FORSCHUNGSINITIATIVE Zukunft BAU

F 3081

Hannes Raddatz, Michael Rethfeldt
Martin Kasparick, Arne Wall, Dirk Timmermann

Absicherung der Gerätekommunikation im Smart Home unter Verwendung des Schutzprofils für Smart Meter Gateways

Bundesinstitut für Bau-, Stadt- und Raumforschung
im Bundesamt für Bauwesen und Raumordnung

Fraunhofer IRB Verlag

F 3081

2018

ISBN 978-3-7388-0229-0

Fraunhofer IRB Verlag
Fraunhofer-Informationszentrum Raum und Bau

Postfach 80 04 69
70504 Stuttgart

Nobelstraße 12
70569 Stuttgart

Telefon 07 11 9 70 - 25 00
Telefax 07 11 9 70 - 25 08

E-Mail irb@irb.fraunhofer.de

www.baufachinformation.de

www.irb.fraunhofer.de/bauforschung

FORSCHUNGSINITIATIVE Zukunft BAU

Absicherung der Gerätekommunikation im Smart Home unter Verwendung des Schutzprofils für Smart Meter Gateways

Endbericht

Der Forschungsbericht wurde mit Mitteln der Forschungsinitiative Zukunft Bau des Bundesinstitutes für Bau-, Stadt- und Raumforschung gefördert.
(Aktenzeichen: SWD-10.08.18.7-15.10 / II3-F20-14-1-012)
Die Verantwortung für den Inhalt des Berichtes liegt bei den Autoren.

Bearbeiter: M.Sc. Hannes Raddatz, M.Sc. Michael Rethfeldt, Dipl.-Inf. Martin Kasparick, M.Sc. Arne Wall

Projektleiter: Prof. Dr.-Ing. Dirk Timmermann

Universität Rostock Traditio et Innovatio

Inhaltsverzeichnis

Abbildungsverzeichnis ... **3**

Tabellenverzeichnis ... **5**

Abkürzungsverzeichnis ... **6**

1. Einführung ... **7**

2. Aktuelle Entwicklungen ... **9**

2.1. Smart Metering und Smart Home ... 9

2.2. Open Source im Smart Home ... 11

2.3. Protokoll-Trends ... 14

3. Grundlagen ... **16**

3.1. Entwicklungen aus vorangegangenen BBSR-Projekten ... 16

3.2. Smart Meter Gateway ... 17

3.2.1. Schnittstellen ... 17

3.2.2. BSI Schutzprofil und Technische Richtlinien ... 18

3.3. Protokolle ... 20

3.3.1. Constrained Application Protocol (CoAP) ... 20

3.3.2. Concise Binary Object Representation (CBOR) ... 23

3.3.3. CBOR Object Signing and Encryption (COSE) ... 23

3.3.4. Object Security for Constrained RESTful Environments (OSCORE) ... 23

3.3.5. OAuth 2.0 & OpenID Connect ... 24

3.3.6. Authentication and Authorization for Constrained Environments (ACE) ... 25

3.3.7. Lightweight M2M (LwM2M) ... 26

3.4. Weiterführende Informationen ... 26

3.4.1. Videostreaming über CoAP ... 26

3.4.2. Online-Ausweisfunktion des neuen elektronischen Personalausweises ... 28

3.4.3. Binary Decision Diagrams (BDDs) ... 29

4. Sicherheits-Framework ... **33**

4.1. Problemstellung ... 34

4.2. Anwendungsszenarien ... 35

4.3. Allgemeine Anforderungen ... 38

4.4. Abgeleitete Anforderungen vom BSI Schutzprofil für das Smart Meter Gateway ... 40

4.5. Konzept ... 41

4.5.1. Kombinierte Smart Home- und Smart Metering-Infrastruktur ... 42

4.5.2. Protokolle des Sicherheits-Framework ... 44

4.5.3. Kommunikationsabläufe 46
4.5.4. Ergänzungen für eine beschleunigte Umsetzung 51
4.6. Erweiterungen auf Basis des Sicherheit-Framework 54
4.6.1. Dezentrales Regelmanagement 54
4.6.2. ANTs 65
5. Demonstrator - Konzeption & Entwicklung von Prototypen 67
5.1. Aktivierung eines Nutzergerätes mithilfe des neuen elektronischen Personalausweises 67
5.2. Hinzufügen und steuern eines neuen Smart Home-Gerätes 67
5.3. Mehrwertdienst durch Anbindung an Service Plattform 70
5.4. Audio/Video-Gegensprechanlage 71
6. Fazit und Ausblick 72
7. Ergebnistransfer 73
Literaturverzeichnis 75
Anlagen 80

Abbildungsverzeichnis

Abbildung 1: Einsatzbereiche des Smart Meter Gateway nach BSI [31] 9
Abbildung 2: Rolloutszenario für intelligente Messsysteme nach dem Gesetz zur Digitalisierung der Energiewende, [13] 11
Abbildung 3: Logos ausgewählter Open Source Smart Home-Projekte 12
Abbildung 4: Logos der Konsortien [26], [27] 13
Abbildung 5: Publish-Subscribe-Beziehung von MQTT, [28] 14
Abbildung 6: Trends in der IoT-Entwicklung - Verwendung der 3 wichtigsten Nachrichtenprotokolle in IoT-Produkten in den letzten Jahren. Das Diagramm wurde auf Grundlage der Daten aus Anlage I erstellt. 15
Abbildung 7: Schnittstellen des Smart Meter Gateways nach [31], [32] 18
Abbildung 8: Übersicht der BSI Dokumente zum Smart Meter Gateway nach [31] 19
Abbildung 9: Eigenschaften einer RESTful-Architektur 21
Abbildung 10: Protokoll-Stack von CoAP 21
Abbildung 11: Allgemeiner Autorisierungsablauf unter ACE 25
Abbildung 12: ACE-Autorisierungsablauf mit Introspection und Rechenlast bei AS (SMGW) 26
Abbildung 13: ACE-Autorisierungsablauf ohne Introspection und Rechenlast bei RS (TV) 26
Abbildung 14: Abrufen von Meta-Informationen eines Streams nach [78] 27
Abbildung 15: Observe-Mechanismus zum Videostreaming nach [78] 27
Abbildung 16: Beispiel Videostreaming zu Nutzergerät durch Observe-Mechanismus 27
Abbildung 17: Kommunikation während der Online-Ausweisfunktion des nPA 28
Abbildung 18: BDD einer booleschen Funktion, [56] 30
Abbildung 19: Eliminierungsregel für BDDs, [56] 31
Abbildung 20: Verschmelzungsregel für BDDs, [56] 32
Abbildung 21: Lokalisierung von Geräten die von Mirai infiziert wurden (November 2016) [5] 33
Abbildung 22: Angriff eines Smart Home-Netzwerks aufgrund von Fehlkonfiguration 34
Abbildung 23: Angriff auf Smart Home-Netzwerk mit Smart Meter Gateway, [32] 35
Abbildung 24: Registrierung neuer Nutzer und Autorisierung von Nutzergeräten, [26] 36
Abbildung 25: Steuerung von Geräten im Smart Home mit integriertem Sicherheits-Framework 37
Abbildung 26: Sichere Kommunikation zwischen Gegensprechanlage und Smartphone 38
Abbildung 27: Infrastruktur des Sicherheits-Framework am Beispiel eines Gebäudes mit vier Wohneinheiten ausgestattet mit Smart Home-Geräten und Smart Metering-System 43
Abbildung 28: Protokolle des Sicherheits-Framework angeordnet im TCP/IP Schichtenmodell 44
Abbildung 29: Interner Kommunikationsablauf, [32] 47

Abbildung 30: Fernzugriff auf Smart-Home-Geräte, [32] 49
Abbildung 31: Verbindungsaufbau von Smart Home-Gerät zu externem Nutzergerät, [26] 51
Abbildung 32: Erweiterung mit verteilten Datenbanken in Smart Home-Netzwerken 52
Abbildung 33: Partielle Integration des Sicherheits-Framework in IKEA TRÅDFRI, [32], [67] 53
Abbildung 34: Regelmanagement mit zentralem Hub 54
Abbildung 35: Dezentrales Regelmanagement 55
Abbildung 36: Automatisierung einer Heizungssteuerung 55
Abbildung 37: Entscheidungsdiagramme (BDDs) der Beispielregeln 62
Abbildung 38: XOR-BDD der Bespielregeln a und c 62
Abbildung 39: Ansatz mit zentralem Regelmanagement 63
Abbildung 40: Ansatz mit verteiltem Regelmanagement 64
Abbildung 41: Regeltransfer beim dezentralen Regelmanagement 65
Abbildung 42: Network Trust Zones nach ANTs 65
Abbildung 43: Netzwerk-Infrastruktur nach ANTs 66
Abbildung 44: In den Demonstrationsszenarien verwendete Geräte: ESP32, Raspberry Pi, Android Smartphone 67
Abbildung 45: RFID-Lesegerät ReinerSCT Standard für nPA [79] 67
Abbildung 46: WLAN-Testgeräte basierend auf ESP32 und Raspberry Pi 3 68
Abbildung 47: Strom bei konstanter Betriebsspannung von 5V während eines Sendevorganges zur Temperaturwert-Übermittlung 69
Abbildung 48: Betriebszeit mit einem 18650er Li-Ion Akku mit 3400mAh und 3,7V 70
Abbildung 49: Kamera-gehäuse für Raspberry Pi [80] 71
Abbildung 50: Gehäuse für Wandpanel [80] 71
Abbildung 51: Raspberry Pi 3 [80] 71

Tabellenverzeichnis

Tabelle 1: Informationsflüsse über Schnittstellen des Smart Meter Gateway nach [34] 20
Tabelle 2: Gegenüberstellung von CRUD und CoAP-Operationen 22
Tabelle 3: Bedeutung von CoAP-Operationen 22
Tabelle 4: Rollen von Teilnehmern in OAuth 2.0 24
Tabelle 5: Ausgewählte Schutzziele der Informationssicherheit für Smart Home-Netzwerke 39

Abkürzungsverzeichnis

BDD ..... Binary Decision Diagram
BSI ..... Bundesamt für Sicherheit in der Informationstechnik
DPWS ..... Devices Profile for Web Services
HTTP ..... Hypertext Transfer Protocol
IoT ..... Internet of Things
LTE ..... Long Term Evolution
M2M ..... Maschine-zu-Maschine
MAC ..... Medium Access Control
NAT ..... Network Address Translation
OASIS ..... Organization for the Advancement of Structured Information Standards
OCF ..... Open Connectivity Foundation
SHGW ..... Smart Home Gateway
SMGW ..... Smart Meter Gateway
TCP ..... Transmission Control Protocol
UDP ..... User Datagram Protocol
WCF ..... Windows Communication Foundation
WLAN ..... Wireless Local Area Network

1. Einführung

Die rasante Entwicklung des Internets und die steigende Anzahl der vernetzten Geräte, gepaart mit mangelhafter oder nicht konsequent durchgesetzter Sicherheit haben dazu geführt, dass nahezu jeder Benutzer in vollem Umfang von unbefugten Dritten überwacht werden kann [1]. Die Steuerungen von Gebäudeautomationssystemen, darunter auch Smart Home-Lösungen, können ohne großen Aufwand von Dritten übernommen werden. Die bestehenden Installationen, die auf den klassischen Gebäudeautomationsstandards, wie z. B. KNX, LON und BACnet, basieren, weisen in der Regel keine oder nur rudimentäre Sicherheitsmaßnahmen auf, die die Gerätekommunikation vor dem Abhören oder der unbefugten Übernahme der Steuerung schützen.

Dieses Problem wurde für Smart Metering-Systeme vom Bundesamt für Sicherheit in der Informationstechnik (BSI) adressiert, indem ein Schutzprofil für Smart Meter Gateways[1] und die Technischen Richtlinien TR-03109 erstellt wurden. Das Schutzprofil beschreibt Bedrohungsszenarien und leitet Sicherheitsanforderungen ab. Beide Dokumente fordern unter anderem die Verwendung von Sicherheitsmechanismen, wie z. B. Verschlüsselung, Integritätsschutz und Authentifizierung. Weiterhin werden die möglichen Kommunikationspfade zwischen den verschiedenen Schnittstellen des Smart Meter Gateways spezifiziert und Anforderungen abgeleitet. Ein Smart Metering-System, das den Anforderungen des Schutzprofils und den Technischen Richtlinien entspricht, kann nach dem aktuellen Wissensstand als sicher angenommen werden.

Ein Ziel des Forschungsvorhabens ist es, die Kommunikation im Smart Home-Netzwerk unter Berücksichtigung des Schutzprofils für Smart Meter Gateways (siehe Abschnitt 3.2) und des aktuellen Standes der Technik (siehe Abschnitt 2) abzusichern. Das im Projekt zusammengestellte Sicherheits-Framework für Smart Home-Systeme bietet den Nutzern hohe Sicherheitsstandards, ohne dabei den Konfigurationsaufwand zu erhöhen. Die Sicherheitsmechanismen arbeiten im Hintergrund und sind folglich für den Nutzer transparent. Der Einrichtungs- und Wartungsaufwand ist unter Beachtung der Sicherheitsanforderungen minimal gehalten, um einen hohen Grad an Benutzerfreundlichkeit und Akzeptanz des Sicherheits-Framework zu erreichen.

Viele Marktteilnehmer versuchten in den letzten Jahren ihre Produktlinien auf eine eigene, oftmals proprietäre Technologie und Infrastruktur zu begrenzen und schränken damit ihre eigene Flexibilität bezüglich eines umfassenden Produktportfolios ein. Dies führt dazu, dass die Geräte verschiedener Hersteller oft nicht kompatibel sind und eine Anbindung der Geräte von Drittherstellern unmöglich ist. Die Verwendung offener Schnittstellen ermöglicht dagegen eine Gerätekommunikation unabhängig vom Hersteller und fördert die Konkurrenzfähigkeit kleiner bzw. neuer Firmen am Markt, da der Entwicklungsaufwand für neue Produkte reduziert wird. Daher ist ein weiteres Ziel dieses Projektes eine frei verfügbare und sichere Lösung für die Gerätekommunikation zu konzipieren, welche als eine solide Grundlage für ein zukunftssicheres und erweiterbares Smart Home von Kunden sowie Herstellern genutzt werden kann. Wir erhoffen uns mit dem zusammengestellten Sicherheits-Framework insbesondere jungen Unternehmen den Einstieg in den Smart Home-Bereich zu erleichtern und folglich eine höhere Innovationsrate durch einen stärkeren Wettbewerb in diesem Wirtschaftssektor zu ermöglichen.

Ausgangsbasis für das Projekt ist die derzeitige Art und Weise der Nutzung von Geräten im Bereich des Internet of Things (IoT). Produkte zur Automatisierung von Gebäuden oder für Sicherheitstechnik

[1] Common Criteria: “Protection Profile for the Gateway of a Smart Metering System (Smart Meter Gateway PP)“

werden häufig ohne die erforderlichen Fachkenntnisse nachgerüstet und verursachen durch Konfigurationsfehler Schwachstellen in den Systemen. Häufig können solche Lücken über lange Zeit ausgenutzt werden, ohne dass die Nutzer sich der Problematik bewusst werden und die die Privatsphäre von Personen in erheblichen Maße gefährden. Es gibt unzählige Beispiele im privaten ([2], [3]) wie im öffentlichen Überwachungsbereich ([4]–[6]), die die Aktualität und Bedeutsamkeit dieser Problematik belegen. In Abschnitt 4.1 wird diese Problemstellung weiterführend erörtert und auf die zunehmende Bedrohung durch Bot-Netzwerke eingegangen. Das Forschungsprojekt befasst sich mit der Lösung dieses Problems, indem ein Konzept eines Sicherheits-Framework für das Smart Home erarbeitet wird. Das Smart Meter Gateway mit seinen Sicherheitsmechanismen als sichere Instanz wird dazu in das Smart Home-Netzwerk integriert. Weiterhin wird das Smart Meter Gateway zu einem Smart Home Gateway aufgewertet, welches die Umsetzung innovativer Mehrwertdienste über eine sichere Infrastruktur ermöglicht. Zusätzlich zu den Sicherheitseigenschaften bietet die Kombination von Smart Metering- und Smart-Home-System Kostenvorteile für den Eigentümer bzw. Mieter sowie Anreize für die Vermarktung und den Rollout neuer Smart Metering-Systeme. Die Einführung von Mehrwertdiensten für das Smart Meter Gateway ist seitens der Industrie und Wohnungswirtschaft gewünscht, wie verschiedene Workshops und Konferenzen gezeigt haben.

Der Projektbericht ist wie folgt aufgebaut: In Abschnitt 2 wird ein Überblick über aktuelle Entwicklungen in den Bereichen Smart Metering Rollout und Smart Home gegeben. Insbesondere die Untersuchung von aktuellen Entwicklungen am Markt und in der Open Source-Szene in Abschnitt 2.2 sowie Umfragen über IoT-Kommunikationsprotokolle in Abschnitt 2.3 haben Trends aufgezeigt, die die Konzeption des Sicherheits-Framework beeinflusst haben. Basierend auf Erkenntnissen und Entwicklungen aus den vorangegangenen BBSR-Projekten (siehe Abschnitt 3.1) wurde eine sichere Kommunikation zwischen Geräten im Smart Home konzipiert. Der Fokus lag dabei auf der Verwendung des Smart Meter Gateways und der Berücksichtigung relevanter Schutzprofile und Technischer Richtlinien, die in Abschnitt 3.2 eingeführt werden. Weiterhin werden etablierte und von Entwicklern bevorzugt genutzte Kommunikations- und Sicherheitsprotokolle sowie neue, effizientere Protokolle benutzt, die in Kapitel 3.3 dieses Berichtes eingeführt werden. Die Demonstration der Absicherung des Smart Home mithilfe von prototypischen Implementierungen umfasst eine Vielzahl von zusätzlichen Teilaspekten, die in Abschnitt 3.3.7 gesondert betrachtet werden. In Kapitel 4 wird das Sicherheits-Framework definiert. Dazu findet in Abschnitt 4.1 eine Beschreibung der Gefahren und Probleme bei der Integration aktueller Smart Home-Geräte statt. Im folgenden Abschnitt werden Anwendungsszenarien konstruiert, die typische Teilaspekte eines Smart Home und seiner Nutzung abbilden. Davon ausgehend werden in Kapitel 4.3 allgemeine Anforderungen an das Sicherheits-Framework aufgestellt. Ausgehend von den Anforderungen des BSI Schutzprofils für Smart Meter Gateways, werden unter Punkt 4.4 Kriterien für das Sicherheits-Framework abgeleitet. In Abschnitt 4.5 wird das unter diesen Kriterien erarbeitete Konzept vorgestellt und detailliert beschrieben. Am Ende dieses Abschnitts (4.5.4) werden Ergänzungen zum Konzept aufgeführt, die eine zügige Integration des Sicherheits-Framework in Smart Home-Produkte ermöglichen. Kapitel 4.6 beschreibt Anwendungen, die auf das Sicherheits-Framework aufbauen, zusätzliche Funktionen realisieren und die Erweiterbarkeit des Konzeptes demonstrieren. In Abschnitt 5 werden die Prototypen und Implementierungen von ausgewählten Aspekten des Sicherheits-Framework, die während der Bearbeitung des Projektes entstanden sind, beschrieben und analysiert. Während Kapitel 6 die Ergebnisse des Forschungsprojektes zusammenfasst, werden in Abschnitt 7 die aus dem Projekt hervorgegangenen Veröffentlichungen aufgeführt.

2. Aktuelle Entwicklungen

Das Smart Metering Rollout befindet sich in Deutschland noch in der Anfangsphase und wird kontrovers in der Öffentlichkeit diskutiert. Abschnitt 2.1 stellt die geplanten Einsatzgebiete der intelligenten Kommunikationseinheit eines Smart Metering-Systems, dem Smart Meter Gateway, dar und fasst die Berichterstattung über das Gateway und das Rollout im Allgemeinen zusammen.

Die in der Industrie vorherrschende Entwicklung proprietärer Smart Home-Lösungen mit geschlossenen Ökosystemen erhält in den letzten Jahren zunehmend Konkurrenz aus dem Open Source-Bereich. Diese Entwicklung und die Erwartungen der Konsumenten bezüglich Interoperabilität von Geräten und Systemen erfordern in Zukunft einen Wandel in diesem Automatisierungssektor. In Abschnitt 2.2 werden die Ansätze der Open Source-Gemeinschaft und Reaktionen der Industrie auf diese Entwicklungen beschrieben.

In Kapitel 2.3 werden Trends aus Umfragen der letzten Jahre im Bereich der IoT-Kommunikationsprotokolle abgeleitet und interpretiert.

2.1. Smart Metering und Smart Home

Die Umstellung auf Smart Metering-Systeme soll in Deutschland mehrere Bereiche der digitalen Infrastruktur unterstützen. Die in Abbildung 1 dargestellten Cluster sind vom Bundesamt für Sicherheit in der Informationstechnik (BSI) definierte Silos, die durch die Verbreitung von Smart Metering-Systemen, hier im speziellen die Installation der Smart Meter-Kommunikationseinheit, das sogenannte Smart Meter Gateway (SMGW), in Zukunft von der Digitalisierung profitieren sollen. Das BSI beschreibt in den zugehörigen Schutzprofilen und Technischen Richtlinien (siehe Abschnitt 3.2.2) die Anforderungen an die Cluster Smart Metering (1) und Smart Grid/Mobility (2) detailliert, während das Cluster Smart Home (3) kaum Erwähnung findet. Paragraf § 21 des Messstellenbetriebsgesetzes enthält die bisher präziseste Aussage öffentlicher Institutionen bezüglich des Clusters Smart Home.

Abbildung 1: Einsatzbereiche des Smart Meter Gateway nach BSI [31]

§ 21 I Nr. 4a Messstellenbetriebsgesetz (MsbG):

„Ein intelligentes Messsystem muss ein Smart-Meter-Gateway beinhalten, das […] *offen für weitere Anwendungen und Dienste ist und dabei über die Möglichkeit zur Priorisierung von bestimmten Anwendungen verfügt, wobei nach Anforderung der Netzbetreiber ausgewählte energiewirtschaftliche und in der Zuständigkeit der Netzbetreiber liegende Messungen und Schaltungen stets und vorrangig ermöglicht werden müssen,* […]*“*

Das BSI beschreibt für die Silos 1 und 2 mögliche Anwendungen, wie eine flexible Tarifwahl, automatische Übermittlung von Verbrauchsdaten an Stromanbieter und die Fernsteuerung regelbarer Verbraucher und Erzeuger (Photovoltaik-Anlagen, Blockheizkraftwerke, …) zur Netzstabilisierung. Die genannten Funktionen werden mit vereinzelten Einschränkungen bereits während des Rollouts in Deutschland unterstützt. Smart Home-Anwendungen, -Beispiele oder zielgerichtete Anforderungen fehlen bisher:

„[…] *Unklar seien außerdem die gesetzlichen Vorgaben zu Schnittstellen für E-Mobilität oder Smart-Home-Anwendungen.* […]“ – Energate-Messenger [7]

Smart Metering Rollout in den Medien

Die Berichterstattung der Presse über den Einbau und die Verbreitung von Smart Metering-Systemen fiel im Jahr 2017 und Anfang 2018 weitestgehend kritisch aus. Im März 2017 berichteten die Tageszeitungen über eine Studie von Forschern der Universität Twente aus den Niederlanden [8]. Messungen zeigten, dass diverse Smart Meter häufig falsche Ergebnisse liefern. Bei der Beschaltung mit stark nicht linearen Lasten konnten Abweichungen um bis zu 582 Prozent ermittelt werden. Die Forscher kritisieren die nicht mehr zeitgemäßen Anforderungen an elektronische Energiezähler in den Niederlanden [9]. Obwohl die getesteten Smart Meter in Deutschland keine Zulassung erhalten hätten, da die deutschen Spezifikationen die neuen Lasten wie LEDs und Energiesparlampen berücksichtigen, wurde die Studie in Deutschland von der Presse häufig als allgemeingültig ausgelegt. Einige deutsche Hersteller von Smart Metern veröffentlichten als Reaktion auf diese Artikel Stellungnahmen, deren Medienwirksamkeit jedoch erheblich geringer ausfiel. Im Mai 2017 attestierten rund 800 Fachexperten auf dem VDE/FNN-Fachkongress „Zählen – Messen – Prüfen“ den am Markt verfügbaren Smart Metern Nachholbedarf im Bereich der Interoperabilität [10]. Ein weiteres Problem über das im April 2018 berichtet wurde, betrifft die Verzögerungen während der Zertifizierung des Smart Meter Gateways durch das BSI. Der in Abbildung 2 dargestellte Zeitplan des Bundesministeriums für Wirtschaft und Energie (BMWi) für das Smart Metering Rollout, sah eigentlich schon für 2017 eine Umrüstung bisheriger Stromzähler zu intelligenten Messsystemen, digitale Verbrauchszähler kombiniert mit der Kommunikationseinheit SMGW (siehe Abschnitt 3.2), für ausgewählte Verbraucher und Erzeuger vor. Recherchen bei den Herstellern Theben, EFR, Discovergy, EMH, Landis+Gyr und PPC sowie beim BSI haben ergeben, dass Smart Meter Gateways nach dem BSI-Schutzprofil und zugehörigen Technischen Richtlinien noch nicht am Markt erhältlich sind (Pilotprojekte ausgeschlossen), weil sich diese teilweise noch in Entwicklung bzw. in der Zertifizierungsphase durch das BSI befinden. Um das benötigte Zertifikat zu erhalten, wird die entwickelte Hard- und Software des Gerätes vom BSI bzw. autorisierten Prüfzentren getestet. Bis zum April 2018 wurde jedoch noch keine Zertifizierung eingereichter Geräte vom BSI abgeschlossen. Folglich sind derzeit keine Smart Meter Gateways am Markt verfügbar und der Einbau intelligenter Messsysteme muss weiterhin verschoben werden [11], [12].

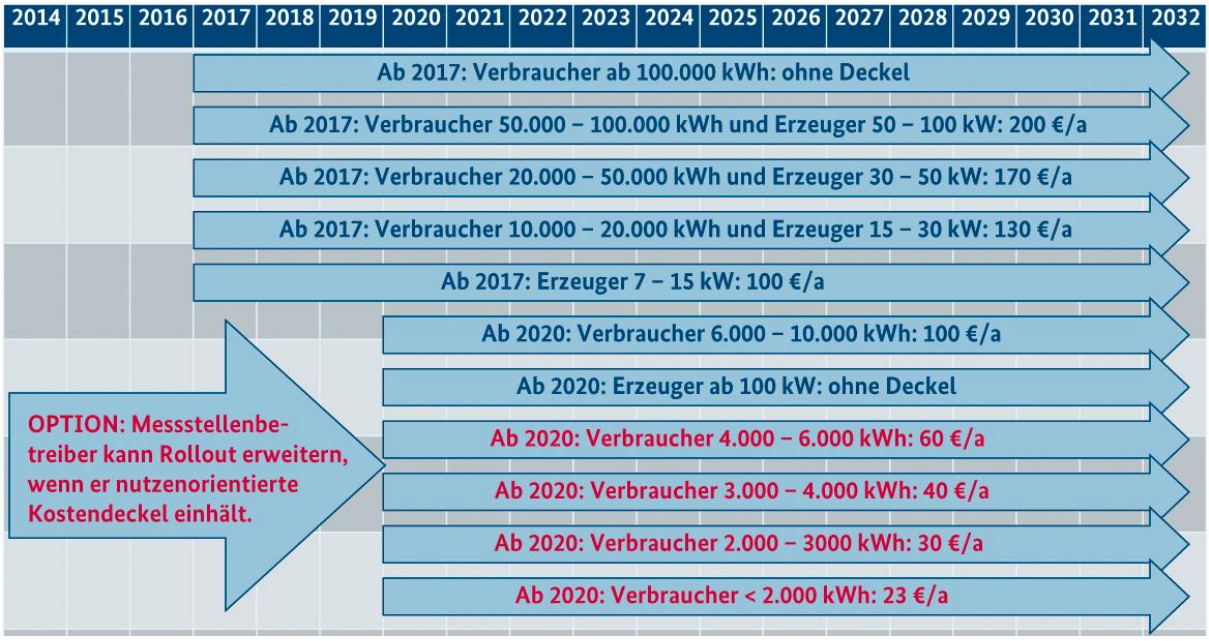


Abbildung 2: Rolloutszenario für intelligente Messsysteme nach dem Gesetz zur Digitalisierung der Energiewende, [13]

2.2. Open Source im Smart Home

Die bisherige Produktpolitik der im Smart Home-Sektor aktiven Unternehmen bestand bislang darin, ein abgeschlossenes Ökosystem für das Smart Home zu schaffen. Dies wird erreicht, indem proprietäre Kommunikationsprotokolle verwendet werden, die die Auswahl an verfügbaren Produkten auf Geräte des gleichen Herstellers beschränken. Die dadurch erzielte Kundenbindung führt jedoch zu einer entsprechenden Nachfrage an Diversität im Produktportfolio. Diesen Bedarf kann ein einzelnes Unternehmen kaum leisten, da im Smart Home eine Vielzahl an verschiedenen Sensoren, Aktoren und Schnittstellen benötigt und folglich Expertise in den jeweiligen Produktbereichen erfordert wird. Aufgrund dieser Bestrebungen in den letzten Jahren können aktuell (Stand: Januar 2017) zwei Trends im Smart Home-Bereich identifiziert werden. Während eine Entwicklung der Open Source-Szene entstammt, wird der zweite Trend durch die Bildung von Konsortien verschiedener Hersteller ausgelöst.

Teile der Open Source-Gemeinschaft, die sich mit dem Smart Home befassen und u. a. quelloffene Software- und Hardwareprojekte bereitstellen, haben ein verstärktes Interesse erfahren und an Bedeutung gewonnen (OpenHAB → Eclipse Smart Home). Die Bestrebungen der Gemeinschaft haben diverse Projekte hervorgebracht, die größtenteils auf dem gleichen Grundkonzept basieren. Dieses Konzept wird oft als „System of Systems“ bezeichnet und steht für die Orchestrierung der verschiedenen offenen und proprietären Smart Home-Systeme, wie z. B. KNX, Z-Wave, ZigBee, Phillips HUE und Apple HomeKit, durch ein übergeordnetes Automatisierungssystem. Es kann als ein eigenständiges System betrachtet werden, welches auf Softwareseite durch Schnittstellen, auch Gateways oder Bindings genannt, und auf Hardwareseite durch entsprechende Adapter mit den vorhandenen Systemen kommuniziert. Dies ermöglicht eine systemübergreifende Interaktion zwischen den Geräten verschiedener Hersteller und über Protokollgrenzen hinweg, sowie eine zentrale Verwaltungsinstanz für den Nutzer. Je nach Popularität und Konzeption des Projektes werden mehr oder weniger fremde Protokolle unterstützt. Projekte, die als „System of Systems“

konzipiert wurden, haben jedoch den konzeptionellen Nachteil, dass sie nur sehr eingeschränkt Sicherheitskonzepte umsetzen können, da sie in dieser Hinsicht keinen Einfluss auf die angeschlossenen Smart Home-Systeme haben. Schutzziele der Informationssicherheit, wie

Abbildung 3: Logos ausgewählter Open Source Smart Home-Projekte

Vertraulichkeit, Authentizität und Integrität, können daher nur auf die Kommunikation zwischen Verwaltungsinstanz und User Interface bzw. Clientgerät, wie z. B. Smartphone und Tablet, angewendet werden. Einige größere Open Source-Projekte, die auf diesem Konzept beruhen, sind OpenHAB [14], Calaos [15], Domoticz [16], Home Assistant [17], Home Genie [18], Ago Control [19], Freedomotic [20], MajorDoMo [21], WOSH [22], LinuxMCE [23] und FHEM [24]. Die Projekte unterscheiden sich stark in den Anforderungen an die Programmierkenntnisse der Nutzer, den unterstützten Schnittstellen zu vorhanden Kommunikationsprotokollen (u. a. KNX, Z-Wave, ZigBee, Ubiquiti, Phillips HUE, MQTT, X10, 1wire, ModBus, EnOcean, Insteon, Intertechno, diverse Multimediageräte wie TVs und A/V-Receiver, …) und wurden zudem in verschiedenen Programmiersprachen (Java, C/C++, Python, Perl, …) geschrieben.

Weiterhin gibt es innerhalb der Open Source-Gemeinschaft Bestrebungen, die unterhalb des Konzeptes „System of Systems“ ansetzen und proprietäre Smart Home-Systeme durch offene Lösungen zur Gerätekommunikation ersetzen. Etablierte Konzepte und Implementierungen existieren in diesem Bereich jedoch nur wenige, weil dazu weitere spezialisierte Kenntnisse u. a. zur Schaltungstechnik, Leiterplattenentwurf und Mikrocontroller-Programmierung nötig sind. Ziel dieser Bestrebungen ist es, eigene Sensoren und Aktoren für das Smart Home bzw. Adapter für existierende Geräte zu realisieren und die Dokumentation des Kommunikationsprotokolls, Schaltpläne, Leiterplattenentwürfe und den Quellcode der Firmware für die Mikrocontroller zu veröffentlichen. Ein solcher Ansatz ermöglicht die Integration von etablierten Sicherheitskonzepten in die Kommunikation zwischen den einzelnen Geräten sowie eine Kontrolle der richtigen Umsetzung durch die Gemeinschaft nach dem Open Source-Prinzip. Um den Aufwand für die Entwicklung solcher Systeme einzugrenzen,

werden die Steuerungsfunktionen und Nutzerinteraktionen an die oben genannten „System of Systems“-Projekte durch die Integration von entsprechenden Schnittstellen ausgelagert. Als der bisher am weitesten fortgeschrittener Vertreter in dieser Kategorie kann auf das SmartHomatic-Projekt [25] verwiesen werden. Es bietet ein verschlüsseltes Kommunikationsprotokoll zwischen den Geräten und dem zentralen Steuergerät und funkt im Sub-GHz-Band mit 868 MHz.

Der zweite Trend, welcher sich derzeit abzeichnet, findet aufseiten der Hersteller proprietärer Smart Home-Systeme statt. Die Konzepte hinter den herstellereigenen Ökosystemen werden aufgeweicht, indem sich Unternehmen zu Gruppen zusammenschließen, in denen gemeinsam Frameworks oder Kommunikationsstandards erarbeitet werden, um die Produkte der teilnehmenden Hersteller miteinander zu vernetzen. Im Falle der Frameworks wird dies durch die Schaffung von Schnittstellen realisiert. Dies ermöglicht den Herstellern einem, der Nachfrage entsprechenden, breiteren Produktportfolio und der Kritik an einem abgeschlossenen Ökosystem zu begegnen. Folglich können sich die Unternehmen auf ihre Kompetenzbereiche konzentrieren und den eigenen Qualitätsansprüchen eher gerecht werden. Diese Entwicklung ist ebenfalls vorteilhaft für den Konsumenten, da keine vollständige Bindung an ein Ökosystem bzw. Produktportfolio eines Herstellers mehr nötig ist, um ein Smart Home-System aufzubauen. Ein Beispiel für diese Öffnung der Ökosysteme ist das Framework des Eclipse Smart Home-Projektes [26]. Dieses ging Ende 2013 aus dem Open Source-Projekt OpenHAB [14] hervor und wird derzeit von den fünf Partnerunternehmen QIVICON (Deutsche Telekom), ProSyst, JUNG, Zoo Automation und aleon unterstützt (Stand Januar 2016). Das

Abbildung 4: Logos der Konsortien [26], [27]

Framework folgt dem „System of Systems“-Ansatz und ermöglicht durch Gateways die Übersetzung der Kommunikation zwischen verschiedenen proprietären und offenen Protokollen auf einem zentralen Vermittlungsgerät. Die Hersteller müssen daher im besten Fall nur ein Binding für ihr Kommunikationsprotokoll bereitstellen, um eine Integration ihrer Produkte in Eclipse Smart Home zu ermöglichen.

Auch die im Februar 2016 umbenannte und neu ausgerichtete Open Connectivity Foundation (OCF), die aus dem Open Interconnect Consortium (OIC) hervorgegangen ist, hat ein neues Ziel für die Bestrebungen der Unternehmensgemeinschaft (mehr als 170 Partner, darunter Cisco Systems, General Electric, Intel, MediaTek, LG, Canon, Samsung, Microsoft, Qualcomm) formuliert [27]. Unter dem Namen OIC wurde das Ziel verfolgt, Standards für das Internet of Things (IoT) basierend auf dem Constrained Application Protocol (CoAP) zu entwickeln. Nach der Umbenennung wurde dieses Ziel angepasst. Es bleibt der Fokus auf IoT bestehen, jedoch soll aufgrund der gesammelten Erfahrungen, u. a. aus dem Smart Home-Bereich, die Vereinheitlichung von proprietären IoT-Protokollen verschiedener Hersteller im Vordergrund stehen, um in Zukunft eine nahtlose Kommunikation zwischen Geräten unabhängig vom Hersteller zu ermöglichen. Die Bestrebungen des OCF erfordern tief greifendere Veränderungen für Hersteller als das Framework von Eclipse Smart Home. Allerdings wird damit das Fundament für eine IoT-Gerätekommunikation gelegt, die sich langfristig als besser skalierende (kein zentraler Vermittler nötig), sicherere und vielseitigere Alternative (mehr Möglichkeiten für IoT/Smart Home-Managementlösungen) erweisen kann. Andererseits bietet das

Eclipse Smart Home Framework eine für Hersteller kurzfristig umsetzbare Lösung, um bereits am Markt existierende Geräte aufzuwerten.

2.3. Protokoll-Trends

Es existiert eine Vielzahl an Kommunikationsprotokollen, die Geräte direkt oder über das Internet miteinander verbinden. Die verschiedenen Lösungen wurden für spezielle Einsatzgebiete entwickelt und jedes besitzt dementsprechend je nach Verwendungszweck Vor- und Nachteile gegenüber anderen Protokollen. Ein weiterer Aspekt ist die Verbreitung in der Entwicklergemeinschaft, die in der Regel eine Folge der Vorteile eines Protokolls für einen spezifischen Anwendungsfall ist. Folglich verändert sich mit der Entwicklung neuer Protokolle oder aufgrund der zunehmenden Verbreitung bereits existierender Lösungen die Antwort auf die Frage nach dem besten Protokoll für ein Szenario. In Abbildung 6 sind die drei am häufigsten verwendeten Kommunikationsprotokolle im Bereich des Internet of Things (IoT) aufgeführt. Die Daten sind der jährlichen IoT Developer Survey der Eclipse Foundation entnommen (siehe Anlage I) und veranschaulichen die meistgenutzten Protokolle unabhängig vom Einsatzgebiet. Während das Hypertext Transfer Protocol (HTTP) weiterhin eines der meistgenutzten Kommunikationsprotokolle für das Internet respektive das Laden von Websites ist, haben sich weitere Protokolle für andere Einsatzgebiete etablieren können. Im Bereich des IoT, das zu einem großen Teil aus Sensornetzwerken besteht, haben sich in den letzten Jahren zwei Protokolle als Favoriten für Entwickler erwiesen. Message Queue Telemetry Transport (MQTT) wurde von IBM und Cirrus Link Solutions entwickelt und ist seit 2013 durch OASIS als IoT-Protokoll standardisiert. Es ist für Maschine-zu-Maschine (M2M) Kommunikation ausgelegt und basiert, wie in Abbildung 5 dargestellt, auf einem Kommunikationsfluss mit Veröffentlichern (engl. Publisher), z. B. Sensoren, einem Vermittler (engl. Broker), der Sensordaten sammelt und anbietet, und Abonnenten (engl. Subscriber), die Messwerte bestimmter Sensoren abonnieren können.

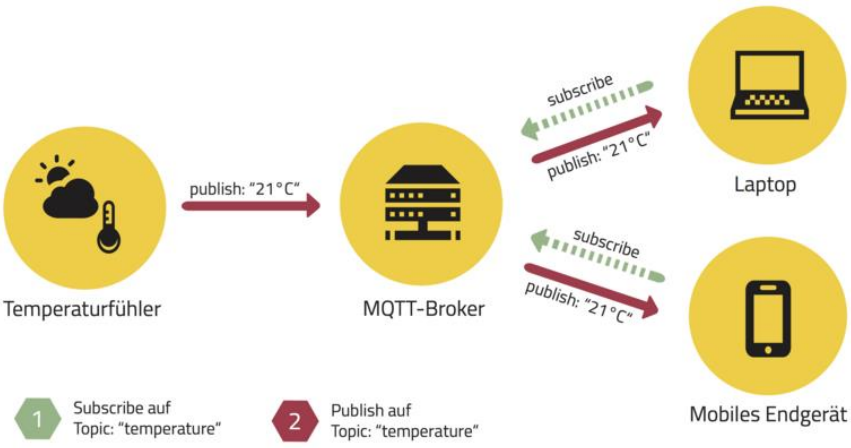


Abbildung 5: Publish-Subscribe-Beziehung von MQTT, [28]

Das Protokoll arbeitet verbindungsorientiert, indem es die Aufgaben des Verbindungsaufbaus zu anderen Teilnehmern und deren Aufrechterhaltung an das Transmission Control Protocol (TCP) auslagert. Die Nachteile dieser Lösung sind einerseits die Zentralisierung durch den Broker und der Aufwand für TCP-Kommunikation im Hinblick auf ressourcenbeschränkte Kleinstgeräte, wie z. B. Sensoren. Das zweite Protokoll, welches sich in der Entwicklergemeinde durchgesetzt hat, ist das Constrained Application Protocol (CoAP). Es ist ebenso wie MQTT ein Client-Server-Protokoll für M2M-Kommunikation. Arbeitet jedoch ausschließlich mit Client- und Server-Rollen, verzichtet auf den

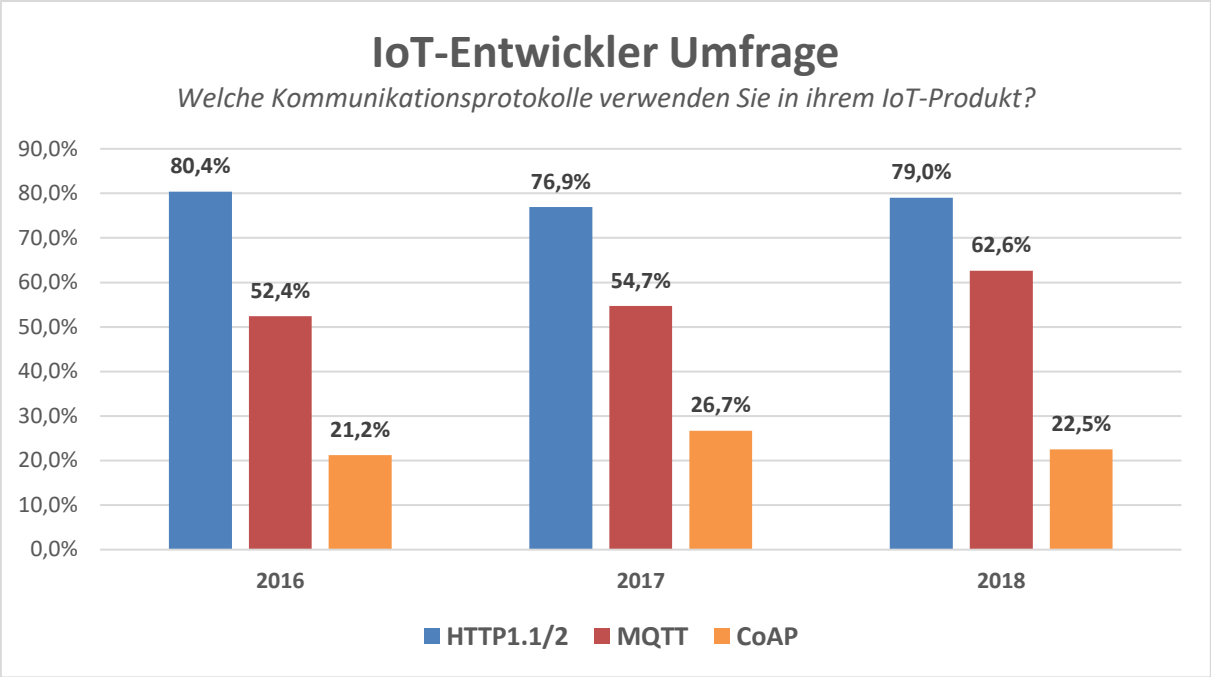


Abbildung 6: Trends in der IoT-Entwicklung - Verwendung der 3 wichtigsten Nachrichtenprotokolle in IoT-Produkten in den letzten Jahren. Das Diagramm wurde auf Grundlage der Daten aus Anlage I erstellt.

zentralen Broker und arbeitet als verbindungsloses Protokoll. Letzteres beschreibt die Kommunikation über das User Datagram Protocol (UDP), welches im Gegensatz zu TCP keine dedizierte Verbindung aufbaut und auch nicht den korrekten Empfang von Nachrichten sicherstellt. Diese Nachteile verringern jedoch den Kommunikationsaufwand, welches eine effizientere Kommunikation ermöglichen kann. Natürlich müssen die fehlenden Funktionen von UDP im Vergleich zu TCP von CoAP teilweise nachgebildet werden, um ein funktionsfähiges Protokoll zu ermöglichen. Der entscheidende Vorteil von CoAP ist die Dezentralisierung der Kommunikation. Ein Ausfall des Brokers bei MQTT hat den kompletten Systemausfall zur Folge. Weitere Informationen zu CoAP sind in Abschnitt 3.3.1 aufgeführt. Das in den vorangegangenen Projekten genutzte Devices Profile for Web Services (DPWS) findet in den Umfragen der letzten Jahre keine Erwähnung und hat demnach keine weitere Verbreitung außer als Nischenlösung erfahren. Microsoft nutzt DPWS als Teil der Technologie Windows Communication Foundation (WCF), welche Teil der Windows Betriebssysteme seit Windows Vista ist.

3. Grundlagen

Im Folgenden werden die Grundlagen für dieses Forschungsprojekt eingeführt. Relevante Entwicklungen aus den vorangegangenen BBSR-Projekten werden in Abschnitt 3.1 aufgeführt. Anschließend werden in den folgenden Abschnitten Protokolle und Technologien erklärt, die zur Umsetzung des Sicherheits-Framework benötigt werden.

3.1. Entwicklungen aus vorangegangenen BBSR-Projekten

Im Folgenden werden die Ergebnisse aus den vorangegangenen BBSR-Projekten „Webservices for Devices als Integrationsplattform für intelligente Dienste der Gebäudetechnik“ (Aktenzeichen: SF-10.08.18.7-11.4 / II 3-F20-11-004) und „Offene Schnittstellen im Smart Home unter Verwendung semantischer Plug&Play-Technologien“ (Aktenzeichen: II 3-F20-13-3-001/SWD-10.08.18.7-13.12) vorgestellt, die die Basis für dieses Forschungsprojekt bilden. Im weiteren Verlauf des Berichtes werden diese Projekte als BBSR WS4D und BBSR Plug&Play bezeichnet. An dieser Stelle sei erwähnt, dass die hier vorgestellten Vorgängerprojekte die Web-Technologie DPWS für die Kommunikation zwischen Geräten nutzen. Mittlerweile werden andere Protokolle, wie z. B. CoAP und MQTT, im IoT-Bereich genutzt, um eine ressourcenschonende Kommunikation zwischen Kleinstgeräten zu ermöglichen (siehe Abschnitt 2.3).

Im ersten Projekt, BBSR WS4D, wurde geprüft und bewiesen, dass sich die Web-Technologie „Devices Profile for Web Services“ (DPWS) basierend auf dem SOAP-Protokoll (Simple Object Access Protocol) für die Gerätekommunikation im Smart Home eignet. Weiterhin wurden Optimierungen präsentiert, um die Effizienz von DPWS zu steigern. So kann der Header einer DPWS-Nachricht durch Abbildung auf das Constrained Application Protocol (CoAP) erheblich komprimiert werden. Der Discovery-Mechanismus zum Auffinden anderer DPWS-fähiger Geräte skaliert schlecht für Netzwerke mit vielen Teilnehmern. Daher wurden Optimierungen erarbeitet, die eine Discovery mit geringer Netzwerkbelastung und hoher Zuverlässigkeit in großen Netzwerken erreicht.

Im Nachfolgeprojekt BBSR Plug&Play wurde die Integration und Orchestrierung neuer und alter (Legacy) Geräte in ein bestehendes Smart Home-Netzwerk untersucht. Es wurde ein Konzept zur dezentralen Konfiguration von Smart Home-Geräten erarbeitet, bei dem der Nutzer mittels eines Endgerätes wie beispielsweise Smartphone oder PC die Smart Home-Geräte untereinander rudimentär verknüpfen kann [29]. Somit können Automatisierungen im Smart Home erstellt und ausgeführt werden, ohne eine zentrale Verwaltungsinstanz zu benötigen, die oft in proprietären Lösungen verwendet, jedoch als Schwachstelle, dem sogenannten Single Point of Failure (SPoF), angesehen wird. Ein erfolgreicher Angriff auf einen solchen SPoF führt in diesem Zusammenhang zu dem Ausfall aller Smart Home-Funktionen, da die zentrale Verwaltungsinstanz ausfällt.

Das Fehlen eines Datenmodells in DPWS stellt eine Hürde für die herstellerunabhängige Kommunikation zwischen DPWS-Geräten dar. Gleiche Geräte unterschiedlicher Hersteller können verschiedene Daten für die Konfiguration benötigen. Um diese Hürde zu überwinden, wurde eine Datenbank im Internet eingerichtet, die den Austausch von Gerätedaten und die Etablierung von Standards für Konfigurationsdaten von spezifischen Gerätetypen ermöglicht.

Die Integration bereits existierender Geräte in das Smart Home, die kein DPWS unterstützen, wurde als besonders wichtiges Teilprojekt für eine zukünftige Umsetzung des Smart Home-Netzwerkes erachtet. Ein kompletter Umstieg auf eine neue Technologie ist mit höheren Kosten verbunden, da alle

bestehenden Installationen in diesem Fall vollständig modernisiert werden müssten. Eine kostengünstige Lösung ist dagegen eine schrittweise Modernisierung. Dabei werden nur einige Geräte ersetzt oder neue installiert. Die bestehenden Geräte sollen weiterhin genutzt und in die neue Installation integriert werden. Um dieses Ziel zu erreichen, müssen alte Geräte mittels eines Gateways mit DPWS-Geräten kommunizieren können. Dies wurde am Beispiel des BACnet-Protokolls gezeigt, da bereits mehrere Gateway-Umsetzungen von BACnet auf andere Protokolle wie z. B. KNX, LON, ZigBee existieren.

Neben der Einbindung von Legacy-Geräten ins Smart Home, wurde die Aufrüstung alter Stoff- und Energiemengenzähler zu Smart Metern als eine kostengünstige Alternative zum Einbau eines teuren Smart Meters untersucht, um den Rollout in Deutschland zu beschleunigen. Im Verlauf des Projektes wurde ein Prototyp basierend auf einer Kamera und einem neuronalen Netzwerk entwickelt.

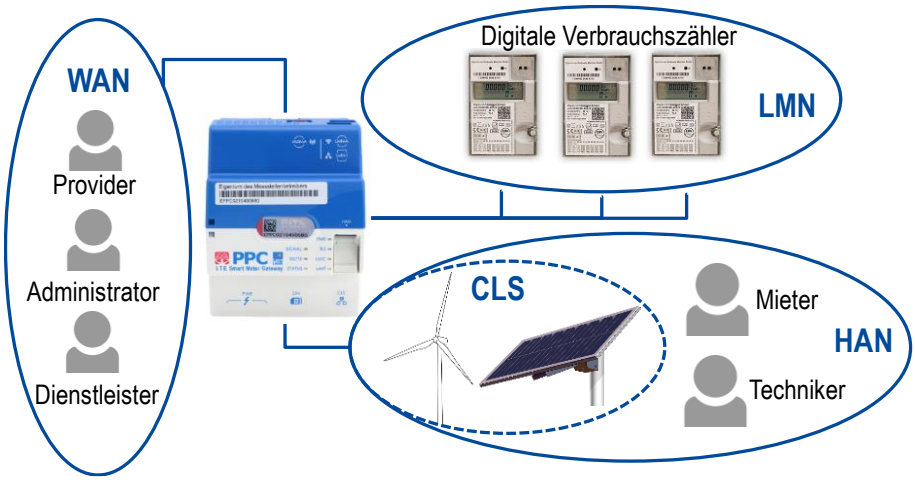
3.2. Smart Meter Gateway

Das Smart Meter Gateway stellt die Kommunikationseinheit eines intelligenten Messsystems dar. In Deutschland werden unter dem Begriff Smart Meter zwei Ausbaustufen eines modernen Verbrauchszählersystems zusammengefasst. Als moderne Messeinrichtung (mME) wird ein digitaler Verbrauchszähler ohne Kommunikationseinheit bezeichnet, der gespeicherte Werte tagesgenau ausgeben kann. Die Möglichkeit zur Anbindung an ein Smart Meter Gateway muss durch Schnittstellen gewährleistet werden. Eine Kombination aus moderner Messeinrichtung und Smart Meter Gateway wird intelligentes Messsystem (iMSys) genannt. Dieses System ermöglicht u. a. eine automatische Datenübertragung der Verbrauchswerte an den Messstellenbetreiber und die externe Steuerung von Controllable Local Systems (CLS), wie z. B. Blockheizkraftwerke, Photovoltaik- und Windkraftanlagen, durch den Energieanbieter. [30]

Im Folgenden wird das Smart Meter Gateway detailliert beschrieben und auf die Anforderungen eingegangen, die das Bundesamt für Sicherheit in der Informationstechnik (BSI) an diese Kommunikationseinheit eines Smart Meter-Systems stellt.

3.2.1. Schnittstellen

Das Smart Meter Gateway besitzt Schnittstellen zu drei (vier) unterschiedlichen Netzwerken. Über das LMN kann eine Verbindung zu den Verbrauchszählern, z. B. Strom und Gas, aufgebaut werden. Externe Verbindungen zum Energieprovider, dem SMGW-Administrator und die Bereitstellung von Mehrwertdiensten wird über den WAN-Anschluss des SMGW ermöglicht. SMGWs werden verschiedene Kommunikationstechnologien, wie z. B. Ethernet, Mobilfunk (LTE) oder Breitband-Powerline (BPL), anbieten, um eine Verbindung mit externen Gegenstellen zu ermöglichen. Über die HAN-Schnittstelle kann einerseits der Mieter/Eigentümer die Verbrauchswerte vom SMGW abrufen und der Techniker vor Ort Einstellungen am SMGW vornehmen. Andererseits wird entweder über die HAN-Schnittstelle, oder je nach Hersteller über eine vierte Schnittstelle, das Netzwerk für CLS eingebunden. Über diese Verbindung können regelbare Verbraucher und Erzeuger (z. B. Blockheizkraftwerk und Photovoltaikanlage) vom Energieprovider aus der Ferne gesteuert werden, um beispielsweise dynamische Tarife zu ermöglichen oder Netzschwankungen auszugleichen.



WAN – Wide Area Network
LMN – Local Metrological Network
HAN – Home Area Network
CLS – Controllable Local Systems

Abbildung 7: Schnittstellen des Smart Meter Gateways nach [31], [32]

3.2.2. BSI Schutzprofil und Technische Richtlinien

Das vom BSI herausgegebene „Schutzprofil für die Kommunikationseinheit eines intelligenten Messsystems für Stoff- und Energiemengen“ (SMGW-PP) definiert Sicherheitsvorgaben für das Gateway eines Smart Metering-Systems. Das Schutzprofil wurde auf Grundlage der Common Criteria/ Norm ISO/IEC 15408, einem weltweiten Standard zur Bewertung von Datensicherheit, erarbeitet. Allgemeines Ziel der Common Criteria ist es, Zertifizierungen zu ermöglichen, die weltweit anerkannt werden. Das Schutzprofil richtet sich an Entwickler und informiert diese über die sicherheitsrelevanten Anforderungen an ein solches Gerät. Dabei werden mögliche Bedrohungsszenarien beschrieben, denen das Gateway ausgesetzt sein kann. Das Schutzprofil wird während der Evaluierung und Zertifizierung eines, von einem Hersteller eingereichten, Smart Meter Gateway vom BSI genutzt. Auf diese Weise soll geprüft werden, ob alle sicherheitsrelevanten Anforderungen erfüllt wurden. Neben dem Schutzprofil für das Smart Meter Gateway (PP-0073), existiert noch ein gesondertes Schutzprofil für das interne Sicherheitsmodul des Smart Meter Gateway (PP-0077), auf das in PP-0073 verwiesen wird. PP-0077 befasst sich ausschließlich mit den kryptografischen Anforderungen an das Sicherheitsmodul, u. a. zur Ver- und Entschlüsselung der Kommunikation, dem Signieren von Daten und zur Gewährleistung eines sicheren Speicherplatzes für Daten, Zertifikate und Schlüssel. Die vom Gateway zu unterstützenden Funktionen und deren Einschränkungen werden in der Technischen Richtlinie BSI TR-03109 spezifiziert. Diese Richtlinie enthält eine Vielzahl untergeordneter Bestimmungen, die die Komponenten des Smart Meter Gateway einzeln spezifizieren. In Abbildung 8 werden diese Teilspezifikationen übersichtlich dargestellt.

Das Schutzprofil erfordert drei physikalisch voneinander getrennte Schnittstellen für die Kommunikation mit dem Smart Meter-Netzwerk (LMN: Local Metrological Network), Home Area Network (HAN) und Wide Area Network (WAN). Während das WAN zur Anbindung an das Internet genutzt wird, um einen sicheren externen Zugriff über RESTful-Dienste (siehe Kapitel 3.2) auf das Gateway und damit verbundene Geräte (HAN) zu ermöglichen, dient die HAN-Schnittstelle zur Anbindung von Haushaltsgeräten und Controllable Local Systems (CLS), wie z. B. EEG-Anlagen (Erneuerbare-Energien-Gesetz) und Blockheizkraftwerke. Für jede der Schnittstellen werden unterschiedliche Anforderungen definiert. Da das Gateway als Vermittler zwischen diesen drei

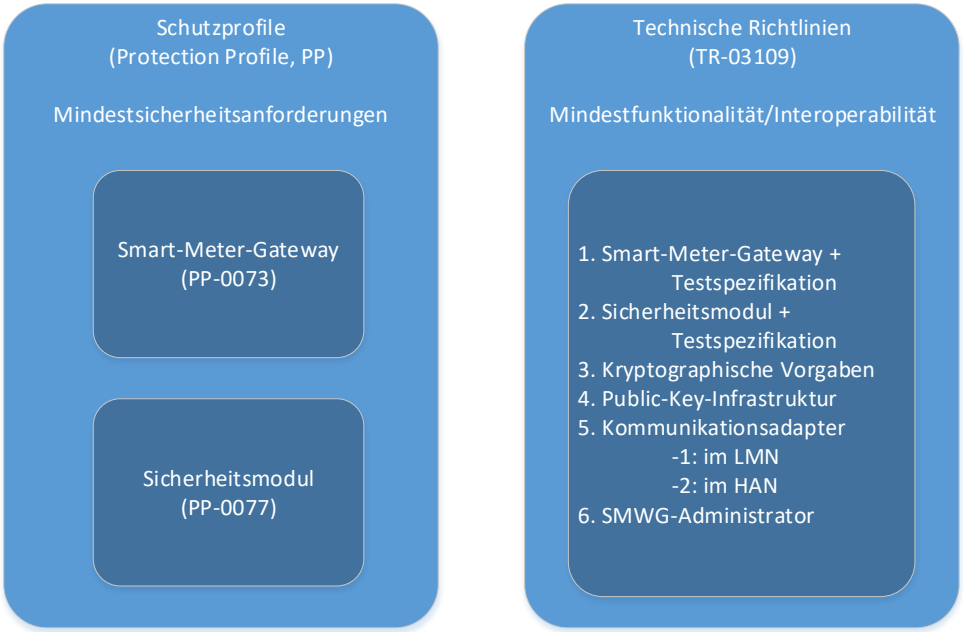


Abbildung 8: Übersicht der BSI Dokumente zum Smart Meter Gateway nach [31]

Netzwerken auftritt, erfordert das Schutzprofil die Integration von Firewall-Mechanismen. Jede Kommunikation über diese Schnittstellen erfolgt erst nach einer gegenseitigen Authentifizierung mit dem Kommunikationspartner und muss verschlüsselt und integritätsgesichert sein. Eine besonders starke Restriktion, die auch bereits in Fachartikeln [33] kritisiert wurde, ist die Bedingung, dass Verbindungen über die WAN-Schnittstelle nur von innen nach außen aufgebaut werden dürfen. Dies erschwert die Interaktion mit dem Gateway über einen Fernzugriff erheblich. Es gibt jedoch die Möglichkeit, den Verbindungsaufbau zu einer sicheren Gegenstelle im Internet von außen zu initiieren. Einzig der Gateway Administrator, i. d. R. der Energieversorgungsanbieter, kann eine so genannte Wake-Up-Nachricht an das Gateway schicken und den Verbindungsaufbau zu einer vordefinierten Adresse auslösen. Ein externer Zugriff muss folglich immer über den Gateway Administrator erfolgen. Dieses Szenario ist allerdings nicht detailliert im Schutzprofil spezifiziert. Weiterhin wird im BSI-Dokument zwischen unterschiedlichen Einsatzgebieten, wie z. B. Einfamilienhaus und Mehrfamilienhaus, und damit einhergehenden variierenden Anforderungen unterschieden.

Die entworfenen Bedrohungsszenarien werden anhand des Aufenthaltsortes und Zugriffs des Angreifers kategorisiert. Ein mögliches Szenario ist, dass sich der Angreifer vor Ort befindet, physischen Zugriff auf das Gateway besitzt und die Kommunikation kompromittiert, verbraucherrelevante Daten ausliest oder die Systemuhr manipuliert. In einem anderen Szenario kommt es zu einem externen Angriff über die WAN-Schnittstelle des Gateways, um u. a. die Kontrolle über das Gateway zu erlangen und Geräte, die sich im HAN-Netzwerk befinden, fernzusteuern oder eine kompromittierte Firmware auf diese Geräte aufzuspielen. Um diesen und anderen Bedrohungen zu begegnen, werden Sicherheitsziele definiert, die bei einer Implementierung des Smart Meter Gateways umgesetzt werden müssen. Weiterhin ist der erlaubte Informationsfluss zwischen den verschiedenen Schnittstellen des Smart Meter Gateways klar definiert worden. Tabelle 1 veranschaulicht die möglichen Informationsflüsse.

Quelle \ Ziel	WAN	LMN	HAN
WAN	Erlaubt, SMGW nimmt nicht an Komm. teil	Keine Verbindung erlaubt	Keine Verbindung erlaubt
LMN	Keine Verbindung erlaubt	Erlaubt, jedoch keine Komm. Zw. LMN-Geräten erwartet	Keine Verbindung erlaubt
HAN	Verschl. Verbindung zu vordef. Endpunkten	Keine Verbindung erlaubt	Erlaubt, SMGW nimmt nicht an Komm. teil

Tabelle 1: Informationsflüsse über Schnittstellen des Smart Meter Gateway nach [34]

3.3. Protokolle

Dieser Abschnitt führt die Protokolle ein, die im Sicherheits-Framework verwendet werden. Es werden sowohl bereits etablierte Standards beschrieben, als auch neuartige Protokolle, die sich noch Standardisierungsprozess befinden.

3.3.1. Constrained Application Protocol (CoAP)

Das Constrained Application Protocol (CoAP) [35] ist ein Vertreter der Representational State Transfer-Protokolle (RESTful), welches sich für verteilte eingebettete Systeme eignet. Protokolle dieser Familie weisen eine Client-Server-Struktur auf. Clients bauen die Verbindung zu Servern auf. Eine Verbindungsinitiierung seitens des Servers ist nicht vorgesehen. Des Weiteren sind Verbindungen zwischen Servern und Clients zustandslos. Dies bedeutet, dass Serveranfragen eines Clients unabhängig von der vorangegangenen Kommunikation verarbeitet werden. Somit sind einzelne Verbindungen voneinander unabhängig. Um die Kommunikationshistorie zu berücksichtigen, ist es möglich, dass jeder Client Caching betreibt. Außerdem ermöglicht eine RESTful-Architektur die unabhängige Entwicklung von einzelnen Diensten, die zusammen eine große Architektur bilden. Eine weitere Eigenschaft von RESTful-Architekturen ist es, Systeme mit mehreren Schichten zu implementieren. Dabei kommuniziert ein Server auf eine Anfrage mit anderen Servern, um die ursprüngliche Client-Anfrage zu beantworten. Außerdem können Clients einen ausführbaren Code von Servern abrufen (Code-on-Demand). Abbildung 9 visualisiert diese Eigenschaften einer RESTful-Architektur.

Um Diensten in einer RESTful-Architektur Adressen zuzuordnen, kommt der Uniform Resource Locator (URL) zum Einsatz. Ein Dienst kann mehrere Ressourcen anbieten. Zur Adressierung der einzelnen Ressourcen wird der Uniform Ressource Identifier (URI) verwendet. Ein URL fügt den Zugriffsmechanismus auf die URI hinzu.

CoAP ähnelt dem Hyper Text Transfer Protocol (HTTP) [36] sehr stark. Es weist allerdings einen kleineren Header als HTTP auf, da der CoAP-Header binär codiert ist. Außerdem basiert CoAP, wie in Abbildung 10 dargestellt, auf UDP [37] anstelle von TCP [38] im Fall von HTTP. Möchte ein HTTP-Client eine Anfrage an einen HTTP-Server senden, so muss zunächst eine Verbindung über TCP aufgebaut

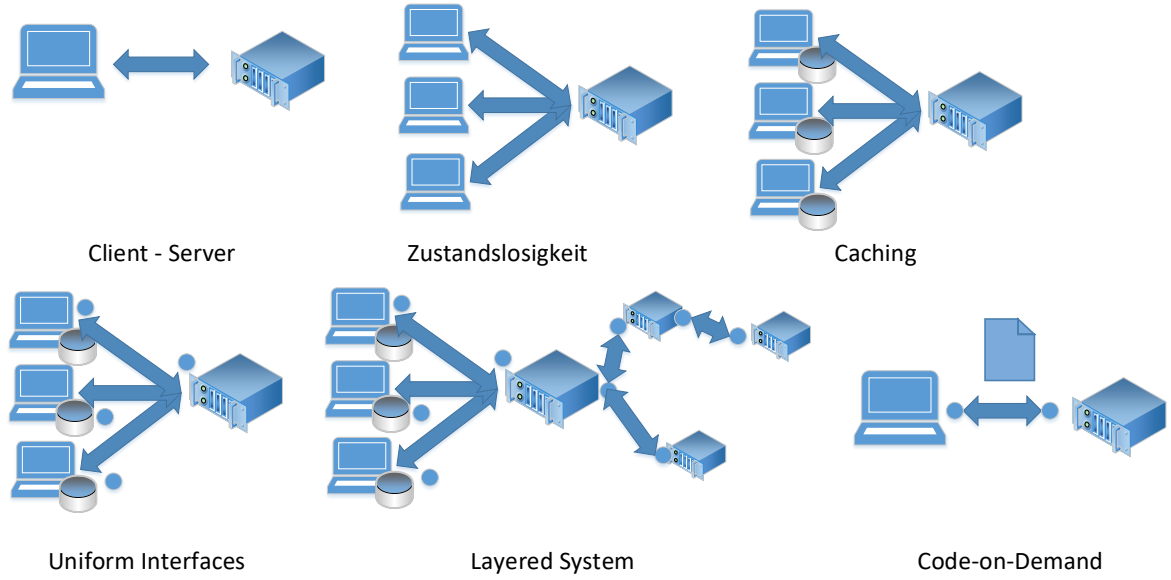


Abbildung 9: Eigenschaften einer RESTful-Architektur

werden. Der TCP-Handshake benötigt drei Nachrichten, bevor die Anwendung über HTTP Nachrichten austauschen kann. Jede TCP-Verbindung wird durch einen weiteren Drei-Wege-Handschlag beendet. Bei UDP hingegen wird keine Verbindung aufgebaut. Die Daten werden sofort über UDP zum Server gesandt. Somit eignet sich CoAP besonders für Kleinstgeräte, da im Vergleich mit HTTP weniger und kürzere Nachrichten versendet werden. Insbesondere bei drahtloser Kommunikation lässt sich durch die Komprimierung gesendeter Daten eine größere Energiemenge einsparen. Von dieser Optimierung profitieren batteriebetriebene Smart Home-Geräte.

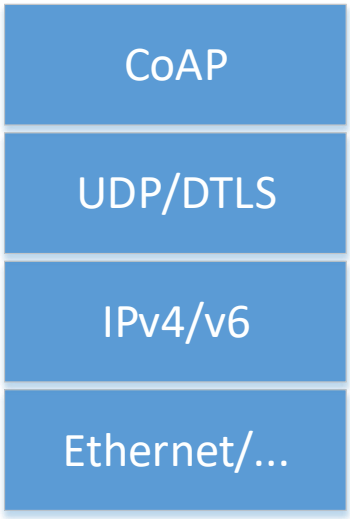


Abbildung 10: Protokoll-Stack von CoAP

Jeder CoAP-Server bietet Ressourcen für Clients an, die über CRUD-Operationen auf die Ressourcen zugreifen. CRUD-Operationen sind **C**reate, **R**ead, **U**pdate und **D**elete. Sie bezeichnen das Erstellen, Lesen, Ändern und Löschen einer Ressource. Umgesetzt werden diese Operationen, wie in Tabelle 2 und Tabelle 3 dargestellt, durch GET, PUT, POST und DELETE. Diese Operationen werden ebenfalls von HTTP unterstützt. Aufgrund der starken Ähnlichkeit von CoAP und HTTP lassen sich Proxys implementieren, die zwischen beiden Protokollen übersetzen. Dabei kann beispielsweise CoAP für

ressourcen-schwache Geräte im Smart Home und HTTP für die Kommunikation mit Web-Diensten genutzt werden.

CRUD-Operation	CoAP (REST)-Operation
Create	PUT oder POST
Read	GET
Update	PUT
Delete	DELETE

Tabelle 2: Gegenüberstellung von CRUD und CoAP-Operationen

CoAP definiert eine Ressource (.well-known/core), die jeder Server anbieten muss. Nach Abfrage dieser Ressource erhält der Client eine Übersicht über alle Ressourcen des Servers. Jede Ressource liefert einen Content Type, der Informationen über die Payload darstellt.

CoAP-Operation	Bedeutung
GET	Lesen von Daten von einem URI
PUT	Neue Ressource auf dem Server anlegen; bestehende Ressource ändern
POST	Daten als Payload an Server senden; Ressourcen auf dem Server anlegen
DELETE	Löschen der Ressource auf dem Server

Tabelle 3: Bedeutung von CoAP-Operationen

Eine weitere Eigenschaft von CoAP ist der asynchrone Eventing-Mechanismus. Dabei abonnieren Clients eine Ressource auf einem Server. Der Server speichert alle Abonnenten und sendet auf eine Änderung der Ressource eine Benachrichtigung.

Funktionalitäten von Smart Home-Geräten lassen sich durch Ressourcen auf einem CoAP-Server repräsentieren. Jedoch sind für dieselbe Funktion mehrere gültige Repräsentationen durch Ressourcen möglich. Soll die Beleuchtung eines Raumes gesteuert werden, sind beispielsweise folgende Ressourcen denkbar:

```
/Beleuchtung/Wohnzimmer/setzen?status=an

/Eigenheim/Zimmer/Wohnzimmer/Licht/AN
```

Daraus ergeben sich die nachstehenden Probleme:

Die Bezeichnung des Zimmers kann unterschiedlich aufgebaut sein. Die einzelnen Räume lassen sich durch beliebige Pfade, die hierarchisch gegliedert sein können, beschreiben. Ein weiteres Problem ist die nicht einheitliche Wahl der Operation, die auf die Ressource angewendet wird. Das erste Beispiel basiert auf einer GET-Anfrage, die die nötigen Parameter in der URI überträgt. In diesem Fall lautet die Query `status=an`. Der Status kann außerdem den Zustand `aus` annehmen. Andere Zustände wie `wahr`, `falsch`, `on` oder `off` sind möglich. Das zweite Beispiel kann entweder durch eine GET-, PUT- oder POST-Anfrage angesprochen werden. Die Semantik der URIs ist für den Menschen nicht zwingend gegeben. Entwickler eines Clients wissen nicht zwingend, welche Ressourcen wie angesprochen werden müssen, um ein bestimmtes Verhalten auszulösen. Hier ist eine gute Dokumentation der

Hersteller nötig. Die sogenannte RESTful-API sollte öffentlich einsehbar sein, damit Geräte verschiedener Hersteller miteinander kompatibel sind.

Standardkonforme Sicherheit von CoAP: Datagram Transport Layer Security (DTLS)

CoAP-Nachrichten basieren auf UDP und besitzen daher keine Sicherheitsfunktionen wie Verschlüsselung und Integrität. Der CoAP-Standard definiert für den Aspekt der Kommunikationssicherheit das Protokoll Datagram Transport Layer Security (DTLS) [39]. DTLS verschlüsselt die Daten, weshalb die Vertraulichkeit der Informationen sichergestellt ist. Außerdem können Daten, die mit DTLS gesichert werden, auf Veränderung während der Übertragung geprüft werden. Die Nachrichtenintegrität wird somit gewahrt. DTLS stellt außerdem die Nachrichtenauthentizität sicher. Der Absender einer Nachricht lässt sich durch die Verwendung von Zertifikaten identifizieren. Ein Zertifikat beinhaltet einen öffentlichen Schlüssel und Metainformationen zum Gerät. Server- und Client-Zertifikate stellen eine bidirektionale Authentizität sicher.

3.3.2. Concise Binary Object Representation (CBOR)

CBOR [40] ist ein von der Universität Bremen mitentwickelter Internet Engineering Task Force (IETF) Standard und stellt ein binäres Datenformat dar, um eine kompakte Nachrichtengröße zu ermöglichen, und benötigt nur vergleichsweise wenige Zeilen Programm-Code zur Erstellung und Auswertung solcher Nachrichten. Es basiert auf dem Datenaustauschformat JavaScript Object Notation (JSON) [41] und kann als binäre Erweiterung angesehen werden. Während JSON ein menschenlesbares Datenformat verwendet, kann der Inhalt und die Struktur von CBOR-Nachrichten nicht ohne Verarbeitung durch ein entsprechendes Computerprogramm von Menschen gelesen werden. Dieser Kompromiss ermöglicht die kompaktere Nachrichtengröße und eignet sich somit besonders für Anwendungen im Internet of Things (IoT).

3.3.3. CBOR Object Signing and Encryption (COSE)

Der IETF Standard COSE [42] basiert auf dem Datenformat CBOR und realisiert rudimentäre Sicherheitsfunktionen wie die Erstellung und Prüfung von Signaturen und Nachrichtenauthentifizierungscodes (engl. Message Authentication Codes, MAC), sowie das Ver- und Entschlüsseln von CBOR-Nachrichten. Es kann als Derivat der JSON Object Signing and Encryption (JOSE) Spezifikation [43] angesehen werden, welche ähnliche Sicherheitsfunktionen für JSON realisiert.

3.3.4. Object Security for Constrained RESTful Environments (OSCORE)

OSCORE ist ein IETF Dokument mit dem Status „Draft“, befindet sich jedoch bereits im Standards Track. Somit kann das Sicherheitsprotokoll in Zukunft zum anerkannten IETF Standard werden. Da OSCORE das Constraint Application Protocol (CoAP, siehe 1. Zwischenbericht) zur Nachrichtenübermittlung nutzt, wird das Protokoll im Folgenden ausgehend vom CoAP-Standard beschrieben.

Der CoAP-Standard empfiehlt mit Datagram Transport Layer Security (DTLS, siehe 1. Zwischenbericht) [44] die CoAP-Kommunikation zu sichern. Das Dokument beschreibt jedoch auch die Nachteile der kombinierten Verwendung von CoAP mit DTLS. Das Sicherheitsprotokoll verhindert z. B. die Unterstützung für Gruppenkommunikation (ein Befehl wird von mehreren Geräten empfangen und ausgeführt, z. B. alle Lampen im Wohnzimmer mit einem Befehl ausschalten) und enthält potenzielle

Sicherheitslücken bei der Verwendung von Proxys (keine Ende-zu-Ende-Verschlüsselung, siehe Cloud-Kommunikation in Abschnitt 4.5.3) [35]. Um diese Einschränkungen zu überwinden, wurde das Sicherheitsprotokoll OSCORE definiert. Es führt das sogenannte Object-Security-Konzept ein. Dieses kann als Applikationsschichtsicherheit bezeichnet werden und ermöglicht eine flexible Einstellung von Sicherheitsparametern für verschiedene Nachrichtentypen. Folglich kann eine Nachricht unverschlüsselt, teilweise oder vollständig verschlüsselt werden. Weiterhin können MAC und Signaturen geprüft werden. Um diese Funktionen mit anerkannten Sicherheitsmechanismen zu realisieren, verwendet OSCORE das Datenformat CBOR und die Sicherheitsspezifikation COSE.

3.3.5. OAuth 2.0 & OpenID Connect

Aktuell existieren keine einheitlichen Protokolle zur Autorisierung von Zugriffen auf IoT/Smart Home-Geräte. Autorisierung beschreibt die Zugriffsberechtigung von Teilnehmern auf Ressourcen. Weitverbreitete Web-Standards wie OAuth 2.0 [45] und OpenID Connect [46] versprechen jedoch eine Lösung, die sich auf Smart Home-Geräte übertragen lässt.

OAuth 2.0 ist die aktuelle Version eines Autorisierungsverfahrens, der Single-Sign-On-Dienste ermöglicht. Es werden folgende Rollen von Teilnehmern unterschieden:

Rolle	Bedeutung
Resource Owner (RO)	Eigentümer einer Ressource
Resource Server (RS)	Rechner, an dem sich die Ressource befindet
ID Provider (IDP)	Server zur Verwaltung der Nutzerkonten
Client	Web Browser oder Smartphone-Applikation
Relying Party (RP)	Drittanbieter

Tabelle 4: Rollen von Teilnehmern in OAuth 2.0

Durch OAuth 2.0 verwendet ein Nutzer dieselben Login-Daten, um sich bei mehreren Dienstanbietern einzuloggen. Möchte der Nutzer sich bei einer Drittanbieter-Webseite anmelden, steht die Verwendung eines schon existierenden Logins zur Verfügung. Unternehmen wie Facebook, Twitter und Google bieten die Möglichkeit, Kundenkonten für Drittanbieterwebseiten zu nutzen. Dadurch können Drittanbieter-Dienste zum Zugriff auf Nutzerdaten, wie z. B. Fotoalben, Facebook-Timelines oder Kontaktinformationen, autorisiert werden. Außerdem lässt sich durch OAuth 2.0 ein autorisierter Zugriff auf APIs (Application Programming Interface, dt.: Programmierschnittstelle) realisieren. Obwohl OAuth 2.0 zur Autorisierung dient, finden sich jedoch eine Vielzahl von Anwendungen, die dieses Verfahren nutzen, um Nutzer zu authentifizieren. Dabei erhält die RP den Zugriff auf einen Nutzer-Identifier, der beim IDP hinterlegt ist.

Da OAuth 2.0 ein weitverbreiteter Internetstandard ist, wurden umfangreiche Sicherheitsanalysen durchgeführt. Darunter ist auch eine formale Sicherheitsanalyse [47], die seine Sicherheit bestätigt. Dies gilt unter der Maßgabe, dass einige Implementierungsdetails in OAuth 2.0 einfließen, die Sicherheitslücken beheben.

OpenID Connect erweitert OAuth 2.0 um eine echte Authentifizierungsfunktion. Dabei können sich Clients bei einem RP unter Nutzung eines IDP authentifizieren und Nutzerprofildaten übermitteln.

3.3.6. Authentication and Authorization for Constrained Environments (ACE)

ACE [48] ist ebenfalls ein IETF Draft im Standards Track und stellt ein leichtgewichtiges Framework für Authentifizierung und Autorisierung von IoT-Geräten dar. Es basiert auf dem häufig verwendeten OAuth 2.0 Framework und nutzt CBOR, COSE und CoAP/OSCORE, um Authentifizierungs- und Autorisierungsfunktionen zwischen IoT-Geräte zu ermöglichen. Ähnlich wie bei OAuth 2.0 werden verschiedene Rollen wie Authorization Server (AS, entspricht IDP von OAuth 2.0), Ressource Server (RS) und Client definiert. Es kann genutzt werden, um Zugriffe auf Geräte zu gewähren oder abzulehnen. Weiterhin ermöglicht ACE die sichere Verteilung von Sitzungsschlüsseln für eine Verbindung zwischen zwei Geräten.

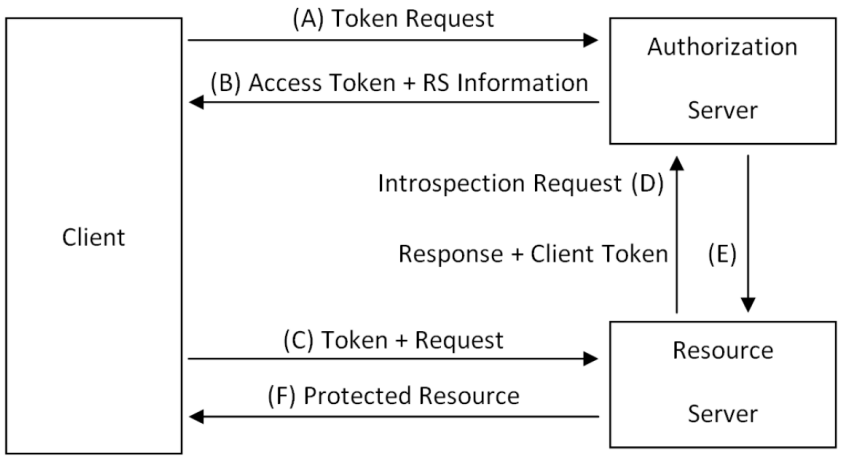


Abbildung 11: Allgemeiner Autorisierungsablauf unter ACE

In Abbildung 11 ist der allgemeine Protokollablauf von ACE dargestellt. Möchte ein Client die Autorisierung für eine Resource erhalten, so stellt er einen Token Request an den Authorization Server (A). Ist der Client berechtigt die Resource zu nutzen, wird ein Access Token (entspricht einer Zugriffsgenehmigung) mit dazugehörigen Ressourcen Informationen ausgestellt (B). Daraufhin sendet der Client den Request samt Token an die geschützte Resource (C). Da der Resource Server nicht sicher sein kann, ob der Token gültig ist, wird in der Introspection eine Prüfung des Tokens durchgeführt (D). Die Bestätigung (E) bestätigt dem Resource Server, dass der Client berechtigt ist die Resource zu nutzen. Abschließend wird mit der Nachricht (F) die angeforderte Ressource übermittelt. Der eben beschriebene Ablauf wird in Abbildung 12 mit Beispielgeräten dargestellt. Außerdem existiert eine Protokollvariante bei der die Introspection des Tokens (C) durch den Resource Server selbst durchgeführt wird anstelle eine Verbindung zum Authorization Server aufzubauen. Wie in Abbildung 13 dargestellt, entfallen dabei die Nachrichten (D) und (E). Die Varianten verwenden unterschiedliche Token. Während bei einem Kommunikationsfluss mit Introspection nur ein Referenzschlüssel ähnlich einer Garderobenmarke zwischen den teilnehmenden Geräten ausgetauscht wird (Informationen über Zugriffsberechtigung verbleiben auf dem Authorization Server), benötigt die Variante, bei der der Resource Server selbstständig die Berechtigung prüfen muss, einen so genannten CBOR Web Token. Dieser ist eine komplexe Datenstruktur, teilweise verschlüsselt und signiert, die es dem Resource Server erst ermöglicht, die Zugriffsberechtigung zu überprüfen. Die beiden Kommunikationsvarianten sind daher ein Kompromiss zwischen geringem Verarbeitungsaufwand für den Resource Server und verringertem Kommunikationsaufwand (keine RS-AS-Kommunikation) mit größerem Token. Weiterhin besitzen die Token einen individuell definierbaren Gültigkeitszeitraum.

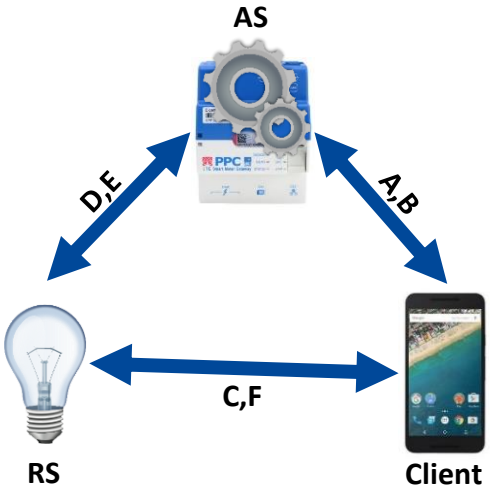


Abbildung 12: ACE-Autorisierungsablauf mit Introspection und Rechenlast bei AS (SMGW)

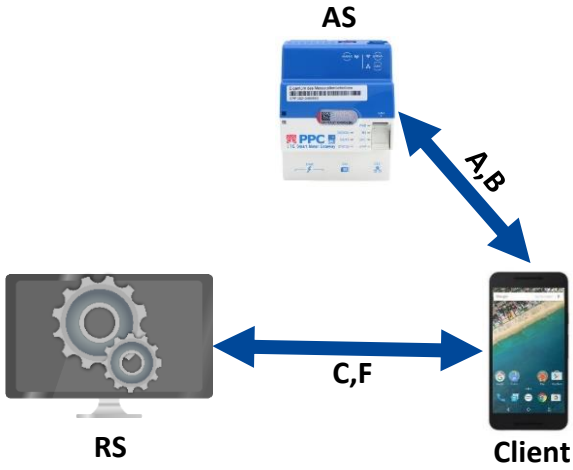


Abbildung 13: ACE-Autorisierungsablauf ohne Introspection und Rechenlast bei RS (TV)

3.3.7. Lightweight M2M (LwM2M)

Die Open Mobile Alliance arbeitet an einheitlichen Schnittstellen wie dem Lightweight Machine to Machine (LWM2M) Projekt. LWM2M basiert auf CoAP und erweitert es durch Datenmodelle. Die einzelnen Datenmodelle stehen öffentlich zur Verfügung. Weiterhin enthält es eine Managementschicht zur benutzerfreundlichen Verwaltung von einer Vielzahl an Geräten. Dies ermöglicht die Automatisierung von Prozessen, wie z. B. Firmware-Updates aller verwalteten Geräte.

3.4. Weiterführende Informationen

Dieser Abschnitt enthält weitere Grundlagen für Erweiterungen des Sicherheits-Framework und die Implementierung der Prototypen.

3.4.1. Videostreaming über CoAP

Mit CoAP lassen sich Anwendungsfälle wie das Lesen von Sensordaten oder das Steuern von Geräten realisieren. Neben einzelnen kleinen Nachrichten können durch den CoAP Block Wise Transfer auch größere Datenblöcke versendet werden. Der MPEG-DASH Standard zum Videodatenstreaming besteht daraus, dass ein Client per Request Videosegmente von einigen Sekunden Länge vom Server abfragt. Je nach verfügbarer Verbindungsgeschwindigkeit und -Qualität können Segmente unterschiedlicher Auflösung und damit unterschiedlicher Datenmenge abgerufen werden. MPEG-DASH basiert auf HTTP, sodass eine Adaption auf CoAP möglich ist [49]. Da HTTP auf TCP aufsetzt, greifen die TCP inhärenten Mechanismen zur Staukontrolle. CoAP hingegen verwendet eine Staukontrolle auf Anwendungsschicht, da UDP solch eine Funktionalität nicht besitzt. In [49] finden sich Untersuchungen zum Videodatenstreaming über CoAP nach dem MPEG-Dash Standard, die zeigen, dass CoAP besonders für den Einsatz von verlustbehafteten Übertragungen geeignet ist. Im Smart Home Anwendungsfall wäre dies eine WLAN Übertragung mit höheren Frame-Loss Raten. In der Literatur findet sich ein weiteres Verfahren, dass das Streaming von Sensordaten nach dem Blockwise Transfer evaluiert [50]. Ein grundsätzlich anderer Ansatz zum Streaming über CoAP macht sich den Publish-Subscribe-Mechanismus, auch Observe-Beziehung genannt, zunutze. Dabei wird der Stream durch

einzelne Benachrichtigungen (Notifications) an alle Abonnenten ausgesandt [51]. Die einzelnen Notifications können optional vom Client durch ein Acknowlegement quittiert werden um eine Neuübertragung nach einem Timeout zu veranlassen. Hierbei muss auf geeignete Zeitspannen für das Timeout geachtet werden, da insbesondere beim Live-Datenstreaming, wie bei einer Tür-Gegensprechanlage, hohe Anforderungen an die Übertragungslatenz gestellt werden und eine Neuübertragung nach wenigen Millisekunden obsolet ist. Die Unterschiede der Kommunikationsvarianten werden in Abbildung 14 und Abbildung 15 dargestellt. Der binär codierte Header von CoAP stellt bei allen Verfahren nur einen geringen Datenoverhead dar. Kombiniert mit leistungsfähigen Videokompressionsverfahren wie H.264 [52] und H.265 [53] lassen sich, wie in Abbildung 16 beispielhaft dargestellt, Video-Livestreams mit hoher Auflösung bei geringer Datenrate übertragen.

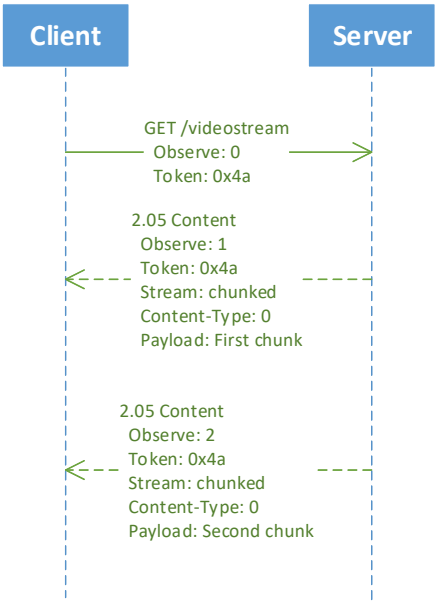


Abbildung 14: Abrufen von Meta-Informationen eines Streams nach [78]

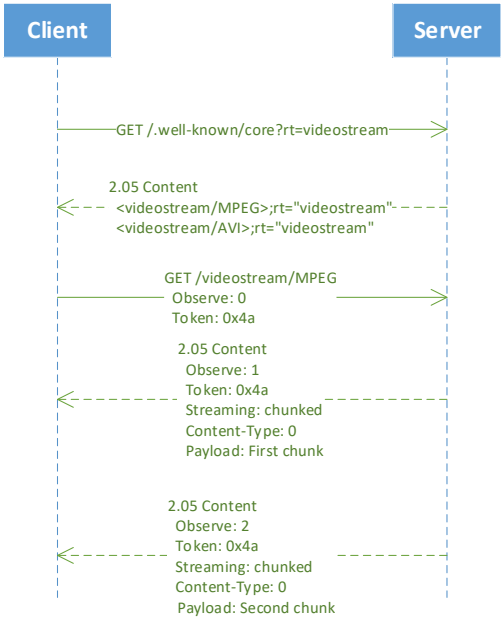


Abbildung 15: Observe-Mechanismus zum Videostreaming nach [78]

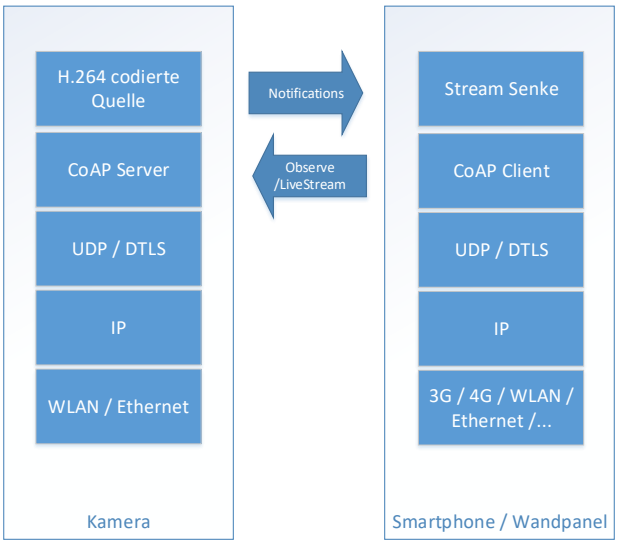


Abbildung 16: Beispiel Videostreaming zu Nutzergerät durch Observe-Mechanismus

3.4.2. Online-Ausweisfunktion des neuen elektronischen Personalausweises

Am 1. November 2010 wurde der neue elektronische Personalausweis (nPA) in Deutschland eingeführt. Der eingebettete Chip mit kontaktloser Kommunikationsmöglichkeit über RFID (Radio-Frequency Identification, dt.: Identifizierung mithilfe elektromagnetischer Wellen) ermöglicht eine Online-Ausweisfunktion, das Signieren digitaler Dokumente und enthält Daten für hoheitliche Funktionen, wie die digitale Version des Lichtbildes und optional Fingerabdrücke des Besitzers. Die Möglichkeit, sich bei Dienstanbietern elektronisch auszuweisen, bietet den Vorteil einer sicheren und schnellen Authentifizierung des Benutzers ohne menschliche Kontrollinstanzen. Um Dokumente und Nachrichten mit dem Personalausweis signieren zu können, muss ein Zertifikat von einer von der Bundesregierung autorisierten Instanz käuflich erworben werden. Dieses Zertifikat mit zeitlich beschränkter Gültigkeit wird in dem Chip des Personalausweises hinterlegt und kann zum elektronischen Unterschreiben von Dokumenten mit einer qualifizierten elektronischen Signatur (QES) genutzt werden. Das dazu benötigte Kartenlesegerät muss die Anforderungen der Technischen Richtlinie BSI TR-03119 erfüllen und wird ebenfalls für die Online-Ausweisfunktion benötigt. Die hoheitlichen Informationen auf dem Chip können nur durch Behörden, wie z. B. Polizei und Zoll, ausgelesen werden.

Um die Funktionsweise der Online-Ausweisfunktion anschaulich zu erklären, wird im Folgenden der Ablauf zum Zugriff auf einen Service eines Dienstanbieters mithilfe des neuen Personalausweises mithilfe der Kommunikationspfade in Abbildung 17 beschrieben.

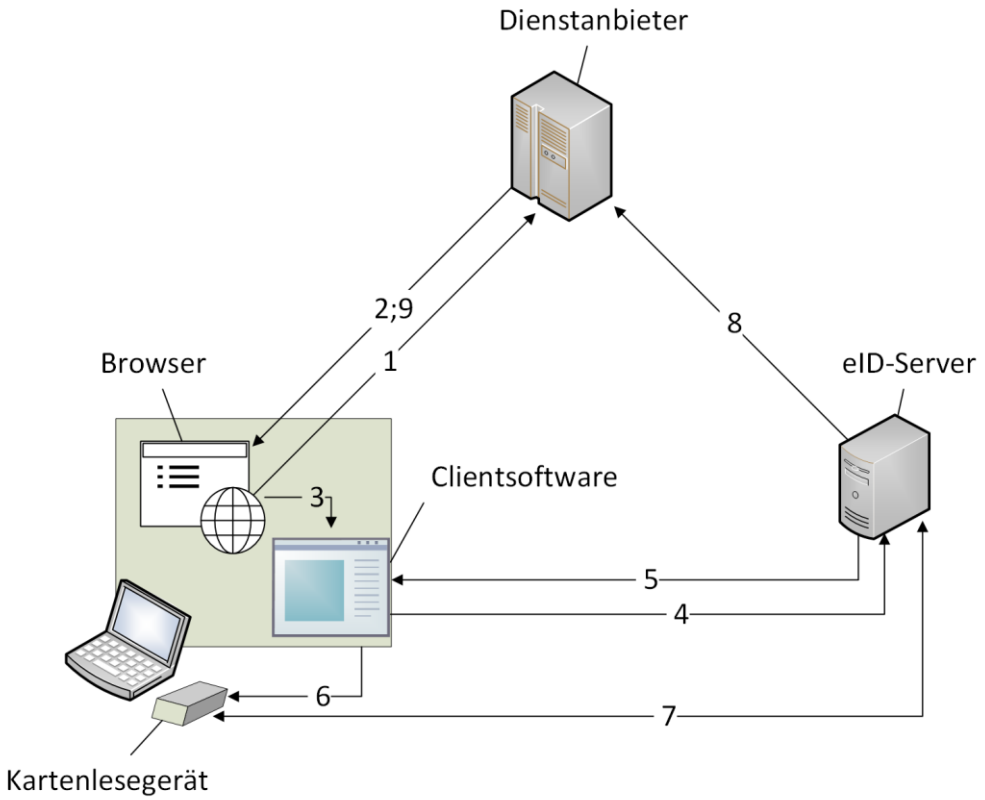


Abbildung 17: Kommunikation während der Online-Ausweisfunktion des nPA

Auf einem Computer mit einem angeschlossenen Kartenlesegerät muss zunächst eine Clientsoftware gestartet werden. Diese, auf dem offiziellen eCard-API-Framework basierende Anwendung, öffnet einen PORT auf dem Localhost und ermöglicht es dem Dienstanbieter, über den Browser einen Authentifizierungsprozess zu initiieren. Während dieses Prozesses ermöglicht die Anwendung eine

Kommunikation über das Kartenlesegerät mit dem neuen Personalausweis. Ruft der Nutzer die Website des Dienstanbieters auf und wählt die Authentifizierungsmethode für den neuen Personalausweis (1), dann übermittelt die Website dem Browser einen Link zum Localhost (127.0.0.1:PORT) mit zusätzlichen Parametern (2). Das Aufrufen dieses Links aktiviert die Clientsoftware auf dem Computer (3) und lässt diese mithilfe der übergebenen Parameter eine Verbindung zu dem eID-Server bzw. -Service Provider des Dienstanbieters aufbauen (4). Der eID-Server, welcher optional von einem Drittanbieter (eID-Service Provider) bereitgestellt werden kann, dient als Kontroll- und Vermittlungsinstanz. Der Server übermittelt der Clientsoftware welche Daten vom Personalausweis erforderlich sind und überträgt das Berechtigungszertifikat des Dienstanbieters (5). Dieses Zertifikat wird nach einer erfolgreichen Prüfung von der Vergabestelle für Berechtigungszertifikate im Bundesverwaltungsamt vergeben und enthält Informationen über den Dienstanbieter sowie die Verwendung der ausgelesenen Daten. Nachdem der Nutzer der Übertragung dieser Daten zugestimmt hat, wird eine sichere Verbindung durch das PACE-Protokoll zwischen Kartenlesegerät und Personalausweis initiiert (6). PACE steht für Password Authenticated Connection Establishment und ist ein vom BSI entwickeltes Authentisierungsverfahren, das auf der Eingabe eines Passwortes beruht und einen hochsicheren Sitzungsschlüssel generiert. Je nach Anforderungen des Dienstanbieters dienen als Passwort u. a. PIN, PUK oder Zugangsnummer des Personalausweises.

Der Personalausweis prüft während der so genannten Terminal-Authentisierung die Gültigkeit des Berechtigungszertifikates und verifiziert den Zugriff des Dienstanbieters auf die angeforderten Daten im Chip des Personalausweises. Es folgt die Chip-Authentisierung auf der Seite des eID-Servers. Diese prüft die Echtheit und Gültigkeit des Ausweis-Chips und initiiert eine hochsichere Ende-zu-Ende-Verschlüsselung der Kommunikation zwischen Personalausweis und eID-Server, um die persönlichen Daten zu übertragen (7). Der eID-Server leitet diese Daten an den vom Nutzer ursprünglich aufgerufenen Service des Dienstanbieters weiter (8). Dieser Vorgang schließt die Nutzer-Authentisierung ab und ermöglicht dem Nutzer den Zugriff auf den Service des Dienstanbieters (9). [54], [55]

3.4.3. Binary Decision Diagrams (BDDs)

Um Problemen beschreiben und lösen zu können, bedarf es einer geeigneten Abstraktion der Problemstellung mithilfe eines Modells. Ein binäres Entscheidungsdiagramm, auch Binary Decision Diagram (BDD) genannt, ist eine Datenstruktur zur Repräsentation von booleschen Funktionen und dient zur Auswertung einer solchen Funktion. Die betrachtete Funktion wird dabei durch eine Baumstruktur repräsentiert. Eingangsgrößen (Variablen) werden als Knotenpunkte dargestellt und über Verzweigungen verbunden, welche in den booleschen Zuständen wahr (T, 1) oder falsch (F, 0) enden. Das Ergebnis spiegelt daraufhin den Wert der booleschen Funktion unter der gewählten Variablenbelegung wider.

Aufbau

Ein BDD ist ein azyklischer, gerichteter Graph G = (V, E). Jedes BDD besitzt einen Anfangsknoten, der auch als Wurzel bezeichnet wird. Abbildung 18 stellt das binäre Entscheidungsdiagramm der booleschen Funktion $f = x_1 \cdot x_2 + x_3$ dar.

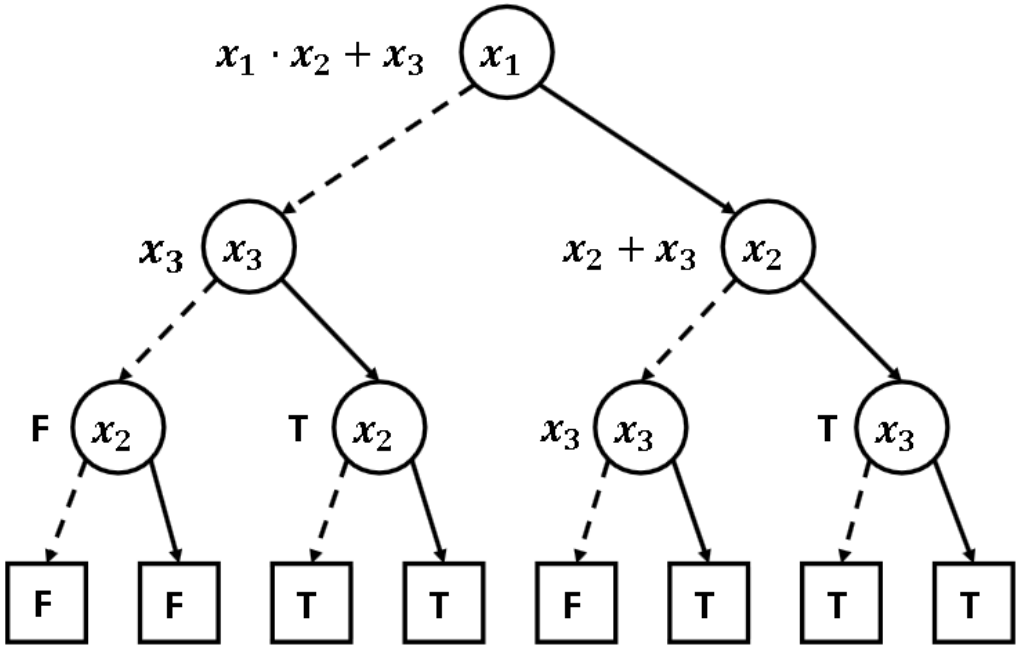


Abbildung 18: BDD einer booleschen Funktion, [56]

Ein BDD ist wie folgt aufgebaut:

- G wird als Baum bezeichnet und besitzt V Knoten
- Jeder Knoten v aus V ist entweder ein Blatt oder ein innerer Knoten
- Blätter (abschließender Knoten) besitzen keine ausgehenden Kanten und sind mit einem Wert aus 0, 1 bzw. F, T beschriftet
- Jeder innere Knoten wird mit einer Variable x_i beschriftet
- Jeder innere Knoten v besitzt zwei ausgehende Kanten. Diese werden als 1, T (ganze Linie) oder als 0, F (gepunktete Linie) bezeichnet und stellen den Entscheidungspfad wahr oder falsch dar.
- Jeder innere Knoten v besitzt zwei Nachfolger bzw. Kinder child (v, F) und child (v, T)

Für die Beschreibung eines BDD werden 2 Eigenschaften verwendet:

(i) Die *Größe* eines BDDs ist gleich der Anzahl der Knoten
(ii) Die *Tiefe* eines BDDs ist gleich der Länge des längsten Pfades

Geordnetes Entscheidungsdiagramm

Man spricht von einem geordneten Entscheidungsdiagramm (englisch: Ordered BDD, kurz OBDD), wenn die Reihenfolge der Variablen auf jedem Pfad von der Wurzel zum Endknoten gleich ist. Die Anordnung der Variablen hat einen Einfluss auf die Größe des OBDD. Deswegen ist es vorteilhaft eine geeignete Reihenfolge zu finden. Ein wichtiger Operator, der benutzt wird, um boolesche Operationen in einem OBDD zu abzubilden, ist der ITE-Operator. Er wird folgendermaßen definiert:

$$ITE(f, g, h) = f \cdot g + \bar{f} \cdot h$$

Wörtlich bedeutet diese Funktion „Wenn f dann g sonst h“, wobei f, g und h drei boolesche Funktionen darstellen. Dies bedeutet, wenn f erfüllt ist, dann ist der Funktionswert von ITE gleich dem

Funktionswert von g, ansonsten gleich dem von h. Alle booleschen Funktionen mit zwei Variablen können über den ITE-Operator dargestellt werden.

Reduziertes Entscheidungsdiagramm

Man spricht davon, dass ein geordnetes BDD reduziert (englisch: reduced, kurz ROBDD) ist, wenn alle Redundanzen im Diagramm entfernt worden sind. Dies geschieht unter Anwendung der folgenden zwei Regeln:

- *Eliminierung*: Entfernen von überflüssigen Knoten
- *Verschmelzen*: Zusammenführen von isomorphen[2] Knoten

Die Verwendung der beiden Regeln verändert jedoch nicht die dargestellte Funktion. Ein Knoten wird eliminiert, wenn seine nachfolgenden Knoten (Kinder) identisch sind. Das bedeutet child (v, F) = child (v, T). Dabei werden, wie in Abbildung 19 dargestellt, der Knotenpunkt und seine Kanten entfernt.

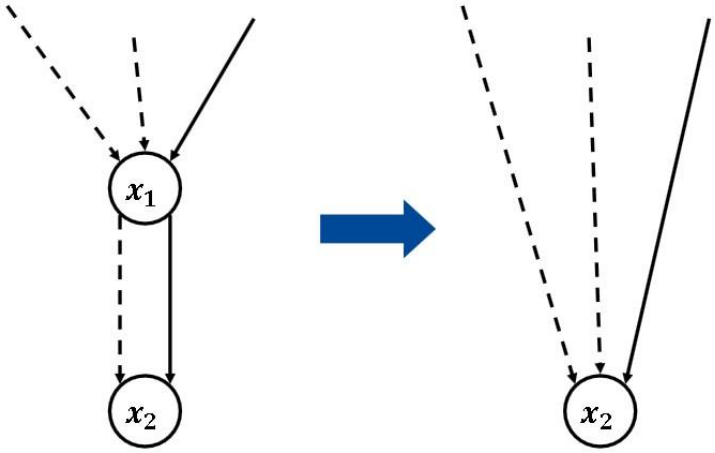


Abbildung 19: Eliminierungsregel für BDDs, [56]

Zwei Knoten werden zusammengeführt, wenn beide die gleiche Variable repräsentieren und identische Kinder besitzen. Mathematisch ausgedrückt bedeutet dies:

$$index\ (v) = index\ (\tilde{v})$$

$$child(v, F\ ) = child(\tilde{v}, F\ )$$

$$child(v, T\ ) = child(\tilde{v}, T\ )$$

Daraufhin wird, wie in Abbildung 20 dargestellt, der Knoten v entfernt und die eingehenden Kanten werden auf $\tilde{v}$ übertragen.

[2] von gleicher Art, Struktur, Form oder Gestalt

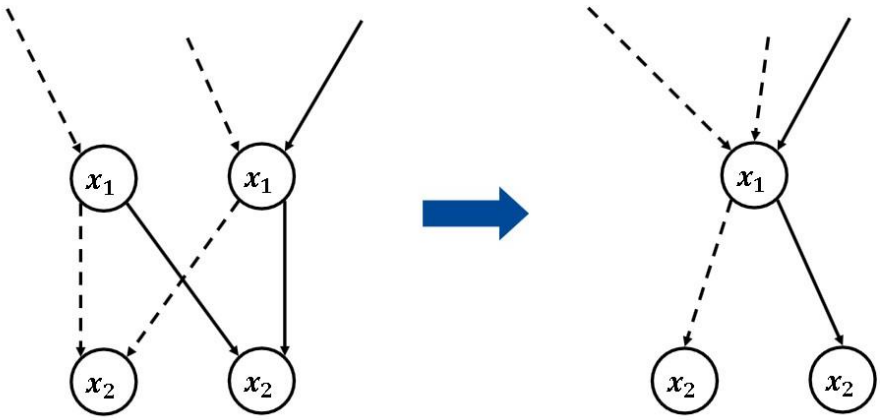


Abbildung 20: Verschmelzungsregel für BDDs, [56]

Entscheidbarkeit

Die wesentliche Eigenschaft eines binären Entscheidungsdiagramm ist die Entscheidbarkeit. Es ist in der Lage eine entscheidbare Menge darzustellen und wird daher als ein Entscheidungsverfahren bezeichnet. Es ist ein Algorithmus, der für jedes Element einer Menge einen Output wahr (1) oder falsch (0) liefern kann.

4. Sicherheits-Framework

Aktuelle IoT- und Smart Home-Geräte können vom Käufer leicht angeschlossen und eingerichtet werden. Es erfordert jedoch Fachkenntnisse vom Nutzer und einen hohen Konfigurationsaufwand, um die Steuerung und Verwaltung dieser Geräte gegen unbefugte Zugriffe abzusichern. Besonders die Steuerung der Smart Home-Geräte aus der Ferne wird häufig als Vorteil solcher Systeme genannt. Wird jedoch die fachkundliche Konfiguration der Geräte und des Netzwerks (u.a. Router) vernachlässigt, bietet eben dieser Vorteil der Fernsteuerung eine erhebliche Angriffsfläche für Angreifer. Um sich die verheerenden Ausmaße solcher Konfigurationsfehler zu verdeutlichen, können IoT-Suchmaschinen wie shodan.io oder insecam.org herangezogen werden. Dort muss nicht lange gesucht werden, um Zugriff auf Überwachungskameras zu erhalten, die Räume privater Wohnungen zeigen. Es gibt eine Vielzahl an möglichen Ursachen, die diesen unbefugten Zugriff ermöglichen, wie z. B. Fehler in der

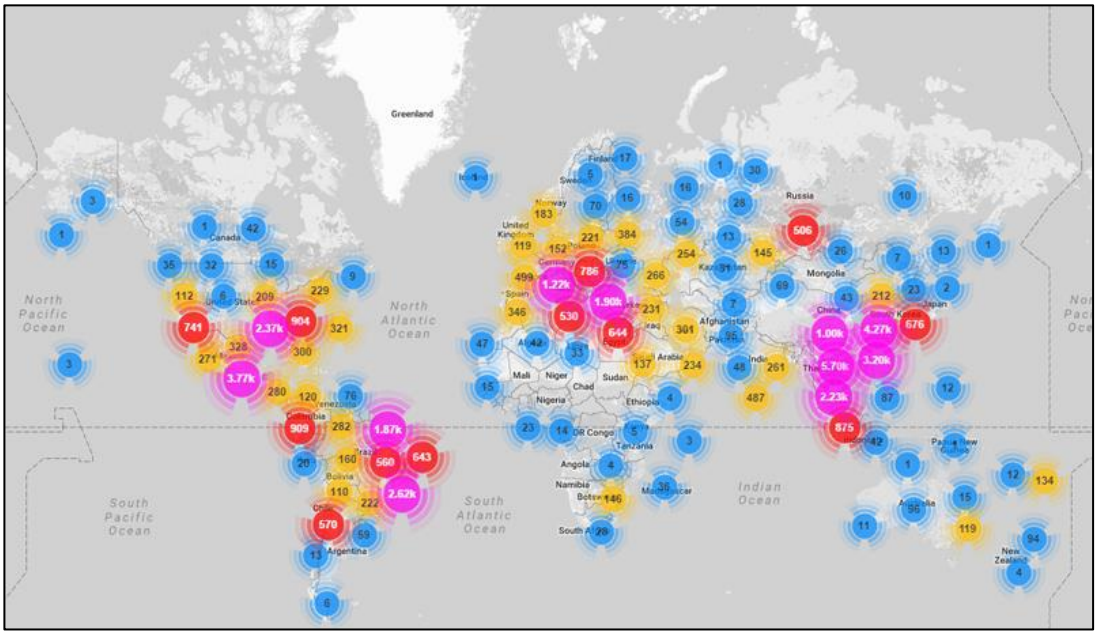


Abbildung 21: Lokalisierung von Geräten die von Mirai infiziert wurden (November 2016) [5]

Firmware der Sicherheitskamera. Jedoch ist die mit Abstand häufigste Schwachstelle eine falsche Konfiguration des Netzwerkes und der teilnehmenden Geräte. Weiterhin hat dieser leichtfertige Umgang mit Sensorik und Aktorik in einer schützenswerten Umgebung das Entstehen von riesigen Botnetzen begünstigt. Eines der bekanntesten Botnetze, „Mirai“, hat weltweit Sicherheitskameras attackiert, infiltriert und für eigene Zwecke missbraucht. Ziel des Botnetzes war der kollektive und gleichzeitige Angriff von Servern im Internet, um deren angebotene Dienste zu blockieren [5]. Solche Attacken werden Distributed Denial of Service-Attacken (DDoS) genannt. Anfang 2018 wurde ein neues Botnetz namens „Hide’n Seek” entdeckt, welches auch in Deutschland Geräte angegriffen hat und Informationen über die Besitzer sammelt. Es wird vermutet, dass die Entwickler das Botnetz zur Spionage und Erpressung einsetzen [57].

Ziel eines neuen Sicherheits-Framework muss es folglich sein, angreifbare offene Verbindungen eines Smart Home-Netzwerkes zu vermeiden und jegliche Kommunikation, interne wie auch externe, abzusichern und somit vor Abhörangriffen oder unbefugter Steuerungsübernahme zu schützen. Um dieses Ziel zu erreichen, muss einerseits die notwendige Konfiguration von Geräten auf ein Minimum reduziert, die Nutzfreundlichkeit priorisiert und die Absicherung der Kommunikation automatisiert werden. Andererseits wird eine sichere und vertrauensvolle Instanz als Ausgangsbasis für solch ein

Sicherheitskonzept benötigt, um von Anfang an Sicherheit zu gewährleisten. Das Projekt für das Sicherheits-Framework basiert auf der Idee einen Teil des neuen Smart Metering Systems, das Smart Meter Gateway (SMGW), als sichere Instanz für ein Smart Home-Netzwerk zu verwenden. In Abschnitt 3.2 wird das SMGW beschrieben und auf die umfangreichen Sicherheitsanforderungen des BSI eingegangen. Die notwendige Zertifizierung durch das BSI erlaubt es, dieses Gerät als eine Instanz mit sehr hohen Sicherheitsstandards anzusehen. Ein weiterer förderlicher Aspekt ist die deutschlandweite Umrüstung von alten Verbrauchszählern auf neue intelligente Messsysteme im Zuge des derzeitigen Smart Metering Rollout. In naher Zukunft wird jedes Gebäude in Deutschland mit einem SMGW ausgestattet sein und erfüllt folglich die Voraussetzungen für das entwickelte Sicherheits-Framework.

4.1. Problemstellung

Die Installation von Smart Home-Geräten ohne Fachkenntnisse führt häufig zu Sicherheitsproblemen durch eine fehlerhafte Konfiguration der Geräte, z. B. keine Authentifizierung durch ein Passwort oder Nutzung der Herstellerwerte (Benutzername, Passwort), und von Elementen der Netzwerkinfrastruktur, wie Routern. Diese Fehler können dazu führen, dass die Privatsphäre der Besitzer aufgrund unerlaubter Steuerung und Überwachung durch Angreifer verletzt werden kann. Die mögliche Angriffsfläche vergrößert sich mit dem Einsatz weiterer Smart Home-Geräte da auch das Potential für eine mögliche Fehlkonfiguration zunimmt. Abbildung 22 veranschaulicht diese Problematik.

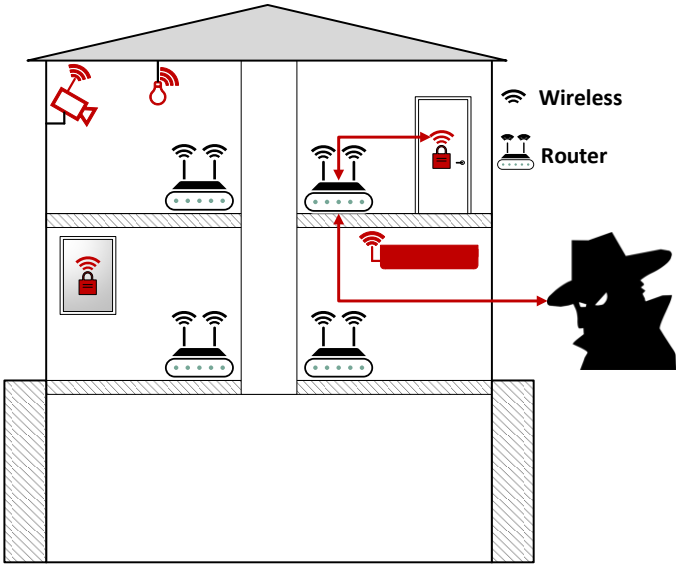


Abbildung 22: Angriffsszenario eines Smart Home-Netzwerks aufgrund von Fehlkonfiguration

Auf Grundlage dieser Problemstellung können drei Anforderungen abgeleitet werden, um ein Netzwerk aus Smart Home-Geräten abzusichern. Bisherige Probleme werden größtenteils durch Fehlkonfigurationen oder Unwissen verursacht. Folglich ist ein Ansatz nötig, der eine größtmögliche Nutzerfreundlichkeit bietet und eine Konfiguration ohne Fachkenntnisse ermöglicht. Weiterhin wird eine Instanz benötigt, die von Anfang an als sicher und vertrauensvoll angenommen werden kann. Ähnlich dem Vertrauensprinzip im Internet. Dort werden Zertifikate genutzt, um Vertrauen zwischen gegenseitig unbekannten Kommunikationspartnern, z. B. Internet-Browser und Website, herzustellen, indem eine vertrauenswürdige Instanz, die Zertifizierungsstelle (Certificate Authority), beiden Teilnehmern garantiert, dass dem jeweils anderen vertraut werden kann. Als sichere Instanz bietet

sich im Smart Home-Szenario das in Abschnitt 3.2 beschriebene Smart Meter Gateway (SMGW) an, welches in Zukunft in Deutschland flächendeckend vorhanden sein wird. Von diesen beiden genannten Anforderungen ausgehend, beschreibt die dritte Anforderung den Bedarf an einer Sicherheitslösung, die eine Absicherung der Gerätekommunikation im Smart Home unter Nutzung des SMGW als sichere, vertrauensvolle Instanz ermöglicht. Die Lösung muss sowohl Kleinstgeräte wie batteriebetriebene Sensorik bis hin zu TV und Smartphone unterstützen. Folglich sind die Ressourcenbeschränkungen (u. a. Rechenleistung, Energieversorgung) einiger Gerätetypen bei der Auswahl einer geeigneten Lösung zu berücksichtigen. Die Anforderung an eine nutzerfreundliche Geräteeinrichtung führt zu einer, für den Nutzer, transparenten Sicherheitskonfiguration und Absicherung der Gerätekommunikation. Somit muss das Sicherheits-Framework selbstständig und automatisiert Sicherheitskonfigurationen der beteiligten Geräte durchführen und einen Verbindungsaufbau über das SMGW zwingend erforderlich machen. Ein Angriff auf ein Smart Home-Netzwerk, das mit diesem Sicherheits-Framework ausgestattet ist, wird nun bereits am SMGW aufgehalten, welches aufgrund der in Abschnitt 3.2.2 beschriebenen hohen Sicherheitsanforderungen als ausreichend geschützte digitale Tür zum Smart Home angesehen werden kann. Das zuvor dargestellte Angriffsszenario verändert sich, wie in Abbildung 23 dargestellt, durch die Integration des Sicherheits-Framework.

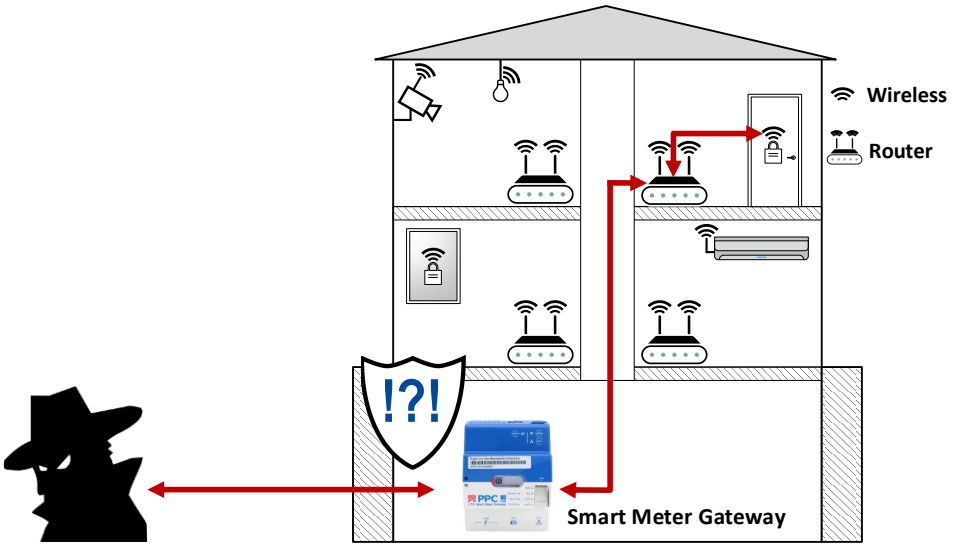


Abbildung 23: Angriff auf Smart Home-Netzwerk mit Smart Meter Gateway, [32]

4.2. Anwendungsszenarien

Die folgenden Szenarien stellen Ausschnitte aus einem beispielhaften alltäglichen Betrieb eines Smart Home dar und fokussieren dabei wichtige Teilaspekte des Sicherheits-Framework.

Authentifizierung von Nutzern und Autorisierung des Nutzergerätes

Der neue Personalausweis kann genutzt werden, um Personen in einem Smart Home zu registrieren und Nutzergeräte, wie z. B. Smartphones, freizuschalten. Personen können zu Berechtigungsgruppen hinzugefügt werden. Auf Basis dieser Informationen lassen sich Autorisierungsvorgänge organisieren. Für Personen ohne Personalausweis, wie z. B. Kinder und Jugendliche unter 16 Jahren, muss entweder ein alternativer Authentifizierungsprozess angeboten werden oder es erfolgt eine Rechtevergabe über authentifizierte Nutzer. So könnte der Mieter einen Zugriff für sein Kind einrichten. Die Rechte dieses

Nutzerkontos müssen jedoch beschränkt sein, um eine Ausnutzung dieser Funktion zu verhindern. Beispielsweise könnte das Kind nur Geräte steuern aber nicht aus dem Netzwerk entfernen oder neue hinzufügen.

An dieser Stelle ist hervorzuheben, dass zwischen dem Authentifizierungs- und Autorisierungsprozess unterschieden werden muss. Zuerst muss sich der Mieter mit dem Personalausweis und durch die Eingabe einer PIN authentifizieren, indem seine Personeninformationen, wie in Abschnitt 3.4.2 beschrieben, abgerufen werden. Daraufhin werden eventuell die erhaltenen Informationen mit einer

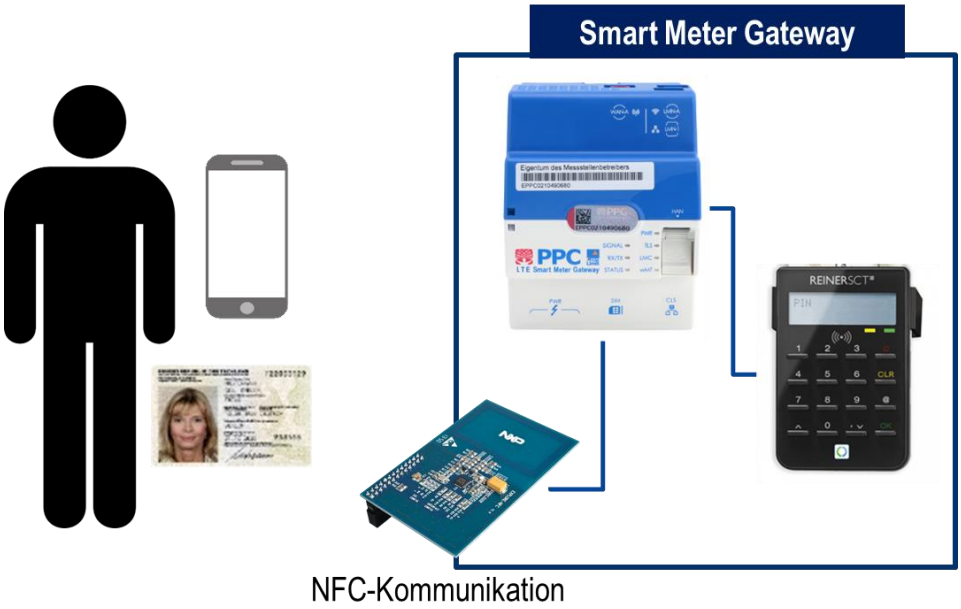


Abbildung 24: Registrierung neuer Nutzer und Autorisierung von Nutzergeräten, [26]

Mieterdatenbank der Wohnungsgenossenschaft abgeglichen, um die Zugriffsberechtigung zu verifizieren, und eine bestimmte Wohnung mit möglicherweise bereits vorhandenen Smart Home-Geräten dem neuen Mieter zuzuordnen. Danach wird der Autorisierungsvorgang gestartet. Während dieses Prozesses wird nicht der Mieter, sondern sein Nutzergerät, z. B. ein Smartphone, autorisiert. Dazu muss eine App für das jeweilige Betriebssystem des Gerätes zur Verfügung stehen, welche den Autorisierungsvorgang auf dem Nutzergerät durchführt und gleichzeitig entsprechende Sicherheitsanforderungen an das Gerät prüft. Beispielsweise muss ein Smartphone adäquate Authentifizierungsmechanismen bieten und aktiviert haben, z. B. Code-Abfrage oder Fingerabdrucklesegerät.

Einbindung neuer Geräte

Geräte mit beschränkten Nutzereingabemöglichkeiten sollen bequem in ein Smart Home-Netzwerk integriert werden können. Nutzergeräte wie Smartphones würden dabei als Vermittler zwischen verschiedenen Gerätetypen dienen und können mithilfe einer sogenannten Out of Band-Kommunikation neue Geräte einfach und sicher hinzufügen [58], [59]. Das Sicherheitsmodul des Smart Meter Gateway dient dabei zur Bestimmung sicherer Passwörter mit hoher Entropie.

Geräte im Smart Home steuern

Wie in Abbildung 25 dargestellt ist, wird für jede Kommunikation zwischen Geräten im Smart Home die Authentizität und Autorisierung der Kommunikationspartner geprüft sowie eine Ende-zu-Ende-

Verschlüsselung für die Verbindung realisiert. Diese Funktionen des Sicherheits-Framework sollen dabei so umgesetzt werden, dass für den Nutzer keine erheblichen Komfortverluste während der Bedienung des Smart Home auftreten und die Kommunikation auch von außerhalb mittels abgesichertem Fernzugriff funktioniert.

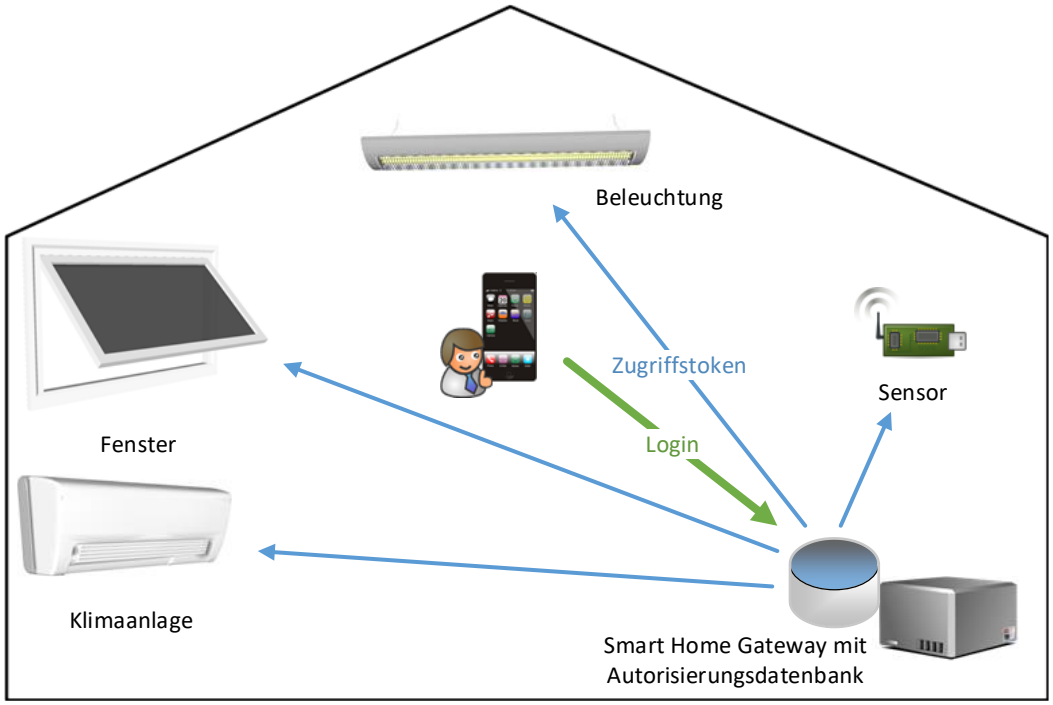


Abbildung 25: Steuerung von Geräten im Smart Home mit integriertem Sicherheits-Framework

Sichere Geräteverwaltung – Beispiel Firmware-Aktualisierung

Basierend auf der sicheren und vertrauenswürdigen Verbindung des Smart Home Gateway zur Service Plattform des Energieanbieters in der Cloud, kann ein Dienst zur Geräteverwaltung und -wartung aufgebaut werden. Dieses Szenario demonstriert die Vorteile eines solchen Dienstes anhand einer Wartungsfunktion zur Prüfung der Firmware-Versionen aller Geräte im Smart Home und Information des Nutzers über vorhandene Aktualisierungen. Der Nutzer kann entscheiden zu welchem Zeitpunkt die Aktualisierung der Geräte durchgeführt wird.

Sicherheitsprofile

Je nach Einsatzort des Smart Meter Gateway werden unterschiedliche Sicherheitsprofile aktiviert. Dieses Szenario soll die Bedeutung der einzelnen Profile veranschaulichen. Im Gegensatz zu Einfamilienhäusern wird in Gebäuden mit mehreren Wohnungen eine Isolierung der Kommunikation der separaten Partien zum Smart Home Gateway erforderlich sein. Weiterhin dürfen Geräte und Personen keinen unbefugten Zugriff auf Datenbankeinträge anderer Smart Home-Systeme im gleichen Gebäude erhalten. Das erarbeitete Sicherheits-Framework bietet für diese Isolierung der unterschiedlichen Teilnehmer/Mieter eine Teillösung über einen für jeden Nutzer individuell verschlüsselten Kommunikationskanal. Es sind jedoch weitere Maßnahmen zu treffen, um die Datenbankabfrage effektiv zu beschränken und besonders eine physikalische Trennung der Heimnetzwerke der Mieter zu erreichen. Diese Probleme gilt es jedoch bereits bei den derzeit in der Zertifizierung befindlichen Smart Meter Gateways zu lösen.

Audio/Video-Gegensprechanlage

Im Rahmen dieses Forschungsprojektes wird eine Audio- & Video-Gegensprechanlage untersucht. Dabei sollen Bild und Ton von der Kamera neben der Haustür zu einem Nutzergerät übertragen werden. Ein Rückkanal in Form einer Audioübertragung ist ebenfalls vorgesehen. Sollte eine Person bei Abwesenheit des Hausbewohners die Türklingel betätigen, wird eine sichere Audio- & Video-Verbindung zum Nutzergerät (z. B. Smartphone) über das Internet aufgebaut. Der Bewohner kann über diese Verbindung in Zukunft beispielsweise per Knopfdruck die Haustür öffnen. Falls der Mieter zuhause ist, kann die Audio- & Video-Übertragung über das lokale Netzwerk des Hauses auf einem Wandpanel wiedergegeben werden. Da die vorgeschlagene Lösung IP-basiert ist, können Standardnetzwerke wie WLAN verwendet werden. Im Gegensatz zu herkömmlichen Lösungen entfallen proprietäre Busstrukturen, die beim Hausbau geplant bzw. mit großem Aufwand nachträglich installiert werden müssten.

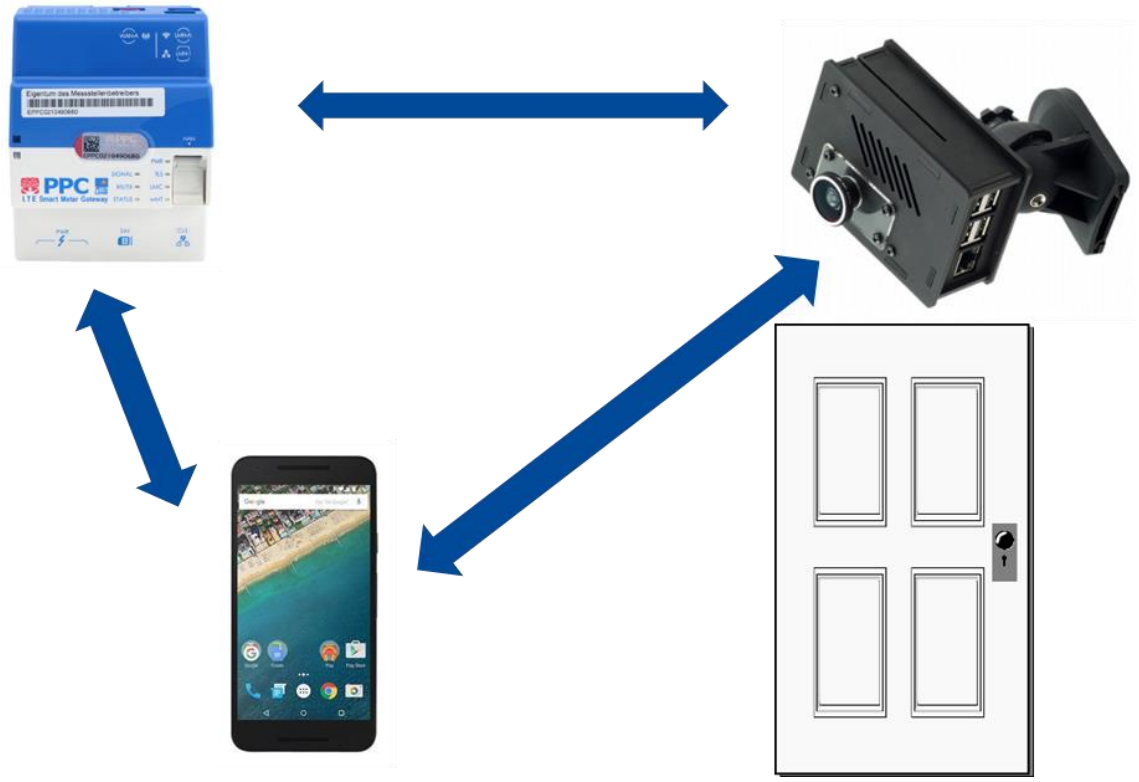


Abbildung 26: Sichere Kommunikation zwischen Gegensprechanlage und Smartphone

4.3. Allgemeine Anforderungen

Das Sicherheits-Framework ist auf Grundlage von vordefinierten Anforderungen, den erarbeiteten Anwendungsszenarien und den aktuellen Entwicklungen am Markt spezifiziert worden. Den Bestrebungen der OCF und dem Markttrend (Abschnitt 2.3, [60], [61]) folgend, basiert die Gerätekommunikation im abgesicherten Smart Home auf CoAP. Dabei werden die durch das Projekt BBSR Plug&Play geschaffenen DPWS-basierten Kommunikationsmodelle und Erweiterungen zu CoAP portiert bzw. nachgebildet. Im Folgenden werden zuerst die allgemeinen Anforderungen und danach die vom BSI Schutzprofil abgeleiteten Anforderungen für das Sicherheits-Framework erläutert.

Ausgewählte Schutzziele der Informationssicherheit für Gerätekommunikation im Smart Home

Die Schutzziele der Informationssicherheit umfassen eine Vielzahl von Anforderungen an ein Kommunikationsnetzwerk. Für den Bereich Smart Home wurden folgende Eigenschaften während der Konzeption als essenziell betrachtet:

Schutzziel	Bedeutung
Vertraulichkeit	… der Kommunikation → Verschlüsselung
Authentizität	… der Kommunikationspartner →Vertrauen durch Schlüsselvergleich, Signaturen und Autorisierung
Integrität	… von Daten und Nachrichten →Empfang von nicht modifizierten Nachrichten, Nachweis durch authentifizierte Prüfsummen
Verfügbarkeit	… von zur Gerätesteuerung benötigten Instanzen und Diensten →Dezentralisierung und Redundanz

Tabelle 5: Ausgewählte Schutzziele der Informationssicherheit für Smart Home-Netzwerke

Abgesicherte Kommunikation im Smart Home-Netzwerk und geschützter Fernzugriff durch externe Nutzergeräte

Diese Forderung stellt den Kern des Sicherheits-Framework dar und bedingt eine abgesicherte Verbindung zwischen Geräten im Smart Home sowie zu Nutzergeräten die sich außerhalb des Smart Home-Netzwerks befinden unter Verwendung des Smart Home Gateway als sichereren, vertrauensvollen Vermittler. Um diese Anforderung zu bedienen, müssen die bereits genannten und für den Smart Home-Bereich relevanten Schutzziele der Informationssicherheit umgesetzt werden. Weiterhin ist die Heterogenität der Smart Home-Geräte zu berücksichtigen. Dies erfordert, dass das Sicherheits-Framework auf Kleinstgeräten, wie z. B. batteriebetriebener Sensorik, über Aktorik mit Netzanschluss bis hin zu Smartphones und Computern eingesetzt werden kann und die Eigenschaften der unterschiedlich performanten Geräte vorteilhaft kombiniert.

Berücksichtigung etablierter Kommunikationstechnologien im Smart Home-Bereich, um Annahme des Sicherheits-Framework durch Industrie zu begünstigen

In den letzte Jahren haben sich verschiedene Kommunikationstechnologien für den Smart Home-Bereich etabliert (siehe Abschnitt 2.2) und bisher zeichnet sich kein Trend zur Vereinheitlichung der physikalischen Kommunikationsschicht ab. Daher muss das Sicherheits-Framework unabhängig von den unteren Schichten des ISO/OSI-Modells konzipiert werden, um keine Nischenlösung zu erarbeiten.

Abgesicherte Kommunikation mit Service Plattform über WAN-Schnittstelle des Smart Home Gateway

Die Umsetzung dieser Anforderung ermöglicht neben der bisher möglichen Abfrage von Verbrauchsdaten durch den Energieanbieter, die Anbindung an einen Service Provider in der Cloud.

Basierend auf dieser Grundfunktionalität können Mehrwertdienste, wie z. B. Ambient Assisted Living (AAL), realisiert werden. Weiterhin kann mithilfe der Cloud-Anbindung ein intelligentes Gerätemanagement angeboten werden, welches den Nutzer beispielsweise über veraltete Firmware von Geräten im Smart Home informiert und eine sichere Firmware-Aktualisierung dieser Geräte ermöglicht.

Integration neuer Geräte ins Smart Home durch autorisiertes Nutzergerät

Die Integration neuer Smart Home-Geräte in das bestehende Netzwerk erfordert ein sicheres und nutzerfreundliches Verfahren, welches nur von autorisierten Nutzergeräten, wie z. B. Smartphone oder Tablet, initiiert werden darf. Für die dazu nötige Autorisierung eines Nutzergerätes wird ein Konzept benötigt, welches auf der Identifikation, Authentifikation und Autorisierung des Mieters der Wohneinheit durch das Smart Home Gateway basiert. Der Vorgang des Hinzufügens neuer Geräte wird in diesem Projekt abstrahiert. Es wird die Initiierung des Hinzufügens, die Schlüsselgenerierung und das Eintragen in die Autorisierungsdatenbank betrachtet. Der Teilvorgang der Kommunikation zwischen autorisiertem Nutzergerät und neuem Smart Home-Gerät vor der Integration ins Netzwerk wird nicht bearbeitet. Diese sogenannte Out-of-Band-Kommunikation ist zum einen gerätespezifisch (Kommunikation über LED-Lampe, WLAN Access Point, NFC) und andererseits ein eigenes Forschungsgebiet, mit dem sich auch Forscher der Universität Rostock bereits auseinandergesetzt haben [58].

Ableiten von Anforderungen an ein Smart Home Gateway basierend auf dem BSI Schutzprofil für das Smart Meter Gateway

Den Ausgangpunkt für das Sicherheits-Framework bildet das Smart Meter Gateway, welches eine sichere und vertrauenswürdige Instanz im Netzwerk darstellt. Dieses wird einerseits als sicherer Speicherort für Nutzerinformation und Autorisierungsdaten der Geräte dienen und andererseits als sicherer Vermittler zwischen Geräten innerhalb des Smart Home sowie für den Remotezugriff (Initiierung einer sicheren Gerätekommunikation mit externen Geräten über WAN-Schnittstelle) agieren. Ausgewählte sicherheitsrelevante Anforderungen des BSI Schutzprofils für das Smart Meter Gateway wurden bei der Konzeption des Sicherheits-Framework berücksichtigt. Weiterhin wurden benötigte Erweiterungen bzw. Änderungen des Schutzprofils für den Betrieb eines Smart Home Gateway erarbeitet. Die übernommenen Anforderungen aus dem BSI Schutzprofil sowie relevante Änderungen in Bezug auf die Smart Home-Integration werden in Abschnitt 4.4 beschrieben.

4.4. Abgeleitete Anforderungen vom BSI Schutzprofil für das Smart Meter Gateway

Eine besonders sicherheitsrelevante Anforderung im Schutzprofil für das SMGW und in der zugehörigen Technischen Richtlinie BSI-TR-03109 ist die Beschränkung der Informationsflüsse zwischen den drei Schnittstellen WAN, LMN und HAN des SMGW. Für das Sicherheits-Framework ist die Kommunikation zwischen WAN- und HAN-Schnittstelle relevant. Das Schutzprofil legt fest, dass ein Informationsfluss vom WAN zum HAN über das SMGW nicht erlaubt ist. Der umgekehrte Datenfluss (HAN → WAN) ist unter Einschränkungen erlaubt. Diese erfordern, dass eine Verbindung nur zu vertrauensvollen vordefinierten Endpunkten im WAN aufgebaut werden darf. Die Informationsflüsse werden in Tabelle 1 in Abschnitt 3.2.2 übersichtlich dargestellt. Eine Besonderheit stellt der Wake-Up-Service des SMGW (BSI Schutzprofil, Abschnitt 1.4.6.5) dar. Der Gateway Administrator darf das SMGW

von außen „aufwecken“, sodass das Gateway ebenfalls eine Verbindung zu einem vordefinierten Endpunkt im WAN aufbaut. Für das Sicherheits-Framework, im Besonderen für das Smart Home Gateway, werden daraus folgende Anforderungen abgeleitet:

- Die Kommunikation zwischen einem Gerät im Smart Home-Netzwerk und einem sich außerhalb befindlichen Nutzergerät darf nur von innen nach außen (HAN/LAN → WAN) aufgebaut werden.
- Eine direkte Kommunikation zwischen externem Nutzergerät und Smart Home-Gerät über das Smart Home Gateway ist nicht erlaubt. Eine vermittelnde Funktion soll unter Beachtung von Einschränkungen möglich sein.

Weitere Anforderungen des BSI Schutzprofils sind die Einschränkung der Interaktion mit dem SMGW auf einen reinen Lesezugriff (Read-Only) und die Passivität des SMGW während einer HAN-zu-HAN-Kommunikation. Erstere erfordert, dass Mieter oder Smart Home-Geräte keine Daten auf dem SMGW schreiben/speichern dürfen. Letztere beschreibt, dass das SMGW nicht in die Kommunikation zwischen zwei Geräten im HAN involviert wird. Folgende Anforderung für das Sicherheits-Framework wurde davon abgeleitet:

- Diese Einschränkung behindert die Nutzung des Potenzials des SMGW für die Smart Home-Integration. Daher wird diese Anforderung für das Framework nicht direkt übernommen. Autorisierte Geräte sollen neben dem Lesezugriff auch Schreibrechte auf ausgewählte Bereiche erhalten. Da diese Anpassung der Anforderungen an das SMGW eventuell unterschiedliche Hardware-Anforderungen von Smart Meter Gateway und Smart Home Gateway impliziert, wird nach der Beschreibung des Konzepts des Sicherheits-Framework in Abschnitt 4.5.4 eine Erweiterung beschrieben, die die Anforderung bezüglich eines reinen Lesezugriffs erfüllt.
- Die Aufwertung des Smart Meter Gateway zu einem Smart Home Gateway erfordert eine Interaktion mit Smart Home-Geräten im HAN. Unter Verwendung des zuvor genannten Workarounds kann die Interaktion auf ein Minimum reduziert werden, grundsätzlich ist jedoch eine genauere Definition an dieser Stelle nötig. Beispielsweise gibt es Kommunikationsabläufe wie in Abschnitt 3.3.6 (ACE) beschrieben, bei denen das Gateway als Authorization Server involviert ist. Dabei wird das Gateway von den beiden anderen Kommunikationspartnern einzeln angesprochen, aber nicht genutzt, um über das Gateway hinweg zu kommunizieren. Aus dem Protection Profile und den Technischen Richtlinien geht nicht eindeutig hervor, ob diese Kommunikation zulässig ist.

4.5. Konzept

Angestrebtes Ziel dieses Projektes ist es, Hersteller und Entwickler von Smart Home-Geräten dabei zu unterstützen, das durch den Smart Metering Rollout bald flächendeckend verfügbare Potential des Smart Meter Gateway als sicheren, vertrauensvollen Vermittler und Zugangspunkt zum Smart Home-Netzwerk zu nutzen. Zuerst wird in Abschnitt 4.5.1 die zugrunde liegende Smart Metering-Infrastruktur eingeführt und erweitert. Darauf folgt in Abschnitt 4.5.2 die Beschreibung und Verknüpfung der zusammengestellten Protokolle des Sicherheits-Framework. Zum Abschluss des Konzeptkapitels wird in Abschnitt 4.5.4 auf Ergänzungen und Anpassungen des Sicherheits-Framework hingewiesen, die die Integration des Framework in aktuelle Smart Home-Produkte beschleunigen können.

4.5.1. Kombinierte Smart Home- und Smart Metering-Infrastruktur

Smart Metering-Systeme im Sinne von intelligenten Messsystemen (siehe Abschnitt 3.2) bestehen aus einem digitalen Verbrauchszähler je Wohneinheit und Energiequelle, sowie einem Smart Meter Gateway (SMGW) als Kommunikationseinheit. Letzteres arbeitet als Vermittler zwischen lokalem digitalen Verbrauchszähler und externem Energieversorger, indem es die empfangenen Verbrauchsdaten an den Versorger bzw. Messstellenbetreiber weiterleitet. Um diese Aufgabe durchzuführen, ist das SMGW über eine WAN-Schnittstelle mit dem Internet verbunden und enthält ein Sicherheitsmodul, welches sicherheitsrelevante und kryptografische Funktionen bereitstellt. Der Internetzugang kann unter anderem über ein Mobilfunkmodem, z. B. ein GSM/UMTS-Modem, oder über Power Line Communication (PLC) hergestellt werden. PLC überträgt Daten über das Stromnetz und erübrigt folglich das Verlegen weiterer Kabel oder Antennen.

In Deutschland muss jedes Smart Meter Gateway (SMGW) von einer akkreditierten Prüfstelle des Bundesamtes für Sicherheit in der Informationstechnik (BSI) gemäß einem spezifischen Schutzprofil (Protection Profile) und den dazugehörigen Technischen Richtlinien zertifiziert werden (siehe Abschnitt 3.2.2). Die daraus resultierenden hohen Sicherheitsstandards zeichnen das Gerät als sichere und vertrauensvolle Instanz aus, deren Potential für die Absicherung der Gerätekommunikation im Smart Home-Netzwerk genutzt werden kann.

Die Aufwertung des Smart Meter Gateway zu einem Smart Home Gateway (SHGW) beinhaltet die Integration des Gateways ins Smart Home-Netzwerk der Wohneinheiten über die HAN-Schnittstelle des Gerätes und die Implementierung des im Folgenden genannten Protokoll-Stack des Sicherheits-Framework. Das aufgewertete SHGW stellt sein integriertes Sicherheitsmodul für Mehrwertdienste und die verbundenen Smart Home-Netzwerke zur Verfügung. Derzeit gibt es keine offiziellen Richtlinien für die Realisierung der Anbindung an die Smart Home-Netzwerke der Wohneinheiten. Im Konzept für das Secure Smart Home-Projekt wird daher eine PLC-Verbindung vorgeschlagen, um die Installationskosten im Falle einer Smart Metering-Nachrüstung zu senken. Dies kann zukünftig durch PLC-fähige Router und Verbrauchszähler erreicht werden. In Abbildung 27 wird die kombinierte Smart Home- und Smart Metering-Infrastruktur dargestellt.

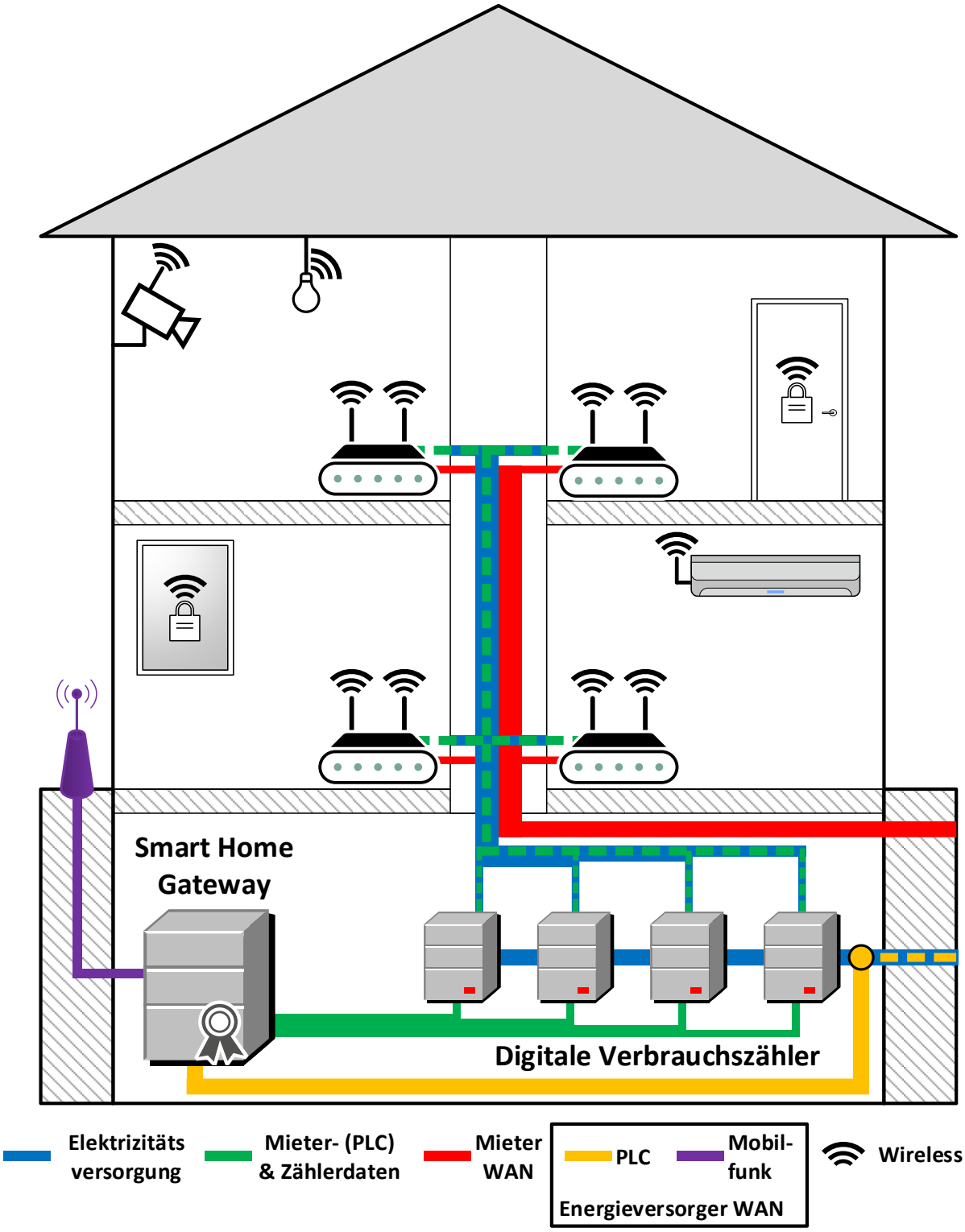


Abbildung 27: Infrastruktur des Sicherheits-Framework am Beispiel eines Gebäudes mit vier Wohneinheiten ausgestattet mit Smart Home-Geräten und Smart Metering-System

4.5.2. Protokolle des Sicherheits-Framework

Der zusammengestellte Protokoll-Stack des Sicherheits-Framework basiert auf aktuellen Tendenzen und Entwicklungen im Bereich des Smart Home (siehe Abschnitt 2). In Abbildung 28 sind die Protokolle des Sicherheits-Framework anhand des TCP/IP-Schichtenmodells dargestellt. Aufgrund der großen Vielfalt an etablierten Übertragungsprotokollen im Smart Home-Bereich ist das Framework unabhängig Verbindungsschicht (engl. Link Layer) konzipiert worden, um eine möglichst generische Lösung für eine große Anzahl an Smart Home-Geräten zu bieten.

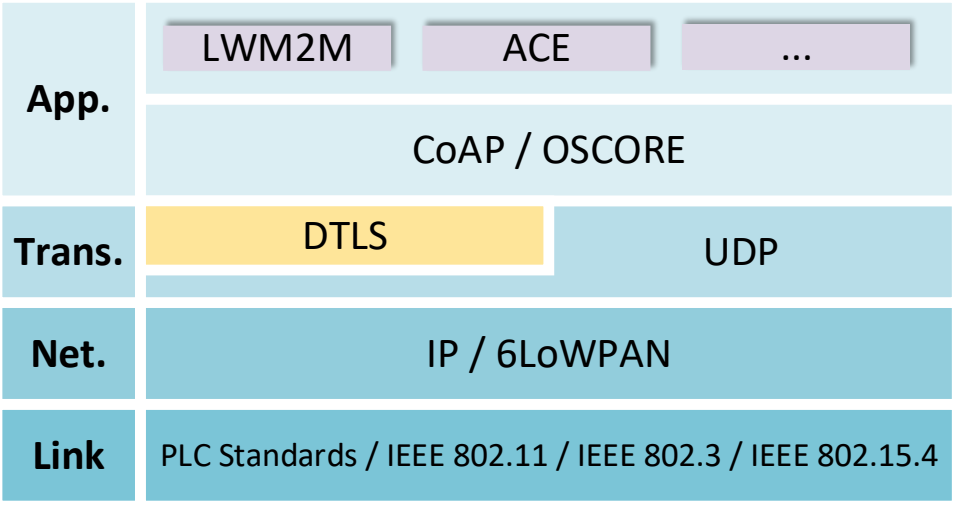


Abbildung 28: Protokolle des Sicherheits-Framework angeordnet im TCP/IP Schichtenmodell

Beispielsweise wird ausgehend von der Darstellung der Infrastruktur in Abbildung 27 eine PLC-Verbindung für die Kommunikation zwischen Router in der Wohneinheit und SHGW (IEEE 1901, HomePlug) sowie zwischen SHGW und Energieversorger (IEEE 1901, verschiedene ITU-Standards) verwendet. Während im Smart Home-Netzwerk für vereinzelte Verbindungen IEEE 802.3 Ethernet [62] verwendet wird, sind die wichtigsten Protokolle IEEE 801.11 WLAN [63] (z. B. für Smartphones und Tablets) und IEEE 802.15.4 [64]. Werden mehrere Funkprotokolle im selben Netzwerk verwendet werden, ist ein Gateway erforderlich, um Geräte über die Protokollgrenzen hinweg miteinander zu verbinden.

Auf der Netzwerkschicht (engl. Network Layer) setzt das Sicherheits-Framework das Internetprotokoll (IPv4/v6) oder ein schichtenübergreifendes Profil wie 6LoWPAN [65] (ebenfalls IP-basiert) voraus. Letzteres ist für ressourcenarme Geräte mit IEEE 802.15.4-Funkschnittstelle entwickelt worden.

Eine Schicht höher im TCP/IP-Modell wird das User Datagram Protocol (UDP) zur Realisierung einer leichtgewichtigen und verbindungslosen Transportschicht (engl. Transport Layer) erfordert. Entscheidender Vorteil dieses Protokolls für den Einsatz im Smart Home ist der kleinere Kommunikations-Overhead im Vergleich zum Transmission Control Protocol (TCP). Diese Eigenschaft verringert die Energiekosten für das Versenden einer Nachricht und wirkt sich folglich positiv auf die Standby-Zeit batteriebetriebener Sensorik aus.

Das Sicherheits-Framework setzt auf Anwendungsebene (engl. Application Layer) das Constrained Application Protocol (CoAP, siehe Abschnitt 3.3.1), ein populäres REST-basiertes Nachrichtenprotokoll für Netzwerke bestehend aus Geräten mit Ressourcenbeschränkungen, z. B. Sensornetzwerke, voraus.

CoAP bietet keine Funktionen zur Absicherung der Kommunikation. Daher beinhaltet das Framework zusätzliche Sicherheitsprotokolle, um Integrität, Vertraulichkeit und Authentizität der Kommunikation zu gewährleisten. Eines dieser Protokolle ist das vom IETF CoAP Standard empfohlene Datagram Transport Layer Security (DTLS), welches das etablierte TLS-Sicherheitskonzept über UDP-Kommunikation realisiert. Aufgrund einiger Nachteile (siehe Abschnitt 3.3.4) enthält das Sicherheits-Framework ein zweites Sicherheitsprotokoll. Object Security for Constrained RESTful Environments (OSCORE, siehe Abschnitt 3.3.4) ist eine neue Protokollerweiterung für CoAP, derzeit noch ein IETF Draft, und ersetzt die Sicherheitsfunktionen von DTLS. Weiterhin behebt es die Nachteile, die bei der Kombination von CoAP und DTLS entstehen. Beide Sicherheitsprotokolle sichern die Gerätekommunikation im Smart Home durch Verschlüsselung (Vertraulichkeit) und signierte Prüfsummen (Authentizität, Integrität) ab. Während DTLS aufgrund des etablierten TLS-Konzeptes Teil des Framework ist, wurde OSCORE aufgrund seiner Vorteile gegenüber DTLS aufgenommen.

Zusätzlich zur Absicherung der Kommunikation enthält das Framework eine Autorisierungsfunktion, um zu prüfen ob zwei Geräte berechtigt sind miteinander zu kommunizieren. Dazu wird ebenfalls ein Protokoll verwendet, welches sich noch im IETF Draft-Zustand befindet. Authentication and Authorization for Constrained Environments (ACE, siehe 3.3.6) dient zur Verwaltung von Zugriffsberechtigungen in Smart Home- und IoT-Szenarien. Es basiert auf OAuth2.0 (siehe Abschnitt 3.3.5) und definiert die Rollen Authorization Server (AS), Resource Server (RS) und Client. Das SHGW übernimmt die Rolle des AS, während Smart-Home-Geräte und Nutzergeräte, wie z. B. ein Smartphone, je nach Szenario RS oder Client repräsentieren. ACE ist eng mit OSCORE verknüpft, kann jedoch auch oberhalb von CoAP mit DTLS verwendet werden. Basierend auf ACE und einer Autorisierungsdatenbank im SHGW wurde ein Zugriffsmanagement entwickelt, das die Steuerung von Geräten nur berechtigten Nutzern ermöglicht. Darüber hinaus ist es möglich, bereits erteilte Berechtigungen im Nachhinein zu widerrufen. Das SHGW mit seinen zertifizierten Sicherheitseigenschaften dient als zentrale Autorisierungsinstanz im Smart Home-Netzwerk und verwaltet Datenbanken für Zugriffsberechtigungen und Sicherheitsschlüssel angemeldeter Geräte. Das Anlegen von Datenbanken auf dem SHGW erfordert ein Abmildern der Lesezugriff-Anforderung des BSI Schutzprofils (siehe Abschnitt 4.4). Autorisierte Nutzergeräte, wie z. B. das Smartphone des Mieters, müssen Funktionen zur Erstellung neuer Einträge oder dem Entfernen alter Einträge in der Autorisierungsdatenbank zur Verfügung stehen. Weiterhin stellt das SHGW aufgrund seiner Rolle als Autorisierungsinstanz einen essenziellen Bestandteil der sicheren Kommunikation und Gerätesteuerung im Smart Home dar. Daher ist auch die Forderung nach passivem Verhalten in einer HAN-zu-HAN-Kommunikation vom SHGW nicht zu gewährleisten.

Die Wartbarkeit von Smart Home-Geräten ist eine essenzielle Voraussetzung für das Bestreben Sicherheit über den gesamten Produktlebenszyklus gewährleisten zu können. Zur Administration dieser Wartungsaufgaben sowie zur allgemeinen Orchestrierung der Geräte beinhaltet das Sicherheits-Framework das Geräteverwaltungsprotokoll LwM2M der Open Mobile Alliance (OMA) [66]. LwM2M verwendet CoAP, um eine nutzerfreundliche Verwaltung mehrerer Geräte zu realisieren. Dies ermöglicht unter anderem eine automatisierte Kontrolle der Firmware-Versionen aller Geräte im Smart Home sowie die Automatisierung notwendiger Firmware-Upgrades, um Schwachstellen zu beheben.

4.5.3. Kommunikationsabläufe

In diesem Abschnitt werden die Abläufe von Verbindungsaufbau und Datenaustauschen für verschiedene Szenarien beschrieben und dargestellt. Als interne Kommunikation wird der Fall bezeichnet, bei dem beide Kommunikationspartner (Client und Resource Server) sich im lokalen Netzwerk befinden. Dementsprechend wird eine Kommunikation eines Smart Home-Gerätes mit einem Client der sich außerhalb des lokalen Netzwerkes befindet als externe Kommunikation bzw. Fernzugriff bezeichnet. Die Darstellungen visualisieren den generellen Ablauf der Kommunikation und enthält keine tiefergehenden Details, wie z. B. Schlüsselaustausch, das Signieren von Nachrichten (Authentizität, Integrität) oder die Verschlüsselung der Verbindung (Vertraulichkeit).

Interne Gerätesteuerung

In diesem Szenario befindet sich das Nutzergerät (Client), z. B. ein Smartphone, im lokalen Netzwerk bzw. in der Wohnung. Abbildung 29 visualisiert den Kommunikationsverlauf für dieses interne Zugriffsszenario. Bevor der Client auf Sensorik oder Aktorik im Smart Home, welche die Rolle des Resource Server einnimmt, zugreifen kann, muss das Smartphone einen Zugriffs-Token beim Authorization Server, dem Smart Home Gateway (SHGW), beantragen. Mithilfe der bei dieser Anfrage übermittelten Informationen kann das SHGW die Authentizität und Autorisierung des anfragenden Clients prüfen. Falls das Smartphone bzw. dessen Nutzer auf Grundlage der Autorisierungsdatenbank die Berechtigung zum Steuern der Sensorik/Aktorik besitzt, generiert das SHGW einen Zugriffs-Token und übermittelt diesen zusammen mit Informationen über die zu steuernde Sensorik/Aktorik an den Client. Diese RS-Informationen enthalten u. a. den öffentlichen Schlüssel zum Aufbau einer sicheren Kommunikation, eine Liste der unterstützten Kommunikationsprotokolle (CoAP+DTLS, OSCORE) und die Art des Tokens (kleiner Token für Prüfung durch Authorization Server oder größerer Web Token für selbstständige Autorisierungsprüfung). Daraufhin kann der Client eine direkte Verbindung zum Resource Server aufbauen und den Token sowie den Steuer- oder Lesebefehl übermitteln. Je nach Token-Typ prüft nun die Sensorik/Aktorik den Token selbstständig oder übergibt ihn optional an das SHGW zu Prüfung. Fällt die Autorisierungsprüfung positiv aus, wird der übermittelte Befehl ausgeführt und eine entsprechende Nachricht an das Smartphone zurückgeschickt. Im Folgenden können weitere Befehle vom Client übertragen werden. Benötigen diese Befehle andere Zugriffsrechte auf die Sensorik/Aktorik, z. B. war der erste Befehl nur ein Lesezugriff, während mit den folgenden Befehlen Daten auf den Resource Server geschrieben werden sollen, muss der Client erst einen neuen Token mit diesen anderen Berechtigungen beim Authorization Server beantragen.

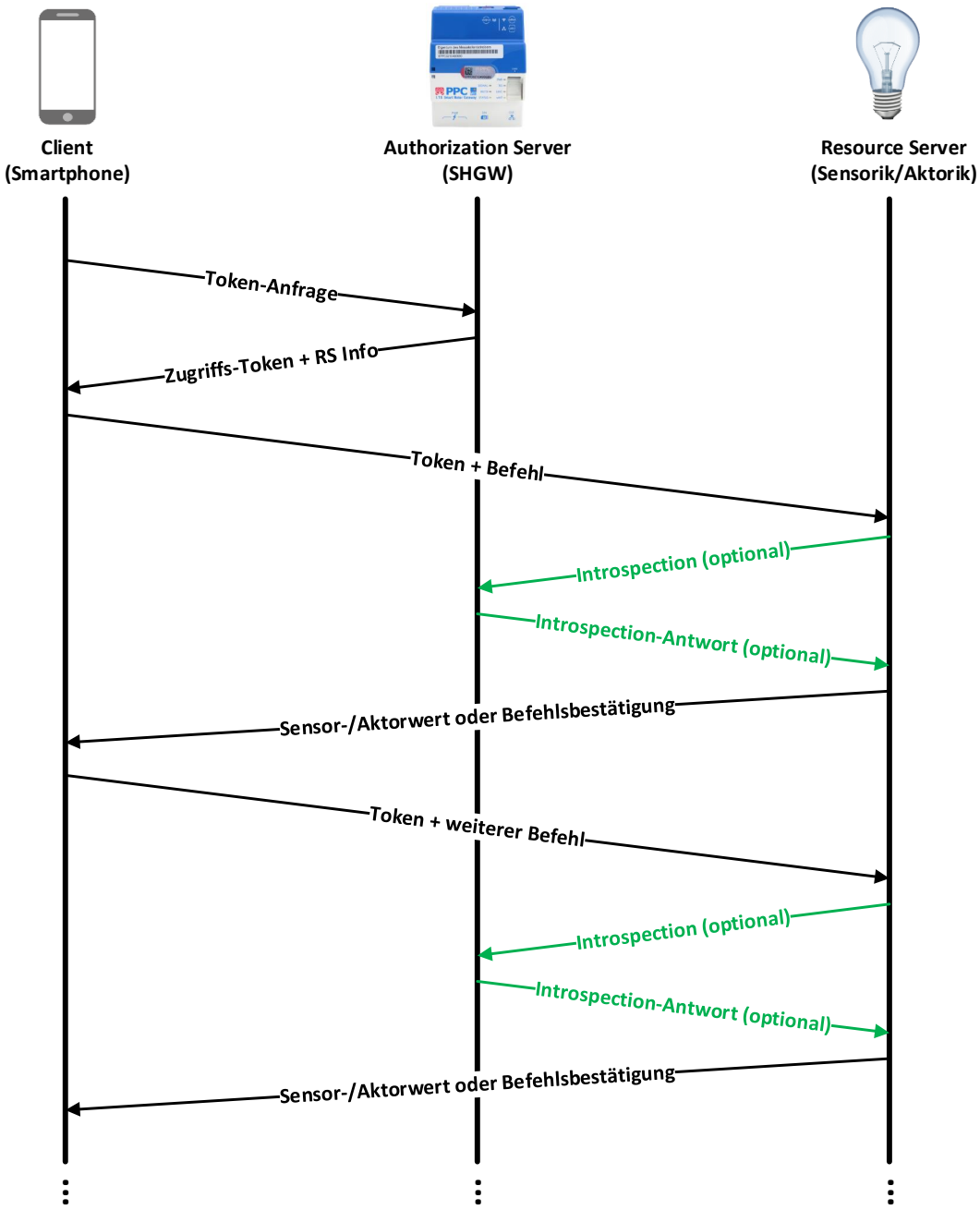


Abbildung 29: Interner Kommunikationsablauf, [32]

Fernzugriff auf Smart Home-Geräte

In Abbildung 30 wird der Nachrichtenaustausch für das Szenario eines Fernzugriffs, z. B. Smartphone im Mobilfunknetz, auf Smart Home-Geräte dargestellt. Um eine Steuerung unter diesen Bedingungen zu ermöglichen, wird ein weiterer Kommunikationspartner integriert. Die Service Platform ist eine Cloud-Anwendung, die im SHGW als vordefinierter Endpunkt über die WAN-Schnittstelle des SHGW erreichbar ist. Bis auf eine sogenannte Wake-Up-Nachricht darf die Kommunikation nur vom SHGW in Richtung Service Platform aufgebaut werden. Innerhalb der Service Platform befinden sich vertrauenswürdige Applikationen, die als Gegenstellen zur Realisierung der intelligenten Funktionen der Smart Metering-Systeme benötigt werden (z. B. Verbrauchsdatenübermittlung an Energieversorger, Steuerung regelbarer Lasten und Erzeuger). Eine dieser Applikationen realisiert im

Rahmen des Sicherheits-Framework den Fernzugriff durch autorisierte Nutzergeräte auf das Smart Home. Um dies zu realisieren, wird für den Client ein Konto in der Service Platform eingerichtet. Nur mit den dort hinterlegten Authentifizierungsdaten kann der Client eine Verbindung zum Smart Home aufbauen. Dabei agiert die Service Platform nur als erste Instanz. Das SHGW prüft die Authentizität und Zugriffs-Berechtigungen mithilfe der Autorisierungsdatenbank (privater Schüssel des Clients nötig, höhere Sicherheitsstufe als Login-Daten der Service Platform). So kann verhindert werden, dass nicht authentifizierte Zugriffsanfragen bis zum SHGW weitergeleitet werden und Ressourcen binden. Weiterhin ist durch eine solche Kaskadierung der Zugriffsprüfung sichergestellt, dass auch Anfragen von Geräten mit kürzlich geänderten Berechtigungen korrekt geprüft werden.

Der sich außerhalb des lokalen Netzwerkes befindliche Client sendet eine Token-Anfrage mittels einer App, die automatisch zwischen lokalem und externem Netzwerk unterscheiden kann, eine Zugriffsanfrage an die Service Platform in der Cloud. Nach erfolgreicher Authentifizierung nutzt die Fernzugriffs-Applikation auf der Service Platform den Wake-Up-Service, um dem SHGW mitzuteilen, dass eine Kommunikation erforderlich ist. Daraufhin baut das SHGW die Verbindung zur vordefinierten Service Platform auf und ruft anstehende Nachrichten ab. Im Zuge dessen erhält das SHGW die Token-Anfrage des Clients prüft die Authentizität, grundlegende Autorisierung erneut und gleicht die Anfrage mit den bestehenden Berechtigungen für den Client ab. Nach erfolgreicher Prüfung wird ein Zugriffs-Token generiert und zusammen mit RS-Informationen über die Cloud an den Client geschickt. Weiterhin wird dem entsprechenden Resource Server mitgeteilt, dass eine externe Verbindung erforderlich ist und zusätzlich Client-Informationen übertragen. Bis zu diesem Schritt im Kommunikationsverlauf fand der Nachrichtenaustausch zwischen externen Kommunikationspartnern (Client, Cloud) und Smart Home (Authorization Server, Resource Server) über die WAN-Schnittstelle des SHGW respektive den Internetzugang des Energieproviders statt. Die anschließenden Kommunikationsschritte werden über den WAN-Anschluss der Wohnung bzw. des Mieters (Internetanschluss über Router/Modem) durchgeführt. Mithilfe der Client-Informationen baut die Sensorik/Aktorik eine direkte Verbindung zur Service Platform auf, teilt dieser mit, welcher Client ihr Kommunikationspartner ist und wartet auf die Übermittlung des Befehls und Token vom Client über die Cloud. Anschließend prüft der Resource Server den Token entweder selbstständig oder leitet ihn weiter zum Authorization Server. Nach erfolgreicher Autorisierung wird der Befehl ausgeführt und eine Antwort an den Client über die Cloud geschickt. Auch hier können weitere Befehle vom Client versendet werden. Falls diese andere Berechtigungen benötigen als der vorhandene Token erlaubt, muss ein neuer Token beim SHGW über die Service Platform beantragt werden. An dieser Stelle kann der Vorteil des OSCORE-Kommunikationsprotokolls, durchgängige Ende-zu-Ende-Sicherheit (E2E), vorteilhaft genutzt werden. Wird die Kommunikation zwischen Client und Resource Server über die Cloud mithilfe von CoAP mit DTLS durchgeführt, müssen die Verbindungen Resource Server <-> Cloud und Cloud <-> Client separat verschlüsselt und die Daten in der Cloud ent- und wieder verschlüsselt werden. Demnach kann letztere Variante nicht als echte E2E-Verschlüsselung bezeichnet werden.

Theoretisch kann auch eine direkt Verbindung zwischen Client und Resource Server hergestellt werden ohne dabei den Umweg über die Cloud als Vermittler zu nutzen. In der Realität funktioniert dies spätestens seit der Einführung der Mobilfunktechnologie Long Term Evolution (LTE) nicht mehr. Ursache ist die Isolation von Mobilfunkgeräten im Internet durch private IP-Adressen. Eine Folge dieser Separation ist die Beschränkung auf einen Verbindungsaufbau vonseiten des Smartphones. Ein Gerät kann über das Internet keine Verbindung zu einem (LTE-)Smartphone aufbauen, da die Network Address Translation (NAT) des Mobilfunkanbieters dies verhindert.

Anstelle des separaten Client-Kontos für die Service Platform kann auch eine (partielle) Kopie der Autorisierungsdatenbank des SHGW in der Service Platform gespeichert werden. Dies vereinfacht den Fernzugriff, da kein zusätzliches Cloud-Konto notwendig ist. Jedoch kann es datenschutzrechtliche Bedenken geben, da nun wichtiges Schlüsselmaterial in der Cloud gespeichert wird.

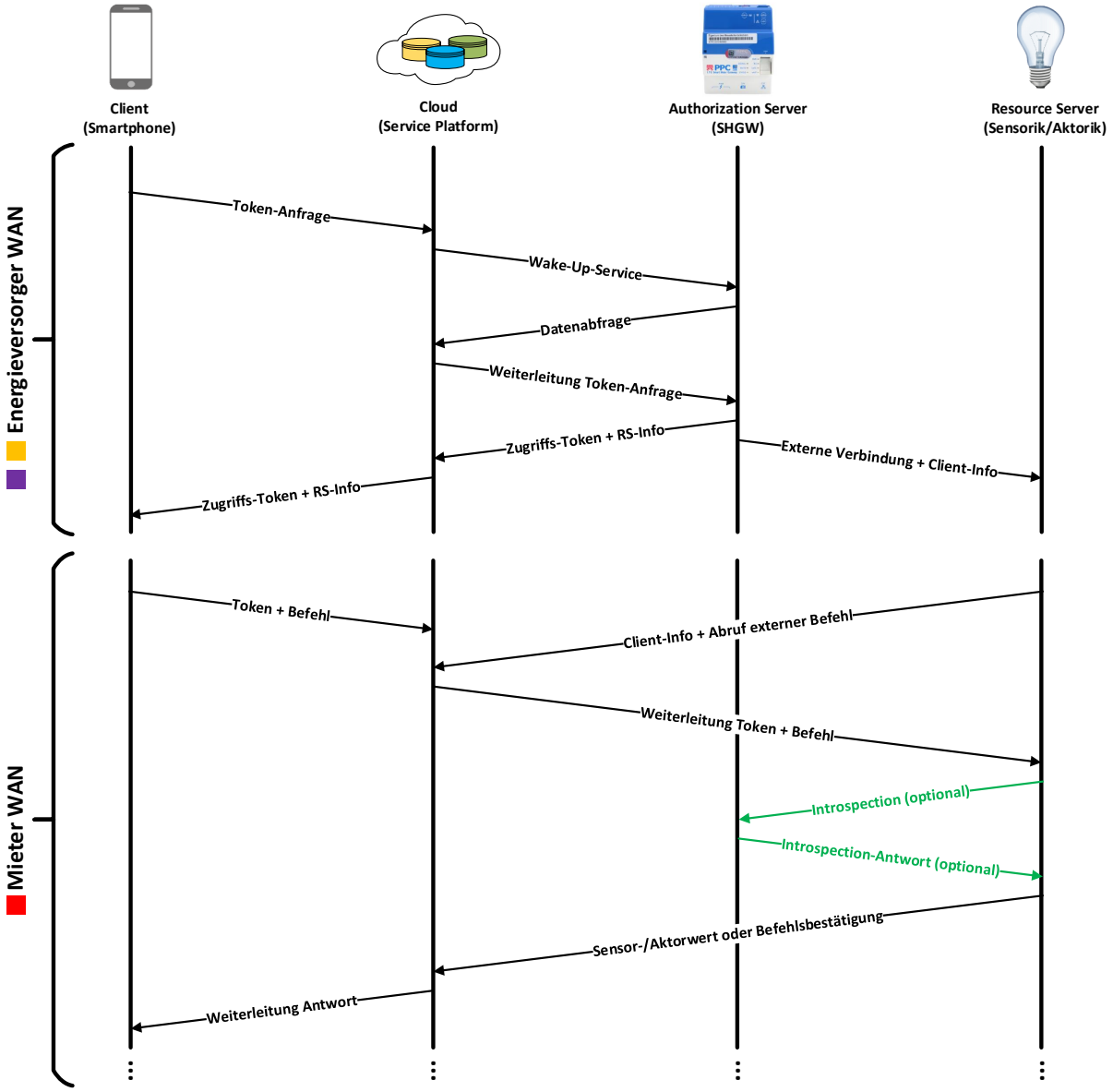


Abbildung 30: Fernzugriff auf Smart-Home-Geräte, [32]

Verbindungsaufbau von Smart Home-Gerät zu externem Nutzergerät

Dieses Szenario setzt voraus, dass Client und Resource Server bereits im lokalen Netzwerk eine Observe-Beziehung (allgemein auch Publish-Subscribe genannt) ausgehandelt haben. Diese Verknüpfung der beiden Geräte bewirkt, dass die Sensorik/Aktorik bei Wertänderungen oder Ereignissen eine Nachricht an den Client sendet. Die Autorisierung über Token findet nur während der Einrichtung dieser Beziehung statt. Danach wird die Autorisierung in Intervallen überprüft, welche von verschiedenen Parametern (z. B. Gültigkeitszeitraum des Tokens) abgeleitet werden können. Mithilfe der Prüfintervalle können unnötige Belastungen des SHGW mit Anfragen verringert werden.

In Abbildung 31 werden die Schritte für Verbindungsaufbau und Kommunikation zu einem externen Nutzergerät dargestellt. Tritt eine Wertänderung oder ein Ereignis ein, dass Teil einer Observe-Beziehung ist, wird geprüft, ob aufgrund des Intervalls für Autorisierungsprüfungen (nur bei Observe-Beziehungen) eine erneute Anfrage an den Authorization Server notwendig ist. Falls diese durchgeführt werden muss und das SHGW keine Berechtigungsänderung mitteilt, sendet die Sensorik/Aktorik eine Benachrichtigung mit dem neuen Wert oder dem eingetretenen Ereignis an den in der Observe-Beziehung definierten Client im lokalen Netzwerk. Antwortet der Client nach einer vordefinierten Zeitspanne (Timeout) und eventuellen Neuübertragungen nicht, wird angenommen, dass sich der Client nicht im lokalen Netzwerk befindet. Daraufhin wird die ausstehende Nachricht als externe Benachrichtigung zusammen mit Client-Informationen an die Service Platform über den WAN-Anschluss des Mieters geschickt. Die Cloud-Anwendung generiert mithilfe der Client-Informationen eine spezifische Notification für Smartphones und schickt diese über einen entsprechenden Notification-Service (abhängig vom Betriebssystem des Nutzergerätes) an den Client. Dieser nutzt die in der Notification enthaltenen Informationen sowie die Authentifizierungsdaten des Cloud-Kontos, um die Benachrichtigung des Resource Server von der Service Platform abzurufen. Erfordert die Benachrichtigung weitere Interaktionen, kann der Client mithilfe des Tokens, welcher während der Aushandlung der Observe-Beziehung generiert wurde, einen Befehl über die aufgebaute Verbindung an den Resource Server schicken. Falls der Token abgelaufen ist oder nicht die benötigten Berechtigungen für den auszuführenden Befehl besitzt, muss ein neuer Token vom SHGW angefordert werden. Dazu wird ein neuer Verbindungsaufbau, wie unter „Fernzugriff auf Smart Home-Geräte“ (Abbildung 30) beschrieben, durchgeführt.

Die Anwesenheitserkennung des Clients (Erreichbarkeit über lokales Netzwerk) kann auf unterschiedliche Weise detektiert werden. Neben dem hier dargestellten Timeout, kann auch der Router abgefragt oder der Client „angepingt“ werden. Diese Methoden sind jedoch u. a. abhängig von der genutzten Kommunikationstechnologie und sind daher implementierungsspezifisch.

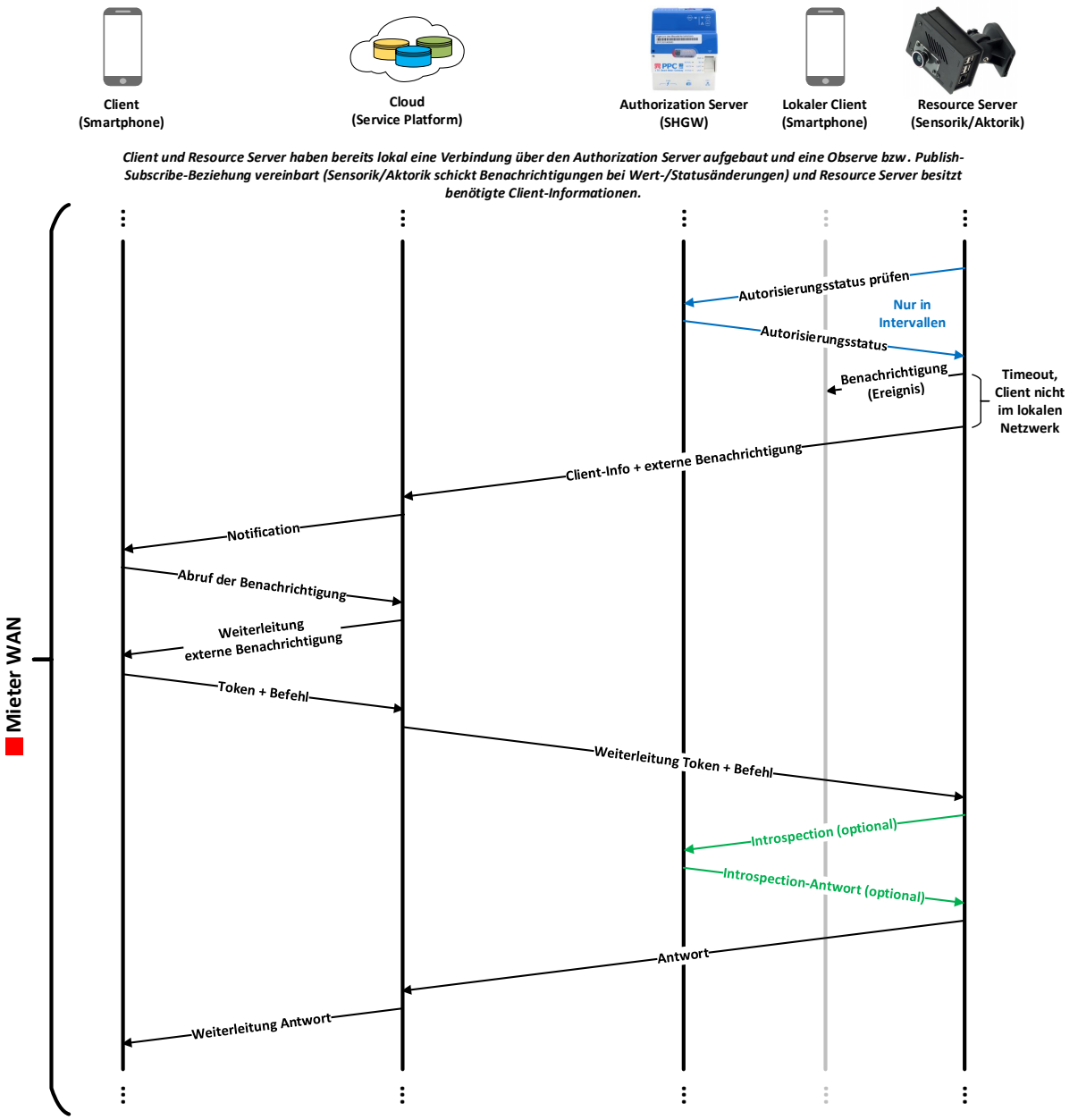


Abbildung 31: Verbindungsaufbau von Smart Home-Gerät zu externem Nutzergerät, [26]

4.5.4. Ergänzungen für eine beschleunigte Umsetzung

Erweiterung zur Realisierung der Anforderung an einen ausschließlichen Lesezugriff auf das Gateway

Ein reiner Lesezugriff auf das Gateway erfordert die Auslagerung der Autorisierungsdatenbank mit den Schlüssel-Informationen und Zugriffsberechtigungen aller Smart Home- und Nutzergeräte. Folglich kann die Datenbank nach Smart Home Geräten je Smart Home-Netzwerk respektive Wohneinheit partitioniert werden. Alle Datenbankeinträge einer Wohneinheit können, wie in Abbildung 32 dargestellt, auf ein dediziertes Gerät im privaten Netzwerk der Mieter oder eventuell auf direkt auf die Router ausgelagert werden. Die Datenbanken müssen dort verschlüsselt gespeichert und vor unbefugtem Zugriff (z. B. gegen Angreifer über WAN-Zugang des Mieters oder lokale Angreifer) geschützt werden. Eine wichtige Voraussetzung für diese Erweiterung ist die Einschränkung, dass nur

das Gateway Schreibrechte (Hinzufügen, Ändern, Löschen) in diesen externen Datenbanken hat. Folglich ist die Generierung von Token weiterhin Aufgabe des Gateways. Bereits generierte und in den externen Datenbanken abgespeicherte Token und Geräteinformationen können jedoch von Smart Home-Geräten direkt abgefragt werden. Neben der Wahrung der Anforderung an ausschließliche Lesezugriffe auf das Gateway, wird dieses zusätzlich entlastet. Diese Erweiterung wirkt sich folglich auch positiv auf die Bestrebung nach einem passiven Verhalten des Gateways bei HAN-zu-HAN-Kommunikation aus.

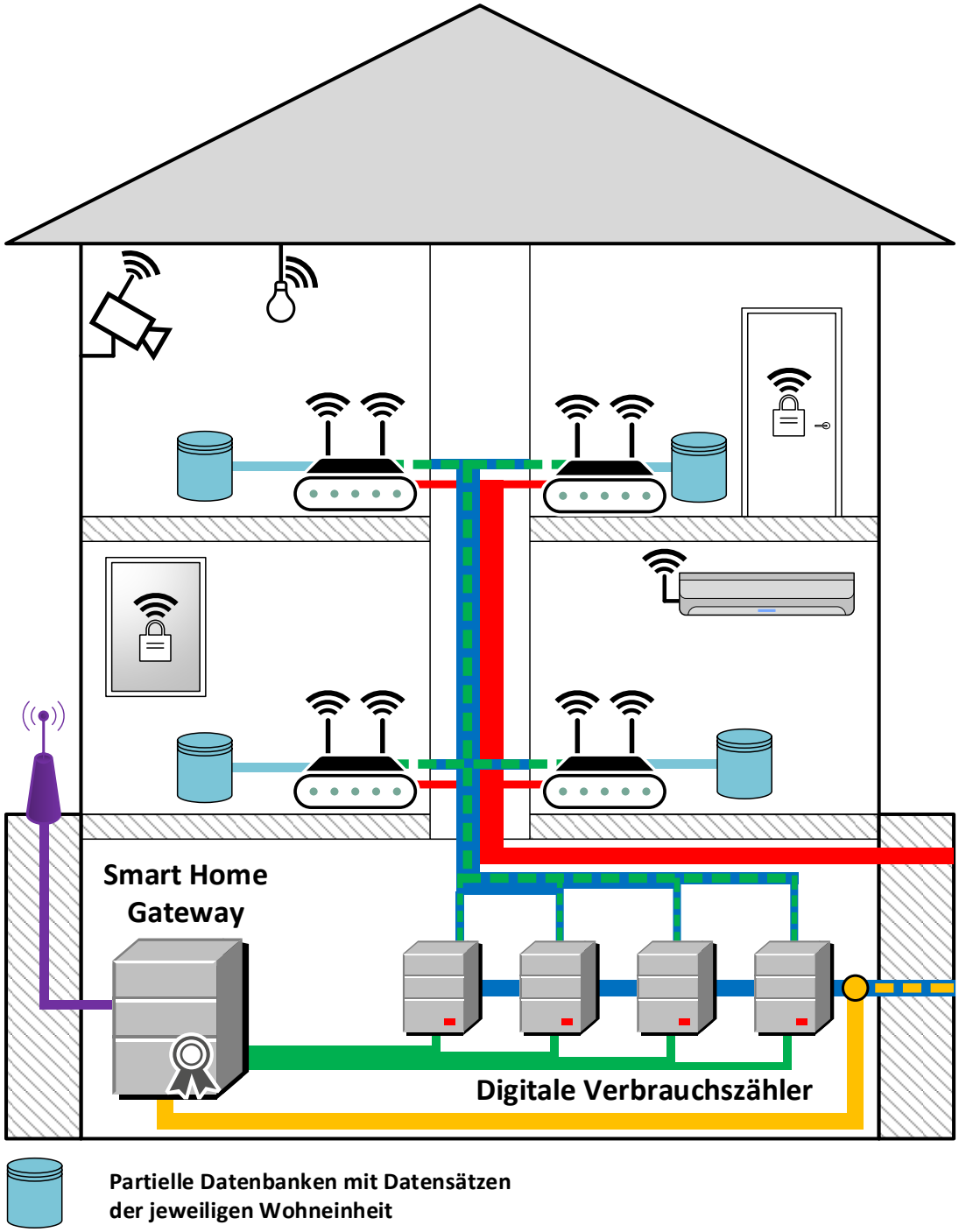


Abbildung 32: Erweiterung mit verteilten Datenbanken in Smart Home-Netzwerken

Ausgewählte Beispiele für eine beschleunigte Integration des Sicherheits-Framework in vorhandene Smart Home-Produkte

Grundsätzlich eignet sich das Sicherheits-Framework für eine zügige Integration und Verbreitung in der Praxis, da es nahezu[3] unabhängig von der verwendeten Funktechnologie ist.

Viele Smart Home-Produkte besitzen eine Möglichkeit für Firmware-Updates, um Fehler zu beheben und neue Funktionen hinzuzufügen. Abhängig von der verwendeten Hardware und der genutzten Funktechnologie, kann das Sicherheits-Framework bzw. dessen Implementierung als Basis für eine neue Firmware genutzt werden, die mithilfe des Update-Mechanismus auf die Geräte übertragen wird. Beispielsweise bietet die Firma Sonoff schaltbare WLAN Steckdosen an, die auf dem Mikrocontroller ESP8266 basieren. Dies ist der Vorgänger des in den Prototypen genutzten Mikrocontrollers ESP32. Für die Geräte dieser Firma sind daher kaum Anpassungen vorzunehmen, um sie mit dem Sicherheits-Framework auszustatten.

Eine weitere Möglichkeit ist die partielle Integration des Sicherheits-Framework in bestehende Lösungen durch eine Aktualisierung der zentralen Smart Home-Steuerungen, auch Hubs genannt. Diese Möglichkeit wird im Folgenden am Beispiel der Smart Home-Produkte des Unternehmens IKEA beschrieben. IKEA nutzt für die eigene Smart Home-Produktlinie TRÅDFRI eine Kombination aus CoAP- und ZigBee-Kommunikation. Das System besteht aus einem zentralen Hub, der mit einem Smartphone über WLAN mithilfe des Kommunikationsprotokolls CoAP gesteuert wird. Der Hub leitet die gesendeten Befehle über einen Zigbee-Adapter (spezifiziert Funktechnologie und Kommunikationsprotokoll) an die Smart Home-Geräte, wie z. B. LED-Lampen, weiter. Bei diesen Smart Home-Produkten kann, wie in Abbildung 33 dargestellt, durch ein Firmware-Upgrade des zentralen Hubs, die Interaktion des IKEA-Systems mit anderen Smart Home-Geräten abgesichert werden. Dabei muss jedoch die ZigBee-Implementierung als ausreichend sicher angenommen werden. Für den Übergangsprozess stellt diese Methode allerdings eine adäquate Lösung dar.

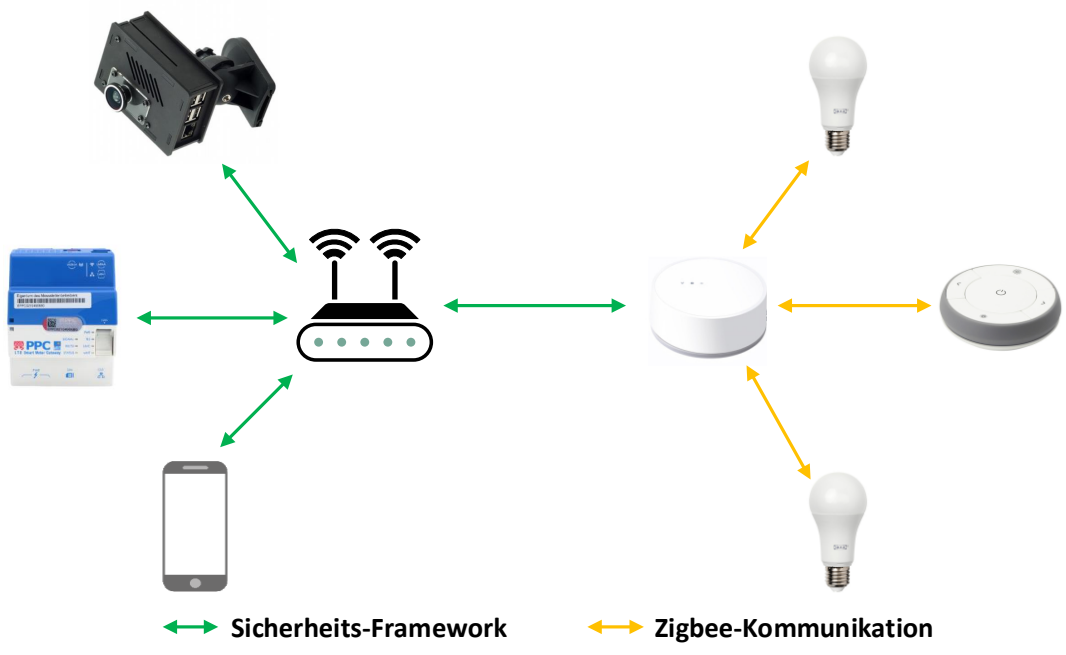


Abbildung 33: Partielle Integration des Sicherheits-Framework in IKEA TRÅDFRI, [32], [67]

[3] Proprietäre Funktechnologien erfordern eventuell protokollbedingt eine längere Integration

4.6. Erweiterungen auf Basis des Sicherheit-Framework

4.6.1. Dezentrales Regelmanagement

Ein Großteil der Smart Home-Systeme am Markt verwendet einen zentralen Hub, um die Kommunikation aller beteiligten Smart Home-Geräte zu verwalten und steuern. Ein Nutzer kann dabei Sensorik, wie z. B. Temperatur-, Wind- und Regensensoren benutzen, um in Abhängigkeit der gemessenen Werte Geräte zu steuern. Bei diesen Geräten handelt es sich u. a. um Aktorik, wie z. B. Fenster mit einem automatischen Kippmechanismus, Markisenantriebe oder Klimaanlagen. Der entscheidende Nachteil einer zentralisierten Lösung besteht darin, dass der Hub einen Single-Point-of-Failure darstellt. Bei einem Ausfall des Hubs fallen, wie in Abbildung 34 dargestellt, sämtliche Smart Home-Funktionen aus.

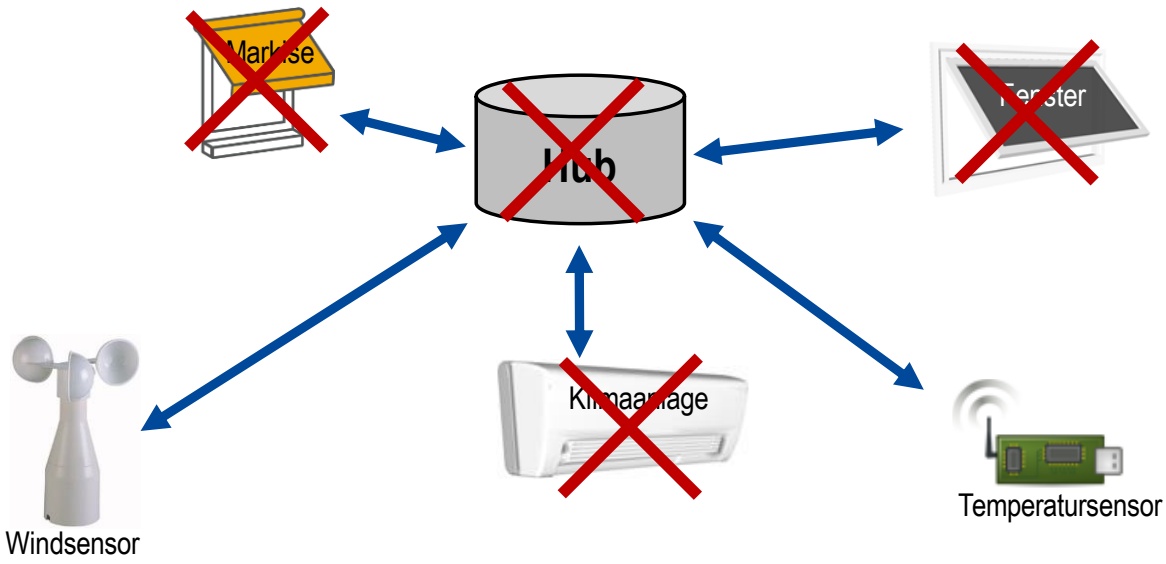


Abbildung 34: Regelmanagement mit zentralem Hub

Um diesem Problem zu begegnen, lässt sich die Logik zur Steuerung von Geräten dezentral integrieren. Abbildung 35 veranschaulicht diese dezentrale Steuerung. Im abgebildeten Szenario fällt die Klimaanlage aus. Die integrierte Regelsteuerung, bisher zuständig für die Absprache zwischen automatisiertem Fenster und Klimaanlage, ist daher ebenfalls nicht mehr funktionstüchtig. Die unabhängig gesteuerte Markise kann jedoch trotz des Defekts der Klimaanlage weiterhin im Smart Home Regelsteuerungen vornehmen. Dazu kann die Markise Sensordaten vom Windsensor abrufen (Request/Response) oder Änderungen abonnieren (Publish/Subscribe) und je nach Windbedingungen eingefahren werden. Diese Kommunikation lässt sich durch Protokolle des Sicherheits-Framework, wie OSCORE und ACE, absichern. Es ist jedoch möglich, dass zwei oder mehr Regeln eine Aktorik zur selben Zeit steuern, was zu einer Kollision führt. Insbesondere wenn diese Regeln auf verschiedenen Geräten ausgeführt werden, muss eine Synchronisierung aller Regeln stattfinden. Alle potenziell in Konflikt stehenden Regeln, da sie dieselbe Aktorik ansteuern, lassen sich durch Binary Decision Diagrams (BDDs) modellieren. Diese Darstellungsform ermöglicht ein automatisiertes Prüfen auf mögliche Kollisionen bei einer Testabdeckung von 100 %. Demnach kann die Abwesenheit von Fehlern nachgewiesen werden. Außerdem besteht die Möglichkeit die Ursache für die Fehlkonfiguration zu finden und den Nutzer darüber zu Informieren.

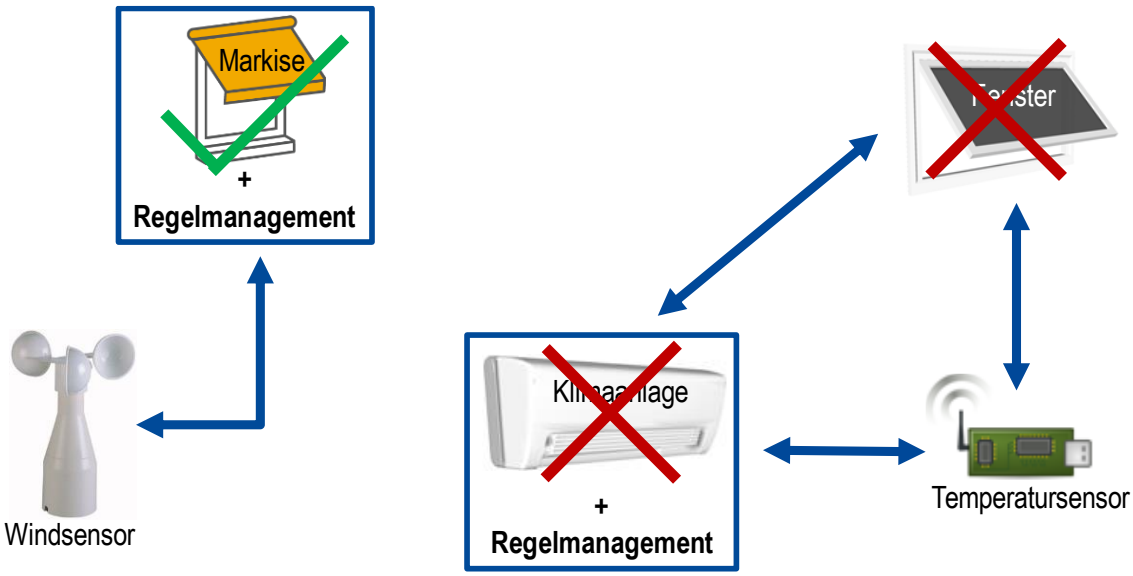


Abbildung 35: Dezentrales Regelmanagement

Im Folgenden werden mithilfe einer Zusammenfassung des Konzeptkapitels einer Masterarbeit [68] zum Thema dezentrales Regelmanagement die Funktionsweise einer solchen dezentralen Umsetzung und allgemeine Probleme von Konfigurationssystemen für Smart Home-Netzwerke beschrieben.

Konfigurationssysteme ermöglichen es Nutzern im Smart Home einen Konfigurationsdienst (Service) mit Sensorik und Aktorik zu verbinden, um gewünschte Aktionen durchzuführen. Ein Beispiel für eine solche Aktion ist die Steuerung einer Heizung. Diese soll beim Erreichen eines Minimalwertes automatisch eingeschaltet werden. Der Service benötigt hierzu die Sensordaten eines Thermometers, damit die Aktion ausgelöst werden kann. Abbildung 36 veranschaulicht dieses Beispiel. Der Nutzer erstellt die gewünschten Regeln auf dem Smartphone und schickt sie an den Service.

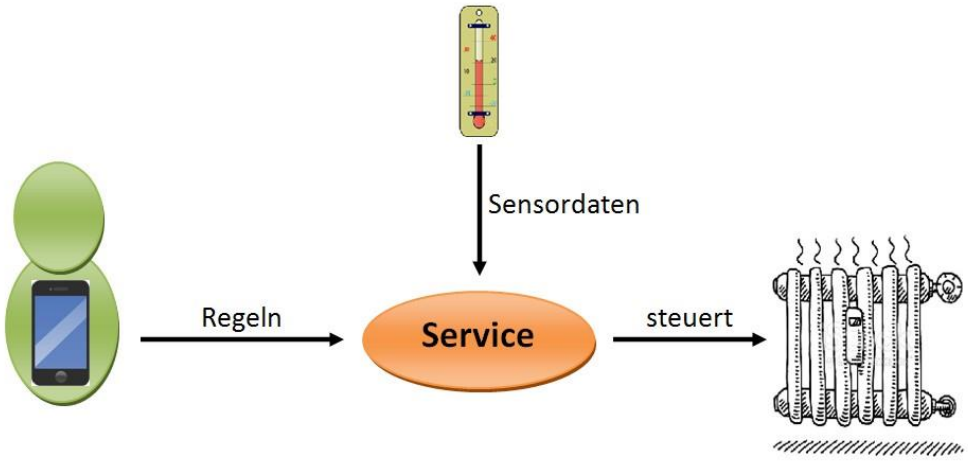


Abbildung 36: Automatisierung einer Heizungssteuerung

Rollen in Konfigurationssystemen

Es gibt vier Rollen, die in diesem Abschnitt erläutert werden. Diese sind grundlegende Komponenten, die durch Interaktion eine automatisierte Steuerung von Geräten ermöglichen.

Nutzer: Der Nutzer entwirft die Regeln und ist damit für die Einrichtung der Gebäudeautomation zuständig. Die Regeln werden über eine Smartphone-App erstellt. Weiterhin kann der Nutzer mit der App mehrere Geräte miteinander verknüpfen, um die Steuerung der Aktorik zu ermöglichen.

Service: Dieser Begriff beschreibt eine Einheit, die zur Ausführung einer Regel angelegt wird. Ziel ist es, eine Aktorik beim Eintreffen bestimmter Konditionen zu steuern.

Aktorik: Aktoren setzten elektrische Signale in mechanische Bewegung oder andere physikalische Größen, wie z. B. Temperatur, um. Folglich haben sie direkten Einfluss auf ihre Umgebung. Wenn die vom Nutzer definierte Regel ausgeführt wird, sendet der Service einen Befehl an den Aktor, um das gewünschte Ziel zu erreichen.

Sensorik: Die Sensorik dient zur qualitativen Erfassung von physikalischen und chemischen Größen, welche in elektrische Signale umgeformt werden. Physikalische Größen sind z. B. Temperatur, Luftfeuchtigkeit, Lichtstärke und Beschleunigung. Die Sensoren liefern die Werte für die Regelparameter der Regeln.

Regelstruktur

Bei den verwendeten Regeln handelt es sich um sogenannte „Trigger-Action Rules“. Eine Aktion wird ausgeführt (getriggert), wenn ein oder mehrere Werte von Parametern der Regel erreicht werden.

Syntax und Datenformat

Trigger-Action-Programming sticht gerade durch die Schlichtheit des Ansatzes hervor und ist daher besonders benutzerfreundlich. Auch unerfahrene Anwender können eine solche Programmierung schnell erlernen. Ein Beispiel dieses Programmierstils ist die IFTTT Umsetzung (If This Then That, [69]). Die Regel ist im Normalfall auf einen Trigger und eine Aktion begrenzt.

a) “If it’s 09:00 then turn the lights on”
b) “If it’s 09:00 and dark then turn the lights on”

In beiden Fällen sollen die Lichter eingeschaltet werden. Beispiel b besteht jedoch aus zwei Triggern (Parameter und Wert), die durch eine Konjunktion (AND-Verknüpfung) verbunden sind. An dem Beispiel ist zu erkennen, dass es vorteilhaft ist boolesche Operatoren in die Regelsyntax miteinzubeziehen. Um jedoch die Komplexität gering zu halten, ist es ratsam nur AND und OR zu benutzen. Eine allgemeine mathematische Darstellung der Regelsyntax kann nach [29] wie folgt beschrieben werden:

$$OPERATION := \{=, \neq, >, \geq, <, \leq\} \tag{1}$$

$$f : PARAMETER \rightarrow Z \tag{2}$$

$$TRIGGER := \{PARAMETER \times OPERATION \times Z\} \tag{3}$$

$$BOOLOPERATOR := \{AND, OR\} \tag{4}$$

$$Regel := P(TRIGGER) \times BOOLOPERATOR \tag{5}$$

Die Parameter sind physikalische Größen und Z ist der Wert, der erreicht werden muss, um eine Aktion zu triggern. Eine Regel könnte beispielsweise wie folgt aussehen:

```
LIGHT < 80 AND TIME > 9:00 AND TIME < 18:00
```

An dieser Stelle ist hervorzuheben, dass eine Regel nur einen Typ von booleschen Operatoren beinhalten darf. Um eine Regel einfacher zu gestalten ist es auch möglich zwei Regeln zu entwerfen anstatt zwei Ereignisse mit einem OR zu verbinden. Zu der eigentlichen Regelsyntax muss dem Service außerdem noch übergeben werden, auf welche Aktorik zugegriffen werden soll. Diesbezüglich muss definiert werden, ob der Aktor aus- oder eingeschaltet werden soll. Im späteren Verlauf wird noch ein Parameter eingeführt, der die Prioritäten von Regeln definiert. Mit diesen zusätzlichen Eigenschaften und der bisherigen Syntax wird eine Regel erstellt, die wie folgt aussehen könnte:

```
TIME > 14:00 AND TIME < 14:10 = 1, 2
```

In diesem Fall soll das Fenster zwischen 14:00 und 14:10 Uhr geöffnet sein. Die 2 nach dem Komma stellt die Priorität der Regel dar. Die 1 entspricht der internen ID der zu steuernden Aktorik. Die Regel wird mithilfe folgender Syntax in die Datenbank aufgenommen:

```
PARAMETER OPERATION Z = AKTOR_ID, PRIO_ID
```

Ein boolescher Operator kann zudem noch mehrere Trigger-Anweisungen verbinden. Die Nummerierung der Aktoren (`Aktor_ID`) muss vorher festgelegt werden, eventuell basierend auf der Reihenfolge der Geräteregistrierung im Smart Home-System. Um eine Regel abzuspeichern, die das Fenster schließen soll, wird vor die Aktorziffer ein Ausrufezeichen gesetzt (`!1`). Zur Auswertung der Regel wird diese in einzelne Teile zerlegt und separat abgespeichert. Die Aufteilung der Regel in ihre Bestandteile dient später zur Überprüfung ihrer logischen Korrektheit und dient zum Vergleich der Regeln untereinander.

Regelvorschriften

Ein wesentlicher Punkt ist die Konsistenz der Regel. Es müssen Vorschriften definiert werden, die eine einheitliche Regelstruktur zur Folge haben. Da es sich in diesem Konzept um ein IoT-Szenario handelt, muss eine Lösung erstellt werden, die einen geringen Kommunikationsaufwand und eine kleinstmögliche Komplexität erfordert. Zunächst wurde der Ansatz diskutiert, dass eine Regel mit einer Definition sowohl zum Einschalten der Aktorik als auch zum Ausschalten dient. Dies bedeutet, dass die Aktorik bei Gültigkeit der definierten Trigger eingeschaltet und sonst ausgestellt wird. Diese Umsetzung beinhaltet jedoch das Problem, dass der Service zu jedem Zeitpunkt über den Zustand der Aktorik informiert sein muss. Da eine Gebäudeautomation eine große Anzahl an Regeln beinhalten kann, ist es sehr wahrscheinlich, dass Regeln miteinander kollidieren und sich gegenseitig aufheben können. Wenn beispielsweise eine Regel definiert wird, die für einen Zeitraum von 10 Minuten die Aktorik einschalten soll, bedeutet dies außerdem, dass die Aktorik für die restliche Zeit durch die Regel ausgeschaltet wird. Es könnte auch der Fall eintreten, dass die aktuelle Regel durch die Ausführung einer anderen unterbrochen wird, aber danach wieder wirksam werden muss. Dies kann nur über eine kontinuierliche Kommunikation zwischen Aktorik und Service sichergestellt werden. Der Kommunikationsaufwand einer solchen Umsetzung ist zu groß, als dass sie für ein IoT-Szenario in Frage kommt. Daher wurde ein Ansatz gewählt, der nur eine Zustandsänderung der Aktorik durch eine Regel zulässt. Durch diesen Ansatz wird das Ein- und Ausschalten voneinander getrennt. Es entstehen dadurch keine Wertebereiche mehr, währenddessen die Regel gültig ist. Die Regel im vorherigen

Beispiel (`TIME > 14:00 AND TIME < 14:10 = 1, 2`) muss also in zwei Regeln aufgeteilt werden. Eine zum Einschalten und eine zum Ausschalten der Aktorik, um die Definition eines Zeitraumes zu verhindern. Eine Trennung der Regel ergibt die folgenden zwei Einzelregeln:

```
TIME > 14:00 = 1, 2
TIME > 14:00 = !1, 2
```

Es entstehen zwei Regeln, die den Zustand der Aktorik nur einmalig ändern können. Ein Fenster kann beispielsweise mit der ersten Regel geöffnet und mit der zweiten geschlossen werden. Damit eine Regel nicht in einem Wertebereich gültig ist, darf der gleiche Parameter nur ein einziges Mal in einer Regel verwendet werden. Unterschiedliche Parameter dürfen aber durchaus miteinander verknüpft werden. Ein Beispiel einer Regel, die das Eintreffen mehrerer Ereignisse überprüft, könnte wie folgt aussehen:

```
TEMP > 25 AND LIGHT > 80 AND TIME > 12:00 = 1, 2
```

Die Aktorik wird eingeschaltet, wenn alle Ausdrücke wahr sind. Der wesentliche Teil der Regel ist jedoch der Zeitparameter (TIME). Die Aktion wird nur einmal am Tag getriggert. Der Operator > ist hierbei mit einem = gleichzusetzen. Es handelt sich hierbei um eine Vereinfachung. Es erleichtert die Trennung der Regel in ihre einzelnen Elemente, da das Gleichheitszeichen in der Regelsyntax bereits verwendet wird. Die anderen Trigger enthalten Sensorwerte. Diese können sich im Gegensatz zur Uhrzeit sprunghaft verändern (Sonderfall der Zeitangabe bei Wechsel von 23:59 auf 0:00 Uhr ausgenommen). Folglich steuert die Regel die Aktorik, wenn es 12 Uhr ist, die Temperatur zu diesem Zeitpunkt einen Wert über 25 Grad Celsius besitzt und die Lichtstärke einen Wert von über 80 Lux annimmt. Anders verhält es sich jedoch, wenn beispielsweise der Parameter Temperatur (TEMP) in der Regel als einziger Trigger definiert wurde. Wichtig ist dabei, wie die Operation gewählt wurde:

```
TEMP > 25 = 1, 2
TEMP < 24 = !1, 2
```

Das Ereignis von der ersten Regel beschreibt den Fall, dass die Temperatur von unter 25 Grad auf einen Wert von über 25 Grad steigt. Das bedeutet, dass die Aktion ausgeführt wird bzw. die Regel in Kraft tritt, wenn ein Temperaturwert von unter 25 Grad aufgenommen worden ist und der darauffolgende Wert eine Temperatur von über 25 Grad aufweist. Die Aktion wird nur einmal ausgelöst. Es kann der Fall auftreten, dass solche Sprünge kurz danach wiederholt auftreten. Diese Fälle müssen in einer Implementierung berücksichtigt werden. Des Weiteren ist es erforderlich, dass die zweite Regel, die für das Ausschalten der Aktorik zuständig ist, nicht zu nahe an dem Definitionsraum des Triggers der ersten Regel liegt. Dies könnte sonst zu einem wechselseitigen Auslösen der Regeln führen, was eine unnötige Belastung Aktorik zur Folge hätte.

Regelkonflikte

Ein wesentlicher Bestandteil dieser Arbeit ist die Detektion von Konflikten der aufgestellten Regeln. Dies bedeutet, dass die Regeln sich nicht gegenseitig behindern (Kollision) und nicht mehrfach definiert (Redundanz) werden dürfen. Es gibt drei verschiedene Arten von Fehlern, die durch Regeln ausgelöst werden können: logische Fehler, interne Kollisionen und externe Kollisionen.

Logische Fehler

Eine Regel beinhaltet einen logischen Fehler, wenn die Voraussetzungen für die Ausführung einer Aktorik nie erreicht werden können. Diese Fehler können häufig durch Tippfehler verursacht werden. Gerade bei der Ausführung einer Aktion, die von mehreren Inputwerten abhängig ist, kann solch ein Fehler auftreten. Es ist zu vermeiden, dass die Regeln zu jeder Zeit wahr (Tautologie) bzw. falsch (Kontradiktion) sind. Zwei Beispiele sollen dies verdeutlichen:

a) Fenstersteuerung öffnen: `TEMP > 250 = 1, 2`
b) Lichtsteuerung: `LIGHT > 80 AND LIGHT < 60 = 3, 2`

In Beispiel a wurde ein Wert für den Temperatursensor übergeben, der unter normalen Bedingungen nicht eintreffen kann. Dies bedeutet, dass die Regel nie ausgeführt wird. Solche Eingabeprobleme müssen in einer Implementierung weitestgehend abgefangen werden, erfordern jedoch auch eine semantische Beschreibung von Parametern, welches ein eigenes Forschungsgebiet ist.

Beispiel b zeigt eine logische AND-Verknüpfung zweier Lichtstärken. Sie bilden jedoch einen Widerspruch zueinander und die Lampe wird somit nie eingeschaltet. Auch in diesem Fall wird der Fehler deutlich, da die Regel nie einen wahren Wert annehmen kann. Dieser Fall ist jedoch schon durch die Regelvorschrift, dass ein Parameter nicht mehrfach verwendet werden kann, behoben. Es muss also durch die Regelprüfung erkannt werden, dass der Nutzer fälschlicherweise einen Wertebereich geschaffen hat, indem er mehrfach den gleichen Parameter zur Regeldefinition verwendet hat.

Interne Kollisionen

Zu einer internen Regelkollision kommt es, wenn sich zwei oder mehr Regeln auf einem Service gegenseitig behindern. So wird durch die eine Regel zu einem Zeitpunkt die Aktorik ausgeschaltet, während die andere Regel gleichermaßen diese aktivieren möchte. Zudem muss auch gewährleistet werden, dass eine Regel nicht mehrfach definiert wird. Weiterhin bedarf es einer Unterscheidung zwischen möglichen Kollisionen und garantierten Kollisionen. Zwei Regeln wirken in jedem Fall gegeneinander, wenn sie durch gleiche Trigger definiert wurden, aber den Zustand der Aktorik entgegengesetzt verändern möchten. Hierzu ein Beispiel:

```
a) TIME > 14 = 1, 2
b) TIME > 14 = !1, 2
```

Beide Aktionen werden im gleichen Moment ausgelöst. Dabei kollidieren die Regeln, da sie auf die gleiche Sensorik zugreifen. Während Regel a die Aktorik einschaltet, wirkt Regel b dem entgegen und möchte dieselbe Aktorik ausschalten. Diese Regelkollision wird garantiert stattfinden. Anders verhält es sich, wenn zwei Regeln verschiedene Parameter enthalten. Unter bestimmten Umständen, kann es passieren, dass die getriggerten Ereignisse beider Regeln gleichzeitig eintreten und die Aktionen sich gegenseitig behindern. Ein Beispiel für ein solches Szenario könnte wie folgt aussehen:

```
a) TEMP > 20 = 1, 2
b) TIME > 14 = !1, 2
```

Die Wahrscheinlichkeit, dass beide Regeln **im gleichen Moment** schalten ist, relativ gering. Wenn jedoch von einer großen Anzahl an Regeln ausgegangen wird, könnte solch ein Fall durchaus eintreffen. Daher ist es wichtig, dass durch eine Prüfung die Möglichkeit erkannt und ausgewertet wird.

Externe Kollisionen

Wie bei der internen Kollision widersprechen sich in diesem Fall zwei oder mehrere Regeln. Jedoch befinden sich diese Regeln auf verschiedenen Endgeräten bzw. Diensten. In diesem IoT-Szenario müssen somit alle Regeln auf den unterschiedlichen Diensten berücksichtigt werden. Es muss eine Kommunikation zwischen den Konfigurationsdiensten stattfinden, die eine Überprüfung der Regeln ermöglicht.

Regelprüfung

In einem ersten Schritt müssen alle Arten von Regelkonflikten detektiert werden. Anschließend ist eine Verarbeitung der Konflikte notwendig. Das bedeutet zum einen, dass durch den Nutzer verursachte logische Fehler erkannt, analysiert und diesem mitgeteilt werden müssen. Zum anderen bedarf es der Detektion von Regelkollisionen. Zwei oder mehr Regeln dürfen nicht gleichzeitig auf eine Aktorik zugreifen. Dabei ist es wichtig, dass sowohl die Regeln auf einem Konfigurationsdienst nach Kollisionen überprüft werden und als auch die von anderen Diensten im Netzwerk, die auf die gleiche Aktorik Zugriff haben. Folglich durchläuft die Überprüfung drei Schritte:

- Detektion logischer Fehler
- Detektion interner Kollisionen
- Detektion externer Kollisionen

Detektion logischer Fehler

Zunächst müssen logische Fehler behoben werden. Um dies durchzuführen, müssen Wertebereiche überprüft werden. Bei „Trigger-Action-Rules“ wird eine Aktion ausgelöst, sobald ein voreingestellter Wert erreicht wird. Beispielsweise haben Temperatur und Lichtstärke nur bestimmte Wertebereiche, die für eine Gebäudeautomation sinnvoll sind, unabhängig vom darstellbaren Wertebereich eines Parameters. Weiterhin stellt die Uhrzeit als Parameter einen Sonderfall dar. Sie ist eine periodische Größe, die Werte von 0:00 bis 23:59 annimmt und dann nach einer sprunghaften Wertänderung wieder von vorne beginnt. Diese Besonderheit muss während einer Implementierung berücksichtigt werden. Mit der Anforderung an die Regelsyntax Wertebereiche bzw. Zeiträume zu vermeiden, kann auch die Problematik des Wertesprungs (23:59 auf 0:00) umgangen werden. Allerdings muss die Regelprüfung die mehrfache Verwendung von Triggern in einer Regel überprüfen. Inwieweit diese Prüfung bereits während der Eingabe der Regel durch die App oder im Nachhinein geprüft wird, kann je nach Implementierung variieren.

Detektion interner Kollisionen

Der nächste Schritt der Regelprüfung besteht in der Behebung von Kollisionen. Es muss verhindert werden, dass sich Regeln, die die gleiche Aktorik steuern, gegenseitig behindern. Um den Vergleich der Regeln zu vereinfachen, werden Prioritäten eingeführt, die eine Wichtung der Regeln ermöglicht. Es handelt sich dabei um ein Attribut, das für jede Regel vergeben wird. Der Nutzer kann dem Regelsystem übergeben, ob es sich um eine Sicherheitsregel (Priorität 1), eine Energieregel (Priorität 2) oder eine Komfortregel (Priorität 3) handelt. Ein Beispiel soll den Zweck dieser Umsetzung verdeutlichen:

Priorität 1: Schließe das Fenster, wenn sich niemand in der Wohnung befindet.
Priorität 2: Schließe das Fenster, wenn die Klimaanlage läuft.
Priorität 3: Öffne das Fenster, wenn die Innentemperatur größer als 24 Grad ist.

Alle drei Regeln betreffen das Fenster. Durch die angegebene Priorität ist jedoch eine klare Rangfolge gegeben. Solange die Regel mit der höchsten Priorität ausgeführt wird, haben andere Regeln keinen Zugriff auf die Aktorik. Die höchste Priorität wird an Regeln vergeben, die direkt die Sicherheit der Wohnung betreffen. Sie dienen zum Schutz der Wohnung und haben daher immer Vorrang. Der Nutzer kann jedoch selbst definieren, welche Prioritäten er den Regeln zuteilen möchte. Die zweite Priorität wird hauptsächlich für die Reduzierung des Energieverbrauchs benutzt. Im Beispiel wird das Fenster nicht geöffnet, wenn die Klimaanlage angestellt ist. So könnte man auch eine Regel mit dieser Priorität definieren, die aussagt, dass das Licht ausgestellt wird, sobald sich niemand mehr in dem Raum befindet. Priorität 3 umfasst Regeln, die dem Komfort dienen. Wie schon erwähnt, wird die Priorität in der Regelsyntax berücksichtigt.
Die Attributverteilung von Prioritäten ist jedoch nur eine Voreinstellung, welche die Regelprüfung erleichtern soll. Die Aufgabe einer Regelprüfung ist es, Regeln mit gleicher Priorität zu vergleichen, die auf dieselbe Aktorik zugreifen möchten. Um dies durchzuführen, werden die Regeln als binäre Entscheidungsdiagramme (BDDs, siehe Abschnitt 3.4.3) dargestellt. Die Idee dahinter besteht darin, dass diese Struktur genutzt werden kann, um den Vergleich von Regeln zu automatisieren. Wie in den Grundlagen schon erläutert wurde, besteht ein BDD aus Knotenpunkten, die entweder einen wahren (1) oder einen falschen Wert (0) annehmen können. Da Regeln aus aussagenlogischen Ausdrücken und booleschen Operatoren bestehen, ist der Einsatz eines BDDs zur Auswertung geeignet. Eine weitere Möglichkeit eines Entscheidungsverfahrens, welches sich für aussagenlogische Anwendungen eignet, ist die Verwendung von Wahrheitstabellen. Es wurde sich jedoch für den Einsatz von BDDs entschieden, da die zur Verarbeitung benötigte Datenmenge und der Vergleichsaufwand geringer ausfallen. Wie im Grundlagenteil beschrieben, wird die Größe der Entscheidungsdiagramme über ihre Anzahl der Knoten beschrieben. Die maximale Anzahl der Knoten beträgt $2n - 1$. Die Größe einer Wahrheitstabelle beträgt $2n$, wobei n die Anzahl der Eingangsgrößen ist. Der Vorteil besteht in der Reduzierbarkeit des BDD. Durch Eliminierungs- und Verschmelzungsregeln kann ein BDD deutlich reduziert werden (im Vergleich zu einer Wahrheitstabelle einer Regel). Dies ist besonders bei einer hohen Anzahl an Eingangsgrößen von Vorteil. Die geringere Größe hat einen positiven Einfluss auf Speicherauslastung und Laufzeit der Regelprüfung auf den Geräten.
Die Umwandlung einer Regel in ein binäres Entscheidungsdiagramm wird über die Substitution der einzelnen Ausdrücke (Trigger) durch Variablen erreicht. Identische Ausdrücke werden dabei der gleichen Variable zugeordnet. Zum Vergleich von zwei Regeln werden diese in BDDs umgewandelt und mithilfe einer XOR-Verknüpfung verbunden. Weiterhin werden Ähnlichkeitsgrenzen berücksichtigt. Wenn ein Ausdruck (Trigger) zweier Regeln identisch ist und sich die Trigger-Werte nur minimal unterscheiden, werden die Ausdrücke durch die gleiche Variable substituiert. Ein Beispiel soll dies verdeutlichen:

```
a) TIME     > 14:00     AND athome  = 1, 2
b) TIME     > 14:05     AND athome  = 1, 1
c) TEMP     > 25        AND !athome = !1, 2
```

Zunächst werden die Regeln in BDDs umgewandelt. Die Ausdrücke werden dabei substituiert. X_1 entspricht `TIME > 14:00` und `TIME > 14:05` aufgrund des ähnlichen Trigger-Wertes ebenfalls. X_2 ersetzt `athome`. Beide Variablen werden über eine AND-Verknüpfung verbunden. `TEMP > 25` wird

ebenfalls durch die Variable X_3 substituiert. Abbildung 37 stellt die Regeln a, b und c als BDDs dar. Wie zu erkennen ist, sind durch die Ähnlichkeits-Substitution die Regeln a und b identisch. Wenn Regel b einem bestehenden Konfigurationssystem mit Regel a hinzugefügt wird, teilt die Regelprüfung dem Nutzer mit, dass die neue Regel bereits vorhanden ist. Die Priorität spielt dabei keine Rolle. Wichtig ist nur, dass beide Regeln dieselbe Aktorik in den gleichen Zustand versetzen möchten (hier: einschalten).

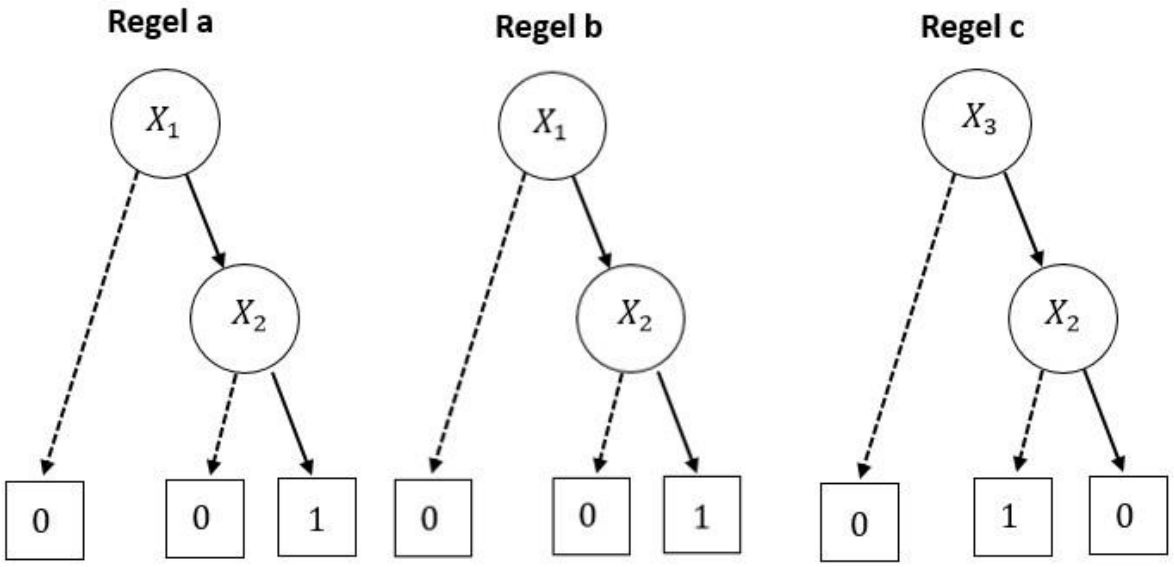


Abbildung 37: Entscheidungsdiagramme (BDDs) der Beispielregeln

Zur Prüfung auf Kollisionen zweier Regeln wird die XOR-Verknüpfung verwendet. Sind die BDDs identisch, besitzen die Enden aller Zweige des resultierenden (XOR-)BDD den Wert 0. Anhand des Vergleichs von Regel a und c wird die XOR-Verknüpfung im Folgenden veranschaulicht. Die XOR-Verknüpfung in Abbildung 38 besitzt auch wahre Ausgänge („1“). Da die XOR-Verknüpfung nur wahr

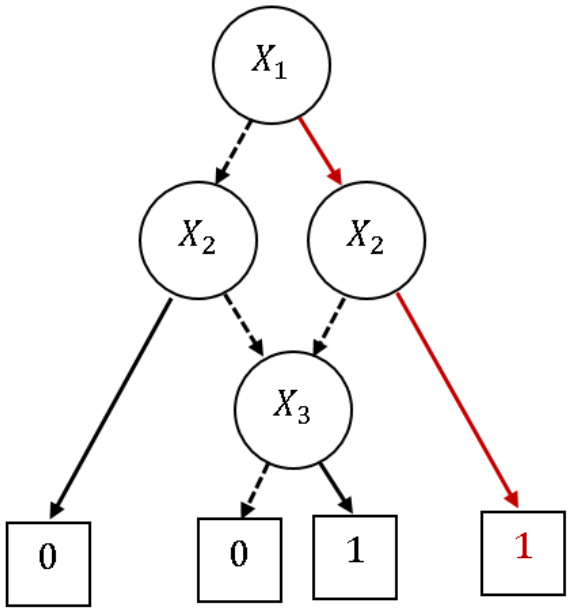


Abbildung 38: XOR-BDD der Bespielregeln a und c

(„1“) wird, falls sich die verknüpften Ausdrücke unterscheiden, bedeutet dies, dass die Regeln unter den Bedingungen dieser Zweige nicht kollidieren würden. An dieser Stelle kann bei der Regelprüfung eine Vereinfachung vorgenommen werden. Eine Regel wird nur dann aktiviert, falls alle Trigger der

Regel ausgelöst werden. Folglich müssen alle Variablen des Regel-BDD den booleschen Wert wahr („1“) annehme damit die Regel insgesamt den Wert wahr („1“) annehmen kann. Daher wird während der Regelprüfung ausschließlich der rechte Pfad bzw. Zweig des BDDs (rot markiert) in Abbildung 38 für die Auswertung berücksichtigt. Der Ausgang dieses Pfades hat den booleschen Wert wahr („1“) angenommen. Folglich kann keine Kombination von Sensorwerten für die Regeln a und c gefunden werden, die bei Berücksichtigung aller Trigger der Regeln, einen gleichzeitigen Zugriff auf die Aktorik ermöglicht. Die Regeln können daher nicht miteinander kollidieren. Dementsprechend tritt eine garantierte Kollision ein, wenn der Ausgang des rechten Pfades des XOR-BDD den Wert falsch („0“, „kein Unterschied vorhanden“) annimmt. Beide Regeln müssen dazu die gleiche Priorität besitzen und gegeneinander wirken.

Detektion externer Kollisionen

Da die Regeln dezentral von verschiedenen Diensten ausgeführt werden, können externe Kollisionen entstehen. Diese Konflikte müssen auch über das erläuterte Verfahren der Kollisionsbehebung gelöst werden. Dabei muss eine neue Regel mit allen Regeln im Konfigurationssystem des Smart Home vorhandenen Regeln verglichen werden. Um dies durchzuführen, wurden zwei Ansätze erarbeitet. Die Regelprüfung kann entweder über eine zentrale Instanz oder aber dezentral durch die Integration der Regelprüfung auf jedem Konfigurationsdienst durchgeführt werden.

Zentraler Ansatz

In dieser Variante wird ein zentrales Regelmanagement verwendet, um die einzelnen Regeln zu verwalten. Der Nutzer definiert Regeln und übergibt, wie in Abbildung 39 dargestellt, diese mittels Smartphone App an das zentrale Regelmanagement. Daraufhin wird die Regel geprüft und nach erfolgreicher Prüfung an einen geeigneten Konfigurationsdienst im Netzwerk weitergeleitet.

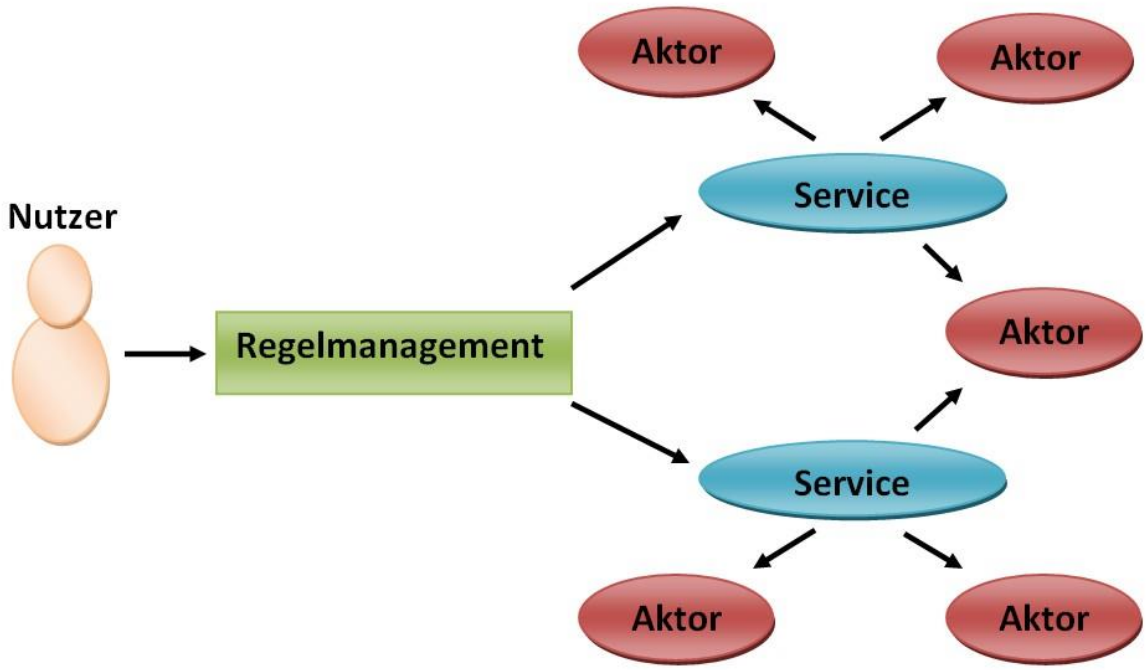


Abbildung 39: Ansatz mit zentralem Regelmanagement

Der Vorteil dieses Ansatzes besteht in der Schlichtheit seiner Umsetzung. Dem System wird nur eine Komponente hinzugefügt und auf diesem zentralen Regelmanagement verbleibt eine Kopie aller Regeln im System. Es ist daher keine Kommunikation zwischen den Diensten während der Regelprüfung notwendig, um alle Regeln abzurufen. Gleichzeitig werden die Dienste nicht mit dem hohen Aufwand der Regelprüfung belastet. Das zentrale Regelmanagement wird nur zum Hinzufügen, Ändern oder Löschen von Regeln im System benötigt. Zur Ausführung der Regeln werden nur die Konfigurationsdienste benötigt. Tritt ein defekt am zentralen Regelmanagement auf, hat dies nur zur Folge, dass dem System keine neuen Regeln übergeben oder Änderungen an bestehenden Regeln durchgeführt werden können.

Dezentraler Ansatz

Im Gegensatz zur zentralen Variante werden die vom Nutzer definierten Regeln direkt an die Konfigurationsdienste (Services) mittels Smartphone App geschickt. Abbildung 40 stellt diese dezentrale Variante dar.

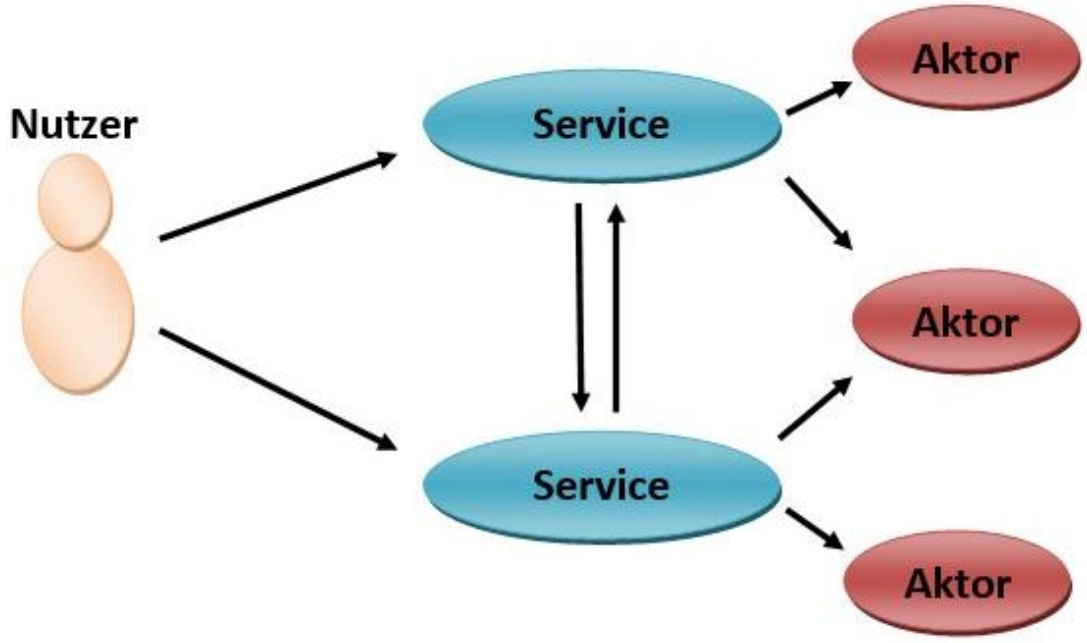


Abbildung 40: Ansatz mit verteiltem Regelmanagement

Da es kein zentrales Regelmanagement gibt, muss jeder Service eine eigene Regelprüfung durchführen. Folglich muss nun jeder Dienst neben der Regelausführung zusätzliche Aufgaben verarbeiten. Weiterhin wird die Komplexität der Regelprüfung erhöht. Die Dienste müssen miteinander kommunizieren, um Regeln auszutauschen, die potenziell mit der neuen Regel kollidieren könnten. Im Fall eines logischen Fehlers oder einer internen Kollision verläuft die Detektion wie im zentralen Ansatz. Der wesentliche Unterschied besteht in der Analyse von externen Kollisionen. Konfigurationsdienste, die Regeln für dieselbe Aktorik besitzen, müssen miteinander kommunizieren und ihre Regeln austauschen, um Kollisionen zu vermeiden. Bei der Erstellung einer neuen Regel muss der Dienst, der die neue Regel erhält, alle relevanten Regeln von anderen Diensten über den Regeltransfer abrufen, zusammenführen und überprüfen.
Der Regeltransfer ermöglicht einen Austausch der Regeln zwischen den Diensten. Dazu legt jeder Dienst eine Tabelle für jede Aktorik an, die von ihm gesteuert werden soll. Fordert ein Dienst im

Konfigurationssystem alle Regeln an, die eine bestimmte Aktorik steuern, schicken die anderen Dienste diese Tabellen als Antwort zurück. Der Ablauf dieser Kommunikation wird in Abbildung 41 dargestellt.

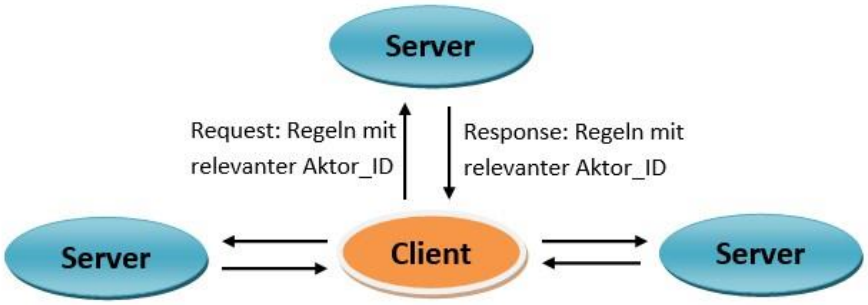


Abbildung 41: Regeltransfer beim dezentralen Regelmanagement

Im Vergleich zum zentralen Ansatz bietet diese Variante einen Vorteil bei der Ausfallsicherheit, da die Konfigurationsdienste unabhängig voneinander arbeiten können und somit der Ausfall eines Gerätes keinen Einfluss auf den Rest des Konfigurationssystems hat.[4] Das Fehlen einer zentralen Einheit spricht grundsätzlich für eine bessere Skalierbarkeit des Systems im Vergleich mit dem zentralen Ansatz.

4.6.2. ANTs

Unabhängig von der Sicherheit von DTLS, OSCORE und ACE können Angreifer einzelne Geräte durch Ausnutzen von Software- und Hardwareschwachstellen kompromittieren, um Angriffe innerhalb des lokalen Smart Home-Netzes auszuführen. Die OWASP (Open Web Application Security Project) [10] erstellt regelmäßig eine Auflistung zu Angriffen auf Web Applikationen (Webserver Bereich), die sich direkt auf Web Services von Smart Home-Geräte übertragen lassen. Das entwickelte Sicherheitskonzept „ANTs: Application-Driven Network Trust Zones on MAC-Layer in Smart Buildings“ bedient sich eines Divide-and-Conquer Ansatzes, um Geräte in sogenannte Trust Zones zu gruppieren. Dabei bestimmt die Anwendungsschicht die Gruppenzugehörigkeit. Da auf Anwendungsschicht bestimmte Kommunikationsflüsse, wie beispielsweise zwischen einem Smart TV und einer Türschließanlage aus praktischer Sicht nicht stattfinden können, wird die Kommunikation auf MAC-Ebene, wie Abbildung 42 in dargestellt, frühzeitig unterbunden. Geräte die in logischer Verknüpfung stehen bleiben auf der MAC-Schicht und damit auf IP-Ebene miteinander verbunden. So kann ein Smart TV beispielsweise das Wohnzimmer in einen Kinomodus versetzen und das Licht dimmen. Zur Re-/Konfiguration der MAC-Ebene werden Passwörter/Schlüssel mit hoher Entropie an die einzelnen Smart Home Geräte individuell verteilt. Das SHGW bietet durch das integrierte Trusted Plattform Module eine Möglichkeit Schlüsselmaterial zu generieren. Zur funktionalen Evaluation und Leistungsanalyse des Konzeptes ANTs entstand eine Implementierung. Dabei wurde für die MAC-Ebene das WLAN Mesh Protokoll nach IEEE 802.11s exemplarisch gewählt. Das SHGW steht über eine

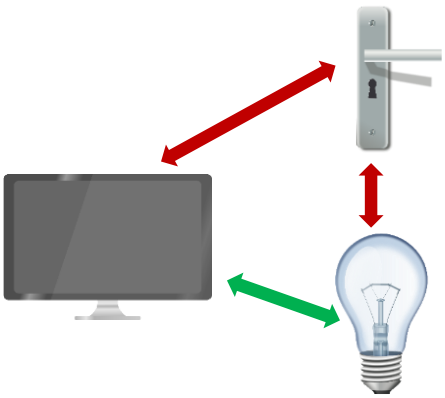


Abbildung 42: Network Trust Zones nach ANTs

[4] Ein ausgefallener Konfigurationsdienst kann eine Regel enthalten, die eine Kollision mit der neuen Regel verursacht. Dieser Fall ist jedoch gesondert zu betrachten, z. B. durch die Integration einer Routine zur Prüfung der Regelkonsistenz im gesamten System, falls der defekte Konfigurationsdienst durch einen neuen ersetzt wird.

drahtgebundene Kommunikation mit dem WLAN Mesh-Netzwerk in Verbindung, was durch einen sogenannten Mesh Portal Point (MPP) realisiert wird. Wie in Abbildung 43 dargestellt, sind alle Geräte über ein Konfigurationsnetz (blau) miteinander verbunden. Über dieses Netz erhalten alle Geräte (End Device) individuelle Zugangsdaten, um der jeweiligen Trust Zone (grün) beizutreten.

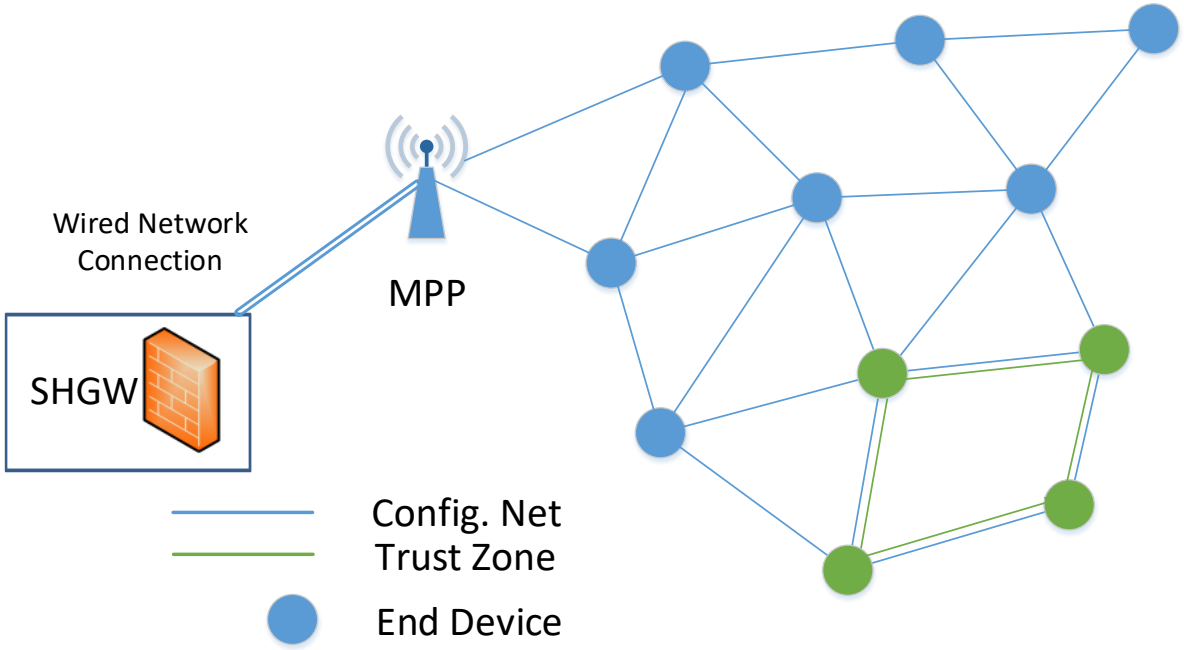


Abbildung 43: Netzwerk-Infrastruktur nach ANTs

5. Demonstrator - Konzeption & Entwicklung von Prototypen

Für die Entwicklung der Prototypen wurden Geräte unterschiedlicher Leistungsklassen genutzt, um den angestrebten Einsatz des Sicherheits-Framework in heterogenen Netzwerken zu modellieren. Das Development Board ESP32 entspricht aufgrund seines geringen Energieverbrauchs von 10 µA – 240 mA der Leistungsfähigkeit batteriebetriebener Sensoren. Während Geräte wie Aktorik, die häufig einen Netzanschluss besitzen und dementsprechend leistungsfähiger sein können, durch Raspberry Pis (200 mA – 2,5 A) im Testaufbau vertreten sind. Als autorisierte Nutzergeräte werden Android Smartphones eingesetzt. Im Folgenden werden die verschiedenen Demonstrationsszenarien erläutert.

Abbildung 44: In den Demonstrationsszenarien verwendete Geräte: ESP32, Raspberry Pi, Android Smartphone

5.1. Aktivierung eines Nutzergerätes mithilfe des neuen elektronischen Personalausweises

Diese Funktionalität wird derzeit entwickelt und ermöglicht die Nutzung des Smart Home Gateways zur Aktivierung eines autorisierten Nutzergerätes, wie z. B. Smartphone. Dazu werden Schnittstellen in das Smart Home Gateway integriert, die eine Kommunikation mit dem neuen elektronischen Personalausweis (nPA) über RFID sowie mit einem Smartphone oder Tablet über NFC (engl. Near Field Communication) ermöglichen. Es wird die Identifikation und Authentifikation des Nutzers sowie die Autorisierung des Nutzergerätes durchgeführt. Die Autorisierung wird mithilfe von SQL-Datenbanken (englisch: Structured Query Language) und Schlüsselpaargenerierung realisiert.

Abbildung 45: RFID-Lesegerät ReinerSCT Standard für nPA [79]

5.2. Hinzufügen und steuern eines neuen Smart Home-Gerätes

Um ein neues Gerät mit limitierten Ein-/Ausgabemöglichkeiten in ein bestehendes Smart Home-Netzwerk zu integrieren, startet das Gerät zunächst einen eigenen Access Point. Der Nutzer kann beispielsweise mit einem Smartphone dem WLAN-Netzwerk des Gerätes beitreten, um dem neuen Endgerät die Zugangsdaten für das Smart Home-Netzwerk übertragen. Nach einem Neustart verbindet

sich das Gerät mit dem existierenden Smart Home-Netzwerk des Nutzers und es kann ein sogenannter Out-of-Band Schlüsselaustausch, z. B. mittels Infrarot-Diode im Gerät und blinkendem Smartphone-Bildschirm, nach [58] durchgeführt werden, um dem neuen Gerät Informationen zur Authentifizierung im Smart Home-Netzwerk zu übermitteln.

An zwei Beispielen soll die Gerätevernetzung mittels CoAP demonstriert werden. Ein pyroelektrischer Sensor (PIR-Sensor) erkennt Bewegungen, um das Licht einzuschalten. Die Lampe ist ein eigenständiges Gerät mit einem ESP32, der die Lampensteuerung über eine RESTful API (CoAP) ermöglicht. Eine PUT-Nachricht an die entsprechende Licht-Ressource (engl.: Light) mit dem Nachrichteninhalt (engl.: payload) „on“ oder „off“ steuert den Zustand der Lampe. Im zweiten Beispiel sendet ein Temperatursensor über eine CoAP PUT-Nachricht zyklisch den aktuellen Temperaturwert an einen CoAP Server, der die Werte anzeigt. Die Temperaturanzeige wurde mit der Java-Implementierung jCoAP [70] auf einem Raspberry Pi 3 mit einem angeschlossenen Display implementiert. Die beiden Beispiele werden in Abbildung 46 dargestellt. Neben der Lampe wurden sowohl der Bewegungsmelder als auch das Thermometer auf Basis eines ESP32 implementiert. Darauf läuft eine in der Programmiersprache C geschriebene CoAP Implementierung. Ein vierter ESP32 dient im Demonstrationsaufbau als WLAN Access Point, um die typische Leistungsaufnahme eines WLAN-Kleinstgerätes im Access Point-Modus zu bestimmen. Die gemessene durchschnittliche Leistungsaufnahme von 600 mW kann in der Produktentwicklung durch Verringerung der Taktfrequenz weiter gesenkt werden.

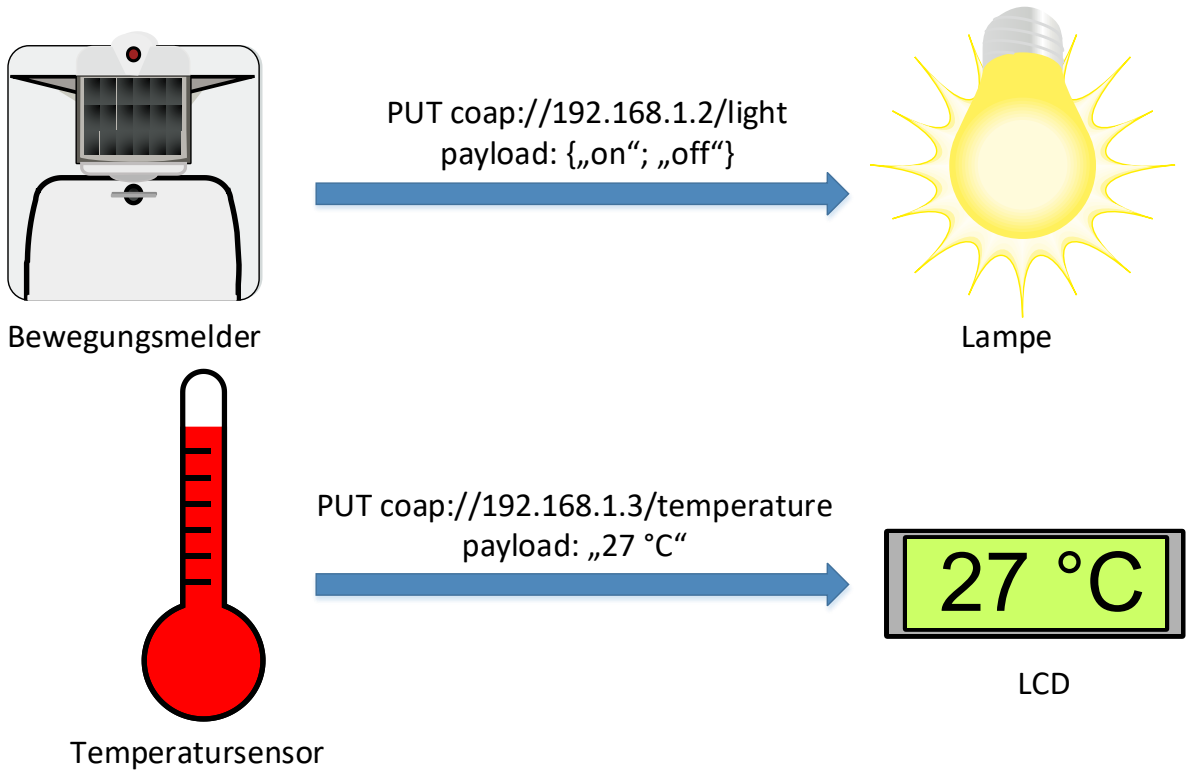


Abbildung 46: WLAN-Testgeräte basierend auf ESP32 und Raspberry Pi 3

Der Temperatursensor wurde in einen sogenannten Deep-Sleep-Modus versetzt, um die Leistungsaufnahme zu reduzieren, wenn keine Daten aufgenommen und versendet werden. Die gemessene Leistung von 25 mW wird fast ausschließlich von der Power-LED verursacht, die auf dem Entwicklungsboard fest verdrahtet ist und nicht über Software deaktiviert werden kann. Bei einem Industrieprodukt entfiele diese LED, sodass sich eine Leistungsaufnahme von 50µW nach dem Datenblatt des ESP32 erzielen ließe. In Abbildung 47 ist der Strom bei einer konstanten Betriebsspannung von 5 V während einer Übertragung dargestellt. Zunächst wird der Chip des ESP32 durch einen Watchdog-Timer aus dem Deep-Sleep-Modus aufgeweckt. Daraufhin verbindet sich das WLAN-Modul mit dem WLAN Access Point durch einen WPA2 Handshake. Darauf ruft der ESP32 über einen 1-Wire Bus den aktuellen digitalisierten Temperaturwert vom Sensormodul ab, baut eine CoAP Nachricht samt Temperaturwert-Payload zusammen und sendet die Nachricht an einen CoAP Server, der auf einem Raspberry Pi 3 ausgeführt wird. Nach erfolgreicher Übertragung versetzt die Software den Chip in den Deep-Sleep-Modus.

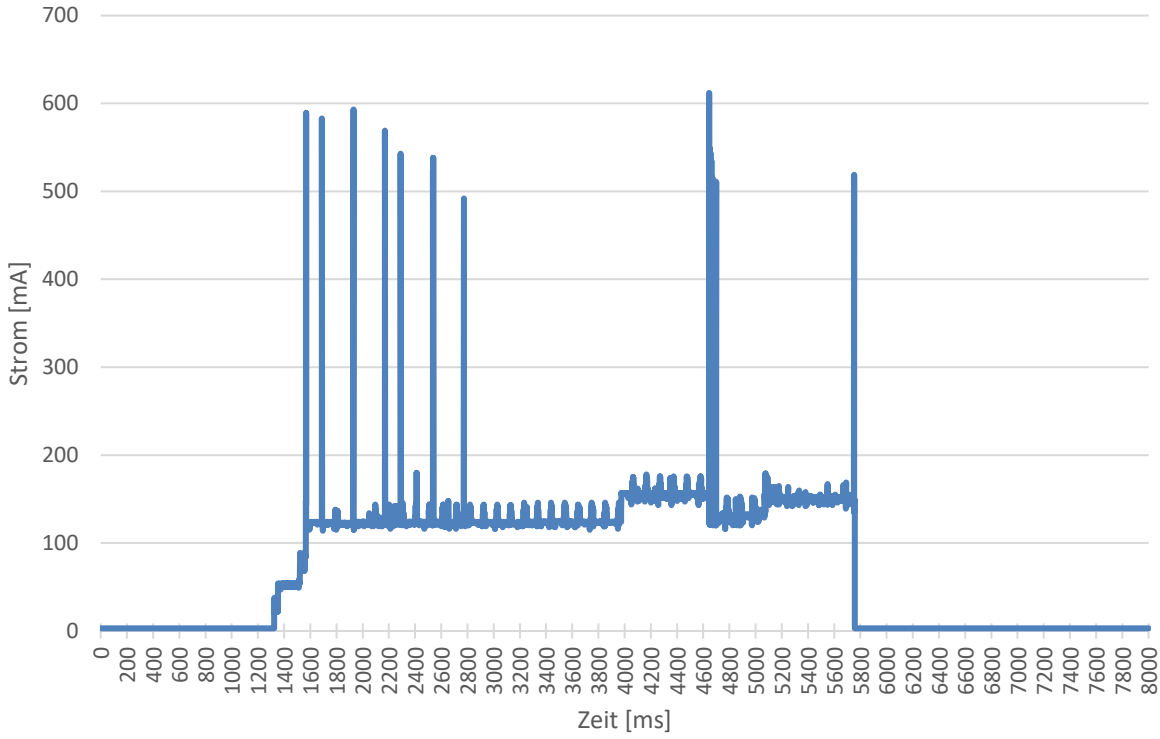


Abbildung 47: Strom bei konstanter Betriebsspannung von 5V während eines Sendevorganges zur Temperaturwert-Übermittlung

Durch weitere Verbesserungen an der Software lässt sich eine bestehende WLAN-Verbindung im Deep-Sleep-Modus aufrecht erhalten, um den 1,2 Sekunden dauernden WPA2-Handshake nicht nach jedem Aufwecken durchführen zu müssen, was den Energieverbrauch eines Sendevorganges um ca. 40 % reduziert. In einem realen IoT-Produkt mit einem typischen Lithium-Ionen-Akku (Typ 18650er mit 3400mAh bei 3,7V) können ca. 25.000 Sendevorgänge ausgeführt werden. Wird vor jedem Sendevorgang ein WPA2-Handshake ausgeführt, sind 15.000 Sendevorgänge möglich. Abbildung 48 stellt die Batterielebensdauer bei einem variablen Sendeintervall dar. Bei einem Intervall von 15 Minuten, z. B. für Messungen der Außentemperatur, liegt die Betriebszeit unter Verwendung einer optimierten Software ohne WPA2-Handshake bei ca. einem ¾ Jahr.

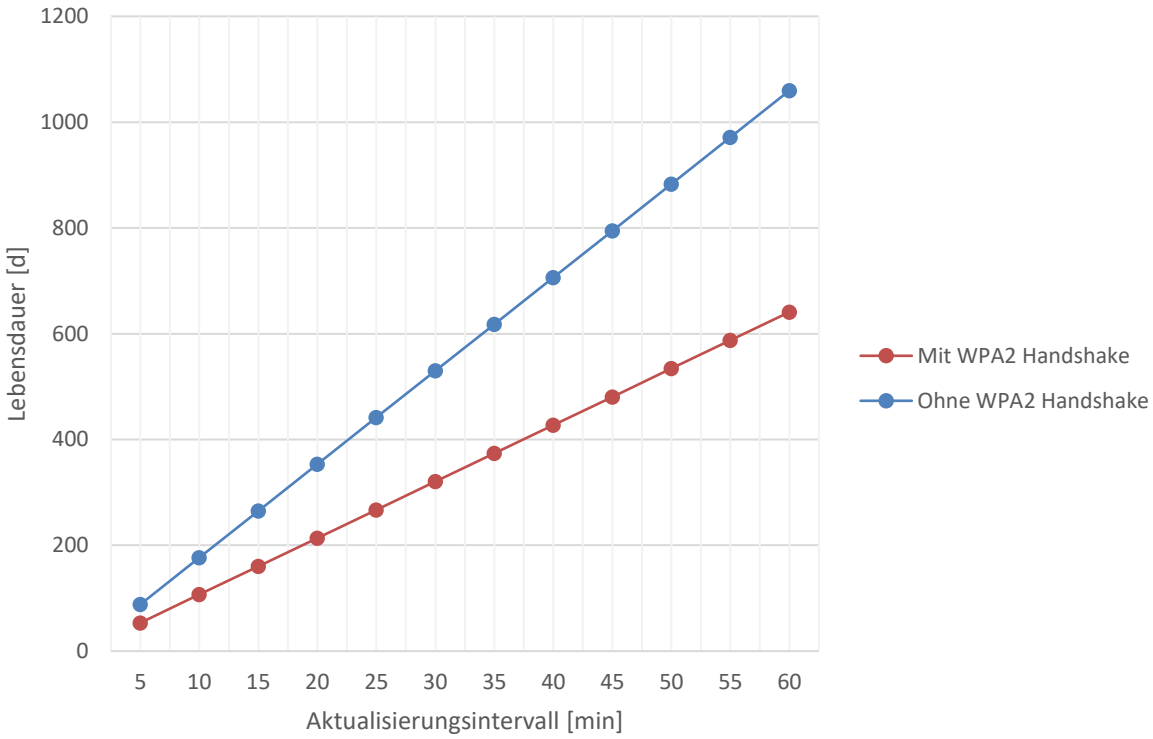


Abbildung 48: Betriebszeit mit einem 18650er Li-Ion Akku mit 3400mAh und 3,7V

5.3. Mehrwertdienst durch Anbindung an Service Plattform

Ein Mehrwertdienst, der sich durch eine Service Plattform im Internet anbietet, ist das sichere Verteilen von wichtigen Firmware Updates an alle Smart Home-Geräte. Bei den Updates kann es sich neben sicherheitskritischen Bug-Fixes auch um funktionale Erweiterungen handeln. In einem exemplarischen Demo-Szenario wird die aktuell installierte Firmware Version von den einzelnen Geräten abgefragt und im SHGW gespeichert. Das SHGW baut eine geschützte Verbindung zur Cloud Plattform auf, um nach verfügbaren Updates zu suchen. Im Falle eines Updates initiiert das SHGW durch das Verteilen von Schlüsselmaterial eine gesicherte Verbindung vom Smart Home-Gerät zum korrespondierenden Server. Dieses Konzept löst das Problem, dass einige eingebettete Systeme keine Kenntnis über vertrauenswürdige CA-Zertifikate haben, da Speicherressourcen limitiert sind. Ein Aufbau einer DTLS/TLS Verbindung durch ein Server-Zertifikat nach dem X.509 Format ist nicht möglich. Die Vertrauensbeziehung vom Smart Home-Gerät zum SHGW, durch die lokale PKI, wird erweitert, da das SHGW als Vertrauensinstanz einen Sitzungsschlüssel an den Cloud Server und das Endgerät senden kann.

5.4. Audio/Video-Gegensprechanlage

Zur Umsetzung des Prototyps bieten sich die in Abschnitt 3.3.1 und 3.4.1 erläuterten Konzepte an. Da das Publish/Subscribe-Verfahren bislang für das Streaming in der Literatur wenig untersucht wurde, wurde jenes Verfahren prototypisch implementiert. Als Geräte-Plattform für die Gegensprechanlage wird ein Raspberry Pi 3 mit ins Gehäuse integrierter Kamera verwendet.

Die Audio/Video-Übertragung kann auf einem Wandpanel oder einem Smartphone bzw. Tablet empfangen und gesteuert werden. Das Wandpanel wird ebenfalls mithilfe eines Raspberry Pi 3 und einem Touchscreen realisiert. Der Aufbau kann aus Abbildung 26 entnommen werden.

Abbildung 49: Kamera-gehäuse für Raspberry Pi [80]

Abbildung 50: Gehäuse für Wandpanel [80]

Abbildung 51: Raspberry Pi 3 [80]

6. Fazit und Ausblick

Die Recherchen zu aktuellen Smart Home-Geräten offenbarten folgende Probleme: Die Integration solcher Geräte in ein Heimnetzwerk erfordert Fachwissen, um Konfigurationsfehler im Gerät und Netzwerk zu vermeiden. Einige dieser am Markt erhältlichen Geräte sind sogar so programmiert, dass sie automatisch eine Verbindung zum Server des Herstellers herstellen, unbekannte Informationen übertragen und dabei neben der Angriffsfläche durch unsachgemäße Konfiguration, weitere Schwachstellen verursachen, die es Angreifern ermöglichen Smart Home-Geräte zu übernehmen. Ein weiterer Aspekt aktueller Produkte ist die bisherige Tendenz von Herstellern zur Entwicklung geschlossener proprietärer Smart Home-Ökosysteme. Jedoch sind die Produktportfolios der Hersteller häufig aufgrund historisch gewachsenen Fachwissens beschränkt und können den Bedarf der Konsumenten nicht mehr ideal bedienen. Daraus resultieren Bemühungen der Open Source-Gemeinschaft zur Entwicklung offener Smart Home-Systeme, die die Interoperabilität zwischen Geräten unterschiedlicher Hersteller ermöglichen. Das Konzept, die einzelnen proprietären Systeme durch übergeordnete offene Systeme zu orchestrieren, ermöglicht eine herstellerübergreifende Kommunikation. Diese sogenannten „Systems of Systems“, mit ihrem provisorischen Charakter, finden Unterstützung in Teilen der Industrie. Jedoch stehen einige Konsortien aus Industriepartnern dieser Entwicklung kritisch gegenüber und engagieren sich für die Vereinheitlichung der Kommunikationsstandards der teilnehmenden Partner. Während der Open Source-Ansatz „System of Systems“ konzeptbedingt keine Sicherheitseigenschaften betrachten kann, da man von den zugrunde liegenden proprietären Lösungen der Hersteller abhängig ist, stagnieren die Arbeiten an einer nachträglichen Integration von Interoperabilitätsschnittstellen aufgrund der hohen Diversität der proprietären Protokolle und Bestrebungen der Hersteller. Letztlich können beide Bestrebungen kaum Einfluss auf Sicherheitsmaßnahmen von Smart Home-Systemen nehmen. Um diese Problematik zu lösen, wurde mit dem Sicherheits-Framework eine Basis für ein Smart Home-System erarbeitet, welche von Beginn an aktuelle Sicherheitsanforderungen des BSI berücksichtigt und diese mit dem Ziel der Konfigurationssicherheit benutzerfreundlich umsetzt. Das Framework kann als eine Art Fundament eines Smart Home-Systems verstanden werden, welches Hersteller bzw. Entwickler als Ausgangspunkt für die Implementierung eigener Smart Home-Geräte und –Systeme verwenden können. In Abschnitt 4.6 werden Erweiterungen beschrieben die dieses Grundelement nutzen und basierend auf dem Sicherheits-Framework weitere Funktionen hinzufügen. Die zu Beginn des Forschungsvorhabens durchgeführten Recherchen wurden genutzt, um trotz des wissenschaftlichen Fokus ein Konzept mit Protokollen zu erarbeiten, das eine zügige Umsetzung in die Realität ermöglicht. Dies kann, wie in Abschnitt 4.5.4 erörtert, u. a. durch die Integration des Sicherheits-Framework in zukünftige Firmware-Updates von Geräten oder als Übergangslösung durch die Aktualisierung der zentralen Einheiten (Hubs) der Smart Home-Systeme ermöglicht werden. Eine reale Umsetzung des erarbeiteten Konzeptes wird durch prototypische Implementierungen ausgewählter Szenarien in Abschnitt 5 dokumentiert.

Während der Bearbeitung des Forschungsprojektes wurden weitere Teilbereiche des Smart Home identifiziert und Ideen entwickelt, die Potential für zukünftige Forschungsprojekte bieten. Eine dieser Ideen ist beispielsweise das dezentrale Protokollieren von Aktivitäten im Smart Home zur redundanten angriffsresistenten Sicherung von Aktivitätsinformationen und zur Erfüllung der Anforderung an die Zurechenbarkeit eines Systems, welches eines der Schutzziele in der Informationssicherheit ist.

7. Ergebnistransfer

Unsere Veröffentlichung („ANTs: Application-Driven Network Trust Zones on MAC Layer in Smart Buildings“) wurde auf der renommierten IEEE Consumer Communications & Networking Conference 2018 (CCNC) [71] angenommen. Auf dieser Konferenz werden wissenschaftliche Arbeiten aus dem akademischen Bereich, sowie dem industriellen Umfeld ausgetauscht. Die CCNC findet im Zusammenspiel mit der angesehenen Consumer Electronics Show (CES) statt, die u. a. als Plattform für Innovationen aus dem Smart Home-Bereich steht. Ein Austausch mit industriellen Vertretern über aktuelle Entwicklungen und Probleme der Heimgerätevernetzung liefert Impulse für anwendungsorientierte Forschungsarbeiten.

> Arne Wall, Hannes Raddatz, Michael Rethfeldt, Peter Danielis, Dirk Timmermann:
> **ANTs: Application-Driven Network Trust Zones on MAC Layer in Smart Buildings**
> In Proceedings of the 15th Annual IEEE Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, Januar 2018

In modernen Smart Home-Systemen ist eine große Anzahl von Geräten miteinander vernetzt. Dies wird u. a. durch Gateways ermöglicht, die zwischen unterschiedlichen Funkprotokollen vermitteln. Jedoch steigt die Gefahr, dass mindestens eines dieser Geräte eine kritische Sicherheitslücke aufweist, die bislang unbekannt war und für Angriffe gegen andere Smart Home-Geräte missbraucht werden kann (Zero Day Exploit). Um den Schaden eines solchen potenziellen Angriffs zu reduzieren, wird die Kommunikation zwischen den Geräten so weit wie möglich eingeschränkt ohne die verteilte Anwendung zu behindern.

Unser entwickelter Ansatz basiert auf einem zentralen Gerät, das eine Übersicht über alle Netzwerkverbindungen, Gerätetypen und Anwendungen speichert und auswertet. Ein zertifiziertes Smart Home Gateway nach dem Schutzprofil für Smart Meter Gateways des BSI bietet sich für eine reale Umsetzung an. Dieses Gerät sendet individuelle, kryptografisch sichere Zugangsdaten (Zufallszahlen als Shared Secret mit hoher Entropie) an die Smart Home-Geräte. Diese Zugangsdaten werden benutzt, um über die IEEE 802.11s WLAN Mesh [72] Technologie anwendungsspezifische Verbindungen aufzubauen. Gesichert werden die Verbindungen der Geräte untereinander über das anerkannte Simultaneous Authentication of Equals (SAE) Verfahren [73]. Unerwünschte Nachrichtenflüsse können durch die Gruppierung von Geräten in sogenannte Trust Zones unterbunden werden, um die Angriffsfläche zu reduzieren. Des Weiteren ist es möglich Geräte mit Sicherheitslücken temporär unter Quarantäne zu stellen, bis die Sicherheitsbedrohungen behoben worden sind.

> Arne Wall, Hannes Raddatz, Michael Rethfeldt, Peter Danielis, Dirk Timmermann:
> **Performance Evaluation of MAC-Layer Trust Zones over Virtual Network Interface**
> 4th Conference On Mobile And Secure Services (MobiSecServ), Miami Beach, Florida, USA, Februar 2018

In einer weiteren Veröffentlichung wird das Konzept von ANTs hinsichtlich der Eignung für die Praxis evaluiert. Das entsprechende Paper wurde auf der von der IEEE gesponsorten MobiSecServ 2018 vor Sicherheitsforschern präsentiert und umfassend diskutiert. Das Paper steht der Allgemeinheit über die IEEE Xplore Onlinebibliothek zur Verfügung.

Hannes Raddatz, Arne Wall, Dirk Timmermann:
SafeBase: A Security Framework for Smart Home Systems Based on Smart Metering Infrastructure
International Conference on Embedded Wireless Systems and Networks (EWSN), Madrid, Februar 2018

Das Paper „SafeBase“ hat den Reviewprozess der International Conference on Embedded Wireless Systems and Networks (EWSN) Konferenz durchlaufen. Die Konzepte zur Nutzung des SHGW als zentralen Vertrauensanker wurden vor Fachpublikum präsentiert.

Arne Wall, Vlado Altmann, Johannes Müller, Hannes Raddatz, Dirk Timmermann:
Decentralized Configuration of Embedded Web Services for Smart Home Applications
11th Annual IEEE International Systems Conference (SYSCON), Montreal, Quebec, Kanada, April 2017

Die Diskussion mit Fachleuten aus dem Bereich der Systemtechnik auf der 11th Annual IEEE International Systems Conference (SYSCON) legten den Grundstein für nachfolgende Arbeiten. Hieraus entstanden die Ideen und Konzepte zur Regelprüfung, um fehlerhafte Gerätekonfigurationen und damit verbundene Sicherheitsprobleme zu erkennen und zu beheben.

Literaturverzeichnis

[1] C. Lee, L. Zappaterra, K. Choi, and H. A. Choi, “Securing smart home: Technologies, security challenges, and security requirements,” *2014 IEEE Conf. Commun. Netw. Secur. CNS 2014*, pp. 67–72, 2014.

[2] “Insecam - World biggest online cameras directory.” [Online]. Available: http://www.insecam.org/. [Accessed: 29-Apr-2018].

[3] “Shodan.” [Online]. Available: https://www.shodan.io/. [Accessed: 29-Apr-2018].

[4] “Security: Ein Botnetz aus Überwachungskameras - Golem.de.”

[5] “Breaking Down Mirai: An IoT DDoS Botnet Analysis.” [Online]. Available: https://www.incapsula.com/blog/malware-analysis-mirai-ddos-botnet.html. [Accessed: 13-Mar-2018].

[6] “Hide’n Seek: IoT-Botnetz mit Spionage-Skills | heise Security.” [Online]. Available: https://www.heise.de/security/meldung/Hide-n-Seek-IoT-Botnetz-mit-Spionage-Skills-3950938.html. [Accessed: 13-Mar-2018].

[7] “Smart-Meter-Rollout: Die Tücken der Digitalisierung - energate messenger+.” [Online]. Available: http://www.energate-messenger.de/news/175879/. [Accessed: 20-Feb-2018].

[8] F. Leferink, C. Keyer, and A. Melentjev, “Static energy meter errors caused by conducted electromagnetic interference,” *IEEE Electromagn. Compat. Mag.*, vol. 5, no. 4, pp. 49–55, 2016.

[9] Heise, “Smart Meter messen oft falsch.” [Online]. Available: https://heise.de/-3644942.

[10] “Gut, aber noch nicht gut genug – Erste Ergebnisse des VDE|FNN-Tests intelligenter Messsysteme.” [Online]. Available: https://www.vde.com/de/presse/pressemitteilungen/fnn-tests-intelligenter-messsysteme. [Accessed: 20-Apr-2018].

[11] “BMWi - Häufig gestellte Fragen rund um das Messstellenbetriebsgesetz (MsbG) und intelligente Messsysteme.” [Online]. Available: https://www.bmwi.de/Redaktion/DE/FAQ/Intelligente-Messsysteme-Zaehler/faq-intelligente-netze-intelligente-zaehler.html. [Accessed: 20-Apr-2018].

[12] “Digitalisierung stockt – Intelligente Stromzähler kommen später | heise online.” [Online]. Available: https://www.heise.de/newsticker/meldung/Digitalisierung-stockt-Intelligente-Stromzaehler-kommen-spaeter-4011821.html. [Accessed: 20-Apr-2018].

[13] “BMWi - Gesetz zur Digitalisierung der Energiewende.” [Online]. Available: https://www.bmwi.de/Redaktion/DE/Downloads/Gesetz/gesetz-zur-digitalisierung-der-energiewende.html. [Accessed: 23-Apr-2018].

[14] “openHAB.” [Online]. Available: http://www.openhab.org/. [Accessed: 26-Jan-2017].

[15] “Calaos, Open Source Home Automation.” [Online]. Available: https://www.calaos.fr/en/. [Accessed: 27-Jan-2017].

[16] “Domoticz.” [Online]. Available: https://domoticz.com/. [Accessed: 27-Jan-2017].

[17] “Home Assistant.” [Online]. Available: https://home-assistant.io/. [Accessed: 27-Jan-2017].

[18] “HomeGenie, Home Automaton Server in the Internet Of Things era.” [Online]. Available: http://www.homegenie.it/. [Accessed: 27-Jan-2017].

[19] “ago control – open source home automation system – home automation with z-wave knx raspberry pi 1wire linux AMQP qpid.” [Online]. Available: https://www.agocontrol.com/. [Accessed: 27-Jan-2017].

[20] “Freedomotic | Open IoT Framework.” [Online]. Available: http://www.freedomotic.com/. [Accessed: 27-Jan-2017].

[21] “MajorDoMo Main/MajorDoMo — open source smart home platform.” [Online]. Available: http://majordomohome.com/. [Accessed: 27-Jan-2017].

[22] “WOSH Framework - Wide Open Smart Home - Homepage.” [Online]. Available: http://wosh.sourceforge.net/. [Accessed: 27-Jan-2017].

[23] “Home: LinuxMCE home automation.” [Online]. Available: http://www.linuxmce.com/. [Accessed: 27-Jan-2017].

[24] “Home of FHEM.” [Online]. Available: http://fhem.de/fhem.html. [Accessed: 27-Jan-2017].

[25] “smarthomatic.” [Online]. Available: https://www.smarthomatic.org/. [Accessed: 27-Jan-2017].

[26] “Eclipse SmartHome - A Flexible Framework for the Smart Home.” [Online]. Available: http://www.eclipse.org/smarthome/. [Accessed: 26-Jan-2017].

[27] “OCF - Open Connectivity Foundation Brings Massive Scale to IoT Ecosystem.” [Online]. Available: https://openconnectivity.org/news/open-connectivity-foundation-brings-massive-scale-to-iot-ecosystem. [Accessed: 26-Jan-2017].

[28] “MQTT Essentials Part 2: Publish & Subscribe.” [Online]. Available: https://www.hivemq.com/blog/mqtt-essentials-part2-publish-subscribe. [Accessed: 29-Apr-2018].

[29] A. Wall, V. Altmann, J. Müller, H. Raddatz, and D. Timmermann, “Decentralized Configuration of Embedded Web Services for Smart Home Applications,” Montreal, Quebec, Kanada, 2017.

[30] “Bundesnetzagentur - Smart Meter.” [Online]. Available: https://www.bundesnetzagentur.de/DE/Sachgebiete/ElektrizitaetundGas/Verbraucher/NetzanschlussUndMessung/SmartMetering/SmartMeter_node.html. [Accessed: 12-Mar-2018].

[31] Bundesamt für Sicherheit in der Informationstechnik, “Das Smart-Meter-Gateway,” p. 40, 2015.

[32] “Smart Meter Gateways (SMGWs) von PPC.” [Online]. Available: https://www.ppc-ag.de/produkte-services/smart-meter-gateways/. [Accessed: 23-Apr-2018].

[33] K.-D. Walter, “Smart-Meter- Kommunikation : sicher , aber funktionsüberladen,” pp. 66–69, 2012.

[34] BSI PP and BSI SMGW PP, “Protection Profile for the Gateway of a Smart Metering System - Schutzprofil für die Kommunikationseinheit eines intelligenten Messsystems für Stoff- und Energiemengen,” 2011.

[35] Z. Shelby, ARM, K. Hartke, C. Bormann, and U. B. TZI, “RFC 7252: The Constrained Application Protocol (CoAP),” 2014.

[36] R. Fielding, U. C. Irvine, and J. Gettys, *RFC 2616: Hypertext Transfer Protocol -- HTTP/1.1*. http://tools.ietf.org/pdf/rfc2616.pdf, 1999, pp. 1–114.

[37] J. Postel, “RFC 768: User Datagram Protocol,” 1980.

[38] Darpa Internet Program, “RFC 793: Transmission Control Protocol,” California, 1981.

[39] E. Rescorla, I. RTFM, N. Modadugu, and I. Google, “RFC 6347: Datagram Transport Layer Security Version 1.2,” 2012.

[40] C. Bormann and P. E. Hoffman, “Concise Binary Object Representation (CBOR),” no. 7049. RFC Editor, 2013.

[41] T. Bray, “The JavaScript Object Notation (JSON) Data Interchange Format,” no. 7159. RFC Editor, 2014.

[42] J. Schaad, “CBOR Object Signing and Encryption (COSE),” no. 8152. RFC Editor, 2017.

[43] R. Barnes, “Use Cases and Requirements for JSON Object Signing and Encryption (JOSE),” no. 7165. RFC Editor, Apr-2014.

[44] E. Rescorla and N. Modadugu, “Datagram Transport Layer Security,” no. 4347. RFC Editor, Apr-2006.

[45] D. Hardt, “RFC 6749: The OAuth 2.0 Authorization Framework,” 2012.

[46] OpenID, “Standard: OpenID Connect Core 1.0 incorporating errata set 1,” 2014. [Online]. Available: http://openid.net/specs/openid-connect-core-1_0.html. [Accessed: 26-Jan-2017].

[47] D. Fett, R. Küsters, and G. Schmitz, “A Comprehensive Formal Security Analysis of OAuth 2.0,” 2016.

[48] L. Seitz, G. Selander, E. Wahlstroem, S. Erdtman, and H. Tschofenig, “Authentication and Authorization for Constrained Environments (ACE),” Internet Engineering Task Force, 2017.

[49] P. Krawiec, M. Sosnowski, J. Batalla, C. Mavromoustakis, and G. Mastorakis, “DASCo: dynamic adaptive streaming over CoAP,” *Multimed. Tools Appl.*, pp. 1–20, 2017.

[50] G. Choi, D. Kim, and I. Yeom, “Efficient streaming over CoAP,” in *{IEEE} 2016 International Conference on Information Networking (ICOIN)*.

[51] O. N. Diaz, S. Loreto, and H. M. Back, “Method and server for sending a data stream to a client and method and client for receiving a data stream from a server.” Google Patents, 2017.

[52] International Telecommunication Union, “Standard: H.264 Advanced video coding for generic audiovisual services,” 2003.

[53] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, 2012.

[54] Bundesministerium des Inneren, “Die Technik des neuen Personalausweises,” 2010.

[55] “Personalausweis und Kosten der Beantragung | Öffentliche IT (ÖFIT).” [Online]. Available: http://www.oeffentliche-it.de/personalausweis#Welche Kriterien muss ein Unternehmen aufweisen, um ein spezielles eID-Zertifikat zu erhalten? [Accessed: 26-Jan-2017].

[56] C. Haubelt, “Course Material of ‘Selected Topics in Embedded Systems Design.’” Institut für Angewandte Mikroelektronik und Datentechnik, Universität Rostock, 2017.

[57] “Hide’n Seek.” [Online]. Available: https://www.heise.de/security/meldung/Hide-n-Seek-IoT-Botnetz-mit-Spionage-Skills-3950938.html.

[58] S. Unger and D. Timmermann, “Bridging the UI gap for authentication in smart environments,” *Proc. - Int. Symp. Comput. Commun.*, 2014.

[59] F. K. Santoso and N. C. H. Vun, “Securing IoT for smart home system,” *Proc. Int. Symp. Consum. Electron. ISCE*, vol. 2015–Augus, pp. 5–6, 2015.

[60] Eclipse IoT Working Group, IEEE IoT, and AGILE IoT, “IoT Developer Survey,” 2016.

[61] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications,” *IEEE Commun. Surv. Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[62] S. I. E. E. E. Association, “ANSI IEEE 802.3 Standard,” 1998.

[63] “IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” *IEEE Std 802.11-2012 (Revision IEEE Std 802.11-2007)*, 2012.

[64] “{IEEE} Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements Part 15.4: Wireless Medium Access Control ({MAC}) and Physical Layer ({PHY}) Specification,” IEEE, 2006.

[65] G. Montenegro, J. Hui, D. Culler, and N. Kushalnagar, “Transmission of IPv6 Packets over IEEE 802.15.4 Networks,” no. 4944. RFC Editor, Sep-2007.

[66] G. Klas, V. Friedhelm Rodermund, V. Zach Shelby, A. Sandeep Akhouri, and E. Jan Höller, “White Paper ‘ Lightweight M2M ’ : Enabling Device Management and Applications for the Internet of Things,” 2014.

[67] “Smart Home Beleuchtung: LED-Lampen & mehr - IKEA.” [Online]. Available: https://www.ikea.com/de/de/catalog/categories/departments/lighting/36812/. [Accessed: 27-Apr-2018].

[68] S. Behl, “Konzeption und Implementierung einer dezentralen Regelverwaltung für IoT und Smart Home,” Universität Rostock, 2018.

[69] “IFTTT helps your apps and devices work together.” [Online]. Available: https://ifttt.com/. [Accessed: 28-Apr-2018].

[70] “WS4D / jCoAP | GitLab.” [Online]. Available: https://gitlab.amd.e-technik.uni-rostock.de/ws4d/jcoap. [Accessed: 28-Mar-2018].

[71] “IEEE CCNC 2018 | IEEE Consumer Communications & Networking Conference | 12-15 January 2018 // Las Vegas // USA.” [Online]. Available: http://ccnc2018.ieee-ccnc.org/. [Accessed: 27-Oct-2017].

[72] L. a N. Man, S. Committee, I. Computer, Man, and Lan, *Specific requirements Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 10 : Mesh Networking IEEE Computer Society*. 2011.

[73] D. Harkins, “Simultaneous Authentication of Equals: A Secure, Password-Based Key Exchange for Mesh Networks,” in *2008 Second International Conference on Sensor Technologies and Applications (sensorcomm 2008)*, 2008, pp. 839–844.

[74] “IoT Developer Survey 2015.” [Online]. Available: https://www.slideshare.net/IanSkerrett/iot-developer-survey-2015. [Accessed: 29-Apr-2018].

[75] “IoT Developer Survey 2016.” [Online]. Available: https://www.slideshare.net/IanSkerrett/iot-developer-survey-2016. [Accessed: 29-Apr-2018].

[76] “IoT Developer Survey 2017.” [Online]. Available: https://www.slideshare.net/IanSkerrett/iot-developer-survey-2017. [Accessed: 29-Apr-2018].

[77] “Key Trends from the IoT Developer Survey 2018 | Eclipse Foundation.” [Online]. Available: https://blogs.eclipse.org/post/benjamin-cabé/key-trends-iot-developer-survey-2018. [Accessed: 29-Apr-2018].

[78] S. Loreto, O. Novo, and Ericsson, “RFC: CoAP Streaming,” 2012.

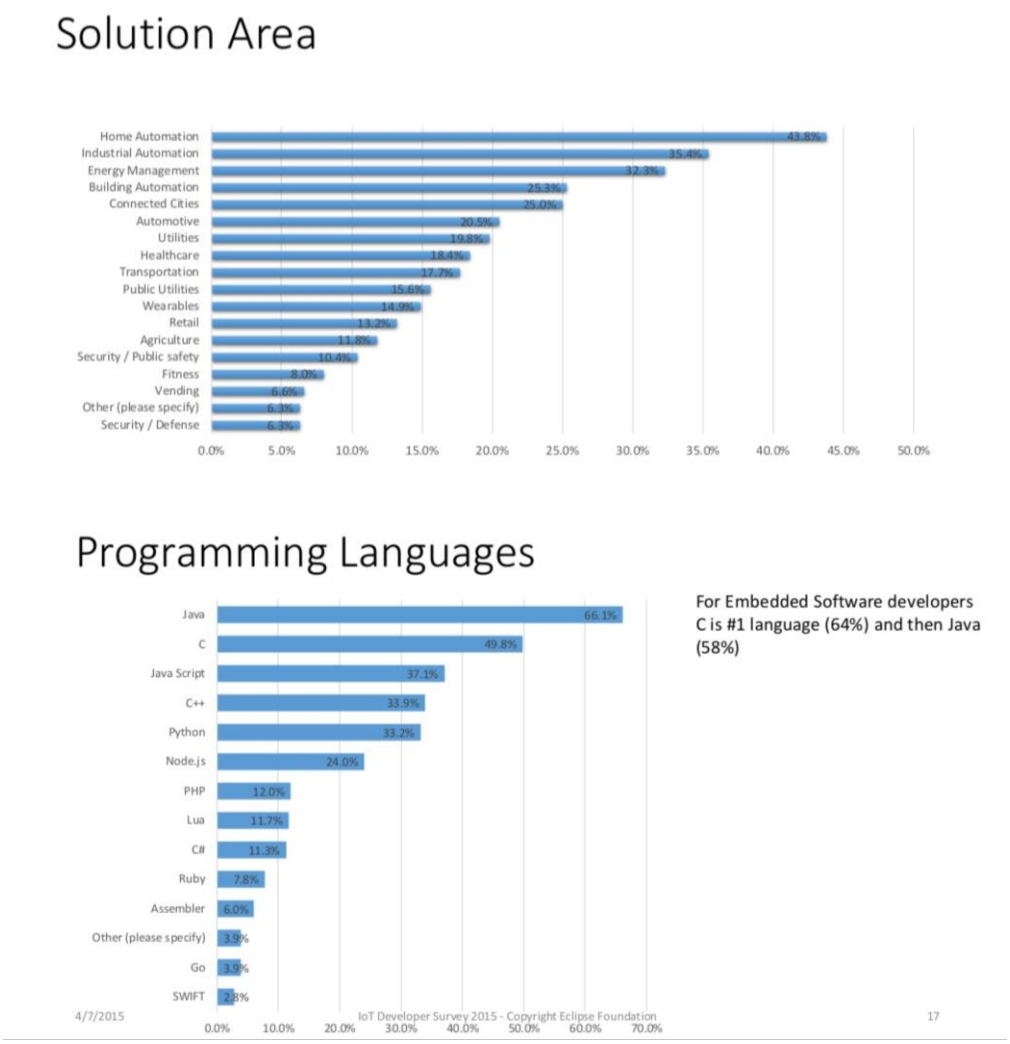
[79] “REINER SCT - Chipkartenleser - cyberJack RFID standard.” [Online]. Available: http://www.reiner-sct.com/produkte/chipkartenleser/cyberJack_RFID_standard.html. [Accessed: 27-Jan-2017].

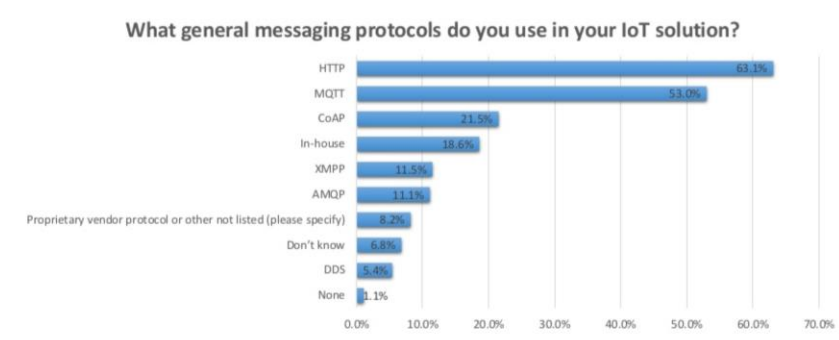
[80] “ModMyPi | Cases for your Raspberry Pi.” [Online]. Available: https://www.modmypi.com/. [Accessed: 27-Jan-2017].

Anlagen

I. IoT-Entwickler Umfragen der Eclipse Foundation

IoT Developer Survey 2015 [74]

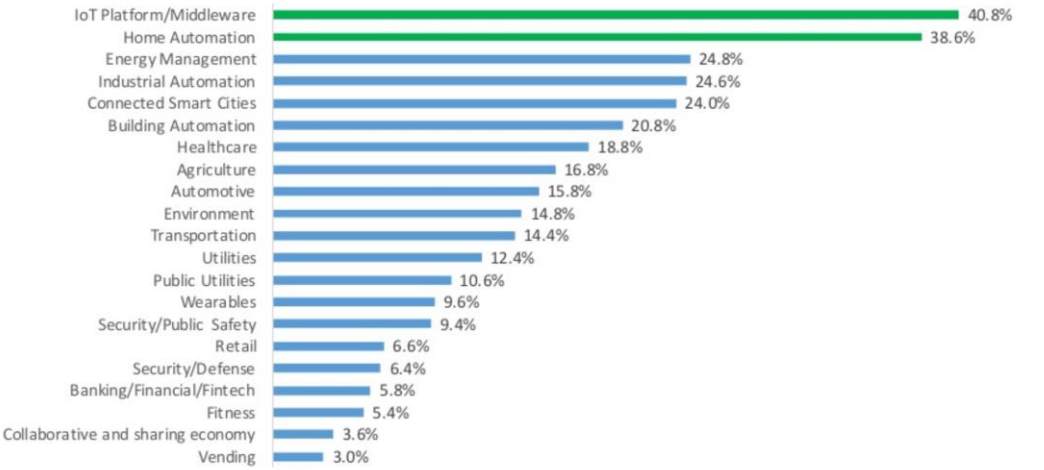




IoT Developer Survey 2016 [75]

KEY INDUSTRIES

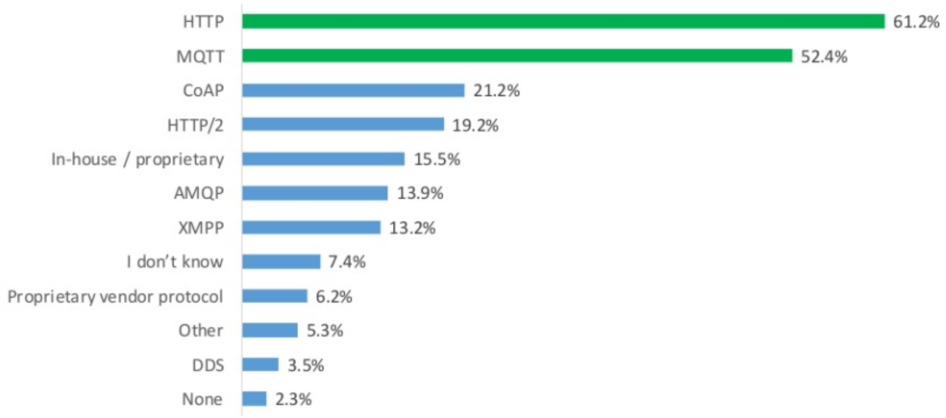
What industry or industries best describe(s) the type of IoT solutions you have built or will build?



4/14/16 IoT Developer Survey 2016 - Copyright Eclipse Foundation 14

MESSAGING STANDARDS

What messaging protocol(s) do you use for your IoT solution?

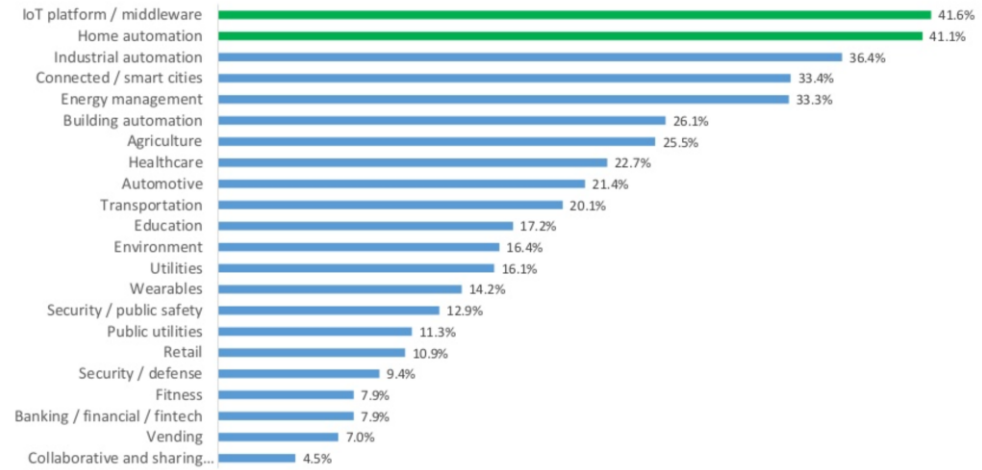


4/14/16 IoT Developer Survey 2016 - Copyright Eclipse Foundation 24

IoT Developer Survey 2017 [76]

KEY INDUSTRIES

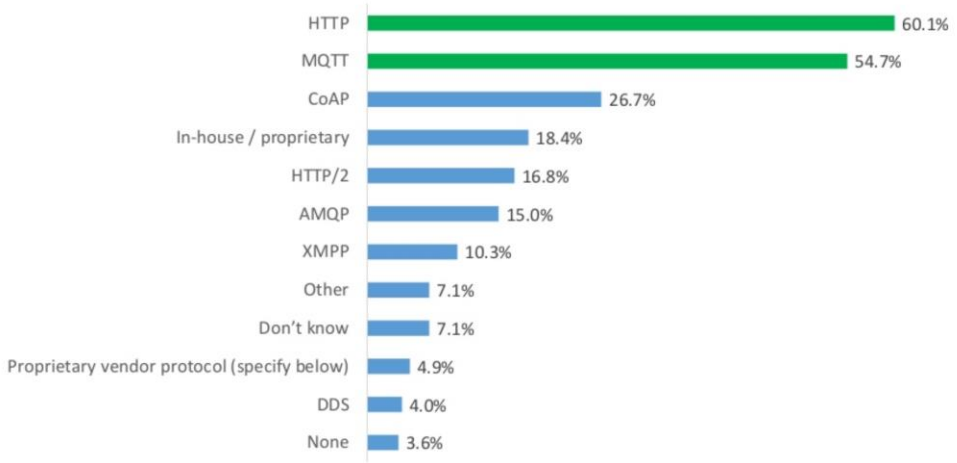
What industry or industries best describe(s) the type of IoT solutions you have built or will build?



IoT Developer Survey 2017 - Copyright Eclipse Foundation, Inc.

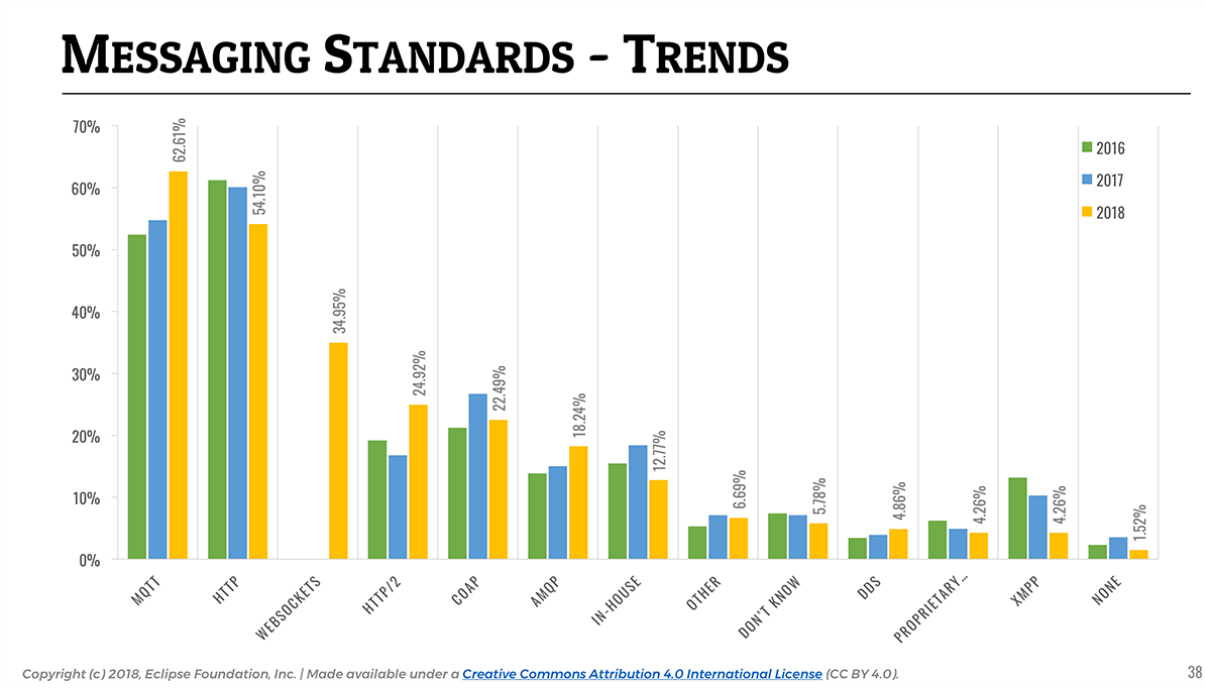
MESSAGING STANDARDS

What messaging protocol(s) do you use for your IoT solution?

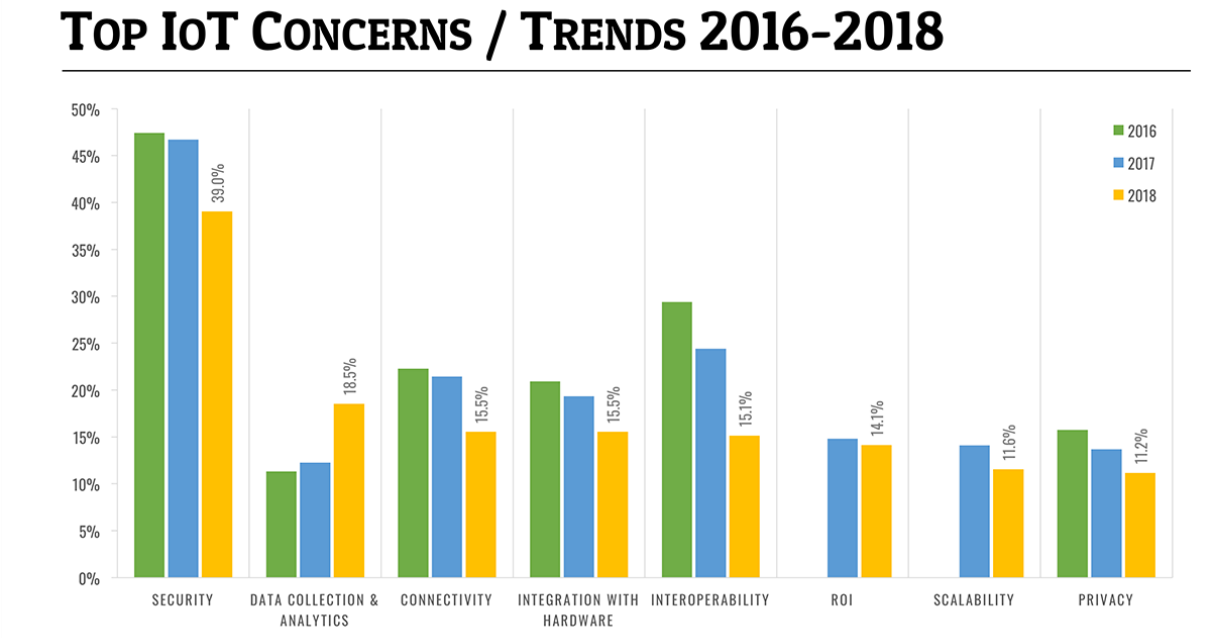


IoT Developer Survey 2017 - Copyright Eclipse Foundation, Inc.

IoT Developer Survey 2018 [77]



38



18