

Jürgen Jasperneite
Lisa Underberg (Hrsg.)

Kommunikation in der Automation

Beiträge des Jahreskolloquiums KommA 2026, Lemgo

OPEN ACCESS

inIT | TH OWL | ifak

Editor:
Benedikt Lücke
Institut für industrielle Informationstechnik - inIT | Technische Hochschule Ostwestfalen-Lippe

Impressum

17. Jahreskolloquium
Kommunikation in der Automation
(KommA 2026)

10. November 2026 • Lemgo

OPEN-Book

ISBN: 978-3-9818463-7-9
DOI: https://doi.org/10.25644/3yhf-pk66

Herausgeber:
Jürgen Jasperneite, Lemgo
Lisa Underberg, Magdeburg

Institut für industrielle Informationstechnik
Technische Hochschule Ostwestfalen-Lippe

Campusallee 6
D-32657 Lemgo
Telefon: +49 5261 702-2400
Internet: www.init-owl.de | www.jk-komma.de

Vorwort

17. Jahreskolloquium Kommunikation in der Automation – KommA 2026



Am 10. November 2026 fand am Institut für industrielle Informationstechnik (inIT) der Technischen Hochschule Ostwestfalen-Lippe das 17. Jahreskolloquium der Reihe KommA – Kommunikation in der Automation statt. Das Kolloquium, das abwechselnd vom inIT in Lemgo und dem ifak in Magdeburg veranstaltet wird, hat sich erneut als bewährtes Forum für Wissenschaft und Industrie zu allen technisch-wissenschaftlichen Fragen rund um die industrielle Kommunikation erwiesen. Unterstützt wurde die Veranstaltung durch die ITG und die Gesellschaft für Informatik.



Im Fokus des Jahreskolloquiums standen moderne Echtzeit-Kommunikationssysteme – vom klassischen Feldbus über Echtzeit-Ethernet bis zur drahtlosen Kommunikation und der Nutzung von IoT-Technologien. Einen besonderen konzeptionellen Schwerpunkt bildete in diesem Jahr der Einsatz Künstlicher Intelligenz im Engineering. Bei der Systemanalyse und dem Entwurf von Kommunikationssystemen widmeten sich die Tagungsbeiträge aktuellen Themen wie Single Pair Ethernet (SPE), Infrastructure as Code (IaC) sowie modellbasierten OPC UA-Ansätzen. Als Kernaspekte vernetzter eingebetteter Echtzeitsysteme behandelte das KommA-Kolloquium zudem die formale Modellierung, Verifikation und Validierung, Interoperabilität, Konformität, Safety und Security sowie die Systemintegration. Neben diesen aktuellen Technologien bildeten die vielfältigen praxisnahen Anwendungsbereiche der industriellen Kommunikation einen zentralen Schwerpunkt.

Alle Beitragskurzfassungen wurden im Vorfeld von mindestens zwei Vertretern des Programmkomitees begutachtet. Der Verzicht auf parallele Sitzungen beförderte auch in dieser Auflage den intensiven Gedankenaustausch. Alle Beiträge (Vortrag, Poster) finden sich als Vollmanuskripte im elektronischen Tagungsband wieder.

Prof. Dr. Jürgen Jasperneite

Institut für industrielle Informationstechnik - inIT
Technische Hochschule Ostwestfalen-Lippe,
Fraunhofer IOSB-INA

Prof. Dr. Lisa Underberg

ifak e.V. - Institut für Automation und Kommunikation e.V.
Otto-von-Guericke-Universität Magdeburg

Komitees

Tagungsleitung:

Prof. Dr. Jürgen Jasperneite

Institut für industrielle Informationstechnik - inIT | Technische Hochschule Ostwestfalen-Lippe, Fraunhofer IOSB-INA

Prof. Dr. Lisa Underberg

ifak e.V. - Institut für Automation und Kommunikation e.V. | Otto-von-Guericke-Universität Magdeburg

Programmkomitee:

Stefan Bollmeyer

ABB Asea Brown Boveri Ltd

Holger Büttner

Beckhoff Automation GmbH & Co. KG

Prof. Dr. Christian Diedrich

Otto-von-Guericke-University

Prof. Dr. Mathias Fischer

Universität Hamburg

Prof. Dr. Mesut Günes

Otto-von-Guericke-University

Prof. Dr. Ulrich Jumar

Otto-von-Guericke-University

Dr. Maren Kobusch

WAGO GmbH & Co. KG

Gunnar Lessmann

PHOENIX CONTACT Electronics GmbH

Dr. Jan Stefan Michels

Weidmüller GmbH & Co. KG

Markus Rentschler

Murrelektronik GmbH

Prof. Dr. Thilo Sauter

Technische Universität Wien

Dr. Sebastian Schriegel

Fraunhofer IOSB-INA

Prof. Dr. René Simon

Hochschule Harz

Prof. Dr. Henning Trsek

Institut für industrielle Informationstechnik -inIT

Technische Hochschule Ostwestfalen-Lippe

Dr. Christoph Weiler

Siemens AG

Prof. Dr. Jörg F. Wollert

FH Aachen

Prof. Dr. Martin Wollschlaeger

Techn. Universität Dresden

Organisationskomitee:

Jasmin Zilz

Institut für industrielle Informationstechnik - inIT | Technische Hochschule Ostwestfalen-Lippe

Benedikt Lücke

Institut für industrielle Informationstechnik - inIT | Technische Hochschule Ostwestfalen-Lippe

Yuliaa Blashchuk

Institut für industrielle Informationstechnik - inIT | Technische Hochschule Ostwestfalen-Lippe

Tim Esau

Institut für industrielle Informationstechnik - inIT | Technische Hochschule Ostwestfalen-Lippe

Jannik Peters

Institut für industrielle Informationstechnik - inIT | Technische Hochschule Ostwestfalen-Lippe

Single Pair Ethernet in Industrie und Bahn: Neue Test- und Validierungsansätze für robuste SPE-Systeme

Jan Letmade [1], Roland Heß[1] und Stefan Witte [2]

Abstract: Elektromagnetische Verträglichkeit (EMV) Prüfungen sind ein obligatorischer Bestandteil der Produktentwicklung und Voraussetzung für Zulassungen in vielen Bereichen wie Industrie oder Bahn. Mit Single Pair Ethernet (insbesondere 10BASE-T1L) ist in den letzten Jahren ein neuer Bereich von Netzwerkkomponenten und insbesondere Switches entstanden, die ebenfalls nach den bestehenden Normen zugelassen werden müssen. Für Single Pair Ethernet (SPE) in Kombination mit Power over DataLine (PoDL) entstehen hier aber erhebliche Schwierigkeiten eine normgerechte Prüfung durchzuführen. Bei einigen Medientypen (100BASE-T1 und 1000BASE-T1) ist eine normgerechte Prüfung mit bisherigem kommerziell verfügbarem Testequipment gar nicht möglich. Die Schwierigkeiten sind technologiebedingt, da bei SPE über nur eine 2-Draht-Leitung sowohl Daten mit höheren Frequenzen als auch höhere Leistungen mit bis zu 52 W übertragen werden. In dieser Arbeit stellen wir daher Ansätze und Lösungen vor, die eine normgerechte Prüfung von SPE-Komponenten erlauben.

Keywords: Single Pair Ethernet, EMV Testing, Surge, Burst, Radiated Emissions, Conducted Emissions, Radiated Immunity, Conducted Immunity

1 Einleitung

Single Pair Ethernet, insbesondere 10BASE-T1L hat in den letzten Jahren zunehmend Verbreitung im industriellen und gebäudetechnischen Umfeld erlangt. Die Technologie ermöglicht neue Anwendungen, da die hohe Reichweite von 1000 m und gleichzeitige Energieversorgung der angeschlossenen Komponenten [22] den Verkabelungsaufwand reduziert und eine einheitliche Kommunikationsinfrastruktur vom Sensor bis in die Cloud ermöglicht. Die Technologie kommt zudem im Güterverkehr mit der Digitalen Automatischen Kupplung (DAK) zum Einsatz [Si23].

Mit den Anwendungsfeldern steigen auch die Anforderungen an die Komponenten und die einzuhaltenden Normen, insbesondere an die Robustheit gegen transiente Störgrößen wie elektrische Bursts (IEC61000-4-4) und Surge-Ereignisse (IEC61000-4-5) sowie Störaussendungen. Die Anwender fordern zudem den Einsatz von ungeschirmten Leitungen, da so die übliche Schaltschrankverkabelung über Reihenklemmen möglich wird und bestehende Leitungen weiterverwendet werden können. Für die Komponenten und die EMV-Prüfung stellt dies eine besondere Herausforderung dar, da Surge und Burst direkt auf die Datenleitungen

[1] OWITA GmbH, Lemgo, Germany, jan.letmade@owita.com, https://orcid.org/0009-0007-0579-8268; roland.hess@owita.com

[2] Institut für industrielle Informationstechnik, Technische Hochschule Ostwestfalen-Lippe, Lemgo, Germany, stefan.witte@th-owl.de, https://orcid.org/0009-0009-6243-422X

eingekoppelt werden müssen. Eine besondere Herausforderung stellen auch die Normen für Schienenfahrzeuge dar, da teils deutlich engere Grenzwerte gefordert sind.

100BASE-T1 ist im Automobilbereich verbreitet und wird zunehmend in Fahrzeugen eingesetzt. Auch hier werden wenn möglich ungeschirmte Leitungen verwendet, sodass keine speziellen Kabel und Stecker nötig sind. Die geringe Reichweite von 15 m ist innerhalb eines Fahrzeugs dabei kein Problem. Aufgrund der zulässigen Kanalparameter ist eine normgerechte EMV-Prüfung von 100BASE-T1 Komponenten aber momentan nicht möglich.

2 Single Pair Ethernet

Single Pair Ethernet beschreibt eine Reihe von Physical Layer (PHY) Medientypen im 802.3 Standard, welche alle eine einzelne Zweidrahtleitung (symmetrisches Leiterpaar) als Übertragungsmedium nutzen. Mit Ausnahme von 10BASE-T1S ist die Übertragung Punkt-zu-Punkt und Vollduplex. Es wird Echo-Cancellation genutzt, um das Datensignal der Gegenseite von dem selbst gesendeten Signal zu unterscheiden. Beim Linkaufbau wird zunächst ein Linktraining zur Clock-Synchronisierung und Schätzung des Übertragungskanals durchgeführt. Nach dem Linkaufbau werden immer entweder Daten oder Idle-Symbole gesendet, um den Link aufrecht zu erhalten und damit beide PHYs dauerhaft synchronisiert bleiben.

10BASE-T1S unterscheidet sich von allen anderen SPE Medientypen. Es ist Multidrop-fähig und kann wie ein übliches Bus-System wie CAN oder RS485 eingesetzt werden, bei denen mehrere Teilnehmer eine Leitung teilen. Diese Leitung oder Mixing-Segment kann bis zu 25 m lang sein. 10BASE-T1S unterstützt PHY Level Collision Avoidance (PLCA) zur Verhinderung von Kollisionen auf dem Datenbus, sodass es ähnlich wie CAN verwendet werden kann. MAC-seitig sind aber viele Möglichkeiten zur Verteilung des Medienzugriffs möglich, wie z.B. feste Zeitslots für jeden Teilnehmer oder ein Koordinator, der Slots vergibt. Mit nur zwei Teilnehmern in Punkt-zu-Punkt Verkabelung sind Vollduplex 15 m möglich. Im Gegensatz zu den anderen SPE Medientypen gibt es keinen Verbindungsaufbau und Linktraining.

Abschnitt in 802.3-2022	Medientyp	Reichweite
Clause 96	100BASE-T1	15 m UTP
Clause 97	1000BASE-T1	15 m UTP, 40 m STP
Clause 146	10BASE-T1L	1000 m UTP oder STP
Clause 147	10BASE-T1S	25 m Multidrop, 15 m Point-to-point
Clause 149	2.5GBASE-T1, 5GBASE-T1, 10GBASE-T1	15 m STP

Tab. 1: SPE Medientypen in IEEE 802.3 [22]

Im Gegensatz zu klassischen Feldbus-Systemen wie CAN oder RS485 gibt es mit Ausnahme von 10BASE-T1S eine Vollduplexübertragung, welche jedoch einen Linkaufbau und dauerhafte Synchronisation der PHYs erfordert. Zudem sind die Datenraten und die vom Medium erforderliche Bandbreite zum Teil deutlich höher.

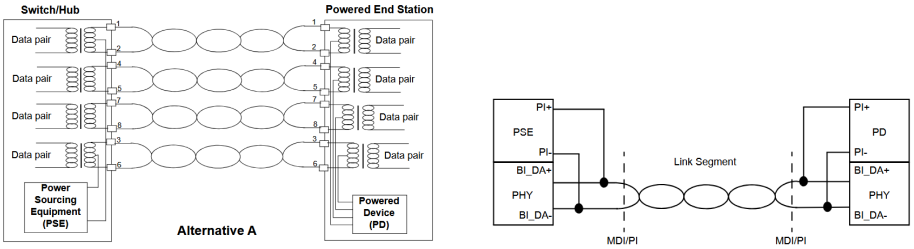


Abb. 1: Vergleich von PoE und PoDL. Links: Schaltbild PoE, Rechts: Schaltbild PoDL

Bei allen SPE Medientypen ist optional die Übertragung von Energie über dieselbe Zweidrahtleitung möglich (PoDL). Damit kann, wie bei Power over Ethernet (PoE), ein Switch die angeschlossenen Geräte versorgen. Anders als bei PoE wird die Energie jedoch im Gegentakt über das Adernpaar anstatt Gleichtakt über zwei Adernpaare übertragen.

3 Herausforderungen bei EMV-Prüfungen für SPE

Je nach Produkt und Umfeld müssen verschiedene Normen eingehalten werden, die zum Teil auch unterschiedliche Prüfverfahren erfordern. In den meisten Fällen sind aber die Normen EN 61000-4-2 bis 61000-4-6 und die EN 55032 im Bereich Industrie und die EN 50155 bzw. EN 50121-3-2 für die SPE-Anschlüsse relevant. Die in den Normen enthaltenen Tests werden im Folgenden kurz erläutert und die Herausforderungen für SPE beleuchtet. Der Test zu Electro Static Discharge (ESD) wird nicht erläutert, da es hier für SPE keine Besonderheiten gibt.

3.1 Radiated Emissions (Gestrahlte Störaussendung)

Bei dem Test wird geprüft, ob das Equipment under Test (EUT) elektromagnetische Störungen ausstrahlt, welche Funkdienste oder benachbarte Geräte beeinträchtigen könnte. Die Messung erfolgt üblicherweise in einer Absorberhalle, wo das EUT auf einem Drehtisch platziert wird. Es wird dann mit einer Antenne die Abstrahlung (üblicherweise von 30 MHz bis 1 GHz) in horizontaler und vertikaler Polarisation gemessen, während das EUT gedreht wird. Es sind die in den relevanten Normen definierten Grenzwerte zu unterschreiten. Für ein EUT, wie einen Switch ergibt sich hier die Herausforderung, dass an jedem Port ein Gerät (oft Auxiliary Equipment (AE) genannt) angeschlossen sein muss, welches kommuniziert und die vom Switch bereitgestellte Leistung konsumiert, da ansonsten die Messung wenig

aussagekräftig ist. Das AE muss mit in der Messkammer verbleiben, da das Herausführen von so vielen Leitungen ohne das Einbringen von Störungen in die Messkammer, in den meisten Fällen nicht möglich ist. Es ist daher ein AE nötig, welches selbst alle Grenzwerte einhält und keine nennenswerten Störaussendungen hinzufügt. Hier sind im Bereich Bahn auch die länderspezifischen Normen einzuhalten (z.B. EMV06 für Deutschland), welche teils deutlich niedrigere Grenzwerte als die üblichen Industrienormen vorschreiben.

3.2 Conducted Emissions (Leitungsgebundene Störaussendung)

Mit diesem Test wird geprüft, ob das EUT über Anschlussleitungen und Kommunikationsleitungen unzulässige Störungen aussendet. Für Netzleitungen nutzt man zur Versorgung des Gerätes eine Netznachbildung (LISN), welche das Gerät versorgt und gleichzeitig die Störspannungen auskoppelt. Für die Kommunikationsleitungen kann ein handelsüblicher Differenztastkopf mit hoher Eingangsimpedanz verwendet werden. Für die Kommunikationsleitungen ist hier wieder eine Gegenstelle erforderlich, die die bereitgestellte Leistung aufnimmt und gleichzeitig selbst keine Störaussendungen verursacht.

3.3 Radiated Immunity (Gestrahlte Störfestigkeit)

Dieser Test prüft, ob ein Gerät in Gegenwart von elektromagnetischen Feldern, wie sie von Mobiltelefonen, Funkgeräten, WLAN oder Radarsystemen erzeugt werden, störungsfrei arbeitet. Die Prüfung findet in einer Absorberhalle statt, in der zuvor ein homogenes Feld kalibriert wurde. Der Prüfling wird an dieser Position aufgestellt und über eine Sendeantenne mit einem Feld von typischerweise 3 oder 10 V/m im Frequenzbereich von 80 MHz bis 1 GHz und darüber bis 6 GHz beaufschlagt. Die Prüfung wird von allen vier Seiten mit horizontaler und vertikaler Polarisation durchgeführt. Für diesen Test ist ebenfalls eine Gegenstelle erforderlich, welche die vom EUT gelieferte Leistung aufnehmen kann und selbst die nötige Störfestigkeit einhält.

3.4 Conducted Immunity (Leitungsgeführte Störfestigkeit)

Bei diesem Test werden hochfrequente Störungen simuliert, die von Sendern wie Rundfunk- oder Mobilfunkanlagen in Kabel eingekoppelt werden und leitungsgebunden ins Gerät gelangen, was insbesondere bei langen Leitungen im Bereich von 150 kHz bis 80 MHz der dominierende Kopplungspfad ist. Der Test muss für die Datenleitungen nach den jeweiligen Normen des Kommunikationssystems durchgeführt werden, was für SPE die IEC 62132-4 ist. Bei SPE wird hierbei vor allem die Common-Mode-Rejection des PHYs, womit die Ergebnisse auch sehr abhängig vom im geprüften Gerät verbauten PHY sind.

3.5 Burst (Schnelle transiente Störgrößen)

Der Burst-Test nach IEC 61000-4-4 simuliert die Störungen, die beim Schalten induktiver Lasten, durch Relaiskontaktprellen oder beim Unterbrechen von Stromkreisen entstehen und sich als Pakete sehr schneller Impulse mit Anstiegszeiten von 5 ns ausbreiten. Diese hochfrequenten Transienten koppeln kapazitiv in benachbarte Leitungen ein und verursachen typischerweise Fehlfunktionen wie Kommunikationsabbrüche oder Paketfehler, weshalb der Test für alle Geräte in industrieller oder gebäudetechnischer Umgebung unverzichtbar ist. Der Prüfling wird auf einer Massefläche aufgebaut und die Störgröße bei Signalleitungen über eine kapazitive Koppelzange eingekoppelt. Der Generator erzeugt Bursts mit Wiederholfrequenzen von 5 oder 100 kHz, einer Burstdauer von 15 ms und einer Periode von 300 ms bei Prüfspannungen von mehreren kV. Auch hier sind insbesondere die Normen für Bahn herausfordernd, da 2 kV mit Kriterium A (keine Bitfehler größer als im Kommunikationsstandard vorgeschrieben) und 4 kV mit Kriterium B (kein Linkabbruch) gefordert sind. Der Test beansprucht bei SPE das EUT und AE gleichermaßen, da beide Seiten ständig zusammenarbeiten müssen, um den Link aufrecht zu erhalten. Um ein EUT korrekt zu testen muss also sichergestellt werden, dass das AE in jedem Fall die Prüfbedingungen einhält.

3.6 Surge (Stoßspannung)

Die Surge-Prüfung nach IEC 61000-4-5 ist notwendig, weil Blitzeinschläge in der Nähe oder Schalthandlungen großer Lasten im Versorgungsnetz energiereiche Überspannungsimpulse erzeugen, die Isolationen durchschlagen, Bauteile thermisch zerstören und zu Totalausfällen führen können. Im Unterschied zu Burst handelt es sich hier um einzelne, vergleichsweise langsame, aber sehr energiereiche Impulse. Der Prüfling wird über ein Coupling Decoupling Network (CDN) an den Hybridgenerator angeschlossen, der einen Impuls mit 1,2/50 µs Spannungsform bei Leerlauf und 8/20 µs Stromform bei Kurzschluss erzeugt. Bei der Durchführung werden je Polarität und Kopplungsart mindestens fünf Impulse mit Prüfpegeln von beispielsweise 1 kV zwischen den Leitern und 2 kV gegen Erde eingeprägt. Auf den SPE-Anschlüssen darf das Gerät weder zerstört werden, noch dürfen Linkabbrüche auftreten. Die Herausforderung ist hier die Prüfung bei ungeschirmten Leitungen. Hier muss der Surge direkt auf die Datenleitungen (und nicht nur auf den Schirm) eingeprägt werden, wofür ein CDN benötigt wird. Da die meisten CDNs für langsamere Kommunikationstechnologien wie CAN oder RS485 entwickelt wurden, erfüllen diese nicht für alle Medientypen die Anforderungen an den Kommunikationskanal (Einfügedämpfung, Nebensprechen, etc.).

Aus den einzelnen Prüfungen ergeben sich im wesentlichen zwei Punkte, welche gelöst werden müssen. Zum einen muss für ein zu prüfendes Gerät für jeden SPE-Port eine Gegenstelle mit SPE-Datenkommunikation und PoDL-Energieübertragung angeschlossen sein, um sinnvolle Prüfergebnisse zu erhalten. Zum anderen ist für die Surge-Prüfung mit ungeschirmten Datenleitungen ein CDN nötig, was die Anforderungen an den Kommunikationskanal einhält.

4 Entwicklung einer SPE-Gegenstelle

Da die SPE-Gegenstelle ein gewöhnliches SPE Powered Device (PD) ist, ist eine technische Umsetzung grundsätzlich einfach möglich und wurde auch von uns vorgenommen. Da ein Switch in der Regel viele Ports hat, müssen an der Gegenstelle ebenso viele Ports vorhanden sein. Daher wurde die Gegenstelle als Einschubkarte konzipiert, sodass 12 Stück in einem Gehäuse verbaut werden können. Jede Karte hat zwei Ports, womit bis zu 24 Ports in einem Gehäuse möglich sind. Beide Ports sind datenseitig mit einem Switch miteinander verbunden, sodass eine kontinuierliche Datenübertragung möglich ist, was für einige Tests benötigt wird. Zusätzlich können im Switch Zähler für die Anzahl an übertragenen Paketen sowie die Anzahl der Paketen mit CRC-Fehler überwacht werden, sodass auch die Zahl der Paketfehler während der Tests überwacht werden kann.

Die Gegenstelle muss auch die vom Switch für den Port bereitgestellte Leistung aufnehmen. Dies wurde über einstellbare elektronische Lasten realisiert, wobei eine pro Port vorhanden ist. Eine Einschubkarte mit 6HP Breite kann dabei PoDL Klasse 10-14 abführen. Für Klasse 15 sind größere Kühlkörper notwendig, womit das Modul 12HP breit wird. Um Störaussendungen zu vermeiden, wurden bei der Spannungsversorgung nur Linearregler verwendet. Durch entsprechendes PCB-Design wurden die Störaussendungen von den PHYs und dem MCU zur Steuerung minimiert.

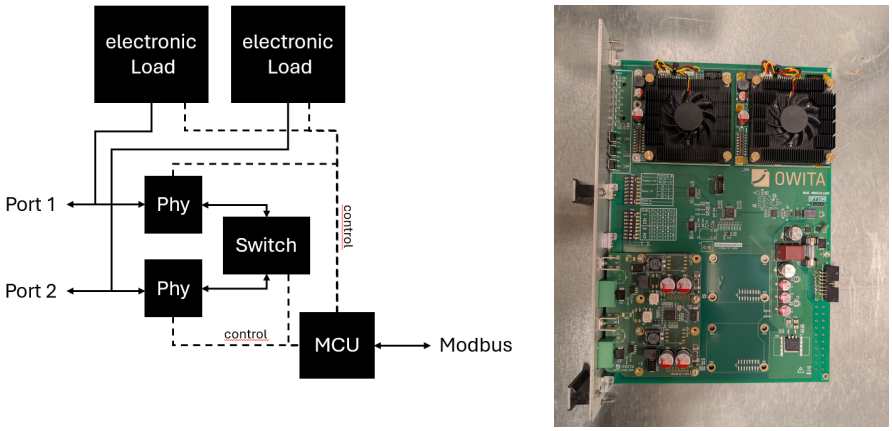


Abb. 2: Implementierung der 10BASE-T1L Gegenstelle. Links: Blockschaltbild, Rechts: Foto der fertigen Einschubkarte

Die Herausforderung ergibt sich aber darin, dass die Gegenstelle das Testergebnis nicht verfälschen darf. So müssen die leitungsgebundenen- und gestrahlten Störaussendungen deutlich unter den Grenzwerten liegen. Da beide Seiten für das Aufrechterhalten des SPE-Links verantwortlich sind, müssen auch alle Kriterien für Burst eingehalten werden.

Die Gegenstelle wurde daher im Labor gegenüber der in Abschnitt 3 beschriebenen Tests und

Normen geprüft. Dabei zeigte sich bei der Messung zu gestrahlten Störaussendungen, dass das Gerät mindestens 15 dBµV/m unter dem EN55032 Grenzwert liegt. Bei der gestrahlten Störfestigkeit zeigten sich auch bei 20 V/m keine Fehler. Bei den leitungsgebundenen Störaussendungen lagen alle Emissionen mindestens 10 dBµV/m unter den EN55032 Grenzwerten. Bei Surge wurden 2 kV bei shield/earth und line/earth mit Kriterium B (kein Linkverlust) eingehalten und bei Burst 2 kV line/earth mit geschirmten Kabeln mit Kriterium A (kein Bitfehler) und 4 kV mit Kriterium B sowohl mit geschirmten als auch mit ungeschirmten Kabeln erreicht. Die Gegenstelle kann daher zur Prüfung anderer 10BASE-T1L SPE-Geräte eingesetzt werden. Die Gegenstelle ist durch Austausch der PHYs und des Frontends grundsätzlich auch auf andere SPE-Standards erweiterbar. Hierfür müsste aber eine erneute, vollständige EMV-Prüfung erfolgen.

5 Coupling Decoupling Network (CDN)

In der IEC61000-4-5 sind für die Prüfung von ungeschirmten, symmetrischen Datenleitungen (Kapitel 7.4 Test setup for tests applied to unshielded symmetrical interconnections communication lines) zwei CDNs definiert [05]. Das CDN für ungeschirmte, symmetrische Datenleitungen (siehe Abbildung 4 links) leitet den Surge über Gasableiter oder TVS-Dioden auf die Datenleitungen. Die Widerstände R_{m2} werden entsprechend der Anzahl der Datenleitungen und des durchzuführenden Tests konfiguriert. Damit besteht die Möglichkeit, dass der Surge unsymmetrisch eingekoppelt wird, da Gasableiter je nach Alterung zu unterschiedlichen Zeiten zünden. Die unterschiedlichen DC-Spannungen bei PoDL zwischen den beiden Datenleitungen können ebenfalls zu einer unsymmetrischen Einkopplung führen. Das Equipment Under Test (EUT) ist direkt mit den Datenleitungen verbunden auf die der Surge eingeprägt wird. Das Auxiliary Equipment (AE) wird über das Decoupling Network und das Protection Equipment geschützt. Im Decoupling Network ist im Wesentlichen eine 20 mH Gleichtaktdrossel verbaut, welche den im Gleichtakt eingeprägten Surge stark abdämpft, gleichzeitig die Datenkommunikation, welche im Gegentakt stattfindet, nicht beeinflusst. Diese 20 mH-Drossel hat aufgrund der hohen Ströme und der Sättigung des Kernmaterials eine sehr große Bauform, was eine hohe Bandbreite auch bei bifilarer Wicklung sehr schwierig macht. Zudem müssen die Magnetfelder in Kernmaterial und Luftspalt gut abgeschirmt werden, um Crosstalk bei mehreren Datenleitungen zu vermeiden.

In Abbildung 3 sind die Einfügedämpfung zweier CDNs dieser Bauart (CDN-UTP8 ED3 und DCD-5 st-4-1) mit der geforderten Einfügedämpfung verschiedener Medientypen aufgetragen. Die Einfügedämpfung der CDNs ist anhand der im Datenblatt garantierten Bandbreite als idealen Tiefpassfilter geschätzt. Die tatsächliche Kurve wird abweichen, ein grundsätzlicher Vergleich ist jedoch möglich. Das CDN ist besser als die für den Medientyp notwendige Einfügedämpfung, wenn die Kurve des CDNs über der des Medientyps liegt. Das DCD-5 st-4-1, welches in Laboren am weitesten verbreitet zu sein scheint, kann danach nur für 10BASE-T1L verwendet werden. Dies deckt sich auch mit unseren empirischen Versuchen, wobei nur ein Kanal des DCD-5 verwendet werden kann. Werden mehrere

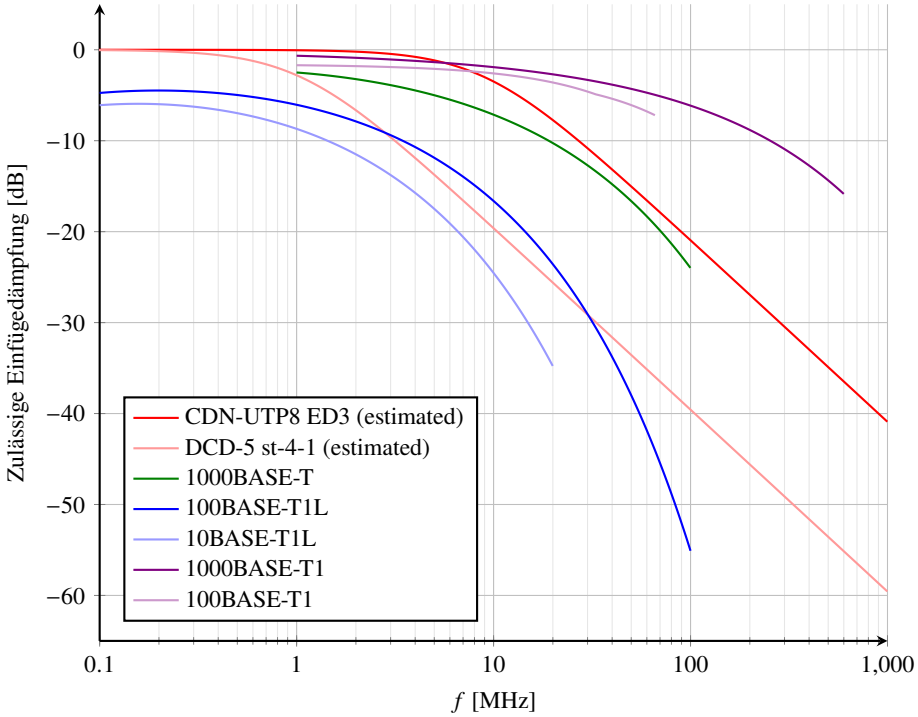


Abb. 3: Vergleich der Einfügedämpfung zweier CDNs (CDN-UTP8 ED3 und DCD-5 st-4-1) mit der geforderten Einfügedämpfung verschiedener Medientypen. [22]

Kanäle verwendet, kann es vorkommen, dass nur einer der beiden SPE-Verbindungen einen Link aufbaut. Möglicherweise ist der Crosstalk zwischen den Kanälen zu groß. Der Hersteller kann dies möglicherweise durch bauliche Maßnahmen beheben.

Das CDN-UTP8 ED3 scheint weniger verbreitet zu sein und eignet sich nach Abbildung 3 auch für 100BASE-T1L und 1000BASE-T. Der Hersteller wirbt auch mit 1000BASE-T, was sich mit der Darstellung deckt. Für 100BASE-T1 und 1000BASE-T1 ist das CDN jedoch nicht geeignet.

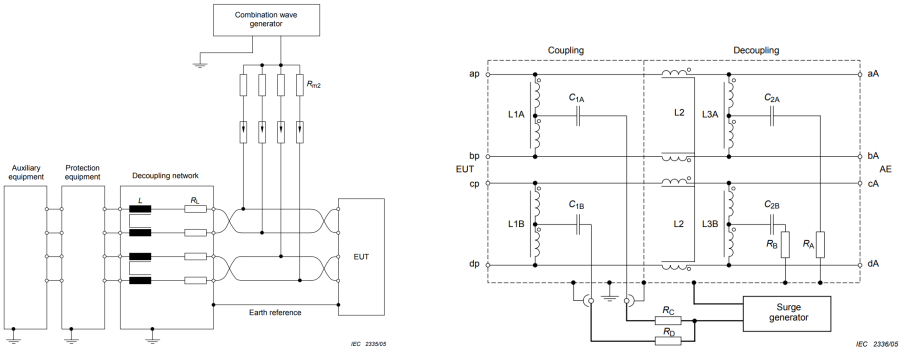


Abb. 4: Vergleich von zwei CDNs aus der IEC61000-4-5 [05]. Links: CDN für ungeschirmte, symmetrische Datenleitungen, Rechts: CDN für ungeschirmte, symmetrische high-speed Datenleitungen

Daher kann für high-speed Datenleitungen wie Ethernet noch ein weiteres CDN verwendet werden (siehe Abbildung 4 rechts). Dieses koppelt den Surge über C1 und die Gegentaktdrossel L1 ein, was zu einer deutlich symmetrischen Einkopplung führt. Die Gleichtaktdrossel L2 verhindert auch hier, dass der Surge zum AE übertragen wird. Jedoch ist keine Induktivität spezifiziert, sodass diese auch kleiner ausfallen kann. Die wesentliche Absicherung des AEs erfolgt über L3, welche Gleichtaktstörungen über C2 an die Erde ableiten. Für SPE allein kann das CDN gut verwendet werden, in Kombination mit PoDL jedoch nicht, da L1 und L3 für DC-Spannung zwischen den differentiellen Leitungen einen Kurzschluss darstellen.

5.1 Coupling Decoupling Network (CDN) für high-speed Datenleitungen

Um die Anforderungen von SPE und PoDL zu erfüllen schlagen wir ein CDN nach Abbildung 5 vor. Dieses trennt die HF-Datensignale von den niederfrequenten Surge, sodass diese separat behandelt werden können. Die Gegentaktdrosseln L1 und L3 verhindern, dass die HF-Datensignale in den oberen Teil der Schaltung gelangen. Diese werden stattdessen über L4 übertragen. Das Hochpassverhalten kann über die Cs eingestellt werden. L4 muss nur die nötige Isolation zwischen EUT und AE ermöglichen und kann daher auf eine hohe Bandbreite optimiert werden.

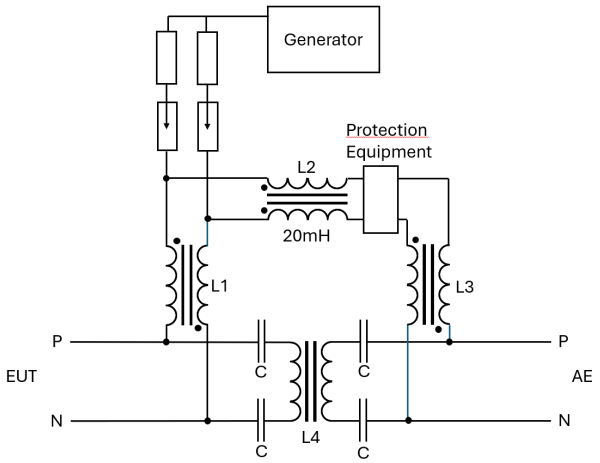


Abb. 5: Vorschlag eines CDNs, welches die Anforderungen von SPE und PoDL erfüllt.

Die Einkopplung des Surge erfolgt im oberen Teil über L1. Da L1 eine Gegentaktdrossel ist, kommt diese durch den Gleichtakt-Surge nicht in die Sättigung und kann daher klein ausgelegt werden. Die niederfrequente Entkopplung des Surge erfolgt wie bei dem anderen CDN auch über eine 20 mH Gleichtaktdrossel. Hinter dieser können zusätzliche Schutzelemente verbaut werden, um das AE besser zu schützen.

Das CDN ermöglicht die vollständige Einkopplung des Surge in die Datenleitung bei gleichzeitigem Schutz des AEs und Datenübertragung mit hoher Bandbreite. Ein solches CDN ist jedoch nach aktueller Norm nicht zulässig. Damit dieses für EMV-Prüfungen verwendet werden kann, müsste dieses in die IEC61000-4-5 mit aufgenommen werden.

6 Fazit

In dieser Arbeit wurden die Herausforderungen bei der normgerechten EMV-Prüfung von SPE-Geräten erörtert. Es zeigen sich zwei wesentliche Probleme: zum einen muss zur Prüfung eines Switches eine spezialisierte Gegenstelle gebaut werden, zum anderen ist für die Surge-Prüfung bei ungeschirmten Leitungen für einige SPE-Medientypen kein CDN vorhanden. In dieser Arbeit wurde eine vollständig geprüfte Gegenstelle für 10BASE-T1L Prüfungen vorgestellt. Zudem wurde eine CDN-Schaltung vorgestellt, die die Anforderungen für SPE und PoDL erfüllt und somit auch ein Testen von 100BASE-T1 und 1000BASE-T1 erlaubt.

Literaturverzeichnis

[05] Electromagnetic compatibility (EMC) - Part 4-5: Testing and measurement techniques - Surge immunity test. IEC, 2005.

[22] IEEE standard for Ethernet 2022. IEEE, 2022.

[Si23] Simon Funke, DB: FP5-TRANS4M-R Deliverable 2.3: Reference Freight System Architecture. 2023.

Infrastructure as Code in the Connected Factory

How Git and CI/CD Find Their Way Into the Cabinet

Dimitri Block [1], Albrecht Lohöfener[2]

Abstract: The increasing convergence of Operational Technology (OT) and Information Technology (IT) in industrial automation impacts the network configuration of machine networks. This article highlights the potential of Infrastructure as Code (IaC) as an IT concept for commissioning and maintaining network-enabled components using YANG models. With manageable effort, operability is simplified, availability and IT security are enhanced, and standards-compliant manufacturer differentiation as well as customer-specific adaptations are enabled.

Keywords: Infrastructure as Code, YANG models, CI/CD, Network Digital Twin, SDN

1 Introduktion and Problem Statement

In industrial automation, Operational Technology (OT) and Information Technology (IT) have been clearly separated traditionally. OT systems are characterized by segmented machine networks (MN) that ensure independence from higher-level IT networks e.g. production site networks or IT enterprise networks as well as from other MNs as illustrated in Figure 1. Due to segmentation production processes within MNs are independent. Hence, downtime, failures or anomaly behavior do not harm production processes in different MNs and increase their availability. Additionally, each segmentation combined with security measures such as firewalls adds up to defense in depth and therefore enhances the cyber resilience against malicious attacks from outside as well as from within another MN. Hence, sensitive process data is protected.

[1] Weidmüller Interface GmbH & Co. KG, Technology Development, Klingenbergstraße 26, 32758 Detmold, Dimitri.Block@weidmueller.com, https://orcid.org/0000-0002-1980-7344
[2] Weidmüller Interface GmbH & Co. KG, BU Industrial Ethernet, Klingenbergstraße 26, 32758 Detmold, Albrecht.Lohoefener@weidmueller.com

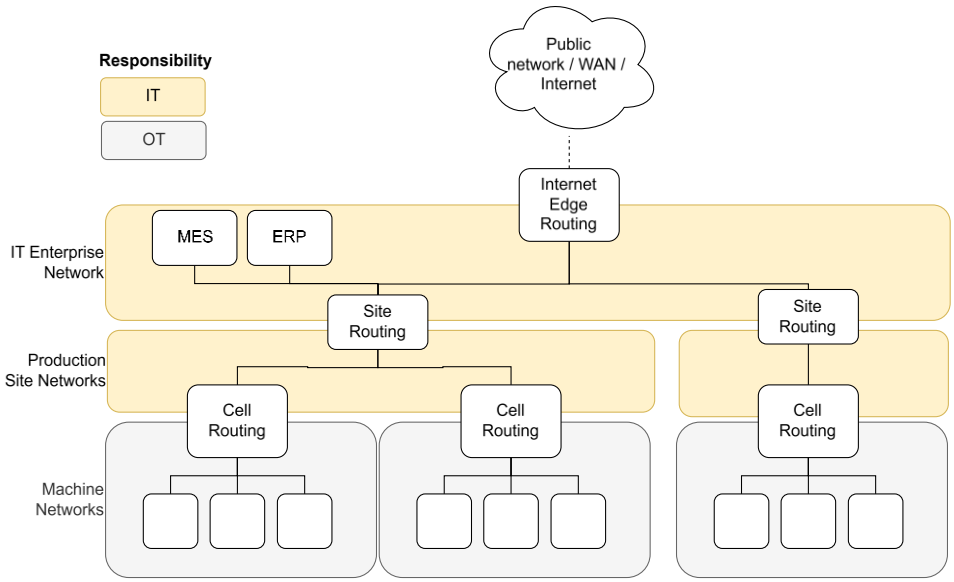


Figure 1: Typical logical network topology in manufacturing production sites

As illustrated in Figure 1 a typical logical network topology of an MN consists of a cell routing functionality (CRF), several Ethernet switches, network hosts such as controllers and field devices. The CRF segments between the MN and the higher-level IT networks. It handles IP routing, Network Address Translation (NAT), firewall functionalities, as well as other segmenting and protection network functions.

Typically, production facilities and machine builders commonly implement CRFs using dedicated physical components, usually termed “routers”. This approach supports portability because configurations can be transferred between routers. Consequently, machines can be relocated between production sites without requiring major reconfiguration and can also be delivered to customers containing the network configuration for the MN. For the same reasons, Ethernet switching and other MN networking features are also typically implemented as dedicated physical components. Consequently, managed networking components such as routers and managed Ethernet Switches use vendor-specific proprietary user interfaces.

In contrast, segmentation in IT networks tends not to be implemented by separate components, but rather logically within fewer, yet more powerful, components. These different technical approaches lead to a fragmentation of the market. Providers of IT network components use powerful, standardized, and scalable solutions, partner ecosystems, while OT networks are characterized by specialized physical components and vendors.

According to a VDMA market study of German machine builders in 2025 [VDMA25], nearly 80% of machine builders plan to connect their MNs to the higher-level IT networks in the future. The convergence of IT and OT networks in general was identified as one of the most important industry trends. Hence, it requires new methods for network configuration of MNs. Current industrial automation practices largely rely on manual, vendor-specific configuration mechanisms that limit scalability, reusability, automation and are also error-prone. It is therefore necessary to investigate how established IT concepts can be applied to MNs without compromising availability, segmentation, or backward compatibility.

This contribution evaluates the network configuration of machine networks (MN) in manufacturing production sites, and therefore the commissioning and maintenance of network-enabled components using “Infrastructure as Code” (IaC) as an established IT concept [MH18]. With IaC, infrastructure resources are managed not manually but through declarative or script-based definitions in the form of code. The concept of IaC originates from the continuous development and deployment processes of software-driven services (Continuous Integration and Continuous Delivery or Continuous Deployment, CI/CD for short).

This paper makes three contributions:

1. It derives requirements for applying IaC to network configuration in segmented MNs.
2. It proposes a YANG-native IaC approach combining version control, reusable configuration templates, inventory data, credential management, and model-driven management protocols.
3. It demonstrates the technical feasibility of the approach in a container-based VLAN bridging proof of concept using Ansible, NETCONF, and emulated Ethernet switches.

2 State of the Art: Network Configuration in Connected Factories

Conventional OT network configuration mechanisms are limited by vendor-specific interfaces and insufficient transactional capabilities. Traditionally, network configuration relies on proprietary command-line connectivity and instructions. While initially network configuration required serial connection directly to the network device. With the rise of Ethernet, migration to the command-line-based unencrypted protocol Telnet (RFC 854) was adopted which enabled remote network configuration via a network itself. Later it was mainly replaced by the popular protocol SSH, specifically the second version SSH-2 (e.g. RFC 4254), which encrypts the command-line-based connection and therefore protects confidential network configuration details and credentials.

Parallel, a dedicated protocol for network configuration was standardized as Simple Network Management Protocol (SNMP) with the RFC 1157. With later versions encryption was introduced. Technically, SNMP operates using a request-response model, allowing to query devices for status and configuration data. Additionally, it supports trap messages for event-based notifications. Thereby, SNMP is based on a data model, called Management Information Base (MIB) introduced in RFC 1155 which enables hierarchical representation of network configuration elements so called managed objects.

Although SNMP supports network configuration operations in principle, it is predominantly applied to monitoring and status retrieval. Moreover, its limited data-modelling capabilities and lack of atomic transactions, structured error handling, and rollback mechanisms restrict its suitability for automated network configuration [FB25].

To address the limitations of SNMP, a new protocol, called NETCONF which abbreviates "network configuration" was introduced in the RFC 6241. Technically, NETCONF uses XML-encoded remote procedure calls and is commonly transported over SSH, providing mechanisms for configuration, retrieval, and management of network devices. Its development marked a shift toward transactional, model-driven configuration management by supporting atomic operations and robust error handling, making it well-suited for modern infrastructure as code workflows.

NETCONF data models are based on YANG. It is a data modeling language developed for defining the structure of configuration and state data manipulated network management protocols introduced in RFC 6020. It addresses the limitations of SNMP, which were less suited for model-driven, programmable, and automated network configuration and monitoring. Its adoption enables consistent and interoperable management across multi-vendor environments. Today, YANG is widely recognized as the foundational modeling language for modern network configuration and orchestration.

Based on the success, multiple alternative transport protocols for YANG-based network configuration were introduced, such as RESTCONF (RFC 8040) which extends the wide-spread concept of Representational State Transfer (REST) interfaces to network configuration, CORECONF which leverages the Constrained Application Protocol (CoAP) for communication and employs CBOR for compact binary serialization for network configuration of resource-constrained devices, and gRPC Network Management Interface (gNMI) which uses gRPC and supports configuration retrieval, modification, and streaming telemetry.

The recently published industry standard IEC/IEEE 60802:2026 [IEC26] uses YANG-based models combined with the transport protocols NETCONF or RESTCONF. Technically, it uses a centralized model-based IT network configuration approach for TSN-based IT and OT networks. Hence, it is an international industry standard for MNs, which is based mainly on standard YANG model network configuration.

3 Proposed YANG-Native IaC Network Configuration Approach

We propose IaC for network configuration approach of MNs based on YANG models as illustrated in Figure 2. YANG defines the schema of configuration and operational data. Depending on the management protocol, corresponding data instances can be encoded in XML, JSON, or CBOR. Hence, templates based on the XML-based YANG models from the corresponding SDOs can be stored as source code files for the declarative state of the network configuration in the version control system (VCS). Together with the inventory files containing the IaC-capable MN device connectivity settings and the configuration files declaring how the YANG model templates should be applied, and the credentials vault containing sensitive information like passwords, the IaC framework enforces the desired state at the target devices using appropriate YANG-capable transport portocols like but not limited to NETCONF, RESTCONF, CORECONF and gNMI.

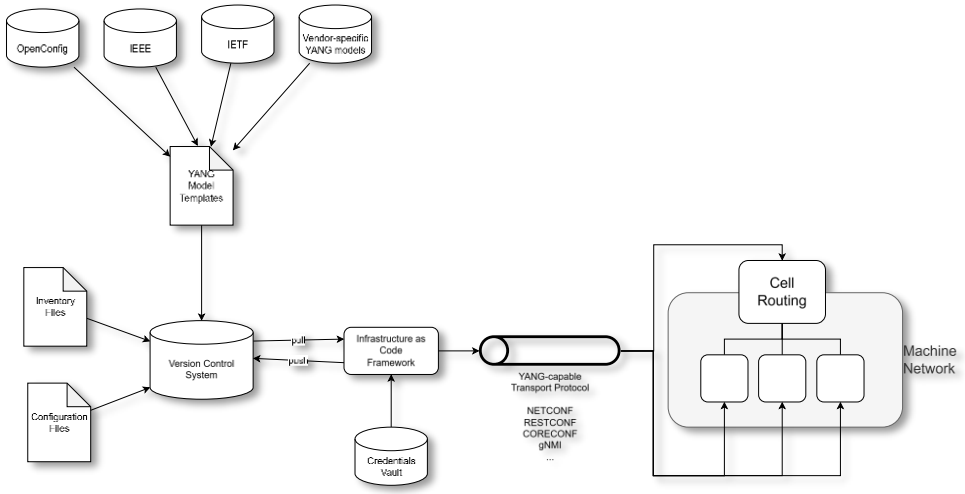


Figure 2: Proposed YANG-native IaC approach for the network configuration of machine networks

Numerous standardized YANG models from standard developing organizations (SDOs) such as IETF, IEEE, and OpenConfig already cover the typical range of industrial functionalities required for Ethernet switching and CRF. An overview of some standard YANG modules is listed in Table 1.

Table 1: Examples of YANG model standards for networking functionalities

YANG STANDARD	RFC / STANDARD REFERENCE	NETWORKING FEATURES
ietf-interfaces	RFC 8343	Network interface management (enable/disable interfaces, interface configuration)
ietf-ip	RFC 8344	IPv4 and IPv6 addressing, multiple IP addresses per interface
ietf-routing	RFC 8349	Static routing and routing configuration
openconfig-interfaces	OpenConfig Interfaces Model	Vendor-neutral operational and configuration model for network interfaces
ietf-access-control-list	RFC 8519	Access control lists (ACLs), packet filtering, firewall policies
openconfig-acl	OpenConfig ACL Model	Vendor-neutral ACL and packet filtering configuration
openconfig-vlan	OpenConfig VLAN Model	VLAN configuration and management
ietf-snmp	RFC 7407	SNMP configuration and management
openconfig-qos	OpenConfig QoS Model	Traffic prioritization, classification, queuing, QoS policies
openconfig-access-points	OpenConfig Access Points Model	WLAN access point mode configuration

The standard NETCONF (RFC 6241) specified transactional configuration management through the optional features of candidate datastores, atomic commit operations and confirmed commits. A candidate datastore is a temporary configuration workspace where changes can be prepared, reviewed, and validated before they are applied to the active device configuration. Multiple configuration modifications can be accumulated and committed as a single transaction. An atomic commit applies to a set of configuration changes as one transaction ensuring that the device remains in a consistent state. A confirmed commit temporarily activates a configuration and requires explicit confirmation within a defined timeout period reverts to the previous configuration otherwise protecting against connectivity loss or invalid settings. This feature enabling safe, consistent, and failure-resilient network configuration.

Furthermore, using YANG models within each network-capable devices in a MN, enables to capture and represent its network topology state, which is the base for a Network Digital Twin (NDT) as proposed in [PA22]. NDT supports use cases for central management, desired and current condition comparison, topology documentation as well as simulation of the effect of network topologies changes.

In addition, standardized manufacturer differentiation and customer-specific function extensions are supported. A vendor can implement standard models (e.g. OpenConfig or IETF) and add proprietary functionality via adding new nodes to an existing model (augment), vendor-specific YANG modules or deviations, which declares where the implementation differs from a standard model. The device remains compatible with the standard model while exposing unique vendor capabilities.

Furthermore, IaC enables policy-based security and compliance processes due to software services like OPA or credential vault services as illustrated in Figure 2. Hence, network configurations may be automatically validated against these policies before deployment and credentials are protected against accidental public exposure.

4 Proof of Concept: VLAN Bridge Use Case

In the following example, the proposed YANG-native IaC network configuration approach is validated. The example use case is to brigde two IP subnets with the network hosts h1 and h2 and Ethernet switches s1 and s2 respectively on OSI layer 2 via a different IP subnet with a VLAN bridge between the Ethernet switches s1 and s2. Hence, it joins both to a single virtual IP subnet containing h1 and h2.

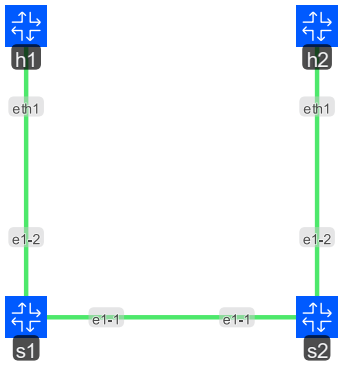


Figure 3: Example use case of Ethernet switch configuration to join two separate IP subnets with VLAN bridge

Therefore, switches s1 and s2 must be reconfigured to:

1. Add a VLAN tagged sub-interface e.g. with the VLAN ID 101 to create VLAN trunk port at the network interface e1-1.
2. Add an untagged sub-interface to create a VLAN access port at the network interface e1-2.
3. Bind the trunk port as well as the access port sub-interfaces to share OSI layer 2 network traffic internally.

The network topology including two hosts and two switches were emulated using the open-source-source software Containerlab using Alpine-based hosts and Nokia SRL Ethernet switches.

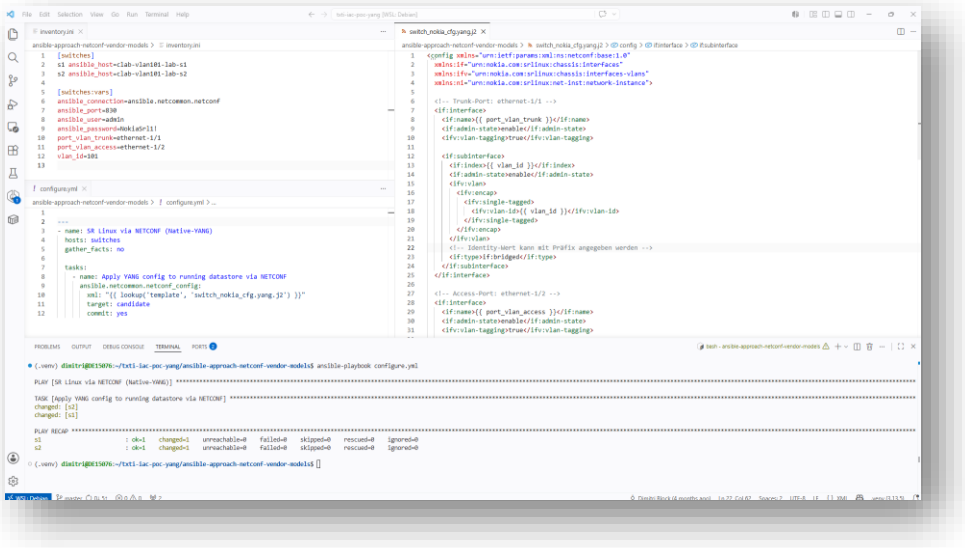


Figure 4: Ethernet switch s1 and s2 are configured via Ansible call (bottom) containing an Ansible inventory file (top left), Ansible playbook as configuration file (left), and a NETCONF/YANG template (right) enabling VLAN-encapsulate communication between two separated IP subnets and therefore connecting hosts h1 and h2 transparently.

The popular open-source IaC framework Ansible was used as illustrated in Figure 4. The inventory file contains the connectivity settings and credentials for the management interfaces of both Ethernet switches. For simplicity reasons no credential vault was used. Additionally, the use case specific VLAN parameters were stored in the inventory file.

The configuration file is called Ansible playbook. It uses the YANG model template, populates it with the actual values, stores the new configuration as candidate, and commits

it finally all at once. The used transport protocol is NETCONF. The YANG model template contains four different YANG models, whereby three are vendor-specific due to the limitation of no standard-based YANG model for the specific Ethernet switch.

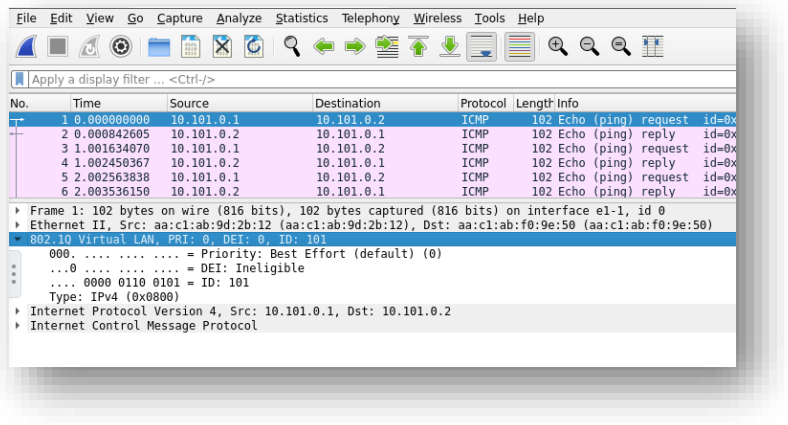


Figure 5: Successful ICMP-based reachability validation of hosts h1 and h2 after Ethernet switch s1 and s2 reconfiguration

After executing the Ansible playbook with the inventory file and YANG model template, both Ethernet switches are reconfigured. Hence, the hosts h1 and h2 can reach each other via ICMP. Figure 5 shows the VLAN tagged network traffic after successful reconfiguration between both Ethernet switches.

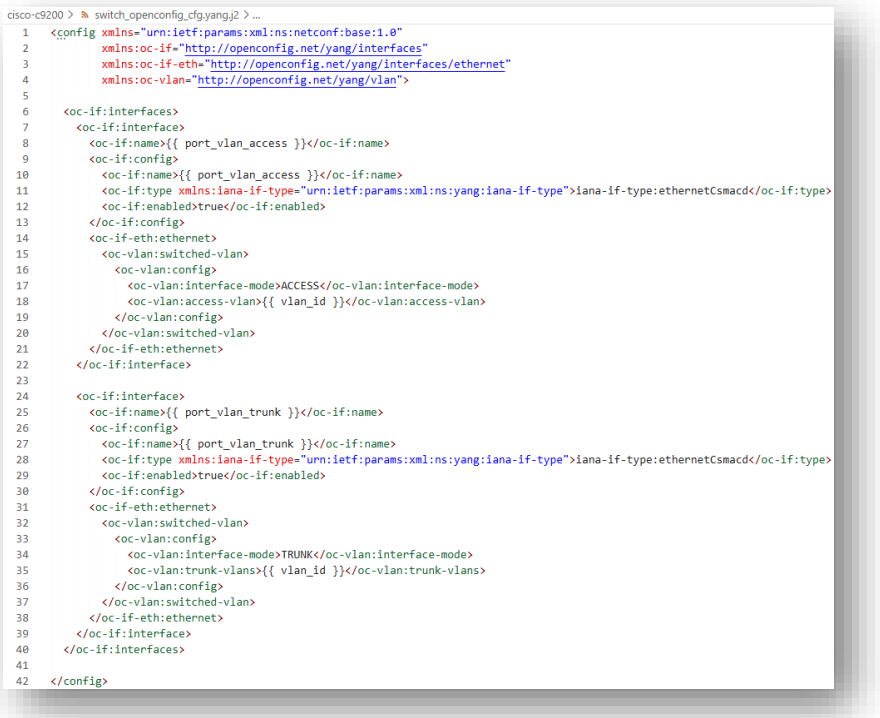


Figure 6: Network configuration template prepared for the VLAN use case on a Cisco Catalyst C9200-48T-E with standard YANG models only

4.1 Network Configuration based on Standard YANG Models

In the proof of concept, the Ethernet switch emulations were limited. While some functionalities were supported, the VLAN use case could not be validated with standard-based YANG models. To investigate whether the use case could be represented without vendor-specific YANG modules, the configuration template was adapted for a Cisco Catalyst C9200-48T-E. It resulted in a successful standard-only YANG model network configuration.

In Figure 6 the used YANG model template is depicted. It contains one IETF YANG model and three OpenConfig YANG models.

5 Conclusion and Outlook

This contribution demonstrates that a YANG-native Infrastructure as Code approach can transfer established configuration-management principles from IT to machine networks (MNs). By combining standardized and vendor-specific YANG models with version control, reusable templates, inventory data, and transactional protocols such as NETCONF, network configurations can be managed consistently and reproducibly across heterogeneous devices. The proof of concept confirms the technical feasibility of this approach:

IaC results in consistent and reproducible configuration of network resources, minimizes potential sources of error through automation, and speeds up the implementation of changes. In connected factory environments, IaC contributes to harmonization by providing scalable management of heterogeneous networks. This approach thus offers a flexible, scalable, and efficient solution for future-ready connected factory environments.

Further work should evaluate the approach with physical industrial network components and additional networking functionalities to identify missing standard YANG models for MNs. Further investigation is also required into automated testing, policy validation, credential management, recovery behavior, and lifecycle integration within industrial CI/CD processes.

References

[VDMA25] M. Rothhöft: Marktstudie Industrielle Kommunikation / Industrie 4.0; Ergebnisse einer Befragung von Unternehmen aus dem Maschinenbau in Deutschland, 2025, VDMA

[O25] Okano, M. T. et al.: Technological Convergence of Operational Technology (OT) and Information Technology (IT) in the Digital Industry, IFIP APMS 2025, Springer.

[FB25] Frank Bär, Dr. Frank Breiter, Alexander Groß: Enhancing TSN Configuration: A Hybrid CORECONF-NETCONF Approach for Constrained Devices, 2025

[PA22] P. Almasan et al.: Network Digital Twin: Context, Enabling Technologies, and Opportunities, in IEEE Communications Magazine, November 2022, doi: 10.1109/MCOM.001.2200012

[IEC26] IEC/IEEE 60802 Time-Sensitive Networking Profile for Industrial Automation, Published Standards, Juni 2026

[MH18] Marc Holness, Ciena: Joint IEEE 802 and ITU-T SG15 Workshop “Building Tomorrow’s Networks”, IEEE 802.1 YANG Data Modeling Overview, January 2018

OPC UA Server Datasheet: A Model to Facilitate Integration of OPC UA Enabled Machines in Engineering and IT tools

Kazeem Oladipupo[1], Sebastian Kaebisch[1] , Matthias Riedl[2], Christian Diedrich[2]

Abstract: Integrating OPC UA-enabled machines into industrial IT/OT systems typically requires access to physical assets at runtime, for machines delivered by machine builder, this usually leads to prolonged commissioning time and integration bottlenecks. To address this, this paper introduces the OPC UA Server Datasheet submodel. The submodel formalizes essential server configuration parameters like NodeSet files, security policies and modes. It also formalizes identification metadata, and runtime endpoints of the asset that allows communication properties to be exchanged ahead of physical delivery. This enables parallel offline engineering, interface simulation, and client pre-configuration. The approach is validated using a laboratory demonstrator with a Siemens TIA Portal Add-In for automated submodels creation and a Node-RED environment for simulated server instantiation and update of runtime endpoints. The workflow is intended to eliminate on-site integration friction, reduces downtime, and accelerates product time to market.

Keywords: Digital Twin, Asset Administration Shell, OPC UA, System Integration, Offline Engineering.

1 Introduction

For decades, information modelling has been a cornerstone of automation activities such as simulation, design, engineering, and operation. In the current era of industrial automation, information modelling of automation systems particularly in the context of digital twins enables businesses to accelerate the integration of machines into engineering and IT tools, ultimately contributing to increased efficiency and revenue. To further enhance integration speed, this paper explores an approach to offline engineering for machines equipped with OPC UA interfaces. It introduces the OPC UA Server Datasheet Submodel [ID26c], designed for the Asset Administration Shell (AAS) [ID26a][ID26b], which enables application-specific configuration to be prepared before the machine is delivered by machine builder and OPC UA server is instantiated.

This paper presents the OPC UA Server Datasheet Submodel, an information model developed to hold the configuration and identification data of an OPC UA server associated with an asset represented by the AAS. The model enables consistent exchange of server information throughout the asset lifecycle, from engineering and commissioning to operation. The paper presents several use cases demonstrating its applicability.

[1]Siemens AG, Technology, Friedrich-Ludwig-Bauer-Straße 3, 85748, Garching bei München, {kazeem.oladipupo & sebastian.kaebisch}@siemens.com.
[2]Institut für Automation und Kommunikation e.V., Werner-Heisenberg-Straße 1, 39106 Magdeburg, {christian.diedrich & matthias.riedl}@ifak.eu

The Submodel incorporates configuration data such as NodeSet files, supported profiles, security policies, and server capabilities required by OPC UA client applications for seamless integration of OPC UA enabled asset into IT infrastructures. In addition, it provides standardized mechanisms for server identification using the *BuildInfo* object defined in IEC 62541 Part 5[IE20c] and for runtime access using endpoint information based on the *EndpointDescription* object defined in IEC 62541 Part 4[IE20b]

Finally, a proof-of-concept on the creation and implementation of the Submodel is presented spanning across multiple stakeholders (machine builder and operator). This paper also demonstrates OPC UA Server Submodel combination with other AAS submodels, such as the Asset Interfaces Description (AID) [ID26d], thereby enabling a coherent and scalable approach to system integration.

2 Background

2.1 Asset Administration Shell (AAS)

The concept of Digital Twin (DT) is significantly growing in the manufacturing domain. This technology (digital twin) represents a physical asset within a digital framework throughout its lifecycle. To ensure a smooth integration between various industrial assets, tools, and data, standardized digital twin technologies are required.

The AAS, also known as the “Verwaltungsschale” in German, is a standardized digital representation of an asset, the cornerstone for the interoperability of Industrie 4.0 components organized in Industrie 4.0 systems. An asset can be a simple component, a machine, a plant, or a product at any level of equipment hierarchy [ID26a]. The information within the AAS is structured into submodels such as Digital Nameplate, Technical Data, and Handover Documentation. These submodel templates handle a specific aspect of an asset, and they are developed and published by the Industrial Digital Twin Association (IDTA).

The structure of the AAS allows different stakeholders to utilize information throughout the asset lifecycle while maintaining semantic consistency. Through standardized data models and APIs [ID26b], the AAS provides interoperability among manufacturers, system integrators, operators, and end users. As a result, the AAS has emerged as one of the foundational technologies for exchanging data and implementing DTs in industrial environments.

2.2 Open Platform Communication Unified Architecture (OPC UA)

OPC UA as a standard defined in IEC 62541[IE25], is a service-oriented, platform-independent middleware that enables a secure and reliable data exchange in industrial automation. It is designed to scale the integration of factory data from field devices to enterprise and cloud system. The core of the standard is its information model, it defines an object-oriented information model capable of exposing machine data raw tags values

as a browsable address space of nodes (Variables, Objects, Methods, References) with types and semantics that gives more meaning to a value and how it is related to other values.

The standard supports various types of communication architecture like client-server and publish-subscribe. The client-server architecture offers a request/response and a subscription-based data access while publish-subscribe architecture offers a broker-based data access which is suitable for fast eventing applications. In addition to communication is a transport method, OPC UA provides various kind of transport methods such as TCP for low-latency communication, HTTP or WebSocket for IT systems Integration and recent transport method, OPC UA over TSN to realize real-time and deterministic potential for field devices in OT networks.

OPC UA offers a secured by design architecture by incorporating various authentication technologies at the protocol level [IE20a]. By configuring the appropriate security mode and policies, OPC UA offers end-to-end encryption and decryption of messages, offers X.509 certificate for authentication and provides an option to restrict user access using the *allowanonymoususer* flag.

A layer above the core architecture of OPC UA is a very crucial specification that helps industry domains to semantically structure their data. This specification is companion specification. The companion specification helps several domains like OPC for AAS, Robotics, PackML, OPC UA for Machinery e.t.c standardized their data into a widely recognized model.

2.3 The Necessity of Offline Engineering

In conventional automation projects, system integration follows a strictly sequential engineering lifecycle. While mechanical, electrical, and control engineering can partially overlap during design, the software integration of shop-floor machines into higher-level information technology (IT) and operational technology (OT) systems such as SCADA, MES, and cloud edge gateways typically begins only after the physical machine has been assembled, wired, and commissioned on the factory floor.

This sequential paradigm introduces substantial risks and inefficiencies:

- ***Extended Commissioning Times:*** Software engineers and system integrators can only browse the OPC UA address space, identify node identifiers, verify data types, and validate security trust relationships once the physical PLC and its integrated OPC UA server are powered and reachable on the operational network.
- ***Production Downtime and Cost Overruns***: Interface mismatches, missing data points, or security certificate incompatibilities discovered during on-site commissioning require iterative debugging cycles, leading to delayed ramp-ups and production downtime.
- ***Lack of Early Validation:*** Developing downstream client applications (e.g., analytical dashboards, recipe management, predictive maintenance

algorithms) concurrently with machine manufacturing is hindered when no concrete representation of the communication interface exists.

Offline engineering and virtual commissioning mitigate these bottlenecks by decoupling software configuration from physical hardware availability. By utilizing standardized digital descriptions of machine interfaces, system integrators can instantiate simulated communication servers, pre-configure client connection parameters, and validate communication pipelines in advance.

3 OPC UA Server DataSheet Submodel

The overall structure of the OPC UA server datasheet submodel is defined in Figure 1. It consists of three core parts: the Configuration part, which describes how the server is configured; the Identification part, which helps identify a specific server; and the Endpoint Definition part, which describes the endpoints of the server and/or the endpoint of the discovery server at runtime. Each of these core parts are described in the following subsection.

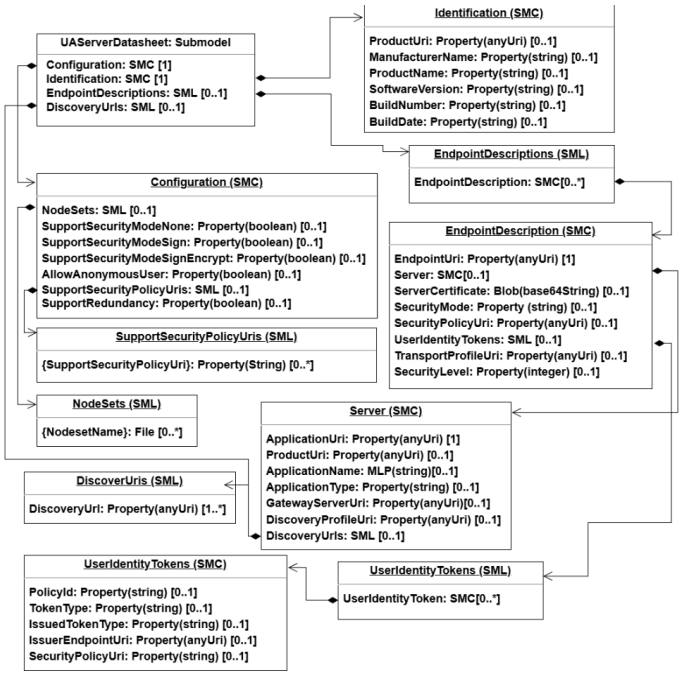


Figure 1: Overall structure of OPC UA Server Datasheet Submodel.

3.1 Configuration Core Part

The configuration core part of the submodel contains information that can be used to configure a simulated server that mirrors the physical machine interface. It can also be used by client application to understand what a server is capable of doing. Within the *Configuration SubmodelElementCollection (SMC)*, is a term like *NodeSets* that contains the NodeSet files used to configure the server. These NodeSet files can also be used by a client application to understand the structure of data that the server provides. There are also flag terms *AllowAnonymousUser* that indicates if anonymous users are granted access or not. *SupportSecurityModeNone*, *SupportSecurityModeSign*, *SupportSecurityMode-SignEncrypt* that indicates the security mode(s) supported by the server. *SupportSecurityPolicyUris* provides a list of supported security policies by OPC UA. It is designed in a way to be capable of handling future policy additions or removals. Lastly, *SupportRedundancy* property that indicates that the server is capable of handling redundancy features defined by OPC UA.

3.2 Identification Core Part

The Identification core part of the submodel contains information that can be used to identify a specific OPC UA server in a network. It uses OPC UA based information which is the *BuildInfo* object information. The BuildInfo object information which is represented by *Identification SMC* contains terms like *ProductUri* that defines a uri identifier for the server, *ManufacturerName* that defines the name of the server manufacturer, *ProductName* that defines the name of the server, *SoftwareVersion* that defines the version of the software at the time it was built as a product. *BuildDate* that defines the date that the server application was built and *BuildNumbe*r that indicates the number associated with the server application build process.

3.3 Endpoint Definition Core Part

The endpoint definition core part of the submodel contains information that is provided at runtime and can be used by client application to properly connect to the server. It uses OPC UA based information which is *EndpointDescription* object information. The EndpointDescription object information provides a list of server endpoints based on number of security modes and security policies supported by the server at runtime. In each of these endpoints, information like *EndpointUri* that provides the endpoint address of the server. *SecurityMode* indicates the type of security mode associated with the endpoint. *SecurityPolicyUri* defines the security policy associated with the endpoint. *TransportProfileUri* defines the type of transport profile that can be used by the endpoint for example, for uabinary, transport Uri http://opcfoundation.org/UA-Profile/Transport/uatcp-uasc-uabinary will be used. *SecurityLevel* provides a numeric value that indicates how secure the endpoint is. Zero indicates weak security and anything greater than zero indicates better security. *ServerCertificate* provides a BLOB container for the server's certificate byte string. In addition to these properties are also *Server* object information as *Server SMC* and list of identity tokens that the server will accept as *UserIdentityToken SubmodelElementList (SML)*. The *Server SMC* provides a description of the server that the endpoint belongs to. In this description are properties like

ApplicationUri that uniquely identifies the server application instance. *ProductUri* that defines a uri identifier for the server. *ApplicationName* that provides a localized name for the server application. *ApplicationType* that defines the type of application, this can be any of Server, Client, ClientAndServer, or DiscoveryServer. Because the focus of this submodel is on OPC UA server, only Server or ClientAndServer is valid for this property. *GatewayServerUri* that defines the URI that identifies the gateway server linked with the discovery URIs. A list of *DiscoveryUris* that is used to query for endpoint information of registered server applications. *DiscoveryProfileUri* defines the type of discovery profile supported by the endpoint.

Because the endpoint information cannot be known unless the server is running, the information is always populated at the point when the server is starting.

4 OPC UA Server Datasheet Creation and Usage

As depicted in Figure 2, the creation and usage of OPC UA Server Datasheet submodel spans between two actors in a machine's lifecycle, namely, the machine builder (vendor) and the operator. The creation of the submodel starts at the machine builder and its data continue to grow as the submodel transits between lifecycle phases. For this paper, a simple demo shown in Figure 3 represents the machine. In the subsections that follows, the creation and usage of the OPC UA Server Datasheet submodel is detailed and discussed.

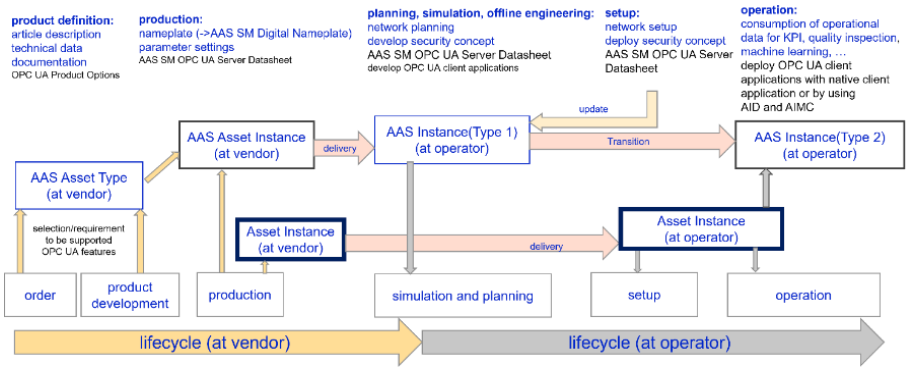


Figure 2: Usage of OPC UA server datasheet across asset lifecycle.

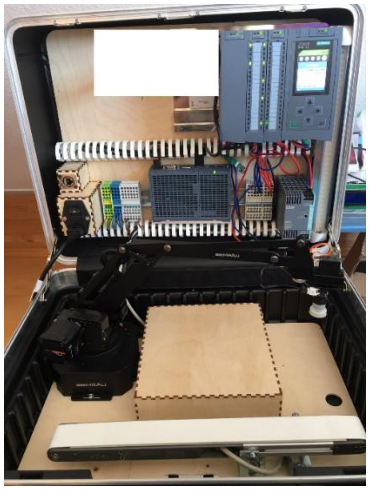


Figure 3: Pick and Place Demo Box.

4.1 OPC UA Server Datasheet Creation at Machine Builder

At the machine builder, during development of the machine, various engineering tasks are executed to get to the finished product which is the machine. Amongst these tasks is the automation engineering where the abstraction of components that will enable the process logic of the machine are collected and connected together. It is also this point that all datapoints used in the machine logic are defined as tags. If the controller orchestrating the machine processes has OPC UA interface, then the engineering of that interface must be done and all the datapoints of the machine can be accessed through the interface. Achieving this engineering task always require the use of an engineering tool.

For this paper, TIA Portal [SI25] is used as the engineering tool to execute the automation engineering task. The major reason for using TIA Portal is because it provides a way for a custom automation task to be executed within the engineering tool. The custom automation task is achieved using the TIA Openness [SI26] and TIA Add-Ins [SI25]. These two services open a way to create a custom feature that generates OPC UA Server datasheet submodel from the OPC UA interface information in the project opened in TIA Portal. Figure 4 shows the feasible flow of how the submodel is generated. In this flow, it is assumed that the Machine AAS has been

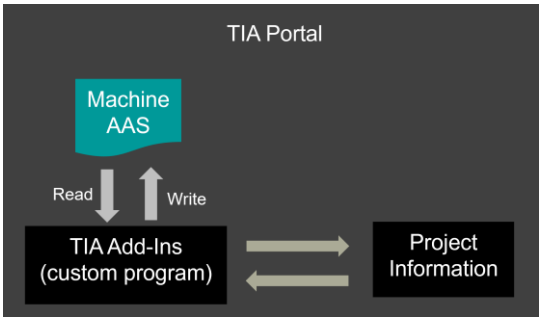


Figure 4: Submodel Creation Flow in TIA Portal.

initially created from the design engineering task (e.g using CAD or EPLAN) that precedes the automation engineering task.

The TIA Add-In custom program mainly parses and reads the machine AAS, reads the OPC UA interface information of the controller controlling the machine, converts the information to OPC UA server datasheet submodel and writes it to the machine AAS.

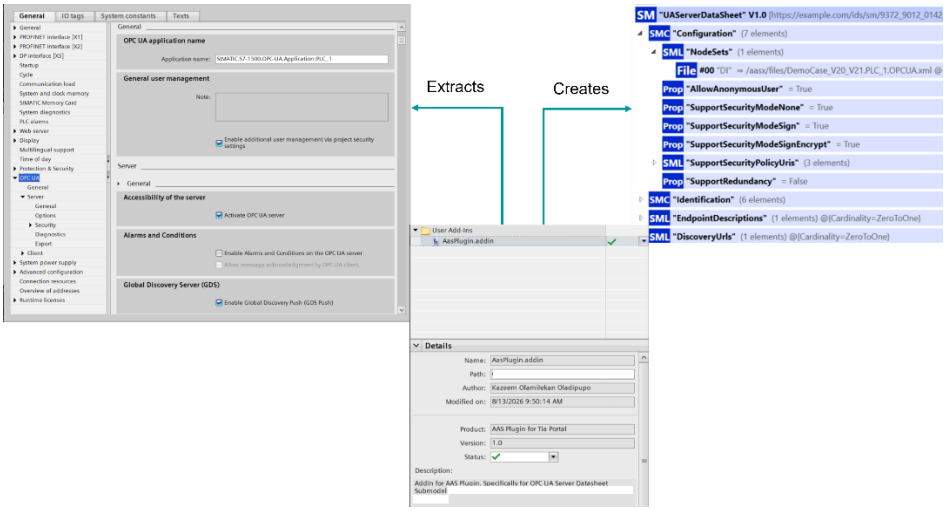


Figure 5: OPC UA Information and Submodel Output Snippet

Figure 5 shows a snippet of information that is read from the TIA project and the output that it provides as a submodel.

4.2 OPC UA Server Datasheet Usage at Operator

As seen in Figure 2, the AAS of the machine is expected to be delivered before the physical machine and it is the time between these deliveries that is used for offline engineering and offline integration activities that then reduces the integration of the physical machine when it is delivered. To realize this offline integration, a simulation of the machine's interface

is needed, the OPC UA server datasheet submodel in the AAS delivered by the machine builder contains information that can be used for simulating the machine's OPC UA interface.

For this paper, an open-source tool called Node-RED [OP26] is used. In node-red, an extension program is added to the OPC UA node package [MK26] to be able to read the machine's AAS and use the OPC UA server datasheet submodel to configure itself. Figure 6 shows the OPC UA simulated server node based on AAS in the UI.

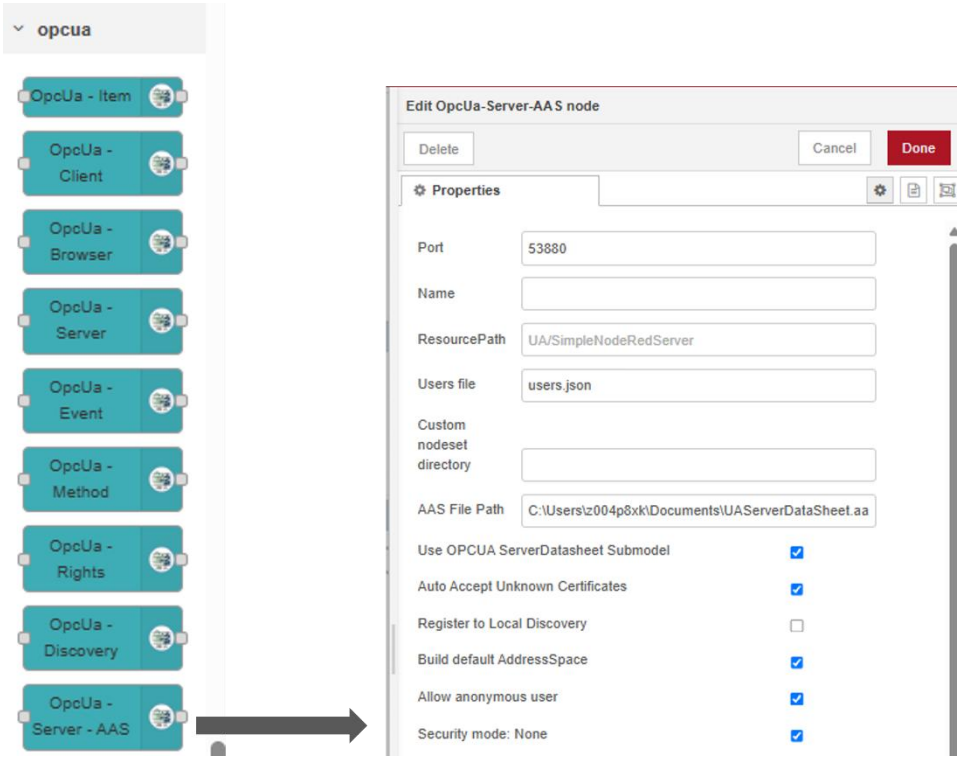


Figure 6: Node-Red UI for OPC UA Server configuration based on OPC UA Server Datasheet.

In addition to the configuration of the simulated server with the AAS, the extension program also updates the identification core part and the endpoint definition core part of the OPC UA server datasheet submodel and writes the output to the same AAS where the submodel is extracted from.

4.3 Synergy Between OPC UA Server Datasheet and AID

Since the simulation server in section 4.2 is presented as a machine interface, from AAS standpoint, client applications can use the information in the OPC UA server datasheet submodel to connect to the server. However, OPC UA server datasheet only provides client application with information needed to connect and communicate with the server.

With the combination of AID submodel, information about datapoints that resides in the server can also be provided to the client application. This additional submodel helps both the operator and the client application focus on the main parameters needed for the operation of the machine instead of devoting time on browsing through the server address spaces for datapoints.

The creation of the AID submodel starts at the machine builder side in section 4.1 since all the datapoints that exist in the machine is known by the machine builder's engineering tool. Figure 7 shows the custom program snippet in Node-RED that is used to also update the AID submodel with endpoint information of the simulated server and other additional nodes that have been added during configuration.

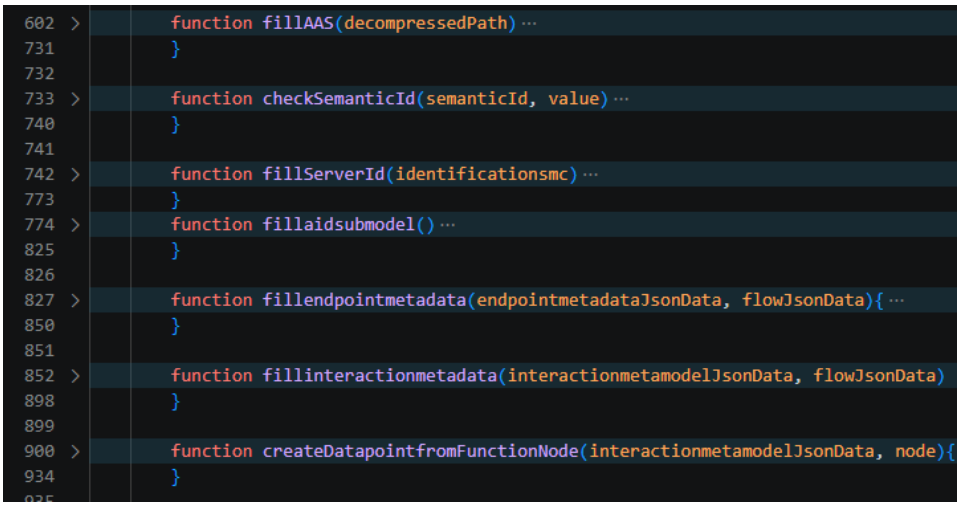


Figure 7: AID Creation Code Snippet.

5 Discussion and Recommendation

The development of OPC UA server datasheet submodel provides a potential for machine operators to save engineering and integration time when installing a new machine into their system. This time saving effort helps the operator cut production downtime and improves time to market for products. While the concept of offline engineering has been around for a long time, having a way to orchestrate this concept by using a model that can be transferred from machine builder to machine operator ensures data integrity within both parties.

During the development of OPC UA server datasheet submodel version 1.0, many questions about what the submodel will contain were raised extensively because OPC UA specification has so many parts and features. At the end, it was finalized that configuration parameters, identification parameters and endpoint parameters should be the major information to be provided. While this information forms the basis for version 1.0, the submodel can be extended with appropriate information if any use-case arises.

Acknowledgement

The development of the submodel was a joint activity between IDTA and OPC Foundation.

Bibliography

[ID26a] Industrial Digital Twin Association: Specification of the Asset Administration Shell; Part 1 - Metamodel, July 2026, Available: https://industrialdigitaltwin.org/en/wp-content/uploads/sites/2/2026/07/IDTA-01001-3-2_AAS-Specification_Part1_Metamodel.pdf.

[ID26b] Industrial Digital Twin Association Specification of the Asset Administration Shell; Part 2 – Application Programming Interfaces, July 2026, Available: https://industrialdigitaltwin.org/en/wp-content/uploads/sites/2/2026/07/IDTA-01002-3-2_SpecificationAssetAdministrationShell_Part2_API.pdf.

[ID26c] Industrial Digital Twin Association: Submodel Template: OPC UA Server Datasheet, May 2026, Available: https://github.com/admin-shell-io/submodel-templates/blob/main/published/OPC%20UA%20Server%20Datasheet/1/1/IDTA%20022009-1-0_Submodel_OPC%20UA%20Server%20Data%20Sheet.pdf

[ID26d] Industrial Digital Twin Association: Submodel Template: Asset Interfaces Description, April 2026, Available: https://github.com/admin-shell-io/submodel-templates/blob/main/published/Asset%20Interfaces%20Description/1/2/IDTA%2002017-1-1_Submodel_Asset%20Interfaces%20Description.pdf

[IE20a] IEC 62541-2: OPC Unified Architecture - Security Model, Edition 3.0; OPC Foundation; International Electrotechnical Commission. November, 2020.

[IE20b] IEC 62541-4: OPC Unified Architecture - Services, Edition 3.0; OPC Foundation; International Electrotechnical Commission. November, 2020.

[IE20c] IEC 62541-5: OPC Unified Architecture - Information Model, Edition 3.0; OPC Foundation; International Electrotechnical Commission. July, 2020.

[IE25] IEC 62541-1: OPC Unified Architecture - Overview and Concepts, Edition 1.0; OPC Foundation; International Electrotechnical Commission. December, 2025.

[MK26] Mika Karaila, Klaus Landsdorf; Node-Red-Contrib-Opcua; Version 0.2.355; September 2026; Available: https://flows.nodered.org/node/node-red-contrib-opcua

[OP26] OpenJs Foundation; Node-Red; Version 5.0.7; September 2026; Available: https://nodered.org/

[SI25] Siemens, Introduction to TIA Portal, Version 21, November 2025, Available: https://docs.tia.siemens.cloud/r/en-us/v21/introduction-to-the-tia-portal

[SI26] Siemens, TIA Portal Openness: API for automation of engineering workflow, Version 21, March 2026, Available: https://docs.tia.siemens.cloud/r/en-us/v21/tia-portal-openness-api-for-automation-of-engineering-workflows

Smarte Produktion in der CIIT FactorySandbox

Marc Riedlinger [1], David Hülsmeier-Reineke [2], Benedikt Ohm[3], Magnus Redeker[1] und Anja Moldehn[4]

Abstract: Um die Anlagenverfügbarkeit in der Produktion zu steigern, benötigen Betreiber über den gesamten Lebenszyklus einer Maschine umfassende Transparenz über den aktuellen Betriebszustand und die verbauten Komponenten. Um dies zu erreichen, werden konkrete Anwendungsfälle in der Multi-Vendor-Testumgebung *CIIT FactorySandbox* umgesetzt. Zentrales Element ist dabei die Asset Administration Shell (AAS), die standardisierte, interoperable und digitale Repräsentation der Assets der realen Welt. Drei umgesetzte Use Cases werden hier detailliert vorgestellt: i) Handover Documentation mit digitalen Typenschildern, ii) Telemetriedaten-basierte Echtzeitüberwachung und -steuerung sowie iii) automatisierte Software-Updates zur automatisierten, freigabegesteuerten Handhabung von Software-Updates.

Keywords: Asset Administration Shell, Multi-Vendor-Umgebung, Anlagenverfügbarkeit

1 Einleitung

Die Digitalisierung der industriellen Produktion macht technische Systeme zunehmend intelligenter, modularer und somit komplexer. Komponentenhersteller, Maschinenbauer und -betreiber stehen vor der Herausforderung, Transparenz in ein interdisziplinäres Umfeld zu bringen. Bspw. um Informationen zu Produktänderungen automatisiert zu kommunizieren oder die operative Verfügbarkeit und Security von Anlagen sicherzustellen. In der FactorySandbox des Centrum Industrial ITs (CIIT) werden dafür technische Möglichkeiten praxisnah umgesetzt: Anders als ein reiner Demonstrator, der nur die Funktionsfähigkeit zeigt, entspricht die heterogene Multi-Vendor-Umgebung realen Szenarien.

Der Produktionsprozess der CIIT FactorySandbox spiegelt alle wesentlichen Module eines Produktionsprozesses wider: Ein Roboter entnimmt Bauteile, sortiert diese nach individuellen Kundenaufträgen und legt sie auf ein Förderband. Montage- und Demontageprozesse beginnen manuell und werden schrittweise automatisiert. Komponenten von CIIT-Partnern sind in der Anlage verbaut und werden im Gesamtkontext interoperabel getestet.

Telemetriedaten werden von Sensoren und IIoT-Geräten kontinuierlich erfasst und in die Asset Administration Shell (AAS) integriert. Dort erfolgt eine standardisierte Verarbeitung,

1 Fraunhofer IOSB-INA, Campusallee 1, 32657 Lemgo, DE, marc.riedlinger@iosb-ina.fraunhofer.de, https://orcid.org/0009-0004-3908-1749; magnus.redeker@iosb-ina.fraunhofer.de
2 Hilscher Gesellschaft für Systemautomation mbH, Rheinstraße 15, 65795 Hattersheim am Main, DE, dhuelsmeier@hilscher.com, https://orcid.org/0000-0002-4644-0592
3 Stäubli Electrical Connectors AG, Stockbrunnenrain 8, 4123 Allschwil, CH, b.ohm@staubli.com
4 Centrum Industrial IT, Campusallee 6, 32657 Lemgo, DE, a.moldehn@ciit-owl.de

sodass Daten systemübergreifend verfügbar und interpretierbar sind. In Echtzeit werden Muster und Anomalien visuell dargestellt, sodass zielgerichtete Automatisierungsschritte abgeleitet und direkt in den Anlagenbetrieb eingegriffen werden können. Dadurch lassen sich Prozesse in Echtzeit optimieren, Ausfälle vermeiden und die Energieeffizienz steigern. Die zentrale Rolle der AAS ermöglicht konsistente Datenmodelle und interoperable Schnittstellen über Herstellergrenzen hinweg. Im Detail betrachtete Anwendungen:

- Handover Documentation: Standardisierte Bereitstellung technischer Dokumentation über AAS-Teilmodell mit digitalen Typenschildern für automatisierten Zugriff.
- Telemetriedaten-basierte Echtzeitüberwachung und -steuerung: Nutzung von Sensor- und IIoT-Daten über die AAS für die Optimierung industrieller Anlagen.
- Automatisierte Software-Updates: Automatisierte, standardisierte und interoperable Updates mithilfe von Produktänderungsmitteilungen.

Die vorgestellten Anwendungen verdeutlichen, wie digitale Technologien nicht nur Funktionsnachweise liefern, sondern in realitätsnahen Szenarien greifbare Mehrwerte für die Produktion und nachhaltige Wertschöpfung bieten. Das Paper ist wie folgt gegliedert: Abschnitt 2 betrachtet den Stand der Technik, Abschnitt 3 präsentiert die drei konkreten Anwendungen zur Steigerung der Verfügbarkeit der CIIT FactorySandbox und Abschnitt 4 diskutiert die drei Anwendungsfälle hinsichtlich Vorteilen, Limitationen und Ausblick.

2 Stand der Technik

In diesem Kapitel wird der Stand der Technik in Bezug auf die Themenfelder Industrielles Internet der Dinge (englisch: Industrial Internet of Things, IIoT), AAS und Produktänderungsmitteilungen beleuchtet.

2.1 Industrielles Internet der Dinge

Das IIoT bezeichnet die Vernetzung von industriellen Anlagen, Maschinen und Sensoren, um Prozessdaten automatisiert zu erfassen und auszuwerten [1]. Dies ermöglicht unter anderem Zustandsüberwachung, Prozessoptimierung und vorausschauende Wartung.

Eine Plattform zur Verwaltung entsprechender Edge-Geräte und Anwendungen ist beispielsweise netFIELD. Diese wird in der CIIT FactorySandbox genutzt, um etwa den Handover Documentation Anwendungsfall umzusetzen.

2.2 Asset Administration Shell

Die Industrie-4.0-Verwaltungsschale (englisch: Asset Administration Shell, AAS) ist eine standardisierte digitale Repräsentanz eines physischen oder digitalen Assets [10] und

bildet somit die Implementierung des Digitalen Zwillings für die Industrie 4.0 [19]. Mithilfe der AAS können Assets über ihren gesamten Lebenszyklus global eindeutig identifiziert und semantisch beschrieben werden [19]. Die AAS zielt darauf ab, Daten interoperabel austauschbar zu machen und damit herstellerübergreifende Geschäftsmodelle zu ermöglichen, wie z. B. die kollaborative Zustandsüberwachung [20].

Die AAS-Spezifikationen umfassen das Metamodell, die API, die semantische Beschreibung von Merkmalen, Sicherheitsaspekte und das Dateiformat; sie wurden ursprünglich von der Plattform Industrie 4.0 entwickelt und werden heute von der IDTA gepflegt und weiterentwickelt [11]. Auf internationaler Ebene wurde der erste Teil der internationalen Normenreihe *Asset Administration Shell for industrial applications* IEC 63278-1:2023 veröffentlicht und umfasst die Struktur der AAS [16]. Der konkrete domänen- oder anwendungsspezifische Inhalt der AAS wird über Teilmodelle realisiert, wie z. B. das Digitale Typenschild oder Technische Daten [10]. Die Teilmodellvorlagen für die jeweiligen Teilmodelle werden in Arbeitsgruppen unter Leitung der IDTA erarbeitet und von der IDTA veröffentlicht [11]. Die HTTP/REST-API der AAS [13] erlaubt es dem Anwender, Elemente hinzuzufügen, zu lesen, zu schreiben oder zu löschen sowie AAS-Operationen auszuführen [21]. Dafür wird eine Laufzeitumgebung benötigt, die die AAS ausführen kann und die API bereitstellt. Hier haben sich mehrere Open-Source-Lösungen etabliert, u. a. Eclipse BaSyx [4], Eclipse FA^{3}ST [6] oder der Eclipse AASX Server [5].

2.3 Produktänderungsmitteilungen

Änderungen an einem Produkt werden vom Hersteller an die betroffenen Parteien der Lieferkette kommuniziert. Man unterscheidet dabei Produktänderungen, die in einer Produktänderungsmitteilung (englisch: Product Change Notification, PCN) dokumentiert werden, und Produktabkündigungen, die als Produktabkündigungsmitteilung (englisch: Product Discontinuance Notification, PDN) übermittelt werden [22].

Das übergeordnete normative Rahmenwerk für das Obsoleszenzmanagement innerhalb eines Unternehmens bildet IEC 62402:2019, das Grundsätze, Prozesse und Verantwortlichkeiten beschreibt [14]. Ergänzend dazu beschreibt IEC 62890:2020-07 den Lebenszyklus von Systemen und Komponenten und welche Fristen beim Übergang in eine neue Lebenszyklusphase kommuniziert werden müssen [15]. Das VDMA-Einheitsblatt 24903:2017-12 regelt die Struktur und Inhalte des Austauschformats von PCN und PDN [22]. Die Grundlage für das Einheitsblatt ist der von der Component Obsolescence Group Deutschland entwickelte Standard smartPCN, der als VDMA 24903 veröffentlicht wurde [2].

Das IDTA-Teilmodell *Product Change Notifications for industrial product types and items in manufacturing* [12] greift die Entwicklungen der Standards IEC 62402:2019, IEC 62890:2020-07 und VDMA 24903 auf und erstellt ein holistisches Informationsmodell, das eine interoperable Bereitstellung von PCN (inklusive PDN) zwischen Lieferanten und Anwendern von industriellen Produkttypen und Artikeln ermöglicht.

Bisherige Implementierungen eines AAS-basierten Produktänderungsmanagements befinden sich im Prototypen- oder Demonstratorstadium. Liepert et al. stellten im Herbst 2024 eine prototypische Implementierung eines AAS-basierten Produktänderungsmanagements im Siemens Product-Lifecycle-Management Teamcenter vor [17]. Der Prototyp adressiert Änderungen an Komponenten während der Engineering-Phase einer Maschine durch einen Maschinenbauer. Die Open Industry 4.0 Alliance hat Ende 2025 einen Demonstrator für AAS-basierte PCN vorgestellt [18], der vorführt, wie Änderungen eines Herstellers an einem Produkt kundenseitig abonnierbar gemacht werden können.

3 Anwendungen zur Steigerung der Anlagenverfügbarkeit

Aufbauend auf den technischen Grundlagen werden im Folgenden drei konkrete Anwendungen zur Steigerung der Verfügbarkeit der CIIT FactorySandbox vorgestellt, die den praktischen Einsatz der AAS demonstrieren: i) Handover Documentation, ii) Telemetriedatenbasierte Echtzeitüberwachung und -steuerung und iii) automatisierte Software-Updates.

3.1 Handover Documentation

Komponenten verschiedener Hersteller mit eigenen Informationsmodellen kommen in der FactorySandbox zusammen, sodass der Zugriff auf technische Dokumentation während Inbetriebnahme, Betrieb und Instandhaltung mehrere Schnittstellen und Zugänge benötigt. Dieser Anwendungsfall untersucht die AAS-basierte Bereitstellung der Informationen über ein einheitliches User Interface.

Mit dem IDTA-Teilmodell *Handover Documentation* wird ein standardisiertes Austauschformat für die technische Dokumentation eines Assets [9] spezifiziert. Dokumente liegen digital vor und werden in Bezug mit einem (physischen) Asset gesetzt. Dieses Teilmodell ermöglicht die strukturierte Bereitstellung von digitalen Dokumenten wie Betriebsanleitungen, Wartungsunterlagen, Konformitätserklärungen und Prüfzeugnisse. Allerdings liegen diese Informationen entweder verstreut auf AAS-Servern bei den Herstellern oder als lokale Kopie im Firmennetzwerk, die ggf. nicht aktualisiert werden.

Jede Komponente der Demonstrationsanlage besitzt eine AAS als Interoperabilitätsschicht, die von Herstellern auf eigenen Servern bereitgestellt werden. Um diese AAS zu finden, müssen Komponenten einer Anlage identifiziert werden. Anschließend können produktbezogene Informationen einheitlich für Nutzer bereitgestellt werden. Die einheitliche Semantik des Teilmodells reduziert herstellerspezifische Unterschiede auf die Integrationsschicht und den Zugang zu den AAS-Servern.

Um Asset-Informationen zu erhalten, werden herstellereigene Shopfloor-Konnektoren als Integrationsschicht genutzt, mit denen Komponenten identifiziert werden können. Zusätzlich wird der innerhalb des Factory-X-Projekts erstellte *Update Hub* [7] genutzt,

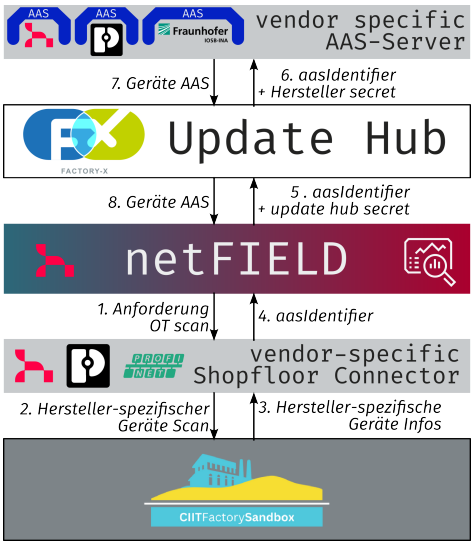


Abb. 1: Bereitstellung von AAS-Informationen durch netFIELD: Ein netFIELD-Gerät (1.) scannt mit (2.) herstellerspezifischen Treibern nach Komponenten der FactorySandbox. Die (3.) Informationen werden in die (4.) Cloud übertragen, um über den (5.) Factory-X *Update Hub* AAS der Komponenten von den (6. + 7.) Hersteller-AAS-Servern abzurufen. Diese Informationen stehen per (8.) Webinterface zur Verfügung.

um einen einheitlichen Zugang zu verschiedenen AAS-Servern bereitzustellen. So kann etwa die netFIELD-Umgebung [8] Asset-Informationen von einer Komponente bis hin zur Anlage über eine einheitliche Benutzerschnittstelle für Benutzer zugänglich machen. Die Dokumente verbleiben dabei fachlich dem jeweiligen Asset zugeordnet und können von Herstellern und Nutzern mit neuen Informationen oder Firmwareversionen versorgt werden. Dabei unterstützt die netFIELD-Umgebung etwa

- Wartungspersonal, um Anleitungen, Schaltpläne oder Ersatzteilinformationen bereitzustellen und so eine manuelle Suche zu reduzieren.
- Installateure, indem Installationshinweise und Parametrierunterlagen strukturiert zur Verfügung stehen.
- Regulatorische Anforderungen (Nachweisführung, Compliance), indem Konformitätserklärungen, Zulassungen und Prüfzeugnisse einzelnen Assets zugeordnet sind.
- den Erhalt von Informationen über den Lebenszyklus wie etwa Versions- und Statusinformationen oder Produktänderungen. Aktualisierte Dokumente oder Firmware können bereitgestellt werden, ohne den Zugriffspfad für Anwender zu verändern.

Limitierungen Dieser Use Case ist abhängig von den Daten der Hersteller und muss daher vom Anlagenbetreiber persistiert werden. Das System ersetzt daher (noch) kei-

ne Dokumentenmanagement- oder Produktdatenmanagementsysteme. Es dient nur als standardisierte Zugriffsschicht, welche Metadaten und Dokumentreferenzen aus herstellerspezifischen Quellen integriert. Die Qualität und Aktualität der bereitgestellten Inhalte verbleiben in der Verantwortung des jeweiligen Herstellers.

Erwarteter Nutzen Die AAS verringert als einheitliche Schnittstelle herstellerspezifische Zugriffshürden und stellt digital Informationen bereit. Diese werden — im Zeitalter von KI — benötigt, um Assets zu vernetzen, Anlagen zu überwachen und resilient gegenüber digitalen Bedrohungen zu werden, oder auch um Engpässe zu erkennen und um bestehende Prozesse zu optimieren. Die AAS ermöglicht somit Transparenz und Nachvollziehbarkeit und bietet einen direkten Mehrwert für Instandhaltungs- und Betriebsprozesse.

3.2 Telemetriedaten-basierte Echtzeitüberwachung und -steuerung

Die Echtzeitüberwachung und -steuerung von industriellen Anlagen ist ein zentraler Baustein zur Steigerung der Anlagenverfügbarkeit und Energieeffizienz. Klassische Ansätze zur Zustandsüberwachung erfordern häufig die Installation zusätzlicher Sensorik und damit verbundene Investitionskosten sowie Installationsaufwände. Die Anwendung in der CIIT FactorySandbox demonstriert, wie vorhandene Infrastruktur – in diesem Fall der Stäubli CombiTac – als Datenquelle für Telemetriedaten aus der Anlage genutzt werden kann, um zwei konkrete Optimierungsziele zu erreichen: Die bedarfsgerechte Kühlung des Schaltschranks und die Erkennung von Schwankungen bei der Druckluftversorgung.

Technische Grundlagen und Datenbereitstellung: Der Stäubli CombiTac ist ein modulares Steckverbindersystem, das in der FactorySandbox an drei strategisch wichtigen Positionen im Schaltschrank verbaut ist. Er dient als Schnittstelle für das erste und das letzte Förderband sowie für die zentrale Druckluftversorgung. Über diese Steckverbinder werden nicht nur Signale und Energie übertragen, sondern auch relevante Prozessdaten erfasst. Die im CombiTac integrierte Sensorik ermöglicht die Messung des Durchleitungsdrucks der Druckluft sowie die Erfassung von Temperaturen direkt am Stecker und an angrenzenden, strategisch wichtigen Positionen im Schaltschrank. Diese Messdaten werden über die AAS der jeweiligen CombiTac-Instanz interoperabel bereitgestellt. Konkret erfolgt die Bereitstellung über das proprietäre AAS-Teilmodell *Operation*, das semantisch beschriebene Betriebsdaten standardisiert zugänglich macht. Durch die Nutzung der AAS wird sichergestellt, dass die CombiTac-Daten mit weiteren interoperablen Kontextdaten automatisch fusioniert werden können und nicht nur für herstellerspezifische Systeme, sondern für beliebige, AAS-kompatible Anwendungen verfügbar sind.

Implementierung der Steuerungslogik: Die Auswertung der Telemetriedaten und die Ableitung von Steuerungsmaßnahmen erfolgt durch einen Control-Agent innerhalb der FactorySandbox. Dieser Agent fungiert als Software-Komponente, die sowohl die interoperablen Daten der AAS abonniert als auch weitere verfügbare Zustandsdaten der Anlage (z. B. Motorströme, Taktzahlen) aggregiert.

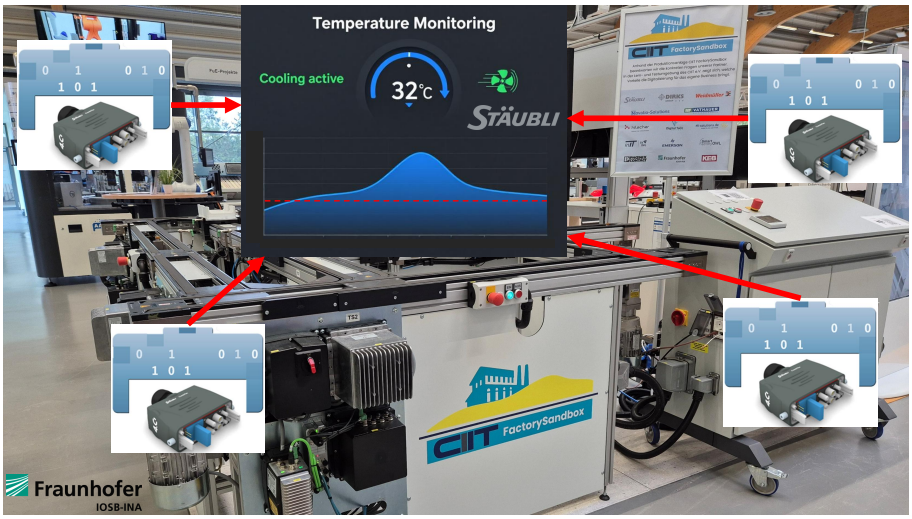


Abb. 2: Echtzeitüberwachung und -steuerung der CIIT FactorySandbox auf Basis der interoperablen Telemetriedaten des Stäubli CombiTac.

Auf Basis dieser Datenbasis implementiert der Control-Agent die Steuerungslogik für die beiden Anwendungsfälle:

- Bedarfsgerechte Kühlung: Der Control-Agent überwacht die vom CombiTac bereitgestellten Temperaturdaten kontinuierlich. Wird ein Grenzwert überschritten, generiert der Agent einen Steuerbefehl zur Aktivierung der Kühlung im Schaltschrank. Sinkt die Temperatur unter den Schwellenwert, wird die Kühlung wieder deaktiviert. Dies ermöglicht eine energieeffiziente Kühlung, die nur bei Bedarf erfolgt.
- Zuverlässige Druckluftversorgung: Parallel überwacht der Agent kontinuierlich den Durchleitungsdruck der Druckluftversorgung der Anlage. Ein unerwarteter Druckabfall außerhalb des normalen Produktionszyklus deutet auf eine Leckage im Druckluftsystem hin. Wird ein kritischer Schwellenwert unterschritten, generiert der Agent eine Warnung, die auf einem Dashboard visualisiert wird, um das Wartungspersonal sofort zu informieren, so dass eine zuverlässige Druckluftversorgung wiederhergestellt werden und die Anlage weiter verfügbar ist.

Alarme können in einem Dashboard visualisiert (Abb. 2) werden und per Direktnachricht an die Produktionsleitung versandt werden. Um Fehlalarme zu vermeiden und die Zuverlässigkeit der Überwachung zu erhöhen, kann der Control-Agent weitere Kontextdaten zu seiner Entscheidungsfindung hinzuziehen. Beispielsweise können die Druckmessdaten des CombiTac mit Zustandsdaten der Anlage korreliert werden: Ein Druckabfall bei geschlossenen Ventilen deutet auf eine Leckage hin, während ein Druckabfall bei geöffneten Ventilen zum normalen Produktionsprozess gehört.

Vorteile und Mehrwert: Die Umsetzung der Telemetriedaten-basierten Echtzeitüberwachung und -steuerung in der FactorySandbox bietet gegenüber konventionellen Ansätzen erhebliche Vorteile:

- Keine Zusatzhardware: Ein wesentlicher Vorteil ist die Nutzung der im CombiTac bereits verbauten Sensorik. Es ist keine nachträgliche Installation separater Sensorik erforderlich, was Kosten und Installationsaufwand minimiert.
- Interoperabilität: Durch die Bereitstellung der Daten im interoperablen AAS-Teilmodell *Operation* können die Daten von beliebigen Clients genutzt werden und mit interoperablen Daten aus weiteren Quellen fusioniert werden, ohne dass herstellerspezifische Treiber oder Schnittstellen notwendig sind.
- Proaktive Wartung: Die frühzeitige Erkennung von Temperaturanstiegen und Druckverlusten ermöglicht ein proaktives Eingreifen zur Vorbeugung von Anlagenstillständen.
- Energieeffizienz: Die bedarfsgerechte Kühlung reduziert den Energieverbrauch der Anlage, da Ventilatoren nicht permanent laufen, sondern nur bei tatsächlichem Bedarf.

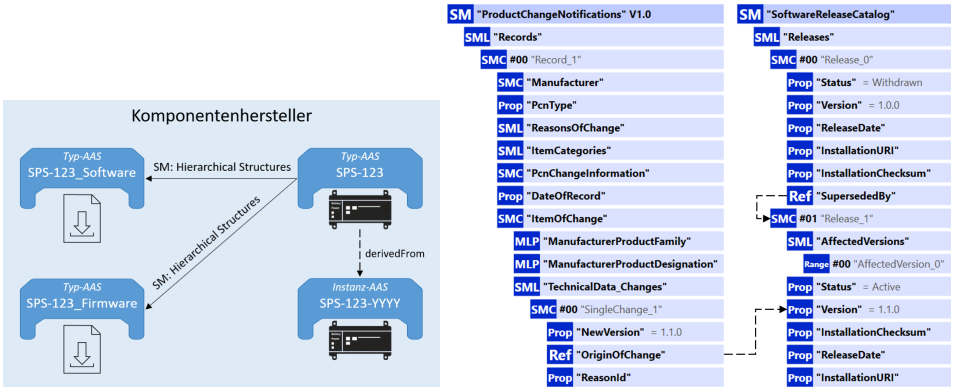
3.3 Automatisierte Software-Updates

Veröffentlicht ein Hersteller von Automatisierungskomponenten ein Software-Update, soll diese Information als PCN an alle betroffenen Nutzer weitergereicht werden. Ein Nutzer kann hier entweder direkt der Betreiber einer Maschine oder aber der Maschinenbauer sein, der die Maschine baut und dann erst an den Betreiber ausliefert. In der FactorySandbox, die als Betreiber auftritt, ist jedoch eine unmittelbare Beziehung zu den Herstellern gegeben, weshalb nur diese Konstellation hier betrachtet wird. Dafür möchte die FactorySandbox auf eine AAS-basierte Lösung setzen. Folgende Anforderungen werden gestellt:

- Ein Update wird automatisch vorgeschlagen.
- Ein Update wird vor der Installation manuell bestätigt oder abgelehnt.
- Nach der Bestätigung soll ein Update automatisch installiert werden.
- Es können zu jeder Zeit auch ältere Updates installiert werden.
- Soweit möglich sollen nur standardisierte IDTA-Teilmodelle verwendet werden.

Auf der Herstellerseite werden die Typen seiner Komponenten (z. B. SPS-123) und alle Instanzen dieser Komponenten eines Betreibers gepflegt (siehe Abb. 3a). Das Betreiben der Instanzen durch den Hersteller hat regulatorische Gründe. Auf diese Weise kann der Hersteller sicherstellen, dass die in der Instanz verfügbaren Informationen, die für eine potenzielle Nachweispflicht notwendig sind, korrekt sind. In der Typ-Komponente sind über das IDTA-Teilmodell *Hierarchical Structures enabling Bills of Material* alle Software-Komponenten, wie Firmwares oder Softwareanwendungen, die auf der Komponente laufen,

referenziert. Dabei wird jede Software als eigenständige AAS modelliert, hat damit einen eigenen Lebenszyklus und kann so im PCN-Teilmodell direkt als Asset referenziert werden.



(a) Herstellerseitiges AAS-Konzept. (b) Notwendige Teilmodelle für ein Software-Update.

Abb. 3: AAS-basierte automatisierte Software-Updates.

Abbildung 3b zeigt einen Ausschnitt des IDTA-Teilmodells für PCN und wie es mit einem proprietären AAS-Teilmodell *Software Release Catalog* (SRC) in Beziehung steht, um die Anforderungen an den hier adressierten Anwendungsfall umsetzen zu können. Beide Teilmodelle sind Teil einer Typ-AAS für Software (z. B. SPS-123_Software). Das PCN-Teilmodell enthält Informationen, die das Software-Update betreffen und an die Betreiber gemeldet werden sollen. Ergänzend dazu enthält das SRC-Teilmodell hinreichende Informationen für den Betreiber, um eine Betroffenheitsanalyse durchführen zu können, und den Download-Link des Updates. *OriginOfChange* referenziert die neue Version (hier 1.1.0) im Release-Katalog des SRC-Teilmodells. Im Vorgängereintrag wird auf die jeweils nächste Version verwiesen mithilfe der Referenz *SupersededBy*. Damit kann durch den Release-Katalog navigiert werden, um für eine Altversion die nächste unterstützte Version zu finden mit dem Status *Active*. In der Liste *AffectedVersions* sind die betroffenen Software-Versionen aufgeführt, die von dem Update betroffen sind.

Der Ablauf eines Software-Updates in der FactorySandbox ist in Abb. 4 skizziert. Sobald ein Update herstellerseitig verfügbar wird, muss es, inklusive dazugehöriger PCN, in die Typ-AAS der betroffenen Software eingetragen werden. Das erledigt der Service *SW-Update Server*, der den gesamten Software-Update-Prozess herstellerseitig orchestriert. In der Typ-AAS wird ein neuer Eintrag jeweils im PCN-Teilmodell und im Release-Katalog erstellt. In der FactorySandbox orchestriert die Software *SW-Update Client* den Update-Prozess. Sie fragt die REST-API des *SW-Update Server* zyklisch ab, ob beim Hersteller neue und relevante Software-Updates vorliegen. Dafür muss der Client die Softwarestände aller Geräte kennen, die in der Sandbox verbaut sind.

Der verantwortliche Betreiber der Sandbox kann mittels der gelieferten PCNs und der Informationen aus den relevanten SRC-Teilmodellen eine Betroffenheitsanalyse durchführen

und anschließend das Update bestätigen, wie in der Abbildung dargestellt, oder ablehnen. Wird das Update bestätigt, löst das den Download des Updates aus. Für die automatische Installation des Updates benötigt der Client die IP-Adresse der Zielhardware und Zugriff, um das Update installieren zu können. Der automatische Installationsprozess kann beispielsweise mit dem open-source Framework Eclipse hawkBit [3] durchgeführt werden. Nach erfolgtem Update wird der Hersteller informiert, dass das Update installiert wurde. Dieser kann nun den Softwarestand im IDTA-Teilmodell *Nameplate for Software in Manufacturing* der betroffenen Komponenten-Instanz (hier SPS-123-YYYY) aktualisieren.

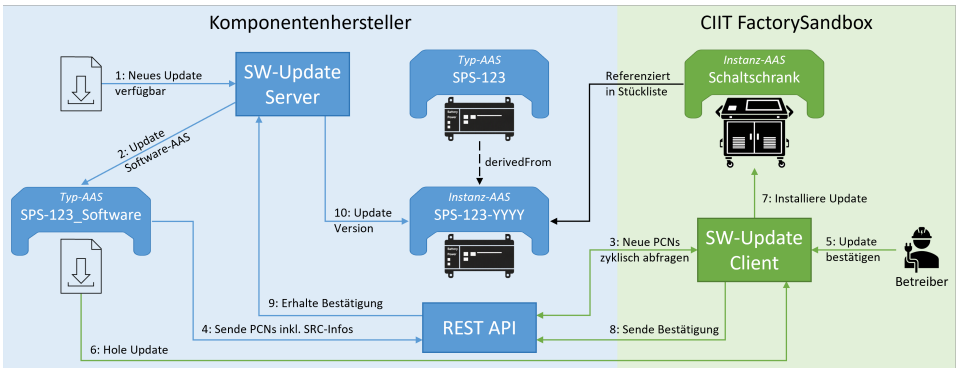


Abb. 4: Konzept des automatisierten Software-Updates in der FactorySandbox.

4 Diskussion und Ausblick

Die in der CIIT FactorySandbox umgesetzten Use Cases zeigen, wie die AAS als standardisierte digitale Repräsentation dazu beitragen kann, Transparenz und Interoperabilität über den Lebenszyklus industrieller Anlagen und deren Verfügbarkeit zu erhöhen.

Der Anwendungsfall Handover Documentation zeigt, wie die AAS als einheitliche Zugriffsschicht auf technische Dokumentation über Herstellergrenzen hinweg genutzt werden kann. Die Bereitstellung von Betriebsanleitungen, Wartungsunterlagen und Konformitätserklärungen über das standardisierte IDTA-Teilmodell *Handover Documentation* reduziert herstellerspezifische Zugriffshürden. Die Qualität und Aktualität der bereitgestellten Inhalte verbleiben in der Verantwortung des jeweiligen Herstellers.

Der Anwendungsfall der Telemetriedaten-basierte Echtzeitüberwachung und -steuerung demonstriert, wie bestehende Infrastruktur ohne Zusatzhardware für IIoT-Anwendungen genutzt werden kann. Die im Stäubli CombiTac integrierte Sensorik und Kommunikationsfähigkeit ermöglicht bspw. die bedarfsgerechte Kühlung. Ein Vorteil liegt in der Nutzung der bereits verbauten Komponenten, was Kosten und Installationsaufwand minimiert. Zukünftig könnte der Ansatz auf weitere Komponenten ausgeweitet werden, wobei die Integration von KI-gestützter Anomalieerkennung noch zu vertiefen ist.

Das vorgestellte Konzept für automatisierte Software-Updates hat durch die Nutzung der AAS den Vorteil, dass ein Hersteller mit allen Nutzern seiner Software in einem einheitlichen Verfahren Software-Updates durchführen kann. Das Konzept benutzt ein proprietäres Teilmodell SRC, um die Historie aller Software-Updates einer Software nachvollziehbar zu machen und somit die Anforderungen an die Betroffenheitsanalyse zu erfüllen. In Zukunft kann das Teilmodell, ggf. mit Erweiterungen, standardisiert werden, um einen einheitlichen Software-Update-Prozess in der Industrie zu ermöglichen.

Die Umsetzung in einer heterogenen Multi-Vendor-Umgebung verdeutlicht dabei das Potenzial der AAS, Informationen und Daten unterschiedlicher Komponenten und Systeme konsistent zusammenzuführen und für verschiedene Akteure nutzbar zu machen. Gleichzeitig zeigt die CIIT FactorySandbox, dass die AAS eine sinnvolle Grundlage für integrierte industrielle Anwendungsszenarien ist. Die Ergebnisse bilden damit eine Grundlage für die weitere Erprobung und Skalierung AAS-basierter Lösungen mit dem Ziel, Anlagenverfügbarkeit, Transparenz und Effizienz entlang des industriellen Lebenszyklus zu steigern.

Literatur

[1] H. Boyes, B. Hallaq, J. Cunningham und T. Watson, „The industrial internet of things (IIoT): An analysis framework," *Computers in Industry*, Jg. 101, S. 1–12, 2018, ISSN: 0166-3615. DOI: https://doi.org/10.1016/j.compind.2018.04.015. Adresse: https://www.sciencedirect.com/science/article/pii/S0166361517307285.

[2] Component Obsolescence Group Deutschland, *Specification of smartPCN 3.0*, März 2018. Adresse: https://smartpcn.org/wp-content/uploads/2021/01/Specification-smartPCN-3.01-released-20180323.pdf.

[3] Eclipse Foundation, *hawkBit*, 2015. Adresse: https://hawkbit.eclipse.dev, Aufruf: 19.08.2026.

[4] Eclipse Foundation, *Eclipse BaSyx*, 2017. Adresse: https://eclipse.dev/basyx, Aufruf: 04.08.2026.

[5] Eclipse Foundation, *AASX Server*, 2022. Adresse: https://projects.eclipse.org/projects/dt.aaspe, Aufruf: 04.08.2026.

[6] Eclipse Foundation, *Eclipse Fa³st*, 2023. Adresse: https://projects.eclipse.org/projects/dt.fa3st, Aufruf: 04.08.2026.

[7] Factory-X Consortium, *Factory-X contributions - Update Hub*, Juni 2026. Adresse: https://github.com/factory-x-contributions/update-hub.

[8] Hilscher Gesellschaft für Systemautomation mbH, *netFIELD Device Management*, https://netfield.io/, 2026, Aufruf: 28.09.2026.

[9] Industrial Digital Twin Association, „IDTA 02004-1-2 Handover Documentation," Industrial Digital Twin Association, Submodel Template of the Asset Administration Shell 02004, März 2023, Version 1.2. Adresse: https://industrialdigitaltwin.org/wp-content/uploads/2023/03/IDTA-02004-1-2_Submodel_Handover-Documentation.pdf.

[10] Industrial Digital Twin Association, *Asset Administration Shell – Reading Guide*, Nov. 2024. Adresse: https://industrialdigitaltwin.org/wp-content/uploads/2024/11/2024-11_IDTA_AAS-Reading-Guide.pdf.

[11] Industrial Digital Twin Association, *Asset Administration Shell – Quick Start Guide*, Juli 2025. Adresse: https://industrialdigitaltwin.org/wp-content/uploads/2025/07/IDTA_AAS-Quick-Start-Guide.pdf.

[12] Industrial Digital Twin Association, „Product Change Notifications for industrial product types and items in manufacturing," Industrial Digital Twin Association, Submodel Template Specification 02036, Apr. 2026, Version 1.0. Adresse: https://idtwin.org/en/wp-content/uploads/sites/2/2026/04/IDTA_02036_Submodel_ProductChangeNotifications.pdf.

[13] Industrial Digital Twin Association, „Specification of the Asset Administration Shell – Part 2: Application Programming Interfaces," Industrial Digital Twin Association, Specification of the Asset Administration Shell 01002, Juli 2026, Version 3.2. Adresse: https://industrialdigitaltwin.io/aas-specifications/IDTA-01002/v3.2/index.html.

[14] International Electrotechnical Commission, „Obsolescence management," International Electrotechnical Commission, International Standard IEC 62402:2019, Mai 2019. Adresse: https://webstore.iec.ch/en/publication/59531.

[15] International Electrotechnical Commission, „Industrial-process measurement, control and automation - Life-cycle-management for systems and components," International Electrotechnical Commission, International Standard IEC 62890:2020, Juli 2020. Adresse: https://webstore.iec.ch/en/publication/30583.

[16] International Electrotechnical Commission, „Asset Administration Shell for industrial applications – Part 1: Asset Administration Shell structure," International Electrotechnical Commission, International Standard IEC 63278-1:2023, Dez. 2023. Adresse: https://webstore.iec.ch/en/publication/65628.

[17] C. Liepert, C. Stary, A. Lamprecht und D. Zügn, „Interoperable Product Change Management Within Engineering: A Digital Twin Approach," in *Subject-Oriented Business Process Management. Models for Designing Digital Transformations*, M. Elstermann und M. Lederer, Hrsg., Cham: Springer Nature Switzerland, 2025, S. 256–261, ISBN: 978-3-031-72041-3.

[18] Open Industry 4.0 Alliance, *Don't Miss Critical Changes! AAS Product Change Notification*, 2025. Adresse: https://openindustry4.com/project/aas-product-change-notification, Github: https://github.com/OI4/AAS_PCN_Demonstrator/tree/main, Aufruf: 07.08.2026.

[19] Plattform Industrie 4.0, *Details of the Asset Administration Shell – from idea to implementation*, Juli 2019. Adresse: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/vws-in-detail-presentation.html.

[20] Plattform Industrie 4.0, *Kollaborative datenbasierte Geschäftsmodelle: Collaborative Condition Monitoring – Wie durch unternehmensübergreifende Kollaboration Mehrwert generiert werden kann.* Juli 2020. Adresse: https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/kollaborative-datenbasierte-geschaeftsmodelle.html.

[21] Plattform Industrie 4.0, *What is the Asset Administration Shell from a technical perspective?* Apr. 2021. Adresse: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/2021_What-is-the-AAS.html.

[22] Verband Deutscher Maschinen- und Anlagenbau, „Obsoleszenzmanagement - Informationsaustausch zu Änderungen und Abkündigungen von Produkten und Einheiten," DIN Media GmbH, Technische Regel VDMA 24903:2017-12, Dez. 2017. Adresse: https://www.dinmedia.de/de/technische-regel/vdma-24903/281802112.

A Lightweight NETCONF Proxy Architecture for Resource-Constrained TSN Devices

Jonas Janzen[1], Hamza Chahed[2], Andreas Kassler[2,3], and Henning Trsek[1]

Abstract: IEC/IEEE 60802 specifies NETCONF/YANG for centralized configuration of industrial Time-Sensitive Networking devices, while allowing resource-constrained devices to be managed through a proxy. The protocol between proxy and target remains open. This paper presents a conceptual architecture that retains NETCONF/YANG toward the controller and uses CORECONF toward constrained targets. A YANG datastore separates the two interfaces: the proxy identifies changes intended for a target and translates the affected YANG instances into CORECONF requests. An `edit-config` example illustrates the path from XML configuration to a SID-based YANG-CBOR update.

Keywords: Time-Sensitive Networking, IEC/IEEE 60802, NETCONF, CORECONF, Management Proxy

1 Introduction

Industrial automation systems increasingly depend on flexible and interconnected communication infrastructures. At the same time, time-critical applications require predictable latency, bounded jitter, and reliable data transmission. Time-Sensitive Networking (TSN) extends Ethernet with mechanisms for deterministic communication and is therefore considered an important technology for converged industrial networks [LS19]. The IEC/IEEE 60802 TSN Profile for Industrial Automation [IE26] specifies how selected TSN mechanisms are applied in interoperable, multi-vendor industrial networks. An important part of this profile is the centralized configuration of network components. A Centralized Network Configuration (CNC) calculates network-wide configurations and remotely applies them to the participating bridges and end stations. This approach can simplify network engineering and support the integration or reconfiguration of industrial devices.

Centralized management requires the participating devices to expose standardized configuration and status information. IEC/IEEE 60802 uses Network Configuration Protocol (NETCONF) [En11] together with Yet Another Next Generation (YANG) data models [Bj10] for this purpose. However, industrial networks also contain resource-constrained embedded devices whose available memory, code space, and processing capacity are limited.

[1] Technische Hochschule Ostwestfalen-Lippe, Institute Industrial IT (inIT), Campusallee 6, 32657 Lemgo, Germany, jonas.janzen@th-owl.de; henning.trsek@th-owl.de

[2] Karlstad University, Universitetsgatan, 651 88 Karlstad, Sweden, hamza.chahed@kau.se; andreas.kassler@th-deg.de

[3] Technische Hochschule Deggendorf, Dieter-Görlitz-Platz 1, 94469 Deggendorf, Germany, andreas.kassler@th-deg.de

Implementing a complete NETCONF server, including XML processing and the required transport and security functionality, can impose a considerable resource burden on such devices [Bh24; Se12].

IEC/IEEE 60802 addresses this limitation through the concept of a management proxy. An industrial automation station with reduced features – IEC/IEEE 60802 Conformance Class B (ccB) devices – may use the NETCONF server of another station. The management proxy provides the standardized NETCONF/YANG interface towards the CNC and represents one or more constrained devices, referred to as proxy targets.

While the standard defines the general roles and interaction between the CNC, proxy, and proxy targets, it intentionally leaves the protocol between the proxy and its targets implementation-specific. Consequently, several design questions remain open. A suitable target-side protocol must provide compact communication while preserving the semantics of the YANG-based configuration received through NETCONF. The proxy must translate between different protocol operations and data representations, associate incoming configuration data with the correct target, and return meaningful results to the CNC.

This paper proposes a proxy architecture for translating IEC/IEEE 60802 configuration toward constrained devices while retaining YANG-modeled data on both interfaces. The central design step is to handle NETCONF operations at a YANG datastore and derive target-specific CORECONF requests from the resulting data changes. We explain this separation through the logical architecture and one `edit-config` example. Implementation details and quantitative evaluation are left to subsequent work.

2 Background

2.1 Management Proxy

IEC/IEEE 60802 uses the IEEE 802.1Q [IE22] fully centralized configuration model: Centralized User Configuration (CUC) entities provide the communication requirements of each industrial application that wants to use the network, and the CNC computes and distributes configuration to the network devices via NETCONF/YANG. Conformance Class B (ccB) devices have lighter TSN and management requirements and may use another device's NETCONF server as a proxy. With this approach, constrained devices that are not capable of running a NETCONF server can still be part of centralized network configuration. The management proxy is a network device that supports NETCONF itself and handles the translation and forwarding of the configuration to a ccB device. The ccB device that receives this configuration is described as the proxy target. Since the use of a proxy is optional for ccB devices, not every ccB device necessarily acts as a proxy target. The protocol used between the proxy and the proxy target is explicitly left user-specific by IEC/IEEE 60802.

2.2 CORECONF

CoAP Management Interface (CORECONF) [Ve26] is a network management interface for constrained devices and networks currently specified by the IETF CoRE Working Group. Similar to NETCONF, CORECONF follows the model-driven approach and operates on configuration and state data described by YANG modules. However, it uses the Constrained Application Protocol (CoAP) to access YANG-defined datastore, data node, and event stream resources. The transferred instance data is encoded according to the YANG-to-CBOR mapping defined in RFC 9254 [Ve22] to reduce payload size. Concise Binary Object Representation (CBOR) is a binary data representation format designed for small message and implementation sizes. The YANG-CBOR mapping defines how configuration data is represented using CBOR. Numeric Schema Item Identifiers (SIDs) [Ve24] replace textual YANG node names in a southbound representation. In benchmarks comparing CORECONF, NETCONF, and RESTCONF on resource-constrained IIoT devices, CORECONF showed promising time, power, and memory characteristics [Bh24]. This provides empirical support for considering it as a candidate for the proxy–target interface.

3 Related Work

Existing work in the area of NETCONF proxies remains limited. A small number of publications describe proxies, gateways, and protocol adapters that expose a uniform northbound interface while translating requests toward heterogeneous or resource-constrained devices. Such approaches are typically targeted toward the domain of the Internet of Things, where device heterogeneity and limited resources motivate the separation between the NETCONF management interface and a simpler device-facing protocol [CJ20; SSS18]. For the combined problem of NETCONF/YANG proxying for resource-constrained targets in industrial TSN configuration, the reviewed literature provides even less coverage.

Sahlmann et al. [SSS18] propose a NETCONF–MQTT bridge that mediates between a NETCONF-based management system and IoT devices communicating through MQTT. The bridge dynamically generates YANG data models from semantic descriptions of device capabilities, thereby allowing heterogeneous IoT devices to be represented through a common NETCONF/YANG interface. However, the MQTT communication and translation follow a customized scheme and are specific to the proposed bridge. In addition, the authors still identify the size of the JSON-encoded configuration data as a challenge for constrained devices [SSS18]. It therefore establishes the feasibility of NETCONF translation, but leaves open the use of a compact, YANG-native southbound configuration protocol.

Sambandan and Thirupathi map YANG models to OPC UA information models [ST24] and extend this mapping to a runtime adapter between OPC UA applications and NETCONF-managed devices [ST25]. However, their approach retains NETCONF on the device-facing side of the translator, while the IEC/IEEE 60802 proxy considered here instead needs

NETCONF toward the CNC and a lightweight protocol toward constrained devices. Their adapter therefore does not address the required southbound translation and introduces an additional semantic mapping step, which does not preserve YANG-based southbound configuration semantics directly.

Albrecht et al. [AHB25] investigate NETCONF proxy configuration for a resource-constrained industrial device in an IEC/IEEE 60802-related TSN scenario. Their concept retains NETCONF/YANG toward the management system and transfers CBOR-encoded configuration information protected by CBOR Object Signing and Encryption (COSE) toward the constrained device. It is the closest approach in application context, but utilizes a custom target-side configuration mechanism. It provides compact protected messages but requires configuration operations and their semantics to be specified separately.

The proposed architecture in this work differs from the NETCONF-to-CBOR proxy of Albrecht et al. [AHB25] by using CORECONF as a YANG-aware southbound management interface with specified operations instead of a custom CBOR-based target protocol.

4 Proposed Proxy Design

The proposed architecture provides a standard NETCONF/YANG interface to the central network controller while using a lightweight, CORECONF-based interface for proxy targets with limited resources. Both interfaces use YANG as their common data model, while the representation changes inside the proxy from XML northbound to YANG-CBOR southbound. Figure 1 shows the logical components of the proxy architecture, separating the CNC, the NETCONF-facing proxy, and the constrained targets. Inside the NETCONF proxy, a standard NETCONF server is available to receive the configuration that is intended for a proxy target. How the CNC knows that this proxy is responsible for a certain proxy target is considered out of scope. The configuration is stored in a YANG datastore inside the proxy. Each target-facing datastore context requires a change-observation mechanism. When a relevant configuration change occurs, the change is passed to the Proxy Engine together with its target context, operation, YANG instance path, and value. The Proxy Engine resolves the corresponding downstream target and preserves the ordering of operations addressed to that target. The translation component converts the textual YANG path and value into a SID-based YANG-CBOR representation, which is transmitted through the CORECONF client. A target registry inside the Orchestration logic keeps device identity separate from the YANG path within that device. For each proxy target, it associates an identifier with a CORECONF endpoint, the applicable YANG modules and SID assignments, the operations supported by the target, and the current state of the endpoint. The Proxy Engine uses this context to select the destination and determine whether a requested data change can be forwarded.

Table 1 shows how NETCONF operations could be mapped to CORECONF communication through the YANG datastore. It follows the distinction between local proxy datastore

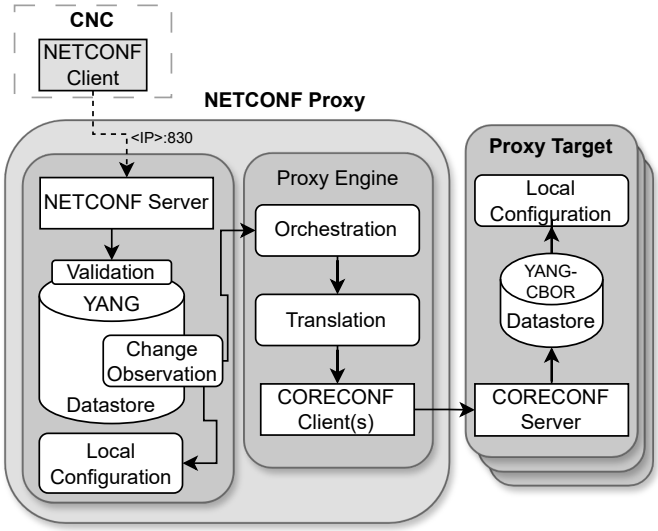


Fig. 1: Logical components of the proposed NETCONF-to-CORECONF proxy.

operations and operations requiring target communication in Albrecht et al. [AHB25]. The proxy derives the effective changes rather than mapping the NETCONF operation directly to a CoAP method. According to the NETCONF RFC [En11], the `<running>` YANG datastore holds the active configuration, the `<candidate>` datastore stages edits, and the `<startup>` datastore holds the configuration used at boot.

Tab. 1: Conceptual mapping through the proxy datastore

NETCONF	YANG datastore action	CORECONF
`get`	Read configuration; Request live state if needed	 `FETCH` for live state
`get-config`	Read stored configuration	None (the proxy already holds the current target configuration)
`edit-config`	<candidate>: edit local configuration; <running>: forward edits to target <startup>: forward edits to target	 `iPATCH` for affected instances `iPATCH` for affected instances
`copy-config`	Copy configuration Forward if target is <running> or <startup>	 `iPATCH` for affected instances
`commit`	Forward <candidate> changes into <running> datastore of the target	`iPATCH` for affected instances
`discard-changes`, `lock`, `unlock`	Handle within the proxy datastore	None

Figure 2 illustrates a configuration update for a bridge port. To assign VLAN 100 as the

Port VLAN Identifier (PVID) of interface `swp1`, the CNC sends a NETCONF `edit-config` request to the proxy. Its XML configuration identifies the interface with `<name>swp1</name>` and sets `<pvid>100</pvid>` inside the `bridge-port` container. This container extends the `ietf-interfaces` model through the `ieee802-dot1q-bridge` model. After YANG validation, the datastore observer identifies the changed `pvid` leaf and the proxy selects the corresponding target. The translation uses the leaf's numeric SID together with the list key `swp1` and encodes the value 100 as a CBOR integer. The proxy sends the resulting CORECONF `iPATCH` request to the target, which validates the update and returns a response. Finally, the proxy reports success to the CNC.

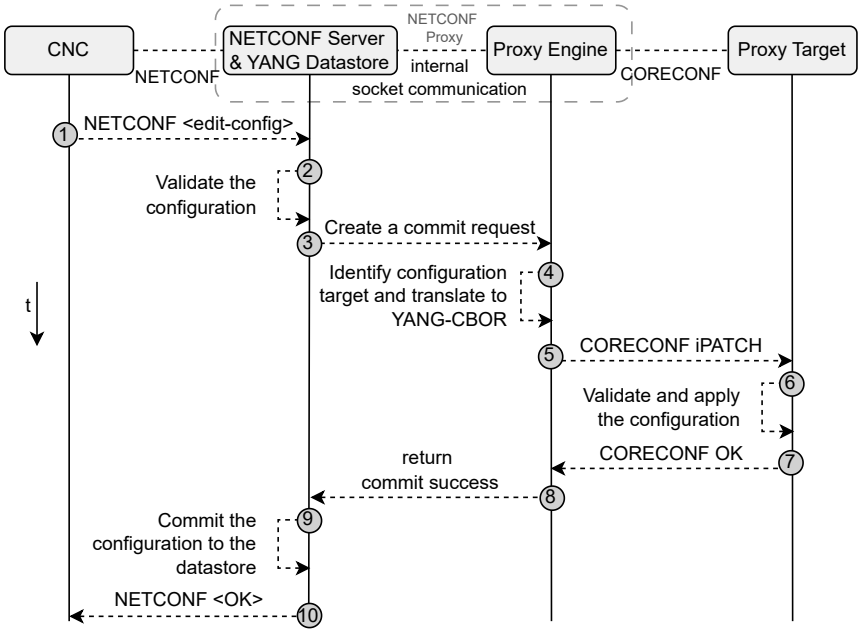


Fig. 2: Conceptual configuration flow for one NETCONF `edit-config` update.

5 Conclusion

IEC/IEEE 60802 allows constrained TSN devices to be managed through a NETCONF proxy, but leaves the protocol between proxy and target open. This paper proposes CORECONF for that interface: CoAP and YANG-CBOR keep communication compact, while YANG provides a common basis for configuration on both sides. It then presents the main idea of a proxy architecture that receives NETCONF/YANG configuration from the CNC at a YANG datastore and translate it into target-specific CORECONF requests from the resulting data changes.

In future work, we plan to implement and further develop the proposed architecture, including support for CORECONF in embedded systems and the management of multiple target devices. The implementation needs to be evaluated for correctness, configuration latency, message size, resource consumption, and scalability. Secure communication, onboarding, and consistency between the proxy datastore and device state remain open design questions.

References

[AHB25] Albrecht, J.; Heubacher, K.; Biendarra, A.: Secure Engineering of Ethernet TSN Networks for Resource Constrained Industrial Devices. In (Underberg, L.; Jasperneite, J., eds.): Kommunikation in der Automation : 16. Jahreskolloquium, 11./12.11.2025 : Tagungsband. Institut für Automation und Kommunikation e.V., An-Institut der Otto-von-Guericke-Universität Magdeburg, Magdeburg, pp. 155–164, 2025, accessed: 11/17/2025.

[Bh24] Bhat, M. G. et al.: CORECONF, NETCONF, and RESTCONF: Benchmarking Network Orchestration in Constrained IIoT Devices. IEEE Internet of Things Journal, p. 1, 2024.

[Bj10] Bjorklund, M.: YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF), 2010.

[CJ20] Chung, C.; Jeong, J. P.: A Design of IoT Device Configuration Translator for Intent-Based IoT-Cloud Services. In: 2020 22nd International Conference on Advanced Communication Technology (ICACT). IEEE, pp. 52–56, 2020.

[En11] Enns, R. et al.: Network Configuration Protocol (NETCONF), 2011, https://www.rfc-editor.org/info/rfc6241.

[IE22] IEEE: 802.3-2022: IEEE Standard for Ethernet. IEEE, New York, NY, 2022.

[IE26] IEC/IEEE: IEC/IEEE 60802 TSN Profile for Industrial Automation, 2026, https://standards.ieee.org/ieee/60802/11358/, accessed: 07/29/2024.

[LS19] Lo Bello, L.; Steiner, W.: A Perspective on IEEE Time-Sensitive Networking for Industrial Communication and Automation Systems. Proceedings of the IEEE 107 (6), pp. 1094–1120, 2019.

[Se12] Sehgal, A. et al.: Management of resource constrained devices in the internet of things. IEEE Communications Magazine 50 (12), pp. 144–149, 2012.

[SSS18] Sahlmann, K.; Scheffler, T.; Schnor, B.: Ontology-driven Device Descriptions for IoT Network Management. In: 2018 Global Internet of Things Summit (GIoTS). IEEE, pp. 1–6, 2018.

[ST24] Sambandan, D.; Thirupathi, D.: Towards a Unified Management Interface for 5G Sensor Networks: Interoperability between Yet Another Next Generation and Open Platform Communication Unified Architecture. Sensors (Basel, Switzerland) 24 (19), 2024.

[ST25] Sambandan, D.; Thirupathi, D.: Improving Efficiency in Industry 4.0 Automation Environment Through a Unified Interface for Interoperability Between NETCONF and OPC UA. In (Rathore, V. S. et al., eds.): Modern Practices and Trends in Expert Applications and Security. Vol. 1379, Lecture Notes in Networks and Systems, Springer Nature Singapore, Singapore, pp. 259–272, 2025.

[Ve22] Veillette, M. et al.: RFC 9254: Encoding of Data Modeled with YANG in the Concise Binary Object Representation (CBOR), 2022.

[Ve24] Veillette, M. et al.: RFC 9595: YANG Schema Item iDentifier (YANG SID), 2024.

[Ve26] Veillette, M. et al.: CoAP Management Interface (CORECONF): Internet-Draft, 2026, https://datatracker.ietf.org/doc/draft-ietf-core-comi/, accessed: 06/13/2026.

6G Roadmap: A Systematic Analysis Derived from 6G Platform Germany White Papers

Parva Yazdani[1] and Lisa Underberg[1]

Abstract: 6G promises to turn wireless connectivity into critical infrastructure for industrial automation. The white paper "German Perspective on 6G" surveys 29 technical building blocks across eight use case families, providing a broad foundation for this development. We build on this survey by applying a validated 70% cross family consensus threshold to its data and find that only security and trust qualifies as a universally mandatory enabler, with sensing, positioning, and time synchronization ranking immediately below it. Both connect to existing automation standards, Time Sensitive Networking, OPC UA, and IEC 62443, rather than to a generic 6G security or sensing concept. Reading this priority against the usability and cost challenges surveyed in a second white paper on lessons learned from 5G, we derive a roadmap where new 6G capabilities are delivered through the standards aligned tooling small and medium sized enterprises already use, rather than a parallel 6G specific stack.

Keywords: 6G, Industrial Automation, Technical Building Blocks, Security and Trust, Sensing and Positioning

1 Introduction

The evolution of wireless communication has moved it beyond simple connectivity toward a fundamental technological infrastructure for industrial ecosystems. Networks are increasingly used as critical enablers for physical autonomy and sophisticated intelligence rather than serving merely as conventional data exchange utilities. In industrial automation, 6G is expected to evolve from best effort delivery models toward supporting highly synchronised, mission critical environments. Collaborative mobile robotics illustrates this shift well, since it requires real time coordination between robotic swarms and human personnel in shared workspaces with strict safety requirements.

The white paper "German Perspective on 6G, Use Cases, Technical Building Blocks, and Requirements" [FD24] offers a comprehensive, use case driven overview of this development. Its breadth offers scope for additional prioritisation work that can help readers decide where to focus limited engineering effort, particularly smaller organisations without dedicated standardisation teams. A second white paper, "From Lessons Learned in 5G to Innovative Solutions in 6G" [Un25], complements this technical perspective with a survey-based account of the challenges hindering the adoption of industrial 5G and the lessons that should contribute to the design of 6G.

[1] ifak e.V., Werner-Heisenberg-Straße 1, 39106 Magdeburg, Germany, parva.yazdani@ifak.eu; lisa.underberg@ifak.eu

This paper builds on that breadth to add a layer of prioritisation and applicability, rather than repeating the content of either white paper. Its contribution is threefold. First, we apply a validated, threshold based filtering methodology to the survey data behind "German Perspective on 6G" to distil a small number of prioritised technical enablers from this large scale use case survey. Second, we examine the two resulting top ranked building blocks, security and trust and sensing, positioning and time synchronization, in technical depth with respect to their concrete relevance for industrial real time communication, including connections to established automation standards. Third, we connect this prioritised technical view with the usability and cost challenges documented in "Lessons Learned in 5G to Innovative Solutions in 6G", translating the resulting usability gap into actionable implications for small and medium sized enterprises in the automation sector. The combination of these three elements yields a roadmap that is specific to industrial automation rather than a general summary of either source.

The remainder of this paper is structured as follows. Section 2 introduces the two source white papers and positions this paper's contribution relative to them. Section 3 describes the methodology, including the threshold based filtering and its validation. Section 4 presents the technical enabler analysis. Section 5 derives the roadmap for industrial automation. Section 6 discusses the limitations of this analysis, and Section 7 concludes.

2 Background

"German Perspective on 6G, Use Cases, Technical Building Blocks, and Requirements" [FD24] was produced by the working group "WG:Roadmap" of the 6G Platform Germany. In November 2023, the working group issued a call for use case contributions to the 33 BMBF funded 6G projects and collected 83 use cases from 21 projects by February 2024. Three task forces then harmonised these use cases into eight use case families, derived 29 technical building blocks organised into seven categories, and surveyed use case authors on requirements and features. Five of the eight use case families, Collaborative Robots, Digital Twins, Immersive Experience, Fully Connected World, and Trustworthy Environments, align with categories already established by the pan-European Hexa-X-II project, while Resilient Society, 3D Mobility, and Environmental Awareness were introduced as new categories reflecting the German contributors' priorities. For each technical building block, use case authors classified their need as mandatory, optional, or not needed, and the resulting scores were visualised per use case family in spider diagrams. This white paper is the primary data source for the enabler analysis in Section 4.

"From Lessons Learned in 5G to Innovative Solutions in 6G" [Un25] was produced by the 6G Platform Germany working group "Innovation Management". It documents an expert survey of 18 respondents from mobile network operators, enterprises, service vendors, industry associations, and researchers, structured around four aspects, technological, economic, standardisation, and regulatory. For each aspect, the white paper reports the challenges encountered during industrial 5G deployment, the lessons that follow from them, and

innovative solutions proposed for 6G. Its findings on the cost and complexity of non-public network deployment form the basis for the roadmap synthesis in Section 5.

Read together, the two white papers leave one specific connection unexplored: neither relates the technical building block priorities behind [FD24] to the usability and cost challenges documented in [Un25], and neither grounds a specific enabler in established industrial automation standards. The following sections build on both white papers to close this connection, rather than repeating either paper's content, deriving this paper's threefold contribution as outlined in Section 1.

3 Methodology

3.1 Data Extraction

The white paper [FD24] scores each technical building block per use case family by averaging use case authors' three point ratings, mandatory equal to one, optional equal to one half, not needed equal to zero, into a family level value between zero and one, and visualises the eight resulting values per building block as a spider diagram with three coloured bands, not needed, optional, and mandatory. For each of the 29 technical building blocks, we read off which of these three bands each of the eight use case families falls into directly from the published spider diagrams, producing a discrete tier label for every building block and use case family pair. We cross checked this extracted dataset against the white paper's own prose discussion chapter [FD24], which independently confirms, for example, that most use case families rate security and trust as mandatory, and found the two consistent.

3.2 Threshold Based Filtering

For a technical building block b and a tier $t \in \{\text{mandatory}, \text{optional}, \text{not needed}\}$, let $c_t(b)$ denote the number of the eight use case families whose spider diagram places b in tier t. We define b as a consensus enabler for tier t if

$$c_t(b)/8 \geq 0.70, \tag{1}$$

that is, if at least six of the eight use case families agree on the same tier for that building block. Applying this rule across all 29 building blocks and all three tiers isolates a small set of building blocks with strong cross family agreement, shown in Table 1. Beyond this qualifying set, we additionally rank all 29 building blocks within each tier by their raw agreement count $c_t(b)$, giving a broader top ranked list per tier that is useful for roadmap planning even where full consensus is not reached. Section 4 reports both the qualifying set and the broader ranking.

3.3 Validation of the 70% Threshold

With only eight use case families, agreement can only take the discrete values $k/8$ for $k = 0, \ldots, 8$, so any threshold strictly between 62.5% and 75% selects exactly the same qualifying set as 70%. The choice of 70% is therefore not a sensitive tuning parameter but a stable operating point within this plateau. Raising the bar above 75%, so that seven or eight of eight use case families would need to agree, would eliminate security and trust and cell-free massive MIMO from the qualifying set entirely, leaving only quantum technologies for network infrastructure, which is too strict for practical prioritisation. Lowering it toward 50% would, conversely, admit building blocks with only weak, four out of eight agreement, undermining the claim of consensus. The 70% threshold is thus the tightest bar that still yields a non-empty, practically useful qualifying set for more than one tier.

Tab. 1: Technical building blocks reaching the 70% cross family consensus threshold, by tier.

Tier	Technical building block	Use case families
Mandatory	Security and Trust	6 / 8 (75%)
Not Needed	Quantum Technologies for Network Infrastructure	8 / 8 (100%)
Not Needed	Cell-free Massive MIMO	6 / 8 (75%)
Optional	none	—

Notably, no technical building block reaches 70% cross family agreement on the optional tier, which suggests that use case families disagree far more on which building blocks are merely beneficial than on which are essential or unnecessary. We return to this asymmetry in Section 4.

4 Technical Enabler Analysis

Table 1 showed that only security and trust reaches cross family consensus on the mandatory tier. Figure 1 extends this with the broader top 10 ranking by raw agreement count introduced in Section 3.2, and Figure 2 shows, for these same 10 enablers, the full mandatory, optional, and not needed pattern across all eight use case families. The two highest ranked enablers, security and trust and sensing, positioning and time synchronization, are examined below in technical depth.

The Security and Trust building block is defined in [FD24] as an on-demand security plane, analogous to the data plane, that spans every network layer from resource to application and is intended to expose capabilities such as key management, trusted computing, and blockchain based trust anchors. Its near universal mandatory rating is not a generic call for more security. It reflects that most of the use case families rating it mandatory, Collaborative Robots, Digital Twins, 3D Mobility, Immersive Experience, and Trustworthy Environments, close a physical actuation loop, where a spoofed sensor reading or a replayed control command can

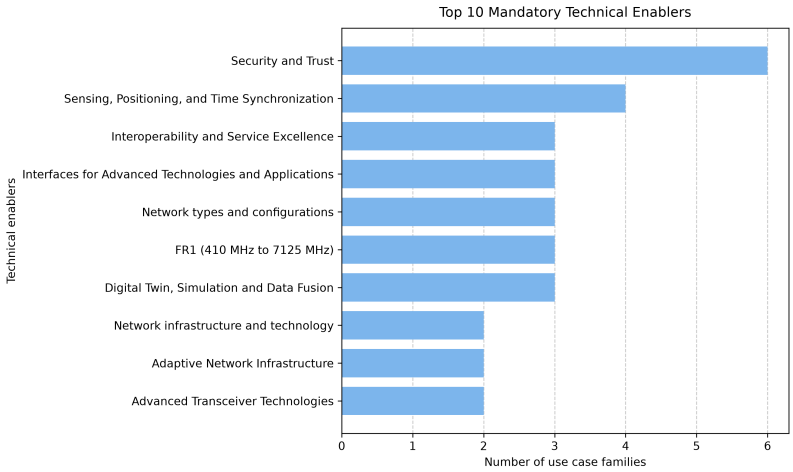


Fig. 1: Top 10 mandatory technical enablers, ranked by the number of use case families that classify them as mandatory.

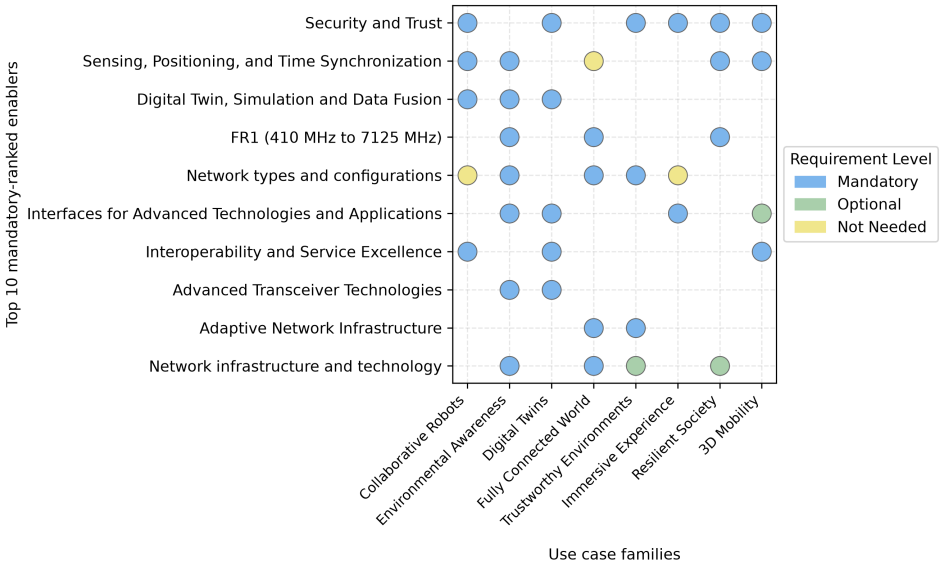


Fig. 2: Full mandatory, optional, and not needed classification of the top 10 mandatory-ranked enablers across the eight use case families.

cause physical harm rather than only a data breach. This changes the engineering problem,

since security checks must fit within the same real time bound as the control loop itself rather than being layered on afterwards.

This is where the connection to existing automation standards becomes concrete. Time Sensitive Networking (TSN) delivers deterministic timing through traffic scheduling, but does not itself authenticate or encrypt frames. In practice, this task increasingly falls to OPC UA, which industry already deploys over TSN as the application layer for real time shopfloor communication [LN23]. OPC UA's own security model signs and encrypts every message over a given connection using per-session symmetric keys and authenticates each application instance through an exchanged certificate [OP25]. It follows that a 6G security plane adding its own authentication step on the radio link has to fit within the timing budget that OPC UA over TSN already guarantees, rather than introduce a second, uncoordinated security boundary. Likewise, IEC 62443 [IE09] already gives automation engineers, including those at small and medium sized enterprises with no dedicated 6G security expertise, an established vocabulary of security zones and conduits for scoping requirements. For 6G's security plane to be adopted rather than bypassed, it needs to map onto these existing zones and conduits instead of asking practitioners to learn a parallel model.

The Sensing, Positioning, and Time Synchronization building block is defined in [FD24] as the localisation of cooperative objects and the distribution of spatial and temporal information through time synchronisation, distinguishing it from joint communication and sensing, which senses non-cooperative objects. The white paper itself notes that its time synchronisation aspect connects directly to TSN [FD24]. This connection is concrete rather than incidental, since TSN's own synchronisation protocol, IEEE 802.1AS generalised precision time protocol, requires sub microsecond clock accuracy across a wired industrial network. Where a 6G radio link extends or replaces part of such a network, for instance as a wireless bridge to a mobile robot, its own time synchronisation must meet the same accuracy bound, or the wireless segment becomes the weak link that breaks the end to end determinism the wired network otherwise guarantees. This matters beyond timing alone, since a closed loop safety function such as collision avoidance between a robotic swarm and human personnel depends on both the accuracy and the update rate of the position estimate the network delivers, not on accuracy in isolation.

Figures 3 and 4 extend the ranking to the other two tiers and make the asymmetry flagged in Section 3.3 visible. The not needed tier mirrors the mandatory tier's clear separation: Quantum Technologies for Network Infrastructure and Cell-free Massive MIMO reach 8 of 8 and 6 of 8 use case families respectively, well above the remaining entries. The optional tier shows no such separation: its top ranked enablers, Edge Computing Innovations and FR2 (24.25 GHz to 71.0 GHz), reach only 3 of 8 use case families, and the ranking flattens quickly from there. Use case families agree readily on what a technology must do and on what it can skip entirely, but disagree on the wide middle ground of merely beneficial capabilities.

Both enablers, one reaching strict cross family consensus and one ranking immediately

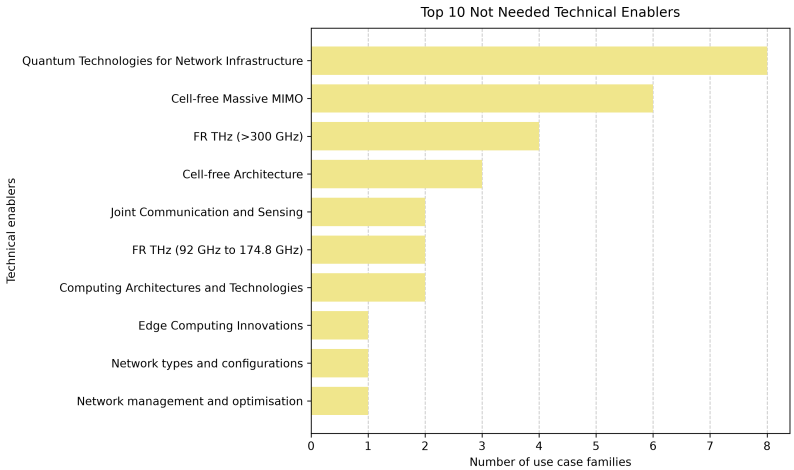


Fig. 3: Top 10 not needed technical enablers, ranked by the number of use case families that classify them as not needed.

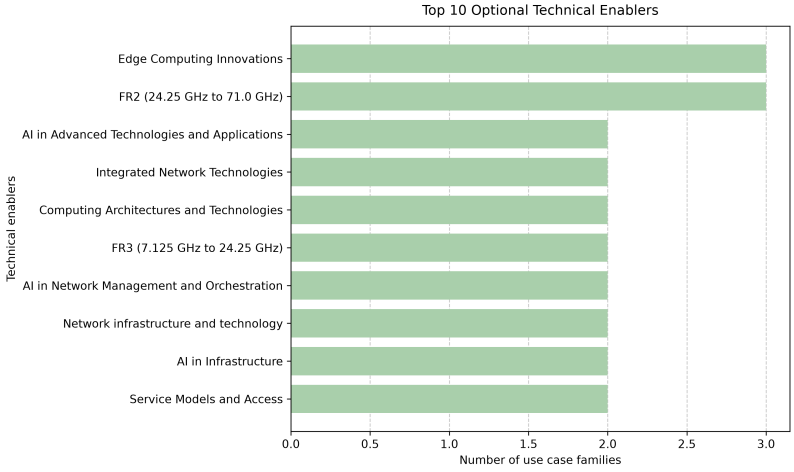


Fig. 4: Top 10 optional technical enablers, ranked by the number of use case families that classify them as optional.

below it, turn out to be exactly the two building blocks that the white paper itself ties most directly to existing deterministic industrial networking [FD24]. This is not a coincidence of the threshold chosen in Section 3, but a sign that the enablers use case families agree on most strongly are the ones with the least room for a purely optional, best effort implementation.

5 Roadmap for Industrial Automation

Even the technical white paper behind Section 4 recognises that small and medium sized enterprises need distinct attention, noting explicitly that contributions from industry partners, especially SMEs, were emphasised to reflect their standardisation needs [FD24]. Underberg et al. [Un25]'s survey makes the resulting usability gap concrete. A large majority, 87%, of respondents agreed that expert knowledge or reliance on external service providers is critical when deploying 5G, and 88% agreed that high deployment cost significantly slows adoption. Most respondents also preferred 5G as an add-on to existing infrastructure rather than a full replacement, and 93% saw a modular hardware and firmware approach, replaceable radio modules with updatable firmware, as a feasible way to move from 5G to 6G.

These findings reframe the two enablers examined in Section 4. Both add configuration complexity, a security plane spanning every network layer and a sensing and positioning capability tied to strict timing accuracy, on top of infrastructure that Underberg et al. [Un25]'s respondents already found too dependent on outside expertise and too costly to deploy. Table 2 translates this into concrete roadmap implications for automation SMEs.

Tab. 2: From the usability gap in [Un25] to roadmap implications for automation SMEs.

Usability finding	Added complexity from Section 4	Roadmap implication
87% need expert knowledge or an external provider to deploy 5G	Security plane spans every layer down to the resource layer	Ship the security plane as pre-configured profiles aligned to IEC 62443 zones and conduits, not a bespoke 6G configuration task
88% cite high cost as a barrier; most prefer 5G as an add-on, not a replacement	Sensing and positioning needs sub microsecond timing accuracy across the network	Deliver sensing and positioning as an add-on radio bridge to the existing wired TSN network, not a parallel infrastructure
93% support a modular hardware and firmware radio approach for 5G to 6G migration	Both enablers assume tight integration with the existing network stack	Expose new 6G capabilities through the same modular, updatable radio hardware, so SMEs upgrade firmware rather than redesign their network

This is not a hypothetical ask. Labs Network Industrie 4.0 [LN23] shows that testbeds pairing OPC UA with TSN, the same standards Section 4 ties the two top enablers to, are already built around small and medium sized enterprise requirements rather than large-scale industrial deployments. The roadmap implication for 6G in industrial automation is therefore not to build new SME-facing tooling from scratch, but to route the two enablers with the strongest cross family consensus through the standards-aligned, SME-oriented tooling that already exists for TSN and OPC UA.

6 Discussion

Three limitations qualify this analysis. First, the tier labels behind Table 1 and Section 4 were read directly off the published spider diagrams in [FD24] rather than from the underlying numeric survey averages, which the white paper does not publish; the cross check in Section 3.1 against its own prose summary reduces, but does not eliminate, the risk of a misread diagram. Second, the technical depth in Section 4 covers only the two enablers with the strongest cross family agreement; the broader top 10 ranking in Figure 1 is not examined to the same depth, so the roadmap in Section 5 is deliberately narrow rather than exhaustive. Third, Underberg et al. [Un25]'s survey does not itself distinguish respondents by company size, so the small and medium sized enterprise framing in Section 5 is our own reasonable inference from general findings on cost and expert knowledge dependency, not a claim Underberg et al. [Un25] tested directly for that group.

The 70% threshold itself withstands this scrutiny. As shown in Section 3.3, with only eight use case families the qualifying set is stable across the entire 62.5% to 75% range, so the specific choice of threshold is not the source of any of the three limitations above. A different threshold within that range would isolate the same two enablers pursued in Section 4 and the same roadmap in Section 5.

7 Conclusion

This paper set out to turn the breadth of the use case and technical building block survey behind [FD24] into a short, prioritised, and technically grounded roadmap for industrial automation, without simply restating either source white paper. The 70% cross family consensus threshold, itself shown to be robust to reasonable variation, isolates exactly one mandatory enabler, security and trust, and two not needed enablers, leaving sensing, positioning, and time synchronization as the next most consistently demanded building block. Both enablers turn out to be the ones the white paper itself ties most directly to existing deterministic industrial networking [FD24], and both connect concretely to TSN, OPC UA, and IEC 62443 rather than to a generic 6G security or sensing concept. Read against Underberg et al. [Un25]'s usability and cost findings, this technical priority list translates into a specific roadmap implication for automation SMEs, that new 6G capabilities should be delivered through the standards-aligned tooling SMEs already use for TSN and OPC UA, rather than through a parallel 6G-specific configuration and security stack.

The limitations noted in Section 6 point directly to future work. The same threshold based methodology and technical-depth treatment applied here to the top two enablers can be extended to the remaining entries in Figure 1, and the SME framing in Section 5, currently a reasonable inference from Underberg et al. [Un25]'s aggregate findings, would benefit from a dedicated, company size stratified survey. Both extensions would narrow the gap between the roadmap derived here and one validated directly against automation SMEs themselves.

References

[FD24] Franchi, N.; Dressler, F., eds.: German Perspective on 6G – Use Cases, Technical Building Blocks and Requirements, Insights by the 6G Platform Germany. White Paper. FAU University Press, Erlangen, 2024.

[IE09] International Electrotechnical Commission: IEC TS 62443-1-1:2009 – Industrial Communication Networks, Network and System Security, Part 1-1, Terminology, Concepts and Models, Standard, 2009.

[LN23] Labs Network Industrie 4.0: LNI 4.0 Testbed TSN – Whitepaper OPC UA over TSN, In cooperation with the OPC Foundation and Standardization Council Industrie 4.0, issued 17 April 2023, 2023, https://www.ieee802.org/1/files/public/docs2023/liaison-LNI40-WhitepaperOPCUAoverTSNv17-0423.pdf.

[OP25] OPC Foundation: OPC Unified Architecture – Part 2: Security Model, OPC 10000-2, Release 1.05; also standardised as IEC 62541-2, 2025, https://reference.opcfoundation.org/Core/Part2/v105/docs/.

[Un25] Underberg, L. et al.: From Lessons Learned in 5G to Innovative Solutions in 6G, 6G Platform Germany White Paper, Working Group Innovation Management, 2025, https://www.6g-platform.com/whitepaper/.

Systematic Analysis of MQTT suitability for Asset Administration Shell

Juilee Tikekar [1] and Tobias Kleinert [2]

Abstract: Industrial shop-floor communication increasingly combines HTTP/REST and MQTT, yet the official Asset Administration Shell (AAS) API specification does not define which operations should be exposed over which protocol. This paper proposes a systematic, rule-based methodology for classifying the MQTT suitability of AAS Repository API operations based on their payload size, communication frequency, and interaction pattern. Applying the methodology to the AAS Repository Service Specification Profile 001 (54 operations), a machine-readable protocol assignment specification is derived, comprising MQTT topic templates, QoS levels, retained flags, and payload schemas. The resulting classification provides a reproducible, traceable foundation for hybrid HTTP/MQTT AAS architectures.

Keywords: Asset Administration Shell, MQTT, HTTP, Industrial Communication

1 Introduction

Achieving standardized and interoperable production systems even among different manufacturers is the challenge raised by Industry 4.0 [Ro17]. To address this challenge, the Plattform Industrie 4.0 introduced the concept of Asset Administration Shell (AAS) [Pl16]. The AAS aims to share the digital representation of all the information about an asset in a standardized way, where the asset can be a software or hardware component. According to Reference Architecture Model for Industry 4.0 (RAMI I4.0), Industry 4.0 applications exchange information via various domain-specific formats and protocols, including OPC UA, AutomationML and MQTT [Pl18]. Together along with AAS, these frameworks cover a broad spectrum of use cases and functionalities.

The communication between different systems or components usually happens via interfaces. In case of AAS, the Part 2 of the specifications of AAS (IDTA Number 01002) i.e., Application Programming Interface (API) defines the interfaces and operations to enable the access to AAS, and thereby enabling the information exchange via AAS [In25]. Although the specification is technology-neutral, the operations defined in the specifications follow a resource oriented approach, which is close to general HTTP/REST principle. However, the network and system architectures of systems on the shop-floor are usually constrained and optimized for minimal consumption of energy and other resources. In such scenarios,

[1] Industrial Digital Twin Association (IDTA) e.V., Lyoner Straße 18, 32657 Frankfurt am Main, Germany, juilee.tikekar@idtwin.org, https://orcid.org/https://orcid.org/0009-0000-3842-4739
[2] RWTH Aachen University , IAT, 52065 Aachen, Germany, kleinert@iat.rwth-aachen.de, https://orcid.org/https://orcid.org/0000-0001-7441-4431

the industrial machine-to-machine communication protocols such as MQTT are used to provide a light-weight data exchange format focusing on reliable data transfer. The event-driven publish-subscribe model of MQTT can overcome the challenges imposed by request-response model of HTTP, such as inefficient polling for frequently changing values (e.g., sensor data), higher network load through redundant requests, delays in detecting state changes, and scalability problems with many simultaneous clients. At the same time, it should also be noted that MQTT does not offer advantages for all the AAS operations. Complex queries with filtering and pagination are more efficient over HTTP. Thus, it is important to study a formalized methodology for deciding which operations should be mapped over which protocol.

Therefore, this research work proposes a systematic approach for analyzing MQTT suitability of each AAS operation based on industrial application scenarios and programmatically generate a machine-readable protocol assignment specification consisting of MQTT configurations. The rest of the paper is structured as follows: Section 2 reviews related work, section 3 describes relevant uses cases considered during the work, section 4 presents the implementation of the proposed methodology, followed by section 5 discusses the results of the methodology.

2 State of the Art

This section reviews existing approaches for accessing the AAS over HTTP/REST and MQTT, and prior work on hybrid architectures. Section 2.1 covers current AAS implementations and their MQTT extensions, whereas section 2.2 discusses hybrid protocol approaches and positions the proposed work against them.

2.1 AAS over HTTP/REST and MQTT

Prior work has focused on realizing AAS access as type 2/ reactive AAS based on IDTA AAS specifications. These standards also provide OpenAPI specifications [25], which define HTTP/REST APIs for AAS as the primary interoperability interface [In25]. The current available implementations i.e., open-source as well as commercial solutions mostly focus on the HTTP/REST implementation of these operations [Ja23][26b], sometimes with MQTT based extensions in isolation. For instance, Eclipse BaSyx's MQTT support publishes hierarchical events for a fixed set of repository-level structural events (e.g., shell and submodel-reference create/update/delete), using full-object payloads such as the complete AAS JSON representation [26c]. This may result in comparatively larger payloads, which is a relevant constraint in bandwidth limited shop-floor networks. On the other hand, Eclipse FA^{3}ST uses implementation-specific eventing types and payload, leading to a vendor-lock-in challenge [26a]. Additionally, these implementations provide the functionality to communicate with the asset via MQTT with the features like 'DataBridge'

by Eclipse BaSyx [Zi23] or 'AssetConnection' by Eclipse FA^{3}ST [Ja22]. However, none of these implementations address protocol selection based on interaction pattern. In addition to these implementations, there have been a few attempts, e.g., [PBD20] [PBD21], made to exchange AAS via MQTT. However, in these approaches, topics and payloads are application-specific, and not derived from AAS API structure.

2.2 Protocol Assignment and Hybrid Approaches

Various authors have explored hybrid protocol approaches in AAS-related as well as AAS-adjacent work. The authors presented an approach where MQTT is used alongside HTTP for peer-to-peer AAS interactions [Gr22], but the approach lacks a systematic per-operation assignment. In the research work [Ol26], a microservice-based RAMI4.0 architecture combines AAS-based Submodels with MQTT publish-subscribe and HTTP/REST communication protocols through standardized interfaces. Although this approach uses standard AAS-compliant payload as MQTT messages, the full-object payload may lead to network inefficiency. Additionally, the approach leaves the MQTT configurations (like QoS or retained flag) unstandardized, deferring them to implementations, which may lead to interoperability challenges.

However, a study from 2012 [Ro12] proposes a hybrid communication model and middleware that selects request-response or publish-subscribe based on message attributes and quality properties. However, the approach works at generic middleware or message level, not at the level of standardized AAS API operations.

These different studies and implementations combine both the communication paradigms, but they typically expose them as independent protocol options rather than leveraging a deterministic protocol assignment specification based on their respective strengths. Thus, the proposed research fills the missing AAS-specific, operation-level, industrial protocol assignment layer deduced from industrial application scenarios, and protocols' strengths and weaknesses.

3 Use Cases and Operation Classification

This section describes common use cases and industrial application scenarios and how operations can be classified between HTTP and MQTT protocols based on the interaction type of these scenarios.

Industrial application scenarios differ mainly in terms of payload size, communication frequency and interaction type, which together drive the choice of protocol. In case of interactions, where the payloads are small and frequency is high, an event-driven mechanism is suitable. For instance, if a temperature sensor in a machine continuously generates values and multiple systems want to access this data, request-response format becomes inefficient

as it involves polling from multiple clients. On the other hand, use-cases like file transfers or accessing devices via engineering tools, which involve large payloads and are mostly human-driven, are best served with request-response type of interaction pattern.

Although the nature of some use cases like operation invocation and parameterization fits the request-response pattern well, they can still benefit from event-driven protocols like MQTT, provided a request-response feature, such as that offered in MQTTv5 [OA19], is used to ensure the client is acknowledged. Table 1 in Section 4.1 summarizes these scenarios together with their frequency, payload, interaction type and the corresponding AAS API endpoints and MQTT suitability.

4 Implementation

This section describes how the protocol assignment specifications for MQTT were derived from current AAS Specs in OpenAPI format. For simplicity, the section makes use of AAS repository Service Specification Profile 001 [In25] [25]. However, the actual implementation covers both AAS and Submodel Repository Service Specification profiles and corresponding results are discussed in the section 5.

4.1 Assignment of scenarios and requirements to AAS API Endpoints

The Table 1 summarizes the AAS entities (e. g. Submodel, Submodel-element, AAS Operation, etc.) involved and the most suitable AAS API operation endpoints as per the application scenarios described in Section 3. For scenarios such as device and sensor monitoring or alarms and events, the value-only endpoint is preferred for MQTT mapping, as it results in small, frequent payloads efficient to publish, whereas the complete JSON-serialized element or its metadata remain suitable for a one-time retrieval via HTTP. In case of parameterization and setpoint updates, the PATCH endpoint with value-only characteristic is mapped to MQTT, provided the update can be verified using the request-response feature of MQTTv5. In contrast, scenarios like device discovery or engineering tools and administration involve pagination, complex queries and filters, which are more efficient over HTTP, hence these are marked as unsuitable for MQTT in the table.

4.2 Decision Criteria for Operation Classification

Below decision matrix (Table 2) is used to programmatically analyze each AAS API operation using OpenAPI Specifications (YAML) for AAS Repository and to assess if the operation is suitable for MQTT mapping. The suitability is returned in three terms i. e. "suitable", "unsuitable" and "conditional". "conditional" represents such operations, which are efficiently executed in HTTP, and notified over MQTT upon successful execution.

Scenario	AAS Entity	AAS Endpoint	Frequency	Payload	Interaction Type	MQTT Suitability
Device and sensor monitoring	Submodel-element	GET /shells/{aasId}/submodels/{submodelId}/submodel-elements/{idShortPath}/$value	High	Small	Event-driven	suitable
Parameterization and setpoint update	Submodel-element	PATCH /shells/{aasId}/submodels/{submodelId}/submodel-elements/{idShortPath}/$value	Low	Small	Request-Response	suitable
Operation invocation	AAS Operation	POST /shells/{aasIdentifier}/submodels/{submodelIdentifier}/submodel-elements/{idShortPath}/invoke/$value; GET .../operation-status/{handleId}; GET .../operation-results/{handleId}/$value	Low	Small–medium	Request-Response	suitable (Async)
Alarms and events	Submodel-element	GET /shells/{aasId}/submodels/{submodelId}/submodel-elements/{idShortPath}/$value	Low–medium	Small–medium	Event-driven	suitable
Diagnostic info and health status	Submodel-element	GET /shells/{aasId}/submodels/{submodelId}/submodel-elements/{idShortPath}/$value	On-demand/ periodic	Small–medium	Event-driven / Request-response	suitable
File transfer	AAS File element	GET/PUT/DELETE /shells/{aasId}/submodels/{submodelId}/submodel-elements/{idShortPath}/attachment	Low	Large	Request-response	conditional
Peer-to-peer device interaction	Submodel-element	GET /shells/{aasId}/submodels/{submodelId}/submodel-elements/{idShortPath}/$value	Medium	Small	Request-response/ event-driven	suitable
	AAS Operation	POST /shells/{aasIdentifier}/submodels/{submodelIdentifier}/submodel-elements/{idShortPath}/invoke/$value and other operation related endpoints				suitable
Device discovery	AAS, Submodel	GET /shells	Low	Medium	Request-Response	unsuitable
Tools and administration	AAS	GET /shells	Medium–high	Medium–high	Request-response	unsuitable
	Submodel	GET /shells/{aasId}/submodels/{submodelId}				unsuitable

Tab. 1: Application Scenario, AAS Operation, Requirements and MQTT Suitability

AAS API Operation	MQTT Suitability	Interaction Type	Reason
GET (single resource, value-only, except AAS Metamodel)	suitable	Event-driven	Read current values, small payloads, high frequency
GET (GetAll, query parameters)	unsuitable	Request-response	Pagination, filtering → HTTP efficient
POST (Operation invocation)	suitable	Request-response	Async execution pattern
POST (create resource)	conditional	Request-response	Large payload → HTTP execution + MQTT notification
PUT (replace resource)	conditional	Request-response	Large payloads → HTTP execution + MQTT notification
PATCH (update value)	suitable	Request-response	Small payloads, low frequency
DELETE (delete resource)	conditional	Event-driven	HTTP execution + MQTT notification
/attachment (file operations)	conditional	Request-response	HTTP execution + MQTT notification
Serialization modifiers	unsuitable	Request-response	Complex queries, large payloads → HTTP efficient

Tab. 2: Decision Matrix for AAS API operation classification

4.3 Decision Criteria for MQTT Bindings

Standardizing MQTT topics and configurations such as Quality of Service or retained flag is necessary to ensure system interoperability across shop-floors. If these settings are not standardized, different systems may apply different settings, thereby making the communication inefficient and non-interoperable, leading to data silos and routing chaos. Additionally, standardizing these configurations ensures predictable network reliability, prevents data loss, and avoids system-wide failures by matching message delivery guarantees to exact operational requirements.

- **Topic Template:**
  MQTT topics are hierarchical strings that route messages between publishers and subscribers. In the proposed work, the HTTP dominated endpoints of AAS API operations are mapped as is to the MQTT topic template. However, special cases like value-only serialization, the patch or delete operations, operation invocation, and file operations, add a suffix such as "/value", "/update" or "/delete", "/invoke", and "/attachment" respectively.
- **Quality of Service**
  QoS in MQTT represents a formal agreement between publisher and subscribers, so

that it can be ensured that the critical messages reach their destination appropriately based on their importance. There are three QoS levels defined by MQTT i. e. QoS 0 = at most once (fire and forget, no acknowledgment), QoS 1 = at least once (duplicates are acceptable), QoS 2 = exactly once (critical commands) [OA19]. Out of these levels, the suitable ones are decided based on if the message loss is acceptable. In the proposed work, QoS 0 is not considered, as it imposes unreliable configuration for industrial scenarios. Thus in case of AAS APIs, operations related to operation invocations and file-transfers can be categorized to have QoS 2 due to their criticality, whereas rest of the operations can be configured with QoS 1 creating the balance between performance and reliability.

- **Retained flag**
  MQTT retained flag instructs the broker to store the last-known message on a topic, so that the latest relevant information is available regardless of publishing frequency. Thus, the new subscribers can receive the current state instantly. In the proposed work, in case of protocol assignment, all the value-only AAS API operation to read the current state are assigned to have retained value as "true". Additionally, all the file-operations and result of operation invocation are also marked as retained. The rest of the AAS API operations in MQTT are not suitable for retaining.

- **Payload schema**
  Similar to MQTT topics and configurations, standardizing payload schema in factories is important to ensure device interoperability, simplify data integration and reduce maintenance costs. The AAS API operations already consist of standardized payload schema. Therefore, the proposed work reuses the same schema in case of identified suitable operations. The official schema does not include timestamps. However, including timestamps in payload ensures data integrity by notifying the recipient exactly when the data was generated. Therefore, in this work, the timestamp has been added as a part of extended schema to the standard AAS schema.

- **Publisher/Subscriber Role Assignment**
  Unlike HTTP, MQTT does not preset the client/server roles, instead it relies on publish/subscribe model. In the industrial scenarios, without specifying which side publishes to a given topic, two independent implementations of the same mapping specification could both try to publish or only subscribe, thereby breaking interoperability. In the proposed work, the entity hosting AAS APIs (e. g., edge gateway) acts as publisher enabling the read access to the current values, alarms and operation results, while consuming applications (e. g., HMI, MES, analytics) subscribe to these topics. On the other hand, for the write operations such as parameterization or operation invocation, these roles invert. The consuming applications invoke the requests and the AAS hosting entity subscribes, executes the operation and publishes the result on the response topic as per MQTTv5. In the implementation, the terms "server" and "client" denote the AAS-hosting entity and the consuming application, respectively.

4.4 Algorithm for Protocol Assignment Specification

Algorithm 1: Derivation of MQTT Binding from Classified AAS API Operation

Input: Operation op (path, method, category) from the AAS Repository API
Output: MQTT binding $(topic, qos, retained, schema, role)$ for suitable operations

$suitability \leftarrow$ `Classify(`op`)` ; `// Decision matrix, Tab. 2`
if $suitability$ = unsuitable **then**
  return $\emptyset$;
$topic \leftarrow$ `DeriveTopic(`$op.path, op.method$`)` ; `// Topic Template 4.3`
$qos \leftarrow$ `DeriveQoS(`$op.category$`)` ; `// QoS 4.3`
$retained \leftarrow$ `DeriveRetained(`$op.category, op.method$`)` ; `// Retained Flag 4.3`
$schema \leftarrow$ `DerivePayloadSchema(`$op.schema$`)` ; `// Payload Schema 4.3`
$role \leftarrow$ `DeriveRole(`$op.category, op.method$`)` ; `// Pub/Sub Role Assignment 4.3`
return $(topic, qos, retained, schema, role)$;

Algorithm 1 is used to programmatically generate the protocol assignment specification based on standard AAS API operations (OpenAPI). For this algorithm the YAML format of the AAS Specifications is provided as input. The algorithm, first of all, marks the suitability of every AAS API operation defined in this YAML file using the decision matrix described in Table 2. For each of the 'suitable' and 'conditional' operations, it then derives the topic template, MQTT configurations like QoS, retained flag and publisher/subscriber role, and payload schema as described in Section 4.3. This generated mapping specification for every operation is then stored in a JSON file so as to apply it as a protocol assignment specification in the systems.

4.5 Generated Protocol Assignment Specification

```
{
        "operationId": "GetSubmodelElementByPath-ValueOnly_AasRepository",
        "mqtt_suitable": "suitable",
        "topic_template": "shells/{aasIdentifier}/submodels/{
            submodelIdentifier}/submodel-elements/{idShortPath}/value",
        "mqtt_config": {"qos": 1, "retained": true, "role": "server-to-
            client"},
        "mqtt_payload_schema": {"schema": "...SubmodelElementValue", "
            extensions": {"timestamp": {"type": "string", "format": "date-
            time"}}}
}
```

List. 1: Generated MQTT binding for the device/sensor monitoring scenario (cf. Table 1)

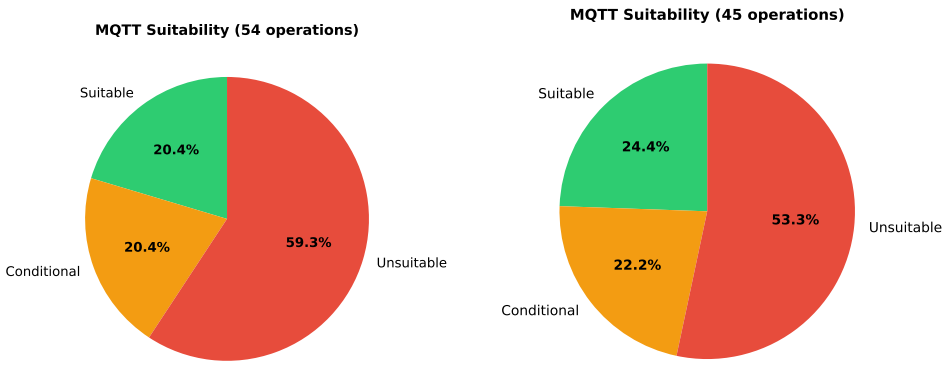
Listing 1 shows the result of protocol assignment specification for one of the AAS API Operations, which is marked as suitable for MQTT in case of device or sensor monitoring scenario as per Table 1. The the additional attributes such as "operationId" and "mqtt_suitable" has been added to the specification for traceability. A complete protocol assignment specification, in the end, consists of such individual mappings and configurations for all the operations defined in original AAS API Specification.

5 Results and Discussion

This section presents the classification results and outlines the direction for future work.

5.1 Results

Figure 1 shows the suitability distribution for the AAS Repository (54 operations) and Submodel Repository (45 operations), derived using the algorithm from Section 4.4 and the decision matrix in Table 2. For the AAS Repository, 11 operations (20.4%) are suitable, 11 (20.4%) are conditional, and 32 (59.3%) are unsuitable. For the Submodel Repository, 11 of 45 operations (24.4%) are suitable, 10 (22.2%) are conditional, and 24 (53.3%) are unsuitable.



(a) AAS Repository (54 operations) (b) Submodel Repository (45 operations)

Fig. 1: MQTT suitability distribution across AAS Repository and Submodel Repository operations

Both specifications show a similar split. Roughly 60% of the operations are unsuitable for the MQTT in both cases, mainly operations requiring pagination, complex filtering, or static metadata access, which HTTP handles more efficiently. The Submodel Repository has a slightly higher suitable share because its API surface is centered on submodel-elements and their values. Value-only and partial-update operations dominate the suitable class, and the

Submodel Repository contains proportionally more of them. Although only about one-fifth operations are classified as suitable, these correspond to the high-frequency, small-payload interactions that dominate typical shop-floor traffic, such as sensor monitoring, alarms, and setpoint updates. The majority of unsuitable operations are low-frequency administrative or query operations.

The methodology offers three advantages over ad hoc or manually curated MQTT eventing implementations. First, classification and binding generation are derived directly from the official AAS OpenAPI specification. The protocol assignment is therefore machine-readable, reproducible, and can be regenerated automatically whenever the specification changes, unlike the fixed, hand-picked event sets described in Section 2. Second, the decision matrix gives a transparent, traceable justification for each assignment. Existing solutions embed the choice of eventable operations implicitly in code rather than in a documented, reviewable ruleset. Third, the methodology systematically derives MQTT-specific configurations: QoS, retained flag, topic template, payload schema, and publisher/subscriber role, as described in Section 4.3. This standardizes MQTT bindings across independently developed systems and avoids interoperability issues that arise when each system applies its own configuration. Finally, the generated specification (Listing 1) is directly reusable. Other systems or tools can apply it to configure their AAS-MQTT interfaces without re-implementing the decision logic.

5.2 Future scope

The proposed methodology forms a foundation for several future extensions.

- **Application in Hybrid Architecture:** The generated protocol assignment specification can be integrated into a hybrid HTTP/MQTT architecture, wherein a routing policy engine dynamically routes suitable operations over MQTT and the remaining operations over HTTP, based on the assignment derived in this work.
- **Use of AI for spec generation:** Given the rule-based nature of the decision matrix, future work can explore AI- or LLM-based methods to further refine or extend the classification, e. g. for other AAS API profiles or to handle edge cases not covered by the current rule set.
- **Protocol-neutral generalization:** Since the underlying methodology separates suitability classification from protocol-specific binding synthesis, the approach can potentially be generalized to other machine-to-machine protocols, such as OPC UA, or to standardized event envelopes, such as CloudEvents, in future work.

References

[25] OpenApi Specifications of Asset Administration Shell, https://app.swaggerhub.com/search?owner=Plattform_i40, Accessed: 03.04.2026, 2025.

[26a] Eventing in FA^{3}ST, https://faaast-service.readthedocs.io/en/latest/interfaces/message-bus.html, Accessed: 31.08.2026, 2026.

[26b] IDTA Solution Hub, https://industrialdigitaltwin.org/solutions-hub, Accessed: 06.04.2026, 2026.

[26c] MQTT Eventing in Eclipse BaSyx, https://wiki.basyx.org/en/latest/content/user_documentation/basyx_components/v2/aas_repository/features/mqtt.html, Accessed: 31.08.2026, 2026.

[Gr22] Grunau, S. et al.: The Implementation of Proactive Asset Administration Shells: Evaluation of Possibilities and Realization in an Order Driven Production. In (Jasperneite, J.; Lohweg, V., eds.): Kommunikation und Bildverarbeitung in der Automation. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 131–144, 2022.

[In25] Industrial Digital Twin Association: Specification of the Asset Administration Shell Part 2: API, tech. rep. IDTA-01001-3-1, IDTA, 2025.

[Ja22] Jacoby, M. et al.: FA3ST Service – An Open Source Implementation of the Reactive Asset Administration Shell. In: 2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA). Pp. 1–8, 2022.

[Ja23] Jacoby, M. et al.: Open-Source Implementations of the Reactive Asset Administration Shell: A Survey. Sensors 23, p. 5229, 2023.

[OA19] OASIS: MQTT Version 5.0, OASIS Standard, https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html, OASIS, 2019.

[Ol26] Oliveira, V. F. d. et al.: Microservice-Based Implementation of the Reference Architectural Model for Industry 4.0. IEEE Access 14, pp. 55877–55890, 2026.

[PBD20] Pribiš, R.; Beňo, L.; Drahoš, P.: An Industrial Communication Platform for Industry 4.0 - case study. In: 2020 Cybernetics & Informatics (K&I). Pp. 1–9, 2020.

[PBD21] Pakala, H. K.; Belyaev, A.; Diedrich, C.: Middleware Architecture for Application Layer Interoperability of Standardized Digital Representations. In: IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society. Pp. 1–6, 2021.

[Pl16] Plattform Industrie 4.0: Implementation Strategy Industrie 4.0: Report on the Results of the Industrie 4.0 Platform, https://www.bitkom.org/sites/default/files/file/import/2016-01-Implementation-Strategy-Industrie40.pdf, Accessed: 06.04.2026, 2016.

[Pl18] Plattform Industrie 4.0: Platform Industrie 4.0: RAMI4.0: A Reference Framework for Digitalisation, https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/rami40-an-introduction.pdf, Accessed: 31.08.2026, 2018.

[Ro12] Rodríguez-Domínguez, C. et al.: A Communication Model to Integrate the Request-Response and the Publish-Subscribe Paradigms into Ubiquitous Systems. Sensors 12 (6), pp. 7648–7668, 2012, https://www.mdpi.com/1424-8220/12/6/7648.

[Ro17] Rojko, A.: Industry 4.0 Concept: Background and Overview. International Journal of Interactive Mobile Technologies (iJIM) 11 (5), pp. 77–90, 2017, https://online-journals.org/index.php/i-jim/article/view/7072.

[Zi23] Zielstorff, A. et al.: Overcoming Challenges in Integrating Legacy Devices with Asset Administration Shells - An OPC UA Case Study. In: 2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA). Pp. 1–8, 2023.

Seamless & Secure IT/OT Integration with PROFINET

Andreas Walz[1] and Oliver Pfaff[1]

Abstract: Security is an omnipresent concern. At the same time, the demand for integrating Operational Technology (OT) with Information Technology (IT) continues to grow. PROFINET addresses both challenges, as reflected in the latest PROFINET specification V2.5. The specification provides the technological foundation for scalable, secure, and future-proof automation solutions spanning from the field level to enterprise IT systems. In this article, we offer two contributions. First, we examine the architecture and features of PROFINET Security as of specification V2.5. Second, we outline potential enhancements for a future version of the PROFINET specification that strengthen support for emerging communication scenarios arising from IT/OT integration. This article reflects and contributes to work conducted within the Security Working Group (CB/PG10) of PROFIBUS & PROFINET International.

Keywords: cyber security, identity and access management, industrial communication, IT/OT integration, PROFINET

1 Introduction

PROFINET is an open Industrial Ethernet communication technology for automation use cases, widely used to interconnect controllers, distributed field devices, drives, safety components, engineering tools, and supervisory systems within industrial automation networks. As an Ethernet-based technology, it combines real-time automation communication with integration into broader network infrastructures. With the latest specification V2.5 [PN26], PROFINET takes the next step from a field-level communication system towards a next-generation open industrial communication platform, with major improvements in areas such as (cyber)security and IT/OT integration [PN25]. While *Information Technology* (IT) focuses on business applications and enterprise services, *Operational Technology* (OT) focuses on facilitating, monitoring and controlling industrial processes. The integration of both domains enables new data-driven and operational use cases across industrial systems [NIST23].

In industrial automation, communication relationships are often described as either *vertical* or *horizontal*, depending on the relative position of the communicating entities in the automation hierarchy [Se03]. PROFINET communication that occurs between IO controllers and IO devices, including cyclic real-time IO exchanges, can be considered horizontal. Vertical PROFINET communication, on the other hand, occurs between

[1] Siemens AG, Digital Industries, Siemenspromenade 1, 91058 Erlangen, Deutschland, {andreas-christoph.walz, oliver.pfaff}@siemens.com

PROFINET-enabled software tools or services on one end and IO controllers or IO devices on the other end. Typical use cases are, for example, reading diagnostics data, managing parametrization, and performing asset management functions.

IT/OT integration is predominantly realized through vertical communication mechanisms. Horizontal communication remains essential for process control, while vertical communication extends the OT domain into enterprise and cloud-based IT ecosystems.

With the application classes of *Secure Realtime* and *Secure Access*, the PROFINET specification V2.5 provides the technological foundation for secure horizontal and secure vertical PROFINET communications [PN26a]. This includes scalable end-to-end cryptographic protection and role-based access control to address omnipresent security requirements such as authentication, authorization, integrity, and confidentiality (see [IEC62443]). In Section 2 of this article, we will provide more detailed information about security in PROFINET V2.5.

OT-specific security solutions are generally acceptable when vertical communications play only a minor role. However, as IT/OT convergence progresses, enterprise IT entities will increasingly take part in communications with OT entities. For these use cases, authentication and authorization must be aligned with established corporate solutions, especially *Identity and Access Management* (IAM). The demand for such integration is raised by the NAMUR Recommendation NE 201 [NA25]. The recommendation addresses IAM for industrial automation systems and promotes token[2]-based authentication and authorization features.

To respond to this demand, Section 3 sketches how enterprise IAM infrastructures can be leveraged in conjunction with the PROFINET specification V2.5. This approach requires new, PROFINET vendor-specific add-on components in the IT infrastructure of PROFINET technology users. Moreover, the approach's authorization capabilities remain limited to the mechanisms available in PROFINET V2.5. In Section 4, we present an alternative in which PROFINET endpoints directly consume tokens issued by enterprise IT infrastructures. We discuss how this approach can address the limitations and outline enhancements for a future PROFINET specification version that would avoid these limitations. This article reflects and contributes to work conducted within the Security Working Group (CB/PG10) of PROFIBUS & PROFINET International.

We end with a summary, conclusion, and outlook in Section 5.

[2] Within our article, a token is understood as a data object that a claimant uses to make assertions (for example, identity or access privileges) about itself towards a verifier.

2 PROFINET Security as of Specification V2.5—in a Nutshell

In this chapter, we outline the key concepts of PROFINET specification V2.5, with an emphasis on security.

With PROFINET specification V2.5, the *Service Exchange Protocol* (SXP) is introduced as a versatile carrier for connection management, record services, and, optionally, alarm and/or IO data for PROFINET. SXP is transport-agnostic, with transport bindings currently specified for TCP/IP and Layer 2 (Ethernet). This allows both IP-based routing and architectural support for software-defined and virtualized automation as well as vertical communication.

Security is covered by the following categories of features.

- **Cryptographic security** based on state-of-the-art standards is provided for communication taking place with PROFINET *Application Relations* (ARs), turning ARs into *secure ARs*. This includes AR endpoint authentication via X.509 certificates (LDevID credentials according to IEEE 802.1AR [IEEE8021AR]) based on TLS as well as communication integrity protection and confidentiality as determined by local configuration.
- **Role-based access control** allows preventing unauthorized access to PROFINET resources. Access decisions are made by the responding endpoint of a secure AR based on the role(s) assigned to the requesting AR endpoint.
- **Auditable events** are recorded and disseminated by PROFINET stations to enable central security monitoring and forensics.
- **Hardening measures** help to limit the attack surface by allowing the user to enable/disable access through unprotected protocols and services via local configuration, as required.

Different use cases demand different levels of security. To accommodate these varying needs, three conceptual stages are defined, each requiring the implementation of a specific set of security capabilities and providing corresponding security properties [PNsec].

- **Secure Cell** is the basic concept for security. It is based on controlled and protected network and physical boundaries around the automation cell and enables continued use of existing equipment without requiring device-level security mechanisms.
- **Secure Realtime** features cryptographically protected horizontal PROFINET communications (i.e., protected PROFINET communication within automation cells).
- **Secure Access** features cryptographically protected vertical PROFINET communications (i.e., protected access to a PROFINET station from outside of a secure cell).

In combination, Secure Cell, Realtime and Access provide a comprehensive set of technical features for addressing the security requirements of current PROFINET use cases, including real-time communication and tool access.

3 Secure vertical communication with PROFINET V2.5

In this section, we describe how existing enterprise IAM infrastructure can be leveraged with the security mechanisms of PROFINET specification V2.5.

Before turning to the details, it is useful to distinguish between two fundamentally different security features.

1. **Authentication** takes a *subject-centric* perspective. It is concerned with statements about authentication events, which may be expressed in the form of **ID tokens**.
2. **Authorization** takes a *resource-centric* perspective. It is concerned with statements about access privileges, which may be expressed in the form of **access tokens**.

These two perspectives are not disjoint; they complement each other. However, the current mainline technologies in IAM focus on one of them:

- The **OAuth 2.0** framework [RFC6749] specified by the *Internet Engineering Task Force* (IETF) focuses on authorization.
- **OpenID Connect** [OIDC23] specified by the *OpenID Foundation* (OIDF) extends OAuth 2.0 and focuses on authentication in its extension.

In the following, we refrain from adopting one perspective and keep the discussion independent of this distinction. Therefore, we use the term “token” as an umbrella term for access tokens and ID tokens and also for *bearer* and *sender-constrained* tokens. Independently from these form factors: token objects in IAM and certificate objects in X.509 are orthogonal data structures. Producers and consumers of tokens cannot produce or consume certificates, respectively, and vice versa.

In the following, we consider the following setting:

- A ***Device Access Tool*** **(DAT)** wants to access a protected resource hosted by an IO controller or IO device (collectively called IOX) located within a PROFINET automation cell using PROFINET communication (i.e., the resource access shall happen via a secure PROFINET AR). The DAT may be a software tool for engineering, commissioning, diagnostics, etc.
- The DAT integrates with existing enterprise infrastructure for *Identity and Access Management* (IAM)[3]. This infrastructure hosts a ***Trusted Third Party*** **(TTP)**, which

[3] This assumes that human-operated DATs can participate in the envisaged authentication schemes for human users, either through native capabilities or by relying on external authentication mechanisms.

is based on off-the-shelf products/services, and which can issue tokens based on established IT standards. For example, the TTP could be an OAuth [RFC6749] authorization server issuing *JSON Web Tokens* (JWTs) [RFC7519]. The TTP is assumed to operate outside the PROFINET automation cell.

- The **IOX** acts as PROFINET responder for requests coming from the DAT. The IOX uses PROFINET Security (i.e., it accepts secure PROFINET ARs).

This setting suffers from an obvious mismatch: while the DAT is assumed to acquire tokens issued by enterprise IAM infrastructure to convey authentication and/or authorization information to the IOX, the IOX expects the DAT to authenticate and authorize itself using an LDevID credential, as mandated by PROFINET specification V2.5 (see Section 2). To bridge this gap, an additional entity can be used: an **LDevID[4] issuer** that serves as an intermediary which exchanges tokens issued by enterprise IAM infrastructure for LDevID credentials to be presented to IOXs. The LDevID issuer issues an LDevID credential to a DAT upon presentation of a token that authorizes the DAT to obtain such credential. The LDevID issuer could, for example, be implemented as a server using the *Enrollment over Secure Transport* (EST) protocol [RFC7030].

In this setting, we assume that the DAT initially

- neither possesses a valid LDevID credential for the intended resource request nor a token valid to obtain an LDevID credential from the LDevID issuer, and it
- neither knows the LDevID issuer nor the TTP.

The interaction between these entities proceeds as follows:

1. **DAT → IOX**: attempt to access a protected resource via PROFINET (by sending an SXP request) without presenting a valid LDevID credential. Note: the DAT may skip this step if it can determine on its own that it does not possess an LDevID credential valid for the intended request.
2. **IOX → DAT**: reject the access attempt, either because DAT authentication fails during secure AR establishment (if the DAT cannot present a valid LDevID) or because the presented LDevID provides insufficient permissions for the request.
3. **DAT**: determine an LDevID issuer (for example, via DNS-based service discovery [RFC6763] or by static configuration).
4. **DAT → LDevID issuer**: request an LDevID credential via EST without presenting a valid token.

[4] Data structures that build upon X.509 certificate objects and that are locally significant, i.e., their certificate portion is created by a local issuer. LDevIDs are also used by claimants to make assertions about themselves towards verifiers but are orthogonal to tokens (in the above-mentioned sense).

5. **LDevID issuer → DAT**: provide an error response, informing the DAT that a valid token is required for LDevID issuance, including details of the expected token and its issuing authority (i.e., the TTP).
6. **DAT → TTP**: request a token that matches the given expectation and that allows the DAT to obtain an LDevID credential from the LDevID issuer.
7. **TTP → DAT**: return a token.
8. **DAT → LDevID issuer**: request an LDevID credential via EST using the previously acquired token.
9. **LDevID issuer → DAT:** return an LDevID credential.
10. **DAT → IOX**: re-attempt to access the protected resource via PROFINET using the previously acquired LDevID credential.
11. **IOX → DAT**: serve the resource access request.

The entities involved in the procedure and their relations are illustrated in Fig. 1.

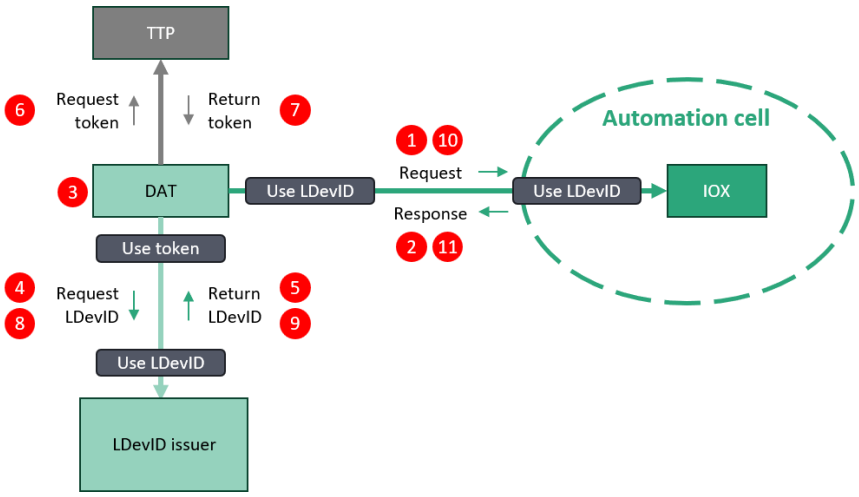


Fig. 1: Illustration of the entities involved in a PROFINET resource access that leverages PROFINET specification V2.5 and existing enterprise IAM infrastructure.

This procedure demonstrates that existing enterprise IAM infrastructures can be utilized in conjunction with PROFINET based on specification V2.5. However, it also exhibits several limitations.

- With the LDevID issuer, an **additional always-on entity** becomes necessary.

- The LDevID issuer is outside the scope of the PROFINET specification V2.5. Consequently, its implementation is **vendor-specific** for the time being, which hinders interoperable and vendor-independent deployments.
- Access control on the side of the IOX remains **role-based**. There is no option for more sophisticated authorization policies.

In the following Section 4, we present an approach which resolves these limitations.

4 Token-based vertical communication with PROFINET

In this section, we present and discuss an approach in which a PROFINET IOX directly consumes and verifies tokens issued by enterprise IAM infrastructure.

The setting is similar to the one given in Section 3:

- A **DAT** wants to access a protected resource hosted by an IOX within an automation cell via PROFINET.
- The DAT integrates with existing enterprise IAM infrastructure that hosts a **TTP**.
- The **IOX** responds to requests from the DAT but it does not initiate communication by itself (neither to the DAT, nor to the TTP, nor to any other entity). This IOX is assumed to be capable of processing tokens by itself. It is also assumed to be capable of providing DATs with remedy instructions in error cases (missing token, present token that is invalid, present token that is valid but inappropriate for the requested resource).

In this setting, we assume that the DAT initially

- does not possess a valid token for the resource request it intends to make and
- does not know the TTP.

Given the setting and assumptions described above, the interaction between these entities to access a protected PROFINET resource proceeds as follows:

1. **DAT → IOX**: attempt to access a protected resource via PROFINET (by sending an SXP request) without presenting a valid token.
2. **IOX → DAT**: provide an error response, informing the DAT that a valid token is required for the requested resource, including details of the expected token and its issuing authority (i.e., the TTP).
3. **DAT → TTP**: request a token that matches the given expectation and that allows the DAT to access the protected resource hosted on the IOX.
4. **TTP → DAT**: return a token.

5. **DAT → IOX**: re-attempt to access the protected resource via PROFINET using the previously acquired token.

6. **IOX → DAT**: serve the resource access request.

The entities involved in the procedure are illustrated in Fig. 2.

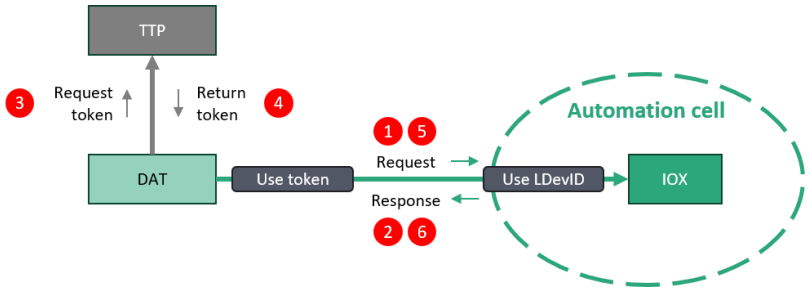


Fig. 2: Illustration of the entities involved in a PROFINET resource access that directly uses tokens to convey authentication and/or authorization information along with the request.

Realizing this procedure requires addressing at least the following two key challenges.

- The PROFINET specification would need to be enabled to support token-based authentication and authorization of PROFINET requestors. This includes conveying tokens as part of PROFINET resource requests to IOX as well as conveying information on expected tokens and their issuer in PROFINET responses from IOXs. This challenge primarily concerns steps 1, 2, and 5.
- The DAT would need to be enabled to interact with off-the-shelf IAM products/services (especially those based on OAuth 2.0 and OpenID Connect). This challenge primarily concerns steps 2, 3, and 4.

In the following, we focus on the first challenge. The second challenge remains within the scope of DAT products and their implementation choices.

- **Step 1 (DAT → IOX)**

  The DAT's resource access attempt is either an SXP request within an existing AR between the DAT and the IOX or an attempt to establish a secure AR.

  As of PROFINET V2.5, authentication and authorization are based on X.509 certificate objects used in a TLS handshake that takes place in the context of secure AR establishment. As a result of PROFINET's use of TLS within the *Extensible Authentication Protocol* (EAP-TLS) [RFC9190], the DAT plays the role of the TLS server. To allow the DAT to convey authentication and/or authorization information to the IOX using tokens *instead* of X.509 certificates, the mandatory TLS server authentication mechanism needs to be bypassed. Ways to achieve this in a standard-

compliant manner exist, for example using *Raw Public Keys* [RFC7250] or self-signed certificates [RFC8705]. The details remain to be determined.

Additionally, the use of bearer tokens (as opposed to sender-constrained tokens) requires a confidential communication channel to prevent token disclosure. Within a secure AR, the acyclic service channel is always cryptographically protected; however, depending on the local configuration, this protection may provide only integrity without confidentiality. The resolution options are i) support bearer tokens and require a confidential record CR for their transfer or ii) require sender-constrained tokens. The resolution details remain to be determined.

- **Between step 1 and step 2 (IOX)**

  If the DAT sends a token with the request, the IOX needs to validate this token before serving the request. This check must include both

  - the token's intrinsic validity (which is determined by, for example, the object's syntactic and semantic sanity, the validity of its signature, whether it is presented within its period of validity) as well as
  - the validity of the token in the context of the given request (which is determined by the authorization policy).

  Checking the signature of a token requires a cryptographic verification key, which is deployment-specific. This results in additional security configuration demand of an IOX to be addressed during its lifecycle.

  To check whether a given token is valid for a given resource request, the IOX must apply the logic of some authorization policy. This authorization policy may be fixed by the PROFINET specification, determined by the vendor of the IOX, or configurable by the owner/operator. The latter option is more flexible than the former options but also results in an additional configuration demand.

  In case the DAT has not sent a token with the request at all, the IOX needs to determine whether the presentation of a token is required to serve the request. If so, or if the token presented by the DAT failed to pass the above checks, the IOX needs to respond to the resource request with an error, providing additional information about the expected token to the DAT (see next step 2 below).

- **Step 2 (IOX → DAT)**

  Since we assumed the DAT did not present a token valid for the request in step 1, the IOX replies with an error response. To allow the DAT to acquire a valid token for a follow-up access attempt, the IOX augments its error response with additional information. This should include:

- **TTP address**: a URL without query parameters[5] through which the DAT can contact the TTP. As the TTP is deployment-specific, this URL is deployment-specific but not request-specific. It results in additional configuration demand for an IOX.
- **Response type** ("`response_type`"): an identification of the token type expected by the IOX (for example, "`id_token`" for authentication and/or "`token`" for authorization). Details depend on whether a subject- or a resource-centric perspective is assumed (see Section 3).
- **Audience**: an identification of the intended token consumer, i.e., the IOX. Profiling the use of this information requires establishing a mapping between entity identifiers in the enterprise IAM and PROFINET namespaces. This may result in additional configuration demand for an IOX.
- **Scope**: an identification of the scope which the required token is expected to include. The IOX should be able to select from a small catalogue of pre-defined scopes, based on the requested resource. This requires harmonization and standardization work on the level of the PROFINET ecosystem.
- **Nonce**: a request-specific unique challenge value that allows the IOX to constrain token usage, either in terms of the number of permitted presentations or the time-period during which token presentation is accepted. Whether this information can be conveyed as a token claim depends on the standards being applied. For example, OpenID Connect defines a dedicated claim for this purpose, whereas OAuth does not.

- **Step 3 (DAT → TTP)**

  This step is not discussed further, as it is assumed to depend on the specific DAT implementation.

- **Step 4 (TTP → DAT)**

  This step is not discussed further, as it is assumed to depend on the specific DAT implementation.

- **Step 5 (DAT → IOX)**

  Details of this step are given in step 1 (see above).

- **Step 6 (IOX → DAT)**

  This step is not discussed further, as it is standard PROFINET behavior.

[5] The absolute path portion of the URL that is typically provided in the "Location" response header of a similar HTTP response.

While several details remain to be elaborated, this procedure shows that it is feasible to directly leverage enterprise IAM infrastructure for PROFINET – if IOX components become token-aware.

Note: The OPC UA specification [OPCUA] already supports token-aware OPC UA server components.

5 Summary and conclusion

We discussed how the PROFINET specification V2.5 supports secure IT/OT integration. We showed how existing enterprise IAM infrastructures can be leveraged today by introducing an LDevID issuer that exchanges IAM tokens for PROFINET credentials. This approach adds a vendor-specific online component and remains bound to the authorization limitations of PROFINET specification V2.5. As a direction for future work, we outlined an alternative in which an IOX directly consumes and verifies tokens issued by enterprise IAM infrastructure.

Bibliography

[IEC62443] International Electrotechnical Commission. *IEC 62443 series: Security for industrial automation and control systems*. IEC.

[IEEE8021AR] IEEE. *IEEE Std 802.1AR-2018: IEEE Standard for Local and Metropolitan Area Networks—Secure Device Identity*. IEEE, 2018.

[NA25] NAMUR. NAMUR Recommendation NE 201: *Identity and Access Management on Automation Devices*. NAMUR, 2025.

[NIST23] Stouffer, K.; Pease, M.; Tang, C.; Zimmerman, T.; Pillitteri, V.; Lightman, S.; Hahn, A.; Saravia, S.; Sherule, A.; Thompson, M. *Guide to Operational Technology (OT) Security*. NIST Special Publication 800-82 Rev. 3, National Institute of Standards and Technology, September 2023.

[OIDC23] Sakimura, N.; Bradley, J.; Jones, M.; de Medeiros, B.; Mortimore, C. *OpenID Connect Core 1.0 incorporating errata set 2*. OpenID Foundation, 15 December 2023.

[OPCUA] OPC Foundation. *OPC Unified Architecture Specification, Part 4: Services*. OPC 10000-4, Version 1.05.07, 21 April 2026. Last accessed: 24 September 2026.

[PN25] PROFIBUS & PROFINET International. *PROFINET V2.5 – A significant step toward the future*. Press release, 24 November 2025. Last accessed: 23 September 2026.

[PN26] PROFIBUS & PROFINET International. *PROFINET Specification: Official Downloads*. PROFINET specification V2.5 documents, including IEC 61158-5-

10, IEC 61158-6-10, IEC 61784-2, and IEC 61784-6-3. Last accessed: 23 September 2026.

[PN26a] PROFIBUS & PROFINET International. *PROFINET Ensures CRA Compliance. PROFINEWS*, 13 February 2026. Last accessed: 23 September 2026.

[PNsec] PROFIBUS & PROFINET International. *PROFINET Security – Comprehensive Concept*. PROFINET Explained, PROFINET.com. Last accessed: 23 September 2026.

[RFC6749] Hardt, D., Ed. *The OAuth 2.0 Authorization Framework*. RFC 6749, IETF, October 2012.

[RFC6763] Cheshire, S.; Krochmal, M. *DNS-Based Service Discovery*. RFC 6763, IETF, February 2013.

[RFC7030] Pritikin, M., Ed.; Yee, P., Ed.; Harkins, D., Ed. *Enrollment over Secure Transport*. RFC 7030, IETF, October 2013.

[RFC7250] Wouters, P., Ed.; Tschofenig, H., Ed.; Gilmore, J.; Weiler, S.; Kivinen, T. *Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)*. RFC 7250, IETF, June 2014.

[RFC7519] Jones, M.; Bradley, J.; Sakimura, N. *JSON Web Token (JWT)*. RFC 7519, IETF, May 2015.

[RFC8705] Campbell, B.; Bradley, J.; Sakimura, N.; Lodderstedt, T. *OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens*. RFC 8705, IETF, February 2020.

[RFC9190] Preuß Mattsson, J.; Sethi, M. *EAP-TLS 1.3: Using the Extensible Authentication Protocol with TLS 1.3*. RFC 9190, IETF, February 2022.

[Se03] Seitz, Matthias. "Speicherprogrammierbare Steuerungen." Fachbuchverlag Leipzig im Carl Hanser Verlag, München (2003).

Acknowledgments

The authors thank the Security Working Group (CB/PG10) of PROFIBUS & PROFINET International for the valuable collaboration and discussions on this topic.

From Unstructured CVE Text to Actionable OT Vulnerability Intelligence: Explainable AI Clustering for Industrial Security and Industrial Communication

Arjun Kalliyath[1] and Dr. David Hülsmeier-Reineke[2]

Abstract: Grouping vulnerability records can make large retrieval results easier to inspect, but high cluster separation alone does not establish real-world usefulness. We examined six clustering pipelines on five product queries in the SRAG analytics system, covering 677 distinct CVE records. Three baselines used TF-IDF or remediation keywords; three hybrid pipelines combined mean-pooled RoBERTa or SecureBERT embeddings with structured weakness and severity features before UMAP and HDBSCAN. The embedding pipelines achieved higher mean assignment coverage than the baselines. RoBERTa had the highest mean coverage and Silhouette score in the main comparison. A separate comparison within the ModernBERT architecture gave mixed results, and whitening changed the encoder ranking on two collections. The findings support hybrid clustering as a candidate organisation layer, while showing that cybersecurity pretraining did not consistently improve the reported diagnostics. Retrieval quality, metadata dependence and preprocessing sensitivity limit the interpretation.

Keywords: operational technology, vulnerability analysis, CVE, hybrid clustering, RoBERTa, SecureBERT, ModernBERT

1 Introduction

Vulnerability review in operational technology (OT) requires more than matching a product name to a CVE identifier. Analysts must establish whether a disclosure applies to the installed version and configuration, then consider the consequences of intervention for an operational system [Ni23]. A retrieval service may return overlapping advisories, repeated descriptions and records for similarly named products. Grouping these records can provide a useful starting point, provided that analysts can still inspect the underlying evidence.

Lexical methods group records that share terms. Transformer encoders offer a different representation of the descriptions, but cybersecurity pretraining does not by itself guarantee a better partition. The outcome also depends on metadata, preprocessing, dimensionality reduction and the density parameters used to form clusters.

We compared two term frequency-inverse document frequency (TF-IDF) baselines, a keyword baseline and three hybrid embedding pipelines. Additional runs compared SecureBERT 2.0 with its ModernBERT base model and examined embedding whitening. We assessed assignment coverage, internal clustering scores and sensitivity to

1 Hilscher Gesellschaft für Systemautomation mbH, Hattersheim am Main, Germany; Technische Hochschule Ostwestfalen-Lippe, In-IT, Lemgo, Germany, akalliyath@hilscher.com, arjun.kalliyath@stud.th-owl.de, ORCID: https://orcid.org/0009-0006-7579-4839

2 Hilscher Gesellschaft für Systemautomation mbH, Hattersheim am Main, Germany, dhuelsmeier@hilscher.com, ORCID: https://orcid.org/0000-0002-4644-0592

preprocessing. The contribution is an empirical assessment of an implemented workflow and its limits.

2 Background and related work

Term frequency-inverse document frequency (TF-IDF) weights terms within and across documents [Sa88]. Combined with a density-based method, it provides a baseline that does not require a pretrained encoder. DBSCAN identifies dense regions using a neighbourhood threshold [Es96]; HDBSCAN builds a density hierarchy and accommodates clusters at different density levels [Ca13]. Both can leave records unassigned. In vulnerability review, these noise records still require attention.

RoBERTa is a general-purpose masked-language model [Li19]. The original SecureBERT adapts a RoBERTa-based model to cybersecurity language and includes tokenizer changes [Ag22]. SecureBERT 2.0 instead builds on ModernBERT [Wa24, Ci26]. We therefore distinguished RoBERTa versus SecureBERT 1.0 from ModernBERT versus SecureBERT 2.0. Comparing RoBERTa directly with SecureBERT 2.0 also changes the architecture and tokenizer. Even within an architecture, differences between complete checkpoints cannot be attributed to one training factor without further controls.

The embedding pipelines used Uniform Manifold Approximation and Projection (UMAP) before density clustering [Mc18], so both the representation and its projection influenced the clustering distances. Contextual language-model embeddings can be anisotropic, with many vectors pointing in similar directions [Et19]. Whitening centres and rescales these representations [Su21]. We used it as a sensitivity check; a change in internal scores does not establish a change in semantic correctness.

3 Data and retrieval scope

We collected CVE records through the CERT@VDE-backed service using five vendor-and-product queries (Cisco IOS XE, Beckhoff TwinCAT, Siemens S7-1500, Yocto Project and ABB Symphony), an empty version filter and a start date of 1 January 2000. Retrieval was capped at 300 records per query. Preparation and identifier deduplication produced five datasets containing 31 to 300 records each, with 677 distinct CVE identifiers in total and no overlap between datasets. Every retained record had a non-empty description. In the retained-record audit, 220 CVEs (32.5%) included Common Platform Enumeration (CPE) entries; 75 of these had only * or - in the CPE version component. This measures the content of the exports, not completeness of the underlying vulnerability database. Vendor and product naming inconsistencies in NVD have also been documented [An22]. We analysed the saved responses; the capped results were not complete product inventories.

4 Methods

4.1 Representations and clustering

Figure 1 shows the two feature paths used in the comparison. The TF-IDF and keyword baselines operated on engineered text containing the description and terms derived from weakness, attack-vector, privilege and impact fields. The embedding pipelines encoded the cleaned description and combined it with a separate 21-dimensional vector derived from Common Weakness Enumeration (CWE) and Common Vulnerability Scoring System (CVSS) fields. The parameter STRUCT_WEIGHT controlled the contribution of this structured vector to the combined representation and was fixed at 0.4 for every encoder. It was a feature-weighting setting, not a measured proportion of the clustering outcome.

Boilerplate and dominant terms, including vendor names and generic terms such as controller, PLC and CPU, were removed during cleaning. The same rules were applied across all five collections, although a term that was boilerplate in one collection could be informative in another. The keyword method assigned records to remediation-oriented buckets by counting matches, subject to a minimum score and category size. Its six possible categories covered updating or patching, reconfiguration, network restrictions, credential hardening, disabling services and workarounds. These labels described matched language rather than verified recommendations.

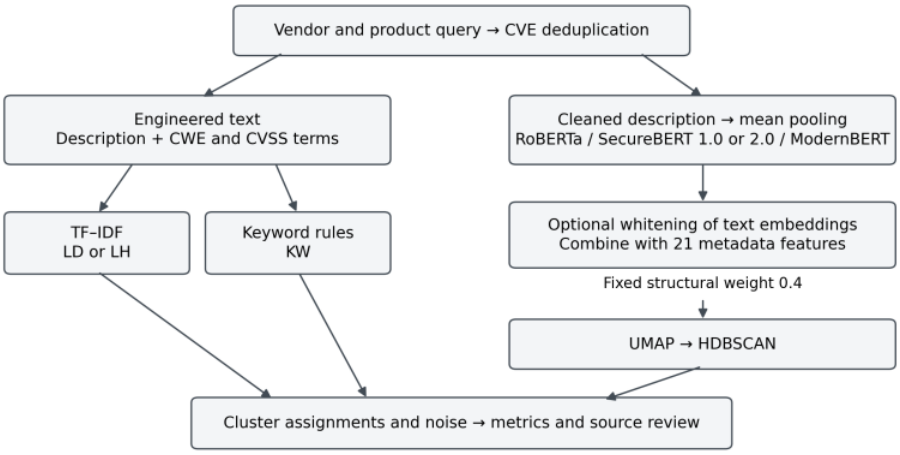


Fig. 1: Feature paths in the evaluated workflow. LD and LH denote TF-IDF with DBSCAN and HDBSCAN; KW denotes keyword rules. Metadata entered both branches, through engineered text on the left and a separate feature vector on the right. Whitening was enabled only in the sensitivity runs.

The main embedding comparison used RoBERTa-base (roberta-base), SecureBERT 1.0 (ehsanaghaei/SecureBERT) and SecureBERT 2.0 (cisco-ai/SecureBERT2.0-base). The separate architecture comparison added ModernBERT-base (answerdotai/ModernBERT-base). Each method used mean-pooled representations, the shared hybrid feature path,

UMAP and HDBSCAN. Whitening was disabled in the main and architecture comparisons and enabled in the sensitivity runs to examine preprocessing effects across the five vendor/product queries.

DBSCAN used EPS = 0.5 and MIN_SAMPLES = 2. HDBSCAN used MIN_CLUSTER_SIZE = 5 and MIN_SAMPLES = 1; UMAP used 15 neighbours and MIN_DIST = 0.1. The keyword rules used a minimum score of 1 and minimum category size of 2. These settings were saved with the runs; the accompanying implementation notes specified UMAP RANDOM_STATE = 42 and the structured-feature configuration.

4.2 Evaluation

For each collection, assignment coverage C was calculated as

$$C = 100 \times (1 - N_{-1}/N) \tag{1}$$

where N is the number of input records and N_{-1} is the number labelled as noise. Coverage ranges from 0 to 100% and describes assignment, not correctness or product applicability. Silhouette ranges from −1 to 1: values near 1 indicate compact, well-separated clusters, values near 0 indicate overlapping clusters, and negative values suggest possible misassignment [Ro87]. The Davies-Bouldin index (DBI) is non-negative; lower values indicate less within-cluster dispersion relative to between-cluster separation, whereas higher values indicate poorer separation [Da79].

We calculated coverage from the saved integer label counts and used the exported internal scores at their available precision. The exports lacked feature matrices and a complete specification of metric distances and noise handling. Silhouette and DBI were therefore treated as diagnostics of the respective pipelines, rather than independently recomputed measurements in a common representation space. CWE labels were not used as an external reference because they already contributed to the input features.

Each reported mean gave the five collections equal weight. For coverage, we first divided the assigned count by the number of records in that collection and then averaged the five percentages. Results are reported as mean ± one standard error across the five collections, with one decimal place for coverage and two for internal scores. These summaries describe variation across the selected collections; they do not estimate variation across random seeds.

Exploratory repeated-measures ANOVAs treated collection as the block and method as the within-collection factor, for both the six main methods and the three main encoders. Before testing, we applied the empirical logit to coverage counts and transformed Silhouette and DBI as follows:

$$z_C = \ln((a + 0.5)/(u + 0.5)) \tag{2}$$

$$z_S = \ln((1 + s)/(1 - s)) \tag{3}$$

$$z_D = \ln(d) \tag{4}$$

Here, a and u are assigned and unassigned counts, s is Silhouette and d is DBI. The 0.5 correction keeps complete-assignment values finite. We applied Greenhouse-Geisser corrections to the ANOVA degrees of freedom. The ModernBERT comparison used two-sided paired t-tests on the same transformed metrics. Holm adjustment covered all nine metric tests, with a significance threshold of 0.05. For the two significant six-method omnibus tests, follow-up paired t-tests compared all 15 method pairs per metric, with a separate Holm correction across these 30 comparisons. With five selected collections, these exploratory tests have limited power and generalisability; internal-score tests compare pipeline diagnostics, not quality in a common feature space.

We also inspected saved cluster keywords and membership overlap on the common ordered inputs to illustrate topic grouping. This exploratory check did not replace a systematic assessment of member descriptions or analyst-labelled semantic quality.

A second saved execution reproduced all 30 main assignment vectors and metric records for identical ordered inputs, confirming repeatability under the same configuration.

5 Results

5.1 Mean performance across collections

Table 1 summarises the method comparisons. The hybrid embedding pipelines had higher mean assignment coverage than either TF-IDF baseline or the keyword method. RoBERTa had the highest mean coverage and Silhouette and the lowest mean DBI in the main comparison. The differences among embedding methods were smaller than their coverage differences from the baselines.

Method	Coverage (%)	Silhouette	DBI
TF-IDF / DBSCAN	81.0 ± 5.2	0.41 ± 0.12	1.26 ± 0.21
TF-IDF / HDBSCAN	72.5 ± 2.9	0.41 ± 0.06	1.51 ± 0.19
Keyword rules	82.8 ± 5.9	0.76 ± 0.07	0.76 ± 0.12
RoBERTa	98.7 ± 0.8	0.88 ± 0.04	0.34 ± 0.06
SecureBERT 1.0	98.2 ± 1.5	0.84 ± 0.05	0.41 ± 0.08
SecureBERT 2.0	97.2 ± 1.3	0.82 ± 0.05	0.45 ± 0.09
ModernBERT	97.7 ± 1.4	0.80 ± 0.08	0.43 ± 0.08

Tab. 1: Mean ± standard error across five collections. Coverage is a percentage. Internal scores describe pipeline-dependent representations. ModernBERT was evaluated in the separate architecture comparison on the same exported CVE objects; the SecureBERT 2.0 results matched the main comparison.

RoBERTa achieved (98.7 ± 0.8)% coverage, compared with (98.2 ± 1.5)% for SecureBERT 1.0 and (97.2 ± 1.3)% for SecureBERT 2.0. The TF-IDF baselines achieved (81.0 ± 5.2)% with DBSCAN and (72.5 ± 2.9)% with HDBSCAN; keyword rules achieved (82.8 ± 5.9)%.

After Greenhouse-Geisser and Holm corrections, the six-method omnibus tests detected differences in coverage ($F(1.59, 6.36) = 14.51$, adjusted $p = 0.042$) and DBI ($F(2.27, 9.07) = 17.48$, adjusted $p = 0.0057$), but not Silhouette ($F(1.72, 6.89) = 9.43$, adjusted $p = 0.084$). The three-encoder tests detected no differences on these metrics (all adjusted $p = 1.00$). Non-significant encoder results do not demonstrate equivalence.

Follow-up comparisons found higher coverage than TF-IDF/HDBSCAN for RoBERTa, SecureBERT 1.0 and SecureBERT 2.0, by 26.3, 25.7 and 24.7 percentage points, respectively (Holm-adjusted $p = 0.0072$, 0.035 and 0.010). RoBERTa also had a lower mean DBI than TF-IDF/HDBSCAN by 1.17 (adjusted $p = 0.041$). No other follow-up pair was significant. Thus, RoBERTa was not established as the best method, and the evidence did not show that every embedding pipeline outperformed every baseline.

The keyword method illustrates why coverage and internal scores must be considered together. Its mean Silhouette was 0.76 ± 0.07, compared with 0.41 ± 0.12 and 0.41 ± 0.06 for the TF-IDF baselines, but its coverage remained below that of the embedding methods. A high internal score can coexist with many unassigned records and does not identify the most useful view for an analyst.

5.2 Comparison within the ModernBERT architecture

The architecture comparison used the same exported CVE objects and downstream settings as the main comparison. As Table 1 shows, SecureBERT 2.0 had a higher mean Silhouette than ModernBERT-base (0.82 ± 0.05 versus 0.80 ± 0.08). ModernBERT-base had slightly higher coverage, $(97.7 \pm 1.4)\%$, and lower DBI, 0.43 ± 0.08, compared with 0.45 ± 0.09 for SecureBERT 2.0. Neither checkpoint led on all three diagnostics. Paired t-tests detected no differences (all Holm-adjusted $p = 1.00$), so the comparison did not establish a consistent benefit from cybersecurity adaptation.

5.3 Whitening sensitivity

Figure 2 compares the main embedding results with the saved whitening runs. RoBERTa retained the highest Silhouette of the three encoders on Cisco, Siemens and ABB. On Beckhoff, whitening reversed its lead over both SecureBERT variants. On Yocto, the main-run lead of SecureBERT 1.0 disappeared and RoBERTa became the highest-scoring encoder. Encoder rankings were therefore sensitive to preprocessing.

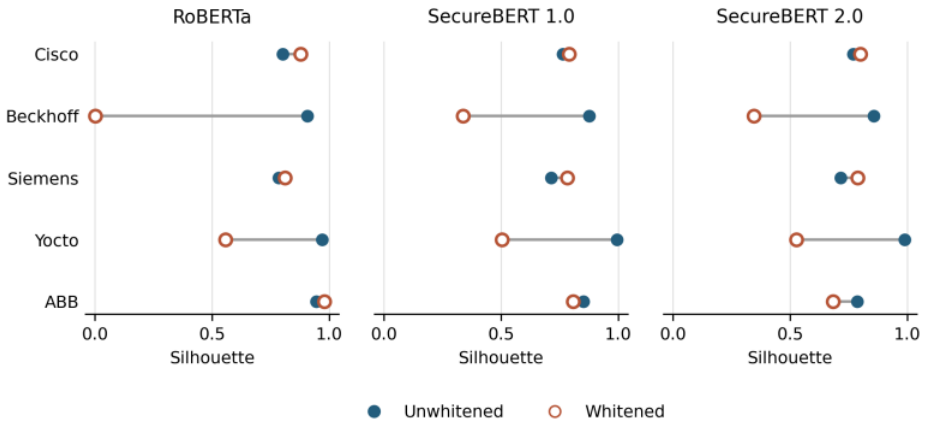


Fig. 2: Silhouette before and after whitening for each query and encoder. Filled blue markers show main-run values; open rust markers show whitening-run values. Connecting lines link paired results, and all panels use the same scale.

Whitening reduced the mean pairwise cosine diagnostic and changed the retained embedding dimension, cluster counts and coverage. Covariance estimation and rank selection on small samples may alter neighbourhood structure, but the exports did not establish the mechanism responsible for the ranking changes. The whitening files omitted their source CVE arrays: query metadata, collection sizes and severity histograms matched the main runs, but record-by-record identity could not be checked. These results support treating whitening as a sensitivity analysis rather than choosing it solely to maximise an internal score.

6 Discussion and limitations

Broad assignment coverage was the most consistent descriptive advantage of the hybrid embedding pipelines. This matters for an interface that presents groups as entry points into a retrieval result, but complete assignment can conceal inappropriate merges. Internal diagnostics favoured RoBERTa on average in the main comparison, but the exploratory tests did not establish differences among the encoders. The comparison within the ModernBERT architecture showed trade-offs. An operational tool should keep every cluster member and all unassigned records visible to the analyst.

The saved keywords and memberships provided examples of recognisable topics. On Beckhoff, all three encoders produced identical groups: web-interface terms (13 records), diagnostics and privilege terms (6), and ADS/UDP protocol terms (12). Their Yocto partitions were also identical, despite different internal scores. On ABB, an 18-record RoBERTa cluster mixed ABB/server and Magento/code-upload terms. SecureBERT 2.0 assigned five of those records to a six-record group labelled with ABB, OPC and 800xA terms, twelve to a Magento/code-upload group, and one to noise. The ABB/OPC group also included one record from another RoBERTa cluster. This example suggests finer topical separation with SecureBERT 2.0, but keywords and overlap alone

do not establish shared weaknesses, remediation or cybersecurity relevance. They support inspecting groups as analyst entry points, not an overall semantic-quality ranking.

Retrieval quality remained a separate constraint. Capped results were not complete inventories, and query matching could include unrelated products. Clustering organised these results without establishing their relevance. Product-specific conclusions would require checks against vendor advisories and affected-version information.

Both feature paths used security metadata, in different forms. The structured component could influence clustering independently of the encoder, and missing metadata could create grouping patterns. Cleaning could also affect collections differently. The study therefore compared complete pipelines and did not isolate the contribution of domain pretraining. No held-out reference, random-seed distribution or analyst assessment was available; practical benefits such as improved review accuracy or time savings remain unmeasured.

7 Conclusion

The main findings across five collections containing 677 distinct CVEs were:

- Hybrid embedding pipelines achieved higher mean coverage than the baselines. RoBERTa reached (98.7 ± 0.8)%, compared with (82.8 ± 5.9)% for keyword rules and (81.0 ± 5.2)% and (72.5 ± 2.9)% for the two TF-IDF pipelines. Follow-up tests found higher coverage for all three encoders than TF-IDF/HDBSCAN (Holm-adjusted $p \leq 0.035$); other coverage pairs were not significant.
- Cybersecurity adaptation did not consistently improve the reported diagnostics. RoBERTa had a mean Silhouette of 0.88 ± 0.04, compared with 0.84 ± 0.05 for SecureBERT 1.0. SecureBERT 2.0 led ModernBERT on mean Silhouette (0.82 ± 0.05 versus 0.80 ± 0.08), while ModernBERT led on coverage and DBI. The encoder comparisons did not establish statistically significant differences.
- Whitening changed encoder rankings on two collections, showing that preprocessing influenced the comparison.
- Keyword and membership inspection showed recognisable topics and finer separation in one ABB example with SecureBERT 2.0. Identical Beckhoff and Yocto partitions and the absence of systematic semantic assessment prevent a general semantic-quality ranking.

These findings support evaluating hybrid clustering as an organisation layer for vulnerability records, with source evidence available for inspection. Product applicability and analyst usefulness require separate validation; the present diagnostics do not establish either.

Acknowledgements

This work was funded on behalf of the German Federal Ministry of Research, Technology and Space (BMFTR) as part of the SRAG project (grant number 16KIS2316K).

References

[Ag22] Aghaei, E.; Niu, X.; Shadid, W.; Al-Shaer, E.: SecureBERT: A Domain-Specific Language Model for Cybersecurity. arXiv:2204.02685, 2022.

[An22] Anwar, A.; Abusnaina, A.; Chen, S.; Li, F.; Mohaisen, D.: Cleaning the NVD: Comprehensive Quality Assessment, Improvements, and Analyses. IEEE Transactions on Dependable and Secure Computing 19(6), pp. 4255-4269, 2022.

[Ca13] Campello, R. J. G. B.; Moulavi, D.; Sander, J.: Density-Based Clustering Based on Hierarchical Density Estimates. In: Advances in Knowledge Discovery and Data Mining (PAKDD 2013), pp. 160-172, 2013.

[Ci26] Cisco AI: SecureBERT2.0-base Model Card. https://huggingface.co/cisco-ai/SecureBERT2.0-base, accessed 16.09.2026.

[Da79] Davies, D. L.; Bouldin, D. W.: A Cluster Separation Measure. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-1(2), pp. 224-227, 1979.

[Es96] Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X.: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: Proc. KDD-96, pp. 226-231, 1996.

[Et19] Ethayarajh, K.: How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. In: Proc. EMNLP-IJCNLP 2019, pp. 55-65, 2019.

[Li19] Liu, Y. et al.: RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692, 2019.

[Mc18] McInnes, L.; Healy, J.; Melville, J.: UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv:1802.03426, 2018.

[Ni23] National Institute of Standards and Technology: Guide to Operational Technology (OT) Security. Technical report SP 800-82 Rev. 3, 2023. https://doi.org/10.6028/NIST.SP.800-82r3.

[Ro87] Rousseeuw, P. J.: Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. Journal of Computational and Applied Mathematics 20, pp. 53-65, 1987.

[Sa88] Salton, G.; Buckley, C.: Term-Weighting Approaches in Automatic Text Retrieval. Information Processing and Management 24(5), pp. 513-523, 1988.

[Su21] Su, J. et al.: Whitening Sentence Representations for Better Semantics and Faster Retrieval. arXiv:2103.15316, 2021.

[Wa24] Warner, B. et al.: Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference. arXiv:2412.13663, 2024.

Feasibility of LLM Based Generation of IEC/IEEE 60802 TSN Configurations Using RAG

Manish Laxman Uplanchi [1], Alexander Biendarra [1], Tobias Ferfers [1]

Abstract: Time-Sensitive Networking configuration requires standards knowledge, YANG model structure, device-specific semantics, and validation beyond XML syntax. This paper evaluates a RAG-assisted prototype that translates structured or natural-language TSN requirements into per-device NETCONF/YANG configurations for IEEE 802.1AS/gPTP and IEEE 802.1Qbv/TAS within the IEC/IEEE 60802 context. The workflow combines requirement extraction, retrieval from a 2,854-record knowledge base, constrained generation, four validation layers, and bounded correction. Evaluation combined increasing-complexity gPTP/TAS scenarios, adversarial cases, reference comparisons, robustness prompts, and a No-RAG direct-LLM baseline. All four positive configuration scenarios passed validation, while refining requirement extraction reduced false acceptance in the adversarial suite. The prototype is therefore useful for translating explicitly specified synchronization roles, network interfaces, and user-defined TAS schedules into validated device-specific configurations. Its current scope excludes schedule optimization, arbitrary topologies, complete IEC/IEEE 60802 conformance, and physical-device interoperability.

Keywords: Time Sensitive Networking, IEC/IEEE 60802, Retrieval Augmented Generation, NETCONF, YANG, Large Language Models

1 Introduction

Industrial automation increasingly relies on converged Ethernet infrastructures that must carry time critical control traffic together with ordinary best effort communication. Time Sensitive Networking (TSN) extends Ethernet with mechanisms for synchronized time, deterministic scheduling, reliability, and resource management [LS19]. IEC/IEEE 60802 profiles these mechanisms for industrial automation and constrains the options, defaults, protocols, and management functions needed for interoperable deployment [II26]. This convergence reduces the need for isolated communication islands, but it increases configuration complexity because synchronization roles, traffic classes, gate schedules, interfaces, namespaces, and YANG hierarchies must remain consistent across multiple devices and standards.

NETCONF and YANG provide a standardized management foundation for representing and transporting configuration data [En11][Bj16]. Schema valid XML alone, however, does not guarantee that a configuration expresses the intended engineering requirement. A payload can target the wrong interface, contain an inconsistent Gate Control List (GCL),

[1] Fraunhofer IOSB INA, Campusallee 1, 32657 Lemgo, Germany; alexander.biendarra@iosb ina.fraunhofer.de; tobias.ferfers@iosb ina.fraunhofer.de; manish.uplanchi@iosb ina.fraunhofer.de

assign an invalid synchronization role, or include operational state instead of configurable data. Automated TSN configuration, therefore, needs both standards grounded generation and explicit validation.

Recent studies investigate LLMs for configuration generation, correction, and validation [Wa24][HFV24][Li24][Li25a][Li25b]. RAG (Retrieval Augmented Generation) is relevant because required knowledge is distributed across standards, data models, and engineering rules [Ma25] [Wi25]. TSN work further shows that complete configuration and schedule-related tasks remain challenging [De25][YI24][He20].

This paper evaluates a bounded RAG-assisted configuration workflow for two mechanisms relevant to IEC/IEEE 60802: IEEE 802.1AS generalized Precision Time Protocol (gPTP) synchronization and IEEE 802.1Qbv Time Aware Shaping (TAS). The prototype was developed specifically for this research rather than being built on a pre-existing network configuration agent framework. Its orchestration, requirement extraction, retrieval strategy, prompt construction, validation logic, and correction workflow evolved iteratively during implementation. The main contribution is therefore not an unrestricted claim that an LLM can configure TSN but an empirical feasibility result for a validation-centered generation process with a precisely delimited configuration scope.

2 Background and Related Work

IEEE 802.1AS specifies protocols and managed objects for distributing synchronized time and selecting the timing source in time sensitive networks *[IE25].* Within the evaluated prototype, gPTP generation covers a bounded parameter set: PTP instance identification, domain number, priority1, priority2, time receiver only, PTP port indices, interface association, and port enablement. The scope was limited to parameters required by the evaluated synchronization and topology scenarios and covered by deterministic semantic checks. Message interval parameters were therefore left outside the prototype scope.

IEEE 802.1Q defines bridges, bridged networks, and the scheduled traffic mechanisms used by TAS [IE22]. A GCL contains ordered gate operations with interval durations and gate state bit masks. Determining a suitable schedule is itself an optimization problem, and prior work has investigated algorithmic GCL synthesis and scalable scheduling [YI24][He20]. The present prototype does not optimize a schedule. It translates a schedule already stated in the requirement into YANG based configuration containing the target interface, gate parameter table, administrative control list, gate control entries, interval durations, masks, and cycle time.

IEC/IEEE 60802 connects these mechanisms to the industrial automation profile and specifies YANG based management concepts [II26]. The prototype consequently uses YANG modules for IETF interfaces, IEEE 1588 PTP, IEEE 802.1AS gPTP, and IEEE 802.1Q bridge and scheduling structures. Runtime schema validation is performed with

yanglint, while *pyang* is used offline to compile schema information for the retrieval knowledge base.

2.1 LLM Assisted Configuration

Automated TSN configuration has previously been approached through dedicated configuration frameworks, software defined networking, and model driven management [Ho21][Sc25]. These systems demonstrate the value of reducing manual parameter translation but generally rely on deterministic models and algorithms rather than free-form natural language interpretation.

LLM based approaches broaden the input interface but introduce reliability concerns. NetConfEval shows that adding documentation alone does not guarantee better configuration output [Wa24]. Hollosi et al. report that complex NETCONF/YANG generation remains difficult [HFV24]. NetPilot combines retrieval, generation, verification, and correction [Wi25], while *ConfAgent* [Li25a], *ConfigChecker* [Li25b], and *Ciri* [Li24] reinforce the need for explicit synthesis or validation around model output. These findings motivate treating the LLM as a constrained generator rather than the acceptance authority.

3 Architecture and Development Evolution

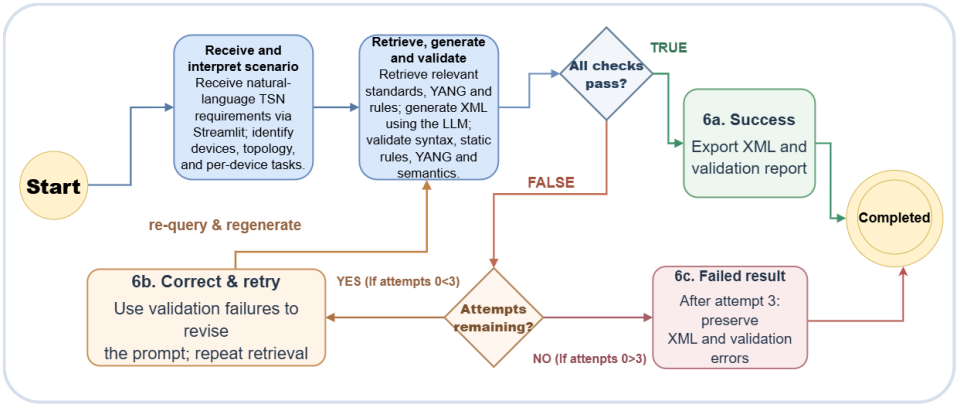


Fig. 1: Flowchart of the implemented RAG assisted TSN configuration system.

Figure 1 shows implementation of the prototype and interaction between user interface, requirement processing, retrieval, generation, and validation. The user interacts with the system through a web interface implemented with Streamlit and provides either a structured scenario or free-form TSN requirement through it. Structured input is parsed deterministically, free-form input is converted by extraction into a strict, structured

representation. Both routes produce ScenarioSpec. The ScenarioSpec is decomposed into one DeviceSpec per device. DeviceSpec is a project specific structure containing role, interfaces, gPTP parameters, and, where applicable, TAS parameters. A deterministic prompt builder creates a device specific instruction, also used for Chroma retrieval.

Retrieval uses the local all MiniLM-L6-v2 embedding model. Retrieved standards passages, YANG schema information, namespaces, and semantic rules are combined with instruction for NETCONF/YANG generation. Extraction and generation are separate LLM tasks, but both use Qwen/Qwen3 Coder Next FP8 through the FHGenie OpenAI compatible endpoint. Correction uses no additional model: validation diagnostics are appended to the instruction and the same retrieval generation path repeats.

Each generated <config> candidate is normalized and checked for XML well formedness, project specific static rules, YANG validity with yanglint, and deterministic semantic consistency against DeviceSpec. Acceptance requires all mandatory checks; otherwise, failures form the correction prompt.

Generation permits three attempts per device: one initial attempt and two corrective retries maximum. These bound limits repeated inference while allowing correction and was not experimentally optimized.

Most runtime components are local: Streamlit, orchestration, Chroma, embeddings, XML processing, and yanglint. LLM inference is remote through FHGenie; pyang is offline for knowledge base preparation. The architecture evolved from structured input to dual input operation, with extraction tightened to preserve unresolved values and contradictions rather than silently repairing them.

3.1 Knowledge Base, Chunking, and Retrieval

The RAG knowledge base is stored in a persistent local Chroma vector database and contains 2,854 records across standards text, YANG schema paths, namespaces, and semantic rules. Text embeddings are generated locally using sentence-transformers/all-MiniLM-L6-v2, producing 384-dimensional vectors for semantic similarity search. Standards text from IEEE 802.1AS, IEEE 802.1Q, and IEC/IEEE 60802 is prepared using clause aware chunking with a maximum target size of 1,200 characters and 180 characters of trailing overlap. Consecutive text belonging to the same clause is merged where possible. Structural knowledge is stored separately as one record per YANG module, compiled schema node, or semantic rule. Table 1 summarizes the resulting knowledge base composition and retrieval limits.

Source type	Records	Preparation	Retrieved records per source type
IEEE 802.1AS 2025	1,314	1,200 char / 180 overlap	4
IEEE 802.1Q 2022	366	1,200 char / 180 overlap	4
IEC/IEEE 60802:2026	327	1,200 char / 180 overlap	4
YANG schema paths	817	one schema node / record	8
YANG namespaces	10	one module / record	4
Semantic rules	20	one rule / record	4
Total	2,854		

Tab. 1: Live knowledge base composition and retrieval quotas.

YANG schema path retrieval retains up to eight records, whereas the remaining source types retain four. This asymmetry is a fixed implementation heuristic and was not formally optimized; no documented rationale was found for selecting eight rather than four. Retrieval is performed independently for each source type rather than through one global similarity search. Chroma first returns a larger candidate set for each category, after which metadata-based score adjustments are applied. Read only YANG schema paths are penalized for configuration-oriented requests, required namespace modules are preferentially included, results are deduplicated, and the retained records are assembled in a fixed source order.

3.2 Prompting and Model Configuration

Natural language extraction and XML generation use the same served model, Qwen/Qwen3 Coder Next FP8. Qwen3 Coder Next is described as an 80 billion parameter mixture of experts coding model that activates approximately 3 billion parameters during inference [Ca26]. In the evaluated prototype, the model is accessed through the FHGenie OpenAI compatible endpoint. Temperature is fixed at 0.0 to reduce sampling variability; no random seed is configured, and other sampling parameters remain at endpoint or library defaults.

The prompting strategy separates three responsibilities. The extraction prompt converts free-form requirements into strict JSON and is instructed not to guess missing values or repair inconsistencies. A deterministic function then builds one instruction per DeviceSpec, a project specific internal data structure containing the configuration requirements for an individual device. The generation prompt requires one NETCONF <config> element, exact YANG namespaces and hierarchy, configurable nodes only, and no explanatory text. On failure, the original device instruction is augmented with detected

XML, static rule, yanglint, or semantic validation failures and resubmitted through the same generation path.

> Generate NETCONF/YANG configurations for a TSN network using IEEE 802.1AS/gPTP and IEEE 802.1Qbv/TAS. Topology: tsnClock connects to localTsnSwitch on eth0, and localTsnSwitch connects to tsnDevice on eth1/eth0. tsnClock is the Grandmaster, localTsnSwitch is the TSN bridge, tsnDevice is a time receiver only slave end station. Use gPTP domain number 0 with standard role based priority defaults. On localTsnSwitch's eth1 interface, configure a Qbv schedule supporting 2 traffic classes, with a 1 millisecond cycle split into two 500 microsecond intervals: during the first interval only traffic class 0 is open, and during the second interval only traffic class 1 is open. Produce one validated NETCONF RPC per device, checked for XML syntax, YANG schema, and semantic correctness.

Listing 1: Example of a natural language input request for the three device Easy scenario.

Short excerpts illustrate the constraints:

```
Extraction: "faithful transcription ... never plausible completion"
Generation: "Use only nodes, hierarchy, types and namespaces found in
the                        retrieved                        context."
Generation:  "Do  not  emit  any  node  marked  config  false."
Correction: "Fix only the issues listed below and keep everything else
... unchanged."
```

Listing 2: Selected constraints from the extraction, generation, and correction prompts.

Listing 1 presents the three-device easy scenario input, while Listing 2 summarizes representative extraction, generation, and correction constraints. These constraints were refined manually. No formal hyperparameter sweep was performed; therefore, the reported setup is a fixed implementation rather than an experimentally optimized configuration.

3.3 Validation and Supported Configuration Scope

As shown in Figure 1, each generated NETCONF candidate passes through four complementary validation layers before acceptance. First, lxml checks XML well formedness. Second, project specific static rules detect selected invalid configuration patterns. Third, yanglint validates the candidate against the implemented IEEE and IETF YANG modules, including hierarchy, namespaces, list keys, types, ranges, and modeled constraints. pyang is used separately during offline knowledge base preparation to compile YANG schema information and generate namespace and schema path records; it is not used for runtime XML validation. Fourth, a deterministic semantic validator performs 20 named checks against the corresponding DeviceSpec, covering interface structure, gPTP

mappings, TAS parameters, GCL entries, cycle time consistency, and prohibited elements. Two additional pre generation guards reject out of range Qbv traffic class assignments and contradictory Grandmaster/time receiver only roles.

If validation fails, the detected XML, static rule, YANG, or semantic failures are returned to the correction path shown in Figure 1. The pipeline allows at most three generation attempts per device and stops when all mandatory checks pass.

The implemented prototype covers selected IEEE 802.1AS/gPTP parameters and IEEE 802.1Qbv/TAS configuration. TAS schedules are supplied by the user and translated into YANG configuration; the prototype does not calculate or optimize schedules. Therefore, successful validation demonstrates consistency with the implemented schemas and project specific semantic requirements, but does not establish complete IEC/IEEE 60802 conformance or replace validation on physical TSN devices.

4 Evaluation Methodology

This chapter evaluates the RAG assisted TSN configuration prototype using positive, adversarial, reference, robustness, runtime, and No-RAG direct-LLM baseline tests. It assesses configuration correctness, rejection of invalid or ambiguous requirements, and preservation of gPTP/TAS semantics within the implemented scope

The evaluation contains 18 cases: four positive-path scenarios, eleven adversarial cases, and three independently prepared reference-fidelity scenarios. Correction attempts are treated as retries rather than independent repetitions. The scenario set was designed for bounded coverage, progressively varying topology size, traffic-class count, GCL length, cascading, branching, and selective TAS application, rather than for statistical representativeness.

Scenario	Devices	Traffic classes	GCL entries	Main increase
Easy	3	2	2	linear GM bridge receiver
Medium	3	4	3	linear; combined gate masks
Complex	4	8	5	two cascaded bridges
Complex branching	5	8	8	branched topology; TAS on one branch

Tab. 2: Four positive path scenarios of increasing configuration complexity.

Easy level topology scenario uses a three device Grandmaster-bridge-receiver topology with two traffic classes and two GCL entries. Medium keeps the topology but increases

traffic classes and introduces combined gate masks. Complex topology adds a second bridge with eight traffic classes and five GCL entries, while Complex branching extends to five devices and two downstream branches, with TAS applied only on the switchB-Node2 path.

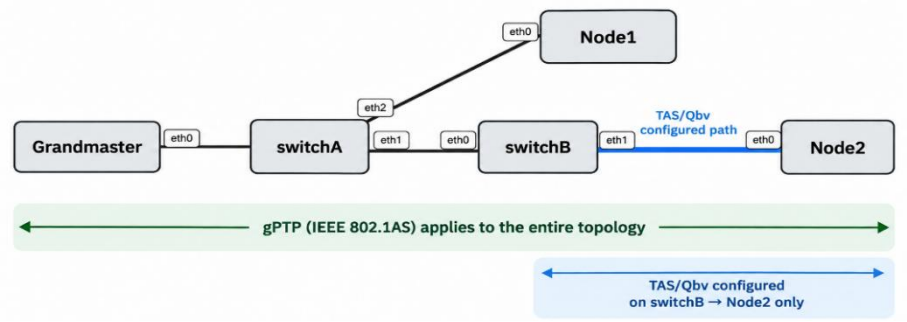


Fig. 2: Five device branching topology of the complex positive path scenario.

All positive scenarios combine gPTP and TAS using user-supplied schedules. Evaluation therefore focuses on faithful configuration translation, with success requiring all device configurations to pass XML, static-rule, YANG, and semantic validation. The eleven adversarial cases contain deliberately problematic natural language requirements, including missing TAS target information, inconsistent interval and gate state lists, missing cycle time, traffic class range violations, inconsistent cycle duration, interface mismatches, contradictory synchronization roles, out of range gPTP values, and requests for operational or read only state. These cases exercise rejection points across requirement extraction, scenario construction, pre generation guards, YANG validation, and semantic validation.

Several adversarial cases were used during development to refine extraction and validation behavior. Consequently, this suite is treated as a development and regression set rather than as held out evidence of robustness to previously unseen faults. Pasted markdown

Reference fidelity is assessed separately from the runtime semantic validator using independently prepared configuration expectations. Comparisons are performed at field or device level rather than through byte identical XML, so formatting and attribute ordering do not affect the outcome. This provides an additional comparison against separately defined expected configuration values rather than relying exclusively on the runtime DeviceSpec.

4.1 Measures and Boundaries

To complement the development-oriented adversarial suite, six additional prompts were fixed before execution and the system was not modified afterward. They cover informal or reordered language, ambiguity in TAS targets or synchronization roles, and previously untested conflicting requirements. Valid inputs were expected to pass all validation layers, while ambiguous or invalid inputs should be rejected or remain unresolved.

End-to-end execution times ranged from less than 1 s to 49 s; cases proceeding through generation required approximately 39-49 s, while early rejection or failure completed within 1-5s. These single-run observations indicate interactive response time rather than benchmark performance. The Easy scenario was also submitted directly to Qwen3-Coder-Next through FHGenie without retrieved standards, YANG schema paths, namespaces, or semantic rules. This No-RAG direct-LLM baseline provides a practical comparison with the complete RAG-assisted workflow.

5 Results and Discussion

After the development correction cycle, all four positive path scenarios passed the complete validation procedure. The largest case contained five devices in a branching topology, eight traffic classes, and eight GCL entries. The workflow therefore demonstrated successful translation of one network level requirement into several device specific gPTP configurations while applying TAS only to the requested branch and preserving the supplied schedule values. This supports feasibility within the implemented parameter set rather than unrestricted TSN configuration generation.

The eleven adversarial cases cover missing TAS targets, inconsistent GCL data, invalid cycle times or traffic classes, interface mismatches, contradictory synchronization roles, out-of-range gPTP values, and requests for read-only state. They exercise multiple rejection points across extraction, scenario construction, pre-generation checks, YANG validation, and semantic validation. Because several cases were used to refine the implementation, the suite is treated as a development and regression set rather than as evidence of robustness to unseen faults.

The separate exploratory robustness evaluation therefore used six additional fixed prompts without modifying the system afterward. Three produced the intended behavior: two informal language requests generated fully validated configurations, and an unspecified TAS target was rejected without guessing the missing interface. Three exposed remaining limitations. An ambiguous synchronization provider description passed without generating the intended gPTP configuration, a conflicting TAS requirement failed for an unrelated interface mismatch rather than the intended contradiction, and conditional Grandmaster eligibility was lost during extraction and therefore remained undetected.

These results show that informal phrasing was handled successfully in the tested cases, while semantic ambiguity and information loss during extraction remain important limitations. They also reinforce that successful XML and YANG validation cannot recover requirements that were incorrectly resolved or discarded earlier in the pipeline.

5.1 Reference Fidelity and Interpretation

Reference scenario	Granularity	Observed match
Single device gPTP	field level	12 / 12
Three device gPTP + TAS	field level	32 / 32
Four device gPTP + TAS	device level	4 / 4 devices

Tab. 3: Independent reference fidelity results for the three evaluated scenarios.

All independently defined conditions in the three reference scenarios were satisfied. The denominators in Table 3 have different meanings: the first two count field-level assertions, whereas the four-device case reports an aggregate pass per device. They should therefore not be combined into a single accuracy percentage. Nevertheless, the comparison provides evidence beyond the internal semantic validator because the reference material was prepared independently of the DeviceSpec used during runtime generation.

Taken together, the positive-path, reference-fidelity, robustness, and No-RAG direct-LLM results support the feasibility of combining natural-language interpretation, standards- and schema-grounded retrieval, constrained generation, YANG validation, deterministic semantic checks, and bounded correction for the selected gPTP and TAS tasks. The No-RAG direct-LLM baseline produced less constrained output with additional unspecified parameters and ambiguous topology or TAS interpretations. However, the comparison does not isolate retrieval as the sole cause of improvement. The robustness evaluation further shows that informal phrasing can be handled, while ambiguity and information loss during extraction remain important limitations.

6 Limitations, Conclusion, and Outlook

This work demonstrates the feasibility of a RAG assisted prototype for generating and validating NETCONF/YANG configurations for a bounded subset of IEEE 802.1AS/gPTP and IEEE 802.1Qbv/TAS. The system supports selected gPTP parameters and the translation of predefined TAS schedules into device specific configurations. Its purpose is configuration translation rather than schedule optimization, and successful XML, YANG, and semantic validation confirms consistency with the implemented models and rules rather than complete IEC/IEEE 60802 conformance or physical device interoperability.

In practical use, the prototype can process structured and free-form requirements for the evaluated linear, cascaded, and branching topologies, including multiple synchronization roles and selective TAS configuration. The evaluation also shows that requirement interpretation remains an important limitation. Informal descriptions were handled successfully in the tested cases, whereas ambiguous roles, incomplete topology information, or contradictory requirements may still be interpreted incorrectly before generation. Clear specification of device roles, connections, interfaces, and TAS parameters therefore improves reliability.

The evaluation combines positive, adversarial, reference fidelity, and exploratory robustness cases. A No-RAG direct-LLM baseline additionally provides a practical comparison with generation without the prototype's retrieved standards and YANG context. This comparison indicates the benefit of the complete RAG assisted workflow, including structured decomposition, retrieval, validation, and correction, while a controlled retrieval only ablation would be required to quantify the contribution of retrieval independently.

Qwen/Qwen3 Coder Next FP8 was selected because it is available through FHGenie and is suitable for code oriented generation. Future work should extend the supported TSN and YANG parameters, strengthen ambiguity and contradiction handling, evaluate a broader range of topologies and independently worded requirements, compare alternative models and retrieval configurations, and test accepted configurations on simulated or physical TSN devices.

7 References

[Bj16] Björklund, M.: The YANG 1.1 Data Modeling Language. RFC 7950, 2016.

[Ca26] Cao, R. et al.: Qwen3 Coder Next Technical Report. CoRR abs/2603.00729, 2026.

[De25] Debnath, R.; Zhao, L.; Barzegaran, M.; Steinhorst, S.: Introducing Large Language Models into the Design Flow of Time Sensitive Networking. arXiv:2509.26368, 2025.

[En11] Enns, R.; Björklund, M.; Schoenwaelder, J.; Bierman, A.: Network Configuration Protocol (NETCONF). RFC 6241, 2011.

[IE24] IEEE: IEEE Standard for Precision Clock Synchronization Protocol for Networked Measurement and Control Systems - Amendment 5: MIB and YANG Modules. IEEE Std 1588e 2024, 2024.

[He20] Hellmanns, D. et al.: Scaling TSN Scheduling for Factory Automation Networks. In: Proc. IEEE International Conference on Factory Communication Systems (WFCS), S. 1 8, 2020.

[HFV24] Hollósi, G.; Ficzere, D.; Varga, P.: Generative AI for Low Level NETCONF Configuration in Network Management Based on YANG Models. In: Proc. 20th International Conference on Network and Service Management (CNSM), 2024.

[Ho21] Houtan, B. et al.: An Automated Configuration Framework for TSN Networks. In: Proc. 22nd IEEE International Conference on Industrial Technology (ICIT), S. 771 778, 2021.

[IE22] IEEE: IEEE Standard for Local and Metropolitan Area Networks Bridges and Bridged Networks. IEEE Std 802.1Q 2022, 2022.

[IE25] IEEE: IEEE Standard for Local and Metropolitan Area Networks Timing and Synchronization for Time Sensitive Applications. IEEE Std 802.1AS 2025, 2025.

[II26] IEC/IEEE: Time Sensitive Networking Profile for Industrial Automation. IEC/IEEE 60802:2026, 2026.

[Li24] Lian, X. et al.: Configuration Validation with Large Language Models. arXiv:2310.09690, 2024.

[Li25a] Li, S. et al.: ConfAgent: Towards Intelligent Network Configuration Via LLM Agent. In: Proc. IEEE/ACM International Symposium on Quality of Service (IWQoS), S. 1 10, 2025.

[Li25b] Li, F. et al.: ConfigChecker: Automated Network Configuration Validation with Large Language Model and Knowledge Graph. In: Proc. IEEE/ACM International Symposium on Quality of Service (IWQoS), S. 1 10, 2025.

[LS19] Lo Bello, L.; Steiner, W.: A Perspective on IEEE Time Sensitive Networking for Industrial Communication and Automation Systems. Proceedings of the IEEE 107/6, S. 1094 1120, 2019.

[Ma25] Majlesara, A.; Majlesi, A.; Mamaghani, A.; Shokrani, A.; Khalaj, B. H.: 5G Network Automation Using Local Large Language Models and Retrieval Augmented Generation. arXiv:2511.21084, 2025.

[Sc25] Schappacher, M.; Welte, D.; Sikora, A.; Lehmann, C.: A YANG Model Based Approach for Configuration and Management of 5G TSN Bridges. In: Proc. 21st IEEE International Conference on Factory Communication Systems (WFCS), S. 1 8, 2025.

[Wa24] Wang, C. et al.: NetConfEval: Can LLMs Facilitate Network Configuration? Proceedings of the ACM on Networking 2/CoNEXT2, Art. 7, S. 1 25, 2024.

[Wi25] Windmann, S.; Albrecht, J.; Friesen, M.; Jasperneite, J.: NetPilot Towards LLM Assisted Configuration of Hybrid TSN/5G Networks. In: Proc. IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), S. 1 4, 2025.

[YI24] Yoshimura, A.; Ito, Y.: A Study on Determination of an Appropriate GCL of Time Aware Shaper in Ethernet Based Industrial Networks. In: Proc. 15th International Conference on Ubiquitous and Future Networks (ICUFN), S. 355 359, 2024.

Asset Administration Shell Based Orchestration and Performance Evaluation of 5G Timeslot Configurations for Industrial Applications

Gustavo P. Cainelli[1], Hasal Kulasekara Pallewaththe Kankanamge[2], Kenji Ota[3], Santiago Soler Perez Olaya[2], Martin Wollschlaeger[2], Takashi Shiba[3], Noriharu Suematsu[3], and Lisa Underberg[1]

Abstract: Industry 4.0 applications are characterized by dynamic communication requirements, often resulting in asymmetric uplink and downlink traffic patterns. Such variability makes static 5G network configurations inadequate for supporting diverse operational scenarios in industrial environments. To address this challenge, this paper presents an Asset Administration Shell (AAS)-based orchestration approach for adaptive 5G network reconfiguration. An exemplary scenario is considered in which 5G timeslot configurations are determined according to application-specific requirements. The proposed method models application demands and network capabilities as AAS submodels, enabling a network orchestrator to match these descriptions and identify the most appropriate configuration. The approach is evaluated using an intralogistics crane use case deployed on a 5G testbed. Results demonstrate that application-aware network adaptation better supports varying communication demands, highlighting the limitations of static network configurations in Industry 4.0 environments.

Keywords: Industry 4.0, Asset Administration Shell, Network reconfiguration, Industrial 5G

1 Introduction

Industry 4.0 (I4.0) applications, such as intralogistics systems, machine vision, automated guided vehicles (AGVs), and collaborative robots, exhibit dynamic communication requirements regarding data rate, transmission time, reliability, and availability. Wireless communication systems (WCS), particularly 5G networks, are expected to support these heterogeneous demands while efficiently utilizing network resources [3rb; 5G21]. However, different operational scenarios can result in asymmetric uplink and downlink traffic; for example, machine vision may require significant uplink capacity for video streams, while other applications rely primarily on downlink communication for control and monitoring. Consequently, static network configurations cannot efficiently support these varying requirements [5G21].

[1] Institut für Automation und Kommunikation e.V. (ifak), Werner-Heisenberg-Straße 1, 39106 Magdeburg, Germany, gustavo.cainelli@ifak.eu; lisa.underberg@ifak.eu

[2] Technische Universität Dresden (TUD), Department of Electrical and Computer Engineering, 01069 Dresden, Germany, hasal_kulasekara.pallewaththe@tu-dresden.de; santiago.soler_perez_olaya@tu-dresden.de; martin.wollschlaeger@tu-dresden.de

[3] Research Institute of Electrical Communication (RIEC), Tohoku University, 2-1-1 Katahira, Aoba-ku, 980-8577 Sendai, Japan, kenji.ota@tohoku.ac.jp; takashi.shiba.e2@tohoku.ac.jp; noriharu.suematsu.a3@tohoku.ac.jp

In industrial 5G systems, different timeslot allocations in Time Division Duplex (TDD) provide different uplink and downlink capacities, as illustrated in Figure 1, directly influencing data rates and transmission times [3ra; 5G26]. Selecting an appropriate configuration therefore requires knowledge of both application requirements and network capabilities. A major challenge is establishing a standardized representation for information exchange between the application and communication domains.

The Asset Administration Shell (AAS), the standardized Digital Twin approach of Industry 4.0, provides a technology-independent and interoperable model for representing asset-related information [Pl20; Pl22]. This paper presents an AAS-based orchestration approach for selecting 5G timeslot configurations based on application-specific communication requirements. The architecture models application requirements and network capabilities using AAS, enabling an orchestrator to match them and recommend suitable configurations. The approach is evaluated using an intralogistics crane use case developed in Germany and emulated on a 5G testbed at RIEC, Tohoku University, Japan, assessing uplink–downlink configurations in terms of data rate and transmission time.

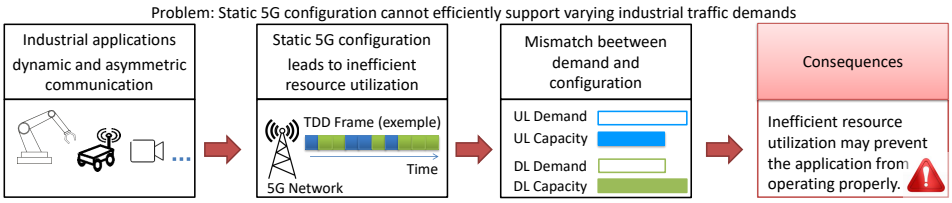


Fig. 1: Limitations of static 5G TDD configurations in supporting dynamic and asymmetric communication demands in industrial applications.

The main contributions are:

- An AAS-based representation of application communication requirements and 5G network capabilities.
- A network orchestration workflow for selecting 5G timeslot configurations.
- An experimental implementation of an intralogistics crane use case over a 5G testbed.
- A performance evaluation of uplink–downlink configurations regarding data rate and transmission time.

The remainder of this paper is organized as follows. Section 2 reviews related work, Section 3 presents the proposed approach, Section 4 describes the use case and experimental setup, Section 5 presents the results, and Section 6 concludes the paper.

2 Background and Related Work

2.1 Parameters of 5G that influence performance

Several configurable 5G parameters significantly affect transmission time, reliability, throughput, and resource utilization. Some of them are described below.

Radio Link Control Mode (RLC Mode): The RLC mode determines the trade-off between reliability and transmission time. Two modes are available through the Centralized Unit (CU) configuration [Ap23]: *RLC_AM*, which retransmits lost packets to ensure high reliability at the cost of increased transmission time, and *RLC_UM_BIDIRECTIONAL*, which avoids retransmissions and prioritizes low transmission time. Although *RLC_UM* may lead to occasional packet loss, it better supports real-time communication. Therefore, the choice of RLC mode represents a trade-off between reliability and low transmission time.

ulSchedInterval: The *ulSchedInterval* parameter defines the periodicity of uplink scheduling requests and influences the responsiveness of the network to uplink traffic. Shorter scheduling intervals improve responsiveness and consequently decrease the overall transmission time, but increase control overhead and processing load which leads to higher energy consumption.

Time Division Duplexing (TDD): The TDD configuration determines how radio resources are divided between downlink (DL) and uplink (UL) transmissions. Different DL/UL slot ratios allow the network to be optimized for traffic patterns dominated by either downlink or uplink communication. Configurations with more uplink symbols improve uplink capacity and reduce UL transmission time, whereas downlink-oriented configurations provide higher downlink throughput.

5G QoS Identifier (5QI): The 5G QoS Identifier (5QI) specifies the quality-of-service characteristics associated with a data flow, including priority level, packet delay budget, and packet error rate. Different 5QI values are defined for different applications and services. Selecting an appropriate 5QI enables the network to meet the transmission time, reliability, and throughput requirements of specific use cases.

2.2 AAS and AAS-based Network Management

The AAS is the standardized digital representation of an asset in the I4.0 framework. An asset may be a physical entity, such as a machine or a communication device, or a logical entity, such as a software application or service. Together, an asset and its AAS form an I4.0 component through which asset-related information can be identified, accessed, and exchanged throughout the asset life cycle [Pl20; Pl22]. Information within an AAS is organized into submodels, each of which represents a specific aspect of the asset, such as identification, operational data, capabilities, or configuration parameters. An application

can describe its required communication behavior through an AAS submodel, while the communication system can expose capabilities through another AAS submodel. The network-management problem can then be treated as a model-based matching problem between application requirements and communication capabilities.

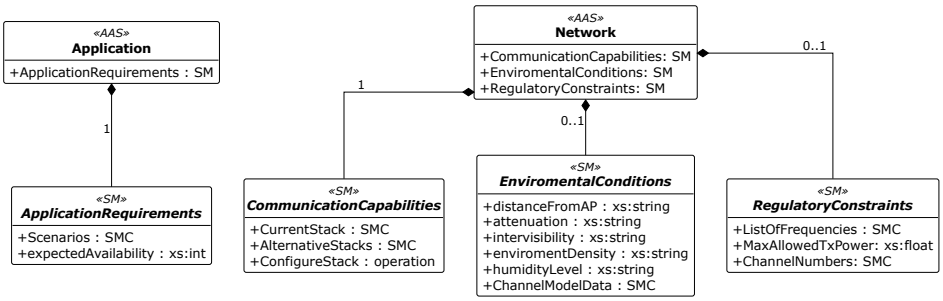


Fig. 2: A UML class diagram for an abstract AAS network management architecture.

This model-based matching problem can be solved by a centralized computing entity which can be called an “orchestrator”. Upon activation of an application function, the orchestrator retrieves the corresponding requirements from the Application AAS, compares them with the capabilities exposed by the Network AAS and selects a feasible configuration.

This direction moves network management from static engineering-time configuration toward model-based, application-aware adaptation. Previous research has progressively applied this concept to industrial 5G systems. Cainelli et al. represented relevant communication-system components using AASs and demonstrated how communication parameters can be adapted through negotiation between the production and communication domains [Ca23]. Olaya et al. subsequently investigated the dynamic reconfiguration of 5G non-public networks in response to changing application requirements [Ol24b]. Their extended architecture distinguishes four information perspectives—application requirements, network capabilities, legal constraints, and environmental conditions—and introduces an orchestrator that evaluates these models before selecting a network configuration [Ol24a]. Related work further applied this architecture to automated adaptation in an industrial intralogistics setting [Ca24a; Ca24b] and to transmit-power reconfiguration based on represented channel conditions [Ka25]. Furthermore, in [Ka25], the authors used a specific wireless channel model to represent the channel condition information in the AAS, and described the computational steps for adapting the coverage and energy consumption. This demonstrates the versatility and adaptability in the generic approach of AAS-based network management.

3 AAS-Based Network Orchestration Architecture

This section introduces the proposed AAS-based network orchestration architecture and describes how application communication requirements and network capabilities are represented and matched to enable adaptive 5G network configuration.

3.1 Overall Architecture

The proposed architecture relies on AAS to provide standardized representations of application requirements and network capabilities, enabling adaptive network orchestration. As illustrated in Figure 3, the Application AAS describes the application communication requirements of the active scenario, while the Network AAS provides the available capabilities. The orchestrator retrieves this information and uses a decision engine to select and apply a suitable 5G TDD configuration at runtime.

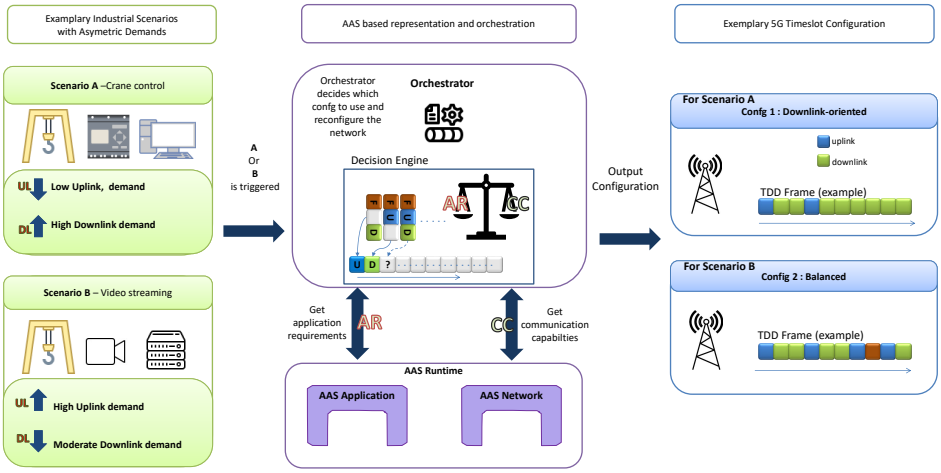


Fig. 3: Overview of the proposed AAS-based orchestration approach, from application-specific communication demands to adaptive 5G TDD configuration.

The approach is illustrated using two scenarios from an industrial intralogistics crane use case. Scenario A represents crane control, characterized by low uplink and high downlink demand, whereas Scenario B represents video streaming, with high uplink and moderate downlink demand. Depending on the active scenario, the orchestrator selects a corresponding TDD configuration.

3.2 Representation of Application Requirements

Application communication requirements are represented through an Application AAS, providing a machine-readable description of the communication needs associated with individual application functions (scenarios). For the intralogistics crane use case, requirements are derived from two representative operational scenarios:

Scenario A: Material Handling Control: Communication focuses on machine status, control commands, safety messages, and coordination. Strict timing requirements (e.g.,

< 50 ms transmission time) and high reliability are critical, while throughput demand remains moderate.

Scenario B: Camera-Assisted Monitoring: Camera-based functions for collision avoidance, load detection, and environment modeling generate video streams and significantly higher uplink traffic. While transmission time and reliability remain important, sufficient uplink capacity becomes the critical constraint.

The Application submodel is shown in Figure 4. It provides a structured representation of application-level communication requirements and their relation to the functions that generate them. Each application function (scenario) is associated with requirements describing transmission time, reliability, traffic demand, and uplink and downlink characteristics. This allows functions of the same application to expose distinct network requirements. For example, crane control has stringent transmission time and reliability constraints with moderate data-rate requirements, whereas camera-assisted monitoring requires substantially higher uplink capacity.

The UML representation provides the structural basis for implementing these requirements as standardized AAS submodel elements accessible through the Application AAS interface. Thus, the orchestrator can operate on explicitly represented communication requirements without knowledge of the application's internal implementation.

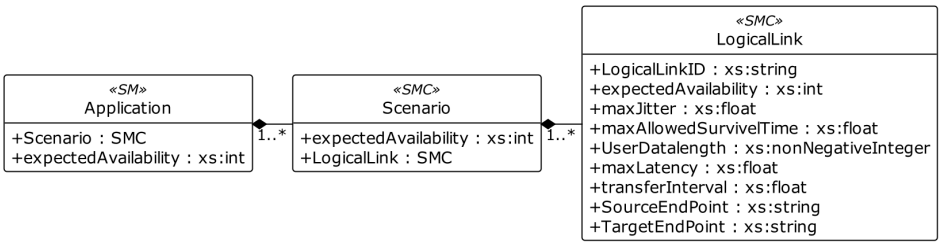


Fig. 4: UML class diagram of the Application Requirements Submodel used to represent communication requirements associated with application functions.

Table 1 summarizes the communication characteristics of the two scenarios, whose different uplink demands and transmission time constraints affect the suitability of 5G TDD configurations.

Tab. 1: Communication characteristics of the evaluated operational scenarios.

Requirement	Manufacturing Control	Camera Monitoring
Primary function	Control/Safety	Environment Sensing
Transmission time sensitivity	High	Medium-High
Reliability requirement	High	High
Uplink demand	Low to Medium	High
Downlink demand	Medium	Medium
Data type	Commands/Status	Video/Sensor Data

3.3 Representation of Network Capabilities

The communication network is represented through a Network AAS that standardizes the description of available capabilities. In the considered 5G system, these capabilities include different Time Division Duplex (TDD) configurations with distinct uplink and downlink resource allocations, supporting different application communication profiles.

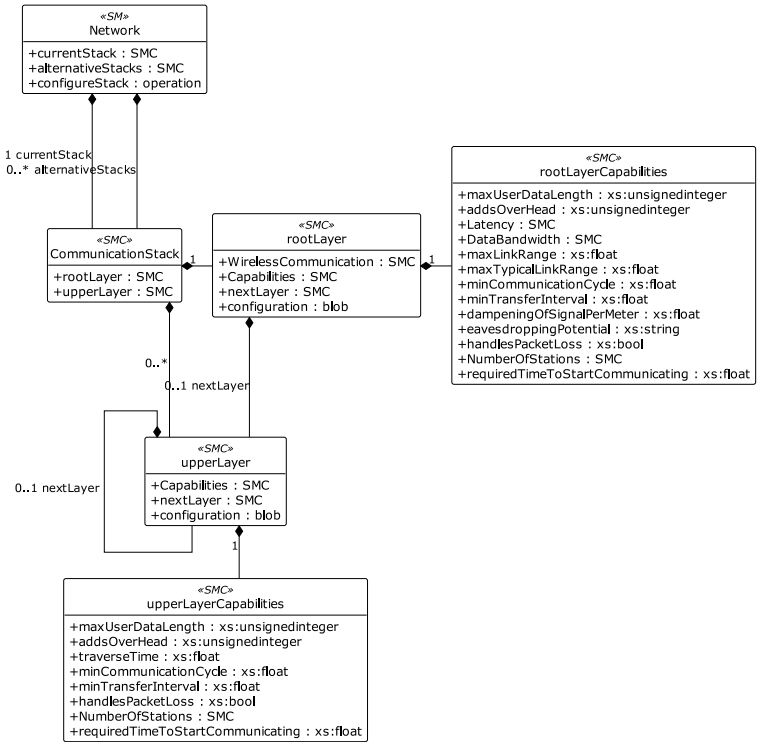


Fig. 5: UML class diagram of the Network Capabilities Submodel.

The Network AAS exposes the current and alternative configurations that the Network Management System (NMS) can activate. External factors, such as environmental conditions, may also limit available capabilities.

3.4 Orchestration Workflow

The orchestration process matches the application requirements represented by the Application AAS (Subsection 3.2) with the capabilities represented by the Network AAS (Subsection 3.3). As illustrated in Figure 3, the workflow proceeds as follows:

1. **Request:** A user or process requests an application function.

2. **Requirement Retrieval:** The orchestrator retrieves the function-specific communication requirements from the Application AAS.
3. **Capability Query:** The available TDD configurations are retrieved from the Network AAS.
4. **Matching & Decision:** The orchestrator matches the requirements with the available configurations and selects a suitable configuration.
5. **Activation:** The selected configuration is activated through the NMS.

The AAS runtime enables interoperable information exchange between the application and communication domains.

4 Experimental Setup

To evaluate the proposed AAS-based orchestration approach, an industrial intralogistics crane use case was emulated in the laboratories of the Research Institute of Electrical Communication (RIEC), Tohoku University, Japan. The objective of the experimental setup was to reproduce the communication environment of the original industrial demonstrator deployed in Germany while enabling controlled evaluation of different 5G network configurations.

4.1 Laboratory Environment

Figure 6 presents the overall architecture of the experimental environment. The setup consists of a physical domain containing the emulated industrial assets and a digital domain containing the AAS infrastructure and orchestration services. The physical and digital domains are interconnected through the RIEC 5G system, enabling end-to-end communication between the industrial assets and the orchestration components.

The physical domain contains the crane emulation devices, camera emulation systems, display devices, measurement equipment, and 5G user equipment (UE). The digital domain hosts the Application AAS, the Network AAS, the EmKoI4.0 orchestrator, and the AAS runtime environment. The AAS runtime and MQTT-based infrastructure are deployed in Germany, while the industrial assets and the 5G network infrastructure are located at RIEC in Japan. This setup enables realistic evaluation of distributed Industry 4.0 communication scenarios involving geographically separated digital and physical components.

4.2 Hardware and Software Components

The crane application was emulated using two NXP i.MX development boards equivalent to those employed in the original industrial demonstrator. These devices generated the

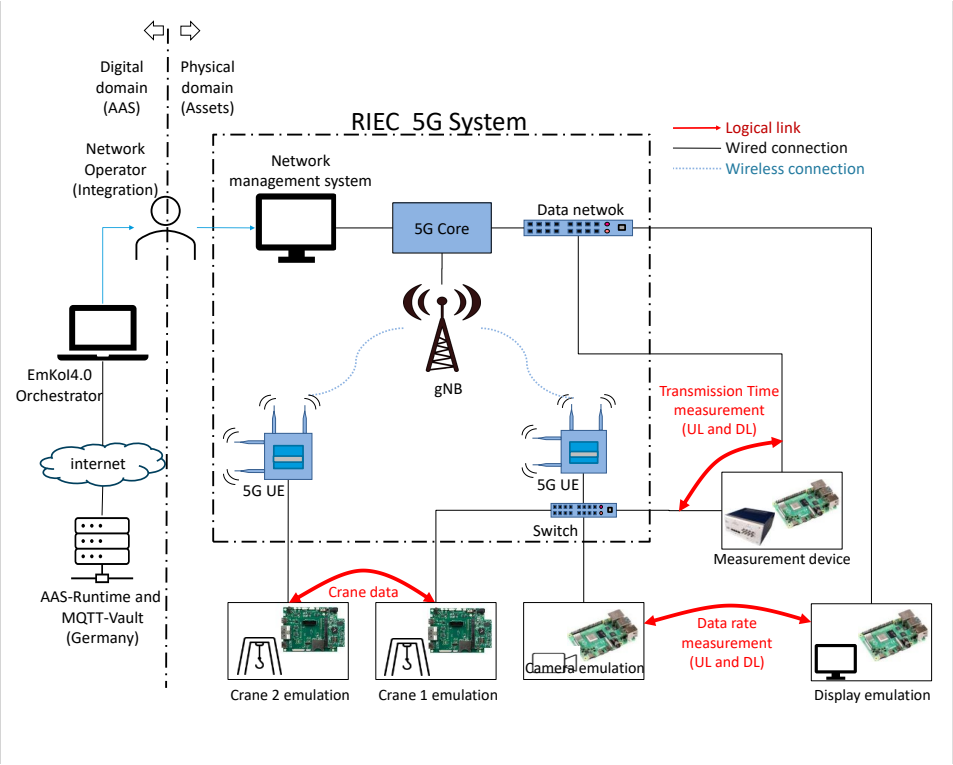


Fig. 6: Experimental setup implemented at RIEC integrating the physical and digital domains through the 5G network.

communication traffic associated with crane operation and exchanged data through the 5G network. Additional devices were used to emulate camera-based monitoring functions, display systems, and network measurement services. The camera emulation system generated configurable traffic patterns representing different video transmission scenarios, while dedicated measurement devices collected communication performance metrics.

The orchestration was deployed in a Raspberry Pi and interacts with the Application AAS and Network AAS through the AAS runtime environment. Based on the information retrieved from the AAS models, the orchestrator determines suitable network configurations according to the communication requirements of the active application scenario.

4.3 5G Network Infrastructure

The communication infrastructure is based on the RIEC private 5G system consisting of a 5G Core, a gNB, network management services, and multiple 5G user equipment devices.

The network supports configurable TDD, allowing different uplink and downlink resource allocations to be evaluated.

The productive traffic generated by the industrial applications is exchanged between the emulated assets through the 5G network, while management traffic is exchanged between the orchestrator and the AAS infrastructure through Ethernet. This separation allows the impact of network configuration changes on application communication performance to be evaluated independently from the orchestration processes.

4.4 Performance Measurements

The experimental evaluation focuses on two key performance indicators: data rate and transmission time. Measurements were performed for both uplink and downlink communication directions. Dedicated monitoring devices were used to collect throughput measurements and transmission time statistics while the crane and camera applications generated representative industrial traffic.

To represent different operational conditions, the camera emulation system was configured with different video transmission rates corresponding to scenarios similar to standard-definition (SD) and high-definition (HD) video streaming. These traffic profiles generate different uplink communication demands and therefore allow evaluation of the impact of application requirements on network performance.

4.5 Evaluated Timeslot Configurations

The experiments investigated multiple 5G timeslot configurations with different uplink and downlink resource allocations. For the purpose of this paper, two representative configurations are selected for detailed analysis.

The first configuration represents a downlink-dominant resource allocation in which the ratio between uplink and downlink resources is configured as 1UL:3DL. This configuration prioritizes downlink capacity and is expected to be suitable for scenarios with relatively low uplink requirements.

The second configuration represents a more balanced allocation with a ratio of 1UL:2DL. Compared to the first configuration, additional resources are assigned to uplink communication, making it potentially more suitable for camera-based monitoring applications that generate significant uplink traffic.

Table 2 summarizes the evaluated configurations.

The following section presents the experimental results obtained for the different operational scenarios and analyzes how the evaluated configurations affect communication performance.

Tab. 2: Evaluated 5G timeslot configurations.

Configuration	UL:DL Ratio	Characteristic
Config 1	1:3	Downlink-dominant
Config 2	1:2	Balanced

5 Results and Discussion

This section presents and discusses the experimental results for the evaluated 5G timeslot configurations, assessing the influence of uplink and downlink resource allocation on transmission time and data rate.

5.1 Transmission Time Analysis

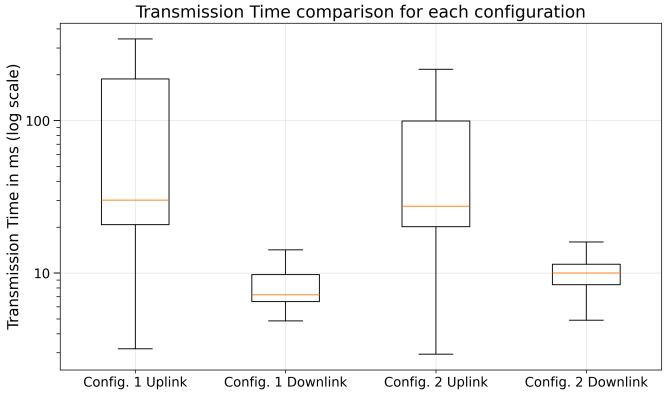


Fig. 7: Transmission time comparison for the evaluated timeslot configurations.

The results indicate that the selected timeslot configuration significantly influences communication performance, particularly in the uplink direction. The downlink-dominant configuration (Config 1) allocates fewer resources to uplink communication and consequently exhibits higher variability in uplink transmission times. In contrast, the balanced configuration (Config 2) provides additional uplink resources, resulting in more stable behavior and improved uplink performance. For downlink communication, both configurations show comparatively stable behavior. Since downlink resources remain sufficient in both evaluated configurations, only minor differences can be observed in the measured transmission times. These observations indicate that uplink-sensitive industrial applications can benefit from configurations that provide additional uplink resources.

The results underline the importance of considering application-specific communication requirements during network configuration. The manufacturing scenario requires a reliable low-latency communication, whereas the camera-assisted monitoring scenario requires a substantially higher uplink load. Therefore, the suitability of a timeslot configuration depends on the active operational scenario. Downlink-optimized configurations therefore serve the

former efficiently but degrade under video traffic, while uplink-enhanced configurations accommodate the latter without compromising overall performance.

These observations confirm that a single static network configuration cannot efficiently support all communication profiles encountered in Industry 4.0 environments.

5.2 Data Rate Analysis

The data rate results further demonstrate the impact of the selected timeslot configuration on communication performance, particularly in the uplink direction. As shown in Figure 8, the balanced configuration (Config 2) achieves substantially higher uplink data rates, typically between approximately 165 and 180 Mbps, whereas the downlink-dominant configuration (Config 1) remains around 52–63 Mbps. This difference reflects the additional uplink resources available in Config 2. In the downlink direction, both configurations achieve comparable mean data rates of approximately 90 Mbps. However, Config 2 exhibits higher downlink variability, which was not anticipated. These results indicate that increasing uplink resource allocation substantially improves uplink capacity, while the observed downlink data rate remains comparable on average between the two configurations.

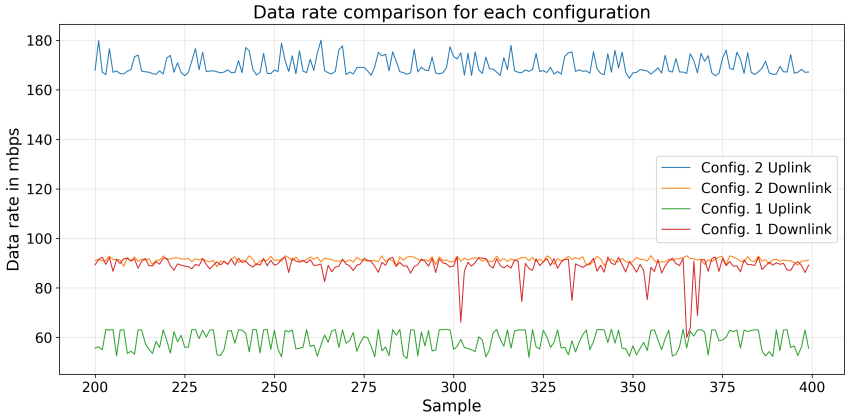


Fig. 8: Data rate comparison for the evaluated timeslot configurations.

The obtained results demonstrate that configurable 5G networks provide significant potential for supporting heterogeneous industrial communication requirements. The evaluated timeslot configurations exhibit different performance characteristics, indicating that no single configuration is optimal for all operating conditions.

The combination of AAS-based information modeling and orchestration mechanisms enables communication networks to adapt according to application demands. This capability is particularly relevant for industrial environments in which production processes, machine functions, and communication requirements change dynamically over time.

Although the present study focuses on two representative timeslot configurations, the proposed approach is not restricted to these configurations and can be extended to additional network parameters and optimization objectives.

6 Conclusion and Future Work

Industrial applications impose uplink and downlink communication requirements that vary with the active function. In the intralogistics crane use case, material-handling control requires reliable, low-latency exchange of commands and status, with moderate throughput demand, whereas camera-assisted monitoring produces high uplink demand through video transmission. The evaluated 5G timeslot configurations, 1UL:3DL and 1UL:2DL, exhibit different transmission-time characteristics, particularly in the uplink, and are therefore suited to different communication profiles. AAS submodels represent both application requirements and network capabilities, and the proposed orchestrator uses this information to select a suitable 5G configuration. The experimental results demonstrate the feasibility of adaptive 5G network management driven by these application requirements.

Future work will evaluate additional 5G configuration parameters beyond timeslot allocation and will investigate alternative RLC operation modes. These modes will be compared through a quantitative reliability–latency analysis under the industrial traffic profiles of this study. A further direction is dynamic online network reconfiguration, in which the orchestrator updates the active 5G configuration as application requirements change during operation.

Acknowledgment

This work was partially funded by the EmKoI4.0 project (Grant No. 01MJ22014A–D), funded by the German Federal Ministry for Research, Technology and Space (BMFTR). This work was also supported by the joint cooperation project between the Research Institute of Electrical Communication (RIEC), Tohoku University, Japan, and ifak – Institut für Automation und Kommunikation e.V., Germany, under Grant No. 2025A14.

References

[3ra] 3rd Generation Partnership Project (3GPP): NR; NR and NG-RAN Overall Description; Stage 2, TS 38.300, 3GPP.

[3rb] 3rd Generation Partnership Project (3GPP): Service Requirements for Cyber-Physical Control Applications in Vertical Domains, TS 22.104, 3GPP.

[5G21] 5G Alliance for Connected Industries and Automation (5G-ACIA): 5G for Connected Industries and Automation, tech. rep., 5G-ACIA, 2021.

[5G26] 5G Alliance for Connected Industries and Automation (5G-ACIA): Mechanisms of Performance Degradation Due to Usage of Different TDD Patterns in Neighboring 5G TDD Networks, 5G-ACIA Industrial 5G User Guide, 2026.

[Ap23] Apresia Systems: Operation Manual ApresiaAERO-Manager, TD61-7862, version 1.0.3, APRESIA Systems, 2023.

[Ca23] Cainelli, G. P. et al.: Managing the 5G System Based on Production Requirements Using an Asset Administration Shell Approach. IFAC-PapersOnLine 56 (2), pp. 2108–2114, 2023.

[Ca24a] Cainelli, G. et al.: Automated Adaptation of Industrial Communication Networks to Meet Application Requirements through Asset Administration Shell. In: Kommunikation in der Automation 2024. Lemgo, Germany, pp. 109–121, 2024.

[Ca24b] Cainelli, G. et al.: Flexible Reconfiguration of 5G Local Networks over Industry 4.0 Digital Twin. In: Automation 2024. VDI Verlag, pp. 699–712, 2024.

[Ka25] Kankanamge, H. K. P. et al.: Flexible Transmit Power Reconfiguration in Private 5G Networks using the Asset Administration Shell. In: Proceedings of the 7th Global Power, Energy and Communication Conference (GPECOM). Bochum, Germany, pp. 1048–1055, 2025.

[Ol24a] Olaya, S. S. P. et al.: Flexible Reconfiguration of Industrial 5G Networks over Asset Administration Shell. In: Proceedings of the IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA). Padova, Italy, pp. 1–8, 2024.

[Ol24b] Olaya, S. S. P. et al.: Management of Industrial 5G Networks over Asset Administration Shell. In: Proceedings of the IEEE 20th International Conference on Factory Communication Systems (WFCS). Pp. 1–4, 2024.

[Pl20] Plattform Industrie 4.0: Digital Twin and Asset Administration Shell Concepts and Application in the Industrial Internet and Industrie 4.0, tech. rep., Plattform Industrie 4.0, 2020.

[Pl22] Plattform Industrie 4.0: Details of the Asset Administration Shell – Part 1, tech. rep., Plattform Industrie 4.0, 2022.

Agent Communication Protocols for Industrial Ecosystems

Parva Yazdani[1], Piyumi Pathirathna[1], and Mario Thron[1]

Abstract: Agent communication protocols provide mechanisms for industrial coordination, but their suitability depends on workflow requirements and supporting controls. This study reviews agent communication protocols and potential industrial applications, then proposes a representative intralogistics transport-allocation workflow and implements it using selected protocols. The Model Context Protocol (MCP) provides shared tool access, while Agent2Agent (A2A), Agent Network Protocol (ANP), and Agora provide alternative paths for requesting bids and assigning work. A controlled coordination experiment was conducted using the implemented workflow to compare the selected inter-agent protocols while varying executor fleet size, repeated interaction, and participation conditions. All variants completed the workflow under common application rules. Median coordination latency increased with fleet size and was lower in repeated rounds. Median latency was similar for A2A and Agora, while the tested ANP implementation path showed higher medians, particularly in first-contact rounds and at larger fleet sizes. The results suggest that all three alternatives are viable for implementing the selected workflow, and that protocol choice should consider coordination latency alongside the mechanisms the workflow requires. A separate fault experiment examined four selected security and dependability scenarios relevant to industrial task allocation: agent identity substitution, replay of a task award request, task-state recovery after interruption, and authorization of agent actions. A2A and ANP support peer-identity verification, but these mechanisms do not establish plant approval; Agora leaves authentication and authorization entirely outside its scope. A2A provides a Task ID and status lookup, but task status remained unresolved after interruption. Action authorization and duplicate-execution protection also depended on controls beyond the protocols. The fault experiment shows that application and deployment components are required alongside protocols to cover security and dependability requirements. Taken together, the experiments show that industrial agent integration requires matching protocol combinations to workflow requirements and adding controls where protocol mechanisms fall short.

Keywords: Agent communication, Agent protocols, Industrial agent systems

1 Introduction

Recent advances in artificial intelligence (AI), particularly large language models (LLMs), have increased interest in agentic workflows. LLMs can interpret operational information, reason over context, and support planning and action selection; connected to tools and other agents, they can enable greater autonomy under defined tasks and controls. In industry, such systems could support resource allocation and coordination. A manufacturing study illustrates this potential: an LLM-based system improved makespan and stability in a physical shopfloor evaluation [ZH26].

Agents use communication protocols to exchange information, request work, and access services. Surveys characterize this landscape, benchmarks compare protocol behavior,

[1] ifak e.V., Werner-Heisenberg-Straße 1, 39106 Magdeburg, Germany, parva.yazdani@ifak.eu; piyumi.pathirathna@ifak.eu; mario.thron@ifak.eu

and security analyses examine risks [AN26; DU25; YA25b]. However, evidence remains limited on how protocol mechanisms support industrial workflows, affect coordination, and handle faults. This study examines that gap by linking protocol mechanisms to workflow requirements, coordination costs, and dependability controls. It reviews protocols and industrial applications, then evaluates a representative workflow through coordination and fault experiments. Four research questions guide the study:

1. Which current protocols are relevant to industrial workflows, and what roles and mechanisms do they provide?
2. What industrial applications are possible, and how can a representative workflow and communication configuration be selected and implemented?
3. How do selected implementations perform as workflow scale, repeated interaction, and participation conditions change?
4. How do they handle selected faults, and what application or deployment components are needed?

The contributions are a review of how protocol roles relate to industrial applications, a measured comparison of a representative workflow under common conditions, and an examination of fault handling that attributes observed protections and identifies additional controls.

2 Agent Communication Protocols

Agent communication protocols address different interaction roles and are not direct substitutes. Yang et al. distinguish context-oriented protocols, which connect agents to tools and resources, from inter-agent protocols, which structure communication among agents [YA25b]. The wider landscape also includes protocols for other interaction settings. This study examines four complementary approaches, selected to represent different communication roles and mechanisms:

- **MCP** provides tool and resource access. A host–client–server architecture uses JSON-RPC and capability negotiation to expose tools, resources, and prompts [MCP]. The host manages connection permissions and security policies; MCP does not determine whether an operation is permitted by an industrial application.
- **A2A** structures inter-agent collaboration through messages and tasks, with progress updates and result artifacts. Agent Cards describe capabilities and security requirements [A2A]. Application policy still determines which actions a participant may perform.
- **ANP** combines agent description and discovery with identity and secure communication. Its `did:wba` method uses signed requests to authenticate control of an identity and is designed for cross-platform identity authentication [CH26; DID], making ANP a relevant candidate for cross-company agent communication.

- **Agora** is a specialized meta-protocol for agent exchanges based on shared protocol descriptions. Research on Agora also explores communication through shared protocol descriptions and generated routines [MA24b]. The Agora Protocol Working Standard specifies two-party JSON exchanges and references to protocol documents, but leaves authentication, authorization, and negotiation beyond protocol discovery out of scope [Agora]. The research prototype and the standard describe related but distinct aspects of Agora.

The ProtocolBench study compares several agent communication protocols on task success, latency, message overhead, and behavior under failure. Its results vary by scenario, supporting requirement-specific selection rather than a universal protocol ranking. Its question-answering, security-probe, queueing, and agent-failure scenarios do not establish performance in industrial workflows [DU25]. Security analyses also distinguish protocol-level threat modeling from implementation behavior. Anbiaee et al. combine lifecycle threat modeling of MCP, A2A, Agora, and ANP with a measurement-driven MCP case on unintended tool-provider selection [AN26]. This work identifies protocol and integration risks, but leaves open how protocol mechanisms support industrial task assignment and completion under common application conditions, and where observed protections originate. The evaluation addresses this workflow-level gap while distinguishing protocol provisions from implementation and application controls.

3 Industrial Applications and Workflow Design

Industrial agent applications could support supplier coordination, production resource allocation, maintenance, inventory replenishment, intralogistics, and quality follow-up. Published work illustrates some of these patterns. Zhao et al. describe agents bidding to select processing machines on a physical shop floor [ZH26]. Berlec et al. present a proof of concept for decentralized warehouse management involving storage offers and capacity pooling [BE25]. Ray's review outlines manufacturing and supply-chain workflows as illustrative A2A applications [RA25]. Other candidate patterns include comparing supplier offers, coordinating maintenance diagnosis, and arranging inspection follow-up.

To illustrate possible agentic applications in industrial domains spanning production, inventory, and logistics, this section proposes a workflow: *intralogistics transport allocation*. A welding cell requires a part held in another factory zone. A cell agent checks stock and requests offers from executor agents representing transport resources such as automated guided vehicles or forklifts. Each request carries a task identifier, part, quantity, source zone, and destination cell. The executor status includes availability, position, and energy; bids add an estimated arrival time and energy cost. The cell agent selects one executor, awards the task, and records its completion. It ranks bids by estimated arrival time, then energy cost, and then executor identifier. If no eligible bid arrives within the response period, it issues no award. Figure 1 shows this shared interaction sequence.

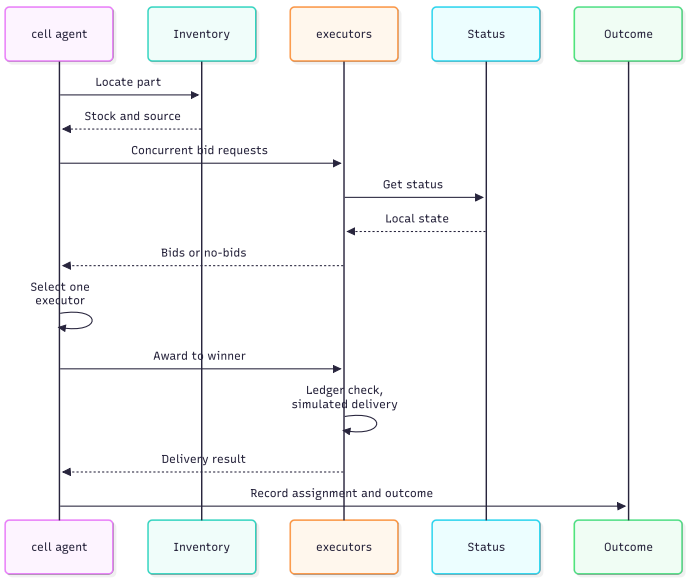


Fig. 1: Shared intralogistics transport-allocation sequence.

The case has a clear request, comparable offers, a single assignment, and an observable outcome. The workflow requires consistent inventory information, comparable bid fields, a bounded response period, and one unambiguous award. It also requires the application to distinguish identity, participation approval, and action permission; prevent duplicate task execution; and correlate completion with the request. Overall, this representative scenario provides a basis for considering how industrial agent systems can be implemented, which features address the use-case requirements, and what safety measures an industrial setting requires.

The implementation and protocol paths for this workflow are as follows. Because MCP provides tool and resource access, it serves as the common layer. The cell agent uses three MCP services: inventory lookup, executor-status lookup, and outcome recording. Each executor reads its status through MCP, while the cell agent uses MCP to locate material and record completion. After an award, the selected executor checks an application task ledger and returns a completion result. Completion is represented as a workflow state; vehicle motion is outside this configuration. The communication configuration combines this common MCP tool layer with alternative inter-agent paths that handle agent discovery or protocol negotiation, where needed, and carry bids and awards:

- **A2A** uses an Agent Card to describe each executor's advertised capabilities. The cell agent sends bid requests and receives bids as Messages; it sends the award as a

Message that starts a Task lifecycle. Figure 2 shows how A2A maps to this use case alongside MCP.

- **ANP** supports peer discovery and retrieval of an Agent Description and DID document. The description identifies the peer's interfaces, and DID-WBA can authenticate calls to them [CH26; DID]. Agents can then exchange the task request, bid, award, and completion through an application interface.
- **Agora** carries bids and awards under a hashed Protocol Document in one conversation [Agora; MA24b]. Peers with a shared document use its hash; if they lack a shared document and support protocol negotiation, they can agree on a common Protocol Document for the task. The implementation supports both pre-shared Protocol Documents and a proposal-and-acceptance path using a deterministic policy.

These protocol-specific workflows can be implemented using current protocol specifications and publicly available libraries. For example, the Python-based Agent Protocols Reference Kit (v0.2.0) provides shared protocol APIs and components [PA26a]; the implementation uses `a2a-sdk` 1.1.2, `anp` 1.0.3, and `mcp` 2.1.1. This combination of protocol software, a design that meets the workflow's operational and safety requirements, and the necessary computing, network, and system interfaces provides the basis for implementing agent systems in industrial ecosystems.

4 Evaluation Approach

Protocol specifications describe mechanisms and expected interactions, while existing studies examine protocols in other scenarios. Evidence therefore remains limited on how the selected implementations coordinate an industrial workflow under common application conditions, how they respond to faults, and which system layer provides protection. To address this gap, two experiments were conducted using the intralogistics workflow described in Section 3: one examined coordination behavior and the other examined responses to selected faults.

The coordination experiment compared A2A, ANP, and Agora with the same MCP services, task fixtures, selection rule, and application controls. It varied the executor fleet size (1, 3, 5, or 10) and whether participation was stable or dynamic. In the stable condition, all executors were configured before a run; in the dynamic condition, one executor was absent initially and introduced during setup using an externally supplied endpoint. Each of the 24 configurations used one warm up and ten measured runs, yielding 240 measured runs and 24 warm-up runs in total. Each run comprised five rounds: a first-contact round followed by four repeated rounds, with a five second executor response deadline. For each round, the experiment measured end-to-end latency from the inventory lookup through outcome recording and inter-agent latency from the first protocol event through simulated delivery. Figure 2 illustrates these timing boundaries on the A2A path; the same boundaries were used for all variants. The experiment also recorded selected protocol events, serialized business-payload bytes, and MCP calls; these measures do not capture complete wire

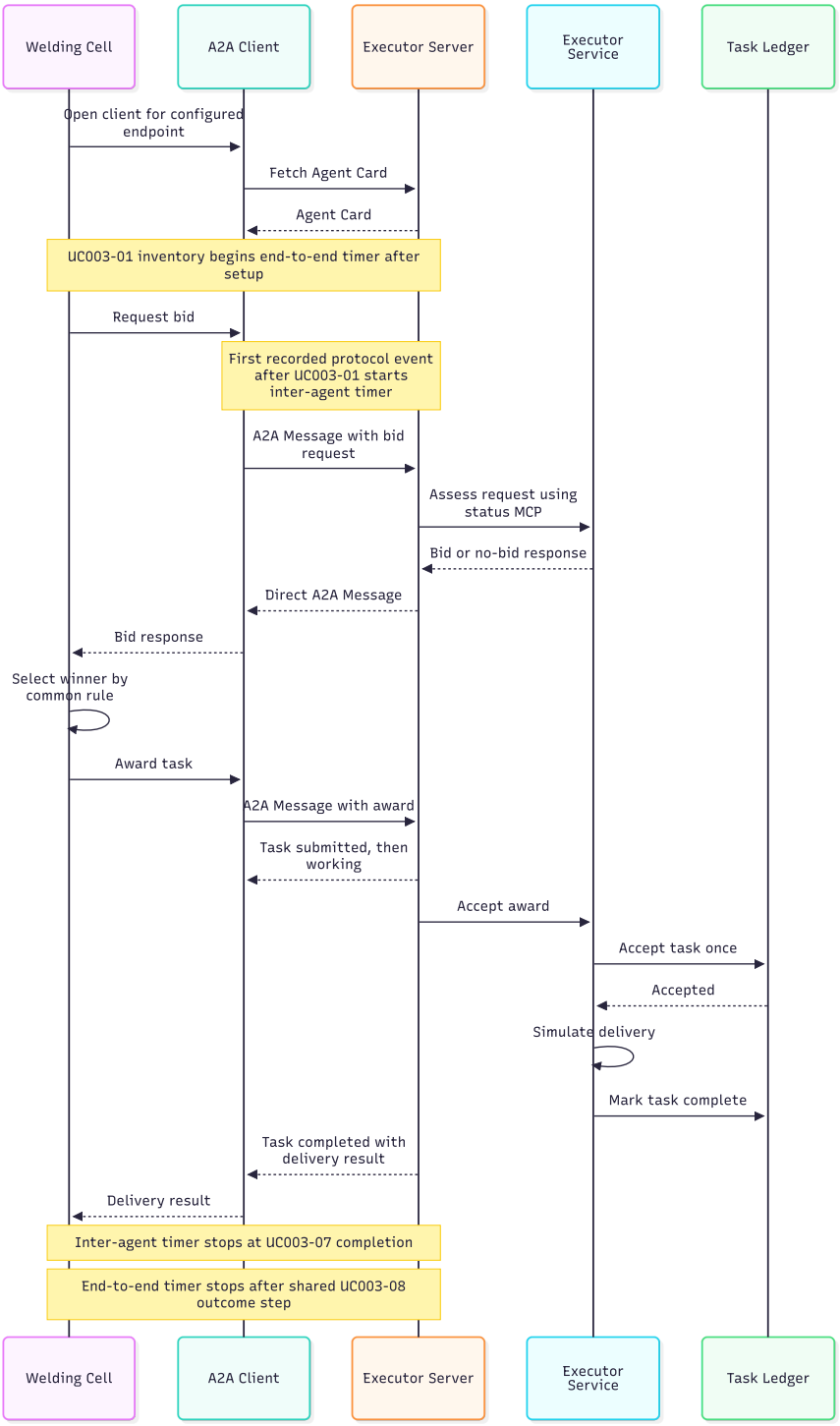


Fig. 2: A2A sequence for one executor, with MCP services.

traffic. With ten independent runs per configuration, latency summaries are descriptive: medians represent typical values and are less affected by unusually high observations, while end-to-end latency also reports the interquartile range to show spread. All valid observations were retained, and repeated rounds were treated as nested within runs.

The four fault scenarios were selected to cover distinct security and dependability concerns in industrial task allocation: admitting the intended peer, preventing duplicate execution, resolving task status after interruption, and enforcing action permissions.

- **Identity admission:** the configured executor endpoint was replaced by a rogue endpoint claiming that identity and presenting valid credentials; the test checked whether it could return a bid.
- **Duplicate execution:** after the original delivery, the experiment harness replayed the exact award request to the executor endpoint once to check whether a second delivery was prevented.
- **Task state recovery:** the executor was stopped after durably accepting the award but before delivery, then restarted to check whether its task status could be recovered.
- **Action authorization:** an application-level permission policy at the executor allowed `request_bid` but denied `award` for the authenticated observer. For each request, the executor authenticated the observer, identified the requested action, and checked it against this policy before returning a bid or award denial.

Each protocol fault baseline had five valid runs. For the crash case, A2A was tested in two five-run conditions to assess whether `GetTask` could retrieve task status after restart: in one, the cell received a Task ID before the crash and queried it afterward; in the other, it received no ID to query. Since credential validity alone does not establish plant approval, the identity admission scenario was also repeated with an application-level approved-identity check as a safeguard. The check compared the endpoint identity established by credential verification with the plant-approved identity for the configured executor and rejected a mismatch before requesting a bid. All four Agora fault profiles used HTTPS, with the verified TLS certificate fingerprint serving as the endpoint identity. The runs recorded caller-visible outcomes and whether fault handling came from a protocol-defined mechanism or from the application and its supporting software.

Both experiments ran on x86-64 Linux under WSL2 with Python 3.13.13, using fresh local processes and state for each coordination run and fault attempt. Coordination exchanges used loopback HTTP for A2A and Agora and HTTPS for ANP; MCP calls in the coordination experiment used stdio. Deterministic fixtures and simulated delivery were used throughout, without LLM decisions or physical transport. Processor and memory specifications were not recorded. Because transport differed across the coordination protocols, latency differences cannot be attributed to protocol alone. The experiments did not measure real network or robot behavior, setup and onboarding time, or complete wire traffic. The experiment repository is publicly available and provides the implementation, configurations, documentation, sequence diagrams, and curated results for inspection and rerunning [PA26b].

Tab. 1: End-to-end coordination latency in milliseconds: median (IQR). Each configuration has ten first rounds and forty repeated rounds nested within the same ten runs. Setup is excluded.

		Stable		Dynamic	
Protocol	Fleet	First	Repeated	First	Repeated
A2A	1	260.3 (24.6)	224.4 (30.4)	260.6 (37.8)	217.1 (28.3)
	3	418.9 (48.7)	357.0 (30.6)	417.2 (44.0)	358.0 (38.2)
	5	567.1 (30.6)	481.8 (40.0)	575.2 (59.0)	484.4 (63.3)
	10	904.3 (63.1)	823.4 (59.0)	883.2 (46.3)	815.3 (50.1)
ANP	1	341.2 (18.2)	233.9 (29.4)	332.8 (14.7)	241.7 (27.1)
	3	625.6 (68.8)	389.1 (29.8)	627.8 (52.1)	417.3 (55.4)
	5	891.1 (22.1)	551.2 (43.6)	883.5 (64.3)	569.7 (67.5)
	10	1,521.2 (90.9)	930.6 (73.7)	1,533.4 (68.7)	943.8 (99.0)
Agora	1	271.6 (47.5)	218.1 (21.3)	262.8 (17.0)	221.0 (18.6)
	3	409.8 (48.7)	370.0 (27.6)	418.2 (29.8)	359.9 (42.2)
	5	559.2 (27.3)	489.1 (54.9)	585.6 (85.6)	511.1 (48.3)
	10	924.2 (50.1)	850.8 (59.9)	910.0 (36.2)	853.0 (84.6)

5 Results

Coordination experiment results are as follows: The experiment included 240 measured runs of five rounds each. Across all 1,200 rounds, the expected winner was selected, one simulated delivery was completed, and a consistent outcome was recorded. These outcomes were observed with the same application controls across variants and therefore do not establish that each protocol supplies them independently. Table 1 reports the end-to-end results for all fleet sizes and participation conditions.

End-to-end median latency increased with fleet size across protocols and participation conditions, while repeated-round medians were lower than first-round medians in all 24 configurations. Inter-agent medians followed the same scale pattern. At ten executors, first-round inter-agent medians were 752–781 ms for A2A and Agora and 1,386–1,390 ms for ANP; repeated-round medians were 726–760 ms for A2A and Agora and 841–850 ms for ANP. Stable and dynamic conditions had similar medians with no consistent ordering. The dynamic condition demonstrates that an executor introduced during setup could subsequently participate in the common workflow. No consistent increase in coordination latency was observed after introduction, although the cost of onboarding itself remains unmeasured. The MCP workload was identical across protocols at each fleet size: each round used $N + 2$ calls for a fleet of N executors (3, 5, 7, and 12 calls for fleet sizes of 1, 3, 5, and 10, respectively).

Fault experiment results are as follows:

- **Identity admission:** across five baseline runs per protocol, the cell received a bid from the rogue endpoint in every run. In five safeguard runs per protocol, a plant-approved

identity mapping rejected the endpoint after credential verification and before the cell sent a bid request.

- **Duplicate execution:** across five runs per protocol, replaying the exact award request after the original delivery resulted in exactly one delivery per run. The A2A reference-kit message-ID cache returned the cached response without a second business dispatch; this suppression is inferred from the response and dispatch count. For ANP and Agora, the replay reached application logic, where the task ledger identified the already completed task and prevented a second delivery.
- **Task state recovery:** In every run, task status remained unresolved and no delivery occurred. ANP and Agora each had five runs; A2A had two conditions of five runs each. When the cell received a Task ID before the crash, `GetTask` returned `TaskNotFoundError` after restart in all five runs. Without a Task ID, the cell could not query task status. The tested ANP and Agora profiles had no award-status lookup.
- **Action authorization:** across five runs per protocol, an authenticated observer was allowed `request_bid` and received a bid, but `award` was denied in every run.

6 Discussion

Current agent communication protocols offer complementary mechanisms that can support the integration of agent systems into industrial workflows. MCP provides shared access to tools and resources; A2A structures message- and task-based interactions; ANP provides agent descriptions, discovery, and identity mechanisms; and Agora supports exchanges based on shared protocol descriptions. Together, these roles suggest that an industrial workflow may use multiple protocols rather than rely on one protocol for every interaction. Prior work describes several potential industrial applications [BE25; RA25; ZH26]. This paper introduces a cross-domain intralogistics transport-allocation scenario spanning production, inventory, and logistics. The workflow was implemented with MCP as the common tool layer and A2A, ANP, or Agora as alternative inter-agent protocols for bids and awards. By holding the workflow and application rules constant while varying the inter-agent protocol, the study shows how workflow requirements can guide protocol integration.

To assess how the selected inter-agent implementations coordinated the same workflow as fleet size, repetition, and participation varied, the experiment held application rules constant. All three variants completed the task with MCP providing the common tool layer, showing that alternative inter-agent protocols can be integrated with the same industrial services for this case. Across all three variants, latency increased as the executor fleet grew, consistent with the added coordination needed to solicit and process more bids. The measurements do not identify a clear overall choice: A2A and Agora had similar latency, while ANP was slower in the tested configurations and offers a different discovery and identity model. Protocol selection therefore depends on workflow requirements as well as measured latency; the observed timings describe the complete implementation paths, which used different transports.

To examine how the selected protocol implementations addressed fault scenarios relevant to secure and dependable industrial deployment, the fault experiment tested identity substitution, award replay, task-state recovery after interruption, and action authorization. The experiment identified protocol-level mechanisms relevant to safety and dependability: A2A and ANP provide peer-identity verification, while A2A provides task identifiers and status lookup. However, peer authentication does not establish plant approval; the authenticated identity must also be checked against the plant-approved identity. Agora leaves authentication and authorization entirely outside its scope. A task reference enables recovery only when the corresponding state is retained. Action authorization and duplicate-execution protection also require controls beyond the protocol. These findings show that dependable industrial workflows require assessing which requirements protocol mechanisms and their implementations cover, then providing missing controls at the application or deployment level.

These fault findings motivate two proposed supporting components. An organizational agent registry and access-control service could address the gaps in plant approval and action authorization. It could map each verified peer identity to an approved agent record, including its owner and operational status, and associate the agent with roles. An authorization policy could map those roles to permitted actions and data. Requests would then be checked for identity, plant approval, and action-level permission before an agent acts or accesses shared information. A specific proposal is to adapt the European Digital Identity (EUDI) Wallet framework [EC23] for plant-level agent systems. The plant registry could issue an agent-held credential attesting that the agent is approved for the plant and has specified roles. A verifier would bind that credential to the agent's identity in the selected protocols through suitable protocol-specific integrations. The plant access-control service would still determine which actions and data the agent may access. The feasibility and suitability of adapting this framework for agent systems, including its integration with the selected protocols, require further study.

A second proposed component is a persistent plant knowledge base that could maintain an asset registry, expose current operational status, and retain task state across interruptions. The proposed agent registry could form part of this plant asset registry. Figure 3 illustrates how the proposed agent registry and access-control service could govern access to a persistent plant knowledge base shared by agents and existing industrial systems. Authorization policies linked to the registry could govern which agents and plant systems may read, create, or update each record. Authorized agents and existing industrial systems could continuously update the knowledge base with current plant status and task state. Persistent task records could support status lookup and reconciliation after interruption, while stable task references and recovery logic would still be needed before work is retried. A vendor white paper proposes an ontology-based knowledge graph for structuring industrial data for AI agents [MA26], addressing one basic element relevant to this direction. However, further research on designing, adapting, implementing, and evaluating the proposed plant-level knowledge base remains future work.

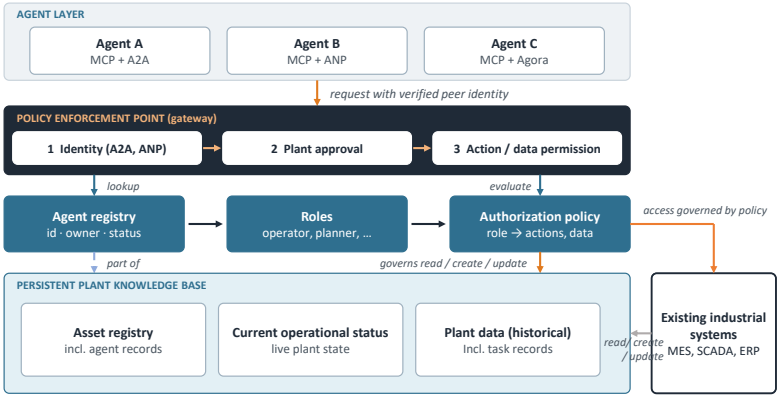


Fig. 3: Proposed architecture for agent access control and a persistent plant knowledge base.

7 Conclusion

This study reviewed agent communication protocols and industrial application opportunities, then implemented and evaluated an intralogistics transport-allocation workflow. The protocols were selected for their different roles: MCP for tool access, A2A for message and task interactions, ANP for discovery and identity, and Agora for exchanges based on shared protocol descriptions. The coordination experiment compared A2A, ANP, and Agora with MCP as the common tool layer, varying fleet size, repeated interaction, and participation conditions. All three variants completed the workflow under the same application rules. Coordination latency increased with fleet size and was lower in repeated rounds. A2A and Agora had similar median latencies, while the tested ANP implementation path had higher medians. These measurements describe the performance of the implemented workflow under the tested conditions; latency differences cannot be attributed to the protocols alone because the transport configurations differed. The results show that all three alternatives can support the selected workflow. Protocol choice should therefore consider measured latency alongside the communication mechanisms and controls the workflow requires.

The fault experiment showed that protocols need supporting controls for secure and dependable operation. Peer authentication required an additional check against plant-approved identities. Implementation and application controls prevented duplicate execution and enforced action permissions. Task status remained unresolved after interruption, showing the need to retain task state and provide recovery logic. These findings show that protocol mechanisms must be combined with application and deployment controls to meet the workflow's security and dependability requirements. The fault experiment also helped identify two supporting components that could address the security and dependability

requirements examined in this study. An agent registry and access-control service could establish plant approval and enforce permissions for actions and data. A persistent plant knowledge base could maintain asset and operational information and retain task state to support recovery after interruption. Future work should design, implement, and evaluate these components and their integration with the selected protocols.

References

[A2A] A2A Project: Agent2Agent Protocol Specification, version 1.0.0, 2026, https://a2a-protocol.org/v1.0.0/specification/, accessed: 09/26/2026.

[Agora] Agora Protocol: Agora Protocol Working Standard, 2025, https://agoraprotocol.org/docs/protocol/specification, accessed: 09/26/2026.

[AN26] Anbiaee, Z. et al.: Security Threat Modeling for Emerging AI-Agent Protocols: A Comparative Analysis of MCP, A2A, Agora, and ANP, 2026, arXiv: 2602.11327.

[BE25] Berlec, T. et al.: Exploring Decentralized Warehouse Management Using Large Language Models: A Proof of Concept. Applied Sciences 15 (10), 5734, 2025.

[CH26] Chang, G.: Agent Network Protocol Specifications (ANP 1.1), 2026, https://agentnetworkprotocol.com/en/specs/.

[DID] Agent Network Protocol: DID-WBA Method Design Specification, version ANP 1.1, 2026, https://github.com/agent-network-protocol/AgentNetworkProtocol/blob/v1.1/03-did-wba-method-design-specification.md, accessed: 09/26/2026.

[DU25] Du, H. et al.: ProtocolBench: Which LLM MultiAgent Protocol to Choose?, 2025, arXiv: 2510.17149.

[EC23] European Commission: The European Digital Identity Wallet Architecture and Reference Framework, 2023, https://digital-strategy.ec.europa.eu/en/library/european-digital-identity-wallet-architecture-and-reference-framework, accessed: 10/05/2026.

[MA24b] Marro, S. et al.: A Scalable Communication Protocol for Networks of Large Language Models. 2024, https://arxiv.org/abs/2410.11905.

[MA26] Mandi tereza, K.: Building Ontology-Driven Intelligence for Industrial AI Agents, 2026, https://www.hivemq.com/resources/building-ontology-driven-intelligence-for-industrial-ai-agents/.

[MCP] Model Context Protocol: Architecture, version 2025-11-25, 2025, https://modelcontextprotocol.io/specification/2025-11-25/architecture, accessed: 09/26/2026.

[PA26a] Pathirathna, P.; Yazdani, P.; Thron, M.: Agent Protocols Reference Kit, version v0.2.0; citation metadata commit cb7f7c9, 2026, https://github.com/piyumipat/agent-protocols-reference-kit/tree/v0.2.0, accessed: 10/01/2026.

[PA26b] Pathirathna, P.; Yazdani, P.; Thron, M.: Agent Protocols Industrial Use Cases, version commit d241f1a, 2026, https://github.com/piyumipat/agent-protocols-industrial-use-cases/commit/d241f1a, accessed: 10/01/2026.

[RA25] Ray, P. P.: A Review on Agent-to-Agent Protocol: Concept, State-of-the-art, Challenges and Future Directions, 2025, DOI: 10.36227/techrxiv.174612014.42157096/v1.

[YA25b] Yang, Y. et al.: A Survey of AI Agent Protocols, 2025, arXiv: 2504.16736.

[ZH26] Zhao, Z. et al.: A Large Language Model-Based Multi-Agent Manufacturing System for Intelligent Shopfloors. Advanced Engineering Informatics 69, p. 103888, 2026.

Evaluation of Hybrid V2X Co-Simulation Frameworks for Research and Qualification in Mixed-Traffic Deployments

Maxim Friesen[1], Jannik Peters[1], Shayan Naveed[1], and Jürgen Jasperneite[1,2]

Abstract: Hybrid V2X combines network-based communication over 5G with direct communication via ITS-G5, LTE-V2X and NR-V2X sidelink. Such systems are studied in simulation before deployment, but comparisons of V2X simulation frameworks rarely consider hybrid operation or their use in both research and teaching. This paper assesses the documented capabilities of nine open-source frameworks against the research and qualification needs of a planned mixed-traffic campus intersection with automated rail and road vehicles, non-equipped cars and vulnerable road users. None of them combines the required channel-access models, 5G Uu capabilities and hybrid operation, their simulator versions diverge, and their open NR sidelink implementations exist only for ns-3 and MATLAB. We therefore propose a toolchain with common scenario definitions, compatible message formats, comparable packet traces and common metric definitions across simulation, laboratory and field, in which a simulator is selected for each learning or research question at basic, advanced and expert level.

Keywords: Hybrid V2X, Co-Simulation, Emulation, NR Sidelink, 5G Campus Network, Qualification

1 Introduction

Vehicle-to-Everything (V2X) communication allows vehicles and roadside infrastructure to share information for coordinating movements at intersections and warning of potential conflicts with vulnerable road users (VRUs). Hybrid V2X combines a direct path between stations via ITS-G5 or the PC5 sidelink of LTE-V2X and NR-V2X with a network path over the 5G Uu interface. The direct path works without network coverage, but its range is limited by obstruction and channel load. The network path reaches stations beyond radio range and supports Quality of Service (QoS) control and applications running close to the vehicles through Multi-access Edge Computing (MEC), but depends on coverage and network load. Combining both allows applications to select a path according to their latency and reliability requirements and the available connectivity [Al21; Kh24].

Simulation is used to study these paths and their effects on vehicle interactions before deployment. For example, Veins connects the traffic simulator Simulation of Urban Mobility (SUMO) with the network simulator OMNeT++. Available frameworks focus on different aspects, such as the Intelligent Transport Systems (ITS) stack of the European Telecommunications Standards Institute (ETSI), cellular networks or perception. Few model both

[1] Technische Hochschule Ostwestfalen-Lippe, Institut für industrielle Informationstechnik, Campusallee 6, 32657 Lemgo, Deutschland, maxim.friesen@th-owl.de; jannik.peters@th-owl.de; shayan.naveed@stud.th-owl.de; juergen.jasperneite@th-owl.de

[2] Fraunhofer IOSB-INA, Campusallee 1, 32657 Lemgo, Deutschland, juergen.jasperneite@iosb-ina.fraunhofer.de

communication paths in comparable detail, although simulation results for hybrid operation depend on how channel access, blockage and interference are modelled [Kh24; Pe26]. Framework selection must therefore consider which models represent these aspects for the intended research question, not only which communication technologies are supported.

A single-rail test track for the automated MonoCab vehicle is planned on the Innovation Campus Lemgo [Te]. The MonoCab will share roads with cars, pedestrians and cyclists. Research and qualification infrastructure is being developed in the KAMOS project to investigate and teach automated driving, V2X communication and cooperative intersection management. Learners and researchers should be able to explore the same scenarios in simulation, examine them with model vehicles and compare the results with field measurements. This requires consistent message formats, metric definitions and analysis methods. Simulation should also help explain why communication succeeds or fails by recording reception events and modelled loss causes at every receiver, whereas field logs cover only instrumented receivers. Taking one intersection as the reference use case, this paper derives requirements for research and qualification, assesses the documented capabilities of nine open-source simulation frameworks against them and proposes a toolchain that selects a simulator for each learning or research question.

2 Related Work

Previous work has compared V2X simulation frameworks based on their technical capabilities and suitability for specific applications. A survey of five OMNeT++-based libraries examines their support for 5G-V2X simulation [Ba21]. Another survey reviews simulation studies of LTE-based cellular and hybrid V2X communication [HB21]. The survey in [SR24] derives requirements for connected automated vehicles, while the paper in [Sa25] weights requirements according to their importance for each use case. These approaches provide a methodological basis for assessing frameworks against requirements derived from the intended use case.

Hybrid V2X combines communication technologies whose interactions also need to be represented in simulation. The simulation study in [Go24] reproduced measured packet delivery ratios for ITS-G5 and LTE-V2X sidelink, but could not model their mutual interference because it lacked a shared radio channel model. The testbed evaluation in [Kh24] demonstrated improved reliability by selecting between direct and network-based communication for each message. Simulating such operation requires both the behaviour of the individual technologies and the selection between communication paths to be represented. Combining standard V2X messages with a simulated 5G network also requires integrating the software that provides them. The work in [KB23] coupled Artery, which provides the ETSI ITS stack, with the 5G simulator Simu5G to investigate collective perception supported by an edge server. Artery had to be adapted because the frameworks required different OMNeT++ versions. Such adaptations add installation and maintenance work, as the combined setup must remain compatible when its underlying frameworks are updated.

The paper in [Tr24] introduced a web interface for the network simulator ns-3 and SUMO to make simulation easier for beginners. The teaching platform in [Wa25] combined miniature vehicles and roadside units for practical V2X exercises. Connecting such exercises with simulation and field experiments requires consistent scenario definitions, message formats and analysis methods, so that learners can compare communication behaviour across the environments. However, the reviewed comparisons are limited to one simulator family, to LTE-based V2X, to perception or to a coarse view of communication, and the studies on hybrid communication, integration and teaching each examine only a single setup. None of them assesses frameworks for hybrid V2X research and teaching together. This paper addresses this gap by assessing available frameworks against the needs of both and by proposing how they can be combined across simulation, laboratory and field.

3 Background

3.1 Hybrid V2X Access Technologies

ITS-G5, LTE-V2X and NR-V2X use different methods to access the radio channel. ITS-G5 follows IEEE 802.11p and uses Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), in which stations check whether the channel is busy before transmitting. Decentralized Congestion Control (DCC) limits transmissions based on the channel busy ratio, the proportion of time the channel is sensed as busy. LTE-V2X provides two ways to allocate transmission times and frequency resources on the PC5 sidelink. The base station can assign them to vehicles (Mode 3), or vehicles can select them autonomously (Mode 4). Autonomous selection uses Sensing-Based Semi-Persistent Scheduling (SB-SPS), which reserves resources for periodic transmissions [Al21]. NR-V2X, standardized in Release 16, also distinguishes between network-scheduled allocation (Mode 1) and autonomous selection (Mode 2) [Al21]. It also supports unicast and groupcast, with Hybrid Automatic Repeat Request (HARQ) feedback for retransmissions [Al21].

The network path uses the Uu interface of a 5G standalone (SA) network. Traffic is assigned to QoS flows whose 5G QoS Identifier (5QI) defines characteristics such as priority and packet delay budget. An edge server can relay messages between vehicles. This forms a vehicle-to-network-to-vehicle (V2N2V) path. Because it shares network resources with other traffic, its performance also depends on network load and resource allocation [Kh24]. The field measurements in [Go24] show that the performance of ITS-G5 and LTE-V2X depends on radio obstruction and interference when the technologies share a channel. The simulation comparison in [Pe26] shows that accounting for the surrounding geometry improves agreement with field measurements and that modelling interference between technologies changes reception predictions. Representing these effects is therefore important when using simulation to compare communication technologies under the conditions of a deployment.

3.2 ETSI ITS Stack and Message Conformance

The ETSI facilities layer provides V2X application messages. Cooperative Awareness Messages (CAMs) announce a station's position and movement, while Decentralized Environmental Notification Messages (DENMs) warn about events. Collective Perception Messages (CPMs) share objects detected by sensors. SPATEM and MAPEM carry Signal Phase and Timing (SPaT) information from traffic lights and intersection layout information (MAP), respectively. Maneuver Coordination Messages (MCMs) support the coordination of planned vehicle movements. Message structures are defined in Abstract Syntax Notation One (ASN.1). The Unaligned Packed Encoding Rules (UPER) specify how these structures are encoded for transmission. The Basic Transport Protocol (BTP) and GeoNetworking provide transport and networking for direct communication. Using compatible message formats makes it possible to decode simulated packets and field traces with the same tools.

3.3 Co-Simulation and Emulation

Co-simulation couples separate simulators so that vehicle movement and communication can be studied together. Coupling frameworks such as Veins use the control interface of SUMO to exchange vehicle positions and control commands with a network simulator in synchronized time steps [Ah25]. Studies of sensor detection and its effect on CPMs additionally require a perception model. CARLA, a driving simulator with camera, lidar and radar models, can provide simulated sensor data for these studies [Ah25], but requires a graphics processing unit (GPU) [SR24]. V2X frameworks are typically built on a general network simulator such as OMNeT++ or ns-3 and extend it with V2X models. INET provides network protocol models for OMNeT++-based setups, while MATLAB is also used for V2X radio simulation. Frameworks therefore depend on specific versions of these simulators and supporting libraries. Conflicting version requirements can prevent frameworks from being combined directly, making code adaptations necessary and increasing installation and maintenance effort. Beyond simulation, network emulation runs the simulated network in real time so that actual applications or devices can communicate through it, and hardware-in-the-loop (HIL) setups connect physical components, such as controllers or radios, with the simulation.

4 Reference Use Case and Requirements

4.1 Campus Intersection and Staged Validation

Figure 1 shows the planned traffic-calmed campus intersection that serves as the reference use case. The automated MonoCab and road shuttle interact with conventional cars, pedestrians and cyclists. Since pedestrians and cyclists carry no V2X devices, infrastructure cameras around the intersection detect their presence. The roadside unit (RSU) shares information

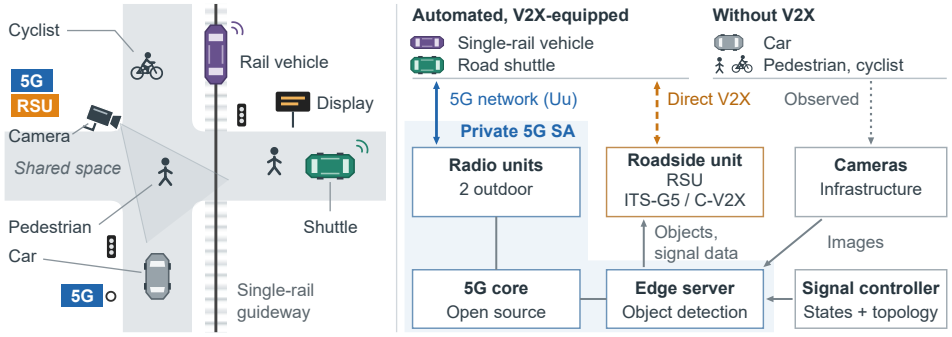


Fig. 1: Schematic representation of the reference mixed-traffic use case and its communication infrastructure.

about the detected objects in CPMs. The planned infrastructure includes a private 5G SA network in the 3.7 to 3.8 GHz band and an edge server. The RSU and on-board units support ITS-G5 and cellular V2X (C-V2X). A traffic signal supplies signal states and intersection topology for SPATEM and MAPEM. During the experiments, vehicles exchange V2X messages with each other and with the infrastructure, both over the private 5G network and directly via ITS-G5 and C-V2X.

A cooperative system coordinates the automated vehicles and resolves conflicts by a fixed priority order. VRUs and non-equipped cars rank above the automated vehicles because they cannot participate in message-based coordination. This order decides which vehicle yields and is separate from communication priorities such as ETSI traffic classes or the 5QI. Delayed or lost position updates can cause the cooperative system to base yielding decisions on outdated vehicle positions. Simulating cooperative behaviour therefore requires both the coordination logic and the communication behaviour of the network and direct paths. Experiments progress from simulation through a small-scale laboratory to the real intersection. The laboratory is planned as an indoor model track with several robot vehicles and model traffic signals, connected through the private 5G network for controlled V2X experiments and teaching. Common scenario definitions, message formats and measurement methods are intended to make results comparable across these environments. Qualification activities address message structure, channel access and message loss at basic level, 5G QoS and edge computing at advanced level, and open research questions at expert level. The intended learning outcomes of these levels are oriented towards levels 6 to 8 of the European Qualifications Framework (EQF) [Co17].

4.2 Simulation Requirements

Table 1 lists the requirements derived from the reference use case. COM follows from the two communication paths, INT from the roadside infrastructure, edge server and priority

Tab. 1: Requirements for research and qualification around the reference use case.

ID	Requirement
Communication fidelity (COM)	
COM-1	Direct path with standard channel access of ITS-G5 (CSMA/CA), LTE-V2X (SB-SPS) and NR-V2X sidelink (Mode 2).
COM-2	Network path over 5G NR Uu with scheduling under load, QoS flows and 5QI.
COM-3	Hybrid operation with several interfaces per node, selection between the network and direct paths, and interference between technologies sharing one channel.
COM-4	Blockage by buildings and vehicles derived from the scenario geometry.
COM-5	Congestion control on the ITS-G5 channel (DCC) based on the channel busy ratio.
System integration (INT)	
INT-1	CAM, DENM and CPM services in ASN.1 UPER, BTP/GeoNetworking on the direct path, extensible to further messages such as MCM.
INT-2	SPATEM from signal states, MAPEM from intersection geometry and CPMs from detected objects.
INT-3	Edge hosts that run applications such as perception and coordination and relay messages between vehicles (V2N2V).
INT-4	Traffic simulation with road and rail vehicles and VRUs. Custom applications, such as priority handling, run on selected participants and can access the positions of non-equipped participants.
Operation and qualification (OPS)	
OPS-1	Transition from simulation to real devices in laboratory and field (same stack, emulation or HIL), compatible message profiles, comparable traces.
OPS-2	Reception and loss statistics per receiver with the modelled loss causes, without additional code.
OPS-3	Low effort for installation, dependencies, maintenance and hardware, without commercial licences, for use on student computers.

order, and OPS from the staged validation and qualification levels. Since they cover research and qualification as a whole, a single experiment usually only needs some of them, such as ITS-G5 with DCC for a basic exercise on congestion.

5 Framework Evaluation

The assessment covers nine open-source frameworks that model at least one communication path of the reference use case, as listed in Table 2. Artery-C, which extended Artery with cellular V2X models, is not rated, because its repository, formerly public under GPL-3.0 [He24], has been removed. The ratings are based on publications, documentation and the repository state.[3] Only the official version of each framework counts. Components

[3] Repositories as of 1 October 2026: SUMO github.com/eclipse-sumo/sumo, OMNeT++ github.com/omnetpp/omnetpp, INET github.com/inet-framework/inet, ns-3 gitlab.com/nsnam/ns-3-dev, CARLA github.com/carla-simulator/carla, Veins github.com/sommer/veins, Artery github.com/riebl/artery, Vanetza github.com/riebl/vanetza, OpenCV2X github.com/brianmc95/OpenCV2X, Simu5G github.com/Unipisa/Simu5G, 5G-LENA gitlab.com/cttc-lena/nr, VaN3Twin github.com/DriveX-devs/VaN3Twin, Eclipse MOSAIC github.com/eclipse-mosaic/mosaic, OpenCAMS github.com/minhaj6/carla-sumo-omnetpp-cosim, WiLabV2Xsim github.com/V2Xgithub/WiLabV2Xsim, CrowNet github.com/roVer-HM/crownet.

Tab. 2: Assessment of the frameworks against the requirements of Table 1 (● full, ◐ partial, ○ no documented support; for OPS-3 low, medium and high effort).

	COM					INT				OPS		
Framework	1	2	3	4	5	1	2	3	4	1	2	3
Veins 5.3.1	◐	○	○	●	○	○	○	○	◐	○	●	●
Artery + Vanetza	◐	◐	◐	●	●	●	◐	◐	●	●	●	◐
OpenCV2X	◐	◐	◐	●	●	◐	○	◐	◐	●	●	○
Simu5G 1.7.0	○	●	◐	○	○	○	○	●	◐	●	●	◐
5G-LENA 5.1 + NR V2X 1.1	◐	●	○	◐	○	○	○	○	○	○	◐	◐
VaN3Twin	●	◐	◐	●	●	●	◐	◐	◐	●	◐	◐
Eclipse MOSAIC 25.2	◐	◐	◐	○	○	◐	○	◐	◐	○	◐	●
OpenCAMS	◐	○	○	●	○	○	○	○	◐	○	●	◐
WiLabV2Xsim 6.1	●	○	◐	●	●	○	○	○	○	○	◐	◐

inherited from other frameworks count where its build integrates them, whereas third-party research extensions are discussed in the findings but do not raise a rating. The row for 5G-LENA combines its official version and the separately released NR V2X extension, which are assessed as separate configurations rather than one compatible installation. A requirement is fully met if the official version provides every element of its row in Table 1 and partially met if it provides some of them, so the ratings describe documented capabilities, not tested model accuracy. OPS-3 instead estimates the documented effort of initial setup, maintenance of a fixed environment and migration of existing experiments. It is low if none of these requires substantial work and academic use needs neither a commercial licence nor a GPU, medium if one of these conditions is not met, and high if setup is not reproducible or maintenance has evidently ended. The findings F1 to F7 explain the ratings.

F1. ***Incomplete support for hybrid V2X.*** Veins centres on IEEE 802.11p, Artery and VaN3Twin on the ETSI ITS stack, OpenCV2X and WiLabV2Xsim on channel access of the direct path, Simu5G and 5G-LENA on 5G radio access, and Eclipse MOSAIC and OpenCAMS on traffic and perception. None of the assessed frameworks provides the complete combination of channel-access models, 5G Uu capabilities and hybrid operation covered by COM-1 to COM-3. VaN3Twin and WiLabV2Xsim model interference between ITS-G5 and C-V2X sharing a channel, but WiLabV2Xsim has no network path and each VaN3Twin vehicle uses a single technology [Pe26]. Artery and OpenCV2X place ITS-G5 and LTE Uu on one node, and MOSAIC applications can send over an ad-hoc or a cellular module. However, none of them demonstrates selecting the path for each message, their network path is limited to LTE or an abstract cellular model, and interference between technologies is not modelled. Simu5G switches flows between Uu and its own direct link, which is not based on the 3GPP sidelink specifications. Path selection as in the testbed of [Kh24] therefore requires own application logic and cannot be combined with 5G NR QoS flows and a standard direct path in any single framework.

F2. ***Diverging simulator versions.*** Artery still builds on OMNeT++ 5.6.2, whereas Simu5G 1.7.0 is compatible with OMNeT++ 6.1 to 6.4 and notes that many releases require changes to existing configurations. On the ns-3 side, 5G-LENA 5.1 builds on ns-3.48, while VaN3Twin and the NR V2X extension each depend on their own ns-3 branch. Combinations of the ETSI stack with the 5G network therefore exist only in individual projects, such as the integration in [KB23] and CrowNet, which maintains its own forks of INET, Artery, Veins and Simu5G. Integration effort and fork maintenance are thus a major obstacle.

F3. ***NR sidelink outside OMNeT++.*** Release-16 NR sidelink with both resource allocation modes and switching to Uu was implemented in OMNeT++ in Artery-C [He24], but is no longer available. Among the reviewed frameworks, open NR sidelink implementations exist only in ns-3, with the NR V2X extension of 5G-LENA [Al21] and VaN3Twin [Pe26], and in MATLAB, with WiLabV2Xsim at the physical and medium access layer. The 5G-LENA extension supports Mode 2 with broadcast, groupcast, unicast and HARQ feedback, but not Mode 1. In OMNeT++, OpenCV2X covers LTE-V2X Mode 4 only, so NR sidelink studies with an OMNeT++-based ETSI stack need another simulator.

F4. ***Uneven ETSI conformance.*** Artery and VaN3Twin encode messages in ASN.1 and transport them via BTP and GeoNetworking. Vanetza, the stack underlying Artery, is a standalone library that can also be used outside the simulator with ITS-G5, C-V2X or GeoNetworking over IP multicast, and OpenCV2X reuses the Artery stack for CAMs and DENMs [Mc21]. Eclipse MOSAIC, in contrast, uses its own binary format for CAMs, DENMs, SPaT and infrastructure-to-vehicle information messages, which Wireshark cannot decode.

F5. ***Weak infrastructure coupling.*** Vanetza and VaN3Twin contain SPATEM and MAPEM in ASN.1, but the official versions of Artery and VaN3Twin do not derive them from a signal program. The paper in [Wá23] connected a traffic signal controller to a digital twin and generated SPaT and MAP from the SAE J2735 definitions, but serialized them as JSON in the HIL demonstration. None of the assessed frameworks therefore generates standard SPATEMs and MAPEMs from a simulated signal program. CPMs are generated in Artery from field-of-view sensor models, including roadside sensors, and in VaN3Twin from a range-based sensor on SUMO vehicles. CARLA integrations in OpenCAMS [Ah25] and VaN3Twin support more detailed sensor simulation. The MCM specification TS 103 561 has not been published yet. VaN3Twin added an MCM service to its official version, which implements version 1.1 of the MCM ASN.1 module, but its correspondence to a particular ETSI draft is not established.

F6. ***Paths towards hardware.*** The Artery testbed connects a device under test through a CUBE adapter or a software-defined radio. OpenCV2X inherits the radio route, which requires the external SEA V2X API, but only for ITS-G5. Its operation and the current availability of the API were not verified. VaN3Twin provides an emulation mode whose packets can be decoded in Wireshark. Its predecessor ms-van3t shares one code base with the field stack OScar, so the study in [Ge24] ported the VRU basic service, which sends awareness messages for VRUs, from simulation to field hardware in a few steps. Simu5G runs as a real-time emulator, in which real applications communicate through a simulated

5G network with MEC. Testbeds and emulation thus connect simulation with laboratory and field without merging simulators at code level, although the endpoints still need adapters.

F7. ***Teaching and observability.*** For teaching, OMNeT++ offers a graphical runtime environment that animates the message exchange step by step, while ns-3 relies on external visualizers. VaN3Twin includes a web-based one. WiLabV2Xsim requires MATLAB, or Octave in a separate branch, and CARLA or ray tracing require GPUs, which limits their use on student computers. For observability, Veins records lost packets per receiver by cause, which Artery and OpenCAMS inherit through the Veins radio. Simu5G records losses per receiver at the MAC and RLC layers, and OpenCV2X distinguishes half-duplex, sensing, propagation and collision errors [Mc21], but has not been updated since 2023. The ns-3-based frameworks expose losses through trace sources, which often require own code, although the NR V2X extension of 5G-LENA includes collectors for decoding errors. No reception log per message exists in a format shared by the frameworks.

6 Reference Toolchain

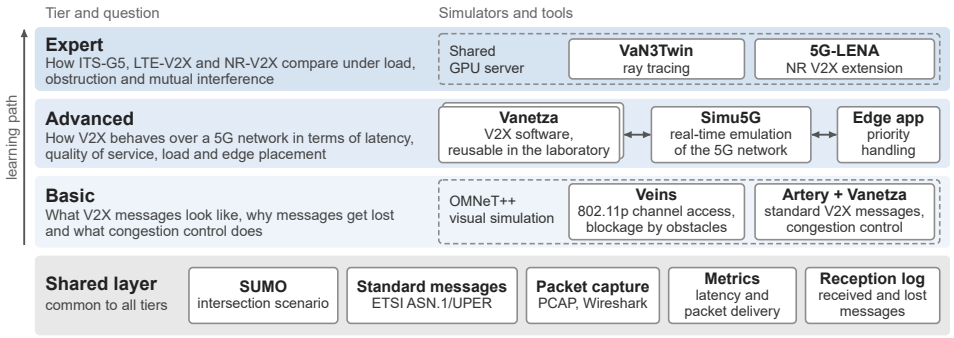


Fig. 2: Toolchain with a shared layer and a simulator for each learning or research question.

Even experiments that only use a private 5G network need standard ETSI messages together with NR Uu and QoS flows. No assessed framework provides both, and combinations at code level remain tied to the simulator versions they were built on (F2). We therefore propose a toolchain that selects a simulator for each learning or research question and connects the simulators through a shared layer of common definitions and formats, intended to keep results comparable across simulators and across simulation, laboratory and field (Figure 2). The toolchain and its tiers are derived from the assessment and have not yet been implemented or evaluated.

A common SUMO model of the intersection serves as the reference scenario for laboratory and field and contains the guideway of the rail vehicle, the paths of the VRUs and the signal program. Vanetza provides standard message encoding in Artery, in emulation and on real devices (F4), while VaN3Twin and field devices with their own stacks remain comparable if they use a compatible message profile. Packet capture (PCAP) files can be decoded like field

traces in Wireshark, which needs an added dissector for MCMs. Common definitions of latency, jitter, packet delivery ratio, inter-arrival time, range and channel load are evaluated by the same scripts for simulated and measured traces. Finally, a common format for a reception log per message covers the receivers under test in laboratory and field and every receiver in simulation. It keeps three sources apart, namely the message content as encoded by the sender, the output of the radio model such as distance, received power and loss cause, and the decisions of the cooperative system, such as which vehicle yields.

On top of the shared layer, three tiers follow the basic, advanced and expert levels of qualification. The basic tier answers what CAMs, DENMs and CPMs look like, why messages get lost and what DCC does. It uses the graphical runtime environment of OMNeT++, where Veins introduces channel access and blockage by buildings and vehicles, and Artery with Vanetza adds the ETSI facilities and DCC. The Veins radio records lost messages per receiver with their causes (F7), whereas CPMs from sensor models need Artery's INET-based radio, in which these causes still have to be recorded. The advanced tier answers how 5QI, load and edge placement affect the latency of V2X messages over the private 5G network. It runs Simu5G as a real-time emulator (F6) with Vanetza processes as endpoints, which exchange their messages over the simulated 5G network with an edge application such as the priority handling of the cooperative system. Adapters have to carry the encoded messages over IP, map the endpoints to simulated devices and supply them with positions and time from the scenario. Since the ETSI stack runs outside the simulator, the version conflict between Artery and Simu5G (F2) does not arise. In return, the emulation has to run in real time, which limits the number of nodes and the traffic load of a scenario. The endpoints, adapters and edge application also have to be maintained separately. Because the emulation mirrors the laboratory, the same V2X software and edge application can later run on the model vehicles over the real 5G network. Interfaces, timing and clock synchronisation then have to be verified on the real devices. The expert tier addresses open research questions, such as how ITS-G5, LTE-V2X and NR-V2X compare under load, obstruction and mutual interference. VaN3Twin with ray tracing is used for these comparisons, and the NR V2X extension of 5G-LENA for studies of the Release-16 NR sidelink, within the limits noted in F3. Since ray tracing relies on GPUs (F7), this tier is intended to run on a shared server rather than on student computers. Path selection between Uu and direct communication together with 5G QoS is not covered by any tier (F1).

Not every simulator provides the shared layer yet, as Veins and 5G-LENA lack standard messages (Table 2), the packet recorder used by Artery writes only IP datagrams, not the GeoNetworking packets of the direct path, and the common reception log format has yet to be defined. Connecting the tools requires services that derive SPATEM and MAPEM from the signal program or controller and CPMs from object lists of infrastructure cameras or vehicle software. It also requires integrating existing bridges between the ETSI stack and the Robot Operating System (ROS) 2, through which the toolchain is intended to be connected to the vehicle software and to a driving simulator such as AWSIM.

7 Gaps and Research Opportunities

Four gaps remain between the assessed frameworks and the reference use case. The first gap is integration. No assessed framework combines 5G NR Uu with QoS flows and MEC with a standard ETSI stack on the direct path in one runtime. The closest combination, CrowNet, relies on its own forks based on Simu5G 1.4.3, a release that predates the QoS-flow support added in Simu5G 1.5.0 (F2). Emulation can bridge this gap for the network path, but hybrid operation, in which a node selects the path for each message, still requires a joint runtime or a causally synchronized coupling (F1). Studies in which ITS-G5 and C-V2X share a channel additionally need a common interference model. The second gap is NR sidelink. In the assessed versions, unicast sidelink with HARQ feedback is implemented only in the 5G-LENA extension, which contains no ETSI stack and has not been updated since 2024 (F3). Studying standard ETSI messages over it therefore requires additional integration. The third gap concerns perception and the rail vehicle. Between geometric sensor models and GPU-based camera simulation (F5), detection error models calibrated with the infrastructure cameras could represent the planned perception at moderate cost. SUMO provides rail motion models, but its built-in railway-crossing controller gives trains priority and therefore does not reproduce the intended priority order, so the rail vehicle's yielding has to be controlled explicitly. The last gap is validation. In the reviewed literature, simulation has been compared with measurements only for individual radio technologies [Ge24; Go24; Pe26], not for hybrid operation. An earlier field test with the rail vehicle used only the network path over a private 5G network [Gu23]. Running the same scenario with the shared layer in simulation, laboratory and field could help close this gap, because a deviation between two of them can then be traced to what changed from one to the other.

8 Conclusion and Outlook

This paper derived requirements for research and qualification around a mixed-traffic campus intersection and assessed the documented capabilities of nine open-source simulation frameworks against them. None of the assessed frameworks provides the complete combination of channel-access models, 5G Uu capabilities and hybrid operation. We therefore proposed a toolchain that selects a simulator for each learning or research question at basic, advanced and expert level and connects the simulators through a shared layer of common definitions and formats, intended to keep results comparable. Because a single experiment needs only a subset of the requirements, the criteria and ratings can also support simulator selection for other use cases. The next steps are implementing the shared layer and the three tiers, evaluating them in teaching and against measurements, contributing SPATEM and MAPEM services to the open frameworks, and comparing simulated and measured hybrid operation.[4]

[4] **Acknowledgement.** This work was funded by the German Federal Ministry of Research, Technology and Space (BMFTR) and the State of North Rhine-Westphalia under grant 13HAW029FQ (KAMOS). The authors are responsible for the content of this publication.

References

[Ah25] Ahmad, M. U. et al.: OpenCAMS: An Open-Source Connected and Automated Mobility Co-Simulation Platform for Advancing Next-Generation Intelligent Transportation Systems Research, 2025, arXiv: 2507.09186 [cs.SE].

[Al21] Ali, Z. et al.: 3GPP NR V2X Mode 2: Overview, models and system-level evaluation. IEEE Access 9, pp. 89554–89579, 2021.

[Ba21] Barbecho, P. et al.: An Evaluation of OMNeT++-based V2X Communication Frameworks: On the Path Towards 5G-V2X Simulations. In: ACM MSWiM. Pp. 75–78, 2021.

[Co17] Council of the European Union: Council Recommendation of 22 May 2017 on the European Qualifications Framework for lifelong learning. Off. J. Eur. Union C 189, pp. 15–28, 2017.

[Ge24] Genovese, A. et al.: ETSI-Compliant Protection of Connected Vulnerable Road Users: Simulation and Field Trial. In: IEEE VNC. IEEE, pp. 343–350, 2024.

[Go24] Gorospe, J. et al.: Performance Comparison of IEEE 802.11p and LTE-V2X Through Field-Tests and Simulations. In: IEEE VNC. IEEE, pp. 81–88, 2024.

[Gu23] Gustin, D. et al.: Outdoor Field Test of 5G-based V2X Communication for Real-Time Monitoring and Remote Control of a Monorail Vehicle. In: IEEE INDIN. IEEE, 2023.

[HB21] Hejazi, H.; Bokor, L.: A Survey on Simulation Efforts of 4G/LTE-based Cellular and Hybrid V2X Communications. In: TSP. Pp. 333–339, 2021.

[He24] Hegde, A. R.: Radio Resource Allocation in Cellular V2X: From Rule Based to Reinforcement Learning Based Approaches, PhD thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2024, https://doi.org/10.25593/open-fau-899.

[KB23] Kovács, G. A.; Bokor, L.: Implementation of MEC-Assisted Collective Perception in an Integrated Artery/Simu5G Simulation Framework. Sensors 23 (18), 7968, 2023.

[Kh24] Khalid, I. et al.: Optimizing hybrid V2X communication: An intelligent technology selection algorithm using 5G, C-V2X PC5 and DSRC. Future Internet 16 (4), 107, 2024.

[Mc21] McCarthy, B. et al.: OpenCV2X: Modelling of the V2X Cellular Sidelink and Performance Evaluation for Aperiodic Traffic, 2021, arXiv: 2103.13212 [cs.NI].

[Pe26] Pegurri, R. et al.: VaN3Twin: The multi-technology V2X digital twin with ray-tracing in the loop. IEEE Trans. Wireless Commun. 25, pp. 16812–16827, 2026.

[Sa25] Santos, J. et al.: A Requirement-based Methodology for Selecting Simulation Solutions in C-ITS Safety Applications. Int. J. ITS Res. 23 (2), pp. 1315–1330, 2025.

[SR24] Stepanyants, V. G.; Romanov, A. Y.: A Survey of Integrated Simulation Environments for Connected Automated Vehicles: Requirements, Tools, and Architecture. IEEE Intell. Transp. Syst. Mag. 16 (2), pp. 6–22, 2024.

[Te] Technische Hochschule Ostwestfalen-Lippe, Institut für Energieforschung: MONOCAB-Campusbahn am Innovation Campus Lemgo, https://www.ife-owl.de/forschung/projekte/monocab-campusbahn-am-innovation-campus-lemgo, accessed: 10/01/2026.

[Tr24] Trúchly, P. et al.: Remote Virtual Laboratory for Simulation of VANET Networks. In: ELMAR. IEEE, pp. 21–24, 2024.

[Wá23] Wágner, T. et al.: SPaT/MAP V2X Communication between Traffic Light and Vehicles and a Realization with Digital Twin. Comput. Electr. Eng. 106, 108560, 2023.

[Wa25] Wang, L. et al.: Application Research of a V2X Semi-Physical Simulation Platform in Vehicle–Road Collaboration Experiments. World Electr. Veh. J. 16 (6), 304, 2025.

Bridging Worlds between AAS and OPC UA

Georg Biehler [1] and Tino Bischoff [2],

Abstract: This paper addresses the fragmentation in industrial information caused by distinct metamodels, APIs, and protocols of Asset Administration Shell (AAS) and OPC UA, which leads to data silos for asset lifecycle versus operational data. The upcoming OPC 30271 specification bridges this gap by defining a common information model based on the expressive OPC UA metamodel. It introduces a bidirectional, lossless, and deterministic mapping between AAS and OPC UA, ensuring semantic identity preservation via a "roundtrip" guarantee. Additionally, OPC 30272 establishes a unified, backward-compatible communication layer. This integration eliminates data silos, simplifies client development, and fosters true interoperability, enabling robust digital twins and holistic asset views.

Keywords: Asset Administration Shell, OPC UA, Common information model, Bidirectional lossless mapping, OPC 30271 (mapping specification), OPC 30272 (communication layer), Interoperability, Digital twin, Metamodel mapping

1 Current Status

As of now, the specification of the IDTA defining the Asset Administration Shell (AAS) and the specification of the OPC Foundation defining OPC UA resulted in different protocols, APIs, metamodels, and in consequence information models. This has an impact onto both clients and servers.

[1] Siemens AG, DI AUT TI TRS STD, Siemenspromenade 1, 91058 Erlangen, georg.biehler@siemens.com
[2] Fraunhofer IOSB, Abteilung ILT, Fraunhoferstraße 1, 76131 Karlsruhe, tino.bischoff@iosb.fraunhofer.de

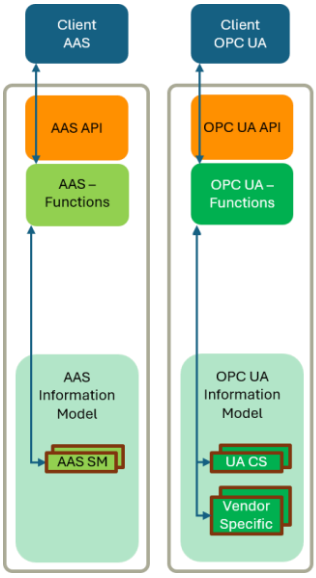


Figure 1 – Current status of accessing information of an asset

Assume that a client needs to access information regarding a particular asset, where information is related to its life cycle and to its operational state. This client needs to use two completely different ways regarding communication protocols and APIs to retrieve the information, whereas the information is hosted within the server in two separated information silos as shown in Figure 1.

To retrieve information related to the asset's life cycle, an AAS Client communicating to an AAS Server must be used. To access information related to the asset's operational state, an OPC UA client communicating to an OPC UA Server must be used.

The information itself is hosted in two separate information silos, one hosting the AAS information model and another hosting the OPC UA information model.

The server itself must implement two different information silos, the APIs for accessing the silos and two different protocol suites.

2 Bridging information silos

The upcoming specifications OPC 30271 addresses how to bring the information silos into one common information model as shown with the green box in Figure 2.

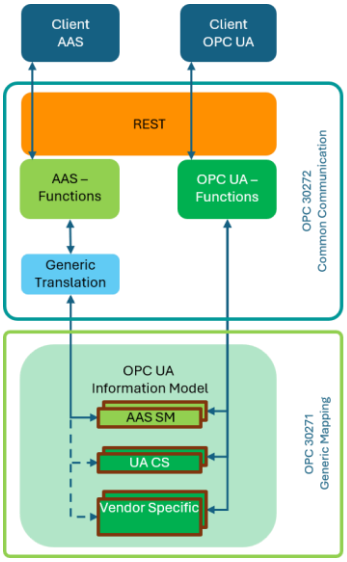


Figure 2 – Bridging Worlds

To break down the wall between the information silos, a **common information model** is defined. It allows to store all information, whether related to the asset's life cycle, or related to operation, in one unified common information model.

As the base for the common information model the OPC UA metamodel is used. This decision is justified by the enhanced expressiveness of the OPC UA metamodel compared to the AAS metamodel. In particular, OPC UA supports complex data types such as structures and arrays, which are not supported by the AAS metamodel. Furthermore, OPC UA enables the representation of rich and flexible references between nodes, which cannot be expressed in an equivalent manner within the AAS metamodel. A mapping of the OPC UA metamodel to the AAS metamodel is, in principle, possible. However, such a mapping would require numerous auxiliary constructs to compensate for the limitations of the simpler AAS metamodel.

Conversely, the representation of the AAS metamodel within the OPC UA metamodel can be achieved largely in a natural manner. One exception arises from the fact that the AAS metamodel permits multiple inheritance of elements, which is not supported by derivation mechanisms in the OPC UA metamodel; however, the specification defines a solution for that.

The **mapping** allows a bidirectional, lossless, deterministic, and type-complete mapping between the AAS information model and the OPC UA information model. The mapping is defined based on mapping AAS metamodel elements to OPC UA metamodel elements and back by preserving all attributes. Therefore, it provides a generic and automatic mapping, i.e. it allows to map automatically any AAS and any Submodels into an OPC

UA information model.

The benefit of such lossless, bidirectional mapping is the ability to maintain both AAS and OPC UA information within a single, coherent, and common information model. This eliminates data silos and the need for complex, error-prone manual mapping. Instead of having separate representations of an asset – one for its digital twin (AAS) and another for its real-time operational data (OPC UA) – they can coexist and interoperate seamlessly within a unified framework. This provides a holistic view of the asset, blending its static description with its dynamic behaviour.

Perhaps one of the most significant advantages for developers, system integrators, and ultimately, the end-users, is the ability to provide a **common information model**. This allows a client to access simultaneously:

- **Rich Semantic Information from an AAS:** Retrieving detailed asset descriptions, Submodels, properties, and their relationships, providing deep context.
- **Real-time Operational Data via OPC UA:** Accessing live sensor readings, control values, alarms, and historical data.

3 Common communication

As both the AAS and the OPC UA specification provide **OpenAPI** defined **REST APIs (HTTP/HTTPS)** for transport and **JSON** for data encoding, their usage is perfectly suited to create a unified common communication layer.

Using this layer offers the union of both REST APIs as defined by AAS and by OPC UA. It provides both AAS-specific interactions (like querying an AAS for Submodels or properties) by using the AAS-Functions as defined by IDTA and OPC UA-specific interactions (like subscribing to a sensor value or invoking a method) using the OPC UA-Functions as defined by OPC Foundation.

This **unified communication endpoint** facilitates both AAS and OPC UA communication in a fully backward-compatible manner. This is a critical aspect: it means that existing clients, whether they rely on OPC UA or on AAS, can continue to operate seamlessly. Simultaneously, new integrations can immediately tap into the advantages of this consolidated, future-proof approach without disrupting current operations.

The **translation** abstracts the AAS API provided on the common communication layer from the implementation of the information model store. It translates AAS API calls into a sequence of OPC UA calls to retrieve the requested information from or to store information in the common information model. The translation layer provides generic and automatic translation, i.e. it allows to translate automatically any AAS function call with IDTA defined JSON document content for request and response into a sequence of OPC UA function calls with OPC Foundation defined JSON document content for request and

response.

This approach dramatically simplifies client-side development, reduces network overhead, and streamlines data acquisition. Instead of managing separate connections and translation logic for each standard, developers can interact with a single, coherent data source.

For the specification of the communication layer, please refer to **Fehler! Verweisquelle konnte nicht gefunden werden.**.

4 Respecting Definitional Authority

The individual definitional sovereignty of each standard is respected:

- **AAS Metamodel:** The authoritative definitions for the AAS metamodel remain firmly established within the IDTA-01001 Metamodel specification.
- **AAS Submodel Templates:** The authoritative definitions for AAS Submodel Templates remain firmly established within the IDTA and partners.
- **AAS REST API:** The authoritative definitions for the AAS REST API remain firmly established within the Fehler! Verweisquelle konnte nicht gefunden werden. Application Programming Interfaces specification. This includes the content of the JSON document used for request and response.
- **OPC UA Metamodel:** The authoritative definitions for the OPC UA metamodel remain firmly established within the OPC UA base specification.
- **OPC UA Companion Specifications:** The authoritative definitions for OPC UA Companion Specifications remain firmly established within the OPC Foundation and partners.
- **OPC UA REST API:** The authoritative definitions for the OPC UA REST API remain firmly established within the **Fehler! Verweisquelle konnte nicht gefunden werden.** and **Fehler! Verweisquelle konnte nicht gefunden werden.** specifications. This includes the content of the JSON document used for request and response.

5 Benefits

Let's delve into what this means:

1. **Bidirectional, Lossless Mapping:** Information is mapped from AAS to OPC UA and vice-versa without any loss of information or semantic meaning.
2. **The “Roundtrip” Guarantee: Semantic Identity Preserved:** A key indicator of

the mapping is the “roundtrip” guarantee. This means that if information is mapped from an AAS into OPC UA, and back from OPC UA into an AAS, the resulting AAS model will be semantically identical to the original.

3. **Fostering a Common Information Model:** The benefit of such lossless, bidirectional mapping is the ability to maintain both AAS and OPC UA information within a single, coherent, and common information model. This eliminates data silos and the need for complex, error-prone manual mapping. Instead of having separate representations of an asset – one for its digital twin (AAS) and another for its real-time operational data (OPC UA) – they can coexist and interoperate seamlessly within a unified framework. This provides a holistic view of the asset, blending its static description with its dynamic behaviour.

4. **Flexible Client Access:**

    a) **OPC UA-Centric Clients:** A client designed to primarily utilize OPC UA functions gains immense flexibility. It can effortlessly access both traditional OPC UA information (like sensor readings, control commands, or historical data) and the rich, semantic information provided by the AAS. This means an OPC UA client can not only read a temperature value but also understand, through the AAS, which part of the machine that temperature relates to, its maintenance schedule, or its manufacturer's specifications – all through its familiar OPC UA interface.

    b) **AAS-Centric Clients:** Similarly, a client primarily leveraging AAS functions can access both AAS information and, to a certain extent, OPC UA information. In this scenario, the OPC UA data is transparently translated into the AAS format before being presented to the client. This allows an AAS client to query an asset's digital twin for a specific property, and if that property's value is dynamically delivered via OPC UA, the translation layer fetches it and presents it as if it were a native AAS property. This provides a consistent AAS view, even for data originating from OPC UA, though the direct interaction with OPC UA-specific services might be abstracted or limited to what can be meaningfully represented within the AAS structure.

This capability is more than just a technical feature; it's an enabler for:

- **True Interoperability:** It breaks down barriers between different industrial communication standards, allowing systems to “speak the same language” at a semantic level.
- **Enhanced Contextualization:** Real-time operational data from OPC UA gains important context from the AAS, transforming raw numbers into meaningful insights about an asset's lifecycle, capabilities, and status.
- **Simplified Development & Integration:** Developers can choose their preferred client paradigm (AAS or OPC UA) and still access a comprehensive view of the

asset, reducing complexity and accelerating development cycles.

- **Robust Digital Twins:** It provides a solid foundation for truly dynamic and comprehensive digital twins, where the digital representation is continuously synchronized with the physical asset's real-time performance and rich descriptive information.
- **Future-Proof Architectures:** It allows organizations to leverage existing OPC UA investments while seamlessly integrating with the evolving AAS ecosystem, ensuring their systems remain adaptable and scalable.

By ensuring that information can flow freely and accurately between these two critical industrial standards, we move significantly closer towards realizing the vision of intelligent, interconnected, and highly automated industrial environments.

Kenngrößen zur Zustandsbewertung industrieller Funkkommunikationsanwendungen

Sarah Willmann[1], Pascal Meyer[2], Marco Meier[1], André Gnad[1] und Lisa Underberg[1]

1 Einleitung

Im Zuge der weiteren Automatisierung und Flexibilisierung von Produktionsprozessen wächst der Bedarf an Unterstützungsfunktionen, wie Assistenzsysteme, oder auf organisatorischer bzw. ergonomischer Ebene, für die Mitarbeitenden im Produktionsumfeld stetig. Steigende Bedeutung und Komplexität erlangt dabei der Informationsaustausch über Fachdomänengrenzen hinweg, sodass industrielle Kommunikationssysteme bereits aus dem parallelen Zusammenspiel unterschiedlichster Technologien bestehen. Neben herkömmlichen Feldbussystemen kommen ethernet-basierte Übertragungswege, sowie lokale Funknetze wie WLAN und Mobilfunkstandards wie 5G zum Einsatz. Gerade diese technische Heterogenität birgt diverse Störungsquellen, etwa an den Übergängen zwischen verschiedenen Technologien und Medien, bei ungünstig gewählten Netzwerkkonfigurationen oder durch zunehmend aufwendige Kommunikationssoftware. Hinzu kommt, dass die Funkübertragung bei hohem Kommunikationsaufkommen durch die wachsende Beanspruchung des Funkspektrums Störungen unterliegt, was unmittelbar die Zuverlässigkeit der betroffenen Anwendungskommunikation beeinträchtigt. Dies stellt Herausforderungen in der Kommunikation in der industriellen Umgebung dar. Wie können die Herausforderungen der Koexistenz angegangen und damit die Zuverlässigkeit der Kommunikation sicher gestellt werden? Zu Beginn des Beitrages wird der Lösungsansatz für einen Teil der Messwerterfassung präsentiert und die Architektur des genutzten Versuchsaufbaus beschreiben. Im Folgenden wird das Verfahren der Messwertaufnahme erläutert, die Messergebnisse präsentiert, Ergebnisse ausgewertet und Schlussfolgerungen gezogen. Die Inhalte dieses Beitrages basieren auf Ergebnissen, die u. a. im Rahmen des Forschungsprojekts „Koexistenzmanagement für Funkanwendungen im industriellen Umfeld“ (KoFundus) Förderkennzeichen 49VF240003, welches vom Ministerium für Wirtschaft und Energie gefördert wird, erarbeitet wurden.

2 Lösungsansatz

Der Beitrag stellt die systematische Identifikation von geeigneten funklösungsbezogenen Kenngrößenwerten die zur Zustandsbewertung von heterogenen, industriellen Funkkom-

[1] Institut für Automation und Kommunikation, Industrielle Kommunikation, Werner-Heisenberg-Straße 1, 39106 Magdeburg, Deutschland, Sarah.Willmann@ifak.eu; Marco.Meier@ifak.eu; Andre.Gnad@ifak.eu; Lisa.Underberg@ifak.eu

[2] Hochschule Magdeburg-Stendal, Institut für Elektrotechnik, Breitscheidstraße 2, 39114 Magdeburg, Deutschland, jean.p.meyer@stud.h2.de

munikationsanwendungen genutzt werden können, dar. Ein Modell für die Ermittlung und Darstellung des Zustands ist die Klarzustandsfunktion (KZF), die in [WKR17] eingeführt wurde. Informationen für die KZF können aus anwendungsbezogenen, funklösungsbezogenen und physikalischen Kenngrößen gewonnen werden. Die anwendungsbezogenen Kenngrößen wie z.B. die Übertragungszeit [Me19] wurden bereits in [DF24] identifiziert, sowie eine exemplarische Untersuchung von geeigneten funklösungsbezogenen Kenngrößen durchgeführt [WMG25]. In diesem Beitrag wird dargestellt, welche zusätzlichen Informationen sich aus den Funklösungen gewinnen lassen und welchen Erkenntnisgewinn sie für die KZF bieten. Aufbauend auf [Wi26] werden die direkten Auswirkungen auf funkbezogene Kenngrößen durch eine aktive Störung unter Laborbedingungen untersucht. Dies betrifft sowohl in der management information base (MIB) standardisierte Kenngrößen [MR91], als auch herstellerspezifische Kenngrößen.

3 Theoretische Grundlagen

Gemäß den VDI/VDE-Richtlinien [An20] und [Ko22] ist Koexistenz ein "Zustand in der Funkkommunikation, in dem alle Funkkommunikationslösungen einer Produktionsanlage, die ein gemeinsames Medium nutzen, alle Kommunikationsanforderungen ihrer jeweiligen Automatisierungsanwendung erfüllen". Den Prozess der Erhaltung und Herstellung von Koexistenz durch technische sowie organisatorische Maßnahmen wird als Koexistenzmanagement bezeichnet [Wi26]. [Ko22] beschreibt verschiedene Varianten, darunter das manuelle, werkzeuggestützte Koexistenzmanagement, das dem aktuellen Stand der Technik entspricht.

Die Heterogenität der Funkkommunikationssysteme bringt vielfältige Fehlerquellen in Form von unzureichend abgestimmten Netzwerkkonfigurationen, Technologie- und Medienübergängen mit sich. Kritisch wird es insbesondere bei Interkoexistenz, wenn verschiedene Funksysteme für unterschiedliche Anwendungen dieselben Zeit-, Raum- und Frequenzbereiche belegen, oder wenn ein einzelnes System mehrere Anwendungen bedient. In beiden Fällen können sich die Anwendungen im Zeit- und Fehlerverhalten gegenseitig beeinträchtigen [Wi26]. Ob die Anwendungsanforderungen erfüllt sind und die Anwendung aus Kommunikationssicht funktionsfähig bleibt kann mithilfe der KZF beurteilt werden. Diese nutzt die in [Me17] definierten Kenngrößen für das Zeit- und Fehlerverhalten logischer Funkverbindungen und bündelt diese standardisierten Kenngrößen in einer einzigen Kennzahl, der KZF [WKR17].

Im Koexistenzmanagement greift die KZF in das Systemverhalten ein, wenn die Koexistenz der Funkanwendungen nicht mehr gegeben oder zu erwarten ist. Voraussetzung ist die Zustandserfassung von Kenngrößen des Kommunikationssystems, aus der sich geeignete Handlungen ableiten lassen. Diese Kenngrößen können in die drei folgenden Kategorien eingeordnet werden [Wi26]. Die anwendungsbezogenen Kenngrößen ermittelt das am ifak entwickelte Werkzeug FTT [WMG25], [CWP25]. Innerhalb des FTT Testsystems bestimmt das Multiface die Kenngrößen und der FTT Manager übernimmt die Verarbeitung und

Visualisierung der Daten. Die funklösungsbezogenen Kenngrößen werden werden durch die Funkgeräte bereitgestellt und über verbreitete Protokolle, bspw. SNMP (SimpleNetworkManagementProtocol), abgerufen. Für die KZF werden beide Kenngrößengruppen fusioniert.

4 Messung

4.1 Messaufbau

Für eine getrennte Analyse der Beeinflussung des Geräteverhaltens, welches durch die Wertänderung der Parameter ersichtlich wird, auf der Sende- und Empfangsseite ist der Messaufbau nach Abbildung 1 aufgebaut und auf zwei Durchläufe nach Abbildung 2 aufgeteilt. Äußere Störeinflüsse werden durch zwei getrennte Schrimboxen erheblich minimiert. In einer Schirmbox befindet sich jeweils ein Kommunikationsteilnehmer. Beide Funkkommunikationsgeräte (Tab. 1) nutzen das 2,4 GHz WLAN Band und sind durch Koaxialkabel über einen Leistungsteiler in H-Form miteinander verbunden und besitzen jeweils einen eigenen unabhängigen Netzwerkswitch. Um die Funkkommunikation reproduzierbar zu beeinträchtigen wird ein Vektor Signal Generator (VSG) verwendet. Das verwendete Störprofil, mit der "center frequency"von 2,447 GHz wurde auf dem VSG selbst erzeugt. Dieses Störsignal wird auf der zu untersuchenden Kommunikationsseite durch ein Koaxialkabel in den Leistungsteiler eingespeist. Für die Validierung der erfolgreichen Störsignaleinspeisung wird das Signal auf der entgegengesetzten Seite ausgekoppelt und von einem Spektrumanalysator (SpA) angezeigt. Über den Netzwerkswitch der Sendeseite, der den Sendeausgang 0 des Multiface mit dem Accesspoint verbindet, werden die zu

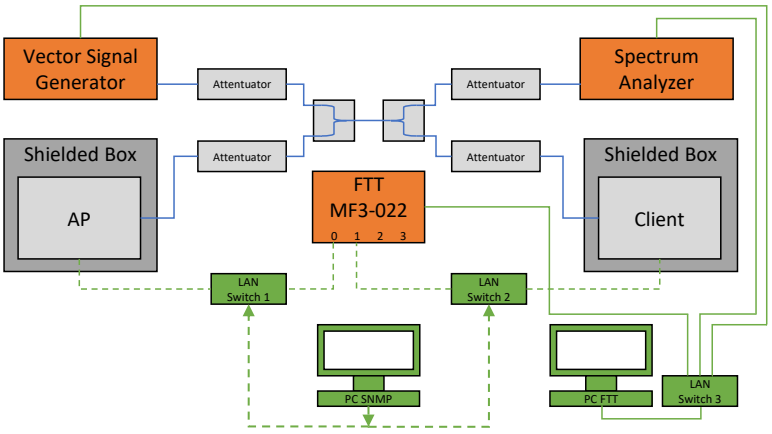
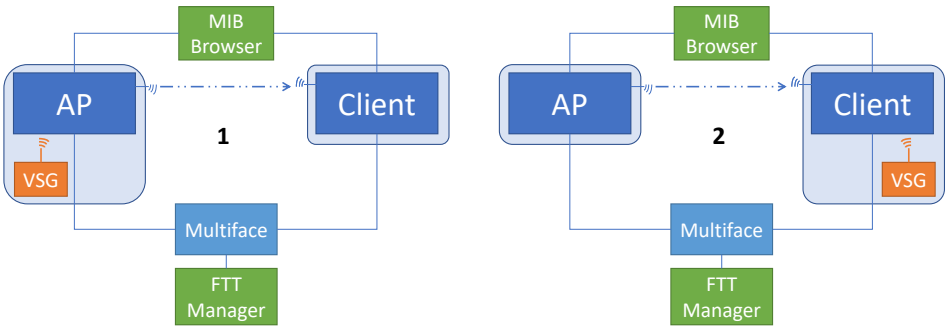


Abb. 1: Schematische Darstellung des aktuellen Netzwerkaufbaus.



(a) Durchlaufnummer 1 (b) Durchlaufnummer 2

Abb. 2: Vergleich der Störposition für die jeweilige Durchlaufnummer 1 | 2

übertragenden Datenpakete eingespeist. Die vom Client empfangenen Datenpakete werden über einen zweiten Netzwerkswitch an den Eingang 1 (empfangene Daten) des Multiface zurückgesendet. Das Multiface ist zusammen mit dem VSG und dem SpA an einen dritten Netzwerkswitch angeschlossen. Der erste Computer (PC FTT) ist mit dem dritten Netzwerkswitch verbunden und nutzt das Programm FTT Manager. Der FTT Manager bedient das Multiface, den VSG und den SpA. Änderungen der zu messenden Geräteparameter werden von einem zweiten Computer (PC SNMP) mit dem Programm MIB Browser dokumentiert. Die Störung der Sendeseite wird in der Abbildung 2a und die Störung der Empfangsseite in der Abbildung 2b dargestellt. Beide Durchlaufnummern nutzen den gleichen Messaufbau nach Abbildung 1. Zur Fehlervermeidung werden je Durchlaufnummer (1 oder 2) die Positionen des VSG und des SpA getauscht.

4.2 Versuchsplanung

Der Versuch wird in zwei Teilabschnitte, Bestimmung der Störleistung 4.2.1 und Bestimmung von aussagekräftigen Parametern für die KZF 4.2.2 unterteilt. Beide nutzen den gleichen Messaufbau nach 4.1.

4.2.1 Bestimmung der Störleistung

Innerhalb der Bestimmung der Störleistung werden die VSG Einstellungen für die Bestimmung von aussagekräftigen Parametern für die KZF ermittelt. Die Messpunktverteilung muss so dimensioniert werden, dass die unbeeinträchtigte Kommunikation, die zunehmend beeinträchtigte (geringe Störung), sowie die abbrechende Kommunikation (starke Störung) hinreichend abgebildet wird. Ebenso sind Messpunkte mit einem außerordentlichen Kommunikationsverhalten mit in die Messpunktverteilung aufzunehmen.

Tab. 1: Auswahl der verwendeten Hard- und Software

Anz.	Hersteller	Gerät / Objekt	Verwendungszweck
1x	Siemens	SIMATIC NET IWLAN AP Scalance WAM766-1EEC	Access Point
1x	Siemens	SIMATIC NET IWLAN Client Scalance WUM766	Client
1x	ifak	Multiface (Nr.22)	Testsystem
1x	ifak	Windows PC1 Software: FTT Manager	Testkonfuguration Steuerung Analyse
1x	iReasoning	Windows PC2 Software: MIB Browser	Datenerfassung Analyse
1x	Siemens	MIB Modul: SN-MSPS-SCW-MIB Revision V03.02.00	Datenerfassung Analyse
1x	John W. Eaton und weitere Autoren	Programm: GNU Octave	Auswertung und Datenvisualisierung

4.2.2 Bestimmung von aussagekräftigen Parametern für die KZF

Die ermittelte Messpunktverteilung des Abschnitts 4.2.1 bildet die Grundlage für alle Messpunkte der Bestimmung von aussagekräftigen Parametern für die KZF. Die Zeitabschnitte zwischen den Messpunkten sind aus Vergleichbarkeitsgründen gleich dimensioniert. Die Dimensionierung der Messintervalllänge muss hinreichend groß gewählt werden, um atypische Kommunikationsverhalten nach Änderungen der Störintensität von den stationären Kommunikationseigenschaften des Systems zu trennen. Empirische Voruntersuchungen zeigten, dass eine Mindestmessdauer von 10 Sekunden erforderlich ist, um eine valide Datengrundlage zu generieren. Für die Optimierung zwischen statistischer Repräsentativität und zeitlicher Effizienz wurde eine Intervalllänge von 120 Sekunden ermittelt. Zudem besteht zum Messende die Notwendigkeit eines Störendverlaufs für die Erzeugung eines natürlichen Synchronisierungspeaks, der die zeitliche Synchronisierung der FTT Manager Daten und MIB Browser Daten, durch die Übertragungszeit (TT) ermöglicht.

4.3 Durchführung

4.3.1 Bestimmung der Störleistung - Durchführung

Für die Bestimmung des Beeinflussungsbereichs der Beeinträchtigung auf die Stabilität der Gerätekommunikation wurde die Störintensität während einer laufenden Datenübertagung stetig manuell erhöht. Dabei wurden die Anzahl der nicht nutzbaren Nachrichten (nicht korrekt im jeweiligen Kommunikationszyklus empfangener Nachrichten (MM)) kontinuierlich überwacht. Als maßgebliches Kriterium für den Beginn einer relevanten Beeinflussung wurde der erstmalige signifikante Anstieg der MM herangezogen. In diesem Bereich wurden Messungen mit jeweils einer Dauer von 60 s in 0,1 dBm Schritten mit sukzessiv steigender Störamplitude mithilfe des VSG durchgeführt, bis ein stabiler und reproduzierbarer Kommunikationsabbruch erreicht wurde. Währenddessen wurden Parameter wie gesendete Nachrichten (TX), empfangene Nachrichten (RX), MM und die Anzahl der Fälle, in denen die Sequenznummer in der empfangenen Nachricht geringer als die erwartete Sequenznummer war (late cycle (LC)), gemessen. Dies wurde für beide Durchlaufnummern nach 4.1 durchgeführt.

4.3.2 Bestimmung von aussagekräftigen Parametern für die KZF - Durchführung

Die Bestimmung von aussagekräftigen Parametern für die KZF wurde nach der Tabelle 2 durchgeführt. Für die Synchronisierung der MIB- und FTT Datenquellen ist es erforderlich, dass beide Daten den beeinträchtigten Kommunikationsverlauf hinreichend abbilden ab merklicher Beeinträchtigung bis zum Kommunikationsabbruch. Die Ermittlung der Messpunkte ist im Abschnitt 4.3.1 beschrieben. Zum Beginn der Messung wurde das Multiface zur Datenübertragung, Steuerung, und Auswertung über den FTT Manager auf PC1 händisch gleichzeitig mit der MIB Parametermessung auf PC2 gestartet. Für einen vergleichbaren Messablauf wurde die Ansteuerung mit einem lua Skript im FTT Manager und dem Multiface realisiert (automatisierter Testablauf). Die Messung wurde mit dem niedrigst einstellbaren Störpegel des VSG begonnen, um die Kommunikation nahezu ungestört automatisiert vermessen zu können. Die vollständige Deaktivierung des Störeinflusses ist mit dem verwendeten lua Skript nicht möglich. Diese konnte nur händisch am VSG mithilfe einer manuellen Eingabe vorgenommen werden. Die Pegeländerung während der Messung wurde zeitgesteuert von dem FTT Manager an den VSG übertragen. Die Störintensitätsänderung erfolgte so schnell wie möglich. Die Messdauer aller Teilmessabschnitte wurde ebenfalls im Skript definiert. Die MIB-Parametermessung war die gesamte Zeit über aktiv. Für jede Messung wurde die entsprechende MIB Tabelle ermittelt und dann mit "Get Next", "Graph View"mögliche Parameter ausgewählt. Teilweise wurden zur Zeiteinsparung auch zwei MIB Tabellen gleichzeitig ausgelesen. Die Aktualisierungszeit wurde mit einer Sekunde so klein wie möglich eingestellt. Mit dem automatischen Ende der FTT Manager Messung wurde die MIB Messung händisch pausiert.

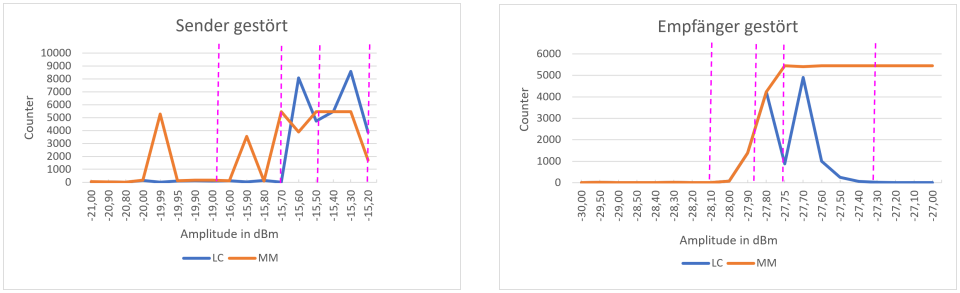
Tab. 2: Messablauf

vergangene Zeit	**Zeit pro Abschnitt**	**Maßnahme**	**Sender gestört, MIBs am Sender ausgelesen**	**Empfänger gestört, MIBs am Empfänger ausgelesen**
in sec	in sec	Status	Amplitude des Störsignals in dBm	Amplitude des Störsignals in dBm
0-120	120	Start: MIB | Multiface	-140.00	-140.00
120-240	120	Aktiviert	-18,60	-28,20
240-360	120	Aktiviert	-15,80	-28.00
360-480	120	Aktiviert	-15,70	-27,90
480-600	120	Aktiviert	-15,60	-27,80
600-720	120	Aktiviert	-15,50	-27,75
720-840	120	Aktiviert	-15,40	-27,70
840-960	120	Aktiviert	-15,30	-27,60
960-1080	120	Aktiviert	-15,20	-27,30
1080-1100	20	Verbindung zum Abbruch bringen	-13.00	-13.00
1100-1110	10	Wiederherstellen der Verbindung	-140.00	-140.00
Messende	Messende	Aus	-140.00	-140.00

4.4 Messergebnisse

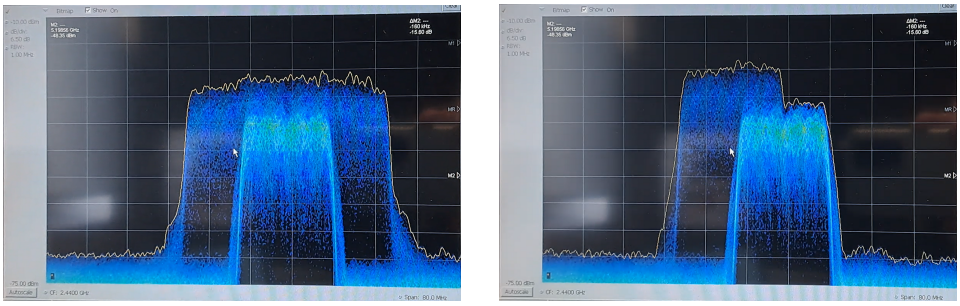
4.4.1 Bestimmung der Störleistung - Messergebnisse

Die rosa gestrichelten Linien in Abbildung 3 zeigen die ermittelten Hauptmesspunkte nach Abschnitt 4.2.1. Auffällig ist der reproduzierbare ersichtliche Einbruch von LC in Abbildung 3 für die Abbildung (a) bei -15,5 dBm (Pegel des VSG) und (b) bei -27,75 dBm. Dies könnte auf den Wechsel der Modulationsart oder die Änderung der Leistung des Funkgeräts basieren. In Abbildung 4 ist die Auffälligkeit der Senderseite bei -15,5 dBm zu erkennen. Die Teilabbildung (a) zeigt den Zustand des ungestörten Zustand bis -15,5 dBm (Pegel des VSG). Die Teilabbildung (b) zeigt den Zustand von -15,5 dBm (Pegel des VSG) bis zum Kommunikationsabbruch. Im Grenzbereich um -15,5 dBm (Pegel des VSG, senderseitig gestört) schwankten die Werte zwischen den Teilabbildungen der Abbildung 4. Bei Störung des Empfängers wurde das gleiche Verhalten gemessen. Eine genauere Analyse mit dem Programm Wireshark ergab für den ungestörten Fall eine stufenweise Erhöhung der Datenrate mit der Zeit und für den gestörten Fall eine sinkende Datenrate mit steigendem Störeinfluss.



(a) Durchlaufnummer 1 (b) Durchlaufnummer 2

Abb. 3: Vergleich der Bestimmung der Störleistung Durchlaufnummer 1 und 2



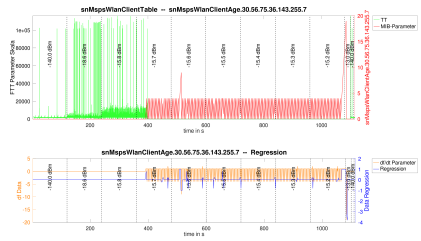
(a) Band vollständig (b) Band unterbrochen

Abb. 4: Auffälligkeit in der Kommunikation

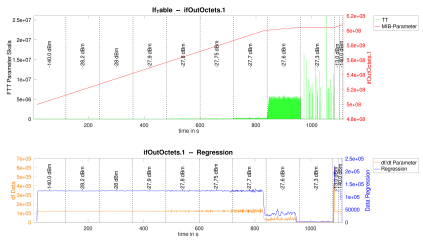
4.4.2 Bestimmung von aussagekräftigen Parametern für die KZF - Messergebnisse

Für die Synchronisierung und Visualisierung der FTT- und MIB Daten wurde ein Skript mit dem Programm Octave erstellt. Mithilfe des Skriptes wurden die vom Multiface gemessenen Daten anhand der Sequenznummerabfolge (SN) neugeordnet und dann die TT Daten anhand des TT Wertsprungs, um 1100 s, und die Geräteparameterdaten (MIB) anhand des letzten Datenpunktes auf der Zeitachse ausgerichtet. Für die Synchronisierung der Daten konnte nicht der Datenbeginn genutzt werden, weil es keine zuverlässige Startzeit des MIB Browsers oder des Mutifaces gibt. Der TT-peak kann durch einen schnellen Wechsel von einer sehr hohen Störintensität zu einer sehr geringen Störintensität erzeugt werden. Die MIB Parameterdaten sind mit zufälligen Fehlern in der Synchronisierung mit den FTT Daten behaftet. Dieser entsteht durch das händische Stoppen der MIB Messung über den PC2 zum Zeitpunkt des Messendes der FTT Messung auf dem PC1. Der daraus resultierende zufällige Fehler in der Zeitsynchronisierung beider Datenreihen beträgt maximal +/- 2 s und zeichnet sich in einer Verschiebung der MIB Parameterdaten, sowie deren Ableitung und die darauf basierende Regression, auf der x-Achse.

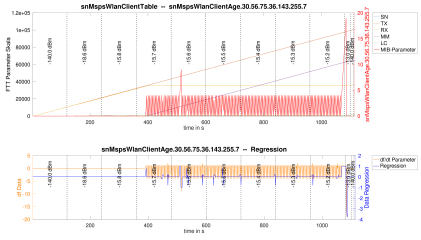
Die Abbildung 5 (a, c) und (b, d) veranschaulicht die Messergebnisse (Tab. 3). Beispielhaft wurden zwei der ermittelten Parameter mit einer Abhängigkeit vom Störeinfluss in Form von 4 Diagrammen dargestellt. Die Diagramme a und c werden folgend als Ergebnisse der Senderseite bezeichnet. Die Diagramme b und d werden folgend als Ergebnisse der Empfängerseite bezeichnet. Die Aufteilung der Daten auf zwei Diagramme dient der besseren Darstellbarkeit der einzelnen Teilgraphen. Diese wurden jeweils für die Senderseite nach der Durchlaufnummer 1 und für die Empfängerseite nach der Durchlaufnummer 2 erstellt. Die Diagramme visualisieren in der oberen Hälfte die Messergebnisse der FTT Daten (nur TT in grün) und die MIB Parameter Daten in rot. Die linke y-Skala wird verwendet für alle darzustellenden anwendungsbezogenen Kenngrößen (SN, TX, RX, MM, LC) die mit dem FTT gemessen wurden. Die Messgrößen SN und LC werden nur sehr klein dargestellt. Die rechte y-Skala in rot wird ausschließlich für den jeweiligen untersuchten MIB Parameter verwendet. In der unteren Diagrammhälfte ist die Ableitung des MIB Parameters und die darauf basierende gleitende Regression [Sm97] dargestellt. Diese Regression wurde für eine Minimierung des Rauschverhaltens und bessere Analyse des Ableitungsgraphs durchgeführt. Die Regressionsberechnung erfolgte lokal über ein gleitendes Zeitfenster von jeweils fünf aufeinanderfolgenden Messwerten (Polynom 1. Grades, n = 5). Dieses Zeitfenster wurde schrittweise über den gesamten Datensatz verschoben. Die farbliche Unterscheidung der Graphen und Achsen visualisiert die Zugehörigkeiten der Graphen zu der jeweiligen y-Achse. Auf eine Änderung der Störamplitude durch den VSG wird durch



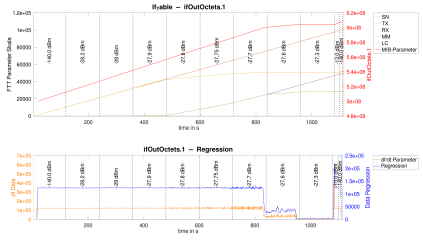
(a) Sender:snMspsWlanClientTablesnMspsWlanClientAge



(b) Empfänger:IfTableifOutOctets



(c) Sender:snMspsWlanClientTablesnMspsWlanClientAge



(d) Empfänger:IfTableifOutOctets

Abb. 5: Vergleich der Messergebnisse der Sende und Empfangsseite
(Y-Skala MIB Parameter (rot) Wert * 10)

eine senkrechte, gestrichelte, schwarze Linie am Änderungspunkt dargestellt. Die daraus entstandenen Abschnitte sind in Tabelle 2 definiert worden.

Als nennenswerte funklösungsbezogene Kenngröße der Senderseite wurde der Parameter snMspsWlanClientAge der SnMspsWlanClientTable unter den Bedingungen der Durchlaufnummer 1 ausgewählt. Der Parameter gibt die Zeit an, die seit dem Empfang des letzten Frames von der Client Station vergangen ist. Repräsentativ für die Störintensität der anwendungsbezogenen Kenngröße der Sendeseite steht die steigende Übertragungszeit (TT: Abbildung 5a). Die TT steigt im Mittel stufenweise im direkten Zusammenhang mit dem Störeinfluss. Kurz bevor die TT im Abschnitt von -15,7 dBm keine Ergebnisse mehr liefert, weil die Datenübertragung abbricht, steigt der Wert des MIB Parameters snMspsWlanClientAge zeitgleich mit RX, ersichtlich in (Abbildung 5c), sprunghaft an. Die zeitliche Ableitung des MIB Parameters in Abbildung 5c zeigt die Entwicklung der Zeit, die seit dem Empfang des letzten Frames von der Client Station vergangen ist. Durch den Graph der Regression (blau) werden Zeitschwankungen ersichtlich. Diese zeigen ein internes Geräteverhalten auf. Dabei handelt es sich vermutlich um Sende- oder Optimierungsversuche. Das bedeutet der Parameter snMspsWlanClientAge steht nicht im direkten Zusammenhang mit der Störpegeländerung.

Als nennenswerte funklösungsbezogene Kenngröße der Empfängerseite wurde der Parameter ifOutOctets der IfTable unter den Bedingungen der Durchlaufnummer 2 ausgewählt. Nach der Parameterbeschreibung des MIB Browser [MR91] gibt dieser Parameter die Gesamtzahl der übertragenen Oktette, einschließlich "framing characters"wieder. Der Parametergraph der Abbildung 5b zeigt eine evidente Abhängigkeit von der Störintensität. Repräsentativ für die Störintensität stehen die anwendungsbezogen Kenngrößen der steigenden Übertragungszeit (TT: Abbildung 5b) und der sinkenden Anzahl der empfangenen Pakete (RX: Abbildung 5d). Während der Störbeeinflussungsbereich des MIB Parameters ifOutOctets aus den Originaldaten (rot) erst ab der -27,6 dBm Grenze ersichtlich wird, beginnt die tatsächliche Beeinflussung wesentlich früher. Die Ableitungsdaten des Parameters in Abbildung 5d zeigen merkliche Abweichungen um den normalen Anstieg ab dem Abschnitt -27,9 dBm (orange). Das bedeutet mithilfe einer Datenaufbereitung, in Form einer Ableitung, werden aussagekräftige Zustände über die Kommunikationsstörung ermöglicht. Deutliche Änderungen der TT und RX Daten an den Grenzen der Störpegeländerung deuten ebenso auf eine direkte Abhängigkeit von der Störintensität hin, wie der Parameter ifOutOctets. Es ist zu erwähnen, dass der Parameter ifOutOctets auch ohne Störeinfluss nahezu linear hochzählt. Dies ist in Abbildung 5d anhand des konstanten Verhaltens der Ableitung in dem Abschnitt -130 dBm und dem annähernd konstantem Verhalten (leichter Störeinfluss) vom Messbeginn bis zum Ende des Abschnitts -27,9 dBm ersichtlich. Der in Abbildung 5d aufgetretene Sprung zum Messbeginn in dem Ableitungs- und Regressionsgraph wird durch den erhöhten Startwert des MIB Parameters zum Messbeginn verursacht.

Zusätzlich konnte ein Unterschied zwischen einer stetigen und diskreten Störpegeländerung festgestellt werden. Eine Messung gilt als stetig, wenn der Störpegel manuell am VSG umgestellt wird. Die verwendete Umstellungsmethode unter der Verwendung des

in Abschnitt 4.3.2 eingeführten Skript erzeugt eine diskrete Störpegeländerung, da die Gerätekonfiguration bei jeder Pegeländerung neu an den VSG übertragen wird. Dieses Neuübertragungsverhalten führt zu folgendem Verhalten. Wird die Messung unter diskreten Bedingungen genau mit dem identischen Störpegel wie bei einer stetigen Messung an einem stark kommunikationsbeeinträchtigtem Punkt kurz vor Kommunikationsabbruch, an dem die Kommunikation unter stetiger Änderung noch reproduzierbar funktionierte, durchgeführt, kann keine Kommunikation hergestellt werden. Die Gerätekommunikation bleibt unter stetiger Störung länger erhalten als bei diskreter Störung. Um einen Kommunikationsabbruch durch eine sendeseitige Störung zu erzielen, muss der Ausgangspegel des VSG etwa 12 dBm stärker als auf der Empfängerseite eingestellt werden. Das bedeutet die Empfängerseite reagiert nennenswert empfindlicher auf Störungen als die Senderseite.

Der Tabelle 3 ist eine Einordnung in zwei Gruppen zu entnehmen. Grün hinterlegte Parameter stehen für ein gesuchtes Verhalten. Ein Parameter gilt als gesucht, wenn sich der Anstieg des Geräteparameters in Abhängigkeit des Störpegels auf die Gerätekommunikation maßgeblich ändert. Gelb hinterlegte Parameter zeigen eine Änderung mit geringer Abhängigkeit zum Störeinfluss und meist keine Abhängigkeit vom Störpegel auf. Rot hinterlegte Parameter sind dauerhaft konstant oder 0.

Tab. 3: Messergebnisübersicht mit Farbcodierung

MIB-Table	Sender gestört, MIBs am Sender ausgelesen	Empfänger gestört, MIBs am Empfänger ausgelesen
IfTable	ifInOctets.1 ifInUcastPkts.1 ifInNUcastPkts.1 ifOutOctets.1 ifOutUcastPKts.1 ifOutNUcastPkts.1 ifIndex.1 ifInDiscards.1 ifInErrors.1 ifInUnknownProtos.1 ifMtu.1 ifOutDiscards.1 ifOutErrors ifOutQLen.1 ifSpeed.1	ifInOctets.1 ifInUcastPkts.1 ifInNUcastPkts.1 ifOutOctets.1 ifOutUcastPKts.1 ifOutNUcastPkts.1 ifIndex.1 ifInDiscards.1 ifInErrors.1 ifInUnknownProtos.1 ifMtu.1 ifOutDiscards.1 ifOutErrors.1 ifOutQLen.1 ifSpeed.1 ifLastChange.1
snMspsWlanStatisticErrorsTable	snMspsWlanStatisticErrorsTxCount.10 snMspsWlanStatisticErrorsTxRetry.10 snMspsWlanStatisticErrorsTxDroppedFrames.10 snMspsWlanStatisticErrorsRxDuplicatedFrames.10 snMspsWlanStatisticErrorsRxCountTotal.10 snMspsWlanStatisticErrorsRxCount.10 snMspsWlanStatisticErrorsRxDecryption.10 snMspsWlanStatisticErrorsRxFcs.10	snMspsWlanStatisticErrorsTxCount.6 snMspsWlanStatisticErrorsTxRetry.6 snMspsWlanStatisticErrorsTxDroppedFrames.6 snMspsWlanStatisticErrorsRxDuplicatedFrames.6 snMspsWlanStatisticErrorsRxCountTotal.6 snMspsWlanStatisticErrorsRxCount.6 snMspsWlanStatisticErrorsRxDecryption.6 snMspsWlanStatisticErrorsRxFcs.6
snMspsWlanStatisticTrafficTable	snMspsWlanStatisticTrafficTxAverageRate.10 snMspsWlanStatisticTrafficTxUniCast.10 snMspsWlanStatisticTrafficRxDataFrames.10 snMspsWlanStatisticTrafficRxUnicast.10 snMspsWlanStatisticTrafficRxMBCast.10 snMspsWlanStatisticTrafficRxAverageRate.10 snMspsWlanStatisticTrafficAssocResponseRx.10 snMspsWlanStatisticTrafficAuthResponseTx.10 snMspsWlanStatisticTrafficDisassocResponseRx.10 snMspsWlanStatisticTrafficDisassocRequestTx.10 snMspsWlanStatisticTrafficAuthRequestRx.10 snMspsWlanStatisticTrafficDeauthRequestTx.10 snMspsWlanStatisticTrafficDeauthRequestRx.10 snMspsWlanStatisticTrafficTxDataFrames.10 snMspsWlanStatisticTrafficDisassocResponseTx.10 snMspsWlanStatisticTrafficAssocRequestRx.10 snMspsWlanStatisticTrafficRxMgmtFrames.10 snMspsWlanStatisticTrafficDeauthResponseRx.10 snMspsWlanStatisticTrafficAssocRequestTx.10 snMspsWlanStatisticTrafficAuthResponseRx.10 snMspsWlanStatisticTrafficDeauthResponseTx.10 snMspsWlanStatisticTrafficAssocResponseTx.10 snMspsWlanStatisticTrafficTxMgmtFrames.10 snMspsWlanStatisticTrafficAuthRequestTx.10 snMspsWlanStatisticTrafficDisassocRequestRx.10	snMspsWlanStatisticTrafficTxAverageRate.6 snMspsWlanStatisticTrafficTxUniCast.6 snMspsWlanStatisticTrafficRxDataFrames.6 snMspsWlanStatisticTrafficRxUnicast.6 snMspsWlanStatisticTrafficRxMBCast.6 snMspsWlanStatisticTrafficRxAverageRate.6 snMspsWlanStatisticTrafficAssocResponseRx.6 snMspsWlanStatisticTrafficAuthResponseTx.6 snMspsWlanStatisticTrafficDisassocResponseRx.6 snMspsWlanStatisticTrafficDisassocRequestTx.6 snMspsWlanStatisticTrafficAuthRequestRx.6 snMspsWlanStatisticTrafficDeauthRequestTx.6 snMspsWlanStatisticTrafficDeauthRequestRx.6 snMspsWlanStatisticTrafficTxDataFrames.6 snMspsWlanStatisticTrafficDisassocResponseTx.6 snMspsWlanStatisticTrafficAssocRequestRx.6 snMspsWlanStatisticTrafficRxMgmtFrames.6 snMspsWlanStatisticTrafficDeauthResponseRx.6 snMspsWlanStatisticTrafficAssocRequestTx.6 snMspsWlanStatisticTrafficAuthResponseRx.6 snMspsWlanStatisticTrafficDeauthResponseTx.6 snMspsWlanStatisticTrafficAssocResponseTx.6 snMspsWlanStatisticTrafficTxMgmtFrames.6 snMspsWlanStatisticTrafficAuthRequestTx.6 snMspsWlanStatisticTrafficDisassocRequestRx.6
SnMspsWlanClientTable	snMspsWlanClientAge.30.56.75.36.143.255.7 snMspsWlanClientChannelNumber.30.56.75.36.143.255.7 snMspsWlanClientDataRate.30.56.75.36.143.255.7 snMspsWlanClientSignalStrength.30.56.75.36.143.255.7 snMspsWlanClientSignalPercentage.30.56.75.36.143.25.7 snMspsWlanClientAID.30.56.75.36.143.255.7	———keine Parameter————

5 Zusammenfassung und Ausblick

Mit der Erkenntnis über aussagekräftigere Kommunikationsqualitätsinformation durch funklösungsbezogene Kenngrößen der Empfängerseite mit einer direkten Abhängigkeit vom Störeinfluss ist zu schlusszufolgern, dass sich die Parameter der Empfängerseite mehr für mögliche Eingriffsmaßnahmen zur Verbesserung des Kommunikationsverhalten eignen. Offen ist die Betrachtung der funkumgebungsbezogenen Kenngrößen um alle drei Kategorien in die KFZ einfließen lassen zu können. Aktuell können Entscheidungen auf Basis der KZF getroffen werden, die Einbeziehung der funkumgebungsbezogenen Kenngrößen würde das Gesamtbild komplettieren.

Literaturverzeichnis

[An20] Funkgestützte Kommunikation in der Automatisierungstechnik – Anforderungen und Grundlagen, Richtlinie VDI/VDE 2185 Blatt 1, Berlin: VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik, 2020.

[CWP25] Cainelli, G. P.; Willmann, S.; Pereira, C. E.: Performance evaluation of wireless communications systems: test system, performance parameters and influencing quantities. In: 2025 IEEE Latin Conference on IoT (LCIoT). S. 339–342, 2025.

[DF24] DFAM - Deutsche Forschungsgesellschaft für Automatisierung und Mikroelektronik e.V.: Reglerbasiertes Koexistenzmanagement verschiedenartiger Funkkommunikationssysteme (REBAKO). In Forschungsbericht Nr. 43/2024, 2024.

[Ko22] Funkgestützte Kommunikation in der Automatisierungstechnik – Koexistenzmanagement von Funksystemen, Richtlinie VDI/VDE 2185 Blatt 2, Berlin: VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik, 2022.

[Me19] Funkgestützte Kommunikation in der Automatisierungstechnik – Messtechnische Performancebewertung von Funklösungen, Richtlinie VDI/VDE 2185 Blatt 4, Berlin: VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik, 2019.

[MR91] McCloghrie, K.; Rose, M.: Management Information Base for Network Management of TCP/IP-based internets: MIB-II, RFC 1213, 1991, https://www.rfc-editor.org/info/rfc1213/.

[Sm97] Smith, S. W.: Moving Average Filters. In: The Scientist and Engineer's Guide to Digital Signal Processing. California Technical Publishing, Kap. 15, 1997.

[Wi26] Willmann, S. et al.: Klarzustandsfunktion zur Überwachung und Beeinflussung industrieller Funkkommunikationsanwendungen, 2026, http://dx.doi.org/10.25673/123342.

[WKR17] Willmann, S.; Krätzig, M.; Rauchhaupt, L.: Methodology for holistic assessment of dependability in wireless automation. In: 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). S. 1–4, 2017.

[WMG25] Willmann, S.; Meier, M.; Gnad, A.: Einsatzbeispiele für ein Werkzeug zur Untersuchung der Zuverlässigkeit industrieller Kommunikationssysteme. KommA 2025 – 16. Jahreskolloquium Kommunikation in der Automation, Magdeburg 11.-12.11.2025, 2025.

Layer 2 Konnektivität in 5G Campusnetzwerken: Vergleich von VXLAN und Ethernet-PDU-Sessions für Industrial Ethernet

Denis Gustin[1], Timo Siekmann[2]

Abstract:
Die Einführung von 5G in der industriellen Automation eröffnet neue Möglichkeiten für drahtlose Kommunikation mit hohen Datenraten, geringen Latenzen und Echtzeitfähigkeit. Gleichzeitig stellen etablierte Industrial-Ethernet-Protokolle wie PROFINET/PROFIsafe bzw. EtherNet/IP mit CIP Safety oder auch die Integration von Time-Sensitive Networking (TSN) mit 5G klare Anforderungen an eine Kommunikation auf Layer 2 (Siemens AG, 2022). Da heutige 5G Systeme primär Layer 3-Konnektivität bereitstellen, werden Layer 2-Anbindungen in der Praxis entweder per Tunneling, wie Virtual eXtensible LAN (VXLAN), oder – mit 3GPP Release 16 – per Ethernet Packet Data Unit (PDU) Sessions realisiert. Die damit verbundenen Auswirkungen auf Durchsatz, Latenz und Jitter für industrielle Anwendungen sind bislang nur unzureichend quantifiziert. Vorarbeiten demonstrieren zwar die prinzipielle Funktionsfähigkeit von Ethernet PDU-Sessions in 5G Netzen und deren Integration mit industriellen LAN-Technologien, liefern jedoch noch keinen systematischen Performancevergleich mit Tunneling-basierten Ansätzen.

In dieser Arbeit wird die Performance dieser beiden Ansätze systematisch untersucht und miteinander verglichen. Hierzu wird ein Messaufbau mit VXLAN Tunneling über 5G sowie Ethernet PDU Sessions über 5G realisiert. Die Messungen erfolgen für die Ethernet-Framegröße 256 Byte, wie sie in typischen OT-Szenarien auftritt und betrachten explizit die Latenz der Übertragungsstrecke bei unterschiedlichen Zykluszeiten.

Die Ergebnisse zeigen, dass VXLAN als etablierte und breit verfügbare Technologie zwar eine einfache Realisierung von Layer 2 Kommunikation über 5G-Layer 3-Netze ermöglicht, der notwendige Kapselungs-Overhead jedoch die effektive MTU reduziert und somit nutzbare Datenrate als auch Latenz und Jitter nachteilig beeinflusst. Ethernet PDU Sessions vermeiden diesen Overhead, erlauben die Nutzung der vollen MTU und führen im Versuchsaufbau zu höherem nutzbarem Durchsatz sowie geringeren Latenzen. Damit eignen sie sich insbesondere für Anwendungen mit hohen Anforderungen an Echtzeitfähigkeit und Verfügbarkeit, etwa bei Safety-Protokollen oder Motion-Control-Aufgaben.

Keywords: 5G NR, Ethernet PDU, Industrial Ethernet, Layer 2, VXLAN

[1] Fraunhofer IOSB-INA, Campusallee 1, 32657, Lemgo, Denis.Gustin@iosb-ina.fraunhofer.de,
[2] Fraunhofer IOSB-INA, Campusallee 1, 32657, Lemgo, Timo.Siekmann@iosb-ina.fraunhofer.de,

1 Einleitung

Die fortschreitende Digitalisierung in der industriellen Automation (Industrie 4.0) erfordert Kommunikationssysteme, die hohe Datenraten, geringe Latenzen und ein hohes Maß an Determinismus gewährleisten. 5G bietet hier das Potenzial, die starre Verkabelung durch flexible drahtlose Verbindungen zu ergänzen oder zu ersetzen. Gleichzeitig basieren etablierte Industrial-Ethernet-Protokolle wie PROFINET/PROFIsafe oder EtherNet/IP mit CIP Safety sowie Konzepte des Time-Sensitive Networking (TSN) zwingend auf einer durchgängigen Layer 2 Konnektivität (Ethernet) zwischen Steuerungen, Peripheriegeräten und Sicherheitskomponenten. Für die nahtlose Integration von 5G in bestehende Automatisierungsnetzwerke ist diese Transparenz auf Layer 2 daher unerlässlich.

Problemstellung

Heutige 5G Systeme stellen in ihrer Standardkonfiguration primär Layer 3-Konnektivität (IP-basierte PDU Sessions) bereit. Um dennoch Layer 2 Kommunikation über 5G Netze abzuwickeln, werden in der Praxis häufig Tunneling-Verfahren wie Virtual eXtensible LAN (VXLAN) eingesetzt. Mit 3GPP Release 16 wurde der Standard um sogenannte PDU Sessions Type „Ethernet“ erweitert, die Layer 2 nativ über das 5G Netz transportieren. Während die prinzipielle Funktionsfähigkeit beider Ansätze in Vorarbeiten gezeigt werden konnte, fehlt bislang eine systematische und quantitative Untersuchung der Auswirkungen auf netzwerkspezifische Leistungskenngrößen wie Durchsatz, Latenz und Jitter. Insbesondere der Overhead durch Kapselungsverfahren wie VXLAN und dessen Einfluss auf die maximale Nutzdatenrate (MTU) bei unterschiedlichen Framegrößen sind für OT-Anwendungen kritisch (Elmadani & Sati, 2024).

Zielsetzung und Aufbau der Arbeit

Ziel dieser Arbeit ist der systematische Performancevergleich von VXLAN Tunneling und Ethernet PDU Sessions zur Herstellung von Layer 2 Konnektivität in privaten 5G Netzen. Hierzu wird ein Messaufbau in einem Labor-Testbed realisiert, der VXLAN Tunneling über 5G (Layer 3) sowie Ethernet PDU Sessions über 5G (Layer 2) umfasst. Die Messungen basieren auf One Way Latenzmessungen für verschiedene Frameraten, in der OT typische Ethernet-Framegröße 256 Byte (5G ACIA, 2019).

Das restliche Paper ist wie folgt strukturiert: Kapitel 2 erläutert die theoretischen Grundlagen zu 5G PDU Sessions und VXLAN. Kapitel 3 beschreibt den Messaufbau, die Hardwarekomponenten und die Methodik. In Kapitel 4 werden die Messergebnisse präsentiert und visuell aufbereitet. Kapitel 5 gibt ein Fazit zu der Arbeit und einen Ausblick.

2 Theoretische Grundlagen

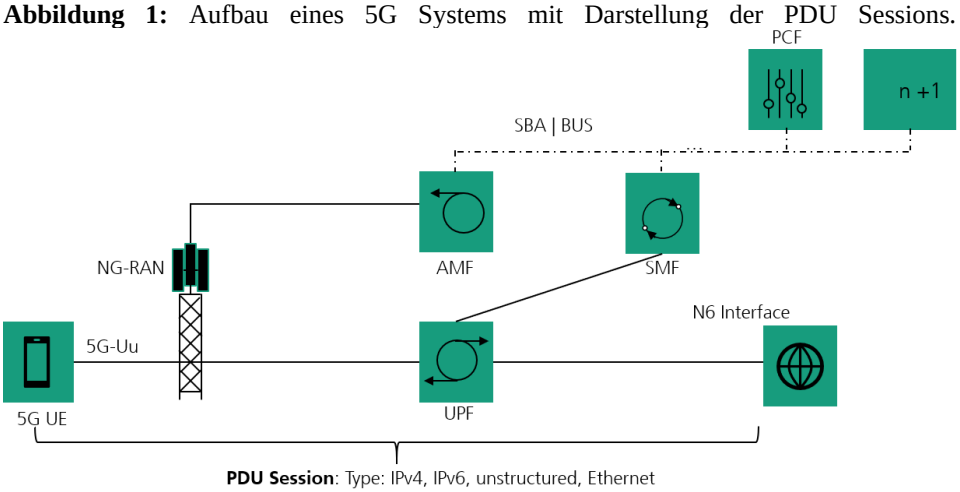
5G PDU Sessions

PDU Sessions beschreiben eine logische Verbindung zwischen UE (User Equipment) und dem Data Network. Die PDU Session agiert als Daten Pipeline und inkludiert DRB (Data Radio Bearers) die wiederum QoS (Quality of Service) Flows beinhalten. Der QoS Flow ist mit seinem QoS Flow Identifier die kleinste Einheit um Frames oder Pakete im 5G System zu markieren und gegeneinander zu priorisieren.

Im 3GPP Release 16 wurde eine Erweiterung der PDU Sessions um den Type Ethernet eingeführt. Damit können Layer 2 Frames zwischen dem 5G UE und der UPF ohne zusätzliche Kapselung, wie Tunneling, über das 5G Netzwerk übertragen werden. Insbesondere für Layer 2 Netzwerke, welche in der Robotik und Industrie sehr häufig verwendet werden, bedeutet das mehr Transparenz und ermöglicht neue Netzwerkarchitekturen. Durch die Layer 2 Transparenz wird es möglich zwei Layer 2 Netzwerke mittels 5G zu verbinden und ebnet somit auf dem Physical Layer den Weg zur TSN (Time Sensitive Networking) Bridge (Bin Muslim, Tönjes, & Bauschert, 2024). Die Layer 2 Transparenz, ist neben der präzisen Zeitsynchronisation ein entscheidender Bestandteil in der Kombination von 5G und TSN Netzwerken.

Hervorzuheben ist, dass die Nutzung von PDU Session Type eine ganzheitliche Unterstützung vom 5G System benötigt. So muss das 5G UE, das 5G RAN und der 5G Core die PDU Session Type Ethernet unterstützen. In der Vorbereitung für diese Veröffentlichung wurde die RAN Unterstützung von den Herstellern Huawei, Nokia und Amarisoft in Versuchen bestätigt. Insbesondere die UPF und damit einhergehend der 5G Core muss die PDU Session Type Ethernet unterstützen, damit geht einher das bestehende 5G Netzwerke mittels eines Core Update PDU Sessions Type Ethernet unterstützen könnten.

Abbildung 1: Aufbau eines 5G Systems mit Darstellung der PDU Sessions.



VXLAN Tunneling

Virtual extensible Local Areas Network ist ein Protokoll der IETF mit dem Standard (RFC 7348), welches ursprünglich als Lösung für die gestiegenen Anforderungen im Bereich Datacenter entwickelt wurde. Als ein Bedarf wird die limitierte Anzahl an MAC Adressen im VLAN (Virtual Local Area Network) und als weiterer Bedarf der Transport von Layer 2 Protokollen über ein IP Netzwerk genannt (Mahalingam, et al., 2014).

Die Übertragung von Layer 2 Ethernet Frames erfolgt mittels Kapselung in Layer 3 IP Pakete. Dadurch entsteht ein Tunnel mit einem VTEP (Virtual Tunneling End Point) am Beginn und Ende, dazwischen verhält sich das Netzwerk wie ein Ethernet Netzwerk. Damit reiht sich VXLAN den Tunnelprotokollen wie GRE oder L2TP ein.

Durch die, für den Tunnel, notwendige Kapselung entsteht zusätzlicher Overhead. Jeder Ethernet-Frame wird mit einem VXLAN Header, einem UDP Header, einem IP Header und einem zusätzlichen Ethernet-Header gekapselt, wodurch sich die gesamte Frame-Größe erhöht. Dieser Overhead führt zu einer reduzierten MTU (Maximum Transmission Unit) und begrenzt damit die effektiv über die Kommunikationsverbindung übertragbare Nutzlast. Zur Veranschaulichung der Auswirkungen der VXLAN Kapselung betrachten wir die folgende Overhead-Berechnung:

8 Byte (VXLAN) + 8 Byte (UDP) + 20 Byte (IP) + 14 Byte (Ethernet) = 50 Byte

Damit beträgt die effektive MTU für einen über VXLAN getunnelten Ethernet-Frame maximal 1450 Byte. Diese Verringerung der MTU bedeutet, dass die maximale Nutzlast, die in einem einzelnen Frame übertragen werden kann, reduziert wird. Dies kann zu einer stärkeren Fragmentierung und höheren Latenzen führen, was sich in industriellen

Anwendungsszenarien negativ auf die Kommunikationsleistung auswirken kann (Mahalingam, et al., 2014).

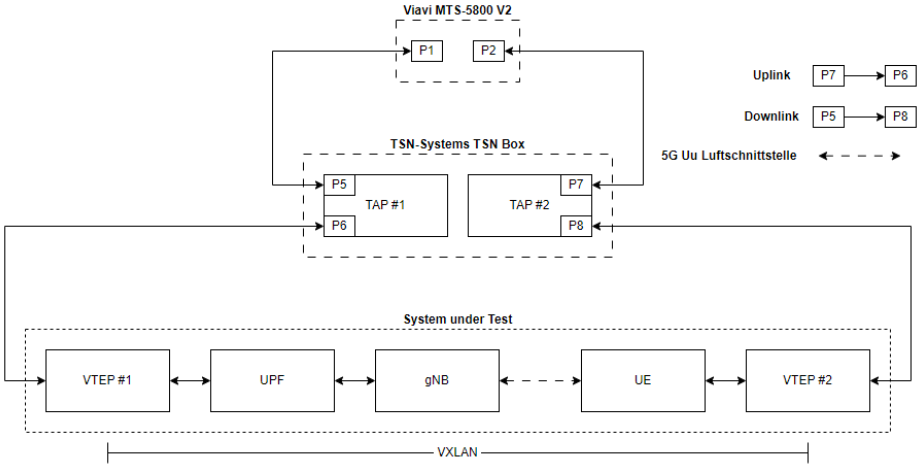
Da das VXLAN Protokoll seit 2012 mit der Kernel version 3.7 nativ im Mainline Linux integriert (KernelNewbies.org, 2017) ist und Siemens es auch für die Nutzung seiner 5G Endgeräte als auch seiner 5G Infrastrukturen empfiehlt, wird das Protokoll VXLAN als Vergleich zum nativen Layer 2 über 5G mittels PDU Session Type Ethernet herangezogen (Siemens AG, 2022).

3 Messaufbau

Der Messaufbau umfasst ein Amarisoft Callbox 5G Campusnetzwerk. Als 5G Router wird ein Teltonika RUTC50 Router verwendet, da dieser die Möglichkeit bietet Layer 2 Traffic mittels VXLAN zu tunneln. Der 5G Router wird mit dem Traffic Generator (Viavi MTS 5800) verbunden und die Verbindung zum 5G System erfolgt über die Luftschnittstelle. Der VXLAN Tunnel wird zwischen dem Teltonika 5G Router über das 5G Netzwerk und bis zum Amarisoft systeminternen VXLAN Endpunkt aufgebaut (VXLAN ist nativ in Linux integriert). Der Teltonika Router und die Amarisoft Netzwerkkarte (Linux System) bilden somit die VTEPs (VXLAN Tunnel Endpoints). Abbildung 2 beschreibt den Messaufbau.

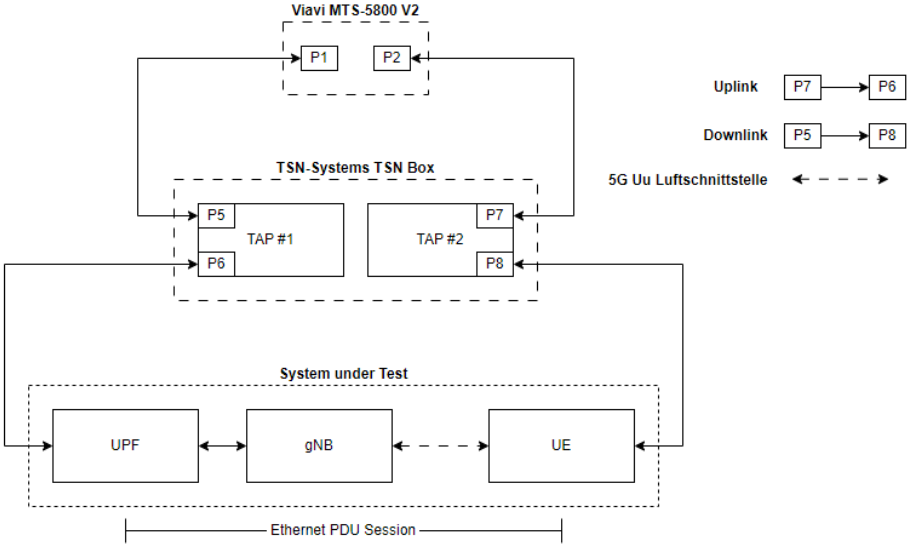
Zwischen den VTEPS und dem Traffic Generator befindet sich eine TSN Systems Testbox, welche zwei TAPs beinhaltet. Mit Hilfe der TSN Testbox werden Frames aus dem Viavi Traffic Generator mitgeschnitten, bevor diese in das System under Test (SUT) übertragen werden. Sobald Sie das SUT wieder verlassen, werden diese Frames auf dem zweiten TAP ebenfalls mitgeschnitten. Die Differenz der Timestamps bildet die Latenz. Sie wird als OWD (One Way Delay) jeweils in Uplink- und Downlink Richtung gemessen.

Abbildung 2: Messaufbau für das VXLAN Szenario.



Der Messaufbau für die Layer 2 Transparente Lösung umfasst neben dem Amarisoft 5G System und dem Traffic Generator, der TSN Testbox ein 5G Modem welches das Feature PDU Session Type ETHERNET unterstützt. Zum Einsatz kommt hier ein Quectel Modem (RM500U-CN), da dieses Modem die Möglichkeit bietet Ethernet PDU Sessions zu initiieren. Das Modem ist in einer Waveshare Hardware eingebettet, welche die Verbindung von M.2 auf RJ 45 ermöglicht. Dadurch kann der Traffic Generator direkt am Modem angeschlossen werden. Der weitere VXLAN Router entfällt in dem Messaufbau, da die Frames nach dem 5G System (N6 Interface) über das Tap in der TSN Testbox an den zweiten Port des Traffic Generators geleitet werden. Der Aufbau wird in Abbildung 3 beschrieben.

Abbildung 3: Messaufbau für das Ethernet PDU Szenario.



4 Messergebnisse und Diskussion

Der direkte Vergleich von Ethernet PDU Sessions mit VXLAN Tunneling bei der Framegröße 256 Byte mit 1000 Frames/Sekunde (1 ms Zykluszeit übertragen auf Industrial Ethernet) zeigt in den Abbildungen 4 bis 7, wie VXLAN Tunneling einen deutlichen Einfluss auf die Verteilung der Frame-Latenzen hat. Die Verteilung gleicht mit Ethernet PDU Sessions der typischen rechtsschiefen Verteilung von 5G Latenzen, wie bspw. durch (Cainelli, et al., 2024) gezeigt. VXLAN Tunneling erzeugt eine deutlich stärker gestreute Latenzverteilung. VXLAN hat einen 49,9 % höheren Mittelwert im Uplink und einen 41,6 % höheren Mittelwert im Downlink. Das P99 Perzentil liegt bei VXLAN im Uplink ca. 41,1 % höher während es im Downlink lediglich 5,3 % höher liegt.

Mittelwert und P99 Perzentil sind somit sowohl im Uplink als auch im Downlink den Ethernet PDU Sessions deutlich unterlegen. Die Unterschiede sind in Tabelle 1 dargestellt.

Tabelle 1: Mittelwert und P99 Perzentil der Latenzen im Vergleich für VXLAN Tunneling und Ethernet PDU Sessions. Gemessen für eine Framegröße von 256 Bytes bei 1000 Frames/Sekunde (1 ms Zykluszeit).

Metrik	**UL VXLAN**	**UL Ethernet PDU**	**DL VXLAN**	**DL Ethernet PDU**
Mittelwert	24,04 ms	16,04 ms	9,77 ms	6,90 ms
P99 Perzentil	44,34 ms	31,42 ms	14,42 ms	13,69 ms

Abbildung 4: Histogramme der Messungen für den Uplink mit 256 Bytes Framegröße und 1000 Frames/s.

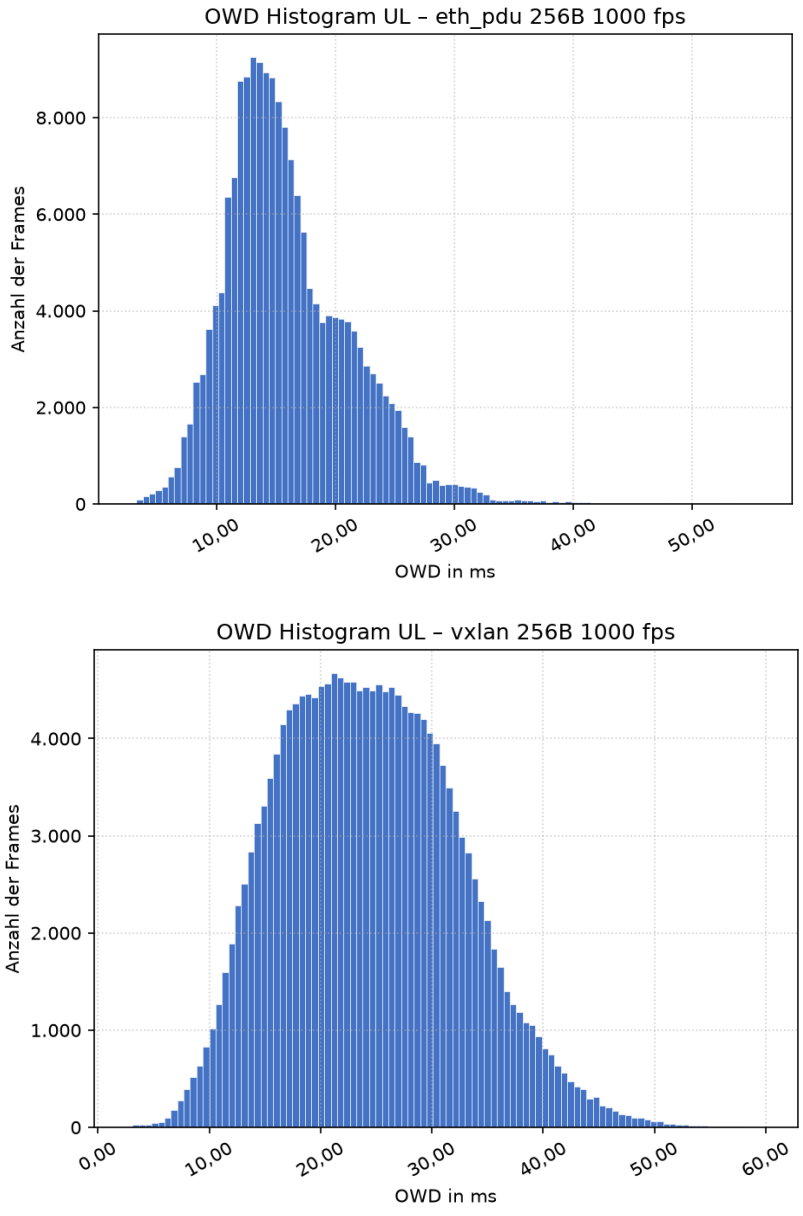
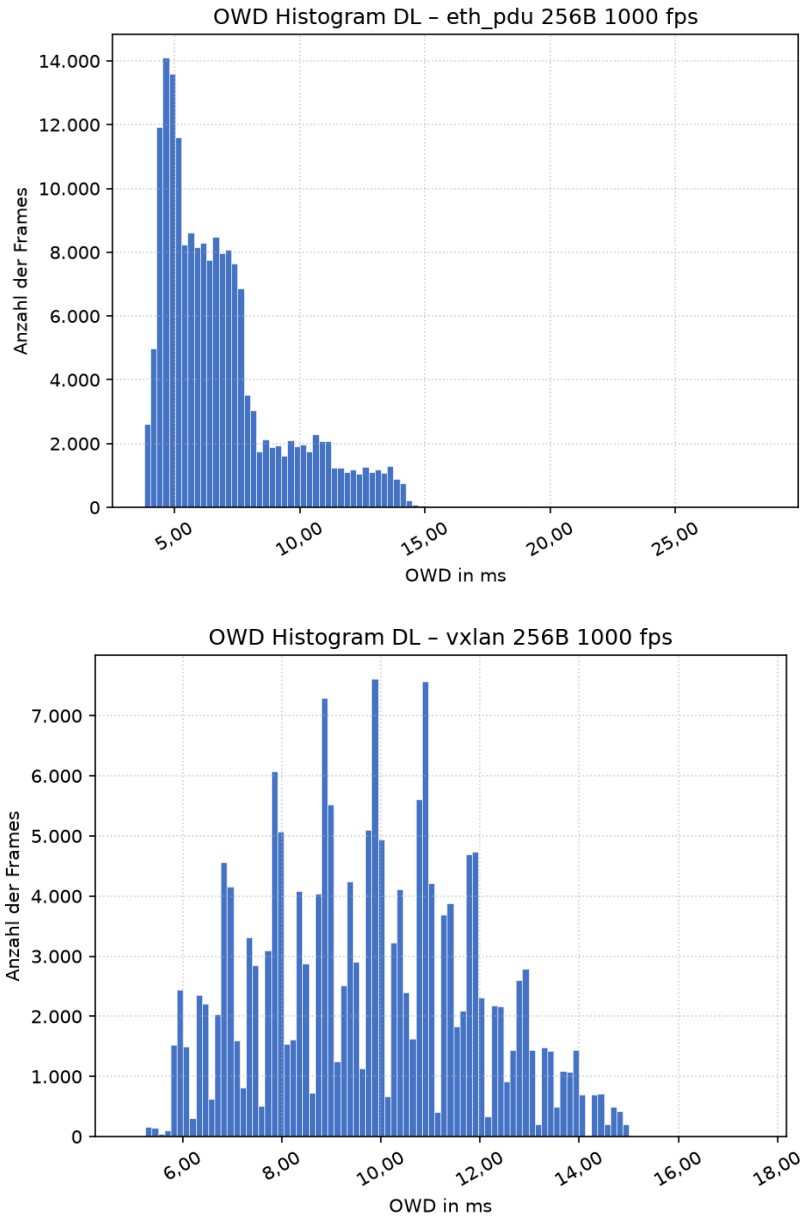


Abbildung 5: Histogramme der Messungen für den Downlink mit 256 Bytes Framegröße und 1000 Frames/s.



5 Fazit und Ausblick

Ethernet PDU Sessions sind besonders bei niedrigeren Zykluszeiten im Vorteil, da hier der Overhead der VXLAN Kapselung zu höherer Auslastung des VTEP-Gerätes führt und somit latenzrelevante Einflüsse häufiger auftreten. Ebenso sind Ethernet PDU Sessions bei niedrigen Framegrößen im Vorteil, da auch hier der Overhead der Kapselung gegenüber der Payload stärker ins Gewicht fällt. VXLAN erzeugt damit sowohl auf den VTEPs als auch auf dem 5G System mehr Last. Besonders bei Skalierung mit mehreren Layer 2 Teilnehmern im 5G System kann VXLAN Tunneling so eine signifikante zusätzliche Last im Kommunikationssystem erzeugen.

Im Vergleich zu anderen Optimierungen am 5G System, wie etwa von (Cainelli, et al., 2024) dargestellt, zeigen Ethernet PDU Sessions einen weniger starken Einfluss auf die Latenzen und somit auf die Performanz von Industrial Ethernet Anwendungen. Der Vorteil von Ethernet PDU Sessions ist jedoch, dass sie neben der Latenzverbesserung nativ Layer 2 Transparenz für 5G Systeme erzeugen. Dieser Schritt ist für Industrial Ethernet Anwendungen und auch für spätere TSN-Integrationen in die 5G Systeme unerlässlich. Zudem wird die Komplexität von UEs reduziert, da weniger Komponenten für ein Layer 2 transparentes 5G Setup benötigt werden. Der Aufwand Ethernet PDU Sessions in bestehenden 5G Systemen zu integrieren, sollte demnach genau gegenüber den Vorteilen für die betriebene Anwendung abgewogen werden. Vorteile werden primär in stärker ausgelasteten Systemen mit zahlreichen Layer 2 Kommunikationsendpunkten zum Tragen kommen.

Industrial Ethernet Anwendungen bestehen jedoch nicht nur aus Datenverkehr mit niedrigen Framegrößen, sodass besonders der Einfluss auf höhere Framegrößen im Nachgang genauer untersucht werden sollte. Besonders hier könnte die MTU-Reduktion durch VXLAN Overhead stark ins Gewicht fallen, sodass der Vorteil der Ethernet PDU Sessions noch deutlicher werden könnte.

6 Literaturverzeichnis

5G ACIA. (November 2019). Integration of Industrial Ethernet Networks with 5G Networks. *Integration of Industrial Ethernet Networks with 5G Networks*. 5G ACIA.

Bin Muslim, A., Tönjes, R., & Bauschert, T. (15. Mai 2024). *Synchronizing TSN Devices via 802.1AS over 5G Networks*. Von MPDI: https://www.mdpi.com/journal/electronics abgerufen

Cainelli, G., Kottkamp, M., Michelangelo, A., Kranefeld, N., Underberg, L., & Saunders, C. (2024). 5G Performance Analysis for Material Handling in Manufacturing. *Beiträge des Jahreskolloquiums KommA.* Lemgo.

Elmadani, M., & Sati, S. O. (28. Januar 2024). MTU Analyzing for Data Centers Interconnected Using VxLAN. 2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS).

KernelNewbies.org. (30. Dezember 2017). Abgerufen am 01. 10 2026 von https://kernelnewbies.org/Linux_3.7

Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., . . . Wright, C. (2014). *RFC Editor*. Von RFC Editor: https://www.rfc-editor.org/info/rfc7348/ abgerufen

Siemens AG. (2022). *PROFINET communication*. Von PROFINET communication: https://assets.new.siemens.com/siemens/assets/api/uuid:b0836a19-adcb-43fb-9fad-36519aa20449/profinet-communication-with-industrial-5g.pdf abgerufen

Flat, Resilient, Decentralized AAS Registries for Cross-Enterprise Value Networks

Jonas Kortner[1], Marc Riedlinger[1], and Florian Pethig[1]

Abstract:
Cross-enterprise resolution of Asset Administration Shell (AAS) identifiers is primarily a governance problem: in most contemporary industrial dataspace architectures, privileged operators concentrate power and have exclusive views of lookup metadata. Here, we propose a flat, non-globally-replicating AAS Registry realized as a permissioned Kademlia DHT over member-operated Registrar nodes with unbiased node identifiers, self-certifying records carrying chains of authority over transferable shell IDs, member-diverse replica placement against correlated failure, and a soft state maintained by republication. Assuming only centrally governed network admission, we analyze its structural security properties and exposure of lookup metadata. Publication and resolution require no privileged participant.
Keywords: Digital Twin • Asset Administration Shell • Dataspace • Governance

1 Introduction and Related Work

The digital twin (DT) is a cornerstone of the digital transformation of sectors and industries, in particular for manufacturing and the Industry 4.0 paradigm [1]. While 'DT' subsumes a range of concepts, it is understood that DT-based value creation derives from the scalable, secure, and sovereign sharing of interoperable, semantics-enriched twin data between value partners [2], [3]. This is particularly true for the IEC 63278-1 Asset Administration Shell (AAS) as the de-facto standard for industrial DTs coupled to asset lifecycles [4]. However, *a precondition of sharing twin data in a value network is that data is actually discoverable and retrievable in the network*.

Discovery of AAS and resolution of AAS endpoints across organizational boundaries is primarily a governance problem. Leaving aside considerations of robustness, operators of enabling services gain an advantage from privileged access to lookup metadata. IDTA, following Belyaev [5], provide and discuss a taxonomy of AAS registries [6]. In principle, a single authorized business entity could operate one global registry known to, and available for, the entire value network. However, such a centralized approach raises issues of power concentration and trust. A scenario of independent, enterprise-operated registries that may or may not interact with each other somewhat addresses the latter, but introduces concerns of complexity, discoverability, and equitable participation. Clearly, then, it is compelling to introduce a so-called 'registry-of-registries' (ROR) as a global (meta-)service that delegates to independent, non-interacting registries. While that approach inherits key issues of the fully centralized one, IDTA argue that an AAS ROR presently is the best feasible compromise, if only as a next step in the evolution of the AAS ecosystem.

In line with that assessment, participants in the Catena-X dataspace run their own, independent registries behind EDC connector endpoints that are discoverable via a chain of

1 Fraunhofer IOSB-INA, Campusallee 1, 32657 Lemgo, Deutschland, jonas.kortner@iosb-ina.fraunhofer.de; marc.riedlinger@iosb-ina.fraunhofer.de; florian.pethig@iosb-ina.fraunhofer.de

operator-run services, effectively implementing a hierarchy of non-interacting registries with top-level connector index as ROR [7], [8]. The Manufacturing-X initiative inherits this architecture [9], [10]: As non-interacting AAS registries behind Tractus-X EDC connectors, MX-Ports of the 'Hercules' variant rely on provision of a federated discovery. The 'LEO' variant skips dataspace-level discovery because IEC 61406-1 Identification Links (ILs) map to organizational domains directly, but requires fall-back on asset or metadata lookup at a federated layer when the IL is unknown. Approaches without a governed index layer exist, but demand global replication: Gaia-X lighthouse projects deploy Eclipse XFSC Federated Catalogs, essentially a globally indexed population of hierarchically equal, redundant, and synchronized catalogs akin to IDTA's interacting-registry model [11]. Others demonstrate twin registration, lookup, and owner change on distributed ledgers, trading integrity and sovereignty for network costs and metadata exposure that scale unfavorably with ecosystem size [6], [12].

To conceptualize decentralized approaches to the DT lookup problem, it pays to look beyond the Industry 4.0 ecosystem for inspiration. For example, the InterPlanetary Name System (IPNS) uses signed, mutable records resolved over a Kademlia distributed hash table to implement peer-to-peer hypermedia and file sharing at scale [13]. Here, we argue that a permissioned Kademlia resolution plane that avoids global AAS descriptor replication and constrains replica placement across organizational failure domains can close the gap of a decentralized, interacting but non-globally-replicating AAS registry in IDTA's taxonomy of AAS registries. To the best of our knowledge, such an approach has not been described yet.

2 Contribution, Scope, and Premise

In this work, we suggest to map the problem of lookup of AAS descriptors to lookup in a permissioned Kademlia distributed hash table (DHT) [14]–[16]. Importantly, a DHT does not rely on privileged nodes or global synchronization of descriptors and exposes only bounded quantities of lookup metadata to any single participant. We contribute elements of an architecture and its governance analysis, not an implementation or evaluation. We

- map the AAS Registry API to lookup in a permissioned Kademlia DHT over member-operated nodes and specify the required components: unbiased node identifiers bound to attested keys, self-certifying records carrying verifiable chains of authority, and a soft Registry state maintained by republication under finite storage leases;
- describe member-diverse replica placement to guard against correlated-failure scenarios of real-world value networks;
- and discuss the structural security properties of the architecture, evaluate the disclosure of lookup metadata as a governance question, and critically evaluate the attribution "decentralized".

Our suggestions of record format and Registry soft state follow established practice for Kademlia DHTs, notably the IPNS [13]. As novel contributions, we discuss a permissioned adaptation to the AAS Registry, member-diverse replica sets, and establishing authority over identifiers that neither derive from their owners' keys nor remain with a single owner.

2.1 The AAS Registry API

Exchange of AAS data at scale is enabled by the AAS API [17]. This interface is a system of independent, REST-ful maps from asset identifiers to shell and submodel IDs to individual data fields (submodel elements). As shown in Fig. 1, *Discovery* maps a globally unique asset ID to globally unique IDs of associated shells; *a Registry resolves one globally unique shell identifier to exactly one endpoint descriptor*. A descriptor, then, exposes all information required to access the shell via lower-level shell interfaces, assuming access privileges. In this work, we address the Registry only and assume clients hold shell IDs.

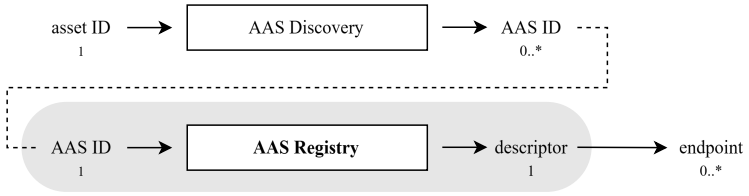


Fig. 1: Lookup of AAS data via the AAS Discovery and AAS Registry APIs [17]. Interfaces to the lower-level AAS and AAS Submodel repositories are not shown.

2.2 Value Network Setting and Assumptions

We consider a cross-enterprise value network of M sovereign members engaged in ongoing, administered AAS data exchange but who are neither necessarily trusting each other nor wanting to grant another a privileged view of their network metadata. We do not presuppose any particular institutional form of the network, but assume a membership authority (MA) that provides attestation of organizational identity via member keys, supplies an enumerable membership view, manages membership revocation, and issues Registrar credentials under an admission policy that limits each member to at most C concurrent (Kademlia) Registry nodes and meters the total number of credentials issued over time. (Both points are discussed in 3.3.)

We note in passing that such a value network resembles a dataspace and that existing dataspaces initiatives issue identifiers and credentials to their members just as assumed here. While we do not assume M, we concede that members in smaller, consortium-sized network may be able to feasibly query one another directly or appoint an operator to hold a global index. As M increases, issues of both engineering and trust arise; our work addresses only that regime. For now, we leave the question after the threshold for later work.

As a threat model, we consider two classes of adversaries: External adversaries are actors without network credentials who may want to observe traffic, attempt to register or modify AAS records, or attempt to inject nodes into the network. Honest-but-curious members have credentials and follow network protocol, but may seek to extract commercially or otherwise useful information from the network. We do not consider actively malicious insiders within

the scope of this work, but comment where design decisions have consequences under a stronger threat model.

2.3 Resource Lookup in Kademlia Peer-to-Peer Networks

Kademlia is a DHT for resource lookup in peer-to-peer overlay networks that do not operate an central, authoritative file register [14]–[16]. A resource is identified by a fixed-length binary string or 'key' of length m, practically a hash of file name or content. Crucially, the N network nodes or 'peers' that serve so-called provider records (i.e. resource endpoint information) have identifiers in the same key space. This way, one can define distance not just between peers, but also between a peer and a record.[2] Consequently, the network can place k redundant copies of a record, so-called replicas, with the k peers closest to its key. This is the DHT.

Peers communicate over the network by issuing remote procedure calls (RPCs) `PING`, `STORE`, `FIND_NODE`, and `FIND_VALUE`. When queried for a record r it does not keep in its replica store, a peer P can route the lookup to those of its contacts in a routing table that are close to the requested key. If any of these contacts hold r, they return it; else they return lists of *their* known closest contacts to P, and P queries *them*. This iterative lookup terminates in $O(\log N)$ steps. Because peers update routing tables (on a most-recently-seen basis with a bias towards established, stable nodes) with every RPC request and response, the Kademlia network learns from traffic. Node churn is addressed by issuing finite times-to-live (TTL) to replica store entries and, hence, a requirement that peers periodically re-publish their records.

As of today, implementations of Kademlia variants drive thousands of (generally large) peer-to-peer networks such as I2P and the IPNS [15], [18], [19]. We refer to the bibliography for more detailed presentations of concepts, algorithms, and applications.

3 Kademlia Registrars of Shell Descriptors

In this section, we discuss a network of dedicated member-operated Registrar (peer) nodes that, collectively, supply a flat, resilient, and decentralized AAS Registry.[3] In dependency order, we discuss components and behavior of Registrar nodes in the network; self-certifying records and their distribution among Registrar replica stores; and network operations and processes that maintain the Registry soft state. Throughout, we assume some cryptographic hash function, h, that maps to the Kademlia key space $\{0,1\}^m$. (For example, for SHA-256, $m = 256$.) An examination of the attribution of "flat, resilient, and decentralized" is provided in sec. 4.

2 In Kademlia, the distance between two keys is the bit-wise XOR interpreted as a base-2 integer. This map has all properties of a metric: positivity, non-degeneracy, symmetry, and the triangle inequality.

3 We capitalize the Kademlia 'Registry' and refer by 'registry' to non-interacting, member-internal services or the general concept.

3.1 Partner-Scoped Shell Registries and Registrar Nodes

In real industrial settings, AAS *hosts* – typically shop-floor devices, type repositories, or PML or ERP systems that expose AAS-compliant interfaces – are generally not reachable from the internet. Hence, these hosts cannot themselves participate in a Registry. Instead, we propose a population of member-operated *Registrars*: internet-reachable DHT nodes that maintain routing state, forward lookups, and store replicas of other partner's records as well as expose a standard-conformant AAS registry API to clients within their own member network. From a member point of view, *this network of Registrars implements the Registry*; c.f. Fig. 2. Importantly, a Registrar is owned and operated by exactly one value network member that can be identified and verified – via its credentials, attested by the MA – by other members.[4]

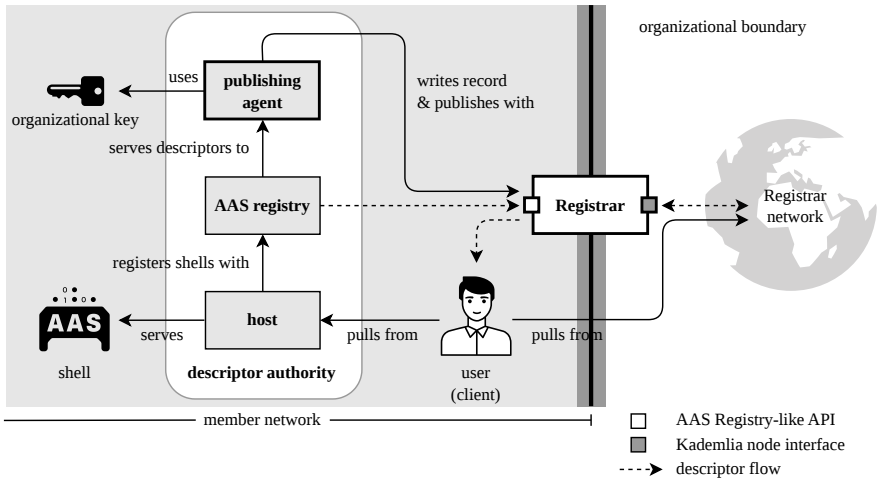


Fig. 2: Outline of the relationships between member AAS, AAS hosts, member-internal AAS registries, and member (Kademlia) Registrar nodes.

Members register some or all of their shells with their Registrars through their *descriptor authority* (DA). In practice, this DA subsumes existing AAS repositories, member-internal registries, and a publication agent. The DA creates, updates, publishes, withdraws, and signs records with the organizational key attested by the value network MA.

3.2 Self-Certifying Records

A Registry *record* wraps around a standard AAS ID and descriptor payload with additional attributes: the organizational identity of the issuer, a strictly monotonic version counter, a

4 We note in passing that not all value network members necessarily need to operate Registrars. Downstream value partners may only act as consumers and attach to Registrars via lean clients. Similarly, in principle, a host could register shells with a Registrar and not contribute to the Registry infrastructure (but thereby trading operational burden for dependency on its sponsoring Registrar for publication and resolution).

validity range, and a JWS-style detached signature (over the JCS-canonicalized record) under the issuer DA's key. The descriptor itself can be resolved to an endpoint at the organizational boundary of the member that owns the shell [17]. Such records are established practice in, e.g., IPNS [13]. However, unlike IPNS names, an AAS identifier bears no relation to organizational key and may change owners; records from a previous and a current owner can coexist in the Registry, with incomparable version counters. Therefore, we posit that each record carries a chain of authority over its shell: its first issuer, followed by any transfer statements, each signed by the then-current authority and naming the successor. A resolver, then, would accept only records issued by the current authority and, among these, the highest version.

3.3 Registrar Function and Form

As outlined in 3.1, Registrar nodes supply a network-wide AAS Registry service. A member publishes shells via their Registrars as well as queries them for descriptors for IDs of their own or another member's shells. Therefore, firstly, a Registrar must expose an AAS registry API conformant to Part 2 of the AAS specification towards its owning member; c.f. Fig. 1. To guarantee that lookups of member-owned descriptors succeed independent of network connectivity, a Registrar must cache, or at least route to, member-internal registries. Records belonging to other members, however, are retrieved from the Kademlia DHT (but using the same AAS registry API); c.f. Fig. 2. This requires, secondly, an outside- or network-facing interface for Registrar-to-Registrar communication via the standard Kademlia RPCs.

To enable these, a Kademlia Registrar ρ must have a unique ID i_ρ in key space. To mitigate key grinding, we suggest to assign to i_ρ the hash of a (public) key submitted for attestation concatenated with a nonce drawn uniformly at random from $\{0, 1\}^m$ by the MA. We require that, furthermore, ρ holds verifiable membership credentials, a Registrar key pair attested by the MA, as well as the trust anchor to verify credentials and signatures of other Registrars.

A Kademlia Registrar participates in the DHT by keeping a *replica store* of records that are near its ID in key space. Essentially, this is a dynamically updated list of copies of records (as key–value pairs) that have an ID close to i_ρ. We discuss placement of records in replica stores in 3.4 but preempt that, as a matter of Registry hygiene, Registrars assign finite times-to-live (TTLs) to stored replicas as well as periodically (re-)publish a cache of their own records to other Registrars; c.f. 3.5.

Lastly, a Registrar must keep a *routing table* of contacts, i.e. other Registrars, to be able to route lookup queries around the network. As the network changes – members join and leave, suffer outages, etc. – as well as upon interacting with other Registrars via RPCs, Kademlia Registrars updates their table with a dedicated bias towards established, long-lived nodes. For a comprehensive discussion also of specific implementations, we refer the reader to the Kademlia literature.

3.4 Publication and Member-Diverse Replica Placement

To publish or update a descriptor of an AAS a with the network, a member's DA assembles the record, attaches the chain of authority over a (extended by a transfer statement where authority has changed), increments the version counter, signs with the member's organizational key, and computes its key-space ID, $i_a = h(a)$. As in the standard Kademlia construction, any member-owned Registrar can publish that record by finding the k Registrars closest to i_a – the so-called *replica set* – and issuing `STORE` RPCs.

Here, however, we propose to enforce a *diversity rule* for replica placement. In real value networks, Registrar failures cannot be modeled as strictly independent; when a member leaves the network or suffers an outage at the enterprise level, all of their Registrars leave simultaneously. In principle, this could affect most or even all Registrars in a replica set (for a time up to the system-wide republication interval, as discussed below).

To mitigate, we require that *no more than μ of the k Registrars closest to a given key shall be operated by the same member* as follows: For placement of a record with key i_a, order all Registrars ρ by increasing distance from i_a in the chain $\mathcal{R}(a)$. Traverse $\mathcal{R}$ in order and admit $\rho \in \mathcal{R}$ to i_a's replica set iff that replica set contains less than μ Registrars owned by the same member, terminating at k entries. The resulting *member-diverse replica set* spans Registrars maximally close under the diversity rule and from at least $\lceil k/\mu \rceil$ distinct members.[5] Consequently, since the member-diverse replica set is in general not a subset of the k Registrars closest to i_a, resolvers must widen the termination condition of the iterative lookup to the $k' = \lceil k \cdot C/\mu_{\text{eff}} \rceil$ closest contacts, with $\mu_{\text{eff}} = \min(\mu, C)$ and C being the maximum number of Registrars per member, to remain guaranteed to reach a replica holder.[6] Clearly, $\mu \geq C$ recovers the standard construction.

3.5 The Registry Soft State

Even in the absence of correlated-failure scenarios, regular network activity – such as (de-)commissioning of shells, member churn, etc. – strains the Registry state. As in standard Kademlia and IPNS , therefore, the Registry state is soft by design [13], [14]: A replica, once published and placed, is assigned a finite expiry equal to the lesser of its validity and a *storage lease* T_{sl} set by its holding Registrar. At expiry, the record is dropped from the store.

Consequently, records must be republished for as long as they are to remain resolvable. We distinguish two roles. Firstly, Registrars republish their own records at a Registry-wide republication interval $T_{\text{re}} < T_{\text{sl}}$. This also re-computes replica sets over the current, potentially changed membership. Furthermore, Registrars republish their replica stores at a

5 We tacitly assume a Registrar population larger then the replication parameter, as well as more than $\lceil k/\mu \rceil$ members in the network.

6 Let the horizon H be the index of the last Registrar in $\mathcal{R}$ inspected by the traversal. Then, $H = k + z$ with z the number of Registrars skipped. A Registrar is skipped only if its member is already saturated, i.e. the replica set already contains μ_{eff} Registrars belonging to its member. Since each Registrar has exactly one member, there are at most $\lfloor k/\mu_{\text{eff}} \rfloor$ saturated different members in a replica set, and the cap C means that each member has at most $C - \mu_{\text{eff}}$ Registrars skipped. Hence $H \leq k + (C - \mu_{\text{eff}}) \cdot \lfloor k/\mu_{\text{eff}} \rfloor \leq \lceil k \cdot C/\mu_{\text{eff}} \rceil = k'$. Consequently, no admitted replica can lie beyond rank k' in the distance ordering, and a resolver whose lookup terminates at the k' closest contacts is certain to reach a replica holder if the record was published at all.

shorter interval, pushing each replica to the current member-diverse replica set (omitting replicas for which a STORE RPC has been received in the current interval). This keeps replica sets current under churn between owner republications and while an owner's Registrars are unavailable, and is entirely permissible since records are self-certifying. A replica passed on that way retains its remaining lease; only a STORE issued by a Registrar of the record's current authority renews it, so that records of departed members still expire.

Practically, the republication interval T_{re} is the policy parameter that determines cost; $T_{\mathrm{sl}} > T_{\mathrm{re}}$ is chosen accordingly. Consequently, the Registry state is soft in the strict sense that no copy of a record persists on its own account. Importantly, removal of records for decommissioned shells or departed members occurs automatically while replica sets are automatically updated under network churn. We note in passing that a soft Registry state – with republication as an obligation – also incentivizes members to operate Registrars in redundancy. A partner with only a single Registrar that is unavailable long enough for leases to lapse will not only lose their own resolution capability, but furthermore observe its replicas dropping off the Registry.

4 Security, Governance, and Limitations

Having outlined the overall Registry architecture in terms of its components and processes, we discuss security concerns, governance matters, and limitations.

4.1 Concerns of Security

At the level of individual records, integrity of Registry content is guaranteed cryptographically because records are signed under the owning member's organizational key and because members holds the MA trust anchor. Claiming authority over Registry content, i.e. another member's shell, is technically possible, but remains attributable: competing claims necessarily carry a chain of authority rooted at, and signed by, the claimant and would be flagged as a conflict by conforming Registrars. (Under the honest-but-curious model, such conflicts arise only by error.)

We observe that a signature under a valid record does not establish the non-existence of more recent, updated versions of that record. Under our threat model, such staleness is not adversarial: expiring leases, republication, and replica sets recomputed under member churn admit windows in which deprecated versions are served in good faith.[7] However, because Kademlia's iterative lookup is performed in parallel over several members of the replica set [14], Registrars and clients can resolve conflicting lookup responses in favor of the highest version within the prevailing chain of authority, and record TTLs are bounded by the lesser of the remaining validity interval and T_{sl}. We acknowledge but do not consider the issue critical in our (weaker) threat model.

At the network level, two attacks on open DHT deployments must be considered [16]. In a Sybil attack, an adversary registers inflated numbers of nodes to attain influence over the network. Here, this threat is answered by admission: the MA caps the number of concurrent

7 An actively malicious actor could serve stale records deliberately.

Registrars per member, and IDs cannot be minted. From this point of view, C is as much a security as a governance parameter. In an Eclipse attack, an adversary surrounds a chosen key with surplus nodes to suppress or falsify its resolution. If i_ρ incorporates entropy drawn by the MA after key commitment (as suggested in 3.3), a member cannot choose its position in key space offline. They *can* discard a credential and request another; a freshly issued Registrar ID is drawn among the k Registrars closest to a given key with probability $\sim k/N$ such that $\sim N/k$ issuances suffice to target that key. Consequently, any robust issuance policy must count, rate-limit, and log every issuance. We note that unbiased placement matters even under the honest-but-curious model: a member able to position Registrars near a competitor's keys could select which lookups it observes, defeating the argument of 4.2. Likewise, a member who legitimately operates many Registrars is, by construction, a legitimate partial eclipse of some region in key space (although the diversity rule of 3.4 can mitigate this, depending on the value of μ).

We note that the preceding arguments presuppose Registrar-to-Registrar communication over mutually authenticated channels as opposed to standard Kademila over UDP. Consequently, credentials must be short-lived or checked against a disseminated status list for one Registrar to recognize a revoked identity of a member.

4.2 Exposure of Replica Store and Lookup Metadata

Since a Registrar stores both records of their own as well as of other members, its operator can infer the existence of shells, their owner, and their shell descriptors (and boundary endpoints) at any time. To mitigate this disclosure, a member concerned with what their endpoint structure reveals may publish only a single indirection endpoint at which their own infrastructure performs internal resolution. Alternatively, a member may encrypt descriptor payloads under a key held by a privileged subset of members.

Furthermore, Registrars on a routing path – and by extension: their operators – observe which keys are resolved, by whom, and when. This is a concern in a value network where patterns of interest reveal sourcing relationships. However, in a centralized or ROR scenario, registry operators observe precisely the same traffic not only in its entirety, but moreover also from a structurally and contractually privileged position [5], [6]. Therefore, the relevant question is not whether lookup metadata is exposed but to whom, under what accountability, and with what capacity to leverage it.

As a governance judgment – not a technical result –, we take the view that disclosure of fragmentary metadata to members bearing symmetric obligations is preferable to complete metadata exposure to any single operator.

4.3 Governance and the Scope of Decentralization

While a Kademlia Registry decentralizes its capacity to resolve shell IDs to descriptors – an activity in which members perform on equal terms –, the architecture described here relies on centralized management of admission and membership of the network. In particular, a trusted MA is required to certify public keys against organizational identities, to draw the

entropy from which Registrar IDs derive, to enforce Registrar issuance policy, maintain a current membership view, and revoke membership. We argue that reliance on a (centralized) MA does not compromise the attribution of "decentralized". Firstly, the MA plays no active role in resolution; a lookup performed while the MA is unreachable succeeds regardless because verification requires only the trust anchor and previously issued credentials, both held locally by every Registrar. Consequently, the MA cannot observe or preferentially resolve queries, suppress records, or in fact, access any Registry metadata beyond the shared membership view. Furthermore, the MA does not attest ownership of individual shells. Crucially, that would require the MA to hold per-shell state and attestate authority along the publication path, reintroducing a privileged-operator role. Instead, authority over shells is established by chains of authority carried in the records themselves. If one accepts that *some* managed membership boundary is necessary – a Registry accepting anonymous registrations would fail the integrity requirements of 4.1 outright –, an MA whose role is strictly constrained to network admission only is perhaps the easiest to justify.

4.4 Limitations and Avenues for Further Research

We have deliberately made no assumptions about the internal structure of AAS identifiers. In a scenario where shell IDs carry an authority component – such as with IEC 61406-1 ILs –, delegated resolution (in the manner of DNS) should be preferred over the approach described here [10]. Without a global consensus mechanism of ledger-based approaches, one cannot choose more than two out of freedom freedom from assumptions on identifier structure, a flat registry without privileged per-shell state, and protocol-level adjudication over shell IDs. Here we suggest to compromise on the adjudication of authority, i.e. relying on the MA to resolve and ultimately revocate conflicting claims. Furthermore, compared to a registry service that indexes shells in member-internal repositories, the Kademlia Registry admits no efficient proof of non-existence of a descriptor for a given shell. A null return only establishes that no record was found within the lookup horizon. Moreover, the `QueryAssetAdministrationShellDescriptors` Operation of the AAS Registry API (or other filtering, semantic search, etc.) is not possible over a DHT.[8] However, such provisions are conceptually closer to catalog services layered above the Registry, out of scope of this work.

The Registry requires active maintenance of its soft state. For records to remain resolvable, the aggregate cost of republication scales roughly as $|D|k/T_{\mathrm{re}}$ in the globally registered records, D. These and other operational efforts – Registrar provision, patching and monitoring, key management, etc. – are distributed among network participants while Registry performance depends in part on the least competent member. Whether or not such technical limitations of a flat architecture are acceptable is a matter of governance considerations.

Regardless of the flatness and decentralized nature of the Kademlia Registry, censorship by a collusion of credentialed members remains a possibility in principle. Unbiased Registrar IDs under a metered issuance budget and a cap on concurrent Registrars per member,

8 Likewise, the AAS Registry Operation `GetAllAssetAdministrationShellDescriptors` is not sensible here.

parallel iterative lookups, and a diversity bound μ chosen sufficiently small raise the number of colluding members required to control majorities of replica sets and deny a record. Addressing such attacks calls for an extension of our threat model and further research.

Finally, the robustness claim of our work remains structural for now. Clearly, more centralized or ROR-like deployments are suggestive of single points of failure (although most robustness concerns can practically be mitigated by redundancies of infrastructure components). However, we have not evaluated our claim. Here, a quantitative treatment is required in terms of the network parameters we introduced but have left unspecified: the replication factor k, the diversity bound μ, and the replication interval, T_{re}, against which the duration of a correlated-failure outage trades directly. Similarly, we have not assumed the number of members in the value network, M, but do acknowledge that iterative lookup of records over a (Kademlia) DHT is unwarranted for in scenarios where M is sufficiently small that members can query each other directly. Considerations of governance aside, investigating the crossover would be an interesting inquiry. We intent to address these matters by simulation in subsequent work.

5 Conclusion

We have proposed a flat AAS Registry in which member-operated Registrars resolve shell identifiers over a permissioned Kademlia DHT, involving no privileged participant for publication or resolution. Furthermore, no global replication of descriptors is required. Building on established DHT practice and unbiased Registrar identifiers, we contributed member-diverse replica placement against the correlated failures of real value networks and chains of authority over transferable shell IDs. Admission to the network remains centrally governed. The architecture we propose requires sustained maintenance of a Registry soft state, is limited to exact-match lookup only, and can only attribute (rather than adjudicate) conflicting claims of authority over shells. A quantitative evaluation of resilience and of the scale at which the proposed architecture becomes warranted, as well as an extension of the threat model to actively malicious insiders, remain for future work.

References

[1] F. Tao, H. Zhang, and C. Zhang, “Advancements and challenges of digital twins in industry,” *Nature Computational Science*, vol. 4, no. 3, pp. 169–177, 2024.

[2] M. Redeker, S. Volgmann, F. Pethig, and J. Kalhoff, “Towards Data Sovereignty of Asset Administration Shells across Value Added Chains,” in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, IEEE, 2020, pp. 1151–1154.

[3] J. Mügge, A. Seegrün, T.-K. Hoyer, T. Riedelsheimer, and K. Lindow, “Digital Twins within the Circular Economy: Literature Review and Concept Presentation,” *Sustainability*, vol. 16, no. 7, p. 2748, 2024.

[4] P. K. Gupta, J. Heller, and B. Lüdemann-Ravit, “Asset Administration Shell: Deutschlands Standard für Digitale Zwillinge,” *Zeitschrift für wissenschaftlichen Fabrikbetrieb*, vol. 120, pp. 755–762, 2025.

[5] A. Belyaev, J. Hasler, C. Urban, and C. Diedrich, "Architektonische Gestaltungsprinzipien einer dezentralen Industrie 4.0 -Infrastruktur - Taxonomie von dezentralen Registry und Discovery," in *EKA 2022 – Entwurf komplexer Automatisierungssysteme, 17. Fachtagung*, 2022, pp. 1–12.

[6] "Decentralized Registries: Taxonomy of decentralized registries and an architectural overview," Industrial Digital Twin Association, Frankfurt, Tech. Rep., Apr. 2023.

[7] C. Schmidt, F. Volz, L. Stojanovic, and H. Kett, "Integration Approaches for Digital Twins in Dataspaces," *Applied Sciences*, vol. 15, no. 21, p. 11 623, 2025.

[8] F. Schöppenthau, F. Patzer, B. Schnebel, K. Watson, N. Baryschnikov, B. Obst, Y. Chauhan, D. Kaever, T. Usländer, and P. Kulkarni, "Building a Digital Manufacturing as a Service Ecosystem for Catena-X," *Sensors*, vol. 23, no. 17, p. 7396, 2023.

[9] M. Neubauer, L. Steinle, C. Reiff, S. Ajdinović, L. Klingel, A. Lechler, and A. Verl, "Architecture for manufacturing-X: Bringing asset administration shell, eclipse dataspace connector and OPC UA together," *Manufacturing Letters*, vol. 37, pp. 1–6, 2023.

[10] F. Zink, L. Fink, B. Wallner, F. Bleicher, and T. Trautner, "Advancing the Collaboration in Manufacturing Networks A Systematic Literature Review on Implementations of Asset Administration Shells and Data Spaces," *Procedia CIRP*, vol. 136, pp. 546–551, 2025.

[11] B. T. Arnold, K. Baydoun, D. Collarana, S. Duda, C. Gillmann, A. Hemid, P. Hertweck, P. Moosmann, D. Sukhoroslov, and C. Lange, "XFSC: A catalogue of trustable semantic metadata for data services and providers," *arXiv preprint arXiv:2501.14473*, 2025.

[12] B. Putz, M. Dietz, P. Empl, and G. Pernul, "EtherTwin: Blockchain-based Secure Digital Twin Information Management," *Information Processing and Management*, vol. 58, no. 1, p. 102 425, 2021.

[13] V. Santos, S. Allen, M. Rataj, H. Dias, and G. Eggert. "IPNS Record and Protocol," IPFS Specifications. (), [Online]. Available: `https://specs.ipfs.tech/ipns/ipns-record/` (visited on 09/28/2026).

[14] P. Maymounkov and D. Mazières, "Kademlia: A Peer-to-Peer Information System Based on the XOR Metric," in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, ser. IPTPS '01, Berlin, Heidelberg: Springer-Verlag, 2002, pp. 53–65.

[15] D. Trautwein, A. Raman, G. Tyson, I. Castro, W. Scott, M. Schubotz, B. Gipp, and Y. Psaras, "Design and evaluation of IPFS," in *Proceedings of the ACM SIGCOMM 2022 Conference*, ACM, 2022, pp. 739–752.

[16] G. Urdaneta, G. Pierre, and M. V. Steen, "A survey of DHT security techniques," *ACM Computing Surveys*, vol. 43, no. 2, pp. 1–49, 2011.

[17] "Specification of the Asset Administration Shell Part 2: Application Programming Interfaces," Industrial Digital Twin Association, Frankfurt, Tech. Rep., Apr. 2023.

[18] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE Communications Surveys and Tutorials*, vol. 7, no. 2, pp. 72–93, 2005.

[19] Y. Thomas, N. Fotiou, I. Pittaras, G. Xylomenos, S. Voulgaris, and G. C. Polyzos, "Peer Clustering for the InterPlanetary File System," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Future of Internet Routing and Addressing*, ACM, 2023, pp. 8–14.

PROFINET IRT mit COTS-Hardware in einer Linux-Umgebung

Dr. Harald Albrecht [1], Dr. Stephan Höme [2] und Marcel Albert[3]

Abstract: Mit TSN-Elementen der IEEE/IEC 60802 in Verbindung mit COTS-Netzwerkkarten und Link-Speeds ab 1 Gbit/s lassen sich hoch performante PROFINET-Controller auf Standard-Hardware umsetzen. Diese skalieren flexibel bezüglich CPU- und Netzwerkleistung, binden existierende Brownfield-Geräte nahtlos ein und ermöglichen die Konvergenz von harter Echtzeit (OT) und Industrial-IoT-Anwendungen (IIoT) auf einer gemeinsamen Plattform. Während softwaredefinierte Architekturen große Freiheiten eröffnen, setzt der Wunsch nach wirtschaftlicher COTS-Hard- und Software – insbesondere unter Realtime-Linux – solide Leitplanken. Dieser Beitrag analysiert die softwarearchitektonischen Voraussetzungen im Linux-Kernel für das Verwalten, Isolieren und Priorisieren von Kommunikations- und Rechenressourcen und zeigt die Grenzen sowie Best Practices für Container-basierte Automatisierungs- und IoT-Workloads auf.

Keywords: PROFINET, Commercial off-the-shelf (COTS), Software-Defined Automation, Industrial IoT, Echtzeit-Linux, IEEE/IEC 60802, Container-Virtualisierung, Docker

1 PROFINET IRT auf COTS-Hardware

PROFINET RT wird heute sowohl auf Controller- als auch auf Device-Seite regelmäßig auf Standard-Hardware realisiert. Die Verarbeitung der zyklischen Echtzeitdaten kann dabei mit den Mechanismen marktüblicher Prozessoren und Netzwerkschnittstellen erfolgen. Für PROFINET IRT (Isochronous Real-Time) gelten hingegen deutlich höhere Anforderungen: Kommunikation und Applikation müssen mit hoher zeitlicher Genauigkeit aufeinander abgestimmt werden, und die Übertragung der Echtzeitdaten muss auch bei zusätzlichem Netzwerkverkehr deterministisch erfolgen [PI-PN].

PROFINET IRT erreicht diesen Determinismus durch Erweiterungen der Ethernet-Kommunikation auf Basis von IEEE 802.1. Die ursprünglichen IRT-Mechanismen wurden allerdings spezifiziert, bevor die heute unter Time-Sensitive Networking (TSN) zusammengefassten Ethernet-Erweiterungen verfügbar waren. Moderne Standard-Netzwerkcontroller integrieren inzwischen wesentliche TSN-Funktionen in Hardware. Damit entsteht die

[1] Siemens AG, DI AUT SP ICC, harald.albrecht@siemens.com, https://orcid.org/0009-0006-6548-8424
[2] Siemens AG, DI AUT SP ICC, stephan.hoeme@siemens.com, https://orcid.org/0000-0002-3543-4330
[3] Siemens AG, DI AUT SP ICC, m.albert@siemens.com

Möglichkeit, die für PROFINET IRT benötigten Funktionen auf COTS-Hardware abzubilden, anstatt hierfür ausschließlich spezifische Kommunikationsbausteine zu verwenden.

Ziel dieses Ansatzes ist es, PROFINET IRT insbesondere auf leistungsfähiger Server-Grade-Hardware skalierbar zu machen und zugleich die Verbreitung im Hardware- und Software-Ökosystem zu erleichtern. Höhere Link-Speeds ermöglichen zudem konvergente Netzwerke, in denen Teilnehmer mit unterschiedlichen Datenraten gemeinsam betrieben werden. So kann ein leistungsfähiger Backbone mit 1 Gbit/s oder mehr über geeignete Switches weiterhin 100-Mbit/s-Stränge mit bestehenden PROFINET-Geräten anbinden.

1.1 Ansatz

Der in diesem Beitrag betrachtete Ansatz konzentriert sich auf einen PROFINET-IRT-Controller, dessen Ethernet-Anbindung auf einem TSN-fähigen Standard-NIC beruht. Als konkrete Beispiele dienen die Intel-Netzwerkcontroller I210 und I226. Deren in Hardware integrierte Funktionen für Zeitsynchronisation und zeitgesteuerte Übertragung werden genutzt, um die erforderlichen PROFINET-IRT-Mechanismen abzubilden. Der architektonische Rahmen ist das TSN-Profil IEC/IEEE 60802 [IEC60802] für die industrielle Automatisierung.

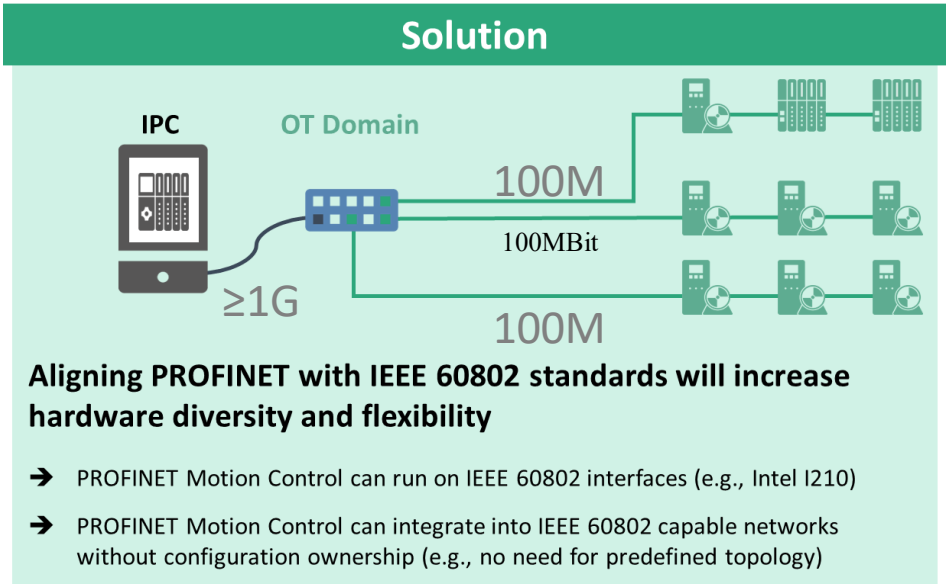


Abbildung 1: PROFINET und IEC/IEEE 60802.

Im Unterschied zu einer rein softwareseitigen Umsetzung werden zeitkritische Funktionen soweit möglich in der Netzwerkkarten-Hardware ausgeführt. Die Controller-Software konfiguriert diese Ressourcen und stimmt Kommunikationszyklus, Zeitsynchronisation und Applikationsverarbeitung aufeinander ab. Dadurch bleibt die Lösung eng an verbreiteter Hardware- und Betriebssystemmechanismen, während die für IRT erforderliche zeitliche Präzision erhalten wird.

Um die strikten Zeitanforderungen von PROFINET IRT auf Standard-Controllern wie dem Intel I210 oder I226 einzuhalten, werden dedizierte Hardware-Mechanismen genutzt: Die Uhrzeitsynchronisation erfolgt konform zu IEEE 802.1AS über die integrierte Hardware-Uhr (PTP Hardware Clock, PHC), wodurch Zeitstempel direkt auf PHY/MAC-Ebene ohne Software-Jitter generiert werden. Das zeitgesteuerte Senden zyklischer Frames wird über zeitbasierte Übertragungsmechanismen des Linux-Netzwerk-Subsystems (wie ETF-Qdisc bzw. Hardware-LaunchTime via SO_TXTIME) an den MAC-Transmitter gekoppelt. Dadurch wird die Frame-Übertragung hardwarenah im Sendepfad synchronisiert, während der Linux-Host die zyklische Aufbereitung der Nutzdaten im Applikationszyklus übernimmt.

1.2 Ergebnis

Die praktische Machbarkeit wurde anhand eines Demonstrators für PROFINET Motion Control nachgewiesen, der auf der SPS 2025 erstmals öffentlich präsentiert wurde. Der Controller lief auf einem handelsüblichen Industrie-PC mit x86-64-Prozessor der Intel-Core-i-Familie und PREEMPT_RT-Linux. Über eine Intel-I226-Netzwerkkarte mit 2,5 Gbit/s steuerte er mehrere reale PROFINET-IRT-Antriebe an.

Im Rahmen der Validierung wurde ein stabiler Applikations- und Sendezyklus von 1 ms realisiert. Die Zyklus-Jitter Messungen über mehrere Millionen Zyklen belegen:

- Sende- und Verarbeitungsjitter: Bei Zuweisung des Echtzeittasks auf einen isolierten CPU-Kern lag der maximale Jitter der zyklischen IRT-Pakete reproduzierbar unter ±10µs. Im Netzwerkbetrieb stellt diese geringe Toleranz kein Problem dar, da moderne PROFINET- bzw. TSN-Bridges durch den Einsatz von relativen Forwardern (Relative Forwarding) diesen Jitter innerhalb des Übertragungsfensters vollständig ausgleichen, sodass die isochrone Synchronisation der nachgelagerten Antriebe zu jedem Zeitpunkt absolut deterministisch gewährleistet bleibt.

- Koexistenz unter Störlast: Auch bei simultaner Beaufschlagung des Systems mit unkorreliertem TCP/IP-Hintergrundverkehr (1 Gbit/s Best-Effort-Traffic via iperf3) traten dank der hardwaregestützten Trennung und Priorisierung weder Zyklusausfälle noch Frame-Verluste im IRT-Kanal auf.

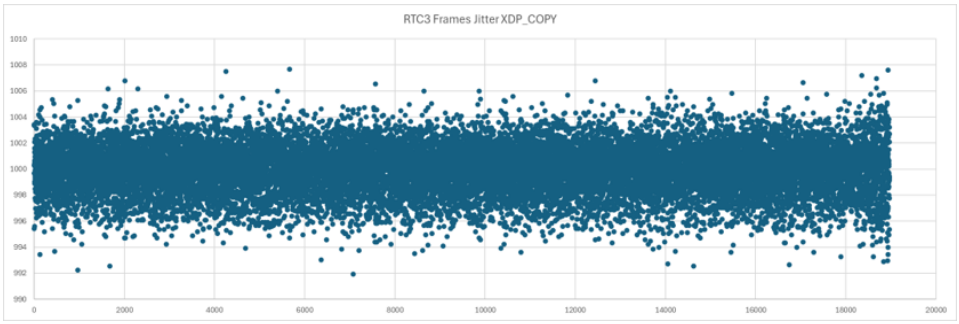


Abbildung 2: Sende- und Verarbeitungsjitter von maximal ±10µs bei 1ms Zykluszeit über Zyklen/Zeit

Dies belegt, dass COTS-basierte Standard-Ethernet-Controller die notwendige zeitliche Schärfe für anspruchsvolle Motion-Control-Anwendungen im PROFINET-Umfeld verlässlich bereitstellen können.

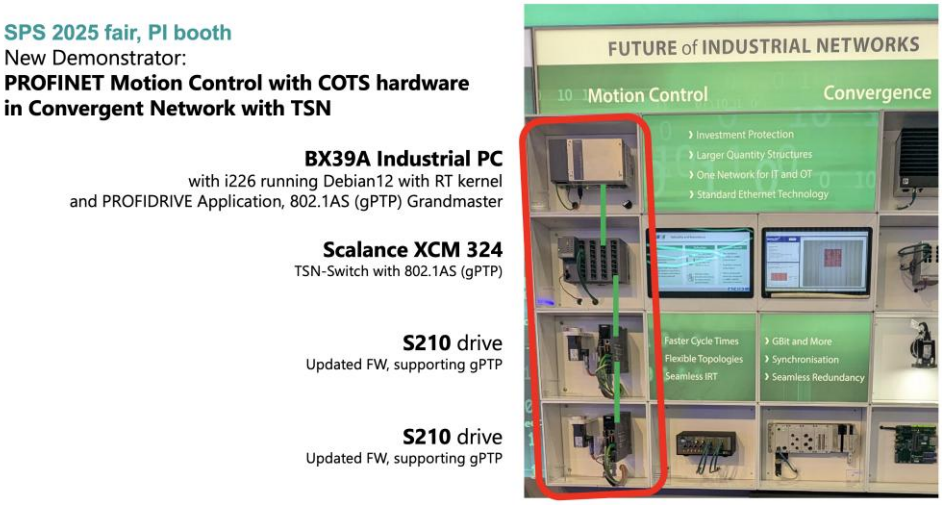


Abbildung 3: Demonstrator für PROFINET Motion Control mit COTS-Hardware.

2 Linux und Docker als COTS-Software-Plattformen

Aus Sicht einer Reihe von Anwendern geht der Einsatz von COTS-Hardware zugleich mit dem Einsatz von COTS-Software und COTS-Software-Plattformen einher. Sie erhoffen sich einen positiven Nutzen aus der Vielfalt bei Hardware und Software, inklusive des breit vorhandenen Wissens und verbesserter Software-Prozesse. Nicht zuletzt trainieren heute wichtige LLM-Anbieter ihre Modelle schwerpunktmäßig in den Bereichen von COTS-Software-Plattformen.

Dieser Beitrag klammert bewusst das „Data Plane Development Kit“ (DPDK, [DPDK]) aus, da es einen eigenen, überschaubaren Satz von Userspace Treibern getrennt vom Linux-Kernel pflegt und damit bis heute nicht unter das COTS-Kriterium fällt, dass die breite Hardware-Unterstützung des Linux-Kernels hingegen problemlos erfüllt. Zudem existiert speziell für Docker keine fertige, gepflegte Docker-Integration, sondern nur die dazu inkompatible Kubernetes/CNI-Integration [CNI] (eine Integration von CNI-Plugins in Docker ist aufgrund des deutlich unterschiedlichen Verwaltungsmodells der Netzwerkschnittstellen bei Docker und dessen libnetwork leider nicht praxistauglich möglich; die zugehörige Diskussion würde leider den Umfang dieses Beitrags sprengen).

Die treibende Motivation für den Einsatz von Standard-Linux und Container-Plattformen liegt in der Konsolidierung ehemals getrennter Welten (OT und IoT) auf einer gemeinsamen Hardware:

- Deterministische Feldkommunikation (OT): Zyklische PROFINET-Steuerung von Antrieben und I/O-Modulen im Sub-Millisekundenbereich.
- Lokale Datenanalyse und Bildverarbeitung (Industrial Edge): Auswertung hochauflösender Bilddatenströme (z. B. 4K-Kameras zur visuellen Qualitätskontrolle mittels Machine Learning) direkt an der Maschine.
- IoT- und Cloud-Konnektivität (IIoT): Protokollübersetzung und Zustandsüberwachung via OPC UA, MQTT oder REST-Schnittstellen zu übergeordneten IT-Systemen.

Das wirtschaftliche Versprechen von COTS besteht darin, diese heterogenen Aufgaben nicht mehr auf separaten SPS- und IPC-Geräten abzuwickeln, sondern auf einer modularen Rechenplattform zu vereinen. Genau dieser Mischbetrieb deckt jedoch die architektonischen Reibungspunkte des Standard-Linux-Kernels auf: Während IoT-Dienste auf hohen Durchsatz und dynamische Ressourcenverwaltung optimiert sind, erfordert die Steuerungs-OT strikten Determinismus und minimale Latenzen.

Linux als COTS-Software-Plattform entwickelt seine Hardware-Unterstützung entlang technischer Anforderungen, verfügbarer Treiber und wirtschaftlich tragfähiger Beiträge aus der Hersteller- und Entwicklergemeinschaft. Zwar ermöglichen offene Quellen auch projektspezifische Anpassungen, deren langfristige Pflege und Integration verursachen jedoch zusätzlichen Aufwand.

Nicht zuletzt stehen Automatisierungsanwender einer „Materialschlacht“ kritisch gegenüber, da sie vom Einsatz von COTS-Software und -Hardware gerade wichtige Einsparungen erwarten. Damit spielt nun auch die Software-Architektur des Linux-Kernels eine wichtige Rolle, gerade wenn wie im Fall der Echtzeit-Automatisierung hohe Leistungsanforderungen gestellt werden, die sich zudem vom Gros gängiger Hochleistungsanwendungen in der allgemeinen IT unterscheiden.

Vor diesem Hintergrund illustriert dieser Beitrag, dass in einem Linux-System die Kommunikation (communication) eng mit dem sogenannten „compute“ oder „Rechenleistung“ verschränkt ist, so dass man durchaus auch von einem „com(m)pute“ sprechen kann. Die weitergehenden Verschränkungen des „com(m)pute“ beispielsweise mit Leistungseigenschaften der unterlagerten Hardware für die Speicher- und Netzwerkanbindung sind hingegen nicht mehr Gegenstand dieses Beitrags.

2.1 Linux-Kernel: Eine Geschichte zweier Architekturen

Automatisierungstechnische Applikationen wie PROFINET erfordern oftmals Zykluszeiten im und unter dem Millisekunden-Bereich. Dieser Beitrag befasst sich deshalb mit der Architektur der (harten) echtzeitfähigen Ausprägung des Linux-Kernels, „preempt_rt“ genannt. Davon zu unterscheiden ist beispielsweise die häufig auf Desktop-Systemen vorzufindende Ausprägung „preempt=voluntary“ als gängiger Ausgleich zwischen System-Durchsatz und -Latenz.

Leider wurde in der Vergangenheit in wissenschaftlichen Publikationen mit Bezug zur Echtzeit-Kommunikation/TSN mit Linux-Systemen einem durchaus wesentlichen Architektur-Unterschied kaum Beachtung geschenkt. Siehe hierzu exemplarisch [FAK22], die den für ihre Tests benutzen Kernel mit „*we changed the distro’s default (5.15) kernel with 5.19-rc6 kernel version*“ beschreiben. Andere, wie [LCZ22] geben überhaupt keine Kernel-Informationen an. Damit bleibt leider in solchen Fällen unklar, welche eine durchaus entscheidende Rolle spielende Kernel-Architektur zum Einsatz bei der wissenschaftlichen Untersuchung kam.

2.2 Preempt-Kernel

Ein häufig benutzter Kernel, weil insbesondere in den vergangenen Jahren leicht verfügbar, sind die Server- und Desktop-Ausprägungen preempt=none und preempt_voluntary. Diese verarbeiten Interrupts – wie beispielsweise von Netzwerkkarten – in zwei getrennten Stufen:

1. Interrupt-Handler (analog zur alten „top half“-Architektur früher Kernel), die alle andere Arbeit unterbrechen und deshalb nur sehr kurz laufen dürfen, so dass sie die eigentliche Arbeit, die empfangenen Pakete in Richtung Netzwerkstack und Applikationen zu „treiben“…

2. …in getrennten Kernel-Threads als sogenannte „soft IRQs“ unter der Kontrolle des Kernel-Schedulers ausführen (die „bottom halves“).

Diese für das „Treiben“ der Netzwerkpakete zuständigen Kernel-Threads gehören dabei fest zum sogenannten „ksoftirqd“, der jeweils pro *logischer* CPU als eigener Kernel-Thread auftritt, siehe auch die nachfolgende Abbildung. Vereinfacht wird hier nur die Empfangsrichtung betrachtet.

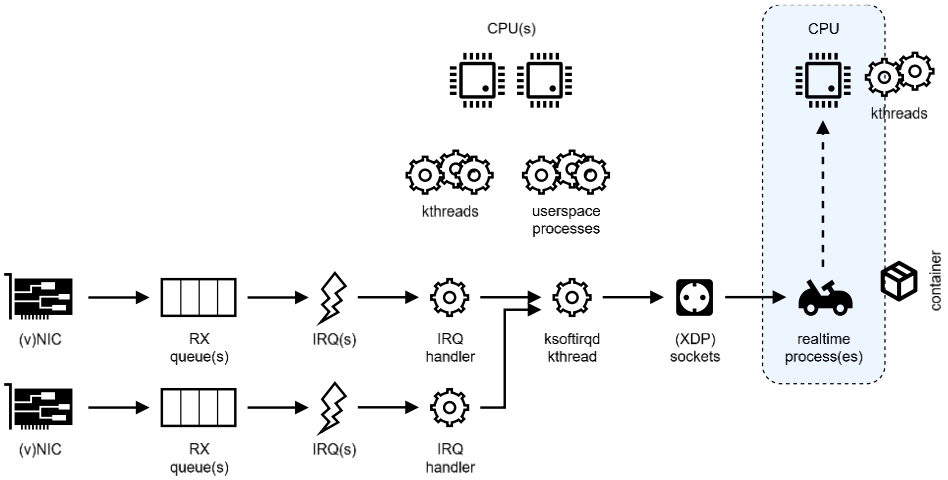


Abb. 4: Linux preempt=none/dynamic Kommunikations- und Ausführungsarchitektur.

Anmerkung: dieser Beitrag macht sich bewusst die Terminologie des Linux-Kernels zu Nutze, der zwischen „logischen“ CPUs und CPUs ohne Zusatz unterscheidet: eine „Zusatz-lose“ CPU gehört dabei in die topologische Hardware-Sicht, während eine logische CPU eine von der Hardware-Topologie losgelöste Abarbeitungseinheit für Programmcode ist.

Wie in der vorangehenden Abbildung illustriert, ist diese Infrastruktur beim Einsatz des sogenannten ‚eXpress Data Path‘ (XDP) im Standard-Kernel im Basisfall immer noch aktiv; die Infrastruktur des Kernels stellt hierbei sicher, dass Sicherheitsregeln eingehalten und kein fremder Speicher angesprochen werden kann. Erst durch die gezielte Kombination eines preempt_rt-Kernels mit dedizierten Treibern für AF_XDP im Zero-Copy-Modus lässt sich dieser Pfad für harte Echtzeitanwendungen vollständig von den üblichen Kernel-Bottlenecks entkoppeln.

Alle Soft-IRQs sind aus der Sicht des ksoftirqd in eine strikte Hierarchie von Soft-IRQ-Klassen eingepfercht, mit der Netzwerkkommunikation auf dem dritten und vierten Platz. Der ksoftirqd agiert dabei als Scheduler für die („best effort“) Abarbeitung von Interrupts pro logischer CPU.

Dabei entscheidet eine sogenannte „IRQ-Affinität“, welcher logischen CPU beispielsweise ein wichtiger Netzwerkkarten-IRQ zur Abarbeitung zugeordnet wird. Grundsätzlich kann diese IRQ-Affinität angepasst werden, jedoch ist das Latenzverhalten des ksoftirqd nur schwer zu kontrollieren, da er nicht nur alle zur jeweiligen logischen CPU affinen IRQs verarbeitet, sondern neben dem Netzwerkverkehr auch noch beispielsweise die Kommunikation mit Massenspeichern abwickelt. Zudem sollte der ksoftirqd nur im Nicht-Echtzeit-Betrieb laufen; Versuche, ihn mit Echtzeit-Priorität zu betrieben enden zumeist in unerwünschten Systemzuständen wie beispielsweise 95% Last der betroffenen logischen CPU, was dem Standard-Grenzwert für das Realtime-Throttling entspricht.

Aufgrund fehlender anderweitiger Hinweise in Forschungsveröffentlichungen muss leider davon ausgegangen werden, dass in vielen Fällen diese Standardkonfiguration zum Einsatz kam. Diese erweist sich bei entsprechend überdimensionierter Hardware und geringen Workloads als offensichtlich durchaus ausreichend. Jedoch spiegeln die in diesen Konfigurationen ermittelten Daten die industrielle Anwenderrealität allenfalls sehr verzerrt wider: bei steigender Last verhalten sich diese Systeme in der Regel stark nicht-linear und können – in Analogie zur Bruchmechanik von Stäben – zum plötzlichen „Einknicken“ neigen, bei denen eine kleine zusätzliche Lasterhöhung von wenigen Prozentpunkten bei einer logischen CPU zum Verlust der Applikation führen.

Ein besseres Verständnis dieser Systeme ist aber erforderlich, weil bei industriellen Anwendern und Anbietern im täglichen Wettbewerb die Hardware-Kosten – nicht zuletzt aufgrund des leicht überschwänglichen Hardware-Bedarfs bei KI-Anwendungen – eine wichtige Rolle spielen. Eine wesentliche Motivation für COTS-Plattformen ist ja gerade das Downscaling, ebenso wie das Upscaling, der erforderlichen Hardware-Ressourcen je nach spezifischem Kundenprojekt.

2.3 Preempt_rt-Kernel

Linux preempt_rt-Kernel besitzen eine abgeänderte Architektur, die den differenzierten Anforderungen der Echtzeit-Kommunikation besser Rechnung trägt. Ein in diesem Kontext wesentlicher Unterschied in der Kernel-Architektur ist der Ersatz von Interrupt-*Handlern* durch komplett eigenständige, vollwertige Kernel-*Threads*. Diese unterliegen damit dem Scheduling, inklusive der Echtzeit- wie „round robin“ und „first in, first out“ Scheduling.

Der zweite wesentliche Unterschied sind sogenannte „NAPI“ Kernel-Threads, siehe auch die nachfolgende Abbildung. Diese müssen erst explizit für eine Netzwerkschnittstelle aktiviert werden, danach werden die – vereinfacht gesprochen – einkommende Netzwerkpakete der betreffenden Schnittstelle nicht mehr im ksoftirqd verarbeitet, sondern in gesonderten NAPI-Kernel-Threads. Diese wiederum unterliegen dem System-Scheduling, so dass sie insbesondere im Zusammenspiel mit Automatisierungs-Prozessen/Threads entsprechend in Maßen anpassbar sind.

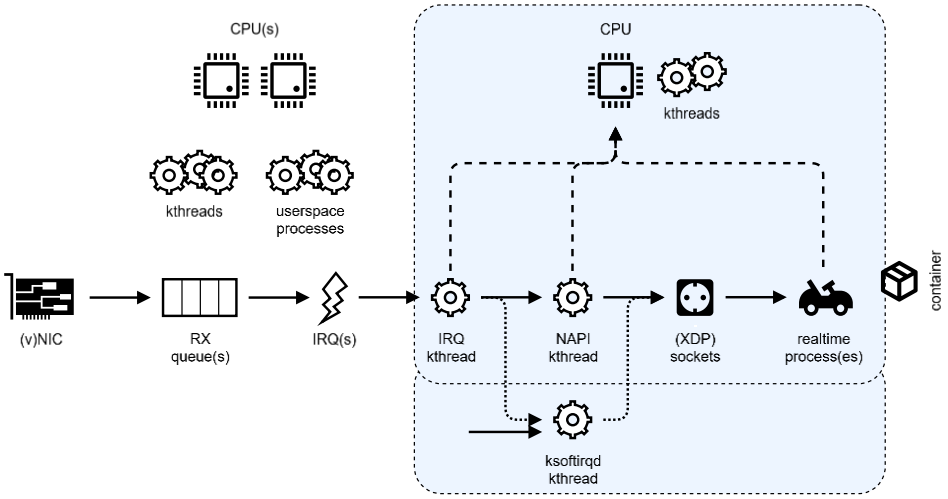


Abb. 5: Linux preempt_rt Kommunikations- und Ausführungsarchitektur.

Sowohl IRQ- als auch NAPI-Kernel-Threads laufen als Voreinstellung in der Scheduling-Strategie first-in-first-out (FIFO) in der Realtime-Priorität 50. Die Echtzeit-Prioritäten bei Linux liegen (aus der Sicht des Userspace) im Bereich 1-99, wobei 1 die niedrigste Echtzeit-Priorität und 99 die höchste darstellt. Die (Echtzeit-) Priorität 0 ist am POSIX API nicht erlaubt.

Letztlich erlauben (neuere) preempt_rt-Kernel in Verbindung mit NAPI-Kernel-Threads eine durchgängige Ressourcen-Reservierung auf dem Empfangspfad.

2.4 Wir sind nicht allein (auf einer logischen CPU).

Wie bereits angedeutet, betreibt der Linux-Kernel eine Reihe von Kernel-Threads für seine Aufgaben, selbst auf „isolierten“ logischen CPUs. Die Kernel-Threads lassen sich hierbei grob in zwei Gruppen einteilen:

1. CPU-spezifische Kernel-Threads, die auf genau einer CPU ausgeführt werden und von dieser nicht entfernt werden können (und dürfen).
2. „Springer“-Kernel-Threads, die je nach Lastsituation auf wechselnden logischen CPUs ausgeführt werden, hier ist noch einmal zu unterscheiden zwischen:
    a. Kernel-Threads, die sich von einer logischen CPU wegmigrieren lassen,
    b. sowie Kernel-Threads, die aus Kernel-internen Gründen nicht wegmigriert werden können.

Diese Situation lässt sich auch interaktiv mit dem OpenSource-Werkzeug „Linux Kernel Namespaces“ [LXKNS] eines der Autoren dieses Beitrags erkunden, wie in der nachstehenden Abbildung zu sehen. Dieses Tool kann ohne weiteres einfach direkt auf einem Linux-System „docker compose“ als Docker-Container gestartet werden.

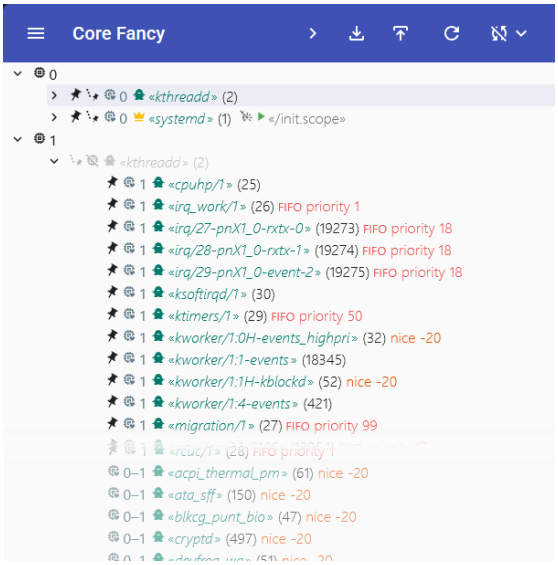


Abb. 6: Unvermeidliche Kernel-Threads auf „isolierten“ logischen CPUs.

Das Beispiel zeigt ein Linux-System mit preempt_rt-Kernel und zwei logischen CPUs, der üblichen Nummerierung folgend als 0 und 1 bezeichnet. Aufgeblendet sind hier Kernel-Threads auf der logischen CPU 1, wobei ein Steckstift symbolisiert, dass der betreffende Kernel-Thread rein nur auf dieser logischen CPU zum Ablauf kommt. Weitere Kernel-Threads in aufgehellter Darstellung zeigen hingegen, dass sie auf allen logischen CPUs 0-1 zum Ablauf kommen können, wobei diese Kernel-Threads ausnahmslos dem nicht-Echtzeit-Scheduling unterliegen und größtenteils ihre Priorität (nice −20) sogar noch unter die Standardpriorität erniedrigt haben.

2.5 Container-Kommunikation

Software-Container – und hier insbesondere OCI/Docker-Container – haben in den vergangenen Jahren das Erstellen, Paketieren und zur Ausführung bringen positiv verändert. Software-Container bilden heute in der IT wesentliche COTS-Software-Plattformen, nicht zuletzt in Form von Docker und Kubernetes. Sie auch im OT-Kontext erfolgreich einzusetzen, erfordert dabei, sich zumindest einiger wesentlicher treibender Architektur-Eigenschaft gerade auch in Bezug auf (Feld-) Kommunikation bewusst zu werden.

Nachdem die vorangegangenen Abschnitte das zu Verfügungstellung diverser Systemressourcen für eine zuverlässige Echtzeit-Kommunikation beleuchtet haben, verbleiben noch die Netzwerkschnittstellen eines Linux-Systems, über die Echtzeit-Kommunikation abgewickelt werden soll. In einer flexiblen Modularisierung von (auch gerade von „3rd Party“) Automatisierungs-Anwendungen spielen Container eine wichtige Rolle, um Integration zu beschleunigen und zu vereinfachen.

Für nicht-Echtzeit-Kommunikation stellt Docker heute jedem Container (es sei denn, ein Container wählt explizit ein Opt-Out) externe Kommunikation über die Host-Schnittstellen bereit. Für harte Echtzeit-Kommunikation ist es jedoch erforderlich, dem zeitkritischen Steuerungs-Task den direkten, latenzarmen Zugriff auf die Netzwerkschnittstelle zu gewähren, ohne dass dieser durch unkoordinierte Zugriffe anderer Container oder des Betriebssystems gestört wird.

Zunächst einmal kennt der Linux-Kernel keine Container, stattdessen stellt er grundlegende Mechanismen zur Ressourcen-Steuerung sowie -Separation; in Bezug auf Netzwerkkommunikation sind dieses die sogenannten „network namespaces“ (im Folgenden kurz als „netns“ bezeichnet). Entgegen ihrem Namen besitzen konkrete netns aber keine „Namen“, sondern nur transiente inode-Nummern. Die netns teilen den Netzwerk-Stack des Linux-Kernels im laufenden Betrieb in dynamisch einrichtbare und anpassbare „Abteilungen“.

Eine Netzwerkkarte kann hierbei allerdings immer nur genau einem netns zu einem Zeitpunkt zugehören. Physikalische Schnittstellen haben dabei die Besonderheit, dass beim Zerstören eines netns immer wieder in den initialen (oder „host“) netns zurückkehren und damit nicht verloren gehen können. Demensprechend verwaltet auch Docker, wie nachfolgend illustriert, die Netzwerkschnittstellen von Containern rein auf Ebene der Netzwerkschnittstellen; die Verwaltung einzelner Sende- und Empfangsqueues wird nicht unterstützt. Eine Übertragung der in der IEEE 60802 insbesondere in Figur 3 illustrierten End-Station-Architektur muss deshalb nur als komplette „end station“ erfolgen, sollen COTS-Plattformen wie Docker ohne aufwändige Modifikationen eingesetzt werden.

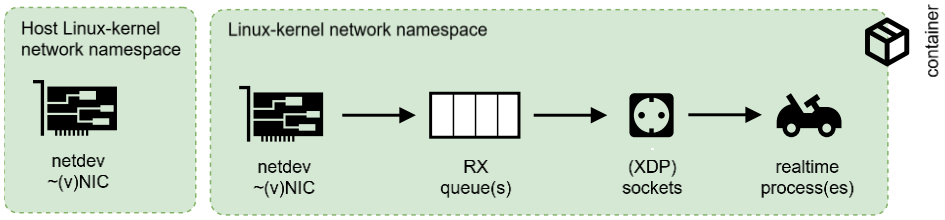


Abb. 7: Granularität der NIC-Verwaltung (nicht nur) bei Containern.

Damit das „Isolieren“ von Host-Netzwerk-Schnittstellen Docker-konform erfolgen kann, ist ein Docker-Netzwerk-Treiber erforderlich, da Docker selbst – anders als die Kubernetes CNI-Plugins – keinen sogenannten „passthrough“-Treiber anbietet. Jedoch lässt sich

ein derartiger 3rd-Party Docker-Treiber problemlos nachrüsten, wie beispielsweise bei der Siemens Industrial Edge-Plattform der Fall ist.

Die architektonische Besonderheit des hier vorgestellten Ansatzes liegt in der echten Konvergenz auf einer einzigen physikalischen Netzwerkschnittstelle (netdev), ohne hierfür auf Kernel-Bypass-Monolithen wie DPDK zurückgreifen zu müssen:

- Isochrone Echtzeit (rote Phase): Für den zyklischen PROFINET-IRT-Verkehr nutzt die Controller-Applikation AF_XDP, idealerweise im Zero-Copy-Modus. Dies reduziert Kopieroperationen und ermöglicht einen verkürzten Datenpfad zwischen NIC-Treiber und Userspace. Die Verarbeitung bleibt jedoch in die NAPI- und Treiberarchitektur des Linux-Kernels eingebunden. Durch preempt_rt, threaded NAPI, geeignete Queue-Zuordnung und CPU-Affinitäten kann dieser Pfad kontrollierter gegenüber konkurrierenden Workloads isoliert werden.
- IT- und IoT-Kommunikation (grüne Phase): Parallel dazu bleibt der reguläre Linux-Netzwerk- und Socket-Stack auf demselben netdev uneingeschränkt funktionsfähig. Nicht-zeitkritische Dienste – wie OPC UA, MQTT-Broker für Cloud-Anbindungen, Diagnose-Webserver oder Standard-TCP/IP-Traffic aus parallelen Docker-Containern – nutzen dieselbe physikalische Leitung über Standard-Linux-Sockets.

Auf der Netzwerkkarten-Hardware wird dieser Mischverkehr so abgebildet, dass der IT- und IoT-Datenverkehr der grünen Phase der höher priorisierten roten Echtzeitphase nachgeordnet wird. Damit bildet dieser COTS-Ansatz die fundamentale Eigenschaft von PROFINET IRT – die Koexistenz von harter Echtzeit und offenem Ethernet auf einem Kabel – mit reinen Linux-Standardmechanismen vollständig ab.

Für das Containermodell löst dieser Ansatz den Zielkonflikt zwischen Isolation und Schnittstellenteilung elegant:

- Der PROFINET-IRT-Steuerungs-Container erhält über gezielte Linux-Capabilities (CAP_NET_RAW, CAP_NET_ADMIN) direkten Zugriff auf die Hardware-Queues des physikalischen netdevs via AF_XDP. Er wickelt die rote Phase im Zero-Copy-Verfahren ab, ohne dass die gesamte Netzwerkkarte exklusiv in einen abgeschotteten Network Namespace verschoben werden muss.
- Parallel dazu verbleibt das netdev als IP-Schnittstelle im primären Netzwerkstack. Weitere Container (wie Edge-Analytics, Vision AI oder IoT-Gateways) greifen transparent über Standard-Docker-Netzwerke (Bridge oder Macvlan) auf denselben Port zu und wickeln ihren Best-Effort-Verkehr in der grünen Phase ab.

Auf diese Weise ermöglicht die Kombination aus XDP und preempt_rt eine echte Workload-Konsolidierung auf einem einzigen COTS-netdev, ohne das bewährte IT-Managementmodell von Docker aufbrechen zu müssen.

3 Zusammenfassung

Mit „Commercial off-the-shelf“ (COTS) Netzwerkkarten wie beispielsweise der Intel I210 oder I226 mit TSN-Elementen der IEEE 60802 und IEEE 802.1AS Zeitsynchronisation ist es mit höheren Link-Speeds ab 1 Gbit/s möglich, eigene PN-Controller umzusetzen, die flexibel bei den CPU- und Netzwerkleistungen skalieren und auch IRT unterstützen.

Dank der höheren Bandbreite im Netzwerk können nun auch neue Applikationen betrieben werden, wie beispielsweise 4K-Kameras für Vision-AI-Anwendungen wie Qualitätskontrolle. Der Rechenbedarf der verschiedenen modularen Anwendungen im gleichen System lässt sich nun bequemer über entsprechend in Leistung und/oder Anzahl dimensionierte CPUs aus Standardangeboten decken. Beispielsweise für Maschinenbauer bedeutet dies eine größere Flexibilität im Auf- und Ausbau der PN-Controller und den weiteren, Maschinen-spezifischen Anwendungen auf der gleichen Plattform.

COTS-Software-Plattformen wie preempt_rt-Linux und Docker bieten zusätzliche Flexibilität in der Modularisierung und Skalierung sowohl bei IT-, also auch OT-Anwendungen und deren bessere Integration. Hier sind jedoch bestimmte Architektur-Regeln der IT zu berücksichtigen, um die Vorteile dieser IT-Architekturen auf OT-Anwendungen erfolgreich zu übertragen.

Literaturverzeichnis

[ALB23] Albrecht, H.: Edgeshark: Einblicke in die virtuelle Kommunikationswelt (nicht nur) von Containern – ein (OpenSource) Beitrag zu einer konvergierenden IT/OT. In: Kommunikation in der Automation, 14. Jahreskolloquium, Magdeburg 2023. http://dx.doi.org/10.25673/111642

[CNI] The Container Network Interface. https://www.cni.dev/

[DPDK] Data Plane Development Kit. https://www.dpdk.org/.

[FAK22] Ferenc, F.; Antal, P.; Kerekes, M.: The TSN Building Blocks in Linux. netdev 0x16, Lissabon, 2022. https://netdevconf.info/0x16/papers/16/netdev0x16_fejes_final.pdf

[LCZ22] Lázaro, J.; Cabrejas, J.; Zuloaga, A.; Muguira, L.; Jiménez, J.: Time Sensitive Networking Protocol Implementation for Linux End Equipment. Technologies, 2022. https://www.researchgate.net/publication/360160668_Time_Sensitive_Networking_Protocol_Implementation_for_Linux_End_Equipment/fulltext/62f997f9c6f6732999cf8e00/Time-Sensitive-Networking-Protocol-Implementation-for-Linux-End-Equipment.pdf

[LXKNS] https://github.com/thediveo/lxkns

[IEC60802] https://webstore.iec.ch/en/publication/60188
IEC/IEEE 60802:2026: Time-sensitive networking profile for industrial automation

[PI-PN] https://www.profibus.de/downloads/profinet-specification/
PROFIBUS Nutzerorganisation e.V. (PNO): PROFINET Specification – Architecture, Protocols and Services, IEC 61158-6-10 / IEC 61784-2.

AI-Security Monitoring for Industrial Control Systems

Felix Specht [1], David Hülsmeier-Reineke [2] und Henning Trsek [3]

Abstract: Industrielle Steuerungs- und Automatisierungssysteme (ICS) sind zunehmend Ziel von Cyberangriffen, während die regulatorischen Anforderungen an die Cybersicherheit durch NIS-2 und den EU Cyber Resilience Act steigen. Security Monitoring unterstützt die Erfüllung dieser Anforderungen, ist in industriellen Netzen jedoch aufwendig umzusetzen: Die Kommunikation ist heterogen, die Datenmengen sind groß und qualifiziertes Personal für die Bewertung von Meldungen fehlt. Dieser Beitrag stellt das Konzept des *AI-Security Monitorings* vor, das Sicherheitsfunktionen wie Network Monitoring, Vulnerability Management, Asset Management, Compliance Checking und Dokumentation unter Einsatz von großen Sprachmodellen (LLM) vereint. Zwei Anwendungsfälle werden vertieft. Für das Network Monitoring wird eine sechsstufige Verarbeitungskette beschrieben. Die ersten fünf Stufen führen vom Netzwerkverkehr zum Alarm und klassifizieren ihn mit einem überwacht trainierten Decision Tree. Die sechste Stufe untersucht denselben Alarm nachgelagert mit einem werkzeuggestützten LLM-Agenten. Beide Ansätze werden auf demselben Testbed gegenübergestellt. Beide Ansätze bestimmen die Angriffsmethode. Der Entscheidungsbaum ordnet Alarme über sieben Methoden in 88 % der Fälle korrekt zu, der LLM-Agent über neun Methoden in 89 % der Fälle. Der Agent benötigt dafür keine Trainingsdaten. Für das Vulnerability Management wird eine Verarbeitungskette von der Identifikation der Netzwerkteilnehmer bis zur Handlungsempfehlung beschrieben.

Keywords: AI-Security Monitoring, LLM, Network Monitoring, Vulnerability Management, ICS

1 Einleitung

Industrielle Steuerungs- und Automatisierungssysteme (ICS) bestehen aus spezialisierten Komponenten der Automatisierungstechnik, die über industrielle Netzwerkprotokolle wie Profinet, OPC UA oder Modbus/TCP kommunizieren [WSJ17]. Diese Komponenten wurden häufig nicht nach dem Secure-by-Design-Prinzip entwickelt und bieten unzureichenden Schutz gegen Cyberangriffe [SOE22]. Durch die steigende Vernetzung der Anlagen sind sie zunehmend exponiert. Gleichzeitig steigen die regulatorischen Anforderungen: Die NIS-2-Richtlinie[4], der Cyber Resilience Act (CRA)[5] und die Normenreihe IEC 62443[6] fordern

[1] Fraunhofer IOSB-INA, Digitale Infrastruktur, Campusallee 1, 32657 Lemgo, Deutschland, felix.specht@iosb-ina.fraunhofer.de, https://orcid.org/0009-0003-9236-2534

[2] Hilscher Gesellschaft für Systemautomation mbH, Rheinstraße 15, 65795 Hattersheim am Main, Deutschland, DHuelsmeier@hilscher.com, https://orcid.org/0000-0002-4644-0592

[3] inIT - Institute industrial IT, Technische Hochschule Ostwestfalen-Lippe, Campusallee 6, 32657 Lemgo, Deutschland, henning.trsek@th-owl.de, https://orcid.org/0000-0002-0133-0656

[4] Richtlinie (EU) 2022/2555 über Maßnahmen für ein hohes gemeinsames Cybersicherheitsniveau in der Union (NIS-2), Amtsblatt der Europäischen Union, 2022.

[5] Verordnung (EU) 2024/2847 über horizontale Cybersicherheitsanforderungen für Produkte mit digitalen Elementen (CRA), Amtsblatt der Europäischen Union, 2024.

[6] IEC 62443: Security for industrial automation and control systems, International Electrotechnical Commission (IEC), 2018.

umfassende Security-Maßnahmen. Security Monitoring ist dabei ein zentrales Werkzeug für Anlagenbetreiber[7]. Es verschafft ihnen einen Überblick über die Geräte in ihren Netzen, deren Kommunikationsverhalten und die darin enthaltenen Schwachstellen. Das so enstandene Lagebild ist die Grundlage, um Angriffe effektiv zu erkennen, verwundbare Geräte zu aktualisieren und den gesetzlich vorgeschriebenen Meldepflichten nachzukommen.

Die Umsetzung des Security Monitorings in industriellen Netzen ist jedoch aufwendig. Proprietäre Protokolle und Legacy-Systeme erschweren die Überwachung, und Echtzeitanforderungen begrenzen den Einsatz klassischer IT-Security-Werkzeuge. Die Kommunikation erfolgt kontinuierlich und erzeugt große Datenmengen, in denen einzelne Auffälligkeiten untergehen. Ein einzelner Cyberangriff erzeugt zudem eine Vielzahl von Einzelbefunden über mehrere Kommunikationsbeziehungen hinweg. Die Bewertung dieser Meldungen erfordert Security-Expertise, die in produzierenden Unternehmen häufig nicht verfügbar ist.

In diesem Beitrag stellen wir das Konzept des *AI-Security Monitorings* vor, das Sicherheitsfunktionen unter Einsatz von großen Sprachmodellen (LLM) vereint. Je nach Anwendungsfall greifen die Sprachmodelle über Werkzeuge auf Alarm-, Zeitreihen- und Paketdaten zu oder erschließen Textquellen wie Schwachstellenbeschreibungen mittels Retrieval Augmented Generation (RAG) [Le20]. Große Sprachmodelle beziehen domänenspezifische Datenquellen ein und sind für die Analyse sicherheitsrelevanter Informationen in natürlicher Sprache geeignet [Fe24]. Zwei Anwendungsfälle werden vertieft: Network Monitoring zur Erkennung und Bewertung von Cyberangriffen sowie Vulnerability Management zur Identifizierung und Behebung von Schwachstellen. Der Beitrag greift dabei auf bereits veröffentlichte Ergebnisse zurück und ordnet diese in ein gemeinsames Konzept ein.

Kapitel 2 gibt einen Überblick über verwandte Arbeiten. Kapitel 3 beschreibt die Gesamtarchitektur und die fünf Anwendungsfälle. Die Kapitel 4 und 5 stellen die beiden vertieften Anwendungsfälle vor. Kapitel 6 fasst die Ergebnisse zusammen.

2 Stand der Technik

Die Anomalieerkennung in industriellen Netzen nutzt die Eigenschaft, dass industrielle Kommunikation überwiegend zyklisch und über lange Zeiträume stabil ist. Verfahren auf Basis von Netzwerkmetriken wandeln den Netzwerkverkehr in Zeitreihen um und erkennen Abweichungen gegen ein gelerntes Normalverhalten [SO24]. Ein vergleichbarer Ansatz überträgt dieses Prinzip auf Logdaten cyber-physischer Produktionssysteme [OS25]. Diese Verfahren erkennen Abweichungen zuverlässig, benennen jedoch nicht die zugrunde liegende Angriffstechnik.

Für die Benennung der Angriffstechnik wird überwachtes maschinelles Lernen eingesetzt, das eine Genauigkeit von über 95 % erreicht [SO24]. Überwachte Verfahren benötigen

[7] Bundesamt für Sicherheit in der Informationstechnik: Monitoring und Anomalieerkennung in Produktionsnetzwerken, BSI-CS 134, 2019.

gelabelte Trainingsdaten für jede Technik, die sie erkennen sollen. Das MITRE-ATT&CK-Framework für industrielle Steuerungssysteme [ABS20] definiert in der aktuellen Fassung 83 Techniken[8]. Nach Kenntnis der Autoren stellt kein öffentlicher Datensatz gelabelte Beispiele in nennenswertem Umfang bereit. Beim Einsatz in einem anderen Netz ist zudem ein Neutraining erforderlich, um an geänderte Kommunikationsmuster, Protokolle und Topologien angepasst zu werden.

Große Sprachmodelle sind eine mögliche Alternative, da sie über Wissen zu Netzwerkprotokollen, Angriffsmustern und Angriffstaxonomien verfügen [Fe24]. Ohne Werkzeugzugriff ist ihre Genauigkeit jedoch gering: Antari et al. erreichen 31 bis 41 % bei der Klassifikation von Netzwerkverkehr [An25], Daniel et al. einen F1-Wert von 0,62 bei der Zuordnung zu MITRE-ATT&CK-Techniken [Da25]. Ann et al. erreichen durch Feinabstimmung mit synthetischen Daten 96,49 % für sechs Techniken, ohne Feinabstimmung dagegen 14 % [An26]. LLM-Agenten erweitern Sprachmodelle um Werkzeuge, deren Ergebnisse iterativ ausgewertet werden [Ya23]. IDS-Agent koordiniert sechs vortrainierte Klassifikatoren und erreicht einen F1-Wert von 0,975, wobei das Sprachmodell keine Rohdaten des Netzwerkverkehrs betrachtet [Li24]. Revelation nutzt einen LLM-Agenten zur Interpretation des Netzwerkverkehrs, benötigt für die Erkennung jedoch ein feinabgestimmtes Modell [Wo25].

Im Bereich des Vulnerability Managements ordnen Verfahren des maschinellen Lernens Schwachstelleneinträgen automatisiert Produktbezeichner zu [WH20]. Domänenspezifische Sprachmodelle wie SecureBERT [Ag22] bilden Schwachstellenbeschreibungen auf Vektorrepräsentationen ab und ermöglichen den Vergleich inhaltlich ähnlicher Einträge.

Im Gegensatz zu bestehenden Ansätzen verbindet der vorgestellte Ansatz die Erkennung, die Klassifikation und die nachgelagerte Bewertung von Cyberangriffen in einer durchgängigen Verarbeitungskette. Die Erkennung benötigt keine gelabelten Angriffsdaten, und die forensische Analyse ist nicht auf die Techniken beschränkt, die in Trainingsdaten enthalten sind.

3 AI-Security Monitoring

AI-Security Monitoring wird als Oberbegriff für unterschiedliche Sicherheitsfunktionen verwendet, die unter Einsatz von LLM realisiert werden. Abbildung 1 zeigt die Gesamtarchitektur. Zentral ist die AI-Security Monitoring Plattform, die über einen lokalen Agenten auf einem Industrie-PC (IPC) auf Daten des Produktionsnetzes zugreift. Ein LLM-Server stellt das Sprachmodell zur Verfügung, welches die eigentliche Verarbeitung von Anfragen des Agenten durchführt Der LLM-Server kann lokal oder in einer Cloud betrieben werden.

Die Plattform umfasst fünf Anwendungsfälle: **Network Monitoring** zur Erkennung von Cyberangriffen, **Vulnerability Management** zur Identifizierung von Schwachstellen, **Asset Management** zur automatisierten Inventarisierung aller Netzwerkkomponenten und

[8] MITRE: ATT&CK for ICS, https://attack.mitre.org/matrices/ics/, abgerufen am 26.08.2026.

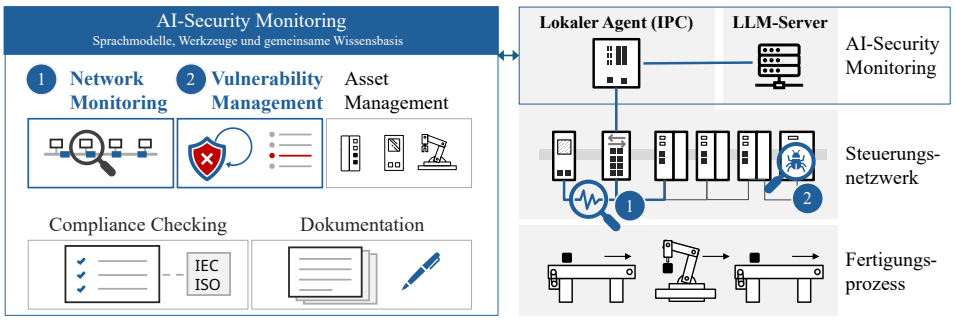


Abb. 1: Gesamtarchitektur des AI-Security Monitorings mit den Anwendungsfällen (1) Network Monitoring und (2) Vulnerability Management, die in diesem Beitrag näher vorgestellt werden, sowie den Bereichen Asset Management, Compliance Checking und Dokumentation.

Firmware-Stände, **Compliance Checking** zur automatisierten Prüfung gegen Normen wie IEC 62443 sowie **Dokumentation** zur LLM-gestützten Erstellung von Audit-Berichten und Security-Reports. Die beiden erstgenannten Anwendungsfälle werden in den Kapiteln 4 und 5 vertieft. Eine tiefere Betrachtung der übrigen Bereiche erfolgt in zukünftigen Veröffentlichungen.

Die fünf Anwendungsfälle greifen auf eine gemeinsame Datenbasis zu. Das Asset Management liefert die im Netzwerk vorhandenen Komponenten mit Hersteller, Produkt und Version und ist damit die Grundlage für das Vulnerability Management. Erkennt das Network Monitoring einen Cyberangriff auf eine Komponente, für die eine offene Schwachstelle vorliegt, erhöht dies die Dringlichkeit der Meldung. Der Einsatz von Sprachmodellen verfolgt in allen Anwendungsfällen dasselbe Ziel: Sicherheitsrelevante Informationen liegen überwiegend als unstrukturierter Text vor, etwa in Schwachstellenbeschreibungen oder Security Advisories. Große Sprachmodelle können diese Quellen automatisiert auswerten und unterstützen damit Personal ohne spezielle Security-Expertise bei der Bewertung sicherheitsrelevanter Ereignisse. Liegen Ausgangsdaten hingegen strukturiert oder in großen Mengen vor, etwa beim Netzwerkverkehr oder in Datenbanken, greifen die Modelle mithilfe von Werkzeugen darauf zu.

4 Anwendungsfall 1: Network Monitoring

Der Anwendungsfall Network Monitoring umfasst (1) eine Verarbeitungskette vom Netzwerkverkehr bis zum Alarm, (2) eine nachgelagerte forensische Analyse der Alarme durch einen LLM-Agenten und (3) einen Vergleich der Alarmklassifikation mittels klassischem Machine Learning und mittels LLM-Agenten.

4.1 Von der Netzwerkaufzeichnung zum Alarm

Die Verarbeitungskette umfasst sechs Stufen. Die ersten fünf Stufen führen vom Netzwerkverkehr zum Alarm, die sechste bewertet den Alarm nachgelagert. Abbildung 2 zeigt die Funktionalität jeder Stufe sowie die zugehörige technische Realisierung. Zentrales Entwurfsmerkmal ist die Trennung von Detektion und Klassifikation: Die Detektion bestimmt, ob ein Vorgang anomal ist, während die Klassifikation ermittelt, um welche Art von Anomalie es sich handelt. Das Modell des maschinellen Lernens wird nur aufgerufen, wenn bereits eine Anomalie vorliegt. Die Detektion benötigt damit keine gelabelten Daten, die in industriellen Umgebungen praktisch nicht verfügbar sind.

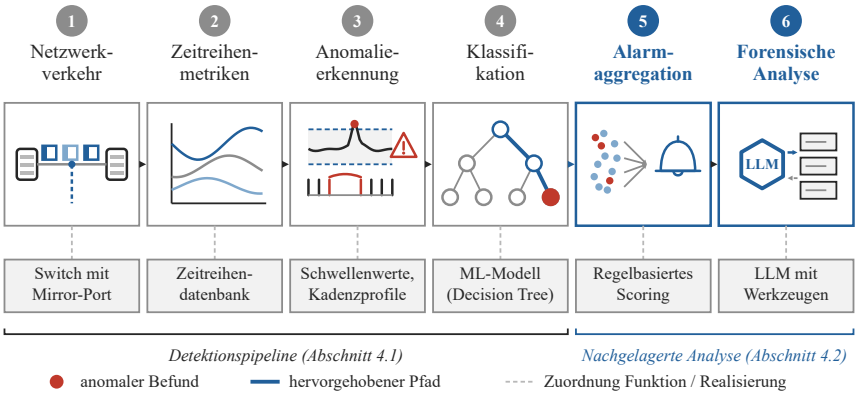


Abb. 2: Verarbeitungskette vom Netzwerkverkehr bis zum Alarm mit nachgelagerter forensischer Analyse. Die obere Reihe zeigt die Funktionalität, die untere Reihe die technische Realisierung.

Stufe 1 – Erfassung. Der Netzwerkverkehr wird über einen Mirror-Port eines industriellen Switches oder ein Netzwerk-TAP abgegriffen. Die Erfassung ist rein passiv: Es werden keine Pakete gesendet oder verändert, sodass die Überwachung die Echtzeitkommunikation der Anlage prinzipbedingt nicht stören kann.

Stufe 2 – Zeitreihenmetriken. Die rohen Netzwerkpakete werden Kommunikationsbeziehungen zugeordnet und zu Zeitfenstern fester Länge aggregiert. Eine Kommunikationsbeziehung wird über einen neunfeldrigen Schlüssel aus Quell- und Ziel-MAC-Adresse, VLAN, IP-Version, Quell- und Ziel-IP-Adresse, Quell- und Ziel-Port sowie Protokollstapel identifiziert. Je Beziehung und Zeitfenster werden zehn Metriken berechnet: sechs Volumenmetriken m_1 bis m_6 zu Paketanzahl, Datenvolumen und Paketgröße sowie vier Timingmetriken m_7 bis m_{10} zur Zwischenankunftszeit. Da industrielle Kommunikation überwiegend zyklisch ist, sind die Timingmetriken hier aussagekräftiger als in IT-Netzen. Die Metriken werden parallel auf drei Sichten desselben Verkehrs berechnet: je Verbindung, je Gerät und je Protokoll. Ein Portscan bleibt auf Verbindungsebene unauffällig, wird auf Geräteebene aber deutlich sichtbar.

Stufe 3 – Anomalieerkennung. Die Anomalieerkennung umfasst zwei sich ergänzende Detektionspfade. Der erste Pfad prüft die Metrikwerte gegen Schwellenwerte, die in einer Lernphase je Metrik und Kommunikationsbeziehung als quantilbasierte Hüllkurve auf einer angriffsfreien Referenzaufzeichnung berechnet werden:

$$b_{\text{unten}} = Q(q) \cdot (1 - v), \qquad b_{\text{oben}} = Q(1 - q) \cdot (1 + v) \tag{1}$$

Dabei bezeichnet $Q(q)$ das q-Quantil der Metrikwerte im Trainingszeitraum und v einen Variationszuschlag. Quantile statt Minimum und Maximum machen das Verfahren robust: Ein einzelner Ausreißer würde die Hüllkurve sonst so weit aufziehen, dass echte Angriffe in den Erwartungsbereich fallen. Ein Zeitfenster gilt als anomal, sobald eine Metrik ihre Schwellenwerte verletzt. Beziehungen ohne gelernte Hüllkurve gelten als anomal, da die Topologie industrieller Netze über lange Zeiträume stabil ist. Der zweite Pfad prüft die zeitliche Kadenz der Datenankunft. Er erkennt Vorgänge, die eine reine Wertprüfung verfehlt: Ein Adversary-in-the-Middle verschiebt die Zykluszeit, ohne die Paketwerte zu verändern, ein durch Denial of Service ausgefallenes Gerät erzeugt gar keine Werte mehr. Die Detektion erzeugt damit vier Anomalietypen: Schwellenwertverletzung und unbekannte Beziehung aus dem ersten Pfad sowie Kadenzabweichung und Ausbleiben aus dem zweiten Pfad. Der jeweilige Typ wird als Basis-Label der Anomalie zugeordnet.

Stufe 4 – Klassifikation. Zielgröße der Klassifikation ist die Angriffsmethode, also das konkrete Vorgehen. Mehrere Methoden können derselben Technik des MITRE-ATT&CK-Frameworks für industrielle Steuerungssysteme entsprechen [ABS20]; diese Zuordnung nimmt erst der Agent in Stufe 6 vor. Ein Decision Tree bildet die Metrikvektoren einer erkannten Anomalie unmittelbar auf eine Angriffsmethode ab. Liegt ein Ergebnis oberhalb der Konfidenzschwelle vor, wird die benannte Methode als weiteres Label der Anomalie zugeordnet. Andernfalls bleibt nur der Basistyp aus Stufe 3 erhalten. Ohne belastbare Aussage des Modells wird der Vorgang damit nicht spekulativ zugeordnet.

Stufe 5 – Alarmaggregation. Ein Cyberangriff erzeugt dutzende bis hunderte Anomalie-Datenpunkte, die ohne Verdichtung nicht handhabbar wären. Anomalien, deren zeitlicher Abstand eine Lückentoleranz nicht überschreitet, werden demselben Alarm zugeordnet. Die Toleranz überbrückt kurze Pausen innerhalb eines Angriffs, verschmilzt getrennte Episoden aber nicht. Bei der Aggregation werden die Labels der Anomalien zusammengeführt. Dies kann jedoch zu Konflikten führen, wenn unterschiedliche Anomalien widersprüchliche Labels haben. Welche davon den gesamten Alarm benennt, entscheidet eine Bewertung je Label. Sie summiert drei Evidenzgrößen: Ereignisrate (Anomalien je Sekunde), Merkmalsabdeckung (Anteil der zehn Metriken) und Persistenz (Zahl der Zeitfenster). Drei Faktoren dämpfen das Ergebnis: mittlere Konfidenz (Sicherheit der Einzelbefunde), Protokollabschlag (generische Stacks wie ARP) und Beobachtungsebene (Verbindung, Gerät oder Protokoll). Ein häufiges, aber unsicheres Label setzt sich damit nicht gegen ein selteneres, belastbares durch. Alle drei Evidenzgrößen sind nach oben gedeckelt. Ein Label mit 500 Zeitfenstern zählt damit nicht mehr als eines mit zehn. Ohne diese Grenze würde ein dauerhaft auffälliger Befund jedes seltenere Label verdrängen.

4.2 Stufe 6 – Agentenbasierte forensische Analyse

Das Label eines Alarms ist nur so gut wie das Machine Learning-Modell, das es vergeben hat. Das Modell ordnet jede Anomalie einer der sieben Angriffsmethoden seines Trainingsdatensatzes zu. Neue oder abweichende Klassen kann es nicht als solche erkennen. Jeder Alarm wird daher anschließend einer LLM-gestützten forensischen Analyse unterzogen [ST26], die das Ergebnis der Aggregation überprüft und die Verarbeitungskette um eine unabhängige Zweitmeinung ergänzt. Der Agent benötigt dafür keine gelabelten Trainingsdaten.

Der Agent erhält eine Alarmkennung und sammelt in einer iterativen Werkzeugschleife Evidenz. In jedem Durchlauf ruft das Sprachmodell entweder Werkzeuge auf oder führt die abschließende Klassifikation durch. Der Werkzeugsatz umfasst 19 lesende Werkzeuge für Alarmdaten, Zeitreihen, Paketaufzeichnung und die MITRE-ATT&CK-ICS-Wissensbasis. Die Vorhersage des ML-Modells aus Stufe 4 wird zudem aus jeder Werkzeugantwort entfernt, bevor sie das Sprachmodell erreicht, um eine methodische Unabhängigkeit des LLM zu gewährleisten. Die Untersuchung folgt dem Vorgehen eines IT-Forensikers: Alarmdaten lesen, Metriken abfragen, Paketaufzeichnungen untersuchen, mit dem Normalbetrieb vergleichen und die Aktivität einer Technik zuordnen.

Die Klassifikation erfolgt zweistufig: Der Agent bestimmt zunächst die Angriffsmethode (z.B. Port Scan, TCP-Flood oder ARP-Cache-Poisoning) und belegt sie aus der Untersuchung, dann ordnet er sie einer MITRE-ATT&CK-ICS-Technik mit Konfidenzwert zu. Der erste Schritt ist dabei die maßgebliche Fehlerquelle, der zweite gelingt zuverlässig, sobald der erste korrekt ist [ST26].

Die Auswertung auf einem industriellen Testaufbau umfasst 450 Alarme aus neun Angriffsmethoden und sechs MITRE-ATT&CK-ICS-Techniken [ST26]. Das Cloud-Modell GPT-5.4 bestimmt die Angriffsmethode in 89 % der Fälle korrekt, das lokal betriebene Nemotron-3-Super mit 120 Milliarden Parametern in 77 %. Die Analyse benötigt 137 bis 467 Sekunden je Alarm und eignet sich damit für die nachgelagerte Bewertung, nicht für die Klassifikation in Echtzeit. Die Analyseergebnisse können anschließend als Grundlage für die Vorfallsreaktion dienen, bei der LLM-Agenten ebenfalls das menschliche Personal unterstützen können. Die Vorfallsreaktion selbst ist jedoch nicht Gegenstand dieses Beitrags.

4.3 Vergleich der Klassifikationsansätze

Für jeden Alarm liegen zwei unabhängige Klassifikationen vor: durch das Decision-Tree-Modell und den LLM-Agenten. Tabelle 1 stellt beide Ansätze gegenüber. Die Referenzwerte des Decision Trees stammen aus einem Industrieprojekt mit derselben Verarbeitungskette und demselben Testbed [SD26].

Beide Ansätze nutzen dasselbe Testbed und dieselbe Verarbeitungskette; in beiden Fällen bildet der aggregierte Alarm die Grundlage der Bewertung. Die zugrunde liegenden

Tab. 1: Gegenüberstellung der Klassifikationsansätze auf dem Testbed des Fraunhofer IOSB-INA. Beide Ansätze bewerten aggregierte Alarme derselben Verarbeitungskette. Beide Ansätze bestimmen die Angriffsmethode; die Klammer nennt die Anzahl der unterschiedenen Methoden. Die Sprachmodelle werden auf einem Datensatz aus 450 Alarmen ausgewertet. Die Nemotron-Modelle werden lokal betrieben. Bei den kleineren 30B und 4B Modellen sinkt die Genauigkeit auf bis zu 20 %. Die Modellgröße begrenzt damit die lokale Ausführung.

Verfahren	**Zielgröße**	**Genauigkeit**	**Trainingsdaten**	**Quelle**
Decision Tree	Methode (7)	88 %	erforderlich	[SD26]
LLM-Agent, GPT-5.4 (Cloud)	Methode (9)	89 %	-	[ST26]
LLM-Agent, Nemotron-3-Super	Methode (9)	77 %	-	[ST26]
LLM-Agent, Nemotron-3-Nano 30B	Methode (9)	40 %	-	[ST26]
LLM-Agent, Nemotron-3-Nano 4B	Methode (9)	20 %	-	[ST26]

Methodenmengen sind jedoch nicht deckungsgleich: Der Decision Tree unterscheidet sieben Methoden, der Agent neun. Die Werte sind daher als Einordnung zu lesen, nicht als direkter Wettbewerb. Beide Ansätze sind komplementär und werden parallel betrieben: Das überwachte Verfahren klassifiziert im laufenden Betrieb in Millisekunden, der Agent bewertet nachgelagert und begründet seine Einschätzung mit Belegen aus dem Rohverkehr. Da beide Aussagen unabhängig entstehen, ist ihre Übereinstimmung ein Hinweis auf die Verlässlichkeit der Klassifikation.

5 Anwendungsfall 2: Vulnerability Management

Vulnerability Management ergänzt das in Kapitel 4 beschriebene Network Monitoring. Es bewertet den bekannten Sicherheitszustand von Netzwerkteilnehmern und führt von der Komponentenidentifikation zur Handlungsempfehlung. Die Betrachtung reicht von Komponenten über Maschinen bis zu Anlagen. Abbildung 3 zeigt die Verarbeitungskette mit Beispielen-Ausschnitten.

Identifikation der Netzwerkteilnehmer: Stouffer et al. [St23] ordnet Asset Inventory und Configuration Management als zentrale OT-Sicherheitsmaßnahmen ein. Die Asset Discovery kombiniert DCP für PROFINET-Geräte,[9] netFIELD,[10] GSDML und nmap.[11] So lassen sich auch unvollständige Gerätedaten erfassen.

Hersteller, Produkt sowie Firmware- oder Softwareversion werden extrahiert und Komponenten Maschinen und Anlagen zugeordnet. Das hierarchische Modell folgt IEC 62443-1-1;

[9] PROFIBUS & PROFINET International: PROFINET System Description, 2018.
[10] Hilscher: netFIELD-Dokumentation, https://hilscher.atlassian.net/
[11] Lyon, Gordon: Network Discovery, https://nmap.org/book/discovery.html, abgerufen 2026-09-17.

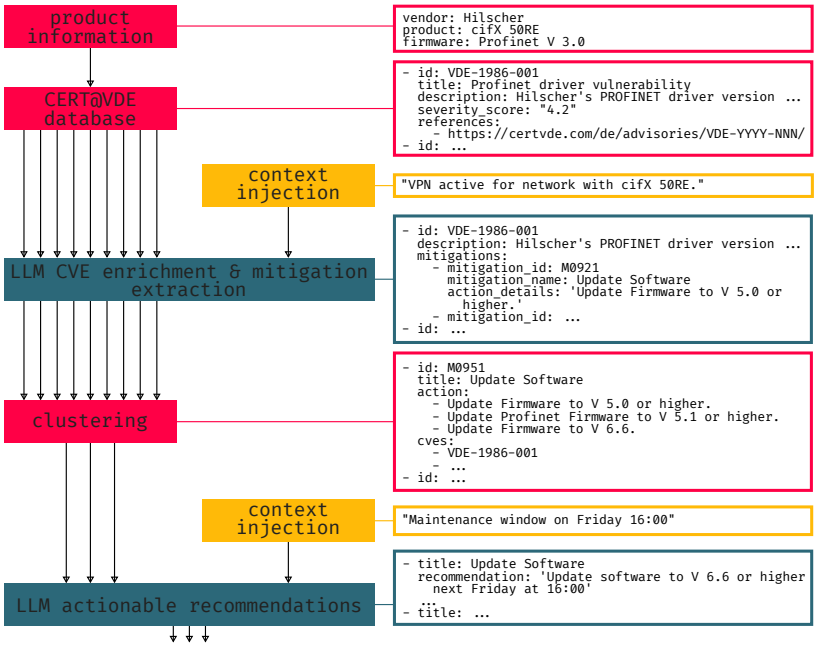


Abb. 3: Verarbeitungskette des Vulnerability Managements: Produktinformationen identifizieren CVEs. Ein LLM ergänzt Lösungsstrategien, bildet Maßnahmencluster und erzeugt kontextbezogene Empfehlungen.

die Risikobetrachtung stützt sich auf NIST SP 800-82 Rev. 3 [10; St23]. Die Aggregation ordnet Schwachstellen auch Maschinen, Prozessen und Anlagen zu und unterstützt CPS-Sicherheitsentscheidungen [OGS24; St23].

Abgleich mit Schwachstellendatenbanken: Uneinheitliche Hersteller- und Produktnamen erschweren die CPE-Zuordnung [Ji25; WH20]. Normalisierte Produktdaten werden mit der CERT@VDE-Datenbank abgeglichen.[12] Die Suche kombiniert CPE-Bezeichner [Nand] und Textmerkmale.

Die Einträge verweisen auf Common Vulnerabilities and Exposures (CVE).[13] CVEs enthalten meist Versionen und Schweregrad, aber nicht immer Lösungsstrategien [KK22; LP17]. Diese werden durch Referenzen und Herstellerhinweise ergänzen.

Eine lokal betriebene KI analysiert CVE-Beschreibung, Referenzen und Suchergebnisse, extrahiert Lösungsstrategien und ordnet Quellen zu. Die Lokale Verarbeitung schützt hierbei sensible Anlageninformationen [Hu25]. Dieses Vorgehen entspricht semi-automatisierten

[12] VDE Verband der Elektrotechnik Elektronik Informationstechnik e.V.: CERT@VDE, https://certvde.com/en/advisories/.

[13] MITRE Corporation: Common Vulnerabilities and Exposures (CVE), https://cve.mitre.org.

Security-Playbooks [Ob26]. Für den produktiven Einsatz wird Qwen3.8-Flash-Next verwendet.[14] Kleinere Modelle waren in ersten Erprobungen weniger zuverlässig.

Pro CVE entsteht eine strukturierte Beschreibung mit Komponente, Kennung, Schweregrad, Evidenz und Lösungsstrategien. Diese bildet die Grundlage der Maßnahmencluster.

Clustering und Generierung von Handlungsempfehlungen: Die Lösungsstrategien werden auf Enterprise Mitigations des MITRE-ATT&CK-Frameworks abgebildet.[15] Je Zuordnung entsteht ein Eintrag mit Kategorie und konkreter Maßnahme [Gr22]. Anschließend werden die Einträge nach Kategorie gruppiert.

Für jedes Cluster generiert das Sprachmodell eine Empfehlung mit Maßnahmenbedarf, CVEs, Komponenten und Quellen. Der höchste CVSS-Wert dient als Schweregrad. Diese konservative, potenziell überschätzende Aggregation hält kritische Schwachstellen sichtbar [Lo22], während die Priorisierung kontextbezogen auf Maschinen- und Anlagenebene erfolgt .

LLM-Assistenz und agentische Unterstützung Eine LLM-Assistenz bereitet Clusterempfehlungen mit Retrieval-Augmented Generation (RAG) [Le20] auf. Der Kontext umfasst Anlagenbeziehungen, Prozesskritikalität, Maßnahmenstatus, Segmentierung, Updates und Wartungsfenster. Durch den Kontext werden etwa bereits umgesetzte Maßnahmen nicht erneut vorgeschlagen. Die Assistenz unterstützt Sichtbarkeit, Priorisierung und Planung.

Die Betrachtung reicht von der Komponente über die Maschine bis zur Anlage [10; St23]. Die Datenbasis verbindet Geräte, gemeinsame Maßnahmen, Prozessabhängigkeiten und Betriebsauswirkungen. Machine Operator erhalten Risiken, OT Maintainer Maßnahmen und Machine Builder Planungshinweise. Eine agentische KI plant Maßnahmenreihenfolgen und bereitet deren Umsetzung vor [Ya23]. Die werkzeuggestützte Verarbeitung entspricht Kapitel 4.2.

Die Assistenz bewertet, begründet, plant und schlägt Maßnahmen vor. Nutzer prüfen Empfehlungen und geben die Umsetzung frei. Automatische Änderungen an produktiven Anlagen bleiben ausgeschlossen, da Change Control, Wartung und Patch Management kontrollierte OT-Prozesse sind [St23].

6 Zusammenfassung

Dieser Beitrag stellt das Konzept des AI-Security Monitorings vor, einen Ansatz zur Bündelung sicherheitsrelevanter Funktionen in industriellen Produktionsanlagen unter Einsatz von großen Sprachmodellen. Das Konzept umfasst fünf Anwendungsfälle, von denen zwei vertieft beschrieben wurden.

[14] Qwen Team: Qwen3.8-Flash-Next, https://qwen.ai/blog?id=qwen3.8-flash-next, abgerufen 2026-09-17.
[15] MITRE: ATT&CK Enterprise Mitigations, https://attack.mitre.org/mitigations/.

Das Network Monitoring erfasst den Netzwerkverkehr passiv, detektiert ohne gelabelte Angriffsdaten und verdichtet Einzelbefunde zu bewerteten Alarmen. Ein LLM-Agent bewertet diese Alarme unter Nutzung von über 19 lesenden Werkzeugen und ohne Trainingsdaten. Eine Gegenüberstellung auf demselben Testbed zeigt, dass ein Decision Tree zwischen sieben Angriffsmethoden in 88 % der Fälle und ein Cloud-Agent zwischen neun Angriffsmethoden in 89 % korrekt klassifiziert. Der Agent benötigte bis zu acht Minuten für die Analyse eines Alarms und ergänzt damit die unabhängige Klassifikation des Decision Trees.

Das Vulnerability Management führt von der Asset Discovery über den CVE-Abgleich zu Maßnahmenclustern und kontextbezogenen Empfehlungen. Sensible Anlageninformationen bleiben durch die lokale Verarbeitung im Unternehmen, Nutzer geben die Umsetzung frei. Die automatisierte Aufbereitung von Belegen beschleunigt die technische Erstbewertung, die Prüfung der Meldepflicht bleibt eine organisatorische Aufgabe.

Danksagung

Diese Arbeit wurde im Auftrag des Bundesministeriums für Forschung, Technologie und Raumfahrt im Rahmen des Projekts SRAG finanziert, Fördernummer 16KIS2316K.

Literatur

[10] IEC 62443-1-1: Security for Industrial Automation and Control Systems – Concepts and Models, International Electrotechnical Commission, 2010.

[ABS20] Alexander, O.; Belisle, M.; Steele, J.: MITRE ATT&CK for Industrial Control Systems: Design and Philosophy, Techn. Ber., The MITRE Corporation, 2020.

[Ag22] Aghaei, E.; Niu, X.; Shadid, W.; Al-Shaer, E.: SecureBERT: A Domain-Specific Language Model for Cybersecurity, 2022, arXiv: 2204.02685, URL: https://arxiv.org/abs/2204.02685.

[An25] Antari, A. et al.: Evaluating Large Language Models for Network Traffic Classification. IEEE Access/, 2025.

[An26] Ann, J. et al.: Fine-Tuning Large Language Models with Synthetic Data for Intrusion Detection in Industrial Control Systems. IEEE Access/, 2026.

[Da25] Daniel, N. et al.: Mapping Network Intrusion Detection Rules to MITRE ATT&CK with Large Language Models. In: Proc. IEEE Conf. on Communications and Network Security. 2025.

[Fe24] Ferrag, M. A. et al.: Revolutionizing Cyber Threat Detection with Large Language Models. IEEE Access 12/, 2024.

[Gr22] Grigorescu, A.; Nica, I.; Dascalu, M.; Rughinis, R.: CVE2ATT&CK: BERT-Based Mapping of CVEs to MITRE ATT&CK Techniques. Algorithms 15/9, S. 314, 2022.

[Hu25] Huang, H.; Li, Y.; Jiang, B.; Jiang, B.; Liu, L.; Liu, Z.; Sun, R.; Liang, S.: A Middle Path for On-Premises LLM Deployment: Preserving Privacy Without Sacrificing Model Confidentiality. In: Proceedings of EMNLP. 2025.

[Ji25] Jiang, Y.; Shang, F.; You, F. T. W.; Wang, H.; Cong, C. R.; Meng, Q.; Nay, O.; Lim, H. W.; Sikdar, B.: VulCPE: Context-Aware Cybersecurity Vulnerability Retrieval and Management, 2025, arXiv: 2505.13895, URL: https://arxiv.org/abs/2505.13895.

[KK22] Khanmohammadi, K.; Khoury, R.: Half-Day Vulnerabilities: A Study of the First Days of CVE Entries, 2022, arXiv: 2303.07990, URL: https://arxiv.org/abs/2303.07990.

[Le20] Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.; Rocktäschel, T.; Riedel, S.; Kiela, D.: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In: NeurIPS. 2020.

[Li24] Li, Y. et al.: IDS-Agent: An LLM Agent for Explainable Intrusion Detection in IoT Systems. In: Proc. NeurIPS Workshop on Open-World Agents. 2024.

[Lo22] Longueira-Romero, Á.; Flores, J. L.; Iglesias, R.; Garitano, I.: Gotta Catch 'em All: Aggregating CVSS Scores. In: Recuperación de Accidentes y Amenazas en Sistemas de Información (RECSI). 2022.

[LP17] Li, F.; Paxson, V.: A Large-Scale Empirical Study of Security Patches. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017.

[Nand] National Institute of Standards and Technology: Common Platform Enumeration (CPE), https://nvd.nist.gov/products/cpe, Accessed: 2026-09-21, n.d.

[Ob26] Oberhofer, D.; Grill, J.; Pernul, G.; Schöning, S.: Generating Semi-Automated Security Playbooks for Vulnerability Mitigation from Unstructured Advisory Data. International Journal of Information Security/, 2026.

[OGS24] Otto, J.; Grüttemeier, N.; Specht, F.: Security Decisions for CPS based on Solving Critical Node Problems with Vulnerable Nodes. In: AAAI-24 Workshop CAIPI. 2024.

[OS25] Otto, J.; Specht, F.: Efficient Cyberattack Detection in Logs of Cyber-Physical Production Systems. In: Proc. 7th SysInt. 2025.

[SD26] Specht, F.; Dhanesha, H. K.: Security-Monitoring und Cyberangriffsklassifikation in OT-Netzen, Vortrag, AUTOMATION 2026, Baden-Baden, 2026.

[SO24] Specht, F.; Otto, J.: Efficient ML-based Security Monitoring and Cyberattack Classification of Encrypted Network Traffic in ICS. In: Proc. 29th IEEE ETFA. 2024.

[SOE22] Specht, F.; Otto, J.; Eickmeyer, J.: Cyberattack Impact Reduction using SDN for Cyber-Physical Production Systems. In: Proc. 20th IEEE INDIN. 2022.

[St23] Stouffer, K.; Pease, M.; Tang, C.; Zimmerman, T.; Pillitteri, V.; Lightman, S.; Hahn, A.; Saravia, S.; Sherule, A.; Thompson, M.: Guide to Operational Technology (OT) Security, Techn. Ber. SP 800-82 Rev. 3, NIST, 2023.

[ST26] Specht, F.; Trsek, H.: Zero-Shot Cyberattack Classification in Industrial Control Systems Using Tool-Augmented Large Language Models. In: Proc. 31st IEEE ETFA. 2026.

[WH20] Wåreus, E.; Hell, M.: Automated CPE Labeling of CVE Summaries with Machine Learning. In: Proc. 17th Int. Conf. on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA). 2020.

[Wo25] Worae, D. et al.: Revelation: Retrieval-Augmented Interpretation of Network Traffic with Large Language Models. arXiv preprint arXiv:2506.00934/, 2025.

[WSJ17] Wollschlaeger, M.; Sauter, T.; Jasperneite, J.: The Future of Industrial Communication. IEEE Industrial Electronics Magazine 11/1, S. 17–27, 2017.

[Ya23] Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; Cao, Y.: ReAct: Synergizing Reasoning and Acting in Language Models, 2023, arXiv: 2210.03629, URL: https://arxiv.org/abs/2210.03629.

Automatisierte Harmonisierung syntaktisch heterogener Kommunikations-Systeme mittels instanzorientierter Erkundungs- und Zugriffsfunktionen

Thomas Fessler [1] und Jochen Müller [1]

Abstract: Bekannte und zugängliche Kommunikationsteilnehmer und Informationshaushalte vereinfachen die Erstellung Digitaler Zwillinge technischer Anlagen. In der heterogenen Praxis der Gebäude- und Industrieautomation ist das selten der Fall. Dieser Beitrag stellt eine Methode zur automatisierten protokollunabhängigen Erkundung von Kommunikationsnetzen und -teilnehmern sowie deren Schnittstellen und zur automatisierten Instanziierung lesender Zugriffsfunktionen vor. Vorab festgelegt werden nur der direkte Zugriff auf das physikalische Medium und die grundlegenden Signaleigenschaften. Ein Large Language Model leitet aus Mitschnitten strukturierte Beschreibungen von Telegrammrahmen, Prüfsummen und Kommunikationssequenzen ab, die zur Laufzeit regelbasiert ausgeführt werden. Die Evaluation der Methode in Netzen mit BACnet MS/TP, Modbus/RTU, BACnet/IP und Modbus/TCP zeigt, dass die passive Erkundung auf den EIA-485-Bussen vollständig und fehlerfrei gelingt und im Ethernet-Netz 23 von 34 Teilnehmern sowie alle 25 Modbus/TCP-Datenpunkte korrekt erfasst werden. In der aktiven Erkundung werden 14 von 16 Erkundungs- und Zugriffsmethoden richtig gewählt; die erzeugten Kommunikationssequenzen erfüllen im Mittel 64 % der Bewertungskriterien, 29 von 218 sind unmittelbar ausführbar. Fehler entstehen vor allem dort, wo Informationen oder Zustände außerhalb der Beschreibung liegen, etwa Routing-Information oder der Verbindungszustand bei TCP, und wo erkannte Anforderungen nicht vollständig umgesetzt werden. Insgesamt zeigt sich, dass die Methode die Erkundung unbekannter Kommunikationsnetze bis hin zum Datenzugriff grundsätzlich ermöglicht, Kommunikationssequenzen jedoch noch nicht zuverlässig instanziiert werden.

Keywords: Interoperabilität, heterogene Netze, Einsatz von LLM

1 Einleitung

Im industriellen und gebäudetechnischen Umfeld erzeugt das Informationsmanagement Digitaler Zwillinge von technischen Anlagen einen erheblichen manuellen Aufwand [MNS24]. Grund hierfür ist die begrenzte Interoperabilität von Systemen in der heterogenen automatisierungstechnischen Praxis (Technologie, Herstellerabhängigkeiten). Syntaktische Interoperabilität als Befähigung zum Datenaustausch durch Verwendung des gleichen Datenformats wird durch die Heterogenität von Datenzugriffen (Kommunikationstechnologien, Datenformate) erschwert, die mangelnde semantische Interoperabilität entsteht durch eine Vielfalt an Bedeutungen von Datenbezeichnungen und -inhalten [DLH11]. Aktuelle Forschungsaktivitäten bieten Lösungsansätze für die automatische Abbildung heterogener Semantik auf informationstechnische Standards [Bo24; Mi22; Xi24], setzen

[1] TH Köln, Cologne Lab for Building Automation and Optimization (COLABO), Betzdorfer Str. 2, 50679 Köln, thomas_georg.fessler@th-koeln.de, https://orcid.org/0009-0005-2999-762X; jochen.mueller@th-koeln.de, https://orcid.org/0009-0003-3429-5329

jedoch bekannte Strukturen, Erkundungs- und Zugriffsfunktionen voraus. So erzeugt etwa [Bo23] Digitale Zwillinge und Monitoring-Anwendungen automatisiert durch semantisches Matching, allerdings für Teilnehmer mit BACnet-Schnittstelle, deren Erkundungs- und Zugriffsmechanismen standardisiert sind. Automatismen zur Erstellung Digitaler Zwillinge im Umfeld heterogener Kommunikationstechnologien und Informationsstrukturen fehlen bis heute.

Architekturen für interoperable Systeme setzen syntaktische Homogenität bereits bei der Errichtung des Netzes voraus, etwa über eine zentrale Instanz, bei der sich Teilnehmer anmelden [Sa24], über konfigurierte Gateways [Az24] oder über gleichberechtigte Teilnehmer eines Peer-to-Peer-Netzes [Dr05]. Verfahren zur Erkundung bestehender Netze, etwa Port-Scans [Vi99] oder die Auswertung von Netzwerkverkehr [Sh18], identifizieren Teilnehmer, erschließen aber nicht deren Informationshaushalt. Werkzeuge, die darüber hinausgehen, sind protokollspezifisch implementiert und erkennen nur, was sie bereits kennen. Large Language Models (LLMs) verfügen dagegen über umfangreiches Wissen, auch zu Kommunikationsprotokollen, und werden zunehmend für Aufgaben der Softwareentwicklung eingesetzt [Zh25]. Unmittelbar zur Steuerung der Kommunikation eignen sie sich jedoch nicht, da ihre Ausgaben nicht deterministisch und für zeitkritische Abläufe zu langsam sind.

Ziel dieser Arbeit ist die automatisierte Harmonisierung syntaktisch heterogener Kommunikationssysteme. Teilnehmer, die über unterschiedliche Protokolle kommunizieren, sollen einheitlich über standardisierte Digitale Zwillinge zugänglich werden. Teil dieser Arbeit ist daher die Erzeugung strukturierter Beschreibungen von Erkundungs- und Zugriffsfunktionen durch LLMs zur Entwurfszeit, die zur Laufzeit regelbasiert und deterministisch ausgeführt werden. Grundlage der Kommunikation ist damit nicht mehr eine gemeinsame Sprache aus Syntax und Semantik, sondern allein die physikalische Befähigung zum Informationsaustausch. Diese Arbeit ergänzt so Methoden der semantischen Harmonisierung, die einheitliche und bekannte Informationsstrukturen voraussetzen [Bo24]. Die Beiträge sind:

- eine vierstufige, protokollunabhängige Methode zur Erkundung von Netzen, Geräten und Schnittstellen bis zur Instanziierung lesender Zugriffsfunktionen,
- Beschreibungsmodelle für Telegrammrahmen, Referenzen auf Telegrammabschnitte, Prüfsummen, Kommunikationssequenzen und den Informationshaushalt eines Geräts,
- eine Evaluierung an drei realen Netzen mit BACnet MS/TP, Modbus/RTU sowie BACnet/IP und Modbus/TCP.

Abschnitt 2 beschreibt das methodische Vorgehen und den Versuchsaufbau. Abschnitt 3 stellt die Methode und die Modelle vor und berichtet deren Evaluation. Abschnitt 4 fasst die Ergebnisse zusammen und gibt einen Ausblick.

2 Methodik

Grundlage für die Entwicklung der Harmonisierungsmethode ist das OSI-Referenzmodell [IS94b]. Es teilt die Abläufe zwischen interagierenden Systemen in sieben Schichten mit abstrakter Zuständigkeit ein. Diese reichen von der Festlegung der Signale, etwa Einsen und Nullen als elektrische oder optische Pegel, auf der untersten Schicht bis zu komplexen Objektstrukturen der kommunizierenden Anwendungen auf der obersten. Einzelne Kommunikationsprotokolle decken Aspekte einer oder mehrerer Schichten ab.

Anhand von diesen Schichten zugeordneten Kommunikationsaspekten werden die Erkundung von Kommunikationsnetzen sowie die Erkundung von und der Zugriff auf Informationshaushalte einzelner Teilnehmer unabhängig von den tatsächlich verwendeten Protokollen abstrakt beschrieben. Ziel ist es, die Festlegungen und daraus folgende Einschränkungen auf bekannte Kommunikationsprotokolle möglichst gering zu halten. Als Dateneinheit werden daher Telegramme (Schicht 2, entsprechen einer einzelnen Nachricht, bei Ethernet als *Frames* bezeichnet) festgelegt. Zudem wird die Ermittlung des Telegramm-Formats aus einem kontinuierlichen Signal-Strom (Schicht 1, Folge von Bits/Bytes) mittels Mustererkennung durch LLMs betrachtet.

Die abstrakten Beschreibungen bilden die Struktur einer Anwendung, welche die Instanziierung von Erkundungs- und Zugriffsfunktionen mittels LLMs automatisiert. Die Funktionen sind instanzorientiert. Sie werden nicht vorab je Protokoll oder Gerätetyp implementiert, sondern für den konkreten Teilnehmer im konkreten Netz aus dessen beobachtetem Datenverkehr und den Ergebnissen der Erkundung erzeugt. Diese Arbeit beschränkt sich auf lesende Zugriffe, schreibende Zugriffe auf den Informationshaushalt der Teilnehmer werden nicht betrachtet. Aktive Kommunikation umfasst damit Erkundungsfunktionen, lesende Zugriffsfunktionen und die dafür notwendigen Kommunikationssequenzen wie einen Verbindungsaufbau.

Protokollspezifische Beispiele in den LLM-Interaktionen wurden bewusst auf grundlegende, nicht automatisierungsspezifische Protokolle beschränkt. So dient als Beispiel einer Erkundungsfunktion nicht ein automatisierungsspezifischer Dienst wie BACnet *Who-Is* oder PROFINET DCP *Identify*, sondern die ARP-Anfrage (*Address Resolution Protocol*) aus der Internetprotokollfamilie. Beide folgen demselben Muster: Auf eine Broadcast-Anfrage antworten die angesprochenen Teilnehmer und geben damit ihre Adresse und Existenz preis. Auf diese Weise bleiben die Beispiel-Eingaben unabhängig von den real erkundeten Netzen.

2.1 Versuchsaufbau zur Evaluation

Die Konzepte wurden prototypisch in Python implementiert und evaluiert, mit PySerial und einem USB-Konverter für EIA-485 (auch RS-485, aktuell TIA-485-A) sowie Raw-Sockets (`AF_PACKET`) für Ethernet, über die vollständige Frames gesendet und empfangen werden. Da der Kernel selbst erzeugte TCP-Verbindungen nicht kennt und das SYN-ACK der

Gegenstelle sonst mit einem RST beantworten würde, wurden ausgehende RST-Segmente per Firewall-Regel unterdrückt. Für BACnet MS/TP kam zusätzlich die Bibliothek *misty* zum Einsatz (Abschnitt 3.7). Als LLM diente GPT-5.5 von OpenAI.

Betrachtet wurden ein BACnet MS/TP- und ein Modbus/RTU-Versuchsstand über EIA-485 sowie ein Ethernet-Labornetz, in dessen Mitschnitt neben zahlreichen nicht automatisierungsrelevanten Teilnehmern 31 BACnet/IP- und drei Modbus/TCP-Teilnehmer nachweisbar waren. Im Modbus/RTU-Versuchsstand erfolgte die passive Erkundung im laufenden Betrieb; für die aktive Erkundung wurde der vorhandene Master wegen des Single-Master-Betriebs vom Bus getrennt. Die erzeugten Kommunikationssequenzen wurden nach einer regelbasierten Validierung, deren Kriterien in den jeweiligen Abschnitten angegeben sind, in den Versuchsnetzen real ausgeführt.

3 Ergebnisse

In diesem Abschnitt werden die entwickelte Harmonisierungsmethode, die zugehörigen Beschreibungsmodelle und die Evaluationsergebnisse vorgestellt.

3.1 Harmonisierungsmethode

Die Harmonisierungsmethode umfasst die Schritte vom Verbinden mit einem unbekannten Kommunikationsnetz bis zur Erstellung standardisierter Digitaler Zwillinge einzelner Teilnehmer. Ihren Kern bildet die Erkundung. Sie wird hier im Überblick beschrieben, die folgenden Abschnitte führen zentrale Aspekte aus.

Die Erkundung wurde als vierstufiger Prozess modelliert (Abb. 1). Im ersten Schritt, der passiven Netzwerkerkundung, wird der stattfindende Datenverkehr aufgezeichnet und analysiert. Es entsteht ein nicht notwendigerweise vollständiger Überblick über vorhandene Teilnehmer bei minimalen Kenntnissen über das Kommunikationsnetz.

Im zweiten Schritt, der aktiven Netzwerkerkundung, sollen weitere Teilnehmer identifiziert werden. Dazu werden anhand der im Mitschnitt erkannten Anwendungsprotokolle geeignete Erkundungsfunktionen ermittelt und ausgeführt. Alternativ kann die aktive Erkundung regelbasiert für eine definierte Menge von Kommunikationsprotokollen erfolgen. Dann ist allerdings zu prüfen, ob die Nutzung nicht vorgesehener Protokolle das Netz stören und so dessen eigentliche Funktion beeinträchtigen kann. Diese Betrachtung ist nicht Teil dieser Arbeit.

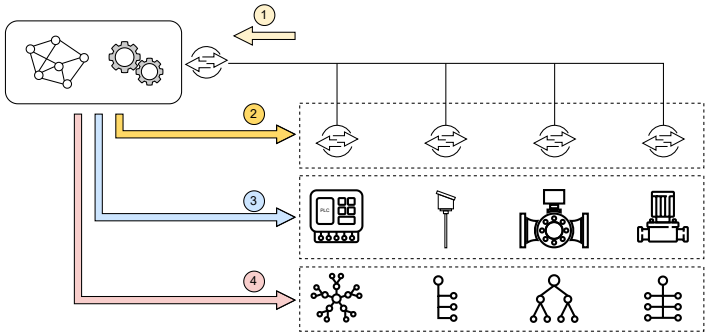


Abb. 1: Vierstufige Erkundung: (1) passive Netzwerkerkundung durch Aufzeichnung des Datenverkehrs, (2) aktive Netzwerkerkundung über Erkundungsfunktionen, (3) Geräteerkundung und (4) Erkundung der Informationsstrukturen.

Der dritte Schritt, die Geräteerkundung, zielt auf Informationen über das Gerät selbst. Dazu gehören etwa Hersteller, Modell und Typ. Diese Informationen sind insbesondere für Geräte mit nicht direkt erkundbaren Kommunikationsschnittstellen relevant. Sie können bspw. als Grundlage einer automatisierten Suche nach Schnittstellendokumentation und der Extraktion der für den Zugriff relevanten Informationen dienen. Diese Ansätze wurden in dieser Arbeit nicht ausgearbeitet.

Der vierte Schritt behandelt die Erkundung der Informationsstrukturen mittels nativer Erkundungsmechanismen von Kommunikationsschnittstellen. Ob ein Protokoll einen solchen Mechanismus bietet, ist dabei selbst Teil der Erkundung: Bietet es keinen, wie Modbus für die Struktur seines Registerraums [Kä25], ist ein Negativergebnis korrekt. Aus den erkundeten Informationsstrukturen werden lesende Zugriffsfunktionen für den jeweiligen Teilnehmer instanziiert. Die gewonnenen Informationen werden im Geräte-Informationsmodell zusammengeführt (Abschnitt 3.3).

3.2 Erkundungs- und Zugriffsfunktionen

Funktionen lassen sich als Transformation von Eingaben in Ausgaben beschreiben, die durch bestimmte Mechanismen erfolgt und bestimmten Steuerungen unterliegt [IS17]. Im Kontext der Kommunikation zwischen Systemen sind Funktionen diejenigen Leistungen, die ein System einem Kommunikationspartner bereitstellt oder von ihm in Anspruch nimmt. Als Erkundungsfunktionen werden Funktionen verstanden, die Informationen über Existenz, Identität oder Struktur eines Kommunikationsnetzes, eines Teilnehmers oder seiner Schnittstellen liefern. Zugriffsfunktionen sind der Mechanismus, mit dem an einer bekannten Adresse im Informationshaushalt eines Teilnehmers gelesen oder geschrieben wird.

Erkundung kann damit auf zwei Wegen erfolgen: über funktionsbasierte Erkundungsfunktionen mit dediziertem Dienst, etwa die *Identify*-Anfrage des PROFINET-Protokolls DCP oder den Befehl *ListIdentity* bei EtherNet/IP, oder über Zugriffsfunktionen auf Informationsobjekte mit Metainformationen, etwa das Lesen des Identity-Objekts aus dem CANopen-Objektverzeichnis. Im zweiten Fall wird eine Erkundung also über Zugriffsfunktionen realisiert.

Erkundungsfunktionen existieren auf verschiedenen Schichten des OSI-Referenzmodells. So gibt es neben den zuvor erwähnten Beispielen auf Anwendungsebene bspw. die der Sicherungsschicht (Schicht 2) zuzuordnende ARP-Anfrage. Diese Arbeit konzentriert sich auf Erkundungsfunktionen der Anwendungsschicht. Auch denkbar ist eine Netzwerkerkundung auf unteren Schichten und eine darauf aufbauende Erkundung vorhandener Anwendungsschnittstellen, bspw. durch Verbindungsversuche nach Trial-and-Error-Prinzip über Standard-Ports in IP-Netzen. Dieser Ansatz wird nicht ausgearbeitet.

3.3 Geräte-Informationsmodell

Als Zielstruktur der Erkundung wurde ein Geräte-Informationsmodell entworfen, das während einer Erkundung schrittweise angereichert wird (Abb. 2). Ein Gerät wird darin als Menge kommunizierender Anwendungen beschrieben. Anwendungen verfügen über einen anwendungsspezifischen Identifikator sowie einen Protokollstapel. Dieser beinhaltet die unterhalb der Anwendung angesiedelten Kommunikationsprotokolle mit ihren protokollspezifischen Identifikatoren, etwa der MAC-Adresse bei Ethernet oder der IP-Adresse beim Internet Protocol. Die zugänglichen Informationen einer Anwendung werden als Datenpunkte abgebildet. Angelehnt an die *Property Affordances* der WoT Thing Description [Wo23] übernehmen sie deren Beschreibungselemente Titel, Beschreibung und Datentyp, letzteren mit den Basistypen von JSON Schema. Datenpunkte vom Typ Objekt oder Array können weitere Datenpunkte enthalten, sodass objektorientierte Strukturen wie die Slot-Subslot-Hierarchie bei PROFINET oder die Klassen-Instanz-Attribut-Struktur bei EtherNet/IP darstellbar sind. An die Stelle der Protokollbindungen der Thing Description (*Forms*) tritt eine Zugriffsfunktion, die eine Kommunikationssequenz nach Abschnitt 3.5 enthält.

3.4 Telegramm-Format

Beschränkt sich die Protokollfestlegung auf OSI-Schicht 1, so sind zwar die grundlegenden Signale definiert, nicht aber die damit übertragenen Telegramme. Daher wurde ein Modell zur Definition von Telegramm-Anfang und -Ende entwickelt. Der Anfang wird entweder über eine feste Bytefolge (Präambel) oder über eine Pause auf der Leitung festgelegt. Das Ende ergibt sich aus Längenregeln. Es kann sich entweder um eine feste oder eine dem Telegramm zu entnehmende Länge handeln. Beide Varianten können durch Bedingungen

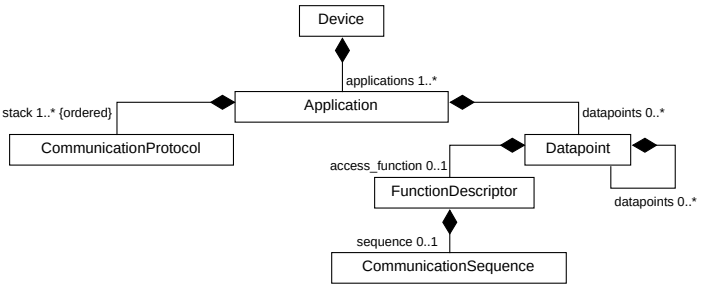


Abb. 2: UML-Klassendiagramm des Geräte-Informationsmodells.

kombiniert und weiter eingegrenzt werden. Damit sind auch Protokolle abbildbar, bei denen Lage und Bedeutung der Längenangabe von anderen Feldern abhängen, etwa vom Startbyte bei PROFIBUS DP, das zwischen Telegrammen fester und variabler Länge unterscheidet. Das Schema ist zudem eine Grundlage der aktiven Kommunikation. Es wird benötigt, um den vollständigen Empfang eines Telegramms zu erkennen, auf das reagiert werden muss. Für Ethernet entfällt diese Ermittlung, da die Telegrammgrenzen durch IEEE 802.3 festgelegt sind.

Validiert wurde das Modell mit je fünf Mitschnitten der beiden EIA-485-Versuchsstände. Sie wurden mit und ohne Zeitstempel zur Format-Ermittlung übergeben. Dabei wurden die Ergebnisse durch die UART-Zeitstempel verschlechtert. So wurde BACnet MS/TP teilweise nicht korrekt identifiziert, bei Modbus/RTU wurde die Angabe der Pausenzeit zwischen zwei Telegrammen verfälscht. Bei allen Testläufen ohne Zeitstempel wurden die Protokolle korrekt identifiziert und standardkonform beschrieben.

3.5 Kommunikationssequenzen

Kommunikationssequenzen beschreiben den in der Regel aus mehreren Telegrammen bestehenden Ablauf einer Interaktion. Sie folgen dabei dem Anfrage-Antwort-Prinzip [IS94a; TW12]: Telegramme werden abwechselnd gesendet und empfangen. Beginnen können Sequenzen mit einem eigenen Telegramm, etwa beim TCP-Handshake, oder mit einem empfangenen, etwa bei PROFIBUS oder BACnet MS/TP, wo ein Master auf eine Abfrage antworten muss, um in den Token-Ring aufgenommen zu werden. Empfangene Telegramme werden über zur Identifikation abzugleichende Abschnitte, gesendete Telegramme über Vorlagen mit Platzhaltern definiert (Abb. 3).

Da die protokollspezifischen Telegrammstrukturen nicht implementiert werden, steht die Bedeutung einzelner Bytes zur Laufzeit nicht zur Verfügung. Zugleich stehen die Telegramme einer Sequenz in Beziehung zueinander, und mitunter müssen Elemente aus empfangenen in zu sendende Telegramme übernommen werden, z. B. die Adresse des

Absenders. Diese Zusammenhänge bilden die Platzhalter ab: *Referenzen* verweisen über Telegramm- und Byte-Index sowie Länge auf Telegrammabschnitte, die *Transformationen* durch mathematische Operationen verändern können (z. B. die Erhöhung der Sequenznummer um 1 während eines TCP-Handshakes). *Schnittstellen-Eigenschaften* übernehmen Eigenschaften der eigenen Kommunikationsschnittstelle wie die MAC-Adresse, *Werte* geben die Position eines gelesenen Wertes in einer Lesebestätigung an, und *Prüfsummen* beschreiben Prüfsummenverfahren abstrakt.

Letzteres ist nötig, da die Telegramminstanziierung erst zur Laufzeit finalisiert wird. Das Prüfsummenmodell umfasst vier Familien: zyklische Redundanzprüfung nach dem parametrischen Modell von Williams [Wi93], arithmetische Summe, XOR-Verknüpfung und Einerkomplementsumme. Die Eingangsbytes sind eine geordnete Folge von Referenzen und Konstanten, sodass auch Pseudo-Header wie bei TCP und UDP abbildbar sind.

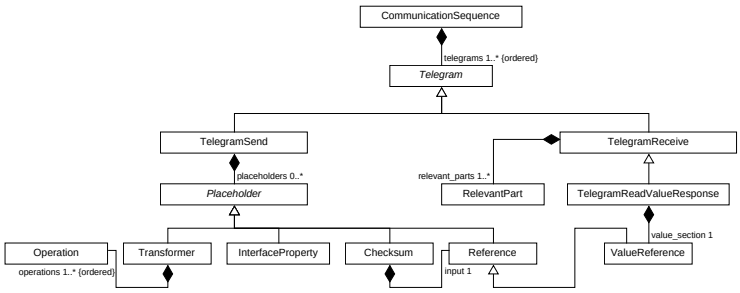


Abb. 3: UML-Klassendiagramm der entwickelten Abstraktion von Kommunikationssequenzen.

3.6 Evaluation der passiven Erkundung

Die passive Erkundung ist auf den während der Beobachtung stattfindenden Datenaustausch im eigenen (Sub-)Netzwerk beschränkt. Bei BACnet MS/TP sind Master über die Telegramme des Token-Handlings sichtbar, zudem wie bei Modbus/RTU die vom Master abgefragten Slaves, in geswitchten Ethernet-Netzen nur Broadcast-, Multicast- und eigener Verkehr. Um eine breite Ausgangslage zu schaffen, wurde im Ethernet-Netz während der Aufzeichnung bewusst Kommunikation angeregt: ein *Who-Is* und Modbus/TCP-Abfragen.

Bewertet wurde die passive Erkundung in einem Lauf gegen die im Mitschnitt nachweisbaren Teilnehmer, nach Vollständigkeit, Fehlidentifikationen (Teilnehmer ohne Beleg oder mit falschem Protokoll) sowie der Korrektheit von Protokollstapel, Identität (MS/TP-Adresse, Slave- bzw. Unit-Adresse, Device-Identifikator) und beobachteten Datenpunkten (Tab. 1).

Auf den EIA-485-Bussen war die passive Erkundung vollständig und korrekt. Im Ethernet-Netz waren erkannte Teilnehmer korrekt beschrieben, und alle 25 Modbus/TCP-Datenpunkte stimmten. Es fehlten jedoch 11 von 31 BACnet/IP-Teilnehmern. Kein Device-Identifikator

Tab. 1: Evaluation der passiven Erkundung. Referenz sind die im Mitschnitt nachweisbaren Teilnehmer.

Netz	vorhanden	erkannt	Fehlident.	Stapel	Identität	Datenpunkte
BACnet MS/TP	4	4	0	4/4	4/4	6/6
Modbus/RTU	4	4	0	4/4	4/4	4/4
BACnet/IP	31	20	1[a]	20/20	0/20	–
Modbus/TCP	3	3	0	3/3	3/3	25/25
weitere Protokolle (Ethernet)	–	24	15[b]	–	–	–

[a] Eine nicht vorkommende Adresse. [b] 14-mal Ethertype 0x8899 als SERCOS III (0x88CD), ein Modbus/RTU-Teilnehmer ohne Beleg.

wurde korrekt als Anwendungsidentifikator übernommen. Stattdessen verwendete das LLM überwiegend die IP-Adresse als Identifikator der Anwendung. Zudem wurden 14 nicht automatisierungsrelevante Teilnehmer über einen falsch gedeuteten Ethertype als SERCOS III und damit als Automatisierungsgeräte eingestuft.

3.7 Evaluation der aktiven Erkundung

Evaluiert wurde die Erzeugung der vier aktiven Funktionsarten anhand von Vollständigkeit, Korrektheit und Ausführbarkeit der Kommunikationssequenzen. Eingabe waren die Geräte-Informationsmodelle der drei Versuchsnetze nach der passiven Erkundung. Die Ergebnisse wurden schrittweise ausgewertet und zur Eingabe des nächsten Schrittes wurde der Referenzzustand hergestellt, um Fehlerfortpflanzungen zu verhindern. Die Funktionen wurden durch das LLM zuerst benannt und beschrieben, bevor im nächsten Schritt die Sequenz-Instanziierung anhand der benannten Funktionen erfolgte. Vor der Ausführung wurden die Sequenzen anhand der folgenden Kriterien bewertet:

K1 Vollständigkeit: Alle Sende- und Empfangsschritte der Sequenz sind in richtiger Reihenfolge enthalten.

K2 Statische Abschnitte: Die festen Abschnitte ausgehender Telegramme stimmen.

K3 Platzhalter: Position, Länge und Transformation stimmen; fest eingetragene Laufzeitwerte gelten als Fehler.

K4 Prüfsummen: Alle Prüfsummen sind korrekt beschrieben.

K5 Rückmeldung: Die relevanten Abschnitte identifizieren die erwartete Antwort eindeutig.

Die Ausführung der BACnet MS/TP-Sequenzen konnte nicht wie vorgesehen direkt über den USB-Konverter erfolgen, da dieser in Kombination mit der Python-Implementierung die zeitlichen Anforderungen des Token-Handlings nicht erfüllen konnte. Sie wurden

daher in Code der *misty*-Bibliothek übersetzt. Diese bietet eine BACnet-konforme Python-Implementierung, der Buszugriff erfolgt aber über eine ausgelagerte vorkompilierte C-Anwendung. Als ausführbar gilt eine MS/TP-Sequenz daher nur, wenn sich die erzeugten Telegramme ohne inhaltliche Korrektur übertragen ließen.

Tab. 2: Evaluation der aktiven Erkundung. Methodenwahl je Protokoll, Kriterien je instanziierter Sequenz.

Funktionsart	Methoden	Seq.[a]	K1	K2	K3	K4	K5	K1–K5	aus-führbar
Netzwerk-erkundung	4/4	2/2	50 %	100 %	50 %	100 %	50 %	70 %	1/2
Geräte-erkundung	4/4	17/42	62 %	77 %	59 %	59 %	77 %	67 %	3/17
Schnittstellen-erkundung	2/4	16/25	54 %	50 %	63 %	56 %	50 %	55 %	4/16
Zugriffs-funktionen[b]	4/4	183/2245[c]	75 %	65 %	57 %	66 %	62 %	65 %	21/183
Gesamt[d]	14/16	218/2314	60 %	73 %	57 %	70 %	60 %	64 %	29/218

[a] erzeugt/erwartet [b] Aufgrund der Datenpunktmenge exemplarisch für ein BACnet- und ein Modbus-Gerät.
[c] Größenbedingt wurden die Datenpunktlisten der BACnet-Geräte nach Objekten aufgeteilt. Das LLM neigte dazu, Datenpunkte zusammenzufassen, und instanziierte nur für wenige Properties je Objekt Zugriffsfunktionen.
[d] Gesamt: Prozentwerte sind Mittelwerte über die Funktionsarten.

Die Methodenwahl war in 14 von 16 Fällen korrekt. Beide Fehler betreffen Modbus, für dessen Registerraum ein Negativergebnis korrekt gewesen wäre. Hinsichtlich der Sequenzvollständigkeit zeigt sich, dass diese mit zunehmender Komplexität (bspw. bei verbindungsorientierter Kommunikation) abnimmt. So wurde der für Modbus/TCP notwendige Handshake nicht durchgängig vollständig instanziiert, obwohl er vorab meist als notwendig erkannt und beschrieben wurde. Bei den anderen Modellkriterien sind keine Abweichungsmuster erkennbar. Mögliche Gründe sind nicht hinreichend genaue Beschreibungen für das LLM und der fehlende Determinismus des LLMs. Sie sind in weiteren Untersuchungen zu analysieren.

Ausführbar waren 29 der 218 Sequenzen. Das Scheitern lag meist nicht an der Syntax, sondern an Informationen und Zuständen außerhalb der Beschreibung, etwa an fehlender Routing-Information für Geräte hinter BACnet-Routern oder am Verbindungszustand bei TCP.

4 Fazit und Ausblick

Die vorgestellte Methode ermöglicht die Erkundung von Kommunikationsnetzen, Teilnehmern und Schnittstellen allein auf Basis der Festlegung grundlegender Signale (Bits

und Bytes). Ein LLM erzeugt dazu strukturierte Beschreibungen, die regelbasiert ausgeführt werden. Bereits das Telegramm-Format ließ sich aus unstrukturierten Byte-Strömen zuverlässig ermitteln, sofern keine ungenauen Zeitstempel übergeben wurden.

Auf den EIA-485-Bussen war die passive Erkundung vollständig und korrekt, im heterogenen Ethernet-Netz dagegen lückenhaft und teils fehlerhaft, insbesondere bei der Geräteidentifikation. Bei der Methodenwahl traten Fehler vor allem dort auf, wo ein Negativergebnis richtig gewesen wäre. Mit zunehmender Komplexität nimmt die Vollständigkeit der Sequenzen ab: Anforderungen wie ein Verbindungsaufbau wurden erkannt, aber nicht vollständig umgesetzt, und bei großen Informationshaushalten wurden Datenpunkte zusammengefasst und nur für einen Bruchteil Zugriffsfunktionen instanziiert. In der Ausführung scheiterten die Sequenzen meist nicht an der Syntax, sondern an Informationen und Zuständen außerhalb der Beschreibung, etwa fehlender Routing-Information oder dem TCP-Verbindungszustand.

Dennoch zeigen die Ergebnisse, dass die Methode funktioniert: Die Methodenwahl war in 14 von 16 Fällen korrekt, die Sequenzen erfüllten rund zwei Drittel der Kriterien, und in jeder Funktionsart ließen sich Sequenzen real ausführen. Da allgemeine Kommunikationsaspekte betrachtet werden, ist die Methode zudem nicht auf die Automatisierungstechnik beschränkt. Fehler ließen sich durch die Implementierung grundlegender Protokolle wie IP, TCP und UDP reduzieren. Das LLM müsste dann nur noch deren Verwendung feststellen und die verbleibenden Elemente beschreiben; Verbindungsaufbau, Verbindungszustand und Pseudo-Header-Prüfsummen lägen außerhalb seiner Verantwortung. Dies widerspricht zwar dem Ansatz, Festlegungen auf bekannte Protokolle gering zu halten, schränkt die Allgemeinheit angesichts der Verbreitung dieser Protokolle aber kaum ein und dürfte die Zuverlässigkeit deutlich erhöhen.

Nicht betrachtet wurden schreibende Zugriffe, segmentierte Übertragungen, Routing sowie ausbleibende oder fehlerhafte Rückmeldungen; sie sind in Methode und Modelle einzuarbeiten. Das Sequenzmodell sollte zudem Wiederholungen unterstützen, um große Informationshaushalte ohne Einzelinstanziierung abzudecken. Ebenso fehlt ein Modell zur Interpretation gelesener Werte, das über ihre Lokalisierung hinaus etwa das Bilden einer Gleitkommazahl aus zwei Modbus-Registern erlaubt.

Literaturverzeichnis

[Az24] Azad, T. et al.: Hierarchical Decentralized Edge Interoperability. IEEE Internet of Things Journal 11 (8), S. 13948–13960, 2024.

[Bo23] Both, M. et al.: Automated monitoring applications for existing buildings through natural language processing based semantic mapping of operational data and creation of digital twins. Energy and Buildings 300, S. 113635, 2023.

[Bo24] Both, M.: Semantische Interoperabilität digitaler Zwillinge als Basis für automatisierte Erkundungsmechanismen: Dissertation. 2024.

[DLH11] Diedrich, C.; Lüder, A.; Hundt, L.: Bedeutung der Interoperabilität bei Entwurf und Nutzung von automatisierten Produktionssystemen. at – Automatisierungstechnik, S. 426–438, 2011.

[Dr05] Drinjakovic, D.: Zugriff auf Informationen und Dienste in einem verteilten Automatisierungssystem mit selbstkonfigurierenden, semantischen Ordnungsstrukturen: Zugl.: Aachen, Techn. Hochsch., Diss., 2004. VDI-Verl., Düsseldorf, 2005.

[IS17] ISO/IEC/IEEE: ISO/IEC/IEEE 24765:2017 – Systems and software engineering – Vocabulary, Norm ISO/IEC/IEEE 24765:2017, International Organization for Standardization, 2017.

[IS94a] ISO/IEC: ISO/IEC 10731:1994 – Information technology – Open Systems Interconnection – Basic Reference Model – Conventions for the definition of OSI services, Norm ISO/IEC 10731:1994, International Organization for Standardization, 1994.

[IS94b] ISO/IEC: ISO/IEC 7498-1:1994 – Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model, Norm ISO/IEC 7498-1:1994, International Organization for Standardization, 1994.

[Kä25] Kämper, B. et al.: Gewinnung von Strukturinformationen aus semantisch heterogenen Gebäudeautomationsnetzen durch Mustererkennung mittels Künstlicher Intelligenz, Abschlussbericht MonKI-Projekt, 2025.

[Mi22] Miny, T.: Konzept für die semantische Interoperabilität zwischen Informationsmodellen: Dissertation. 2022.

[MNS24] Müller, J.; Nellessen, G. T.; Scholz, H.: Optimierung der Gebäudeautomation in kommunalen Bestandsgebäuden: Performance Monitoring der Technischen Gebäudeausrüstung. atp magazin 66, S. 74–79, 2024.

[Sa24] Sadeghi, M. et al.: Interoperability of heterogeneous Systems of Systems: from requirements to a reference architecture. The Journal of Supercomputing 80 (7), S. 8954–8987, 2024.

[Sh18] Shahid, M. R. et al.: IoT Devices Recognition Through Network Traffic Analysis. In: 2018 IEEE International Conference on Big Data (Big Data). S. 5187–5192, 2018.

[TW12] Tanenbaum, A. S.; Wetherall, D. J.: Computernetzwerke. Pearson Studium, München, 2012.

[Vi99] de Vivo, M. et al.: A review of port scanning techniques. ACM SIGCOMM Computer Communication Review 29 (2), S. 41–48, 1999.

[Wi93] Williams, R. N.: A Painless Guide to CRC Error Detection Algorithms, 1993.

[Wo23] World Wide Web Consortium: Web of Things (WoT) Thing Description 1.1, 2023, https://www.w3.org/TR/wot-thing-description11/, Stand: 23. 02. 2026.

[Xi24] Xia, Y. et al.: Generation of Asset Administration Shell With Large Language Model Agents: Toward Semantic Interoperability in Digital Twins in the Context of Industry 4.0. IEEE Access 12, S. 84863–84877, 2024.

[Zh25] Zheng, Z. et al.: Towards an understanding of large language models in software engineering tasks. Empirical Software Engineering 30 (2), 2025.

Towards LLM-Based Configuration of 5G Networks: Pipeline and Integration of Commercial 5G System

Steffen Fricke[1] und Jürgen Jasperneite[1,2]

Abstract:

Wireless communication networks form the backbone of today's communication infrastructure. Increasing heterogeneity and complexity, particularly in 5G, Time-Sensitive Networking (TSN), and emerging Sixth Generation Wireless Networks (6G) systems, make manual network configuration inefficient and error-prone, while decreasing expert availability increases the need for intelligent automation. Large Language Models (LLMs) have demonstrated significant potential across various domains and are therefore promising for network management and configuration. This paper investigates the current state, trends, and limitations of Retrieval Augmented Generation (RAG) and fine-tuning techniques in telecommunications. Based on these findings, a comprehensive concept for Artificial Intelligence (AI)-assisted 5G configuration combining local LLMs, a RAG system for contextual knowledge, and a verification component for validating the syntax of the generated configurations is presented. The proposed pipeline is evaluated using 500 test prompts covering different difficulty levels and parameter settings. The results provide insights into the capabilities and limitations of LLM-based approaches for automated 5G network configuration.

Keywords: 5G, Network Engineering, LLM, AI

1 Introduction

Wireless communication networks are a key component of modern communication infrastructures. Traditional manual configuration approaches, together with currently available management tools, are increasingly reaching their limits due to growing system complexity, high susceptibility to configuration errors, and the declining availability of expert knowledge. These challenges highlight the need for intelligent and automated approaches to network configuration.

Recent research has investigated the use of well-known LLMs, such as GPT and Llama, for 5G network management and configuration. However, many existing approaches rely on simulations or Q&A tasks, and do not address the complete configuration process of real-world 5G systems, including the translation of high-level user intents into executable low-level network configurations and their subsequent validation and execution.

This paper presents an extensive concept for LLM-based 5G system configuration that addresses this gap. The proposed approach uses locally deployed models trained on technical data to translate high-level user intents, provided through a web-based user interface, into low-level network configurations. A RAG system based on Qdrant provides contextual knowledge about the underlying commercial 5G system, MECSware, while an Application Programming Interface (API) enables interaction with the system. To prevent unintended

1 TH OWL Institute Industrial IT - inIT, Lemgo, Germany, steffen.fricke@th-owl.de, juergen.jasperneite@th-owl.de,

2 Fraunhofer IOSB-INA, Lemgo, Germany, juergen.jasperneite@iosb-ina.fraunhofer.de,

system behavior, a formal verification component is indispensable and validates the syntax of the generated curl commands before execution. The validated commands are subsequently executed on a real 5G system.
The proposed pipeline is evaluated using 500 test prompts covering different difficulty levels and evaluates the performance of different models Qwen3-8B-Instruct, Gpt-oss, and phi3-mini-4k-instruct. The evaluation provides insights into the performance and limitations of the proposed approach for automated 5G network configuration.
The main contributions of this work are:

- **Literature Review:** An analysis of the current state of research on automated network configuration and RAG-based retrieval approaches, identifying existing methods, trends, and research gaps
- **Concept for 5G Configuration:** A concept for automated configuration of the MECSware 5G system, including configuration verification and validation
- **Evaluation:** A comprehensive evaluation, as well as the execution performance on real 5G hardware, using a dataset of 500 test prompts

The remainder of this work is structured as follows: Section 2 describes the related work regarding the practical use of LLMs in Network Engineering. After that, Section 3 presents the extended concept for the configuration of a 5G system. Furthermore, Section 4 presents the evaluation methodology and the test setup. Section 5 shows the evaluation results. The work concludes with a conclusion and an outlook in Section 6.

2 Related Work

The work [12] presents a theoretical and high-level concept for the configuration of a hybrid 5G/TSN network using AI. The concept consists of a user-interface, a RAG-system, an orchestrator, and a fine-tuned LLM. Additionally, the concept includes a verification component to check the syntax and the semantic to avoid unintended system behaviour. In this approach, the practical implementation, as well as the evaluation and the definition of practical components is missing.
Furthermore, the publication [6] presents a survey regarding the state of the art, trends and challenges of using LLMs in Network Engineering for the technologies 5G, TSN, and 6G. The author explains, that there is substantial body in 5G for the different categories network configuration, network analysis, and network fault management. Limitations of the current work are that most of the approaches are simulation-based, and often well known LLMs are used instead of fine-tuned specialized LLMs.
The work [7] outlines, that most of the current work using RAG and fine-tuning approaches are used in telecommunication Q&A systems, which is not sufficient for the configuration of real 5G systems. The author addresses the mentioned issues by presenting a work in progress pipeline for the configuration of a 5G network. The concept consists of a web-based user interface using Prime React, a flask API, and the framework Qdrant for the implementation of a RAG system.
In the approach, specialized, on technical data trained models like Qwen-3-8B-Instruct and

Phi-2 are used for a first feasibility evaluation. In this evaluation, a small test dataset of 50 prompts for the configuration of the commercial 5G-system MECSware is used. As a result the generated output and the ground truth are compared, calculating the metrics Precision and BLEU. The results visualize the improvements of using a RAG-system, espacially for the model Qwen-3-8B-Instruct. The evaluations shows that the concept is a promising baseline with a general understanding of the generation of curl commands paving the way for future research on refining the RAG pipeline to ensure reliable compliance and robustness. In this approach, the verification of the generated output, the transfer of the commands to the hardware and the fine-tuning and the iterative refinement is missing.
In contrast, the publication [4] proposes a different approach to the configuration of 5G systems without using AI. The presented concept, 5GAutoConf, uses the 5G system Open Air Interface to enable rapid reconfiguration of 5G systems. It provides an automated 5G base station configuration tool that requires only a set of basic input parameters. The configuration process follows an automated, data-driven approach that is compliant with 5G standards and generates lookup tables. This approach could complement an RAG-based configuration pipeline by providing a formally derived set of valid configuration options and enabling the verification of generated commands against these configurations.

2.1 Practical implementations regarding real-world system implementation

Real-world 5G systems are typically controlled via APIs. The authors of [8] proposed a method by processing Network Exposure Function (NEF) and Representational State Transfer (REST) APIs and generate a dataset for fine-tuning the system. GPT-4 is used to generate a dataset of 700 valid configurations. These are used to fine-tune a model Phi-2 using Quantized-Low-Rank Adaptation.
In addition, the publications [3] and [2] explore methods to improve the efficiency of RAG systems and to reduce hallucinations. In particular, [3] addresses single-pass performance by introducing strategic context reordering to mitigate the “lost-in-the-middle” problem, where LLMs tend to pay less attention to information located in the middle of the input. By placing the most relevant passages at the beginning and the query at the end, retrieval effectiveness can be improved especially for small models.
The publication [5] presents a chatbot for the automated configuration of a 5G modem on the device side, translating high-level user intents into low-level Attention (AT) and modem-specific commands. The system employs a RAG approach based on ChromaDB to store relevant documentation, including information on AT commands and their configuration procedures. For the verification regular expressions are used.
The work [9] compares traditional RAG with Agentic RAG. Traditional RAG follows a single-pass retrieval and generation pipeline that integrates domain-specific knowledge into LLMs at low inference latency, but its static design cannot evaluate or iteratively improve intermediate results. Agentic RAG addresses this by adding external tools and iterative refinement, allowing the system to dynamically evaluate and adapt its retrieval and generation process.

3 Concept

As described in the Chapter 2, the work [7] proposes a RAG based pipeline for the configuration of the 5G system MECSware and shows a feasibilty evaluation. In the previous concept [7], the web-based user interface for visualizing the system state and displaying user input and output, the Flask API for bidirectional communication between the backend and frontend, the RAG system Qdrant for managing different collections, and the time-series database InfluxDB for snapshot of the system for further evaluation were already implemented. Figure 1 visualizes the main components required to implement the extended concept of configuring a 5G network using AI. Already deployed components are visualized in grey, not yet implemented components are visualized with a dashed border.

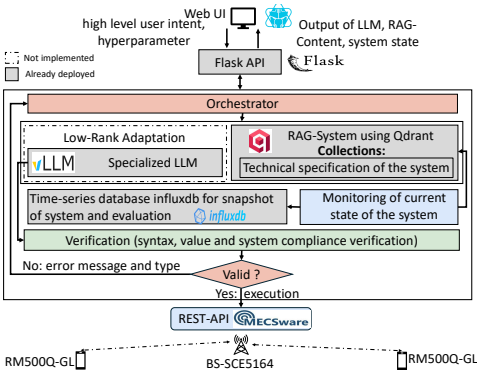


Fig. 1: Schematic overview of the main system components of the conceptual framework

The main improvements of the concept are:

- Enhancement of the system prompt
- Implementation of a Verification Component to validate curl command syntax
- Refinement loop to improve the quality of the generated curl commands
- Execution of the generated curl command on real 5G systems MECSware

```
"curl -X PUT https://172.19.22.34:6327/5g/bs/conf -k -u admin:admin -d '"Name":"BS2","Frequency":"3750",
"Ratio": "4:1" ' -H 'Content-Type: application/json'"
```

List. 1: Structure of a curl command

HTTP Method Endpoint URL Credentials Body Header

At first, the system prompt is extended by providing additional information about requirements, the general structure of curl command and improved instructions. Addtionally, the concept is extended by a verification component that validates the syntax and structure of the generated `curl` commands. Listing 1 illustrates the structure of a `curl` command used to configure the MECSware base station. The command contains the Hypertext Transfer Protocol (HTTP) method, the Uniform Resource Locator (URL) and path, authentication information, request body, and headers.

First, the verification component checks the syntax and structure of the generated command. Subsequently, the specified parameter values and their valid ranges are verified. Before executing the valid command on the physical hardware, the requirements and current system state is additionally compared and validated. This ensures that the generated command is compatible with the current configuration and that the required base station and user equipment (UE) are available.

Furthermore, the concept will be extended by a refinement loop. In the event of a failed

verification, the validation component provides feedback consisting of an error message and an error type indicating the part of the `curl` command in which the error occurred. This feedback is then added to the prompt and provided to the AI model which allows the model to iteratively refine the generated `curl` command and thereby improve its correctness and quality.
Additionally, correctly validated commands will be transmitted to the MECSware system for execution on the physical hardware. The experimental setup consists of an indoor base station of type SCE5164 and a connected RM500 modem manufactured by Quectel.

4 Evaluation methodology

This section describes the evaluation setup, including the test environment, the models used, the structure of the dataset, and the evaluation metrics.

4.1 Test setup

The proposed approach is evaluated in the industrial environment of the SmartFactoryOWL laboratory at Technische Hochschule Ostwestfalen-Lippe using the MECSware SCE5164 indoor base station and a Quectel RM500Q-GL UE modem operating within the campus network frequency range, with the test setup illustrated in Figure 2.



Fig. 2: Test setup for the evaluation

The system control and management are handled via the MECSware API, which provides a total of 14 core commands for administering base stations and UEs. Through this interface, all key operations like adding, deleting, modifying, and querying system status can be performed under both Linux and Windows. The configuration workflow is comparable to managing a conventional router and is fully documented in the system user guide. To enable efficient, AI-driven retrieval of technical details, this manual is stored and indexed in the Qdrant vector database using semantic chunking. The models are deployed using vLLM on an NVIDIA A100 GPU with 80 GB of VRAM.

4.2 Models

Tab. 1: Overview of the evaluated LLMs and their key properties

LLM	Developer	Parameters	Context window	Embedding Model	Key Strength
Qwen-3-8B-Instruct [11]	Alibaba Cloud	8.19 billion	128,000 tokens	Qwen3-Embedding-0.6B	Integrated "Thinking Mode"
Phi-3-mini-4k-instruct [1]	Microsoft	3.8 billion	4,000 tokens	all-MiniLM-L6-v2	Strong reasoning per parameter
Gpt-oss-20b [10]	OpenAI	21 billion (3.6 billion active, MoE)	131,072 tokens	all-MiniLM-L6-v2	Configurable reasoning effort

This Section shows in Table 1 the key properties of the evaluated LLMs Qwen-3-8B-Instruct, as well as those of the comparable model Gpt-oss and the smaller variant Phi-3-mini-4k-instruct.

4.3 Dataset

The evaluation is based on a dataset of 500 test prompts. Each prompt is assigned a *difficulty level*, which can be *Low*, *Moderate*, or *High*. Table 2 describes these levels in detail. The dataset distribution is visualized in Figure 3 as a stacked bar chart, faceted by severity level (Low, Moderate, High) and operating system (Linux, Windows), with each bar further segmented into three categories: 5G BS Configuration, Base Station Management, and User Equipment. As Linux is the primary target platform of the analysis, it is more frequently represented in the dataset than Windows. Nevertheless, both operating systems are included to cover the full range of API capabilities and enable comprehensive evaluation across platforms. The majority of data points in the dataset are classified as Low severity, with the analysis focusing primarily on the Base Station Management and Configuration categories. To construct the dataset, a set of representative seed examples was initially created manually for each category and difficulty level. These examples served as the basis for expanding the dataset using an LLM-based generation process using Claude Sonnet. After the generation process, the complete dataset was manually reviewed to verify the correctness and consistency of all generated instances. Incorrect or inconsistent examples were revised or removed before the dataset was used for subsequent evaluation. The structure of one datapoint is shown in Listing 2 and contains the prompt and the ground truth but also and ID, a category, a platform and a level of difficulty.

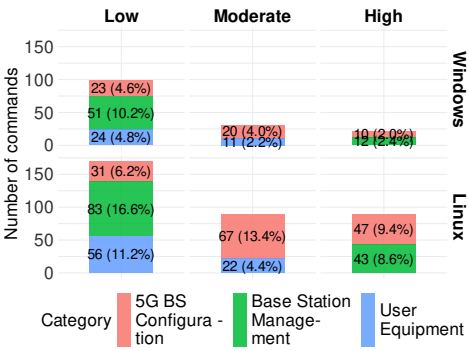


Fig. 3: Distribution of evaluation dataset

Tab. 2: Complexity categories with description and example prompt

Category	Description	Example Prompt
Low	Easy use case by directly defining all relevant parameters in the prompt.	*Create the Windows command to change the frequency of 'Hall_1' to '3730' MHz.*
Moderate	Combination of parameters and the conversion of units like kHz to MHz or mW to dBm.	*Configure base station 'Server_Room_BS' using a Linux curl command at 3,740,000 kHz, 20 MHz bandwidth, and 10 mW transmission power.*
High	Complex parameter combinations requiring knowledge of spectrum ranges, bandwidth and transmit power, including edge cases for requirement verification.	*Configure base station 'BS' using a Windows curl command. The operator allocated a spectrum slice starting at f_low = 3.750 GHz and ending at f_high = 3.79 GHz. Calculate the center frequency and the channel bandwidth in MHz. Set the TDD ratio to 1:1.*


```
"id": 1,
"category": "Base Station Management - Add",
"platform": "Linux",
"level_of_difficulty": "low",
"prompt": "Create a curl command for Linux to add a new base station named 'Halle1', type 'SCE5164', IP
      '10.0.2.110', and MAC address '3C:62:F0:17:24:AA'. The description should be 'Produktionshalle
      1'.",
"ground_truth": "curl -X POST https://172.19.22.34:6327/5g/bs/conf -k -u admin:admin -d '{\"Name\": \"
      Halle1\", \"Type\": \"SCE5164\", \"Description\": \"Produktionshalle 1\", \"IP\": \"10.0.2.110\",
      \"MAC\": \"3C:62:F0:17:24:AA\"}' -H 'Content-Type: application/json'"
```


List. 2: Structure of a single evaluation data point

4.4 Evaluation Metrics

For the evaluation of the correctness of the predicted commands, the following metrics in Table 3 are used. The selected metrics capture different aspects of the generated output, including the correctness and completeness of key-value pairs, textual similarity to the ground-truth command, and semantic similarity between the generated and reference outputs. In addition, Generation Time is measured to evaluate the computational performance to generate the curl configurations.

Metric	Formula	Description
Precision	$\frac{\lvert KV_{\text{gen}} \cap KV_{\text{gt}} \rvert}{\lvert KV_{\text{gen}} \rvert}$	Fraction of correct KV pairs among all generated pairs
Recall	$\frac{\lvert KV_{\text{gen}} \cap KV_{\text{gt}} \rvert}{\lvert KV_{\text{gt}} \rvert}$	Fraction of correct KV pairs among all ground-truth pairs
F1-Score	$2 \cdot \frac{P \cdot R}{P + R}$	Harmonic mean of precision and recall
ROUGE-1	$\frac{\lvert U_R \cap U_C \rvert}{\lvert U_R \rvert}$	Unigram overlap between reference (U_R) and candidate (U_C) text
ROUGE-L	$\frac{\text{LCS(Ref, Cand)}}{\text{Length(Ref)}}$	Longest common subsequence, sentence-structure aware
BLEU	$\text{BP} \cdot \exp\left(\sum_n w_n \log p_n\right)$	N-gram overlap with brevity penalty
Cosine Similarity	$\frac{\vec{e}_{\text{gen}} \cdot \vec{e}_{\text{gt}}}{\lVert \vec{e}_{\text{gen}} \rVert \, \lVert \vec{e}_{\text{gt}} \rVert}$	Semantic similarity between embedding vectors of generated and ground-truth text
Generation Time	$t_{\text{end}} - t_{\text{start}}$	Time to generate the curl command

Tab. 3: Evaluation metrics used to assess the correctness of the predicted commands

5 Evaluation Results

This Section presents the evaluation results and visualizes the impact of the improved system prompt and the evaluation metrics, comparing the different models with and without RAG. It further covers the effect of the refinement loop, the time required to generate the configuration and the success of command execution.

5.1 Impact of improved system prompt

As a contribution of the concept, the system prompt is improved by providing additional information about the requirements, the general structure of a curl command, and refined instructions. Figure 4 shows the number of correctly generated commands for both system prompts, the old system prompt from [7] and the improved version for the three models, comparing the configurations without RAG and with RAG. The left side shows the results for the old system prompt, the right side those for the new one. With the old system prompt and without RAG, none of the models generated a successfull configuration. With the new system prompt, 20 % of the commands generated by Qwen3 are successful. RAG generally improves the quality of the generated output, and for Gpt-oss the new system prompt yields an improvement of 9.6 %.

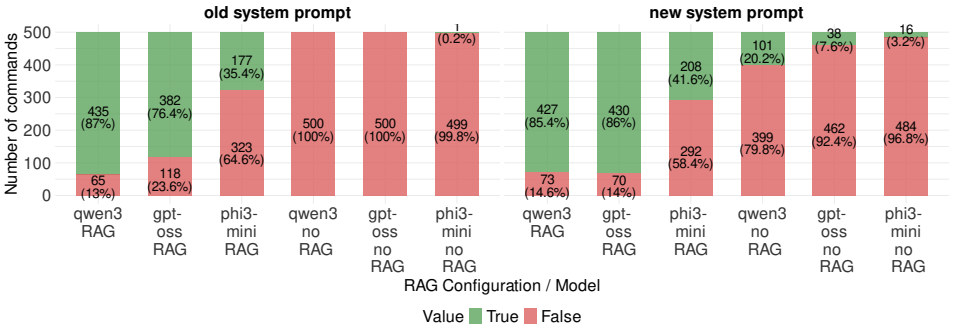


Fig. 4: Comparison of three different models and impact of improved system prompt

Table 4 lists the mean ROUGE-1 scores and their standard deviations for the three models with the old and the improved system prompt, each without and with RAG. ROUGE-1 measures the unigram overlap, i.e., the proportion of single words that match between the generated output and the reference. The improved system prompt increases the score for all models and configurations.

Tab. 4: Mean ROUGE-1 scores ± standard deviation for the old and the improved system prompt

System Prompt / RAG configuration	Qwen no RAG	Qwen RAG	Phi3 no RAG	Phi3 RAG	Gpt-oss no RAG	Gpt-oss RAG
Old system prompt	0.611 ± 0.141	0.767 ± 0.153	0.477 ± 0.271	0.562 ± 0.365	0.0508 ±0.182	0.646 ± 0.311
New system prompt	0.716 ± 0.408	0.967 ± 0.0981	0.538 ± 0.339	0.720 ± 0.248	0.220 ± 0.402	0.897 ± 0.235

5.2 Metric-Based Evaluation

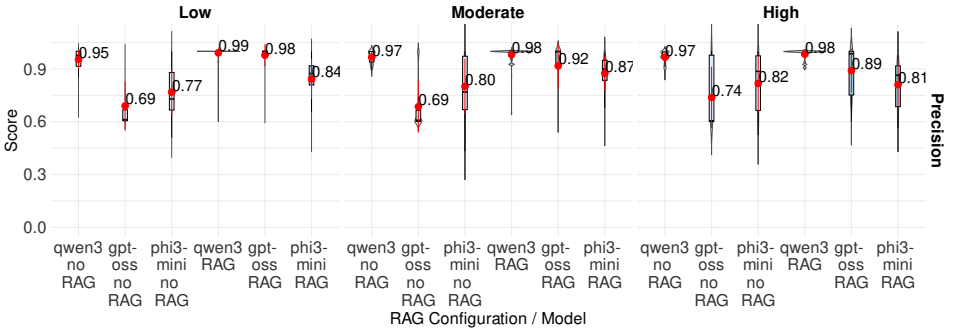


Fig. 5: Comparison of three different models and impact of RAG on metrics devided by level of difficulty

This Section describes the metrics defined in Chapter 4.4 for the three models and compares the two strategies, RAG and no RAG. Figure 5 shows the metric precision, separated by level of difficulty (low, moderate, and high). The results show the improvements achieved by RAG across the difficulty levels. Table 5 compares the mean and standard deviation of

the metrics defined in Chapter 4.4 for the three models and the two strategies, RAG and no RAG.

Tab. 5: Comparison of evaluation metrics across three models with and without RAG

Use Case / Metric	F1-Score	Recall	Precison	BLEU	Rouge-L	Rouge-1	Cosine similarity
Qwen no RAG	0.855 ± 0.173	0.838 ± 0.193	0.875 ± 0.150	0.562 ± 0.365	0.699 ± 0.400	0.716 ± 0.408	0.833 ± 0.224
Qwen RAG	0.973 ± 0.0578	0.974 ± 0.0546	0.972 ± 0.0630	0.879 ± 0.213	0.964 ± 0.100	0.967 ± 0.0981	0.985 ± 0.0425
Gpt-oss no RAG	0.644 ± 0.169	0.605 ± 0.189	0.692 ± 0.146	0.179 ± 0.342	0.214 ± 0.393	0.220 ± 0.402	0.390 ± 0.322
Gpt-oss RAG	0.932 ± 0.114	0.930 ± 0.120	0.936 ± 0.108	0.790 ± 0.292	0.894 ± 0.236	0.897 ± 0.235	0.949 ± 0.160
Phi-3 no RAG	0.783 ± 0.146	0.959 ± 0.0818	0.780 ± 0.143	0.402 ± 0.322	0.525 ± 0.334	0.538 ± 0.339	0.768 ± 0.255
Phi-3 RAG	0.841 ± 0.0997	0.789 ± 0.153	0.839 ± 0.114	0.460 ± 0.244	0.649 ± 0.238	0.720 ± 0.248	0.833 ± 0.149

5.3 Impact of refinement loop

The concept further includes a refinement loop that feeds the output of the verification component back to the model as feedback on incorrect commands, so that the output is refined iteratively. Figure 6 shows the number of correctly generated commands for the strategies RAG and RAG with refinement. RAG improves the results compared to the baseline. Qwen3 and Gpt-oss achieve comparable results with 85.4 % and 86.0 % correctly generated commands, respectively, whereas the smaller model Phi-3 reaches only 38 %. With the refinement loop, the share of correct commands for Qwen3 increases from 85.4 % to 92.8 %, and the largest improvement is achieved for Phi-3, which rises from 38 % to 78 %. Figure 7 shows the error categories identified by the verification component for both strategies. The number of errors decreases from RAG to RAG with refinement. This indicates that the refinement loop is especially beneficial for smaller models.

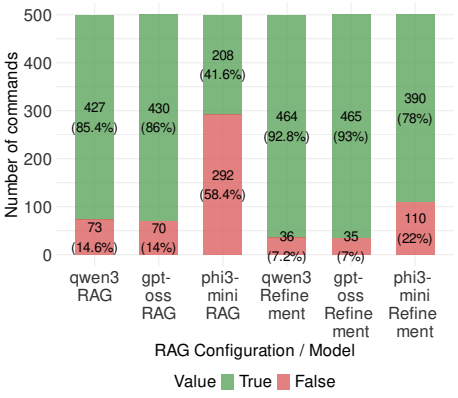


Fig. 6: Impact of RAG and refinement on command correctness across models

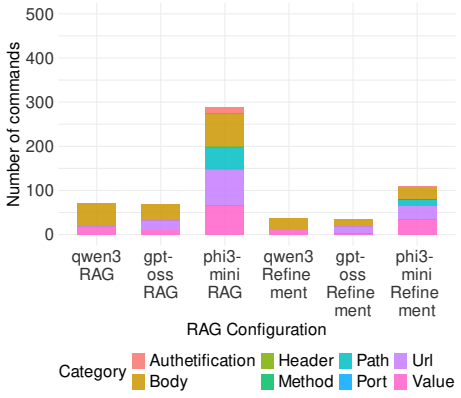


Fig. 7: Effect of RAG and refinement on error categories of generated commands

5.4 Generation time to curl configuration

Figure 8 shows the generation time required to generate the curl command for each model and difficulty level. Phi-3 and Qwen3 have a low generation time of approximately 1 s, which remains constant across all difficulty levels. In contrast, the generation time of Gpt-oss increases with the difficulty level, from 1.99 s for Low to 3.68 s for High. This increase is likely caused by the additional reasoning tokens that the model produces for more complex prompts. Nevertheless, all models generate commands within a few seconds, which is sufficient for interactive use.

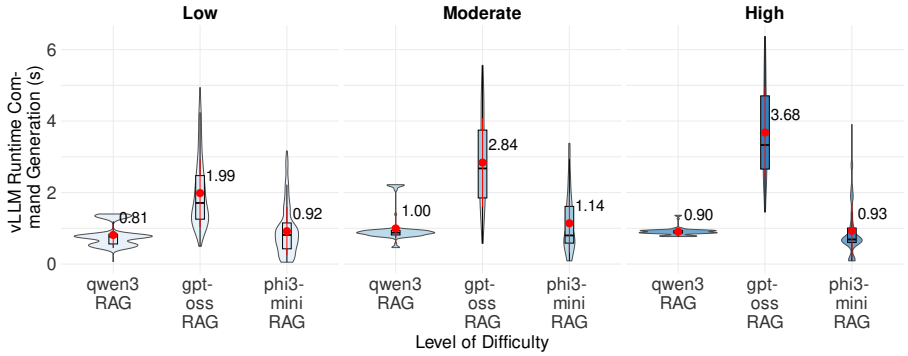
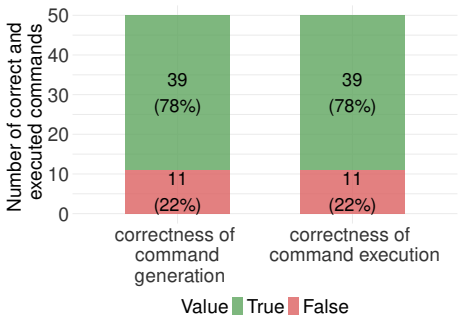


Fig. 8: Command generation time of the three models per difficulty level

5.5 Evaluation of command execution

As a further improvement, the commands are executed on real 5G hardware. This Section presents a positive test, showing that verified commands generated by Qwen 3 can be executed successfully on the MECSware 5G system. In addition, a negative test demonstrates the impact of the verification component, which checks the syntax of each command and, before execution, verifies that the command matches the current system state.

5.5.1 Positive Tests: Execution of verified commands



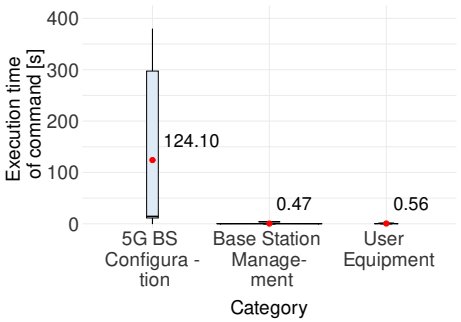


Fig. 9: Comparison of correct commands and Fig. 10: Distribution of execution time for the successfull executed commands for positive tests three different categories

Figure 9 shows the number of correctly generated and successfully executed commands. Of the 50 test prompts, 39 commands were generated correctly and passed the verification

component. All valid commands were executed successfully on real hardware. Figure 10 shows the execution time per use case: 5G Base Station Configuration, Base Station Management, and User Equipment. The execution time for Base Station Management and UE commands is low at approximately 0.5 s, whereas 5G BS Configuration has a mean execution time of 124 s. This considerably higher value is caused by the system reboot required when configuring cell parameters, and is therefore a hardware-related delay rather than an overhead of the proposed system.

5.5.2 Negative Tests: Rejection of invalid commands

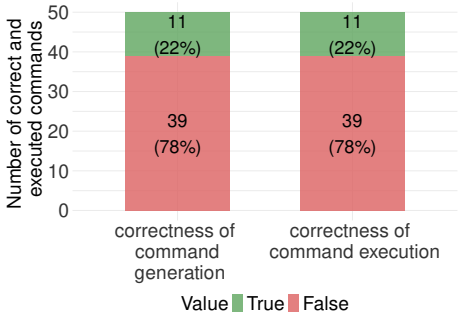


Fig. 11: Comparison of correct commands and successful executed commands

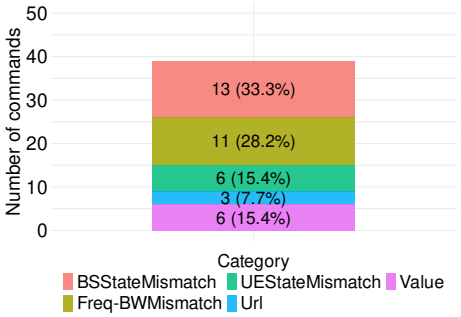


Fig. 12: Distribution of error categories for negative testing

In addition to the positive tests, negative tests were performed using 50 test prompts containing partly invalid or non-existing base stations, UEs, and configurations that do not match the current system state. Figure 11 shows the number of correctly generated commands and the number of successfully executed commands. As expected, the majority of the commands were rejected by the verification component and therefore not executed. Figure 12 shows the distribution of the error categories. The verification component detects syntax and value errors, checks the availability of base stations and UEs, and validates the commands against the current system state. This demonstrates that invalid requests are reliably prevented from reaching the system.

6 Conclusion and Outlook

In conclusion, this work presents a concept for the automated configuration of the MECSware commercial 5G system using LLMs. The proposed approach extends an existing concept by improving the system prompt, implementing a verification component, enabling the execution of generated `curl` commands on a real 5G system, and introducing a refinement loop to improve the quality of the generated output. To evaluate the proposed pipeline, an extensive dataset of 500 test prompts covering different levels of difficulty and parameter settings was used to assess the performance of the three models Qwen3-8B-Instruct, phi3-mini-4k-instruct, and Gpt-oss in generating `curl` commands for configuring the MECSware 5G system.

The evaluation shows that incorporating RAG and improving the system prompt substantially

improves the generation of correct `curl` commands compared to the baseline approach. Additionally, it can be visualized, that the refinement loop by using the feedback of the verification component can improve the quality of the generated output. Furthermore, the generated commands are executed on real hardware by positive test and showing by negative tests, that the verification component prevents the execution of errorneous commands.

Further Work: Several directions remain for future research. First, the proposed approach could be extended to support the configuration of UE modems in addition to base stations. This would enable the configuration pipeline to cover both sides of the 5G communication system. Building upon this, a combined configuration of base stations and UE modems could be investigated to enable end-to-end configuration of complete 5G scenarios. Second, fine-tuning of the underlying LLMs on domain-specific network configuration data could be investigated to further improve the generation of valid and system-specific commands. Finally, the approach should be evaluated on additional 5G systems and implementations to assess its generalizability and identify system-specific limitations.

References

[1] Marah Abdin, Sam Ade Jacobs u. a. „Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone“. In: *arXiv preprint arXiv:2404.14219* (2024).

[2] Samar AboulEla u. a. „Exploring RAG Solutions to Reduce Hallucinations in LLMs“. In: *2025 IEEE International systems Conference (SysCon)*. 2025, S. 1–8.

[3] Yufei Chen, Zhiliang Huang und Youyan Zhang. „Efficient-RAG: Competitive Single-Pass Performance in RAG via Strategic Context Reordering“. In: *2025 18th International Conference on Advanced Computer Theory and Engineering (ICACTE)*. 2025, S. 188–194.

[4] Niels Hendrik Fliedner, Florian Klingler und Henning Trsek. „5GAutoConf: A Rapid Data-Driven Autoconfiguration Tool for 5G Base Stations“. In: *2026 22nd International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*. 2026, S. 1325–1328.

[5] Mattia Fontana u. a. „Leveraging LLM-Powered Intelligent Chatbots for Intent-Based Networking in 5G Modem Reconfiguration“. In: *2025 IEEE 11th International Conference on Network Softwarization (NetSoft)*. 2025, S. 209–213.

[6] Steffen Fricke, Maxim Friesen und Jürgen Jasperneite. „LLMs for Network Engineering Automation: State of the Art and Research Directions“. In: *NOMS 2026-2026 IEEE Network Operations and Management Symposium*. 2026, S. 1–6.

[7] Steffen Fricke, Jie Wang und Jürgen Jasperneite. „Towards Automated 5G Configuration: Design and Evaluation of a RAG-Enhanced AI Pipeline“. In: *31st IEEE International Conference on Emerging Technologies and Factory Automation, Industrial Communication Technologies and Systems*. to appear. IEEE, Sep. 2026.

[8] Zainab Khan u. a. *NEFMind: Parameter-Efficient Fine-Tuning of Open-Source LLMs for Telecom APIs Automation*. 2025. arXiv: 2508.09240.

[9] Fnu Neha und Deepshikha Bhati. „Traditional RAG vs. Agentic RAG: A Comparative Study of Retrieval-Augmented Systems“. In: *2025 IEEE International Conference on Future Machine Learning and Data Science (FMLDS)*. 2025, S. 383–387. DOI: 10.1109/FMLDS67896.2025.00073.

[10] OpenAI. *gpt-oss-120b & gpt-oss-20b Model Card*. 2025. arXiv: 2508.10925 [cs.CL]. URL: https://arxiv.org/abs/2508.10925.

[11] Alibaba Cloud Qwen Team. *Qwen3: Think Deeper, Act Faster*. https://qwenlm.github.io/blog/qwen3/. Accessed: 5. Oktober 2026. 2025.

[12] Stefan Windmann u. a. „NetPilot - Towards LLM-Assisted Configuration of Hybrid TSN/5G Networks“. In: *2025 IEEE 30th International Conference on Emerging Technologies and Factory Automation (ETFA)*. 2025, S. 1–4.

Engineering Data Space – Generating Actionable Maintenance Recommendations from Heterogeneous Data

Joschka Kersting[1], Oliver von Heißen[2], Daniel Eckertz[2], and Gesa Benndorf[1]

Abstract: Generative AI offers potential to support engineers in complex tasks. However, effective application requires domain- and situation-specific context distributed across heterogeneous maintenance and engineering artifacts, operational data, and expert knowledge. Existing retrieval-augmented generation approaches focus on individual sources with limited support for integrating structural knowledge and dynamic operational information. This paper presents an engineering and maintenance data space approach for context-aware AI assistance integrating heterogeneous engineering information according to the current task. It combines operational and maintenance information with semantic system context from SysML v2 models. A folding-machine maintenance scenario illustrates how fault information, anomaly results when available, and structural context on affected components and relationships can be brought together in an assistant. The contribution is a prototypical architecture for providing operational and system-model context; diagnostic effectiveness has not yet been evaluated.

Keywords: Retrieval-Augmented Generation, Anomaly Detection, Predictive Maintenance, Multimodal Assistance Systems, Engineering Automation, SysMLv2

1 Introduction

Maintenance activities rely on information that is distributed across heterogeneous artifacts, data sources, and software systems throughout the lifecycle of technical systems. This information constitutes the **engineering context** of a task. In this work, engineering context refers to the information required to interpret an engineering situation and to derive appropriate actions or decisions. It comprises system context (such as requirements, functions, components and their relationships) as well as operational context (such as sensor data, machine states and historical information). Further context may originate from engineering processes, tools and IT systems, expert knowledge, previous decisions, regulatory constraints, and organizational structures. The relevance of such context depends on the specific engineering task and situation. Providing LLMs with task-relevant domain context can improve their ability to access manufacturing knowledge, while context-aware reduction can restrict the provided information to the knowledge relevant to a given query [Mo25]. Providing this context is particularly challenging in industrial maintenance, where relevant information is distributed across heterogeneous sources and experienced technicians often rely on implicit knowledge to identify and relate the information required for fault diagnosis.

[1] Fraunhofer IOSB-INA, Campusallee 1, 32657 Lemgo, Deutschland, joschka.kersting@iosb-ina.fraunhofer.de; gesa.benndorf@iosb-ina.fraunhofer.de

[2] Fraunhofer IEM, Zukunftsmeile 1, 33102 Paderborn, Deutschland, oliver.von.heissen@iem.fraunhofer.de; daniel.eckertz@iem.fraunhofer.de

At the same time, the German Federal Employment Agency classifies technical service and maintenance occupations at the specialist level as shortage occupations [Bu26], further increasing the need for systematic assistance for less experienced technicians.

Recent advances in **Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG)**, which grounds generated answers in retrieved documents [Le20], have enabled new forms of intelligent assistance systems that can retrieve and synthesize information from heterogeneous sources. However, an LLM does not inherently know the specific application situation in which it is used. It lacks knowledge about the particular technical system, its current state, the engineering artifacts available in an organization, and information or experience held by users or other experts. Consequently, the usefulness of an AI-based engineering assistant depends not only on the capabilities of the underlying language model, but also on its ability to access and combine the relevant engineering context for the task at hand.

Existing work in **industrial maintenance** typically addresses individual aspects of this context or isolated subproblems. Text-focused systems [Lu25] evaluate fault-code recommendation pipelines over technical manuals, hybrid approaches [HP25] combine vibration-based fault detection with manual and web retrieval, and anomaly detection methods [Pi20] identify statistical deviations in sensor data. While these approaches provide valuable information for individual aspects of fault diagnosis, engineering tasks often require information from multiple sources and perspectives to be considered jointly. In particular, the systematic integration of operational context, such as machine states, sensor data, anomalies, and historical information, with system context, such as components, functions, requirements, and their relationships, remains a challenge for LLM-based maintenance assistance.

To address this gap, we present an engineering data space approach for the provision and integration of heterogeneous context for Generative AI in engineering. It specifically integrates two source types, operational data and semantic system context from SysML v2 models, for the current task. The architecture is illustrated with an industrial maintenance scenario involving a folding machine from an industrial partner. It combines two complementary perspectives: operational context, including technical documentation, machine information, fault knowledge, expert knowledge, and operational data, and system context derived from SysML v2 engineering models, primarily components and their relationships. **Engineering context** is provided through the maintenance assistance solution developed at Fraunhofer IOSB-INA, while semantic **system context** is retrieved from SysML v2 models using the context extraction mechanisms developed at Fraunhofer IEM. By integrating both perspectives, the assistant can relate observations from the operating machine to knowledge about the engineered system.

This paper is guided by the following **research questions (RQ)**:

RQ1 How can operational context and semantic system context from engineering models be integrated and combined to support cross-source reasoning for Generative AI in industrial maintenance scenarios?

RQ2 What architectural mechanisms are required to dynamically retrieve and orchestrate context from heterogeneous engineering sources for LLM-based assistance?

The contribution of this paper is threefold: (1) we introduce an engineering data space approach for the task-specific provision and orchestration of heterogeneous engineering context for Generative AI; (2) we present two complementary mechanisms for providing operational context and semantic system context from heterogeneous sources, including operational data, technical documentation, expert knowledge, and SysML v2 engineering models; and (3) we illustrate how these mechanisms can provide context for an industrial maintenance scenario and discuss architectural trade-offs. Diagnostic effectiveness and the benefit of combining sources remain to be evaluated.

2 Related Work

Retrieval-Augmented Generation for Industrial Maintenance. Recent work has applied RAG to specific diagnostic tasks in industrial maintenance. PARAM [HP25] extends RAG with multi-agentic generation for prescriptive maintenance of rotating machinery, combining bearing vibration analysis with structured maintenance recommendations based on retrieved manuals and web search. An evaluation framework for fault diagnosis using technical manuals in RAG-based LLMs is presented in [Lu25], comparing different system configurations for fault-code recommendation.

While these approaches focus primarily on text-based retrieval from maintenance documentation, MindRAG [Lö25] introduces a modular agent framework that combines multimodal RAG with a semi-structured vector store designed for condition monitoring data. By integrating condition monitoring data, annotations, and maintenance work orders, MindRAG moves toward joint text-and-signal analysis through an explainable agent interface; a direction that our approach extends further by incorporating additional knowledge sources such as mapped OPC UA variable names, rule-based fault codes, and SysML v2 models within a single diagnostic architecture.

Semantic System Context from Engineering Models. Complementing the use of models for retrieval, von Heißen et al. [vo26] propose a methodology for semantic-aware change management in SysML v2. Their approach leverages LLMs to translate high-level natural-language requests into verifiable CRUD operations over the SysML v2 API, utilizing vector-based retrieval combined with scoped graph expansion to provide the necessary model context while minimizing token overhead. By implementing layered validation to ensure both syntactic conformance (e.g., [KB26]) and semantic intent preservation, their work demonstrates how LLMs can automate the evolution of complex systems engineering models.

LLM Agents for Fault Diagnosis and Decision Support. Beyond pure retrieval, LLM-based agentic systems have emerged for industrial fault diagnosis. LLM agents as Prognostics

and Health Management copilots [Lu24] introduce self-evaluation frameworks for maintenance recommendation. Recent work combines LLM planning agents with external causal inference models to rank candidate failure modes and interventions [Ki25], and assesses in simulation the feasibility of LLM agents diagnosing faults directly from raw sensor data [Le25]. A semi-automated knowledge graph construction pipeline designed as a foundation for GraphRAG-based retrieval in manufacturing maintenance has also been proposed [Zh25]. Our approach contributes to this line of work by orchestrating multiple specialized knowledge channels, each grounded in a different data source, within a unified architecture, rather than relying on a single retrieval pipeline.

Anomaly Detection and Knowledge-Driven Diagnostics. Statistical learning methods for industrial time-series data, such as control charts and supervised classifiers [Pi20], detect anomalous machine behavior. Recent work executes image-based anomaly detection models within an OPC UA server, using a state machine based on the OPC UA Machine Vision companion specification for near-real-time integration [VV24]. The COMETH service [ZB24] employed in the proposed approach uses active learning to iteratively learn and improve anomaly detection. While anomaly detection alone provides statistical alerts, the approach enriches these alerts with explanatory context from documentation, rule knowledge, and engineering models to transform raw anomaly signals into context for maintenance recommendations.

Tab. 1: Comparison of the proposed approach with representative related systems. Text RAG: retrieval from unstructured documents or records; Structured Rules: structured fault or rule knowledge (e.g., fault-code tables) used as a retrieval source; Time-Series: use of sensor data, derived features, or sensor-based alerts; Engineering Models: CAD/CAE or system models; Integrated Interface: one interactive user interface (e.g., chat) to the system's sources. ✓: described; (✓): partially or only in an evaluation setting; –: not described.

System	Text RAG	Structured Rules	Time-Series	Engineering Models	Integrated Interface
Trilla et al. [Tr24]	✓	–	–	–	–
MindRAG [Lö25]	✓	–	✓	–	✓
PARAM [HP25]	✓	–	✓	–	(✓)
MechRAG [LC25]	✓	–	–	✓	–
PHM Copilots [Lu24]	✓	–	✓	–	–
Lukens et al. [Lu25]	✓	✓	–	–	(✓)
Proposed approach	✓	✓	✓	✓	✓

Research Gap and Positioning. Table 1 compares the proposed approach with a selection of related systems along five architectural capabilities. The selected systems address different aspects of maintenance assistance, such as document retrieval, condition monitoring, and engineering model access. In this comparison, no individual system is shown to combine manual retrieval, structured fault-code lookup, sensor-derived anomaly results, and SysML v2 structural context. Our contribution is the architectural provision of these sources within one assistance interface; the table does not establish uniqueness across all existing work or a diagnostic benefit from combining the sources.

3 Context-Orchestrated Maintenance Assistance

The considered use case addresses fault diagnosis and maintenance of an industrial machine as described in Sec. 1. Transferring all available information into the assistant is neither feasible nor desirable, as it exceeds context limits and dilutes relevant information. Instead, the sources are accessed on demand and only the context relevant to the current task is provided to the assistant.

General Architecture. Figure 1 gives an overview of the approach. The technician working at the machine interacts with a single **chat interface**. It accepts free-form natural-language descriptions as well as structured inputs such as identifiers or fault codes, and presents answers together with references to the sources they are based on. Behind the interface, an **orchestration layer** manages the conversation, decides which sources are required for a given request, and assembles the retrieved information into the prompt of the LLM. Each source is connected through a dedicated **channel**. A channel encapsulates the access mechanism appropriate for its source, for instance semantic search over documents, deterministic lookup in structured tables, calls to analysis services, or model-based context extraction from SysML v2 models. Towards the orchestration layer, channels are designed to share a common interface: they receive the current request together with its conversational context and return task-relevant context fragments enriched with provenance metadata, such as document and page, data source, or model element. This allows channels to be enabled or disabled independently of each other and is intended to ease the integration of further sources. In this sense, the approach forms an engineering data space: heterogeneous, separately maintained sources are accessed on demand through source-specific channels, and both operational context (e.g., machine states, fault information, detected anomalies) and system context (e.g., components, functions, requirements, and their relationships) are accessible through one interface.

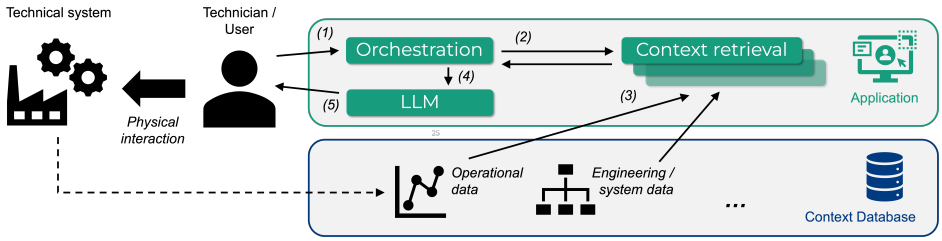


Fig. 1: Overview of the context-orchestrated maintenance assistance solution

General Workflow. Each conversational turn follows a recurring sequence. (1) *Request:* The user describes a problem or question through the chat interface, for example a reported fault or unexpected behavior of the technical system. (2) *Channel selection:* The orchestration layer interprets the request in the light of the conversation history and the descriptions of the available channels, and selects only those channels that are expected to contribute relevant context. (3) *Context retrieval:* Each selected channel queries its source using its own access mechanism and returns a task-specific excerpt instead of the complete source; a

model-based channel, for instance, identifies the model elements related to the problem and extracts their relevant neighborhood. (4) *Context assembly:* The returned fragments are combined with the request and the conversation history into a structured prompt in which the contributions of the individual channels remain distinguishable. (5) *Response generation:* The LLM generates an answer that relates the operational observations to the system context and presents it together with references grouped by channel, so that the user can trace the answer to its sources. Follow-up questions start a new turn, in which previously retrieved context can be reused and further channels can be consulted as the diagnosis progresses.

The approach thus separates context provision, performed by the channels, from context utilization, performed by the orchestration layer and the LLM. This allows information from the physical machine and its operation to be linked with knowledge originating from its engineering, without transferring complete data sources or engineering models into the assistant. The following subsections describe the prototype implementation with its knowledge sources and the channel for semantic system context from SysML v2 models, which is based on Semantic-Aware Change Management (SACM).

4 Prototype Implementation

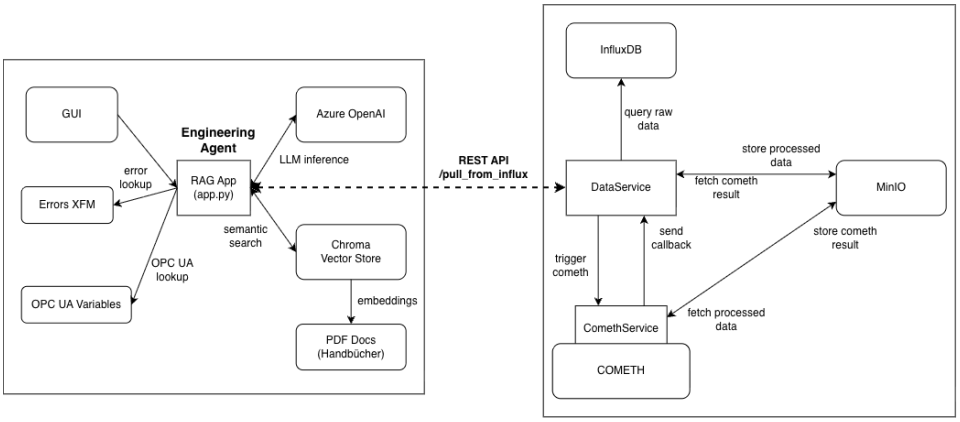


Fig. 2: Architecture of the proposed system.

The prototype, the concrete instantiation of the engineering data space approach, provides a unified chat-based diagnostic interface for the folding machine from an industrial partner. The system accepts structured inputs such as fault codes (e.g., `8251 208`) and OPC UA variable names (e.g., `eb_B31_Gurtstopp`) as well as free-form natural-language descriptions of machine problems. It returns proposed diagnostic actions with verifiable source references (document, page, data source, and model element) and supports optional voice input and output for hands-free operation at the machine. Figure 2 shows the architecture, which

consists of two main subsystems connected by a REST interface: a RAG application on the left and a data processing pipeline on the right.

4.1 Knowledge Sources and Retrieval

Tab. 2: Knowledge Sources used by the prototype

Knowledge Source	Type	Retrieval Method	Provided Metadata
Operating Manual	PDF / Vector Store	Semantic search	Document name, page reference
Fault Codes	Structured lookup table	Regex + deterministic lookup	Error description, root cause, remedy, required qualification
OPC UA Variables	Structured lookup table	Regex + deterministic lookup	Browse name, display name, node ID, data type, signal path
COMETH Anomaly Detection	Time-series + ML service	REST call + active learning	Anomaly time window, quality score, responsible signals, operator feedback
Expert Notes	Structured notes	Semantic search	Note content, priority, validity period, author
SysML v2 Model (via SACM)	Engineering model	Semantic-aware context extraction	Components and their relationships

Table 2 provides an overview of the six knowledge sources used by the prototype. The left side contains the RAG application, a web-based frontend that coordinates six knowledge sources. Fault codes and OPC UA variable names are extracted from user input via pattern matching and resolved against precompiled lookup tables. The OPC UA lookup provides variable metadata, not current sensor measurements. Semantically matching passages from the PDF operating manual are retrieved from a vector store and returned with document name and page reference (see Table 2 for details). Documents can be activated, deactivated, or updated incrementally without a full index rebuild, and the source PDF can be viewed or downloaded directly from the chat interface. The semantic search uses the user's problem description and can be enriched with a fault-code lookup result. OPC UA variable names are used directly when mentioned in the user input; they are not used to select or constrain subsequent retrieval steps.

4.2 Anomaly Detection Pipeline

The right side of the architecture contains the COMETH [ZB24] data processing pipeline. Sensor measurements from the machine's OPC UA interface are stored as time series, aggregated and normalized, and analyzed by the active-learning-based COMETH anomaly detection service. The prototype initiates analysis runs via a REST call (red dashed connection). COMETH results include anomaly time windows, quality scores, responsible

signals, and operator feedback (see Table 2); sensor-derived information reaches the assistant through these results, not through the OPC UA variable lookup.

4.3 Routing and Orchestration

A central design decision in the proposed approach is to avoid a static, predefined tool call pipeline. Instead, the orchestration layer is realized as a per-turn LLM-based router that receives the current question, the full conversation history, and the available tool definitions, and selects only the subset of knowledge channels relevant to the query. This approach lets the LLM assess the semantic intent of the question and is intended to limit irrelevant context in the answer. The routing logic separates channel selection from channel execution: each channel can be independently enabled or disabled through the sidebar, and two safety nets preserve conversational continuity when the router elects no tool. If COMETH results already exist in the current session, they are reused without launching a new analysis; otherwise, the document retrieval channel activates as a general fallback.

Operational experience such as known issues, maintenance hints, safety notices, and confirmed false positives can be captured in structured notes with priority levels, and optional validity dates. Notes are created manually, imported in batch, or proposed automatically by an LLM-based extraction that scans the chat history for reusable diagnostic patterns. At query time, semantically matching notes are injected into the LLM prompt with higher priority than generic documentation, and conflicts between expert notes and document content are explicitly reported to the user.

The router decision also governs the use of the SysML v2 model (see Section 4.4), which provides assembly part lists for a given sub-assembly and cross-component references indicating relations among parts in the machine. Access follows a two-tier strategy with a live REST endpoint preferred over a local model export, protected by a circuit breaker against repeated timeouts. When selected, COMETH, documentation, expert notes, and SysML provide complementary context to the assistant. The OPC UA variable-name lookup is used directly; its results do not trigger or narrow other retrievals.

The final LLM prompt assembles the question, conversation history, structured metadata, retrieved documents, expert notes, COMETH results, and SysML context in clearly delimited sections. The user-facing output separates the main answer from a collapsible source panel that groups references by channel: PDF pages, expert notes with confidence levels, COMETH anomaly details, and model elements. Voice output produces a short summary optimized for spoken delivery, generated independently of the full written response.

4.4 Semantic System Context via SACM and SysML v2

To provide a formal semantic context for the machine's structure and behavior, the proposed approach integrates the approach for Semantic-Aware Change Management (SACM) [vo26]. While the primary goal of SACM is the automation of model evolution, its underlying mechanism for context-aware retrieval is leveraged to resolve complex structural queries.

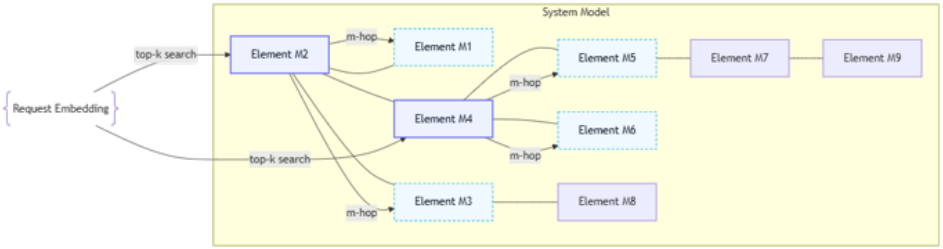


Fig. 3: Workflow of the Semantic-Aware Change Management (SACM) approach [vo26].

The core of the SACM retrieval strategy is a two-stage process designed to balance precision with token efficiency (Figure 3). First, a vector-based retrieval identifies the most relevant starting elements within the SysML v2 model based on the semantic similarity between the user query and the model elements' metadata. To overcome the limitations of purely associative retrieval, which may miss critical structural dependencies, the system then performs a scoped graph expansion. This expansion traverses the model's relationships (e.g., compositions, allocations, and refinements) to include neighboring elements that provide the necessary architectural context. This precise retrieval strategy is used to gain the context of the system model as part of the overall maintenance assistance.

5 Case Study: Prototype on an Industrial Folding Machine

The engineering data space architecture is instantiated as a diagnostic assistant for a folding machine from an industrial partner, deployed at the Fraunhofer IOSB-INA application center. The prototype integrates six knowledge sources listed in Table 2 specific to this machine.

Example Scenario. A technician observes fault code `2128 100` ("conveyor blocked") together with unusual vibrations. He enters both the fault code and a short natural-language description: "*Fault `2128 100` – conveyor blocked, plus strong vibrations at the infeed.*" The following outlines possible channel contributions for this illustrative query; it is not a recorded execution trace:

- The fault-code lookup can resolve `2128 100`, and the manual search can retrieve relevant passages.
- If selected by the router, matching expert notes add site-specific information.

- If selected by the router, SysML v2 context extraction provides information on related assembly parts and their relationships.
- If selected by the router, COMETH supplies anomaly information from results already in the session or from a newly requested analysis.

Where these results are available, the assistant can combine fault and maintenance information with structural context and detected anomalies in a response with source references. The example illustrates how complementary sources can contribute to the same diagnostic question; it does not establish diagnostic effectiveness.

The machine operating manual comprises approximately 200 pages with functional descriptions, maintenance procedures, and a fault-code catalog, which is additionally extracted into a structured lookup table. The machine exposes approximately 150 OPC UA variables, whose metadata is covered by a lookup table; sensor time series are analyzed by COMETH. A SysML v2 model provides assembly context and cross-component references (see Table 2).

6 Discussion

The proposed architecture addresses RQ1 at the level of context provision: it adds operational information, technical documentation, expert knowledge, and SysML v2 models. RQ2 is addressed by the per-turn LLM-based router for channel selection. Whether this context improves cross-source reasoning or diagnostic quality remains to be evaluated.

The per-turn LLM based router is a central design decision in the proposed approach. It enables flexible channel selection that adapts to the semantic intent of each query, rather than following a predefined pipeline. The primary benefit is coverage: queries that do not contain an explicit fault code or variable name still receive useful diagnostic context through semantic search over the manual. The risk is unpredictability: the router may fail to activate a relevant channel or may activate irrelevant channels that dilute the LLM prompt with noise. The current design prioritizes flexibility over determinism, and whether this trade-off is justified in practice requires empirical validation.

Expert notes introduce a second trade-off. They allow site specific knowledge to override generic documentation, preventing unnecessary maintenance actions when a known false positive is present, for instance. However, notes require curation, and stale or incorrect notes can actively harm diagnostic quality. The automated extraction of notes from chat history reduces manual entry effort but introduces a quality risk that warrants further investigation. Some channels exhibit partial overlap. Fault codes may appear both in the structured lookup table and in semantically retrieved manual passages, leading to redundant content in the LLM prompt. The router currently does not deduplicate across channels, and whether this redundancy degrades answer quality is an open question.

The current work has several limitations. The approach has been instantiated for a single machine type, an industrial folding machine. Generalizability to machines with different data modalities, such as vision based inspection or acoustic monitoring, has not been demonstrated. The COMETH service requires a seed of labeled data, but no anomalies need to be present in the dataset; the active learning loop introduces latency. SysML v2 model quality is assumed, but in practice models age as machines are modified and model maintenance must be integrated into operational workflows. Diagnostic quality, source synergy, and router accuracy have not yet been empirically evaluated, and the contributions are architectural and design contributions pending empirical validation.

7 Conclusion and Future Work

The main architectural contribution of this paper is a per-turn LLM-based router that selects knowledge channels dynamically based on query semantics, conversation history, and channel definitions, avoiding a rigid pipeline while preserving source transparency through structured provenance information. The diagnostic benefit of the assembled context remains to be evaluated.

Future work proceeds in three directions. First, a controlled laboratory evaluation with technically qualified participants and synthetic fault scenarios will provide empirical evidence on diagnostic quality, source synergy, and router accuracy. Second, the source spectrum can be extended to additional data modalities, in particular CAD data for visual component identification, and to Graph RAG based retrieval over maintenance knowledge graphs [Zh25] that formalize relationships between error codes, components, and root causes. Third, a field study in a production environment is needed to assess long term usability, retrieval drift, and the practical maintenance effort for the knowledge base.

Declaration of generative AI and AI-assisted technologies in the writing process. During the preparation of this work the authors used OpenAI ChatGPT to reword and rephrase text. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

[Bu26] Bundesagentur für Arbeit: Berichte: Blickpunkt Arbeitsmarkt – Fachkräfteengpassanalyse 2025, tech. rep., Tab. 8, S. 13, Nürnberg: Bundesagentur für Arbeit, 2026, https://statistik.arbeitsagentur.de/Statistikdaten/Detail/202512/arbeitsmarktberichte/fachkraefte-engpassanalyse/fachkraefte-engpassanalyse-dl-0-202512-zip.zip?__blob=publicationFile&v=1, accessed: 09/29/2026.

[HP25] Harbola, C.; Purwar, A.: Prescriptive Agents based on RAG for Automated Maintenance (PARAM), arXiv:2508.04714, 2025, https://arxiv.org/abs/2508.04714.

[KB26] Kersting, J.; Benndorf, G.: From Code Generator to Engineering Companion: A Plant-aware Agent for PLC Programming. In: Proceedings of the 31st IEEE International Conference on Emerging Technologies and Factory Automation (ETFA2026). Västerås, Sweden, 2026.

[Ki25] Kirubanandan, R.: Causal-Aware LLM Agents for PHM Co-pilots: Health Monitoring and Intervention Planning. In: Annual Conference of the PHM Society (Poster). 2025, https://papers.phmsociety.org/index.php/phmconf/article/view/4321.

[LC25] Li, S.; Corney, J.: MechRAG: a multimodal large language model for mechanical engineering. Communications Engineering 4, p. 187, 2025, https://www.nature.com/articles/s44172-025-00517-z.

[Le20] Lewis, P. et al.: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In: Advances in Neural Information Processing Systems. Vol. 33, Curran Associates, Inc., pp. 9459–9474, 2020, https://dl.acm.org/doi/abs/10.5555/3495724.3496517.

[Le25] Lee, X. Y. et al.: Exploring LLM-based Agentic Frameworks for Fault Diagnosis. In: Annual Conference of the PHM Society. 2025, https://papers.phmsociety.org/index.php/phmconf/article/view/4350.

[Lö25] Löwenmark, K. et al.: Agent-based Condition Monitoring Assistance with Multimodal Industrial Database Retrieval Augmented Generation, arXiv:2506.09247, 2025, https://arxiv.org/abs/2506.09247.

[Lu24] Lukens, S. et al.: Large Language Model Agents as Prognostics and Health Management Copilots. In: Annual Conference of the PHM Society. 2024, https://papers.phmsociety.org/index.php/phmconf/article/view/3906.

[Lu25] Lukens, S. et al.: An Evaluation Framework for Fault Diagnosis Using Technical Manuals in Retrieval-Augmented Large Language Models. In: Annual Conference of the PHM Society. 2025, https://papers.phmsociety.org/index.php/phmconf/article/view/4549.

[Mo25] Monka, S. et al.: Enhancing Manufacturing Knowledge Access with LLMs and Context-aware Prompting, 2025, arXiv: 2507.22619 [cs.AI], https://arxiv.org/abs/2507.22619.

[Pi20] Pittino, F. et al.: Automatic Anomaly Detection on In-Production Manufacturing Machines Using Statistical Learning Methods. Sensors 20 (8), p. 2344, 2020, https://www.mdpi.com/1424-8220/20/8/2344.

[Tr24] Trilla, A. et al.: Industrial-Grade Smart Troubleshooting through Causal Technical Language Processing: a Proof of Concept. In: Causal Inference and Machine Learning in Practice Workshop (KDD 2024). 2024, https://arxiv.org/abs/2407.20700.

[vo26] von Heißen, O. et al.: Semantic-Aware Change Management for SysML v2 Using Large Language Models. Procedia CIRP 142, 36th CIRP Design Conference (CIRP Design 2026), pp. 546–551, 2026, https://www.sciencedirect.com/science/article/pii/S2212827126008802.

[VV24] Velesaca, W.; Velesaca Lara, H.: Anomaly Detection in Industry 4.0: Supervised vs. Semi-supervised Approaches with OPC-UA Integration. In: LatinX in AI Workshop at NeurIPS 2024. 2024, https://neurips.cc/virtual/2024/107793.

[ZB24] Zelba, F.; Benndorf, G.: Active Learning for Condition-Based Maintenance of Industrial Machinery Using COMETH. In: 2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA). Pp. 1–8, 2024.

[Zh25] Zhang, H. et al.: Knowledge Graph Construction towards a Graph RAG-Enhanced Intelligent Maintenance Chatbot. In: Workshop on Scaling Knowledge Graphs for Industry (SKGi) at SEMANTiCS 2025. 2025, https://ceur-ws.org/Vol-4064/SKGi-paper3.pdf.

Semantic extension of agentic AI communication protocol using Industry 4.0 language and AAS

Rafiul Islam[1]*, Christian Diedrich*, Lisa Underberg*

Abstract:

AI systems are increasingly considered for industrial production environments, yet their general-purpose communication protocols — most notably the Agent-to-Agent (A2A) protocol — lack formally defined, machine-interpretable semantics for domain-specific data. In parallel, the Industry 4.0 Language (VDI/VDE 2193), designed for interoperable interaction among Asset Administration Shells (AAS), provides structured message semantics and interaction protocols but has not achieved broad industrial adoption. This paper investigates how the specifications of the I4.0 Language, together with the AAS information model, can be integrated into the A2A protocol. A detailed structural comparison of both protocols' message elements is presented, identifying direct equivalences (e.g., MessageID, Role), partial correspondences requiring metadata extensions (e.g., ReplyBy, Purpose), and complementary mechanisms unique to each protocol (e.g., AgentCard in A2A, Interaction Manager state machines in I4.0). Two integration options — Streaming and Push Notification — are evaluated for their suitability to carry I4.0 Language interactions over A2A. The proposed mapping leverages A2A's extension and metadata mechanisms to embed I4.0 semantics without modifying the core protocol, making the full range of AAS submodel data available for agent-based industrial applications.

Keywords: Agentic AI, A2A protocol, I4.0 Language, Asset Administration Shell, interaction pattern, peer-to-peer, semantic interoperability

1 Introduction

Agentic AI has become a topical subject regarding the implementation of distributed, autonomous software systems. Consequently, it is worth investigating how these systems can be deployed in industrial production. In line with the focus of KommA, the emphasis here is placed on the interaction mechanisms between agents. Alongside the MCP (Model Context Protocol), "general-purpose" protocols such as A2A, ACP, Agora, and ANP are currently gaining prominence. Review articles by Atri [At26], Hongyi Du et al. [Du25], and Yang et al. [Ya25] characterize and compare these protocols. From a technical standpoint, the A2A protocol appears well-suited for use in industrial environments (the paper provides a detailed rationale for this). Regarding the semantic aspects of the data being transmitted, references to Linked Data—if any—are scarce. Furthermore, the overall concept certainly envisions integrating LLMs via MCP to interpret data semantics; this is undoubtedly a sensible approach in the consumer sector.
In industrial environments, however, formally defined, machine-interpretable semantics are a fundamental prerequisite for reliable and scalable communication. Standardized Digital Twins such as Asset Administration Shell (AAS) are set to establish themselves as the standard for communication between the representatives of assets. The AAS can take various forms of representation: as a file, as a server, or as a proactive software component. Research by Diedrich et. al. [DBSK20] demonstrates how the AAS, utilizing the I4.0 Language, enables order-driven production scenarios, while Is23 [Is23] shows how an AAS can be coupled with an agent to implement a predictive machine maintenance scenario. However, the I4.0 language and its mapping to the AAS have not yet seen widespread

[1] *Otto von Guericke University Magdeburg, Magdeburg, (christian.diedrich@ovgu.de, rafiul.islam@ovgu.de, lisa.underberg@ovgu.de).

adoption outside of demonstrators. Combining the underlying principles of this language with current agent initiatives appears to be a promising way to realize the potential of Industry 4.0 use cases. Therefore, this paper demonstrates how the specifications of the I4.0 language—in conjunction with the AAS information model—can be investigated using the A2A protocol. This would make the full range of application data contained in the Submodels available for agent-based applications. Such an approach could increase acceptance of the AAS as an implementation method for Agentic AI.

2 State of the art in automation

The concept of Industry 4.0 was first introduced in Germany. Since its introduction, extensive research has been conducted on this topic. Various standards, architectures, and technologies have been proposed to support its implementation. The AAS metamodel specification (Part 1), based on IEC 63278 (Asset Administration Shell Specification - Part 1: Metamodel Schema, 2023), defines the structural principles for representing real-world assets such as machines, robots, and processes. It organizes asset information through Submodels and standardized classes, providing a robust framework for semantic interoperability across the value chain.
In addition, the Specification of the Asset Administration Shell Part 2: Application Programming Interfaces published in 2026 specifies standardized APIs for accessing and managing information provided by the AAS. However, current AAS implementations primarily serve as semantic information repositories, focusing on standardized asset representation and interoperability. Although they effectively describe asset characteristics, capabilities, and process variables, they lack intrinsic mechanisms for autonomous reasoning, coordination, and decision-making.

To realize the full potential of Industry 4.0, assets must evolve into intelligent and collaborative entities capable of autonomous interaction, conflict resolution, and context-aware behavior. To support this vision, the I4.0 Language was introduced as a communication framework to enable interoperable communication and cooperation among industrial assets [BeDi19a, Indu00]. Despite its potential to facilitate autonomous collaboration among industrial assets, the I4.0 Language has not yet achieved widespread industrial adoption. To date, the literature only presents proof-of-concept and prototype-based implementations (Belyaev & Diedrich, 2019b; Hurtado et al., 2025; Islam et al., n.d.; Sakurada, de la Prieta, et al., 2025). A key limitation is its dependency on predefined state machines, and engineering teams found it too complex to scale and implement for dynamic, changing factories. [DiLV26, SaPL25b].

Currently, we can see a paradigm shift in the industry toward "Agentic AI Architectures" driven by Large Language Models (LLMs) that can handle data ambiguity and perform dynamic reasoning [Thak26]. However, many LLM-based agentic AI systems in the industry are developed independently. Differences in their interfaces, communication mechanisms, and interaction models can create interoperability and coordination challenges, making it difficult for heterogeneous agents to communicate and collaborate effectively. As a result, standardized mechanisms for communication, tool integration, data exchange, task coordination, and conflict resolution to ensure efficient and scalable collaboration across distributed industrial systems are necessary (Sharma et al., 2026). To address these challenges, several agent communication protocols have already been standardized; Yang et al. [YCSQ25] Provide a comprehensive overview. This paper focuses only on the Agent-to-Agent (A2A) protocol, which enables agent discovery, security and task-oriented interaction through standardized interfaces, while the Model Context Protocol (MCP) focuses on integrating agents with external tools and contextual resources. The A2A protocol establishes communication mechanisms and interaction patterns that facilitate interoperability across heterogeneous agent ecosystems.

Agent protocols such as MCP and A2A are intentionally general-purpose. They standardize discovery, message structure, and task handling independent of any specific vendor, framework, or industry. However, the domain semantics are out of scope of protocol specifications, meaning there is no common means to define the message content. The Agent protocol community assigns domain rules to extensions, not to the core protocol (Jones, 2026). An IETF draft observes agentic protocols define no common ontology and proposes a semantic layer for agent-to-agent interaction [ZSYH26]. Some domain-specific projects already follow this approach. In commerce, UCP (Universal Commerce Protocol Project) and AP2 (Agent Payments Protocol) are A2A extensions that carry structured domain data [Appr26, Univ26]. In laboratory automation, LAP (Lab Agent Protocol) proposes an A2A-based profile in which every value references a standard unit (UCUM) and quantity type (QUDT), so that meaning is fixed by shared vocabularies [ZGCZ26]. In the healthcare domain, a scoping review found thirteen agent communication protocols implementations projects, ten used MCP, two combined MCP and A2A, and one used A2A alone [HPGG26]. Where clinical semantics are needed, they come from the established HL7 FHIR standard, accessed through MCP [EhSK25]. This suggests that domain semantics are best taken from existing standards, not redefined by the agent protocol.

In industrial automation, this semantic layer remains largely absent from emerging LLM-agent implementations. Machine capabilities are often exposed to agents through natural-language descriptions [SiKG25] , while inter-agent coordination relies on free-text messages [KARG26]. Without a shared ontology or controlled vocabulary, agents have no explicit, machine-interpretable reference for the meaning of industrial terms. Consequently, the LLM must infer the intended meaning from linguistic context rather than rely on formally defined concepts. This can lead to inconsistencies in how industrial information is interpreted. For example, in one study, a request for “stainless steel” failed because the tool recognized only the term “stainless”; the model inferred the intended term and retried [SiKG25]. Although this recovery was successful, it depended on the model's ability to infer an appropriate mapping rather than on an explicit, verifiable semantic correspondence. Independently developed agents may therefore exchange messages successfully without sharing a formally defined and verifiable understanding of the industrial concepts involved.

This limitation highlights an important research gap at the intersection of emerging agent protocols and established Industry 4.0 interoperability concepts: rather than defining a new industrial agent protocol, existing general-purpose protocols can potentially be extended or semantically grounded using established industrial models. On the other side, the integration of a standardised vocabulary such as AAS models enhances the quality of general-purpose Agent protocols.

3 Protocol descriptions

3.1 I4.0 Language

The I4.0 language is an interaction pattern at application level between AAS (Figure 1). It is derived from the human interaction and is structured in three levels, the vocabulary (lowest level), the message structure (middle level) and the interaction protocols. The concept is standardised in VDI/VDE 2193.

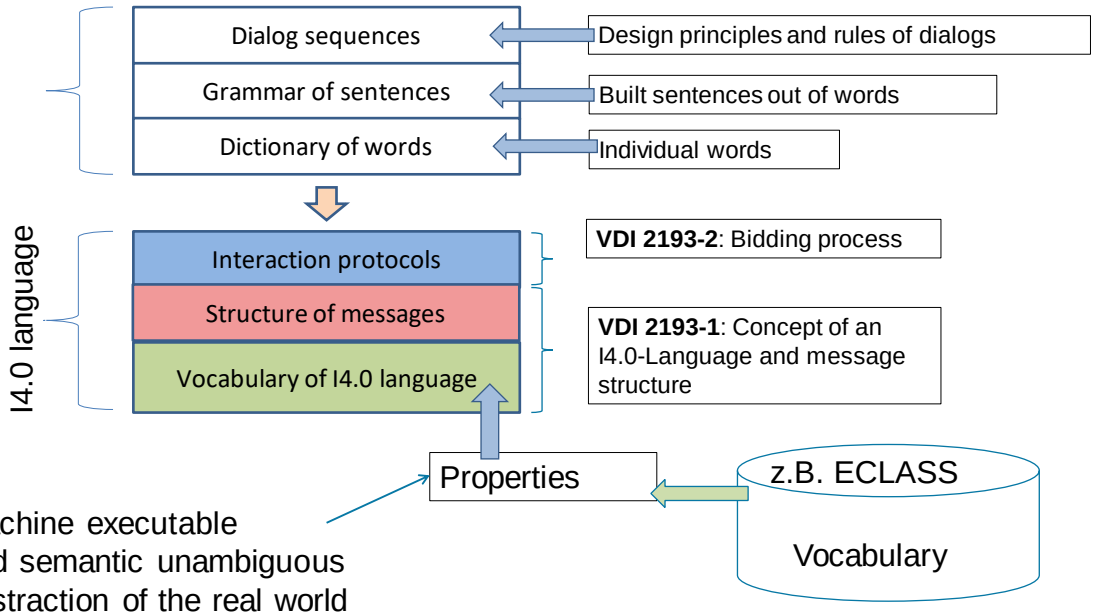


Figure 1: General structure of I4.0 Language

The I4.0 language is used among applications which provide an interaction manager (Figure 2). The interaction manager contains a state machine which controls the interaction protocol. The interaction protocol depends on the purpose if the interaction. Examples are bidding among application agent, or the interaction according to the IEC 62264-part 5 communication pattern.

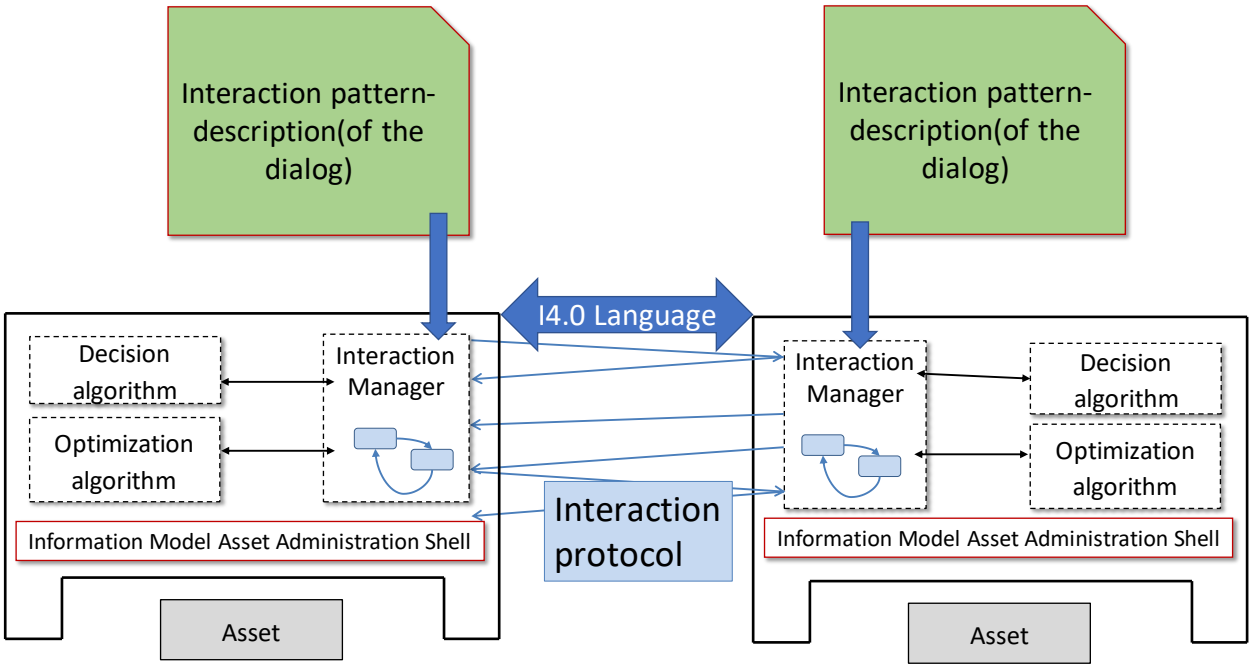
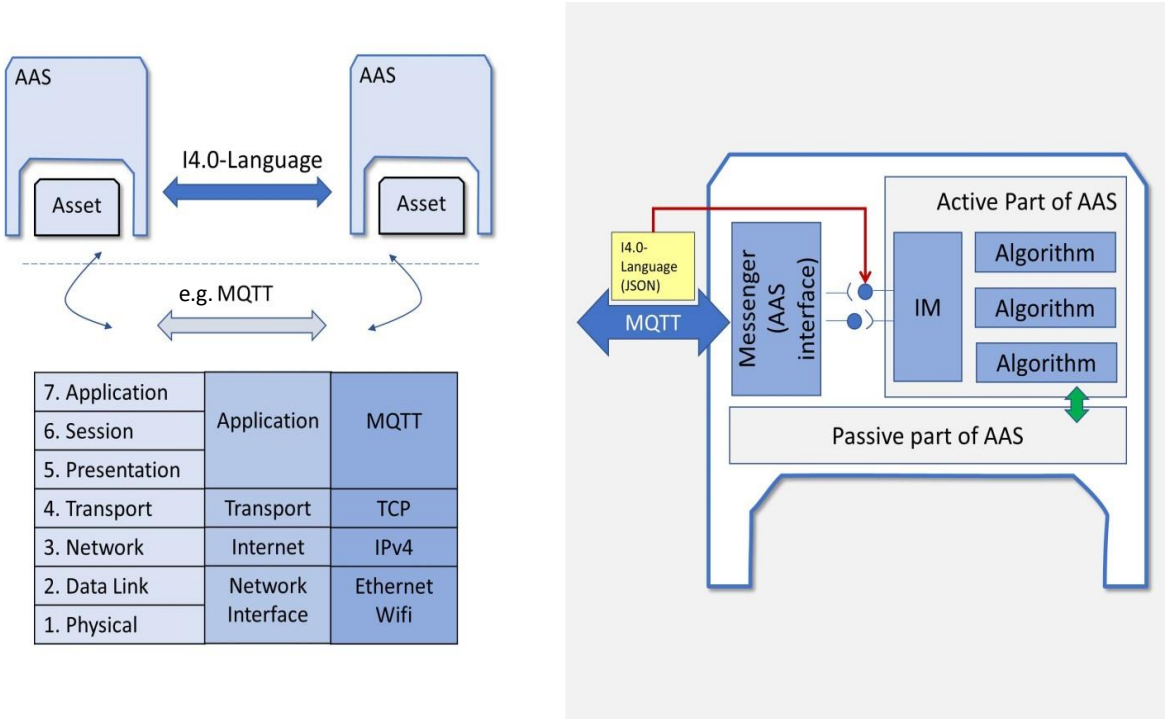


Figure 2: I4.0 Language for interaction among applications

The I4.0 Language is on top of the OSI Reference Model (Figure 3).



Legend: IM – Interaction Manager

Figure 3: Relation of the I4.0 Language to OSI Reference model

Messages trigger the transition in the state machine of the interaction manager. The message structure (Figure 4) consists of an identification and a data area part. The Identification elements are:

- ConversationID – unambiguous identifier of one instance of an interaction pattern (e.g. one bidding process)
- MessageID – unambiguous identifier of one message within one conversation instance
- InReplyTo – MessageID of the messages to which the message refers to
- ReplyBy – date time the response of the messaged has to be provided

The Data area elements are:

- Purpose
  - ProtocolID – unambiguous identifier of the interaction pattern
  - Role – Role the interaction partner has in the interaction pattern (e.g. Client, Service Requester)
- Data elements – JSON data based on the AAS meta model

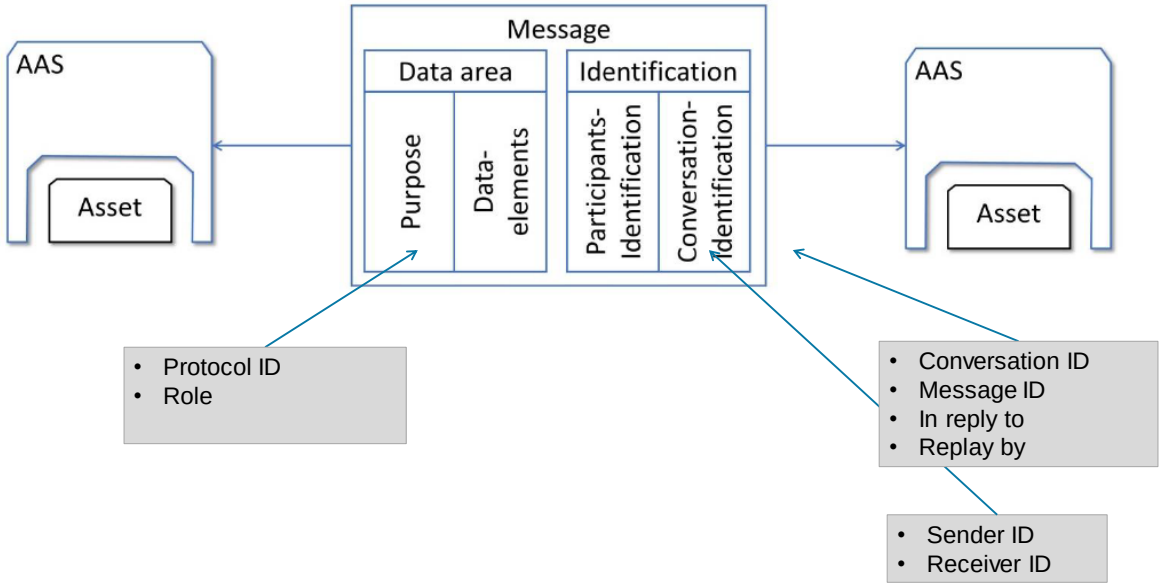


Figure 4: VDI/VDE 2193-1: Structure of messages

3.2 A2A protocol

The A2A protocol [A2A25] is designed as general-purpose protocol to be applicable in different industrial and business domains. It is structured in the A2A Client and Server as well as task (Figure 5). Entities can be both A2A Clients and server. A potential user interacts in a bilateral manner with the A2A Client.

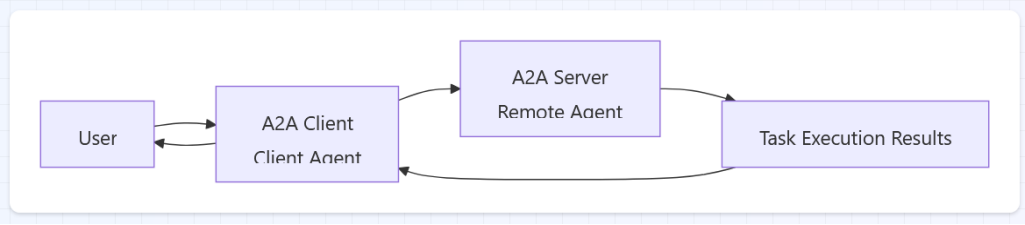


Figure 5: AI Agent- A2A concept [A2A25]

The main features are:

- Agent Cards
  - Self-description of the A2A Remote Agent in a standardized manner

- Service types
    - Requ./Resp. – classical client server service
    - Streaming – communication mode in which the server keeps a single HTTP connection open after the initial request and continuously pushes a sequence of discrete events — task status changes and artifact chunks — to the client as they occur, without the client having to ask again. (via SSE for real-time progress updates)
    - Push Notifications – mechanism that the Remote Agent can send an information to the client without a request (There is no persistent connection. The agent independently opens a fresh HTTP POST to the client's pre-registered webhook URL each time a single event occurs. The client does not need to be connected — it just needs to be reachable at that URL when the agent decides to call it.)
- Stateful interaction (using task mechanism)
    - Task inform about its execution state via StreamResponse (state: submitted, working, input-needed, completed, failed, canceled, and rejected)
- Message formats
    - plain text, files, structured data (JSON) and application specific

An example of the usage of the A2A protocol is shown in Figure 6. The upper layer functionality in an engineering application interacts with the A2A protocol. The interaction between the domain specific engineering tools and databases are using MCP.

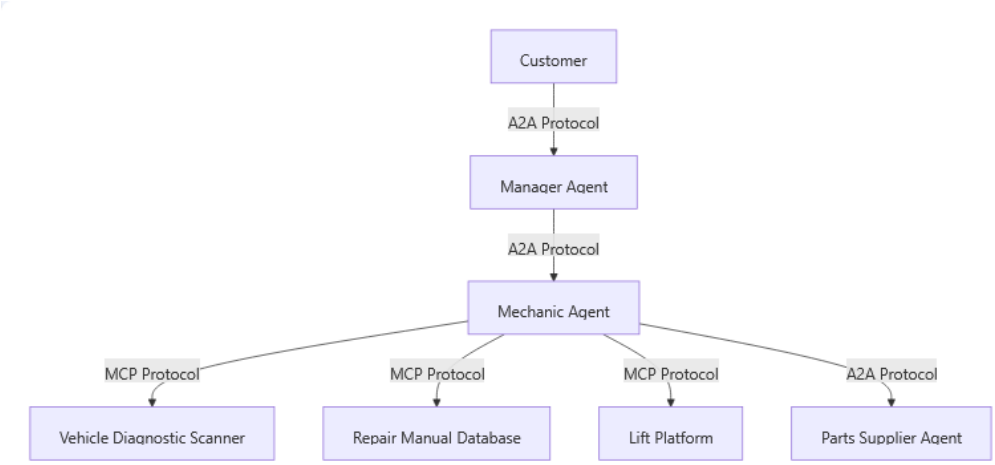


Figure 6: AI Agent - Example Complementation of A2A and MCP – Car workshop [A2A25]

The A2A protocol stack is based on TCP and the https RPC V2.0 protocol which built the envelope of the A2A messages. The A2A message have the identifying elements and the payload with is the element "parts[]". This can be one or more than one. The parts can be plain text, JSON or binary.

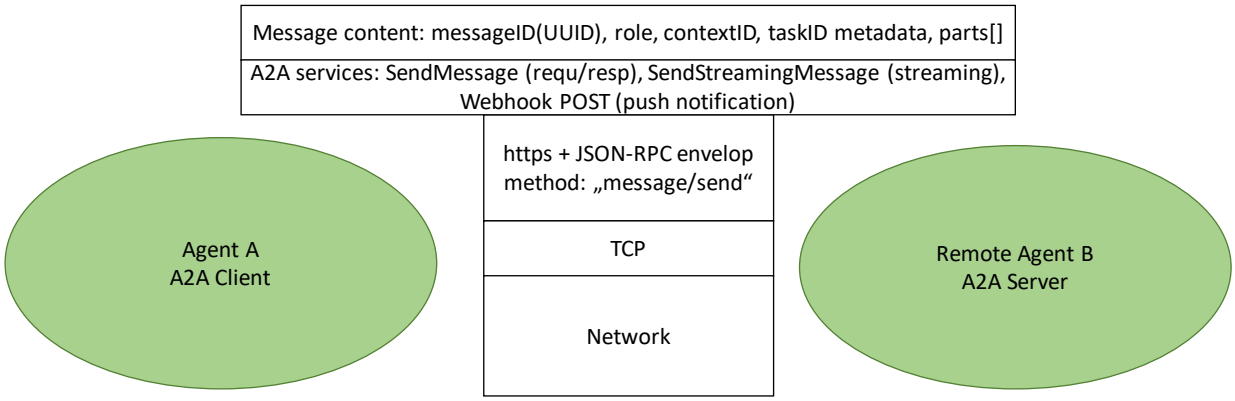


Figure 7: A2A protocol overview

According to the service type there are a set of messages with their data elements. There is a common set of data elements which are common for all messages. They are described in Table 1.

Table 1: Common data elements in A2A messages

A2A message data elements	Meaning
messageID	ID of the messages
role	Fixed enums: ROLE_USER, ROLE_AGENT, ROLE_UNSPECIFIED
DataPart	flexible container that holds exactly one content field using a oneof structure. The content variants are text (plain string), url (URI referencing external file content), raw (inline binary bytes), and data (a structured JSON value for machine-readable content). Every Part additionally caries mediaType (MIME type), an optional filename, and a metadata key-value map
contextID	ContexttD allows the grouping of messages (optional)
taskID	ID of the task (optional)
Metadata	Integrate namespace in A2A (optional)

Before the productive interaction between agents, the Agent Client requests the AgentCard from the Remote Agent. It contains the capabilities and skills of the Remote Agent. Capabilities means the supported services (requ./resp., …) and the skill the specific service features such as security means. The AgentCard elements are listed in Table 2. This gives the client the opportunity to adjust direct to the specific server opportunities. The AgentCard is a self-declaration of the Remote Agent.

Table 2: Selected AgentCard elements

Names	Meaning
Name	Human readable label
Description	Help text for human
url	A2A server url
Version	Version of the A2A protocol
Capabilities	Supported services (requ/resp, streaming, pushNotification)
defaultInputModes	Types of payload data (plain text, JSON, …)
Skill.id, skill.name, …	Plain text of skill related information

4 I4.0 Language integration into A2A protocol

The comparison of the I4.0 Language with the A2A protocol in detail has shown that some message elements are overlapping, some are mutually different and there are different mechanisms for the stateful interaction. The so called

AgentCard of the A2A protocol is something with no equivalent in the I4.0 Language. The explicit specification of the dialog in terms of state machines is special for the I4.0 language.

The integration concept suggests to using the agent and the A2A protocol as top-level interaction. This gives the opportunity to get advantage of the accompanying agent and security mechanisms as well as the potential implementation frameworks.

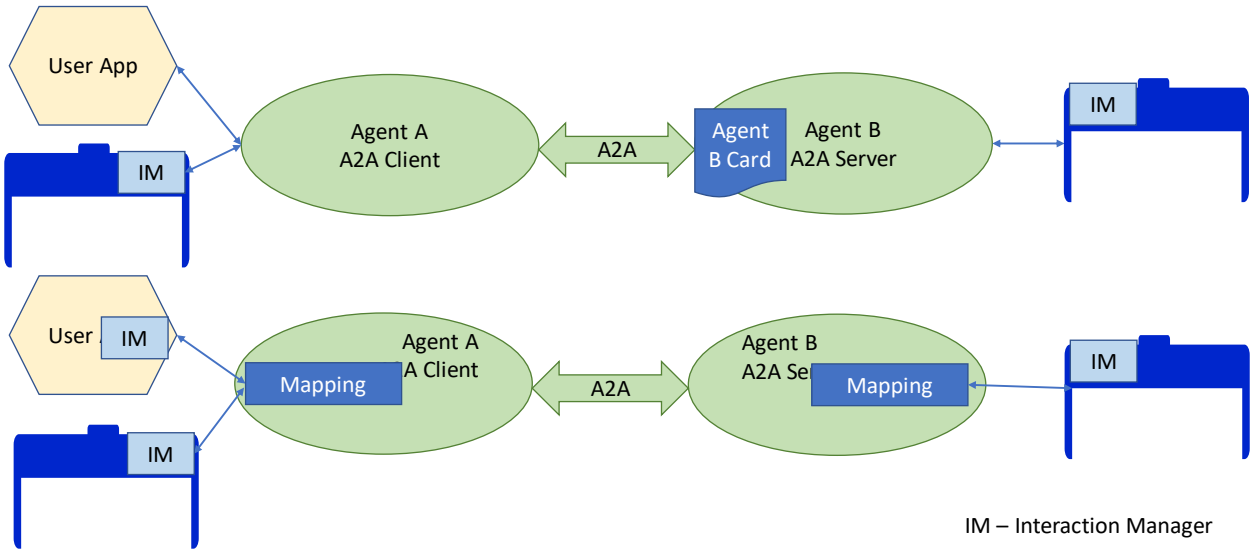


Figure 8: General concept of the I4.0 Language mapping in A2A protocol

In principle there are three protocol mapping options. Using the client server option is out of scope. The main reason is that this would mean that the Agent client has to poll the Remote Agent. Yes, both agents can take the client role. This is also not a suitable solution because the intended interaction is peer-to-peer.

The Streaming and PushNotification protocol are both suitable.

Streaming starts with SendStreamingMessage. The connection stays open and the server pushes events. This is a simple setup with low latency and follows in principle the intention of the I4.0 Language design. But the connection has to stay open until the dialog is ready. This is not very suitable for a long waiting time. The Streaming option needs the support of the streaming protocol of the agents.

PushNotification starts with the PushNotificationConfig (it contains the webhook url and identifier to allocate incoming events in the client to the PushNotification session) which is answered by the server with a taskID. The connection is closed. The client can request optionally with a following SendMessage the status of the task. Now the server is prepared and have the offered url from the client, where the Remote Agent can send its POST messages. There are separate StreamResponse.statusUpdate and StreamResponse.ArtifactUpdate events. The *.statusUpdate response carrries only the status of the task and the *.artifactUpdate contains the dataParts[] element which can contain the I4.0 Language payload data. This option needs pre-configuration and the client has to provide a webhook url, so it is more complex than the streaming option. On the other side there is no open connection and the longer dialogs can be managed. Additionally, multiple Remote Agent can be involved in one dialog. Another pro is that the POST message is more firewall friendly.

One can see that both options are possible. In both options the interaction manager is outside the kernel of the A2A agent implementation. The selection of the option depends on the duration of the dialog and if more than one couple of Agent is involved. The easier streaming option is good for a one-to-one dialog with durations an HTTP connection can stay. The PushNotification option is for multiple peer-to-peer dialogs with potentially longer waiting times between the messages.

The mapping between main message elements are shown in Table 3 and Table 4. One can see that the message elements are sometimes not overlapping or have different meanings. A2A offers the Message.extension and Message.metadata means to extend the namespace with the necessary I4.0 Language elements. Of course the entire I4.0 messages can be tunnelled in the dataparts[] element of the A2A messages.

Table 3: Comparison of message elements

I4.0 Language (VDI/VDE 2193)	A2A Protocol V1.0	Remarks
ConversationID	ContextID	Similar, ConversationID for entire interaction, ContextID related to the execution of one task. I4.0 dialog can have multiple tasks
MessageID	messageID	Equivalent
InReplyTo	ReferenceTaskID + taskID	Equivalent if A2A task interaction
ReplyBy	-	Use of Message.metadata[“x-i40.replyBy”] Integrate namespace in A2A
Role	Role	A2A limited enums not in the sense of the I4.0 Language Use of Message.metadata[“x-i40.role”] Integrate namespace in A2A
ProtocolID	AgentCard.skill	Element of the AgentCard, not in the message and not related to the I4.0 protocol AgentCard.skill + Message.extension[SkillUri=ProtocolID] Example: “https://vdi-vde2193.de/a2a-extension/i40-language/v10”
Purpose	TextPart or DataPart	No native element in A2A Use of Message.metadata[“x-i40.purpose”] Integrate namespace in A2A
DataElement	DataPart	Both JSON possible

Table 4: Comparison of conceptual elements

I4.0 Language (VDI/VDE 2193)	A2A Protocol V1.0	Remarks
Conversation	Task	A2A task fixed state machine related to one function I4.0 Language multiple functions possible
Interaction Manager	Agent internal not visible function	Not part of the A2A protocol
ReplyBy concept, AAS typed vocabulary, Protocol types and roles		I4.0 Language only
-	AgentCard	A2A Protocol only
	Message.extension[]	Example: “https://vdi-vde.de/a2a-extension/i40-language/v10”
	AgentCard.skill.name	Example: “i40-bidding-v10”
	Skill.description	Example: point to ProtocolID + AAS submodel semanticID

The analysis of the A2A and I4.0 Language protocol has shown that the AAS models can be integrated in the A2A protocol as well as use the peer-to-peer dialog interaction from the I4.0 Language. This is because the A2A protocol is a general purpose one. There are two options which distinguish on one hand in their complexity and on the other hand in their usability to different application needs.

Security was not evaluated in this paper. A2A provides transport encryption, authentication, and signed Agent Cards. An implementation must still consider the threats specific to agent-based systems. The MAESTRO framework provides a structured threat model for agentic AI [Mult25]. It has already been applied to A2A [HHNK25] . Applying this analysis to I4.0 interactions over A2A is the subject of future work.

5 Summary

This paper addresses a gap at the intersection of emerging AI agent communication protocols and established Industry 4.0 interoperability standards. General-purpose agent protocols like A2A standardize discovery, messaging, and task handling deliberately leave domain semantics out of scope. In industrial automation, this absence means agents must rely on LLM inference rather than formally defined vocabularies — an approach that has already shown fragility in practice (e.g., term mismatches like “stainless steel” vs. “stainless”).

The paper describes the A2A and I4.0 Language protocols in detail. The I4.0 Language (VDI/VDE 2193) provides a three-level structure — vocabulary, message format, and interaction protocols — with an Interaction Manager state machine controlling dialog flow between AAS instances. The A2A protocol offers agent discovery via Agent Cards, three service types (Request/Response, Streaming, Push Notification), and a task lifecycle with defined states, all transported over JSON-RPC 2.0 / HTTPS.

The core contribution is a field-level mapping between the two protocols. Several message elements map directly (MessageID ↔ messageID, ConversationID ↔ ContextID, Data elements ↔ DataPart). Others require A2A’s metadata or extension mechanisms to carry I4.0 semantics that have no native A2A counterpart — notably ReplyBy, the I4.0-specific Role semantics, Purpose, and ProtocolID. The paper also identifies conceptual elements unique to each side: the AgentCard (A2A only) provides pre-interaction capability discovery absent in I4.0, while the Interaction Manager and typed AAS vocabulary (I4.0 only) provide formally defined dialog control and domain semantics absent in A2A.

For transport, two A2A service types are evaluated. Streaming (via SSE) offers simplicity and low latency for short one-to-one dialogs but requires a persistent connection. Push Notification, using pre-registered webhook URLs, supports longer dialogs and multi-party interactions without open connections, at the cost of more complex setup. Both options keep the Interaction Manager outside the A2A agent kernel as an internal component.

The paper concludes that AAS models can be integrated into the A2A protocol and that the I4.0 Language’s peer-to-peer dialog interaction can be realized through either transport option, with the choice depending on dialog duration and the number of agents involved. This approach positions the AAS as a viable semantic foundation for agentic AI in industrial settings.

6 Acknowledgement:

Claude Opus 4.6 was used for a part of the literature research, for some overview about the A2A protocol as well as spelling and grammar check. The authors reviewed and revised all AI-assisted content and remain responsible for the final manuscript.

7 Literature references

[A2A25] *A2A Protocol: 2025 Full Guide to the A2A Protocol (Blog post), 2025. URL https://a2aprotocol.ai/blog/2025-full-guide-a2a-protocol-de – Accessed 2026-10-12*

[AP2_26] *AP2 Protocol Community: Agentic Payment Protocol (AP2) Specification, Version 0.2, 2026. URL https://ap2-protocol.org/ap2/specification/ – Accessed 2026-10-12*

[Atri26] *Atri, Saurabh: Standardizing AI Agent Protocols: Definitions, Layered Architectures, and Future Research Directions. In: International Journal of Artificial Intelligence, Data Science, and Machine Learning Vol. 7 (2026), No. 1, pp. 39–41. DOI 10.63282/3050-9262.IJAIDSML-V7I1P109*

[BeDi19] *Belyaev, Alexander ; Diedrich, Christian: Specification „Demonstrator I4.0-Language“, Version 3.0, 2019*

[DBSK20] *Diedrich, Christian ; Belyaev, Alexander ; Schröder, Tizian ; et al.: Produktgesteuerter Produktionsablauf basierend auf I4.0-Verwaltungsschalen. In: AUTOMATION 2020 (Baden-Baden, virtual, 30.06.–01.07.2020), VDI-Berichte 2375. Düsseldorf : VDI Verlag, 2020, p. 89 ff. DOI 10.51202/9783181023754-89*

[DiLV26] *Dietrich, David ; Lechler, Armin ; Verl, Alexander: Analysis of Asset Administration Shell-based Negotiation Processes for Scaling Applications. arXiv:2602.13029, 2026. DOI 10.48550/arXiv.2602.13029*

[DSLD25] Du, Hongyi ; Su, Jiaqi ; Li, Jisen ; Ding, Lijie ; et al.: Which LLM Multi-Agent Protocol to Choose? arXiv:2510.17149, 2025. DOI 10.48550/arXiv.2510.17149

[EhSK25] Ehtesham, Abul ; Singh, Aditi ; Kumar, Saket: Enhancing Clinical Decision Support and EHR Insights through LLMs and the Model Context Protocol: An Open-Source MCP-FHIR Framework. arXiv:2506.13800, 2025. DOI 10.48550/arXiv.2506.13800

[HBAC25] Hurtado, Ekaitz ; Burgos, Arantzazu ; Armentia, Aintzane ; Casquero, Oskar: Self-configurable Manufacturing Industrial Agents (SMIA): A Standardized Approach for Digitizing Manufacturing Assets. In: Journal of Industrial Information Integration Vol. 47, Elsevier (2025), Art. 100915. DOI 10.1016/j.jii.2025.100915

[HHNK25] Habler, Idan ; Huang, Ken ; Narajala, Vineeth Sai ; Kulkarni, Prashant: Building A Secure Agentic AI Application Leveraging A2A Protocol (2025)

[HPGG26] Haider, Syed Ali ; Prabha, Srinivasagam ; Gomez-Cabello, Cesar Abraham ; Genovese, Ariana ; et al.: Emerging Agent Communication Protocols for Healthcare AI Systems: A Scoping Review of Model Context Protocol (MCP) and Agent-to-Agent (A2A) Frameworks (Preprint). JMIR Preprints, 2026. DOI 10.2196/preprints.93344

[Indu23] Industrial Digital Twin Association: Specification of the Asset Administration Shell – Part 1: Metamodel (IDTA-01001), Version 3.0. Frankfurt am Main, 2023. URL https://industrialdigitaltwin.org/ – Accessed 2026-10-12

[Indu26] Industrial Digital Twin Association: Specification of the Asset Administration Shell – Part 2: Application Programming Interfaces (IDTA-01002-3-2). Frankfurt am Main, 2026. DOI 10.62628/idta.01002-3-2

[IWRS23] Islam, R. ; Wand, A. ; Röder, Ch. ; Stamm, S. ; et al.: Erfahrungsbericht bei der Umsetzung der VWS Type 3 Interaktionen in einer Maintenance-Anwendung. In: Jumar, U. ; Jasperneite, J. (Eds.): KommA 2023 – 14. Jahreskolloquium Kommunikation in der Automation (Magdeburg, 21.–22.11.2023). Magdeburg : ifak/inIT, 2023. DOI 10.25673/111634

[KARG26] Köhle, Kay ; Anicic, Darko ; Runkler, Thomas A. ; Graf, René: LLM-Driven Multi-Agent Control for Skill-Based Smart Manufacturing. arXiv:2610.01364, 2026. DOI 10.48550/arXiv.2610.01364

[Mult25] Multi-Agentic system Threat Modelling Guide OWASP GenAI Security Project-Agentic Security Initiative, 2025

[Plat18] Plattform Industrie 4.0: I4.0-Sprache: Vokabular, Nachrichtenstruktur und semantische Interaktionsprotokolle der I4.0-Sprache (Diskussionspapier). Berlin : Bundesministerium für Wirtschaft und Energie (BMWi), 2018

[SaPL25a] Sakurada, Lucas ; De la Prieta, Fernando ; Leitão, Paulo: An Agent-Based Approach for Implementing Asset Administration Shell Type 3. In: IEEE Access Vol. 13 (2025), pp. 206617–206638. DOI 10.1109/ACCESS.2025.3640265

[SaPL25b] Sakurada, Lucas ; De la Prieta, Fernando ; Leitão, Paulo: The Role of Multi-Agent Systems in Realizing Asset Administration Shell Type 3. In: Future Internet Vol. 17 (2025), No. 7, Art. 270. DOI 10.3390/fi17070270

[SiKG25] da Silva, Luis Miguel Vieira ; Köcher, Aljosha ; Gehlhoff, Felix: Beyond Formal Semantics for Capabilities and Skills: Model Context Protocol in Manufacturing. In: 2025 IEEE 30th International Conference on Emerging Technologies and Factory Automation (ETFA) (Porto, Portugal, 09.–12.09.2025). IEEE, 2025. DOI 10.1109/ETFA65518.2025.11205601

[SVCR26] Sharma, Rishi ; de Vos, Martijn ; Chari, Pradyumna ; Raskar, Ramesh ; Kermarrec, Anne-Marie: Position: Collaborative Agentic AI Needs Interoperability Across Ecosystems. arXiv:2505.21550, 2026. DOI 10.48550/arXiv.2505.21550

[Thak26] Thakur, Lukesh: Artificial Intelligence Agents and Agentic AI Systems: Architectures, Capabilities, Applications, Challenges, and Future Directions. In: International Journal of Applied Resilience and Sustainability Vol. 2 (2026), No. 2, pp. 721–744. DOI 10.70593/deepsci.0202030

[UCP26] *Universal Commerce Protocol Project: Universal Commerce Protocol (UCP) Specification: Overview and A2A Checkout Binding, 2026. URL https://ucp.dev/specification/overview/ – Accessed 2026-10-12*

[YCSQ25] *Yang, Yingxuan ; Chai, Huacan ; Song, Yuanyi ; Qi, Siyuan ; et al.: A Survey of AI Agent Protocols. arXiv:2504.16736, 2025. DOI 10.48550/arXiv.2504.16736*

[ZGCZ26] *Zhu, Linwu ; Gao, Liqiang ; Chen, Yan ; Zhu, Dan ; Huang, Jian: LAP: An Agent-to-Instrument Protocol for Autonomous Science. arXiv:2606.03755, 2026. DOI 10.48550/arXiv.2606.03755*

[ZWLY26] *Zhang, Lianhua ; Wang, Shoufeng ; Li, Yun ; Yang, Huiling: Ontology-based Semantic Interaction for Internet of Agents. IETF Internet-Draft draft-zhang-dmsc-ioa-semantic-interaction-02, 2026. URL https://datatracker.ietf.org/doc/draft-zhang-dmsc-ioa-semantic-interaction/02/ – Accessed 2026-10-12*

Impressum

17. Jahreskolloquium
Kommunikation in der Automation
(KommA 2026)

10. November 2026 • Lemgo

OPEN-Book

ISBN: 978-3-9818463-7-9
DOI: https://doi.org/10.25644/3yhf-pk66

Herausgeber:
Jürgen Jasperneite, Lemgo
Lisa Underberg, Magdeburg

Institut für industrielle Informationstechnik
Technische Hochschule Ostwestfalen-Lippe

Campusallee 6
D-32657 Lemgo
Telefon: +49 5261 702-2400
Internet: www.init-owl.de | www.jk-komma.de