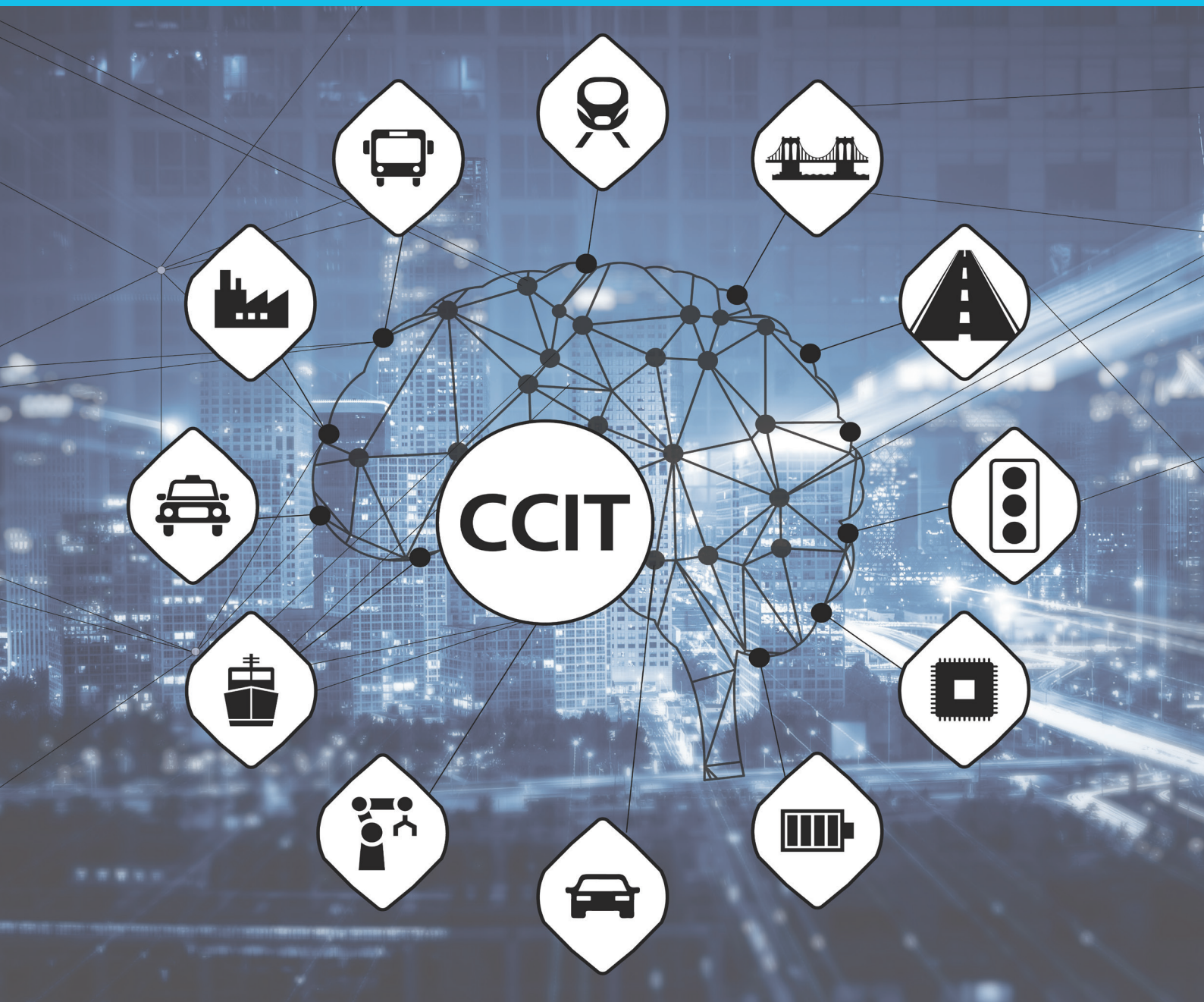
Fraunhofer

Fraunhofer Cluster of Excellence Cognitive Internet Technologies

TrackChain: Blockchain-basiertes Track & Trace in Supply Chains mit dem Industrial Data Space

Mark Gall



Problemstellung

Supply Chains in modernen Produktionsprozessen bestehen heutzutage aus kompletten Wertschöpfungsketten, die eine Vielzahl von Lieferanten, Herstellern, Händlern, Zollbehörden, bis hin zu Logistik- und Finanzdienstleistern umspannen. Die Nachverfolgung von Güterströmen über die gesamte Supply Chain hinweg ist derzeit nahezu unmöglich, ganz zu schweigen von einer unternehmensübergreifenden Aufzeichnung von Logistik-Ereignisse (Supply Chain Events). Dies stellt Unternehmen an mehreren Punkten der Supply Chain vor Herausforderungen: produzierende Unternehmen können die Lieferungen ihrer Produkte nicht bis zum Endkunden nachverfolgen und verlieren so die Möglichkeit, präzise Daten bezüglich Lieferzeiten, Verbrauch und möglichen Transportschäden zu erheben. Logistik-Unternehmen müssen den ordnungsgemäßen Transport nachweisen und etwaige Verluste ermitteln können. Darüber hinaus sind sie bestrebt, Endkunden eine lückenlose Sicht auf den aktuellen Sendungsstatus zu liefern.

Aktuell werden Supply-Chain-Event-Management-Systeme (SCEM) eingesetzt, um individuelle Logistik-Events zu erfassen und zu einer einheitlichen Sicht zusammenzufassen. Dies funktioniert allerdings nur, soweit alle Schritte der Logistik-Kette an das System angeschlossen sind, was bei unternehmens- und länderübergreifenden Supply Chains nicht gegeben ist. Solche SCEM-Systeme um die Möglichkeit einer beliebig erweiterbaren, dezentralen und vor allem vertrauenswürdigen Infrastruktur zur Erfassung von Supply-Chain-Events zu erweitern, ist das Ziel von TrackChain.

Was ist TrackChain?

TrackChain ist eine Blockchain-basierte Infrastruktur für das »Committment« von Events und die kontrollierte Freigabe von Event-Daten für berechtigte Clients. Unter einem »Committment« wird das irreversible Speichern eines Events im Zeitverlauf verstanden. Das bedeutet, dass jedes in TrackChain »comittete« Event nachvollziehbar und nicht-abstreitbar gespeichert wird. Auf diese Weise lassen sich - auch unabhängig von Supply Chains – beliebige Sequenzen von Ereignissen nachvollziehbar und manipulationsgeschützt aufzeichnen. Die möglichen Einsatzszenarien reichen von Audit Logs, die bei der Administration eines technischen Systems angelegt werden, bis hin zur Umsetzung von »Clearing Houses«, die Geschäftsvorfälle wie bspw. abrechenbare Transaktionen aufzeichnen, bis eben hin zur Nachverfolgung von Events in Supply Chains.

Eine Vielzahl ähnlicher Blockchain-basierter Lösungen existiert bereits. Sie haben jedoch in der Regel das Problem, dass die in der Blockchain gespeicherten Daten allen Teilnehmern des Systems zugänglich sind und nicht mehr gelöscht werden können. Diese Kerneigenschaft der Blockchain ist gerade gewünscht, um nicht-abstreitbare Ereignisketten zu erstellen, steht jedoch in Konflikt mit Datenschutzanforderungen und dem Bedarf, geschäftskritische Daten vertraulich zu halten. Nicht erst seit Inkrafttreten der EU-DSGVO müssen personenbezogene Daten ausschließlich zweckgebunden verarbeitet und auf Verlangen des Benutzers gelöscht werden, was eine Speicherung solcher Daten in einer Blockchain praktisch ausschließt. Einige Blockchain-Lösungen versuchen dieses Problem durch »Permissioned Ledgers« zu lösen – also Blockchains, auf die nur berechtigte Teilnehmer zugreifen können.

TrackChain löst das Problem durch eine spezielle Verschlüsselung der comitteten Events. Sämtliche Event-Inhalte werden mittels sogenannter Attribute-Based Encryption verschlüsselt, bevor sie in die Blockchain gespeichert werden. In der klassischen Public-Key-Kryptografie, werden Daten für einen spezifischen Empfänger verschlüsselt. Dies hat jedoch den Nachteil, dass dessen (öffentlicher) Schlüssel im Voraus bekannt sein muss. Im Gegensatz dazu ist Attribute-based Encryption eine Form der Public-Key-Kryptografie bei der Daten für den Inhaber eines (oder mehrerer) Attribute verschlüsselt werden, die jedoch selbst zu diesem Zeitpunkt weder einen Schlüssel benötigen, noch überhaupt bekannt sein müssen. Erst wenn der Inhaber eines Attributes sich zu einem späteren Zeitpunkt als solcher ausweisen kann, wird für ihn ein spezifischer privater Schlüssel erzeugt, mit dem er die für sein Attribut verschlüsselten Einträge in der Blockchain lesen kann. Dadurch lassen sich Konzepte der traditionellen Zugriffskontrolle, für die bislang immer ein vertrauenswürdiger Server die Entscheidung über den Zugriff treffen musste, in kryptographische Operationen abbilden, die sich auch auf dezentral gespeicherte Daten wie bspw. in einer Blockchain anwenden lassen.

Das TrackChain-System verbirgt die Komplexität der Blockchain und der Kryptografie hinter einer einfachen REST-API, über die (Logistik-) Events »comitted« werden können. Über eine zweite REST-API können berechtigte Teilnehmer die Daten in der Blockchain einsehen, die für die bestimmt sind – die Blockchain selbst kann dabei vollkommen öffentlich und dezentral betrieben werden. Derzeit ist TrackChain auf Basis von Multichain implementiert – die Wahl der Blockchain ist jedoch grundsätzlich frei. Ein alternatives Setup von TrackChain wurde bereits auf Basis von Hyperledger betrieben. Die Weiterentwicklung TrackChain++ wird auf Quorum entwickelt.

Einbinden von TrackChain in das IDS Ecosystem

TrackChain stellt eine REST-API bereit, über die Logistik-Events comitted werden können und die wie jede andere REST-Schnittstelle verwendet wird. Die Dokumentation der Schnittstelle ist über Swagger.io verfügbar.

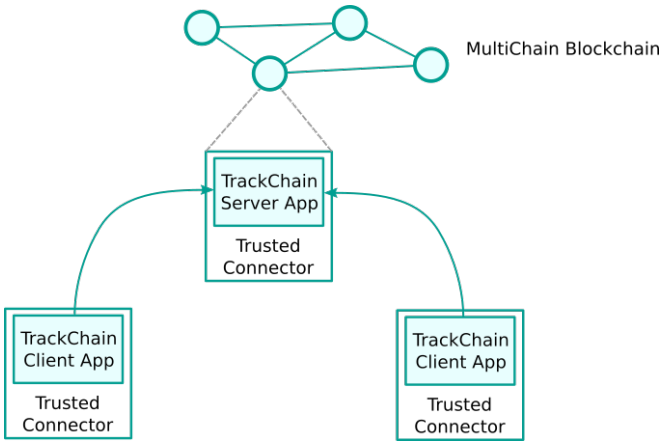


Abb. 1: Integration in den IDS

Committen von Events mit Curl

Ein Logistik-Event kann bspw. folgendermaßen comittet werden:

```
curl --header „Content-Type: application/json“ --request POST --data
‚{ „location“: „52.520008, 13.404954“, „event_type“: „incoming“,
„timestamp“: 1548835195, „eta“: 0 }‘ http://trackchain.example.
com:8000/trackchain/events
```

Committen von Events mit Camel

Alternativ kann eine Camel-Route verwendet werden, um parallel zur Datenverarbeitung Events in die TrackChain zu comitten:

```
<?xml version=“1.0“ encoding=“UTF-8“?>
<blueprint xmlns=“http://www.osgi.org/xmlns/blueprint/v1.0.0“
    xmlns:xsi=“http://www.w3.org/2001/XMLSchema-instance“
    xmlns:camel=“http://camel.apache.org/schema/blueprint“
    xmlns:camel-cxf=“http://camel.apache.org/schema/blueprint/cxf“
    xsi:schemaLocation=“http://www.osgi.org/xmlns/blueprint/v1.0.0
http://www.osgi.org/xmlns/blueprint/v1.0.0/blueprint.xsd http://
camel.apache.org/schema/blueprint http://camel.apache.org/schema/
blueprint/camel-blueprint.xsd“>

    <!-- Routes -->
    <camelContext xmlns=“http://camel.apache.org/schema/blueprint“>
        <route id=“demo-rest-route“>
            <from uri=“activemq:queue:demo.rest“/>
            <setHeader headerName=“Content-Type“
inheritErrorHandler=“true“ id=“setHeader3“>
                <constant>application/json</constant>
            </setHeader>
            <setHeader headerName=“Exchange.HTTP_METHOD“>
                <constant>POST</constant>
            </setHeader>
            <toD uri=“http://localhost:8181/cxf/person“ />
        </route>
    </camelContext>
</blueprint
```

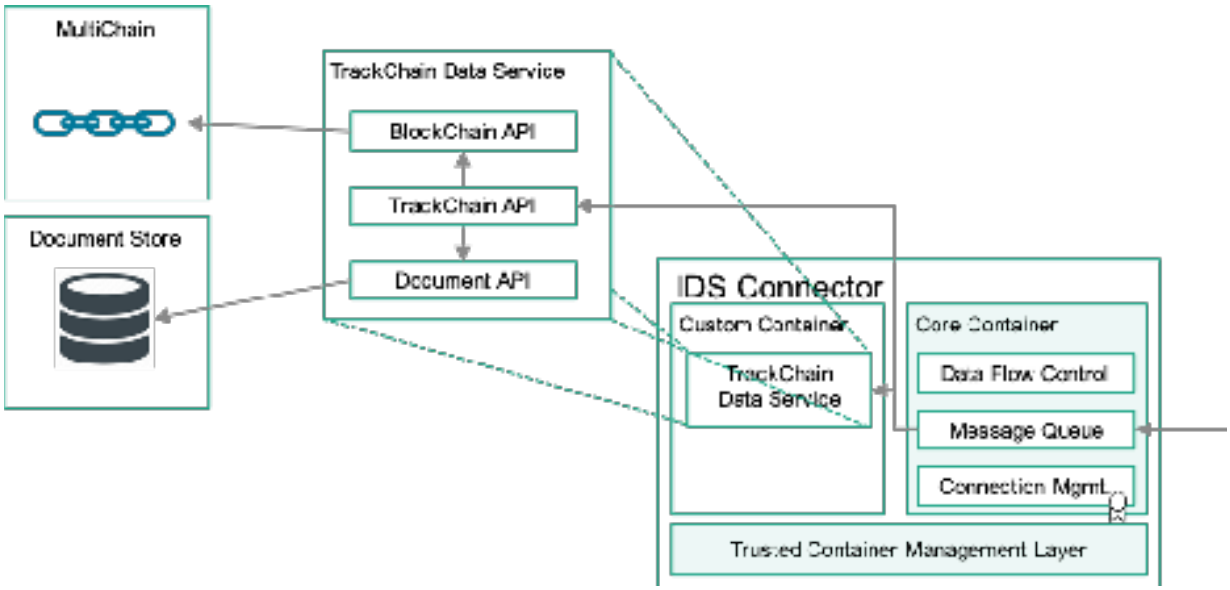
Betreiben eines TrackChain-Knotens auf dem Trusted Connector

Der Industrial Data Space stellt die Infrastruktur für den vertrauenswürdigen Datenaustausch zwischen Unternehmen bereit und bietet mit dem Trusted Connector eine Ausführungsumgebung für »Apps«. Apps werden in Form von Containern bereitgestellt und können innerhalb des Trusted Connectors in den IDS-Datenfluss mit einbezogen werden, indem bspw. entsprechende Routen zwischen einem externen IDS-Endpunkt und der App mittels Apache Camel eingerichtet werden.

Ein TrackChain-Client wird als eine solche App bereitgestellt und kann damit im Trusted Connector betrieben werden. Je nach Anbindung der TrackChain-Client-App in den Datenfluss des jeweiligen Connectors, kann so entweder ein IDS-Dienst »Blockchain-as-a-Service« erstellt werden, der es allen anderen IDS-Teilnehmer erlaubt, die TrackChain-API über das IDS-Protokoll zu nutzen. Alternativ kann der TrackChain-Client direkt in den internen Datenfluss zwischen Anwendungen, die im Trusted Connector laufen, einbezogen werden. In diesem Fall würde der Connector TrackChain als »Audit-Log« verwenden, um kritische Ereignisse und Nachrichten innerhalb seiner Anwendungen zu dokumentieren (d.h. in die Blockchain zu comitten).

Trackchain HMI Demonstrator

Gemeinsam mit dem Anwendungsprojekt Warenverfolgung entstand ein Demonstrator, der die Nutzung von TrackChain im Kontext eines Logistik-Use-Cases zeigt. Die Architektur des Demonstrators folgt den oben beschriebenen Mechanismen für die Einbindung in das IDS Ecosystem.



Die Abbildung zeigt einen Trusted Connector, in welchem der TrackChain-Demonstrator als Anwendung in einem Container betrieben wird. Der TrackChain-Demonstrator selbst besteht aus drei REST-Schnittstellen, die miteinander interagieren:

- **BlockChain-API:** Die BlockChain-API ist zuständig für die Kommunikation mit der BlockChain. Wann immer von der BlockChain gelesen oder in die BlockChain geschrieben werden soll, wird die BlockChain API aufgerufen. Die BlockChain-API kapselt die Blockchain vom Rest der Anwendung ab und ist der einzige Teil, der bei einem Wechsel der Blockchain geändert werden muss.

- **Document-API:** Die Document-API ist zuständig für das Ver- und Entschlüsseln von Daten mit Attribute-based Encryption. Verschlüsselte Daten können in einem öffentlichen Document Store gelagert werden. Umgesetzt wird die Document-API mit der r-abe von AISEC

- **TrackChain-API:** Die TrackChain-API ist zuständig für das Protokollieren von Events des Anwendungsfalles, z.B. der Übergabe eines Paketes von einem Logistikdienstleister zu einem anderen. Hierfür nutzt die Trackchain-API die BlockChain-API, um Hashes der Events in einer BlockChain zu speichern. Die eigentlichen Daten des Anwendungsfalles, z.B. Empfängeradresse eines Pakets, wird über die Document API verschlüsselt im Document Store abgelegt. Ein Verweis zu den Daten wird beim Loggen des Events in der BlockChain angefügt.

Abb. 2:
Architektur TrackChain

Attribute-Based Encryption

Attribute-Based Encryption (ABE), genauer Ciphertext-Policy Attribute-Based Encryption (CP-ABE), beschreibt eine Klasse von Verschlüsselungsverfahren, die sich sehr gut für die Umsetzung von feingranularer Zugriffskontrolle eignen. Dabei muss keinem sog. Monitor vertraut werden, damit dieser die Zugriffskontrolle durchführt. Die Zugriffskontrolle wird implizit dadurch erreicht, dass die verschlüsselten Ciphertexte nur von berechtigten Personen entschlüsselt werden können. Dies wird bereits bei der Verschlüsselung sichergestellt. Im Gegensatz zu einer Verschlüsselung mit einem klassischen Public-Key-Verfahren, bei dem der öffentliche Schlüssel des Empfängers für die Verschlüsselung genutzt wird, wird bei ABE ein systemweiter öffentlicher Schlüssel für alle Benutzer genutzt. Zusätzlich wird eine Zugriffspolicy aus Attributen benötigt, die bei der Verschlüsselung in den Ciphertext eingebettet wird. Ein Nutzer kann dann einen Ciphertext entschlüsseln, wenn die Attribute, die er besitzt, die Zugriffspolicy erfüllt, die bei Verschlüsselung in den Ciphertext eingebettet worden ist.

Im Trackchain Demonstrator wird ABE transparent für die Nutzer eingesetzt. Daten, die an die Trackchain-API im Plaintext geschickt werden, werden verschlüsselt und die Ciphertexte werden zur Speicherung in Datenbank und Blockchain weitergereicht. Bei der Abfrage von Daten muss die Abfrage einen Identifier enthalten, der eine Zuordnung von Attributen zu diesem Identifier ermöglicht. Im Demonstrator erfolgt dies durch einen OAuth Flow. Mit Hilfe des Identifiers wird vom Keyserver der Schlüssel geholt und der Ciphertext entschlüsselt.

Die Daten eines Pakets werden nicht alle mit der gleichen Zugriffspolicy verschlüsselt. Vielmehr wird für einen Datentyp zu Beginn eine Struktur definiert, die den unterschiedlichen Teilen eines Datentyps unterschiedliche Zugriffspolicies zuweist. Dadurch kann der Fall entstehen, dass ein Nutzer bei der Abfrage eines Datensatzes nur Teile eines Datensatzes erhält, da er nur bestimmte Teile entschlüsseln konnte.

Datenspeicherung

Die Speicherung der Daten erfolgt nicht vollständig in der Blockchain. Die Ciphertexte der Daten werden im Demonstrator in einer Datenbank gespeichert. Durch die Verwendung von ABE wäre hier auch ein öffentlicher Fileserver denkbar. Hashes der Ciphertexte werden in der Blockchain gespeichert. Diese Trennung hat den Vorteil, dass weniger Daten in der Blockchain gespeichert werden müssen, denn jedes Byte in der Blockchain kostet potenziell Geld. Hinzukommt, dass die Daten zwar nachvollziehbar und manipulationsgeschützt sind, aber nicht von allen Benutzern der Blockchain eingesehen werden können.

Sensordaten

Im Demonstrator werden zwei verschiedene Sensoren als Datenquellen genutzt: Bluetooth-Sensoren in den Paketen und RFID-Tags für die Simulation des Status bei der Warenauslieferung. Die Sensordaten der Bluetooth-Sensoren werden ständig gesendet und über die Schnittstelle des Trusted Connectors empfangen. Da in diesem Use Case aber nicht die Rohdaten der Sensoren, sondern nur das Überschreiten von Schwellwerten der Sensoren aufgezeichnet werden sollte, werden die Rohdaten im Trusted Connector ausgewertet und nicht an TrackChain geschickt. Stellt der Trusted Connector bei der Auswertung fest, dass ein Schwellwert überschritten worden ist, so schickt er ein Event an die TrackChain-API, die diesen Vorgang protokolliert.

Die RFID Tags senden ihrerseits nur dann Informationen an die TrackChain API, wenn sie ein Paket in der Nähe detektieren. Diese Informationen protokolliert die TrackChain-API ebenso.

Darstellung

Für die Darstellung der Informationen gibt es am Demonstrator selbst vier kleine Tablets, die die Sichten unterschiedlicher Stakeholder auf die in der TrackChain gespeicherten Daten darstellen. Angezeigt werden die Daten des aktuell im Versand befindlichen Pakets. Dabei authentifizieren sich die Tablets als unterschiedlicher Nutzer an der TrackChain-API und erhalten nur die Daten, die für diesen Nutzer entschlüsselt werden konnten.

Weiterentwicklung in TrackChain++

TrackChain wird derzeit im Projekt TrackChain++ weiterentwickelt. Ziel der Weiterentwicklung ist die Anwendung der Konzepte von Trackchain, also das irreversible Speichern von Events und die kontrollierte Freigabe der Event-Daten, im Kontext eines Clearing Houses im IDS Ecosystem. Die Aufgaben des Clearing Houses im IDS sind:

- Überwachung und Logging von Datenaustausch und Daten-basierten Wertschöpfungsketten
- Überwachung der Einhaltung von Policies bei der Datennutzung
- Bereitstellung einer Plattform für die Rechnungslegung

Im Zuge der Weiterentwicklung werden die Systemkomponenten überarbeitet, um sie von der Anwendungsdomäne Logistik zu lösen und sie an die Anforderungen des Clearing Houses anzupassen. Dabei soll darauf geachtet werden, die Kernkomponenten so allgemein wie möglich zu halten, um die Nutzung von TrackChain für weitere Anwendungsdomänen vorzubereiten.

Für die Umsetzung der Plattform für die Rechnungslegung ist die Nutzung von Smart Contracts geplant. Als Voraussetzung hierfür wurde die BlockChain-API angepasst, um mit Quorum arbeiten zu können. Die bisher genutzte Blockchain *Multichain* bietet keine Smart Contracts an. Im Rahmen von TrackChain++ ist auch die Entwicklung von Tools für die Erstellung von Smart Contracts geplant, um es Entwicklern einfacher zu machen, korrekte Smart Contracts zu entwicklen.

Kontakt

AUTOR UND ANSPRECHPARTNER

Dr. Julian Schütte
Service & Application Security

Mark Gall
Service & Application Security

Fraunhofer AISEC
Parkring 4, 85748 Garching bei München

Telefon: +49 (0) 89 322 99 86 124
info@aisec.fraunhofer.de

Förderrahmen
Die TrackChain wird im Rahmen der Aktivitäten des Forschungsclusters Cognitive Internet Technologies CCIT gefördert.