R1.2: IoT-Prüfanforderungen im Projekt

Anforderungen mit Prioritäten für IoT-Testware und das IoT-Testlab

Gefördert durch:

Bundesministerium für Wirtschaft und Energie

aufgrund eines Beschlusses des Deutschen Bundestages

Version 11.1, Datum: 02.06.2017

Autoren:

Frank-Walter Jäkel (Ed) – Fraunhofer IPK
Theo Margraf – AUDI AG
Stefan Stölzle – AUDI AG
Paul Hopton – Relayr
Yuliya Brynzak – Relayr
Michael Wagner – Fraunhofer FOKUS
Axel Rennoch – Fraunhofer FOKUS
Rutten, Stefan – DEKRA
Andre Wardaschka – DEKRA

Inhalt

1 Einleitung

Das Projekt „Ein Testlab und Testware für Internet der Dinge -Lösungen und -Geräte“ des BMWi kurz IoT-T hat als Ziel Firmen bei der Erstellung von IoT-basierten Lösungen und Produkten in den Bereichen Qualitätssicherung und Zertifizierung zu unterstützen. Hierzu sind die Erstellung einer IoT-Testware und die Etablierung mindestens eines IoT-Testlab geplant. Die IoT-Testware wird beim automatisierten Testen von IoT-relevanten Technologien wie z.B. Protokollen helfen und u.a. im IoT-Testlab zum Einsatz kommen. Dabei wird das IoT-Testlab Technologien wie Constrained Application Protocol (CoAP) und Message Queue Telemetry Transport (MQTT) adressieren und gleichzeitig auf Standards wie z.B. TTCN-3 aufsetzen. Das IoT-Testlab wird als praktisches Angebot für Firmen durch die DEKRA etabliert werden. Es soll Firmen ermöglichen auf IoT-Testexpertise zugreifen zu können und Anwendungen zertifizieren zu lassen. Langfristig soll Firmen ermöglicht werden qualitative, sichere und interoperable IoT-Lösungen zu erstellen.

Der vorliegende Report beinhaltet die Zusammenfassung der erarbeiteten IoT-Prüfanforderungen, welche insbesondere auf der Basis der Anwendungsszenarien in R1.1 definiert wurden. Dieses erfolgte unter der Einbeziehung der Erfahrung von DEKRA. DEKRA hat einen wesentlichen Anteil an den dokumentierten Prüfanforderungen insbesondere in Bezug auf die Arbeiten in AP4 und die Entwicklung des IoT-TestLab. Wie schon in R1.1 beschrieben, gibt es unterschiedliche Interessengruppen:

- Endanwender von IoT-Lösungen (z.B. Audi),
- Entwickler von IoT-Lösungen (z.B. Relayr)
- Anbieter von IoT-Lösungen (z.B. Relayr)
- Research (z.B. FOKUS, IPK)
- Zulassungs- und Zertifizierungsstellen (z.B. DEKRA)

Mit diesen Gruppen wurden die Prüfanforderungen in Workshops und auf Messen diskutiert. Im IoT-T Projekt sind Repräsentanten jeder dieser Gruppen vertreten. Neben den direkten Kontakten mit der Industrie wurden auch weitere Quellen aus Standardisierungsbestrebungen und der Gesetzgebung für die Erarbeitung der Prüfanforderungen verwendet [2, 3, 4, 5]. Hierbei standen insbesondere Sicherheitsaspekte sowie Teststandards im Fokus.

Auf Endanwenderseite lag der Fokus dabei stärker auf der Prüfung der bereitgestellten Services und den damit verbundenen Prozessen. Demgegenüber sind die Entwickler und Anbieter stärker an den

Tests und der Sicherstellung der Protokolle und Gateways interessiert. Das drückt sich auch in der an die DEKRA angelehnte Gliederung in Prozess, Komponente und Systeme aus. Spezifische Prüfanforderungen an Protokollen wurden in einem zusätzlichen Punkt zusammengefasst.

Die Prüfanforderungen wurden in einem Anforderungsmanagementsystem (AMS) erfasst, welches über Internet von allen Partnern erreichbar ist. Die Anforderungen wurden in Telefonkonferenzen diskutiert und der Beschreibungsumfang sukzessive erweitert. Dieser Prozess kann auch nach Abschluss von AP1 weiter geführt werden um ggf. neu identifizierte Anforderungen oder Detaillierungen der Beschreibungen zu ergänzen. Insbesondere die Kriterien zu den einzelnen Prüfanforderungen müssen in den weiteren Arbeiten in AP4 identifiziert werden. Abschließend wurde ein Workshop zur Priorisierung und ersten Auswahl der IoT-Prüfanforderungen durchgeführt. Die Ergebnisse wurden im AMS vermerkt. Dabei wurde insbesondere auch die Machbarkeit von entsprechenden Tests innerhalb des Projektes diskutiert. Die tatsächlich für eine Umsetzung in Testware und TestLab sowie in den Demonstratoren geeigneten Prüfanforderungen werden in den Umsetzungsarbeitspaketen AP2, AP3 und AP4 ausgewählt.

1.1 Erfassung und Dokumentation der IoT-Prüfanforderungen

Die Beschreibung der Prüfanforderungen orientiert sich an Standards für das Anforderungsmanagement wie Volere [1] und ISO/IEC 15504. Der Beschreibungsumfang wurde sukzessive erweitert um den Beschreibungsbedarfen der Projektpartner gerecht zu werden. Eine vollständige Liste findet sich in der nachfolgenden Zusammenfassung. Die Überschriften wurden in Englisch angegeben um auch englischsprachige Partner zu unterstützen:

- **ID /** Identifikationsnummer

  Die ID ist die eindeutige Identifikation der Anforderung.

- **Name /** Name

  Der Name ist die selbstsprechende Benennung der Prüfanforderung.

- **Description /** Beschreibung

  Die Beschreibung beinhaltet Merkmale und Eigenschaften der Prüfungsanforderung.

- **Group /** Gruppe

  Die Gruppe definiert die Zugehörigkeit zu einer Anforderungsgruppe (Prozess, Komponente, System oder Kommunikationsprotokoll). Diese Gruppe wird für die Strukturierung der Prüfanforderungen genutzt.

- **Rationale /** Begründung

  Die Begründung der Prüfanforderung ist eine detaillierte Beschreibung warum die Prüfanforderung notwendig ist.

- **Fit Criterion /** Abnahmekriterium

  Fit Criterion setzt fest, wie die Prüfanforderung verifiziert werden kann.

- **Source /** Referenzen

  Referenzen sind Verweise auf z.B. Standards, Regelungen, Gesetze.

- **Organisation /** Organisation

  Organisation aus deren Umfeld die Prüfanforderung abgeleitet ist. Typischerweise sind das die Anwendungspartner Relayr oder Audi sowie der Zertifizierungspartner DEKRA.

- **Relevance /** Wichtigkeit

  Relevance beschreibt die Art des Tests z.B. ob es sich um einen internen Test handelt oder einen Test entsprechend eines existierenden Standards.

- **Status** / Status

  Der Status beschreibt inwieweit die Prüfanforderung sich in der Umsetzung befindet- von der ersten Idee bis zur Abarbeitung.

- **Related conditions /** Bedingungen

  Die Bedingungen beinhalten Voraussetzungen der Prüfanforderung z.B. wie die Existenz von Bedingungen gegen die getestet / geprüft werden kann.

- **Requirement Anforderungstyp /** Anforderungstyp

  Der Anforderungstyp gibt an, ob es sich um eine „nicht funktionale Prüfanforderung“ handelt, welche nicht durch einen automatischen Test geprüft wird, sondern beispielsweise durch die Prüfung von Dokumenten. Der Anforderungstyp gibt an, ob es sich um eine Prüfung durch einen automatischen Test oder durch die Prüfung der Dokumentationen handelt.

- **Use cases /** Anwendungsfall

  Der Anwendungsfall beinhaltet ein Szenario zum Hintergrund der Prüfanforderung.

- **Limitations /** Einschränkungen

  Bedingungen einer Prüfanforderung, welche die Prüfung einschränken. Beispielsweise die Erfassung von personenbezogenen Daten um Auswirkungen von unterschiedlichen Nutzerverhalten zu testen. Hier kann es zu Beschränkungen des Tests durch den Betriebsrat oder das Datenschutzgesetz kommen.

- **Focus/** Fokus

  Der Fokus gibt die Zielgruppe an, für die die Prüfanforderung relevant ist.

- **Priority /** Priorität

  Die Priorität ist ein Kennzeichen für die Wichtigkeit der Prüfanforderung.

- **Target application /** Zielanwendung

  Die Anwendung, welche die Prüfanforderung für die Entwicklung und Ausführung von Tests nutzt. Dabei sind im Wesentlichen das Testlabor und die Testware gemeint. Es kann aber auch einen speziellen Bezug zu den beiden Anwendungsfällen bei Audi und Relyr geben.

- **Feedback /** Feedback

  Feedback dient zur Diskussion der Prüfanforderungen. Das Feld wird in Chapter 2 nicht ausgeführt.

- Author / Autor

  Der Autor der Prüfanforderung und die Organisation können voneinander abweichen. Dieses Feld wurde nur zur Vereinfachung der Diskussion genutzt um den Ansprechpartner zu identifizieren In der darstellung der Anforderungen in Kapitel 2 wurde dieses Feld nicht berücksichtigt.

- Classification / Klassifizierung der Sichtbarkeit

  Eine Klassifizierung kann mit „Public“, „Private“ oder „Restricted“ vorgenommen werden.

  Nur Anforderungen, welche als „Public“ eingestuft sind, werden auch in dieses Dokument übernommen. (Eine Anzahl von Prüfanforderungen wurde in der Diskussion zwischen den Partnern als „Private“ oder „Restricted“ eingestuft. Diese Prüfanforderungen sind im System weiterhin verfügbar, werden aber im aktuellen Dokument nicht aufgeführt.)

- Version / Version

  Die Version wird automatisch gesetzt und zeigt die Zahl der Änderungen zu einer Prüfanforderung an. Diese können auch in Historie zur Prüfanforderung nachgeschlagen werden.

- English_description / Englisch

  Ein Teil der Prüfanforderungen liegt in englischer Sprache vor, daher ist das Hinterlegen im System in deutscher sowie in englischer Sprache möglich.

- Priorisierung von den Partnern

  Jeder der 5 Projektpartner hat die Möglichkeit eine eigene Priorisierung der Anforderungen zu hinterlegen. Dieses dient als Basis für die Diskussion der Wichtigkeit einzelner Prüfanforderungen.

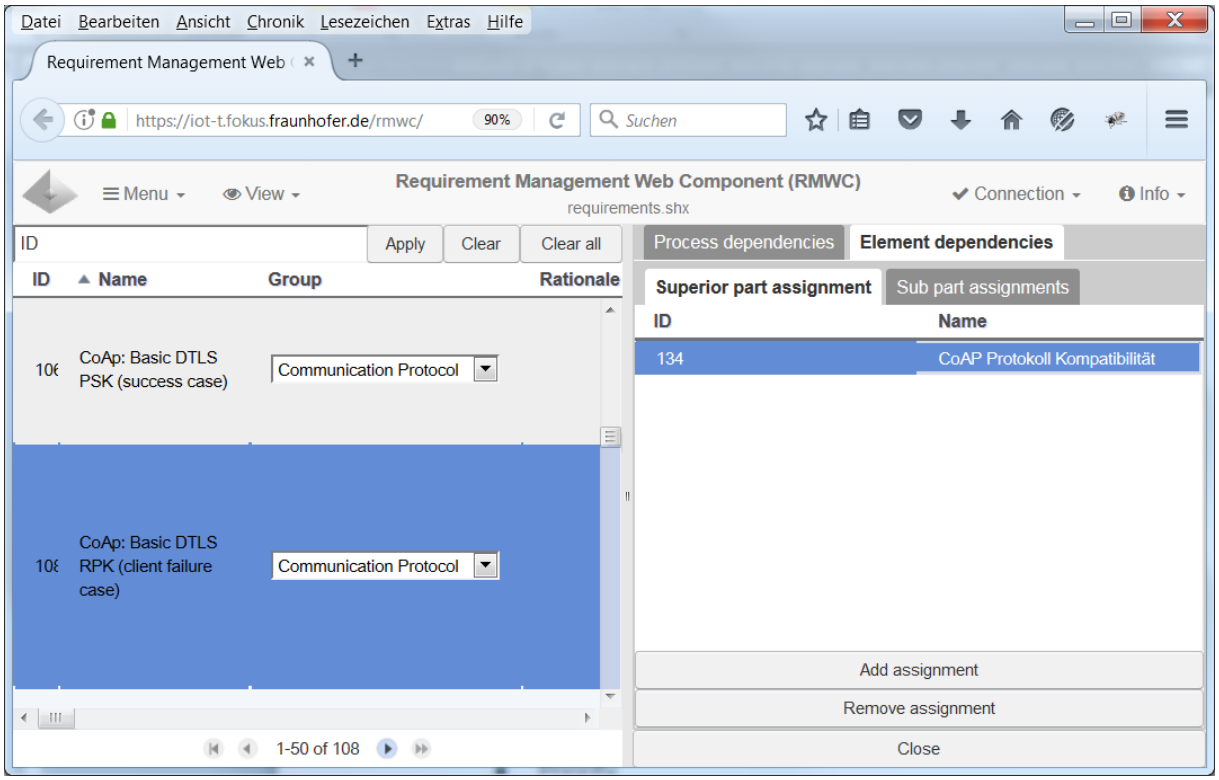

Abbildung 1: Zuordnung einer Prüfanforderung zu einer übergeordneten Prüfanforderung

Im Kapitel 2 werden im Wesentlichen der Name, der Typ, die potentielle Zielanwendung und die Beschreibung der Prüfanforderungen widergegeben um ggf. vorhandene urheberrechtliche Aspekte auszuklammern. In einigen Fällen ist die Beschreibung in englischer Sprache erfolgt.

Im Laufe der Erfassung wurden Abhängigkeiten zwischen Prüfanforderungen identifiziert. Diese wurden zusammengefasst und sind in der Beschreibung der einzelnen Prüfanforderungen als zugehörige Prüfanforderungen aufgelistet. Abbildung 2 illustriert die Zugehörigkeit einer Prüfanforderung „CoAP: Basic DTLS RPK (client failure case)“ zu einer Übergeordneten Prüfanforderung „CoAP Protokoll Kompatibilität“. In Abbildung 3 ist die übergeordnete Prüfanforderung auf der linken Seite mit den entsprechend untergeordneten Prüfanforderungen auf der rechten Seite gezeigt. Hierdurch konnten inhaltlich zusammengehörige Prüfanforderungen zusammengefasst werden.

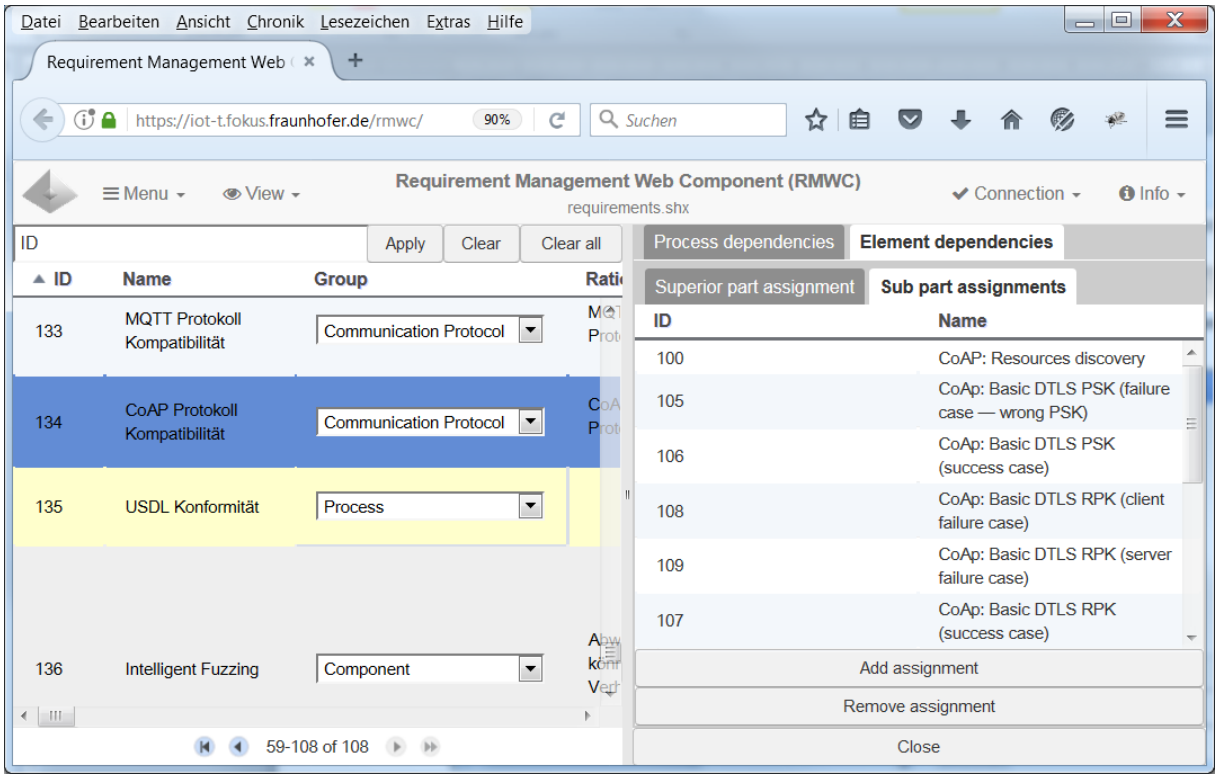

Abbildung 2: Prüfanforderung mit zugehörigen Prüfanforderungen

1.2 Vorgehen

Die Prüfanforderungen wurden von allen Projektpartnern erfasst und in Workshops und Telefonkonferenzen diskutiert. Abschließend fand ein mehrtägiger Workshop zur Priorisierung der Prüfanforderungen statt. An diesem Workshop in Berlin am Fraunhofer IPK haben Vertreter von allen Projektpartnern teilgenommen.

Im Zuge dieser Veranstaltungen wurden die Prüfanforderungen auch in den geplanten Anwendungsszenarien und Demonstratoren verortet. Ein Beispiel hierfür ist des AUDI Anwendungsszenario „modulare Shopfloor IT“ (siehe auch R1.1). Es kann in zwei Formen auftreten. In der virtuellen Realisierung werden die Anlagen und IoT-Komponenten simuliert oder auch emuliert. In der realen Welt erfolgen die Tests auf realen Anlagen und den entsprechenden IoT-Komponenten. Eine Anzahl von Tests kann in beiden Formen durchgeführt werden, andere sind ausschließlich in der realen Welt zu finden. Tests bzgl. der Reaktionszeit entsprechend der technischen Vorgaben (37) machen aber eher Sinn in der realen Welt. Die nachfolgende Liste von Prüfanforderungen und die

Abbildung 3 illustrieren das Vorgehen. Ausgewählte Prüfanforderungen für dieses Anwendungsszenario sind:

4 Prüfen von Ausnahmen/Fehlern

32 Modularer Zusammenbau eines gesamten Prozesses testen

33 Plug and Produce / Plug and play

34 Verfügbarkeit von Services zur Laufzeit in der Infrastruktur

35 Vollständigkeit von Services bezüglich Anwendungsfall

36 Robustheit in Umgebungen mit vielen sich überlagernden Frequenzen

37 Reaktionszeit entsprechend der technischen Vorgaben

123 Test auf erforderliche Rückmeldungen

124 Test auf Existenz der Services zur Steuerung der Produktionsressourcen (auch Funktion) für alle verfügbaren Businessservices

125 Test auf korrekte Zahl und Syntax von Parametern zu einer Funktion inkl. Rückgabewerte

126 Prüfung auf korrekte Semantik von Services und Rückmeldungen

127 Formale Beschreibung von Funktionen

128 Protokoll/Architektur: OPC-UA

131 Konsistente An- und Abmeldung von CPS zur Service-Registry

137 Transaktionen müssen explizit bestätigt werden

139 Testen auf Konsistenzbedingungen des Prozesses im Betrieb

140 Test der Vollständigkeit der Informationszufuhr der Services in der realen Welt

Die Verortung im Anwendungsszenario „modulare Shopfloor IT“ ist in Abbildung 3 illustriert. Dabei überwiegen die Prüfanforderungen für die Absicherung der von den Fertigungsanlagen bereitgestellten Services zu den Fertigungsmodulen der Modularen Shopfloor IT.

Das Vorgehen bildet die Basis für die Übernahme der IoT-Prüfanforderungen von unterschiedlichen Unternehmen, da sich in den Workshops herausgestellt hat, dass es hier Anwendungs- und branchenspezifische Anforderungen geben kann.

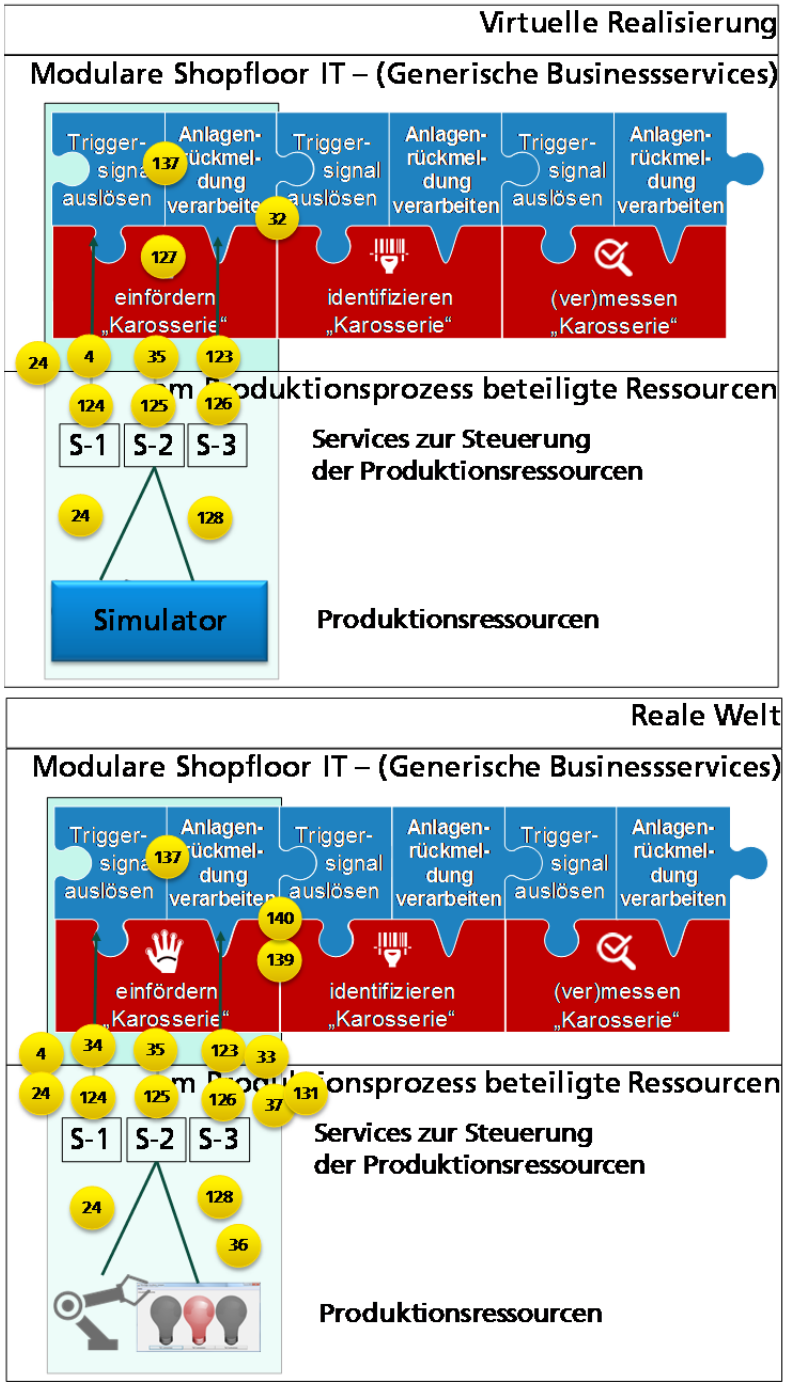

Abbildung 3: Zuordnung von Prüfanforderungen zu Anwendungsscenario „modulare Shopfloor IT“

Hier konnte folgende Einteilungen identifiziert werden:

1. Test der einzelnen IoT-Komponente nach allgemeingültigen Prüfanforderungen, wie beispielsweise die Umsetzung von Sicherheitsaspekten und Vollständigkeit unterstützter Protokolle.
2. Test der Vernetzung der IoT-Komponenten und ihres zusammen Spiels, hier sind Tests der Kompatibilität / Interoperabilität erforderlich.
3. Tests welche während des Betriebs der IoT-Komponenten erforderlich werden, wie das vorhanden sein erforderlichen Services.

Der erste Punkt kann im Wesentlichen verallgemeinert werden, wobei hier Unterschiede zwischen der privaten Nutzung und der industriellen Nutzung insbesondere bzgl. der Kriterien zu den Prüfanforderungen auftreten können. Der Punkt 2 bezieht sich auf die Einbetung einer IoT-Komponenten in eine bestehende Umgebung und ist daher auch Abhängig von den Gegebenheiten in dieser Umgebung. Der dritte Punkt entspricht einem durch Tests unterstützten Monitoring des Verhaltens der IoT-Komponenten im Betrieb und kann das Testen spezifischer prozessbezogener Regeln erfordern z.B. das Vorhandensein anderer IoT Komponenten wie RFID Tags.

2 IoT-Prüfanforderungen

Die hier aufgeführten Prüfanforderungen sind eine Untermenge der erfassten Anforderungen. Im Bericht erscheinen nur Anforderungen, welche als „public“ markiert wurden. Die Anforderungen sind automatisch aus dem AMS herausgezogen und in dieses Dokument übertragen worden.

2.1 Prozess

2.1.1 Prüfen von Ausnahmen/Fehlern (4)

Beschreibung	Bei einem Fehler müssen Maßnahmen zur Abstellung des Fehlers definiert sein. Ein Fehler muss über eine entsprechende Routine grundsätzlich gemeldet werden und möglichst weiter behandelt werden (min. Vorschlag zur Fehlerbehebung).
Anwendungsfall	**Ein Beispiel für einen Test wäre folgender:** Test ob im Fall des Abschaltens einer Anlage noch ein entsprechendes Ereignis gemeldet wird. Maschine stoppen z.B. aus Sicherheitsgründen. Nach Abschaltung erfolgt durch einen parallelen Monitoring Service eine entsprechende Meldung. Falls es hiervon Abweichungen gibt, müssen diese dokumentiert sein.
Einschränkungen	Es könnte z.B. bei Überhitzung ein sofortiges Abschalten nötig sein, dann wäre eine Nachricht nicht mehr möglich. Es sei denn die Anlage wird extern überwacht und es erfolgt eine Meldung "nicht mehr erreichbar".
Begründung	Der erste Schritt zur Behebung eines Fehlers ist die zeitnahe Meldung z.B. eine nicht eindeutige Zuordnung von RFID Identifikationen.
Anforderungstyp	Non-functional Requirements::Maintenance and Support Requirements

2.1.2 Test der logischen Verknüpfung der Funktionen (111)

Beschreibung	Die logische Verknüpfung der Funktionen, welche in einem realistischen Zusammenhang, im Sinne von Reihenfolge stehen, muss getestet werden. Die gesamte Kette muss ebenfalls die erwarteten Ergebnisse liefern. Ein Service muss als Nachfolger eines anderen Services die erforderlichen und erwarteten Informationen erhalten.
Anwendungsfall	Fertigungsplan in der Shopfloor IT
Bedingungen	Die Möglichkeiten der Ablauflogik müssen in einem Referenzmodell definiert werden. Ein Test zur Ausführung vordefinierter Services in ihrer Vernetzung wäre erforderlich.
Begründung	Bei der Verknüpfung von Diensten können nicht sinnvolle Folgen auftreten beispielsweise das Schreiben eines RFID bevor ein RFID entnommen wurde. Diese Probleme in der Ablauflogik bei der Vernetzung von Diensten soll aufgedeckt werden.
Anforderungstyp	Non-functional Requirements

2.1.3 Prüfung auf korrekte Semantik von Services und Rückmeldungen (126)

Beschreibung	Der Service oder der Controller einer Hardwarekomponente muss die erwartete Bedeutung bezüglich einer Funktion inklusive ihrer Parameter erfüllen.
Anwendungsfall	Der Anwendungsfall kann über den Shopfloor IT Demonstrator am IPK aufgezeigt werden. Hier muss die Semantik als Vorgabe dokumentiert werden und dann manuell getestet. Dabei muss das genaue Testvorgehen noch definiert werden. - Semantik der Funktionen – virtuell - Semantik an den Robotern - physikalisch

Bedingungen	Beschreibung der Semantik ggf. Ontologie - Semantik ist verfügbar - Kommunikationsprotokoll ist bekannt - Kommunikationsarchitektur ist bekannt
Einschränkungen	Offen ist inwieweit vollautomatische Tests möglich sind insbesondere bezüglich der Semantik, ggf. ist eine Ontologie hierfür erforderlich. Offen bleibt weiterhin die Prüfung der Ausführung von Funktionen direkt an Maschine/Roboter/Sensor. Diese muss weiterhin manuell geprüft werden. Die Kommunikationsschnittstelle ist Unternehmens- oder zumindest Branchen- spezifisch. Daher könnte es Testware geben aber keine Tests im TestLab.
Begründung	Es muss sichergestellt werden, dass die Services semantisch korrekt realisiert sind. Beispielsweise dürfen ein Stop im Prozess und ein "Weiter" nicht dazu führen, dass der Prozess wieder von vorne losgeht. (Sofern es nicht so definiert wurde).
Anforderungstyp	Non-functional Requirements

2.1.4 Modularer Zusammenbau eines gesamten Prozesses testen (32)

Beschreibung	Ein System aus vernetzten IoT-Komponenten muss als Ganzes getestet werden.
Anwendungsfall	Ein RFID Tag kommt in Reichweite einer RFID-Antenne und löst ein Event aus. Wird der Event korrekt verarbeitet und auf Unstimmigkeiten geprüft?
Einschränkungen	Es ist schwer jede Kombination von IoT-Lösungen und Geräten vorherzusehen aber Inkonsistenzen können im Betrieb getestet werden.
Begründung	Unterschiedliche IoT-Lösungen und Geräte müssen zusammen arbeiten können und die erwarteten Ergebnisse liefern.
Anforderungstyp	Functional Requirements

2.1.5 Verfügbarkeit von Services zur Laufzeit in der Infrastruktur (34)

Beschreibung	Es muss geprüft werden, ob der Service zur Laufzeit vorhanden ist. Test auf Vorhandensein der Services für die interne Steuerung auf der Anlage. Test der USDL, ob die Services in der Registry (z.B. OPC-UA Server) stehen (Test zur Laufzeit im Debug Mode).
Begründung	Eine Anlage könnte ausgefallen sein und damit auch ihre Services aus der Registry entfernt haben.
Anforderungstyp	Functional Requirements

2.1.6 Vollständigkeit von Services bezüglich Anwendungsfall (35)

Beschreibung	Test auf Vollständigkeit der Services für einen Anwendungsfall (z.B. Naht abdichten, Fenster einbauen, Roboterzelle, Pick by Light)
Anwendungsfall	Software Services zur Ausführung von Naht abdichten, Fenster einbauen, Roboterzelle, Pick by Light
Anforderungstyp	Functional Requirements

2.1.7 Test auf erforderliche Rückmeldungen (123)

Beschreibung	Die generischen Businessservices oder der Controller einer Hardwarekomponente müssen Rückmeldungen (Events) in einem vergebenen Format liefern.
Anwendungsfall	Der Anwendungsfall kann über den Shopfloor IT Demonstrator am IPK aufgezeigt werden. Vorgabe sind ca. 3-4 erforderliche Events pro. Prozessschritt auf einem Roboter. Beispiel1: Maschine/Roboter hat Arbeitsgang beendet, ein entsprechender Event wird ausgelöst. Beispiel2: Maschine/Roboter kann nicht weiter Arbeiten (z.B. Fehler) auch hier muss ein entsprechendes Event erfolgen. Beispiel3: Mensch ist in Sicherheitskreis eingedrungen, dann muss der Roboter sofort reagieren aber zusätzlich ein Event schicken.
Bedingungen	Offen ist inwieweit voll automatische Tests möglich sind. Da für bestimmte

	Zustände der IoT-Komponenten ggf. menschliche Eingriffe nötig sind (z.B. Erzeugen eines Fehlers).
Begründung	Es soll sichergestellt werden, dass ein Service in die IoT- Infrastruktur passt und in der Lage ist über ihren Zustand aktive Auskunft zu geben. Dieses ist erforderlich um beispielsweise ein Shopfloor IT Cockpit korrekt mit Daten zu versorgen und schnell auf Fehler zu reagieren.
Anforderungstyp	Functional Requirements

2.1.8 Test auf Existenz der Services zur Steuerung der Produktionsressourcen (auch Funktion) für alle verfügbaren Businessservices (124)

Beschreibung	Es muss jeder bereitgestellte (generische) Businessservice geprüft werden, ob dieser die Services zur Steuerung der Produktionsressourcen abdeckt. Damit ein Objekt in die modulare Shopfloor IT eingebunden werden kann, müssen Mindestanforderungen erfüllt sein, dazu gehören das Starten, Stoppen und Pausieren von Anlagen. Diese Funktionalitäten sollten standardmäßig getestet werden, dabei ist im ersten Schritt die Art der Kommunikation zweitrangig. Wichtig ist hier, dass die Anlage entsprechend angesprochen werden kann.
Anwendungsfall	Der Anwendungsfall kann über den Shopfloor IT Demonstrator am IPK aufgezeigt werden. Vorgabe sind ca. 8 erforderliche Funktionen, welche durch die OPC-UA Schnittstelle unterstützt werden müssen.
Bedingungen	Ein wichtiger Faktor bleibt die Sicherheit der IoT-Infrastruktur. Daher muss hier zwischen dem Testbetrieb mit Rückmeldungen und dem realen Einsatz ohne diese Rückmeldungen zu den Funktionen unterschieden werden.
Begründung	Diese Anforderung ist Voraussetzung für Plug-and-Play in der modularen Shopfloor IT, Test der Steuerung der Produktionsressourcen vor dem Einsatz in der Fertigung, also zur Design Zeit als Vorgabe für den Anlagenlieferanten und ggf. auch für den Betreiber und im Ramp-Up als Test und Absicherung der Schnittstelle einer Anlage.

	z.B.: Servicebasierte Inbetriebnahme von Anlagen: Basis hierfür sind die in den Funktionen jeweils hinterlegten „Services“. Sie beinhalten zum einen den feinst granular möglichen Ablauf einer Funktion, wie auch die zugehörige Anwendung (Anwendung im Sinne von Anwendungsprogramm) und die zugehörigen Technologien (im Sinne von Hardware).
Anforderungstyp	Functional Requirements

2.1.9 Test auf korrekte Zahl und Syntax von Parametern zu einer Funktion incl. Rückgabewerte (125)

Beschreibung	Der Service oder der Controller einer Hardwarekomponente muss bezogen auf die jeweilige Funktion geprüft werden, ob die Anzahl der Parameter und deren Aufbau übereinstimmen. Beispiel:"starte_Auftrag(Auftragsnummer, Maschinenprogramm, RFID Kennung)" zu "starte_Auftrag(Auftragsnummer, RFID Kennung)": Beide könnten funktionieren aber in der konkreten Implementierung darf es nur eine Form geben. Entweder erhält die Anlage das Programm im Aufruf oder sie sucht sich das Programm selber über die Auftragsnummer.
Anwendungsfall	Anwendungsfälle sind die USDL Beschreibungen des Shopfloor IT Demonstrators am IPK und bei AUDI.
Einschränkungen	Offen ist, ob zukünftig eine Alternative zu USDL verwendet wird. Der Hintergrund der Anforderung bleibt aber bestehen. Nur die Programmierung des Tests muss dann entsprechend angepasst werden.
Begründung	Es muss sichergestellt werden, dass die Parametrisierung von Funktionen zwischen dem Ausführungsservice und der IoT- Komponente übereinstimmen, da es sonst zu Fehlern bei der Abarbeitung z.B. des Fertigungsplans kommt.
Anforderungstyp	Functional Requirements

2.1.10 An- und Abmeldung von CPS zur Service-Registry konsistent (131)

Beschreibung	Bei der Anmeldung einer Anlage als CPS (Cyber Physical System) müssen die verfügbaren Dienste in die Registry geschrieben werden. Beim Abmelden müssen die Dienste, ohne andere Dienste oder CPS zu beeinflussen, gelöscht werden
Begründung	Es muss sichergestellt werden, dass die Registry mit den Service-Beschreibungen konsistent ist.
Anforderungstyp	Functional Requirements

2.1.11 Transaktionen müssen explizit bestätigt werden (137)

Beschreibung	Ein SUT darf bestimmte Funktionen nur nach Rückfrage ausführen. Diese Funktionen betreffen zum Beispiel finanzielle Transaktionen, Vertragsänderungen oder Systemupdates. Hierzu gehört auch die Sicherung des Datenzugriffes auf Anlagen z.B. Fernwartung.
Begründung	Eine Komponente könnte sonst unkontrolliert Bestellungen durchführen, siehe Beispiel: "Amazon Echo".
Anforderungstyp	Functional Requirements

2.2 System / Komponente

2.2.1 Plug and Produce / Plug and play (33)

Beschreibung	Test auf Austauschbarkeit von Anlagen oder unterschiedlichen technischen Plattformen (SPS/Robotersteuerung - Schnittstellen)
Einschränkungen	Hängt von der Spezifikation an den Anlagenlieferanten ab.
Begründung	Eine neue IoT-Lösung oder ein Gerät muss in eine vorgegebene Infrastruktur passen. Keiner wird aufgrund eines neuen Heizungsthermostates programmieren. Dieses sollte auch für neue Fertigungsanlagen gelten.
Anforderungstyp	Functional Requirements

2.2.2 Cryptographic algorithms (38)

Zielanwendung	TestLab
Beschreibung	Das SUT muss kryptografische Sicherheitsmechanismen ausschließlich gemäß international anerkannter und geprüfter Sicherheitsempfehlungen und -praktiken benutzen.
Begründung	Der Entwurf und die Implementierung von Sicherheitsmechanismen wie z.B. Verschlüsselungsalgorithmen sind komplex und daher fehleranfällig. Da jeder Fehler bei der Umsetzung von Sicherheitsmechanismen neue Angriffsmöglichkeiten bietet, ist hier auf schon bewährte Praktiken und Empfehlungen zurückzugreifen.
Anforderungstyp	Functional Requirements::Class - Data confidentiality::Family - Data confidentiality

2.2.3 Information confidentiality (39)

Zielanwendung	TestLab
Beschreibung	Das SUT muss die Vertraulichkeit von Informationen, die für die explizite Lese-Autorisierung erforderlich ist, während der Übertragung, Verarbeitung und der Speicherung schützen.
Begründung	Die Entscheidung, welche Information schützenswert ist, ist kontextabhängig.

	Der Umstand, dass einzelne Informationen Leserechte erfordern, ist jedoch ein Indiz dafür, dass eben diese Informationen als schützenswert zu betrachten sind.
Anforderungstyp	Functional Requirements::Class - Data confidentiality::Family - Data confidentiality

2.2.4 Purging information (40)

Zielanwendung	TestLab
Beschreibung	Das SUT muss die Möglichkeit bieten, vertrauliche Informationen, die für die explizite Lese-Autorisierung erforderlich sind, bei der Außerbetriebnahme oder Deaktivierung von Komponenten löschen zu können.
Begründung	Das Löschen von vertraulichen Informationen bei Außerbetriebnahme oder Deaktivierung dient der Vorbeugung versehentlicher Datenveröffentlichung bei Wiederinbetriebnahme einzelner Komponenten.
Anforderungstyp	Functional Requirements::Class - Data confidentiality::Family - Data confidentiality

2.2.5 Centralized account management support (46)

Zielanwendung	TestLab
Beschreibung	Das SUT muss die Möglichkeit bieten, die Accounts direkt über eine Management-Schnittstelle zu konfigurieren und/oder die Möglichkeit bieten, sich in ein Account-Management-System integrieren zu lassen.
Begründung	Die Accounts, mit denen auf ein SUT zugegriffen wird, müssen sich innerhalb eines Account-Management-Systems zentral steuern lassen. Eine typische Lösung stellt hierbei z.B. die Verwendung von RADIUS oder Kerberos dar. Die Verwendung von Accounts, dessen Details im SUT gespeichert und über eine Management-Schnittstelle verändert werden können, ist ebenfalls möglich.
Anforderungstyp	Functional Requirements::Class - Identification and authentication control::Family - Account management

2.2.6 Default accounts (47)

Zielanwendung	TestLab
Beschreibung	Das SUT darf keinen anonymen Zugriff auf Applikationen erlauben.
Begründung	Jeglicher nicht authentisierter Zugriff auf Applikationen stellt einen Kontrollverlust dar und ist daher unbedingt zu unterbinden.
Anforderungstyp	Functional Requirements::Class - Identification and authentication control::Family - Account management

2.2.7 Multifactor authentication (48)

Zielanwendung	TestLab
Beschreibung	Das SUT muss die Möglichkeit von Mehrfaktoren-Authentisierung für alle menschlichen Zugriffe bieten.
Begründung	Bei Security-kritischen Systemen ist ein einzelner Authentisierungsfaktor nicht ausreichend und mit einem weiteren Authentisierungsfaktor zu ergänzen.
Anforderungstyp	Functional Requirements::Class - Identification and authentication control::Family - Authentication

2.2.8 Service access (49)

Zielanwendung	TestLab
Beschreibung	Das SUT darf einem Anwender den Zugriff auf Services erst nach erfolgreicher Authentisierung dieses Anwenders erlauben.
Begründung	Die Möglichkeit, physischen Zugriff auf das System zu haben, stellt noch keine Authentisierung dar. Da die vorige Authentisierung Grundlage für folgende Autorisierung und Auditierung bildet, ist diese auch bei lokaler Verbindung zwingend notwendig.
Anforderungstyp	Functional Requirements::Class - Identification and authentication control::Family - Authentication

2.2.9 Shared accounts (50)

Zielanwendung	TestLab
Beschreibung	Das SUT muss menschlichen Anwendern die Möglichkeit bieten, sich zu identifizieren und zu authentisieren.
Begründung	Eine erfolgreiche Authentisierung bildet die Grundlage für Autorisierung und Auditierung.
Anforderungstyp	Functional Requirements::Class - Identification and authentication control::Family - Authentication

2.2.10 Unique accounts (51)

Zielanwendung	TestLab
Beschreibung	Das SUT muss die Möglichkeit bieten, menschliche Anwender individuell zu identifizieren und zu authentisieren.
Begründung	Da jeder Anwender nur die Zugriffsrechte bekommen sollte, welche er zur Erfüllung seiner Aufgaben benötigt, ist eine Individualisierung der Accounts notwendig. Geteilte Accounts würden im Gegensatz hierzu weder den Grundsatz der Minimalrechte unterstützen, noch eine Teilung von Aufgaben unterstützen.
Anforderungstyp	Functional Requirements::Class - Identification and authentication control::Family - Authentication

2.2.11 Account changeability (52)

Zielanwendung	TestLab
Beschreibung	Das SUT muss die Möglichkeit bieten, lokal gespeicherte Account-Details zu ändern.
Begründung	Jeder Account (insbesondere das Passwort) muss änderbar sein.
Anforderungstyp	Functional Requirements::Class - Identification and authentication control::Family - Authenticators

2.2.12 Authenticator feedback (53)

Zielanwendung	TestLab
Beschreibung	Das SUT muss Rückmeldungen bei Authentisierungsversuchen verschleiern.
Begründung	Bei fehlgeschlagenen Authentisierungsversuchen würde die Information, ob ein Versuch aufgrund eines falschen Passworts oder falscher Identifikation fehlgeschlagen ist, einem Angreifer Zusatzinformationen liefern, die einen weiteren Angriff erleichtern würden. Des Weiteren ist das Passwort bei der Eingabe nicht in Klartext anzuzeigen, da ansonsten die Möglichkeit besteht, dieses durch "Schulter-Blicke" auszuspähen.
Anforderungstyp	Functional Requirements::Class - Identification and authentication control::Family - Authenticators

2.2.13 Basic brute force attacks (55)

Zielanwendung	TestLab
Beschreibung	Das SUT muss einen Schutz gegen "Brute-Force" Account Attacken aufweisen.
Begründung	Wörterbuch-Passwort-Attacken sind leicht anzuwenden und weit verbreitet und daher zu erschweren.
Anforderungstyp	Functional Requirements::Class - Identification and authentication control::Family - Authenticators

2.2.14 Credential confidentiality (56)

Zielanwendung	TestLab
Beschreibung	Das SUT muss Authentifikationen gegen nicht authentisierten Zugriff und Modifikation während des Gebrauchs, der Übertragung und Abspeicherung schützen.
Begründung	Die nicht gesicherte Übertragung von Authentifikationen wie z.B. die Login-Details bei FTP oder Telnet sind abhörbar und somit durch Angreifer zu erlangen. Ungeschützte Authentifikationen bei Speicherung und Verarbeitung würde

	einem Angreifer u.U. die Authentifikationen eines Accounts mit mehr Rechten liefern.
Anforderungstyp	Functional Requirements::Class - Identification and authentication control::Family - Authenticators

2.2.15 Default credentials (57)

Zielanwendung	TestLab
Beschreibung	Das SUT muss die erfolgreiche Anwendung von Standard- Authentifikationen für den Fernzugriff auf Anwendungen verbieten.
Begründung	Standard Authentifikationen werden typischerweise zur Erst-Einrichtung von Systemen verwendet. Da diese Standard- Authentifikationen wohlbekannt sind, sollten diese unter keinen Umständen für den Fernzugriff verwendbar sein.
Anforderungstyp	Functional Requirements::Class - Identification and authentication control::Family - Authenticators

2.2.16 SUT identification (58)

Zielanwendung	TestLab
Beschreibung	Das SUT soll die Möglichkeit bieten, sich gegenüber anderen Systemen zu identifizieren und zu authentifizieren.
Begründung	Ein mögliches Angriffsszenario bildet die Vortäuschung des Ziel-SUT durch einen Angreifer, um z.B. Login-Details abzugreifen. Daher muss ein Kommunikationspartner des SUTs prüfen können, ob sie tatsächlich mit dem Ziel-SUT kommunizieren.
Anforderungstyp	Functional Requirements::Class - Identification and authentication control::Family - Authenticators

2.2.17 Unique SUT identification (60)

Zielanwendung	TestLab
Beschreibung	Das SUT muss die Möglichkeit bieten, seine Identifikation und Authentifikationen zu verändern.
Begründung	Bei einem SUT muss der Identifikation und Authentifikation veränderbar sein, um die Einstellung einzigartiger Werte zu ermöglichen. Die oftmals werkseitig voreingestellten Werte unterliegen hierbei typischerweise einem pseudo-Zufall und sind somit oftmals her leitbar.
Anforderungstyp	Functional Requirements::Class - Identification and authentication control::Family - Authenticators

2.2.18 Identifier management (61)

Zielanwendung	TestLab
Beschreibung	Das SUT muss die Möglichkeit bieten, seine Identifikation über eine externe Schnittstelle bekanntzugeben.
Begründung	Das SUT muss sich in ein Inventarisierungstool integrieren lassen.
Anforderungstyp	Functional Requirements::Class - Identification and authentication control::Family - SUT management

2.2.19 Session lock (63)

Zielanwendung	Testware&TestLab
Beschreibung	Für den Fall, dass das SUT eine Mensch-Maschinen-Schnittstelle bereitstellt, muss das SUT die Sitzung nach einem konfigurierbaren Zeitraum der Inaktivität sperren. Die Sperrung soll so lange aktiv bleiben, bis die Sitzung von dem Nutzer oder einem anderen hierfür autorisierten Nutzer mittels Identifikation und Authentifikation wieder aufgenommen wird.
Begründung	Nach Ablauf eines Zeitraums der Inaktivität des Nutzers muss davon ausgegangen werden, dass sich der Nutzer nicht mehr bei der Mensch-Maschinen-Schnittstelle befindet und diese somit von einem anderen Nutzer

	missbraucht werden kann.
Anforderungstyp	Functional Requirements::Class - Identification and authentication control::Family - Login

2.2.20 System use banner (64)

Zielanwendung	TestLab
Beschreibung	Das SUT muss für alle Mensch-Maschinen-Schnittstellen vor der Authentisierung System-Benutzungshinweise (Banner) anzeigen.
Begründung	Die Anzeige von Benutzungshinweisen hat einen rechtlichen Hintergrund. Es soll dem Nutzer vor der Nutzung des SUT rechtliche Hinweise bei z.B. Missbrauch oder Daten-Mitschnitt und -Speicherung geben.
Anforderungstyp	Functional Requirements::Class - Identification and authentication control::Family - Login

2.2.21 System use configuration (65)

Zielanwendung	TestLab
Beschreibung	Das SUT muss die Möglichkeit bieten, die System-Benutzungshinweise (Banner) durch hierfür autorisierte Nutzer zu konfigurieren.
Begründung	Die Benutzungshinweise sind z.B. im geschäftlichen Umfeld typischerweise an individuelle Firmenpolitiken anzupassen.
Anforderungstyp	Functional Requirements::Class - Identification and authentication control::Family - Login

2.2.22 Unsuccessful logins (66)

Zielanwendung	Testware&TestLab
Beschreibung	Das SUT muss nach einer festgelegten Anzahl fehlgeschlagener Authentisierungen eines Nutzers innerhalb eines festgelegten Zeitraums den Zugang verweigern. Diese Sperrung muss sich nach einem festgelegten Zeitraum automatisch aufheben und das SUT danach erneute Authentisierungsversuch erlauben.

Begründung	Ein verbreiteter Angriff sind "brute force" Attacken, bei denen versucht wird, durch eine Vielzahl von Passwort und Identifikations-Versuchen Zugriff zum SUT zu erlagen. Durch eine zeitweise Sperrung wird ein solcher Angriff nicht verhindert, jedoch zeitlich gestreckt und somit erschwert.
Anforderungstyp	Functional Requirements::Class - Identification and authentication control::Family - Login

2.2.23 Password history (68)

Zielanwendung	Testware&TestLab
Beschreibung	Das SUT muss verhindern, dass ein Nutzer sein Passwort erst nach einer festgelegten Anzahl von Änderungen wiederverwenden darf (Passwort Historie). Die minimale Passwort Historie soll hierbei 5 Einträge pro Nutzer betragen.
Begründung	Eine Passwort Historie soll den Gebrauch von neuen Passwörtern fördern.
Anforderungstyp	Functional Requirements::Class - Identification and authentication control::Family - Passwords

2.2.24 Password lifetime (69)

Zielanwendung	Testware&TestLab
Beschreibung	Das SUT muss die Möglichkeit bieten, die Nutzungsdauer eines Passwortes zu begrenzen. Dem Nutzer muss nach Ablauf dieser Nutzungsdauer ohne vorherige Änderung des Passwortes der Zugang verwehrt werden.
Begründung	Die Begrenzung der Lebensdauer eines Passwortes erzwingt die Änderung innerhalb dieses Zeitraums.
Anforderungstyp	Functional Requirements::Class - Identification and authentication control::Family - Passwords

2.2.25 Password policy (70)

Zielanwendung	Testware&TestLab
Beschreibung	Das SUT muss Passwort-Regeln für neue Passwörter implementiert haben.
Begründung	Um Wörterbuch-Attacken zu erschweren, wird oftmals die Verwendung zusätzlicher Zeichen vorgeschrieben.
Anforderungstyp	Functional Requirements::Class - Identification and authentication control::Family - Passwords

2.2.26 PKI support (71)

Zielanwendung	TestLab
Beschreibung	Im Falle, dass eine PKI-Infrastruktur verwendet wird, muss das SUT alle notwendigen PKI-Operationen unterstützen.
Anforderungstyp	Functional Requirements::Class - Identification and authentication control::Family - PKI

2.2.27 Automatic session closing (72)

Zielanwendung	TestLab
Beschreibung	Das SUT muss eine ungenutzte Sitzung nach Ablauf eines festgelegten Zeitraums schließen.
Begründung	Das automatische Schließen einer Sitzung verringert das Vorhalten von Ressourcen für nicht genutzte Sitzungen.
Anforderungstyp	Functional Requirements::Class - Identification and authentication control::Family - Session control

2.2.28 Token invalidation (73)

Zielanwendung	Testware&TestLab
Beschreibung	Das SUT muss die Verwendung temporärer Sitzungs- oder Authentisierungs-Merkmale nach dem Beenden einer Sitzung verbieten.
Begründung	Temporäre Sitzungs-Merkmale, wie z.B. temporäre Schlüssel, Tokens o.Ä.

	müssen nach dem Beenden der Sitzung ihre Gültigkeit verlieren, um eine unerlaubte Sitzungs-Weiterführung durch Angreifer zu verhindern.
Anforderungstyp	Functional Requirements::Class - Identification and authentication control::Family - Session control

2.2.29 Session identifier (87)

Zielanwendung	Testware&TestLab
Beschreibung	Das SUT muss einzigartige und nicht vorhersagbare Sitzungs-Kennungen für jede Sitzung generieren und in Gebrauch nehmen.
Begründung	Vorhersagbare Sitzungs-Kennungen erlauben einem Angreifer, eine Sitzung zu übernehmen.
Anforderungstyp	Functional Requirements::Class - Identification and authentication control::Family - Session control

2.2.30 Session identifier randomness (88)

Zielanwendung	Testware&TestLab
Beschreibung	Das SUT muss Sitzungs-Kennungen mit allgemein akzeptierten und getesteten Methoden generieren.
Begründung	Dier Generation von pseudo-zufälligen Kennungen kommt einer besonderen Bedeutung zu, die ausschlaggebend für das Nicht-Erraten von Sitzungs-Kennungen ("Session Identifier") ist. Wird diese nicht nach allgemein akzeptierten und getesteten Methoden erzeugt, besteht die Gefahr der Vorhersehbarkeit und damit der Session-Übernahme durch einen Angreifer.
Anforderungstyp	Functional Requirements::Class - Identification and authentication control::Family - Session control

2.2.31 Session invalidation (90)

Zielanwendung	Testware&TestLab
Beschreibung	Das SUT muss Sitzungs-Kennungen nach Beenden einer Sitzung entwerten (inklusive Browser-Sitzungen).

Begründung	Bei Nicht-Entwertung von Sitzungs-Kennungen ("Session-Identifier") ist eine Weiterführung der Sitzung durch einen Angreifer möglich.
Anforderungstyp	Functional Requirements::Class - Identification and authentication control::Family - Session control

2.2.32 Symmetric key authentication (74)

Zielanwendung	Testware&TestLab
Beschreibung	Im Falle der Anwendung von symmetrischen Schlüsseln muss das SUT in der Lage sein a) das geteilte Geheimnis zu validieren. b) die Integrität des geteilten Geheimnisses zu verifizieren.
Begründung	Das SUT nutzt den symmetrischen Schlüssel zur Überprüfung des Kommunikationspartners ohne diesen preiszugeben. Typische Anwendungen sind hier "Challenge-Response", "Cipher-based message authentication" (CMAC) oder "Galois counter mode" (GCM)/ "Galois message authentication code" (GMAC) Operationen.
Anforderungstyp	Functional Requirements::Class - Identification and authentication control::Family - Symmetric keys

2.2.33 Basic overload protection (75)

Zielanwendung	Testware&TestLab
Beschreibung	Das SUT muss in der Lage sein, während eines DoS Angriffs weiterhin operabel zu sein.
Begründung	Ein DoS-Angriff darf ein SUT zwar in seiner Leistung beeinträchtigen, es jedoch nicht völlig unbrauchbar machen.
Anforderungstyp	Functional Requirements::Class - Protection::Family - DoS

2.2.34 ICMP echo request (ping) (76)

Zielanwendung	Testware&TestLab
Beschreibung	Das SUT muss "ICMP echo" im Werkszustand ausgeschaltet haben.
Begründung	ICMP echo wird im Normalbetrieb nur in Ausnahmefällen benötigt, bietet einem potentiellen Angreifer jedoch ein Werkzeug zur Netzwerkerkundung.
Anforderungstyp	Functional Requirements::Class - Protection::Family - DoS

2.2.35 Recovery (77)

Zielanwendung	Testware&TestLab
Beschreibung	Das SUT muss sich nach einer Unterbrechung oder kritischem Fehler in einen bekannt sicheren Zustand begeben – z.B. 30 Sekunden.
Begründung	Inkonsistente Zustände, wie sie nach Unterbrechungen und kritischen Fehlern vorkommen, können Angreifern eine zusätzliche Angriffsfläche bieten.
Anforderungstyp	Functional Requirements::Class - Protection::Family - DoS

2.2.36 Hardening, general (78)

Zielanwendung	Testware&TestLab
Beschreibung	Das SUT muss bei der Standard-Konfiguration (ab Werk) alle ungenutzten Ports / Schnittstellen / Services geschlossen / ausgeschaltet / gestoppt haben.
Begründung	Ungenutzte Ports, Schnittstellen und Services bieten eine unnötig erweiterte Angriffsfläche. Daher muss als Teil der SUT-Härtung dafür gesorgt sein, dass nur die minimal notwendigen Ports, Schnittstellen und Services aktiviert sind.
Anforderungstyp	Functional Requirements::Class - Protection::Family - Hardening

2.2.37 Root access (80)

Zielanwendung	TestLab
Beschreibung	Das SUT muss direkten Zugriff auf Betriebssystem-Services und -Ressourcen verbieten.
Begründung	Ein direkter Zugriff auf Betriebssystem-Services oder -Ressourcen ("root-

	access") umgeht die Autorisierung nach dem Minimalprinzip. Bei einem erfolgreichen Angriff auf ein solches Nutzer-Konto hätte ein Angreifer vollen Zugriff auf alle Services und Ressourcen.
Anforderungstyp	Functional Requirements::Class - Protection::Family - Hardening

2.2.38 MITM resistance (81)

Zielanwendung	Testware&TestLab
Beschreibung	Das SUT muss resistent gegen "Man-in-the-middle"-Attacken sein.
Begründung	Das SUT muss feststellen können, ob der Kommunikationspartner auch der erwünschte Kommunikationspartner ist. Eine Umleitung des Verkehrs zu einem Angreifer darf daher nicht zu einer Verletzung der Datenintegrität oder Vertraulichkeit führen.
Anforderungstyp	Functional Requirements::Class - Protection::Family - MITM

2.2.39 Replay resistance (82)

Zielanwendung	Testware&TestLab
Beschreibung	Das SUT muss resistent gegen "Replay"-Attacken (Datenwiederholung) sein.
Begründung	Der Mitschnitt und späteres Wiederholen von Kommunikationsfragmenten kann (sofern nicht erkannt und unterbunden) zu einer nicht autorisierten Interaktion mit dem SUT führen.
Anforderungstyp	Functional Requirements::Class - Protection::Family - Replay

2.2.40 Communication integrity (83)

Zielanwendung	Testware&TestLab
Beschreibung	Das SUT muss die Integrität der übertragenen Daten schützen.
Begründung	Jegliche Veränderungen von Daten während der Übertragung können zu einer nicht autorisierten Interaktion mit dem SUT oder seinem Kommunikationspartner führen.
Anforderungstyp	Functional Requirements::Class - System Integrity::Family - Integrity protection

2.2.41 Error handling (84)

Zielanwendung	TestLab
Beschreibung	Das SUT darf im Fehlerfall keine Informationen liefern, die von Angreifern ausgenutzt werden könnten.
Begründung	Spezifische, detaillierte Fehler-Rückmeldungen können einem Angreifer Zusatzinformationen liefern, die einen zielgerichteten Angriff erlauben.
Anforderungstyp	Functional Requirements::Class - System Integrity::Family - Integrity protection

2.2.42 File integrity (85)

Zielanwendung	Testware&TestLab
Beschreibung	Das SUT muss die Integrität von heruntergeladenen Dateien, wie z.B. Konfigurations-Dateien oder Software-Pakete, vor der Inbetriebnahme überprüfen.
Begründung	Beschädigte oder modifizierte Dateien können das SUT kompromittieren oder ganz außer Betrieb nehmen.
Anforderungstyp	Functional Requirements::Class - System Integrity::Family - Integrity protection

2.2.43 Input validation (86)

Zielanwendung	Testware&TestLab
Beschreibung	Das SUT muss vor Verwendung Syntax und Inhalt jeder Nachricht überprüfen.
Begründung	Ungeprüfte Weiterverarbeitung von Eingaben kann z.B. zu Puffer-Überläufen oder dem Einschleusen von schädlichem Code führen.
Anforderungstyp	Functional Requirements::Class - System Integrity::Family - Integrity protection

2.2.44 Session integrity (89)

Zielanwendung	TestLab
Beschreibung	Das SUT muss die Authentizität von Kommunikationssitzungen sicherstellen.
Begründung	Nicht authentisierte Kommunikation kann prinzipiell auch von einem Angreifer

	stammen.
Anforderungstyp	Functional Requirements::Class - System Integrity::Family - session control

2.2.45 Remote Software updates (91)

Zielanwendung	Testware&TestLab
Beschreibung	Das SUT muss die Möglichkeit bieten, Software-Aktualisierung mittels Fernwartung durchzuführen.
Begründung	Da bei einer Software potentiell auch nach Auslieferung Security-relevante Sicherheitslücken aufgedeckt werden können, muss es eine Möglichkeit geben, die Korrekturen effektiv aus der Ferne einspielen zu können.
Anforderungstyp	Functional Requirements::Class - Updates::Family - Software updates

2.2.46 Software package signing (92)

Zielanwendung	Testware&TestLab
Beschreibung	Das SUT muss die digitale Signatur von Software-Paketen überprüfen und darf das Software-Paket nur bei erfolgreicher Überprüfung aktivieren.
Begründung	Unsignierte Software Pakete können aus unsicheren Quellen stammen und dürfen daher nicht in Betrieb genommen werden.
Anforderungstyp	Functional Requirements::Class - Updates::Family - Software updates

2.2.47 Audit event information (93)

Zielanwendung	TestLab
Beschreibung	Das SUT muss bei Ereignis-Aufzeichnungen mindestens Zeit und Datum, Art des Ereignisses (falls anwendbar) und das Ergebnis (Erfolg oder Misserfolg) aufzeichnen können.
Einschränkungen	ggf. wegen Betriebsrat abschaltbar
Begründung	"Audit-Records" sind zur Aufzeichnung von Interaktionen eines Nutzers mit dem SUT gedacht, um einen Aktionsnachweis zu generieren. Um aussagefähig zu sein, müssen diese Aufzeichnungen ein Minimum von Informationen

	beinhalten.
Anforderungstyp	Functional Requirements::Class - Use control::Family - Auditing

2.2.48 Audit read permissions (94)

Zielanwendung	TestLab
Beschreibung	Das SUT muss den Scheib- und Lesezugriff auf Ereignis-Aufzeichnungen beschränken und darf den Zugriff nur entsprechend autorisierten Nutzern erlauben.
Begründung	Ereignis-Aufzeichnungen können sensitive Daten und Informationen beinhalten und sind daher vor unberechtigtem Lesezugriff zu schützen. Ein nicht autorisierter Schreibzugriff auf Ereignis-Aufzeichnungen würde es erlauben, die Spuren eines unerlaubten Ereignisses zu verschleiern und ist daher zu verhindern.
Anforderungstyp	Functional Requirements::Class - Use control::Family - Auditing

2.2.49 Auditable events (95)

Zielanwendung	TestLab
Beschreibung	Das SUT muss alle (Nutzer-)Ereignisse im Zusammenhang mit Zugriffs-Kontrolle, Konfiguration und Backup- und Wiederherstellung aufzeichnen.
Begründung	Änderungen der Zugriffs-Kontrolle, Konfiguration und Backup/Restore zählen zu den wichtigsten, aufzuzeichnenden Ereignissen.
Anforderungstyp	Functional Requirements::Class - Use control::Family - Auditing

2.2.50 Authorization enforcement (96)

Zielanwendung	TestLab
Beschreibung	Das SUT muss es ermöglichen, jedem menschlichen Nutzer unterschiedliche Autorisierungen gemäß seiner Aufgaben und unter Berücksichtigung geringstmöglicher Rechte zu vergeben.
Begründung	Autorisierungen sollten nur so weit reichen, wie es die Erfüllung von Aufgaben

	erfordert. Darüber hinaus reichende Rechte bieten die Möglichkeit des Missbrauchs.
Anforderungstyp	Functional Requirements::Class - Use control::Family - Authorization

2.2.51 Least privilage and segregation of duties (97)

Zielanwendung	TestLab
Beschreibung	Das SUT muss bei jedem menschlichen Nutzer einer (logischen) Schnittstelle vor Benutzung die Identifikation, Authentisierung und Autorisierung prüfen.
Begründung	Die Überprüfung von Identifikation, Authentisierung und Autorisierung erlaubt es, den Aufgaben entsprechend Rechte individuell nach dem Minimalprinzip zu vergeben.
Anforderungstyp	Functional Requirements::Class - Use control::Family - Authorization

2.2.52 Rights of Roles (98)

Zielanwendung	TestLab
Beschreibung	Das SUT muss (direkt oder durch einen alternativen Mechanismus) in der Lage sein, einem Nutzer Autorisierungen mittels Rollenzuweisungen zu vergeben.
Begründung	Die Wartung der Rechte eines Nutzers sind mittels Rollenkonzept einfacher.
Anforderungstyp	Functional Requirements::Class - Use control::Family - Authorization

2.2.53 Intelligent Fuzzing (136)

Beschreibung	Das SUT muss bei externen Schnittstellen resistent gegen Protokoll-Abweichungen sein.
Begründung	Abweichungen bei Semantik und Syntax können zu Fehlfunktionen oder ungewolltem Verhalten des SUTs führen.
Anforderungstyp	Functional Requirements

2.3 Kommunikationsprotokoll

2.3.1 Verwendung/Übersetzung unterschiedlicher Protokolle (24)

Beschreibung	Protokolle unterschiedlicher Hersteller müssen integrierbar sein.
Anwendungsfall	Eine neue Anlage soll integriert werden. Jetzt muss geprüft werden, ob diese die Spezifikation bzgl. der Infrastruktur erfüllt und für zukünftige Entwicklungen einsetzbar ist. Diese wird voraussichtlich gegen die Dokumentation der Anlage geprüft.
Begründung	Test, ob Controller oder Services unterschiedliche Kommunikationsprotokolle unterstützen können entsprechend des jeweiligen Standes der Technik verändern. Dieses gilt insbesondere bezüglich Abwärtskompatibilität.
Anforderungstyp	Non-functional Requirements

2.3.2 Unterstützung OPC-UA (128)

Beschreibung	Eine IoT-Komponente, welche mit OPC-UA Schnittstelle ausgeliefert wird, muss OPC-UA vollständig abbilden. Externe OPC-UA-Schnittstellen des SUT müssen zum Standard konform sein und diesen vollständig unterstützen.
Anwendungsfall	Austausch einer IoT-Komponente in einer OPC-UA Umgebung.
Bedingungen	Hier wird kein neuer Test erwartet sondern die Möglichkeit gegeben existierende Tests zu nutzen, da bereits Ansätze für den Test von OPC-UA existieren.
Einschränkungen	Dieser Test sollte bereits vom Lieferanten der Komponente veranlasst werden.
Begründung	Es muss sichergestellt werden, dass bei der Verwendung von OPC-UA dieses auch vollständig und korrekt unterstützt wird. Externe OPC-UA-Schnittstellen des SUT müssen zum Standard konform sein und diesen vollständig unterstützen.
Anforderungstyp	Non-functional Requirements

2.3.3 Formale Beschreibung von Funktionen (127)

Beschreibung	Services müssen nach einem formalen Standard beschrieben sein und damit ausführbar sein. Funktionen müssen nach einem formalen Standard (z.B. USDL) beschrieben

	sein.
Anforderungstyp	Non-functional Requirements

2.3.4 MQTT Protokoll Kompatibilität (133)

Beschreibung	Externe MQTT-Schnittstellen des SUT müssen OASIS Standard konform sein und diesen vollständig unterstützen.
Anwendungsfall	(Server UND (!) Client-seitige Protokoll Tests)
Begründung	MQTT ist eines der zentral benutzen Protokolle im IoT-Kontext und daher im Fokus.
Anforderungstyp	Functional Requirements

Die folgenden zugehörigen Prüfanforderungen sind dabei besonders zu berücksichtigen:

- MQTT Connection
- MQTT: Allows single connectivity per credentials
- MQTT: Can connect with good credentials
- MQTT: Connectivity is established in X secs
- MQTT: Handle > 200 clients connecting every 5s
- MQTT: Handle > 200 clients publish every 1s
- MQTT: Reject mailformed payload
- MQTT: TLS is respected
- Ping functionality for different protocol implementation (MQTT, CoAP))

2.3.5 CoAP Protokoll Kompatibilität (134)

Beschreibung	Externe CoAP-Schnittstellen des SUT müssen RFC7252 konform sein und diesen vollständig unterstützen.
Begründung	CoAP ist eines der zentral benutzen Protokolle im IoT-Kontext und daher im Fokus.
Anforderungstyp	Functional Requirements

Die folgenden zugehörigen Prüfanforderungen sind dabei besonders zu berücksichtigen:

- CoAP: Resources discovery
- CoAp: Basic DTLS PSK (failure case — wrong PSK)
- CoAp: Basic DTLS PSK (success case)
- CoAp: Basic DTLS RPK (client failure case)
- CoAp: Basic DTLS RPK (server failure case)
- CoAp: Basic DTLS RPK (success case)
- CoAP: Handle GET blockwise transfer for large resource
- CoAp: Perform GET transaction (CON mode, piggybacked response) in a lossy context
- CoAP: Perform GET transaction containing several URI-Path options (CON mode)
- Ping functionality for different protocol implementation (MQTT, CoAP))

3 Zusammenfassung und Ausblick

Ein erster Satz an IoT-Prüfanforderungen steht zur Verfügung und wird im Laufe des Projektes weiter entwickelt. Insbesondere werden die erforderlichen Kriterien für die Prüfung der Anforderungen ergänzt. Eine Auswahl der Prüfanforderungen wird mit Testware für die automatische Prüfung in den weiteren Arbeitspaketen unterlegt werden.

Entsprechend ihrer spezifischen IoT-Umgebungen können Unternehmen eine Auswahl der Prüfanforderungen übernehmen um ihre IoT-Komponenten und deren Zusammenspiel zu überprüfen. Dieses wird innerhalb des Projektes im weitere in zwei Demonstratoren auf der Basis von Vorgaben von AUDI und Relayr getestet.

Für die Verfolgung der Entwicklung der Prüfanforderungen wurde ein Verwaltungssystem für die Prüfanforderungen etabliert, welches für alle Partner zugänglich ist. Hierrüber wurden bereits die bestehenden Prüfanforderungen von allen Projektpartnern erfasst und parallel bearbeitet. Jede beteiligte Organisation besitzt einen eigenen Zugang zum System und kann die Prüfanforderungen über eine Internetseite bearbeiten. Eine Auswahl dieser Prüfanforderungen wurde als „öffentlich“ Klassifiziert und bildet den Inhalt von Kapitel 2 des vorliegenden Reports. Im Zuge der Weiterentwicklung wird der aktuelle Beschreibungsumfang weiter ergänzt und ggf. werden weitere Prüfanforderungen identifiziert oder verworfen. Dieses wird in AP3 insbesondere im Verlauf der Tests der Testverfahren und der Testware an realistischen Szenarien von Relayr und Audi überprüft. AP2 wird die Anforderungen für die Erarbeitung von Testware nutzen um möglichst automatische Tests zu unterstützen. Für AP4 bilden die Prüfanforderungen eine erste Basis für die Entwicklung eines Testlaps.

4 Referenzen

1. http://www.volere.co.uk/. Letzter Zugriff Februar 2017.
2. IEC 62443 Industrial communication networks – Network and system security
3. Common Criteria Protection Profile vom Bundesamt für Sicherheit in der Informationstechnik (BSI)
4. ISO/IEC 17025 General requirements for the competence of testing and calibration
5. IT-Sicherheitsgesetz - Schutz für die digitale Infrastruktur. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Broschueren/IT-Sicherheitsgesetz.pdf?__blob=publicationFile&v=5, Zugriff Juni 2017

5 Anhang A: Struktur der Prüfanforderungen

- Requirement
  - Non-functional Requirements
    - Maintenance and Support Requirements
      - Prüfen von Ausnahmen/Fehlern
    - Verwendung/Übersetzung unterschiedlicher Protokolle
    - Test der logischen Verknüpfung der Funktionen
    - Prüfung auf korrekte Semantik von Services und Rückmeldungen
    - Unterstützung OPC-UA
    - Formale Beschreibung von Funktionen
  - Functional Requirements
    - Modularer Zusammenbau eines gesamten Prozesses testen
    - Plug and Produce / Plug and play
    - Verfügbarkeit von Services zur Laufzeit in der Infrastruktur
    - Vollständigkeit von Services bezüglich Anwendungsfall
    - Class - Data confidentiality
      - Family - Data confidentiality
        - Cryptographic algorithms
        - Information confidentiality
        - Purging information
    - Class - Identification and authentication control
      - Family - Account managment
        - Centralized account management support
        - Default accounts
      - Family - Authentication
        - Multifactor authentication
        - Service access
        - Shared accounts
        - Unique accounts

- Family - Authenticators
  - Account changeability
  - Authenticator feedback
  - Basic brute force attacks
  - Credential confidentiality
  - Default credentials
  - SUT identification
  - Unique SUT identification
- Family - SUT management
  - Identifier management
- Family - Login
  - Session lock
  - System use banner
  - System use configuration
  - Unsuccessful logins
- Family - Passwords
  - Password history
  - Password lifetime
  - Password policy
- Family - PKI
  - PKI support
- Family - Session control
  - Automatic session closing
  - Token invalidation
  - Session identifier
  - Session identifier randomness
  - Session invalidation
- Family - Symmetric keys
  - Symmetric key authentication

- Class - Protection
  - Family - DoS

        - Basic overload protection
        - ICMP echo request (ping)
        - Recovery
    - Family - Hardening
        - Hardening, general
        - Least functionality
        - Root access
    - Family - MITM
        - MITM resistance
    - Family - Replay
        - Replay resistance
- Class - System Integrity
    - Family - Integrity protection
        - Communication integrity
        - Error handling
        - File integrity
        - Input validation
    - Family - session control
        - Session integrity
- Class - Updates
    - Family - Software updates
        - Remote Software updates
        - Software package signing
- Class - Use control
    - Family - Auditing
        - Audit event information
        - Audit read permissions
        - Auditable events
    - Family - Authorization
        - Authorization enforcement
        - Least privilage and segregation of duties

        - Rights of Roles
    - Test auf erforderliche Rückmeldungen
    - Test auf Existenz der Services zur Steuerung der Produktionsressourcen (auch Funktion) für alle verfügbaren Businessservices
    - Test auf korrekte Zahl und Syntax von Parametern zu einer Funktion incl. Rückgabewerte
    - An- und Abmeldung von CPS zur Service-Registry konsistent
    - MQTT Protokoll Kompatibilität
    - CoAP Protokoll Kompatibilität
    - Intelligent Fuzzing
    - Transaktionen müssen explizit bestätigt werden