



KASSENÄRZTLICHE
BUNDESVEREINIGUNG

Definition Schnittstelle 112 - 116117

Thema: Erläuterung zur technischen Schnittstelle



Version 1.11
Stand: 19. September 2023

Erstellt durch:



D.I.E. PROJEKT GmbH
Buchenstraße 12
01097 Dresden

Tel.: +49 351 / 4 79 36 0
Fax: +49 351 / 4 79 36 299
E-Mail: P1508@diep-ingenieure.de

Inhaltsverzeichnis

1	Zielstellung	4
2	Schnittstellendefinition und Realisierung	5
3	JSON-Format	7
4	Bereitstellung und Übermittlung relevanter Daten	8
4.1	Allgemein	8
4.2	Informationsobjekt „Einsatzinformationen“	9
4.2.1	Versions-DTO	9
4.2.2	Zusatzdaten (Appendices)	10
4.2.2.1	Incident Appendices“	10
4.2.2.2	Informer-Appendices	12
4.2.2.3	Patient Appendices	13
4.3	Informationsobjekt „Statusmeldung Einsatz“	14
4.4	Informationsobjekt „Statusmeldung Ressource“	14
4.5	Informationsobjekt „Einsatzrecherche“	14
5	Recherchefunktionen	15
6	Architektur der Kommunikation	16
7	Kommunikationsablauf	18
8	Organisatorische Rahmenbedingungen	20
9	Anlagen	21

1 Zielstellung

Die Kassenärztliche Bundesvereinigung (KBV) mit Sitz in Berlin ist einer der maßgeblichen Akteure im deutschen Gesundheitswesen. Gemeinsam mit den Kassenärztlichen Vereinigungen (KVen) steht die KBV für die flächendeckende, wohnortnahe, qualitätsgesicherte, ambulante Versorgung rund um die Uhr.

Die Aufgaben des Rettungsdienstes sind nach dem Föderalismusprinzip des Grundgesetzes durch Landesgesetze geregelt und wird durch die BOS- Rettungsleitstellen (bzw. Integrierten Leitstellen) 112 wahrgenommen. Die Rettungsleitstellen 112 sind die Hauptakteure in der Notfallrettung des deutschen Gesundheitswesens.

Im Rahmen dieses sogenannten Sicherstellungsauftrags betreibt die KBV u. a. die bundesweit einheitliche Rufnummer 116117 für den Ärztlichen Bereitschaftsdienst. Zur Weiterentwicklung dieses Dienstes benötigt die KBV die Definition einer bundeseinheitlichen Software-Schnittstelle zwischen den KV-Servicestellen 116117 und den Rettungsleitstellen 112, damit die jeweils erhobenen Patientendaten bidirektional übergeben werden können.

Gleichzeitig wird eine Kommunikationsarchitektur benötigt, in der festgelegt ist, wie die Kommunikationswege zwischen den o.g. Akteuren (KV-Servicestellen 116117 / Rettungsleitstellen 112) erfolgt.

Das vorliegende Dokument beschreibt die dafür notwendige Schnittstelle sowie die Kommunikationsarchitektur für den Datenaustausch zwischen den KV-Servicestellen 116117 und den Rettungsleitstellen 112.

2 Schnittstellendefinition und Realisierung

Für die technische Implementierung der geforderten Schnittstelle wird der Einsatz eines standardisierten Kommunikationsverfahrens empfohlen. Damit wird sichergestellt, dass eine herstellernerneutrale und barrierefreie Integration in verschiedene Softwarearchitekturen möglich ist. Zudem können zusätzliche Informationsflüsse und Datenstrukturen einfach und schnell umgesetzt werden, um somit flexibel auf geänderte Anforderungen seitens der KV-Servicestellen 116117 und den Rettungsleitstellen 112 reagieren zu können.

Die Schnittstelle muss schwerpunktmäßig die Maschine-zu-Maschine-Kommunikation sicherstellen, welche die Disponenteninformationen automatisch zum zentralen Kommunikationsserver übermittelt und weiteren Disponenten bereitstellt. Eine Quittierung der eingehenden sowie abgehenden Informationen soll ebenso möglich sein wie die zusätzliche Verknüpfung mit weiteren Informationen aus anderen Systemen der eigenen Leitstelle (beispielsweise der Dokumentenverknüpfung zu Patientendaten). Hierfür eignen sich die folgenden standardisierten Programmierschnittstellen (Application Programming Interfaces [API]) bzw. Schnittstellenbeschreibungssprachen:

- Representational State Transfer (REST)-API: Ein standardisiertes Übertragungsmodell zur Abstraktion des Aufbaus und des Verhaltens des World Wide Web. Mit Hilfe von unterschiedlichen Befehlen (GET, POST, PUT etc.) können mittels des HTTP bzw. HTTPS-Protokoll Daten und Programmzustände übertragen werden.
- gRPC Remote Procedure Calls (kurz gRPC): Ein Protokoll zum Aufruf von Funktionen mittels HTTP/HTTPS-Standards. Das quelloffene System erlaubt die standardisierte Übertragung von Funktionen und Zuständen in verteilten Netzen.
- Simple Object Access Protocol (SOAP): Standardisiertes Netzwerkprotokoll zur Übertragung von Daten zwischen verteilten Systemen. Die Verwendung von SOAP wird häufig in der Kombination mit dem Web Services Description Language (WSDL) verwendet, welche eine plattform-, programmiersprachen- und protokollunabhängige Übertragung ermöglicht.

Im Hinblick auf das Einsatzszenario bei der KBV (Übertragung/Anforderung/Quittierung von Einsätzen) soll eine einfache und schlanke Schnittstellenimplementierung bevorzugt werden. Hierfür muss der Einsatz einer REST-Schnittstelle erfolgen, welche vorrangig für eine Punkt-zu-Punkt-Kommunikation entwickelt wurde und an das Prinzip des World Wide Web angelehnt ist. SOAP hingegen folgt dem Prinzip der Übertragung von XML-basierten Daten, welche zwar eine komplexere Struktur (SOAP-Header, SOAP-Body, SOAP-Fault) erlaubt, jedoch eher für den Einsatz im verteilten Rechnen designt wurde. Durch die Verwendung unterschiedlicher Erweiterungen von SOAP in Kombination mit weiteren Webservices (WS-*) können wesentlich umfangreichere Implementierungen verfolgt werden, stehen jedoch dem KBV-Bedarf einer „einfachen“ (lightweight) Umsetzung gegenüber.

Die Implementierung von gRPC hat gegenüber der REST-API einige Vorteile. Das HTTP/1.1-Protokoll, welches REST ausschließlich unterstützt, hat den Nachteil einer hohen Komplexität im Aufbau (Standardisierung durch RFC 7230, 7231, 7232, 7233, 7234, und 7235), einer höheren Latenz (jedes TCP-Paket muss quittiert werden) sowie die Restriktion des gleichzeitigen Aufbaus von mehreren Verbindungen zu einer gleichen Domäne (bis zu 2 gleichzeitige Ver-

bindungen). HTTPS/2, welche von gRPC unterstützt wird, löst nicht nur diese Probleme, sondern erlaubt auch das Streamen von Daten während einer Punkt-zu-Punkt-Verbindung. Die Übertragung von Daten von einem Client zum Server kann solange aufrechterhalten werden wie nötig, anschließend erfolgt die einmalige Quittierung des Streams. Ein zusätzlicher Vorteil in der Verwendung des HTTP/2-Protokolls liegt darin, dass dieses ausschließlich im Binärformat vorliegt, welche einfacher verschlüsselt und komprimiert werden kann. Jedoch bietet REST mit HTTP/1.1 und der damit verbundenen Verwendung des JSON-Formats einen entscheidenden Vorteil für die Implementierung als KBV-Schnittstelle: deren einfache Erweiterung der Übertragungsdaten. gRPC unter Verwendung von HTTPS/2 akzeptiert und überträgt Protobuf-Nachrichten, auch Protocol-Buffer genannt. Deren Inhalte müssen zum Implementierungszeitraum genau bekannt sein und dürfen während der gesamten Laufzeit nicht geändert werden.

```
message Mission {  
  required string SharedIncidentID = 1;  
  required string TransactionNumber = 2;  
  required string Classification = 3;  
  required string Issue = 4;  
}
```

```
Einsatz mission = Mission.newBuilder()  
    .setSharedIncidentID (4477)  
    .setClassification("Rettungseinsatz")  
    .setTransactionNumber("2019-11-21-001")  
    .setIssue("Notfall am Gleis")  
    .build();  
output = new FileOutputStream(args[0]);  
mission.writeTo(output);
```

Abbildung 1: Aufbau des Protocol Buffer (links) und deren Initialisierung und Verwendung in Java (rechts)

Dieses sogenannte „Strong-Typing“ gewährt zwar bei der Implementierung die Sicherheit, dass alle Parameter zwingend definiert und bereitgestellt werden, macht jedoch eine nachträgliche Erweiterung fast unmöglich. Neue Definitionen und Änderungen an der Schnittstelle müssen allen Herstellern der Distributionssoftwares in den KBV-Leistellen vorab genannt und bis zum Umschalttag implementiert sein. Anschließend erfolgt eine komplette Umstellung des Systems mit eingeplanter Downtime, währenddessen der Dienst nicht zur Verfügung steht. Die REST-Schnittstelle unter Verwendung des JSON-Formates erlaubt die dynamische Serialisierung während der Laufzeit eines Programmes. Dabei können neue Felder problemlos dem JSON-Format hinzugefügt und in die Schnittstelle implementiert werden. Anschließend können die Softwarehersteller der Distributionssoftware die neuen Feldtypen implementieren und in einer neuen Version verteilen. Eine Rückwärtskompatibilität ist somit stets gewährleistet.

3 JSON-Format

Unter Zuhilfenahme des JSON-Formates können alle notwendigen Daten übertragen und verteilt werden. Das Format benutzt standardmäßig die UTF-8 Zeichenkodierung und definiert die folgenden Datentypen.

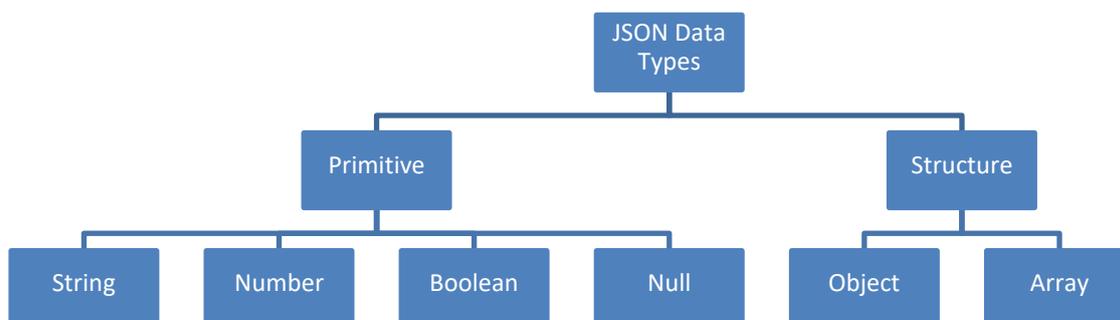


Abbildung 2: JSON-Datentypen nach Standard RFC-8259

Das Datenformat liegt in einfacher Textform vor und erlaubt somit eine einfache Lesbarkeit des Anwenders. Zudem ist JSON unabhängig von jeglichen Programmiersprachen, entsprechende Bibliotheken werden jedoch von jeder gängigen Sprache bereitgestellt und vollumfänglich unterstützt. Für die Schnittstellenprogrammierung von Vorteil ist seine Serialisierbarkeit: Das Objekt existiert mehrfach. Änderungen während der Laufzeit eines Programms am Objekt ändern nicht die Ausgangsdaten (z.B. auf den Remoteserver), sofern dies nicht explizit angeordnet wird (deserialisiert). Änderungen des Ausgangsobjektes haben ebenfalls keine Auswirkungen auf das Objekt während der Programmlaufzeit, es kann somit beliebig erweitert werden. Dies erlaubt die Bereitstellung neuer Felder und Typen durch den zentralen Vermittlungsserver, welche, sofern bereits durch die Clientsoftware berücksichtigt, mitverarbeitet oder ignoriert werden. Eine Abwärtskompatibilität ist somit uneingeschränkt möglich.

4 Bereitstellung und Übermittlung relevanter Daten

4.1 Allgemein

Für eine optimale Einsatzbearbeitung und Weiterleitung von einer KV-Servicezentrale zu einer Rettungsleitstelle 112 und umgekehrt erfolgt die Übermittlung über einen zentralen Kommunikationsserver. Dieser Server hat lediglich die Funktion einer Vermittlung zwischen Sender und Empfänger, d.h. eines typischen Hubs (siehe Abschnitt 5). In diesem zentralen Vermittlungsserver werden neben den Verbindungsdaten auch Sachdaten in Form von technischen Logfiles verarbeitet.

Die dabei zu übertragenden Informationen werden in 4 Informationsobjekte (Klassen) eingeteilt:

- **Einsatzinformationen:** Die Aufnahme der relevanten Daten durch den Anrufer in der jeweiligen KV-Servicezentrale 116117 oder Rettungsleitstelle 112 werden in einer Einsatzinformation von einem Disponenten / Agenten aufgenommen. Zusätzliche Daten können aus einem Einsatzleitsystem ebenfalls mit einfließen (z.B. Geo-Koordinaten des Notrufstandortes). Die aufgenommenen Daten werden anschließend an den zentralen Vermittlungsserver übertragen und können von hier an weitere Rettungsstellen 112 und KV-Servicezentralen 116117 vermittelt werden. Die adressierte Leitstelle holt sich diese Daten mittels einer Client-Software ab und verarbeitet diese entsprechend in ihrem Einsatzleitsystem.
- **Statusmeldung Einsatz:** Enthält aktuelle Informationen über den Zustand und Fortschritt eines Einsatzes. Mit Hilfe einer Einsatz.ID, welche ebenfalls in der Einsatzinformation definiert ist, kann eine entsprechende Zuordnung erfolgen. Die Statusmeldung enthält zudem Informationen zum aktuellen Zustand, seiner Priorität und Meldungstyp (z.B. Historie, Chat etc.).
- **Statusmeldung Ressource:** Jedem Einsatz müssen entsprechend Ressourcen zugeordnet werden, wie z.B. Rettungskräfte oder Ärzte. Die Verknüpfung mit der Einsatzinformation erlaubt eine genaue Zuordnung der notwendigen Ressourcen. Informationen über ihren Status („Rücke an“, „vor Ort“, „In Behandlung“) sowie weitere Daten werden der zusätzlich eingebundenen Leitstelle kontinuierlich übermittelt. Zudem erlaubt dieses Objekt auch die Verlinkung mit weiteren Einsatzressourcen, um eine effektive Koordinierung zu ermöglichen.
- **Einsatzrecherche:** Um doppelte und kostenintensive Einsätze zu verhindern, weil bereits ein Einsatz angelegt wurde und sich in der Bearbeitung befindet, wird bei jeder Einsatzannahme ein minimaler Datensatz erzeugt und angelegt. Dieser Datensatz enthält nur die notwendigen Daten für eine Recherche (z.B. Zeit, Anrufer, angerufene Leitstelle, Einsatznummer usw.)

4.2 Informationsobjekt „Einsatzinformationen“

Der Aufbau und die Bezeichnung der nachfolgenden Felder orientiert sich an der Universellen Leitstellenschnittstelle UCRI in der Version 1.0 vom 17.06.2020 (siehe Anlage 005).

Die farbig hinterlegten Felder sind Pflichtfelder, sie stellen das Minimum Set of Data (MSD) dar und müssen in jedem Einsatzdatensatz vorhanden sein.

Die rot gekennzeichneten Felder, sind zusätzliche (ergänzende) Datenfelder zur o.g. Leitstellenschnittstelle UCRI, die für die Patientenübergabe zwischen der 112 und 116117 erforderlich sind. Die Leitstellenschnittstelle UCRI erlaubt die Erweiterung um zusätzliche Felder.

4.2.1 Versions-DTO

Beschreibt die Version der Einsatzdatenaustausch-Schnittstelle.

Feld (Eigenschaft)	Typ (JSON)	Beispiel	Beschreibung	Hinweise
ucriVersion	String	1.0	Version der UCRI Schnittstellen Spezifikation	obligates Feld

4.2.2 Zusatzdaten (Appendices)

4.2.2.1 Incident Appendices“

Aktuell sind folgende Appendices für Einsätze bekannt und erforderlich:

Feld (Key)	Typ (JSON)	Beispiel	Beschreibung	Hinweise
sharedIncidentID	String	4711	GUID zur Identifizierung des Einsatzes auf allen beteiligten Systemen inkl. dem Leitstellenhub	obligates Feld , wird erst später vom Leitstellenhub befüllt
IncidentID	String	2021-01-05-KV381	eindeutige ID des Einsatz	obligates Feld
SourceID	String	58	Identifikation der sendenden Leitstelle	obligates Feld
TargetID	String	28	Identifikation der Ziel Leitstelle	obligates Feld
ControlCenterName	String	RLSt. Musterstadt	Name der sendenden Leitstelle	obligates Feld
IncidentStartTimeStamp	Time-Stamp	2021-01-05T13:20:00.000Z	Zeitstempel des Einsatzbeginns (UTC)	fakultatives Feld
TransactionNumber	String	KV381	Vorgangsnummer (ggf. lesbare ID)	obligates Feld
Classification	String	KV, Rettungsdienst	Stichworte (Komma getrennt)	obligates Feld
Issue	String	Notfallpatient	Taktische Bewertung des Einsatzes: Sachverhalt	obligates Feld
IncidentAdditionalInfo	String	Eingang befindet sich auf der Gebäuderückseite	Freitext zur Beschreibung des Einsatzes, Zusatzinfo	fakultatives Feld
IncidentDispatcherID	String	2274	ID des Disponenten der Sender-Leitstelle	fakultatives Feld
IncidentDispatcherName	String	Schenker	Name des Disponenten der Sender-Leitstelle	fakultatives Feld
IncidentDispatcher-Reach	String	+4930123456789	Erreichbarkeit (Rufnummer) des Disponenten der Sender-Leitstelle	fakultatives Feld
sentByDispatcherAt	Time-Stamp	2021-01-05T13:19:00.000Z	Zeitpunkt der Übergabe der Anfrage (UTC) durch den Dispatcher (Sender)	obligates Feld
acceptedByDispatcherAt	Time-Stamp	2021-01-05T13:13:00.000Z	Zeitpunkt der Annahme der Anfrage (UTC) durch den Dispatcher (Empfänger)	fakultatives Feld
TargetDispatcherID	String	2274	ID des Disponenten der Empfänger-Leitstelle	fakultatives Feld
TargetIncidentID	String	2021-01-05-2553	Eindeutige ID des Einsatzes (z.B. Einsatznummer) der Empfänger-Leitstelle	fakultatives Feld
IncidentStartTimeStamp	Time-Stamp	2021-01-05T13:20:00.000Z	Zeitstempel des Einsatzbeginns (UTC) der Empfänger-Leitstelle	fakultatives Feld

Wichtig, die Einsatzstelle ist die Stelle, an der sich der Patient aktuell befindet (u.U. abweichend zum Standort des Mitteilers).

Feld (Key)	Typ (JSON)	Beispiel	Beschreibung	Hinweise
Einsatzstelle				
MissionDestination	String	Berlin	Einsatzstelle: Ort	obligates Feld
MissionCityDistrict	String	Charlottenburg	Einsatzstelle: Stadtbezirk	fakultatives Feld
MissionPostalCode	String	10585	Einsatzstelle: PLZ	obligates Feld
MissionObject	String	Rathaus	Einsatzstelle: Bezeichnung Objekt, Firmenname, ...	obligates Feld
MissionObjectType	String	Hochhaus	Einsatzstelle: Objekttyp (z.B. Krankenhaus, Krankenhausabteilung, Kindergarten, Tankstelle, ...)	fakultatives Feld
MissionSuperiorObject	String	Hauptgebäude	übergeordnetes Objekt der Einsatzstelle (z.B. Krankenhaus bei Angabe einer Abteilung als Objekt)	obligates Feld
MissionAccessibility	boolean	true	Zugang zur Einsatzstelle uneingeschränkt möglich [true false]	obligates Feld
MissionAdditionalInfo	String	Klingel defekt, XXX anrufen / Schlüsselnummer HNR1567 in Tresor W12 Vorsicht Gefährdung vorhanden!	Freitext zur Beschreibung von Besonderheiten an der Einsatzstelle	obligates Feld
MissionStreet	String	Otto-Suhr-Allee	Einsatzstelle: Straße	obligates Feld
MissionHouseNumber	String	100	Einsatzstelle: Hausnummer	obligates Feld
MissionStreetSegment	String	mitte	Einsatzstelle: Straßensegment	obligates Feld
MissionFloor	String	3	Einsatzstelle: Etage	obligates Feld
MissionDoor	String	304	Einsatzstelle: Tür	obligates Feld
MissionStaircase	String	5	Einsatzstelle: Treppenhaus	obligates Feld
MissionLongitude	String	52.51738218624819	Einsatzstelle: Längengrad	obligates Feld
MissionLatitude	String	13.308641578255477	Einsatzstelle: Breitengrad	obligates Feld
AdditionalField01	String		Zusätzliches freies Feld 01	fakultatives Feld
AdditionalField02	String		Zusätzliches freies Feld 02	fakultatives Feld
AdditionalField03	String		Zusätzliches freies Feld 03	fakultatives Feld

4.2.2.2 Informer-Appendices

Aktuell sind folgende Appendices für den Mitteleiler bekannt und erforderlich:

Feld (Key)	Typ (JSON)	Beispiel	Beschreibung	Hinweise
InformerName	String	Meier	Nachname des Mitteleilers	obligates Feld
InformerFirstName	String	Isolde	Vorname des Mitteleilers	fakultatives Feld
InformerMiddleName	String	-	Zweitname des Mitteleilers	fakultatives Feld
InformerAdditionalInfo	String	Schwester Isolde	Freitext zur Beschreibung des Mitteleilers	fakultatives Feld
InformerAge	Number	58	Alter des Mitteleilers	fakultatives Feld
InformerGender	String	female	Geschlecht des Mitteleilers [female male diverse]	fakultatives Feld
InformerReach	String	+4930987654321	Erreichbarkeit des Mitteleilers	obligates Feld
Adresse				
InformerCity	String	Berlin	Mitteleiler: Stadt	fakultatives Feld
InformerCityDistrict	String	Charlottenburg	Ortsteil des Mitteleilers	fakultatives Feld
InformerPostalCode	String	10585	Mitteleiler: PLZ	fakultatives Feld
InformerStreet	String	Otto-Suhr-Allee	Mitteleiler: Straße	fakultatives Feld
InformerHouseNumber	String	100	Mitteleiler: Hausnummer	fakultatives Feld
InformerSegment	String	Mitte	Mitteleiler: Straßensegment	fakultatives Feld
InformerFloor	String	3	Mitteleiler: Etage	fakultatives Feld
InformerDoor	String	301	Mitteleiler: Wohnungstür / -nummer	fakultatives Feld
InformerStaircase	String	5	Mitteleiler: Treppenhaus	fakultatives Feld
InformerObject	String	Rathaus	Mitteleiler: Bezeichnung Objekt, Firmenname, ...	fakultatives Feld
InformerObjectType	String	Hochhaus	Mitteleiler: Objekttyp (z.B. Krankenhaus, Kindergarten, Tankstelle, ...)	fakultatives Feld
InformerSuperiorObject	String	Hauptgebäude	Mitteleiler: Übergeordnetes Objekt (z.B. Krankenhaus bei Angabe einer Abteilung als Objekt)	fakultatives Feld

4.2.2.3 Patient Appendices

Zu diesem Zeitpunkt sind folgende Appendices für Patienten bekannt und erforderlich:

Feld (Key)	Typ (JSON)	Beispiel	Beschreibung	Hinweise
PatientFirstName	String	Rita	Vorname des Patienten	obligates Feld
PatientMiddleName	String	-	Zweitname des Patienten	fakultatives Feld
PatientLastName	String	Werner	Nachname des Patienten	obligates Feld
PatientReach	String	+4930987654321	Erreichbarkeit (Rufnummer) des Patienten	obligates Feld
PatientDateOfBirth	Number	1973-01-05	Geburtsdatum des Patienten (ISO 8601, Coordinated Universal Time)	fakultatives Feld
PatientAge	Number	48	Alter des Patienten	obligates Feld
PatientGender	String	female	Geschlecht des Patienten [female male diverse]	obligates Feld
Diagnosis	String	Kopfverletzung	Diagnose des Patienten (Achtung eine Diagnose darf nur durch einen Arzt gestellt werden!)	fakultatives Feld
PatientRMI	String	327	(3 stellige) Rückmeldeindikation = RMI	obligates Feld
PatientRMC	String	112323 oder b1,a2,v1,y0,k2,s1,n2	Rückmeldecode = RMC aus SNA	obligates Feld
PatientPZC	String	327453	(6 stelliger) Patientenzuweisungscode PCZ = RMI + Alter + Dringlichkeit	obligates Feld
PatientAilment	String	Atem erschwert, Kreislaufbeschwerden, Kopfverletzung	Freitext zur Beschreibung der Beschwerden (Meldebild)	obligates Feld

Das Informationsobjekt „Einsatzinformationen“ mit den in der oben dargestellten Abbildung beschriebenen Feldern muss von der Softwareschnittstelle als Mindestkriterium befüllt werden, um bei einer Übermittlung zu einem weiteren Disponenten voll auskunftsfähig zu sein. Die farbig hinterlegten Felder sind laut UCRI Spezifikation Pflichtfelder und müssen in jedem Einsatz-Datensatz vorhanden sein. Wichtige Informationen über den Anrufer wie seine Rufnummer und die Angaben zum Einsatzort werden vom jeweiligen Disponenten manuell oder automatisiert aufgenommen. Die Erfassung in der Notrufaufnahme und zusätzliche Übermittlung der Geokoordinate ist zudem ein sicheres Mittel, um bei unzureichender Lokalisierung durch Straße und Hausnummer (z.B. Einsatz im ländlichen Raum) eine gültige Einsatzposition anzugeben. Diese sind von einer entsprechenden Dispositionssoftware bereitzustellen, sind jedoch ebenfalls als Mindestkriterium anzusehen.

Die Felder „MissionLocation“, „Issue“, „Classification“ und „AilmentAdditionalInfo“ erlauben neben der örtlichen Komponente eine erste Einschätzung des Disponenten selbst, um später geeignete Einsatzmittel auszuwählen. Ergebnisse aus einem standardisierten Notruf-Abfrageprotokoll (SNA) der Rettungsleitstellen 112 bzw. aus SmED bei den KV-Servicestellen 116117 werden durch die entsprechende Auswahl direkt in das Feld „Issue“ eingesetzt. Sollte diese Software nicht im Einsatz sein, erfolgt eine manuelle Eingabe durch den Disponenten. Wichtig hierbei ist, dass durch das Feld „AilmentAdditionalInfo“ dem Mitarbeiter in weiteren Leitstellen die Möglichkeit gegeben wird, mittels Freitext weitere Informationen zu hinterlegen.

„Classification“ kann beispielsweise über vordefinierte Status (Drop-Down-Menüs) ausgewählt werden. Zusätzliche Angaben wie „IncidentID“ und „TargetIncidentID“ erlauben bei späterer Dokumentation und Auswertung früherer Einsätze eine genaue Zuordnung der jeweiligen Meldung. Die Syntaxprüfung der eingegebenen Angaben sowie das Ausfüllen aller relevanten Felder (obligat/fakultativ) ist durch die entsprechende Clientsoftware zu integrieren. Das Objekt „Einsatzinformation“ stellt in seiner derzeitigen Form die Mindestanforderungen dar, welche zwingend auszufüllen sind.

Durch die Verwendung von JSON als Übertragungsmedium kann das als „Einsatzinformation“ definierte Objekt über die Mindestkriterien problemlos erweitert werden. Spezielle Angaben oder zusätzliche Informationen in den Rettungsleitstellen 112 können sicher implementiert und gleichzeitig über die Schnittstelle übertragen werden. Es steht anschließend der eingesetzten Dispositionssoftware frei, ob nach zusätzlichen Feldern geparkt werden muss oder lediglich die Mindestkriterien umzusetzen sind.

4.3 Informationsobjekt „Statusmeldung Einsatz“

Diese Funktion wird vorerst zurückgestellt.

4.4 Informationsobjekt „Statusmeldung Ressource“

Diese Funktion wird vorerst zurückgestellt.

4.5 Informationsobjekt „Einsatzrecherche“

Diese Funktion wird vorerst zurückgestellt.

5 Recherchefunktionen

Zusätzlich zur Übermittlung von Einsätzen zwischen den Leitstellen und den KVen besteht der Wunsch nach einer Recherchefunktion, die es einer Leitstelle ermöglicht, bei Anrufeingang anhand der Rufnummer zu ermitteln, ob sich dieser Anrufer bereits in einem bestimmten Zeitfenster bei der 116117 oder 112 gemeldet hat und was das Ergebnis seiner Anfrage war.

Diese Funktion wird vorerst zurückgestellt.

6 Architektur der Kommunikation

Derzeit betreiben die KVen bundesweit 18 KV-Servicestellen 116117, und es werden in Deutschland 253 Rettungsleitstellen 112 betrieben.

In Summe handelt es sich um 271 Kommunikationsteilnehmer, somit sind maximal $n(n-1)/2 = 36.585$ Verbindungen nötig, damit alle Teilnehmer miteinander kommunizieren könnten.

Es gibt dafür prinzipiell 2 Architekturansätze:

- Punkt zu Punkt Verbindung (Point to Point / PtP)
- Zentrale Vermittlung (Hub and Spoke / HaS)

Unter Berücksichtigung der Anzahl der Verbindungen, des Aufwands zur Einrichtung und des Betriebsaufwands im laufenden Betrieb ergibt sich die logische Schlussfolgerung, dass lediglich die Kommunikation über einen zentralen Leitstellenhub die einzig vernünftige und wirtschaftliche Lösung darstellt.

Die Hub-and-Spoke-Architektur eines Netzwerks hat die Form eines Sterns. Die verschiedenen Kommunikationsendpunkte (Teilnehmer) sind jeweils mit einer Netzwerkverbindung (Spoke) mit dem zentralen Netzwerkknoten, dem Hub, verbunden. Sämtlicher Netzwerkverkehr fließt über den Hub. Die einzelnen Endpunkte des Netzes sind sternförmig über jeweils genau EINE Netzwerkverbindung mit dem zentralen Netzwerkknoten verbunden. Möchten einzelne Endpunkte miteinander kommunizieren, werden die Daten zunächst an den zentralen Knoten und dann weiter zum Endpunkt übermittelt. Es existieren keine direkten Verbindungen zwischen den einzelnen Endpunkten. Der Hub ist zentraler Knoten im Netzwerk und an jeglicher Kommunikation beteiligt. Ohne den Hub ist kein Datenaustausch möglich.

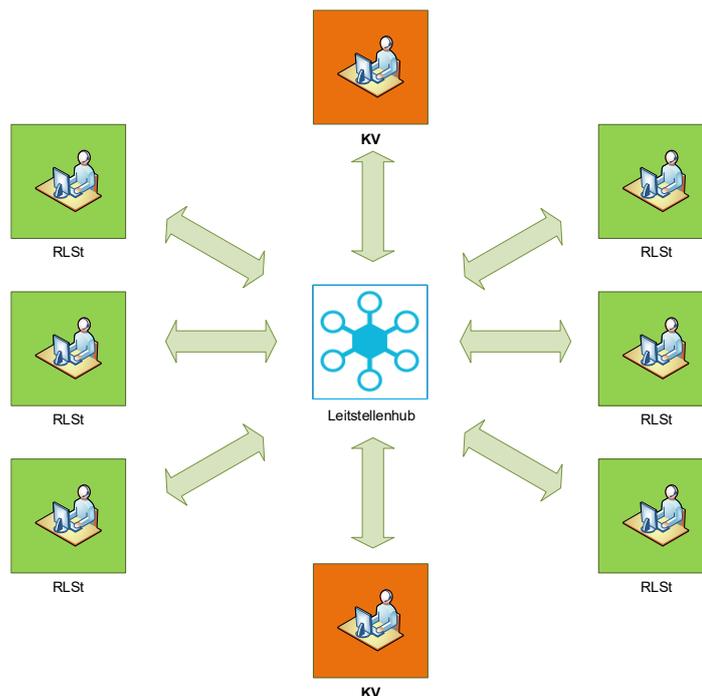


Abbildung 3: Kommunikationsarchitektur mit Leitstellenhub

Bekannt ist diese Architektur insbesondere bei Telefonanlagen und öffentlichen Telefonvermittlungen. Innerhalb eines Unternehmens sind die Telefone genau 1x (über eine Verbindung) mit der lokalen Telefonanlage (PBX) verbunden und diese Telefonanlage dann lediglich über deutlich weniger Kommunikationskanäle (x Verbindungen) mit der Telefonvermittlung (ISM) des zugehörigen Telefonanbieters (Carrier). D.h. die lokale Telefonanlage sorgt dafür, dass alle Teilnehmer innerhalb des Unternehmens, obwohl sie nur eine physische Telefonverbindung besitzen, mit allen anderen Teilnehmern sprechen können. Der öffentliche Telefonanbieter sorgt wiederum über seine Vermittlungsfunktion dafür, dass jeder Teilnehmer jeden anderen Teilnehmer auf der Welt über das Telefon anrufen kann, ohne dass die Beteiligten eine direkte physische Verbindung besitzen.

Die Anzahl benötigter Leitungen lässt sich in einem Hub-and-Spoke-Netz sehr einfach bestimmen. Sie entspricht der Anzahl der zu verbindenden Teilnehmer (ohne den zentralen Hub). Bei den vorbeschriebenen 271 Kommunikationsteilnehmern würden genau 271 Kommunikationskanäle (Spokes) zum zentralen Leitstellenhub aufgebaut und betrieben werden. Über diesen zentralen Leitstellenhub läuft dann nicht nur der Datenaustausch, sondern er bekommt zusätzlich folgende Funktionen:

- Autorisierung der Teilnehmer
- Festlegung der Kommunikationswege (wer darf mit wem kommunizieren) und damit einhergehend eine deutliche Reduzierung der notwendigen vorzuhaltenden Verbindungsmöglichkeiten.
- Management und Monitoring der Kommunikationswege (Bandbreiten usw.)
- Bereitstellung und Management von redundanten Kommunikationskanälen (im Bedarfsfall)

Da der zentrale Leitstellenhub einen single point of failure darstellt, muss er zwingend georedundant aufgebaut werden (x-fach Cluster).

D.h. ein relativ geringer technischer und organisatorischer Aufwand ist erforderlich, um ein solches System aufzubauen und zu betreiben und damit diese hohe Anzahl von Teilnehmern einfach miteinander kommunizieren zu lassen. Es bedarf jedoch eines Betreibers (inkl. Organisation und Prozesse), um diesen Leitstellenhub aufzubauen und 24/7/365 zu betreiben. Der Änderungsaufwand betrifft dann jedoch nur noch diesen Betreiber und nicht mehr die restlichen Kommunikationsteilnehmer.

Über diesen Leitstellenhub können dann sämtliche Teilnehmer kommunizieren.

In einigen Bundesländern sind bereits solche Hub and Spoke Systeme zum Datenaustausch zwischen den Rettungsleitstellen 112 im produktiven Einsatz (z.B. im Freistaat Sachsen, im Land Hessen).

7 Kommunikationsablauf

Nachfolgend wird der Kommunikationsablauf zwischen zwei Kommunikationsteilnehmern beschrieben. Dieser Ablauf gilt für beide Kommunikationsrichtungen.

Wichtig ist, bei der Entwicklung der Schnittstelle wurde hoher Wert auf eine synchrone Kommunikation gelegt. Das bedeutet, beide Kommunikationspartner (bzw. die Prozesse) beim Senden oder Empfangen von Daten werden immer synchronisiert, sie warten (blockieren) so lange, bis die Kommunikation abgeschlossen ist. Sowohl beim Senden als auch beim Empfangen wird immer gewartet. Das entspricht dem „normalen“ bzw. gewohnten Kommunikationsverhalten in Leitstellen. Der Sender stellt eine Anfrage und wartet auf die Antwort des Empfängers, vorher wird der Prozess (in diesem Fall der Einsatz) nicht beendet. Diese Art der Kommunikation findet man bei Faxprotokoll oder auch dem http-Protokoll.

Das Gegenteil ist die asynchrone Kommunikation, hier sendet der Sender ohne auf eine Antwort zu warten. Diese Art der Kommunikation findet man beim Versand von E-Mails oder Briefen.

Zuerst muss sich die Leitstelle am Leitstellenhub authentifizieren. D.h. es muss dazu beim Betreiber des Leitstellenhubs ein organisatorischer Prozess etabliert werden, in dem der Leitstelle ein Login zzgl. einem Passwort und ein privater Schlüssel für die Verschlüsselung der zu übertragenden Daten übermittelt wird. D.h. jede Leitstelle, die an diesem Datenaustausch teilnehmen will, muss sich im Vorfeld registrieren und authentifizieren.

Danach erhält die Leitstelle die entsprechenden technischen Voraussetzungen, um an der Kommunikation teilzunehmen.

Diese Informationen (inkl. der privaten Schlüssel) werden ausschließlich im LSH-Proxy hinterlegt. Der LSH-Proxy befindet sich in der DMZ der jeweiligen Leitstelle. Er bildet das sichere Bindeglied zwischen Einsatzleitsystem und Leitstellenhub. D.h. das ELS der Leitstelle hat keine direkte physische Verbindung zum Leitstellenhub, sondern kommuniziert ausschließlich mit dem LSH-Proxy. Der LSH-Proxy wiederum baut eine verschlüsselte Verbindung (TLS Tunnel) zum Leitstellenhub (LSH) auf.

Der Leitstellenhub erfüllt lediglich eine reine Vermittlungsrolle und hat selbst keinen Zugriff auf die Informationen, welche die Leitstellen untereinander austauschen. Auf dem Leitstellenhub werden keine privaten Schlüssel gespeichert!

Sobald der Disponent der sendenden Leitstelle einen Datensatz an eine andere Leitstelle versenden will, erfolgt die Übermittlung des zu sendenden Datensatzes aus dem Einsatzleitsystem an den LSH-Proxy. Der LSH-Proxy fragt beim LSH nach dem öffentlichen Schlüssel der Empfangsleitstelle. Dieser öffentliche Schlüssel wird dem sendenden LSH-Proxy mitgeteilt und er kann mit diesem Schlüssel den Datensatz verschlüsseln, mit seinem privaten Schlüssel signieren und danach an den LSH senden. Der LSH ermittelt aus dem Datensatz die Adresse der Empfangsleitstelle und sendet den verschlüsselten und signierten Datensatz an den LSH-Proxy der Empfangsleitstelle. Der LSH-Proxy der Empfangsleitstelle kann mit seinem privaten Schlüssel den Datensatz entschlüsseln, mit dem öffentlichen Schlüssel die Echtheit des Datensatzes prüfen und nach erfolgreicher Verarbeitung (Entschlüsselung und Prüfung) dem Einsatzleitsystem zur Verfügung stellen.

Während der Übertragung des Datensatzes erfolgt ein Kommunikationsaustausch mit einem automatisch technischen ACKnowledgement (ACK) und einem manuellen ACK durch den Disponenten oder automatisiert durch das Einsatzleitsystem der empfangenden Leitstelle mit Beginn der Einsatzbearbeitung in Form der Einsatzübernahme. Es werden dazu standardisierte HTTP-Statuscodes verwendet.

D.h. die Sendeleitstelle bekommt mindestens 2 Bestätigungen:

1. 200 OK: der Verbindungsaufbau war erfolgreich, der Datensatz ist beim Empfänger angekommen
2. 200 OK: der Disponent in der Empfangsleitstelle hat den Einsatz angenommen (manuelle Bestätigung erforderlich)

Sobald beide Bestätigungen angekommen sind, wird der Einsatz im ELS der Sendeleitstelle entsprechend als erfolgreich weitergeleitet dokumentiert. In allen anderen Fällen muss der Einsatz als offen bzw. nicht bearbeitet usw. angezeigt werden.

Folgende Fehlermeldungen werden der Sendeleitstelle angezeigt:

1. 400 Bad Request: Zielsystem erreichbar, aber gibt Fehlercode zurück
z.B. nicht berechtigt, falsches Zertifikat, internes KV-CC Problem, usw.
2. 408 Request Timeout: Zielsystem nicht erreichbar
z.B. Timeout-Fehler, falscher DNS-Eintrag, usw.
3. 408 Request Timeout: Zielsystem antwortet nicht „in time“ oder ist nicht erreichbar
Antwort kommt zwar, aber zu spät – gilt als nicht erfolgreich übermittelt

8 Organisatorische Rahmenbedingungen

Für den vor beschriebenen Kommunikationsablauf müssen neben den technischen Voraussetzungen auch die notwendigen Prozesse, Strukturen und Verantwortlichkeiten benannt werden.

Dazu ist eine zentrale Institution / Dienststelle zu benennen, die neben dem technisch fachlichen Betrieb auch die notwendigen Prozesse, Strukturen und Verantwortlichkeiten festlegt.

Diese beauftragte Dienststelle veranlasst, koordiniert und verantwortet die weiterführende Feinspezifikation dieser Schnittstelle und die Festlegung der inhaltlichen Beschreibung des über diese Schnittstelle geführten Datenaustausches.

Um den Anforderungen der DSGVO, insbesondere in Bezug auf Sozialdaten gerecht zu werden, muss ein datenschutzrechtliches Konzept unter Berücksichtigung der Länderregelung durch diese Dienststelle erarbeitet werden.

9 Anlagen

Anlage 001 Aufbau der Datensätze in UML

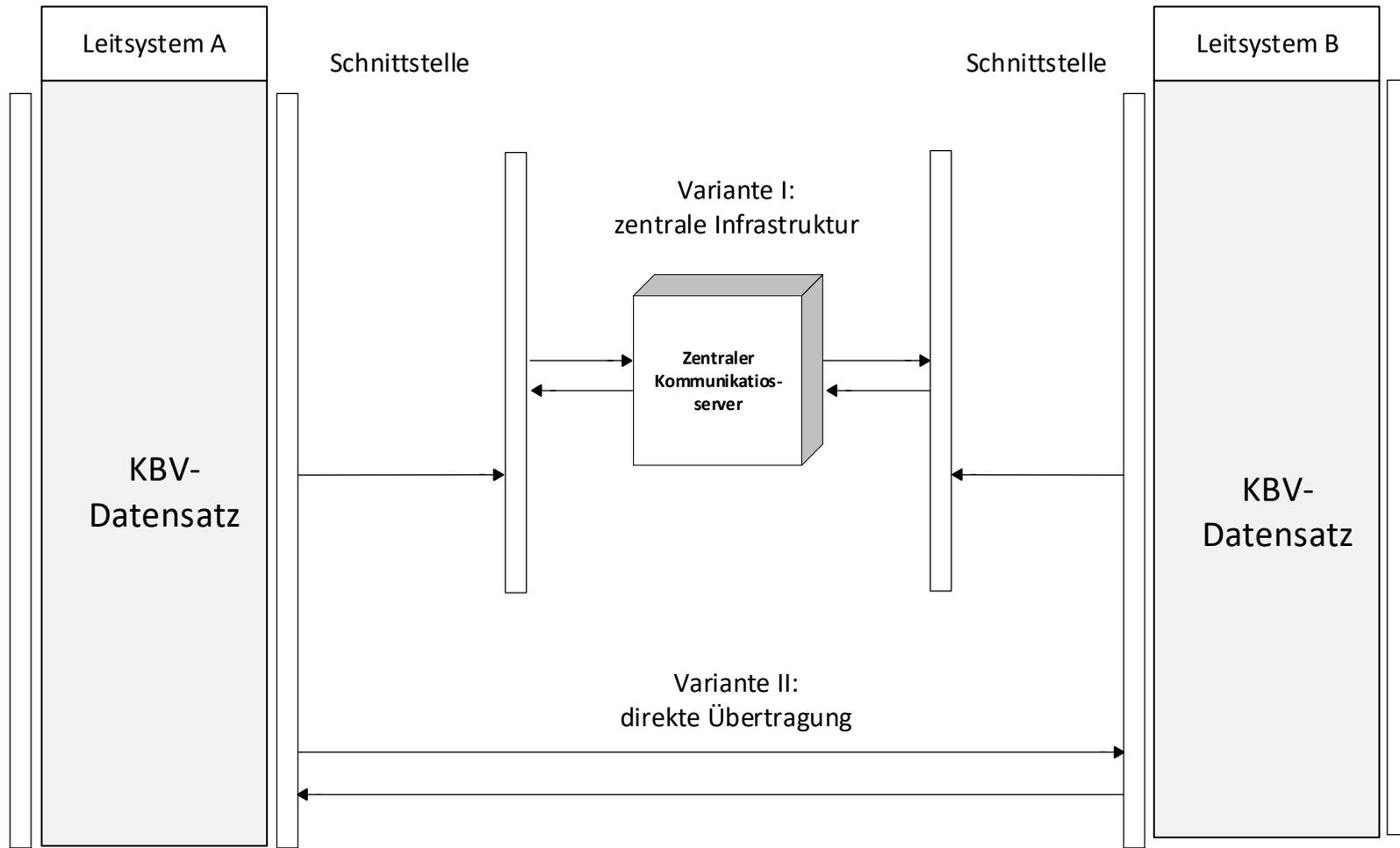
Anlage 002 Kommunikationsarchitektur

Anlage 003 Verschlüsselung

Anlage 004 Kommunikationsablauf

Anlage 005 Schnittstellenspezifikation UCRI V1.0

Objektübertragung zwischen den Leistellen



Proprietäres
Datenformat

JSON-Format

Proprietäres
Datenformat

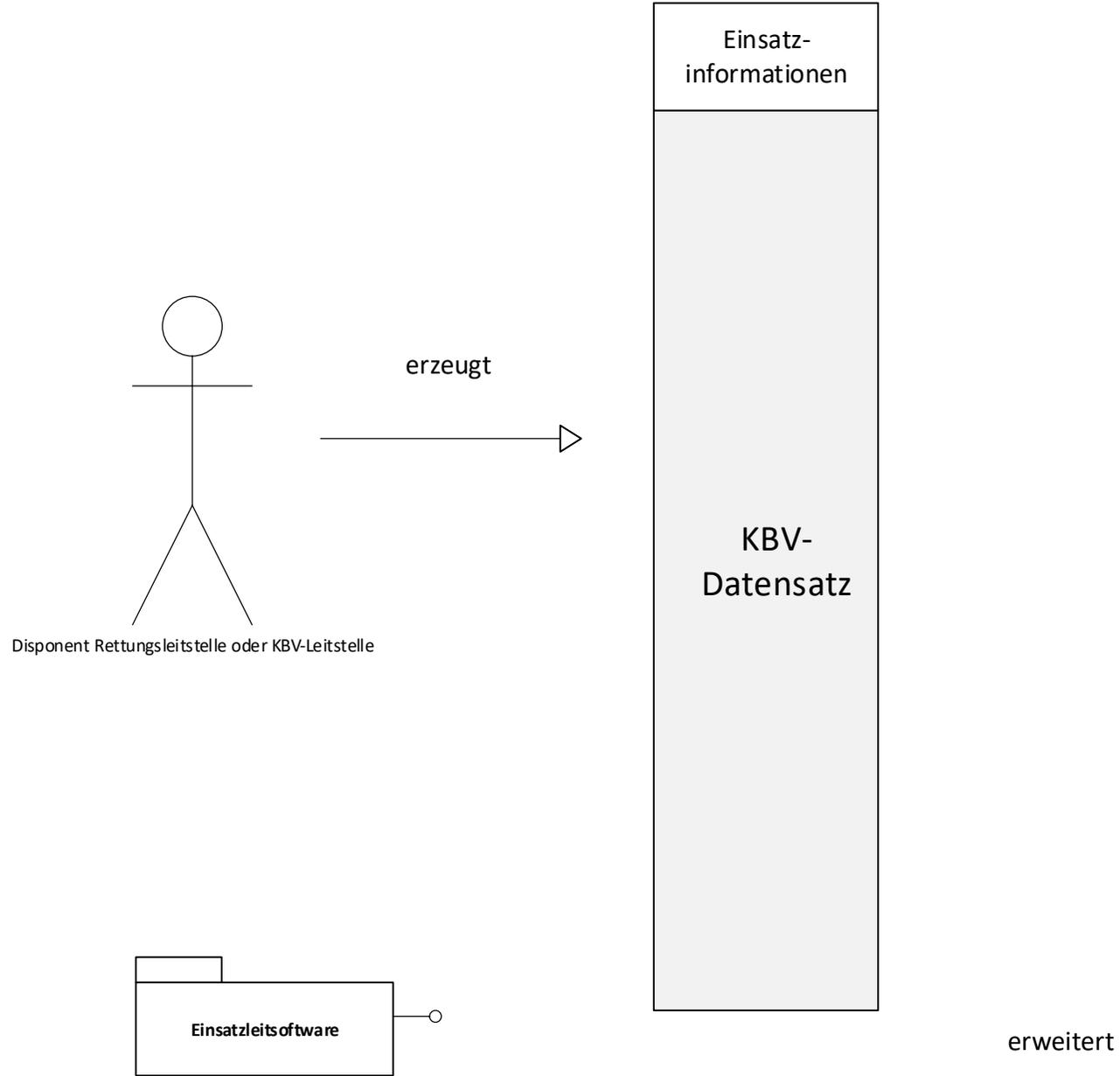
	Datum	Name
Bearbeitet	10.02.2023	Henry Lakatos
Geprüft		
Datename	1908EW_2001_UML-Diagramme_und_Objektzerlegung_nur_Einsatzinformation_v1-11.wdx	
Seite	4	

KBV-Schnittstelle
UML-Diagramm



D.I.E. PROJEKT GMBH Buchenstraße 12 01097 Dresden
Tel. +49 351 4 7936-0 Fax +49 351 4 7936-299
info@diep-ingenieur.de www.diep-ingenieur.de

Bereitstellung Einsatzinformationen



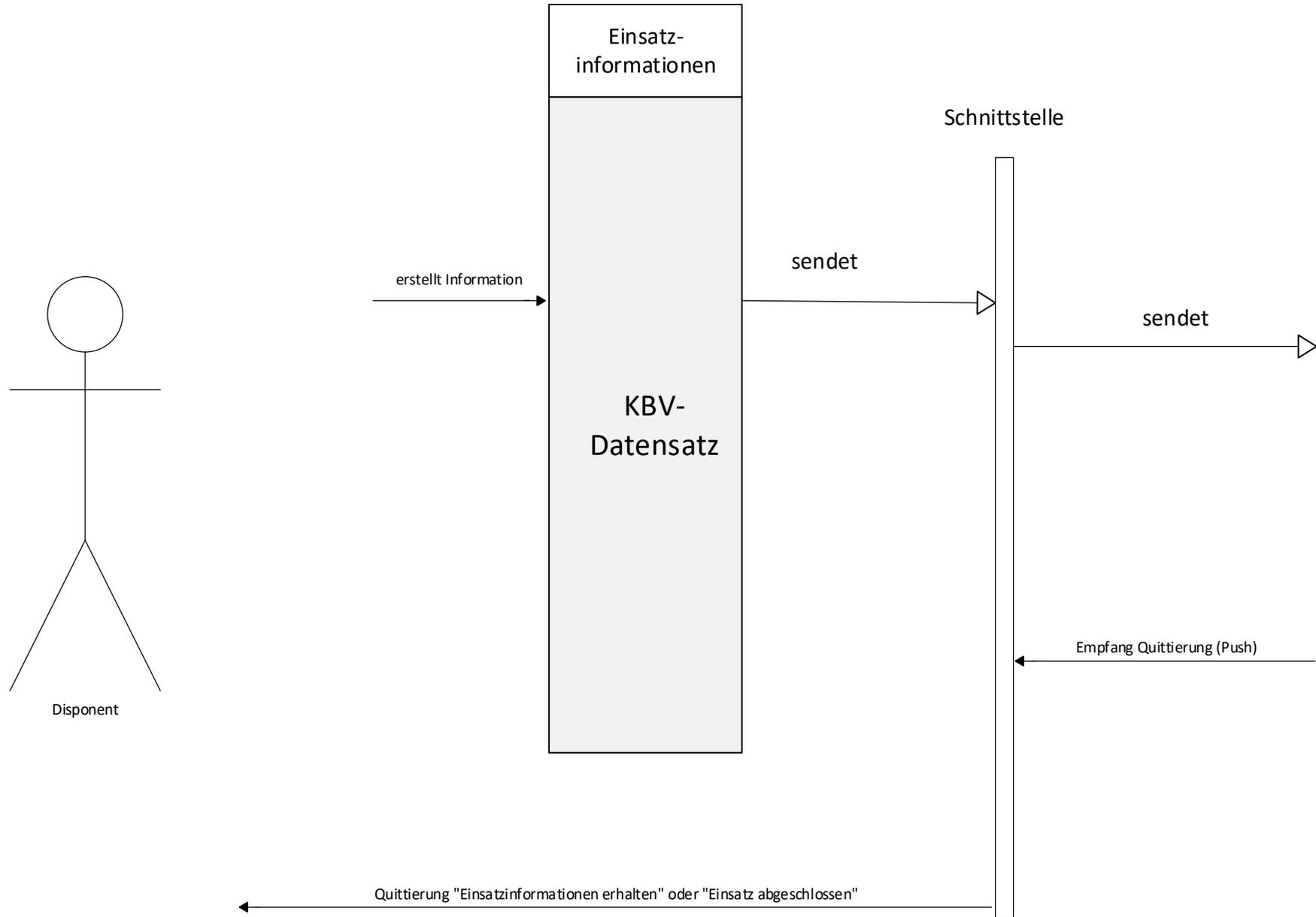
	Datum	Name
Bearbeitet	10.02.2023	Henry Lakatos
Geprüft		
Datename	190BEW_2001_UML-Diagramme_urd_Objektzeugung_nur_Einsatzinformation_v1-11.wxd	
Seite	4	

KBV-Schnittstelle
UML-Diagramm



D.I.E. PROJEKT GMBH Buchenstraße 12 01097 Dresden
Tel. +49 351 4 7936-0 Fax +49 351 4 7936-299
info@diep-projekt.de www.diep-projekt.de

Quittierung Einsatzinformationen PUSH



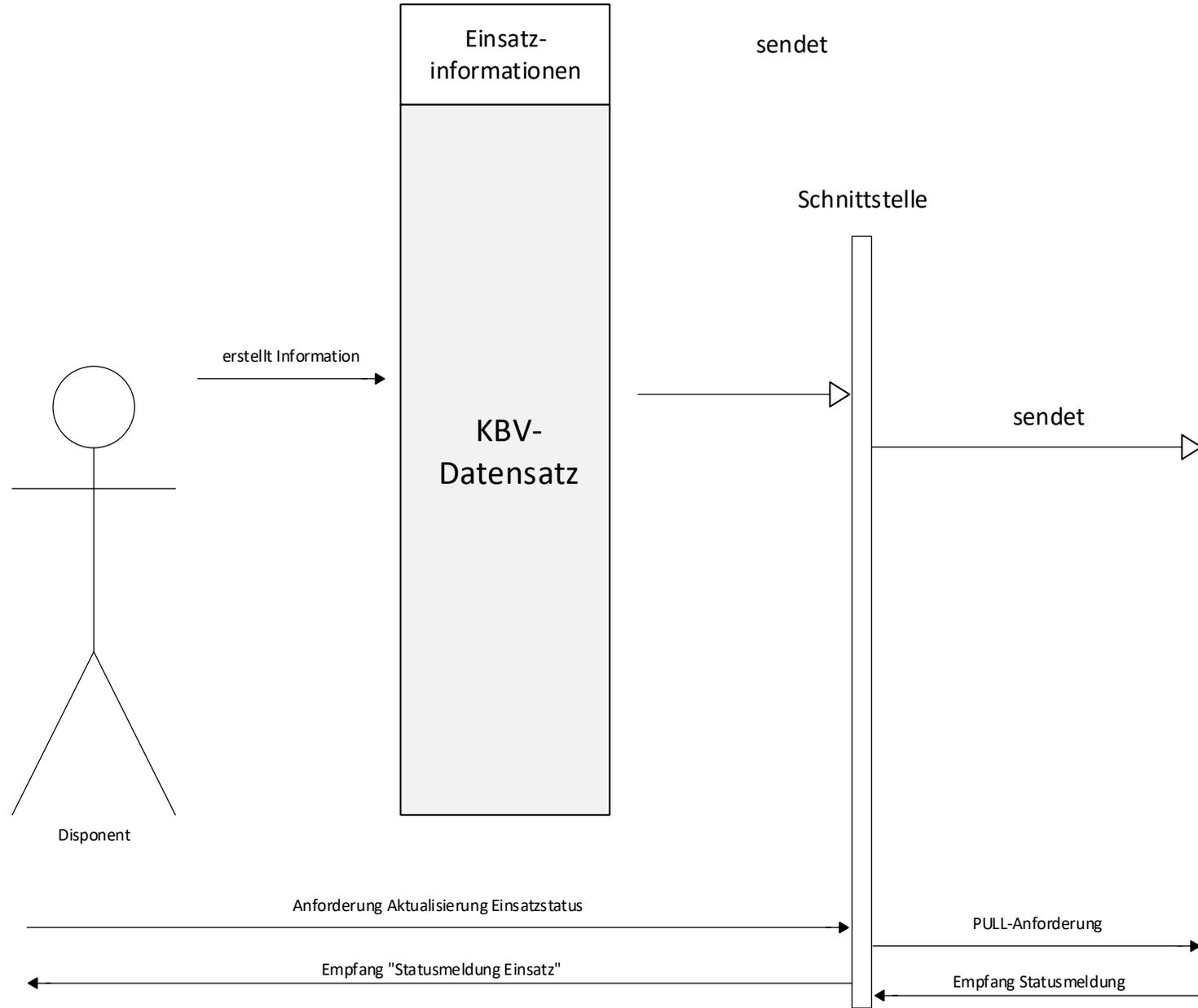
	Datum	Name
Bearbeitet	10.02.2023	Henry Lakatos
Geprüft		
Datiname	190BEV_2001_UML-Diagramme_urd_Objektzeugung_nur_Einsatzinformaton_v1-11.wdx	
Seite	4	

KBV-Schnittstelle
UML-Diagramm



D.I.E. PROJEKT GMBH Buchenstraße 12 01097 Dresden
Tel. +49 351 4 7936-0 Fax +49 351 4 7936-299
info@diep-projekt.de www.diep-projekt.de

Aktualisierung Statusmeldung Einsatz



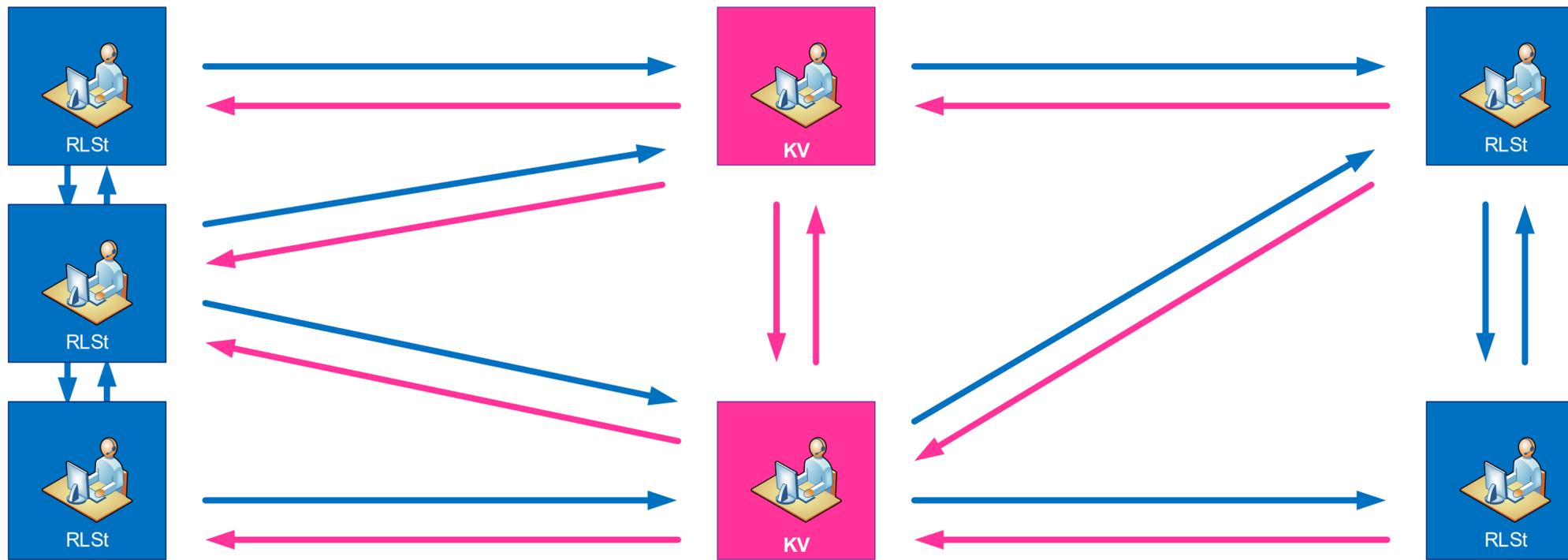
	Datum	Name
Bearbeitet	10.02.2023	Henry Lakatos
Geprüft		
Datiname	190BEW_2001_UML-Diagramme_urd_Objektzeugung_nur_Einsatzinformation_v1-11.vsd	
Seite	4	

KBV-Schnittstelle
UML-Diagramm

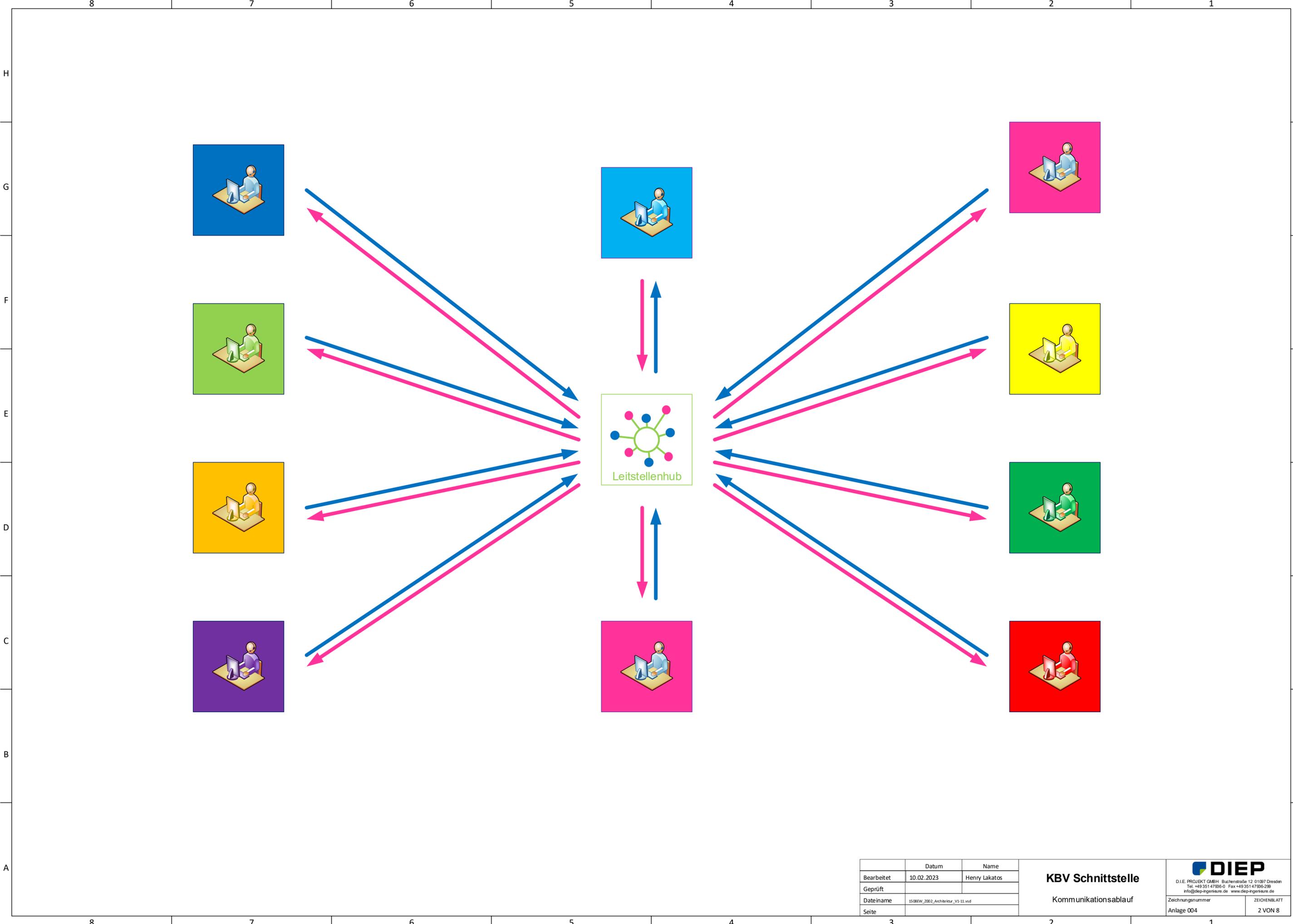
Point to Point Kommunikation



Point to Multipoint Kommunikation



	Datum	Name			
Bearbeitet	10.02.2023	Henry Lakatos	KBV Schnittstelle Kommunikationsablauf		
Geprüft					
Dateiname	150REW_2002_Architektur_V3-11.vsd				
Seite			D.I.E. PROJEKT GmbH Buchenstraße 12 01057 Dresden Tel. +49 351 47896-0 Fax +49 351 47896-299 info@diep-engineure.de www.diep-engineure.de	Zeichnungsnummer Anlage 004	ZEICHENBLATT 1 VON 8



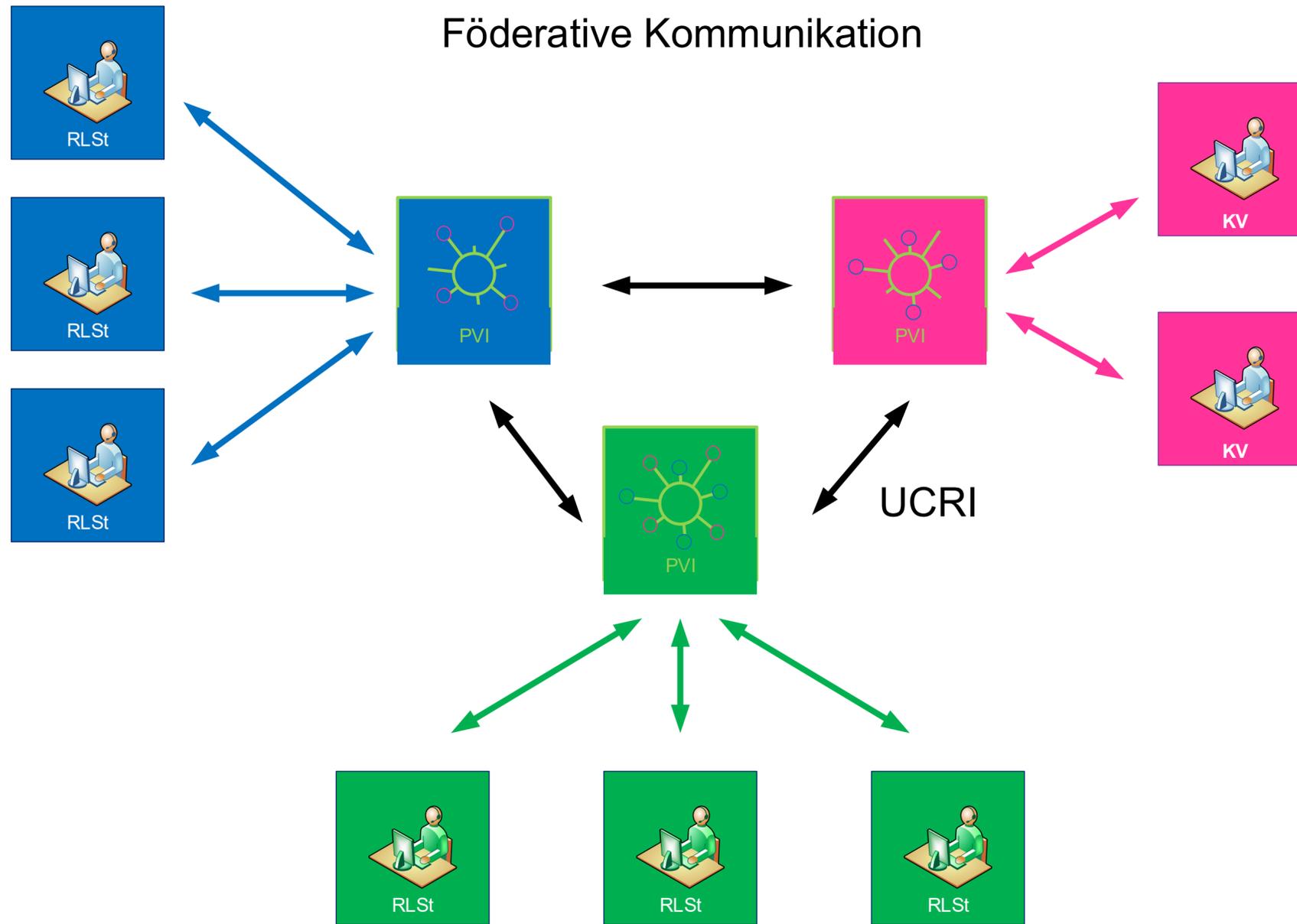
	Datum	Name
Bearbeitet	10.02.2023	Henry Lakatos
Geprüft		
Dateiname	1508EW_2002_Architektur_V3-11.vsd	
Seite		

KBV Schnittstelle
 Kommunikationsablauf

DIEP
 D.I.E. PROJEKT GMBH Buchenstraße 12 01097 Dresden
 Tel. +49 351 47896-0 Fax +49 351 47896-299
 info@diep-ingenieure.de www.diep-ingenieure.de

Zeichnungsnummer	ZEICHENBLATT
Anlage 004	2 VON 8

Föderative Kommunikation



	Datum	Name
Bearbeitet	10.02.2023	Henry Lakatos
Geprüft		
Dateiname	1508EW_2002_Architektur_V3-11.vsd	
Seite		

KBV Schnittstelle

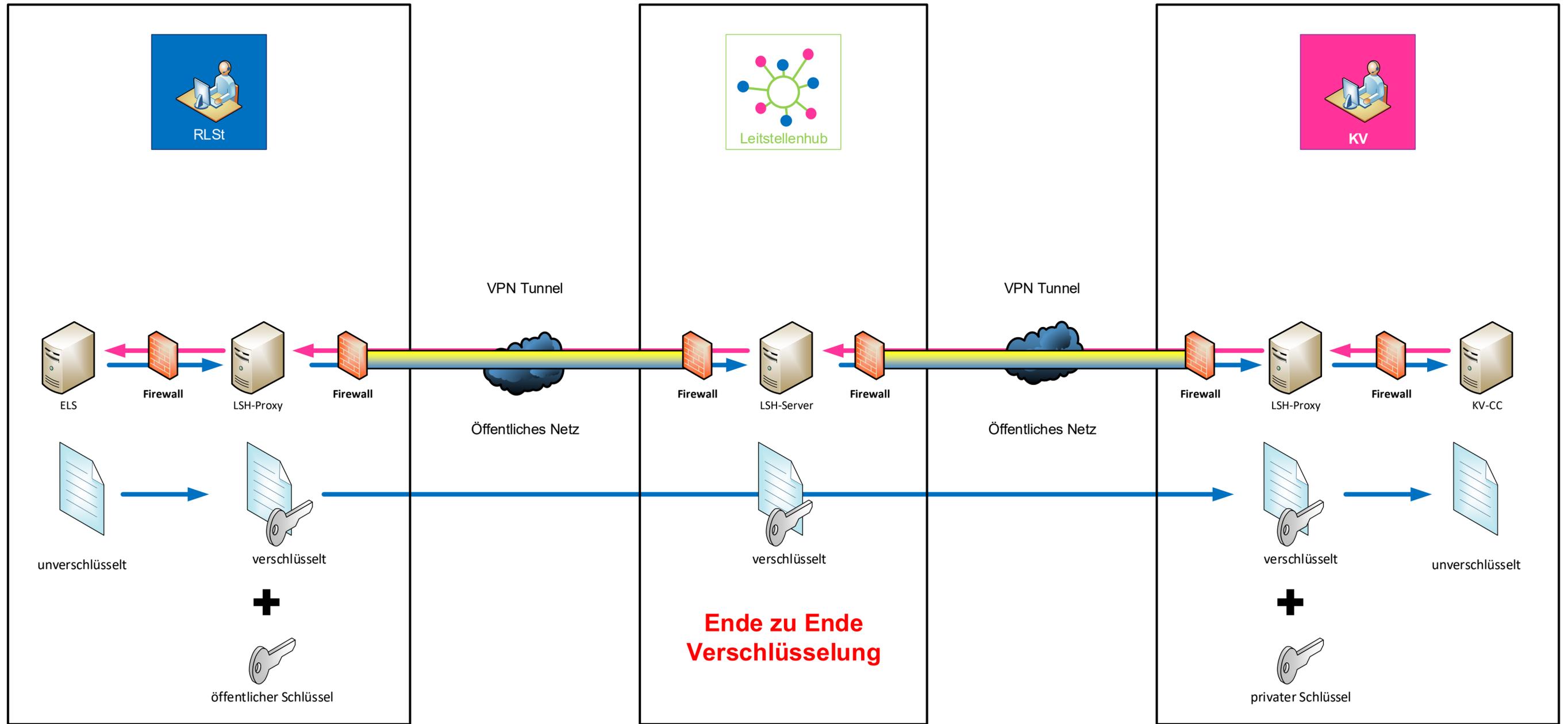
Kommunikationsablauf



D.I.E. PROJEKT GMBH Buchenstraße 12 01097 Dresden
Tel. +49 351 47896-0 Fax +49 351 47896-299
info@diep-engineering.de www.diep-engineering.de

Zeichnungsnummer	ZEICHENBLATT
Anlage 004	4 VON 8

Architektur der IT-/Kommunikations- und Medientechnik

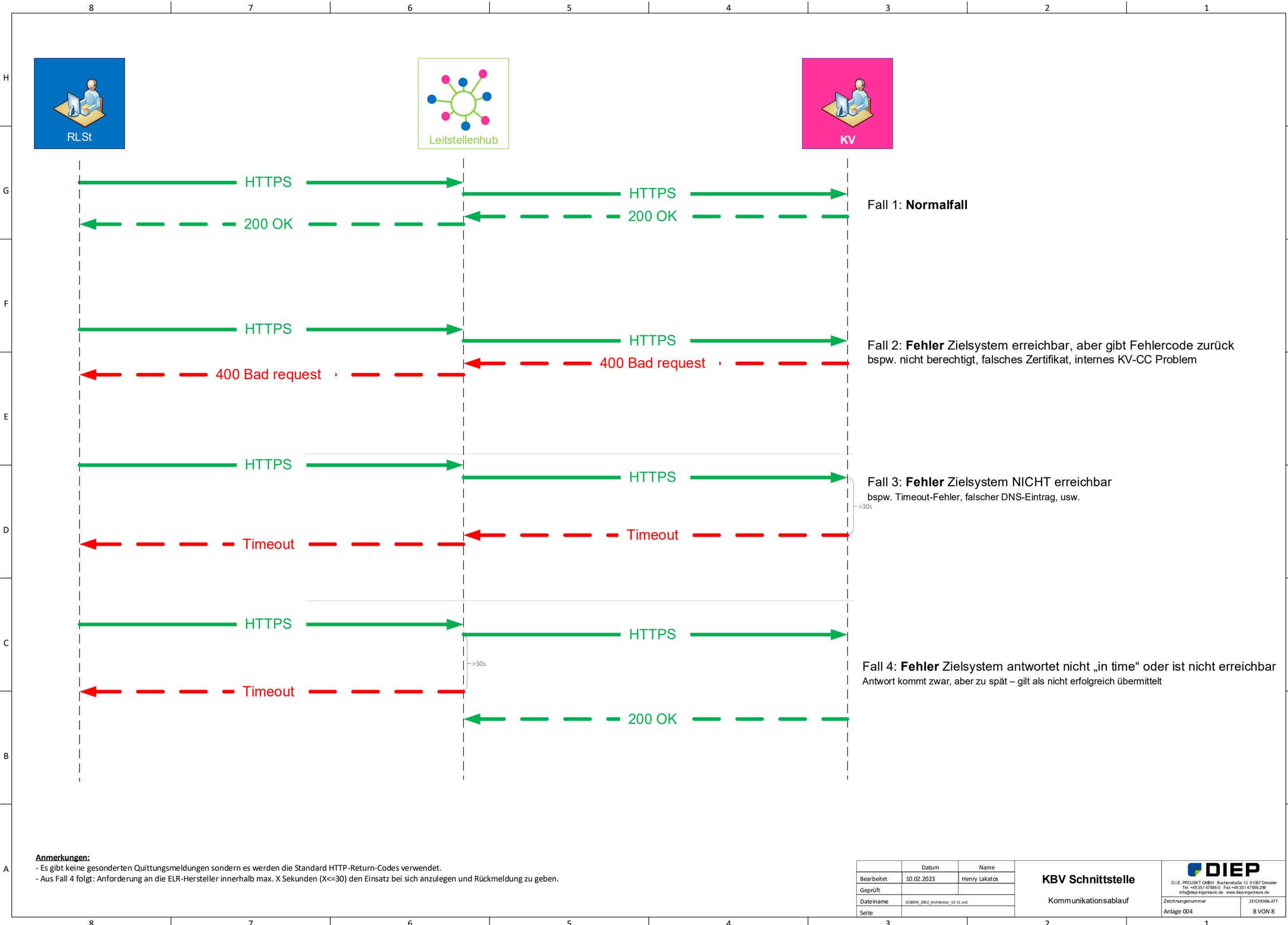


	Datum	Name
Bearbeitet	10.02.2023	Henry Lakatos
Geprüft		
Dateiname	1508EW_2002_Architektur_V3-11.vsd	
Seite		

KBV Schnittstelle
Kommunikationsablauf

DIEP
D.I.E. PROJEKT GMBH Buchenstraße 12 01097 Dresden
Tel. +49 351 47896-0 Fax +49 351 47896-299
info@diep-engineure.de www.diep-engineure.de

Zeichnungsnummer: Anlage 004
ZEICHENBLATT: 7 VON 8



Fall 1: **Normalfall**

Fall 2: **Fehler** Zielsystem erreichbar, aber gibt Fehlercode zurück
bspw. nicht berechtigt, falsches Zertifikat, internes KV-CC Problem

Fall 3: **Fehler** Zielsystem NICHT erreichbar
bspw. Timeout-Fehler, falscher DNS-Eintrag, usw.

Fall 4: **Fehler** Zielsystem antwortet nicht „in time“ oder ist nicht erreichbar
Antwort kommt zwar, aber zu spät – gilt als nicht erfolgreich übermittelt

Anmerkungen:
 - Es gibt keine gesonderten Quittungsmeldungen sondern es werden die Standard HTTP-Return-Codes verwendet.
 - Aus Fall 4 folgt: Anforderung an die ELR-Hersteller innerhalb max. X Sekunden ($X \leq 30$) den Einsatz bei sich anzulegen und Rückmeldung zu geben.

	Datum	Name
Bearbeitet	10.02.2023	Henry Lakatos
Geprüft		
Dateiname	1508EW_2002_Architektur_V3-11.vsd	
Seite		

KBV Schnittstelle

Kommunikationsablauf

DIEP
 D.I.E. PROJEKT GMBH Buchenstraße 12 01097 Dresden
 Tel. +49 351 47896-0 Fax +49 351 47896-299
 info@diep-ingenaure.de www.diep-ingenaure.de

Zeichnungsnummer Anlage 004	ZEICHENBLATT 8 VON 8
--------------------------------	-------------------------

Universal Control Room Interface (UCRI)

Schnittstellenspezifikation Leitstelle- Leitstelle

Version 1.0

Veröffentlichung des Expertenforums universelle
Leitstellenschnittstelle (EFUL)

17.06.2020

Inhalt

1	Einleitung	4
1.1.	Zweck des Dokumentes.....	4
1.2.	Voraussetzungen zum Einsatztausch	4
1.3.	Abkürzungserläuterungen.....	4
2	Architektur Vorschlag	5
2.1.	Systemaufbau Übersicht.....	5
2.2.	UCRI Web-Service integrale Anforderungen	5
2.2.1.	Stichwort Mapping	5
2.2.2.	Einsatzmitteltyp Mapping	6
2.2.3.	Datenkontrolle.....	7
2.3.	Ablauf	7
2.3.1.	Einsatzübergabe	7
2.3.2.	Einsatzmittelanforderung	8
2.3.3.	Kooperativer Einsatz	9
2.4.	Verwendete HTTP-Request-Attribute	9
2.4.1.	Authorization.....	9
2.4.2.	ETag	9
2.4.3.	If-None-Match	10
3	Sicherheit	11
3.1.	Verfügbarkeit der Daten	11
3.2.	Transport- und Applikationssicherheit	11
3.3.	IdentityManager	11
3.4.	Sicherheitsspezifikationen.....	11
4	Data-Transfer-Objects (DTOs/Datentypen)	13
4.1.	Custom Hypermedia Format	13
4.1.1.	application/vnd.simplemedia+json	13
4.1.2.	Links	15
4.2.	Besondere Daten.....	16
4.2.1.	Datum und Zeit	16
4.2.2.	Geographische Koordinaten.....	16
4.2.3.	Geschlecht	17
4.2.4.	Zusatzdaten (Appendices).....	17
4.3.	Home-DTO	22

4.4.	Version-DTO	23
4.5.	JSON Web Token	23
4.6.	NotAuthorized-DTO	24
4.7.	Incident-DTO	24
4.8.	Message-DTO	30
4.9.	Informer-DTO	30
4.10.	Resource-DTO	31
4.11.	Patient-DTO	33
4.12.	Appendix-DTO	35
4.13.	ResourceRequest-DTO	36
4.14.	Confirmation-DTO	37
4.15.	PositiveConfirmation-DTO	37
4.16.	NegativeConfirmation-DTO	38
4.17.	ResourceTurnOutConfirmation-DTO	38
4.18.	TemporaryPositiveConfirmation-DTO	39
4.19.	Classification-DTO	39
4.20.	ResourceType-DTO	39
5	API-Requests	41
5.1.	Home	41
5.2.	Authorize	41
5.3.	Einsatzdaten entgegennehmen	41
5.4.	Nachricht zum Einsatz an korrespondierende Leitstelle	42
5.5.	Quittierung Einsatzübergabe	42
5.5.1.	Übertragung des Einsatzes an remote ELS quittieren	42
5.5.2.	Einsatzübergabe quittieren	43
5.6.	Quittierung Einsatzmittelanforderung	43
5.6.1.	Positive Quittierung	43
5.6.2.	Negative Quittierung	44
5.6.3.	Vorläufig positive Quittierung	44
5.6.4.	Quittierung Fahrzeug ausgerückt	45
5.7.	Stichworte abfragen	45
5.8.	Einsatzmitteltypen abfragen	45
6	Status Codes	47

1 Einleitung

1.1. Zweck des Dokumentes

Das Dokument beschreibt den Einsatzdatenaustausch zwischen zwei oder mehreren Einsatzleitsystemen (*im weiteren Verlauf ELS*) über das **Einsatzdatenaustausch-API** (im Weiteren API).

Das API bietet eine sichere Möglichkeit Einsatzdaten zwischen Leitstellen auszutauschen.

1.2. Voraussetzungen zum Einsatz austausch

Betreiber der Schnittstelle, die über das API Einsatzdaten austauschen, müssen untereinander die *Client-ID (Identifikation am Fremdsystem)*, das zugehörige *Client-Secret (Secret zur Autorisierung am Fremdsystem)* und die *Base-Url der Schnittstelle (Url des Einsatzdatenaustausch-API)* austauschen. Die *Client-ID*, *Base-Url* und das *Client-Secret* sind hierbei auf organisatorischem Wege auszutauschen. Die *Base-Urls* müssen, wenn sich beide Schnittstellen in unterschiedlichen Netzwerken befinden, öffentlich bzw. jeweils für die andere Schnittstellen-Instanz per HTTP/HTTPS erreichbar sein. Es wird empfohlen zur Kommunikation das HTTPS-Protokoll zu verwenden.

1.3. Abkürzungserläuterungen

Abkürzung	Beschreibung
API	Application Programming Interface (Wikipedia)
ELS	Einsatzleitsystem
STS	Security-Token-Service (Wikipedia)
JSON	JavaScript Object Notation (Wikipedia)
JWT	JSON Web Token (Wikipedia)
REST	Representational State Transfer (Wikipedia)
DTO	Data Transfer Object (Wikipedia)
UUID	Universally Unique Identifier (Wikipedia)
Etag	HTTP Entity Tag (Wikipedia)
URL	Uniform Resource Locator (Wikipedia)

2 Architektur Vorschlag

2.1. Systemaufbau Übersicht

Es wird empfohlen die Implementierung der Schnittstelle in einem eigenständigen Prozess (UCRI Web-Service) zu realisieren. Dadurch ist es möglich die Schnittstelle in einer demilitarisierten Zone (DMZ) außerhalb des sensiblen Leitstellen-Netzwerks zu betreiben. Somit kann auch sichergestellt werden das, nur ausgehende Anfragen vom Leitstellen-Netz erlaubt werden. In folgender Grafik wird ein solcher Aufbau dargestellt:

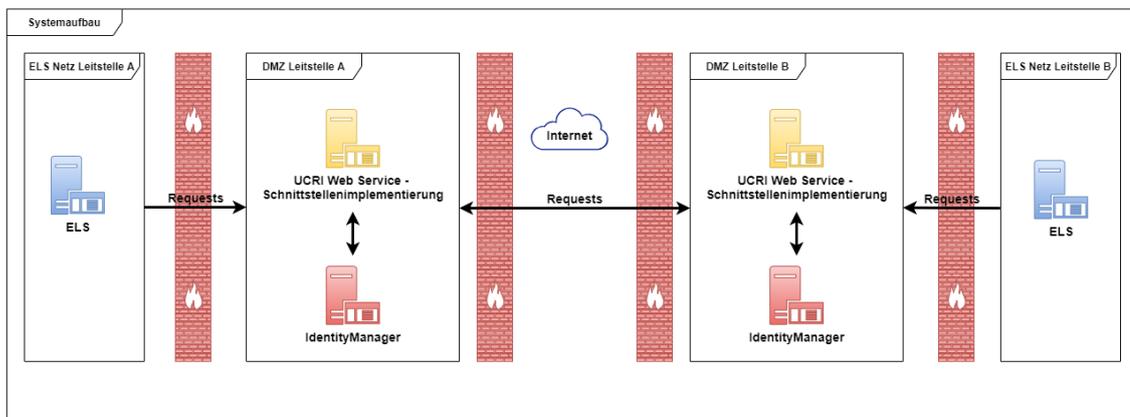


Abbildung 1: Systemaufbau

2.2. UCRI Web-Service integrale Anforderungen

2.2.1. Stichwort Mapping

Da bundesweit kein einheitlicher Stichwortkatalog definiert ist, ist für einen möglichst reibungslosen Einsatzdatenaustausch ein Mapping der Stichworte essentiell.

Aus diesem Grund werden folgende Regeln definiert:

1. Die aussendende Leitstelle überträgt immer das in ihrem System definierte Stichwort.
2. Die empfangende Leitstelle mapped das fremde Stichwort auf ein in ihrem System definiertes Stichwort.

Das konkrete Mapping einzelner Stichwörter kann nicht automatisiert, sondern nur manuell im Rahmen der Stammdatenpflege durch einen (autorisierten) Mitarbeiter jeder beteiligten Leitstelle erfolgen.

Um das Stichwort Mapping zu optimieren, definiert die UCRI Web-Service-Schnittstelle eine Methode zur Abfrage des gesamten Stichwortkatalogs einer Partnerleitstelle. Hierbei muss ein Stichwort mit Kürzel und Beschreibung (Langtext) übergeben werden (siehe Classification-DTO). Somit können typische Übertragungsfehler beim Austausch der

Stichwortkatalog außerhalb des Systems (z.B. eMail- oder FAX-Kommunikation) vermieden werden und für den autorisierten Mitarbeiter innerhalb des Web-Service (UI-gestützte) Mapping-Funktionalitäten realisiert werden.

Da Stichwortkataloge mit der Zeit geändert/aktualisiert werden, kann nicht garantiert werden, dass zu jedem Zeitpunkt für jedes Stichwort ein Mapping vorhanden ist. Daher sollen nicht gemappte Stichwörter grundsätzlich an das empfangende Einsatzleitsystem durchgereicht werden. Der Einsatz soll im empfangenden System mit dem Hinweis zur Anzeige gebracht werden, dass das Stichwort dem System unbekannt ist, damit der Disponent z. B. ein passendes vorhandenes Stichwort auswählt oder die Klärung einleitet. Einsätze dürfen auf keinen Fall wegen unbekannter Stichworte unberücksichtigt bleiben.

2.2.2. Einsatzmitteltyp Mapping

Ebenso wie bei den Stichworten, sind Einsatzmitteltypen nicht einheitlich definiert. Aus diesem Grund besteht auch hier die Anforderung für ein Mapping.

Grundsätzlich werden folgende Regeln definiert:

1. Die aussendende Leitstelle überträgt immer den in ihrem System definierten Einsatzmitteltyp.
2. Die empfangende Leitstelle mapped den fremden Einsatzmitteltyp auf einen in ihrem System definierten Einsatzmitteltyp.

Das konkrete Mapping einzelner Einsatzmitteltypen kann nicht automatisiert, sondern nur manuell im Rahmen der Stammdatenpflege durch einen autorisierten Mitarbeiter jeder beteiligten Leitstelle erfolgen.

Um das Einsatzmitteltyp Mapping zu optimieren, definiert die UCRI Web-Service-Schnittstelle eine Methode zur Abfrage aller definierten Einsatzmitteltypen einer Partnerleitstelle. Hierbei muss ein Einsatzmitteltyp mit Kürzel und Beschreibung (Langtext) übergeben werden (siehe ResourceType-DTO). Somit können typische Übertragungsfehler beim Austausch der Einsatzmitteltypen außerhalb des Systems (z.B. eMail- oder FAX-Kommunikation) vermieden werden und für den autorisierten Mitarbeiter innerhalb des Web-Service (UI-gestützte) Mapping-Funktionalitäten realisiert werden.

Auch wenn Einsatzmitteltypen mit der Zeit nicht so häufig wie Stichworte geändert werden, kann nicht garantiert werden, dass zu jedem Zeitpunkt für jeden Einsatzmitteltyp ein Mapping vorhanden ist. Daher sollen nicht gemappte Einsatzmitteltypen grundsätzlich an das empfangende Einsatzleitsystem durchgereicht werden. Der Einsatz soll im empfangenden System mit dem Hinweis zur Anzeige gebracht werden, dass ein unbekannter Einsatzmitteltyp angefordert wird, damit der Disponent nach

Klärung den richtigen Einsatzmitteltyp zuordnen kann. Einsätze dürfen auf keinen Fall wegen unbekanntem Einsatzmitteltyp unberücksichtigt bleiben.

2.2.3. Datenkontrolle

Per Definition unterstützt die UCRI Web-Service-Schnittstelle grundsätzlich immer den definierten Gesamtdatensatz ohne jegliche Einschränkung. Auf Grund der unterschiedlichen fachlichen Ausrichtungen der potentiellen Leitstellenpartner (nPol-BOS / Pol-BOS / ÖPNV / EVU), ergibt sich hieraus ein zu lösendes Konfliktpotential bzgl. Datenschutz und Datenhoheit.

Aus diesem Grund ergibt sich die funktionale Anforderung eines Datenkontrollmechanismus als integraler Bestandteil des UCRI Web-Service oder des ELS. Mittels dieses Mechanismus muss eine Leitstelle (UI-gestützte) in die Lage versetzt werden, mittels einer Blacklist für jede Partnerleitstelle separat den Gesamtdatensatz auf einen grundsätzlichen, zur Übertragung freigegeben Teildatensatz zu reduzieren.

2.3. Ablauf

2.3.1. Einsatzübergabe

Die Einsatzdaten und eine Kennung der Empfänger-Leitstelle werden vom ELS an die Schnittstelle übergeben und von dort an die Schnittstelle der Leitstelle B weitergeleitet. Das ELS der Leitstelle A fragt zyklisch den Status der Übertragung und die Übernahme des Einsatzes anhand von Quittierungsmeldungen in Form von REST-Ressourcen von der eigenen Schnittstellen-Implementierung ab.

Die Schnittstellen-Implementierung in Leitstelle A authentifiziert sich beim IdentityManager der Leitstelle B und erhält bei Erfolg ein JSON Web Token (JWT). Das Token muss in jedem Request zwischen den Leitstellen im "Authorize-Header" enthalten sein muss.

Die Einsatzdaten werden von der Schnittstelle in Leitstelle A an die Schnittstelle in Leitstelle B übertragen. Der Status der Übertragung wird dem ELS der sendenden Leitstelle (Leitstelle A) durch eine entsprechende Quittierungsmeldung bereitgestellt. Hierbei handelt es sich um eine rein technische Quittierung (HTTP-Ebene).

Die empfangende Leitstelle (Leitstelle B) quittiert der sendenden Leitstelle (Leitstelle A) den Empfang der Einsatzdaten (technische Quittierung auf Anwendungsebene). Der Disponent der empfangenden Leitstelle teilt der sendenden Leitstelle durch eine weitere Quittierung mit, ob er den Einsatz übernimmt oder ablehnt.

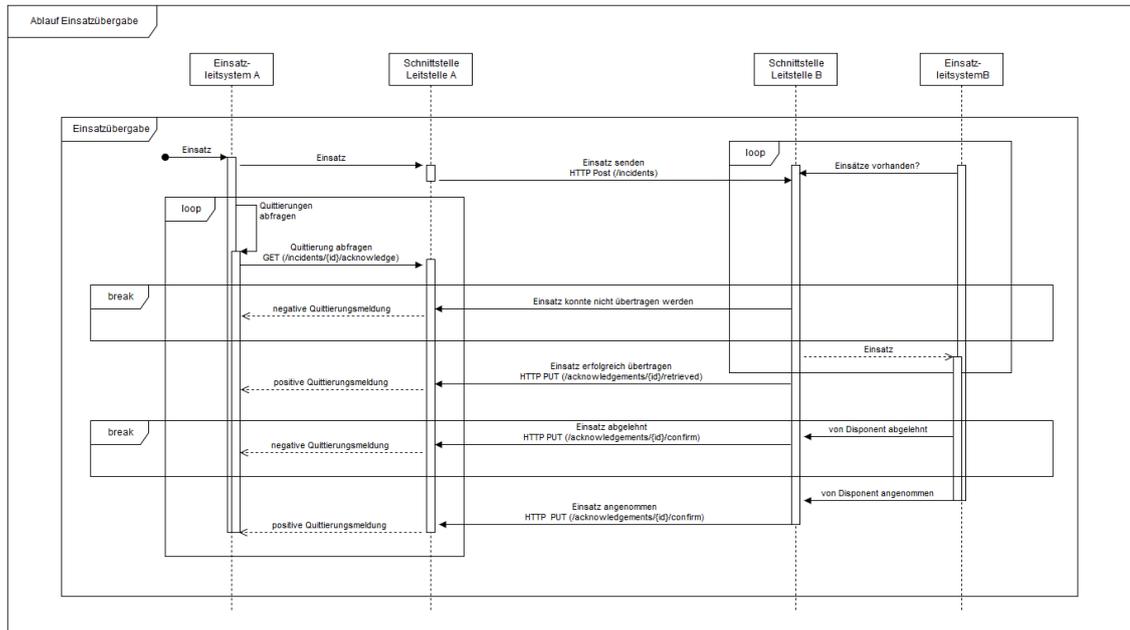


Abbildung 2: Ablauf Einsatzübergabe

2.3.2. Einsatzmittelanforderung

Die Einsatzdaten werden mit Einsatzmittelanforderungen und Kennung der Empfänger-Leitstelle/n vom ELS an die Schnittstelle übergeben und von dort an die Empfänger-Leitstelle/n weitergeleitet. Das ELS der anfordernden Leitstelle fragt zyklisch den Status der Übertragung, die Übernahme des Einsatzes und Quittierungsmeldungen der Einsatzmittel ab.

Die empfangende Leitstelle quittiert der anfordernden Leitstelle jede Anforderung mit entsprechender Nachricht. Kann die Anforderung erfüllt werden, wird positiv quittiert. Die Nachricht enthält den Einsatzmitteltyp und optional den Namen und die ID des Einsatzmittels. Kann die Anforderung nicht erfüllt werden, wird negativ quittiert. Hierbei kann optional ein Grund angegeben werden. Weiter kann eine Anforderung *vorläufig positiv* quittiert werden, wenn Einsatzmittel entsendet werden zum Zeitpunkt der Quittierung aber noch kein konkretes Einsatzmittel genannt werden kann. Es folgt dann zu einem späteren Zeitpunkt eine Quittierung, die der anfordernden Leitstelle Informationen zum ausgerückten Einsatzmittel zu Verfügung stellt.

Auf eine Einsatzmittelanforderung können mehrere positive Quittierungen erfolgen, wenn nicht nur das angeforderte Einsatzmittel entsendet wird. Über Nachrichten zum Einsatz kann der Disponent der anfordernden Leitstelle mitteilen, dass zusätzlich zum angeforderten, weitere Einsatzmittel ausrücken.

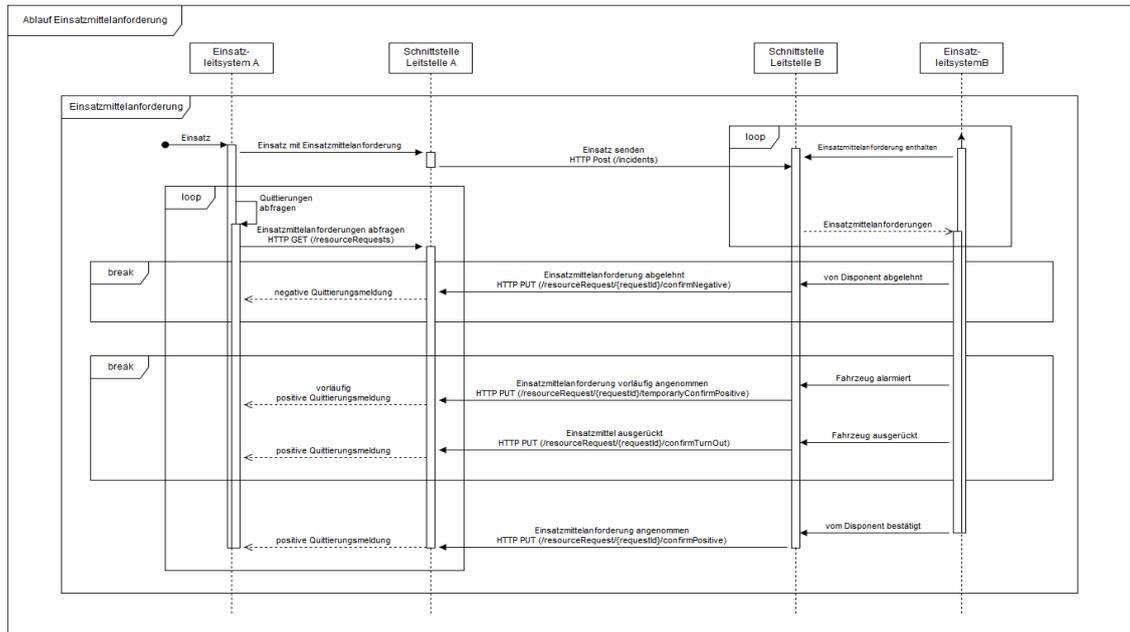


Abbildung 3: Ablauf Einsatzmittelanforderung

2.3.3. Kooperativer Einsatz

Der kooperative Einsatz unterscheidet sich von der eigentlichen Einsatzübergabe dadurch, dass der Einsatz in beiden beteiligten Leitstellen bearbeitet wird. Ein kooperativer Einsatz wird von der sendenden Leitstelle auch nachdem die empfangende Leitstelle die Übergabe positiv quittiert hat nicht beendet. Disponenten beider beteiligten Leitstellen können über Nachrichten zum Einsatz mit der kooperierenden Leitstelle kommunizieren. Das Senden von Nachrichten ist sowohl von der Leitstelle, die den Einsatz übergeben hat, als auch von der empfangenden Leitstelle möglich.

2.4. Verwendete HTTP-Request-Attribute

2.4.1. Authorization

Jeder Request muss im HTTP-Request-Attribut *Authorization* ein von der entfernten Schnittstelle ausgestelltes Token zur Authorization enthalten. Die Schnittstelle beantragt über den Authorize-Endpunkt ein Token unter Angabe der Client-ID und einem Client-Secret.

2.4.2. ETag

ETag (HTTP Entity Tag) wird verwendet um zu verhindern, dass Daten gesendet werden obwohl ein Client diese Daten bereits hat bzw. sich die Daten nicht geändert haben. Das API erstellt hierzu für jede Ressource einen Hashwert. Fragt nun ein Client diese Ressource an, so wird dieser Hashwert als HTTP-Request-Attribut *ETag* der Antwort hinzugefügt. Der Client kann sich nun diesen Wert merken und bei künftigen Anfragen

derselben Ressource diesen Wert als *If-None-Match* (siehe *If-None-Match*) hinzufügen. Hat sich die Ressource nicht geändert wird das API mit *Not Modified (304)* antworten. In diesem Fall sind keine Daten im HTTP-Body enthalten.

2.4.3. If-None-Match

Hat ein Client bereits eine Ressource des API angefragt, so hat er in der Antwort ein HTTP-Request-Attribut ETag (siehe ETag) erhalten. Bei weiteren Anfragen dieser Ressource sollte er nun dieses ETag als HTTP-Request-Attribut If-None-Match hinzufügen. Sollte der Client die Ressource noch nicht angefragt haben oder möchte diese Ressource auf jeden Fall in der Antwort erhalten, so muss er das HTTP-Request-Attribut If-None-Match nicht hinzufügen.

Es bleibt den Client überlassen ob er das *ETag* und das *If-None-Match* HTTP-Header-Attribut benutzen möchte. Es wird jedoch empfohlen, da sich hiermit der Datenverkehr teils drastisch reduzieren kann.

3 Sicherheit

3.1. Verfügbarkeit der Daten

Einsatzdaten sind über die Schnittstelle nur solange abrufbar, wie es für die Verarbeitung erforderlich ist. Dies bedeutet, dass Einsatzdaten nur solange zur Verfügung stehen, bis der Einsatz vom entfernten ELS entweder übernommen oder abgelehnt wurde.

3.2. Transport- und Applikationssicherheit

Zur Transportsicherheit wird ausschließlich eine TLS/SSL Verbindung mit dem HTTPS-Protokoll erlaubt. Die Endpunkte werfen bei jeder Anfrage aus, ob der Client ein gültiges *JWT* besitzt. Ein gültiges Token kann am *Authorize*-Endpunkt (*siehe Authorize*) angefordert werden. Das Token enthält Informationen über den Client oder Benutzer und dessen Rollen, berechnete Scopes, sowie Informationen über die zeitliche Gültigkeit und eine Signatur. Die Signatur gewährleistet, dass das Token nicht manipuliert wurde.

3.3. IdentityManager

Am *Authorize*-Endpunkt kann der Client ein Token für den **Einsatzdatenaustausch**-Scope beim *Security Token Service (STS)* beantragen. Das erhaltene *JWT* enthält ein *Access Token* das bei allen weiteren HTTPS-Requests als HTTP-Header-Attribut *Authorization* hinzugefügt werden muss. Hier muss es als **Bearer** Token eingetragen werden.

Es werden die unter *Sicherheitsspezifikationen* genannten Technologien unterstützt.

Das *JWT* verfügt über ein Property *ExpiresIn*, das im Format Unix-Timestamp vorliegt. Dieser Wert gibt an wie lange das *JWT* gültig ist. Der Client muss sich darum kümmern rechtzeitig, vor Ablauf der Gültigkeit des *JWT*, ein neues *JWT* zu beantragen. Ist das *JWT* abgelaufen wird das API mit *Unauthorized (401)* antworten. Spätestens jetzt muss der Client ein neues *JWT* beantragen, da ab diesem Zeitpunkt keine Daten mehr ausgeliefert werden.

3.4. Sicherheitsspezifikationen

- OpenID Connect Core 1.0 ([spec](#))
 - basic and implicit flow
- OpenID Connect Discovery 1.0 ([spec](#))
- OpenID Connect Session Management 1.0 - draft 22 ([spec](#))
- OpenID Connect HTTP-based Logout 1.0 - draft 03 ([spec](#))
- OAuth 2.0 ([RFC 6749](#))

- implicit, resource owner password credentials and client credentials grant

4 Data-Transfer-Objects (DTOs/Datentypen)

Das API überträgt komplexe Daten als JSON-Objekte. Deshalb werden zur Beschreibung des API die Grunddatentypen von JavaScript verwendet:

- Array
- Bool
- Date
- Number
- Object
- String

4.1. Custom Hypermedia Format

Um die abrufbaren Daten sinnvoll miteinander zu verbinden, verwendet das API Hypermedia-Formate. Das bedeutet, dass neben dem Format *application/json* auch weitere Formate abgerufen werden können.

Der Client kann bestimmen, welches Hypermedia-Format er als Antwort erhalten möchte, indem er im HTTP-Header-Attribut *Response-Content-Type* das entsprechende Format angibt. Zum Beispiel würde das API für *Response-Content-Type: application/json* JSON im Response-Body enthalten.

4.1.1. application/vnd.simplemedia+json

Response-Content-Type: application/vnd.simplemedia+json

Dieses Hypermedia-Format wird vom UCRI-Expertenforum eingeführt um Daten miteinander verknüpfen zu können und dennoch nicht sehr umfangreiche Datenmengen bei jedem Abruf übertragen zu müssen. Es handelt sich auch hierbei um ein *JSON*-Format mit *Links*-Arrays. Das Format ändert sich, wenn nur eine einzelne Ressource abgefragt wird.

application/vnd.simplemedia+json - Beispiel für mehrere Ressourcen

```
1  {
2    "itemcollection": [
3      {
4        "item":
5          {
6            "id": 1,
7            "thumbprint": "xxx"
8          },
9        "links": [
10         {
11           "rel": "self",
12           "href": "https://[host-address]/api/incidents/1/",
13           "label": null,
14           "name": null
15         },
16         {
17           "rel": "resources",
```

```
18         "href": "https://[host-address]/api/incidents/1/resources/",
19         "label": "Ressourcen",
20         "name": null
21     },
22     {
23         "rel": "classifications",
24         "href": "https://[host-address]/api/incidents/1/classifications/",
25         "label": "Stichworte",
26         "name": null
27     }
28 ]
29 },
30 {
31     "item":
32     {
33         "id": 2,
34         "thumbprint": "xxx"
35     },
36     "links": [
37     {
38         "rel": "self",
39         "href": "https://[host-address]/api/incidents/2/",
40         "label": null,
41         "name": null
42     },
43     {
44         "rel": "resources",
45         "href": "https://[host-address]/api/incidents/2/resources/",
46         "label": "Ressourcen",
47         "name": null
48     },
49     {
50         "rel": "classifications",
51         "href": "https://[host-address]/api/incidents/2/classifications/",
52         "label": "Stichworte",
53         "name": null
54     }
55 ]
56 }
57 ],
58 "links": [
59     {
60         "rel": "self",
61         "href": "https://[host-address]/api/incidents",
62         "label": null,
63         "name": null
64     }
65 ]
66 }
```

application/vnd.simplemedia+json - Beispiel für eine Ressource

```

1  {
2    "item":
3    {
4      "id": 1,
5      "thumbprint": "xxx"
6    },
7    "links": [
8      {
9        "rel": "self",
10       "href": "https://[host-address]/api/incidents/1/",
11       "label": null,
12       "name": null
13     },
14     {
15       "rel": "resources",
16       "href": "https://[host-address]/api/incidents/1/resources/",
17       "label": "Ressourcen",
18       "name": null
19     },
20     {
21       "rel": "classifications",
22       "href": "https://[host-address]/api/incidents/1/classifications/",
23       "label": "Stichworte",
24       "name": null
25     }
26   ]
27 }

```

4.1.2. Links

Verwendet ein Hypermedia-Format *Links*-Arrays, so sind diese *Links* immer gleich aufgebaut:

Property	Type	Description
rel	String	<i>relation</i> - beschreibt die Verbindung (z.B.: self: die Ressource selbst, classifications: die Stichworte zur Ressource). Sofern verfügbar wird darauf geachtet, dass registrierte Werte verwendet werden (<i>siehe IANA, RFC 8288</i>)
href	String	<i>hyper reference</i> - die Adresse der Verbindung
label (optional)	String	sollte der Link auf einer UI angezeigt werden, so ist dies der Vorschlag, mit dem er angezeigt wird
name (optional)	String	ein Name der die Ressource beschreibt

Aufbau eines Links

```
1 {  
2   "rel": "self",  
3   "href": "https://[host-address]/api/incidents/1/",  
4   "label": null,  
5   "name": null  
6 }
```

4.2. Besondere Daten

Beim Senden und Empfangen von Daten müssen diese Daten standardisiert übertragen werden können. Damit soll sichergestellt sein, dass beide Seiten der Kommunikation das Gleiche verstehen und Daten nicht missinterpretiert werden. Zum einen erreichen wir das durch das verwenden der Hypermedia Formate, zum anderen durch das Festlegen wie mit besonderen Daten umgegangen werden soll.

4.2.1. Datum und Zeit

Datum und Zeit stellen eine Besonderheit dar, da es nicht nur eine Vielzahl an Darstellungsmöglichkeiten gibt (pro Sprache und/oder Land unterschiedliche Schreibweisen), sondern auch unterschiedliche Zeitzonen. Dadurch ist es notwendig sich auf eine Zeitzone und eine Schreibweise zu einigen. Folgende Regeln gelten für das Senden als auch das Empfangen von Daten:

- Datum und Zeit werden stets im [ISO 8601 Standard](#) übertragen (siehe [International Organization for Standardization, ISO 8601:2004](#))
- Datum und Zeit werden stets in [Coordinated Universal Time \(UTC\)](#) übertragen (siehe [RFC 3339](#))

Beispiel: "timestamp": "\"2018-10-22T13:13:00.000Z\""

Werden Daten gesendet, die nicht diesen Regeln entsprechen, werden sie nicht angenommen.

4.2.2. Geographische Koordinaten

Für den Austausch von geographischen Koordinaten existiert eine Vielfalt an möglichen Formaten, die zudem noch unterschiedlichen Koordinatensystem referenzieren.

Innerhalb dieser Schnittstelle basieren alle ausgetauschten Koordinaten auf dem Referenzsystem [World Geodetic System 1984](#) (WGS84). Die Einzelwerte für Längen- und Breitengrad werden in rein dezimalem Format übertragen.

Beispiel: Breitengrad 53.552886, Längengrad 10.006583 für den Hamburger Hauptbahnhof

4.2.3. Geschlecht

Damit Angaben zum Geschlecht einer Person von den verschiedenen Konsumenten der API einheitlich verarbeitet werden können, sind hierfür nur die Werte "female", "male" und "diverse" zu verwenden. Dies gilt auch dann, wenn Geschlechtsangaben innerhalb der Zusatzdaten (Appendices) eines z.B. Patienten geliefert werden.

4.2.4. Zusatzdaten (Appendices)

Um eine möglichst anpassbare API zu ermöglichen und sowohl Lieferanten wie auch Konsumenten von Daten Anpassungen zu ermöglichen, können Einsatzdaten Ressourcen ein Array von *Appendices* enthalten. Appendices werden überall dort genutzt, wo eine einfache Erweiterung möglich oder auch notwendig ist. Appendices werden durch *Key-Value-Paare* dargestellt. Verfügt eine Ressource über Appendices, so können an sie in einfachster Weise weitere Daten angefügt werden. So ist es ohne weiteres möglich, dass der Client bestimmte Daten zu einem Einsatz ebenfalls erhalten möchte, die aber noch nicht über die API zurückgegeben werden. Es liegt nun an dem ELS diese Daten als Appendices mit einem Einsatz über die Schnittstelle zu übertragen. Sofort stehen dann auch diese Daten über die API zur Verfügung. Appendices werden dabei immer als Appendix-DTO übertragen.

Achtung: Da sich die unterschiedlichen Leitstellenpartner auf die Existenz der hier definierten Appendices verlassen, darf der Key eines Appendix, sofern er hier beschrieben ist, **nicht geändert** werden! Andererseits kann sich der Client darauf verlassen, dass sich der Key eines Appendix, sofern er hier beschrieben ist, nicht ändert.

4.2.4.1 Incident Appendices

Zu diesem Zeitpunkt sind folgende Appendices für Einsätze bekannt. Die farbig hinterlegten Felder sind Pflichtfelder und müssen in jedem Einsatz-Datensatz vorhanden sein.

Prompt	Key	Beschreibung
Incident-ID	IncidentId	eindeutige ID des Einsatzes
Absender-ID	SourceId	Identifikation der sendenden Leitstelle
Empfänger-ID	TargetId	Identifikation der Ziel Leitstelle
Leitstellename	ControlCenterName	Name der sendenden Leitstelle
Einsatzbeginn	IncidentStartTimeStamp	Zeitstempel des Einsatzbeginns (UTC)
Vorgangsnummer	TransactionNumber	Vorgangsnummer (ggf. lesbare ID)
Stichwortkürzel	Classification	Stichworte (Komma getrennt)
Sondersignal	IncidentFlashingLights	Sonderrechte des Einsatzes [true false]

Prompt	Key	Beschreibung
Sachverhalt	Issue	Taktische Bewertung: Sachverhalt
Einsatzstelle		
Einsatzort	MissionDestination	Einsatzstelle: Ort
PLZ	MissionPostalCode	Einsatzstelle: PLZ
Objekt	MissionObject	Einsatzstelle: Bezeichnung Objekt, Firmenname, ...
Objekt Typ	MissionObjectType	Einsatzstelle: Objekttyp (z.B. Krankenhaus, Krankenhausabteilung, Kindergarten, Tankstelle, ...)
Übergeordnetes Objekt	MissionSuperiorObject	übergeordnetes Objekt der Einsatzstelle (z.B. Krankenhaus bei Angabe einer Abteilung als Objekt)
Einsatzstraße	MissionStreet	Einsatzstelle: Straße
Hausnummer	MissionHouseNumber	Einsatzstelle: Hausnummer
Straßensegment	MissionStreetSegment	Einsatzstelle: Straßensegment
Etage	MissionFloor	Einsatzstelle: Etage
Tür	MissionDoor	Einsatzstelle: Tür
Treppenhaus	MissionStaircase	Einsatzstelle: Treppenhaus
Längengrad	MissionLongitude	Einsatzstelle: Längengrad
Breitengrad	MissionLatitude	Einsatzstelle: Breitengrad
Quelle Position	MissionPositionSource	Einsatzstelle: Gibt an aus welcher Quelle die Geokoordinaten stammen. [manualGIS, externalSource, object, segment, street, city]
ECall		
Automatische Aktivierung	eCallAutomaticActivation	Wurde der eCall automatisch oder manuell ausgelöst [true false]
Positions-vertrauens-würdigkeit	eCallPositionCanBeTrusted	Ist die angegebene Fahrzeugposition vertrauenswürdig [true false]. Wert aus MSD
Fahrzeugtyp	eCallVehicleType	Fahrzeugtyp aus MSD z.B. M1 für PKW
Fahrzeug-kennung	eCallVin	Fahrzeugidentifikationsnummer
Fahrzeug-energiespeichera rt	eCallVehiclePropulsion-StorageType	Kommaseparierte Liste der Energiespeicherarten. Mögliche Werte [gasolineTank dieselTank compressedNaturalGas liquidePropaneGas electricEnergyStorage hydrogenStorage]
Zeitstempel	eCallTimestamp	Zeitpunkt des eCall aus MSD

Prompt	Key	Beschreibung
Längengrad	eCallVehicleLatitude	Längengrad der Fahrzeugposition
Breitengrad	eCallVehicleLongitude	Breitengrad der Fahrzeugposition
Fahrtrichtung	eCallDirectionOfTravel	Letzte bekannte Fahrtrichtung in Grad. Auflösung in 2° Schritten
Längengrad N1	eCallLatitudeN1	Längengrad Position N1
Breitengrad N1	eCallLongitudeN1	Breitengrad Position N1
Längengrad N2	eCallLatitudeN2	Längengrad Position N2
Breitengrad N2	eCallLongitudeN2	Breitengrad Position N2
Anzahl der Fahrzeuginsassen	eCallPassengers	Anzahl der Fahrzeuginsassen

4.2.4.2 Informer Appendices

Prompt	Key	Beschreibung
Mitteiler - Name	InformerName	Nachname des Mitteilers
Mitteiler - Vorname	InformerFirstName	Vorname des Mitteilers
Mitteiler - Zweitname	InformerMiddleName	Zweitname des Mitteilers
Mitteiler - Geburtsdatum	InformerDateOfBirth	Geburtsdatum des Mitteilers (UTC)
Mitteiler - Geschlecht	InformerGender	Geschlecht des Mitteilers [female male diverse]
Mitteiler - Erreichbarkeit	InformerReach	Erreichbarkeit des Mitteilers
Adresse		
Mitteiler - Ort	InformerCity	Mitteiler: Stadt
Mitteiler - PLZ	InformerPostalCode	Mitteiler: PLZ
Mitteiler - Straße	InformerStreet	Mitteiler: Straße
Mitteiler - Hausnummer	InformerHouseNumber	Mitteiler: Hausnummer
Mitteiler - Segment	InformerSegment	Mitteiler: Straßensegment
Mitteiler - Etage	InformerFloor	Mitteiler: Etage
Mitteiler - Tür	InformerDoor	Mitteiler: Wohnungstür / -nummer
Mitteiler - Treppenhaus	InformerStaircase	Mitteiler: Treppenhaus
Mitteiler - Objekt	InformerObject	Mitteiler: Bezeichnung Objekt, Firmenname, ...
Mitteiler - Objekt Typ	InformerObjectType	Mitteiler: Objekttyp (z.B. Krankenhaus, Kindergarten, Tankstelle, ...)

Prompt	Key	Beschreibung
Mitteiler – Übergeordnetes Objekt	InformerSuperiorObject	Mitteiler: Übergeordnetes Objekt (z.B. Krankenhaus bei Angabe einer Abteilung als Objekt)

4.2.4.3 Resource Appendices

Zu diesem Zeitpunkt sind folgende Appendices für Einsätze bekannt. Die farbig hinterlegten Felder sind Pflichtfelder und müssen bei jeder Ressource vorhanden sein.

Prompt	Key	Beschreibung
Funkrufname	RadioCallSign	Funkrufname
Fahrzeugtyp	Type	Fahrzeugtyp
Fahrzeugname	Name	Fahrzeugname
Hilfsorganisation	Department	Hilfsorganisation (z.B. Feuerwehr Musterstadt, DRK Musterkreis, ...)
Hilfetyp	DepartmentType	Hilfetyp (z.B. Feuerwehr, Rettungsdienst, Stadtwerke, ...)
Auftragsnummer	OrderNumber	Auftragsnummer
Führungsfahrzeug	CommandVehicle	Führungsfahrzeug [true false]
Position		
Zeitstempel	PositionTimeStamp	Der Zeitpunkt an dem die Position der Ressource zum letzten Mal aktualisiert wurde.
Längengrad	Longitude	Der Längengrad der Position einer Ressource.
Breitengrad	Latitude	Der Breitengrad der Position einer Ressource.
Status		
Zeitstempel	StatusTimeStamp	Der Zeitpunkt an dem der Status der Ressource zum letzten Mal aktualisiert wurde.
Status	Status	Der Status der Ressource.
abweichende Einsatzstelle		
Ort	MissionLocation	Ort der abweichenden Einsatzstelle
PLZ	MissionPostalCode	PLZ der abweichenden Einsatzstelle
Straße	MissionStreet	Straße der abweichenden Einsatzstelle
Hausnummer	MissionHouseNumber	Hausnummer der abweichenden Einsatzstelle
Objekt	MissionObject	Bezeichnung Objekt, Firmenname, ... der abweichenden Einsatzstelle

Prompt	Key	Beschreibung
Objekt Typ	MissionObjectType	Typ des Objekts der abweichenden Einsatzstelle (z.B. Krankenhaus, Krankenhausabteilung, Kindergarten, Tankstelle, ...)
Übergeordnetes Objekt	MissionSuperiorObject	übergeordnetes Objekt der abweichenden Einsatzstelle (z.B. Krankenhaus bei Angabe einer Abteilung als Objekt)
Längengrad	MissionLongitude	Längengrad der abweichenden Einsatzstelle
Breitengrad	MissionLatitude	Breitengrad der abweichenden Einsatzstelle
Quelle Position	MissionPositionSource	Einsatzstelle: Gibt an aus welcher Quelle die Geokoordinaten stammen. [manualGIS, externalSource, object, segment, street, city]

4.2.4.4 Patient Appendices

Zu diesem Zeitpunkt sind folgende Appendices für Patienten bekannt:

Prompt	Key	Beschreibung
Vorname	PatientFirstName	Vorname des Patienten
Zweitname	PatientMiddleName	Zweitname des Patienten
Nachname	PatientLastName	Nachname des Patienten
Geburtsdatum	PatientDateOfBirth	Geburtsdatum des Patienten (ISO 8601 , Coordinated Universal Time)
Geschlecht	PatientGender	Geschlecht des Patienten [female male diverse]
Sichtungskategorie	TriageTag	Im MANV, Kategorie des Patienten nach der Sichtung [green, yellow, red, blue, black]
Verletztenanhängekarte ID	RegistrationCardId	ID der Verletztenanhängekarte
Krankenkasse	HealthInsuranceCompany	der Name der Krankenkasse des Patienten
Krankenkassennummer	HealthInsuranceCompanyNumber	die Nummer der Krankenkasse des Patienten
Versichertennummer	InsuranceNumber	die Versichertennummer des Patienten
Diagnose	Diagnosis	Diagnose des Patienten
	Adresse	
Ort	PatientLocation	Ort der Adresse
PLZ	PatientPostalCode	PLZ der Adresse

Prompt	Key	Beschreibung
Straße	PatientStreet	Straße der Adresse
Hausnummer	PatientHouseNumber	Hausnummer der Adresse
Objekt	PatientObject	Bezeichnung Objekt, Firmenname, ... der Adresse
Objekt Typ	PatientObjectType	Typ des Objekts der Adresse (z.B. Krankenhaus, Krankenhausabteilung, Kindergarten, Tankstelle, ...)
Übergeordnetes Objekt	PatientSuperiorObject	übergeordnetes Objekt der Adresse (z.B. Krankenhaus bei Angabe einer Abteilung als Objekt)
	Transportziel	
Ort	TransportLocation	Ort des Transportziels
PLZ	TransportPostalCode	PLZ des Transportziels
Straße	TransportStreet	Straße des Transportziels
Hausnummer	TransportHouseNumber	Hausnummer des Transportziels
Objekt	TransportObject	Bezeichnung Objekt, Firmenname, ... des Transportziels
Objekt Typ	TransportObjectType	Typ des Objekts des Transportziels (z.B. Krankenhaus, Krankenhausabteilung, Kindergarten, Tankstelle, ...)
Übergeordnetes Objekt	TransportSuperiorObject	übergeordnetes Objekt des Transportziels (z.B. Krankenhaus bei Angabe einer Abteilung als Objekt)

4.3. Home-DTO

Beschreibt die Version der Einsatzdatenaustausch-Schnittstelle

Property	Type	Description
version	Version-DTO	Informationen zur Schnittstellen Versionen
links	Array of Link	Links zu Schnittstellenressourcen

Home-DTO

```

1 {
2   "version":
3   {
4     "ucriVersion": "1.0",
5     "version": "3.1.0.e0c65d",
6     "applicationName": "universal.IX"

```

```

7   },
8   "links": [
9     {
10    "rel": "incidents",
11    "href": "https://[host]/api/incidents/",
12    "label": null,
13    "name": null
14  },
15  {
16    "rel": "classifications",
17    "href": "https://[host]/api/classifications/",
18    "label": null,
19    "name": null
20  },
21  {
22    "rel": "resourceTypes",
23    "href": "https://[host]/api/resourceTypes/",
24    "label": null,
25    "name": null
26  }
27 ]
28 }

```

4.4. Version-DTO

Beschreibt die Version der Einsatzdatenaustausch-Schnittstelle

Property	Type	Description
ucriVersion	String	Version der UCRI Schnittstellen Spezifikation
version	String	Hersteller Produktversion
applicationName	String	Hersteller Produkt-/Schnittstelle-Name

Version-DTO

```

1  {
2  "ucriVersion": "1.0",
3  "version": "3.1.0.e0c65d",
4  "applicationName": "universal.IX"
5  }

```

4.5. JSON Web Token

Das *JWT* (JSON Web Token) beinhaltet das *Access Token* mit dem der Zugriff auf das API verifiziert wird.

Property	Type	Description
accessToken	String	JSON Web Token - verifiziert den Zugriff auf die API
expiresIn	Number	Unix-Timestamp - nach Ablauf dieser Zeit ist das JSON Web Token ungültig
tokenType	String	z.B. 'Bearer' - der Typ des JSON Web Tokens

JSON Web Token-DTO

```

1  {

```

```

2  "access_token": "dea839e9d3e09b4d4c00ba1fb479646a",
3  "expires_in": 3600,
4  "token_type": "Bearer"
5  }

```

4.6. NotAuthorized-DTO

Wird im HTTP-Body zurückgegeben, wenn ein Zugriff nicht autorisiert war. Der enthaltene *Link* kann genutzt werden um den Client neu zu autorisieren (*siehe Authorize*).

Property	Type	Description
message	String	Eine Nachricht, die beschreibt warum man nicht autorisiert ist.
authorizeLink	Link	Links zum Authorize-Endpoint

NotAuthorized-DTO

```

1  {
2  "message": "You are Unauthorized, check if your token is valid or if you didn't send
3  one, follow the AuthorizeLink pls.",
4  "authorizeLink":
5  {
6  "rel": "authorize",
7  "href": "https://[host]/api/authorize/",
8  "label": null,
9  "name": null
10 }

```

4.7. Incident-DTO

Beschreibt ein Einsatzdatenobjekt.

Property	Type	Description
id	String	Eindeutige ID des Einsatzes
classifications	Array of String	Stichwortkürzel des Einsatzes
sharedIncidentId	String	GUID wird vor der Übergabe vom Absendersystem festgelegt. Dient zur Identifizierung des Einsatzes auf allen beteiligten Systemen.
sender	String	Identifikation der sendenden Leitstelle
target	String	Identifikation der empfangenden Leitstelle
sentByDispatcherAt	Timestamp	Zeitpunkt an dem die Übergabe vom Disponenten angestoßen wurde
sentByElsAt	Timestamp	Zeitpunkt zu dem das sendende ELS den Einsatz übermittelt hat
requestedResources	Array of ResourceRequest-DTO	Die angeforderte Einsatzmittel und/oder Einsatzmitteltypen

Property	Type	Description
resources	Array of Resource-DTO	Die am Einsatz beteiligten Ressourcen
patients	Array of Patient-DTO	Die am Einsatz beteiligten Patienten, die noch keiner Ressource zugeteilt wurden
informers	Array of Informer-DTO	Die Mitteleiler zum Einsatz
appendices	Array of Appendix-DTO	Zusatzattribute des Einsatzes
sections	Array of Incident-DTO	Die Abschnitte des Einsatzes
links	Array of Links	Eine Liste mit Links zu Commands zu diesem Einsatz

Incident-DTO

```

1  {
2  "Id": "100001",
3  "sharedIncidentId": "045840e7-fced-4c52-8d88-504b073f46b5",
4  "sender": "els1",
5  "target": "els2",
6  "sentByDispatcherAt": "\2018-10-22T13:15:10.000Z",
7  "sentByElsAt": "\2018-10-22T13:15:11.000Z",
8  "classifications": [
9    "b1",
10   "b2"
11  ],
12  "requestedResources": [
13    {
14      "requestId": "10002003",
15      "resourceType": "KTW",
16      "targetId": "els2",
17      "resourceName": "KTW-1",
18      "resourceId": "1234"
19    },
20    {
21      "requestId": "10002005",
22      "resourceType": "RTW",
23      "targetId": "els2",
24      "resourceName": "RTW-3",
25      "resourceId": "5674"
26    }
27  ],
28  "resources": [
29    {
30      "resourceId": "200001",
31      "stationId": "300001",
32      "alarmMarchOutOrderNo": "1",
33      "patients": [
34        {
35          "id": "400001",
36          "appendices": [
37            {
38              "key": "PatientFirstName",
39              "value": "Max"
40            },
41            {
42              "key": "PatientMiddleName",

```

```

43      "value": "Hubertus"
44      },
45      {
46      "key": "PatientLastName",
47      "value": "Mustermann"
48      },
49      {
50      "key": "PatientDateOfBirth",
51      "value": "\"1992-01-01T00:00:00.000Z\""
52      },
53      {
54      "key": "PatientGender",
55      "value": "Male"
56      },
57      {
58      "key": "TriageTag",
59      "value": "T2"
60      },
61      {
62      "key": "RegistrationCardId",
63      "value": "XX-XXXX"
64      },
65      {
66      "key": "HealthInsuranceCompany",
67      "value": "Musterkasse"
68      },
69      {
70      "key": "HealthInsuranceCompanyNumber",
71      "value": "XXXXXXXXXXXX"
72      },
73      {
74      "key": "InsuranceNumber",
75      "value": "XXXXXXXXXXXX"
76      },
77      {
78      "key": "Diagnosis",
79      "value": "akute Bronchitis"
80      },
81      {
82      "key": "PatientLocation",
83      "value": "Musterstadt"
84      },
85      {
86      "key": "PatientPostalCode",
87      "value": "XXXXX"
88      },
89      {
90      "key": "PatientStreet",
91      "value": "Musterstraße"
92      },
93      {
94      "key": "PatientHouseNumber",
95      "value": "10a"
96      },
97      {
98      "key": "TransportLocation",
99      "value": "Musterstadt"
100     },
101     {

```

```

102         "key": "TransportPostalCode",
103         "value": "XXXXX"
104     },
105     {
106         "key": "TransportStreet",
107         "value": "Hospitalstraße"
108     },
109     {
110         "key": "TransportHouseNumber",
111         "value": "1"
112     },
113     {
114         "key": "TransportObject",
115         "value": "Hospital"
116     },
117     {
118         "key": "TransportObjectType",
119         "value": "KH"
120     }
121 ]
122 }
123 ],
124 "appendices": [
125     {
126         "key": "RadioCallSign",
127         "value": "M-0815"
128     },
129     {
130         "key": "Type",
131         "value": "KTW"
132     },
133     {
134         "key": "Name",
135         "value": "M-0815"
136     },
137     {
138         "key": "Department",
139         "value": "DRK"
140     },
141     {
142         "key": "DepartmentType",
143         "value": "RET"
144     },
145     {
146         "key": "OrderNumber",
147         "value": "O-XX-XXXX"
148     },
149     {
150         "key": "CommandVehicle",
151         "value": "False"
152     },
153     {
154         "key": "AlarmTimeStamp",
155         "value": "\"2018-10-22T13:13:00.000Z\""
156     },
157     {
158         "key": "PositionTimeStamp",
159         "value": "\"2018-10-22T13:12:59.000Z\""
160     },

```

```

161     {
162         "key": "Longitude",
163         "value": "0.00"
164     },
165     {
166         "key": "Latitude",
167         "value": "0.00"
168     },
169     {
170         "key": "StatusTimeStamp",
171         "value": "\"2018-10-22T13:13:00.000Z\""
172     },
173     {
174         "key": "Status",
175         "value": "2"
176     }
177 ]
178 }
179 ],
180 "appendices": [
181     {
182         "key": "AlarmTimeStamp",
183         "value": "\"2018-10-22T13:10:00.000Z\""
184     },
185     {
186         "key": "IncidentStartTimeStamp",
187         "value": "\"2018-10-22T13:12:30.000Z\""
188     },
189     {
190         "key": "IncidentFlashingLights",
191         "value": "True"
192     },
193     {
194         "key": "TransactionNumber",
195         "value": "T-XX-XXX"
196     },
197     {
198         "key": "IncidentNumber",
199         "value": "I-XX-XXX"
200     },
201     {
202         "key": "SectionName",
203         "value": "Haupteinsatz"
204     },
205     {
206         "key": "Issue",
207         "value": "Kranke Person in Praxis"
208     },
209     {
210         "key": "MissionDestination",
211         "value": "Musterstadt"
212     },
213     {
214         "key": "MissionPostalCode",
215         "value": "XXXXX"
216     },
217     {
218         "key": "MissionObject",
219         "value": "Arztpraxis"

```

```

220     },
221     {
222         "key": "MissionStreet",
223         "value": "Praxisgasse"
224     },
225     {
226         "key": "MissionHouseNumber",
227         "value": "5c"
228     },
229     {
230         "key": "MissionFloor",
231         "value": "Souterrain"
232     },
233     {
234         "key": "MissionDoor",
235         "value": "4"
236     }
237 ],
238 "patients": [],
239 "informers": [
240     {
241         "id": "I-0815",
242         "appendices": [
243             {
244                 "key": "InformerName",
245                 "value": "Mustermann"
246             },
247             {
248                 "key": "InformerFirstName",
249                 "value": "Max"
250             },
251             {
252                 "key": "InformerCity",
253                 "value": "Musterstadt"
254             },
255             {
256                 "key": "InformerReach",
257                 "value": "+49 (0) XXXX XXXXXX"
258             }
259         ]
260     }
261 ]
262 links: [
263     {
264         "rel": "confirm",
265         "href": "https://[host]/api/incidents/045840e7-fced-4c52-8d88-
266             504b073f46b5/acknowledgements/confirm/",
267         "label": null,
268         "name": null
269     },
270     {
271         "rel": "retrieved",
272         "href": "https://[host]/api/incidents/045840e7-fced-4c52-8d88-
273             504b073f46b5/acknowledgements/retrieved/",
274         "label": null,
275         "name": null
276     }

```

```

277     "href": "https://[host]/api/incidents/045840e7-fced-4c52-8d88-
278         504b073f46b5/messages/ ",
279     "label": null,
280     "name": null
281   }
282 ]

```

4.8. Message-DTO

Beschreibt eine Nachricht zum Einsatz.

Property	Type	Description
sharedIncidentId	String	GUID zur Identifizierung des Einsatzes auf beteiligten Systemen
message	String	Nachrichtentext (Freitext)

Message-DTO - Beispiel

```

1  {
2    "sharedIncidentId": "ABCDEFGH-1234-5678-90IJ-KLMNOPQRSTU",
3    "message": " Lorem ipsum dolor sit amet..."
4  }

```

4.9. Informer-DTO

Beschreibt einen Mitteiler eines Einsatzes

Property	Type	Description
incidentId	String	die ID des Einsatzes zu dem der Mitteiler gehört
id	String	die ID des Mitteilers
appendices	Array of Appendix-DTO	Zusatzattribute des Mitteilers

Informer-DTO - Beispiel

```

1  {
2    "incidentId": "I-2018-10-24_0001",
3    "id": "I-0815",
4    "appendices": [
5      {
6        "key": "InformerName",
7        "value": "Mustermann"
8      },
9      {
10       "key": "InformerFirstName",
11       "value": "Max"
12     },
13     {
14       "key": "InformerCity",
15       "value": "Musterstadt"
16     },
17     {
18       "key": "InformerReach",
19       "value": "+49 (0) XXXX XXXXXX"
20     }
21   ]
22 }

```

4.10. Resource-DTO

Beschreibt eine Ressource eines Einsatzes.

Property	Type	Description
incidentId	String	die ID des Einsatzes zu dem die Ressource gehört
id	String	die ID der Ressource
appendices	Array of Appendix-DTO	Zusatzattribute der Ressource
patients	Array of Patient-DTO	Liste von Patienten die der Ressource zugeordnet sind

Resource-DTO - Beispiel

```

1  {
2    "incidentId": "I-2018-10-24_0001",
3    "id": "200001",
4    "appendices": [
5      {
6        "key": "RadioCallSign",
7        "value": "M-0815"
8      },
9      {
10       "key": "Type",
11       "value": "KTW"
12     },
13     {
14       "key": "Name",
15       "value": "M-0815"
16     },
17     {
18       "key": "Department",
19       "value": "DRK"
20     },
21     {
22       "key": "DepartmentType",
23       "value": "RET"
24     },
25     {
26       "key": "OrderNumber",
27       "value": "O-XX-XXXX"
28     },
29     {
30       "key": "CommandVehicle",
31       "value": "False"
32     },
33     {
34       "key": "AlarmTimeStamp",
35       "value": "\"2018-10-22T13:13:00.000Z\""
36     },
37     {
38       "key": "PositionTimeStamp",
39       "value": "\"2018-10-22T13:12:59.000Z\""
40     },
41     {
42       "key": "Longitude",
43       "value": "0.00"

```

```

44     },
45     {
46         "key": "Latitude",
47         "value": "0.00"
48     },
49     {
50         "key": "StatusTimeStamp",
51         "value": "\"2018-10-22T13:13:00.000Z\""
52     },
53     {
54         "key": "Status",
55         "value": "2"
56     }
57 ],
58 "patients": [
59     {
60         "incidentId": "I-2018-10-24_0001",
61         "resourceId": "200001",
62         "id": "400001",
63         "thumbprint": "9841894165196841",
64         "appendices": [
65             {
66                 "key": "PatientFirstName",
67                 "value": "Max"
68             },
69             {
70                 "key": "PatientMiddleName",
71                 "value": "Hubertus"
72             },
73             {
74                 "key": "PatientLastName",
75                 "value": "Mustermann"
76             },
77             {
78                 "key": "PatientDateOfBirth",
79                 "value": "\"1992-01-01T00:00:00.000Z\""
80             },
81             {
82                 "key": "PatientGender",
83                 "value": "Male"
84             },
85             {
86                 "key": "TriageTag",
87                 "value": "T2"
88             },
89             {
90                 "key": "RegistrationCardId",
91                 "value": "XX-XXXX"
92             },
93             {
94                 "key": "HealthInsuranceCompany",
95                 "value": "Musterkasse"
96             },
97             {
98                 "key": "HealthInsuranceCompanyNumber",
99                 "value": "XXXXXXXXXXXX"
100            },
101            {
102                "key": "InsuranceNumber",

```

```

103     "value": "XXXXXXXXXXXXX"
104     },
105     {
106         "key": "Diagnosis",
107         "value": "akute Bronchitis"
108     },
109     {
110         "key": "PatientLocation",
111         "value": "Musterstadt"
112     },
113     {
114         "key": "PatientPostalCode",
115         "value": "XXXXX"
116     },
117     {
118         "key": "PatientStreet",
119         "value": "Musterstraße"
120     },
121     {
122         "key": "PatientHouseNumber",
123         "value": "10a"
124     },
125     {
126         "key": "TransportLocation",
127         "value": "Musterstadt"
128     },
129     {
130         "key": "TransportPostalCode",
131         "value": "XXXXX"
132     },
133     {
134         "key": "TransportStreet",
135         "value": "Hospitalstraße"
136     },
137     {
138         "key": "TransportHouseNumber",
139         "value": "1"
140     },
141     {
142         "key": "TransportObjectId",
143         "value": "123456"
144     },
145     {
146         "key": "TransportObject",
147         "value": "Hospital"
148     },
149     {
150         "key": "TransportObjectType",
151         "value": "KH"
152     }
153 ]
154 }
155 ]
156 }

```

4.11. Patient-DTO

Beschreibt einen Patienten.

Property	Type	Description
incidentId	String	die ID des Einsatzes zu dem der Patient gehört
resourceId	String	die ID der Ressource falls der Patient einer Ressource zugewiesen ist
id	String	die ID des Patienten
appendices	Array of Appendix-DTO	Zusatzattribute des Patienten

Patient-DTO - Beispiel

```

1  {
2    "incidentId": "I-2018-10-24_0001",
3    "resourceId": null,
4    "id": "400002",
5    "appendices": [
6      {
7        "key": "PatientFirstName",
8        "value": "Maxine"
9      },
10     {
11       "key": "PatientMiddleName",
12       "value": "Hildegard"
13     },
14     {
15       "key": "PatientLastName",
16       "value": "Mustermann"
17     },
18     {
19       "key": "PatientDateOfBirth",
20       "value": "\"1994-02-20T00:00:00.000Z\""
21     },
22     {
23       "key": "PatientGender",
24       "value": "Female"
25     },
26     {
27       "key": "TriageTag",
28       "value": "T2"
29     },
30     {
31       "key": "RegistrationCardId",
32       "value": "XX-XXXX"
33     },
34     {
35       "key": "HealthInsuranceCompany",
36       "value": "Musterkasse"
37     },
38     {
39       "key": "HealthInsuranceCompanyNumber",
40       "value": "XXXXXXXXXXXX"
41     },
42     {
43       "key": "InsuranceNumber",
44       "value": "XXXXXXXXXXXX"
45     },
46     {

```

```

47     "key": "Diagnosis",
48     "value": "akute Bronchitis"
49   },
50   {
51     "key": "PatientLocation",
52     "value": "Musterstadt"
53   },
54   {
55     "key": "PatientPostalCode",
56     "value": "XXXXX"
57   },
58   {
59     "key": "PatientStreet",
60     "value": "Musterstraße"
61   },
62   {
63     "key": "PatientHouseNumber",
64     "value": "10a"
65   },
66   {
67     "key": "TransportLocation",
68     "value": "Musterstadt"
69   },
70   {
71     "key": "TransportPostalCode",
72     "value": "XXXXX"
73   },
74   {
75     "key": "TransportStreet",
76     "value": "Hospitalstraße"
77   },
78   {
79     "key": "TransportHouseNumber",
80     "value": "1"
81   },
82   {
83     "key": "TransportObject",
84     "value": "Hospital"
85   },
86   {
87     "key": "TransportObjectType",
88     "value": "KH"
89   }
90 ]
91 }

```

4.12. Appendix-DTO

Beschreibt Zusatzdaten zu einer Ressource oder einem Einsatz.

Property	Type	Description
key	String	der Schlüssel des Appendix
value	String	der Wert des Appendix

Appendix-DTO - Beispiel

```

5   {
6     "key": "PositionTimeStamp",
7     "value": "\"2018-10-22T13:12:59.000Z\""

```

8 }

4.13. ResourceRequest-DTO

Beschreibt eine Einsatzmittel-Anforderung

Property	Type	Description
requestId	String	Eindeutige ID der Anforderung
resourceType	String	Ressourcentyp
targetId	String	ID der empfangenden Leitstelle
resourceName	String	Ressourcenname
resourceId	String	RessourcenId der empfangenden Leitstelle
links	Array of Links	Eine Liste mit Links zu Commands zu dieser Einsatzmittelanforderung

ResourceRequest-DTO

```

1  {
2    "requestId": "10002003",
3    "resourceType": "KTW",
4    "targetId": "els2",
5    "resourceName": "KTW-1",
6    "resourceId": "1234",
7    "links": [
8      {
9        "rel": "acceptResourceRequest",
10       "href": "https://[host]/api/incidents/1/resourceRequests/1234/accept/",
11       "label": null,
12       "name": null
13     },
14     {
15       "rel": "rejectResourceRequest",
16       "href": "https://[host]/api/incidents/1/resourceRequests/1234/reject/",
17       "label": null,
18       "name": null
19     },
20     {
21       "rel": "temporarilyAcceptResourceRequest",
22       "href":
23         "https://[host]/api/incidents/1/resourceRequests/1234/acceptTemporary/",
24       "label": null,
25       "name": null
26     },
27     {
28       "rel": "confirmTurnOutForResourceRequest",
29       "href": "https://[host]/api/incidents/1/resourceRequests/1234/confirmTurnOut/",
30       "label": null,
31       "name": null
32     }
33   ]
34 }
```

4.14. Confirmation-DTO

Beschreibt eine Quittierung im Sinne der Einsatzübergabe und ist nur für diese relevant.

Property	Type	Description
sharedIncidentId	String	GUID zur Identifizierung des Einsatzes auf beteiligten Systemen.
status	String	Status der Quittierung (mögliche Status: Accepted, Rejected)
timestamp	String	Zeitstempel an dem die Quittierung erfolgte. (ISO 8601 , Coordinated Universal Time)
cause	String	Grund der Quittierung bei Fehler oder wenn der Einsatz abgelehnt wurde
appendices	Array of Appendix-DTO	Zusatzattribute der Quittierung

Confirmation-DTO

```

1  {
2  "sharedIncidentId": "ABCDEFGH-1234-5678-90IJ-KLMNOPQRSTU",
3  "status": "Rejected",
4  "timestamp": "\"2019-06-11T08:24:06.676Z\"",
5  "cause": "Keine Einsatzmittel verfügbar",
6  "appendices": [
7  {
8  "key": "RefMissionId",
9  "value": "4711GD"
10 }
11 ]
12 }
```

4.15. PositiveConfirmation-DTO

Beschreibt eine positive Quittierung einer Einsatzmittelanforderung und ist nur für diese relevant. Auf eine Einsatzmittelanforderung können mehrere positive Quittierungen folgen. Der Disponent kann der anfordernden Leitstelle über eine Nachricht zum Einsatz mitteilen, dass weitere Einsatzmittel kommen.

Property	Type	Description
requestId	String	Eindeutige Id der Einsatzmittelanforderung
timestamp	DateTime	Zeitstempel der Quittierung
resourceType	String	Entsendeter Ressourcentyp
resourceName	String	Name der entsendeten Ressource

Property	Type	Description
message	String	Nachricht zur Bestätigung

PositiveConfirmation-DTO

```

1 {
2   "requestId": "10002003",
3   "timestamp": "\"2018-10-22T13:12:59.000Z\"",
4   "resourceType": "KTW",
5   "resourceName": "KTW-1",
6   "resourceId": "1234"
7   "message": "Freitextfeld"
8 }
```

4.16. NegativeConfirmation-DTO

Beschreibt eine negative Quittierung einer Einsatzmittelanforderung und ist nur für diese relevant.

Property	Type	Description
requestId	String	Eindeutige Id der Einsatzmittelanforderung
timestamp	DateTime	Zeitstempel der Quittierung
reason	String	Grund der Ablehnung
message	String	Nachricht zur Bestätigung

NegativeConfirmation-DTO

```

1 {
2   "requestId": "10002003",
3   "timestamp": "\"2018-10-22T13:12:59.000Z\"",
4   "reason": "Einsatzmittel nicht verfügbar"
5   "message": "Freitextfeld"
6 }
```

4.17. ResourceTurnOutConfirmation-DTO

Beschreibt eine Quittierung, die angibt, dass das angeforderte Einsatzmittel ausgerückt ist.

Dieses DTO ist nur für die Einsatzmittelanforderung relevant und darf erst nach dem Versand eines PositivConfirmation-DTO eingesetzt werden

Property	Type	Description
requestId	String	Eindeutige ID der Einsatzmittelanforderung
timestamp	DateTime	Zeitstempel der Quittierung
resourceType	String	Entsendeter Ressourcentyp
resourceName	String	Name der entsendeten Ressource
resourceId	String	RessourceId der entsendeten Ressource
message	String	Nachricht zur Bestätigung

ResourceTurnOutConfirmation-DTO

```

1 {
2   "requestId": "10002003",
3   "timestamp": "\"2018-10-22T13:12:59.000Z\"",
4   "resourceType": "KTW",
5   "resourceName": "KTW-1",
6   "resourceId": "1234"
7   "message": "Freitextfeld"
8 }

```

4.18. TemporaryPositiveConfirmation-DTO

Beschreibt eine vorläufige positive Quittierung einer Einsatzmittelanforderung und ist nur für diese relevant.

Sie kann optional vor dem PositivConfirmation-DTO versendet werden, um dem Disponenten eine schnelle erste Rückmeldung zu ermöglichen, bei der die konkreten Einsatzmittel noch nicht festgelegt sind.

Property	Type	Description
requestId	String	Eindeutige Id der Einsatzmittelanforderung
timestamp	DateTime	Zeitstempel der Quittierung
message	String	Kommentar

TemporaryPositiveConfirmation-DTO

```

1 {
2   "requestId": "10002003",
3   "timestamp": "\"2018-10-22T13:12:59.000Z\"",
4   "message": "Schicke das Einsatzmittel sobald es Bereit ist"
5 }

```

4.19. Classification-DTO

Beschreibt ein Einsatz-Stichwort

Property	Type	Description
abbreviation	String	Stichwort-Kürzel
description	String	Beschreibung zum Stichwort
classificationGroup	String	Name der Stichwortgruppe der das Stichwort zugehört

Classification-DTO

```

1 {
2   "abbreviation": "B1",
3   "description": "kleiner Brand",
4   "classificationGroup": "fw sofort"
5 }

```

4.20. ResourceType-DTO

Beschreibt einen Einsatzmitteltyp

Property	Type	Description
abbreviation	String	Einsatzmitteltyp-Kürzel

Property	Type	Description
description	String	Beschreibung des Einsatzmitteltyps

ResourceType-DTO

```
1 {  
2   "abbreviation": "TLF",  
3   "description": "Tanklöschfahrzeug"  
4 }
```

5 API-Requests

5.1. Home

https://[host]/api/home		
Zweck	Liefert Informationen zur Schnittstellenversion und den verfügbaren Ressourcen	
HTTP-Method	GET	
HTTP-Header	Authorization: Bearer [Access-Token]	
Wann	Beim Aufbauen der Verbindung zwischen 2 Systemen	
Links-Relation	Home	
Responses		
Status Code	Message	HTTP-Body
200	OK	Home-DTO
401	Client is not authorized yet, token is invalid or expired	NotAuthorized-DTO
500	Something went wrong.	-

5.2. Authorize

https://[host]/api/authorize		
Zweck	stellt ein Access-Token aus	
HTTP-Method	GET	
HTTP-Header	Authorization: Basic [Base64-Encoded-Client-Secret] ClientId: [Base64-Encoded-Client-Id]	
Wann	das Access-Token ist abgelaufen, es ist keines vorhanden oder es wird ein Status Code Unauthorized (401) zurück gegeben	
Links-Relation	authorize	
Responses		
Status Code	Message	HTTP-Body
200	OK	JWT
400	The request was incomplete.	-
401	Not allowed to authorize.	-
500	Something went wrong.	-

5.3. Einsatzdaten entgegennehmen

https://[host]/api/incidents	
Zweck	Entgegennehmen eines Einsatzes einer anderen Leitstelle
HTTP-Method	POST
HTTP-Header	Authorization: Bearer [Access-Token]
Request-Body	Incident-DTO

https://[host]/api/incidents		
Responses		
Status Code	Message	HTTP-Body
201	Incident has been created	-
400	Incident is invalid	-
401	Client is not authorized yet, token is invalid or expired	NotAuthorized-DTO
500	Internal Server Error	-

5.4. Nachricht zum Einsatz an korrespondierende Leitstelle

https://[host]/api/incidents/{sharedIncidentId}/messages		
Zweck	Übermittelt eine Nachricht zum Einsatz	
HTTP-Method	POST	
HTTP-Header	Authorization: Bearer [Access-Token]	
Request-Body	Message-DTO	
Wann	Wenn die Disponenten insbesondere bei kooperativen Einsätzen Informationen austauschen wollen.	
Responses		
Status Code	Message	HTTP-Body
200	OK	-
400	Bad Request - see message in body for more Informations	message string
401	Client is not authorized yet, token is invalid or expired	NotAuthorized-DTO
404	Not Found	-
500	Something went wrong	-

5.5. Quittierung Einsatzübergabe

5.5.1. Übertragung des Einsatzes an remote ELS quittieren

https://[host]/api/incidents/{sharedIncidentId}/acknowledgements/retrieved	
Zweck	Quittiert der sendenden Schnittstelle das die Einsatzdaten an das Remote ELS übertragen wurden
HTTP-Method	PUT
HTTP-Header	Authorization: Bearer [Access-Token]
Request-Body	Confirmation-DTO
Wann	Wenn die Einsatzdaten in das Remote-ELS übernommen wurden

https://[host]/api/incidents/{sharedIncidentId}/acknowledgements/retrieved		
Responses		
Status Code	Message	HTTP-Body
200	OK	-
400	The request was incomplete.	-
401	Client is not authorized yet, token is invalid or expired	NotAuthorized-DTO
500	Something went wrong.	-

5.5.2. Einsatzübergabe quittieren

https://[host]/api/incidents/{sharedIncidentId}/acknowledgements/confirm		
Zweck	Quittiert der sendenden Schnittstelle das die empfangende Leitstelle den Einsatz angenommen oder abgelehnt hat	
HTTP-Method	PUT	
HTTP-Header	Authorization: Bearer [Access-Token]	
Request-Body	Confirmation-DTO	
Wann	Wenn der Disponent den Einsatz angenommen oder abgelehnt hat	
Responses		
Status Code	Message	HTTP-Body
200	OK	-
400	Bad Request - see message in body for more Informations	message string
401	Client is not authorized yet, token is invalid or expired	NotAuthorized-DTO
404	Not Found	-
500	Something went wrong	-

5.6. Quittierung Einsatzmittelanforderung

5.6.1. Positive Quittierung

https://[host]/api/incidents/{sharedIncidentId}/resourceRequests/{requestId}/accept		
Zweck	Positive Quittierung der Einsatzmittelanforderung	
HTTP-Method	POST	
HTTP-Header	Authorization: Bearer [Access-Token]	
Request-Body	PositiveConfirmation-DTO	
Responses		
Status Code	Message	HTTP-Body
202	Accepted	-

https://[host]/api/incidents/{sharedIncidentId}/resourceRequests/{requestId}/accept		
400	The request body is invalid or incomplete	message string
401	Client is not authorized yet, token is invalid or expired	NotAuthorized-DTO
500	Internal Server Error	-

5.6.2. Negative Quittierung

https://[host]/api/incidents/{sharedIncidentId}/resourceRequests/{requestId}/reject		
Zweck	Negative Quittierung der Einsatzmittelanforderung	
HTTP-Method	POST	
HTTP-Header	Authorization: Bearer [Access-Token]	
Request-Body	NegativeConfirmation-DTO	
Responses		
Status Code	Message	HTTP-Body
202	Accepted	-
400	The request body is invalid or incomplete	message string
401	Client is not authorized yet, token is invalid or expired	NotAuthorized-DTO
500	Internal Server Error	-

5.6.3. Vorläufig positive Quittierung

https://[host]/api/incidents/{sharedIncidentId}/resourceRequests/{requestId}/acceptTemporary		
Zweck	Vorläufige positive Quittierung der Einsatzmittelanforderung	
HTTP-Method	POST	
HTTP-Header	Authorization: Bearer [Access-Token]	
Request-Body	TemporaryPositiveConfirmation-DTO	
Responses		
Status Code	Message	HTTP-Body
202	Accepted	-
400	The request body is invalid or incomplete	message string
401	Client is not authorized yet, token is invalid or expired	NotAuthorized-DTO
500	Internal Server Error	-

5.6.4. Quittierung Fahrzeug ausgerückt

http://[host]/api/incidents/{sharedIncidentId}/resourceRequest/{requestId}/confirmTurnOut		
Zweck	Quittierung über ein ausgerücktes Einsatzmittel aus der Einsatzmittelanforderung	
HTTP-Method	POST	
HTTP-Header	Authorization: Bearer [Access-Token]	
Request-Body	ResourceTurnOutConfirmation-DTO	
Responses		
Status Code	Message	HTTP-Body
202	Accepted	-
400	The request body is invalid or incomplete	message string
401	Client is not authorized yet, token is invalid or expired	NotAuthorized-DTO
500	Internal Server Error	-

5.7. Stichworte abfragen

http://[host]/api/classifications		
Zweck	Zur Abfrage der im Leitsystem gepflegten Einsatz-Stichworte. Ermöglicht ein Mapping zwischen den Stichworten der fremden und der eigenen Leitstelle. Das Mapping muss durch das empfangende System erfolgen.	
HTTP-Method	GET	
HTTP-Header	Authorization: Bearer [Access-Token]	
Request-Body	-	
Responses		
Status Code	Message	HTTP-Body
200	OK	Array of Classification-DTOs
401	Client is not authorized yet, token is invalid or expired	NotAuthorized-DTO
500	Internal Server Error	message string

5.8. Einsatzmitteltypen abfragen

http://[host]/api/resourceTypes	
Zweck	Zur Abfrage der im Leitsystem gepflegten Einsatzmitteltypen. Ermöglicht ein Mapping zwischen Einsatzmitteltypen der fremden und der eigenen Leitstelle. Das Mapping muss durch das empfangende System erfolgen.
HTTP-Method	GET
HTTP-Header	Authorization: Bearer [Access-Token]

http://[host]/api/resourceTypes		
Request-Body	-	
Responses		
Status Code	Message	HTTP-Body
200	OK	Array of Resource Type-DTOs
401	Client is not authorized yet, token is invalid or expired	NotAuthorized-DTO
500	Internal Server Error	message string

6 Status Codes

Status Code	Bedeutung
200	Erfolgreich
204	angefragte Ressource wurde entfernt oder ist nicht vorhanden
304	angefragte Ressource hat sich nicht verändert
400	Fehler im Request
401	kein JWT vorhanden oder JWT abgelaufen (bei HTTP-Accept-Header application/vnd.simplemedia+json oder application/vnd.collection+json wird ein NotAuthorized mit Authorize Link übertragen)
403	das JWT ist gültig, enthält aber nicht die entsprechende Berechtigung
404	angefragte Ressource wurde nicht gefunden
500	serverseitiger Fehler