UNIVERSITÄT
KOBLENZ · LANDAU
Fachbereich 4: Informatik

Enhancing the Security Design of Industrial IoT Platforms

Bachelorarbeit

zur Erlangung des Grades Bachelor of Science (B.Sc.)
im Studiengang Informatik

vorgelegt von

Maximilian Käsgen

Erstgutachter: Dr. Amir Shayan Ahmadian
Univ. Koblenz-Landau, Institut für Softwaretechnik

Zweitgutachter: M.Sc. Marco Ehl
Univ. Koblenz-Landau, Institut für Softwaretechnik

Koblenz, im Mai 2022

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

	Ja	Nein
Mit der Einstellung der Arbeit in die Bibliothek bin ich einverstanden.	☒	☐

. . . . . .

(Ort, Datum) (Unterschrift)

Zusammenfassung

Das (Industrial) Internet of Things und Plattformen zum Aufbau von (Industrial) Internet of Things Anwendungen gewinnen immer weiter an Bedeutung. (Industrial) Internet of Things Anwendungen verarbeiten große Mengen an eventuell Datenschutz relevanten Daten. Zudem können durch Fehlfunktionen, unbeabsichtigte und beabsichtigte, etwa durch einen Hackerangriff, schwere Schäden entstehen, die sich eventuell auch auf das physische Umfeld der Gräte und Menschen in der Nähe der Geräte auswirken können. Mit dem Anstieg an Bedeutung und die möglichen Schäden steigen auch die Anforderungen an die Sicherheit und den Datenschutz innerhalb der Plattformen. Die gesteigerten Anforderungen an die Sicherheit und den Datenschutz erfordern neue oder verbesserte Methoden des Software Engineering um ihnen gerecht zu werden. Dazu soll in dieser Arbeit ein Ansatz vorgestellt werden um Sicherheits- und Datenschutzziele nach Stand aktuellem der Technik bei der Entwicklung von (Industrial) Internet of Things Plattformen besser erfüllen zu können. Um herauszufinden wie Sicherheits- und Datenschutzziele aktuell umgesetzt werden wurden zwölf unterschiedliche Plattformen untersucht. Die Ergebnisse der Untersuchung wurden in einem UMLsec Profil, dem IoTComponentsProfile zusammengefasst. Das IoTComponentsProfile verfügt über einen automatisierten Mechanismus um die Einhaltung von Schutzzielen zu überprüfen und Empfehlungen für nicht erfüllte Schutzziele zu geben. Abschließend wird das IoTComponentsProfile in die aktuell verfügbaren Optionen zur Verbesserung in der Entwicklung von (Industrial) Internet of Things Plattformen eingeordnet.

Summary

The (Industrial) Internet of Things and platforms for building (Industrial) Internet of Things applications are becoming increasingly important. (Industrial) Internet of Things applications process large amounts of data that may be relevant to data protection. In addition, malfunctions, unintentional and intentional, such as a hacker attack, can cause serious damage, which may also affect the physical environment of the devices and people in the vicinity of the devices. With the increase in importance and the possible damage, the requirements for security and data protection within the platforms are also increasing. The increased security and data protection requirements require new or improved methods of software engineering to meet them. For this purpose, an approach is to be presented in this thesis in order to be able to better meet security and data protection goals based on the current state of technology in the development of (industrial) Internet of Things platforms. Twelve different platforms were examined to find out how security and data protection goals are currently being implemented. The results of the investigation were summarized in a UMLsec profile, the IoTComponentsProfile. The IoTComponentsProfile has an automated mechanism to check compliance with protection goals and to make recommendations for protection goals that have not been met. Finally, the IoTComponentsProfile is classified into the currently available options for improving the development of (Industrial) Internet of Things platforms.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1. Einleitung

Das *Industrial Internet of Things* (IIoT) ist eine Erweiterung des Internets um Geräte und Sensoren mit besonderem Fokus auf *Machine-to-Machine* (M2M) Kommunikation, Big Data und maschinelles Lernen im Kontext von Industrieanlagen und Anwendungen. Durch den Einsatz von IIoT erhalten die Nutzer dieser Systeme einige Vorteile, wie zum Beispiel einen erhöhten Grad an Automatisierung und Ressourcenoptimierung von Produktions und Geschäftsprozessen, besser nachvollziehbare Produktionsprozesse und Lieferketten sowie eine bessere Überwachung der laufenden Produktion. Durch das verbinden der Produktions und Lieferketten mit dem Internet entstehen allerdings auch neue Sicherheitsrisiken, die zu monetären Schäden und Reputationsverlust führen können[28]. Um die unterschiedlichen Technologien der diversen IIoT Geräte besser unter Kontrolle bringen zu können und die gesammelten Daten verwerten und weiterleiten zu können werden IIoT Plattformen als zusätzliche Abstraktionsebene eingesetzt. Plattformen erleichtern dabei den Umgang mit IIoT Anwendungen, indem sie Unterstürzung in der Skalierung der Anwendungen, weitere Sicherheitsfeatures sowie genauere Analysemöglichkeiten bieten [15]. Allerdings können diese Plattformen selber neue Schwachstellen mit sich bringen. Die *Open Web Application Security Project* (OWASP) gibt eine Top Ten Liste mit den häufigsten Fehlern für *Internet of Things* (IoT) Anwendungen heraus. Unter anderem die Punkte „3 Insecure Ecosystem Interfaces“ und „4 Lack of Secure Update Mechanism“ sowie weitere betreffen auch (I)IoT Plattformen. Diese Fehler lassen sich durch geeignete Maßnahmen der Softwaretechnik einschränken [20]. Ein Beispiel für die neuartigen Gefahren durch IoT und IIoT Anwendungen ist die Malware Mirai. Diese Malware nutzt aus, dass viele (I)IoT Geräte unter dem Standard Benutzernamen und Passwort betrieben werden. Dadurch logt Mirai sich auf entsprechenden Geräten ein und infiziert diese. Danach können von den infizierten Geräten weitere Angriffe, wie ein Denial-of-Service(DDoS), aufgeführt werden oder Daten vom infizierten Gerät gestohlen werden [6]. Ein Aspekt in der Entwicklung von IIoT Plattformen ist die Verwendung externer Bibliotheken oder Komponenten. Um Anforderungen an die Sicherheit und den Datenschutz gerecht zu werden, müssen IT-Sicherheits- und Datenschutzfeatures der Bibliotheken und Komponenten betrachtet und entsprechend den Anforderungen ausgewählt werden. Dabei sind andere funktionale und nicht funktionale Anforderungen nicht zu vernachlässigen. Um im Designprozess die Anforderungen and Komponenten in Bezug auf Datenschutz und IT-Sicherheit eindeutig darstellen zu können sollten diese durch eine neues Profil in UML Diagrammen mit festgehalten werden. Zudem soll dieses neue Profil auch eine automatische Verifikation erlauben, welche Prüft, ob die Anforderungen an die IT-Sicherheit und den Datenschutz von den Modellierten Komponenten erfüllt werden. Ein solches Profil soll dabei im Rahmen des Projekt Next Level Ecosphere for Intelligent Industrial Production (IIP-Ecosphere)[1] [1] erstellt werden.

[1] Das Projekt Next Level Ecosphere for Intelligent Industrial Production (IIP-Ecosphere) wird vom Deutschen Bundesministerium für Wirtschaft und Klimaschutz (BMWK) finanziert und in Kooperation mit Wirtschaftspartnern durchgeführt

Das Projekt IIP-Ecosphere versucht eine Plattform zu entwickeln um Kein- und Mittelstandunternehmen zu Unterstürzen. In der IIP-Ecosphere Plattform liegt ein Fokus auf Künstlicher Intelligenz und Datensicherheit. Da sich die Plattform noch in der Entwicklung befindet, bietet sie die Möglichkeit die in dieser Arbeit vorgestellten Methoden begleitend zur Entwicklung der Plattform zu testen.

1.1. Ziel der Arbeit

Das Ziel der Arbeit besteht darin die folgender drei Forschungsfragen (RQ) zu beantworten.

- **RQ 1 : Wie werden aktuell Sicherheits- und Datenschutz Anforderungen in IIoT-Plattformen umgesetzt ?**
  Es gibt eine große Menge an IIoT-Plattformen auf dem Markt. Dabei gehen die unterschiedlichen Plattformen die Themen IT-Sicherheit und Datenschutz durch unterschiedliche Methoden und Mechanismen an. Um eine Auswahl und Einstufung unterschiedlicher Ansätze zur Umsetzung von IT-Sicherheit und Datenschutz zu bedarf es eines Überblickes, wie Plattformen die jeweiligen Anforderungen umsetzen. Dieser Überblick soll die Möglichkeit geben bei dem Entwurf neuer IIoT-Plattformen die passenden Methoden und Mechanismen entsprechend der Anforderungen auszuwählen.

- **RQ 2 : Welche Sicherheits- und Datenschutzmechanismen werden von den externen Komponenten der IIP-Ecosphere Plattform zur Verfügung gestellt oder benötigt und wie lassen sich diese Mechanismen in einem UML Profil darstellen ?**
  Um die Anforderungen an IT-Sicherheit und Datenschutz zu erfüllen müssen die Sicherheits- und Datenschutzmechanismen der externen Komponenten des Projektes klar definiert sein. Nur so lässt sich feststellen, ob die jeweilige Komponente die Anforderungen bereits erfüllt, oder ob durch andere Maßnahmen nachgebessert werden muss. Um einen besseren Überblick über die Sicherheits- und Datenschutzmechanismen der externen Komponenten benötigt es ein einheitliches Profil aus dem sich diese Mechanismen ablesen lassen. Um die Lesbarkeit weiter zu erhöhen und die Informationen über die Sicherheits- und Datenschutzmechanismen besser zu dokumentieren sollte das Profil auch in UML darstellbar sein. Ein solches Profil gibt es zur Zeit allerdings noch nicht.

- **RQ 3 : Wie lässt es sich, im Kontext der IIP-Ecosphere Plattform, automatisch überprüfen, welche Sicherheits- und Datenschutzmechanismen in einem Systemmodell verwendet werden ?**
  Große Projekte, wie IIP-Ecosphere, verfügen oft über eine Vielzahl an externen Komponenten, die teilweise an mehreren unterschiedlichen Stellen im Projekt und entsprechend auch mehreren UML Modellen vorkommen. Dabei kann es zu Fehlern kommen, bei denen Anforderungsgen an Komponenten übersehen werden. Um die Häufigkeit solcher Fehler zu verringern bedarf es eines automatisierten Mechanismus um zu überprüfen, ob die Anforderungen an die IT-Sicherheit und den Datenschutz durch die modellierten Komponenten erfüllt werden. Dieser automatisierte Mechanismus kann durch die Verifizierung von UML Modellen gegenüber einem vordefinierten Profils erfüllt werden. Allerdings benötigt diese Verifizierung ein Profil und einen dazugehörigen Regelsatz, welche es zur Zeit

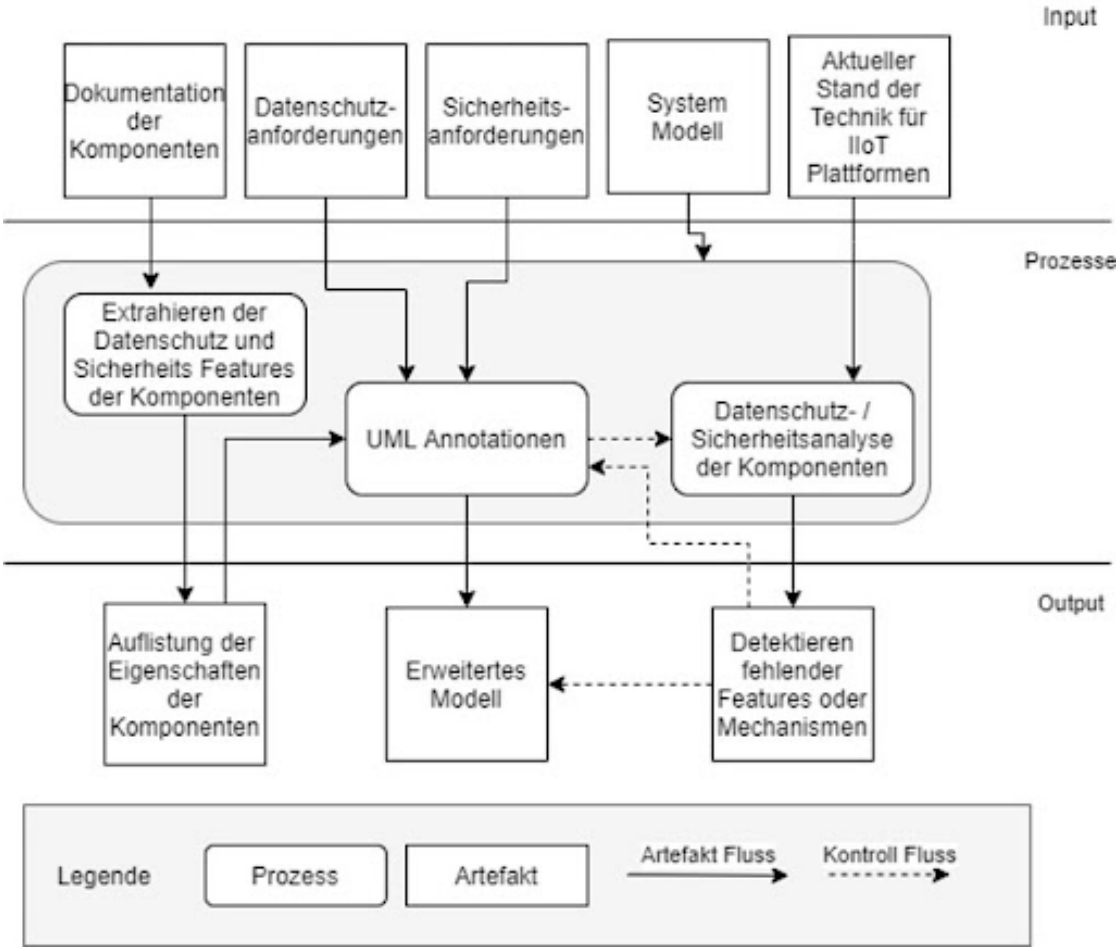

Abbildung 1.1.: Der Arbeits Prozess der in Rahmen dieser Arbeit entwickelt wurde..

noch nicht für den konkreten Anwendungsfall, die IT-Sicherheits- und Datenschutzanforderungen im IIP-Ecosphere Projekt, gibt.

Durch die Antworten auf die Fragen soll ein Prozess entstehen um die Sicherheit und den Datenschutz von IIoT-Plattformen in der Entwurfsphase zu verbessern. Dabei entsteht eine Auflistung der Eigenschaften von Plattformkomponenten, in der Dokumentation. Die Eigenschaften der Komponenten zusammen mit den Datenschutzanforderungen und Sicherheitsanforderungen werden durch UML Annotationen in das Systemmodell hinzugefügt um das Systemmodell zu erweitern. Zuletzt wird auf dem erweiterten Modell eine Analyse durchgeführt um, auf Basis der aktuellen Stand der Technik für IIoT Plattformen, fehlende Mechanismen zu erkennen und gegebenenfalls Empfehlungen auszusprechen. Der Prozess ist in Abbildung 1.1 dargestellt.

1.2. Aufbau der Arbeit

Kapitel 2 erklärt um was es sich bei der *Unified Modeling Language* (UML) und IIoT-Plattformen handelt. Zudem werden Definitionen für Schutzziele in der IT-Sicherheit und im Datenschutz eingeführt. In Kapitel 3 wird der aktuelle Stand der Technik für IIoT Plattformen untersucht. Dazu wird eine Fallstudie anhand von zwölf Plattformen durchgeführt um herauszufinden welche Technologien verwendet werden. Danach werden die Technologien in einem Modell hierarchisch angeordnet und Schutzzielen zugeteilt, sowie ein Dokumentationstemplate für Komponenten von IIoT-Plattfromen vorgestellt. Kapitel 4 beschreibt das erstellen von zwei Plugins für die Eclipse *Integrated Development Environment* (IDE). Das erste Plugin ist ein UML-Profil um Systemmodelle mit Schutzzielen und Technologien zur Erfüllung der Schutzziele zu erweitern. Das zweite Plugin erweitert CARiSMA um einen weiteren Check um zu prüfen ob die mit dem UML-Profil aus dem ersten Plugin erstellten Anforderungen erfüllt sind. In Kapitel 5 werden die Plugins in den aktuellen Stand für UML-Profile im Secure Software Engineering eingeordnet. Anschließend wird die Funktion der Plugins im IIP-Ecosphere Projekt demonstriert. Kapitel 6 schließt die Arbeit mit einem Überblick über andere Ansätze, einem Ausblick für eine Weiterentwicklung der vorgestellten Konzepte und Software und einer Bewertung der Zielerfüllung ab.

2. Grundlagen

In diesem Kapitel werden die für das Verständnis dieser Arbeit notwendigen Grundlagen eingeführt. Die Grundlagen umfassen die *Unified Modeling Language* (UML), das Internet of Things, (Industrial) Internet of Things Plattformen und eine Definition für Schutzziele in der IT-Sicherheit und des Datenschutz.

2.1. Unified Modeling Language

Die *Unified Modeling Language* (UML) ist eine, von der *Object Managament Group*(OMG), entwickelte Modellierungssprache. Die Verwendung einer Modellierungssprache bietet eine Vielzahl an Vorteilen in der Entwicklung von Software. Modellierungssprachen bieten in ihrer Verwendung einen hohen Freiheitsgrad, da sie nicht an eine spezifische Programmiersprache, wie C++ oder Java, gebunden sind. So kann der Fokus in der Entwicklung komplett auf der Programmlogik und Architektur der Software liegen. Zudem kann dadurch eine neutrale Dokumentation über die Funktionalität von Software erstellt werden, die auch bei Änderungen von verwendeten Technologien und Programmiersprachen im Software Lebenszyklus gilt. Modellierungssprachen wie UML überlassen zudem dem Modellierer den Detailgrad des Modells. Dadurch, dass der Detailgrad sich so frei bestimmen lässt eignen sich die Modell auch zur Kommunikation zwischen den Stakeholdern der Software, da der Detailgrad und die Komplexität an das Verständnis der jeweiligen Kommunikationspartner anpassen lässt. Aufgrund dieser Vorteile hat sich der Ansatz des modellbasierten Entwicklung hervorgetan. In der modellbasierten Entwicklung werden erst Modelle der zu entwickelnden Software erstellt um die Komplexität zu beherrschen und die Funktionsweise der Software zu bestimmen bevor man sich für Programmiersprachen und Technologien entscheidet[26].

Die Sprache UML ist hierarchisch mit vier Schichten aufgebaut. Die oberste Schicht bildet die *Meta Object Facility* (MOF) M3. Die MOF gibt die Elemente für die Schicht unter sich vor und wird zeitweise als Meta-Metamodell bezeichnet. Unter der Schicht der MOF befindet sich die Schicht des Metamodell M2. Das Metamodell gibt die Bausteine und Anwendungsregeln der darunter gelegenen Schicht vor. Mit diesen Bausteinen können Nutzer dann eigene Modelle erstellen. Unter dem Metamodell befindet sich dann die angewandte Instanz von des UML Modell M1. Auf dieser Schicht werden die Modelle für neue Software entwickelt. Die Element in diesem Modell sind Instanzen des Metamodells M2. Die unterste Ebene ist das Objekt M0. Auf dieser Ebene befinden sich konkrete Umsetzungen des Modell. Hier werden den im UML Modell auf Ebene M1 modellierten Elementen konkrete werte zugeteilt[12].Ein Beispiel für den Aufbau von UML befindet sich in Abbildung 2.1.

In Abblidung 2.1 bestimmt auf der Obersten Ebene die MOF die Elemente des Metamodell. Darunter bestimmt das Metamodell wie die Elemente angewendet werden können und ordnet

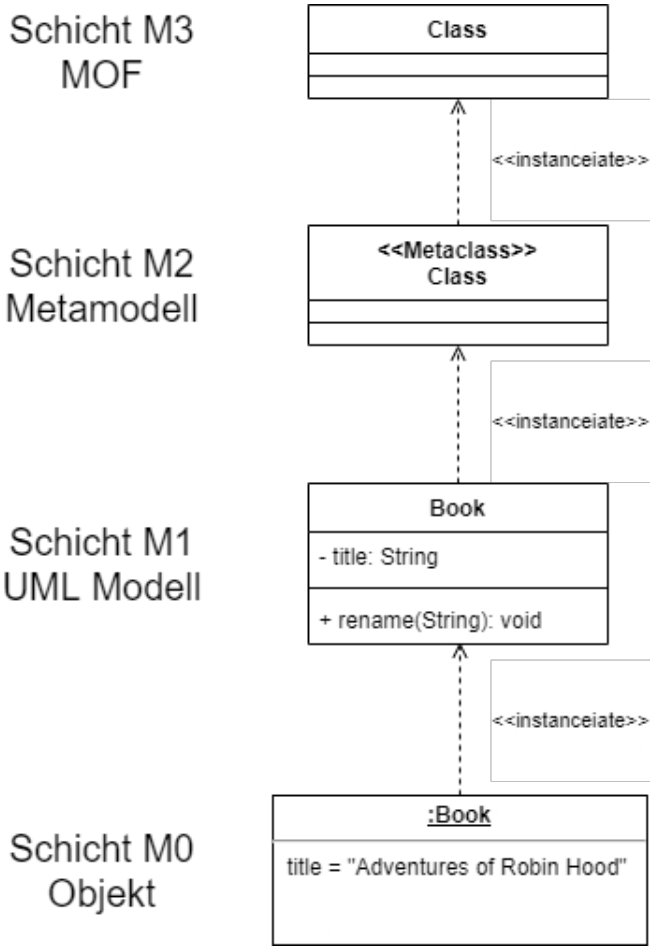

Abbildung 2.1.: Beispiel der Hierarchischen Architektur von UML.

ihnen eine Bedeutung zu. Hier wird das Element einer Klasse definiert, das einer Klasse in einer Objektorientierten Programmiersprache, zum Beispiel Java, entspricht. Auf der Ebene unter dem Metamodell wird von einem Nutzer die Klasse mit dem Namen „Book“modelliert. Die Klasse „Book“verfügt über ein Feld „title“und eine Methode „rename“. Auf der Schicht darunter wird dann ein Objekt als eine konkrete Instanz der Klasse „Book“erstellt. Dem Objekt ist dann für das „title“Feld auch ein Wert zugeordnet.

Um Modelle zwischen unterschiedliche Werkzeugen und eventuellen Erweiterungen von Werkzeugen austauschen zu können existiert zudem das Format *XML Metadata Interchange* (XMI). XMI basiert auf *Extensible Markup Language* (XML) und ist ein standardisiertes, hierarchisches und textbasiertes Austauschformat. Die genauen Formatvorgaben für XMI werden ebenfalls von der OMG herausgegeben[13].

Ein Vorteil von UML Modellen ist, ist dass Entwickler aus UML Modellen automatisch Code generieren können. Das generieren von Code aus UML Modellen wurde ebenfalls von OMG in der sogenannten *Model Driven Architecture* (MDA) formalisiert[26][19].

2.1.1. Erweiterungen der Unified Modeling Language

Um UML auch an spezifischer Anwendungen anpassen zu können lässt sich die Sprache durch sogenannte Profile erweitern. Dabei können neue Stereotypen erstellt werden, indem das UML Metamodell erweitert wird. Diese neuen Stereotypen können dann genutzt werden um UML Modelle um zusätzliche Informationen zu erweitern. Zudem können den Stereotypen noch Key Value Paare, sogenannte Tags, zugeordnet werden.[12]In Abbildung 2.2 wird beispielhaft ein neues Profil erstellt. Es wird eine neues Stereotype `<<Confidentiality>>` eingeführt, das über ein Tag mit Namen `Tag` verfügt. `Tag` wird ein Integer als Wert zugeordnet. Das Stereotype `<<Confidentiality>>` kann dann in UML Modell zu Elementen vom Typ Klasse hinzugefügt werden. In unserm Beispiel könnte so etwa in einem Modell auf einer numerischen Skala bewertet werden wie wichtig die Vertraulichkeit der Daten der Klasse „DataStore“ist.

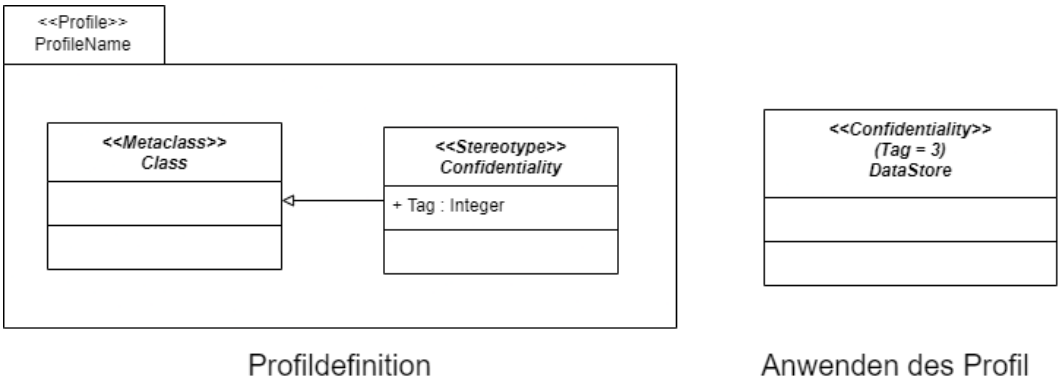

Abbildung 2.2.: Beispiel für die Definition und Anwendung eines neuen UML Profil.

Des weiteren können über die *Object Constraint Language* (OCL) weitere Bedingungen an die Modelle gestellt werden[11].So könnte in unserm Beispiel in Abbildung 2.2 zusätzlich verlangt werden, dass der Wert des Tag "Tag"nicht kleiner als eins und nicht größer als fünf ist.

Neben Profilen gibt es weitere Erweiterungsmechanismen in UML. So können unter anderem auch neue Metaklassen erstellt werden. Diese Erweiterungen sind allerdings aufwendiger als UML Profile und für die Arbeit nicht relevant.

2.1.2. UMLsec

Eine für diese Arbeit wichtige Erweiterung der UML ist UMLsec. UMLsec ist ein UML Profil, dass Entwicklern erlaubt UML Modelle mit Stereotypen zu ergänzen die die Sicherheitsfaktoren des Modell beschreiben. Zum Beispiel können so Verbindungen zwischen zwei Elementen eines UML Modells als verschlüsselt beschrieben werden. Zudem gibt UMLsec Entwicklern die Möglichkeit ein Angreifermodell zu erstellen. Durch die Kombination aus dem erweiterten UML Modell und dem Angreifermodell soll es möglich sein die Sicherheitsaspekte von Software bereits in der Entwurfsphase zu überprüfen und falls nötig Verbesserungen vor zu nehmen. Dabei wurde das Profil so entworfen, dass die Entwickler keine Sicherheitsexperten sein müssen um das Profil korrekt anwenden zu können und so die Sicherheit ihrer Anwendungen zu

verbessern[16].

UMLsec wurde zudem um die Funktionalität des UMLsec-Profil erweitert.[24] Die Funktionalität des UMLsec-Profil ist dreigeteilt. Der erste Teil wäre das Profil. Durch diesen Teil kann UMLsec um neue Stereotypes erweitert werden. Der Prozess ist verwendet den Standard Prozess für die Definition von UML Profilen. So können mit den neuen Stereotypen annotierte Elemente weiter über die standardisierten Wege, zum Beispiel ein XMI Austauschformat, verarbeitet werden Die durch die Profile hinzugefügten Stereotypen beinhalten die neuen Informationen. Der zweite Teil wäre ein Analyse Modell. Das Analyse Modell erstellt ein neues Metamodell und kann genutzt werden um automatische Analysen von Modellen durchzuführen. Der dritte Teil ist eine Transformationskomponente, die eine Transformation von der Profilkomponente auf das Analyse Modell ermöglicht. Diese Trennung hat den Vorteil, dass Modell in denen der Profil Teil des UMLsec-Profil angewandt wurde weiterhin von unabhängigen Werkzeugen verwendbar sind [24].

2.2. Internet of Things und Industrial Internet of Things

Das Internet der Dinge, eng. *Internet of Things* (IoT) beschreibt eine Erweiterung des Internets um verschiedene Sensoren oder Geräte, auch Dinge (eng. Things) genannt. Bei diesen Dingen handelt es sich um Geräte die bis dato nicht mit dem Internet verbunden waren, wie zum Beispiel Glühbirnen, Schlösser oder auch Kaffeemaschinen[29].

Das Industrielle Internet de Dinge, eng.*Industrial Internet of Things* (IIoT) ist eine Erweiterung des IoT. Im IIoT werden zusätzlich Systeme mit dem Internet verbunden die zuvor in den Bereich der *Operational Technology* (OT) gehörten. Zu den System aus der OT, die so mit dem Internet verbunden werden zählen unter anderem *Industrial Control Systems* (ICSs), *Supervisory Control And Data Acquisition* (SCADA)Systeme und *Programmable Logic Controllers* (PLCs). Durch einen Fokus auf *Machine-to-Machine* (M2M) Kommunikation, Big Data und maschinellem Lernen erhoffen sich Nutzer von IIoT System eine Steigerung der Effizienz und Zuverlässigkeit ihrer Geschäftssysteme[28].

Das Einführen solcher Geräte in das Internet birgt allerdings auch Gefahren. Zum einen Besteht eine Gefahr für die Privatsphäre der Nutzer von IoT Geräten, beziehungsweise der Kunden von IIoT Anwendungen. Die Geräte besitzen die Möglichkeit neue Arten von personenbezogenen Daten zu sammeln. Zudem funktionieren die Geräte besser je mehr Daten sie sammeln. Das stellt Entwickler vor die Herausforderung eine Balance zu finden die, zwar die gewünschte Funktionalität von (I)IoT Anwendungen ermöglicht aber dennoch die Privatsphäre der Nutzer angemessen schützt. Zum anderen entstehen durch die neuen Geräte und Systeme auch neuartige Sicherheitsprobleme. Durch die hohe Verbundenheit zwischen Geräten und die große Anzahl an Geräten in einem (I)IoT System kann ein einzelnes Gerät mit einer Schwachstelle leicht zu einem Einfallstor werden über das das komplette Netzwerk angegriffen werden kann. Weitere Probleme entstehen dadurch, dass es sich die Geräte selbst physisch Manipuliert werden können und dadurch Fehlfunktionen ausgelöst oder Angriffe durchgeführt werden können. Zudem ist das Schadenspotenzial durch die Verbindung von IT und OT wesentlich größer als zuvor. Neben bisher möglichen Schäden durch Hackerangriffe, wie zum Beispiel finanziellen oder Reputationsschäden, können nun physische Schäden an Geräten oder Menschen hinzukommen[29][28].

2.3. Aufbau einer IIoT Plattform

Eine Plattform beschreibt im allgemeinen ein Produkt, eine Dienstleistung oder eine Technologie, die als Basis für weitere, neue Produkte, Dienstleistungen oder Technologien dient[4]. Dabei erlaubt diese Definition den Aufbau einer Plattform auf einer anderen Plattform. Um als IIoT-Plattform zu zählen muss dazu die Anwendung der Plattform in dem Industrial IoT liegen. Die Plattformen sind dabei in einem Schalen Prinzip aufgebaut. In der Mitte steht ein Plattform-Sponsor, der die Plattform entworfen hat und dem das geistige Eigentum an der Plattform gehört. Am Beispiel von Microsoft Azure wäre das Microsoft. Darum gibt es einen Plattform-Betreiber der die Plattform betreibt. Am Beispiel von Microsoft Azure wäre dies der Betreiber des Rechenzentrums. Diese beiden inneren Schalen bilden den Kern der Plattform. Der Kern ändert sich selten und langsam, da er die Infrastruktur im Plattform Ökosystem bildet. Um den Kern der Plattform herum gibt es dann Anwendungsentwickler. Diese entwickeln neue Produkten, Dienstleistungen oder Technologien auf Basis der Plattform. Am Beispiel von Microsoft Azure wäre dies ein Kunde der den Cloudservice nutzt und eine Anwendung entwickelt, um zum Beispiel die Temperatur im inneren einer Fertigungsanlage zu messen. Die äußerste Schale um die Anwendungsentwickler herum bilden dann die Endnutzer. Am Beispiel der auf Microsoft Azure aufgebauten Anwendung zum Überwachen der Temperatur wäre dies der Mitarbeiter des Unternehmens der für die überwachte Maschine zuständig ist. Die Anwendungsentwickler und Endnutzer lassen sich als die Peripherie der Plattform zusammenfassen[4]. Dieser Schalenaufbau wir auch in Abbildung 2.3 nochmal grafisch dargestellt.

Plattformen bieten technisch, insbesondere im (I)Iot Umfeld, den Vorteil, dass sie eine weitere Abstraktionsebene einführen. Das erlaubt eine einfachere und daher schnellere Entwicklungen von Anwendungen an der Peripherie der Plattform. Die Entwicklung von Anwendungen wird einfacher, da die Komplexität, die durch die Vielzahl unterschiedlicher Technologien aus dem (I)IoT Umfeld durch einheitliche Schnittstellen ersetzt wird. Zudem kann die Plattform Funktionen bereitstellen die von mehreren Anwendungen gebraucht werden zur Verfügung stellen um die Entwicklung zu Beschleunigen. Die Funktionen der Plattform können dazu noch verwendet werden um die Sicherheit der Plattform zu erhöhen, da für die Sicherheit wichtige Komponenten der Plattform und Anwendung, wie zum Beispiel Authentifizierung und Autorisierung, korrekt in der Plattform selbst implementiert werden und sich so zentral auf für alle Anwendungen an der Peripherie durchsetzen lassen. Zudem helfen Plattform durch die leichte Anwendungsentwicklung bei der Skalierung und Erweiterung von Anwendungen und Geschäftsmodellen, da in diesen Fällen bereits eine Infrastruktur besteht[15]. Die Art der Abstraktion die durch eine (I)IoT-Plattform hinzugefügt wird ist vereinfacht in Abbildung 2.4 dargestellt.

2.4. Schutzziele

Im Rahmen dieser Arbeit werden die Schutzziele der Confidentiality, Integrity, Availability, Unlikability, Transparancy und Intervenability betrachtet. Unterschideliche Autoren in der Literatur, definieren auch die jeweiligen Schutzziele unterschiedlich. Zudem werden auch Teilweise weitere Schutzziele oder präzisere Teilziele vorgeschlagen und betrachtet[8]. Confidentiality, Integrity und Availability werden dabei klassisch der IT-Sicherheit zugeordnet und sind auch

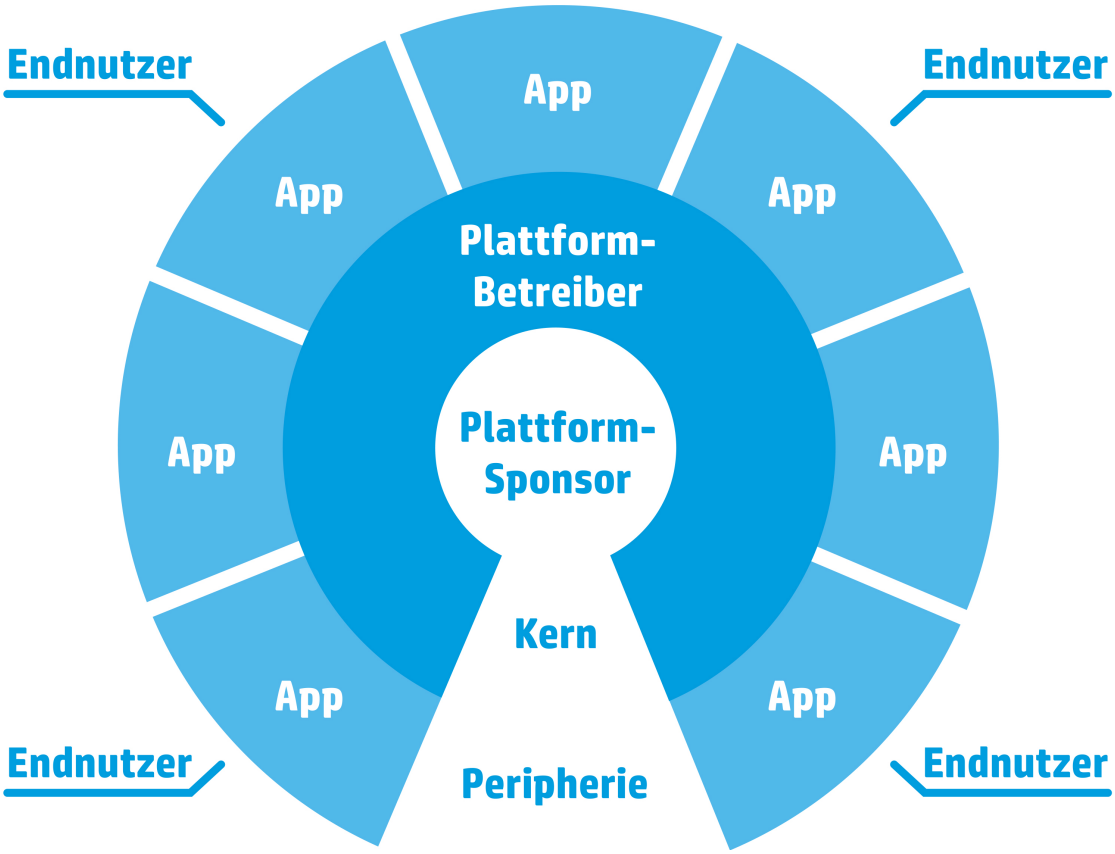

Abbildung 2.3.: Aufbau des Schalenmodells einer Plattform. Abbildung aus [4].

als CIA-Triad bekannt. Confidentiality, Integrity, Availability, Unlikability, Transparancy und Intervenability werden für diese Arbeit wie folgt definiert.

2.4.1. Confidentiality

Das Ziel der Confidentiality ist die Geheimhaltung von Daten gegenüber Unautorisierten[8][5][14]. Dazu können unterschiedliche Methoden eingesetzt werden. Neben der Verschlüsselung der Daten gehört eine Form von Access Control auch zu den Möglichkeiten um die Confidentiality zu erfüllen[5].

2.4.2. Integrity

Um die Integrity zu erfüllen dürfen Daten nicht unautorisiert verändert werden[8][5][14]. Zudem müssen die Daten verlässlich und nicht abstreitbar sein[14]. Die Anforderungen an die Authentizität der Daten erweitert sich hierbei allerdings auch auf die Authentizität der Datenquelle[5].

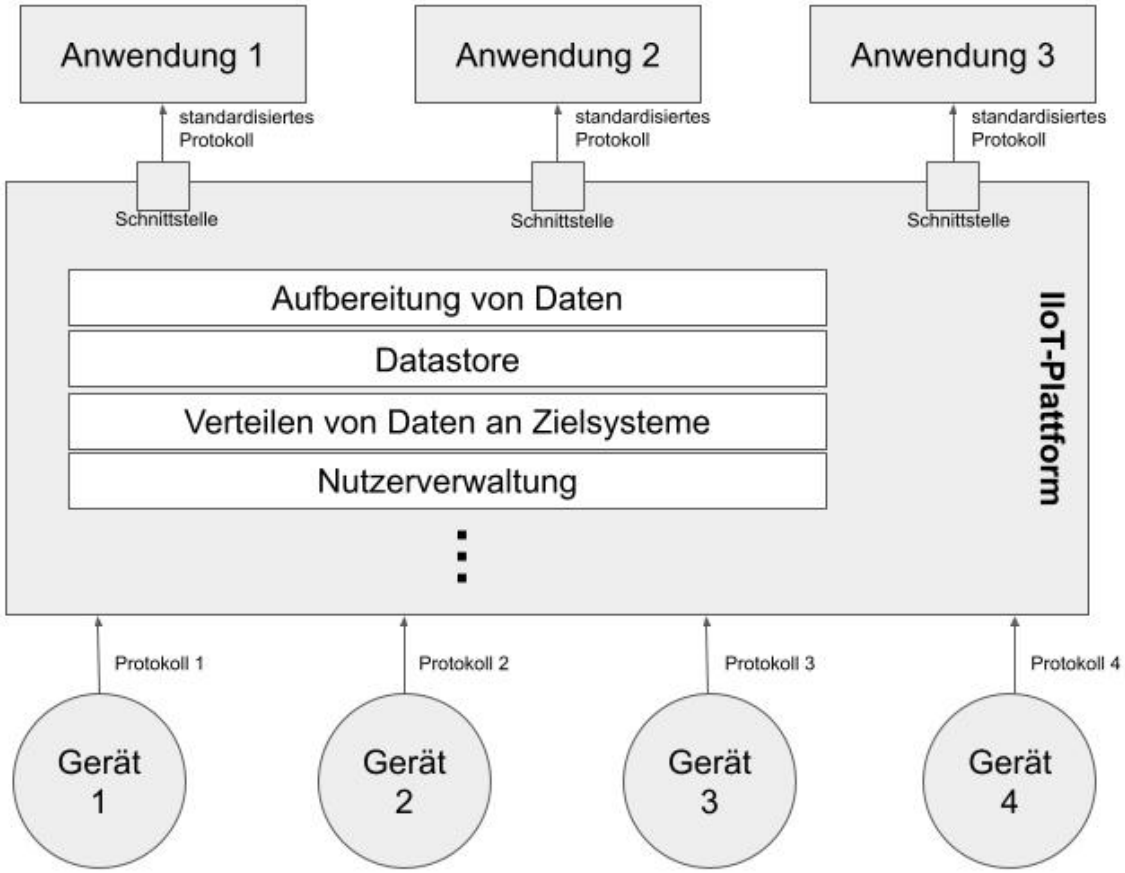

Abbildung 2.4.: Grobe Darstellung des Aufbau einer (I)IoT-Plattform. Die unterschiedlichen Protokolle der Geräte werden über Schnittstellen zu standardisierten Protokollen für die Anwendungen..

2.4.3. Availability

Die Availability verlangt, dass in angemessenerer Zeit Daten abgerufen und genutzt werden können[5][14]. Der Zugriff und die Nutzung der Daten darf dabei nicht durch Unautorisierte gestört werden[8].

2.4.4. Unlinkability

Das Ziel der Unlikability verlangt, dass Daten einer Domäne nicht mit Daten einer andere Domäne verknüpfbar sein dürfen[14][7]. Das bedeutet, dass Daten aus der einen Domäne keine zusätzlichen Informationen zu Daten aus einer anderen Domäne liefern dürfen.

2.4.5. Transparency

Um das Ziel Transparency zu erfüllen muss die komplette Verarbeitung von Daten verstanden werden und Rekonstruierbar sein. Dabei sind gesetzliche, organisatorische und technische Prozesse eingeschlossen. Zudem müssen diese Anforderungen auch nach Löschung der Daten noch erfüllt sein[14][7].

2.4.6. Intervenability

Das Ziel der Intervenability verlangt, dass den Eigentümern von Daten die Möglichkeit gegeben sein muss nachträgliche Änderungen an ihren Daten durchzuführen um die Daten zu korrigieren oder ihr Anrecht auf eine Löschung durchzusetzen[14][7].

2.4.7. Gegensätzliche Schutzziele

Es ist nicht möglich alle der Ziele zugleich, im gleichen Maß zu erfüllen. Einige der Ziele stehen in direktem Widerspruch zueinander.

Confidentiality und Availability widersprechen sich. Die Availability verlangt, dass Nutzer Daten schnell abrufen und einsehen können. Confidentiality hingegen verlangt, dass Daten geheimgehalten werdenden und nicht an andere preisgegeben werden.

Das Ziel Integrity und das Ziel Intervenability widersprechen sich, da Integrity verlangt, dass Daten unverändert bleiben wären Intervenability fordert, dass Änderungen an Daten möglich sein müssen.

Die Zile Transparency und Unlikability widersprechen sich. Transparency fordert, dass Informationen bereitgestellt werden um die Verarbeitung von Daten nach zu vollziehen. Die bereitgestellten Informationen könnten aber genutzt werden um Daten aus unterschiedlichen Domänen zu verknüpfen, was der Unlikability widerspricht.

Dieses Verhältnis zwischen den Zielen wird in Abbildung 2.5 dargestellt.

Neben den sich direkt widersprechenden Zielen gibt es noch weitere Konflikte und Synergien zwischen den Schutzzielen.

Eine solche Synergie besteht zum Beispiel zwischen der Confidentiality und der Unlikability. Werden etwa Daten verschlüsselt um die Confidentiality zu wahren, so wird auch die Unlikability verbessert, da man unleserliche Daten nicht mit anderen Daten verknüpfen kann.

Ein weiterer Konflikt entsteht zum Beispiel zwischen der Confidentiality und der Transparency. Die Confidentiality verlangt, dass Daten geheimgehalten werden. Die Transparency verlangt hingegen, dass die Daten herausgegeben werden um die Prozesse nachvollziehen zu können.

Aufgrund der Widersprüche und Konflikte muss eine Balance gefunden werden um Systeme zu entwickeln die sicher sind und die Anforderungen an den Datenschutz erfüllen[14].

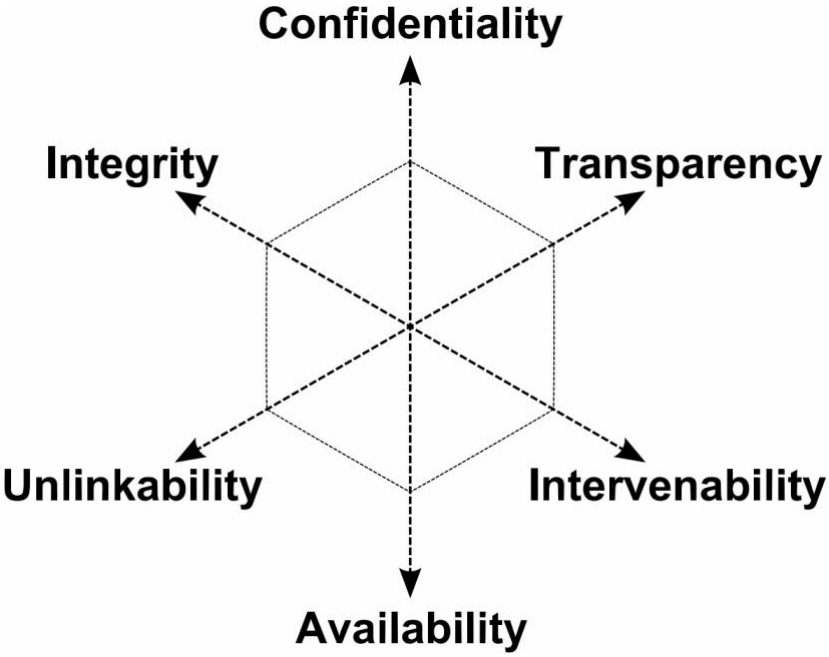

Abbildung 2.5.: Sich widersprechende Schutzziele. Abbildung aus [14].

3. Konzeption

Um eine Grundlage für eine automatische Verifikation über die Einhaltung von IT-Sicherheits- und Datenschutzzielen zu schaffen muss zunächst bekannt sein, welche Methoden und Technologien aktuell in IoT-Plattformen verwendet werden. Dazu wurden die Dokumentationen von zwölf IoT-Plattformen untersucht. Die Ergebnisse der untersuchen können dann als Grundlage verwendet werden um im Entwurf neuer IoT-Plattformen zu kontrollieren, ob die verwendeten Komponenten der Plattform über ausreichenden Schutz verfügen.

3.1. Untersuchung des aktuellen Stand der Technik für die Sicherheit und den Datenschutz von IIoT Plattformen

Um eine Übersicht über die aktuell verwendeten Methoden und Mechanismen zur Umsetzung von IT-Sicherheitsanforderungen und Datenschutzanforderungen zu gewinnen wurden zwölf (I)IoT-Plattformen untersucht. Es wurden die Plattformen :

- Amazon - AWS IoT
- Bosch – Bosch IoT Suite
- Cisco - Kinetic
- IBM - Watson IoT Suite
- Microsoft - Azure IoT Suite
- Oracle – Oracle Cloud IoT
- PTC - Thing Worx
- SAP - Leonardo
- Siemens - Mindsphere

ausgewählt, da diese aufgrund ihrer Umsatz- und Gewinnzahlen, sowie Relevanz für dem Industrie 4.0-Ansatz von den Autoren von [25] ausgewählt wurden. Dabei wurde auf die Expertise von Partnern des Projekts vertraut.

Zusätzlich wurden noch die Plattformen:

- B& R - Automation mapp Technology
- General Electrics - Predix

- Google – Google Cloud IoT Core

ausgewählt die ebenfalls in [25] vorkamen. Diese drei Plattformen wurden dabei aus folgenden Gründen gewählt. B& R - Automation mapp Technology wurde gewählt, da es sich um eine sehr spezialisierte Plattform mit Fokus auf direkte Steuerungselemente wie z.B. die Positionierung von Einzelachsen von Maschinen handelt. Die Google Cloud IoT Core wurde ausgewählt, da die Google Cloud eine hohe Marktkapitalisierung[1] hat. General Electrics - Predix wurde zufällig aus den in [25] ausgewählt.

3.1.1. Fragekatalog

Um die von den Plattformen verwendeten Methoden und Mechanismen zum Umsatz von IT-Sicherheits- und Datenschutzanforderungen zu erfassen wurde darauf ein Fragekatalog erstellt. Der Fragekatalog wurde dabei in vier Teile unterteilt, die jeweils betrachtet wurden : die Edge-Geräte, die Plattform, Authentifizierung und Autorisierung von Nutzern und die Übertragung von Daten aus der Plattform hinaus. Bei der Auswahl der Fragen wurde versucht den Daten zu folgen. Dabei entstehen die Daten an der Edge. Dort werden die Daten eventuell gespeichert oder bereits zu teilen verarbeitet, bis sie in die zentralen Komponenten der Plattform übertragen werden. Um die Sicherheit und den Datenschutz der Daten in diesem Stadium soll durch die Fragen in dem Teil Edge-Geräte des Fragekatalogs geklärt werden. Nachdem die Daten von der Edge an die Plattform übertragen wurden werden die Daten dort weiter verarbeitet, gespeichert und zwischen verschiedenen Komponenten übertragen. Die Aspekte der Sicherheit und des Datenschutz in diesem Bereich werden im Teil des Fragekatalogs Plattform betrachtet. Nachdem die Daten in der Plattform gespeichert oder verarbeitet wurden können sie auch wieder aus der Plattform hinaus übertragen werden. Das Ziel kann dabei eine andere Plattform, ein Datenspeicher, der nicht der Plattform zugeordnet ist, oder auch Systeme einer anderen Organisation sein. Die Sicherheit dieser Übertragung wird im Teil externe Anwendungen des Fragekatalogs betrachtet. Während des kompletten Lebenszyklus der Daten, von der Edge bis zur externen Anwendung, können Menschen mit den Daten interagieren. Um zu verstehen wie der Zugriff auf die Daten erfolgt wurden die Fragen in dem Teil Nutzverwaltung gestellt. Der Prozess des Lebenszyklus der Daten wird dabei vereinfacht in Abbildung 3.1 dargestellt. Dazu wurden die Daten bei ihrem Ursprung, in Transit und in Ruhe betrachtet. Nach den Fragen zu den Daten wurde der Fragekatalog um Fragen zur Sicherheit, insbesondere Integrität, der Infrastruktur erweitert.

Um zu Validieren, dass die Auswahl der Fragen ein möglichst vollständiges Bild über die eingesetzten Methoden und Mechanismen gibt wurde die Auswahl mit den Kriterien einer Untersuchung der Autoren Yu und Kin zu Sicherheit von IoT Plattformen [31] sowie eines Berichts der Deloitte zu relevanten Sicherheitsfeatures bei der Auswahl von IoT Plattformen [21] verglichen und um die Frage nach der Klassifizierung von Daten erweitert.

Um die Fragen zu beantworten wurde jeweils nur die öffentlich verfügbare Dokumentation der Hersteller verwendet. Dieser Ansatz unterliegt dabei zwei möglichen Fehlern. Es können weitere oder neu Aspekte der IT-Sicherheit oder des Datenschutz hinzu kommen die im Fragekatalog nicht betrachtet wurden. Da nur die öffentlich verfügbare Dokumentation verwendet wurde entstehen weitere Fehlerquellen. Da die öffentliche Dokumentationen der Plattformen

[1] https://kinsta.com/google-cloud-market-share/ zuletzt abgerufen 15.02.2022

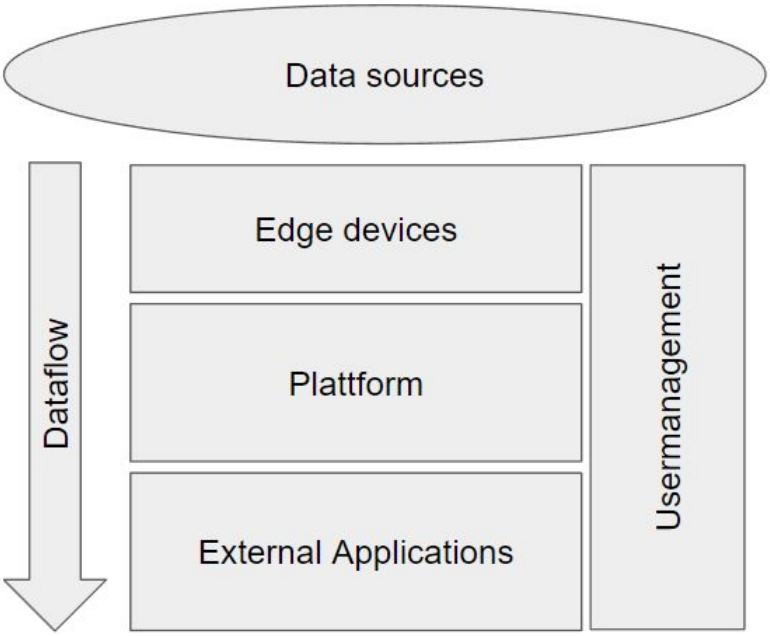

Abbildung 3.1.: Einfache Darstellung des Datenfluss in einer IIoT-Plattform.

nicht den genauen Aufbau der Plattformen beschreiben besteht die Möglichkeit, dass die Ergebnisse des Fragebogens unvollständig sind.

Der Fragekatalog umfasst die folgenden Fragen:

Zu den Edge-Geräten:

In diesem Bereich des Fragekatalogs wurden die Geräte an der Edge betrachtet. Als Geräte an der Edge werden in diesem Fragekatalog alle mit dem Netzwerk verbundenen Geräte, die nicht zur eigentlichen Plattform gehören. Das umfasst dabei alles von Sensoren, Protokollbrücken und Gateway-Server. Dabei lag der Fokus darauf wie die Daten nach der Erfassung gespeichert werden und wie sie an die Plattform übertragen werden.

- **Wie wird Vertraulichkeit der Übertragung von der Edge zur Plattform sichergestellt ?**
  Das Ziel dieser Frage war herauszufinden wie die unterschiedlichen Geräte an der Edge versuchen auf ihnen erhobene und gespeicherte Daten vor Einsicht und Zugriff durch Unbefugte schützen, während die Daten zwischen Geräten oder Komponenten der Plattform übertragen werden. Eine mögliche Antwort auf diese Frage wäre die Verwendung von Protokollen, die Daten verschlüsselt übertragen.

- **Wie wird Authentizität der Übertragung / des Thing von der Edge zur Plattform sichergestellt ?**
  Das Ziel dieser Frage war herauszufinden wie sich die unterschiedlichen Geräte an der Edge vor einer Übertragung von Daten authentifizieren um sicherzustellen, dass Unbefugte

nicht an der Kommunikation im Plattform Ökosystem beziehungsweise an der Kommunikation an der Edge teilnehmen. Eine mögliche Antwort auf diese Frage wäre die Verwendung von Authtifizierungsmechanismen wie X.509 Zertifikaten.

- **Wie wird Integrität der Übertragung von der Edge zur Plattform sichergestellt ?**
  Das Ziel dieser Frage war herauszufinden wie die unterschiedlichen Geräte an der Edge versuchen auf ihnen erhobene und gespeicherte Daten vor Veränderungen durch Unbefugte oder Störungen schützen, während die Daten zwischen Geräten oder Komponenten der Plattform übertragen werden. Eine mögliche Antwort auf diese Frage wäre die Verwendung von Checksummen in den zur Übertragung von Daten verwendeten Protokollen.

- **Wie wird Vertraulichkeit von Daten auf Edgegeräten sichergestellt ?**
  Das Ziel dieser Frage war herauszufinden wie die unterschiedlichen Geräte an der Edge versuchen auf ihnen erhobene und gespeicherte Daten vor Einsicht und Zugriff durch Unbefugte schützen. Eine mögliche Antwort auf diese Frage wäre die Verschlüsselung von Daten auf dem Gerät.

- **Wie werden Updates durchgeführt ?**
  Ziel der Frage ist heraus zu finden, wie Updates der Geräte durchgeführt werden um die Integrität der Geräte sicherzustellen. So sollen die Geräte an der Edge vor neue entdeckten Schwachstellen durch veraltete Software geschützt werden. Eine mögliche Antwort auf diese Frage wäre ein von der Plattform bereitgestelltes System um Edge-Geräte automatisch und in Gruppen upzudaten.

- **Gibt es ein Flottenmanagement (für On- und Offboarding / monitoring)?**
  Flottenmanagement bezeichnet für diese Frage die Möglichkeit mehrere Geräte an der Edge zu überwachen, konfigurieren oder ein Onboarding oder Offboarding durchzuführen.

Zu der Plattform :

Dieser Abschnitt des Fragekatalogs betrachtet die Komponenten der Plattform.

- **Wie wird Vertraulichkeit der Übertragung innerhalb der Plattform(-komponenten) sichergestellt ?**
  Das Ziel dieser Frage war herauszufinden wie die unterschiedlichen Komponenten der Plattform versuchen Daten vor Einsicht und Zugriff durch Unbefugte schützen, während die Daten zwischen Komponenten der Plattform übertragen werden. Eine mögliche Antwort auf diese Frage wäre die Verwendung von Protokollen, die Daten verschlüsselt übertragen.

- **Wie wird Authentizität der Übertragung / der Komponenten innerhalb der Plattform sichergestellt ?**
  Das Ziel dieser Frage war herauszufinden wie sich die unterschiedlichen Komponenten der Plattform vor einer Übertragung von Daten authentifizieren um sicherzustellen, dass Unbefugte nicht an der Kommunikation im Plattform Ökosystem teilnehmen. Eine mögliche Antwort auf diese Frage wäre die Verwendung von Authtifizierungsmechanismen wie X.509 Zertifikaten.

- **Wie wird Integrität der Übertragung innerhalb der Plattform(-komponenten)sichergestellt ?**
  Das Ziel dieser Frage war herauszufinden wie die unterschiedlichen Komponenten der Plattform versuchen auf ihnen erhobene und gespeicherte Daten vor Veränderungen durch Unbefugte oder Störungen schützen, während die Daten zwischen Komponenten der Plattform übertragen werden. Eine mögliche Antwort auf diese Frage wäre die Verwendung von Checksummen in den zur Übertragung von Daten verwendeten Protokollen.

- **Wie wird Vertraulichkeit von Daten innerhalb der Plattform sichergestellt ?**
  Das Ziel dieser Frage war herauszufinden wie die unterschiedlichen Komponenten der Plattform versuchen auf ihnen gespeicherte Daten vor Einsicht und Zugriff durch Unbefugte schützen. Eine mögliche Antwort auf diese Frage wäre die Verschlüsselung von Daten auf dem Gerät.

- **Wie wird die Nachvollziehbarkeit von Aktionen in der Plattform sichergestellt ?**
  Das Ziel dieser Frage war herauszufinden wie der Zugriff auf Daten und Veränderungen an Daten und Komponenten der Plattform nachvollzogen werden können. Änderungen an Daten und Plattformkomponenten nachvollziehen zu können ist relevant um die Schutzziele Integrity und Transparency zu schützen. Eine mögliche Antwort auf diese Frage wäre der Einsatz von Logging von Nutzeraktionen.

- **Gibt es ein Automatisches Patchmanagement ?**
  Ziel der Frage ist heraus zu finden, wie Updates in der Plattform durchgeführt werden um die Integrität der Komponenten sicherzustellen. So sollen die Komponenten vor neue entdeckten Schwachstellen durch veraltete Software geschützt werden.

- **Wie wird Verfügbarkeit der Plattform sichergestellt ?**
  Ziel der Frage ist heraus zu finden wie die Plattform Ausfälle verhindert und den Betrieb sicherstellt. Die Frage versucht zudem herauszufinden wie die Plattform im Fall eines Ausfall den Betrieb wieder aufnimmt. Eine mögliche Antwort auf diese Frage wäre der Einsatz von automatischen Backups um im Falle eines Ausfalls möglichst wenig Daten zu verlieren und schneller wieder einsatzbereit zu sein.

- **Verfügt die Plattform über Features zur automatischen Klassifikation von Daten ?**
  Um angemessenen Umgang mit datenschutzrelevanten Daten sicherzustellen ist notwendig, dass das benötigte Schutzniveau der Daten bei der Erhebung bekannt ist. Das Ziel dieser Frage ist herauszufinden ob die Plattform über ein System verfügt die Daten automatisch zu klassifizieren und einen dem benötigten Schutzniveau angemessenen Umgang zu forcieren.

- **Besonderheiten**
  Dieser Teil des Fragekatalogs hat als Ziel Mechanismen, die nicht von anderen Fragen erfasst werden oder organisatorische Aspekte der IT-Sicherheit oder des Datenschutz zu erfassen.

Zu der Authentifizierung und Autorisierung von Nutzern:

Um Confidentiality, Integrity und Unlinkability umzusetzen ist notwendig, dass nur authentifizierte Nutzer zugriff auf Daten erhalten, die sie auch Benötigen. Dazu soll untersucht werden wie diese Anforderungen innerhalb der Plattform umgesetzt werden.

- **Wie authentifizieren sich Nutzer im Plattform Ökosystem ?**
  Das Ziel dieser Frage war herauszufinden wie sich Nutzer an der Plattform authentifizieren. Eine mögliche Antwort auf diese Frage wäre die Verwendung von Authtifizierungsmechanismen wie X.509 Zertifikaten.

- **Wie werden Berechtigungen im Plattform Ökosystem vergeben ?**
  Das Ziel dieser Frage war herauszufinden nach welchen Schema Berechtigungen in der Plattform vergeben werden. Da dieses Schema meist Plattform weit eingesetzt wird gilt es in der Regel auch für Edge-Geräte. Eine mögliche Antwort auf diese Frage wäre der Einsatz von Role-Based Access Control.

Zur Übertagung von Daten aus der Plattform hinaus :

In diesem Bereich des Fragekatalogs wurde nur die Absicherung der Übertragung an das externe System betrachtet. Alle weiteren Anforderungen an die Sicherheit und den Datenschutz liegen in dem Fall bei dem Zielsystem, an das die Daten übertragen werden.

- **Wie wird Vertraulichkeit der Übertragung von der Plattform zu externen Anwendungen sichergestellt ?**
  Das Ziel dieser Frage war herauszufinden wie die Plattform versuchen auf ihr gespeicherte Daten vor Einsicht und Zugriff durch Unbefugte schützen, während die Daten aus dem Ökosystem der Plattform heraus übertragen werden. Eine mögliche Antwort auf diese Frage wäre die Verwendung von Protokollen, die Daten verschlüsselt übertragen.

- **Wie wird Authentizität der Übertragung / der Plattform zu externen Anwendungen sichergestellt ?**
  Das Ziel dieser Frage war herauszufinden wie sich die Plattform vor einer Übertragung von Daten authentifiziert, wenn sie aus ihrem eigenen Ökosystem heraus kommuniziert. Eine mögliche Antwort auf diese Frage wäre die Verwendung von Authtifizierungsmechanismen wie X.509 Zertifikaten.

- **Wie wird Integrität der Übertragung von der Plattform zu externen Anwendungen sichergestellt ?**
  Das Ziel dieser Frage war herauszufinden wie die Plattform versuchen auf ihr gespeicherte Daten vor Veränderungen durch Unbefugte oder Störungen schützen, während die Daten aus ihrem eigenen Ökosystem heraus übertragen werden. Eine mögliche Antwort auf diese Frage wäre die Verwendung von Checksummen in den zur Übertragung von Daten verwendeten Protokollen.

3.1.2. Auswertung des Fragekatalogs

Die Fragen wurden für jede der zwölf Plattformen mittels der öffentlich verfügbaren Dokumentation beantwortet. Die Antworten für die jeweiligen Plattformen befinden sich in Anhang A.2. Die Ergebnisse des Fragekatalog befinden sich aggregiert in den Tabellen 3.1, 3.2, 3.3 und 3.4.

Edgegeräte	
Wie wird Vertraulichkeit der Übertragung von der Edge zur Plattform sichergestellt ?	11 / 12 mal werden die jeweiligen Verbindungen durch TLS abgesichert. 2 / 12 mal kann zusätzlich ein VPN genutzt werden um zu den Edgegeräten zu verbinden. 1 / 12 mal DTLS genannt.
Wie wird Authentizität der Übertragung / des Thing von der Edge zur Plattform sichergestellt ?	9/12 Unterstützung für Zertifikate 5 / 12 mal über Tokens
Wie wird Integrität der Übertragung von der Edge zur Plattform sichergestellt ?	Nie expliziet erwähnt. TLS Stellt MAC zur Verfügung.
Wie wird Vertraulichkeit von Daten auf Edgegeräten sichergestellt ?	Falls verfügbar (5 / 12) nur die nötigen Tools bereitgestellt. Einmal by Default, aber nur auf dem Gateway.
Wie werden Updates durchgeführt ?	Falls es ein automatisches Patchmanagement (7 / 12)für die Edge gibt, dann gibt es auch ein Flottenmanagemnt. In der Regel OTA oder MQTT.
Gibt es ein Flottenmanagement (für On- und Offboarding / monitoring)?	10 / 12 verfügen über eine Form des Flottenmanagement. Patch / Config / Management : 7 / 12 Monitoring : 5 / 12 Onboarding : 4 / 12 Offboarding : 4 / 12

Tabelle 3.1.: Ergebnisse der Untersuchung in der Kategorie Edgegeräte.

Bei der Auswertung der Ergebnisse in den Tabellen 3.1, 3.2, 3.3 und 3.4 wurden die verwendeten Technologien nach der Häufigkeit ihrer Verwendung Kategorisiert. Um als häufige verwendet zu zählen muss die Technologie von mindestens 50 Prozent der untersuchten Plattformen verwendet werden. Das bedeutet in diesem Fall, dass mindestens sechs verschieden Plattformen diese Technologie verwenden. Als selten zählen Technologien, die von weniger als 50 Prozent der Plattformen überhaupt umgesetzt werden. Also Technologien die von fünf oder weniger Plattformen verwendet werden.

Technologien, die genutzt werden um das Schutzziel der Confidentiality zu erfüllen werden mit der Häufigkeit mit der die Technologien genutzt werden in Abbildung 3.2 dargestellt. Nach

Plattform	
Wie wird Vertraulichkeit der Übertragung innerhalb der Plattform(-komponenten) sichergestellt ?	11 / 12 mal werden die jeweiligen Verbindungen durch TLS abgesichert. 1 / 12 mal ein VPN genutzt.
Wie wird Authentizität der Übertragung / der Komponenten innerhalb der Plattform sichergestellt ?	2 / 12 Zertifikat 7 / 12 Tokenbasiert 2 / 12 Http Basic Auth 2 / 12 durch eigenen Identity Provider
Wie wird Integrität der Übertragunginnerhalb der Plattform(-komponenten)sichergestellt ?	Nie expliziet erwähnt. TLS Stellt MAC zur Verfügung.
Wie wird Vertraulichkeit von Daten innerhalb der Plattform sichergestellt ?	Für die Langfristige Speicherung wird die Verantwortng an den Datastore (DB) abgegeben. 5 / 12 Verschlüsselte speicherung in der Plattform by Default. Einmal keine Datenhaltung innerhalb der Plattform.
Wie wird die Nachvollziehbarkeit von Aktionen in der Plattform sichergestellt ?	9 / 12 Zentrales Logging
Gibt es ein Automatisches Patchmanagement ?	Ein automatisches Patchmanagement wird nur auf 3/12 Plattformen und nur für Geräte beworben.
Wie wird Verfügbarkeit der Plattform sichergestellt ?	6 / 12 regelmäßige Backups 3 / 12 Skalierung und Loadbalancing gegen DDoS 3 / 12 Nachrichten Buffer
Verfügt die Plattform über Features zur automatischen Klassifikation von Daten ?	4 / 12 Plattformen erlauben Datenquellen zu Klassifizieren
Besonderheiten	Keine Angabe da spezifisch für die jeweilige Plattform

Tabelle 3.2.: Ergebnisse der Untersuchung in der Kategorie Plattform.

Nutzerverwaltung	
Wie authentifizieren sich Nutzer im Plattform Ökosystem ?	4 /12 Eigener Identity Provider 3 / 12 MFA Support 5 / 12 OAuth 2.0 2 / 12 SAML 2 / 12 SSO 3 / 12 Basic / Username passwort 1 / 12 Azure AD / Zertifikat
Wie werden Berechtigungen im Plattform Ökosystem vergeben ?	12 / 12 RBAC 1 / 12 ABAC

Tabelle 3.3.: Ergebnisse der Untersuchung in der Kategorie Nutzerverwaltung.

der oben genannten Einschränkung waren der Einsatz von TLS um Vertraulichkeit und Integrität von Datenübertragungen zu gewährleisten häufig. Selten wurde ein VPN verwendet, um Übertragungen zu sichern. Selten wurde eine Möglichkeit bereitgestellt um die gespeicherten Daten

Verbindung mit Externen Anwendungen	
Wie wird Vertraulichkeit der Übertragung von der Plattform zu externen Anwendungen sichergestellt ?	Meist Anwendungs oder Plattformspezifisch, falls angegeben TLS (5 / 12) Ein mal verschlüsseltes PDF
Wie wird Authentizität der Übertragung / der Plattform zu externen Anwendungen sichergestellt ?	2 / 12 Anwendungsspezifisch 1 / 12 Eindeutiger API key mit Zertifikat
Wie wird Integrität der Übertragung von der Plattform zu externen Anwendungen sichergestellt ?	5 / 12 MAC in TLS

Tabelle 3.4.: Ergebnisse der Untersuchung in der Kategorie Verbindung mit Externen Anwendungen.

zu verschlüsseln oder die Daten standardmäßig zu verschlüsseln. Oftmals wurde die Verantwortung über die Vertraulichkeit der Daten an eine externe Speicherkomponente abgegeben. Alle betrachteten Plattformen verwenden *Role Based Access Control* (RBAC) als Autorisierungsschema für die Zugriffskontrolle für Nutzer und Komponenten innerhalb der Plattform. Eine Plattform kann zusätzlich zu RBAC auch noch *Atribute Based Access Control* (ABAC) verwenden. Technologien, die genutzt werden um das Schutzziel der Integrity zu erfüllen werden mit der Häufigkeit mit der die Technologien genutzt werden in Abbildung 3.3 dargestellt. Um Nachvollziehbarkeit, als Bestandteil der Schutzziele von Integrity und Transparency, sicherzustellen wurde häufig eine Form von zentralem Logging verwendet. Häufig verfügen die Plattformen auch über ein automatisches Patchmanagement für die Geräte an der Edge. Noch häufiger als ein automatisches Patchmanagement für die Edge ist eine Form von Flottenmanagement. Dabei ist auffällig, dass alle Plattformen die über ein automatisches Patchmanagement für die Edge verfügen auch die Anforderungen an ein Flottemanagement erfüllen. Innerhalb der Plattform waren zwei Arten der Authentifikation häufig. Geräte an der Edge authentifizieren sich meist über Zertifikate. Die Komponenten innerhalb der Plattform selbst authentifizieren sich hingegen häufig über einen Tokenmechanismus. Seltener kommen dabei eine Http Basic Authentication, eine Authentifikation über Username/Password oder an einem Identity Provider vor.

Technologien, die genutzt werden um das Schutzziel der Availability zu erfüllen werden mit der Häufigkeit mit der die Technologien genutzt werden in Abbildung 3.4 dargestellt. Selten wurde explizit erwähnt, dass durch Skalierung von Ressourcen oder den Einsatz von Messagebuffern versucht wird die Verfügbarkeit der Plattform aufrecht zu erhalten. Nur Backups wurden mit sechs Erwähnungen häufig genug gefunden um einen Trend festzustellen. Es besteht allerdings auch hier die Möglichkeit, dass die Verantwortung für die Verfügbarkeit von Daten an Speicherkomponenten außerhalb der Plattform abgegeben wird. Dies wurde allerdings in keiner der Dokumentationen explizit erwähnt.

Es wurden nur wenige, bis gar keine Technologien identifiziert um die Schutzziele Unlikability, Transparancy und Intervenability zu erfüllen. Technologien, die genutzt werden um das Schutzziel der Transparency zu erfüllen werden mit der Häufigkeit mit der die Technologien genutzt werden in Abbildung 3.5 dargestellt. Selten werden Daten bei ihrer Erhebung entsprechend ihres Schutzbedarfes klassifiziert. Es wurden keine Technologien identifiziert um Unlikability

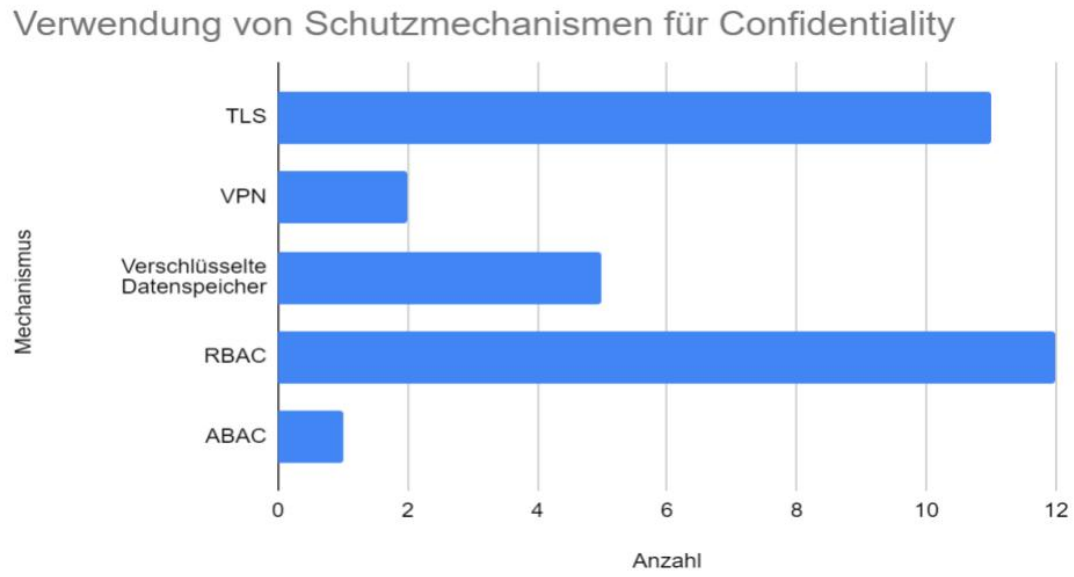

Abbildung 3.2.: Absolute Häufigkeit an Plattformen die verschiedene Technologien nutzen um das Schutzziel der Confidentiality zu erreichen..

Abbildung 3.3.: Absolute Häufigkeit an Plattformen die verschiedene Technologien nutzen um das Schutzziel der Integrity zu erreichen..

durchzusetzen, die nicht auch für die Confidentiality verwendt werden. Auffällig ist, dass das Ziel Intervenability über keine zugeordneten Technologien verfügt.

Abbildung 3.4.: Absolute Häufigkeit an Plattformen die verschiedene Technologien nutzen um das Schutzziel der Availability zu erreichen..

Ein Sonderfall war der Weg auf dem sich Nutzer innerhalb der Plattform authentifizieren. Es gab keine Technologie die häufig genug aufgetreten ist, um einen klaren Trend zu identifizieren. Allerdings verfügen alle untersuchten Plattformen über eine definierte Art, wie sich die Nutzer authentifizieren können. Die unterschiedlichen Technologien werden in Abbildung 3.6 dargestellt. Es ist zu beachten, dass OAuth 2.0 mit nur fünf Erwähnungen die am häufigsten verwendete Technologie in den zwölf Plattformen ist.

Es wurde selten ein sicherer Weg definiert um Daten aus dem Plattformökosystem hinaus zu transportieren. Aus diesen Beobachtungen ergibt sich die Schlussfolgerung, dass der Fokus bei der Entwicklung der untersuchten Plattformen der Fokus auf den Technologien lag, die klassisch als Teil der IT-Sicherheit zugeordnet wurden. Dazu zählen insbesondere die Ziele Confidentiality, Integrity und Availability. Methoden um den Datenschutz durchzusetzen wurden hingegen eher selten umgesetzt.

3.1.3. Erstellen eines Documentationtemplate aus dem Fragekatalog

Um die Dokumentation von Komponenten in der Entwicklung von IIoT-Plattformen zu erleichtern wurde aus dem Fragekatalog in 3.1.1 ein Dokumentationstemplate erstellt. Dazu wurden doppelte Fragen, z.B. die Frage nach der Vertraulichkeit der Übertragungen, entfernt. Dann wurden die Fragen zum Patchmanagement und dem Monitoring von Komponenten entfernt, da es sich dabei um Features der Plattform handelt die durch die Komponenten umgesetzt werden sollen. Zuletzt wurden dann die Technologien aus Tabellen 3.1,3.2,3.3,3.4 als Schlagworte hinzugefügt, um die Suche nach Antworten für das Template zu erleichtern.

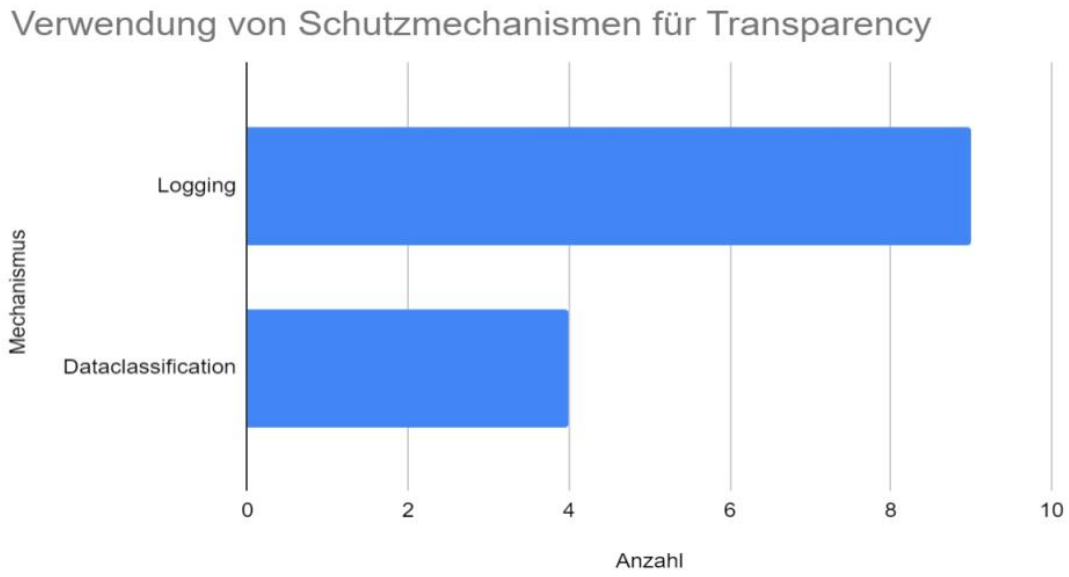

Abbildung 3.5.: Absolute Häufigkeit an Plattformen die verschiedene Technologien nutzen um das Schutzziel der Transparency zu erreichen..

Abbildung 3.6.: Absolute Häufigkeit verschiedener Technologien um Authentifizierung von Nutzern innerhalb der Plattformen durchzuführen..

Name	
Documentation	
Wie wird Vertraulichkeit der Übertragung sichergestellt Schlagworte : TLS, AMQP, Modbus, MQTT, HTTPS, IPSec, VPN	
Wie wird Authentizität der Übertragung sichergestellt ? Schlagworte : TLS, Signatur, Zertifikat, X.509	
Wie wird Integrität der Übertragung sichergestellt Schlagworte : TLS, HMAC, MAC	
Wie wird Vertraulichkeit von gespeicherten Daten sichergestellt ? Schlagworte : Verschlüsselung, AES	
Wie werden Updates durchgeführt ? Schlagworte :	
Wie wird die Nachvollziehbarkeit von Aktionen an der Komponente sichergestellt ? Schlagworte : Logging	
Wie wird Verfügbarkeit der Komponente sichergestellt ? Schlagworte : Nachrichtenbuffer, Loadbalancer, automatische Skalierung, Backup	
Verfügt die Plattform über Features zur automatischen Klassifikation von Daten ? Schlagworte :	
Wie authentifizieren sich Nutzer an der Komponente ? Schlagworte : Authentifizierung, Zugriffskontrolle, Zertifikat, Token, JWToken, OAuth	
Wie werden Berechtigungen in der Komponente vergeben ? Schlagworte : ABAC, RBAC	

Tabelle 3.5.: Dokumentarionstemplate.

3.2. Featuremodell

Die Ergebnisse des Fragekatalogs waren zu diesem Zeitpunkt nur eine Ansammlung verschiedener Technologien. Als nächstes wurden die Ergebnisse in zwei Kategorien unterteilt. Die erste

Kategorie war eine abstrakte Darstellung der Ergebnisse und wird im weiteren auch als Technologien bezeichnet. Beispiele für Ergebnisse des Fragekatalogs, das weiterhin als Technologien bezeichnet werden wären etwa „Verschlüsselter Datenspeicher“oder „Logging“. Es wird für diese Beispiele zwar angegeben, dass Daten verschlüsselt werden aber es wird nicht angegeben, wie die Daten verschlüsselt werden, also welche Verschlüsselungsalgorithmen oder welche Software zum verschlüsseln verwendet wird. Die zweite Kategorie bilden die ab dieser Stelle so genannten Umsetzungen. Als Umsetzungen werden Ergebnisse des Fragekatalogs bezeichnet, die eine konkrete Implementierung einer Technologie sind. Ein Beispiel für eine Umsetzung wäre ein Netzwerkprotokoll wie zum Beispiel MQTT.

Nach dem die Ergebnisse des Fragekatalogs in die zwei Kategorien eingeteilt waren wurden die Ergebnisse hierarchisch,nach dem Detailgrad der Ergebnisse geordnet. Dabei gelten Umsetzungen als detaillierter als Technologien. Auch innerhalb der einzelnen Kategorien konnten sich der Detailgrad unterscheiden. Bau zum Beispiel ein Protokoll wie HTTPS auf einem anderen Protokoll wie TLS auf so zählt das erste Protokoll, in diesem Beispiel HTTPS, als detaillierter.

Zuletzt wurden dann die Technologien mit den Schutzzielen verknüpft, die sie erfüllen.

Aus der Anordnung von Technologien und Umsetzungen sowie der Verbindung von Technologien und Schutzzielen entstand ein Featuremodell [17].

Die Notation des Featuremodell in dieser Arbeit wurde vereinfacht und weicht von der Notation in [17] ab, da viel Elemente der Standardnotation für diese Arbeit nicht nötig sind. Alle Verbindungen zwischen Elementen im Featuremodell dieser Arbeit entsprechen einem logischen `or`.

Das Featuremodell wurde mit dem Ziel erstellt die Ergebnisse des Fragekatalog so aufzubereiten, dass eine Basis entsteht aus der mögliche Methoden zum umsetzen von Schutzzielen abzulesen. Dieses Featuremodell soll dann die theoretische Basis für eine automatische Verifikation für die Erfüllung von Schutzzielen in Kapitel 4 bilden.

Das Featuremodell das die Ergebnisse der Tabellen 3.1,3.2,3.3 und3.4 ordnet ist in Abbildung 3.7 dargestellt.

Die oberste Ebene des Modell bilden dabei die Schutzziele Confidentiality, Integrity, Availability, Intervenability, Unlinkability, und Transparency. Confidentiality, Integrity, Availability stellen zentrale Anforderungen an sowohl die Sicherheit als auch den Datenschutz dar. Intervenability, Unlinkability, und Transparency sind spezifisch für den Datenschutz[14].

Die mittlere Ebene des Modells stellt die Technologien dar, die verwendet werden um die Ziele umzusetzen. Eine Technologie kann genutzt werden um mehrere Ziele zu erfüllen. So wird zum Beispiel die Technologie Access Control eingesetzt um sowohl Anforderungen an die Confidentiality als auch die Unlinkability umzusetzen.

Die unterste Ebene des Modells besteht aus konkret benannten Umsetzungen der Technologien der mittleren Ebene. Die Auswahl der Umsetzungen bezieht sich dabei auf die explizit genannten Umsetzungen der Technologien innerhalb der untersuchten IoT-Plattformen. Das Ziel der Intervenability hat keine Technologien oder Umsetzungen zugeordnet bekommen, da die Umsetzung dieses Ziels sich nicht durch eine zusätzliche Komponente innerhalb der Plattform realisieren lässt, sondern in der Konzeption und Architektur der kompletten Anwendung bedacht werden muss[14].

Eine navigierbare Association von einem Ziel zu einer Technologie stellt dar, dass die Technologie genutzt werden kann um einen Teil der Anforderungen des Zieles zu erfüllen. Ob eine

Technologie ausreicht um ein Ziel zu erfüllen hängt dabei von der Untersuchten Komponente ab. Zum Beispiel wird bei der Umsetzung des Ziel Confidentiality für Daten im Transit nur die Technologie Encryption und nicht Access Control benötigt. Bei ruhenden Daten werden allerdings sowohl Encryption als auch Access Control benötigt um die Anforderungen an das Ziel Confidentiality zu erfüllen.

Eine Navigierbare Association von einer Technologie zu einer anderen Technologie konkretisiert die Verwendung der Technologie. Zum Beispiel gibt es eine Association von der Technologie Encryption zu der Technologie Verschlüsselte Datenspeicher. Dabei wird die Verwendung der Technologie konkretisiert ohne eine genaue Umsetzung zu nennen. Im genannten Beispiel wird nur angemerkt, dass der Datenspeicher verschlüsselt wird, aber nicht wie.

Eine Navigierbare Association von einer Technologie zu einer Umsetzung stellt eine explizite Umsetzung der Technologie dar. Eine Association von einer Technologie zu mehreren Umsetzungen gibt dabei Alternativen an, die sich nicht gegenseitig ausschließen aber den selben Zweck erfüllen. Zum Beispiel kann die Technologie Encryption sowohl durch die Umsetzung IPSec als auch durch die Umsetzung TLS umgesetzt werden. Beide Umsetzungen können gleichzeitig betrieben werden und erfüllen den selben Zweck, Daten bei der Übertragung zu verschlüsseln.

Eine Navigierbare Association von einer Umsetzung zu einer anderen Umsetzung stellt dar, dass eine Umsetzung die andere Umsetzung verwendet, beziehungsweise auf ihr Aufbaut. Zum Beispiel gibt es eine Associacion von TLS nach HTTPS. Das HTTPS Protokoll verwendet TLS um Verbindungen zu verschlüsseln. Dabei wird bei den Protokollen davon ausgegangen, dass Securityfeatures verwendet werden, falls es möglich ist.

Abbildung 3.7.: Hierarchische Anordnung der Ergebnisse des Fragekatalog aus 3.1.1 in einem vereinfachten Featuremodell.

4. Implementierung in UML

Um die Erkenntnisse aus Kapitel drei im der Entwicklung neuer IIoT-Plattformen anwenden zu können soll ein UMLsec-Profil entwickelt werden. Dazu werden zwei Plugins entwickelt. Das erste Plugin stellt ein UML Profil zur Verfügung und ermöglicht Nutzern UML Modelle um Schutzziele und Technologien zur Einhaltung der Schutzziel zu erweitern. Dieses Plugin entspricht dem Profil des UMLsec-Profil und wird im Folgenden auch unter dem Namen „IoTComponentsProfilePlugin“bezeichnet. Das zweite entwickelte Plugin soll überprüfen ob die Schutzziele in einem Modell erfüllt werden und falls nötig Lösungsmöglichkeiten aufzeigen wenn die Schutzziele nicht erfüllt sind. Das zweite Plugin entspricht dabei dem Analysemodell des UMLsec-Profil und wird im Folgenden auch unter dem Namen „IoTComponentCheck“bezeichnet. Dieser Ansatz hat wie bereits in [24] beschrieben den Vorteil, dass das erste Plugin mit dem UML Profil unabhängig von, und ohne das zweite Plugin angewendet werden kann. Das UMLsec-Profil, also die Kombination des IoTComponentsProfilePlugin und des IoTComponentCheck, wird im Folgenden unter dem Namen „IoTComponentsProfile“erwähnt.

4.1. Verwendete Technologien

Die Grundlage für die in dieser Arbeit entwickelte Software bildet das Werkzeug *CompliAnce/Risk/Security-Model-Analyzer* (CARiSMA)[30].CARiSMA ist ein Werzeug um Analysen basierend auf UMLsec durchzuführen. CARiSMA wurde als Plugin für die *Integrated Development Environment* (IDE) *Eclipse* [9] entwickelt. Durch die Integration in die Eclipse IDE ist es CARiSMA möglich Modelle unterschiedlicher Modellierungstools zu analysieren. Da CARiSMA auf dem *Eclipse Modeling Framework* (EMF) aufbaut unterstützt CARiSMA auch domänenspezifische Modelliersprachen wie BPMN, was das Werkzeug vielseitiger einsetzbar macht. Das Werkzeug CARiSMA ist selbst nach einer Plugin Architektur aufgebaut. Die Plugin Architektur macht es möglich, dass CARiSMA mit neuen UML Profilen und automatischen Testverfahren (Checks) erweitert wird. Bei der Installation verfügt CARiSMA bereits über ein mehrere UML Profile, unter anderem UMLsec [30], und mehrere Checks. Durch Eclipse-Plugins lässt sich CARiSMA um neue Profile und Checks erweitern[30].

Zur Entwicklung der Software diese Arbeit wurde neben der Eclipse IDE und EMF noch *Eclipse Papyrus* [10] verwendet. Eclipse Papyrus integriert sich als Plugin in die Eclipse IDE und stellt einen Editor für UML Modelle zur Verfügung. Neben der als Plugin in die Eclipse IDE integrierten Version von Papyrus gibt es noch eine alleinstehende Version. Die alleinstehende Version von Papyrus wurde bei dieser Arbeit nicht betrachtet.

Die jeweiligen Versionen der Verwendeten Werkzeuge und Technologien befinden sich in Tabelle 4.1.

Name	Version
Java	16
Eclipse IDE	2021-12 (4.22.0)
Papyrus	6.0.0.202112011019
EMF	2.26

Tabelle 4.1.: Versionen der für diese Arbeit verwendeten Software.

4.2. Umsetzung des Featuremodel in ein CARiSMA Plugin

Das Plugin soll den Nutzern ermöglichen, ein UML Modell mit den Schutzzielen an die IT-Sicherheit und den Datenschutz, sowie mit den verwendeten Methoden um die Schutzziele zu erreichen zu erweitern. Dazu sollen diese Informationen als Stereotypen in einem UML-Profil dargestellt werden. Das Profil soll dazu eine Erweiterung des Profil UMLSec [16] umgesetzt werden. Das Featuremodell (siehe Abbildung 3.7) wurde daher erst in ein eigenes UML Profil Diagramm übertragen. Um das Featuremodell in ein UML Profil zu übertragen wurde die Navigationsrichtung invertiert und durch eine Extension ersetzt. So wurde aus einer gerichteten Associacion von TLS zu MQTT, „MQTT extends TLS“. Ein Beispiel wie die Übersetzung zwischen dem Featuremodell und dem UML Profil abläuft befindet sich in Abbildung 4.1.

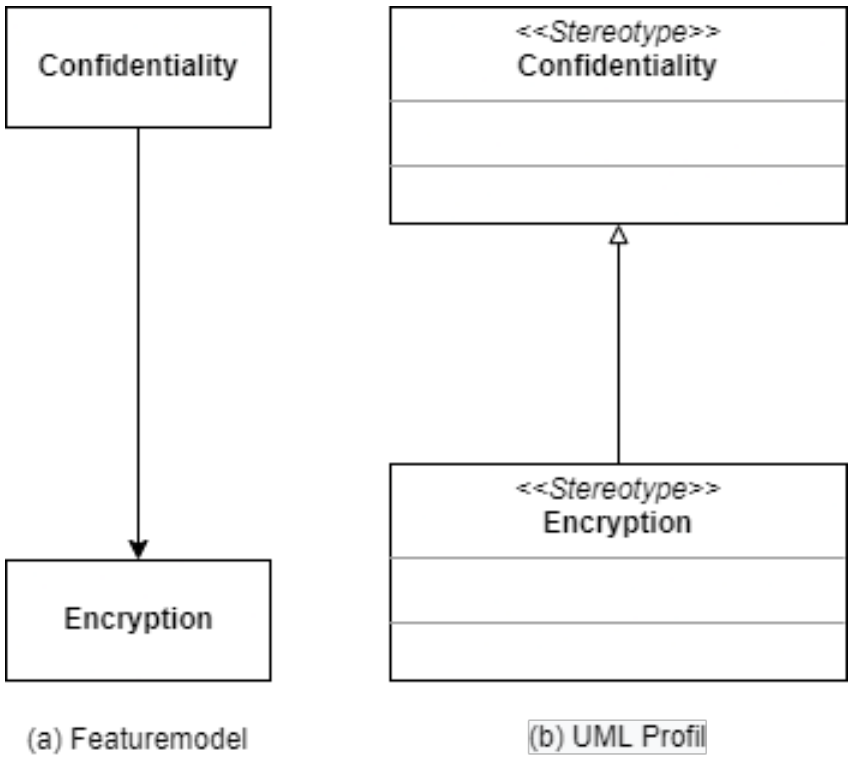

Abbildung 4.1.: Beispiel für die Übersetzung von der Inhalte aus dem Featuremodell zum UML Profil.

Als Metaklassen wurde für alle Ziele die `<<Metaclass>> Class` und `<<Metaclass>> Interface` angegeben um alle Stereotypes an Klassen und Interfaces anbringen zu können. Es wird davon ausgegangen, dass Klassen und Interfaces Komponenten in Modellen von (I)IoT-Plattformen repräsentieren können. Da das Featuremodell und somit auch das Profil eine Vielzahl an Technologien und Umsetzungen aufführt um die Confidentiality und Integrity von Daten im Transport zu schützen wurden die Ziele Confidentiality und Integrity zusätzlich die `<<Metaclass>> Associacion` angegeben. So soll es möglich sein auch die Confidentiality und Integrity von Daten im Transport überprüfen zu können. Die Zuordnung von Schutzzielen zu Metaklassen wird in Abbildung 4.2 dargestellt.

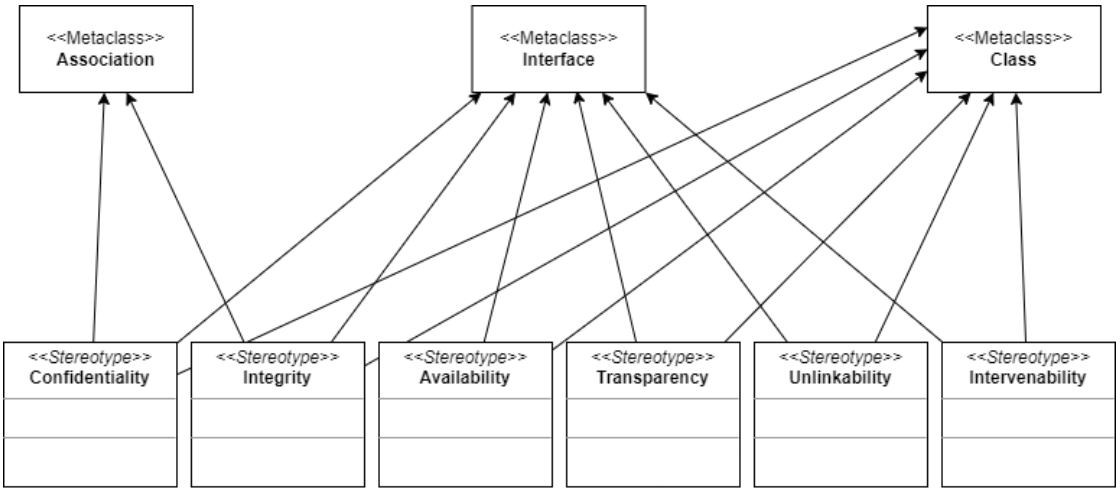

Abbildung 4.2.: Zuordnung von Schutzzielen zu Metaklassen im UML Profil.

Im Anschluss wurde auf das UML Profil Diagramm noch das Ecore Profil aus dem EMF angewandt. Das Ecore Profil ist nötig um aus dem UML Profil Diagramm Code für das Eclipse Plugin generieren zu können. Das Plugin wurde dann nach Anleitung der aus der Dokumentation des Papyrus User Guide erstellt. Die Anleitung um ein UML Profil Plugin zu erstellen sich im Help Menü der Eclipse IDE (zu finden unter „Help/Help Contents“) unter „Papyrus Guide > User Guide > Tasks > Using UML Profiles“finden. Das Ergebnis war das IoTComponentsProfilePlugin. Das komplette Profil ist in Abbildung A.1 abgebildet.

4.3. Erstellen eines Regelsatzes zur automatischen Verifikation der Einhaltung des Profils in einem CARiSMA Plugin

In einem zweiten Plugin wurde ein neuer Check für das Werkzeug CARiSMA erstellt. Das Ziel des Check ist zu überprüfen, ob ein Modell, welches das in dieser Arbeit entwickelte UML Profil anwendet, die modellierten Schutzziele erfüllt.

4.3.1. Theoretischer Ablauf der Verifikation

Zu erst soll geprüft werden, ob der Check anwendbar ist. Dazu soll geprüft werden, ob im untersuchten Modell mindestens ein Element existiert, das mit mindestens einem Schutzziel annotiert wurden. Wurde kein Element mit einem Ziel annotiert, wird ein Fehler ausgegeben und die Analyse beendet. Ist mindestens ein Element des Modell mit einem Schutzziel annotiert ist die Analyse durchführbar. Im folgenden wird der Einfachheit halber nur von Klassen gesprochen. Falls eine Regel oder ein Algorithmus für eine Klasse gilt, gilt es auch für ein Interface. Bei Interfaces wird dabei anstelle einer Association die Realisation äquivalent verwendet. Als Vorbereitung auf die Analyse wird versucht aus den Namen der Klassen im Modell weitere Informationen abzuleiten. Klassen deren Name bereits über die Informationen verfügt die ein Stereotyp des UML Profil vermittelt müssen so nicht extra annotiert werden, was dem Nutzer Arbeit abnimmt und das Modell übersichtlicher und leichter lesbar macht.

Nachdem die zusätzlichen Informationen gewonnen wurden wird für jedes der mit Zielen annotierten Elemente geprüft, ob die Ziele erfüllt sind. Die Prüfung, ob Schutzziele erfüllt sind, geschieht dabei nach folgenden Regeln: Für alle Beispiele gelten, dass das Stereotype `<<Confidentiality>>` von den Stereotypen `<<TLS>>` und `<<MQTT>>` erfüllt wird.

- **Regel 1:** Ein Ziel an einer Klasse gilt als erfüllt, wenn eine über ein Association verbundene Klasse mindestens ein Stereotyp besitzt, das das Ziel erfüllt. In diesem Fall wird davon ausgegangen, dass die zweite Klasse der ersten Klasse eine Technologie zur Verfügung stellt um das Ziel der ersten Klasse zu erfüllen. Ein Beispiel für diesen Fall Befindet sich in Abbildung 4.3. In Abbildung 4.4 befindet sich ein Beispiel, das diese Regel nicht erfüllen würde. Es werden nur die Stereotypen von Klassen betrachtet, die direkt über eine Association mit der untersuchten Klasse verbunden sind. Klassen die über mehr als eine Association und somit auch über weitere Klassen mit der untersuchten Klasse verbunden sind werden nicht betrachtet, da diese nicht unbedingt in die Funktionalität der untersuchten Klasse eingebunden sind.

Abbildung 4.3.: Ein Beispiel für ein Modell das Regel 1 erfüllt..

- **Regel 2:** Ein Ziel an einer Klasse gilt als erfüllt, wenn die Klasse selbst über mindestens ein Stereotype besitzt, das das Ziel erfüllt. In diesem Fall wird davon ausgegangen, dass die Klasse selbst über Technologien verfügt um das Ziel zu erfüllen. Ein einfaches Beispiel befindet sich in Abbildung 4.5.

- **Regel 3:** Ein Ziel an einer Association gilt als erfüllt, falls beide über die Association verbundenen Klassen mindestens ein gleiches Stereotyp besitzen, das das Ziel erfüllt. In diesem Fall wird verlangt, dass beide Klassen über das gleiche Stereotyp verfügen, da

Abbildung 4.4.: Ein Beispiel für ein Modell das Regel 1 nicht erfüllt..

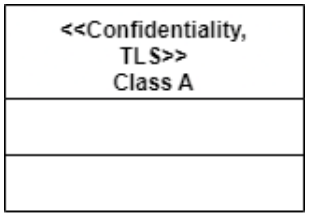

Abbildung 4.5.: Ein Beispiel für ein Modell das Regel 2 erfüllt..

bei einer Kommunikation von beiden Klassen über die annotierte Association die gleiche Technologie verwendet werden muss. So kann zum Beispiel bei der Kommunikation,die Association, zwischen zwei Komponenten, den Klassen, davon ausgegangen werden, dass die Kommunikation nur erfolgreich ist,falls beide Komponenten das gleiche Protokoll verwenden und das Protokoll eine Technologie darstellt die das Schutzziel der Kommunikation erfüllt. Ein Beispiel für ein UML Modell in dem diese Regel erfüllt wird befindet sich in Abbildung 4.6. Ein Beispiel für UML Modelle in denen diese Regel nicht erfüllt wäre befindet sich in Abbildung 4.7.

Abbildung 4.6.: Ein Beispiel für ein Modell das Regel 3 erfüllt..

Sind alle Ziele im Modell erfüllt ist die Analyse erfolgreich beendet. Die erfüllten Ziele werden aufgelistet und ausgegeben. Sind nicht alle Ziele erfüllt wird nach Lösungen gesucht. Als Lösungensvorschläge sollen für alle Elemente die über nicht erfüllte Schutzziele verfügen andere Elemente im Modell aufgelistet werden die über mindestens ein Stereotyp verfügen mit dem sich das Schutzziel erfüllen lässt und auflisten um welche(s) Stereotyp(en) das Schutzziel erfüllen. Danach wird die Analyse als nicht erfolgreich beendet. Dieser Ablauf ist in Abbildung 4.8 dargestellt.

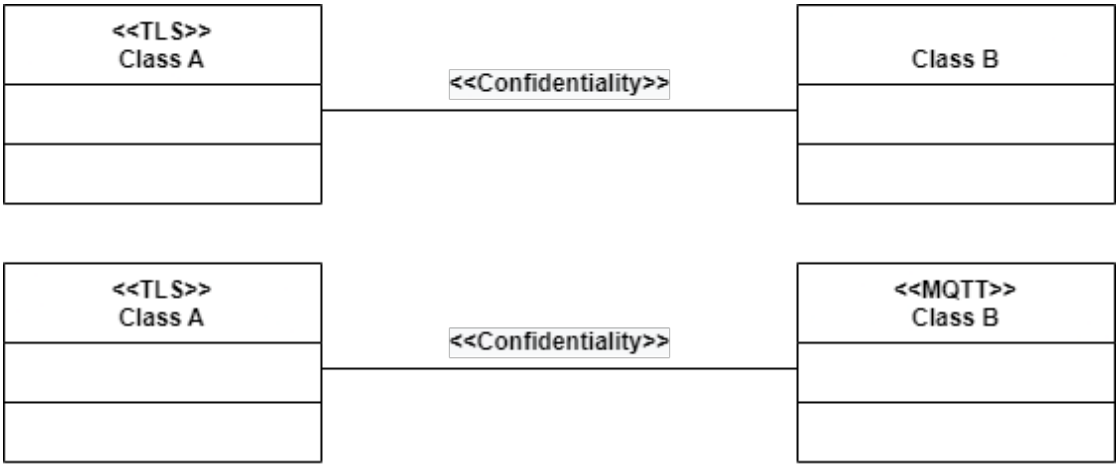

Abbildung 4.7.: Ein Beispiel für ein Modell das Regel 3 nicht erfüllt..

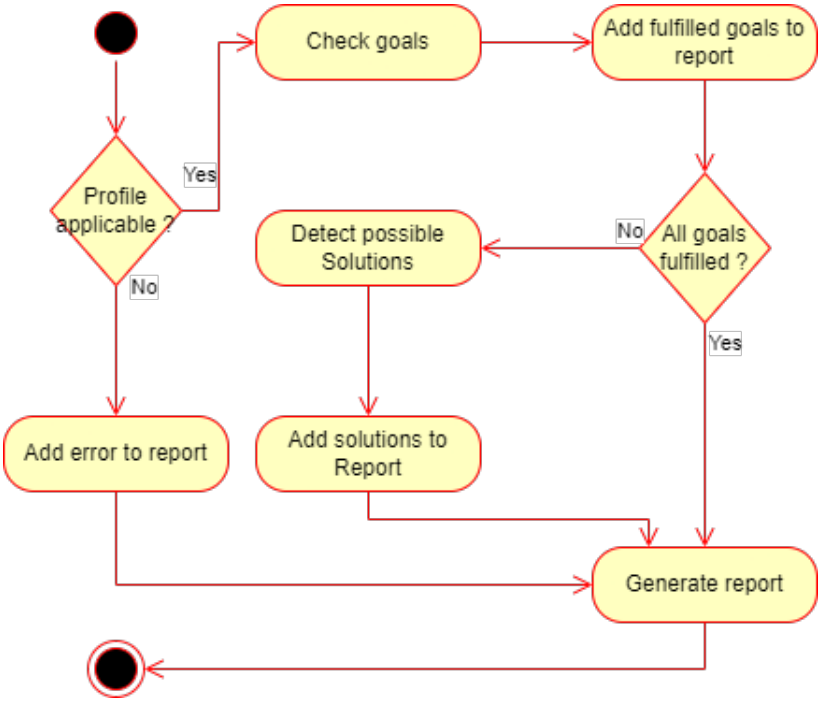

Abbildung 4.8.: Ablauf des Checks.

4.3.2. Umsetzung der Verifikation

Um mit möglichen Erweiterungen des Profils weiterhin funktionieren zu können wurde bei der Entwicklung der Verifikation darauf geachtet die Parameter dynamisch aus dem angewandten Profil zu generieren.

Der in Abbildung 4.8 abgebildete Kontrollfluss wird durch die in Listing 4.1 abgebildete

`perform` Methode durchgeführt.

```
1
2  public boolean perform(Map<String, CheckParameter> parameters, AnalysisHost
       host) {
3      this.host = host;
4      this.numOfElements = 0;
5      Resource currentModel = host.getAnalyzedModel();
6      if (currentModel.getContents().isEmpty()) {
7        host.addResultMessage(new AnalysisResultMessage(StatusType.WARNING, "
       Empty model"));
8        return false;
9      }
10
11     if (currentModel.getContents().get(0) instanceof Package) {
12       Package model = (Package) currentModel.getContents().get(0);
13       printContent(model, "");
14
15       // Check if Goals are applied to parts of the Model
16       if (!checkIsApplicable(model, false)) {
17         host.addResultMessage(new AnalysisResultMessage(StatusType.WARNING, "
       No verifiable Goals"));
18         return false;// No goal's => Check not applicable => Fail
19       }
20       host.appendLineToReport("
       ------------------------------------------------------------------------------
       ");
21       host.appendLineToReport("Goals detected");
22       host.appendLineToReport("
       ------------------------------------------------------------------------------\
       n");
23
24
25       Map<UMLsec, List<Stereotype>> mapping = generateFulfillmentMapping(
       model);
26       //Enrich model
27       enrichModel(model, mapping);
28       host.appendLineToReport("
       ------------------------------------------------------------------------------
       ");
29       host.appendLineToReport("Model after adding stereotypes based on names
       :");
30       printContent(model, "");
31
32       //Check which Goals are fulfilled
33
34       boolean returnBool = true;
35       HashMap<UMLsec, HashMap<Element, Boolean>> fullfilment = new HashMap
       <>();
36       for (UMLsec umLsec : GOALS) {
37         HashMap<Element, Boolean> goalFulfillment = checkGoalFulfillment(
       model, umLsec, mapping);
38         fullfilment.put(umLsec, goalFulfillment);
39
```

```
40          //Generate Message
41          if(goalFulfillment.isEmpty()) {
42            host.addResultMessage(new AnalysisResultMessage(StatusType.INFO,
    umLsec.toString() + " not applied to model"));
43          }else if(! goalFulfillment.keySet().stream().map(x -> goalFulfillment
    .get(x)).toList().contains(false)) {
44            host.addResultMessage(new AnalysisResultMessage(StatusType.INFO,
    umLsec.toString() + " fulfilled" ));
45          }else {
46            if(umLsec.equals(UMLsec.INTERVENABILITY_IOTCOMP)) {
47              host.addResultMessage(new AnalysisResultMessage(StatusType.
    WARNING, umLsec.toString() + " not fulfilled but the Profile doesn't
    provied Stereotypes to fulfill the  it"));
48            }else {
49              host.addResultMessage(new AnalysisResultMessage(StatusType.ERROR,
     umLsec.toString() + " not fulfilled" ));
50              returnBool = false;//unfulfilled always contains
    INTERVENABILITY_IOTCOMP if its in the model => set returnBool to false
51            }
52          }
53        }
54
55        // Make recomendations
56        HashMap<UMLsec, List<Element>> unfulfilled = spotUnfulfilledGoals(
    fullfilment);
57
58        if(unfulfilled.isEmpty()) {
59          return true;//No improvements needed => check not successful
60        }
61
62        recommendImprovements(model, unfulfilled, mapping);
63
64
65
66        return returnBool;//Improvements needed => check not successful
67      }
68
69      host.addResultMessage(new AnalysisResultMessage(StatusType.WARNING, "
    Content is not a model!"));
70      return false;
71    }
```

Quellcode 4.1: Die perfom Methode. Diese Methode gibt den Kontrollfluss des Code vor..

Die Methode `checkApplicable` prüft ob der Check ausführbar ist. Der Code der Methode `checkApplicable` ist in Listing 4.2 abgebildet. Um zu Prüfen ob der Check ausführbar ist werden rekursiv alle Elemente des Modell durchlaufen und es wird geprüft, ob ein Element mit einem Stereotype annotiert ist, das im statischen Array `GOALS` enthalten ist. Das statische Array `GOALS` beinhaltet die Stereotypen die den Schutzzielen entsprechen. Sobald mindestens ein Element über eins der Stereotypes verfügt gibt die Methode den Wahrheitswert `true` zurück. Wird kein solches Stereotype gefunden gibt die Methode den wert `false` zurück und die `perform` Methode beendet den Check mit einem Fehler. Dieser Teil des Code wird in den Zeilen 16 bis

19 der `perform` Methode in Listing 4.1 ausgeführt.

```
1  private boolean checkIsApplicable(Element element, boolean value) {
2    EList<Stereotype> appliedStereotypes = element.getAppliedStereotypes();
3    for (Stereotype stereotype : appliedStereotypes) {
4      for (UMLsec umLsec : GOALS) {
5        if(umLsec.isEqual(stereotype)) {
6          return true;
7        }
8      }
9    }
10   for (Element child : element.allOwnedElements()) {
11     value = value || checkIsApplicable(child, value);
12   }
13   return value;
14 }
```

Quellcode 4.2: Die checkApplicable Methode.

In Zeile 25 der `perform` Methode wird ein Mapping erstellt, das jedem Schutzziel eine Liste mit Stereotypen zuordnet die das jeweilige Schutzziel erfüllen. Dieser Schritt wird einmal durchgeführt da so das Mapping an mehreren Stellen im Programm übergeben werden kann statt jedes mal die Erfüllung eines Schutzziel erneut zu überprüfen.

Ist der Check ausführbar, so wird als nächstes versucht das Modell automatisch zu erweitern. Diese Funktionalität wird durch die in Listing 4.3 aufgeführte Methode `enrichModel` umgesetzt. Dazu wird versucht Klassen für den Check mit Stereotypes zu annotieren, die sich aus dem Namen der Klasse ergeben. Die durch den Check hinzugefügten Stereotypes existieren dabei nur für die Laufzeit der `perform` Methode und werden nicht im Modell gespeichert. Ein Stereotype wird an eine Klasse angefügt, falls der Name des Stereotypes im Namen der Klasse vorkommt. Dabei wird nicht auf Groß- und Kleinschreibung geachtet. Es muss aber der komplette Name des Stereotypes im Namen der Klasse enthalten sein. Diese Einschränkung ist relativ strikt um möglichst keine Fehler durch falsche Annahmen in das Modell einzufügen. Es erlaubt aber dennoch, dass eine Klasse mit Name „MQTT Broker“nicht extra mit dem Stereotype `<<MQTT>>` annotiert werden muss.

```
1  private void enrichModel(Package model, Map<UMLsec, List<Stereotype>> mapping
       ) {
2    //Generate list of all classes in the model
3    List<Element> classes = model.getOwnedElements().stream().filter(x -> x
     instanceof Class || x instanceof Interface).toList();
4    //Generate list of all stereotypes in the fulfillment mapping
5    List<Stereotype> stereotypes=mapping.values().stream().flatMap(Collection
     ::stream).collect(Collectors.toList()).stream().distinct().toList();
6    for (Element element : classes) {
7      List<Stereotype> applyStereotypes = new LinkedList<Stereotype>();
8      String className = ((NamedElement) element).getName();
9
10     for (Stereotype stereotype : stereotypes) {
11       if(className.equals(stereotype.getName())|| className.toLowerCase().
     contains(stereotype.getName().toLowerCase())) {
12         applyStereotypes.add(stereotype);
13       }
```

```
14        }
15
16        if(!applyStereotypes.isEmpty()) {
17          host.appendLineToReport("Applying Stereotypes " + applyStereotypes.
      stream().map(x -> x.getName()).toList().toString() +
18                  " to " + element.eClass().getName() + " because of the Name " +
      ((NamedElement) element).getName());
19          applyStereotypes.forEach(x -> element.applyStereotype(x));
20        }
21      }
22    }
```

Quellcode 4.3: Die enrichModel Methode.

Dieser Teil des Code wird in Zeile 26 der `perform` Methode in Listing 4.1 ausgeführt. Nach dem das Modell erweitert wurde wird geprüft welche der Ziele im Modell erfüllt sind. Dazu wird für jedes Ziel die Methode `checkGoalFulfillment` ausgeführt. Der Code der Methode `checkGoalFulfillment` wird in 4.4 abgebildet. Neben dem Modell und dem zu prüfenden Ziel wird ein Mapping übergeben, das jedem Ziel eine Liste mit allen Stereotypes, die das jeweilige Ziel erfüllen, zuordnet. Die Methode sucht in Zeile 6 des Listing 4.4 zuerst alle Elemente im übergebenen Modell die mit dem jeweiligen Ziel Annotiert sind. Dann wird für jedes mit dem Schutzziel annotierte Element überprüft ob das Ziel erfüllt ist. Dabei wird in Zeile 9, beziehungsweise Zeile 23 und Zeile 36 zuerst geprüft, ob es sich bei dem zu prüfenden Element um eine Klasse, ein Interface oder eine Association handelt. Handelt es sich um eine Klasse wird zuerst geprüft, ob die Klasse selbst über ein weiteres Stereotype verfügt, das das Ziel erfüllt. Falls dies der Fall ist wird das Ziel für die Klasse als erfüllt vermerkt. Das entspricht **Regel 2** um die Erfüllung zu prüfen und geschieht in Zeile 10, beziehungsweise Zeile 24 für Interfaces. Ist das Ziel noch nicht als erfüllt vermerkt werden alle Klassen betrachtet, die über eine Association mit der untersuchten Klasse verbunden sind. Verfügt eine der über eine Association mit der untersuchten Klasse verbunden Klassen über ein Stereotype, das das Ziel erfüllt, so wird die untersuchte Klasse als erfüllt vermerkt. Falls auch keine der über eine Association mit der untersuchten Klasse verbunden Klassen das Ziel erfüllt wird die untersuchte Klasse als nicht erfüllt vermerkt. Das entspricht **Regel 1** um die Erfüllung zu prüfen und geschieht in Zeile 12 bis Zeile 19 von Listing 4.4. Für Interfaces wird der selbe Prozess über die Realisierungen in den Zeilen 25 bis 33 durchgeführt. Ist das untersuchte Element eine Association so werden die Klassen geladen, die über die untersuchte Association verbunden sind. Falls beide über die Association verbundenen Klassen über mindestens ein Stereotype verfügen, das das Ziel erfüllt und beiden Klassen zugeordnet ist, wird vermerkt, dass die Association das Ziel erfüllt hat. Das entspricht **Regel 3** und geschieht in Zeile 37 bis Zeile 41. Als Ergebnis wird ein Mapping zurückgegeben in dem den jeweiligen Elementen ein Wahrheitswert zugeordnet wird der widerspiegelt, ob das Ziel erfüllt ist.

```
1
2 private HashMap<Element, Boolean> checkGoalFulfillment(Package model, UMLsec
      goal, Map<UMLsec , List<Stereotype>> mapping){
3     host.appendLineToReport("
      ----------------------------------------------------------------------------------
      ");
```

```
    host.appendLineToReport("Checking for " + goal.toString() + " fulfillment
    ");
    HashMap<Element, Boolean> result = new HashMap<Element, Boolean>();
    List<Element> annotatedElements = UMLsecUtil.getStereotypedElements(model
    , goal);
    for (Element element : annotatedElements) {
      boolean stereotypeFulfillment=false;
      if (element instanceof Class) {
        stereotypeFulfillment = element.getAppliedStereotypes().stream().map(
    x -> mapping.get(goal).contains(x)).toList().contains(true);//Fulfills
    Goal itself

        if(!stereotypeFulfillment) {// check associated classes
          for (Association association : ((Class) element).getAssociations())
     {
            List<Element> members = association.getRelatedElements();
            for (Element member : members) {
              EList<Stereotype> s = member.getAppliedStereotypes();
              stereotypeFulfillment = stereotypeFulfillment || s.stream().map
    (x -> mapping.get(goal).contains(x)).toList().contains(true);
            }
          }

        }

      }else if(element instanceof Interface) {
        stereotypeFulfillment = element.getAppliedStereotypes().stream().map(
    x -> mapping.get(goal).contains(x)).toList().contains(true);//Fulfills
    Goal itself
        if(!stereotypeFulfillment) {

          List<Element> allRealizationInModel = ((Interface) element).
    getPackage().allOwnedElements().stream().filter(x -> x instanceof
    InterfaceRealization ).toList();
          List<EList<NamedElement>> members = allRealizationInModel.stream().
    filter(x -> ((InterfaceRealization) x).getContract().equals(element)).map
    (x -> ((InterfaceRealization) x).getClients()).toList();
          List<NamedElement> distinctMembers = members.stream().flatMap(
    Collection::stream).collect(Collectors.toList()).stream().distinct().
    toList();
          for (Element member : distinctMembers) {
            EList<Stereotype> s = member.getAppliedStereotypes();
            stereotypeFulfillment = stereotypeFulfillment || s.stream().map(x
     -> mapping.get(goal).contains(x)).toList().contains(true);
          }
        }

      }else if (element instanceof Association) {
        List<EList<Stereotype>> memberStereotypes = ((Association) element).
    getRelatedElements().stream().map(x -> x.getAppliedStereotypes()).toList
    ();
        EList<Stereotype> memberOne = memberStereotypes.get(0);
        EList<Stereotype> memberTwo = memberStereotypes.get(1);
        List<Stereotype> both = memberOne.stream().filter(x -> memberTwo.
```

```
    contains(x)).toList();
41          stereotypeFulfillment = both.stream().map(x -> mapping.get(goal).
    contains(x)).toList().contains(true);
42        }else {
43          host.appendLineToReport(((NamedElement) element).getName() + " is of
    type " + ((NamedElement) element).eClass().toString() + " and shouldn't
    be applicable");
44          continue;
45        }
46        result.put(element, stereotypeFulfillment);
47        host.appendLineToReport(element.eClass().getName() + " " + ((
    NamedElement)element).getName() + (stereotypeFulfillment ? " fulfills " :
     " doesn't fulfill ") + goal.toString());
48      }
49
50      if(result.isEmpty()) {
51        host.appendLineToReport("No Element tries to fulfill " + goal.toString
    ());
52    }
53
54      return result;
55    }
```

Quellcode 4.4: Die checkGoalFulfillment Methode.

Die `checkGoalFulfillment` wird jeweils einmal für jedes Stereotyp im statischen `GOALS` Array durchgeführt. Für jedes Schutzziel wird dann in den Zeilen 41 bis 52 der `perform` Methode in Listing 4.1 ausgegeben ob es erfüllt ist oder nicht. Ist das Schutzziel erfüllt wird eine Meldung mit Level INFO ausgegeben. Ist das Schutzziel nicht erfüllt wird eine Meldung mit Level ERROR ausgegeben und der Rückgabewert der `perform` Methode wird auf `false` gesetzt. Je nachdem, ob `goalFulfillment`, der Rückgabewert der `checkGoalFulfillment` Methode, leer ist, nur den Wert `true` enthält oder mindestens einmal den Wert `false` enthält wird für das jeweilige Schutzziel ein Analyseergebnis gesetzt. Dabei gibt es in Zeile 46-47 einen Sonderfall für das Schutzziel Intervenability, da die aktuelle Version des Profil kein Stereotyp beinhaltet um das Schutzziel zu erfüllen. Anstatt das Ziel als Unerfüllt zu berichten und einen Fehler auszugeben wird eine Warnung ausgegeben. Der Rückgabewert der `perform` Methode wird in diesem Sonderfall auch nicht auf `false` gesetzt.

Nachdem für alle Ziele überprüft wurde, ob sie erfüllt sind, wird nach Lösungen gesucht um nicht erfüllte Ziele erfüllen zu können. Die Suche nach Lösungen wird von der in Listing 4.5 abgebildeten Methode `recommendImprovements` durchgeführt. Um eine mögliche Lösung für nicht erfüllte Ziele zu finden werden alle Klassen im Modell aufgelistet, die über mindestens ein Stereotype verfügen, das das Ziel erfüllt. Mit jeder aufgelisteten Klasse werden zudem noch alle Stereotypes der Klasse aufgelistet, die das Ziel erfüllen.

```
1
2 private void recommendImprovements(Package model, HashMap<UMLsec, List<
    Element>> unfulfilled,
3       Map<UMLsec, List<Stereotype>> mapping) {
4     List<Element> allClassesInModel = model.allOwnedElements().stream().
    filter(x -> x instanceof Class).toList();
5     for (UMLsec umLsec : unfulfilled.keySet()) {
```

```
6       for (Element element : unfulfilled.get(umLsec)) {
7         StringBuilder sb =new StringBuilder();
8         sb.append(((NamedElement) element).getName() + " recommendations for
    " + umLsec.toString() + ":\n" );
9         for (Element c : allClassesInModel) {
10          List<Stereotype> solutions = c.getAppliedStereotypes().stream().
    filter(x -> mapping.get(umLsec).contains(x)).toList();
11          if (!solutions.isEmpty()) {
12            List<String> solutionNames = solutions.stream().map(x -> x.
    getName()).toList();
13            sb.append(  "Class " + ((NamedElement) c).getName() + " provides
    : " + solutionNames.toString() + "\n");
14          }
15        }
16        host.appendToReport(sb.toString());
17      }
18      host.appendLineToReport("
    ------------------------------------------------------------------------------\
    n");
19     }
20
21  }
```

Quellcode 4.5: Die recommendImprovements Methode.

Nachdem die `recommendImprovements` Methode durchlaufen ist, ist der Check abgeschlossen. Der vollständige Code des Checks befindet sich in Listing A.1.

5. Evaluierung

Um die Effektivität des IoTComponentProfil zu demonstrieren und evaluieren wurde das Profil in den Design Modellen der IIP-Ecosphere Plattform angewendet. Zudem wurde das IoTComponentProfil mit dem UML Profil IoTsec verglichen, da IoTsec ebenfalls versucht die Sicherheit im Entwurf von IoT Systemen zu verbessern.

5.1. Fallstudie: Anwendung des neuen UML-Profil im IIP-Ecosphere Projekt

Um die Funktionsweise des IoTComponentProfil zu demonstrieren soll der in Abblidung 1.1 an einem Fallbeispiel der IIP-Ecosphere Plattform durchlaufen werden. Die IIP-Ecosphere Plattform ist eine sogenannte virtuelle Plattform [27]. Die virtuelle Plattform erhält ihre Daten von einer oder mehreren IoT-Plattformen, die mit den Geräten verbunden sind. Die virtuelle Plattform übernimmt dann die Aufgabe die ihr zugelieferten Daten zu Speichern oder zu Verarbeiten. Das Ziel der IIP-Ecosphere Plattform ist eine gesteigerte Effizienz, Produktivität, Flexibilität und Robustheit von Produktionsprozessen in der Industrie. Um dieses Ziel zu erreichen soll ein Werkzeugkasten für den Einsatz von Künstlicher Intelligenz in der Industrie 4.0 zur Verfügung gestellt werden.[1].

Wir betrachten das Modell der IIP-Ecosphere Plattform, beziehungsweise den View „Connectors“ des Modell. Ein Ausschnitt des Connectors View ist in Abbildung 5.2 dargestellt. Der ... wird in Abbildung 5.1 zusammengefasst.

Als erstes wird das IoTComponentProfilPlugin zum Modell der IIP-Ecosphere Plattform hinzugefügt. Dazu wird das UML Profil des IoTComponentProfil über das Menü „UML-Editor/Package/Apply Profile...“ zum Modell hinzugefügt. Als erstes werden dann die Schutzziele an entsprechenden Stellen im Modell, beziehungsweise View, hinzugefügt. Für diese Demonstration wird die Klasse „Connectors “ mit den Stereotypen `<< Confidentiality >>` und `<< Integrity >>` annotiert. Die Auswahl der für die Klasse Connectors relevanten Schutzziele erfolgte dabei in einer Diskussion mit der zuständigen Arbeitsgruppe des IIP-Ecosphere Projekt.

Als nächster Schritt wird die Dokumentation der externen Komponenten nach dem in Kapitel 3 vorgestellten Template (Tabelle 3.5) angefertigt. Dieser Schritt kann auch an einer früheren Stelle im Entwicklungszyklus durchgeführt werden, wenn die Dokumentation der externen Komponenten erstellt wird.

In dieser Demonstration werden die Komponenten „Eclipse Leshan“ und „Eclipse Californium “ betrachtet. Die ausgefüllten Dokumentationstemplates befinden sich in den Tabellen 5.1 und 5.2.

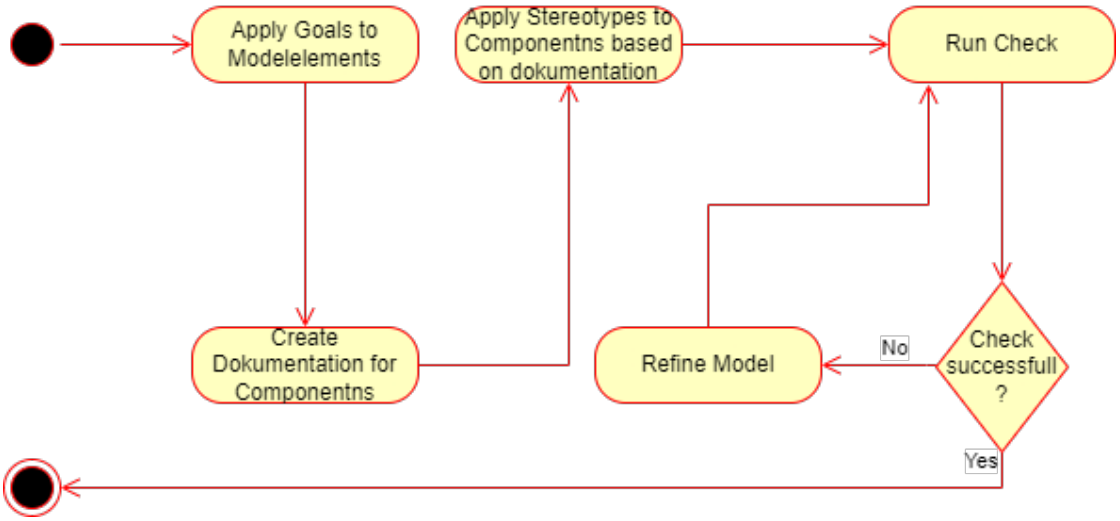

Abbildung 5.1.: Ablauf einer Beispielhaften Nutzung des IoTComponentProfil und des Dokumentationtemplate..

Aus der Dokumentation der externen Komponenten werden die Stereotype `<<HMAC,encryption>>` für die beiden Klassen Eclipse Californium und Eclipse Leshan abgeleitet.

Um die Analyse durchzuführen müssen die folgenden Schritte durchgeführt werden. Die Benutzeroberfläche von CARiSMA ist in Abbildung 5.3 dargestellt. Die Nummerierung in der Abbildung entsprechen der Nummerierung der Schritte.

1. Modell auswählen
2. Check hinzufügen
3. Check starten
4. Ergebnis

Das Ergebnis zeigt, dass nicht alle Schutzziel im Modell erfüllt sind. Das Ergebnis ist in Abbildung 5.4 abgebildet. Der detaillierte Report zeigt, dass weder Confidentiality noch Integrity an Connectors erfüllt ist. Als Vorschlag um die Integrity zu erfüllen wird auf `<<HMAC>>` von den Klassen Eclipse Californium und Eclipse Leshan hingewiesen. Als Vorschlag um die Confidentiality zu erfüllen wird auf `<<encryption>>` von den Klassen Eclipse Californium und Eclipse Leshan hingewiesen.

Um nach der fehlgeschlagenen Analyse die entdeckten Fehler zu beheben werden die Klassen Lwm2mConnector und CoapConnector mit den Stereotypen `<<encryption,HMAC>>` annotiert. Diese Annotation von Stereotypen können wir hier durchführen, da die Funktion der Klassen Lwm2mConnector und CoapConnector an dieser Stelle im Modell von den Klassen Eclipse Californium und Eclipse Leshan umgesetzt. Daher werden an dieser Stelle die Stereotypen der Klassen Eclipse Californium und Eclipse Leshan von den Klassen Lwm2mConnector und CoapConnector übernommen. Wird nachdem die Klassen Lwm2mConnector und CoapConnector mit

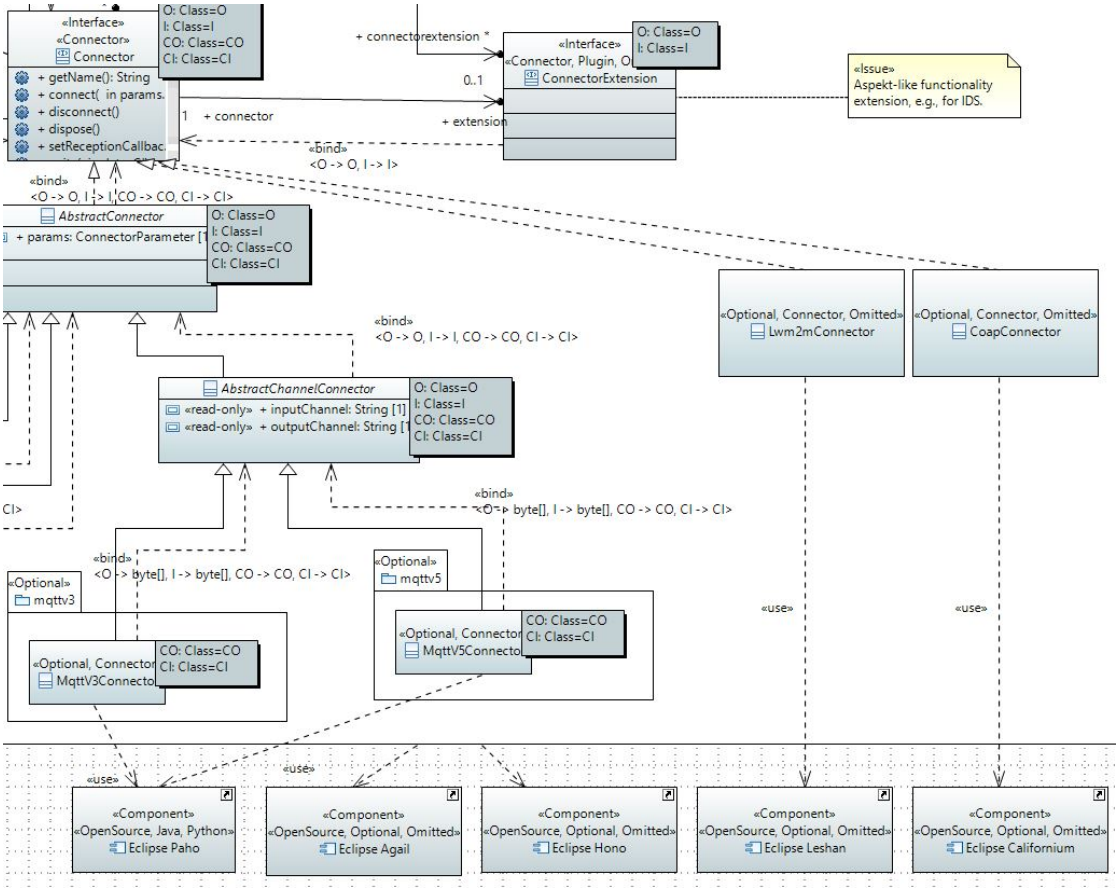

Abbildung 5.2.: Ausschnitt aus dem Connectors View des Modell der IIP-Ecosphere Plattform.

`<<encryption,HMAC>>` annotiert wurden wird die Analyse erneut ausgeführt. Das Ergebnis der zweiten Analyse ist ein erfolgreicher Check und in Abbildung 5.5 abgebildet.

5.2. Vergleich des neu erstellten UML-Profils mit bereits verfügbaren UML-Profilen

In [23] weisen die Autoren Robles-Ramirez, Escamilla-Ambrosio und Tryfonas auf das Problem hin, dass es wenige Möglichkeiten für Entwickler gibt Sicherheitsanforderung spezifisch für IoT Anwendungen zu modellieren. Als Lösung des Problem wird das UML Profil IoTsec vorgestellt. Im Folgenden wird das IoTsec Profil vorgestellt und mit dem IoTComponentsProfile verglichen.

5.2.1. Das IoTsec Profil

IoTsec führt neue Stereotypen ein um Nutzern des Profil zu ermöglichen Sicherheitsanforderungen an IoT Systeme grafisch darzustellen. Die von IoTsec neu eingeführten Stereotype beschreiben Konzepte um die Sicherheitsanforderungen an IoT Systeme zu durchzusetzen beziehungs-

Name	Eclipse Leshan
Documentation	`https://github.com/eclipse/leshan/wiki`
Wie wird Vertraulichkeit der Übertragung sichergestellt Schlagworte : TLS, AMQP, Modbus, MQTT, HTTPS, IPSec, VPN	Daten werden durch DTLS verschlüsselt
Wie wird Authentizität der Übertragung sichergestellt ? Schlagworte : TLS, Signatur, Zertifikat, X.509	Unterstüzuung für X.509 Zertifikate
Wie wird Integrität der Übertragung sichergestellt Schlagworte : TLS, HMAC, MAC	DTLS stellt Integrität durch HMAC sicher
Wie wird Vertraulichkeit von gespeicherten Daten sichergestellt ? Schlagworte : Verschlüsselung, AES	N/A
Wie werden Updates durchgeführt ? Schlagworte :	Muss manuell oder einen anderen Service durchgeführt wer
Wie wird die Nachvollziehbarkeit von Aktionen in der Komponente sichergestellt ? Schlagworte : Logging	N/A
Wie wird Verfügbarkeit der Komponente sichergestellt ? Schlagworte : Nachrichtenbuffer, Loadbalancer, automatische Skalierung, Backup	N/A
Verfügt die Komponente über Features zur automatischen Klassifikation von Daten ? Schlagworte :	Nein
Wie authentifizieren sich Nutzer an der Komponente ? Schlagworte : Authentifizierung, Zugriffskontrolle, Zertifikat, Token, JWToken, OAuth	Authentifizierung durch X.509
Wie werden Berechtigungen in der Komponente vergeben ? Schlagworte : ABAC, RBAC	Keine eigne Implementierung von Zugriffskontrolle.

Tabelle 5.1.: Anwenden des Dokumentarionstemplate auf Eclipse Leshan.

weise sie zu implementieren und geben keine Technologien vor. Zum Beispiel wird das Ste-

Name	Eclipse Californium
Documentation	`https://github.com/eclipse/californium/tree/master/scandium-core`
Wie wird Vertraulichkeit der Übertragung sichergestellt Schlagworte : TLS, AMQP, Modbus, MQTT, HTTPS, IPSec, VPN	Daten werden durch DTLS verschlüsselt
Wie wird Authentizität der Übertragung sichergestellt ? Schlagworte : TLS, Signatur, Zertifikat, X.509	Unterstüzuung für X.509 Zertifikate
Wie wird Integrität der Übertragung sichergestellt Schlagworte : TLS, HMAC, MAC	DTLS stellt Integrität durch HMAC sicher
Wie wird Vertraulichkeit von gespeicherten Daten sichergestellt ? Schlagworte : Verschlüsselung, AES	N/A
Wie werden Updates durchgeführt ? Schlagworte :	Muss manuell oder einen anderen Service durchgeführt werden
Wie wird die Nachvollziehbarkeit von Aktionen in der Komponente sichergestellt ? Schlagworte : Logging	N/A
Wie wird Verfügbarkeit der Komponente sichergestellt ? Schlagworte : Nachrichtenbuffer, Loadbalancer, automatische Skalierung, Backup	N/A
Verfügt die Komponente über Features zur automatischen Klassifikation von Daten ? Schlagworte :	Nein
Wie authentifizieren sich Nutzer an der Komponente ? Schlagworte : Authentifizierung, Zugriffskontrolle, Zertifikat, Token, JWToken, OAuth	Authentifizierung durch X.509
Wie werden Berechtigungen in der Komponente vergeben ? Schlagworte : ABAC, RBAC	Keine eigne Implementierung von Zugriffskontrolle

Tabelle 5.2.: Anwenden des Dokumentarionstemplate auf Eclipse Californium.

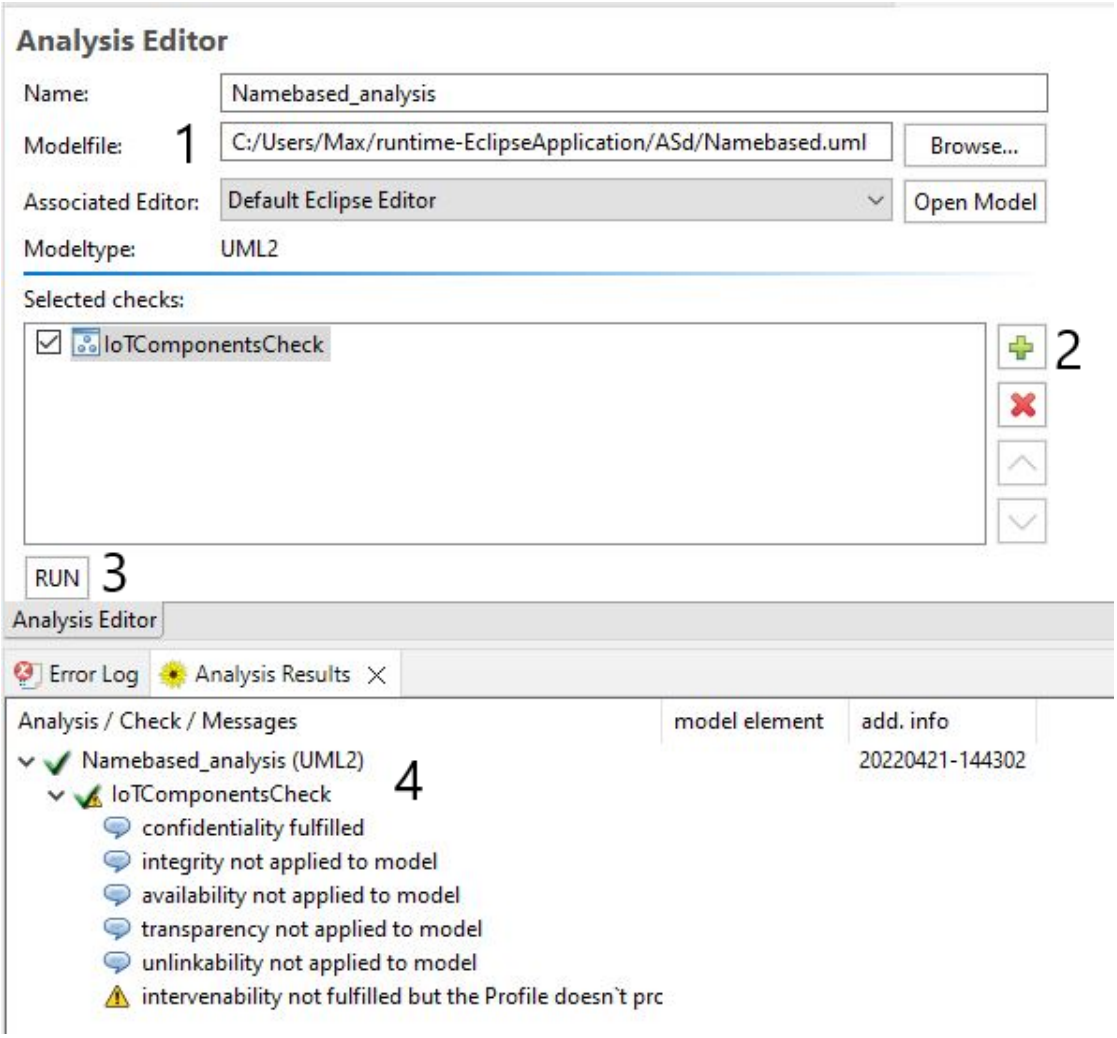

Abbildung 5.3.: Durchführen der CARiSMA Analyse.

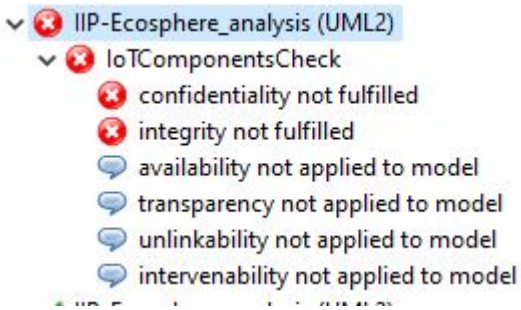

Abbildung 5.4.: Ergebnis der ersten Analyse des IIP-Ecosphere Modell. Die Schutzziele Confidentiality und Integrity sind nicht erfüllt..

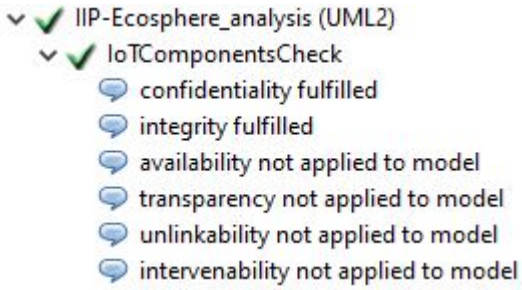

Abbildung 5.5.: Ergebnis der zweiten Analyse des IIP-Ecosphere Modell. Die Schutzziele Confidentiality und Integrity sind erfüllt..

reotyp `<<N: Authentication>>` eingeführt. IoTsec spezifiziert allerdings keine Technologie die die Authentifizierung durchführt. Die Authentifizierung könnte über sowohl ein X.509 Zertifikat als auch über einen Username/Passwort Mechanismus erfolgen. Das Ziel von IoTsec war es durch den kompletten Produkt-Lebenszyklus des IoT Systems anwendbar zu sein. Um im kompletten Produkt-Lebenszyklus einsetzbar zu sein unterstützt IoTsec zehn verschiedene UML Diagramm Arten [23].

5.2.2. Unterschiede zu IoTsec

Unterschiedliche Möglichkeiten die spezifischen Sicherheitsanforderung für IoT Anwendungen zu modellieren wurden in [23] bereits miteinander verglichen. Tabelle 5.3 wurde [23] entnommen und um einen Eintrag für das in dieser Arbeit entwickelte UMLsec-Profil als „UMLsec + IoTComponentsProfile“ erweitert.

Extension or language	Extension specific for IoT.	System security concerns model.	UML extension or visual representation	Security requirements modelling
UMLsec	×	✓	✓	✓
IoT-A	✓	×	✓	×
SysML	×	×	✓	×
SysMLsec	×	✓	✓	✓
UML4IoT	✓	×	✓	×
IBM approach	✓	×	✓	✓
ThingML	✓	×	×	×
UML	×	×	×	×
IoTsec	✓	✓	✓	✓
UMLsec + IoTComponentsProfile	✓	✓	✓	✓

Tabelle 5.3.: Vergleich verschiedener Erweiterungen um IoT Sicherheitsbedenken zu modellieren..

Entsprechend dem Vergleich in Tabelle 5.3 scheinen IoTsec und das in dieser Arbeit entwickelte UMLsec-Plugin den selben Zweck zu erfüllen. Allerdings unterscheiden sich die beiden Erweiterungen in mehreren Punkten.

1. **Zieldefinition**
   IoTsec betrachtet Sicherheitsanforderungen. Das IoTComponentsProfile betrachtet Schutzziele. Sicherheitsanforderungen sind präziser als Schutzziel und werden in der Regel aus den Schutzzielen abgeleitet. Zudem Betrachtet IoTsec nur die Sicherheitsschutzziele Confidentiality, Integrity und Availability. Das IoTComponentsProfile betrachtet zudem noch die Schutzziele Unlikability, Transparency und Intervenability, die dem Datenschutz zuzuordnen sind.

2. **Anwendungsbereich**
   IoTsec wurde entworfen um über den kompletten Produkt-Lebenszyklus eines IoT Systems einsetzbar zu sein. Dazu wird eine Vielzahl an UML Diagrammarten unterstützt. Das IoTComponentProfil kann nur auf Klassen und Associations in UML Diagrammen angewandt werden und ist nur für die Modellierung externer Komponenten im Entwurf von Plattformen entworfen. Dadurch ist der Anwendungsbereich des IoTComponentProfil eingeschränkter als IoTsec's.

3. **Detailgrad der Profile**
   Die Stereotypen von IoTsec sind absichtlich relativ abstrakt. Der Detailgrad entspricht etwa der Technologien-Ebene des Featuremodell aus Abbildung 3.7. Das IoTComponentProfil verfügt über einen höheren Detailgrad als IoTsec, indem auch konkrete Umsetzungen von Technologien modelliert werden können.

4. **Automatisierung**
   Durch den IoTComponentCheck verfügt das IoTComponentProfil über die Möglichkeit automatisiert eine Analyse durchzuführen und und eventuell Verbesserungen vorzuschlagen. Zum aktuellen Zeitpunkt verfügt IoTsec über keine solche Funktion.

Diese Unterschied sind zu Bedenken wenn eine Erweiterung für UML in der Entwicklung neuer IoT Systeme ausgewählt wird. Dabei gilt allerdings auch, dass sich IoTsec und das IoTComponentProfil nicht gegenseitig ausschließen, sondern auch in Kombination verwendet werden können.

6. Fazit

(I)IoT-Systeme und somit auch (I)IoT-Plattformen gewinnen immer mehr an Bedeutung. Daher ist es wichtig, dass die Entwickler solcher Systeme und Plattformen über Methoden und Verfahren verfügen, die Entwickler ermächtigen sichere Plattformen und System zu entwerfen und entwickeln. Um Entwicklern diese Methoden zur Verfügung zu stellen wurde im Rahmen dieser Arbeit wurde das IoTComponentsProfile erstellt. Es gibt neben IoTComponentsProfile und anderen Profilen einige weitere Methoden um die Entwicklung sicherer (I)IoT-Plattformen, die Daten angemessen schützen, zu unterstützen. Diese Methoden sind dabei auch nicht an die UML gebunden sondern können auch an anderen Stellen des Produktlebenszyklus der (I)IoT-System Einsatz finden. Zudem lässt sich das IoTComponentsProfile auch auf mehrere Arten erweitern um Entwickler im Entwurf von (I)IoT-Plattformen besser unterstützen zu können. Es könnte etwa ein Mechanismus zur Risikobewertung in das UML Profil integriert werden.

6.1. Verwandte Arbeiten

Es gibt neben dieser Arbeit noch weitere Ansätze um die Sicherheit und den Datenschutz von (I)IoT-Plattformen in der Entwicklungsphase zu verbessern. Diese Ansätze sind dabei auch nicht nur auf (I)oT-Plattformen beschränkt. Im weiteren Sinne sind auch Ansätze für die Entwicklung von IoT-Systemen oder eine nicht auf IoT-Systeme beschränkte Anwendung von UML für Verbesserungen in der Entwicklung von (I)IoT-Plattformen relevant. In 5.2 wurde bereits IoTsec [23] als ein mit dem IoTComponentsProfile vergleichbares UML Profil erwähnt. IoTsec bietet eine allgemeinere Perspektive durch die höhere Abstraktion und kann an mehr Stellen im Entwicklungszyklus angewendet werden. In [2] wird auf den Ansatz von Model-Based Privacy by Design eingegangen. Es wird ein Verfahren für eine Analyse und für eine Einschätzung für potenzielle Schäden auf Basis von Systemmodellen entwickelt. Es wird auch ein UML Profil vorgestellt um den Datenschutz in Modellen darzustellen. Das Profil stellt dabei Stereotypen für Strategien und Designpatterns zur Verfügung. Ein anderer Ansatz um die Sicherheit und den Datenschutz in IoT-Systemen zu verbessern wäre eine Verbesserung in der allgemeinen Qualität und Funktion der Systeme. Um die Qualität von IoT-Systemen allgemeinen zu erhöhen wird in [18] ThingML vorgestellt. ThingML bildet eine eigene Modelliersprache, ähnlich zu UML, mit Toolsupport und Übertragungsmöglichkeiten zu UML. Zudem stellt ThingML eine Möglichkeit zur Verfügung um automatisch Code zu generieren. Neben der Modellierung von System in UML gibt es weitere Ansätze um die Sicherheit und den Datenschutz von IoT-Systemen bereits in der Entwicklungsphase zu verbessern. In [3] wird auf die Herausforderungen für den sicheren Entwurf von IoT-Systemen hingewiesen. Um auf diese Herausforderungen reagieren zu können wird ein ausführliches Framework für den Entwurf sicherer IoT-Systeme vorgestellt. Das vorgestellte Framework betrachtet unter anderem Aspekte wie die physische Sicherheit der

Endgeräte, systemweite Authentifizierungs- und Zugriffskontrollmechanismen und die Sicherheit des Netzwerks. Alternativ wird in [22] ein Modell eingeführt um die für die Sicherheit von IoT-Systemen notwendigen Akteure und deren jeweiligen Anforderungen in Kontext zu setzen. Als Akteure werden Personen, technologische Ökosysteme, Prozesse und intelligente Objekte betrachtet. Danach wird dargestellt wie die Anforderungen an die Sicherheit des Systems zwischen den jeweiligen Akteuren in konkrete Anforderungen in der Entwurfsphase von Systemen übertragen werden können.

6.2. Ausblick

Das in dieser Arbeit entwickelte IoTComponentsProfile bietet eine Grundlage, die auf mehrere Arten weiterentwickelt werden kann. Sowohl das UML Profil als auch das Analysemodell, also der Check, können weiterentwickelt werden. Ein einfacher Weg das UML Profil weiter zu entwickeln wäre es um zusätzliche Schutzziele, Technologien und Umsetzungen zu erweitern. Die im UML Profil dieser Arbeit ausgewählten Technologien und Umsetzungen entstammen den Ergebnissen des Fragekatalog aus 3.1.1. Es gibt weitere Technologien, wie zum Beispiel *Intrusion Detection Systems* (IDS) um die Integrity zu schützen, oder Umsetzungen, wie das Protokoll OPC UA, die das Profil ergänzen können. Es könnten auch weitere Schutzziele hinzugefügt oder die bereits vorhandenen Schutzziele durch Subziele verfeinert werden. So wird zum Beispiel Authentizität in [8] als Schutzziels der IT-Sicherheit benannt, was in dieser Arbeit als Teil des Schutzziel der Integrity betrachtet wird. Das UML Profil könnte auch wie IoTsec für weitere UML Diagrammarten erweitert werden [23]. Alternativ könnten auch wie im UML Profil aus [2] Strategien und Designpattern zum Erfüllen der Schutzziel in das Profil aufgenommen werden. Eine andere Möglichkeit das IoTComponentsProfile zu erweitern wäre den Tag-Mechanismus für UML Profile zu nutzen, um den Stereotypen einen Risikowert zuzuordnen. So könnte in einem Beispiel dem Stereotype `<<Confidentiality>>` einer Klasse ein Risikowert von vier zugeteilt werden. Um das Schutzziel dieser Klasse zu erfüllen müssten die Technologien und Umsetzungen die zum Erfüllen des Schutzziel auch einen Risikowert von mindestens vier besitzen. Dieses Verfahren könnte umgesetzt werden indem die Risikowerte der Stereotype die das Schutzziel erfüllen würden aufsummiert werden. Eine solche Erweiterung würde auch Änderungen am Analysemodell nach sich ziehen um die Verifikation durchführen zu können.

Literaturverzeichnis

[1] *IIP-Eosphere*, 2021. https://www.iip-ecosphere.eu/.

[2] Amirshayan Ahmadian: *Model-based privacy by design*. doctoralthesis, Universität Koblenz-Landau, Universitätsbibliothek, 2020.

[3] Subho Shankar Basu, Somanath Tripathy und Atanu Roy Chowdhury: *Design challenges and security issues in the Internet of Things*. In: *2015 IEEE Region 10 Symposium*, Seiten 90–93, 2015.

[4] Ansgar Baums: *Digitale Plattformen – DNA der Industrie 4.0*, 2016. http://plattform-maerkte.de/dna/, Zuletzt aufgerufen 18.04.2022.

[5] M.A. Bishop: *Introduction to Computer Security*. Addison-Wesley, 2005, ISBN 9780321247445.

[6] Cloudflare: *Industrial Internet of Things (IIoT)*, 2021. https://www.cloudflare.com/learning/ddos/glossary/mirai-botnet/.

[7] George Danezis, Josep Domingo-Ferrer, Marit Hansen, Jaap Henk Hoepman, Daniel Le Métayer, Rodica Tirtea und Stefan Schiffner: *Privacy and Data Protection by Design*, 2015. https://www.enisa.europa.eu/publications/privacy-and-data-protection-by-design.

[8] Claudia Eckert: *IT-Sicherheit: Konzepte - Verfahren - Protokolle*. De Gruyter Oldenbourg, 2018, ISBN 9783110563900. https://doi.org/10.1515/9783110563900.

[9] Eclipse Foundation: *Eclipse Website*, Stand 20.04.22. https://www.eclipse.org/.

[10] Eclipse Foundation: *Papyrus Website*, Stand 20.04.22. https://www.eclipse.org/papyrus/.

[11] Object Management Group: *Object Constraint Language*, Februar 2014. https://www.omg.org/spec/OCL/2.4.

[12] Object Management Group: *Unified Modeling Language*, Juli 2011. https://www.omg.org/spec/UML/2.4.1.

[13] Object Management Group: *XML Metadata Interchange*, März 2014. https://www.omg.org/spec/XMI/2.4.2.

[14] Marit Hansen, Meiko Jensen und Martin Rost: *Protection Goals for Privacy Engineering*. In: *2015 IEEE Security and Privacy Workshops*, Seiten 159–166, 2015.

[15] Alex Jabolkov: *A Introduction to IoT Platforms*, 2016. https://www.ptc.com/en/blogs/iiot/introduction-iot-platforms.

[16] Jan Jürjens: *Secure systems development with UML*. Springer Science & Business Media, 2005.

[17] Kyo Kang, Sholom Cohen, James Hess, William Novak und A. Peterson: *Feature-Oriented Domain Analysis (FODA) Feasibility Study*. Technischer Bericht CMU/SEI-90-TR-021, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 1990. http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=11231.

[18] Brice Morin, Nicolas Harrand und Franck Fleurey: *Model-Based Software Engineering to Tame the IoT Jungle*. IEEE Software, 34(1):30–36, 2017.

[19] Object Management Group Object und Reference Model Subcommittee: *Model Driven Architecture*, 2014. https://www.omg.org/mda/specs.htm.

[20] OWASP: *OWASP Internet of Things Project*, 2019. https://wiki.owasp.org/index.php/OWASP_Internet_of_Things_Project.

[21] Sean Peasley, Tyler Lewis, Brian Wolfe, Robert Schmid und Mahesh Chandramouli: *Secure IoT by design Cybersecurity capabilities to look for when choosing an IoT platform*, October 2018. by Deloitte Development LLC.

[22] Arbia Riahi, Yacine Challal, Enrico Natalizio, Zied Chtourou und Abdelmadjid Bouabdallah: *A Systemic Approach for IoT Security*. In: *2013 IEEE International Conference on Distributed Computing in Sensor Systems*, Seiten 351–355, 2013.

[23] David Alejandro Robles-Ramirez, Ponciano Jorge Escamilla-Ambrosio und Theo Tryfonas: *IoTsec: UML extension for Internet of things systems security modelling*. In: *2017 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE)*, Seiten 151–156. IEEE, 2017.

[24] Thomas Ruhroth und Jan Jürjens: *Supporting Security Assurance in the Context of Evolution: Modular Modeling and Analysis with UMLsec*. In: *2012 IEEE 14th International Symposium on High-Assurance Systems Engineering*, Seiten 177–184, 2012.

[25] Christian Sauer, Holger Eichelberger, Amir Shayan Ahmadian, Andreas Dewes und Jan Jürjens: *Aktuelle Industrie 4.0 Plattformen – Eine Übersicht*. Nummer DE: IIP-2020/001, EN: IIP-2020/001-en. 2021. https://www.iip-ecosphere.eu/wp-content/uploads/2021/02/IIP-2020_001.pdfhttps://www.iip-ecosphere.eu/wp-content/uploads/2021/02/IIP-2020_001-en.pdfhttps://zenodo.org/record/4485756.

[26] Bernhard Schaetz, Alexander Pretschner, Franz Huber und Jan Philipps: *Model-Based Development*. Technischer Bericht, 2002.

[27] Klaus Schmid, Holger Eichelberger und Christian Kröher: *Domain-Oriented Customization of Service Platforms: Combining Product Line Engineering and Service-Oriented Computing*. Journal of Universal Computer Science (JUCS), 19:233–25, April 2013.

[28] Trendmicro: *Industrial Internet of Things (IIoT)*, 2021. https://www.trendmicro.com/vinfo/us/security/definition/industrial-internet-of-things-iiot.

[29] Trendmicro: *Internet of Things (IoT)*, 2021. https://www.trendmicro.com/vinfo/us/security/definition/internet-of-things.

[30] Website: *CARiSMA Website*, Stand 20.04.22. https://rgse.uni-koblenz.de/carisma/index.shtml, https://rgse.uni-koblenz.de/carisma/index.shtml.

[31] Jin Yong Yu und Young Gab Kim: *Analysis of IoT Platform Security: A Survey*. In: *2019 International Conference on Platform Technology and Service (PlatCon)*, Seiten 1–5, 2019.

A. Appendix

A.1. Begleitmaterial

Unter der URL `https://github.com/Eztrophie/BachelorarbeitUpload` können zusätzlich die Ergebnisse dieser Arbeit in folgender Dateistruktur abgerufen werden.

- /source
  - /carisma.check.iotcomponents : Source Code des CARiSMA Check Plugin
  - /carisma.profile.umlsec.iotcomponents : Source Code des Profil Plugins
- /eclipse : Installation der Eclipse IDE
- /modeling : Dateien des Eclipse Modeling Framework
- eclipse.exe : Eine Eclipse installation in der die Plugins bereits installiert sind
- Ausarbeitung.pdf : Eine digitale Fassung der Bachelorarbeit
- README.md : Eine Installationsanleitung

A.2. Tabellen mit Auswertung der untersuchten Plattformen

Plattform	**Amazon - AWS IoT**
Edgegeräte	
Wie wird Vertraulichkeit der Übertragung von der Edge zur Plattform sichergestellt	Unterstützte Protokolle mit TLS abgesichert.
Wie wird Authentizität der Übertragung / des Thing von der Edge zur Plattform sichergestellt	X.509 Zertifikate zu Authentifizierung und Amazon AWS Cognito Identitys. Die Verantwortung für die Authentifizierung liegt beim Message Broker
Wie wird Integrität der Übertragung von der Edge zur Plattform sichergestellt	MAC in TLS

Wie wird Vertraulichkeit von Daten auf Edgegeräten sichergestellt ?	FreeRTOS (Amazons OS für Micorcontroller) stellt PKCS# 11 Library zur Verfügung, diese muss aber selbst verwendet werden
Wie werden Updates durchgeführt ?	Over-the-Air (OTA) Updates mit signiertem Code
Gibt es ein Flottenmanagement (für On- und Offboarding / monitoring)	Umsetzung des Flottenmanagement durch AWS IoT Device Management, Device Defender und Greengrass. Ermöglichen Fernzugriff über einen VPN und Gruppenmanagement der Geräte.
Plattform	
Wie wird Vertraulichkeit der Übertragung innerhalb der Plattform(-komponenten) sichergestellt	Unterstützte Protokolle (Vorwiegend MQTT / HTTPS) mit TLS abgesichert
Wie wird Authentizität der Übertragung / der Komponenten innerhalb der Plattform sichergestellt	X.509 Zertifikate zu Authentifizierung und Amazon AWS Cognito Identitys
Wie wird Integrität der Übertragunginnerhalb der Plattform(-komponenten)sichergestellt	MAC in TLS
Wie wird Vertraulichkeit von Daten innerhalb der Plattform sichergestellt ? Wie wird die Nachvollziehbarkeit von Aktionen in der Plattform sichergestellt ?	Sämtliche Daten in AWS sind by default mit Ämazon owned keys"verschlüsselt Zentrales Logging über Amazon CloudTrail. Erlaubt echtzeit monitoring und lässt Events für Arlarmierungen einstellen.
Gibt es ein Automatisches Patchmanagement ?	Nur für devices durch ÄWS IoT Device Management".
Wie wird Verfügbarkeit der Plattform sichergestellt ?	Durch IoT Core wenden automatisch Backups und Device Shaddows erstellt, die nach einem ausfall Geräte wiederherstellen können. Verfügbarkeit der Plattform durch AWS Cloud Infrastruktur. Automatische Backups für Daten. AWS Cloud skaliert Ressourcen um DDoS abzufangen.
Verfügt die Plattform über Features zur automatischen Klassifikation von Daten ?	N/A

Besonderheiten	AWS IoT Device Defender : Überwachen von auffälligem verhalten an Edgegeräten, zentrales verwalten von Sicherheitseinstelleungen. Support für OCI container auf Edge
Nutzerverwaltung	
Wie authentifizieren sich Nutzer im Plattform Ökosystem ?	Amazon Cognito Identities (AWS Cloud IAM) innerhalb des Ökosystems. Accounts können zusätzlich über MFA abgesichert werden.
Wie werden Berechtigungen im Plattform Ökosystem vergeben ?	ABAC mit Rollen und Gruppen. Identity-based policies, Resource-based policies, Service control policies und Service control policies als JSON definierbar soweie Access control lists (ACLs) für Nutzer und Rollen.
Verbindung mit externen Anwendungen	
Wie wird Vertraulichkeit der Übertragung von der Plattform zu externen Anwendungen sichergestellt	Auf Basis von AWS Cloud Anwendungen. Unterstützte Protokolle mit TLS abgesichert.
Wie wird Authentizität der Übertragung / der Plattform zu externen Anwendungen sichergestellt	Auf Basis von AWS Cloud Anwendungen.Spezifisch für Anwendung.
Wie wird Integrität der Übertragung von der Plattform zu externen Anwendungen sichergestellt	Auf Basis von AWS Cloud Anwendungen. MAC in TLS.
Quellen	

`https://aws.amazon.com/de/iot/`
`https://aws.amazon.com/de/iot-core/`
`https://aws.amazon.com/de/freertos/`
`https://aws.amazon.com/de/greengrass/`
`https://aws.amazon.com/de/iot-device-defender/`
`https://docs.aws.amazon.com/iot/latest/developerguide/security.html`	

Tabelle A.1.: Auswertung AWS.

Plattform	**Bosch – Bosch IoT Suite**
Edgegeräte	
Wie wird Vertraulichkeit der Übertragung von der Edge zur Plattform sichergestellt	HTTP, WebSocket, AMQP, MQTT durch TLS gesichert.

Wie wird Authentizität der Übertragung / des Thing von der Edge zur Plattform sichergestellt	Entwerder durch Nutzername / Passwort oder X.509 Zertifikat
Wie wird Integrität der Übertragung von der Edge zur Plattform sichergestellt	MAC in TLS
Wie wird Vertraulichkeit von Daten auf Edgegeräten sichergestellt ?	Möglichkeit für Verschlüsselung in Remote Manager aber nicht by default
Wie werden Updates durchgeführt ?	firmware and/or software updates over the air (FOTA/SOTA). Automatisierbar durch device management
Gibt es ein Flottenmanagement (für On- und Offboarding / monitoring)	Bosch IoT Device Management unterstützt bei der Konfiguration und Monitoring der Geräte.
Plattform	
Wie wird Vertraulichkeit der Übertragung innerhalb der Plattform(-komponenten) sichergestellt	HTTP Webhooks, WebSocket, AMQP, MQTT, Apache Kafka, HTTPS durch TLS
Wie wird Authentizität der Übertragung / der Komponenten innerhalb der Plattform sichergestellt	Durch OAuth tokens, Unterstützug für OAuth2 und OpenID
Wie wird Integrität der Übertragunginnerhalb der Plattform(-komponenten)sichergestellt	MAC in TLS
Wie wird Vertraulichkeit von Daten innerhalb der Plattform sichergestellt ?	N/A
Wie wird die Nachvollziehbarkeit von Aktionen in der Plattform sichergestellt ?	Logging über IoT Insights
Gibt es ein Automatisches Patchmanagement ?	Für devices durch Device Management
Wie wird Verfügbarkeit der Plattform sichergestellt ?	Automatische Backups und Bulkupdates
Verfügt die Plattform über Features zur automatischen Klassifikation von Daten ?	N/A
Besonderheiten	Support für OCI container auf Edge. On Premise und Private Cloud Lösungen möglich
Nutzerverwaltung	

Wie authentifizieren sich Nutzer im Plattform Ökosystem ?	Amazon Cognito Identities (AWS Cloud IAM) innerhalb des Ökosystems. Accounts können zusätzlich über MFA abgesichert werden.
Wie werden Berechtigungen im Plattform Ökosystem vergeben ?	RBAC, verwaltet durch Customer Identity and Access Management (CIAM)
Verbindung mit externen Anwendungen	
Wie wird Vertraulichkeit der Übertragung von der Plattform zu externen Anwendungen sichergestellt	Anwendungsspezifisch
Wie wird Authentizität der Übertragung / der Plattform zu externen Anwendungen sichergestellt	Anwendungsspezifisch
Wie wird Integrität der Übertragung von der Plattform zu externen Anwendungen sichergestellt	Anwendungsspezifisch
Quellen	
https://bosch-iot-suite.com/ https://bosch-iot-suite.com/iot-device-management/ https://bosch-iot-suite.com/service/ bosch-iot-device-management/ https://bosch-iot-suite.com/service/rollouts/ https://bosch-iot-suite.com/service/remote-manager/ https://bosch-iot-suite.com/iot-data-management-analytics/ https://bosch-iot-suite.com/service/insights/ https://bosch-iot-suite.com/iot-edge/ https://bosch-iot-suite.com/service/ bosch-iot-edge-agent/ https://bosch-iot-suite.com/service/ bosch-iot-edge-services/ https://docs.bosch-iot-suite.com/device-management/ Feature-list.html https://docs.bosch-iot-suite.com/hub/concepts/ security/ https://docs.bosch-iot-suite.com/rollouts/Features. html https://docs.bosch-iot-suite.com/remote-manager/ on-premises/v71/en/index.htm#rm_overview.htm https://bosch-iot-insights.com/static-contents/ docu/html/Introduction.html https://docs.bosch-iot-suite.com/edge/#108408.htm	

Tabelle A.2.: Auswertung Bosch – Bosch IoT Suite.

Plattform	**B&R - Automation mapp Technology**
Edgegeräte	
Wie wird Vertraulichkeit der Übertragung von der Edge zur Plattform sichergestellt	N/A
Wie wird Authentizität der Übertragung / des Thing von der Edge zur Plattform sichergestellt	N/A
Wie wird Integrität der Übertragung von der Edge zur Plattform sichergestellt	N/A
Wie wird Vertraulichkeit von Daten auf Edgegeräten sichergestellt ?	N/A
Wie werden Updates durchgeführt ?	N/A
Gibt es ein Flottenmanagement (für On- und Offboarding / monitoring)	N/A
Plattform	
Wie wird Vertraulichkeit der Übertragung innerhalb der Plattform(-komponenten) sichergestellt	Zugriff durch verschlüsseltes VPN
Wie wird Authentizität der Übertragung / der Komponenten innerhalb der Plattform sichergestellt	Zertifikat bei VPN login
Wie wird Integrität der Übertragunginnerhalb der Plattform(-komponenten)sichergestellt	VPN implementierung
Wie wird Vertraulichkeit von Daten innerhalb der Plattform sichergestellt ?	N/A
Wie wird die Nachvollziehbarkeit von Aktionen in der Plattform sichergestellt ?	Alle Benutzeraktionen werden mit einem Zeitstempel und einem Benutzernamen protokolliert
Gibt es ein Automatisches Patchmanagement ?	N/A
Wie wird Verfügbarkeit der Plattform sichergestellt ?	N/A
Verfügt die Plattform über Features zur automatischen Klassifikation von Daten ?	N/A

Besonderheiten	Integrierte Firewall
Nutzerverwaltung	
Wie authentifizieren sich Nutzer im Plattform Ökosystem ?	N/A
Wie werden Berechtigungen im Plattform Ökosystem vergeben ?	RBAC, Berechtigungen werden von Gatemanager verwaltet
Verbindung mit externen Anwendungen	
Wie wird Vertraulichkeit der Übertragung von der Plattform zu externen Anwendungen sichergestellt	Export als verschlüsselte PDF möglich
Wie wird Authentizität der Übertragung / der Plattform zu externen Anwendungen sichergestellt	N/A
Wie wird Integrität der Übertragung von der Plattform zu externen Anwendungen sichergestellt	N/A
Quellen	
`https://www.br-automation.com/en/products/software/mapp-technology/` `https://www.br-automation.com/en/products/software/mapp-technology/mapp-cockpit/` `https://www.br-automation.com/en/products/software/mapp-technology/mapp-control/` `https://www.br-automation.com/en/products/software/mapp-technology/mapp-view/`	

Tabelle A.3.: Auswertung B&R - Automation mapp Technology.

Plattform	**Cisco - Kinetic**
Edgegeräte	
Wie wird Vertraulichkeit der Übertragung von der Edge zur Plattform sichergestellt	Edge: alle Verbindungen durch TLS gesichert ; Gateway: VPN, TLS und IPsec
Wie wird Authentizität der Übertragung / des Thing von der Edge zur Plattform sichergestellt	Edge: Passwort oder Token ; Gateway: Passwort oder Token
Wie wird Integrität der Übertragung von der Edge zur Plattform sichergestellt	MAC in TLS
Wie wird Vertraulichkeit von Daten auf Edgegeräten sichergestellt ?	Verschlüsselung empfohlen aber kein mechanismus bereitgestellt bzw expliziet erwähnt

Wie werden Updates durchgeführt ?	N/A
Gibt es ein Flottenmanagement (für On- und Offboarding / monitoring)	Cisco Kinetic Gateway Management Module (GMM) erlaubt monitoring und Management von Gateway komponenten. Keine Angabe zu einzelnen Geräten.
Plattform	
Wie wird Vertraulichkeit der Übertragung innerhalb der Plattform(-komponenten) sichergestellt	Alle unterstützten Protokolle(MQTT, RMQ, HTTPS) erfordern TLS
Wie wird Authentizität der Übertragung / der Komponenten innerhalb der Plattform sichergestellt	Innerhalb der Komponenten durch Passwort oder Token
Wie wird Integrität der Übertragunginnerhalb der Plattform(-komponenten)sichergestellt	MAC in TLS
Wie wird Vertraulichkeit von Daten innerhalb der Plattform sichergestellt ?	Keine Datenhaltung auf Cisco DCM
Wie wird die Nachvollziehbarkeit von Aktionen in der Plattform sichergestellt ?	Zentrales Logging für Gateway komponenten mit täglichen verschlüsselten backups
Gibt es ein Automatisches Patchmanagement ?	Nein bzw Keine Angabe
Wie wird Verfügbarkeit der Plattform sichergestellt ?	Lokale Buffer für Daten im Falle eines Netzwerkausfalls, Multi-Hop-Kommunikation zwischen Brokern
Verfügt die Plattform über Features zur automatischen Klassifikation von Daten ?	Data Control Module (DCM) bietet die Möglichkeit, Richtlinien zu erstellen, die Daten je nach Gerätetyp für verschiedene Anwendungen verfügbar machen, oder mithilfe von Regelwerken benutzerdefinierte Regeln festzulegen.
Besonderheiten	Umfangreiche möglichkeiten um Regeln zur durchsetzung von Dateneigentum
Nutzerverwaltung	
Wie authentifizieren sich Nutzer im Plattform Ökosystem ?	Multi-Factor Authentication (MFA) und Whitelists für Administrator accounts
Wie werden Berechtigungen im Plattform Ökosystem vergeben ?	RBAC in der Cloud

Verbindung mit externen Anwendungen	
Wie wird Vertraulichkeit der Übertragung von der Plattform zu externen Anwendungen sichergestellt	Alle API's über TLS verschlüsselt; Verbindungen zu anderen Plattformen (Microsoft Azure oder IBM Watson) über MQTT über TLS
Wie wird Authentizität der Übertragung / der Plattform zu externen Anwendungen sichergestellt	Authentifizierung über API Key's. Jeder API key ist einzigartig. Validierung der Server Zeertifikate.
Wie wird Integrität der Übertragung von der Plattform zu externen Anwendungen sichergestellt	MAC in TLS
Quellen	

`https://www.cisco.com/c/dam/en/us/solutions/collateral/internet-of-things/kinetic-datasheet-gmm.pdf`
`https://www.cisco.com/c/dam/en/us/solutions/collateral/internet-of-things/kinetic-datasheet-efm.pdf`
`https://www.cisco.com/c/dam/en/us/solutions/collateral/internet-of-things/kinetic-datasheet-dcm.pdf`
`https://www.cisco.com/c/en/us/solutions/internet-of-things/iot-kinetic.html#~capabilities`
`https://www.cisco.com/c/dam/en/us/td/docs/cloud-systems-management/kinetic/tech_notes/kinetic-security.pdf`	

Tabelle A.4.: Auswertung Cisco Kinetic.

Plattform	**General Electrics – Predix**
Edgegeräte	
Wie wird Vertraulichkeit der Übertragung von der Edge zur Plattform sichergestellt ?	Unterstützung für EGD (Ethernet Global Data), MQTT, Modbus, OPC-UA, OSI-Pi an der Edge. Management über HTTPS. Übertragungen an die Cloud über HTTPS, wss, MQTT, gRPC. Alles durch TLS gesichert. zusätzlich Site-to-Site VPN.
Wie wird Authentizität der Übertragung / des Thing von der Edge zur Plattform sichergestellt ?	X.509 Zertifikate

Wie wird Integrität der Übertragung von der Edge zur Plattform sichergestellt ?	MAC in TLS
Wie wird Vertraulichkeit von Daten auf Edgegeräten sichergestellt ?	N/A
Wie werden Updates durchgeführt ?	Fleetmanagement durch Predix Edgemanager
Gibt es ein Flottenmanagement (für On- und Offboarding / monitoring) ?	Predix Edge Manager verfügt über mehrere Microservices um das Management zu erleichtern
Plattform	
Wie wird Vertraulichkeit der Übertragung innerhalb der Plattform(-komponenten) sichergestellt ?	Alle verbindungen durch TLS gesichert
Wie wird Authentizität der Übertragung / der Komponenten innerhalb der Plattform sichergestellt ?	N/A
Wie wird Integrität der Übertragunginnerhalb der Plattform(-komponenten)sichergestellt ?	MAC in TLS
Wie wird Vertraulichkeit von Daten innerhalb der Plattform sichergestellt ?	Credential Store and Encryption Vault Service erlaubt sicheres ablegen von Tokens, Passwörtern und API keys, sowie einen Service zum verschlüsseln von Daten
Wie wird die Nachvollziehbarkeit von Aktionen in der Plattform sichergestellt ?	Zentrales Logging über Predix Audit Service über REST API erreichbar.
Gibt es ein Automatisches Patchmanagement ?	N/A
Wie wird Verfügbarkeit der Plattform sichergestellt ?	N/A
Verfügt die Plattform über Features zur automatischen Klassifikation von Daten ?	Verfügt über Möglichkeiten zur Datenklassifizierung.
Besonderheiten	Anwendung als Dockercontainer auf der Edge. Eigenes Edgebetriebssystem basierend auf Yocto Linux. Datenklassifizierung und verarbeitung. Starker Fokus auf Security support durch GE in form von SOC und Pentests
Nutzerverwaltung	

Wie authentifizieren sich Nutzer im Plattform Ökosystem ?	User Account and Authentication (UAA) service, integrierter OAuth 2.0 service und anbindung über SAML and drittanbieter identity Provider
Wie werden Berechtigungen im Plattform Ökosystem vergeben ?	Definiert über Permission sets. Durch Group-feature RBAC möglich. Zusätzlich erlaubt Access Control Services (ACS) Policies zu definieren und anbindung an Spring Security
Verbindung mit externen Anwendungen	
Wie wird Vertraulichkeit der Übertragung von der Plattform zu externen Anwendungen sichergestellt	TLS über vordefinierte Konnektoren
Wie wird Authentizität der Übertragung / der Plattform zu externen Anwendungen sichergestellt	N/A
Wie wird Integrität der Übertragung von der Plattform zu externen Anwendungen sichergestellt	MAC in TLS
Quellen	

`https://www.ge.com/digital/iiot-platform/predix-essentials`
`https://www.ge.com/digital/iiot-platform`
`https://www.ge.com/digital/iiot-platform/predix-edge`
`https://www.ge.com/digital/iiot-platform/cyber-security-trust-center`
`https://www.predix.io/resources/security`
`https://www.ge.com/digital/documentation/predix-essentials/latest/user-permission-sets.html`
`https://www.ge.com/digital/documentation/predix-platforms/sls.html`
`https://www.ge.com/digital/documentation/predix-platforms/acs-overview.html`
`https://www.ge.com/digital/documentation/predix-platforms/vault-service-overview.html`
`https://www.ge.com/digital/documentation/predix-platforms/pas-overview.html`
`https://www.ge.com/digital/documentation/predix-platforms/tms-get-started.html`
`https://www.ge.com/digital/documentation/predix-platforms/uaas-overview.html`	

https://www.ge.com/digital/documentation/edge-software/c_predix_edge_architecture_overview.html https://www.ge.com/digital/documentation/edge-software/c_predix_edge_os_architecture.html https://www.ge.com/digital/documentation/edge-software/c_predix_edge_agent_architecture_intro.html https://www.ge.com/digital/documentation/edge-software/c_predix_edge_manager_architecture_intro.html https://www.ge.com/digital/documentation/edge-software/c_predix_edge_technician_console_architecture.html https://www.ge.com/digital/documentation/edge-software/c_predix_edge_application_components_architecture.html https://www.ge.com/digital/documentation/edge-software/c_Predix_Edge_Protocol_Adapters_Overview.html https://www.ge.com/digital/documentation/edge-software/c_edge_to_cloud_data_flow.html https://www.ge.com/digital/documentation/edge-software/c_about_predix_cloud_connect.html https://www.ge.com/digital/documentation/edge-software/c_About_Enterprise_Connect.html

Tabelle A.5.: Auswertung General Electrics – Predix.

Plattform	**Google Cloud IoT Core**
Edgegeräte	
Wie wird Vertraulichkeit der Übertragung von der Edge zur Plattform sichergestellt ?	MQTT oder HTTPS über TLS
Wie wird Authentizität der Übertragung / des Thing von der Edge zur Plattform sichergestellt ?	Asymetrische Schlüssel in TLS und Zertifikate
Wie wird Integrität der Übertragung von der Edge zur Plattform sichergestellt ?	MAC inTLS
Wie wird Vertraulichkeit von Daten auf Edgegeräten sichergestellt ?	N/A

Wie werden Updates durchgeführt ?	Gerätemanager erlaubt grobe konfiguration der Geräte.
Gibt es ein Flottenmanagement (für On- und Offboarding / monitoring)	Gerätemanager erlaubt grobe konfiguration der Geräte.
Plattform	
Wie wird Vertraulichkeit der Übertragung innerhalb der Plattform(-komponenten) sichergestellt ?	HTTPS , MQTT , REST über TLS
Wie wird Authentizität der Übertragung / der Komponenten innerhalb der Plattform sichergestellt ?	JSON Web Tokens zur Authentifizierung
Wie wird Integrität der Übertragunginnerhalb der Plattform(-komponenten)sichergestellt ?	MAC in TLS
Wie wird Vertraulichkeit von Daten innerhalb der Plattform sichergestellt ?	Daten werden in Google Pub/Sub abgelegt
Wie wird die Nachvollziehbarkeit von Aktionen in der Plattform sichergestellt ?	Zentrales Logging über Google Cloud Monitoring und Logging
Gibt es ein Automatisches Patchmanagement ?	N/A
Wie wird Verfügbarkeit der Plattform sichergestellt ?	Skalierung durch Google Cloud
Verfügt die Plattform über Features zur automatischen Klassifikation von Daten ?	N/A
Besonderheiten	N/A
Nutzerverwaltung	
Wie authentifizieren sich Nutzer im Plattform Ökosystem ?	Anmeldung über Google Coud IAM
Wie werden Berechtigungen im Plattform Ökosystem vergeben ?	RBAC basierend auf Google Cloud IAM
Verbindung mit externen Anwendungen	
Wie wird Vertraulichkeit der Übertragung von der Plattform zu externen Anwendungen sichergestellt ?	N/A
Wie wird Authentizität der Übertragung / der Plattform zu externen Anwendungen sichergestellt ?	N/A
Wie wird Integrität der Übertragung von der Plattform zu externen Anwendungen sichergestellt ?	N/A

Quellen
https://cloud.google.com/iot-core https://cloud.google.com/iot/docs/concepts/overview https://cloud.google.com/iot/docs/concepts/ device-security https://cloud.google.com/iot/docs/how-tos/iam https://cloud.google.com/iot/docs/how-tos/ logging-monitoring

Tabelle A.6.: Auswertung Google Cloud IoT Core.

Plattform	**IBM - Watson IoT Suite**
Edgegeräte	
Wie wird Vertraulichkeit der Übertragung von der Edge zur Plattform sichergestellt ?	Kommunikation über MQTT mit TLS. Alternativ auch HTTP Messaging API
Wie wird Authentizität der Übertragung / des Thing von der Edge zur Plattform sichergestellt ?	Jedes Gerät hat einen ünique device identifier". Ab dann Token based. Optional Zertifikate.
Wie wird Integrität der Übertragung von der Edge zur Plattform sichergestellt ?	MAC in TLS
Wie wird Vertraulichkeit von Daten auf Edgegeräten sichergestellt ?	N/A
Wie werden Updates durchgeführt ?	Updates über Device Management Protocol(basiert auf MQTT).
Gibt es ein Flottenmanagement (für On- und Offboarding / monitoring)	Device Management durch Device Management Agents auf geräten umgesetzt. Device Management Agent erlaubt konfiguration und zusätzliches Monitoring, sowie Firmwareupdtes über das eigene Device Management Protocol.
Plattform	
Wie wird Vertraulichkeit der Übertragung innerhalb der Plattform(-komponenten) sichergestellt ?	Liegt in der Verantwortung der jeweiligen Komponente. Meist durch TLS abgesichert.
Wie wird Authentizität der Übertragung / der Komponenten innerhalb der Plattform sichergestellt ?	Liegt in der Verantwortung der jeweiligen Komponente. Meist wird IBMid als mechanismus verwendet.

Wie wird Integrität der Übertragunginnerhalb der Plattform(-komponenten) sichergestellt ?	MAC in TLS
Wie wird Vertraulichkeit von Daten innerhalb der Plattform sichergestellt ?	Liegt in der Verantwortung des jeweiligen Datastore
Wie wird die Nachvollziehbarkeit von Aktionen in der Plattform sichergestellt ?	N/A
Gibt es ein Automatisches Patchmanagement ?	N/A
Wie wird Verfügbarkeit der Plattform sichergestellt ?	Last event cache Speichert je letzte Nachricht der Geräte. Automatische Backups durch IBM
Verfügt die Plattform über Features zur automatischen Klassifikation von Daten ?	N/A
Besonderheiten	Zusätzliche Absicherung durch Security Policies z.B. Black-/Whitelisting. Viele Sicherheitsanforderungen zusätzlich durch Watson IBM Cloud abgedeckt.
Nutzerverwaltung	
Wie authentifizieren sich Nutzer im Plattform Ökosystem ?	OAuth oder Facebook und Google+ für Webapps
Wie werden Berechtigungen im Plattform Ökosystem vergeben ?	RBAC, alternativ resource-level access control um Geräte zugriff zu verwalten.
Verbindung mit externen Anwendungen	
Wie wird Vertraulichkeit der Übertragung von der Plattform zu externen Anwendungen sichergestellt ?	N/A
Wie wird Authentizität der Übertragung / der Plattform zu externen Anwendungen sichergestellt ?	N/A
Wie wird Integrität der Übertragung von der Plattform zu externen Anwendungen sichergestellt ?	N/A
Quellen	
`https://www.ibm.com/docs/en/watson-iot-platform?topic=features-product-architecture`	

Tabelle A.7.: Auswertung IBM - Watson IoT Suite.

Plattform	Microsoft - Azure IoT Suite
Edgegeräte	
Wie wird Vertraulichkeit der Übertragung von der Edge zur Plattform sichergestellt ?	Unterstüzung für MQTT, MQTT over WebSockets, AMQP AMQP over WebSockets und HTTPS über TLS bzw DTLS für UDP basierte Protokolle
Wie wird Authentizität der Übertragung / des Thing von der Edge zur Plattform sichergestellt ?	SAS token-based authentication oder X.509 certificate authentication.
Wie wird Integrität der Übertragung von der Edge zur Plattform sichergestellt ?	MAC inTLS
Wie wird Vertraulichkeit von Daten auf Edgegeräten sichergestellt ?	Verschlüsseln der Daten auf dem Gerät
Wie werden Updates durchgeführt ?	Bulk / Batch management möglichkeiten und OTA Updates. Updates Signiert.
Gibt es ein Flottenmanagement (für On- und Offboarding / monitoring)	IoT Hub verfügt über umfangreiche Methoden um Updates zu verwalten, Geräte zu überwachen und zum On- und Offboarding.
Plattform	
Wie wird Vertraulichkeit der Übertragung innerhalb der Plattform(-komponenten) sichergestellt ?	Sichern der Kommunikation über TLS und DTLS. Trennung der Umgebungen in Virtuelle Netzte möglich.
Wie wird Authentizität der Übertragung / der Komponenten innerhalb der Plattform sichergestellt ?	Security Tokens
Wie wird Integrität der Übertragunginnerhalb der Plattform(-komponenten) sichergestellt ?	MAC inTLS
Wie wird Vertraulichkeit von Daten innerhalb der Plattform sichergestellt ?	Daten werden Verschlüsselt von IoT Hub abgelegt, sonst verantwortung des Datastore.
Wie wird die Nachvollziehbarkeit von Aktionen in der Plattform sichergestellt ?	Logging und Threat Detection durch Azure Infrastruktur
Gibt es ein Automatisches Patchmanagement ?	Updates Schedulbar
Wie wird Verfügbarkeit der Plattform sichergestellt ?	Backups durch Azure Infrastruktur und caching von Nachrichten. Skalierung durch Azure gegen DDoS

Verfügt die Plattform über Features zur automatischen Klassifikation von Daten ?	N/A
Besonderheiten	Azure RTOS und Azure Sphere als Geräte OS. Starke Unterstützung für TPM (Trusted Platform Module)
Nutzerverwaltung	
Wie authentifizieren sich Nutzer im Plattform Ökosystem ?	Azure Active Directory oder Shared access signatures oder Per-device security credentials. Bei Azure AD werden OAuth 2.0 accestokens verwendet. MFA für Nutzeraccounts.
Wie werden Berechtigungen im Plattform Ökosystem vergeben ?	Azure AD erlaubt RBAC
Verbindung mit externen Anwendungen	
Wie wird Vertraulichkeit der Übertragung von der Plattform zu externen Anwendungen sichergestellt ?	N/A
Wie wird Authentizität der Übertragung / der Plattform zu externen Anwendungen sichergestellt ?	N/A
Wie wird Integrität der Übertragung von der Plattform zu externen Anwendungen sichergestellt ?	N/A
Quellen	

`https://docs.microsoft.com/de-de/azure/architecture/reference-architectures/iot`
`https://docs.microsoft.com/en-us/azure/iot-hub/`
`https://docs.microsoft.com/en-gb/azure/iot-edge/about-iot-edge?view=iotedge-2020-11`	

Tabelle A.8.: Auswertung Microsoft - Azure IoT Suite.

Plattform	**Oracle – Oracle Cloud IoT**
Edgegeräte	
Wie wird Vertraulichkeit der Übertragung von der Edge zur Plattform sichergestellt ?	MQTT und HTTPS über TLS
Wie wird Authentizität der Übertragung / des Thing von der Edge zur Plattform sichergestellt ?	Tokenbasiert mit Zertifikaten und einzigartiger ID beim Onboarding

Wie wird Integrität der Übertragung von der Edge zur Plattform sichergestellt ?	MAC in TLS
Wie wird Vertraulichkeit von Daten auf Edgegeräten sichergestellt ?	N/A
Wie werden Updates durchgeführt ?	N/A
Gibt es ein Flottenmanagement (für On- und Offboarding / monitoring)	Oracle IoT Asset Monitoring Cloud Service erlaubt monitoring. Keine angaben zu Patches oder On-/Offboarding.
Plattform	
Wie wird Vertraulichkeit der Übertragung innerhalb der Plattform(-komponenten) sichergestellt ?	MQTT, HTTPs und REST über TLS
Wie wird Authentizität der Übertragung / der Komponenten innerhalb der Plattform sichergestellt ?	Tokenbasiert oder HTTP basic
Wie wird Integrität der Übertragunginnerhalb der Plattform(-komponenten) sichergestellt ?	MAC in TLS
Wie wird Vertraulichkeit von Daten innerhalb der Plattform sichergestellt ?	N/A
Wie wird die Nachvollziehbarkeit von Aktionen in der Plattform sichergestellt ?	N/A
Gibt es ein Automatisches Patchmanagement ?	N/A
Wie wird Verfügbarkeit der Plattform sichergestellt ?	N/A
Verfügt die Plattform über Features zur automatischen Klassifikation von Daten ?	N/A
Besonderheiten	
Nutzerverwaltung	
Wie authentifizieren sich Nutzer im Plattform Ökosystem ?	HTTP Basic Authentication oder OAuth
Wie werden Berechtigungen im Plattform Ökosystem vergeben ?	RBAC
Verbindung mit externen Anwendungen	
Wie wird Vertraulichkeit der Übertragung von der Plattform zu externen Anwendungen sichergestellt ?	N/A

Wie wird Authentizität der Übertragung / der Plattform zu externen Anwendungen sichergestellt ?	N/A
Wie wird Integrität der Übertragung von der Plattform zu externen Anwendungen sichergestellt ?	N/A
Quellen	

`https://docs.oracle.com/en/solutions/internet-of-things-options-connect-devices/index.html`
`https://docs.oracle.com/en/solutions/internet-of-things-options-connect-devices/learn-oracle-internet-things-itcon.html`
`https://docs.oracle.com/en/cloud/paas/iot-cloud/iotrq/QuickStart.html`
`https://docs.oracle.com/en/cloud/paas/iot-cloud/iotsu/understand-oracle-internet-things-cloud-service-user-roles.html`
`https://docs.oracle.com/en/cloud/paas/iot-cloud/iotgs/connectors.html`	

Tabelle A.9.: Auswertung Oracle – Oracle Cloud IoT.

Plattform	**PTC - Thing Worx**
Edgegeräte	
Wie wird Vertraulichkeit der Übertragung von der Edge zur Plattform sichergestellt ?	Verschlüsseln mittels OpenSSL/TLS
Wie wird Authentizität der Übertragung / des Thing von der Edge zur Plattform sichergestellt ?	Server authentifiziert sich am Gerät über Zertifikat. Optional : Client authentifiziert sich am Server über zertifikat
Wie wird Integrität der Übertragung von der Edge zur Plattform sichergestellt ?	MAC in TLS
Wie wird Vertraulichkeit von Daten auf Edgegeräten sichergestellt ?	N/A
Wie werden Updates durchgeführt ?	N/A
Gibt es ein Flottenmanagement (für On- und Offboarding / monitoring)	N/A
Plattform	

Wie wird Vertraulichkeit der Übertragung innerhalb der Plattform(-komponenten) sichergestellt ?	Verschlüsseln mittels OpenSSL/TLS. HTTP Strict-Transport-Security (HSTS) in Tomcat
Wie wird Authentizität der Übertragung / der Komponenten innerhalb der Plattform sichergestellt ?	Application keys als security tokens während der Ausführung
Wie wird Integrität der Übertragunginnerhalb der Plattform(-komponenten) sichergestellt ?	MAC inTLS
Wie wird Vertraulichkeit von Daten innerhalb der Plattform sichergestellt ?	Daten werden verschlüsselt gelagert. Datenbanken, keystores und Konfigurationen verschlüsselt.
Wie wird die Nachvollziehbarkeit von Aktionen in der Plattform sichergestellt ?	Zentrales Logging von Authentifizierungen
Gibt es ein Automatisches Patchmanagement ?	N/A
Wie wird Verfügbarkeit der Plattform sichergestellt ?	High Availability (HA) environment verfügbar mit zusätzlichen Hardware (server, loadbalancer etc...) und Software (Synchronisations Services) Lösungen
Verfügt die Plattform über Features zur automatischen Klassifikation von Daten ?	N/A
Besonderheiten	Starker Fokus auf hohe Verfügbarkeit
Nutzerverwaltung	
Wie authentifizieren sich Nutzer im Plattform Ökosystem ?	basic user-password, single sign-on (SSO) access, certificate-based data decryption, SAML
Wie werden Berechtigungen im Plattform Ökosystem vergeben ?	Rechte werden durch Gruppen, User und Organisationen verteilt.
Verbindung mit externen Anwendungen	
Wie wird Vertraulichkeit der Übertragung von der Plattform zu externen Anwendungen sichergestellt ?	N/A
Wie wird Authentizität der Übertragung / der Plattform zu externen Anwendungen sichergestellt ?	N/A
Wie wird Integrität der Übertragung von der Plattform zu externen Anwendungen sichergestellt ?	N/A

Quellen
https://www.ptc.com/en/resources/iiot/brochure/thingworx-overview https://support.ptc.com/help/thingworx_hc/thingworx_8_hc/en/#page/ThingWorx/Help/Composer/Security/Security.html# https://support.ptc.com/help/edge_sdk_c/r2.2.2/en/#page/c_sdk%2Fc_security_overview.html%23

Tabelle A.10.: Auswertung PTC - Thing Worx.

Plattform	**SAP - Leonardo**
Edgegeräte	
Wie wird Vertraulichkeit der Übertragung von der Edge zur Plattform sichergestellt ?	https, MQTT, SNMP, Modbus, CoAP, OPC UA Verschlüsselt durch TLS
Wie wird Authentizität der Übertragung / des Thing von der Edge zur Plattform sichergestellt ?	Protokollabhänig bei verbindung zu Gateways, Verbindungen zur Cloud durch X.509 Zertifikate, Basic Authentication für REST API
Wie wird Integrität der Übertragung von der Edge zur Plattform sichergestellt ?	MAC in TLS
Wie wird Vertraulichkeit von Daten auf Edgegeräten sichergestellt ?	Die Edge Plattform stellt Verschlüsselung für Daten zur Verfügung
Wie werden Updates durchgeführt ?	N/A
Gibt es ein Flottenmanagement (für On- und Offboarding / monitoring)	SAP Leonardo verfügt über eine Device Managemanet Komponente die sicheres Onboarding durch Zertifikats Authentifizierung und Offboarding durch eine API
Plattform	
Wie wird Vertraulichkeit der Übertragung innerhalb der Plattform(-komponenten) sichergestellt ?	Alle unterstützten Verbindungen durch TLS verschlüsselt
Wie wird Authentizität der Übertragung / der Komponenten innerhalb der Plattform sichergestellt ?	Basic Authentication für REST API innerhalb der Plattform
Wie wird Integrität der Übertragung innerhalb der Plattform(-komponenten) sichergestellt ?	MAC in TLS

Wie wird Vertraulichkeit von Daten innerhalb der Plattform sichergestellt ?	Verschlüsselte Speicherung in der SAP Cloud
Wie wird die Nachvollziehbarkeit von Aktionen in der Plattform sichergestellt ?	Automatisches Logging in der Plattform. Je nach Annotation werden auch lesende Zugriffe auf sensible Daten protokolliert. Logs werden nach 201 Tagen automatisch gelöscht
Gibt es ein Automatisches Patchmanagement ?	N/A
Wie wird Verfügbarkeit der Plattform sichergestellt ?	N/A
Verfügt die Plattform über Features zur automatischen Klassifikation von Daten ?	Datenquellen können als annotations versehen werden um sie zu klassifizieren
Besonderheiten	Einfaches Sperren und Löschen von personenbezogenen Daten
Nutzerverwaltung	
Wie authentifizieren sich Nutzer im Plattform Ökosystem ?	Abgleich mit Identityprovider. Dabei werden unterschiedliche Anbieter genutzt um Single Sign-On (SSO) zu ermöglichen. Dabei kommen SAML 2.0, OAuth 2.0, Basic Authentication und weitere Protokolle zum Einsatz.
Wie werden Berechtigungen im Plattform Ökosystem vergeben ?	Zweidimensionales RBAC mit objektinstanzbasierten- und funktionalen Berechtigungen. Alle Nutzer und Systemkomponenten erhalten durch das komplette Ökosystem eindeutige Identitäten.
Verbindung mit externen Anwendungen	
Wie wird Vertraulichkeit der Übertragung von der Plattform zu externen Anwendungen sichergestellt ?	N/A
Wie wird Authentizität der Übertragung / der Plattform zu externen Anwendungen sichergestellt ?	N/A
Wie wird Integrität der Übertragung von der Plattform zu externen Anwendungen sichergestellt ?	N/A
Quellen	
`https://help.sap.com/viewer/product/SAP_Leonardo_IoT/1904a/en-US`	

https://help.sap.com/viewer/7f425dfcbb474a28b9d07829f524665c/1904a/en-US/934fd0db9d9d4db2b9975f8594c766a1.html https://help.sap.com/viewer/e7dae2e1ffa44f70a2959d69f75686d5/1904a/en-US/4dcdc6add96749a382221dc9c2c0f239.html https://help.sap.com/viewer/1dd02d18a8674e89ac7dfde19ebb6c66/2108/en-US https://help.sap.com/viewer/91d9184adbe941e68aafb8724005a479/Cloud/en-US

Tabelle A.11.: Auswertung SAP - Leonardo.

Plattform	**Siemens - Mindsphere**
Edgegeräte	
Wie wird Vertraulichkeit der Übertragung von der Edge zur Plattform sichergestellt ?	OPC UA, SIMATIC S7, SIMATICS, Modbus TCP, EtherNet/IP und andere zur Verbindung zum Gateway. Verbindung vom Gateway zur Cloud durch MQTT und REST gesichert durch TLS.
Wie wird Authentizität der Übertragung / des Thing von der Edge zur Plattform sichergestellt ?	Tokenbasiert durch JSON Web Token
Wie wird Integrität der Übertragung von der Edge zur Plattform sichergestellt ?	MAC in TLS
Wie wird Vertraulichkeit von Daten auf Edgegeräten sichergestellt ?	Verschlüsselte Konfigurationsdateien. Verschlüsselung von gespeicherten Daten auf der Gateway Komponente.
Wie werden Updates durchgeführt ?	Mindsphere sucht automatisch nach Firmware updates, die dann durch die Plattform istalliert werden können
Gibt es ein Flottenmanagement (für On- und Offboarding / monitoring)	Mindsphere unterstützt On- und Offboarding. Insbesondere Onboarding wird durch einzigartige Geräte ID's und Sicherheitstokens abgesichert.
Plattform	
Wie wird Vertraulichkeit der Übertragung innerhalb der Plattform(-komponenten) sichergestellt ?	Alle unterstützten Verbindungen durch TLS verschlüsselt
Wie wird Authentizität der Übertragung / der Komponenten innerhalb der Plattform sichergestellt ?	JSON Web Tokens zur Authentifizierung

Wie wird Integrität der Übertragunginnerhalb der Plattform(-komponenten) sichergestellt ?	MAC in TLS
Wie wird Vertraulichkeit von Daten innerhalb der Plattform sichergestellt ?	Daten werden verschlüsselt in der Plattform gespeichert
Wie wird die Nachvollziehbarkeit von Aktionen in der Plattform sichergestellt ?	Alle Loggs, einschließlich der Geräte, werden zentral gesammelt.
Gibt es ein Automatisches Patchmanagement ?	N/A
Wie wird Verfügbarkeit der Plattform sichergestellt ?	Die Daten werden täglich gesichert und 30 Tage lang aufbewahrt. Dienste zum Schutz von Datenspeichern werden redundant in mindestens zwei Verfügbarkeitszonen ausgeführt
Verfügt die Plattform über Features zur automatischen Klassifikation von Daten ?	Mindsphere verfügt über möglichkeiten zur Klassifikation von Daten. Dabei gibt es Angebote durch Siemens um bei der Klassifikation zu helfen.
Besonderheiten	
Nutzerverwaltung	
Wie authentifizieren sich Nutzer im Plattform Ökosystem ?	JSON Web Tokens zur Authentifizierung. Mindsphere verfügt über integriete IAM-Service mit OAuth 2.0 authorization. Accounts können zusätzlich über MFA gesichert werden.
Wie werden Berechtigungen im Plattform Ökosystem vergeben ?	RBAC
Verbindung mit externen Anwendungen	
Wie wird Vertraulichkeit der Übertragung von der Plattform zu externen Anwendungen sichergestellt ?	MQTT über TLS
Wie wird Authentizität der Übertragung / der Plattform zu externen Anwendungen sichergestellt ?	N/A
Wie wird Integrität der Übertragung von der Plattform zu externen Anwendungen sichergestellt ?	MAC in TLS
Quellen	
`https://siemens.mindsphere.io/en/industrial-iot/industrial-edge`	

https://new.siemens.com/global/en/products/ automation/topic-areas/industrial-edge/ it-specialists.html https://siemens.mindsphere.io/en/industrial-iot/ mindsphere https://siemens.mindsphere.io/en/about/ for-developers https://assets.new.siemens.com/siemens/assets/api/ uuid:6b876b5e-5594-4da4-90e0-e9e0c6f1f1e1/version: 1557483304/siemens-plm-mindsphere-security-model-wp-75966-a7. pdf# https://developer.mindsphere.io/concepts/index.html

Tabelle A.12.: Auswertung Siemens - Mindsphere.

A.3. Vollständiger Code des IoTComponentCheck

```
1
2  package carisma.check.iotcomponents;
3
4  import java.util.Collection;
5  import java.util.HashMap;
6  import java.util.LinkedList;
7  import java.util.List;
8  import java.util.Map;
9  import java.util.stream.Collectors;
10
11 import org.eclipse.emf.common.util.EList;
12 import org.eclipse.emf.ecore.resource.Resource;
13 import org.eclipse.uml2.uml.Association;
14 import org.eclipse.uml2.uml.Class;
15 import org.eclipse.uml2.uml.Classifier;
16 import org.eclipse.uml2.uml.Element;
17 import org.eclipse.uml2.uml.Interface;
18 import org.eclipse.uml2.uml.InterfaceRealization;
19 import org.eclipse.uml2.uml.NamedElement;
20 import org.eclipse.uml2.uml.Package;
21 import org.eclipse.uml2.uml.Profile;
22 import org.eclipse.uml2.uml.Stereotype;
23
24 import carisma.core.analysis.AnalysisHost;
25 import carisma.core.analysis.result.AnalysisResultMessage;
26 import carisma.core.analysis.result.StatusType;
27 import carisma.core.checks.CheckParameter;
28 import carisma.core.checks.CarismaCheckWithID;
29
30 import carisma.profile.umlsec.iotcomponents.UMLsec;
31 import carisma.profile.umlsec.iotcomponents.UMLsecUtil;
32
33
```

```
34
35 /** Contains a Simple CARiSMA Check which returns all elements of a given
       Model.
36  *
37  */
38
39 public class IoTComponentsCheck implements CarismaCheckWithID {
40
41   // Check IDs
42   public static final String CHECK_ID = "carisma.check.iotcomponents"; //$NON
       -NLS-1$
43   public static final String PARAM_CONFIGURATION = "carisma.check.
       iotcomponents.configuration"; //$NON-NLS-1$
44   public static final String CHECK_NAME = "IoT Components"; //$NON-NLS-1$
45
46   public static final UMLsec[] GOALS = {  UMLsec.CONFIDENTIALITY_IOTCOMP,
47                         UMLsec.INTEGRITY_IOTCOMP,
48                         UMLsec.AVAILABILITY_IOTCOMP,
49                         UMLsec.TRANSPARENCY_IOTCOMP,
50                         UMLsec.UNLINKABILITY_IOTCOMP,
51                         UMLsec.INTERVENABILITY_IOTCOMP};
52
53   AnalysisHost host;
54   int numOfElements = 0;
55
56   @Override
57   public boolean perform(Map<String, CheckParameter> parameters, AnalysisHost
        host) {
58     this.host = host;
59     this.numOfElements = 0;
60     Resource currentModel = host.getAnalyzedModel();
61     if (currentModel.getContents().isEmpty()) {
62       host.addResultMessage(new AnalysisResultMessage(StatusType.WARNING, "
       Empty model"));
63       return false;
64     }
65
66     if (currentModel.getContents().get(0) instanceof Package) {
67       Package model = (Package) currentModel.getContents().get(0);
68       printContent(model, "");
69
70       // Check if Goals are applied to parts of the Model
71       if (!checkIsApplicable(model, false)) {
72         host.addResultMessage(new AnalysisResultMessage(StatusType.WARNING, "
       No verifiable Goals"));
73         return false;// No goal's => Check not applicable => Fail
74       }
75       host.appendLineToReport("
     ------------------------------------------------------------------------------
     ");
76       host.appendLineToReport("Goals detected");
77       host.appendLineToReport("
     ------------------------------------------------------------------------------\
     n");
```

```
78
79
80      Map<UMLsec, List<Stereotype>> mapping = generateFulfillmentMapping(
    model);
81      //Enrich model
82      enrichModel(model, mapping);
83      host.appendLineToReport("
    ------------------------------------------------------------------------------
    ");
84      host.appendLineToReport("Model after adding stereotypes based on names
    :");
85      printContent(model, "");
86
87      //Check which Goals are fulfilled
88
89      boolean returnBool = true;
90      HashMap<UMLsec, HashMap<Element, Boolean>> fullfilment = new HashMap
    <>();
91      for (UMLsec umLsec : GOALS) {
92        HashMap<Element, Boolean> goalFulfillment = checkGoalFulfillment(
    model, umLsec, mapping);
93        fullfilment.put(umLsec, goalFulfillment);
94
95        //Generate Message
96        if(goalFulfillment.isEmpty()) {
97          host.addResultMessage(new AnalysisResultMessage(StatusType.INFO,
    umLsec.toString() + " not applied to model"));
98        }else if(! goalFulfillment.keySet().stream().map(x -> goalFulfillment
    .get(x)).toList().contains(false)) {
99          host.addResultMessage(new AnalysisResultMessage(StatusType.INFO,
    umLsec.toString() + " fulfilled" ));
100       }else {
101         if(umLsec.equals(UMLsec.INTERVENABILITY_IOTCOMP)) {
102           host.addResultMessage(new AnalysisResultMessage(StatusType.
    WARNING, umLsec.toString() + " not fulfilled but the Profile doesn't
    provied Stereotypes to fulfill the  it"));
103         }else {
104           host.addResultMessage(new AnalysisResultMessage(StatusType.ERROR,
     umLsec.toString() + " not fulfilled" ));
105           returnBool = false;//unfulfilled always contains
    INTERVENABILITY_IOTCOMP if its in the model => set returnBool to false
106         }
107       }
108     }
109
110     // Make recomendations
111     HashMap<UMLsec, List<Element>> unfulfilled = spotUnfulfilledGoals(
    fullfilment);
112
113     if(unfulfilled.isEmpty()) {
114       return true;//No improvements needed => check not successful
115     }
116
117     recommendImprovements(model, unfulfilled, mapping);
```

```
118
119
120
121       return returnBool;//Improvements needed => check not successful
122     }
123
124     host.addResultMessage(new AnalysisResultMessage(StatusType.WARNING, "
      Content is not a model!"));
125     return false;
126   }
127
128   public void printContent(Element element, String indent) {
129     numOfElements++;
130     host.appendToReport(indent+element.eClass().getName()+": ");
131     if (!element.getAppliedStereotypes().isEmpty()) {
132       host.appendToReport("<<");
133       for (Stereotype st : element.getAppliedStereotypes()) {
134         host.appendToReport(st.getName()+",");
135       }
136       host.appendToReport(">> ");
137     }
138     if (element instanceof NamedElement) {
139       NamedElement namedElement = (NamedElement)element;
140       host.appendToReport(namedElement.getName());
141     }
142     host.appendLineToReport("");
143     for (Element child : element.allOwnedElements()) {
144       printContent(child, indent+"  ");
145     }
146   }
147
148   @Override
149   public String getCheckID() {
150     return CHECK_ID;
151   }
152
153   @Override
154   public String getName() {
155     return CHECK_NAME;
156   }
157
158
159   /**
160    * Checks if at least one Element in the Model has one of the goal
       stereotypes applied
161    * @param element - Model
162    * @param value
163    * @return
164    */
165   private boolean checkIsApplicable(Element element, boolean value) {
166     EList<Stereotype> appliedStereotypes = element.getAppliedStereotypes();
167     for (Stereotype stereotype : appliedStereotypes) {
168       for (UMLsec umLsec : GOALS) {
169         if(umLsec.isEqual(stereotype)) {
```

```
170             return true;
171           }
172         }
173       }
174       for (Element child : element.allOwnedElements()) {
175         value = value || checkIsApplicable(child, value);
176       }
177       return value;
178     }
179
180     /**
181      * If the name of a class in the model contains the the name of a
          stereotype
182      * the stereotype with the same name is applied to the class for the
183      * duration of the check
184      * @param model
185      * @param mapping
186      */
187     private void enrichModel(Package model, Map<UMLsec, List<Stereotype>>
          mapping) {
188       //Generate list of all classes in the model
189       List<Element> classes = model.getOwnedElements().stream().filter(x -> x
          instanceof Class || x instanceof Interface).toList();
190       //Generate list of all stereotypes in the fulfillment mapping
191       List<Stereotype> stereotypes=mapping.values().stream().flatMap(Collection
          ::stream).collect(Collectors.toList()).stream().distinct().toList();
192       for (Element element : classes) {
193         List<Stereotype> applyStereotypes = new LinkedList<Stereotype>();
194         String className = ((NamedElement) element).getName();
195
196         for (Stereotype stereotype : stereotypes) {
197           if(className.equals(stereotype.getName())|| className.toLowerCase().
          contains(stereotype.getName().toLowerCase())) {
198             applyStereotypes.add(stereotype);
199           }
200         }
201
202         if(!applyStereotypes.isEmpty()) {
203           host.appendLineToReport("Applying Stereotypes " + applyStereotypes.
          stream().map(x -> x.getName()).toList().toString() +
204                   " to " + element.eClass().getName() + " because of the Name " +
          ((NamedElement) element).getName());
205           applyStereotypes.forEach(x -> element.applyStereotype(x));
206         }
207       }
208     }
209
210     /**
211      * Loads all stereotypes from the applied profile and checks for each GOAL
212      * which stereotypes fulfill the goal
213      *
214      * @param model
215      * @return Map with goals as key and a list of all stereotypes which
          fulfill the goal as value
```

```
216   */
217  private Map<UMLsec , List<Stereotype>> generateFulfillmentMapping(Package
       model){
218    Map<UMLsec, List<Stereotype>> mapping = new HashMap<UMLsec, List<
       Stereotype>>();
219    Profile profile = model.getAllAppliedProfiles().stream().filter(x ->
       UMLsec.DESCRIPTOR.getProfileName().equals(x.getName())).toList().get(0);
220    List<Stereotype> allStereotypes = profile.allApplicableStereotypes();
221    for (UMLsec umLsec : GOALS) {
222      List<Stereotype> stereotypes = new LinkedList<Stereotype>();
223      for (Stereotype element : allStereotypes) {
224        if ((!umLsec.isEqual(element) )&& stereotypeFulfillsGoal(element,
       umLsec)) {
225          stereotypes.add(element);
226        }
227      }
228      mapping.put(umLsec, stereotypes);
229    }
230    return mapping;
231  }
232
233  /**
234   *
235   * @param model - the model
236   * @param goal - a goal stereotype to check fulfillment for
237   * @param mapping - a mapping with the goal stereotypes as keys and
238   * a list containing all stereotypes which fulfill the goal stereotypes as
       vale
239   * @return A map with the key being a class which applies the goal
       stereotype and
240   * the value as boolean if the goal is fulfilled
241   */
242  private HashMap<Element, Boolean> checkGoalFulfillment(Package model,
       UMLsec goal, Map<UMLsec , List<Stereotype>> mapping){
243    host.appendLineToReport("
       ----------------------------------------------------------------------------------
       ");
244    host.appendLineToReport("Checking for " + goal.toString() + " fulfillment
       ");
245    HashMap<Element, Boolean> result = new HashMap<Element, Boolean>();
246    List<Element> annotatedElements = UMLsecUtil.getStereotypedElements(model
       , goal);
247    for (Element element : annotatedElements) {
248      boolean stereotypeFulfillment=false;
249      if (element instanceof Class) {
250        stereotypeFulfillment = element.getAppliedStereotypes().stream().map(
       x -> mapping.get(goal).contains(x)).toList().contains(true);//Fulfills
       Goal itself
251
252        if(!stereotypeFulfillment) {// check associated classes
253          for (Association association : ((Class) element).getAssociations())
       {
254            List<Element> members = association.getRelatedElements();
255            for (Element member : members) {
```

```
256             EList<Stereotype> s = member.getAppliedStereotypes();
257             stereotypeFulfillment = stereotypeFulfillment || s.stream().map
    (x -> mapping.get(goal).contains(x)).toList().contains(true);
258           }
259         }
260
261       }
262
263     }else if(element instanceof Interface) {
264       stereotypeFulfillment = element.getAppliedStereotypes().stream().map(
    x -> mapping.get(goal).contains(x)).toList().contains(true);//Fulfills
    Goal itself
265       if(!stereotypeFulfillment) {
266
267         List<Element> allRealizationInModel = ((Interface) element).
    getPackage().allOwnedElements().stream().filter(x -> x instanceof
    InterfaceRealization ).toList();
268         List<EList<NamedElement>> members = allRealizationInModel.stream().
    filter(x -> ((InterfaceRealization) x).getContract().equals(element)).map
    (x -> ((InterfaceRealization) x).getClients()).toList();
269         List<NamedElement> distinctMembers = members.stream().flatMap(
    Collection::stream).collect(Collectors.toList()).stream().distinct().
    toList();
270         for (Element member : distinctMembers) {
271           EList<Stereotype> s = member.getAppliedStereotypes();
272           stereotypeFulfillment = stereotypeFulfillment || s.stream().map(x
     -> mapping.get(goal).contains(x)).toList().contains(true);
273         }
274       }
275
276     }else if (element instanceof Association) {
277       List<EList<Stereotype>> memberStereotypes = ((Association) element).
    getRelatedElements().stream().map(x -> x.getAppliedStereotypes()).toList
    ();
278       EList<Stereotype> memberOne = memberStereotypes.get(0);
279       EList<Stereotype> memberTwo = memberStereotypes.get(1);
280       List<Stereotype> both = memberOne.stream().filter(x -> memberTwo.
    contains(x)).toList();
281       stereotypeFulfillment = both.stream().map(x -> mapping.get(goal).
    contains(x)).toList().contains(true);
282     }else {
283       host.appendLineToReport(((NamedElement) element).getName() + " is of
    type " + ((NamedElement) element).eClass().toString() + " and shouldn't
    be applicable");
284       continue;
285     }
286     result.put(element, stereotypeFulfillment);
287     host.appendLineToReport(element.eClass().getName() + " " + ((
    NamedElement)element).getName() + (stereotypeFulfillment ? " fulfills " :
     " doesn't fulfill ") + goal.toString());
288   }
289
290   if(result.isEmpty()) {
```

```
291       host.appendLineToReport("No Element tries to fulfill " + goal.toString
      ());
292     }
293
294       return result;
295     }
296
297
298     /**
299      * Checks if the stereotype stereo fulfills the UMLsec goal by
300      * checking if stereo is a generalization of goal
301      *
302      * @param stereo - a stereotype to check
303      * @param goal - a goal stereotype
304      * @return true if stereo is a generalization of, or equal to goal
305      */
306     private boolean stereotypeFulfillsGoal(Stereotype stereo, UMLsec goal) {
307       if (stereo.allParents().isEmpty()) {
308         if (goal.isEqual(stereo)) {
309           return true;
310         }else {
311           return false;
312         }
313       }else {
314         boolean returnval = false;
315         for ( Classifier parents : stereo.allParents()) {
316           returnval = returnval || stereotypeFulfillsGoal((Stereotype)parents,
      goal);
317         }
318         return returnval;
319       }
320     }
321
322
323     /**
324      * Reduces map of fulfillment to only the classes that don't fulfill
325      * the goals.
326      * @param fullfilment - map of goal stereotypes and
327      * a map of the classes applying them and if they are fulfilled
328      * @return map of goal stereotypes as key and a list of all classes which
       are annotated with
329      * this stereotype but don't have it fulfilled
330      */
331     private HashMap<UMLsec, List<Element>> spotUnfulfilledGoals(
332         HashMap<UMLsec, HashMap<Element, Boolean>> fullfilment) {
333       HashMap<UMLsec, List<Element>> unfulfilled = new HashMap<UMLsec, List<
      Element>>();
334       for (UMLsec umLsec : fullfilment.keySet()) {
335         List <Element> b = new LinkedList<Element>();
336         for (Element key : fullfilment.get(umLsec).keySet()) {
337           if(!fullfilment.get(umLsec).get(key)) {
338             b.add(key);
339           }
340         }
```

```
341       unfulfilled.put(umLsec, b);
342     }
343
344     return unfulfilled;
345   }
346
347   /**
348    * Lists for each class with unfulfilled goals all classes in the
349    * model which have at least one stereotype fulfilling the unfulfilled
350    * goals in the report
351    * @param model
352    * @param unfulfilled
353    * @param mapping
354    */
355   private void recommendImprovements(Package model, HashMap<UMLsec, List<
     Element>> unfulfilled,
356       Map<UMLsec, List<Stereotype>> mapping) {
357     List<Element> allClassesInModel = model.allOwnedElements().stream().
     filter(x -> x instanceof Class || x instanceof Interface).toList();
358     for (UMLsec umLsec : unfulfilled.keySet()) {
359       for (Element element : unfulfilled.get(umLsec)) {
360         StringBuilder sb =new StringBuilder();
361         sb.append(((NamedElement) element).getName() + " recommendations for
     " + umLsec.toString() + ":\n" );
362         for (Element c : allClassesInModel) {
363           List<Stereotype> solutions = c.getAppliedStereotypes().stream().
     filter(x -> mapping.get(umLsec).contains(x)).toList();
364           if (!solutions.isEmpty()) {
365             List<String> solutionNames = solutions.stream().map(x -> x.
     getName()).toList();
366             sb.append(  "Class " + ((NamedElement) c).getName() + " provides
     : " + solutionNames.toString() + "\n");
367           }
368         }
369         host.appendToReport(sb.toString());
370       }
371       host.appendLineToReport("
     ------------------------------------------------------------------------------\
     n");
372     }
373
374   }
375
376
377 }
```

Quellcode A.1: Der komplette Code der IoTComponentsCheck.java Klasse.

A.4. Abbildung des kompletten UML Profil

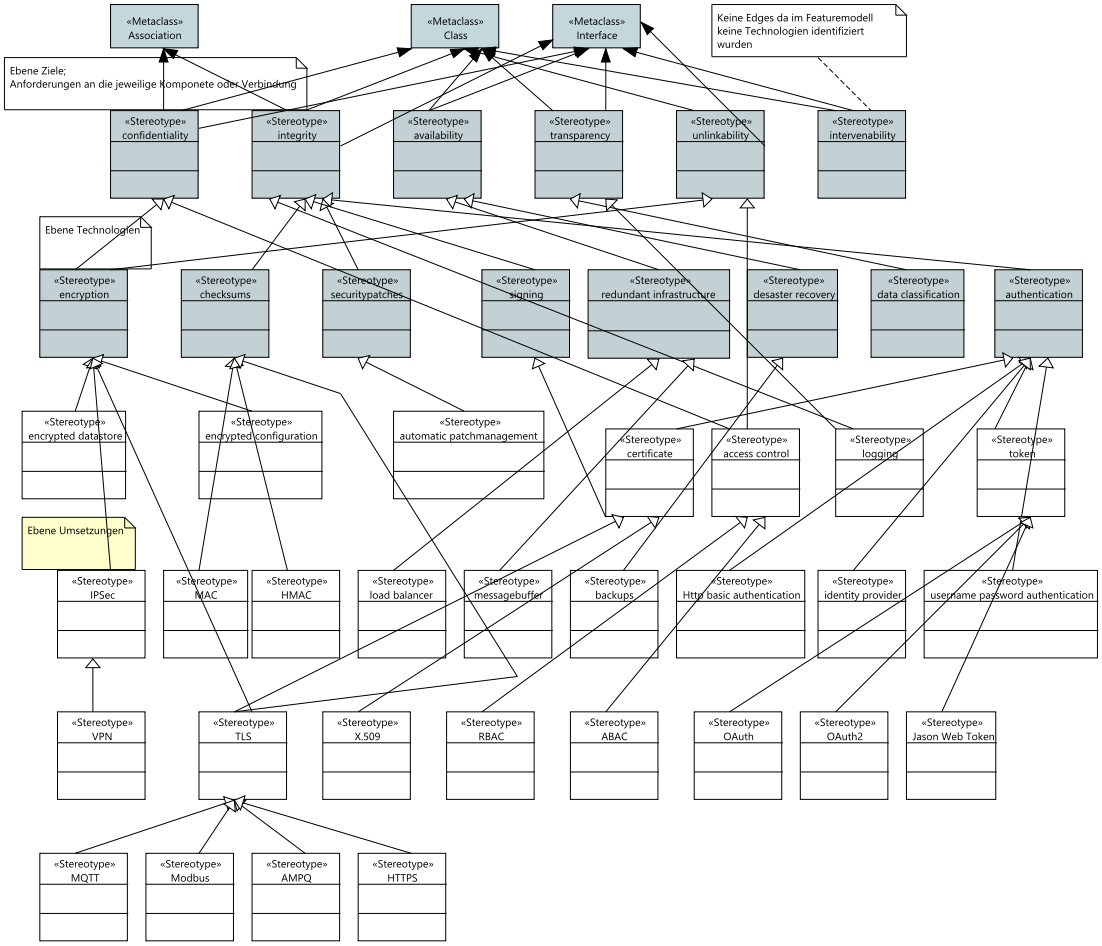

Abbildung A.1.: Das komplette IoTComponentProfile.