

openHAB

Automatisiertes Heim – Teil 1

Kai Kreuzer, Thomas Eichstädt-Engelen

Kai Kreuzer, Thomas Eichstädt-Engelen

openHAB

Automatisiertes Heim – Teil 1

ISBN: 978-3-86802-559-0

© 2015 entwickler.press

Ein Imprint der Software & Support Media GmbH

Inhaltsverzeichnis

1 Heimautomatisierung und das Internet der Dinge

2 Marktüberblick zu Systemen und Protokollen

3 Geräteabstraktion und ihre Schwierigkeiten

4 Standards, Open Source und Forschung

Die Autoren

1 Heimautomatisierung und das Internet der Dinge

2014 war das Jahr des Smart Homes. Nie zuvor wurde so viel über das intelligente Zuhause gesprochen. Nicht nur Apple hat mit HomeKit in iOS 8 den Markteintritt gewagt, auch Google hat mit dem Zukauf von Nest und Dropcam für einigen Wirbel gesorgt. Doch was genau verstehen wir unter einem Smart Home, und was hat das Internet der Dinge damit zu tun?

Normalbürger sprechen vom intelligenten Zuhause, Techniker dagegen eher von Heimautomatisierung – gemeint ist das Gleiche: Die Integration von verschiedenen Dingen im Haus wie Licht, Heizung und Rollläden, aber auch von Multimediageräten wie Radio, Fernsehen und Stereoanlage oder der so genannten weißen Ware, also Waschmaschinen, Kühlschränken, Geschirrspülern und Ähnlichem. Integration bedeutet hierbei Zweierlei: Zum Ersten die zentrale Erreichbarkeit z. B. über ein Touchpanel, um Geräte zu schalten bzw. deren Status zu visualisieren. Zum Zweiten aber auch die Möglichkeit, dass Geräte automatisch aufeinander reagieren, also z. B. dass sich ein Licht einschaltet, sobald Bewegung erkannt wird.

Geräteklassen werden klassisch in Sensorik und Aktorik aufgeteilt. Während Sensoren Daten aus der Umgebung aufnehmen und weitergeben können, sorgen Aktoren für Aktionen wie Schaltvorgänge. Bisher waren insbesondere die Gewerke Licht und Rollläden immer sehr eng mit der Hauselektrik verbunden – bei der Aktorik hat man prinzipiell die Wahl, diese nach am Gerät (also z. B. in einer schaltbaren Steckdose) zu installieren oder sie zentral im Schaltschrank (Abb. 1.1) unterzubringen. Letzteres ist die etwas kostengünstigere Variante, allerdings eigentlich nur im Neubau oder zumindest einer Kernsanierung eine valide Option. In beiden Fällen kommt man um fundierte Elektrokenntnisse und damit einen Elektriker nicht herum – und so waren Smart Homes bislang nur ein teurer Nischenmarkt.

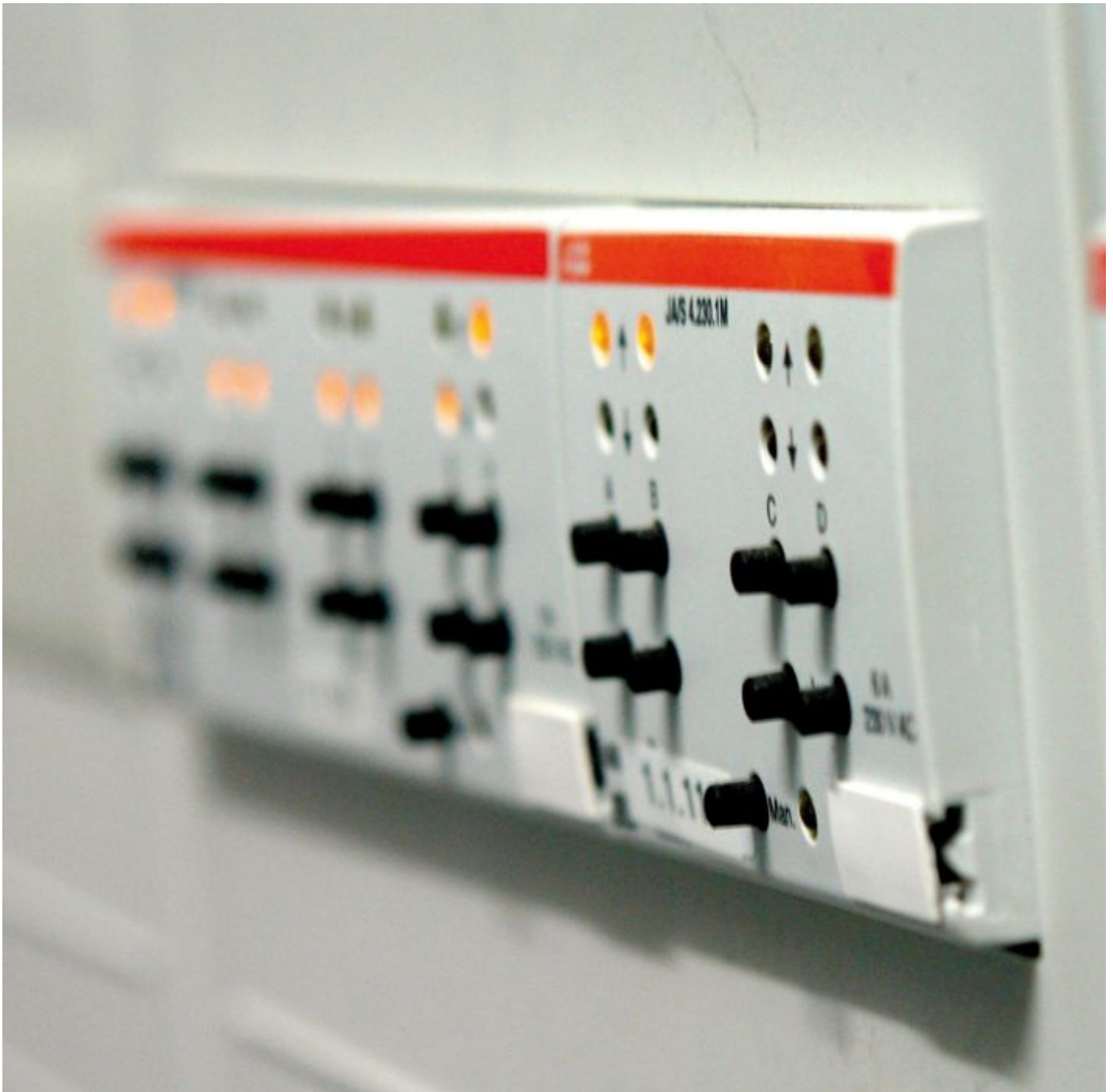


Abbildung 1.1: Klassische Businstallation mit Aktorik im Schaltschrank

Intranet der Dinge

Mit zunehmender Digitalisierung und insbesondere der Verbreitung von Breitbandanschlüssen nimmt das Internet der Dinge (Internet of Things, IoT) nun aber an Fahrt auf und ist zu einem der Buzzwords des Jahres geworden. Das Internet der Dinge geht jedoch weit über Smart Homes hinaus – man versteht darunter jegliche Vernetzung von Geräten auf Basis von Internettechnologien, das heißt im Wesentlichen mithilfe des Internetprotokolls (IP). Hierzu zählen unter anderem auch die Themen Industrie 4.0 oder Smart Cities – Smart Homes sind also nur ein kleiner Bereich des Internets der Dinge. Genauer gesagt bildet ein Smart Home ein

„Intranet der Dinge“ [1]. Der Gedanke, dass eine Heimautomatisierungslösung auch autark, das heißt ohne Verbindung zum restlichen Internet, funktionieren sollte, wird allerdings mehr und mehr von aktuell auf den Markt gebrachten Geräten in Frage gestellt. So ist es heutzutage leider normal, dass sich ein Gerät wie ein Smart-TV, ein Pflanzensensor, eine LED-Lampe oder ein intelligentes Thermostat nach dem Einschalten erst einmal mit dem Cloud-Dienst seines Herstellers verbindet und auch von diesem seine Kommandos erhält. Dies ist insbesondere aus Datenschutz- sowie aus Sicherheitsbetrachtungen keine erfreuliche Entwicklung. Hier ist insbesondere das neue Entwicklerprogramm von Google Nest [2] kritisch zu sehen, denn an der Cloud scheint es keinen Weg vorbei zu geben. Einen durchaus positiven Ansatz verfolgt Apple mit dem kürzlich vorgestellten HomeKit-Framework [3], welches mit iOS 8 im Herbst verfügbar gemacht werden soll. Apple setzt hierbei auf lokale Kommunikation vornehmlich über WiFi und Bluetooth und hält die erhobenen Daten dabei lokal vor. Der Hauptkritikpunkt an HomeKit ist somit die Festlegung auf das Apple-Ökosystem – mit einem Android-Smartphone wird man es kaum nutzen können. Das ist aus Geschäftssicht von Apple natürlich nachvollziehbar, für den Endkunden aber dennoch ein unschöner Aspekt. Der Ansatz, auf Funktechnologien zu setzen, ist allgemein im Markt zu erkennen, da nur hiermit ein echter Massenmarkt erreicht werden kann. Neben WiFi und Bluetooth gibt es aber noch viele andere Systeme, wie zum Beispiel das in Deutschland verbreitete HomeMatic oder Standards wie ZigBee.

Abhängigkeiten verringern

Um sich nicht der Lösung eines Herstellers mit primär eigenen kommerziellen Interessen ausliefern zu müssen, wäre es doch vorteilhaft, ein Smart Home ohne eine solche Abhängigkeit realisieren zu können. Genau diesen Gedanken verfolgt die AllSeen Alliance mit dem AllJoyn-Framework [4]. Dabei handelt es sich um eine offene Open-Source-Lösung, welche eine Peer-2-Peer-Kommunikation zwischen Geräten im (lokalen) Netzwerk gewährleisten soll. Allerdings fehlt einem solchen Ansatz die Möglichkeit, über eine Zentrale einen Fernzugriff zu realisieren, beziehungsweise übergreifende Automatisierungsregeln zu verwalten, da es keine übergeordnete Instanz gibt. Eine solche Instanz nennt man üblicherweise „Home-Gateway“ (Abb. 1.2) und sie erfüllt – ähnlich dem Router – zentrale Aufgaben und dient allgemein der Verbindung von

Geräten. Im Gegensatz zum Router kann ein Home-Gateway hierfür nicht nur auf Ethernet/WiFi zurückgreifen, sondern beliebig über USB-Dongles für weitere Protokolle oder Schnittstellen fit gemacht werden. Ein solches Gateway eignet sich also bestens als neutraler Integrationspunkt, um die verschiedensten Systeme ins eigene Smart Home einzubinden und zu einer Gesamtlösung zu verbinden. Es darf dabei allerdings nicht von der Strategie eines Herstellers abhängen, welche Fremdsysteme integriert werden können. Vielmehr muss das Gateway so offen gestaltet sein, dass jeder die Möglichkeit hat, weitere Anbindungen zu realisieren.



Abbildung 1.2: openHAB läuft auf einem Raspberry Pi als Home-Gateway

openHAB – Open Source und herstellerneutral

Einen solchen offenen und herstellerneutralen Softwarestack entwickelt das Open-Source-Projekt openHAB [5]. Dieses Projekt wurde bereits 2010 ins Leben gerufen, um einheitliche Oberflächen und übergreifende Automatisierungsregeln über Systemgrenzen hinweg realisieren zu können. Es wird von einer großen und stetig wachsenden Community getragen und bietet die Integration von nahezu einhundert verschiedenen Technologien, Protokollen und Systemen an. Die derzeit in Entwicklung befindliche Version 2.0 setzt auf dem Eclipse-SmartHome-Projekt [6] der Eclipse Foundation auf, welche sich der Neutralität und des Open-Source-

Gedankens verschrieben hat.

Als Benutzeroberflächen stehen bei openHAB sowohl webbasierte UIs als auch native Apps für iOS und Android zur Verfügung. Grafische Benutzerschnittstellen dienen üblicherweise primär zur Visualisierung und nur sekundär zum Schalten. Ein echtes Smart Home sollte seine Sache so gut machen, dass nur selten die Notwendigkeit besteht, per Smartphone, Tablet oder PC Befehle auszuführen. Stattdessen kontrolliert man eher den Zustand, lässt sich also Charts zu Temperaturverläufen oder Energieverbrauch zeigen oder prüft, dass auch wirklich alle Fenster geschlossen sind. Wichtiger als eine solche „Pull“-Nutzung ist daher oft die „Push“-Funktionalität. Das Heim versorgt den Nutzer während seiner Abwesenheit aktiv mit Informationen per Push-Nachricht auf das Smartphone (Abb. 1.3). Beispiele dafür sind Alarmmeldungen von Sensoren wie Fensterkontakten, Wasser- oder Rauchmeldern oder Infos zu verpassten Anrufen oder an der Haustür klingelnden Gästen.



Schaltzentrale

Für die Steuerung bieten sich statt einem grafischen Interface oft andere Alternativen an: Zuallererst ist da der klassische (Licht-)Schalter an der Wand. Diesen kennt jeder, er ist intuitiv zu bedienen und gibt keine Rätsel auf. Will man aber reichhaltige Funktionen realisieren, so hat man es schnell mit einer ganzen Schalterbatterie an der Wand zu tun, bei welcher schnell selbst der Hausherr keinen Überblick mehr hat. Der Vorteil an einer intelligenten Haustechnik: Die Funktion der Schalter ist nicht mehr durch die Elektroinstallation vorgegeben, sondern kann nach Belieben jederzeit umprogrammiert werden. Auch können so genannte Szenen abgelegt werden, bei der zum Beispiel gleichzeitig mehrere Lampen und Rollläden in einen bestimmten Zustand gebracht werden. Aber selbst dann ist es ratsam, die Anzahl der Schalter an der Wand klein zu halten und sich auf das Nötigste zu beschränken.

Seit der Einführung von Siri ist Sprachsteuerung wortwörtlich „in aller Munde“ – und gerade zu Hause, wo man mit wenig Nebengeräuschen zu kämpfen hat, kann diese Technologie sehr zuverlässige Dienste verrichten. So kann man bestimmte Szenen wie für das Essen oder den Fernsehabend einfach per Sprachbefehl anwählen. Auch lässt sich darüber beispielsweise die Heizungsanlage oder das Musiksistem ansteuern.

Spricht man selbst zu dem Haus, wird schnell auch die Gegenrichtung gewünscht – dass das Haus zu einem selbst spricht. Dies lässt sich per Text-to-Speech-Funktionalität (TTS) realisieren: Die Sprachausgabe kann hierbei beispielsweise direkt über an das Gateway angeschlossene Lautsprecher geschehen oder per Audiostreaming durch AirPlay oder Sonos auf entfernte Wiedergabegeräte geschickt werden. Solche Sprachausgaben sind als zeitnahe Benachrichtigungskanal sehr nützlich. So kann man sich beispielsweise darüber informieren lassen, dass die Waschmaschine im Keller fertig ist oder dass noch Fenster im Haus geöffnet sind, wenn man dieses gerade verlassen möchte.

NFC und Bluetooth Low Energy

Eine weitere praktische Möglichkeit zur Steuerung sind NFC-Tags. Diese günstigen, batterielosen Aufkleber können von zahlreichen Android-Geräten erkannt und für die Ausführung von Funktionen genutzt werden. Die native Android-openHAB-App unterstützt NFC direkt. Man kann damit

Tags so programmieren, dass ein Schaltvorgang dadurch ausgelöst wird, indem das Smartphone nah an den Tag gehalten wird. Die App braucht dazu noch nicht einmal geöffnet zu sein. Dies kann gerade im Außenbereich sehr nützlich sein: So kann man per NFC-Tag beispielsweise Außensteckdosen schalten, Garagentore öffnen oder sogar die Haustüre entriegeln. Dass dies kein Fremder ebenso machen kann, wird durch Credentials in der Smartphone-App gewährleistet sowie durch die Tatsache, dass nur dieses überhaupt im eigenen WLAN eingebucht ist.

Neben NFC ist auch Bluetooth Low Energy, ebenfalls bekannt als Bluetooth 4.0, sehr relevant. Dies ist insbesondere der Tatsache geschuldet, dass Apple sich weigert, seine Geräte NFC-fähig zu machen und stattdessen auf Bluetooth setzt. Bluetoothtags wie zum Beispiel die iBeacons sind zwar deutlich teurer als NFC-Tags und brauchen oft auch eine Batterie für den Betrieb, bieten aber auch mehr Möglichkeiten: Neben der Erkennung des direkten Kontakts (üblicherweise < 5 cm bei NFC), kann ein Bluetoothtag über größere Entfernungen von über 10 Metern geortet und über dessen Signalstärke auch der ungefähre Abstand ermittelt werden. Diese Information lässt sich hervorragend für einen besseren Kontext nutzen: Befindet sich der Nutzer beispielsweise in der Nähe des Tags im Wohnzimmer, so kann das Sprachkommando „Licht aus“ anders interpretiert werden als wenn er sich im Schlafzimmer aufhält. Dienste wie Google Now zeigen schon heute, wie mächtig die Information des Kontexts sein kann: Wenn diese Information aber zeitgleich auch für Werbung genutzt wird, kann dies durchaus unheimlich sein.

Fazit

Man kann erkennen, wie vielfältig und spannend die Themen rund ums Smart Home sind, nicht zuletzt, da der ganze Markt in großer Bewegung ist. Im nächsten Kapitel wollen wir für etwas Transparenz sorgen und einen Überblick über relevante und interessante Systeme und Protokolle geben. Durch die herrschende Fragmentierung ist der Markt insbesondere für Einsteiger beinahe unüberschaubar, sodass die Auswahl einer passenden Lösung sehr schwer fällt.

Links & Literatur

[1] <http://kaikreuzer.blogspot.de/2014/02/privacy-in-smart-home-why-we-need.html>

[2] <https://nest.com/works-with-nest/>

[3] <https://developer.apple.com/homekit/>

[4] <https://www.alljoyn.org>

[5] <http://www.openhab.org>

[6] <https://www.eclipse.org/smarthome/>

2 Marktüberblick zu Systemen und Protokollen

„Bevor ich mich falsch entscheide, lasse ich es lieber gleich ...“ – ein häufig gehörtes Argument gegen das Smart Home. Richtig ist, dass der Markt schon jetzt unübersichtlich ist und sich das in Zukunft nicht verbessern wird. Wer allerdings die richtigen Fragen kennt, kann die Vielfalt einschränken und damit schließlich zu seiner Entscheidung kommen. In diesem Kapitel stellen wir die wichtigsten Systeme vor und geben Entscheidungshilfen.

Im vorherigen Kapitel haben wir festgestellt, wie vielfältig und spannend die Themen rund um das Smart Home sind. Nun wollen wir konkreter werden und uns verschiedene Systeme genauer anschauen, mit denen die Smart-Home-Anwendungsfälle umgesetzt werden können.

Am Anfang jeder Kaufentscheidung dieser Tragweite steht eine möglichst detaillierte Bestands- und Bedarfsanalyse. Hierbei werden Parameter wie die baulichen Gegebenheiten erhoben, die aktuellen und zukünftigen Anforderungen des Nutzers ermittelt sowie der finanzielle Rahmen festgelegt. Kann die Erhebung dieser Parameter noch leicht durch den Benutzer selbst durchgeführt werden, braucht es für die Auswertung und Entscheidungsvorbereitung eine gute Marktkennntnis der verschiedenen Systeme mit ihren spezifischen Stärken und Schwächen.

Bei der Suche des einen Experten, der hier eine eingehende Beratung leisten kann, wird man allerdings nur selten fündig. Der schnell kontaktierte Elektriker des Vertrauens bietet häufig eher kabelgebundene Systeme an und bezieht selten neue Technologien auf „Gadget-Level“ mit ein. Dem Radio- und Fernsehtechniker fehlt das Wissen über die Normen und Gepflogenheiten der Starkstromwelt und dem IT-Spezialisten fehlt ausbildungsseitig das Hardwarewissen gänzlich. Der Berufszweig, der die relevanten Technologien ausreichend versteht, sie konfigurieren kann und sie vor allem installieren darf, ist somit noch nicht erfunden. Ein Problem, das inzwischen auch von den Branchenverbänden erkannt wurde. Als Maßnahme wurden Interessensgruppen wie der Verband der KNX-Professionals, der SmartHome Deutschland e.V., die Initiative Intelligentes Wohnen des ZVEI und diverse Arbeitskreise wie der AK „ConnectedHome“ der BITKOM mit dem Ziel gegründet, die Probleme der stockenden

Markteinführung zu analysieren, Transparenz zu schaffen und gemeinsam an Lösungen zu arbeiten.

Aktuell heißt es also immer noch „Selbst ist der Smart-Häuslebauer“, weshalb wir nun einige Systeme vorstellen und ihre Relevanz für den Smart-Home-Markt beleuchten wollen. Generell lassen sich die vorhandenen Systeme in kabelgebundene, stromleitungsnutzende und kabellose Systeme unterscheiden. Im Neubau oder Sanierungsfall werden eher die kabelgebundenen Systeme präferiert, während im Bestands- und Renovierungsfall eher die kabellosen Systeme zum Einsatz kommen. Jede Stufe dazwischen kann natürlich ebenfalls vorkommen. Diese grobe Kategorisierung kann dann mit Kriterien wie Anzahl unterstützter Drittsysteme, Herstellerunabhängigkeit, Erweiterbarkeit und Zukunftssicherheit verfeinert werden. Welches System dabei für den individuellen Einsatz das Richtige ist, hängt insbesondere von der oben beschriebenen Bestands- und Bedarfsanalyse ab. In jedem Fall sollten die Visualisierung, also das User Interface zur Darstellung von Geräten, Bildern und Temperaturen sowie die Automationsregeln von den konkret verwendeten Technologien unabhängig sein. Das geht am Besten mit herstellerunabhängigen Integrationsplattformen, die nötige Abstraktionen und übergreifende Funktionen technologieunabhängig bereitstellen.

Um uns einen Überblick über einige Smart-Home-Systeme zu schaffen, starten wir zunächst mit den kabelgebundenen Systemen.

KNX

Mit KNX bezeichnet man ein Bus-System für die Gebäudeautomatisierung, dessen Kommunikationsprotokoll weltweit standardisiert (EN 50090, ISO/IEC 14543) ist. Auch wenn der Standard verschiedene Kommunikationsmedien (wie RF, Infrarot, Powerline) vorsieht, kommt in der Praxis meist nur die Twisted-Pair-Variante zum Einsatz. Alle Sensoren wie bspw. Schalter, Fühler und Bewegungsmelder werden über ein zusätzliches, zweiadriges Steuerkabel mit den Aktoren verbunden (Abb. 2.1).



Abbildung 2.1: Als kabelgebundenes Feldbus-System besitzt jedes KNX-System eine Schaltzentrale; im Bild sind Rollladenaktoren zu sehen

Praktisch alle großen Hersteller bieten KNX-kompatible Produkte an. Auch wenn KNX als so genanntes Feldbus-System ausgelegt ist, bei dem sich die Aktoren bei den zu steuernden Geräten (im Feld) befinden, wird im Einfamilienhaus eher eine zentrale, sternförmige Topologie bevorzugt. Für die meisten Gewerke gibt es Produkte mit einer integrierten KNX-Schnittstelle. Die Parametrierung der Anlage erfolgt über eine recht

kostspielige Software, die so genannte Engineering Tool Software (ETS).

Loxone

Loxone ist ein 2009 in Österreich gegründeter Hersteller für Smart-Home-Produkte. Sein Hauptprodukt, der Miniserver, dient als zentrale Steuereinheit. Es werden eine Reihe von Extensions angeboten, die im Schaltschrank als Reiheneinbaugeräte (REG) eingebaut werden können, falls die enthaltenen Schnittstellen nicht mehr ausreichen. Dem Benutzer stehen ein Webinterface und native Apps zur Verfügung. Die Loxone-Produktpalette stellt neben den Apps keine „nativen“ Schalter oder Sensoren bereit. Es werden vielmehr „normale“ Schalter und Sensoren über Funk oder die digitalen bzw. analogen Eingänge des Miniservers angebunden. Ebenfalls angeboten werden Erweiterungsgateways für viele weitere Systeme, wie KNX, DMX, EnOcean, 1-Wire, RS232, Infrarot und Modbus, allesamt Schnittstellenerweiterungen mit spezifischen Stärken. Die Inbetriebnahme erfolgt mithilfe der kostenlos mitgelieferten „Loxon Config“-Software.

Der höhere Aufwand kabelgebundener Systeme resultiert aus der meist sternförmigen Verkabelung, bei der die Stromkabel jeweils von der Unterverteilung in die einzelnen Räume und zum Teil sogar einzeln zu den Verbrauchern verlegt werden. Ist dies nicht möglich oder gewünscht, so ist es eine Option, auf die vorhandenen Stromkabel zurückzugreifen, wie das folgende System beweist.

digitalSTROM

Seit 2011 werden die farbigen digitalSTROM-Lüsterklemmen (Abb. 2.2) – und inzwischen auch andersförmige Produkte – der digitalSTROM AG (vormals Aizo AG) angeboten. Die Steuersignale werden nicht über ein gesondertes Steuerkabel, sondern über die Stromleitung übertragen (Power Line Communication, PLC), weshalb sich digitalSTROM besonders gut für die Nachrüstung im Bestandsbau eignet. Derzeit ist die Produktvielfalt noch recht beschränkt, wird aber laufend den Bedürfnissen des Markts angepasst. Im Gegensatz zu den vorgenannten Systemen warten die zentralen digitalSTROM-Serverkomponenten mit offenen und gut dokumentierten Schnittstellen auf, die dazu einladen, digitalSTROM im großen Kontext zu integrieren. Die Inbetriebnahme und Konfiguration erfolgt über eine kostenlos mitgelieferte Software. Neben dem Webinterface stehen auch native Apps zur Verfügung.

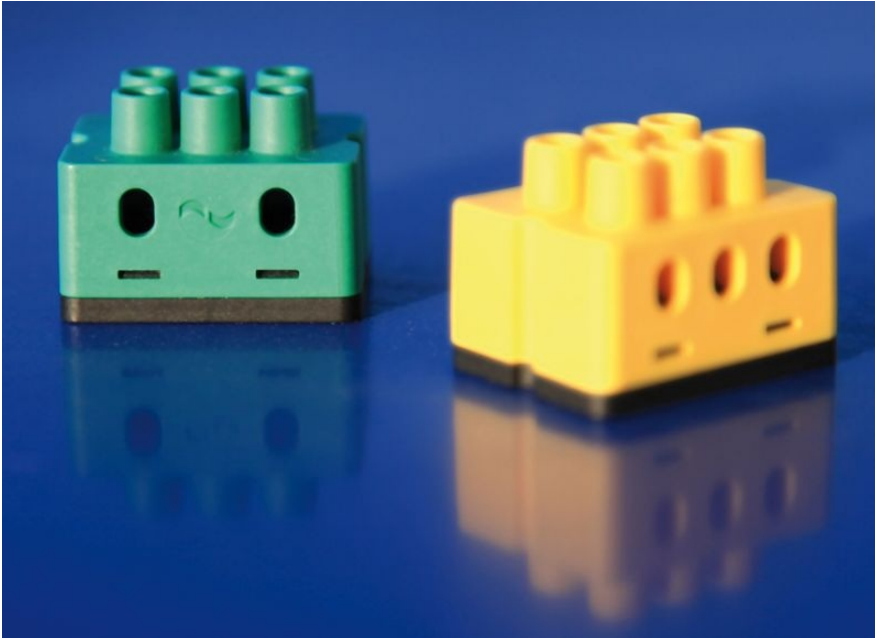


Abbildung 2.2: Die digitalSTROM-Lüsterklemmen sind in verschiedenen Farben erhältlich

Soll aus der Bestandsimmobilie oder der gemieteten Wohnung ohne umfangreiche Renovierungen ein Smart Home gemacht werden, scheiden die oben genannten Systeme aus, und Funksysteme rücken aufgrund der leichten Installation in den Fokus der Betrachtung.

Homematic

Der von der Firma eQ3 hergestellte Heimautomatisierungsbestseller heißt Homematic. Die Geräte funken im 868-MHz-Band über das proprietäre BidCos-Protokoll. Die angebotene Produktpalette ist inzwischen sehr umfangreich und lässt kaum einen Wunsch offen (Abb. 2.3). Generell werden die Geräte mit der zentralen Steuereinheit CCU gekoppelt. Über sie können dann umfangreiche Regelwerke abgebildet werden. Die Integration mit anderen Smart-Home-Integrationsplattformen kann über eine dokumentierte RPC-Schnittstelle erfolgen.



Abbildung 2.3: Die Heimautomatisierungsproduktpalette von Homematic umfasst unter anderem Lösungen für Licht, Heizung und Sicherheit

Dect ULE

Dect ist ein sehr robuster weltweiter Funkstandard, der in erster Linie in schnurlosen Telefonen Verwendung findet (Abb. 2.4). Im Jahr 2011 wurde die Erweiterung ULE (Ultra Low Energy) vorgeschlagen, um insbesondere die Laufzeit batteriegetriebener Sensoren zu erhöhen. Dect funkt im 1 880-MHz-Bereich und hat daher weniger Interferenzen als andere Systeme, die in den dicht besiedelten ISM-Bändern 868 MHz oder im 2,4-GHz-Band funken.

In Deutschland hält sich die Verbreitung und vor allem die Nutzung im Smart Home noch in Grenzen.



Abbildung 2.4: Der Hersteller Fritz! bietet Produkte an, die den Dect-ULE-Standard unterstützen

EnOcean

Der weltweite EnOcean-Standard zeichnet sich durch seinen niedrigen Energieverbrauch aus (Abb. 2.5). Einige Fensterkontakte haben beispielsweise Akkus, die durch integrierte Minisolarzellen aufgeladen werden. Darüber hinaus werden Schalter angeboten, bei denen die Bewegungsenergie des Tastendrucks für die Aussendung eines Schaltbefehls ausreicht und daher keine Batterien nötig sind.



ZigBee

ZigBee ist ein auf dem IEEE 802.15 basierender Funkstandard, der speziell für kleine Embedded-Geräte entwickelt wurde (Abb. 2.6). Er ist vor allem deswegen interessant, weil er eine Mesh-Networking-Funktionalität anbietet. Diese sorgt für eine größere Reichweite, da jeder Knoten im Netzwerk auch als Repeater fungieren und damit das Signal auffrischen kann. Obwohl ZigBee eine Reihe von Profilen (bspw. ZigBee Home Automation) anbietet, die die einheitliche Interpretation der übertragenen Daten ermöglicht, werden diese nur selten von Firmen genutzt, was zu (möglicherweise gewollten) Inkompatibilitäten führt.



Abbildung 2.6: Die Lampen aus der Philips-Hue-Modellreihe lassen sich beispielsweise via ZigBee Light Link steuern

Bluetooth Low Energy (BLE)

BLE ist die Niedrigenergievariante des Bluetooth-Standards. Im Gegensatz zu den anderen hier genannten Funkstandards ist Bluetooth explizit für den Kurzbereich, also wenige Zentimeter bis einige Meter ausgelegt. Der äußerst niedrige Energieverbrauch macht BLE insbesondere für die Anbindung so

genannter Wearables (Fitnesstracker, Smartwatches etc.) interessant, da sie ohnehin auch immer nur kurze Funkdistanzen (nämlich in der Regel bis zum Smartphone) überbrücken müssen. Für den Smart-Home-Bereich hat sich BLE vor allem durch die Verbreitung der iBeacons für die Indoor-Navigation interessant gemacht.

Z-Wave

Der Z-Wave-Funkstandard ist ein weltweit sehr gut verbreiteter Standard für Smart-Home-Produkte (Abb. 2.7). Der Z-Wave-Allianz gehören inzwischen mehr als 250 Mitglieder an, die mehr als 1 100 verschiedene Produkte auf den Markt gebracht haben. War der Standard zunächst hauptsächlich in den USA verbreitet, so scheint nun auch der europäische Markt immer mehr erschlossen zu werden. Auch Z-Wave funkt in Europa im 868-MHz-ISM-Band.



WiFi

Der sicher verbreitetste Funkstandard im Heim ist WiFi. Auch wenn inzwischen sehr viele Haushalte mit Internetanschlüssen und damit häufig auch WiFi-Routern ausgestattet sind, wird es für Smart-Home-Anwendungen vergleichsweise selten verwendet. Grund ist der recht hohe Energieverbrauch, weshalb sich eine Nutzung in batteriegetriebenen Geräten nicht anbietet.

Nachdem wir uns einen Überblick über die relevante Hardware verschafft haben, wollen wir uns nun mit den Softwareprotokollen beschäftigen. Die traurige Nachricht gleich vorne weg: Es gibt nach wie vor leider kein allgemeingültiges „Smart-Home-Protokoll“, also ein Protokoll auf Applikationsebene, über das sich beliebige Dinge im Smart Home magisch finden, verstehen und interagieren könnten. Zur besseren Übersicht haben wir nun die relevanten Kandidaten genauer beleuchtet.

MQTT (Message Queue Telemetry Transport)

MQTT ist ein auf TCP/IP aufbauendes Protokoll, das auf dem Publish-Subscribe-Verfahren beruht. Clients melden sich an einem Message-Broker, einer zentralen Softwarekomponente, an und tauschen Nachrichten über so genannte Topics aus. Der Broker sorgt für die Verteilung der Nachrichten und kümmert sich um deren Lifecycle. Ursprünglich war das Protokoll für die Anbindung verteilter Clients über eine schmalbandige und unzuverlässige Transportschicht vorgesehen. Daher wurden dem Protokoll Mechanismen zur Konsistenzprüfung, Wahrung der Reihenfolge und Nachrichtenwiederholung spendiert. MQTT ist ein OASIS-Standard und liegt inzwischen in der Version 3.1.1 vor (siehe <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html>).

In der jüngsten Vergangenheit wurde MQTT häufiger als „das Protokoll des IoT“ proklamiert. In der Tat eignet es sich aufgrund seiner Fehlerkorrekturmechanismen sehr gut zur Anbindung im Feld befindlicher Sensoren. Dennoch handelt es sich bei MQTT lediglich um ein Transportprotokoll, das keinerlei Semantik für den speziellen Anwendungsbereich mitbringt. So gibt es beispielsweise keine Konventionen zur Benennung der Topics, auf denen die Clients Nachrichten veröffentlichen. Auch ist die Bedeutung der Daten in keiner Weise

klassifiziert. Ohne spezielles Wissen über den jeweiligen Teilnehmer kann also keine Kommunikation aufgebaut werden.

Constrained Application Protocol (CoAP)

CoAP ist ein sehr leichtgewichtiges und effizientes Protokoll zur Anbindung von Devices mit sehr geringen Systemressourcen. Es nutzt UDP und ist aufgrund seiner Ausgestaltung leicht in HTTP zu übersetzen. Der Kern des CoAP-Protokolls ist im RFC 7252 (siehe <http://tools.ietf.org/html/rfc7252>) spezifiziert.

Ebenso wie MQTT ist auch CoAP aus Sicht des Smart Home eher als Transportprotokoll zu bezeichnen, weil hier ebenfalls keine spezielle Semantik für Sensoren und Aktoren enthalten ist.

Thread

Thread ist ein IPv6-basiertes Protokoll, welches kürzlich von den NEST Labs (inzwischen von Google gekauft) veröffentlicht wurde und auf dem IEEE-802.15.4-Standard aufbaut. Aufgrund dieses Aufbaus steht Thread in einer Reihe mit Bluetooth Low Energy (BLE) und ZigBee. Aufgrund seines verwendeten Softwarestacks soll Thread als reine Softwareerweiterung auf allen 802.15.4-kompatiblen Geräten verwendet werden können.

Über die versprochene Kompatibilität auf Netzwerkebene hinaus wird auch von Thread nicht die Interoperabilität auf Applikationsebene adressiert. Welchen Beitrag Thread also schließlich zur Interoperabilität der Geräte leisten wird, und ob es sich gegenüber anderen (Funk)Protokollen durchsetzen wird, muss die Zukunft zeigen.

HomeKit

HomeKit wurde von Apple als eine der beiden großen Neuigkeiten in iOS8 auf dem vergangenen WWDC vorgestellt. Es definiert Schnittstellen auf Applikationsebene, die beschreiben, wie Geräte gefunden und gesteuert werden können. Es werden Geräteprofile, wie z. B. Schalter, Thermostat, Licht und Türschloss definiert. Die so kategorisierten Geräte können dann in eine (Haus-)Hierarchie eingebunden und kontrolliert werden. Jedes einzelne Gerät, also auch einzelne Lampen, die sich hinter einer Bridge befinden, können einzeln identifiziert (identify me) und über Siri angesprochen werden. Das iPhone avanciert damit zur universellen, sprachgesteuerten Fernbedienung.

Apple spezifiziert mit HomeKit ein Protokoll, welches das Auffinden und Benutzen unterschiedlichster Geräte vereinheitlicht. Dieses Protokoll funktioniert wohlgemerkt nur in der „Apple-Welt“, da die Implementierung auf anderen Plattformen aus lizenzrechtlichen Gründen nicht erlaubt ist.

AllJoyn

AllJoyn ist ein Open-Source-Software-Framework, das 2011 von Qualcomm veröffentlicht und inzwischen der Linux Foundation übergeben wurde. AllJoyn bietet Funktionen, mit denen Geräte über alle gängigen Plattformen (Microsoft Windows, Linux, Android, iOS, OS X, OpenWRT) hinweg sich finden, kommunizieren und kollaborieren können. Das Protokoll ist dabei so implementiert, dass es grundsätzlich unabhängig vom unterliegenden Netzwerkstack arbeiten kann. Tatsächlich arbeitet beispielsweise die WiFi-Alliance derzeit am „WiFi-Direct“-Standard, der adhoc Peer-to-Peer-WiFi-Verbindungen ermöglichen wird.

Open Internet Consortium (OIC)

Auch das Open Internet Consortium, welches kürzlich von den Branchengrößen Atmel, Broadcom, Dell, Intel, Samsung und Wind River gegründet wurde, wird ein Protokoll auf Applikationsebene anbieten. Den Ankündigungen zufolge wird ebenso eine Open-Source-Implementierung veröffentlicht, die eine leichte Adaption des Protokolls ermöglichen soll. Ob diese allerdings nur den Mitgliedern des OIC oder allen Interessierten zur Verfügung stehen wird, bleibt bisher offen.

Von allen genannten Protokollen scheint AllJoyn am ehesten den Ansprüchen an ein allgemeingültiges Smart-Home-Protokoll zu genügen. Tatsächlich wurden auch auf der vergangenen IFA erste AllJoyn-kompatible Geräte wie ein Backofen der Firma AEG gezeigt. Auch wenn die AllSeen-Alliance (die Community Organisation um das AllJoyn-Protokoll) inzwischen schon eine illustre Liste von Mitgliedern hat (siehe <https://allseenalliance.org/about/members>), wird es noch eine Weile dauern, bis genug Lösungen am Markt sind, um die tatsächliche Interoperabilität beurteilen zu können.

	KNX	Loxone	digitalSTROM
Standard	✓	✗	✗
zus. Steuerleitung	✓	✓	✗
Powerline	✓	✗	✓

Konfigurationssoftware enthalten	✗	✓	✓
Produktvielfalt	groß	mittel	klein
Vendor-Locking	nein	ja	ja
Komplexität Konfiguration	mittel	niedrig	niedrig
zentrale Verkabelung	✓	✓	✗
dezentrale Verkabelung	✓	✗	✓

Tabelle 2.1: Die Merkmale von KNX, Loxone und digitalSTROM im Überblick

	Homematic	Dect ULE	EnOcean	ZigBee	Bluetooth LE	Z-Wave	WiFi
Standard	✗	✓	✓	✓	✓	✗	✓
Frequenz	868 MHz	1 880 MHz	868 MHz	868 MHz/2,4 GHz	2,4 GHz	868 MHz	2,4 GHz/5 GHz
Meshing	✗	✗	✗	✓	✗	✓	✗
IP-basiert	✗	✗	✗	✓✗	✗	✗	✓
offenes API	✗	✗	✓	✓	✓	✗	n.a.

Tabelle 2.2: Die Merkmale von Homematic, Dect ULE, EnOcean, ZigBee, Bluetooth LE, Z-Wave und WiFi im Überblick

Fazit

Die angebotenen Systeme könnten unterschiedlicher kaum sein, wie dieser Überblick über die wichtigsten Player im Smart Home zeigt: kabelgebunden oder kabellos, weltweiter Standard mit großer industrieller Akzeptanz oder proprietäres Protokoll eines Herstellers. Jedes der Systeme hat seine Stärken und Schwächen, die in der jeweiligen Situation individuell gewichtet werden müssen.

Festzuhalten bleibt: In den meisten Fällen wird im Smart Home nicht nur eines, sondern mehrere der oben genannten (oder ähnlichen) Systeme im Einsatz sein. Um dann keine Automationssilos im smarten Heim zu haben, müssen Integrationsplattformen wie openHAB, SmartThings, Revolv, Wink, Ninja Sphere, Homee oder QIVICON Vermittlungsarbeit leisten. Dazu abstrahieren diese von der jeweiligen konkreten Technologie und operieren auf einem Metadatenmodell. Wie so ein Modell aussehen kann, und welche Schwierigkeiten sich dabei ergeben, werden wir im nächsten Kapitel beleuchten.

Links & Literatur

- [1] <http://kaikreuzer.blogspot.de/2014/02/privacy-in-smart-home-why-we-need.html>
- [2] <https://nest.com/works-with-nest/>
- [3] <https://developer.apple.com/homekit/>
- [4] <https://www.alljoyn.org>
- [5] <http://www.openhab.org>
- [6] <https://www.eclipse.org/smarthome/>

3 Geräteabstraktion und ihre Schwierigkeiten

Für den Softwareentwickler ist ein Schalter nichts weiter als ein boolescher Wert. Doch bei der Programmierung von Smart-Home-Hardware ist fast jedes System bzw. Gerät unterschiedlich anzusteuern. Woran das liegt, und was man dagegen tun kann, wollen wir jetzt näher untersuchen.

Im vorigen Kapitel haben wir gesehen, wie vielfältig der Smart-Home-Markt ist und dass es nicht die „eine“ beste Lösung gibt. Vielmehr haben alle Systeme ihre Stärken und Schwächen. Es gilt also, die jeweiligen Stärken in ein Gesamtsystem zu kombinieren. Hierbei stellt man schnell fest, dass inzwischen fast jeder Hersteller bzw. jede Allianz zu einer eigenen Lösung gefunden hat, wie Geräte per Software ansprechbar sind (Abb. 3.1). Dabei sind auf den ersten Blick die Möglichkeiten nicht allzu vielfältig: Sensoren liefern einerseits Messdaten – also einfache (Fließkomma-)Zahlen – und Aktoren empfangen diskrete Kommandos wie „an“, „aus“ oder auch Prozentwerte. Wäre es für Smart-Home-Anwendungen nicht schön, wenn dies für jedes Gerät auf ein und die gleiche Weise geschehen könnte? Wir suchen also eine Abstraktionsschicht, die uns unabhängig von der konkreten Hardware macht.



Auf der Suche nach einem universellen Typsystem

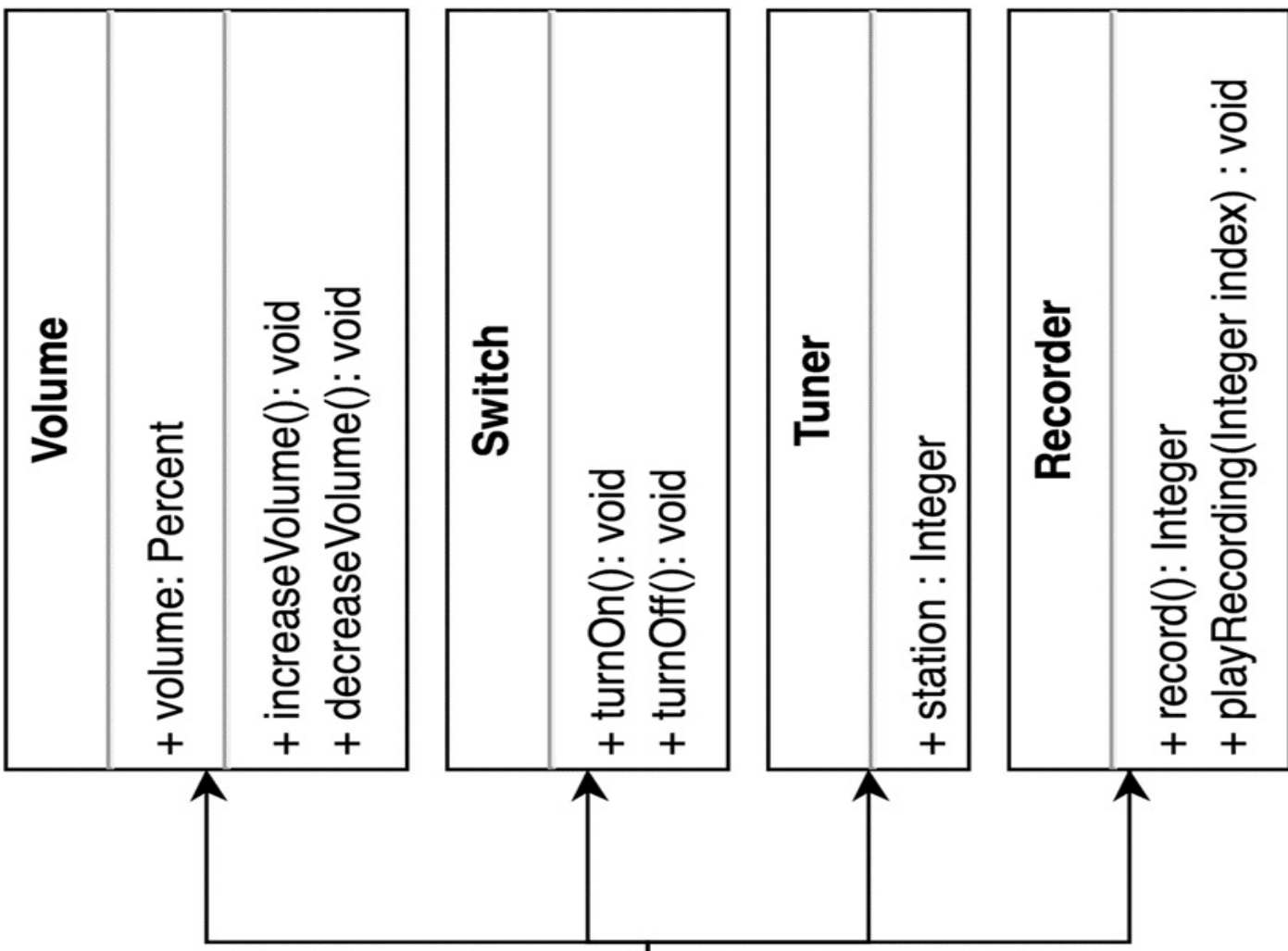
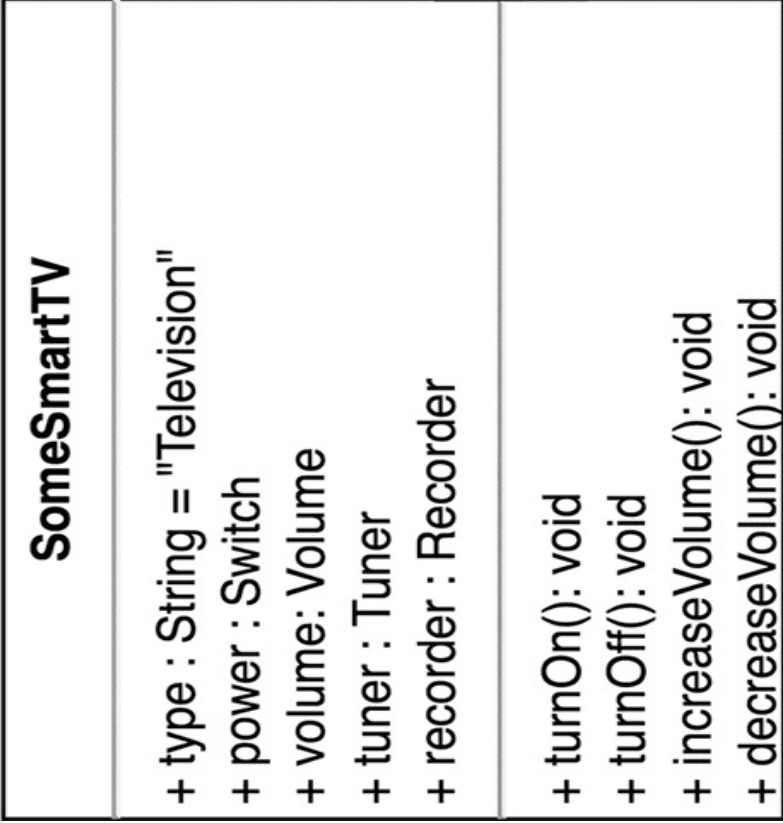
Wie uns die Informatik gelehrt hat, lassen sich alle Informationen in Form von Zahlen darstellen und in Objekten repräsentieren. Was ist also einfacher, als einen Gerätetyp als abstrakte Klasse mit Properties mit primitivem Datentyp zu beschreiben? Mögliche Kommandos lassen sich als Methoden definieren, die je nach konkreter Klasse gerätespezifisch ausimplementiert sein können – fertig ist unsere Geräteabstraktion (Abb. 3.2).

Television
+ volume: Percent + station : Integer
+ turnOn(): void + turnOff(): void + increaseVolume(): void + decreaseVolume(): void + record(): Integer + playRecording(Integer index) : void

Abbildung 3.2: Abstraktion auf Geräteebe ist zu starr, um die reale Vielfalt abdecken zu können

Doch halt! Was genau ist eigentlich der oben genannte „Gerätetyp“? Bei einer Abstraktion kann dies nicht ein konkretes Gerätemodell sein, sondern muss sich auf eine Gerätegattung beziehen, wie z. B. „Fernseher“. Wie kann man aber allgemeingültig einen Fernseher und seine Funktionen in einer abstrakten Klasse beschreiben? Zwischen dem alten Röhrenfernseher mit Einschalter, Lautstärkeregler und drei Programmen und einem aktuellen Smart-TV mit drei verschiedenen Tunern, integriertem Festplattenrekorder und eigenem App Store liegen Welten. Eine Option wäre es, einfach den größtmöglichen Umfang zu modellieren und diverse Properties und Methoden als optional zu deklarieren. Doch wie schreibt man eine Anwendung gegen ein Objekt, bei dem 95 Prozent der angebotenen Methoden eine NotImplementedException werfen? Und wie sollte man den

größtmöglichen Umfang bestimmen können? Wir müssen also einsehen, dass dies eine Sackgasse ist. Die wiederverwendbaren Bestandteile müssen demnach feingranularer gewählt werden, also auf der Ebene einzelner Funktionen. Ein Gerät ist dann einfach eine Menge von Funktionen (oft auch Capabilities genannt), und unterschiedliche Fernseher haben unterschiedliche große Mengen an Funktionen (Abb. 3.3).



Metadaten – die unvermeidbaren Spezifika

Können Gerätefunktionen nun mit primitiven Datentypen beschrieben werden? Ein an/aus wird leicht zu true/false, aber schon bei offen/geschlossen ist nicht klar, ob true/false oder false/true die bessere Abbildung ist – das ist eine Frage der Perspektive. Um ein gut verständliches Mapping zu machen, bietet es sich also an, Enumerations zu verwenden. Verständlichkeit bedeutet aber nun gleichzeitig inhärentes Wissen, was am Beispiel einer Waschmaschine gut veranschaulicht werden kann: Allgemein können wir sagen, dass eine Waschmaschine als Funktion die Auswahl des Waschprogramms hat, das über einen Integer bestimmt werden kann. Wenn wir nun aber eine Enumeration mit 1 = Wollprogramm, 2 = Buntwäsche usw. einführen, stehen wir wieder vor dem gleichen Dilemma: Unterschiedliche Waschmaschinen haben unterschiedliche Waschprogramme. Während die Repräsentation als Integer also abstrakt genug ist, können Enumerations zu spezifisch sein.

Das gilt ganz allgemein für jede Einschränkung des Wertebereichs von Funktionen: Welche Frequenzbereiche unterstützt ein Tuner? Welche minimale Schrittweite akzeptiert mein Thermostat? Welche maximale Helligkeit liefert meine Wetterstation? All diese Informationen sind also sehr gerätespezifisch und können nicht sinnvoll generalisiert werden – sie sollten als Metadaten an dem abstrakten Modell zur Verfügung stehen.

Sehr ähnlich verhält es sich mit den Einheiten: Liefert ein Sensor Messwerte, so macht er dies in einer bestimmten Einheit. Eine Abstraktionsebene sollte keine Einheit vorgeben, denn ein Temperatursensor im LHC bei CERN wird vermutlich lieber in Kelvin messen, während die Kühlschranktemperatur in Grad Celsius – in den USA aber gerne auch in Grad Fahrenheit – gemessen wird. Bei der Einheit handelt es sich demnach ebenfalls um Metadaten, die für die Interpretation ausgewertet werden können.

Konfiguration als zusätzliche Hürde

Eine spezielle Menge von Gerätefunktionen bezieht sich ausschließlich auf deren Konfiguration. Wir möchten diese explizit getrennt von den „operativen“ Funktionen betrachten, da sie sich nochmals sehr viel schwieriger abstrahieren lassen.

Betrachten wir den Fall eines einfachen Dimmers. Operativ lässt sich dessen Funktion recht einfach beschreiben: Er hat einen Status von x Prozent und akzeptiert einen Wert zwischen 0 und 100, um die gewünschte Helligkeit zu wählen. Um dieses Verhalten aber zu konfigurieren, werden oft zahlreiche Optionen angeboten: Welche Art Leuchtmittel wird verwendet, welche Dimmkurve ist also am besten geeignet? Wie schnell wird der neue Wert angefahren? Darf das angeschlossene Gerät überhaupt gedimmt werden oder soll nur ein- und ausgeschaltet werden? Soll beim Einschalten auf 100 Prozent geregelt werden oder auf den zuletzt eingestellten Wert vor dem letzten Abschalten? Man sieht, selbst eine so einfache Funktion kann sehr komplex in ihren Konfigurationsmöglichkeiten sein. Aus diesem Grund ist es ratsam, die Gemeinsamkeiten von Geräten nur auf der operationellen Seite zu suchen und diese zu abstrahieren und mit der Tatsache zu leben, dass es sehr hardwareindividuelle Konfigurationsoptionen gibt.

Die Essenz der Dinge

Denkt man näher über den Dimmer nach, so stellt man fest, dass das, was den Anwender interessiert, gar nicht der Dimmer selbst, sondern die daran angeschlossene Lampe ist. Auch ob ein Magnetkontakt geöffnet oder geschlossen ist, ist lediglich sekundär. Die relevante Information ist hingegen, ob die Tür offen steht oder nicht. Noch offensichtlicher ist der Fall bei einem Bluetooth-Tag im Hundehalsband: Das Halsband ist auch hier nur Mittel zum Zweck, um den Hund zu lokalisieren. Ebenso interessiert nicht die Temperatur eines Sensors, sondern nur die des Raums, in dem er sich befindet. Eine abstrakte Beschreibung braucht also Dinge wie „Haustier“, „Person“, „Tür“, „Raum“, „Lampe“, und die Geräte sind nur Hilfsmittel, um Informationen über diese zu sammeln oder Aktionen auszulösen. Hierbei redet man üblicherweise von einer Ontologie – also einem festgelegten Begriffssystem, in dem die Dinge in definierten Beziehungen zueinander stehen (Abb. 3.3). Die Beziehungen der Ontologie ergeben die Semantik, d. h. Informationen, aus denen konkrete Aussagen abgeleitet werden können, da der Kontext „verstanden“ wird.

Eine Ontologie bedeutet aber für eine Geräteabstraktion schwerwiegende Probleme: Der Gerätehersteller kann niemals das Wissen über dessen konkrete Verwendung haben, und somit gibt es keine Eingliederung eines Geräts in die Ontologie „ab Werk“. Vielmehr hat nur der Installateur bzw. Nutzer das notwendige Wissen über den Einsatzzweck, also den Kontext.

Ein einfaches Beispiel lässt sich mit einem „Alles aus“-Use-Case veranschaulichen: Keine noch so gute Geräteklassifikation kann bestimmen, welche Zwischenstecker beim Verlassen des Hauses die angeschlossenen Geräte vom Strom trennen dürfen. Beispielsweise könnte ja auch ein Kühlschrank daran angeschlossen sein, für den der Zwischenstecker nur dazu dient, den Energieverbrauch aufzuzeichnen. Solches Wissen muss das System also entweder initial mitgeteilt bekommen oder durch Benutzerinteraktion im Laufe der Zeit erlernen.

Ein weiteres Problem ist die Kreativität der Nutzer: Wer weiß schon, wofür sie einen Magnetkontakt verwenden? Kleben sie ihn an ein Fenster, an eine Tür oder vielleicht an den Briefkasten? Oder doch eher an den Kühlschrank, den Safe oder die Mausefalle? Sollen all diese Nutzungsszenarien abgedeckt und semantisch erfasst werden, so muss das Begriffssystem quasi das komplette Vokabular aller in Häusern vorhandenen und verwendeten Dinge umfassen – eine unlösbare Aufgabe.

Als Konsequenz der geschilderten Schwierigkeiten muss man ziehen, dass der Nutzer in vielen Fällen für notwendige Zusatzinformationen sorgen muss und es insofern eher Sinn ergibt, Applikationen so zu gestalten, dass sie nicht notwendigerweise das Wissen über den Kontext selbst benötigen, sondern den Nutzer so führen, dass er den Kontext implizit selbst einfließen lässt, während er seine Use Cases definiert.

Class hierarchy (inferred)

Class hierarchy: 'Multi level sensor'



- Thing
 - Device
 - Function
 - Alarm
 - 'Boolean control'
 - 'Boolean sensor'
 - Keypad
 - Meter
 - 'Multi level control'
 - 'Multi level sensor'
 - 'Wake up'
 - 'function type'
 - OperationMetadata
 - PropertyAccess
 - PropertyMetadata
 - 'Service status'
 - 'Service status detail'
 - 'Unit of measure'

Annotations

Annotations: 'Multi level sensor'



label [type: string]
Multi level sensor

Description: 'Multi level sensor'



- 'has property name' value "data -contains the current state of MultiLevelSensor"^^string
- ('has function type' value Door) or ('has function type' value Flow) or ('has function type' value Gas) or ('has function type' value Humidity) or ('has function type' value Light) or ('has function type' value Liquid) or ('has function type' value Noisiness) or ('has function type' value Other) or ('has function type' value Power) or ('has function type' value Pressure) or ('has function type' value Rain) or ('has function type' value Smoke) or ('has function type' value Temperature) or ('has function type' value Window)

Function

Ortsbestimmung

Eine Folgerung aus der Trennung von Gerät (genauer: einer Gerätefunktion) und dessen Verwendung ist, dass der Ort unterschiedlich ausfallen kann. „Alle Lampen im Wohnzimmer abschalten“ sollte beispielsweise auch die entsprechenden Schaltkanäle des Schaltaktors im Zählerschrank mit einbeziehen. Ein solcher Schaltaktor hat also als Gerät einen „Installationsort“, hingegen mit seinen Funktionen mehrere verschiedene „Wirkorte“. Diese Unterscheidung ist sehr essenziell und deckt sich weitestgehend mit der Trennung von Konfiguration und Operation eines Geräts: Für die Konfiguration ist dessen Installationsort der wichtige Aspekt, für die tatsächliche tagtägliche Nutzung hingegen die jeweiligen Wirkorte. Auch hier ist es wieder der Installateur oder Nutzer, der dem System mitteilen muss, welche Wirkorte es für die Funktionen eines Geräts in seinem Fall gibt.

Tücken der Laufzeit

Ist erst einmal alles klassifiziert, in generischen Einzelfunktionen beschrieben und konfiguriert, gelangt man zur nächsten Hürde: dem Laufzeitverhalten. So schön die einmalige statische Beschreibung auch ist, die Dynamik im Betrieb kann alles sehr schnell wieder durcheinander bringen:

- Abhängig von der (zur Laufzeit gewählten) Betriebsart, kann sich die Menge der angebotenen Funktionen komplett verändern. Eine Heizungssteuerung im Automatikmodus hat beispielsweise deutlich weniger Optionen als im manuellen Modus. Wie kann man solche wechselnden Funktionsumfänge formal sinnvoll beschreiben?
- Wie oft werden Sensordaten übermittelt? Passiert das in festen Intervallen oder bei Veränderung (bei welcher Schwelle) des Messwerts? Dieses Verhalten kann entscheidende Auswirkungen auf die Art und Weise haben, wie Applikationen mit diesen Daten interagieren.
- Was genau passiert nach dem Abschicken eines Kommandos an ein Gerät? Wird es überhaupt sofort übermittelt oder kann dies einige Zeit dauern, bis das Gerät wieder „lauscht“, wie dies oft bei batteriebetriebenen Lösungen der Fall ist? Wie lange dauert die Ausführung des Kommandos? Ein Rollladen braucht beispielsweise eine

erhebliche Zeit, um in eine bestimmte Position zu fahren.

- Welche Fehlerfälle gibt es und wie und wann meldet dies das Gerät? Sind diese „recoverable“, d. h. durch geeignete Fehlerbehandlung abfangbar?
- Welche Zustände durchläuft ein Gerät bei einer Aktion, und wie können gültige Kommandos für gewisse Zustände bestimmt werden? Braucht ein Gerät gar Transaktionalität?

Zu diesen Fragestellungen gibt es in der Praxis nur selten ausgereifte Lösungen. Die bevorzugte Behandlung ist eine statische Beschreibung mit dann in der Applikation implizit implementierten Fallbehandlungen. Eine wirkliche Abstraktion wird aus diesen Gründen nur selten erreicht. Vielmehr bleiben Applikationen bis zu einem gewissen Grad immer gerätespezifisch, auch wenn dies oft auf den ersten Blick nicht deutlich wird.

Fazit

Zusammenfassend darf man festhalten, dass Geräteabstraktion ein wichtiges und zugleich hochkomplexes Thema ist. Auch die EU-Kommission beschäftigt sich derzeit damit und hat eine Untersuchung bestehender Abstraktionen in die Wege geleitet. Erste Berichte dazu wurden unter [1] und [2] veröffentlicht. Als Ergebnis wird eine vereinheitlichte Ontologie in Aussicht gestellt – aufgrund der zahlreichen in diesem Kapitel aufgeworfenen Fragen dürfen wir darauf sehr gespannt sein. Im folgenden Kapitel werden wir uns näher mit den Allianzen sowie dem aktuellen Stand der Forschung im Smart-Home-Bereich beschäftigen.

Links & Literatur

[1] https://docs.google.com/file/d/0B2nnxMhTMGh4M2F6c2JLRfpEcXc/edit?usp=drive_web

[2] https://docs.google.com/file/d/0B2nnxMhTMGh4alB0MGVRcHBKZWs/edit?usp=drive_web

4 Standards, Open Source und Forschung

Der Internet of Things und im Speziellen der Smart-Home-Markt sind so breit, dass einzelnen Anbietern schlicht das Durchhaltevermögen für die große Lösung fehlt. Ein Ausweg sind Allianzen, ein anderer echte Standards. Im folgenden Kapitel wollen wir uns einen Überblick verschaffen und eine Einordnung vornehmen.

Auch wenn in Fachkreisen schon lange über das Smart Home diskutiert und geforscht wird und es sicher auch schon den einen oder anderen (Smart-)Häuslebauer gibt, so ist das Thema „Smart Home“ (leider) immer noch kein Massenphänomen. Zu umfangreich ist das Angebot schon jetzt, und es kommen täglich neue Produkte auf den Markt. Zu selten finden sich echte Spezialisten, die bei der Beratung, bei der Installation und vor allem bei der unweigerlich folgenden Erweiterbarkeit helfen können (siehe Kapitel 2).

Die Ursache für die große Unsicherheit lässt sich dabei auf technischer Ebene recht schnell ausmachen: Die meisten angebotenen Produkte sind schlicht inkompatibel und können daher nicht miteinander kombiniert werden.

Zum einen liegt das an den unterschiedlichen Protokollen auf der Transportebene. Hier konkurrieren Funkprotokolle wie Z-Wave, BidCoS, EnOcean, ZigBee, Bluetooth und WiFi gegen die kabelgebundenen Systeme und unter diesen wiederum die Powerline-basierten Lösungen gegen die Systeme mit zusätzlichen Steuerleitungen.

Selbst wenn man dieses Problem gelöst und sich auf ein einheitliches Transportprotokoll wie beispielsweise IP geeinigt hätte, gäbe es auch auf den darüber liegenden Schichten derzeit noch keine guten Antworten auf offene Fragen, wie etwa nach der protokollübergreifenden Repräsentation von Geräten oder Regeln oder dem einheitlichen Einrichtungsprozess. Auch hier kochen viele Hersteller lieber ihr eigenes Süppchen anstatt an gemeinsamen Ideen zu arbeiten. Dabei liegen die Lösungen für das Problem unterschiedlicher Protokolle und vorherrschender Inkompatibilität auf der Hand. Es könnte a) ein Standard aus der Taufe gehoben werden, der es Geräten unterschiedlicher Hersteller ermöglicht, sich in einem Netzwerk

(transport-)protokollunabhängig zu finden, ihre Fähigkeiten (Capabilities) auszutauschen, sich zu verbinden/pairen und sich schließlich gegenseitig entsprechend ihrer Fähigkeiten zu benutzen. Im Gegensatz dazu könnte b) eine Integrationsplattform verwendet werden, die als zentrale Drehscheibe verschiedene Schnittstellentechnologien adaptiert, Protokolle übersetzt und ihrerseits einheitliche Schnittstellen bereitstellt. Die wahrscheinlichste Lösung wird vermutlich c) eine Mischform aus a) und b) sein. In diesem Szenario integriert die Integrationsplattform einige wenige Applikationsprotokolle und ermöglicht damit eine einheitliche Nutzung.

(Keine) Einigkeit auf Applikationsebene

Sind wir mal ehrlich: Wäre es nicht ein Traum, wenn alle Geräte im Smart Home dem gleichen Applikationsprotokoll gehorchen würden? Keine Inkompatibilität zwischen Farblampe und Waschmaschine sowie zwischen Router und Garagentor? Einfach das jeweilige Gerät einstecken und der Rest geschieht automatisch?

Erreicht werden könnte das durch ein einheitliches Applikationsprotokoll, das für alle nötigen Ebenen Spezifikationen bereithält. Verantwortlich für den Erfolg des jeweiligen Applikationsprotokolls ist dann neben seiner Qualität auch die Adoption-Rate, also die Verbreitung im Markt. Um die kritische Masse dafür zu erreichen, neigen Unternehmen dazu, sich in Allianzen zusammenzuschließen.

Fast zwanzig solcher Allianzen haben wir bei unseren Recherchen mit dem Fokus Internet of Things und Smart Home gefunden (Tabelle 4.1), deren interessanteste Vertreter wir im Folgenden etwas genauer betrachten wollen.

AllSeen

AllSeen wurde 2013 von Qualcomm als Stiftung zur Verwaltung und Betreuung des Open-Source-Frameworks AllJoyn gegründet. AllJoyn selbst wurde bereits 2011 von Qualcomm gegründet und dann 2013 an die AllSeen Alliance (unterhalb der Linux Foundation) übergeben. AllJoyn ist ein Applikationsframework, das Funktionen anbietet, mit denen sich Geräte über alle gängigen Plattformen (Microsoft Windows, Linux, Android, iOS, OS X, OpenWrt) hinweg finden und miteinander kommunizieren und kollaborieren können. Erste AllJoyn-Geräte sind bereits auf dem Markt (siehe Backofen der Firma AEG Electrolux [1]) und viele weitere sind angekündigt. Problematisch scheint bei AllSeen zu sein, dass die

verwendete ISC-Open-Source-Lizenz keine Patentklausel hat. Auch verfolgt die Allianz scheinbar keine Pläne, ihr Framework zu standardisieren, was eine Portierung, beispielsweise in andere Programmiersprachen, erschweren dürfte.

OIC – Open Interconnect Consortium

Diese Allianz wurde Anfang 2014 von den Branchengrößen Atmel, Broadcom, Dell, Intel, Samsung und Wind River als direkter „Konkurrent“ zu AllSeen gegründet. Neben der Standardisierung des Applikationsprotokolls wird auch eine Referenzimplementierung angeboten. Diese wurde in diesen Tagen unter dem Projektnamen IoTivity veröffentlicht [2]. Im Gegensatz zu AllSeen veröffentlicht das OIC allen Sourcecode unter der Apache-Lizenz.

Thread

Thread ist ein IPv6-basiertes Protokoll, das 2014 von den Nest Labs (inzwischen von Google gekauft) veröffentlicht wurde und auf dem IEEE-802.15.4-Standard aufbaut. Aufgrund seines verwendeten Softwarestacks soll Thread durch Firmwareersatz auf 802.15.4-kompatiblen Geräten verwendet werden können. Leider ist über die Offenheit und die Verbreitung außerhalb der Nest-Gruppe noch wenig bekannt.

HomeKit

HomeKit wurde von Apple als eine der beiden großen Neuigkeiten in iOS 8 auf der vergangenen WWDC vorgestellt. Es definiert Schnittstellen auf Applikationsebene, die beschreiben, wie Geräte gefunden und gesteuert werden können. Es werden Geräteprofile, wie z. B. Schalter, Thermostat, Licht und Türschloss definiert. Die so kategorisierten Geräte können dann in eine (Haus-)Hierarchie eingebunden und kontrolliert werden. Jedes einzelne Gerät, also auch einzelne Lampen, die sich hinter einer Bridge befinden, können einzeln identifiziert (identify me) und über Siri angesprochen werden. Das iPhone avanciert damit zur universellen, sprachgesteuerten Fernbedienung. HomeKit funktioniert wohlgemerkt nur in der „Apple-Welt“, da die Implementierung auf anderen Plattformen aus lizenzrechtlichen Gründen nicht erlaubt ist.

QIVICON

QIVICON ist eine von der Deutschen Telekom initiierte Allianz vornehmlich deutscher Unternehmen, die seit 2011 gemeinsam an der Entwicklung einer

Smart-Home-Plattform arbeiten. Dabei geht es nicht nur um Software, sondern um eine Gesamtlösung aus Software, Hardware und Cloud-Diensten, die durch die Partner direkt an Endkunden vermarktet werden kann. Softwareseitig wird versucht, auf offene Standards und Open Source zu setzen [3].

Eclipse IoT

Die Eclipse-IoT-Community wurde 2013 unter dem Namen Eclipse M2M gestartet und im Laufe des Jahres 2014 aufgrund des deutlich erweiterten Fokus in Eclipse IoT umbenannt. Der Working Group gehören heute mehr als zweiundzwanzig Organisationen an, die in mehr als siebzehn Projekten engagiert sind. Das Ziel von Eclipse IoT ist es, (quell)offene Frameworks und Implementierungen offener IoT-Standards anzubieten.

EEBus Initiative e. V.

Die EEBus Initiative.V. wurde 2011 aus dem Förderprogramm E-Energy des BMWi und BMU mit dem Ziel gegründet, die Energieeffizienz durch „den Austausch von Anwendungen und Diensten“ zu erhöhen. In diesem geförderten Projekt wurde in erster Linie an einer (Daten-)Spezifikation für den Datenaustausch der verschiedenen Beteiligten insbesondere im Smart-Grid-Bereich gearbeitet (der eigentliche EEBus). Darüber hinaus ist eine Integrationsplattform namens SHIP zur Verbindung der im EEBus spezifizierten Datenmodelle über IP-Technologien beschrieben worden [4].

Die meisten Ergebnisdokumente sind nur Mitgliedern des EEBus e. V. sowie einiger assoziierter Verbände (VDE, BITKOM) zugänglich, sodass derzeit nur schwer Aussagen über den Status und die Relevanz für das Smart Home gemacht werden können.

Die schiere Anzahl der Allianzen lässt schon vermuten, dass es mehr gibt, als es eigentlich brauchen würde – vorausgesetzt, man wäre wirklich an Interoperabilität und dem einen übergreifenden Applikationsprotokoll interessiert. Häufig spielen bei kommerziellen Entscheidungen dann eben doch nicht nur die harten Spezifikationen, sondern auch weiche Faktoren wie Not Invented Here oder ähnliche Befindlichkeiten eine Rolle. Deswegen gründet man doch lieber seine „eigene“ Allianz, statt einer existierenden beizutreten.

Erfreulicherweise gibt es unter den Allianzen inzwischen dennoch erste Zusammenschlüsse. Ende letzten Jahres kündigten Agora (Frankreich),

EEBus (Deutschland) und Energy@home (Italien) ihre Zusammenarbeit auf europäischer Ebene an [5]. Am Ende wird der Markt entscheiden – es werden sich De-facto-Standards bilden, die langfristig zu einer Konsolidierung des Markts führen werden.

Allianz	Gründung	Sitz	Ausrichtung	Software/Spezifikation verfügbar
Agora (http://bit.ly/1ASt80p)	2014	F	SO	für Partner/Mitglieder
AllSeen Alliance (http://bit.ly/13AuNtp)	2013	USA	SO	öffentlich
Connected Comfort (http://bit.ly/1z1mg06)	?	D	B	ohne Angabe
Connected Living e.V. (http://bit.ly/1BbeP5T)	2009	D	H/SO	für Partner/Mitglieder
Continua (http://bit.ly/1KsvPJ3)	2012	USA	SO/Z	für Partner/Mitglieder
EEBus Initiative e. V. (http://bit.ly/1sdt88a)	2009	D	H/S/SO	für Partner/Mitglieder
Energy@home (http://bit.ly/1C9LuHn)	2012	I	S/SO	für Partner/Mitglieder
HomeKit (http://bit.ly/1mL5gj7)	2014	USA	SO	für Partner/Mitglieder
IIC (http://bit.ly/1pA9JXV)	2014	USA	S/SO	öffentlich
IOT Consortium (http://bit.ly/1AwKH3j)	2014	USA	/	/
M2M Alliance (http://bit.ly/1DG9Tp9)	2007	D	/	ohne Angabe
OIC/loTivity (http://bit.ly/1Blxjec)	2014	USA	S/SO/Z	öffentlich
QIVICON (http://bit.ly/1wClcd7)	2011	D	H/SO	für Partner/Mitglieder
SmartHome Initiative Deutschland e.V. (http://bit.ly/1y46jF0)	2008	D	/	ohne Angabe
Thread (http://bit.ly/1w5CfmS)	2014	USA	SO	für Partner/Mitglieder
UNIVERSAL HOME				

(http://bit.ly/14PwraR)	?	D	H/SO	ohne Angabe
Web of Things (http://bit.ly/1DvlQAJ)	2013	USA	SO	für Partner/Mitglieder

Tabelle 4.1: Smart-Home-/IoT-Allianzen im Überblick; Erklärung der Ausrichtung: Beratung (B), Hardware (H), Software (SO), Spezifikation (S), Zertifizierung (Z)

Standardisierung

Neben den eher auf kommerzielle Interessen ausgerichteten Allianzen sind auch die „klassischen“ Gremien (Tabelle 4.2) um Standardisierung im Internet of Things bemüht. Bereits vor mehr als fünfzehn Jahren startete beispielsweise die OSGi Alliance mit der Spezifikation einer modularen und flexiblen Laufzeitumgebung für Embedded Gateways. Weitere Initiativen wie oneM2M und ETSI M2M kommen ursprünglich aus dem M2M-Bereich, haben ihren Fokus aber inzwischen auch auf das Internet of Things ausgeweitet. Andere eher Telko-lastige Organisation wie IETF und OMA bringen große Erfahrung in der Anbindung mobiler (Embedded-)Geräte über die Luftschnittstelle mit und können daher mit entsprechend spezialisierten Protokollen (kleiner Footprint, garantierte Zustellung) aufwarten.

Im Gegensatz zu den oben aufgeführten Industrieallianzen, deren Interesse auch manchmal nur darin besteht, den eigenen Technologien einen Vorteil zu verschaffen, bemühen sich die Standardisierungsinitiativen (SDO), Standards in Form von meist technologieunabhängigen Spezifikationen bereitzustellen.

Standardisierungsinitiative	Gründung	Ausrichtung	Protokoll/Standards	Verfügbarkeit
Eclipse IoT (http://bit.ly/1hJhhGk)	2014	SO	CoAP, MQTT, ESH	öffentlich
ETSI M2M (http://bit.ly/1tTcTOJ)	2012	S	siehe http://bit.ly/1tTcTOJ	öffentlich
HGI (http://bit.ly/1BV4mdf)	2004	S	siehe http://bit.ly/1C3t2IF	öffentlich
IEC (http://bit.ly/17BeKxp)	1906	S	IEC 60870-5-101, IEC 60870-5-104, IEC 61850 u. v. m.	kostenpflichtig
IEEE IoT (http://bit.ly/14Eb4jr)	2013	S	IEEE P2413	kostenpflichtig
IETF	2012	S	CoAP, DTLS, 6LoWPAN	öffentlich

(http://bit.ly/1DGap6G)				
IPSO (http://bit.ly/14CRRZk)	2008	SO	Smart Objects	öffentlich
OASIS (http://bit.ly/1tQYi0l)	2014	S	MQTT	öffentlich
OMA (http://bit.ly/1AwPxxG)	2002	S	LWM2M	öffentlich
oneM2M (http://bit.ly/1I2chZL)	2012	S/SO	siehe http://bit.ly/14R1Yt3	öffentlich
OSGi Alliance (http://www.osgi.org/Markets/SmartHome)	1999	S	OSGi	öffentlich
W3C Web of Things (http://bit.ly/14FJyMc)	2013	S	/	öffentlich

Tabelle 4.2: Standardisierungsinitiativen mit dem Fokus auf IoT/M2M; Erklärung der Ausrichtung: Beratung (B), Hardware (H), Software (SO), Spezifikation (S), Zertifizierung (Z)

Interesse der Forschung

Die Entwicklung neuer Standards ist nicht zuletzt auch das Ergebnis intensiver Forschung und der Kooperation von Bildung und Wirtschaft. Darum haben neben den Anbietern und Herstellern auch die Forschungseinrichtungen großes Interesse am Internet of Things bzw. am Smart Home. Die Interessensgebiete erstrecken sich dabei von der Entwicklung intelligenter und vor allem effizienter Protokolle auf allen Schichten, über Themen zur Energieeffizienz, künstlicher Intelligenz, der Optimierung der Bedienungsmöglichkeiten bis hin zur Entwicklung neuer Geschäftsfelder im Dienstleistungsbereich.

Ein besonders interessantes Forschungsgebiet ist der Bereich Ambient Assisted Living (AAL). Hier wird das Ziel verfolgt, Menschen so lange wie möglich ein selbstbestimmtes Leben in den eigenen vier Wänden zu ermöglichen. Neben den psychologischen Aspekten stehen dabei auch der Kostendruck im Pflegebereich und der fortschreitende demografische Wandel im Vordergrund.

In diesem Kontext hilft die intelligente Haustechnik beispielsweise beim Verlassen des Hauses unnötige oder gefährliche Stromverbraucher abzuschalten oder sie entdeckt ungewöhnliche Situationen wie einen Sturz der Bewohner. Dank der allgegenwärtigen, aber für den Bewohner dezent im Hintergrund befindlichen Technik können z. B. Servicerufe automatisch

ausgelöst werden und damit gefährliche Situation vermieden und die Rettungszeit deutlich verkürzt werden.

Gerade (aber nicht ausschließlich) bei AAL kommen viele spezialisierte Sensoren wie Kameras, Belegungssensoren und intelligente Fußböden zum Einsatz, um ungewöhnliche Situationen möglichst schnell und fehlerfrei erkennen zu können.

Der praktische Einsatz dieser neuen Technologien wird häufig in so genannten „Living Labs“ getestet. Praktisch jede Forschungseinrichtung mit dem Interessengebiet „Smart Home“ betreibt daher ein mehr oder weniger groß ausgelegtes Lab und testet darin das Zusammenspiel der zu erforschenden Komponenten. Beispiele für solche Labs finden sich in Tabelle 4.3.

FZI Karlsruhe	https://www.fzi.de/forschung/fzi-house-of-living-labs/fzi-living-lab-smarthomeaal/
Hochschule Ruhr West	http://www.hochschule-ruhr-west.de/forschung/institut-energiesysteme-und-energiewirtschaft/forschung-und-projekte/smart-home-lab.html?PHPSESSID=78c63cf08ee30d1a4ee4602a91a549ba
Fraunhofer-inHaus-Zentrum	http://www.inhaus.fraunhofer.de/de/inHaus_entdecken.html
DFKI Bremen	http://www.dfki.de/web/living-labs-en/baall-bremen-ambient-assisted-living-lab?set_language=en&cl=en
TU Dortmund	http://www.kt.e-technik.tu-dortmund.de/cms/de/forschung/Laufende_Projekte/Connected_Home_Lab/index.html
FH Köln	http://www.verwaltung.fh-koeln.de/aktuelles/2013/08/verw_msg_06233.html
RWTH Aachen	http://www.humtec.rwth-aachen.de
TU Kaiserslautern	http://www.assistedliving.de

Tabelle 4.3: Beispiele für Living Labs der Forschungseinrichtungen

Offen und (idealerweise) Open Source

Auch wenn sich in der Leserschaft sicher leicht Konsens darüber erzielen ließe, dass Standards im weitesten Sinne „offen“ sein sollten, ist das nicht der Normalfall. Einige Standardisierungsorganisationen verkaufen die Spezifikationen ihrer Standards zu teils erheblichen Lizenzgebühren. Die Allianzen erheben in der Regel Mitgliedsbeiträge, für welche die Mitglieder dann im Gegenzug Zugriff auf die Dokumente erhalten.

Für die gute und schnelle Verbreitung eines Standards ist es allerdings erforderlich, die Zugangshürde so niedrig wie möglich zu halten. Aus diesem Grund liefern einige Standardisierer neben der Spezifikation auch

eine Referenzimplementierung (idealerweise unter einer gängigen, kommerziell freundlichen Open-Source-Lizenz) zur freien Nutzung mit. Hiermit wird potenziellen Nutzern eine möglichst große Rechtssicherheit in Bezug auf die Verwendung und die weitere Distribution der Quellen gegeben.

Im besten Fall stehen nicht nur die Spezifikation und die Protokolle, sondern auch die Integrationsplattform selbst quelloffen zur Verfügung, um sie schnell an die Bedingungen des sich rasant bewegenden Internet-of-Things-Markts anpassen zu können. Für Nutzer und Anbieter gibt es dabei gleichermaßen gute Argumente für den Einsatz offener statt geschlossener Plattformen. Aus Nutzersicht:

- Keine ausschließliche Bindung an den Anbieter der Hard-/Software (Vendor Lock-in)
- Community, die neben dem Anbieter bei der (Weiter-)Entwicklung der Plattform und bei technischen Problemen hilft
- Fortbestand des Projekts, auch wenn Protagonisten wechseln, da Quellen weiter verfügbar bleiben (bei vernünftiger Lizenz)
- Keine Lizenzkosten
- Investitionssicherheit aufgrund der gegebenen Erweiterbarkeit

Aus Sicht der Systemanbieter wird leider häufig schon wesentlich weniger als „offene Plattform“ bezeichnet. Für sie reicht es meist schon aus, dass die Plattform (technisch) grundsätzlich erweitert werden könnte. Nur in seltenen Fällen sind bei kommerziellen Lösungen jedoch die Kriterien für echte Offenheit wie die Verfügbarkeit der Quellen unter einer anerkannten Open-Source-Lizenz, die Verfügbarkeit einer (Entwickler-)Dokumentation und die aktive Unterstützung der Community vorhanden. Gegen die Öffnung der Plattform könnte aus Sicht der Anbieter auch sprechen:

- Kannibalisierung: Wenn die Produkte austauschbar sind, ist das möglicherweise mit Umsatzeinbußen verbunden.
- Mehr Support: Neben den „normalen“ Nutzern sind plötzlich die Entwickler zu unterstützen, die mit diesem neuen Wissen dann eigene Geräte/Applikationen anbieten, was wiederum zu weiteren Umsatzeinbußen führen könnte.
- Fehlende Kontrolle: Die Entwicklungswünsche der Community gehen vor kommerzielle Interessen des Anbieters.

- Sicherheit: Wenn die Quellen verfügbar sind, sind mögliche Sicherheitslöcher leichter zu finden, und das System wird angreifbarer.

Diesen vermeintlich negativen Argumenten stehen aus unserer Sicht aber eine Reihe positiver Argumente gegenüber:

- Weniger Support: Neben dem Anbieter selbst (wenn es denn überhaupt einen gibt) hilft auch die Community bei der Lösung technischer Probleme.
- Erhöhte Entwicklungsgeschwindigkeit: Weil nicht mehr nur das eigene Entwicklungsteam, sondern auch externe Entwickler mithelfen, kann in der Summe der Output steigen.
- Höhere Sicherheit: Aufgrund der Quelloffenheit können alle Entwickler (nicht nur die des Anbieters) die Sicherheitsmechanismen reviewen und bewerten. Sollten Lücken vorhanden sein, werden diese schneller entdeckt und behoben als im Closed-Source-Fall.

Fazit

Um das Smart Home tauglich für den Massenmarkt zu machen, müssen die Anbieter immer noch einige Hausaufgaben erledigen. Nach den Analysten von Deloitte werden allzu oft technische Details und kostenorientierte Anwendungsfälle einer Lösung hervorgehoben, anstatt auf nutzungorientierte Vermarktung, die Betonung des Verbraucherlifestyles, eine intelligente Preisgestaltung, die Bündelung von Produkt und Installation, Transparenz über Verkaufskanäle und offene Plattformen zu setzen [6].

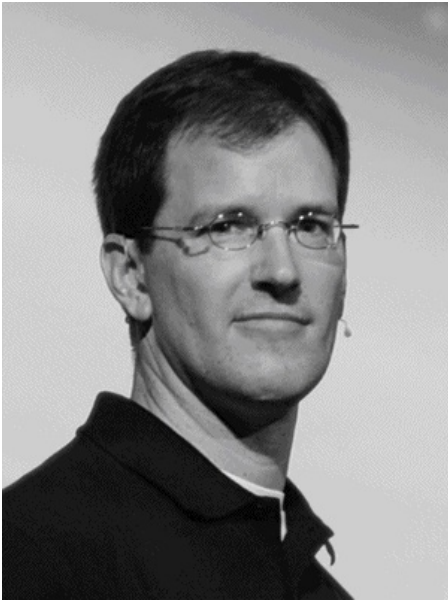
Damit offene Plattformen entstehen können, bedarf es zunächst offener Standards und quelloffener Implementierungen dieser Standards. Und hier liegt der Hase im Pfeffer: Haben es die Standardisierungsgremien mittlerweile geschafft, relevante und gute Protokolle auf den unteren Ebenen zu spezifizieren, sieht es auf der Applikationsebene bisher sehr dünn aus. Hier bringen auch die zahlreich entstandenen (und weiterhin entstehenden) Allianzen kaum Besserung. Mit AllSeen und OIC scheinen zwei Kandidaten mit großem Potenzial auf dem Markt zu sein, die erfahren genug wären, diese Mammutaufgabe der Spezifizierung eines solchen Protokolls zu stemmen. Die Zeit wird zeigen, welcher der Kandidaten dauerhaft mit den Entwicklungen des Internet of Things mithalten bzw. diese mitbestimmen kann.

Im nächsten Kapitel wird es wieder etwas technischer und wir wollen uns mit den Aspekten Steuerung und Automatisierung beschäftigen.

Links & Literatur

- [1] <http://newsroom.electrolux.com/de/2014/09/01/electrolux-tritt-allseen-alliance-bei-um-nahtlose-vernetzung-von-geraten-zu-ermöglichen/>
- [2] <https://www.iotivity.org>
- [3] <http://www.heise.de/developer/meldung/Deutsche-Telekom-unterstuetzt-Eclipse-bei-Heimautomatisierung-2293819.html>
- [4] <http://www.eebus.org/eebus-technik-konzept/>
- [5] <http://www.golem.de/news/agora-energy-home-und-eebus-einheitliche-sprache-fuer-europaeische-smart-homes-1411-110319.html>
- [6] http://www.deloitte.com/view/de_DE/de/branchen/technology-media-telecommunications/7697b5ff6df82410VgnVCM3000003456f70aRCRD

Die Autoren



Kai Kreuzer arbeitet als Developer Evangelist im Connected-Home-Bereich der Deutschen Telekom AG. Er ist Projekt-Lead des Eclipse-SmartHome-Projekts und gründete 2010 das Open-Source-Projekt openHAB, um damit alle Aspekte rund um sein Haus zu automatisieren.



Thomas Eichstädt-Engelen arbeitet als Principal Consultant bei innoQ Deutschland. Dort beschäftigt er sich in erster Linie mit der Entwicklung individueller Kundenprojekte im Smart-Home-Umfeld. Er ist Co-Projekt-Lead des Eclipse-SmartHome- und openHAB-Projekts.