Universal Control Room Interface Version 2 (UCRI2)

UCRI steht für Universal Control Room Interface - ein Protokoll für die Kommunikation zwischen zwei oder mehreren Einsatzleitsystemen.

Nach einer erfolgreichen Einführung und eingehender Approbation im Feld wurden viele Erfahrungen gesammelt und Anforderungen identifiziert, die Weiterentwicklung des UCRI Protokolls auf einer neuen architektonischen Basis erforderten. UCRI2 ist eine komplett überarbeitete Version des Protokolls, die alle Anwendungsfälle der Vorgängerversionen (die letzte UCRI Version 1.1) unterstützt und Grundlage für flexible Weiterentwicklung des Protokolls darstellt.

UCRI2 Ziele

Ziele für die Entwicklung von UCRI2 und die Unterschiede zu UCRI 1.x:

- Einfache, schnell zu implementierende API, schnelles PoC möglich
  - standardisierte maschinenlesbare Spezifikation
  - sichere Zustellung und Verarbeitung von Meldungen
- Trennung technischer und fachlicher Aspekte
  - Einfache Erweiterbarkeit
  - Technische Komponenten müssen bei fachlichen Erweiterungen nicht angepasst werden
  - Fachliche Erweiterungen können ohne Anpassungen der Infrastruktur erfolgen
- Routing im Kern des Konzeptes
  - Einfaches Zusammenspiel mehrerer Hersteller
  - Unterstützung zentraler als dezentraler Anbindungen
- Die RPC/REST basierte Architektur in eine Messaging basierte Architektur überführen
- Authentifizierung OAuth2 konform

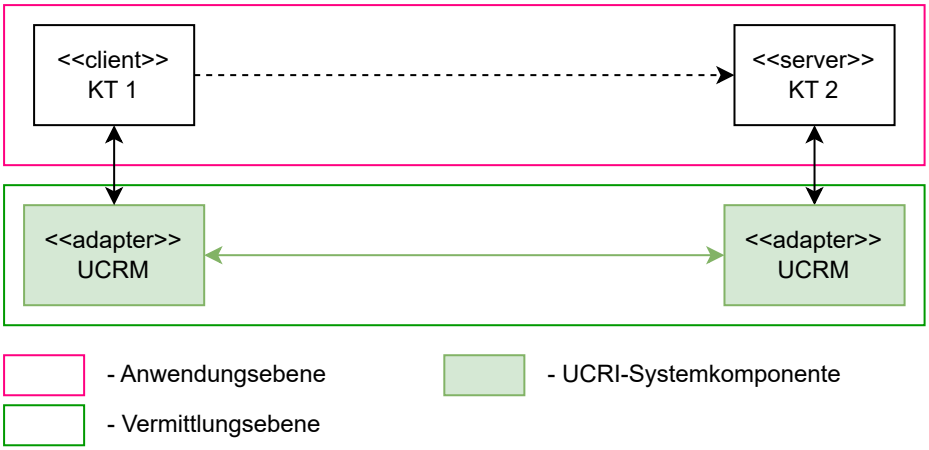
UCRI2 Systemarchitektur

Messaging

Bei der Strukturierung der UCRI2-Schnittstelle wird der Architekturstil Messaging verwendet. Bei diesem Architekturstil kommunizieren verteilte unabhängige Systemkomponenten (im allgemeinen Kommunikationsteilnehmer genannt - KT) miteinander mit Hilfe von Nachrichten.

Messaging-Systeme trennen die fachliche Anwendung (gewöhnlich strukturiert nach Client-Server-Prinzip) von der Vermittlungsebene - also von technischen Aspekten der Nachrichtenübermittlung zwischen technischen Systemen der KT. Nachrichten können während der Übertragung umgewandelt werden, ohne dass Sender oder Empfänger von der Umwandlung wissen. Die Entkopplung ermöglicht es Integratoren, je nach Anforderung unterschiedliche Kommunikationstopologien zu unterstützen, von dezentralen P2P-Protokollen bis zu zentralisierten Broker-Architekturen.

Messaging-Systeme ermöglichen es den Komponenten, entkoppelt zu bleiben und sich auf ihre eigenen Aufgaben zu konzentrieren, während sie gleichzeitig in der Lage sind, mit anderen Komponenten im System zu kommunizieren und zusammenzuarbeiten. Es verringert die Anzahl der Abhängigkeiten zwischen den Komponenten, wodurch das System flexibler und leichter zu warten ist.



Adapter-Komponente

Das andere wichtige Architekturmuster, das bei der Strukturierung der UCRI2-Schnittstelle Verwendung findet ist das Adapter-Muster. Bei diesem Muster erfolgt die Kommunikation zwischen dem technischen System der KT (Anwendungsebene) und der Vermittlungsebene mittels einer Adapter-Komponente (UCRI Control Room Module - UCRM). Der Adapter ermöglicht bidirektionale Kommunikation zwischen den Ebenen in einer standardisierten Form und ermöglicht die Komplexitätsreduzierung der angebundenen Schnittstellen.

Der Adapter kann auch zusätzliche Aufgaben übernehmen, wie z. B. Datentransformation oder Sicherheitsaufgaben wie Ende-zu-Ende-Verschlüsselung von übermittelten Nachrichten.

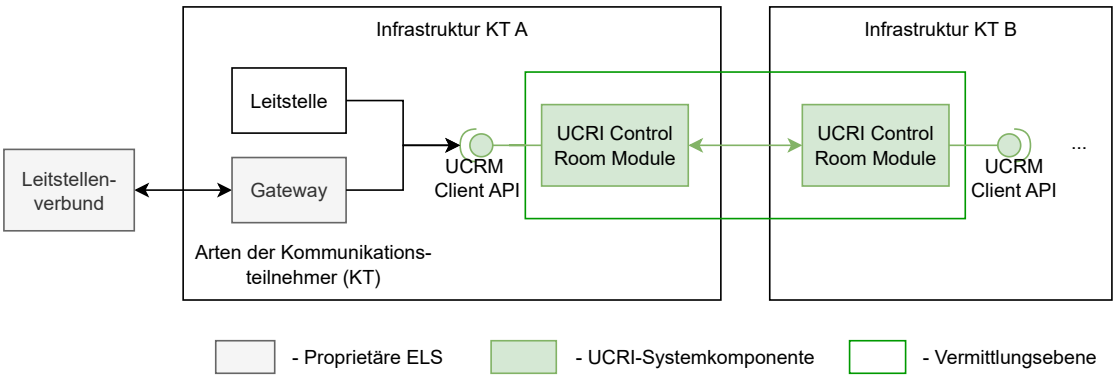
Die Adapter an der Seite von technischen Systemen der KT und optional der Broker (bei einer zentralen Broker-Architektur) kümmern sich somit um die technischen Aspekte der Kommunikation - also die Vermittlungsebene, während die fachlichen Systemkomponenten Clients und Server sich auf die eigentliche Anwendungslogik fokussieren - also die Anwendungsebene.

Vermittlungsebene

Zentrale Aufgabe der Vermittlungsebene ist die Zustellung von Meldungen zwischen Sender und Empfänger. Außerdem müssen auf der Vermittlungsebene unterschiedliche querschnittliche Aufgaben übernommen werden. Hier sind nur einige Beispiele: KT-Authentifizierung und Autorisierung, Meldungsvalidierung, eventuell E2E-Verschlüsselung. Bei der P2P-Topologie kommunizieren die UCRM der zwei KT dabei direkt miteinander.

UCRI Gateway

Die Systemkomponente Gateway stellt einen speziellen KT dar. Das Gateway wird am Übergang zu Gruppen von KT eingesetzt, die auf der Vermittlungsebene nicht direkt erreichbar sind und über eine proprietäres Leitstellenprotokoll angebunden werden können (Leitstellenverbunde). Das Gateway stellt eine Gateway-Funktion bereit zum Mapping zwischen externe Quell- bzw. Zieladressen und UCRI-internen KT-Adressen.



Das Leitstellenmodul (UCRM) stellt die UCRM API bereit - die einzige Kommunikationsschnittstelle für direkt verbundene Kommunikationsteilnehmer wie Leitstellensysteme oder andere technische Knoten, sowie weitere externe Systeme.

Einzelne Aufgaben der Vermittlungsebene, die bei der P2P-Topologie durch die Kommunikation zwischen einzelnen UCRMs umgesetzt werden müssen, sind:

- Verwaltung der Kommunikationstopologie inkl. Adressierungskonzept und KT-Register
- Konzept Authentisierung, Autorisierung, Accounting
- E2E-Verschlüsselung
- Übermittlung von Nachrichten unter der Verwendung eines UCRI-Adressierungskonzepts inkl. Routing-Funktion
- Validierung von Anwendungsmeldungen

Die Vermittlungsebene ist frei von Fachlichkeit. Sie realisiert nur den Datentransport und sichert optional die Ende-zu-Ende-Verschlüsselung der Daten.

Anwendungsebene

Eine UCRI2-Anwendung wird durch folgende Artefakte definiert:

- Ein Satz von standardisierten Nachrichten-Schemata (JSON, kanonisches Datenmodell)
- Ablaufmodell (definierte Abfolge von Nachrichten)
- Prozessdefinitionen (Festlegungen bezüglich Anwendungslogik, die bei der Implementierung in technischen KT-Systemen berücksichtigt werden müssen)

Wichtig ist das Prinzip der Trennung der UCRI2-Anwendungen untereinander und deren Unabhängigkeit von der Vermittlungsebene. Das erlaubt eine freie Weiterentwicklung jeder einzelnen Anwendung.

UCRI2 Vermittlungsebene

Die Vermittlungsebene umfasst technische Aspekte der Nachrichtenübermittlung zwischen technischen Systemen der KT. Dabei können unterschiedliche Kommunikationstopologien unterstützt werden, von dezentralen Peer-To-Peer (P2P) Protokollen bis zu zentralisierten Broker-Architekturen.

Die aktuelle UCRI2 Version spezifiziert nur ein P2P-Protokoll für die Kommunikation zwischen den UCRM-Modulen.

Im Weiteren werden einzelne Aspekte der Nachrichtenübermittlung detailliert beschrieben:

- Adressierungskonzept

- UCRI Leitstellenmodul
- Kommunikationsprotokoll

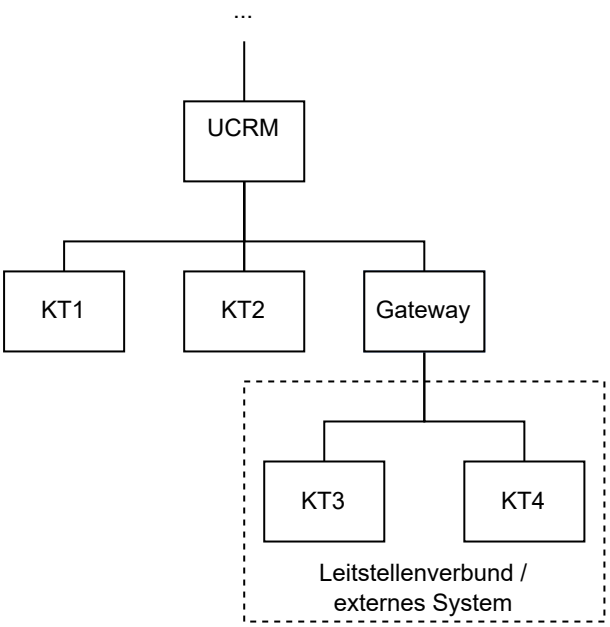
Vermittlungsebene

UCRI2 Adressierungskonzept

Für die Adressierung der einzelnen Kommunikationsteilnehmer (KT) auf der Vermittlungsebene werden eindeutige Kennungen in Form von Object Identifier (OID Spezifikationen ISO/IEC 9834, DIN 66334) verwendet.

OID-Hierarchie

Adressierung von einzelnen KT ist hierarchisch organisiert und spiegelt die hierarchische Kommunikationsstruktur wider:



Diese Struktur ermöglicht Implementierung einer einfachen und einheitlichen Routing-Funktion, die in jeder Systemkomponente (Leitstellenmodul - UCRM, Vermittlungsplattform, Gateway) die Weiterleitung von übermittelten Nachrichten unterstützt (TODO Routing-Konzept).

Auch wenn die Bildung einer eigenen Adressierungsebene für Leitstellenmodule (CRM-Ebene im Bild) nicht zwingend notwendig ist, unterstützt diese Ebene das Implementieren einer uniformen Routing-Funktion. Zusätzlich bringt direkte OID-Adressierung der Leitstellenmodule folgende Vorteile:

- es ermöglicht Umsetzung nützlicher technischer Funktionen, z.B. Kommunikationsdurchstich Modul zu Modul unter Verwendung von einheitlichen Adressierungs- und Routing-Funktionen.
- es unterstützt Vereinheitlichung von Identity-Management für UCRI-Systemkomponenten.

Die Systemkomponente Gateway stellt einen speziellen KT dar. Das Gateway wird am Übergang zu externen Systemen eingesetzt und stellt eine Gateway-Funktion bereit zum Mapping zwischen externe Quell- bzw. Zieladressen und internen OID. Ein externes System bekommt dabei einen entsprechend reservierten OID-Bereich (Unterbaum) und das Gateway bekommt die Wurzel-OID-Adresse dieses Unterbaums.

OID-Nomenklatur

OID-Nomenklatur soll Adressierung von KT über die staatliche Grenzen hinaus ermöglichen. Dabei steht jedem Land frei, ein eigenes Adressierungsschema unterhalb des Landes-OID-Unterbaums zu definieren. Wurzel-Adresse eines Landes-OID-Unterbaums ist wie folgt definiert:

- .1. (Beispiel: 1.2.3.1.276 für Deutschland)

Auch jedes Bundesland in Deutschland ist frei, ein eigenes Adressierungsschema unterhalb des Bundesland-OID-Unterbaums zu definieren. Wurzel-Adresse eines Bundesland-OID-Unterbaums ist wie folgt definiert:

- .1.276. (Beispiel: 1.2.3.1.276.5 für NRW)

Unterschiedliche Organisationen können auch unterschiedliche Adressierungsschemata verwenden. Wurzel-Adresse eines nPOLGA-OID-Unterbaums ist wie folgt definiert:

- .1.276.5.<Kennzahl von POL/nPOLGA, etc.> (Beispiel: 1.2.3.1.276.5.1 für nPOLGA in NRW)

Die OID-Adresse der einzelnen Kommunikationsteilnehmer (KT) wird nach dem amtlichen Gemeindeschlüssel (AGS) gebildet:

#	Ebene	Beispiel
1	Root-OID	1.2.3 - Festlegung in einem geschlossenen UCRI-System
2	Satzart	1 - Einzeladresse, 2 - Gruppe
3	Kennzahl des Landes	276 - Deutschland nach ISO 3166
4	Kennzahl des Bundeslandes	5 - für NRW nach AGS
5	Kennzeichnung von POL/nPOL Gefahrenabwehr (KRITIS, etc.)	Polizeidienststellen, Gesundheitsämter, Veterinärämter, etc. Definition folgt. Annahme für Beispiel: 1 - nPOLGA, 99 - Test/Pilot
6	Kennzahl des Regierungsbezirks	1 - Regierungsbezirk Düsseldorf nach AGS
7	Kennzahl des Landkreises oder der kreisfreien Stadt ("0" bei zentralen Einrichtungen auf der Regierungsbezirksebene)	58 - Landkreis Mettmann nach AGS
8	Gemeinde ("0" bei kreisfreien Städten)	28 - Stadt Ratingen, Stadtbezirk Ratingen nach AGS
9	Leitstellenmodul	1 - erstes Leitstellenmodul, fortlaufende Nummerierung von Leitstellenmodulen
10	Kommunikationsteilnehmer	1 - z.B. ELS, fortlaufende Nummerierung von KT

Beispiel OID Feuerwehr ELS in Ratingen: 1.2.3.1.276.5.1.1.58.28.1.1

Folgende OID wird zur Identifizierung der UCRI-Infrastruktur als Sender von technischen Quittungen (siehe UCRI2-App Technische Quittungen) festgelegt:

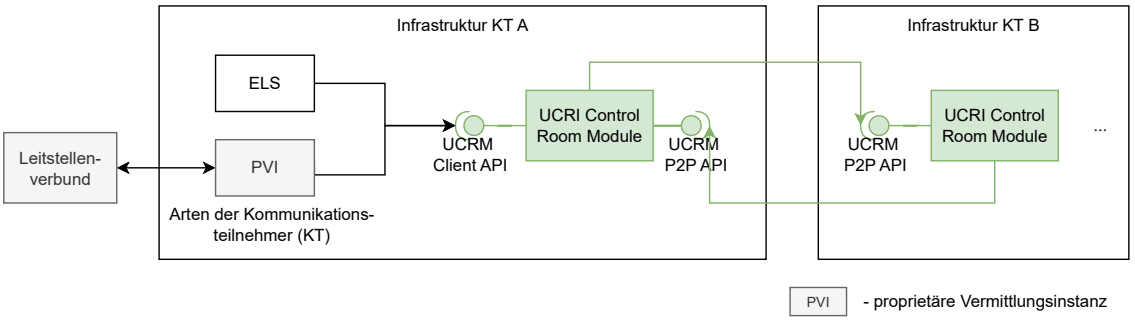
UCRI-Infrastruktur: 1.2.3.1.276.5.0.0.0.0.1.1

Vermittlungsebene Vermittlungsebene

UCRI Leitstellenmodul

Überblick

Das UCRI Leitstellenmodul (UCRI Control Room Module, UCRM) stellt die API bereit - die einzige Kommunikationsschnittstelle für verbundene Kommunikationsteilnehmer wie zum Beispiel Einsatzleitstellensysteme, proprietäre Vermittlungsinstanzen (PVI), sowie andere technische Systeme:



UCRM API Technologie

Das Ziel des UCRI-Systems ist eine Digitalisierung der menschlichen Kommunikation und impliziert (im Gegensatz zu etwa Steuerungsprozessen in M2M-Kommunikation) keine harten Echtzeit-Anforderungen. Aus diesem Grund wird für die Umsetzung der Kommunikationsschnittstelle UCRM API die technologische Variante REST API mit Polling festgelegt. REST API mit Polling von Nachrichten hat folgende Vorteile:

- einfach zu implementieren
- einfach zu konsumieren

Um die Auswirkung des Pollings auf die Systemreaktionszeit bei Meldungsaustausch (die maximale Zeit zwischen den Meldungssende- und Meldungsempfangszeitpunkten) zu minimieren, kann bei Meldungsaustausch ein Long Polling vorgesehen werden.

Protokoll

Wegen der Anforderung zur sicheren Nachrichtenzustellung muss eine technische Nachrichtenempfangsbestätigung in der UCRM API vereinbart werden.

Das Prinzip der sicheren Nachrichtenzustellung ist ein E2E-Prinzip, das auch die Empfangslogik bis zum Persistieren der Nachrichtendaten auf der Seite des KT-Systems einschließt. Um die Ausfälle in dieser Empfangslogik zu kompensieren wird ein zweistufiges Protokoll für Meldungsempfang vereinbart:

1. Meldungen abfragen - idempotent, kann mehrmals wiederholt werden mit dem gleichen Ergebnis.
2. Meldungsempfang bestätigen - idempotent, kann mehrmals wiederholt werden mit dem gleichen Ergebnis. Bestätigte Meldungen werden aus der Vermittlungsebene verworfen und stehen beim erneuter Meldungsabfrage nicht mehr zur Verfügung.

Zum Signalisieren der Nachrichtenzustellung werden technische Quittungen für eine gesendete Nachricht implementiert (positive sowie negative Zustellbestätigungen).

Ein Nachrichtensender bekommt folgende Quittungen:

1. Zustellbestätigung: nachdem der Empfänger die Entgegennahme einer Nachricht bestätigt hat
2. Benachrichtigung über fehlgeschlagene Zustellung: nachdem ein für die Nachrichtenzustellung vordefiniertes Timeout verstrichen ist, ohne dass der Empfänger die Entgegennahme einer Nachricht bestätigt hat

Die technischen Quittungen sind in Form eines JSON-Schemas auf der untergeordneten Seite beschrieben.

Empfehlungen zur technischen Umsetzung

Der Client muss die Schnittstelle periodisch abfragen, um Nachrichten zu empfangen. Das klassische Polling-Intervall ist dabei ein Maß zwischen Systemreaktionszeit (die maximale Zeit zwischen den Sende- und Empfangszeitpunkten) und Systemauslastung. Ein Polling-Intervall in Sekundenbereich (3 - 5 Sekunden) scheint optimal zu sein. Um die Auswirkung des Pollings auf die Systemreaktionszeit bei Meldungsaustausch zu minimieren, wird Verwendung eines Long Polling empfohlen.

Polling bei Nachrichtenabfragen

Folgende Schleife ist bei Nachrichtenabfragen bei Polling zu empfehlen:

1. Nachrichten abfragen mit Angabe maximaler Nachrichtenzahl Nmax
2. Nachrichten verarbeiten
3. Nachrichtenempfang bestätigen
4. Ist die Anzahl von empfangenen Nachrichten gleich Nmax - sofort zum Schritt 1 übergehen
5. Konfigurierte Pause einlegen

Wichtig: bei Programmausfällen zwischen Schritten 2 und 3 kann es zum wiederholten Empfang von gleichen Nachrichten kommen. Die Client-Logik muss somit mit Nachrichtenduplikaten umgehen können, z.B. unter Berücksichtigung von eindeutigen Nachrichten-ID.

Long Polling bei Nachrichtenabfragen

Folgende Schleife ist bei Nachrichtenabfragen bei Long Polling zu empfehlen:

1. Nachrichten abfragen mit Angabe maximaler Nachrichtenzahl Nmax und maximaler Wartezeit Tmax
2. Nachrichten verarbeiten
3. Nachrichtenempfang bestätigen

Wichtig: bei Programmausfällen zwischen Schritten 2 und 3 kann es zum wiederholten Empfang von gleichen Nachrichten kommen. Die Client-Logik muss somit mit Nachrichtenduplikaten umgehen können, z.B. unter Berücksichtigung von eindeutigen Nachrichten-ID.

UCRM REST API

UCRM API Design verwendet REST API Design Richtlinien erarbeitet bei TM Forum.

UCRM API Spezifikation verwendet Standards erarbeitet bei der OpenAPI Initiative.

UCRM API ist in folgende fachliche Bereiche aufgeteilt:

- KT-Registry - Abfragen von Eigenschaften der registrierten Kommunikationsteilnehmer
- Messaging - Versenden und Empfangen von Nachrichten
- Info - Die Info API liefert Informationen über die Version und den Betreiber der Schnittstelle

Berechtigungskonzept

Die UCRM API wird sowohl KT-seitig als auch für die Inter-CRM-Kommunikation verwendet.

Es wird ein einfaches Berechtigungskonzept verwendet, das folgende Rollen vorsieht:

- KT - für die Kommunikation von KT-Systemen zu UCRM
- UCRM - für Inter-CRM-Kommunikation

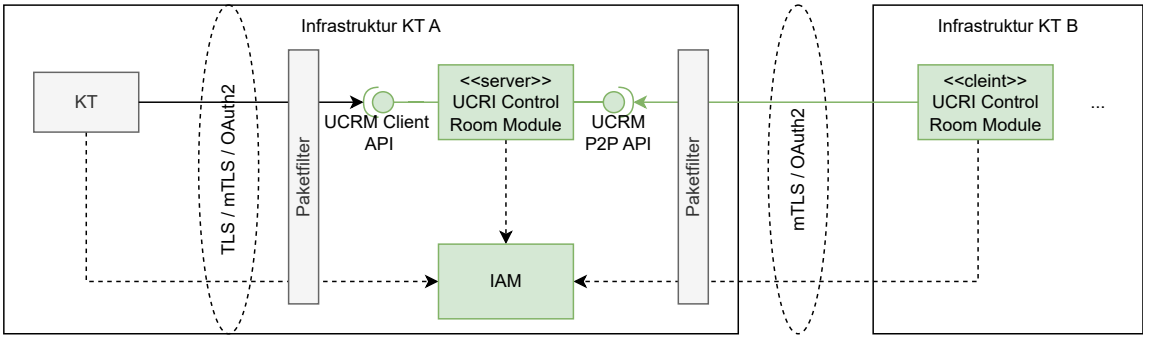
Die Rolle wird an die API-Implementierung mittels eines HTTP-Headers übergeben.

Vermittlungsebene Vermittlungsebene

UCRI2 Kommunikationsprotokoll

UCRI2 unterscheidet zwei Kommunikationsdomäne, siehe Abbildung:

- Kommunikation zwischen einem Leitstellensystem (KT-System) und einem UCRI Leitstellenmodul (UCRM). Diese Kommunikation findet anhand der UCRM Client API statt und erfolgt typischerweise innerhalb einer durch einen Leitstellenbetreiber kontrollierten Infrastruktur.
- Kommunikation zwischen zwei UCRM. Die Inter-UCRM-Kommunikation verwendet UCRM P2P API. Diese Kommunikation kann über das öffentliche Internet stattfinden und braucht dementsprechend besondere Sicherheitsmaßnahmen.



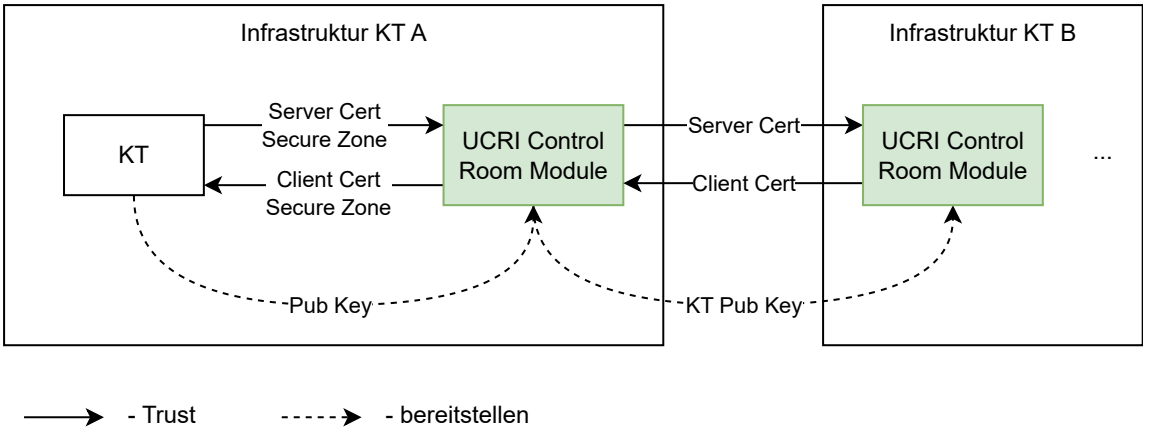
Die Infrastruktur eines KTs sollte eine P-A-P-Struktur aufweisen, die aus Paketfilter als Trennung zu vertrauenswürdigen internen Systemen (Leitstelle), Application-Layer-Gateway (UCRM) und Paketfilter als Trennung zu nicht vertrauenswürdigem Netz (Internet) besteht. Vgl. Netzarchitektur und -design - BSI.

UCRM API Implementierung verwendet in beiden Kommunikationsdomänen OAuth 2.0 mit Client Credentials Grant Type zur sicheren Authentifizierung und Autorisierung von Clients auf der Applikationsebene.

Trust Konzept

Zwischen zwei UCRM wird mTLS als Sicherung der Inter-UCRM-Kommunikation verwendet. Sowohl Client als auch Server bauen eine gegenseitige Vertrauensbeziehung über digitale Zertifikate auf. Dabei kann der Paketfilter auf der Seite des nicht vertrauenswürdigen Netzes die mTLS-Verbindung terminieren. Dazu kann der Paketfilter eine Liste von zugelassenen Domainnamen beinhalten (White Listing) und somit die Kommunikation auf eine geschlossene Gruppe der Teilnehmer beschränken.

Die Kommunikation zwischen einem KT-System und dem UCRM-Modul kann je nach lokalen Sicherheitsanforderungen wahlweise per TLS oder mTLS erfolgen.



Die Vertrauensbeziehungen zwischen den Systemkomponenten bei der Client/Server-Kommunikation basieren dabei auf folgenden Mechanismen (siehe Abbildung):

1. Server-Authentifizierung. Die Vertrauensbeziehung zu UCRM als Server basiert auf dem Domainname (DNS) des Servers und wird durch Server-Zertifikat abgesichert. Der Domainname des Servers ist im Server-Zertifikat verankert.
2. Client-Authentifizierung. Die Vertrauensbeziehung zu KT oder UCRM in der Client-Rolle basiert auf der Object Identifier (OID) des Clients (siehe Adressierungskonzept) und wird durch Client-Zertifikat abgesichert. Die OID des Clients ist im Client-Zertifikat verankert.
3. Sowohl Client als auch Server haben eine Liste vertrauenswürdiger Zertifizierungsstellen (CAs), sogenannte Root-CAs, die als Vertrauensanker dienen. Die ausgestellten Zertifikate von Client und Server müssen von einer CA signiert sein, die jeweils in der vertrauenswürdigen Liste des Gegenübers enthalten ist.

Anmerkung 1: In den konkreten Kundensituationen können unterschiedliche Mechanismen des Zertifikatsmanagements umgesetzt werden - basierend sowohl auf zentraler Authority als auch auf spezifischen Vereinbarungen zwischen einzelnen KTs.

Anmerkung 2: Als ALternative kann das Vertrauen zwischen KT-Systemen und dem UCRM in einer durch einen Leitstellenbetreiber kontrollierten Infrastruktur durch das Bilden von einer Sicherheitszone auf der Netzwerkebene hergestellt werden.

Um den sichernen Ursprung und unverfälschten Inhalt von Applikationsmeldungen zu garantieren, können die Meldungen durch Sender-KT signiert werden. Zur Signaturvalidierung erstellt der KT ein kryptographisches Schlüsselpaar und stellt sein Public Key dem lokalen UCRM-Modul über sicheren Kanal bereit. Bei der Inter-UCRM-Kommunikation wird das KT Public Key an die fremden UCRM mittels UCRM P2P API übermittelt und steht anschließend den potentiellen Meldungsempfängern über die UCRM Client API zur Signaturvalidierung zur Verfügung.

Nachrichtenübermittlung

Die ausgehenden Nachrichten übergibt das KT-System an das UCRM-Modul m.H.v. Client API-Endpunkt /send. Die Nachrichten werden dann durch lokales UCRM mittels P2P API-Endpunkt /send gegen Empfänger-UCRM gepuscht. Für das Handling von Nichtverfügbarkeit der Partner-UCRMs wird in dem lokalen UCRM ein Ausgangspuffer implementiert.

KT-Register

Jeder UCRM baut einen lokalen Cache des KT-Registers durch regelmäßige Aufrufe der KT-Register API /registry an allen bekannten Partner-UCRM.

Die Umsetzung des UCRM API-Endpunktes /registry unterscheidet sich in beiden UCRM API:

- Der Client API-Endpunkt /registry liefert alle bekannten KT.
- Der P2P API-Endpunkt /registry liefert nur eigene KT.

Da die Umgebung dynamisch ist, d.h. neue KT können dazukommen, existierende KT können abgebaut werden, sollen KT-Register-Abfragen sowohl KT-seitig als auch in der Inter-UCRM-Kommunikation regelmäßig durchgeführt werden. Empfehlung: mindestens 1 Mal pro Stunde, maximal alle 5 Minuten.

E2E Verschlüsselung und Datenintegrität

Als Richtlinie für die Auswahl kryptographischer Verfahren für die Verschlüsselung und Signieren von Nachrichten dient die BSI Technische Richtlinie (https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102.pdf?__blob=publicationFile&v=9).

Das asymmetrische kryptographische RSA-Verfahren RSASSA-PKCS1-v1_5 (RFC 3447: Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1) wird sowohl zum Verschlüsseln als auch zum digitalen Signieren der Nachrichten verwendet.

Das Verfahren verwendet ein für jeden KT generiertes Schlüsselpaar, bestehend aus einem privaten Schlüssel und einem öffentlichen Schlüssel. Der private Schlüssel wird im Keystore des KT-Systems gespeichert.

Die öffentlichen Schlüssel werden über das KT-Register als Teil der verfügbaren KT-Daten in Form von JSON Web Key (JWK, RFC 7517: JSON Web Key (JWK)) ausgetauscht.

Verschlüsselung

Da jegliche Kommunikation zwischen KT und UCRM und zwischen UCRMs TLS verschlüsselt stattfindet, kann es auf eine E2E-Veschlüsselung des Nachrichteninhaltes verzichtet werden. Um die strengeren Sicherheitsanforderungen vor allem bei der Kommunikation über das Internet zu erfüllen, kann es später jedoch sinnvoll sein, den Nachrichteninhalt zusätzlich Ende-Zu-Ende, d.h. zwischen dem Sender-UCRM und dem Empfänger-UCRM oder sogar zwischen den KTs (in diesem Fall ohne Möglichkeit, die Meldungen durch UCRM validieren zu lassen), zu verschlüsseln.

Für die Verschlüsselung wird dann das RSA-Verfahren RSASSA-PKCS1-v1_5 (RFC 3447: Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1) verwendet.

Das konkrete Verschlüsselungsverfahren wird bei Bedarf später spezifiziert.

Signaturverfahren

NEU von @PZernicke Nachrichtensignaturen stellen sicher, dass eine Nachricht während der Übertragung nicht verändert wurde und dass sie von der angegebenen Quelle stammt. Hierzu sind zwei Schritte notwendig: Die Erzeugung eines eindeutigen Hash-Wertes für die Nachricht sowie die Signierung dieses Hashwertes durch das sendende System.

- Für das Hashing wird die Nachricht zuerst gemäß JSON Canonicalization Scheme (JCS, RFC 8785: JSON Canonicalization Scheme (JCS)) in eine kanonische Form gebracht und diese kanonische Form dann mit SHA3-256 gehasht.
- Für das Signieren und die Prüfung der Signatur wird das Verfahren nach dem IETF Standard JSON Web Signature (JWS, RFC 7517: JSON Web Key (JWK)) in der Variante Compact JWS in Kombination mit dem RSA-

Verfahren RSASSA-PKCS1-v1_5 verwendet. Diese Schritte werden in den folgenden Unterkapiteln genauer dargestellt.

Hashing der Nachrichten

Vor der Anwendung des JCS erstellt das signierende System eine Kopie der Nachricht, in der nur die folgenden Felder enthalten sind:

- "source"
- "destinations"
- "payload"

Somit werden alle anderen Felder nicht beim Hashing berücksichtigt. Dies ist notwendig, da andere Felder freiwillig sind oder vom UCRM gesetzt werden, falls sie nicht vom Client gesetzt wurden. Als Hashing-Algorithmus kommt SHA3-256 zum Einsatz (Use of the SHA3 One-way Hash Functions in the Cryptographic Message Syntax (CMS)).

Signierung der Nachrichten

Als JWS-Header kommt ein Header mit Angabe des RSA-256-Algorithmus zum Einsatz:

```
{
  "typ": "UCRI_PLAIN",
  "alg": "RS256"
}
```

Der berechnete Hashwert der Nachricht wird als JWS-Payload verwendet. Dieser wird mit dem privaten Schlüssel des Nachrichtensenders signiert. Das Ergebnis (digitale Signatur) wird in Form einer JSON Web Signature (JWS, RFC 7517: JSON Web Key (JWK)) im Feld signatur übertragen:

```
BASE64(Header) || '.' || BASE64(Hash) || '.' || BASE64(Signatur)
```

Prüfung der Signaturen

Um die übermittelte JWS zu prüfen, muss das empfangende System oder UCRM wiederum

- den Hashwert der empfangenen Nachricht berechnen
- die übermittelte Signatur auf Gültigkeit prüfen
- den signierten Hashwert (JWS-Payload) auf Gleichheit mit dem Hashwert der empfangenen Nachrichten prüfen

Die Ermittlung des Hashwertes erfolgt gemäß "Hashing der Nachrichten".

Die Signatur wird aus dem "signature"-Feld der Nachricht extrahiert und gegen den im KT-Register hinterlegten "key" (öffentlichen Schlüssel) validiert.

ENDE NEU von @PZernicke

Nachrichtensignatur stellt sicher, dass eine Nachricht während der Übertragung nicht verändert wurde und dass sie von der angegebenen Quelle stammt. Für das Signieren wird das Verfahren nach dem IETF Standard JSON Web Signature (JWS, RFC 7517: JSON Web Key (JWK)) in Kombination mit dem RSA-Verfahren RSASSA-PKCS1-v1_5 verwendet.

Mit Hilfe eines Headers wird das Verschlüsselungsverfahren beschrieben:

```
{
  "typ": "UCRI_PLAIN",
```

```
  "alg": "RSA256"
}
```

Die zu signierende Meldung ist ein JSON-Objekt. Es gibt im Allgemeinen zwei Schritte, um JSON zu signieren:

1. Serialisierung und Hashing
2. Signatur erstellen

Serialisierung und Hashing

Zuerst muss sichergestellt werden, dass das JSON-Objekt in einer standardisierten Form vorliegt (z.B. durch Canonicalization), damit die Signatur später exakt verifizierbar ist. Dazu wird der Standard JCS / RFC 8785 verwendet. Als Ergebnis entsteht eine kanonische serialisierte Form einer Meldung, die direkt gehasht werden kann.

Folgende Meldungsattribute werden bei der Serialisierung berücksichtigt:

- source
- destinations
- payload

Als Hash-Funktion wird das Verfahren SHA3-256 verwendet (Use of the SHA3 One-way Hash Functions in the Cryptographic Message Syntax (CMS)).

Signatur erstellen

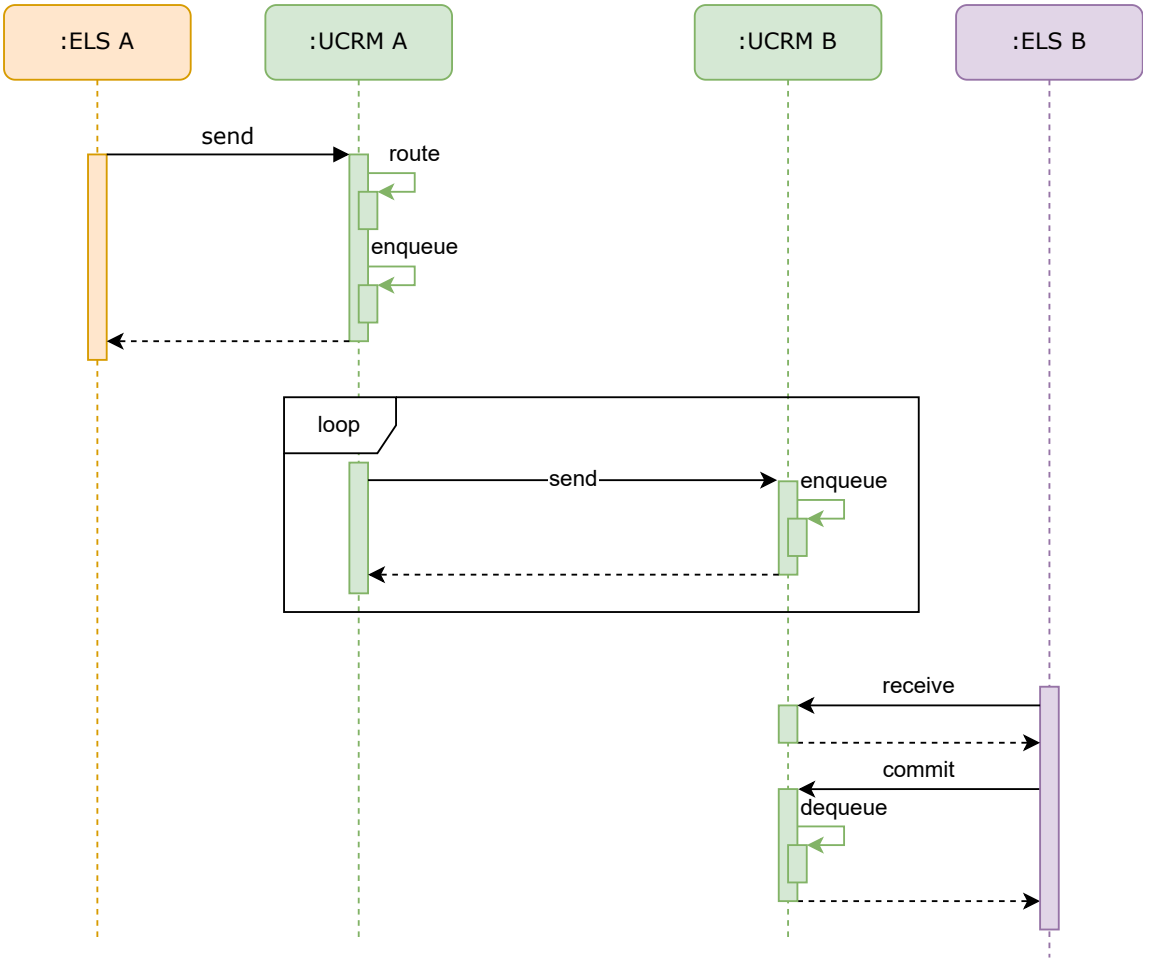
Der errechnete Hash-Wert wird mit dem privaten Schlüssel des Nachrichtensenders verschlüsselt. Das Ergebnis (digitale Signatur) wird als JSON Web Signature in Compact Serialization Form (JWS, RFC 7517: JSON Web Key (JWK)) im Feld signatur übertragen.

Der Empfänger der Nachricht (UCRM) entschlüsselt die digitale Signatur mit dem öffentlichen Schlüssel des Senders und vergleicht den gewonnenen Hash mit dem selbst erstellten Hash der empfangenen Nachricht.

Wenn die Hashes übereinstimmen, kann der Empfänger sicher sein, dass die Nachricht nicht verändert wurde und dass sie tatsächlich vom angegebenen Absender stammt.

Das konkrete Verschlüsselungsverfahren samt entsprechende Konfigurationsparameter können auch aus der UCRM API Version abgeleitet werden.

Zustellung von Nachrichten



Für die Zustellung von Nachrichten können unterschieldiche Routing-Algorithmen umgesetzt werden:

- Routing durch Adressierungshierarchie
- Routing-Tabellen

Validierung von Meldungen

Meldungen werden in UCRM m.H.v. Meldungs-Schemata validiert. Die Validierung erfolgt synchron beim Senden. KT-Register liefert Auskunft über die durch KT unterstützten Schemata.

Vermittlungsebene

UCRI2 Anwendungen

Detaillierte Beschreibung der UCRI2 Anwendungen befindet sich in dem Verzeichnis docs/apps. Hierbei gibt das Dokument "UCRI2 App im Überblick" einen Überblick über die verschiedenen Apps und die PDF-Dateien in den Unterordnern beinhalten die detaillierte Dokumentation für verschiedenen Apps.