LogIQ
VDA5050 Analyzer

PRODUKTBROSCHÜRE

Konformität nachweisen, statt sie zu vermuten.

Prüfwerkzeug für VDA5050, M2X und VDMA LIF — für Logdateien und für laufenden Betrieb.

112
Prüfungen VDA5050 und M2X

13
Prüfungen VDMA LIF

2.1.0 / 3.0
unterstützte VDA5050-Versionen

EIN PRODUKT VON

LogistiXpert

www.logistixpert.de
vda5050.logistixpert.de · m.merz@logistixpert.de

Auf einen Blick

Der LogIQ VDA5050 Analyzer liest den Nachrichtenverkehr zwischen Flottenleitsystem und Fahrzeugen, prüft ihn gegen die Spezifikation und sagt nachvollziehbar, was abweicht — aus einer Logdatei genauso wie aus laufendem Betrieb.

Drei Spezifikationen

VDA5050 2.1.0 und 3.0, M2X 0.9.0 als Vorschau und VDMA LIF 1.0.0 in einem Werkzeug.

Drei Zugänge

Logdatei importieren, live am Broker mitschneiden oder transparent als Proxy dazwischen sitzen.

Prüfbares Urteil

Der Bericht nennt nicht nur Befunde, sondern auch den Prüfumfang: was lief, was mangels Daten nicht griff, was abgeschaltet war.

Flottentauglich

Sequenzen und Lebenszyklen werden je Fahrzeug gewertet — so, wie VDA5050 sie zählt.

Sehen statt lesen

Live-Ansicht auf dem echten Fahrkurs, Replay abgeschlossener Läufe, Order-Pfad über dem LIF-Layout.

Im eigenen Netz

Docker Compose, on-premise oder gehostet. Broker-Zugangsdaten verschlüsselt, Rollen mit klaren Grenzen.

Inhalt

01 Warum es dieses Werkzeug gibt

VDA5050 verspricht, dass Fahrzeuge und Flottenleitsysteme verschiedener Hersteller zusammenarbeiten. Ob sie es tun, entscheidet sich in der Inbetriebnahme — und dort fehlt meist das, worauf sich beide Seiten einigen könnten: ein gemeinsamer, prüfbarer Blick auf die Nachrichten, die tatsächlich über den Broker gegangen sind.

„Wer hat Recht?“

Fahrzeug und Leitsystem kommen von verschiedenen Herstellern. Bei einer Störung steht Aussage gegen Aussage. Ohne Beleg dauert die Klärung Tage und endet oft bei dem, der zuerst nachgibt.

„Was ist eigentlich geprüft?“

Ein grüner Haken ohne Prüfumfang ist wertlos. Für eine Abnahme muss belegt sein, welche Regeln gelaufen sind, welche mangels passender Nachrichten nicht griffen und welche bewusst abgeschaltet waren.

„Was war da los?“

Der Fehler tritt einmal am Freitagabend auf. Ohne Aufzeichnung mit Kontext bleibt nur, ihn nachzustellen — und zu hoffen, dass er sich zeigt.

Der Ansatz

Der Standard beschreibt Nachrichten. Er beweist nicht, dass zwei Systeme sie gleich verstehen. Genau diese Lücke schließt der Analyzer: Er wertet den echten Verkehr aus, belegt jeden Befund an der auslösenden Nachricht und dokumentiert, unter welchen Regeln das Urteil zustande kam.

Was der Analyzer nicht ist

Er ersetzt keine Sicherheitsbetrachtung, keine Abnahmefahrt und keine Prüfstelle. Er erteilt keine Zertifizierung und keine Freigabe. Was er liefert, ist Beweismaterial: welche Nachricht wann von wem gesendet wurde, welche Regel darauf zutrifft und wo genau sie verletzt ist. Die Bewertung dieses Materials bleibt bei Ihnen.

02 Drei Wege hinein

Je nachdem, was vorliegt — eine Datei aus dem Feld, ein laufender Broker oder eine Anlage, in die niemand eingreifen darf.

1 Logdatei importieren

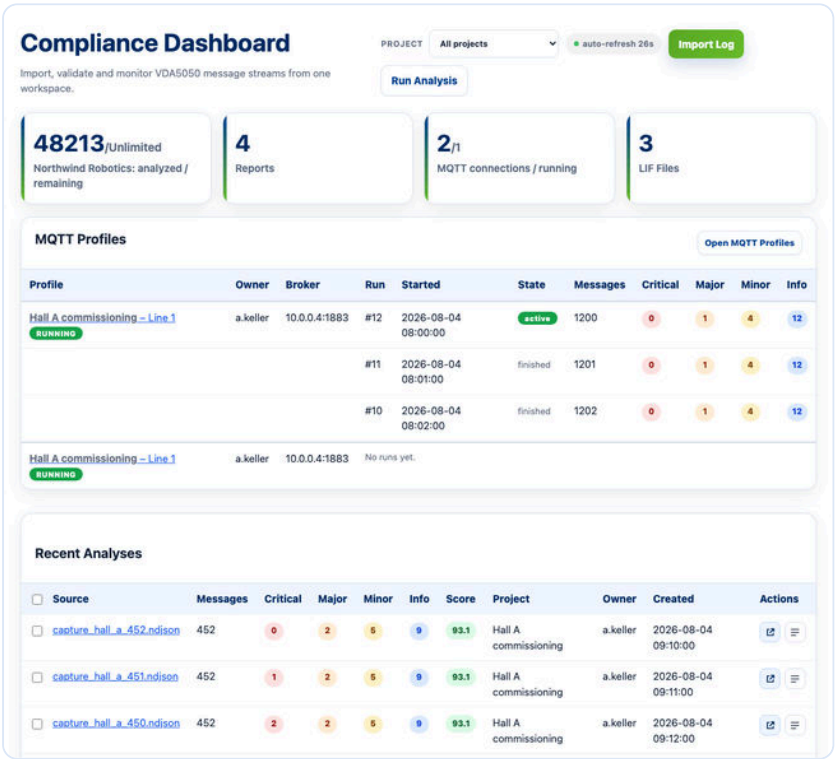
Mitschnitte im Format .log, .txt, .json oder .ndjson. Der Importer erkennt den Nachrichtentyp, meldet Zeilen, die er nicht lesen konnte, und behält die Feldreihenfolge der Quelle bei — Nachrichtenansichten stimmen damit Zeichen für Zeichen mit Ihrer Datei überein. Cleaned JSON, NDJSON und das unveränderte Original bleiben herunterladbar.

2 Live mitschneiden

Ein Capture-Profil hält Broker, Zugangsdaten, Topics, Zielversion, Prüfauswahl und optional ein LIF-Layout. Gestartete Profile zeichnen im Hintergrund auf und analysieren jede Nachricht sofort; mehrere Profile laufen gleichzeitig. Jeder Start-Stopp-Zyklus wird ein Lauf mit eigenen Nachrichten und Befunden.

3 Transparent dazwischen

Dasselbe Profil kann den aufgezeichneten Verkehr auf einen zweiten Broker spiegeln. Der Analyzer sitzt dann als transparenter Proxy zwischen Fahrzeugen und Leitsystem, ohne den Betrieb zu verändern — nützlich, wenn nur ein Netzsegment erreichbar ist.



Ein Arbeitsplatz für Importe, Analysen, Layouts und laufende Mitschnitte.

> Für den Import ist kein Zugriff auf die Anlage nötig. Für den Mitschnitt genügt Lesezugriff auf den Broker; das Spiegeln ist optional und wird pro Profil eingeschaltet.

03 Wie geprüft wird

Die Prüfung ist eine Kette: Erst muss eine Nachricht überhaupt dem Schema entsprechen, dann ihre Struktur stimmen, dann ihr Verlauf, und erst dann lässt sich beurteilen, ob Auftrag und Zustandsmeldung zusammenpassen.

Schema

Jede Nachricht gegen das offizielle JSON-Schema der Zielversion. 2.1.0 und 3.0 werden getrennt geführt, denn 3.0 hat Felder umbenannt — ein Werkzeug, das nur eine Version kennt, meldet beim Versionswechsel Fehler, die keine sind.

Struktur

Pflichtfelder, Datentypen, ISO-8601-Zeitstempel in UTC, semantische Version, Sammlungen als Arrays. Enumerationen werden gegen die erlaubten Werte der Zielversion geprüft.

Verlauf

headerId-Lücken und -Rückschritte, Zeitstempel-Rückschritte, Node- und Edge-Sequenzen, Aktions-Lebenszyklen von der Erteilung bis zum Endzustand, Batterie- und Safety-Meldungen.

Zusammenhang

Wird eine erteilte Order quittiert? Passt der gemeldete Fortschritt zu ihr? Bleibt lastNodeSequenceId innerhalb der Order monoton? Diese Korrelation läuft je Fahrzeug.

Schweregrade

Critical	Die Nachricht ist nicht spezifikationskonform; das Gegenüber darf sie ablehnen.
Major	Klare Abweichung mit Wirkung auf den Ablauf.
Minor	Abweichung ohne unmittelbare Wirkung, aber berichtspflichtig.
Info	Hinweis zur Einordnung, kein Mangel.

Eine Flotte ist nicht ein Fahrzeug mal zehn

VDA5050 zählt headerIds je Topic, und das Topic trägt Hersteller und Seriennummer. Genau so wertet der Analyzer: jede Sequenz, jeder Lebenszyklus, jede Order-State-Korrelation je Fahrzeug. Würde man alle Nachrichten in eine Reihe legen, sähen zwei Fahrzeuge, die beide bei headerId 1 anfangen, aus wie ein Sender, der ständig zurückspringt — hunderte Fehlbefunde, die keine sind. Jeder Befund trägt zudem „Hersteller/Seriennummer“ und lässt sich danach filtern.

Punktzahl und Urteil

Aus den Befunden entsteht je Prüfbereich eine Punktzahl und daraus ein Gesamtwert sowie ein Urteil (PASS, WARNING, FAIL). Beides ist nur so viel wert wie der Prüfumfang, der daneben steht — deshalb die nächste Seite.

04 Zusage gegen Wirklichkeit

Jedes Fahrzeug erklärt in seinem Factsheet, was es beherrscht. 22 Prüfungen halten den Verkehr an diese Erklärung — und beantworten damit eine andere Frage als der Rest des Regelsatzes: nicht „folgt diese Nachricht dem Standard“, sondern „passt sie zu diesem Fahrzeug“.

Aktionen

Jeder actionType aus Order und instantActions muss im Katalog des Fahrzeugs stehen — im richtigen Gültigkeitsbereich, mit deklariertem blockingType und mit den Parametern, die das Fahrzeug erwartet.

Optionale Parameter

Die Spezifikation ist hier scharf: was nicht gelistet ist, gilt als nicht unterstützt. Der Analyzer meldet optionale Felder, die gesendet werden, obwohl das Fahrzeug sie nie zugesagt hat.

Grenzen

Nachrichtenlänge, Länge und Zeichenvorrat der Bezeichner, Knoten und Kanten je Order, Aktionen je Knoten, nodeStates und actionStates je State — beide Richtungen werden gegen die deklarierten Grenzen gemessen.

Zeiten

Mindestabstand zwischen Orders, Mindest- und Regelabstand zwischen States. Geprüft mit einer Toleranz, die je Analyse eingestellt wird — Prozentband plus fester Zuschlag, weil die Uhren der beteiligten Systeme nicht synchron laufen.

Wer war es?

Jeder Befund dieser Gruppe nennt die Seite, die ihn verursacht hat: das Leitsystem, das etwas Undeklariertes verlangt, oder das Fahrzeug, das seine eigene Zusage nicht einhält. Im Bericht ist das ein Filter. Damit endet die häufigste Inbetriebnahme-Diskussion nicht in Behauptungen, sondern in einer Zuordnung.

Ohne Factsheet keine Aussage

Fehlt das Factsheet, oder erklärt es einen Bereich nicht, meldet die betroffene Prüfung „ohne Daten“ — niemals „bestanden“. Ein Bericht, der Konformität gegen eine Referenz behauptet, die niemand geliefert hat, wäre genau die falsche Sicherheit, gegen die dieses Werkzeug antritt.

Drei Wege zum Factsheet

Aus dem Verkehr, wenn die Aufzeichnung das factsheet-Topic enthält. Importiert als eigene Datei — Logexporte aus einem Leitsystem tragen das Topic selten. Oder beim Einrichten der Analyse ausgewählt; diese Wahl gewinnt gegen das Factsheet im Verkehr, und der Bericht hält fest, welches galt. Ein importiertes Factsheet wird beim Hochladen selbst geprüft und lässt sich formatiert lesen — es wird zur Referenz von 22 Prüfungen, also nicht ungelesen übernommen.

05 Bericht und Nachvollziehbarkeit

Ein Prüfbericht, der nur Fehler aufzählt, taugt nicht für eine Abnahme. Der Analyzer legt neben das Ergebnis, wie es zustande kam.

Aktiv

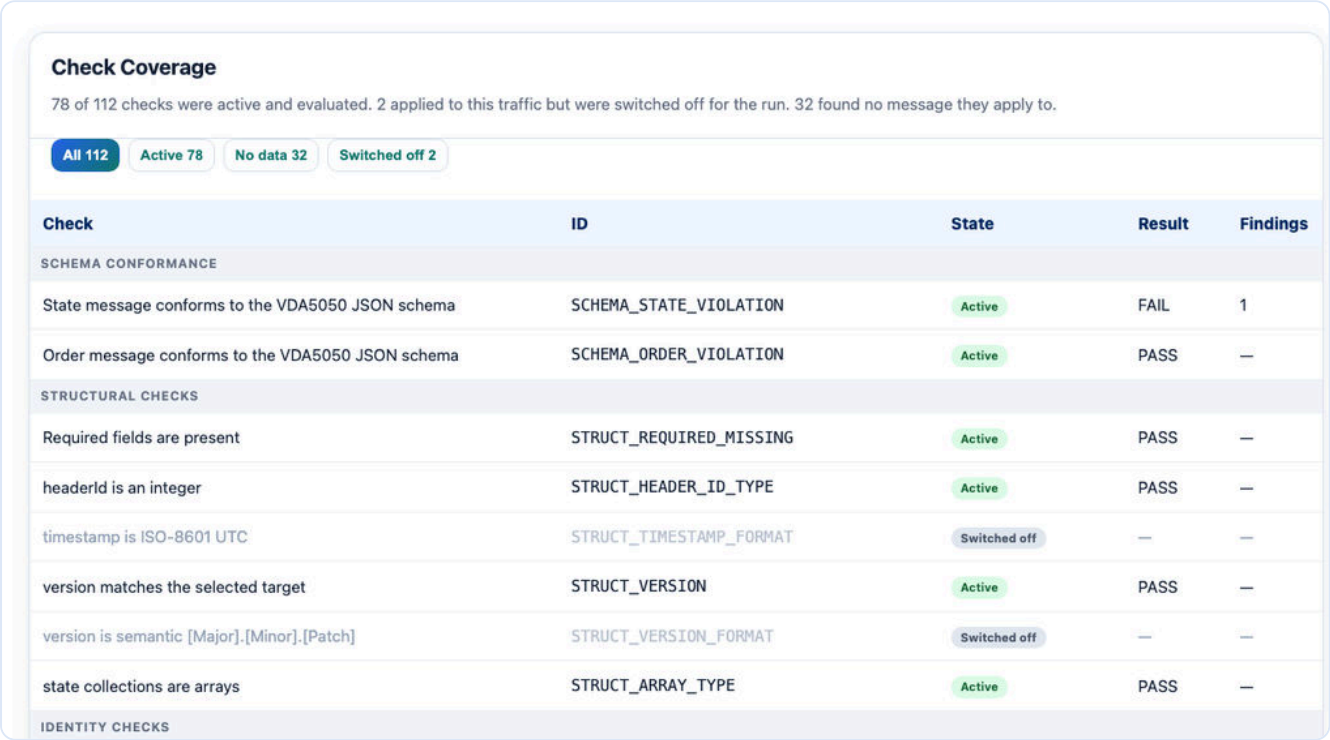
Die Regel lief und hat geurteilt — PASS oder FAIL, mit der Anzahl der Befunde.

Ohne Daten

Die Regel lief, fand aber keine Nachricht, auf die sie zutrifft. Sie besteht nicht still, sondern wird als „ohne Daten“ ausgewiesen.

Abgeschaltet

Bewusst abgewählt — und im Bericht sichtbar, nicht versteckt.



Check Coverage

78 of 112 checks were active and evaluated. 2 applied to this traffic but were switched off for the run. 32 found no message they apply to.

All 112 | Active 78 | No data 32 | Switched off 2

Check	ID	State	Result	Findings
SCHEMA CONFORMANCE				
State message conforms to the VDA5050 JSON schema	SCHEMA_STATE_VIOLATION	Active	FAIL	1
Order message conforms to the VDA5050 JSON schema	SCHEMA_ORDER_VIOLATION	Active	PASS	—
STRUCTURAL CHECKS				
Required fields are present	STRUCT_REQUIRED_MISSING	Active	PASS	—
headerId is an integer	STRUCT_HEADER_ID_TYPE	Active	PASS	—
timestamp is ISO-8601 UTC	STRUCT_TIMESTAMP_FORMAT	Switched off	—	—
version matches the selected target	STRUCT_VERSION	Active	PASS	—
version is semantic [Major].[Minor].[Patch]	STRUCT_VERSION_FORMAT	Switched off	—	—
state collections are arrays	STRUCT_ARRAY_TYPE	Active	PASS	—
IDENTITY CHECKS				

Prüfumfang eines Laufs: 78 von 112 aktiv, 32 ohne passende Daten, 2 abgeschaltet.

Prüfprotokoll

Jeder Lauf hält fest: Werkzeug- und Regelsatzversion, Zielversion der Spezifikation, die verwendeten Schema-Revisionen, die Anzahl der Eingangsnachrichten, eine Prüfsumme über die Eingabe sowie Prüfungen gesamt / aktiviert / ausgewertet. Damit ist ein Ergebnis wiederholbar und ein Bericht überprüfbar.

Ausgabe

Jeder Bericht steht als JSON und als Markdown zum Download bereit — für die Ablage im Projekt, für den Anhang einer Abnahme oder zur Weiterverarbeitung.

06 Befunde im Kontext

Ein Befund ist erst brauchbar, wenn man ihn nachvollziehen kann. Deshalb führt jeder Eintrag zur auslösenden Nachricht.

Die Stelle

Der Befund nennt Prüfung, Schweregrad, Feldpfad und die Nachricht, in der er aufgetreten ist — und zeigt sie mit hervorgehobenem Feld, nicht nur als Zeilennummer.

Der Kontext

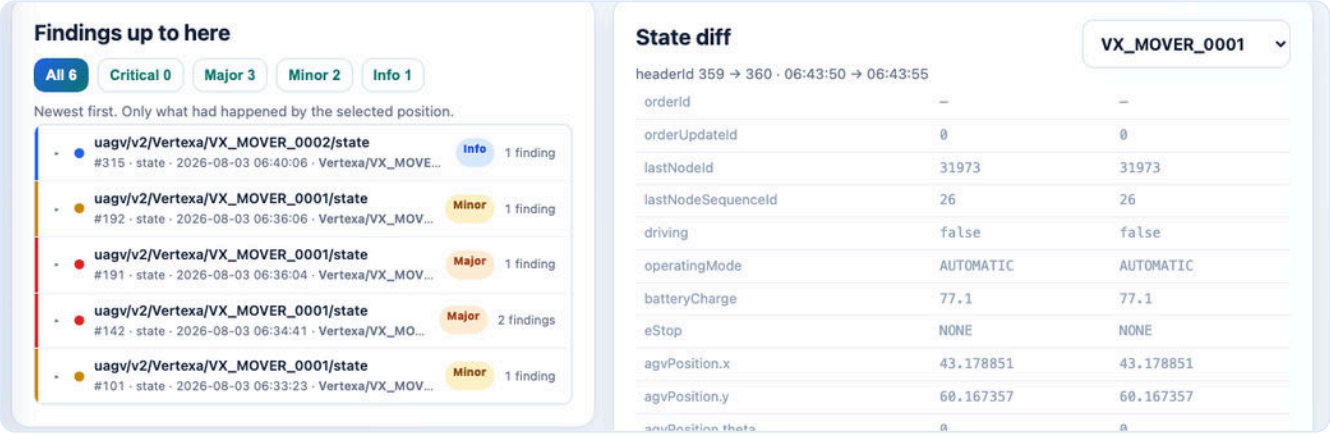
Vorherige, aktuelle und nächste Nachricht liegen nebeneinander und scrollen im Gleichlauf. Umschaltbar zwischen dem Strom des Fahrzeugs — vorherige und nächste Nachricht desselben Typs desselben Fahrzeugs — und dem Log, wo die Nachbarn oft zu einem anderen Roboter gehören.

Die Sicht

Filter nach Schweregrad und nach Fahrzeug; Gruppierung nach Prüfung oder nach Nachricht — je nachdem, ob man ein Muster oder einen einzelnen Vorfall sucht.

Der Vergleich

Zwei aufeinanderfolgende State-Nachrichten desselben Fahrzeugs Feld für Feld nebeneinander, Änderungen hervorgehoben. Die Frage „was hat sich geändert“ beantwortet sich damit ohne JSON-Vergleich von Hand.



Befunde nach Nachricht gruppiert, daneben der Feld-für-Feld-Vergleich.

07 Layout und Visualisierung

Aufträge beziehen sich auf einen Fahrkurs. Wenn Auftrag und Layout nicht zusammenpassen, lehnt das Fahrzeug ab — und ohne Bild sucht man lange.

1

LIF prüfen

13 Regeln nach VDMA LIF 1.0.0: Schema, Knoten- und Kanten-Integrität, Erreichbarkeit im Netz, Stationsbezüge. Das Ergebnis ist ein eigener Bericht mit Layoutzeichnung.

2

Order über Layout

Der Pfad einer Order wird über das importierte Layout gelegt. Der Abgleich meldet Knoten für Knoten und Kante für Kante, ob der Auftrag ins Layout passt, und benennt im Fehlerfall jeden fehlenden Knoten und jede fehlende Kante.

3

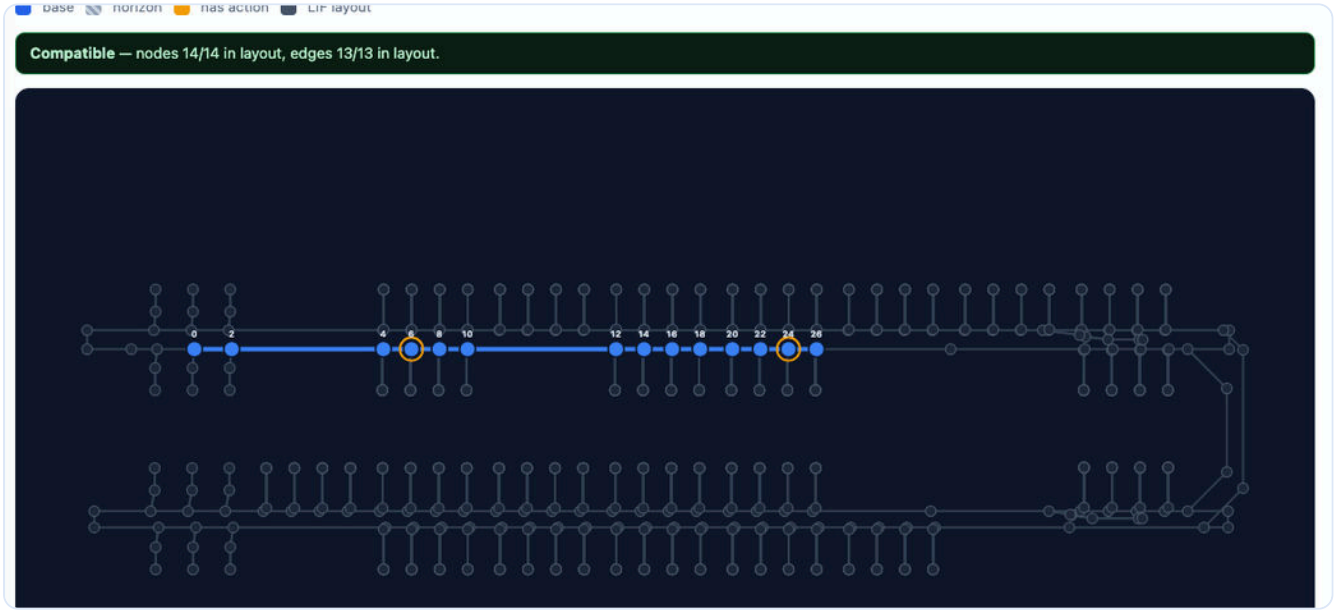
Zustandsverlauf

Neben dem Order-Pfad zeigt die Visualisierung den Verlauf ausgewählter Zustandsfelder über die Zeit — etwa Position, Batterie oder Betriebsart.

4

Bleibt beim Bericht

Das Layout wird schon bei der Analyse ausgewählt und bleibt dem Bericht zugeordnet. Wer ihn später öffnet, sieht dieselbe Grundlage.



Order-Pfad über dem LIF-Layout mit Abgleichmeldung.

08 Live View

Während ein Profil aufzeichnet, lässt sich der Anlage bei der Arbeit zusehen — auf demselben Layout, das auch die Berichte verwenden.

- Alle Fahrzeuge auf dem LIF-Layout, jedes in eigener Farbe.
- Zuletzt erteilter Auftrag als Pfad, letzte gemeldete Position als Marke.
- Fahrzeugtabelle mit Auftrag, letztem Knoten, Position, Batterie, Zustand und Alter der Meldung.
- Befunde erscheinen, während sie entstehen, filterbar nach Schweregrad und Fahrzeug.
- State-Diff eines im Dropdown gewählten Fahrzeugs daneben.
- Aktualisierung im Sekundentakt; im Hintergrund pausiert die Seite.

Zwei Fahrzeuge mit eigenen Routen und letzter bekannter Position.

Der eigentliche Punkt

Die Karte zeichnet den zuletzt gesendeten Auftrag. Die Tabelle zeigt, was das Fahrzeug selbst meldet. Weichen beide voneinander ab, steht es dort — genau in dieser Lücke sitzen die meisten Inbetriebnahmefehler.

09 Replay

Derselbe Blick, aber für alles, was schon passiert ist: abgeschlossene Mitschnitte und importierte Logdateien lassen sich abspielen.

- Vorwärts, rückwärts, Schritt für Schritt, bis zwanzigfache Geschwindigkeit.
- Die Abstände sind die echten Nachrichtenabstände, gedeckelt, damit lange Pausen eine Vorführung nicht aufhalten.
- Die Befundliste zeigt nur, was zum gewählten Zeitpunkt bereits bekannt war — kein Wissen aus der Zukunft.
- Das überlagerte LIF-Layout ist umschaltbar, ohne den Bericht zu verändern.

Replay an Position 402 von 433 mit Befunden bis zu diesem Zeitpunkt.

Wofür man es benutzt

Für die Nachbesprechung. Ein Vorfall lässt sich vor dem Lieferanten abspielen, statt ihn zu beschreiben; alle im Raum sehen dieselbe Abfolge, mit denselben Befunden an denselben Stellen.

10 Dauerbetrieb und Reporting

Ein Mitschnitt über ein Wochenende bringt hunderttausend Nachrichten. Niemand sieht die durch — der Analyzer meldet sich stattdessen selbst.

1. **Bericht nach Zeitplan**

   Alle X Stunden ein Auszug dessen, was seit dem letzten Bericht auffiel. Auch ein ruhiger Zeitraum wird gemeldet, damit Stille eindeutig bleibt.

2. **Alarm nach Schwelle**

   Oder erst, wenn genug zusammengekommen ist: ab X Befunden, wahlweise nur Critical, Critical und Major oder alle Schweregrade.

3. **Empfänger je Profil**

   Die Adressen hängen am Capture-Profil. Verschiedene Anlagen können verschiedene Verteiler haben.

4. **Fortsetzbar**

   Ein Neustart nimmt die Analyse dort wieder auf, wo sie stand. Nichts wird doppelt gewertet, nichts übersprungen.

Was in der E-Mail steht

Zeitraum, Auslöser und Anzahl der neuen Befunde, aufgeschlüsselt nach Schweregrad, nach Fahrzeug und nach Prüfung — mit direktem Link in den laufenden Mitschnitt.


LogIQ
VDA5050 Analyzer

Hall A line: capture report

41 new findings since 2026-08-02 06:00.

Capture profile	Hall A line
Run	#5
Period	since 2026-08-02 06:00
Trigger	41 Major+ findings since the last report (threshold 20)
New findings	41

BY SEVERITY
- Critical: 2
- Major: 27
- Minor: 11
- Info: 1

BY VEHICLE
- Vertexa/VX_MOVER_0001: 26
- Vertexa/VX_MOVER_0002: 15

BY CHECK
- [Critical] SCHEMA_STATE_VIOLATION - 22
- [Major] STATE_ENTRY_INCOMPLETE - 12
- [Minor] HEADER_GAP - 7

Open the capture

https://vda5050.logistixpert.de/mqtt

Auszug eines Schwellenwert-Alarms.

11 Team, Projekte und Kontingente

Prüfergebnisse gehören nicht auf einen Laptop. Der Analyzer kennt Gruppen, Projekte und klare Rechte.

1. ### Enterprise

   Alle Mitglieder sehen, was die Gruppe produziert hat; jede Liste nennt den Besitzer. Geteilte Sichtbarkeit heißt aber nicht geteiltes Löschrecht: fremde Daten löschen dürfen nur Superuser.

2. ### Superuser

   Verwaltet Mitgliedschaft und Projekte und darf fremde Daten löschen. Konten löschen bleibt der Administration vorbehalten.

3. ### Fallback-Superuser

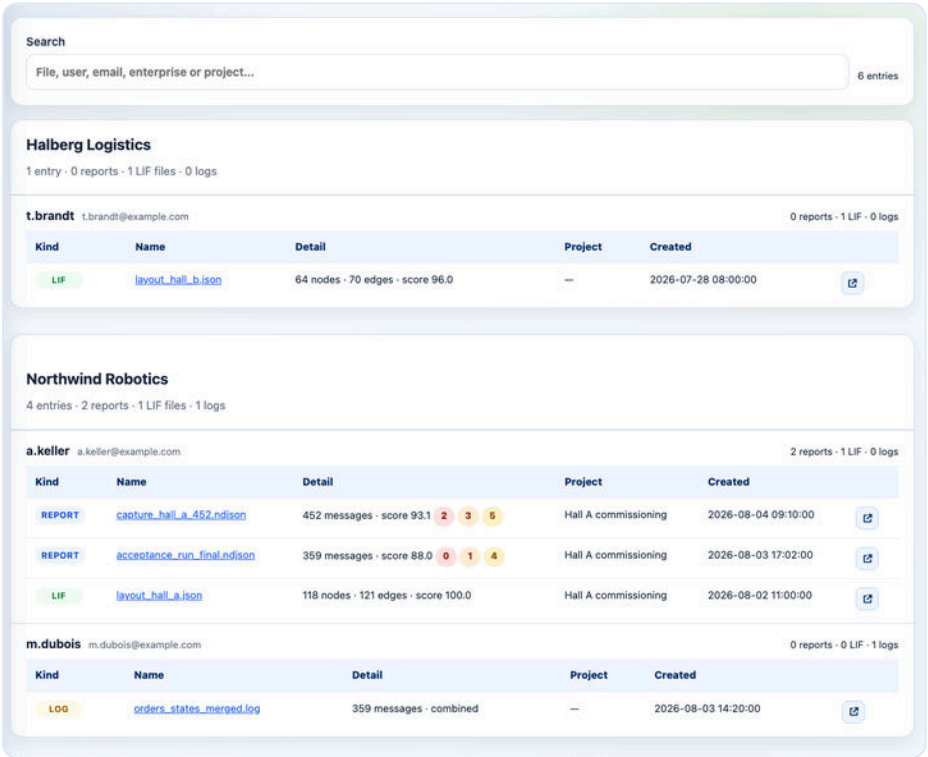
   Genau einer pro Enterprise. Schließt ein Mitglied sein Konto, gehen dessen Logs, Layouts, Berichte und Mitschnitte an diese Person über, statt mit dem Konto verloren zu gehen.

4. ### Projekte

   Logs, Layouts, Profile und Berichte lassen sich einem Auftrag zuordnen und danach filtern. Ein abgeschlossenes Projekt verschwindet aus den Listen, ohne gelöscht zu werden, und kommt beim Wiederöffnen zurück.

5. ### Kontingente

   Analysierte Nachrichten werden gezählt. Ein Enterprise rechnet gegen ein gemeinsames Kontingent, einzelne Nutzer gegen ihr eigenes. Gezählt wird der Verbrauch, nicht der Bestand — ein gelöschter Bericht gibt kein Kontingent zurück.



Administratorsicht über alle Enterprises, Nutzer und Artefakte, mit Volltextsuche.

12 Betrieb und Sicherheit

Mitschnitte aus einer Anlage sind sensibel. Der Analyzer ist so gebaut, dass sie das Werk nicht verlassen müssen.

Im eigenen Haus

Docker Compose, eine Datei, ein Kommando. Betrieb auf eigener Hardware, im eigenen Netz oder als gehostete Instanz — die Wahl ändert nichts am Funktionsumfang.

Verschlüsselte Verbindung

MQTT über TLS, wahlweise mit dem eigenen CA-Zertifikat der Anlage. Das Proxy-Ziel hat denselben Schalter, damit die gespiegelte Verbindung nicht schwächer ist als die gespiegelte.

Zugangsdaten verschlüsselt

Broker-Passwörter werden verschlüsselt gespeichert, nicht im Klartext. Sie bleiben für Reconnects nutzbar, sind aber nicht lesbar.

Lizenz ohne Anruf nach Hause

Im eigenen Netz trägt die Installation eine signierte Lizenzdatei, lokal geprüft. Nach Ablauf vierzehn Tage Karenz, danach schreibgeschützt — alles Aufgezeichnete bleibt erreichbar.

Berichte unter eigenem Namen

Mit Enterprise-Lizenz tragen Bericht, Export und E-Mail-Digest Name und Zeichen des Kunden. Der Hinweis, was das Werkzeug ist, bleibt darunter stehen.

Nicht aufzählbare Links

Berichte, Läufe, Importe und Layouts hängen an unerratbaren Kennungen. Zusätzlich prüft jeder Zugriff die Besitzverhältnisse.

Rollen mit Grenzen

Mitglied, Superuser, Administrator. Wer was löschen darf, ist an einer Stelle im Code definiert und wird bei jeder Aktion geprüft.

Konto schließen

Nutzer können ihr Konto selbst löschen. Ohne Enterprise geht alles mit; in einer Gruppe bleibt die Arbeit bei der Gruppe und geht an den Fallback-Superuser über.

Belegte Qualität

654 automatisierte Tests laufen vor jeder Auslieferung, darunter Regressionstests gegen feste Beispieldaten.

Daten und Aufbewahrung

Uploads, bereinigte Nachrichten, Berichte und aufgezeichnete Mitschnitte liegen in einem Datenverzeichnis der Instanz. Gelöscht wird über die Anwendung — einzeln oder als Auswahl mehrerer Einträge. Beim Löschen eines Logs bleiben die daraus erstellten Berichte erhalten, weil ein Bericht das Protokoll eines Prüflaufs ist und seine Quelle überdauert. Für die Kontingentrechnung gilt das Gegenteil: verbrauchte Nachrichten bleiben verbraucht.

13 Technische Daten

Spezifikationen	VDA5050 2.1.0 und 3.0 · M2X 0.9.0 (Vorschau) · VDMA LIF 1.0.0
Prüfungen	112 für VDA5050 und M2X, 13 für LIF; einzeln aktivierbar, Auswahl je Analyse und je Capture-Profil
Nachrichtenarten	order, state, connection, instantActions, factsheet, visualization; M2X: Lastaufnahme, Tür, Aufzug, Signalleuchte, Ladegerät, Transportauftrag
Eingabeformate	.log, .txt, .json, .ndjson — Rohmitschnitte, MQTT-Captures und NDJSON-Exporte
Ausgabe	Bericht als HTML, JSON und Markdown; Mitschnitte als NDJSON exportierbar
Live-Betrieb	MQTT 3.1.1/5 über paho-mqtt, unverschlüsselt oder über TLS (Port 8883, eigene CA möglich), Benutzer/Passwort, mehrere Profile gleichzeitig, optionales Spiegeln auf einen zweiten Broker
Benachrichtigung	E-Mail über SMTP, Zeitplan oder Schwellenwert, Empfänger je Profil
Betrieb	Docker Compose (Anwendung und Broker), SQLite als Datenhaltung, Betrieb on-premise oder gehostet
Bedienung	Webanwendung, keine Installation am Arbeitsplatz; Oberfläche in Englisch
Datenhaltung	Uploads, bereinigte Nachrichten, Berichte und Mitschnitte liegen in einem Datenverzeichnis; Löschen erfolgt über die Anwendung

Voraussetzungen

- Ein Rechner oder eine VM mit Docker; für dauerhafte Mitschnitte ausreichend Plattenplatz.
- Lesezugriff auf den MQTT-Broker der Anlage, für das Spiegeln zusätzlich Schreibzugriff auf den Zielbroker.
- Für E-Mail-Berichte ein SMTP-Zugang; ohne ihn werden Mails in eine Datei geschrieben.

Grenzen und Annahmen

- Der Analyzer wertet aus, was er sieht. Nachrichten, die den Broker nie erreichen, kann er nicht beurteilen.
- Ein Replay zeigt bis zu 20 000 Nachrichten; längere Läufe werden gekürzt und die Kürzung im Werkzeug benannt.
- M2X folgt einer Vorabversion der Spezifikation; Ergebnisse sind als Vorschau gekennzeichnet.
- Für die Zuordnung von Befunden zu Fahrzeugen müssen Hersteller und Seriennummer im Topic bzw. in der Nachricht stehen.

14 Prüfkatalog

Alle Prüfungen des Werkzeugs, gruppiert nach Bereich. Bezeichner und Kurztexte entsprechen denen in der Anwendung und im Bericht; jede Prüfung führt dort zusätzlich die Fundstelle in der Spezifikation.

VDA5050 und M2X (112)

Schema Conformance (2)

`SCHEMA_STATE_VIOLATION`
State message conforms to the VDA5050 JSON schema

`SCHEMA_ORDER_VIOLATION`
Order message conforms to the VDA5050 JSON schema

Structural Checks (6)

`STRUCT_REQUIRED_MISSING`
Required fields are present

`STRUCT_HEADER_ID_TYPE`
headerId is an integer

`STRUCT_TIMESTAMP_FORMAT`
timestamp is ISO-8601 UTC

`STRUCT_VERSION`
version matches the selected target

`STRUCT_VERSION_FORMAT`
version is semantic [Major].[Minor].[Patch]

`STRUCT_ARRAY_TYPE`
state collections are arrays

Identity Checks (2)

`IDENTITY_MANUFACTURER_CHANGED`
Report streams covering more than one manufacturer

`IDENTITY_SERIAL_CHANGED`
Report streams covering more than one vehicle

Header / Time Checks (4)

`HEADER_DUPLICATE`
No duplicate headerId values

`HEADER_REGRESSION`
headerId increases monotonically

`HEADER_GAP`
Detect headerId gaps

`TIMESTAMP_REGRESSION`
timestamp increases or stays equal

Node / Edge Checks (7)

`STATE_ENTRY_TYPE`
nodeStates/edgeStates entries are objects

`STATE_ENTRY_INCOMPLETE`
nodeId/edgeId and sequenceId are present

`STATE_SEQUENCE_TYPE`
sequenceId is an integer

`STATE_SEQUENCE_REGRESSION`
sequenceIds increase within a message

`NODE_EDGE_SEQUENCE_OVERLAP`
Report node/edge sequenceId overlaps

`LAST_NODE_REGRESSION`
lastNodeSequenceId does not decrease per orderId

`LAST_NODE_JUMP`
lastNodeSequenceId does not jump by more than 4

Action Lifecycle Checks (5)

`ACTION_ENTRY_TYPE`
actionStates entries are objects

`ACTION_REQUIRED_MISSING`
actionId and actionStatus are present

`ACTION_STATUS_INVALID`
actionStatus uses an allowed enum value

`ACTION_INVALID_TRANSITION`
Action transitions are allowed

`ACTION_NOT_TERMINAL`
Actions reach a terminal state

Operating Mode Checks (1)

`STATE_OPERATING_MODE_INVALID`
operatingMode uses an allowed enum value

Driving / Velocity Checks (4)

`PAUSED_AND_DRIVING`
paused=true and driving=true are not both active

`VELOCITY_MISSING_WHILE_DRIVING`
Velocity is present when driving=true

`DRIVING_ZERO_VELOCITY`
Report driving=true with zero velocity

`NOT_DRIVING_WITH_VELOCITY`
Report driving=false with velocity

Battery Checks (2)

`BATTERY_CHARGE_RANGE`
battery charge is between 0 and 100

`BATTERY_CHARGE_JUMP`
Detect battery charge jumps

Safety Checks (4)

`SAFETY_FIELD_MISSING`
safetyState contains the eStop and fieldViolation fields

`SAFETY_ESTOP_VALUE_INVALID`
emergency-stop value uses an allowed enum

`ESTOP_WHILE_DRIVING`
Report active eStop with driving=true

`FIELD_VIOLATION_WHILE_DRIVING`
Report fieldViolation=true with driving=true

Error Reporting Checks (1)

`STATE_ERROR_STRUCTURE`
errors contain errorType and a valid errorLevel

Order Structural Checks (5)

`ORDER_REQUIRED_MISSING`
Required order fields are present

`ORDER_HEADER_ID_TYPE`
Order headerId is an integer

`ORDER_TIMESTAMP_FORMAT`
Order timestamp is ISO-8601 readable

`ORDER_VERSION`
Order version matches the selected target

`ORDER_ARRAY_TYPE`
nodes and edges are arrays

Order Update Checks (3)

`ORDER_UPDATE_DUPLICATE`
No duplicate orderUpdateId values

`ORDER_UPDATE_REGRESSION`
orderUpdateId increases monotonically

`ORDER_UPDATE_GAP`
Detect orderUpdateId gaps

VDA5050 und M2X (Fortsetzung)

Order Graph Checks (7)

ORDER_SEQUENCE_TYPE
node and edge sequenceId values are integers

ORDER_SEQUENCE_PARITY
Node and edge sequenceId parity is valid

ORDER_SEQUENCE_ORDER
Nodes and edges are sorted by sequenceId

ORDER_DUPLICATE_ID
Node, edge and action IDs are unique

ORDER_EDGE_ENDPOINT_UNKNOWN
Edges reference existing nodes

ORDER_GRAPH_CHAIN
Graph sequence is continuous

ORDER_RELEASED_CONTIGUITY
Released base is contiguous

Order Position Checks (1)

ORDER_POSITION_MISSING
Node positions contain coordinates and mapId

Order Action Checks (2)

ORDER_ACTION_REQUIRED_MISSING
Order actions contain required fields

ORDER_BLOCKING_TYPE
blockingType uses allowed values

Order Trajectory Checks (1)

ORDER_TRAJECTORY_STRUCTURE
Edge trajectory structure is valid

Connection Checks (4)

CONN_REQUIRED_MISSING
Required connection fields are present

CONN_HEADER_ID_TYPE
Connection headerId is an integer

CONN_VERSION
Connection version matches the selected target

CONN_STATE_INVALID
connectionState uses an allowed enum value

Instant Action Checks (6)

IA_REQUIRED_MISSING
Required instantActions fields are present

IA_HEADER_ID_TYPE
instantActions headerId is an integer

IA_VERSION
instantActions version matches the selected target

IA_ACTION_REQUIRED_MISSING
Instant actions contain required fields

IA_BLOCKING_TYPE
Instant action blockingType uses allowed values

IA_DUPLICATE_ACTION_ID
Instant action IDs are unique

Factsheet Checks (3)

FS_REQUIRED_MISSING
Required factsheet objects are present

FS_HEADER_ID_TYPE
Factsheet headerId is an integer

FS_VERSION
Factsheet version matches the selected target

Mission Lifecycle Checks (8)

MISSION_STATE_UNKNOWN_ORDER
State references a known orderId

MISSION_ORDER_NOT_ACKED
State acknowledges the orderId after an order

MISSION_ORDER_ACK_SLOW
Order is acknowledged within the timeout

MISSION_NODE_NOT_IN_ORDER
nodeStates contain only nodes from the order

MISSION_EDGE_NOT_IN_ORDER
edgeStates contain only edges from the order

MISSION_ACTION_NOT_IN_ORDER
actionStates contain only actions from the order

MISSION_ORDER_UPDATE_NOT_REJECTED
Stale order updates are rejected

MISSION_INSTANT_ACTION_NOT_REFLECTED
instantActions are reflected in state

VDA5050 und M2X (Fortsetzung)

Capability Checks (22)

CAP_ACTION_NOT_DECLARED
Action type is declared in the factsheet

CAP_ACTION_SCOPE_INVALID
Action is used within its declared scope

CAP_ACTION_BLOCKING_TYPE_INVALID
blockingType is one the vehicle declares

CAP_ACTION_PARAM_UNDECLARED
Action parameters are declared

CAP_ACTION_PARAM_REQUIRED_MISSING
Required action parameters are sent

CAP_ACTION_PARAM_TYPE_MISMATCH
Action parameter values match the declared type

CAP_ACTION_CANCEL_NOT_ALLOWED
cancelOrder respects cancelAllowed

CAP_ACTION_PAUSE_NOT_ALLOWED
startPause respects pauseAllowed

CAP_OPTIONAL_PARAM_UNSUPPORTED
Only declared optional parameters are sent

CAP_OPTIONAL_PARAM_REQUIRED_MISSING
Parameters declared REQUIRED are sent

CAP_OPTIONAL_PARAM_UNKNOWN_PATH
Declared optional parameters exist in the schema

CAP_LIMIT_MSG_LEN
Messages stay within the declared length

CAP_LIMIT_ID_LEN
Identifiers stay within the declared length

CAP_LIMIT_ID_NUMERICAL
Identifiers are numeric when declared so

CAP_LIMIT_LOAD_ID_LEN
loadId stays within the declared length

CAP_LIMIT_TOPIC_LEN
Topic elements stay within the declared length

CAP_LIMIT_ARRAY_LEN_ORDER
Orders respect the declared array limits

CAP_LIMIT_ARRAY_LEN_STATE
States respect the declared array limits

CAP_TIMING_ORDER_INTERVAL
Orders keep the declared minimum interval

CAP_TIMING_STATE_INTERVAL_MIN
States keep the declared minimum interval

CAP_TIMING_STATE_INTERVAL_DEFAULT
States arrive within the declared default interval

CAP_TIMING_VISUALIZATION_INTERVAL
Visualization keeps the declared interval

M2X Schema Conformance (6)

M2X_SCHEMA_LHS_VIOLATION
Load handling station messages conform to the M2X JSON schema

M2X_SCHEMA_DOOR_VIOLATION
Door messages conform to the M2X JSON schema

M2X_SCHEMA_ELEVATOR_VIOLATION
Elevator messages conform to the M2X JSON schema

M2X_SCHEMA_SIGNAL_VIOLATION
Signaling device messages conform to the M2X JSON schema

M2X_SCHEMA_CHARGING_VIOLATION
Charging device messages conform to the M2X JSON schema

M2X_SCHEMA_ORDER_VIOLATION
Transport order messages conform to the M2X JSON schema

M2X Lifecycle Checks (6)

M2X_REQUEST_NOT_REFLECTED
Requests appear in the entity state

M2X_MULTIPLE_ACTIVE_REQUESTS
Only one request is active per entity

M2X_REQUEST_INVALID_TRANSITION
Request status transitions are allowed

M2X_INTERACTION_SEQUENCE
Interaction steps follow the specified order

M2X_STATE_HEARTBEAT_MISSED
Load handling stations publish state at least every 30 seconds

M2X_REQUEST_TYPE_NOT_SUPPORTED
Requests match the addressed interface

VDMA LIF 1.0.0 (13)

Schema Conformance (1)

`SCHEMA_LIF_VIOLATION`
LIF file conforms to the LIF JSON schema

LIF Structural Checks (4)

`LIF_DUPLICATE_LAYOUT_ID`
layoutId is unique

`LIF_DUPLICATE_NODE_ID`
nodeId is unique across the file

`LIF_DUPLICATE_EDGE_ID`
edgeId is unique across the file

`LIF_DUPLICATE_STATION_ID`
stationId is unique across the file

LIF Graph Checks (5)

`LIF_EDGE_START_NODE_UNKNOWN`
Edge startNodeId is part of the same layout

`LIF_EDGE_END_NODE_UNKNOWN`
Edge endNodeId references an existing node

`LIF_ISOLATED_NODE`
Every node is connected by at least one edge

`LIF_STATION_NODE_UNKNOWN`
Station interactionNodeIds reference existing nodes

`LIF_ROTATION_CONFLICT_AT_NODE`
Rotation allowances agree where edges meet

LIF Vehicle Type Checks (2)

`LIF_DUPLICATE_VEHICLE_TYPE_NODE_PROPERTY`
Only one vehicleTypeNodeProperty per vehicle type per node

`LIF_DUPLICATE_VEHICLE_TYPE_EDGE_PROPERTY`
Only one vehicleTypeEdgeProperty per vehicle type per edge

LIF Trajectory Checks (1)

`LIF_TRAJECTORY_KNOT_VECTOR_SIZE`
Trajectory knotVector has the expected size

LogIQ
VDA5050 Analyzer

Der nächste Schritt

1 **Demonstration**

Vorführung an einem mitgelieferten Demopaket — ein sauberer und ein fehlerbehafteter Datensatz mit passendem Layout. Sofort vorführbar, ohne Zugriff auf Ihre Anlage.

2 **Pilot mit Ihren Daten**

Ein Mitschnitt aus Ihrer Anlage, ausgewertet und gemeinsam durchgesprochen. Ergebnis ist ein Bericht, den Sie behalten.

3 **Betrieb**

Gehostet unter vda5050.logistixpert.de oder im eigenen Netz per Docker.

LogistiXpert

Matthias Merz
LogistiXpert
m.merz@logistixpert.de
www.logistixpert.de
vda5050.logistixpert.de