

EUROCOMMAND GMBH

CIMgate.HUB

Architekturdokumentation & Technische Spezifikation

Sichere organisationsübergreifende Einsatzzusammenarbeit
für CommandX (CX5)

STAND

27.10.2026

Inhalt

CIMgate.HUB

- Problem
- Ziele
- Dokumentation
- Links

Arten der Vernetzung

- UC1 - Instanzspiegelung
- UC2 - Befehlsstellen-Synchronisation
- UC3 - Ad-hoc-Synchronisation
- UC4 - Vorgeplante Einsatz-Synchronisation
- Out of Scope

Funktionale Anforderungen

- Wesentliche Anforderungen
- Funktionale Anforderungen

Überblick

1. Einführung und Ziele

- Anforderungen
- Aufgabenstellung
- Qualitätsziele
- Stakeholder

2. Randbedingungen

- Technische Randbedingungen
- Organisatorische Randbedingungen
- Konventionen

3. Systemkontext

- Fachlicher Kontext
- Technischer Kontext

4. Lösungsstrategie

- Kernideen
- Sync-Profile je Vernetzungsart
- Kombination von Vernetzungsarten

Versionshandling
Auslieferungsstufen

5. Bausteinsicht

5.1 CIMgate.HUB Knoten (Whitebox)
5.2 Ebene 2
5.3 Ebene 3

6. Laufzeitsicht

6.1 Kopplung (UC1, UC2, UC4)
6.2 Snapshot / Initialabgleich
6.3 Delta-Sync
6.4 Verbindungsabbruch und Wideranlauf
6.5 Teilen beenden
6.6 Einladungscode
6.7 Schema Abgleich
6.X ...

7. Verteilungssicht

Infrastruktur Ebene 1

9. Architekturentscheidungen

ADR-1: Strategie zum Neuladen der NATS-Konfiguration

11. Risiken und technische Schulden

11.1 Risiken
11.2 Technische Schulden

Offene Fragen

F1
F2
F3
F4
F5-7 Data handling
F8
F10 - Kombination von Vernetzungsarten
F11
Dev Notes

CIMgate.HUB

CIMgate.HUB ist ein **Synchronisations-Gateway** je Organisation, über das getrennte **CommandX (CX5)**-Führungs- und Stabsinstanzen operative Einsätze miteinander teilen können: selektiv je Modul, mit Lese-/Schreibrechten, bidirektional, nahezu in Echtzeit und mit vollständigem Audit-Trail.

Problem

Organisationen der Gefahrenabwehr (Feuerwehr, BOS, Behörden) müssen behördenübergreifende Lagen gemeinsam bewältigen, doch jede Behörde betreibt ihr eigenes, isoliertes CX5.

Heute gibt es keinen sicheren, berechtigungsbasierten Weg, einen laufenden Einsatz mit einer anderen Organisation zu teilen und beide Seiten aktuell zu halten ohne Führungssysteme zusammenzulegen oder mehr Daten als nötig preiszugeben.

Ziele

- Explizites Ad-hoc-Teilen von Einsätzen zwischen Organisationen
- Lese-/Schreibrechte auf Modulebene, durchgesetzt im Broker *und* in der Apply-Schicht
- Bidirektionale Synchronisation nahezu in Echtzeit mit Audit
- Betrieb in eingeschränkten Netzen (RTT 600ms – 2s)

Dokumentation

Arten der Vernetzung

Definition der Vernetzungsarten UC1 – UC4

Funktionale Anforderungen

Anforderungen A-00X, B-00X

Architektur

Überblick über alle Kapitel der Architekturdokumentation

Offene Fragen

Aktuelle offene Fragen

Links

- [Youtrack HUB](#)

Arten der Vernetzung

Folgend die Definition von UC1 - UC4 aus PM Doc.

UC1 - Instanzspiegelung

gegenseitige Spiegelung des gesamten Einsatz- und Stammdatenbestands zweier Instanzen, die sich vorher kennen. Festgelegte Bereiche wie lokale Benutzer und Kommunikationseinstellungen bleiben lokal. Es handelt sich daher nicht um eine Datenbankreplikation.

Zwei Instanzen, die sich vorher kennen, spiegeln gegenseitig ihren gesamten Einsatz- und Stammdatenbestand, einschließlich Funktionen und Rechten. Auf beiden Instanzen wird gleichzeitig gearbeitet, Änderungen fließen in beide Richtungen.

Es gibt keine bevorzugte Instanz: Beide sind getrennte, gleichberechtigte Instanzen mit eigener Adresse, häufig an verschiedenen Orten. Lokal bleiben nur festgelegte Bereiche (siehe Umfang). Es handelt sich deshalb nicht um eine Datenbankreplikation, sondern um eine fachliche Spiegelung.

UC2 - Befehlsstellen-Synchronisation

Synchronisation der Daten einer oder mehrerer Befehlsstellen zwischen zwei Instanzen, die sich vorher kennen.

Zwei Instanzen, die sich vorher kennen, synchronisieren nur die Daten einer oder mehrerer bestimmter Befehlsstellen und die dafür nötigen Stammdaten. Die Daten sind auf die berechnete(n) Befehlsstelle(n) beschränkt. Beide Instanzen bearbeiten die Einsätze dieser Befehlsstelle(n).

Die Hauptinstanz führt die Befehlsstellenstruktur. Die Partnerinstanz ist die Instanz, der die Befehlsstelle(n) zugewiesen sind.

UC3 - Ad-hoc-Synchronisation

Zusammenführung zweier bisher unbekannter Instanzen für einen gemeinsamen Einsatz, wenn die Netzwerke frei sind. Die Berechtigungen werden erst im Einsatz gesetzt.

Zwei Instanzen, die sich vorher nicht kennen, werden für die Zusammenarbeit in einem gemeinsamen Einsatz zusammengeführt. Übertragen werden nur die Einsatzdaten, die Stammdaten beider Seiten bleiben lokal. Die Berechtigungen werden erst im Einsatz beim Einladen gesetzt. Die Nutzer melden sich weiterhin nur an ihrer eigenen Instanz an.

UC 3 setzt freie Netzwege voraus. Sind die Netzwege eingeschränkt, etwa durch Firewalls oder getrennte Netze, gilt UC 4.

UC4 - Vorgeplante Einsatz-Synchronisation

Ad-hoc-Teilen eines Einsatzes über eine Verbindung, die wegen Netzwerkbeschränkungen vorab eingerichtet und laufend geprüft wird. Ein Code ist dafür nicht nötig.

Zwei Instanzen arbeiten im Einsatzfall ad hoc zusammen wie in UC 3. Wegen Netzwerkbeschränkungen lässt sich die Verbindung aber nicht erst im Einsatz herstellen, etwa weil Firewall-Freigaben oder feste Gegenseiten nötig sind. Sie wird deshalb vorab von den Systemadministrationen beider Seiten eingerichtet und laufend geprüft, auch wenn gerade nichts geteilt wird.

Im Einsatzfall wird ein Einsatz ohne Code über diese Verbindung geteilt. Die Berechtigungen werden wie bei UC 3 erst beim Teilen im Einsatz gesetzt. Die Systemadministration kann je Verbindung Obergrenzen festlegen.

Out of Scope

- **UC5 - Stammdatenportal:** Austausch administrativ vorgepflegter Stammdaten zwischen Instanzen über ein zentrales Portal. UC 5 ist eine spätere Weiterentwicklung und nicht Teil der ersten Umsetzung.

Funktionale Anforderungen

Anforderungen wurden aus PM Usecases Instanzvernetzung übernommen.

Wesentliche Anforderungen

- **Organisationsübergreifende Zusammenarbeit**
Getrennte CX5 Führungs- und Stabsinstanzen können sich verbinden und laufende Einsätze miteinander teilen.
- **Selektive, durchgesetzte Berechtigungen**
Freigabe gelten auf Modulebene mit Lese- und Schreibrechten.
- **Kombination von Vernetzungsarten**
Alle Beteiligten eines Einsatzes sehen sich; ein UC1-Paar wirkt nach außen wie eine Seite; ein UC2-Einsatz kann zusätzlich ad hoc geteilt werden
- **Bidirektionale Synchronisation**
Änderungen sollen nahezu in Echtzeit übertragen und angewendet werden.
- **Vollständiger Audit-Trail**
Nachvollziehbarkeit aller Kopplungs- und Teilungsaktionen im Einsatztagebuch bzw. Systemprotokoll
- **Betrieb unter schwierigen Netzwerkbedingungen**
System soll mit einem RTT von 600 - 2s zurechtkommen.

Funktionale Anforderungen

Generelle (A-00X)

ID	Anforderung	Akteur,Modul
A-001	Namen der im Einsatz beteiligten Befehlsstellen und Partner.	System, CX5
A-002	Kombination UC und mehrere Beteiligte Alle Beteiligten sehen sich. Ein UC-1-Paar wirkt wie eine Seite. Ein UC-2-Einsatz kann zusätzlich ad hoc geteilt werden.  Offene Frage: Datenhoheit bei Mischung UC2 + UC3? Was geht dann an UC3>UC1 in Offene Fragen	System, Collaboration, Connectivity
A-003	Konfliktlösung je nach Feld mit LWW + Doku an betroffene Nutzer	System, Operations, Sync
A-004	Verbindungs Status aller Partner in der Sync-Anzeige (Kopfzeile)	Unbekannt, CX5
A-005	Synchronisation im Hintergrund Eingeschränkt je nach UC1-4 Missionsdaten, Stammdaten, Blobs, ...	System, Sync

ID	Anforderung	Akteur,Modul
A-006	Priorisierung (QoS) Bei schmaler Leitung gehen Meldungen, Status und Lage vor Anhängen und großen Dateien	System, Sync
A-007	Verschlüsselung und gegenseitige Authentisierung Verbindungen sind verschlüsselt. Kopplungen verwalten nur die Systemadministration und Funktionen mit „Ad-hoc teilen“.	System, Connectivity
A-008	Nachvollziehbarkeit; Protokollierung Einladen, Annehmen, Ablehnen, Widerrufen, Zugriffsänderungen, Weiterteilen, Beenden und Verbindungsabbrüche landen im Einsatztagebuch bzw. Systemprotokoll	System, Operations
A-009	Fehler bei einzelnen Datensätzen Ein fehlerhafter Datensatz blockiert nichts. Er wird gemeldet (Mouseover, SysAdmin) und automatisch erneut übertragen.	System,SysAdmin, Operations
A-010	Versionsprüfung Beim Koppeln oder Annehmen wird geprüft. Ein Hinweis nennt die Module, die nicht synchronisiert werden können; der Rest läuft weiter.	System, Connectivity
A-011	Bestehende Konfigurationen übernehmen Master/Client-Konfigurationen werden beim Update übernommen; der bisherige Master gilt als einrichtende Seite	System, CX5
A-012	Einladung per beidseitigem Vertrauen	System, Connectivity
A-013	Einladungen sind single-use; laufen ab; und koennen revoked werden	System, Connectivity

UC1-UC4 (B-00X)

ID	Anforderung	UC	Modul
B-001	Kopplung einrichten System > Synchronisierung / Client-Synchronisation, Art "Vollständige Synchronisation"	UC1, UC2, UC4	SysAdmin, Connectivity
B-002	Kopplung bestätigen Gegenseite bestätigt die Kopplung	UC1, UC2, UC4	SysAdmin, Connectivity
B-003	Kopplung auflösen Es wird nichts gelöscht, beide behalten ihren Stand. Nach einer rein technischen Trennung wird zusammengeführt, nach dem Auflösen wird die Spiegelung neu aufgebaut.	UC1, UC2, UC4	SysAdmin, Connectivity
B-004	Anmeldung von Funktionen übertragen Benutzer dürfen sich nur 1mal anmelden und Funktion je Befehlsstelle eventuell auch	UC1, UC2	Sync
B-005	Einschränkungen auf aktive Einsätze nur aktive Einsätze oder alle Einsätze; bei nur aktive Einsätzen werden alte gelöscht	UC2	SysAdmin, Collaboration, Sync
B-006	Einladung erzeugen (mit Recht Ad-Hoc teilen) Direkt im Einsatz: Name der Gegenseite, Zugriffsart, Gültigkeitsdauer (Standardwert vorgeschlagen), bei Teilzugriff der Abschnitt	UC3	User, Connectivity

ID	Anforderung	UC	Modul
B-007	Einladung annehmen (mit Recht Ad-Hoc annehmen) Einsatz Einladung Akzeptieren und annehmen, dabei wird die Einladung einer bestehenden Befehlsstelle zugeordnet, besteht der Einsatz bereits wird zusammengeführt	UC3	User, Connectivity
B-008	Einladung widerrufen (mit Recht Ad-Hoc teilen) Alle offenen Einladungen des Einsatzes sind sichtbar und widerrufbar, auch die anderer Funktionen. Danach ist der Code sofort ungültig.	UC3	User, Connectivity
B-009	Erweiterten Zugriff anfragen Gegenseite mit Lesezugriff soll mit einer Schaltfläche "Anfrage" Zugriffsanfragen versenden	UC3	Unkn, CX5, Connectivity
B-010	Sync nur von Missionsbezogen Stammdaten beider Instanzen bleiben Lokal	UC3	System, Collaboration, Sync
B-011	Einladungscode / Datei Einmal gültig, Zeitlich begrenzt, Widerrufbar, Schutz gegen Brutforce, Hinweis Ungültiger Code	UC3	System, Connectivity
B-012	Verknüpfen von geteiltem Einsatz zu eigenem Einsatz Führt die Gegenseite bereits einen eigenen Einsatz zur selben Lage, kann sie den geteilten Einsatz mit der Lizenz Großschadenslage bzw. Sonderlage mit ihrem eigenen verknüpfen oder zusammenführen.  Offene Frage: Was bedeutet hier das Verknüpfen von den Einsätzen?	UC3	Unkn, Sync, CX5
B-013	Teilen beenden Beide Seiten können Teilen beenden; automatisch bei Einsatz beenden und sperren. Jede Seite behält eine Kopie; Endstand wird immer nachgereicht.	UC3, UC4	System, Connectivity
B-014	Lesekopie in eigenen Einsatz umwandeln Weiterarbeit ohne Partner an einem geteilten Einsatz	UC3, UC4	Unkn, CX5
B-015	Annahmemodus von geteilten Einsätzen Einstellung ob "Annahme erforderlich" oder geteilte Einsätze direkt angenommen werden	UC4	SysAdmin, CX5, Connectivity
B-016	Einschränkung auf Befehlsstelle Auswahl der zu synchronisierenden Befehlsstelle	UC2	SysAdmin, Collaboration, Sync
B-017	Geteilten Einsatz annehmen Nur wenn so eingestellt. Die Frist setzt die teilende Seite.	UC4	Unkn, CX5, Connectivity

Überblick

CIMgate.HUB – Architekturdokumentation & Technische Spezifikation

Stand: 2026-10-13

Gliederung nach [arc42](#)

1. Einführung und Ziele

CIMgate.HUB ist ein Synchronisations-Gateway je Organisation, das getrennte CommandX-(CX5)-Instanzen Einsätze und – je nach Vernetzungsart – Stammdaten miteinander teilen lässt: selektiv, berechtigt, bidirektional, nahezu in Echtzeit und auditierbar.

Anforderungen

Die Anforderungen sind hier einzusehen

- [Funktionale Anforderungen](#)

Aufgabenstellung

Jede Organisation der Gefahrenabwehr betreibt ein eigenes CX5 mit eigener Datenbank. CIMgate.HUB verbindet diese Instanzen, ohne sie zusammenzulegen. Es unterstützt vier Vernetzungsarten.

	Name	Partner bekannt?	Umfang	Einrichtung
UC1	Instanzspiegelung	ja, vorab gekoppelt	gesamter Einsatz- und Stammdatenbestand; lokale Benutzer und Kommunikationseinstellungen bleiben lokal	SysAdmin (B-001/B-002)
UC2	Befehlsstellen-Synchronisation	ja, vorab gekoppelt	Daten einer oder mehrerer Befehlsstellen; optional nur aktive Einsätze (B-005)	SysAdmin
UC3	Ad-hoc-Synchronisation	nein	ein Einsatz, nur einsatzbezogene Daten; Stammdaten bleiben lokal (B-010)	Nutzer mit Recht „Ad-hoc teilen“ per Einladung (B-006)
UC4	Vorgeplante Einsatz-Synchronisation	ja, Verbindung vorab eingerichtet und überwacht	ein Einsatz, ad hoc geteilt, ohne Code	SysAdmin richtet ein, Nutzer teilt

UC1 ist keine Datenbankreplikation: Es werden fachliche Datensätze synchronisiert, festgelegte lokale Bereiche bleiben ausgenommen.

Qualitätsziele

Prio	Ziel	Bedeutung
1	Vertraulichkeit / Datenminimierung	Keine Daten verlassen die Organisation ohne explizite Freigabe; kein Leck über Einsatz-, Modul- oder Organisationsgrenzen

Prio	Ziel	Bedeutung
2	Zuverlässigkeit	Kein Datenverlust bei Verbindungsabbruch; Konvergenz nach Wiederverbindung; Fehler einzelner Datensätze blockieren nichts
3	Aktualität	Meldungen, Status und Lage erreichen Partner auch bei 2 s RTT innerhalb weniger Sekunden
4	Nachvollziehbarkeit	Jede Kopplungs- und Teilungsaktion ist auf beiden Seiten korreliert protokolliert
5	Betreibbarkeit	Installation je Knoten mit einem Befehl; lesbare Fehlerzustände für Administratoren

Stakeholder

Rolle	Erwartung
Einsatzbearbeiter / Funktion in einer Befehlsstelle	Einsatz mit wenigen Klicks teilen, Einladung annehmen, Partnerstatus sehen
Systemadministration der Organisation	Kopplungen einrichten, bestätigen und auflösen; Annahmemodus festlegen; Fehler einsehen
Betrieb / Support	Diagnose von Verzögerung, Konflikten, DeadLetterQueue
Datenschutz / Informationssicherheit	DSGVO-konforme Datenminimierung, Verschlüsselung, Audit

2. Randbedingungen

Technische Randbedingungen

Herkunft	Randbedingung	#
.NET 10, NATS 2.10 (JetStream, Leafnodes), Debezium 2.7 (pgoutput), PostgreSQL 18	POC	TC-01
Ein HUB-Knoten je Organisation bzw. CX5-Instanz, platziert in der DMZ	PM - Okt. 2025	TC-02
Eingeschränkte Netze: RTT 600 ms–2 s, Jitter, Abbrüche; teils nur vorab freigeschaltete Netzwerke (UC4)	UC	TC-03

Organisatorische Randbedingungen

Herkunft	Randbedingung	#
CX5-Seitige Oberfläche setzt das CX5-Team um; der HUB liefert Verträge (API)	CTO	OC-01
Umsetzung von Schema-Tooling liegt bei externen Entwicklern	CTO	OC-02

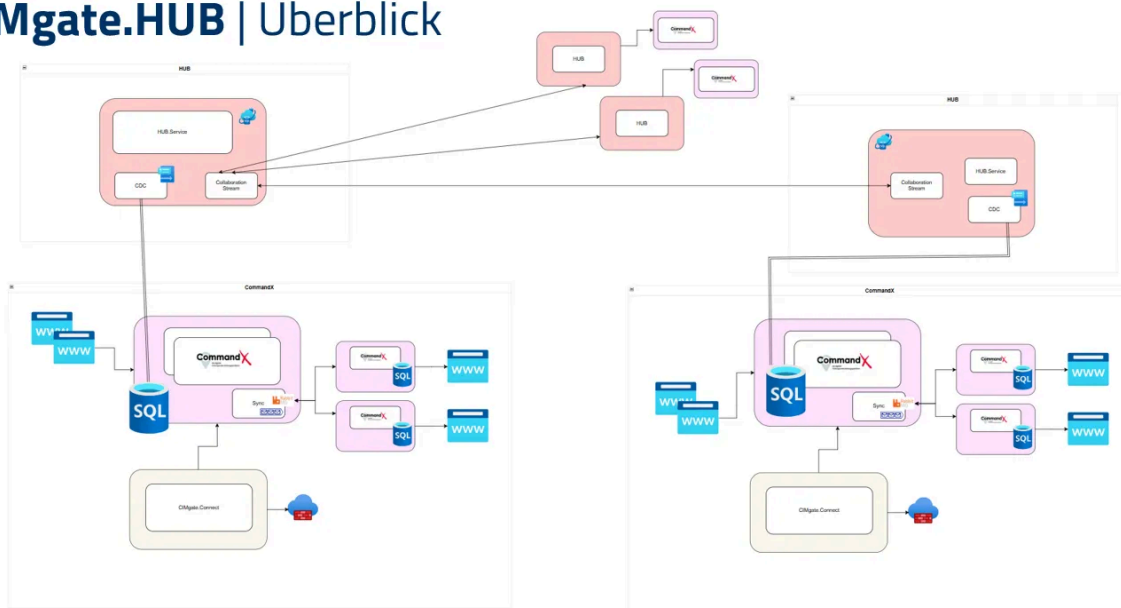
Konventionen

- Generierte Konfiguration (NATS / Debezium) wird in Produktion nie von Hand bearbeitet
- Tests müssen immer sowie auf MSSQL als auch auf PostgreSQL vorgenommen werden, dabei kann die Entwicklung auf gemischten Nodes passieren, jedoch muss auf Unterschiede zwischen Freigebenem und Annemhendem Partner geachtet werden.

3. Systemkontext

Der zentrale Architekturgegenstand ist das Produkt CIMgate.HUB nach den Vernetzungsarten UC1-UC4. CIMgate.HUB sitzt zwischen der eigenen CX5-Datenbank und der Aussenwelt. Er kommuniziert als Schnittstelle mit den Hub-Knoten der Partner. Als Verbindungsweg und für den Datenaustausch wird ausschließlich NATS genutzt. Nutzer interagieren über CX5 und Administratoren können zusätzlich über die HUB-UI Metriken einsehen. Daten verlassen die Organisation nur über den HUB, CX5 selbst spricht nie direkt mit Partnern.

CIMgate.HUB | Überblick



Fachlicher Kontext

Element	Eingang in den HUB	Ausgang aus dem HUB
CX5-Instanz (eigene Org.)	Datenänderungen (CDC), Teilen-/Einladen-/Annehmen-Aktionen, Kopplungskonfiguration	Synchronisierte Datensätze, Partner- und Verbindungsstatus, Konflikt- und Fehlerhinweise, Einsatztagebuch-Einträge
Partner-HUB (UC1, UC2, UC4: vorab gekoppelt)	Änderungen der Gegenseite, Snapshots, Statusmeldungen, Teilungsangebote	Eigene freigegebene Änderungen, Snapshots, Bestätigungen
Partner-HUB (UC3: ad hoc)	Einlösung einer Einladung, Änderungen des geteilten Einsatzes	Einladung (Code/Datei), Einsatzdaten gemäß Freigabe
Systemadministration	Kopplung einrichten/bestätigen/auflösen, Annahemodus, Freigaben	Dashboard, Systemprotokoll, Fehlerliste
Nutzer mit Recht „Ad-hoc teilen/annehmen“	Einladung erzeugen/widerrufen, annehmen, Teilen beenden, Zugriff anfragen	Einladungscode, Status, Zugriffsanfragen
Monitoring	-	Metriken, Traces, Logs (OTLP)

Technischer Kontext

Schnittstelle	Protokoll	Inhalt
CX5-DB lesen	PostgreSQL logische Replikation (pgoutput) → Debezium	Zeilenänderungen aller klassifizierten Tabellen
CX5-DB schreiben	EF Core, Transaktion je Nachricht/Batch	Anwenden eingehender Änderungen inkl. Sync-Metadaten
CX5-UI ↔ HUB	HTTPS/JSON (lokale Control-Plane-API)	Einladungen, Kopplungen, Status, Zugriffsanfragen
HUB ↔ HUB Data Plane	NATS-Leafnode über TLS 1.3, Credentials je Peer	Collab-, Snapshot-, Status- und Objekt-Subjects
HUB ↔ HUB Control Plane	HTTPS + kurzlebige JWT	Handshake, Versionsprüfung, Einlösung, Kopplungsbestätigung
LAN-Erkennung (optional)	mDNS <code>_cimgate-hub._tcp.local</code>	Ankündigung von HUB-Knoten
Monitoring	OTLP → Grafana	Telemetrie

Zeilendaten fließen ausschließlich über NATS, nie über HTTP. Bei UC4 ist der Netzweg vorab freigeschaltet und wird laufend geprüft; bei UC3 muss der Teilende-Partner eine Seite für die Gegenseite direkt erreichen können.

4. Lösungsstrategie

Alle vier Vernetzungsarten laufen über eine Pipeline und unterscheiden sich nur im *Sync-Profil*, das festlegt, welche Daten welchen Partner mit welchen Rechten erreichen.

Kernideen

Hinweis

aus den [Funktionale Anforderungen](#) und Querschnittsanforderungen von PM

Quality Ziel	Ansatz	ADR
Vertraulichkeit	Deny-by-default; Getrennter Partner Account im Broker mit Subject-Berechtigungen; Creds mit Expiry	
Zuverlässigkeit	JetStream-Persistenz, explizites Ack, Snapshot + LSN-Catch-up, Tombstones, Retry + Backoff	
Throughput und Priorisierung (QoS)	Leaf-Verbindungen mit Source Streams; Prioritätsklassen mit getrennten Streams; Blobs on demand	 Prioklassen
Konsistenz	Feldweises LWW auf Basis von Änderungszeitstempeln je Feld; Konfliktprotokoll an Betroffene	 Feldweise LWW
Nachvollziehbarkeit	Append-only-Audit mit knotenübergreifender Korrelations-ID; Spiegelung relevanter Ereignisse ins Einsatztagebuch	
Betreibbarkeit	HUB-DB als Source of Truth, Reconcile-Loop rendert NATS-Konfiguration; Hub per docker compose	

Sync-Profile je Vernetzungsart

Ein Sync-Profil ist die einheitliche Abstraktion über UC1-UC4. Eine *Kopplung* (Partnerbeziehung) trägt ein Profil, ein *Share* ist die konkrete Freigabe eines Datenumfangs.

Sync-Profil	Kopplung	Share-Umfang (Scope)	Stammdaten	Lebensdauer	Share Annahme
INSTANCE (UC1)	dauerhaft, SysAdmin beidseitig bestätigt	Instanz (alle Einsätze)	ja, außer lokale Bereiche	bis zur Auflösung	implizit durch Kopplung
DEPARTMENT (UC2)	dauerhaft, SysAdmin	Befehlsstelle(n); optional nur aktive Einsätze	nur zugehörige	bis zur Auflösung	implizit
ADHOC (UC3)	temporär, durch Einladung	ein Einsatz (optional Abschnitt)	nein	bis Teilen beendet / Einsatz beendet	immer explizit

Sync-Profil	Kopplung	Share-Umfang (Scope)	Stammdaten	Lebensdauer	Share Annahme
FEDERATED (UC4)	dauerhaft, SysAdmin; Verbindung überwacht	ein Einsatz je Share	nein	bis Teilen beendet	konfigurierbar (B-015)

Anmerkungen

Um die Vernetzungsarten UC3 und UC4 umzusetzen, müssen in CX5 Einsätze, sowie deren Missions- Ressourcen und andere missionsbezogene Daten unabhängig voneinander arbeiten. Zum Beispiel erwartet CX5 aktuell für jede Meldung die in einer Mission geschrieben wird, das die Meldungsart mit dieser bestimmten ID in den Stammdaten angelegt und konfiguriert ist. Diese Meldungsarten werden jedoch je Instanz gepflegt und sollen nicht übertragen und in der anderen Instanz erscheinen. Ein solche Übertragung der Stammdaten würde ansonsten zu doppelt benannte Meldungsarten mit unterschiedlichen Konfigurationen führen können, oder bei Verknüpfung der gleichbenannten Meldungsarten, zu einer Unterschiedlichen und somit inkonsistenten Verarbeitung der Meldungen.

UC1: Instance Sync

Für UC1 bestehen keine Einschränkungen und alle Daten können Synchronisiert werden. Hier müsste nur auf ein Snapshotting für neue Nodes im Sync, sowie Catchup für lange inaktive Nodes geachtet werden.

UC2: Department Sync

Für UC2 besteht zu Punkten aus UC1 zusätzlich die Einschränkung auf Departments. Missions- sowie Stammdaten müssen hier bei der Synchronisierung zu Befehlsstellen zugeordnet, gefiltert und anschließend übertragen werden.

UC3: Adhoc Sync

Für UC3 muss CX5 mit asynchronem Stammdatensatz und Unterschieden in Missionen zurechtkommen, dort besteht die Einschränkung auf keine Stammdaten und nur bestimmte Missionsdaten. (Eine vorerst vereinfachte Variante wäre die Möglichkeit Missions-Ids zu verbinden, feste Empfängergruppen für externe Systeme zu erstellen und diese wie Email-Eingang zwischen den Systemen zu behandeln. Jedoch sind die Anforderungen weit mehr als eine einfache Meldungskommunikation)

UC4: Federated Sync

Für UC4 besteht gegenüber UC3 keine weiteren Einschränkungen, der Unterschied liegt in der Verwaltung der Verbindung von Adhoc zu Dauerverbindung.

Kombination von Vernetzungsarten

Datenhoheit bei Mischform ist fachlich offen (A-002),

Bis dahin: Nimmt ein NNode im UC1/2 Sync eine Mission per Adhoc-Share an bleibt diese auf dem Node.

Versionshandling

Gekoppelte Instanzen laufen nicht zwingend auf derselben CX5-Version. Damit trotzdem synchronisiert werden kann, wird die Kompatibilität **je Modul** und nicht je Gesamtversion bewertet (A-010).

Schema als Code: Analog zu EF-Migrationen werden die Avro-Schemata der synchronisierten Tabellen im CX5-Source gepflegt, zusammen mit der Tabellenklassifizierung (Tabelle → Modul). Jede Schemaänderung ist damit Teil des Releases, versioniert und reviewbar.

Ablauf:

1. `Hub.Schema` liest beim Start die Schemata und Klassifizierung der lokalen Instanz und bildet je Modul einen Fingerprint.
2. Beim Koppeln bzw. Annehmen eines Shares tauschen die HUBs ihre Modul-Schemata aus.
3. `Hub.Schema` prüft je Modul die Avro-Kompatibilität in **beide Richtungen** (eigenes Schema liest Partner-Schema und umgekehrt).
4. Kompatible Module werden synchronisiert; Unterschiede gleicht die Avro Schema Resolution beim Lesen aus (Up-/Down-Kompatibilität).
5. Inkompatible Module werden für diese Kopplung nicht synchronisiert. Beide Partner erhalten einen Hinweis mit den betroffenen Modulen, alle anderen Module laufen weiter.
6. Nach einem Update eines Partners wird die Prüfung erneut ausgeführt; der Modulstatus kann sich dadurch in beide Richtungen ändern.

Beispiel: Instanz A hat in der Checkliste eine Spalte im Typ geändert, Instanz B noch nicht. Das Modul Checklisten wird zwischen A und B ausgesetzt und bei beiden als inkompatibel angezeigt, Meldungen, Kräfte usw. werden weiter synchronisiert.

Änderung am Schema	Bewertung	Behandlung
Neues Feld mit Default	kompatibel	ältere Partner ignorieren das Feld, neuere setzen den Default
Feld entfernt, das einen Default hatte	kompatibel	ältere Partner setzen den Default
Typenerweiterung (z. B. <code>int</code> → <code>long</code>)	nur aufwärts kompatibel	Modul nur in eine Richtung lesbar, gilt für Sync als inkompatibel
Umbenennung mit Avro- <code>aliases</code>	kompatibel	Zuordnung über Alias
Umbenennung ohne Alias, Typänderung, neues Pflichtfeld ohne Default, geänderter Schlüssel	inkompatibel	Modul wird ausgesetzt, Hinweis an beide Partner

Anmerkungen:

- Für Sync-relevante Tabellen gilt Avro-Kompatibilität `FULL` (vor- und rückwärts). Ein Build-Check prüft neue Schemata gegen die der bisher ausgelieferten Versionen, sodass Breaking Changes bewusst entschieden werden statt unbemerkt ins Release zu gelangen.
- Jede Nachricht trägt die Schema-ID bzw. den Fingerprint des Senders im Header, damit der Empfänger das passende Writer-Schema zur Resolution nutzt.
- Avro-Defaults dürfen beim Anwenden keine echten Werte überschreiben. Ein älterer Partner kennt ein neues Feld nicht, sein Update darf das Feld beim Empfänger daher nicht auf den Default zurücksetzen. Angewendet werden nur Felder, die im Writer-Schema enthalten sind (passt zum feldweisen LWW, A-003).
- Avro prüft nur die Struktur. Semantische Änderungen (z. B. neue Bedeutung eines Enum-Werts oder Status) erkennt die Prüfung nicht und müssen als Breaking Change behandelt werden.

Auslieferungsstufen

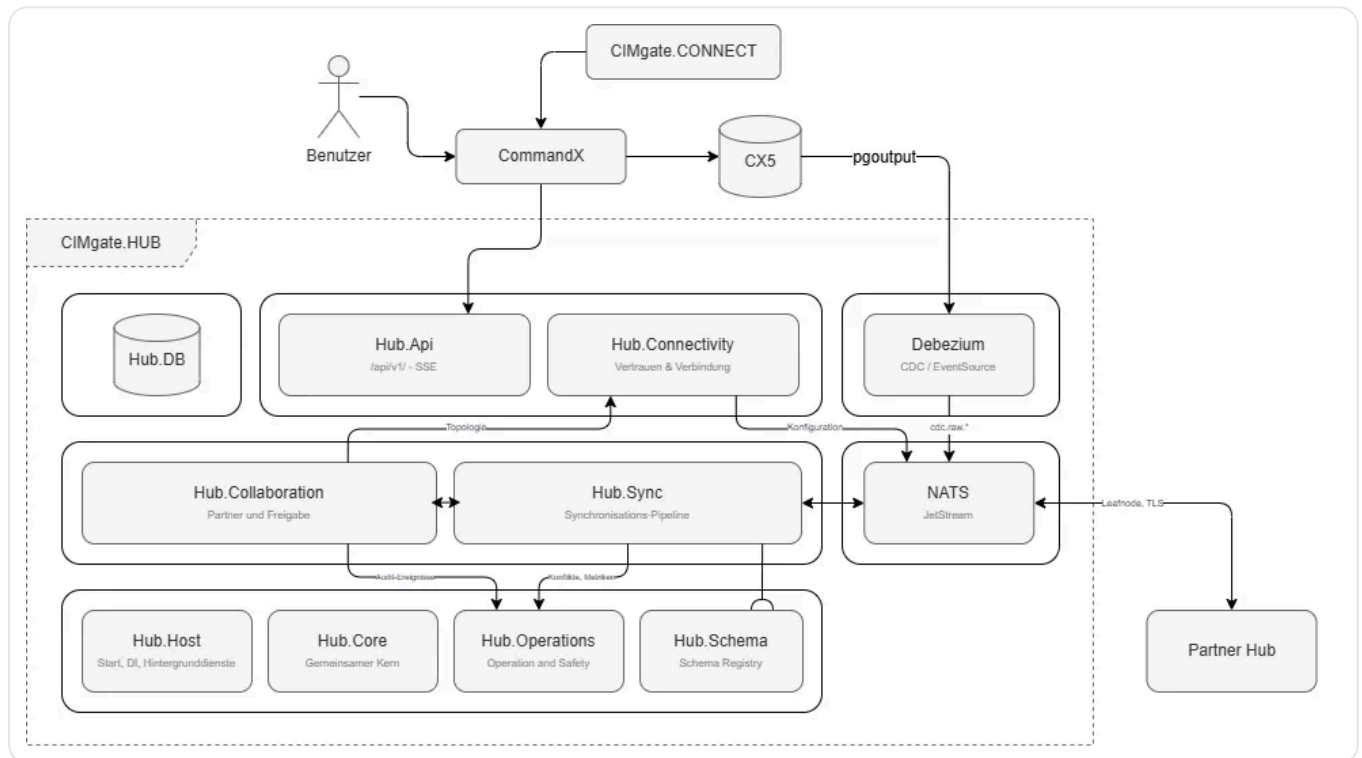
- v0.1 - Kernfunktionen
 - gehärtete Pipeline (DB → CDC → NATS → HUB → NATS → Partner)
 - Fest konfigurierbare mit Sync Profil `INSTANCE`
 - HUB Protokoll v1 (Partner ↔ Partner)
 - Validierung Avro/Schema-Registry
 - Sync von Daten ohne Blobs
 - Dashboard und einfache Metriken
- v0.1+1
 - A-003: Feldweise LWW
 - Versionsaushandlung auf Modul-Ebene per Schema Abgleich
- v0.2 - Kopplung
 - Trust / Enrollment
 - Profile `DEPARTMENT`, `FEDERATED`, `ADHOC`
Aktuell bereits bekannter Blocker für weitere Profile in v0.2 ist das Verknüpfte Stammdaten-Handling in CX5 Missionen.
 - Erweiterung HUB Protokoll (Partner ↔ Partner)
 - (optional) mDNS Discovery
- v0.3 - Erweiterung
 - Blobs/Anhänge
 - A-011: Übernahme alter Master/Client Konfiguration

Annahme: Die Reihenfolge ist ein Vorschlag und muss noch abgestimmt werden

5. Bausteinsicht

Die Bausteinsicht beschreibt die statische Zerlegung des Systems in Bausteine (Module, Komponenten, Subsysteme, Pakete ...) und deren Beziehungen. Sie zeigt die Gesamtstruktur des Quellcodes. Diese Sicht ist in einer Top-down-Hierarchie organisiert.

5.1 CIMgate.HUB Knoten (Whitebox)



In dem Diagramm sind einige Abhängigkeiten zur Übersicht nicht eingezeichnet. Dies sind unter anderem Verbindungen zu **Hub.Core**, das von allen Projekten referenziert wird, sowie der Schreibzugriff von **Hub.Sync** auf die CX5-Datenbank.

Motivation der Zerlegung

Durch die bereits bestehende Arbeit vor der Interschutz, sowie dem Einsatz der PowerSync Komponente in Mobile, hat sich das Konzept den WAL der Datenbank auszulesen durchgesetzt. Dieses Konzept hat an CommandX einen automatischen Event-Source-Stream angehängt, ohne den Programmcode von CommandX an vielen Stellen kontinuierlich erweitert werden musste.

Somit bleiben für den CIMgate.HUB die folgenden Aufgaben für sein Betrieb zu erfüllen:

- Aufbereiten und Verstehen der anfallenden Daten und Events (**Sync**)
- Entscheiden, welche Daten welchen Partner erreichen (**Collaboration**)
- Verbinden und Authentifizieren von Partner auf technischer Ebene (**Connectivity**)

- Protokollieren und Schaffen von Transparenz für Administratoren ([Operations](#))
- Steuerung durch Benutzer in CX5 ([Api](#))

Ergänzend gibt es drei Bausteine ohne eigene Fachaufgabe:

- [Hub.Schema](#) ist eigenständig, da die Tabellenklassifizierung als Teil von CommandX.WEB gepflegt wird und das Schema-Tooling gesondert umgesetzt wird (OC-02). Der Baustein kapselt diese Abhängigkeit.
- [Hub.Core](#) hält die gemeinsamen Verträge, damit die Fachbausteine sich nicht gegenseitig referenzieren müssen.
- [Hub.Host](#) bündelt Start, Konfiguration und Lebenszyklus, damit die Fachbausteine unabhängig vom Hosting bleiben (Betreibbarkeit, docker-compose je Knoten).

Enthaltene Bausteine

Diese Aufgaben lassen sich wie folgt auf die Bausteine aufteilen:

Baustein	Aufgabe
Hub.Sync	Entgegennahme aller Events, Klassifizierung nach Schema, Übernehmen von eingehenden Änderung, Konfliktauflösung, Initialabgleich, zyklenfreie Weiterleitung von Änderungen
Hub.Collaboration	Verwaltung über Partner, Sync-Profile sowie deren Kopplung, Shares und Einladungen. Liefert Regeln für Hub.Sync , wird von Hub.Connectivity genutzt um die Topologie aufzubauen
Hub.Connectivity	Verbindet und authentisiert Partner, ordnet Knotenidentität Partnern zu, verwaltet Credentials, führt das Hub-Protocol, erzeugt NATS-Konfiguration, mDNS Discovery
Hub.Operations	Audit, Fehlerliste, Metriken und Admin Dashboards, verwaltet Debezium und die Anbindung an die CX5 Datenbank
Hub.Api	Schnittstellen zur Steuerung der Bausteine aus CX5
Hub.Schema	Schema Registry: Tabellenklassifizierung, Avro-Schemata, Validierung und Versionsabgleich auf Modulebene
Hub.Core	Hält gemeinsame Verträge die über alle Bausteine genutzt werden und enthält selbst keine Fachlogik
Hub.Host	AppHost, Konfiguration, Lebenszyklus, Dependency Injection, Hintergrunddienste

Fremd- und Infrastrukturbauwerke (nicht selbst entwickelt, Teil des Knotens):

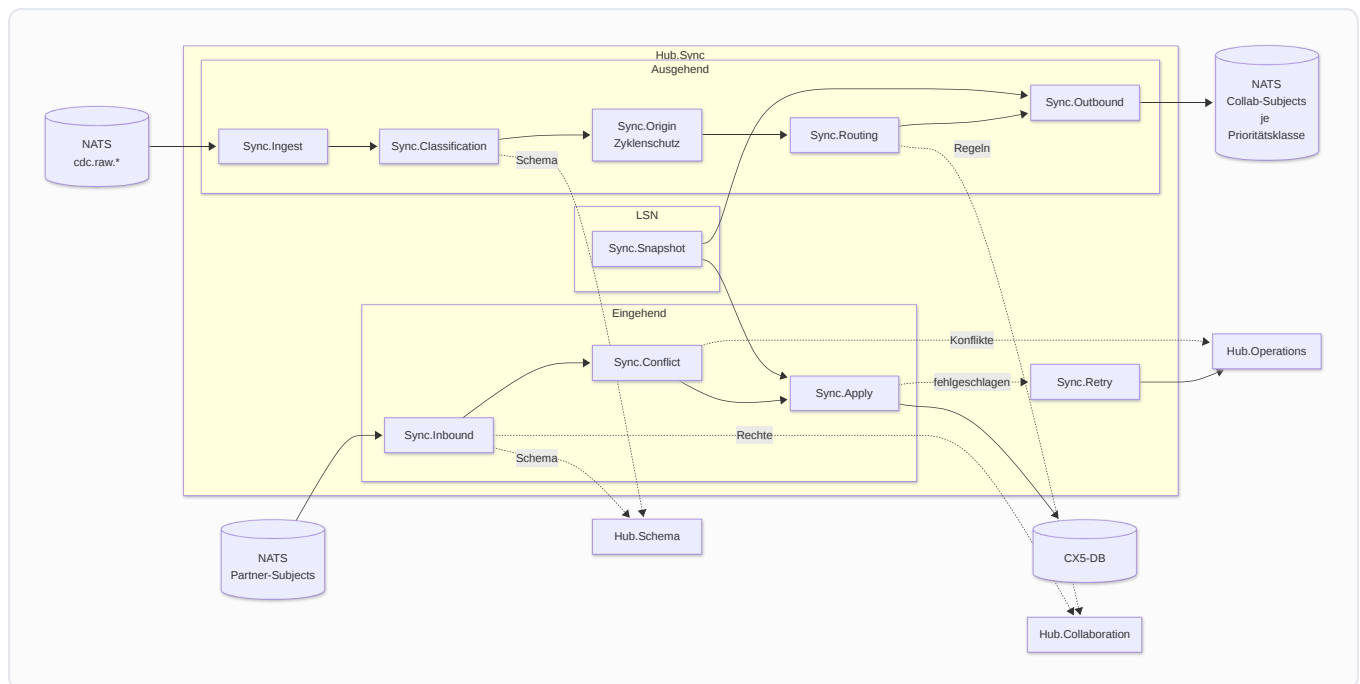
Baustein	Technologie (TC-01)	Aufgabe
Hub.DB	PostgreSQL 18 (Annahme)	Source of Truth des HUB: Partner, Kopplungen, Shares, Credentials, Audit, Sync-Zustand
Debezium	Debezium 2.7 (pgoutput)	Liest die logische Replikation der CX5-DB und publiziert Zeilenänderungen auf cdc.raw.*
NATS JetStream	NATS 2.10	Persistente Streams, Leafnode-Verbindungen zu Partnern, Accounts, Subject-Berechtigungen

Wichtige Schnittstellen

Von → Nach	Bezeichnung	Protokoll	Inhalt
CommandX → Hub.Api	Control-Plane-API	HTTPS/JSON/api/v1, SSE	Einladungen, Kopplungen, Shares, Zugriffsanfragen; Status als Push
CX5-DB → Debezium	CDC	PostgreSQL pgoutput MSSQL CDC	Zeilenänderungen aller klassifizierten Tabellen
Debezium → NATS	cdc.raw.*	NATS JetStream	Rohereignisse je Tabelle
NATS ↔ Hub.Sync	verarbeitung cdc, Sync-Streams	NATS JetStream, explizites Ack	Rohereignisse ein; Collab-, Snapshot- und Status-Subjects ein/aus
Hub.Sync → CX5-DB	Apply	Transaktion je Nachricht/Batch	Eingehende Änderungen inkl. Sync-Metadaten
Hub.Sync → Hub.Schema	Klassifizierung	In-Process	Modulzuordnung, Schema-Validierung, Schema-Version
Hub.Collaboration → Hub.Connectivity	Topologie	In-Process	Welche Partner mit welchen Profilen verbunden sein müssen
Hub.Connectivity → NATS	Konfiguration	Generierte Konfigurationsdateien, Reload, Hub-Protokoll	Accounts, Subject-Berechtigungen, Leafnodes, Streams, Handshakes, Einladungen, Partner-Verbindungen
NATS ↔ Partner-HUB	Data Plane	NATS-Leafnode, TLS 1.3	Collab-, Snapshot-, Status- und Objekt-Subjects, Handshake, Versionsprüfung, Einlösung, Kopplungsbestätigung
Hub.Collaboration → Hub.Operations	Audit-Ereignisse	In-Process	Kopplungs- und Teilungsaktionen mit Korrelations-ID
Hub.Sync → Hub.Operations	Konflikte, Metriken	In-Process	Konfliktprotokoll, fehlerhafte Datensätze, Durchsatz und Latenz
Hub.Operations → Monitoring	Telemetrie	OTLP → Grafana	Metriken, Traces, Logs

5.2 Ebene 2

5.2.1 Hub.Sync (Whitebox)



Motivation

- Ausgehender und eingehender Pfad sind getrennt, da sie unterschiedlich skalieren und unterschiedliche Fehlerfälle haben.
- Rechte werden im Broker (Subject-Berechtigungen) und in `Sync.Inbound` geprüft. Eine fehlerhafte Broker-Konfiguration führt so nicht zu unberechtigten Schreibzugriffen.
- Der Zyklenschutz ist ein eigener Baustein, da jede angewendete Änderung über den WAL erneut als CDC-Ereignis erscheint. Ohne ihn würden Änderungen zwischen gekoppelten Knoten endlos gespiegelt.
- Snapshot und Initialabgleich haben einen anderen Lebenszyklus als der laufende Strom und sind deshalb getrennt.

Enthaltene Bausteine

Baustein	Aufgabe	Anforderung
<code>Sync.Ingest</code>	Konsumiert <code>cdc.raw.*</code> aus JetStream wartet auf die Verarbeitung durch die anderen Komponenten und gibt explizites Ack	A-005
<code>Sync.Classification</code>	Ordnet jede Zeilenänderung über <code>Hub.Schema</code> einem Modul und ggf. Einsatz zu; verwirft lokale Bereiche und nicht klassifizierte Tabellen	B-010, B-016
<code>Sync.Origin</code>	Erkennt anhand der Sync-Metadaten Änderungen, die aus einem eigenen Apply stammen, und leitet sie nicht erneut weiter (PropagationKey)	–

Baustein	Aufgabe	Anforderung
Sync.Routing	Ermittelt anhand der Regeln aus Hub.Collaboration die Ziel-Partner und die Prioritätsklasse, überprüft das Event auf Konfiguration, Regeln und fehlerhafte Payloads	A-006
Sync.Outbound	Publiziert auf das Collab-Subject	A-006
Sync.Inbound	Konsumiert Änderungen der Partner, validiert Schema-Version und prüft Schreibrechte je Modul	A-010
Sync.Conflict	Feldweises LWW auf Basis von Änderungszeitstempeln je Feld; protokolliert Konflikte an Betroffene	A-003
Sync.Apply	Schreibt per EF Core in einer Transaktion je Nachricht/Batch inkl. Sync-Metadaten in die CX5-DB	–
Sync.Snapshot	Initialabgleich per Snapshot + LSN-Catch-up, Tombstones, Zusammenführen nach Trennung, Nachreichen des Endstands, Bereinigung inaktiver Einsätze	B-003, B-005, B-013
Sync.Retry	Retry mit Backoff, DeadLetterQueue, Meldung fehlerhafter Datensätze an Hub.Operations	A-009

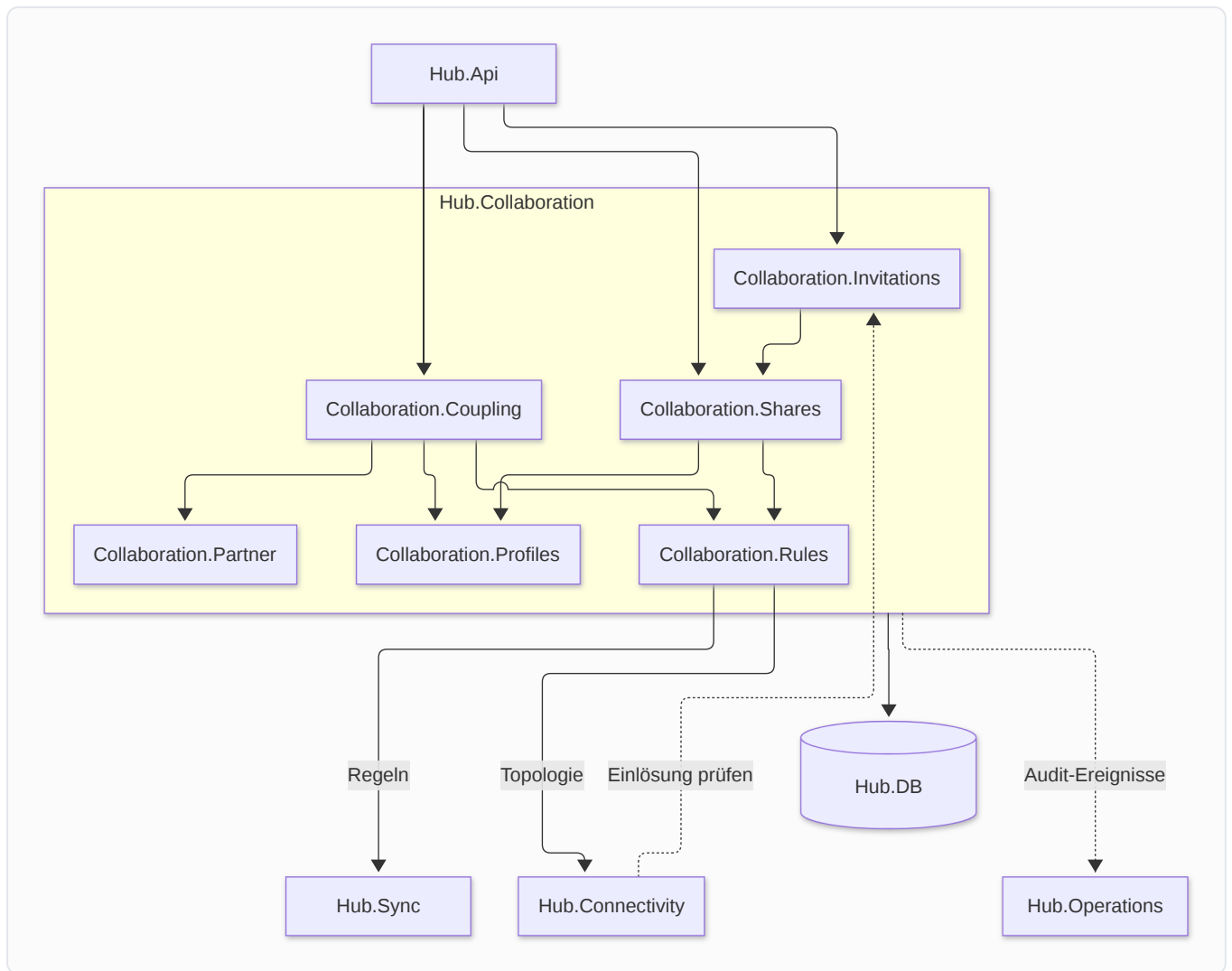
Wichtige Schnittstellen

Schnittstelle	Beschreibung
Sync.Routing → Hub.Collaboration	Abfrage der effektiven Regeln: für eine Änderung (Modul, Einsatz, Befehlsstelle) die Partner und Rechte
Sync.Inbound → Hub.Collaboration	Prüfung, ob der sendende Partner für das Modul schreibberechtigt ist
Sync.Apply → CX5-DB	Transaktionaler Schreibzugriff; die Sync-Metadaten sind Grundlage für Sync.Origin und Sync.Conflict

5.2.2 Hub.Collaboration (Whitebox)

Achtung

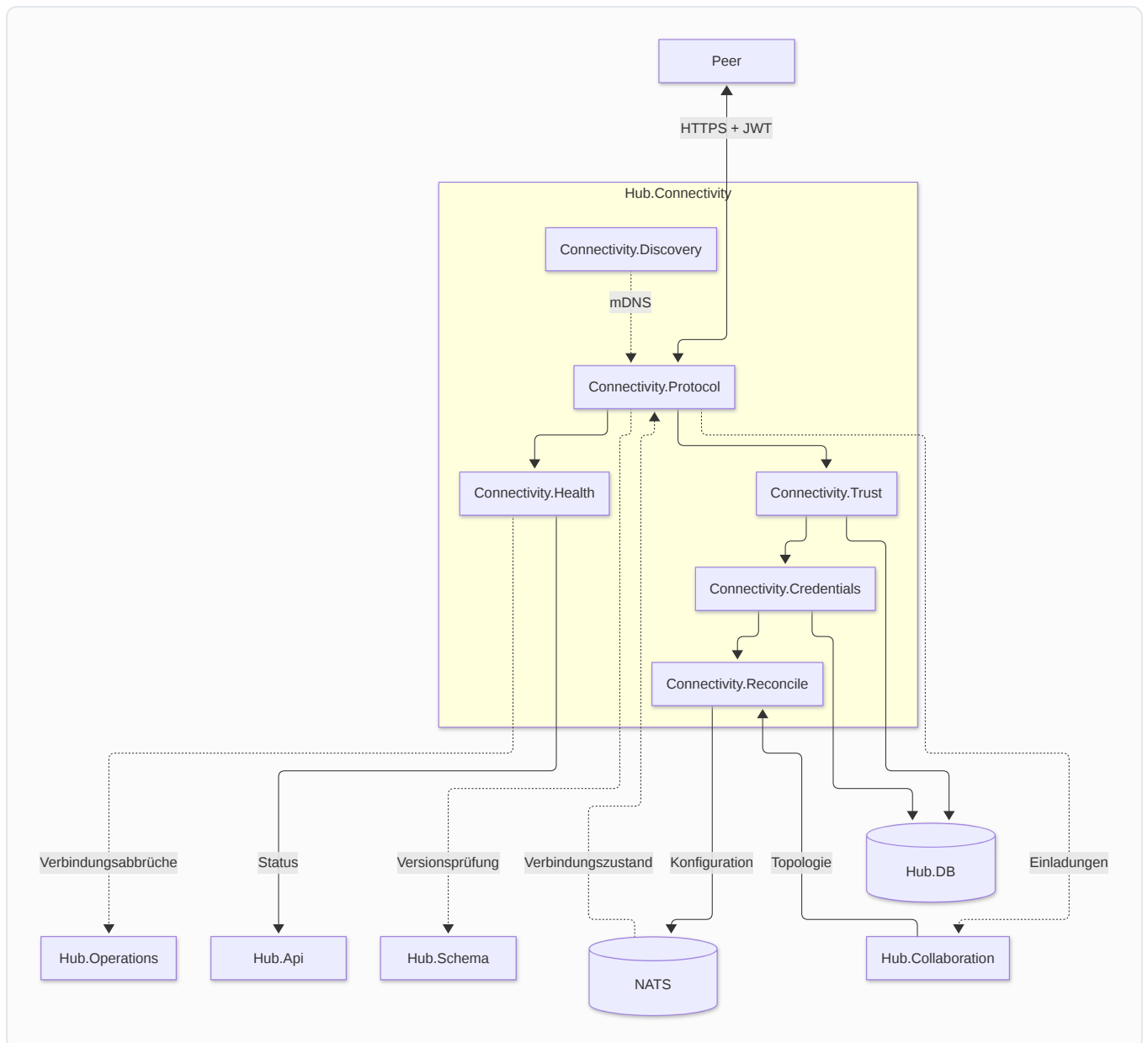
Entwurf ist nicht ausgearbeitet. Folgende Zeichnung nicht als Korrekt anzusehen!



5.2.3 Hub.Connectivity (Whitebox)

Achtung

Entwurf ist nicht ausgearbeitet. Folgende Zeichnung nicht als Korrekt anzusehen!



5.3 Ebene 3

⚠ Achtung

Entwurf ist noch nicht ausgearbeitet!

Sync.Conflict (feldweises LWW, A-003)

6. Laufzeitsicht

Beschreibung von Szenarien, welche die kritischen Abläufe decken. Jeder Ablauf schreibt Audit-Einträge mit derselben Korrelations-ID auf beiden Seiten.

Kritische Abläufe sind wie folgt definiert (6.1 - 6.5):

- Kopplung Verbindungsaufbau (Einsatz-Teilen)
- Snapshot und Initialabgleich
- Laufende Verbindung (Delta-Sync)
- Verbindungsabbruch und Wiederanlauf
- Teilen beenden

6.1 Kopplung (UC1, UC2, UC4)

SysAdmin A richtet eine Kopplung zu einem neuen Partner ein. Die Kopplung trägt ein Sync-Profil (`INSTANCE`, `DEPARTMENT` oder `FEDERATED`)

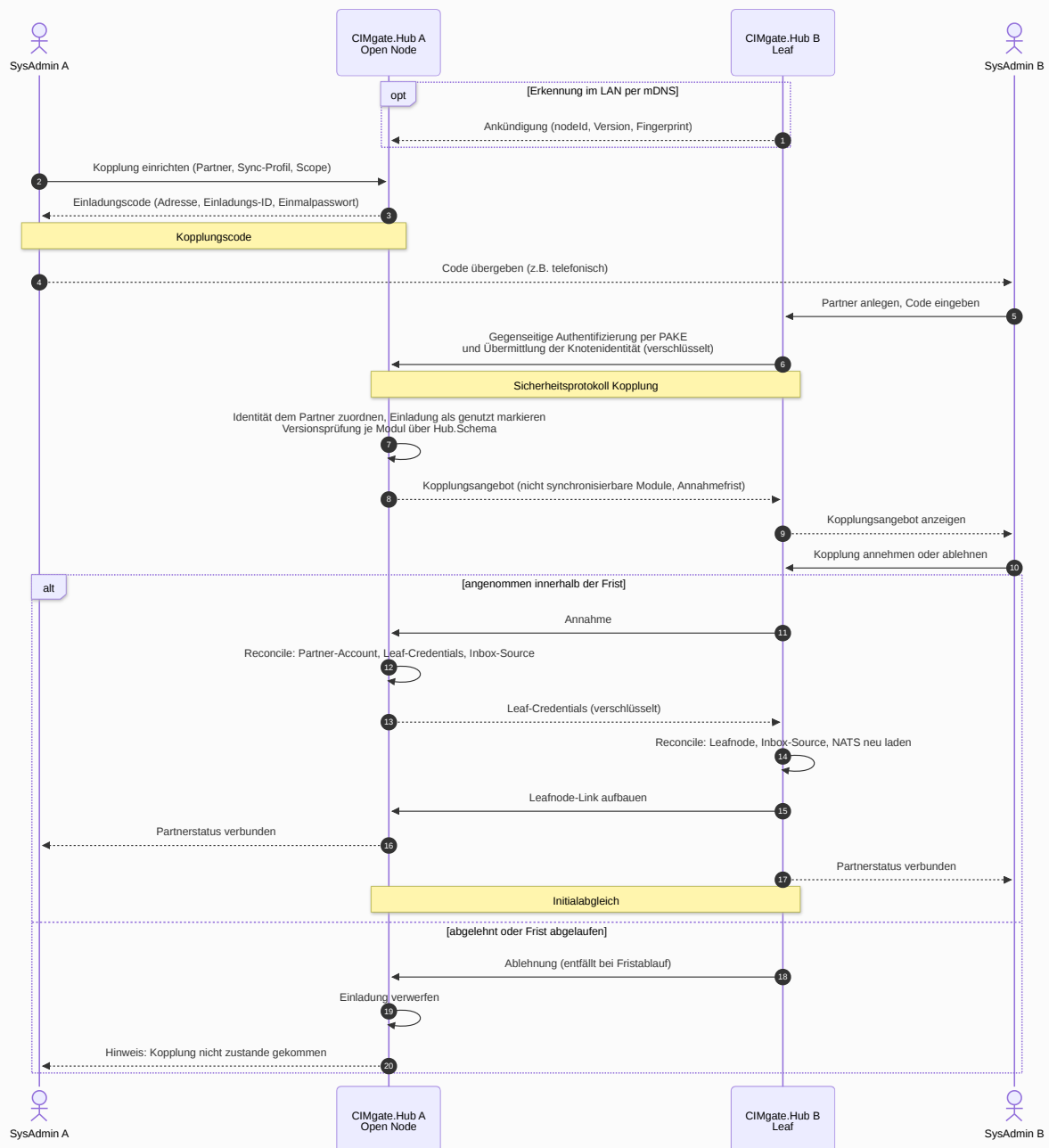
Vorbedingungen

- Beide Hubs laufen und haben eine Node-Identität
- Hub B erreicht NATS von Hub A über das Netz (Adresse und Port aus dem Einladungscode).
- Beide SysAdmins können sich direkt austauschen, z.B. per Telefon oder Email

Ergebnis

- Der Leafnode-Link von B zu A ist aktiv, beide Seiten zeigen den Partnerstatus „verbunden“.
- B sieht ausschließlich die Subjects, die A laut Scope exportiert.
- Die Einladung ist verbraucht, die Bootstrap-Verbindung ist getrennt.
- Anschließend startet der Initialabgleich (6.3).

Einsatz teilen (UC3): Das Konzept der Kopplung kann für das Einsatz teilen wiederverwendet werden



6.2 Snapshot / Initialabgleich

TODO

Abschnitt noch nicht ausgearbeitet.

Aus dem POC

- Über eine registrierte Serviceanfrage im Nats kann ein Snapshot ausgelöst werden
- Snapshot anfrage wird an Debezium weitergegeben und alle Tabellen werden neu eingelesen
-

6.3 Delta-Sync

TODO

Abschnitt noch nicht ausgearbeitet.

- Benutzer ändert Mission in CX5 A; Datenbank schreibt WAL, Debezium published nach `cdc.raw.*`
- `Sync.Ingest` liest das Event, es werden Einsatz, Modul, Befehlsstelle und betroffene Shares ermittelt.
- `Sync.Routing` identifiziert den betroffenen Stream und das Event wird in die Outbox gelegt
- JetStream-Source auf Hub B zieht das Event über die Leafnode Verbindung in die Inbox
- Dort wird es eingelesen, auf Berechtigung geprüft, LWW je Feld angewendet und das Event in CX5 B eingepflegt

6.4 Verbindungsabbruch und Wideranlauf

TODO

Abschnitt noch nicht ausgearbeitet.

- Leafnode bricht ab; Shares wechseln nach Ablauf des Heartbeat auf `Timeout`; Kopfzeile zeigt Partner als getrennt
- Beide Seiten arbeiten lokal weiter; Outbox-Streams sammeln Events
- Nach Wiederverbindung sammeln sich die Events durch die JetStream-Source beim jeweils anderem Partner
- Liegt die Lücke jenseits der Stream Retention, wechselt der Share auf `OutOfSync`; der Initiator löst einen Snapshot aus; Daten werden zusammengeführt

6.5 Teilen beenden

TODO

Abschnitt noch nicht ausgearbeitet.

- Eine Seite beendet den Share, oder der Einsatz wird beendet bzw. gesperrt (B-013)
- Der teilende Hub published einen Ende-Marker auf den Stream; ausstehende Änderungen werden eingespielt, alle neuen verworfen, Share wechselt in `Suspended`
- Credentials und Subject Berechtigungen werden entzogen; jede Seite behält ihr Kopie

6.6 Einladungscode

Für das Einsatzteilen in UC3 benötigt der CIMgate.HUB die Funktion einen Code zum Verbindungsaufbau und teilen eines Einsatzes zu erstellen. Zum sicheren Datenaustausch über NATS werden als Credentials JWT+NKey benötigt. Im Initialen Konzept wurde hierfür nur eine Datei angedacht, welche per Email oder andere wege zum Partner übermittelt werden kann. Dieses hat den Vorteil das der Inhalt von der Größe unbeschränkt wäre und so direkt Credentials, Zertifikate und weitere Verbindungsdetails in einem Packet stecken .

In Gesprächen mit PM ist die undokumentierte Anforderung aufgekommen diesen Code per Funk oder Telefon an den Partner zu übergeben. Darauß entsteht das Problem den Verbindungsaufbau, sowie das Vertrauen in ein Format zu bringen, welches schlank und für die mündliche Übermittlung dennoch alle Daten auf einen sicheren Weg übergibt.

Durch diese Anforderung ist das folgende Konzept des EinladungsCodes entstanden, der alle Informationen für den ersten Kontakt von Hub B mit Hub A trägt. Dieser Code kann für das Einsatzteilen UC3, sowie die Kopplung in UC1/2/4 genutzt werden.

- IP und Port
- Inviteld und Password

Derselbe Code in beiden Darstellungen, die verlustfrei ineinander umgewandelt werden:

Aufbau des Payloads und Regeln der Wortliste: siehe ADR Einladungscode

Darstellung	Beispiel	
Payload	IP: 203.0.113.42:4222 Inviteld: 7B8C PAKE: 8155CC	Verbindungsdaten
Einladungs-Code	SC07-2AGG-FT3K-Q5SN-HC0G	NATO-Alphabet in Viererblöcken
Telefon-Code	 Dampfer ·  Elefant ·  Becher ·  Igel ·  Ente ·  Taschenlampe ·  Laptop ·  Banane ·  Kegel ·  Ziege ·  Teleskop ·  Fuchs	Autovervollständigung bei der Eingabe als Hilfe über feste Wörterliste

6.7 Schema Abgleich

TODO

Abschnitt noch nicht ausgearbeitet.

6.X ...

TODO

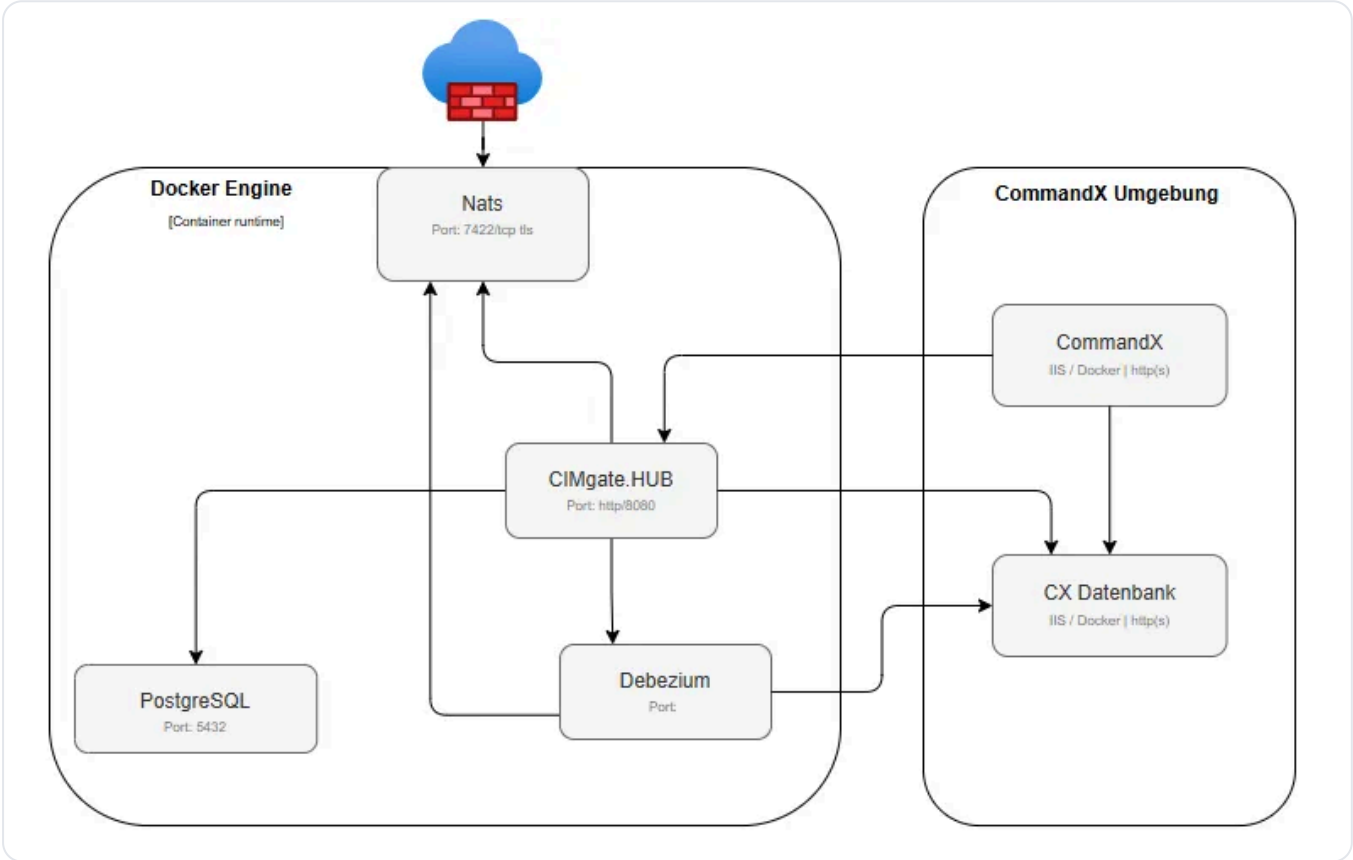
Weitere

7. Verteilungssicht

Der CIMgate.HUB unterstützt vorerst nur die Einrichtung in einer Docker-Umgebung.

Gegenüber einem Installer bringt es den Vorteil das die Externen Komponenten (Debezium, Nats, Datenbanken) immer in der gleichen Konfiguration und Version bei allen Installationen ohne viel Aufwand zu starten, neuaufzusetzen und zu bereinigen sind.

Infrastruktur Ebene 1



Container	Image / Version	Aufgabe	Netz	Volumes
hub	CIMgate.HUB (.NET 9)	Control Plane, Sync-Engine, Admin-UI	intern + Reverse Proxy, Zugriff auf CX5-DB	–
hub-db	PostgreSQL 17	HUB-Datenbank (Topologie, Shares, Audit)	intern	Volume hubdb
nats	NATS 2.10 mit JetStream	Lokaler Broker + Leafnode-Endpunkt	intern; Port 7422/TLS extern	Volume jetstream
debezium	Debezium Server 2.7	CDC aus CX5-DB nach NATS	intern, Zugriff auf CX5-DB	Offset-Store in NATS-KV
proxy	Reverse Proxy (z. B. Caddy/Traefik)	TLS-Terminierung Control Plane	Port 443 extern	Zertifikate

Container	Image / Version	Aufgabe	Netz	Volumes
otel-collector (optional)	OpenTelemetry Collector	Telemetrie-Export	intern	–

Verbindungswerge von Vernetzungsarten

Der Teilende CIMgate.HUB (Master / Remote) muss immer für die Klienten per IP und Port erreichbar sein und die Leafnode-Verbindung zulassen.

Klient bauen die Leafnode-Verbindung zum Remote auf und benötigen dafür die Adresse des Remotes, sowie die Credentials des Accounts. Der Account bestimmt die Subject-Berechtigungen

Annahme: Ein Knoten pro CX5-Instanz, Hochverfügbarkeit des HUB ist nicht gefordert.

9. Architekturentscheidungen

[examples](#)

ADR-1: Strategie zum Neuladen der NATS-Konfiguration

Kontext: Der NATS Server wird als Docker-Container betrieben. Änderungen an der Konfiguration müssen übernommen werden können, ohne den laufenden Betrieb unnötig zu unterbrechen. Ein Neustart des Containers kann bestehende Client-Verbindungen unterbrechen und die Nachrichtenverarbeitung beeinträchtigen. Daher wird ein definiertes Verfahren benötigt, um Konfigurationsänderungen zu validieren, anzuwenden und Fehler zu behandeln.

Entscheidung: Wir gehen davon aus das jede Konfiguration unter Docker läuft. Die NATS-Konfiguration wird über eine eingebundene Konfigurationsdatei bereitgestellt. Vor dem Anwenden einer Änderung wird die Konfiguration validiert. Unterstützte Konfigurationsänderungen werden durch das Senden eines SIGHUP-Signals an den laufenden NATS Server übernommen. Konfigurationsänderungen, die kein Hot Reload unterstützen, erfordern einen kontrollierten Neustart des Containers. Das Neuladen und Neustarten muss protokolliert werden. Fehler bei der Übernahme müssen durch die Betriebs- oder Deployment-Werkzeuge erkennbar sein.

```
docker exec nats nats-server -c /etc/nats/nats.conf -t \  
&& docker kill --signal=SIGHUP nats
```

Positiv: Unnötige Container-Neustarts werden bei unterstützten Konfigurationsänderungen vermieden. Das Verfahren zur Übernahme von Konfigurationsänderungen ist einheitlich definiert. Die Validierung der Konfiguration ist von ihrer tatsächlichen Übernahme getrennt.

Negativ: Nicht alle Konfigurationsänderungen können zur Laufzeit übernommen werden. Eine erfolgreiche Validierung garantiert keine erfolgreiche Übernahme durch den laufenden Server. Die Betriebswerkzeuge müssen zwischen Hot-Reload-fähigen und neustartpflichtigen Änderungen unterscheiden.

11. Risiken und technische Schulden

Die Risiken unterscheiden sich stark von den Risiken zum aktuellen Master-Klient-Sync in CX5.

Darunter sind folgende Themen:

- Mehrwehr für Kunden gegenüber aktuellem Master-Klient-Sync
- Offene Themen im CIMgate.Hub in der Architektur
- Migration der Laufenden Sync-Umgebungen zum Hub
- Organisatorische Risiken

Die Bewertung (Eintritt / Auswirkung: hoch · mittel · gering) ist eine erste Einschätzung und muss im Team validiert werden.

11.1 Risiken

Risiko	Beschreibung	Eintritt	Auswirkung	Maßnahme	#
Höhere Betriebskomplexität	Der bestehende Sync wird aus CommandX genommen und durch ein Docker Stack aus PostgreSQL, Debezium, NATS und dem CIMgate ersetzt. D.h. mehr bewegliche Teile und mehr Betriebs-Know-How intern und bei Kunden.	hoch	mittel	Anleitungen und frühe Einarbeitungen	RI-01
Kein Kunden Mehrwert für UC1/UC2	Der Nutzen für Kunden gegenüber aktueller Master-Client-Sync ist aktuell in Frage vor allem durch RI-01	hoch	hoch	Stabile Verbindung, Geringerer Entwicklungsaufwand bei neuen Modulen, Retention und Replay von Events zur Fehlersuche (auch abseits des Hubs), Größter Mehrwert ist der spätere UC3/UC4 Fall	RI-02
Nutzen in schmalen Netzen gegenüber aktuell Sync nicht belegt	Der Betrieb bei 600 ms–2 s RTT ist aktuell nicht belegt. Es gibt kein Expliziten ansatz Priorklassen zu nutzen. Jeder Stream wird von NATS Source-Streams eigen verwaltet. Ausgeschlossen sind aktuell nur angedachte on-demand Blobs.	hoch	mittel	Tests und Konzept für Priorklassen	RI-03

Risiko	Beschreibung	Eintritt	Auswirkung	Maßnahme	#
Datenbank-Voraussetzungen beim Kunden	Die Architektur per CDC setzt PostgreSQL mit logischer Replikation oder eine MSSQL Lizenz voraus, aktuelle MSSQL Express Installationen werden nicht unterstützt.	mittel	mittel	Umstellung in CX5 auf PostgreSQL im Release 26.09 ohne Migrator vorhanden.	RI-04
Anpassung Stammdaten FK in CommandX	Für UC3 und UC4 müssen alle Missionszugehörigen Objekte in CX5 unabhängig von Stammdaten "funktionieren". Teilt eine Organisation seine Mission mit einer andern Organisation sollen keine Stammdaten in der anderen Datenbank landen.	hoch	hoch	Durch die Anforderung der Revisionssicherheit von abgeschlossenen Mission in CX5 "sollten" Stammdaten bereits unabhängig von der Mission leben. Eine aktuelle Veränderung einer Meldungsart von "Eilmeldung" auf "Gedöns" wirkt sich auf alle (auch abgeschlossene) Missionen aus. Stammdaten in CX5 sollten nur als Templates nutzbar sein und müssten in jede Mission kopiert werden. Zusammenarbeit im Hub wäre dann über Missionsobjekte ohne Stammdaten möglich. UC5 bringt dann importierbare Templates.	RI-05
Retention-Policy und Datenmengen unklar	Es gib aktuell keine konkreten Ansatz, welcher die echten anfallenden Datenmengen bei der Nutzung im Produktiv-System aufzeigt. Somit sind Aussagen zu der benutzbaren Retention-Policy, sowie Systemanforderungen zur Festplattengröße nicht möglich.	mittel	mittel		RI-06
		mittel			RI-07
Unterstützung von gleichzeitigem Betrieb von CIMgate.Hub und RabbitMQ	Events aus WAL werden nicht ohne weiteres Zutun im Rabbit ankommen				RI-08
sensibele Daten	Durch den Hub könnten sensibele Daten an Dritte gelangen, wenn bei der Weitergabe der Events nicht genau vorher alle Events und Tabellen überprüft und auf die Spalten, sowie die Fremdnutzung von Spalten geachtet wird.	mittel			RI-09

11.2 Technische Schulden

Weitere entstehenden technischen Schulden müssen während der Entwicklung identifiziert und ergänzt werden.

Technischer Debt	Komponente	Beschreibung
------------------	------------	--------------

Offene Fragen

F1

Was ist die Definition / Begriff von Partnern?

- Fester Partner (UC1 - INSTANCE-Sync, UC2 DEPARTMENT-Sync)
- AdHoc Partner (UC3 - ADHOC-Sync)
- Federated Partner (UC4 - FEDERATED-Sync)

F2

B-012 Verknüpfen von geteiltem Einsatz zu eigenem Einsatz Führt die Gegenseite bereits einen eigenen Einsatz zur selben Lage, kann sie den geteilten Einsatz mit der Lizenz Großschadenslage bzw. Sonderlage mit ihrem eigenen verknüpfen oder zusammenführen. Was bedeutet hier das Verknüpfen von den Einsätzen?

F3

Ist es korrekt das es die Anforderung gibt: Der Einladungscode zum Einsatzteteilen soll über Funk oder Telefon übergeben werden können

Hosting und Freigabe von NATS (6.e) Kann jeder der ein NATS betreiben möchte und Initiator sein will um Einsätze nach außen zu teilen seinem NATS per IPv4 + Port freigeben? Oder muss Domainname genutzt werden können?

F4

Was nehmen wir zu identifizierung eines Hub-Nodes? Client, Id, Beschreibung aus der CX5 Lizenz, was ist wenn die mehrfach aktiviert sind, sollen die überhaupt oder fest eine lizenz pro HUB die ihre ID festlegt?

F5-7 Data handling

F5 - Handing von Dokumenten

Nur AdHoc Dokumente?

Retention und wie andere die Abholen ist offen.

Haben alle die Berechtigungen alle Meldungenanhänge im ETB, oder andersweitige verknüpfte Dokumente von der Mission einzusehen?

Sicherheitsbedenken wegen Byte-Austausch von schädlichen Dateien ist hier hoch.

Dokumente könnten in Nats Jetstream Documents geladen werden.

übernimmt transfer und chunking der bytes. Dokumente können dann wie ein Topic abgeholt werden? Wie sind die Berechtigungen in Nats?

F6 - Stammdatenhandling & Globaleids

Betreffende Stellen in CX5 sind aktuell ungeklärt.

Zur Folge der Änderung sind UI Updates, Backend Updates, Datenbank Updates und voraussichtlich CX.Sync Updates in verschiedenen Modulen.

Welche Stammdaten werden bei Ad-Hoc Verbindung übertragen? → Am besten gar keine. Nur im FullSync und nur die Berechtigten (Ressourcen Administration)

Werden Stammdaten permanent importiert? → Verbindung wurde wegen einer geteilten Mission erstellt. Sollte die Mission beendet werden liegen bei Teilen der Stammdaten Informationen von dem anderen bereit, da Klären des Löschens von der Mission bei Beendigung nicht geklärt ist.

Was passiert bei Einsatzende? →

Wer darf welche Daten verändern? →

Was passiert wenn ein Objekt bereits existiert? → Einige Daten (wie Rollen) in CX werden mit festen ID's Preseeded, können anschließend vom Benutzer unter der ID umbenannt und umkonfiguriert werden. Der Hub darf diese Informationen bei unterschiedlichen CX nicht verknüpfen. Symbole sind ebenfalls preseeded Daten mit fester ID.

Wie bekommen Ressourcen Symbole wenn Symbole (Stammdaten und nachträglich erstellte) nicht ausgetauscht werden? Gibt es Objekte in 2 verschiedenen CX5 Instanzen die das gleiche Objekt sind mit unterschiedlicher ID?

Sollen z.B. Prod- und Test-System verbunden werden hat die selbe Ressource eventuell eine andere ID. Da CIMgate.Connect die Ressource angelegt hatte. Gleiche ExternalID anderer PK.

Ansich fein wenn nur Mission geteilt wird und alles in der Mission lebt, keine Stammdaten ausgetauscht, angelegt oder überschrieben werden.

Aber CIMgate müsste auf HubOrigin achten und Ressourcen nur von seiner erstellten Instanz updaten, oder Mechanismen zum Übergeben von Ressourcen, wer der Owner ist wird updaten. Fortlaufende Meldungsnummern? Missionsnummern?

F7 - GPS und Positionsdaten

Wieviele Updates an MissionRessourcen werden durch GPS erzeugt.

Muss eine gesonderte Tabelle für Positionsdaten erstellt werden, um die RWs von der MissionsRessourcen-Tabelle zu entnehmen.

Muss eine besondere Sync-Strategie hierfür her? Wie ist es mit Positionshistorisierung, ist ein Klient Offline für X Zeit würde aktuell?

collab.mission.123.missionressource.FW01

nur der letzte Datensatz behalten werden.

Darf überhaupt Historisierung bestehen? aktuell wird keine Historien-Tabelle beachtet im Sync. Muss man sehen was das für den MapReplayStream bedeutet. Historisierung der Positionsdaten im FullSync. Soll da nur die Letzte ResourcePosition rüberkommen, weil der Lagekartenfilm sonst falsche Sicht anzeigt. Master kannte die Position der Ressource ja nicht zu dem Stand auf der Lagekarte. Oder soll es den Verlauf widerspiegeln und einspielen ist gewünscht, damit der Master die Daten vorhalten kann und Verlauf der Ressourcen Historisieren kann?

F8

UC1 - Spiegelung Es wird nur von A <> B gesprochen. Was ist wenn es C gibt, A <> B A <> C. A ist die einrichtende Stelle, Verbindung zu A geht verloren.

F10 - Kombination von Vernetzungsarten

3.6 Kombination und mehrere Beteiligte

Darf B die Mission von A einfach Teilen, wenn die Mission Owner A ist? Wie kann A die Mission zu B übergeben? Wie lösen sich die Verbindungen. B muss zum Master werden, alle Teilnehmer müssen sich dann zu B als Master verbinden. Oder Multi-Sharing handling muss erlaubt sein. Muss innerhalb AdHoc <> AdHoc <> FullSync Verbindung beachtet werden.

☐ An einem Einsatz können mehrere Instanzen beteiligt sein, z. B. Polizei und THW gleichzeitig. Alle Beteiligten sehen sich gegenseitig. ☐ Ein gespiegeltes Paar (UC 1) wirkt nach außen wie eine Seite. Geteilt wird von der Instanz aus, an der die teilende Funktion arbeitet. Fällt eine Instanz des Paares aus, läuft das Teilen über die andere weiter. ☐ Ein Einsatz einer UC-2-Befehlsstelle kann zusätzlich ad hoc geteilt werden, von der Instanz des Besitzers aus. Wie ist die Zusammenarbeit zwischen UC bei Kombination gewünscht? Wenn ein UC1 eine UC3-Mission annimmt, syncet die zu UC1 Partnern?

F11

StammdatenPortal

UC 5 ist fachlich unabhängig von der Synchronisation und keine Kopplung. Über das Portal übernommene Daten behalten aber ihre Herkunftskennung (External ID). Das hilft nebenbei, gemeinsame Daten beim Teilen

in UC 3 und UC 4 eindeutig zu erkennen, etwa wenn beide Seiten dasselbe Fahrzeugpaket nutzen. Was ist wenn die das gleiche Paket nutzen aber Fahrzeuge umbenennen? Wobei soll die External-Id helfen?

Dev Notes

CX5 Anpassung

- FK Keys in Tabellen
- CX5 Stammdaten in Mission

—;

Kann wir B-015 wie B-010 iwi schlau regeln? hängen wir da in das Subject / Header immer Department mit rein, dann müssen wir das für jeden Datensatz erst berechnen. Aktuelle MissionId Erkennung dann zu MissionShare und alle Departments, aber dann Teilt jemand die Mission zu nem weiteren Department und der hat die vorherigen Events nicht, also muss er zurück springen und die Events für diese mission nachholen

—;

RetentionPolicie 30Tage, höher besser aber mehr speicherverbrauch - verbrauch aktuell unbekannt

6.4 Liegt die Lücke jenseits der Stream Retention, wechselt der Share auf OutOfSync; der Initiator löst einen Snapshot aus; Daten werden zusammengeführt

Wie Kann man die Luecke und fehlenden Daten erkennen? Consumer muss MSG Sequenzen tracken und sagen das dort zwischen der vorherigen und neuen nachricht einige verlogen gegangen sind

—;

Prio-Klassen nicht festgelegt

Umsetzung von A-006 muss geprüeft werden und Prio-Buckets sollten ermittelt werden.

Prio wie "nur aktive Mission" ist fragwuerdig ob daruch Performance gewonnen wird. Einige Tabellen z.B. Stammdaten-Tabellen ausschliessen ist plausibel und umsetzbar, haengt jedoch von CX5 FK Keys ab

—;

Deletes braucht Tombstone Message mit Ids

Metadaten für LoopPrevention, geht das wenn before nicht geholt wird? Wie erfolgt ohne before LoopPrevention? MSSQL und PG beachten.