



## Auf den Leib geschneidert

# Ein rollenbasiertes Domänenmodell für das Smarthome

Thomas Eichstädt-Engelen, Phillip Ghadir

*Es gibt in der Domäne Smarthome viele unterschiedliche Sichtweisen, proprietäre Produktfamilien, einige offene Standards und verschiedene Konsortien, die versuchen, langfristig tragfähige, standardisierte Kommunikationsstandards zu definieren. Wir skizzieren für das Umfeld des sogenannten Ambient Assisted Living eine mögliche Programmierschnittstelle für die relevanten Anwenderrollen. Anhand derer stellen wir verschiedene Security-Aspekte zur Diskussion, um dann ein Sicherheitskonzept zu entwerfen.*

## Altersgerechte Assistenzsysteme für ein selbstbestimmtes Leben

► Beim Ambient Assisted Living [AAL] soll das Smarthome den Menschen unaufdringlich dabei helfen, die Aufgaben des Alltags zu bewältigen. Was für gesunde Menschen einfach nur bequem ist, erhält Leuten mit körperlichen Einschränkungen – gleich ob krankheits- oder altersbedingt – ein gewisses Maß an Unabhängigkeit von anderen Menschen. Moderne technische Helfer können Pflegediensten oder Ärzten durch aufgezeichnete Protokolle und Messungen schnell ein Bild von der Situation liefern.

Als Bewohner eines Smarthomes möchte man aber sicherlich auch einen gewissen Grad an Privatsphäre. Privatsphäre geht einher mit Zugriffsbeschränkungen auf Funktionen und Informationen sowie der persönlichen Entscheidung, mit wem welche Informationen geteilt werden.

Je eingeschränkter das Leben wird, desto weniger selbstbestimmt ist man. Mit zunehmenden körperlichen Einschränkungen werden die Anforderungen an die Assistenzsysteme größer. Manche Aufgaben, die zuvor noch in eigener Verantwortung lagen, werden nach und nach an Dritte transferiert.

Aber obwohl solche Zukunftsvisionen viele negative Assoziationen wecken könnten, mag das Leben im Smarthome trotz körperlicher Einschränkungen dennoch einigermaßen selbstständig und wirtschaftlich möglich sein.

Einhergehend mit graduell ansteigenden körperlichen Einschränkungen steigt der Bedarf an Funktionalität im Smarthome. Während ein gesunder, sportlicher Mensch keinen Aufpasser auf den Herd oder die Wasserhähne benötigt, schalten heutzutage Pflegedienste vielfach Fernseher oder Herdplatten für ihre Schützlinge ab, die so etwas ab und an vergessen.

Ohne „Ambient Assisted Living“-Funktionalität erfordert so ein fürsorglicher Eingriff die körperliche Anwesenheit eines Pflegers. Niemand synchronisiert aber die Zeit zwischen den Einsätzen von Pflegern und den außerplanmäßigen Einsätzen der Herdplatten durch den in Pflege befindlichen Bewohner.

Ambient Assisted Living kann solche Eingriffe zum Teil aus der Ferne ermöglichen. Solche Funktionalität erfordert dann das Reagieren auf Informationen, die über verschiedene Kanäle im Haushalt gesammelt werden können. Die Integration von Informationen kann zum Teil analytisch über die Zeit hinweg oder auch über verschiedene Kanäle aggregiert erfolgen.

Im Laufe der Zeit wird das Smarthome also hardwareseitig erweitert. Wenn der Bedarf erst einmal akut ist, werden die neuen Geräte vermutlich ins Smarthome integrierbar sein. Welche (Quasi-)Standards es dafür gibt, besprechen wir gleich. In vielen Fällen werden aber Aufgaben hard- und softwaretechnisch integriert, die zuvor stets vom Menschen manuell ausgeführt wurden. Per Software Geräte zu integrieren, die dafür nicht gebaut wurden, ist eine der heutigen Herausforderungen vieler Modernisierer.

Auf welche Standards sollte man heute setzen, um auch langfristig Systeme nahtlos zu integrieren?

## Smarthome/IoT-Standards

Die Frage nach dem „richtigen“ Standard im Smarthome-Bereich zu beantworten, ist kein leichtes Unterfangen, denn schon nach oberflächlicher Recherche finden sich dabei über 20 Allianzen, die das Internet of Things (IoT) im Allgemeinen und zum Teil das Smarthome im Speziellen auf Applikationsebene zu standardisieren versuchen. Schön wäre es natürlich, wenn alle Geräte im Smarthome dem gleichen Applikationsprotokoll gehorchen würden und sich aus diesem Grund selbstständig finden, konfigurieren und übergreifend benutzen ließen. Der Erfolg eines solchen Protokolls ist dabei aber nicht nur von der Qualität der Spezifikation, sondern insbesondere auch von der Akzeptanz in der jeweiligen Community abhängig.

Gerade um dem Problem der Akzeptanz zu begegnen, schließen sich Unternehmen in Allianzen zusammen, von denen wir nun die wichtigsten kurz vorstellen wollen.

### AllSeen

AllSeen wurde 2013 von Qualcomm als Stiftung zur Verwaltung und Betreuung des Open-Source-Frameworks AllJoyn gegründet. AllJoyn ist ein Applikations-Framework, das Funktionen anbietet, mit denen sich Geräte über die gängigen Plattformen (Microsoft Windows, Linux, Android, iOS, OS X, OpenWRT) hinweg finden, kommunizieren und kollaborieren können.

Auch wenn die AllSeen-Community sehr aktiv zu sein scheint, ist die Menge der auf dem Markt befindlichen Geräte bisher noch eher überschaubar. Der auf der vergangenen IFA 2014 gezeigte Backofen der Firma AEG Electrolux [AEG] ist bisher eher die Ausnahme. Problematisch ist bei AllSeen außerdem, dass die verwendete ISC-Open-Source-Lizenz keine Patent-Klausel hat, was die Bereitschaft der Mitglieder zur Veröffentlichung ihrer Assets eher bremsen dürfte.

### OIC – Open Interconnect Consortium

OIC wurde Anfang 2014 von den Branchengrößen Atmel, Broadcom, Dell, Intel, Samsung und WindRiver als direkter „Mitbewerber“ zu AllSeen gegründet. Neben der Standardisierung des Applikationsprotokolls wird auch eine Referenzimplementierung unter dem Projektnamen IoTivity angeboten [IoTivity]. Im Gegensatz zu AllSeen veröffentlicht OIC ihren Quellcode unter der Apache-Lizenz.

### HomeKit

HomeKit wurde von Apple als eine der beiden großen Neuigkeiten in iOS8 vorgestellt. Es definiert Schnittstellen auf Applikationsebene, die beschreiben, wie Geräte gefunden und gesteuert werden können. Es werden Geräteprofile, wie Schalter, Thermostat, Licht und Türschloss, definiert. Die so kategorisierten Geräte können dann in eine (Haus-)Hierarchie eingebunden und kontrolliert werden.

Das iPhone avanciert damit zur universellen, sprachgesteuerten Fernbedienung. HomeKit funktioniert wohlgerneht nur in der „Apple-Welt“, da die Implementierung auf anderen Plattformen aus lizenzrechtlichen Gründen nicht erlaubt ist.

## Google Weave

Auf der vergangenen Entwicklerkonferenz Google IO wurde das HomeKit-Pendant Google Weave [Weave] vorgestellt. In Ergänzung zum bisherigen Engagement in der Thread-Initiative [Thread] soll mit Weave nun ein Konzept auf der gleichen Abstraktions-Ebene wie HomeKit angeboten werden. Die bisher sehr spärlichen Informationen deuten darauf hin, dass die Abstraktions-Ebene irgendwo zwischen AllSeen/OIC und HomeKit liegen dürfte. Die Veröffentlichung wird für Q4.15 bis Q1.16 erwartet.

## EEBus

Der EEBus e.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 (die EEBus-Datenspezifikation). Darüber hinaus ist eine Integrationsplattform zur Verbindung der im EEBus spezifizierten Datenmodelle über IP-Technologien beschrieben worden (SHIP) [EEBus].

Bei genauerer Betrachtung dieser und der knapp fünfzehn weiteren Allianzen liegt die Vermutung nahe, dass nicht die harten Spezifikationen zur Gründung einer „eigenen“ Allianz geführt haben, sondern eher weiche Faktoren wie Not-Invented-Here oder ähnliche Befindlichkeiten eine Rolle gespielt haben.

Erfreulicherweise gibt es unter den Allianzen inzwischen erste Zusammenschlüsse. Ende letzten Jahres kündigten AGORA (Frankreich), EEBus (Deutschland) und Energy@Home (Italien) ihre Zusammenarbeit auf europäischer Ebene an [EE-BusAgora]. Wenig später kündigte ebenfalls der EEBus eine Kooperation mit dem OIC an [EEBusOIC]. Am Ende wird der Markt entscheiden – es werden sich De-facto-Standards bilden, die langfristig zu einer Konsolidierung des Marktes führen werden.

## Domäne Smarthome

Der De-facto-Standard für die Integration beliebiger Geräte im Smarthome ist das Projekt Eclipse SmartHome [ESH], das aus dem in dieser Kolumne bereits vorgestellten Open-Source-Projekt openHAB (siehe [openHAB] und [GhaEi13]) hervorgegangen ist.

Dank seiner Plug-in-Architektur erlaubt Eclipse SmartHome (sowie auch openHAB), die Funktionalität beliebiger Geräte über verschiedenste Bindings einzubinden. Die Geräte werden an verschiedenen Stellen am und im Haus beziehungsweise der Wohnung installiert.

Im Kern eines Smarthomes aus der Perspektive von Eclipse SmartHome stehen die Funktionen von Dingen, die über Schalter oder Regeln gesteuert werden. Die zentrale Abstraktion dafür ist das Item, das mit Hilfe von Applikationen visualisiert oder

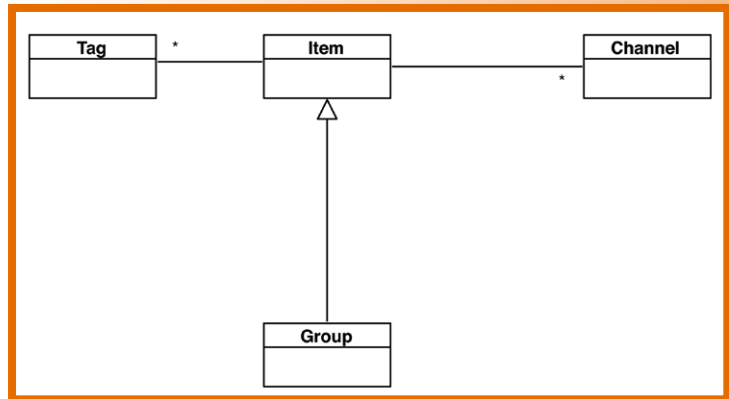


Abb. 1: Im Smarthome können Items gruppiert und getagged werden. Mit Items kann über die Channels mit dem Smarthome interagiert werden

gesteuert werden kann – sowie über beliebige Ereignisse aus der realen Welt – wie per physische Schalter oder Sonnenstand.

Abbildung 1 stellt den Kern des Modells dar. Es besteht aus sogenannten Items, die durch beliebige Tags geordnet sowie an einen oder mehrere Kanäle (Channel) gebunden werden können. Kanäle fallen nicht vom Himmel, sondern sind Bestandteile von installierten Geräten, den sogenannten Things, die in das Smarthome eingebunden werden.

Zu den Kanälen, die ein Thing mit sich bringen könnte, zählen sicherlich neben üblichen Sensoren – zum Beispiel zum Messen von Kohlendioxid, der Raumtemperatur, der Luftfeuchtigkeit, des Luftdrucks oder der Stellung des Fenstergriffs – auch Aktoren, die gemeinsam (manuell oder regelbasiert) gesteuert werden sollen.

Im Smarthome werden die über Channel integrierbaren Features von Geräten konfiguriert. Abbildung 2 skizziert das Integrationsmodell für die Smarthome-Plattform. Die in Abbildung 2 dargestellte Klasse Channel ist dieselbe wie die in Abbildung 1.

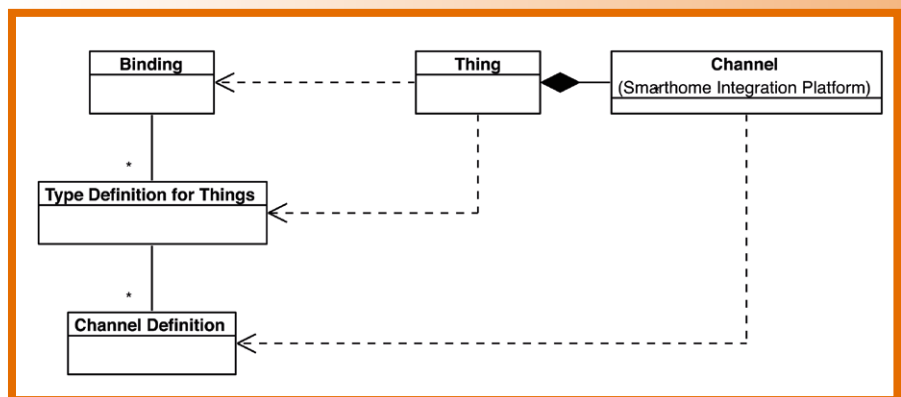


Abb. 2: Channels sind Bestandteile von Geräten (Things) und werden über Bindings technisch integriert

Im Smarthome kann ein Anwender mit einer einzelnen Aktion – zum Beispiel durch Betätigen eines Schalters – gleich mehrere Geräte steuern. Aus der Sicht des Smarthomes sind solche Benutzeraktionen gewöhnliche Ereignisse, auf die es potenziell beliebig komplexe Reaktionen geben kann.

Abbildung 3 zeigt das Metamodell, mit dem Eclipse SmartHome Regeln verwaltet, die das Smarthome „smart“ machen.

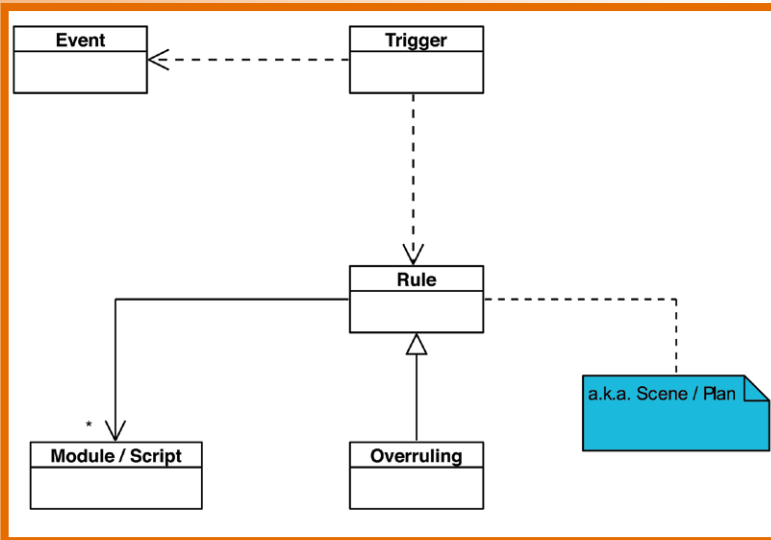


Abb. 3: Metamodell für Regeln zur kontextsensitiven Steuerung des Smarthome

### Kontextsensitives und situatives Steuern

Im Smarthome spielt die räumliche Lage bestimmter Geräte mit deren Kanälen eine wichtige Rolle für kontextabhängige automatisierte oder anwendergesteuerte Reaktionen.

Die räumliche Zuordnung findet in Eclipse SmartHome jedoch nicht auf Ebene der Geräte und Kanäle statt, sondern erfolgt durch die Zuordnung der verbundenen Items zu entsprechend benannten Gruppen. Spätestens durch die Zuordnung von Gruppen zu Gruppen können beliebig komplexe Strukturen entstehen, mit denen man das hierarchische Schema für das umgebende Gebäude aus Abbildung 4 darstellen kann.

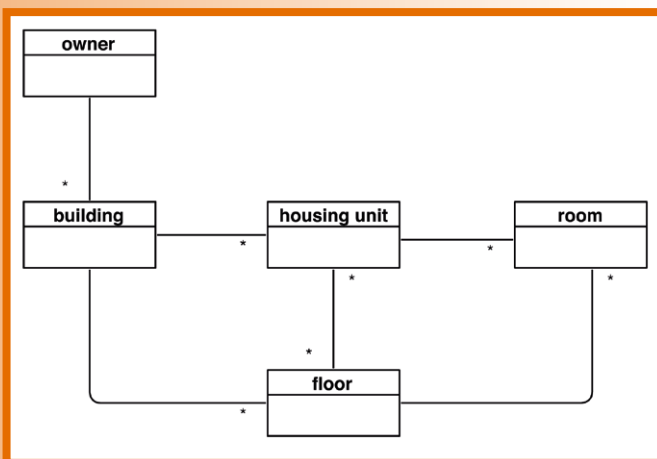


Abb. 4: Modell für hierarchische Gebäude-Schemata

Tags werden in Eclipse SmartHome verwendet, um Geräte (und damit Items) zu selektieren und dann entsprechend zu steuern. Solche Selektionen sind auch aus Regeln heraus möglich. Dadurch lassen sich dann komplexe Reaktionen auf Ereignisse realisieren, wie die Rollläden auf der Sonnenseite der Sonnenintensität anzupassen. Dies kann zum Beispiel als „Programm“ automatisch fahren oder von einem Bewohner angestoßen werden.

### Bedarf nach Rollen und Rechten

Kommen wir auf die Eingangs beschriebenen Situationen zurück, in denen Bewohnern aus der Ferne geholfen werden soll. Der Zugriff von außerhalb auf die heimische Smarthome-Installation lässt sich durch übliche Verschlüsselung der Übertragung und durch entsprechende Authentisierungsprüfung absichern. Praktisch allen Integrationsplattformen (Eclipse SmartHome eingeschlossen) fehlt heute ein darüber hinaus rollen- und rechtebasiertes Berechtigungssystem.

Wir möchten in diesem Artikel zunächst die typischerweise vorhandenen Stakeholder-Gruppen skizzieren, die ein rollenbasiertes Berechtigungssystem (mindestens) unterscheiden können sollte:

- ▼ Der *Netzbetreiber* will Smartgrid-relevante Informationen, wie Last- und Steuerprofile der Geräte im Smarthome, erhalten und anders herum seine Tarifinformationen und Steuerbefehle übermitteln, um eine bessere Netzauslastung zu erreichen.
- ▼ Der *Hersteller* möchte Informationen darüber bekommen, wie seine Geräte *tatsächlich* genutzt werden, um damit seine Produkte verbessern zu können.

- ▼ Der *Plattformbetreiber* stellt den fehlerfreien, technischen Betrieb der zentralen Plattform sowie des angeschlossenen Smarthome-Systems sicher. Dafür benötigt er in Fehlersituationen Zugriff auf den aktuellen Zustand der Installation, die Logfiles und kann bestimmte Systeminformationen, wie die Ressourcenauslastung, abfragen.
- ▼ Der *Bewohner* ist der Endanwender des Systems. Er hat ein großes Interesse, seine Privatsphäre zu schützen. Es kann unterschiedliche Bewohner mit verschiedenen Berechtigungsstufen (Großeltern, Eltern, Kind) geben. Er nimmt selbst nicht die Inbetriebnahme und Konfiguration der angeschlossenen Geräte vor, sondern überlässt das dem *Smarthome Designer*.
- ▼ Der *Smarthome Designer* integriert und konfiguriert neue Geräte im Smarthome.
- ▼ Der *Familienmanager* legt neue Bewohner an und ordnet ihnen Berechtigungsprofile (Rollen und Rechte) zu. Er kann auch weitere Rollen ergänzen.
- ▼ Der *Stellvertreter* übernimmt für eine bestimmte Zeit die Rolle des *Bewohners* (ähnlich dem *Dienstleister*).
- ▼ Der *Dienstleister* kann einmalig (Paketbote), gelegentlich (Gärtner) oder regelmäßig (Haushaltshilfe) Aufgaben im und ums Smarthome übernehmen.
- ▼ Der *Gast* hat temporären, sehr eingeschränkten Zugriff auf den für ihn relevanten Bereich (Gästezimmer, Bad, Hauszutritt) im Haus.

Als Nächstes bilden wir diese Rollen auf die zentralen Eclipse SmartHome-Konzepte ab. Ein naheliegender Ansatz zur Abbildung auf die Items ist die Nutzung der in Abbildung 1 gezeigten Tags. Da Tags einfache Textelemente und damit weitgehend frei wählbar sind, könnten sie auch Rollen- oder Rechte-Informationen für Items beinhalten und erscheinen damit zunächst als geeigneter Container.

Würde man diesem Ansatz folgen, könnte allerdings eine feingranulare Konfiguration der Rollen für einzelne Operationen eines Items schnell zu einer Explosion der Tags je Item führen. Viel gewichtiger erscheint aber, dass nur Items mit Tags attribuiert werden können (und sollen) und somit nicht alle abzusichernden Konzepte über diesen Mechanismus erreicht werden können.

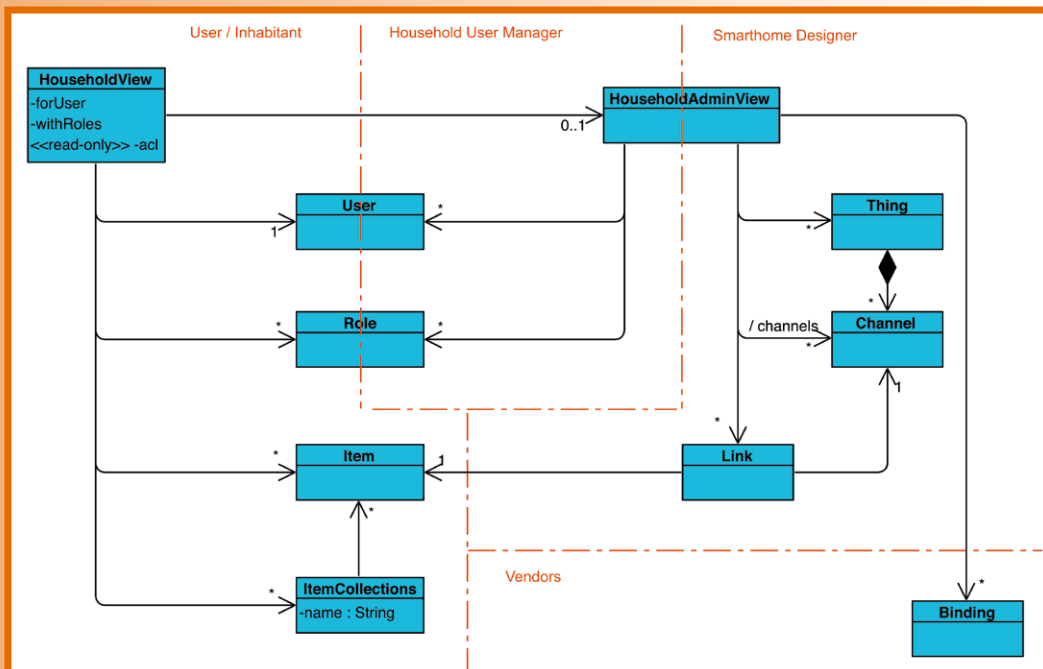


Abb. 5: API-Entwurf für das Smarthome

Ebenfalls zu kurz würde vermutlich der Ansatz springen, Mappings von Rollen und Routen zu administrieren. Die je nach Größe der Installation schnell ansteigende Zahl von Items, Things und Channels und den daraus folgenden Routen dürfte bald zu unwartbaren Mappings führen.

Ein API, das den verschiedenen Rollen gerecht wird, könnte hingegen ähnlich wie in Abbildung 5 dargestellt aussehen.

Wesentlich ist hierbei die Erweiterung des Domainmodells um die Elemente *User* und *Role*. Ebenfalls neu ist die Gliederung des bisherigen API in die *HouseholdView* und die *HouseholdAdminView*, die den Zugriff auf die unterliegenden Ressourcen mit Hilfe der vorliegenden Informationen über *User* (forUser) und *Role* (withRoles) steuern können.

## Literatur und Links

[AAL] [https://de.wikipedia.org/wiki/Ambient\\_Assisted\\_Living](https://de.wikipedia.org/wiki/Ambient_Assisted_Living) (Stand 17.08.2015)

[AEG] <http://newsroom.electrolux.com/de/2014/09/01/electrolux-tritt-allseen-alliance-bei-um-nahtlose-vernetzung-von-geraten-zu-ermöglichen/>

[EEBus] <http://www.eebus.org>

[EEBusAgora] <http://www.eebus.org/aktuelles/news-single/datum///entscheidende-entwicklung-fuer-smart-home-fuehrende-europaeische-initiativen-vereinbaren-einheitlich/>

[EEBusOIC] <http://www.eebus.org/aktuelles/news-single/datum///eebus-initiative-ev-und-open-interconnect-consortium-oic-starten-strategische-zusammenarbeit-zur/>

[ESH] <http://www.eclipse.org/smarthome>

[GhaEi13] Ph. Ghadir, Th. Eichstädt-Engelen, openHAB auf dem Raspberry Pi, in: *JavaSPEKTRUM*, 06/2013, s.

[http://www.sigs-datacom.de/fileadmin/user\\_upload/zeitschriften/js/2013/06/ghadir\\_JS\\_06\\_13.pdf](http://www.sigs-datacom.de/fileadmin/user_upload/zeitschriften/js/2013/06/ghadir_JS_06_13.pdf)

[IoTivity] <https://www.iotivity.org/>

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

[Thread] <http://threadgroup.org>

[Weave] <http://m.heise.de/newsticker/meldung/Google-Brillo-und-Weave-Ein-Android-fuers-Internet-der-Dinge-2670592.html>



**Thomas Eichstädt-Engelen** arbeitet als Principal Consultant bei innoQ Deutschland. Dort beschäftigt er sich in erster Linie mit der Entwicklung individueller Kundenprojekte im IoT und Smarthome-Umfeld. Er ist Co-Project-Lead des Eclipse SmartHome-Projektes und der Integrationsplattform openHAB.  
E-Mail: [Thomas.Eichstaedt-Engelen@innoq.com](mailto:Thomas.Eichstaedt-Engelen@innoq.com)

**Phillip Ghadir** baut am liebsten tragfähige, langlebige Softwaresysteme. Er ist Mitglied der Geschäftsleitung bei innoQ und hat sich früh auf Architekturen für verteilte, unternehmenskritische Systeme spezialisiert. Darüber hinaus ist er Mitbegründer und aktives Mitglied des ISAQB, des International Software Architecture Qualification Board.  
E-Mail: [phillip.ghadir@innoq.com](mailto:phillip.ghadir@innoq.com)

## Von DIY zu LOTCOI

Im Umfeld von Internet of Things, Smarthome und Ambient Assisted Living gibt es verschiedene Spieler und Allianzen. Wir bauen hier auf eine Konsolidierung über die Zeit und hoffen auf eine allianzübergreifende Interoperabilität.

Eclipse SmartHome erlaubt als Java-basierter De-facto-Standard für die Heimautomatisierung die nachträgliche Integration von unterschiedlichsten Elementen in die Steuerung des Smarthomes. Die Mächtigkeit und Flexibilität entsteht durch die Entkopplung von zu steuernden Elementen (Items) von den tatsächlichen Geräten (Things und Channels), die diese Elemente ins Smarthome einbringen, und den Protokollen, über die diese angebunden werden (Bindings).

Aktuell unterscheidet Eclipse SmartHome noch nicht verschiedene Anwendergruppen. Man spürt noch die Wurzel des openHAB-Projekts, das den Fokus auf den Do-It-Yourself (DIY)-Heimautomatisierungs-Bastler legte, der die technische Umsetzung im Smarthome selbst prägt.

Mit zunehmender Reife sind Rollenmodelle gefragt, in die einzelne Berechtigungen an neue Anwender vergeben können – zum Beispiel an das Pflegepersonal oder den Leitstand eines Smarthome-Plattformbetreibers. In diesem Artikel haben wir dafür relevante Anwendergruppen zusammengetragen.