



Smarter Home middleware between IBM IoT Foundation and Apple HomeKit

Bachelor Thesis

at Course of Studies Applied Computer Science
at the Cooperative State University Baden Württemberg in Stuttgart

by

Andreas Rau

August 31 2015

Supervisor 1
Supervisor 2

Jochen Burkhardt
Hans-Erich Lorke

Andreas Rau
Student ID, Course
Company

4186494, TINF12A
IBM, Böblingen

Abstract

Background: The term Internet of Things is getting more popular these days and one application is Smart Home. Many vendors are currently fighting for the pole position but no standard is in sight. With the availability of multiple technologies cross compatibility is suffering. In the course of this thesis a middleware between IBM SDPvNext and Apples HomeKit is developed to enable cross compatibility of two big vendors.

Concept: Terminology like home and smart home is discussed. Further smart home functions like indoor environment, living quality, security and more which emerged from scanning common iot and smart home definitions are mapped to basic home functions like economic, social and protective. These functions are aggregated to measurements and extended by valuing Loius Sul-livans idea of *form follows function* and the within lying idea of minimalism emerging in the final measurements **security, smart home functions, cross compatibility and costs**. With the help of these measurements current smart home technologies like Apple HomeKit, Google Nest, Samsung Smart Home and SmartThings are compared against each other. HomeKit as overall winner of the evaluation is considered to be connected to SDPvNext by a middleware. Two possible middleware solutions are outlined and discussed for pros and cons.

keywords: Smart environment, smart home, Apple HomeKit, Google Nest, Samsung Smart Home, SmartThings, middleware, measurements, evaluation, home gateway, iOS App

Hintergrund Das Internet der Dinge ist ein Begriff welcher schon seit 20 Jahren existiert aber erst seit Kurzem in der breiten Masse angekommen ist. Eine Anwendung der IoT ist ein smartes Zuhause oder um den offiziellen Duktus zu verwenden *Smart Home*. In 2015 gibt es viele Vertreter einer Smart Home Lösung jedoch ist kein Standard zu erkennen. Trotz der vielen Technologien welche für die Realisierung einer Smart Home Lösung verwendet werden gibt es nur wenige Systeme welche mit denen von anderen Herstellern kompatibel sind. Im Rahmen dieser Bachelorarbeit wird eine sogenannte Middleware, eine Stück Software welches mehrere zuerst inkompatible Systeme miteinander verbindet, entwickelt.

Konzept Zu aller erst werden Begrifflichkeiten wie Heim und Smart Home geklärt. Des weiteren werden Smart Home Aufgaben wie Wohnraumklima, Lebensqualität, Sicherheit und Andere durch Evaluation von Smart Home und IoT Definitionen aufgestellt und später auf grundlegende Wohnraum Zwecke abgebildet. Die Ergebnisse werden zu Vorraussetzungen für eine erfolgreiche und gute Smart Home Lösung aggregiert und mit einfachen Grundsätzen wie "*form follows function*" (*Sullivan*) und der inne liegenden Wertschätzung des Minimalismusses erweitert. Sicherheit, Smart home Aufgaben, Interkompatibilität und Kosten sind letztlich die Spezifikationen nach denen aktuelle Systeme wie Apple HomeKit, Samsung Smart Home, SmartThings und Googles Nest bewertet werden. Als Gewinner stellt sich das Apple HomeKit heraus und wird weiterführend eine Rolle in der middleware Lösung spielen, den zweiten Part übernimmt IBM's SDPvNext. Als middleware Lösung werden zwei Systeme vorgestellt von welchen eine implementiert wird. Die Auswahl des Systems ist ebenfalls dokumentiert.

Author's declaration

Unless otherwise indicated in the text or references, this paper is entirely the product of my own scholarly work. This paper has not been submitted either in whole or part, for a degree at this or any other university or institution. This is to certify that the printed version is equivalent to the submitted electronic one.

Gemäß § 5 (2) der „Studien- und Prüfungsordnung DHBW Technik“ vom 18. Mai 2009. Ich habe die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Stuttgart, August 31 2015

Andreas Rau

Contents

1	Introduction	1
1.1	Topic and Motivation	1
1.2	Limitations	2
1.3	Goals	3
1.4	Tasks	3
2	Theory	4
2.1	Smart Home	4
2.1.1	What is a home?	4
2.1.2	Smart home use cases	6
2.1.3	Definitions: Smart home - iot	7
2.1.4	<i>Feature-ism</i>	8
2.1.5	Mapping of smart home and basic home functions	9
2.1.6	Final measurements	9
2.2	Current Systems	11
2.2.1	Apple HomeKit	11
2.2.2	Samsung Smart Home	12
2.2.3	SmartThings	13
2.2.4	Google Smart Home	14
2.3	Battle for Standards	15
2.3.1	Home gateway solution	15
2.3.2	Radios	15
2.3.3	Cross compatibility	15
2.3.4	Fog underneath the cloud	15
2.3.5	Market dominance	16
2.3.6	User VS company	16
2.4	Technologies	17
2.4.1	HomeKit enabled	17
2.4.2	Non-HomeKit	17
2.4.3	Radios	18
2.5	HomeKit and SDPvNext	19
2.5.1	Apple HomeKit	19
2.5.2	WWDC15	22
2.5.3	HAP - HomeKit Accessory Protocol	25
2.5.4	Siri	26
2.5.5	iDevices	26
2.5.6	Bluemix	26
2.5.7	IoT Foundation	27
2.5.8	SDPvNext Rest API	29
2.6	Constraints	30
2.7	Development Tools	30
2.7.1	Mac OSX	30

2.8	Middleware	31
3	Concept of Smart Home Solution	32
3.1	Applying measurements	32
3.1.1	Some words on trustworthiness	32
3.1.2	Security	33
3.1.3	Smart home functions	36
3.1.4	Cross compatibility	37
3.1.5	Cost	39
3.1.6	Extra	40
3.1.7	Comparison	41
3.1.8	The winner	41
3.2	Concept	42
3.2.1	Possible solutions	43
3.2.2	Advantages and disadvantages	45
3.3	Solution	46
3.3.1	Design	46
3.3.2	Implementation	47
3.3.3	Additional work	55
4	Summary	56
4.1	Reflection	56
4.2	Result	57
	List of figures	IV
	List of tables	V
	Source code listings	VI
	Bibliography	VII
	Appendix	IX

1 Introduction

1.1 Topic and Motivation

The term *iot* is around for more than two decades. The concept of a smart device being connected to the internet was realized by four students of the Carnegie Mellon University by modifying a coke vending machine to be able to report its inventory and temperature in 1982 (Martellaro, 1990). Several companies have started their *iot* approaches among them Microsoft's "At work" and Novell's "Nest". The concept of *iot* became popular in 2002 when Paolo Magrassi introduced MIT's RFID and internet of things technology to the industrial and business world (Magrassi, 2001). The idea of identifying and connecting every device possible to control and manage them over a centralized computer has driven *iot* to today's understanding of smart devices (Magrassi, 2002). Smart Home is one application of *iot*. Being able to monitor and control the mechanical, electrical and electronic systems independent of your location offers new opportunities in terms of security and economization. As of 2014 smart home arrived at the average user by using the possibilities of smart phones to control your smart devices at home (McLaughlin, 2014). A lot of movement in the IoT and Smart Home sector is going on right now. Smart Home is a great opportunity for startups to develop innovating products that "push the boundaries of smart" ("In The U.S. Smart Home Market, Don't Rule Out China," 2015). Established companies like Samsung and Google seek their dominance by acquiring these startups and integrate them into their own product line ("In The U.S. Smart Home Market, Don't Rule Out China," 2015).

presentation of the problem Current systems may have problems with connectivity or setup time and frequency ("Why Is My Smart Home So Fucking Dumb?" 2015). However another big issue is cross compatibility when it comes to connect **all** of your home. It seems that hardly any company or vendor makes the approach to be able to connect or control devices of the competitors in a

suitable way. This is the part where this thesis turns up the heat. The content of this thesis deals with smart home solutions and their cross compatibility. First of all basic definitions are clarified then theory is build up to emerge in a middleware solution which connects multiple vendors smart home solutions.

1.2 Limitations

In the course of this bachelor thesis the middleware solution is limited to the guidelines of given systems. In other words the systems that have to be connected by a middleware are economically chosen by IBM. On one hand there is IBM SDPvNext which suits more or less a user management role for devices that are registered in the IoT Foundation which stores devices and their configurations in a database (Mielke, 2015). At the time this paper is published SDPvNext will be integrated in the IoT Foundation Platform and be extended by an authentication layer (PatriziaGufler, 2015) but the two products are treated separately.

On the other hand there is Apple HomeKit, a relatively new approach on smart home. Where Apple is relying on local databases for device information storage as well as waterproof communication encryption with multiple security layers McLaughlin (2014) evaluated amongst other systems on the market in clause *Measurements*. IBM SDPvNext is a more basic approach Mielke (2015). This thesis will not cover explicit code explanations on the given solution nor specific programming language explanations, more describing the idea and the technique used to solve the given problem. However the code can be easily reproduced with the given explanations of what is done and some basic programming skills.

Further the psychological significance of a home is not considered in this paper. Nevertheless the state as well as changes of a home has a strong influence

on behavior, emotions and overall mental health *Raising young refugees' voices in Europe and beyond* (2008).

1.3 Goals

The goal of this thesis is to define measurements for a suitable smart home environment in the eyes of the user. Cross compatibility of vendors and devices as well as minimalism in the amount of extra hardware is valued. Further current systems are rated and two concepts of a middleware connecting multiple natively non connectable systems are developed and rated for further realization. The better solution is implemented and also rated.

1.4 Tasks

The following tasks are written down in an artistic way which suits the creator of TEX Donald E. Knuth and its favor of describing programming as an art-form (Knuth, 1998). At first terminology is declared to provide a consistent base for further argumentation. By specifying Smart Home measurements the solution gets its outlines. Thirdly a weighted overview of the functionality of every involved system and technology is given to provide the colors that will fill in the outlines. Last a solution is drawn with the technologies discussed earlier.

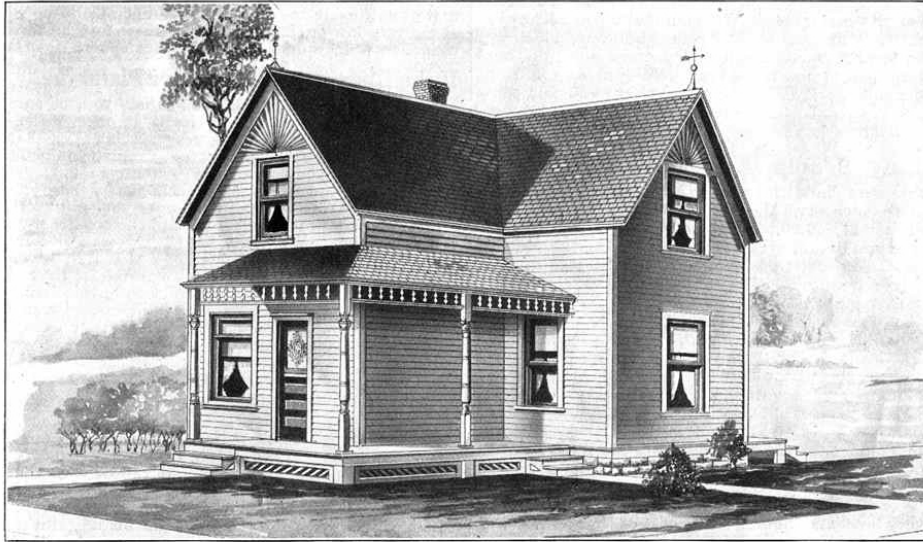
2 Theory

2.1 Smart Home

There is a necessity to provide a sweeping substantiated knowledge-base in order to develop smart home middleware. Hence basic terms like *home* have to be defined which later build up the basis for justification on decision taking in the development process.

2.1.1 What is a home?

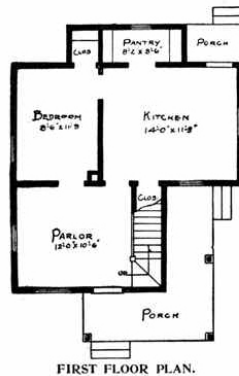
The general definition of a home *the place where a person (or family) lives* (Webster's New World College Dictionary, *n.d.*) is not sufficient enough for the domain of smarter home. Due to this fact information on the term *home* is collected and used to create a suitable definition. The famous citation of Louis Sullivan *form follows function* further used as *Sullivan's rule*, originally altered from papers of nature observation where Horatio Greenough came to the conclusion that form only changes if function changes (Sullivan, 1930) also applies to buildings such as homes. The following figure shows a plan of a house where every room has its specific functions assigned to it:



MODERN HOME No. 115

With Wood Foundation, Not Excavated.

On the opposite page we illustrate a few of the materials we specify on this, our \$725.00 house.



The arrangement of this house is as follows:

FIRST FLOOR.

Parlor - - - 12 feet by 10 feet 6 inches
 Bedroom - 8 feet 6 inches by 11 feet 9 inches
 Kitchen - - - 14 feet by 11 feet 9 inches
 Pantry - - - 8 feet 2 inches by 3 feet 6 inches

SECOND FLOOR.

Front Bedroom, 8 feet 3 inches by 10 feet 6 inches
 Rear Bedroom, 8 feet 6 inches by 11 feet 9 inches
 Large Attic - - - 14 feet by 11 feet 9 inches
 All bedrooms have roomy closets.

Size: Width, 24 feet; length, 28 feet, exclusive of porch



Figure 1: Social functions for rooms

Since the function of a home is in general a place to live and the question of life is a little to complicated to clarify in this paper, more general abstractions are defined to meet the daily approaches on living. The National Institute of Schooling started an approach on naming the functions of a home (**homeFunctionality**):

- Protective
- Economic
- Religious
- Educative
- Social
- Affectional
- Status-giving

These functions are the root for further more complex and current *daily needs* which are not discussed further due too unnecessary specific information. Religious, Educative, Affectional and Status-Giving functions are not considered as necessary for a good smart home implementation and therefore not applied later. Further a home is a geographical location like a city or suburb and used as a permanent or semi-permanent residence, whereas transitory accommodations (hospital, prison, college, etc) are not considered as home but in the scope of smart home.

2.1.2 Smart home use cases

Smart Home use cases are pretty much bound to the services that are run. So in the following table some services are provided as use cases:

Table 1: Homekit Accessory Profiles

Services	Characteristics
Garage door openers	Lock state
Lights	Brightness
Door Locks	Lock state
Thermostats	Current temperature
IP camera controls	Model number
Switches	Switch state
Custom	Custom

These services represent actions that can be performed with the help of smart home devices. A more general way of describing use cases for smart home is done by naming the functions it has to serve in order to accomplish the given actions.

2.1.3 Definitions: Smart home - iot

To get an inside view smart home and suitable iot definitions are shown below:

- **Smart home** “Smart Home is the topic for technical procedures inside of houseings that are supposed to improve indoor environment and living quality, security and efficient energy using remotely controlled devices.”
- **IoT** “the Internet of Things (IoT) refers to identifiable objects and their virtual representation in an Internet-like structure.” (Ashton, 1999)
- **IoT** “moving small packets of data to a large set of nodes, so as to integrate and automate everything from home appliances to entire factories.” (Raji, n.d.)
- **IoT** “the term commonly is used to signify advanced connectivity devices, systems and services that go beyond machine-to-machine communications,

and covers a variety of protocols, domains and applications.” (“Internet of Things - Techcrunch,” 2015)

Smart home, general things to keep in mind The following list is an aggregation of the previous definitions:

- improvement of indoor environment and living quality
- improvement of security
- improvement of energy efficiency
- devices are remotely controlled
- devices are identifiable objects and have a virtual representation
- actions are automated

2.1.4 *Feature-ism*

Computing power hasn’t increased much since 2010, a more stagnating trend is visible. Companies like Intel are improving power consumption instead of computing power because there is a physical end, namely the size of transistors can’t get any smaller. The idea is to scale in width rather than height, which means more cores instead of more power (Chacos, 2013). So we have come to the point that the everyday user is used to the fact that computing power won’t increase drastically anymore (Chacos, 2013). To keep the users buying and attracted to new hardware tricks have to be played. Samsung and many others know the path of feature-ism to keep the users attracted to their devices. Every year all new features that "revolutionize" the user experience are thrown into the market and cause technological entanglement (Machine, 2012). New feature are always "cool" to show them to your friends, but longterm thoughts suffer. Users are "often enticed by the lure of interesting and exotic technologies that look like fun, but in the end, they don’t serve us very well for what we want

to achieve." (Machine, 2012) John Martellario, a writer for the MacObserver mag, states that: "there is only a proper, considered subset of all the available technologies out there that are required to get any specific job done." (Machine, 2012) With that idea in mind and combined with the Sullivans rule minimalism in technology is what makes it useful and good. Later on measurements are also taken on future proof ness because its aim able and necessary to create a worthwhile user experience (Machine, 2012).

2.1.5 Mapping of smart home and basic home functions

There is little information about valuation methods for IT systems in the internet that cover topics like usability for the user and extensibility as well as futer proof ness, instead dozens of financial valuing methods. Due to this fact the definitions of smart home are aggregated and mapped to the basic functions of a home in order to provide use cases for a proper valuation.

Table 2: Mapping of smart home services to basic home functions

Smart home	home functions
Indoor environments	Economic and Social
Living quality	Social
Security	Protective
Energy efficiency	Economic
Remotely controlled	Protective and Economic
Identifyable devices	Protective and Economic
Automation	Protective and Economic

2.1.6 Final measurements

These smart home functions have to be mappable to basic home functions. Due to the fact that device functions do not change when replaced by smart home devices, the amount of set up and carry on work should be as low as possible: *minimalism*. Moreover *Where function does not change, form does*

not change (**SullivansRule**), implies that the absence of minimalism namely extra hardware is negatively valued. A switch is still a switch and shouldn't need to be connected to extra hardware in order to work. smart home devices are costly and therefore should be future proof and not fall into the garbage whenever a new version hits the market.

For economic reasons costs of required hardware, installation and maintenance have to be considered. Due to the fact that IBM doesn't provide homegateway hardware, costs have to be roughly evaluated from current systems on the market not considering the purchase price.

The following list shows all measurements that are considered in this thesis:

- security
- smart home functions
 - 1. feature-ism
- cross compatibility
 - 1. sullivan's rule / minimalism
 - 2. futureproofness
- costs
- other

Further the evaluation is done on values ranging between +2 for excellent fulfillment all to -1 for not fulfilling the requirements.



Figure 2: Evaluation Overview

2.2 Current Systems

In the scope of this thesis more familiar and emerging smart home solutions in the sheer flood of vendors are discussed. Therefore the selection of vendors for a basic analysis are the following:

2.2.1 Apple HomeKit

Overview Apple builds up its functionality on two radio systems: WiFi and Bluetooth Low Energy. Where all HomeKit enabled devices connect directly to an iDevice there is also a possibility of connecting a HomeBridge which serves more radio and cable systems in order to connect more decent devices. Since version 2.0 Homekit comes with the possibility of using the Apple TV as a tunnel to the internet. All iDevices that are registered to the same Apple ID can communicate to each other for a home automation interaction. Further approaches do not rely any more on the Apple TV. Third party HomeBridges are enabled to take communication over the internet. The iCloud itself will serve as a switchboard where iDevices can communicate to all HomeKit enabled ones. Currently there are no news on a car connection or further approaches into smart cars.

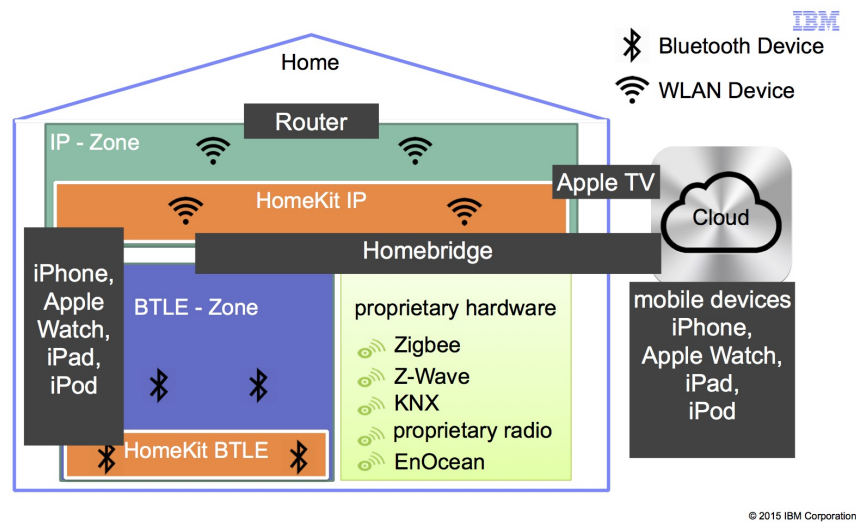


Figure 3: Overview of Apple HomeKit functionality

2.2.2 Samsung Smart Home

Overview Samsung Smart Home solution provides connectivity between all connected devices at home with a single app that can be accessed from smart phones, tablets, tvs and watches. Further information about Samsungs Smart Home solution is rare but three key features are explained below:

Device Control Device Control allows the user to interact with several home devices. Moreover voice control is implemented and enables the possibility to use special phrases like "I am leaving" to trigger predefined action sets like turning of the lights, the air condition and start the cleaning robots (Pochanke, 2014).

Smart Customer Service The Smart Customer Service notifies the user whenever a smart device needs to be cleaned or repaired (Pochanke, 2014).

Home View Home View is considered as a sort of home surveillance system featuring connected cameras to monitor everything happening at home (Pochanke, 2014).

It should be mentioned that at the current state only devices out of Samsungs own product line are supported for smart home purposes. Nevertheless Samsung claims third party support with the aid of its Smart Home Protocol (Pochanke, 2014).

2.2.3 SmartThings

Overview SmartThings started as a Kickstarter project and was acquired by Samsung in 2014. The idea behind it is a combination of a hub and an app. The hub provides the interconnect ability to other devices and aggregates communication for the app. In other words the app can talk to devices from multiple vendors. This is made possible by supporting additional radios compared to HomeKit and Samsung Smart Home namely Zigbee and Z-wave. A big downside is the absence of voice control. Only with more or less elaborate workarounds it is possible to control your devices with your voice (“SmartThings: Security Issue,” 2012).

2.2.4 Google Smart Home

Overview In 2014 Google acquired the Smart Home specialist Nest with its productline consisting of a thermostat (Nest Thermostat) and a smoke sensor (Nest Protect) (“Nest: Erstmals neue Smart-Home-Geräte seit Übernahme durch Google,” 2015). After nearly one and a half year of silence in Googels smart home sector Nest revealed the Nest Cam, a 1080p smart home connected camera. For additional \$ 10 per month the user acquires the ability to review video material of the last 10 days as well as getting notifications from nest whenever suspicious activity is detected.

Brillo "Brillo extends the Android platform to all your connected devices, so they are easy to set up and work seamlessly with each other and your smart-phone" (“Google Developers - Brillo,” 2015). With Brillo comes Weave which serves a consistent api for device communication. An instant communicaiton with Android is guaranteed by the developers at Google (“Google Brillo und Weave: Ein Android fürs Internet der Dinge,” 2015). The bedrock of Brillo is a downscaled version of Android supposed to run on systems with low hardware specks like smart home devices (“Google Brillo und Weave: Ein Android fürs Internet der Dinge,” 2015).

2.3 Battle for Standards

2.3.1 Home gateway solution

Insteon, Lutron and many others competing in the HomeKit market aren't developing dedicated devices to connect to Apples Smart Home solution. The common answer is a bridge which serves as an allocator for connecting existing smart home devices. Where Elgato is the pioneer with it's EVE series on dedicated devices.

2.3.2 Radios

Where the market in the industry is ruled by radio systems like Zigbee and Z-wave, the smart home segment is more common radio systems friendly by using widely spread technology like WiFi and Bluetooth Low Energy.

2.3.3 Cross compatibility

It doesn't matter how good every single solution on smart home is on it self. For many users the usage and the installation of multiple vendors solutions is difficult. So cross compatibility is desirable. Approaches on this topic are poorly solved with bridges that are able to connect more devices to an existing platform than it actually supports.

2.3.4 Fog underneath the cloud

These bridges have to have more logic components than end devices in order to pre filter information and handle communication over different radios. An in official term for that kind of technology is the fog. The fog itself stands for logical pre computation before pushing information to the cloud.

2.3.5 Market dominance

The market dominance is at the moment not important because there is no standard in smart home communication. When thinking back to the early ages of the internet a top dog like http and ip where only possible by defining standards. And rash decisions on the market leader are not appropriate in terms of iot.

2.3.6 User VS company

It remains the question if the user drives the development of new smart home technology or the companies do.

2.4 Technologies

2.4.1 HomeKit enabled

Lutron Electronics , a company born when electricity was rare and lightbulbs were generating that much heat that they were rather used to boil some eggs than to read a book at night. It's founder Joel Spira was a pioneer in dimmable lights, a technology former only known in theatre. Spira liked the idea of saving energy and money by dimming lights. 1959 he filled out his first patent of a dimmable switch which would fit into a wall plug, revolutionary at that time. Nowadays Lutron holds over 2,700 worldwide patents, one of them is a window shading technology ("Lutron History," 2015). In 2015 Lutron steps into the HomeKit smart home area by revealing a new bridge which can be controlled over an iOS app as well as Siri and Apple Watch. All existing caseta and caseta wireless devices can be connected to the bridge.

Insteon Technologies Insteon Technologies is a home automation company which was founded by Joe Dada in 1992 and has coined the term smart home. In January 2015 Insteon announced a gateway (bridge) to allow an interconnection with the Apple HomeKit system.

Elgato Eve Elgato started with Eve it's HomeKit compatibility, where all devices are equipped with Bluetooth Low Energy to connect directly to an iDevice like the iPhone.

2.4.2 Non-HomeKit

Raspberry Pi There are multiple HomeKit accessory bridge implementations for the Raspberry Pi including HAP-Java.

Belkin WeMo Belkin WeMo is a series of lights, switches and sensors that come with a dedicated app. Control takes place over WiFi where nearly every

device is able to connect itself to the local network.

2.4.3 Radios

Belkin is more the home approach for an automation where Zigbee and Z-wave are more top dogs in the industry.

Zigbee Zigbee is a specification for low powered radio devices which are used for small data rates, perfect for home automation.

Z-wave Z-wave itself is pretty close to Zigbee.

2.5 HomeKit and SDPvNext

2.5.1 Apple HomeKit

Apple HomeKit is an API developed by Apple. A common database stores all home information to provide consistency and is available to all Apps. Users can use Siri to interact with their home accessories. Homekit also provides remote access and uses end to end encryption between iDevices in order to maintain user privacy and security (McLaughlin, 2014).

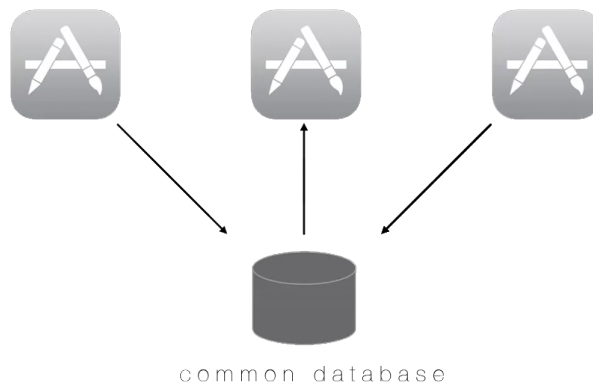


Figure 4: Common database provided by HomeKit (McLaughlin, 2014)

The Home Manager is the entry point for all apps. With the help of the Home Manager changes to the database are made. Homes can be added or removed and will notify the user in case of changes (McLaughlin, 2014).

The organigram used by Homekit is build up by **Homes**. Homes create their own namespace and have to be named uniquely, contain Rooms and Accessories. Rooms have to be named uniquely in their Home domain. (McLaughlin, 2014)

An Accessory corresponds to a physical device and is assigned to a Room. Accessory Objects allow to access the device state (McLaughlin, 2014).

Services represent functionality of an Accessory and are like parameters you can interact with. Services may have a name because some Services are not ment to interact with a user(update firmware). Services are a collection of **Characteristics** like range or units, e.g. the power state of a lightbulb Service. (McLaughlin, 2014).

Characteristics can be of a fiew variety:

- read-only - e.g. the current temperature
- read-write - e.g. lightbulb power state
- write-only - e.g. identify accessorie

Homes, Rooms, Accessories, Services and Characteristics are recognized by Siri and can therefore be accessed without a direct interaction with an app, making the use of HomeKit very comfortable. Services have Apple defined types, that can be accessed by natural language like synonyms or expression that do not exactly refer to the correct name provided to the Service making interaction experience more natural like. (McLaughlin, 2014).

The Accessory Browser is used to find any new Accessories available and is used to add them to a Home. When it is added to a Room it's time to name the Accessory and assign it to a Room (McLaughlin, 2014).

Initial setup workflow:

- Create a Home - User provides name
- Add Rooms to the Home - User provides names
- Add Accessories - Use an Accessory Browser - add Accessory to Home - User provides name - User chooses Room

- interact through application and/or Siri

A natural way of referring to Accessories in a Home is done by grouping. Apples HomeKit enables the user to group Accessories in several ways, name them and provide access to Siri (McLaughlin, 2014).:

Zones are arbitrary, uniquely named groups of Rooms e.g. *upstairs*, where Rooms can be in any number of Zones and are recognized by Siri (McLaughlin, 2014).

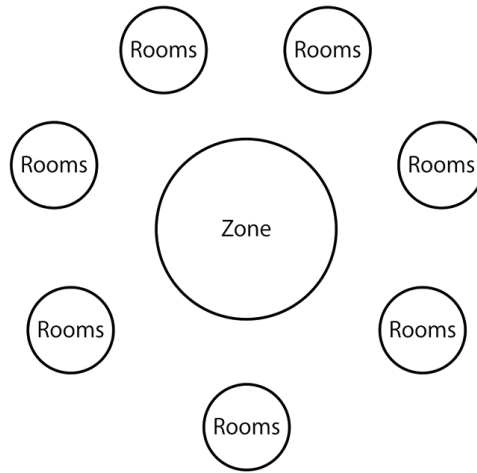


Figure 5: Zones (McLaughlin, 2014)

Service Groups are arbitrary, uniquely named groups of Services and are a convenient way to control Services across accessories e.g. *nightlight* (McLaughlin, 2014).

Actionsets Collection of actions that are executed together in an undefined order, e.g. *night* (McLaughlin, 2014).

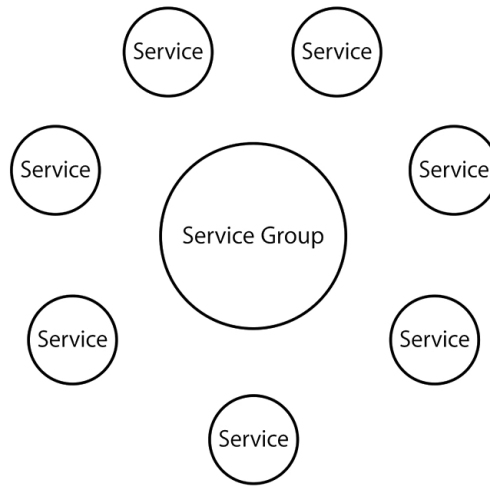


Figure 6: Service Groups (McLaughlin, 2014)

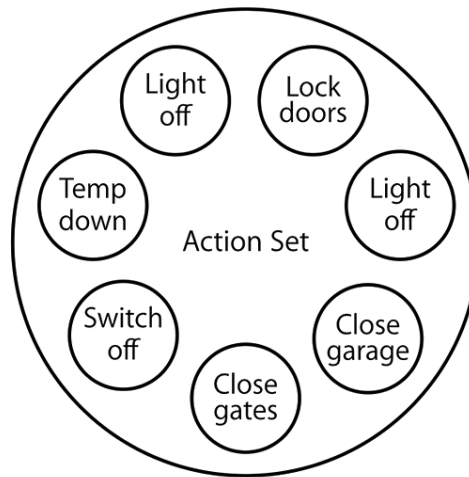


Figure 7: Actionsets (McLaughlin, 2014)

2.5.2 WWDC15

Security All communication between the Accessories and Apple HomeKit are end to end encrypted. Moreover Keys for encryption are changed after each session. Keys are not able to decrypt data from the past or the future. All data is encrypted using keys that are local on the device which ensures privacy of data. Further on administrative features for user management are added.

Maintain existing objects In order to guarantee uniqueness of Accessories, Apple has introduced a unique identifier (NSUUID) in iOS 9.

Predefined Scenes are scenes that occur in a regular every day cycle. Apple has provided some standard scenes like:

- Get up
- Leave
- Return
- Go to bed

Moreover these predefined scenes can not be deleted but customized and are recognized by Siri. These scenes are managed action sets, as known from iOS 8. Visual clues are information that are added by an Accessory Category.

Apple Watch Homekit is now available on watchOS2. All the home data is mirrored on the Apple Watch. Homes can be viewed, Accessories can be controlled and scenes can be executed by hand or over Siri.

Event Triggers Scenes that are executed at specified times of the day are known from iOS8. Events respond to the state of an Accessory or location based events by using geofences. Furthermore Events can be triggered:

- Time-based
- Significant events like:
 - Sunrise
 - Sunset

Time based triggers can be used in a natural way. One way to use them is to specify a time-based trigger to fire after 6pm another to trigger on Sundays.

Every trigger can be used in logical addition. To tie it all together every trigger has an associated scene which will be executed when the trigger is fired.

Remote Access allows the user to control their Accessories even if they are not at home. In order to get access to that feature the user needs an Apple TV 3rd gen. Verification is done by using the same Apple ID on both devices. For users without an Apple TV remote access is managed with the *HomeKit Accessory Protocol (HAP)* over iCloud. This means that you can control your Accessories and get notifications without an Apple TV, no matter where you are.

Bluetooth Low Energy The distance to an Accessory is crucial in order to control it over BTLE. Devices that are too far away from the user cannot be controlled. A secure connection is possible in iOS9 by HAP secure tunneling provided by an intermediate device. The Accessory is connected over BTLE to the intermediate device which is then exposed as an Object over WiFi. The range extender will also be able to provide remote access to all connected BTLE Accessories that works with HomeKit. To ensure privacy the intermediate device can not see the content of the HAP protocol.

Notifications BTLE Accessories fully support notifications and metadata for custom characteristics. Notifications transported over different channels are recognized by the iDevice featuring iOS9 as redundant.

Programmable Switches are used to map this event to a trigger and execute a scene. This feature is very powerful and useful to map an amount of actions to physical switches.

2.5.3 HAP - HomeKit Accessory Protocol

Transports There are two ways to connect Accessories to HomeKit, one is BTLE the other one is over IP.

Security is achieved by *Bi-directional authentication* and *Per-session encryption*.

Bi-directional authentication or mutual authentication is a technology where both parties involved in authentication are aware of each other. This means that the server is authenticating itself to the user as well as the user is authenticating itself to the server. The term *Authentication* describes the process of confirming identities and is done by verifying the validity of a client with a digital certificate.

SSL is a "cryptographic protocol designed to provide communication security over a computer network". The way that SSL works is by "authenticating the counterpart with whom they are communicating and to negotiate a symmetric key". "This session key is then used to encrypt data flowing between the parties". "An important property in this context is forward secrecy, so the short-term session key cannot be derived from the long-term asymmetric secret key". SSL is initialized in layer 5 (Session Layer) of the ISO/OSI model and has a handshake using an asymmetric cipher in order to establish cipher settings and shared key for that session. Afterwards the presentation layer (ISO/OSI 6) is providing the encryption for the data.

Network

2.5.4 Siri

HomeKit itself is an app independent service running in the background like Siri. Siri is Apples own natural language interpreter which is able to answer questions and perform actions on spoken and written user commands. Actions can be delegated to third party web services like Wolfram Alpha or in the domain of this thesis actions can control smart devices. Apple has logically concatenated the HomeKit database with Siri to enable seamless interaction between both services.

2.5.5 iDevices

Apple TV The Apple TV serves as a tunnel to the outside world in order to access your smart home devices from everywhere internet is accessible.

iPhone The proprietary HomeKit service is running in the background of all iDevices and can be accessed and managed from all iDevices registered to the same Apple ID.

Apple Watch The Apple Watch serves as an external display for iPhones. Further more it is packed with a microphone as well as a speaker which makes the watch perfect for interacting with Siri and your home.

2.5.6 Bluemix

Bluemix is an upcoming web service of IBM based on OpenStack software where users can create environments to test their code. Several programming languages are supported among them Java and NodeJS. Further there is a possibility to host an IoT environment called IoT Foundation. This service is backed up with a database and able to connect to IoT Foundation clients of

your choice. The clients are devices which are able to connect them selves to the internet and build up an connection to the IoT Foundation - smart home devices. Because of the fact that the devices have to handle ip communication its standing to a reason that they mostly serve as home gateways which are connected to more proprietary smart home devices.

2.5.7 IoT Foundation

The IoT Foundation is the smart home cloud solution of IBM where configured gateways connect themselves over mqtt to the cloud.

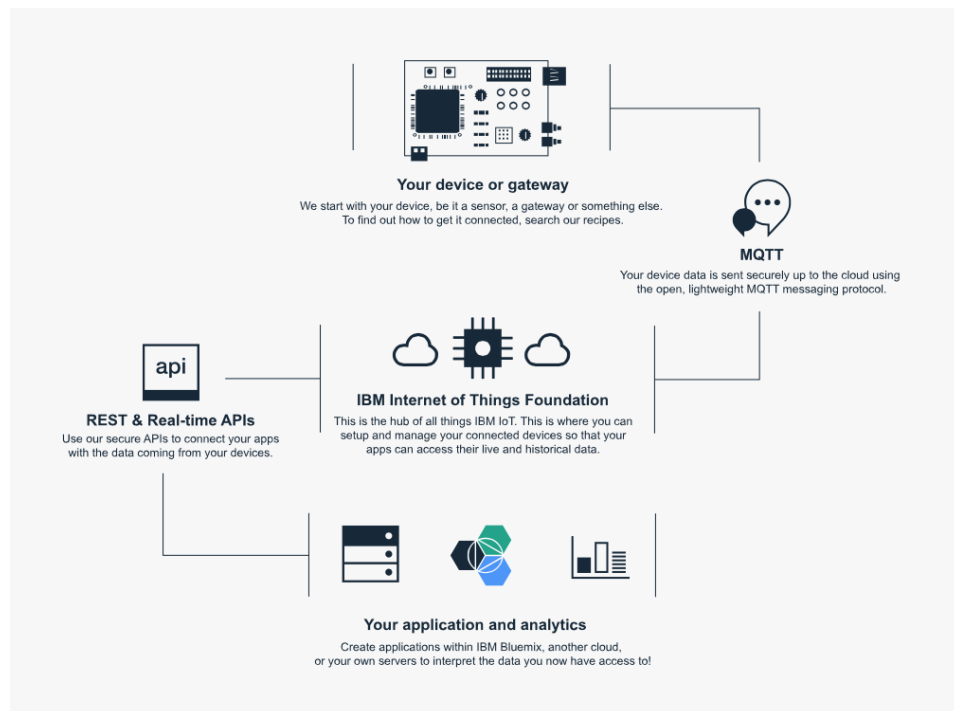


Figure 8: IoT Foundation overview (“IoT Foundation Ceoncept,” 2015)

Organizations IBM implemented the model of organizations into their home automation solution. After registration on the Internet of Things Foundation website a unique organization ID consisting of a six character identifier is paired to your account. Data from devices and applications are only accessible with the organization ID. Applications are bound to the organization ID by so

called API keys. Once an Application connects itself to the IoT Foundation it is directly bound to the organization ID which owns the Api key. Cross-organization communication is not implemented due to security reasons ("IoT Foundation Ceoncept," 2015).

Devices A device is per IBMs definition anything that can connect itself to the internet "and has data it wants to get into the cloud" ("IoT Foundation Ceoncept," 2015). Further a devices is not able to interact with other devices, however it accepts commands from applications. Devices can only be managed when registered to the IoT Foundation with a unique authentication token ("IoT Foundation Ceoncept," 2015).

Applications An Application is like a device anything that has a connection to the internet, but interacts with multiple devices. Applications have to identify them as well as devices. Additionally to the authentication token an application has to register itself to the IoT Foundation with an Api key ("IoT Foundation Ceoncept," 2015).

Events Data is published to the IoT Foundation with a mechanism called event, where devices control the content of an event as well as the name for each event ("IoT Foundation Ceoncept," 2015). The IoT Foundation is performing identification checks with the help of credentials. Therefore every event can be mapped to a specific device. IBM even claims that "with this architecture it is impossible for a device to impersonate another device" ("IoT Foundation Ceoncept," 2015). Further devices are unable to receive events, whether their own or from others. Applications are able to process events where they have access to the source of the event as well as the data it contains. "Applications can be configured to subscribe to all events from all devices, a subset of events, a subset

of devices or a combination of these events" ("IoT Foundation Ceoncept," 2015).

Commands To accomplish communication between applications and devices a mechanism called **command** is used, where only applications can send commands to specific devices. On the other side actions performed after receiving commands on the device side have to be determined by the device itself.

2.5.8 SDPvNext Rest API

SDPvNext is the front end and user management system for the IoT Foundation service. Currently being integrated into the IoT Foundation and extending its functionality by adding an authentication layer to it. SDPvNext is equipped accessible by a Rest API which provides basic rest calls to get devices and their configurations.

2.6 Constraints

HomeKit and SDPvNext HomeKit and SDPvNext are the systems to connect in the course of this bachelor thesis.

HomeKit aatabase accessibility The Homekit database is only accessible through iDevices, which means that only apps executed on an iPhone, iPad and iPod are able to add new devices to the HomeKit db.

SDPvNext Rest API The given Rest API is limited to only read values in form of states out of the IoT Foundation. Thus restricting interaction with the platform to a unidirectional communication from SDPvNext to the iOS app. It is yet not stated whether adding or removing devices is possible through the rest API.

General This thesis is being written from June 1st until August 21st 2015. Apple itself released only rare information about the functionality of HomeKit on June 8th. Further more during this period there were no user or professional reviews on HomeKit devices since they just hit the market on 28 of July. Any experiences are gathered through simulation of devices and may not be equivalent or differ to real ones.

2.7 Development Tools

2.7.1 Mac OSX

Xcode Xcode is the standard tool to develop apps for iOS devices.

Hardware IO Tools for Xcode 6 In oder to test the functionality of apps being developed Apple makes a good job in device simulation. Due to the

unavailability of real HomeKit devices the use of simulated ones is mandatory. Apple provides Hardware IO Tools to guarantee a testing environment which suits the real devices.

Homekit Accessory Simulator The Homekit Accessory Simulator is used to simulate Accessories and Bridges and acts just like real accessories. By default every Accessory has an Access Information Service containing basic information like the name, manufacturer, model and serial number. Bridges are a special type of Accessory that provide the functionality of a Hub, whereas Hubs provide access to Accessories that can't connect themselves directly to iOS. Every Accessory provided by a bridge is listed as separate Accessory later in order to interact directly with it (McLaughlin, 2014).

2.8 Middleware

Goal The goal of this thesis is to develop a middleware which is capable of communicating with the IoT Foundation as well as the HomeKit database. Changes in either db's should be applied to the corresponding one by adding or removing devices from the databases (for additional information refer to constraints). Further the Apple Watch should be able to monitor and control the listed devices.

3 Concept of Smart Home Solution

3.1 Applying measurements

The measurements defined in 2.1.4 are applied to Google, Apple and Samsung smart home solutions are compared against each other. SmartThings is a Kickstarter project acquired by Samsung but acting on their own and therefore listed separate. Evaluation is done on a scale from +2 for an excellent compliance to the measurements, all to -1 for not fulfilling the measurements.

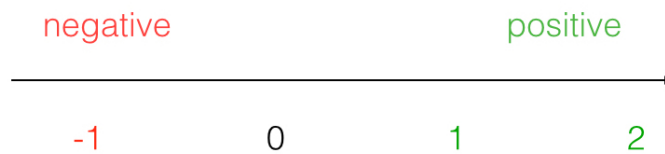


Figure 9: Evaluation Overview

3.1.1 Some words on trustworthiness

Trustworthiness in smart environments is relying on privacy and security. The Ubiquitous Computing Acceptance Model (UCAM) consists of the three mentioned parts and "explains (a) how privacy, security and trust are linked to one another; and (b) how trust is related with usage intention" (Oliver Sack, 2014). Where simple lightbulb controls have nearly no security or privacy impacts, devices like door locks do. The first approach of most iot and smart home novices is to control their light from any device which is regularly not considered to do so. Managing this first hurdle a new wave of home automation is being triggered. Two categories of devices are directly distinguishable:

- sensors

- actuators

Where sensors are used to measure or detect temperature, humidity, daylight or motion, actuators such as motors and switches have a higher risk to do significant damage.

3.1.2 Security

Apple Apple has reveiled its HomeKit Accessory Protocol (HAP) which is explained in 2.6.3. In summary Apple has invested quite some time in developing a fool proof information transport system between the supported devices. The only downside is that devices need to have more calculation power to provide the HAP security, which leads to lag on non suiting hardware and therefore leads to more expensive devices. Elgato "tweaked the firmware and added additional on-chip memory to handle the heavy-duty encryption" (Rossignol, 2015) to suit Apples requirements. Further all device information is stored locally at home.

Apples security system is valued positive (+2).

Google Google claims secure information transportation between devices both locally and through the cloud over Brillos weave. As well as availability for Android and iOS. No further information about technologies were found in the internet **therefore valued neutral (0).**

Samsung Samsung Smart Home solutions dropped 2014 in South Korea and The United States of America (Lohmann, 2014). Information on any security systems implemented by Samsung is hardly to find, maybe because there is none. David Lodge, who works for Pen Test Partners has disclosed some major security flaws in Samsungs Smart TVs (Mottl, 2015). voice data recorded over the Smart TV is send en clair to third party companies as well

as facial recognition information on persons standing in front of the tv (Mottl, 2015).



Figure 10: Overview of Samsung Smart Home functionality

Considering the fact that Samsungs Smart Home solution is sharing its data over the cloud and no adumbration on any security actions nor on it's official website or elsewhere are made, the educated assumption that there is no special security involved can be made.

Samsungs security system is valued negative (-1).

SmartThings Lately the SmartThings smart home solution experienced a vulnerability with its cloud communication. Data is in general pushed into the

cloud and then send to the according app to notify the user about anything happening at his home. This communication is encrypted using Secure Socket Lacer (SSL) standard. Validation of the cloud servers identity on the hub side is crucial for a secure communication, this is exactly where the failure happened. "This means that an attacker with privileged access to a user's home network (e.g. physical access) could have executed a "man-in-the-middle" attack that could have decrypted the communications between the SmartThings Hub and the SmartThings Cloud." ("SmartThings: Make Your World Smarter," 2015). This issue was fixed in a firmware update and further a third party security firm was hired to perform penetration tests on the system. Due to the fact that in general an SSL secured server communication is guaranteed and the security breach was fixed immediately with non affirmed user exploits and the fact that SmartThings will head off the cloud ("Smart Home: SmartThings wendet sich von der Cloud ab," 2015). Further Zigbee and Z-wave are encrypted radios. **Therefore smartThings is valued positive (+1).**

Inter-resume With safety in mind Apple is way ahead of other systems like Samsung. Samsung lacking in security is thrown out of the competition at first. SmartThings uses in general secured communication but currently stores all data in the cloud ("Smart Home: SmartThings wendet sich von der Cloud ab," 2015).

3.1.3 Smart home functions

Apple Apple build up a reliable security standard with the side effects of hardware claiming software. Hence at the time this thesis is written only few manufacturers have fulfilled Apples MFI program and only a handful of devices hit the market. But companies like Insteon are on the right track by providing bridges that connect all of their previously non homekit compatible hardware. Never the less basic functionalities like switches, lightbulbs and environment scanning devices as well as door locks did hit the market, fulfilling six out of seven smart home functions and **therefore Apple is rated positive (+1).**

Google Google covers with it's Nest family consisting of a thermostat, webcam and a smoke sensor relatively small domain of smart home devices compared to Apple, Samsung and SmartThings. Besides the fact that Googles devices comply with five out of seven smart home functions, basic light, switch and door lock compatibilities aren't fulfilled. **Therefore Google gets a negative credit on smart home functions (-1).**

Samsung Samsung is connecting it's smart productline consisting of lightbulbs, vacuum cleaners, refrigerators, washers, door locks and smart tvs all together in it's smart home solution. This quite rich smart home environment complies with six out of seven smart home functions only lacking in indoor environment controlling. **Therefore a positive rating (+1).**

SmartThings SmartThings has a wide variety of smart home devices solving nearly every problem you want to solve with your smart home solution as well as problems that didn't even exist before *feature-ism*. **Therefore a positive rating (+1).**

3.1.4 Cross compatibility

In this thesis cross compatibility pays special attention to extensibility of systems with third party products. Cross compatibility is valued combining the measurements of the Sullivan rule and future proof ness.

Apple In the Apple domain all systems are working like a charm but when it comes to cross compatibility Apple is very poor. Elgato and Insteon are therefore developing a bridge solution to connect their Smart Home pallet of devices which are inherently not able to speak Cupertino's language ("Future Proofing Your Smart Home for Apple HomeKit Compatibility," 2015). Keeping this in mind cross compatibility can be extended to an nearly infinity amount as the bridges are providing connectivity to more proprietary radios like Zigbee and Z-wave. Moreover the *sullivansRule* is fulfilled with the standard HomeKit devices being connected over already existing WiFi or Bluetooth Low Energy. When considering bridges which extend the HomeKit domain extra hardware is necessary, but the benefits overweight the use of one extra hardware:

"Insteon's offering makes all of their existing products compatible with HomeKit. This means you can already outfit your entire home, from outlets and plugs to thermostats and locks, with devices that will work with HomeKit. That's something that not very many companies can offer" ("Future Proofing Your Smart Home for Apple HomeKit Compatibility," 2015).

On the downside currently there is no possibility to connect non Insteon Zigbee or Z-wave solutions ("Future Proofing Your Smart Home for Apple HomeKit Compatibility," 2015).

All in all Apple HomeKit is valued positive (+1).

Google Google drives its cross compatibility with the help of its own IoT operation system Brillo, but their considered system nest, consisting only of a thermostat, a smoke sensor and a webcam is not sufficient enough for

overall smart home purposes nor fulfilling the smart home functions defined earlier nor in any way cross compatible, **therefore considered negative (-1)**.

Samsung Samsung has introduced its Smart Home Protocol (SHP) to enable the communication between different home appliances. Further it is intended to enable third party manufacturer to control their devices over the Samsung cloud. Future plans include Home-Energy, Secure Home Access, Healthcare und Eco Home Applications enabled (Pochanke, 2014). Due to the fact that no third party vendors were announced for over a year since Samsung dropped their Smart Home initiative the ability of cross compatibility is valued less than already available products. Current Samsung smart home solutions are

Samsung is valued negative (-1).

SmartThings SmartThings was developed with wide range compatibility in mind. "Its open system works with a wider selection of gadgets, and developers keep adding compatibility to more devices on a regular basis" (Carey, 2015). The wide variety of SmartThings compatible devices which can be connected to the hub are in parts useless for the daily life and fall into the category of *feature-ism* therefore being valued negative. Devices always have to be connected to a hub which doesn't serve the measurement of minimalism, but to keep in mind they are cross compatible to many devices and therefore valued positive. **All in all cross compatibility is valued positive (+2).**

Inter-resume Further a distinction must be made between smart home solutions like Apple which only provide software in form of services and protocols for third party companies which then have to implement Apples standards in order to play their part, where Samsung and Google provide their own hardware. SmartThings on the other hand provides software in form of an app as well as a

dedicated hub which allows third party systems to connect to. At the time this thesis is written, only Apple and SmartThings provide the ability to connect devices that are not specifically intended to work with the system. Moreover of the two just Apple HomeKit is out of the box voice controlled. Furthermore "mature" vendors like Elgato and Insteon provide a hub which provide connectivity of their none HomeKit product line to control them over the Apple solution. Google with its thin product line loses the cross compatibility race and its non-compliance in the predefined smart home functions.

intermediate result

- Apple +4
- Google -2
- Samsung -1
- SmartThings +4

3.1.5 Cost

All of the compared systems require smart phones in order to send commands and read out sensor data from smart home devices. For the cost calculation these smart phones will not be added to the total costs. It is questionable whether or not smart watches should be considered for the total cost estimation. Due to the fact that they are not required to operate a smart home and just have a subset of functions compared to there big brother phones they are not considered in the computation of costs.

Apple Apple certified smart devices are in the range of 50 to 200 Dollar. (1)

Google Nest device costs are in the range of 99 to 200 Dollar.(1)

Samsung Samsung smart home solutions are pretty cost intensive because their line of devices are regular domestic appliances with a little logic in it

to communicate to an app. Therefore cost ranges start at 150 up to multiple thousand Dollars.(0)

SmartThings A smartthings hub costs 100 Dollar.(1)

3.1.6 Extra

Extra points for minimalism are assigned if the smart home solution is providing voice control.

Apple Apple provides out of the box voice control with the help of its speech assistant siri (McLaughlin, 2014). this is a big step into seamless smart home controlling and fulfilling the minimalism criteria. Therefore an extra point for Apple (+1).

Google Google as well provides voice control functionality and gets an extra point (+1).

Samsung

Samsung as well provides voice control functionality and gets an extra point (+1).

SmartThings SmartThings is natively not supporting voice control, therefore zero extra points (0).

3.1.7 Comparison

The points total are shown in the following table:

Table 3: Evaluation of smart home solutions

Services	Apple	Google	Samsung	SmartThings
Security	2	0	-1	1
Smart Home Functions	1	-1	1	1
Cross Compatibility	1	-1	-1	2
Costs	1	1	0	1
Extra	1	1	1	0
Overall	6	0	0	5

3.1.8 The winner

Apple wins the competition hence the solution will be build upon HomeKit.

3.2 Concept

The basic idea of connecting SDPvNext and Apple HomeKit is to write a middleware which is running on a home gateway and providing a connection to SDPvNext as well as pushing the devices with HomeKit meta data into the local network in order to control them over iDevices and Siri. For clarification purposes a graphic of the idea is given:

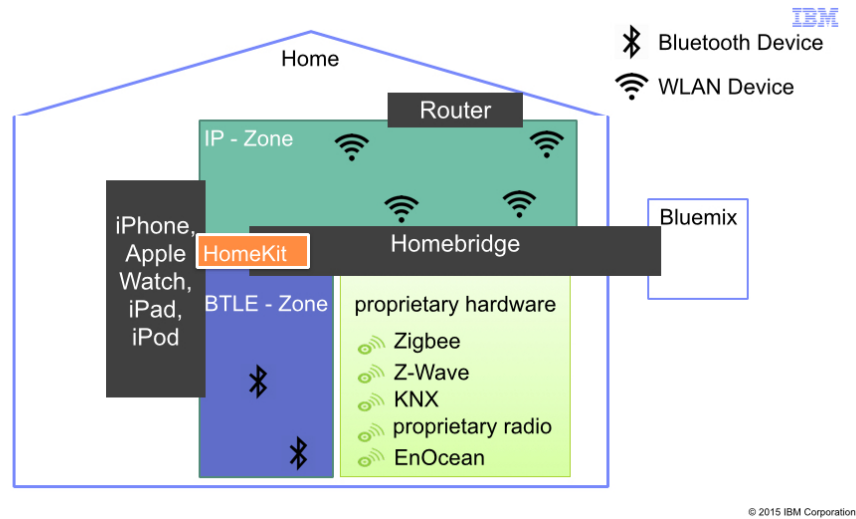


Figure 11: Concept idea

3.2.1 Possible solutions

Revisiting the graphic of HomeKits functionality, there are two possible solutions for an interconnectivity to SDPvNext marked in blue:

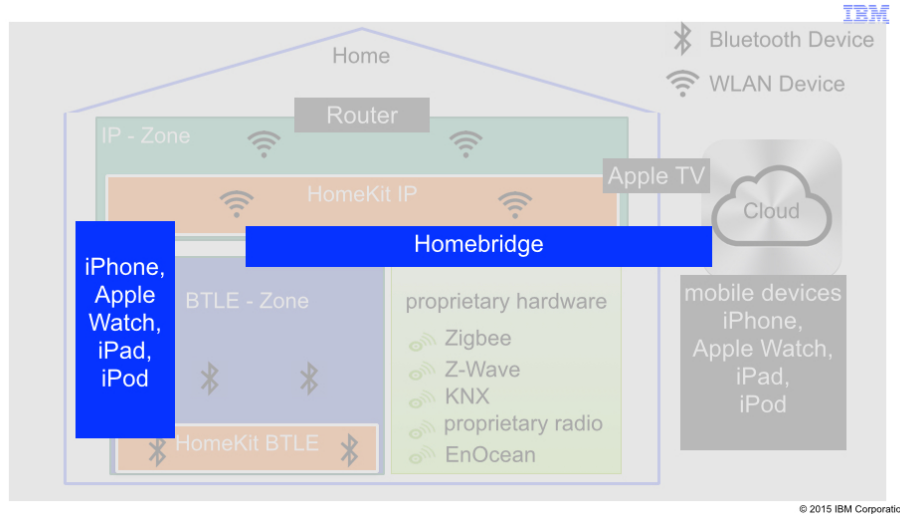


Figure 12: HomeKit middleware possibilities

On one hand a bridge equipped with extra software providing non HomeKit, SDPvNext provided devices the ability to be managed by Apple. On the other hand an iOS app using the SDPvNext rest and HomeKit api to provide inter connect ability between the two smart home solutions.

Bridge solution The bridge solution would aggregate non HomeKit devices connected to a home gateway in order to present them as HomeKit compatible devices in a WiFi environment.

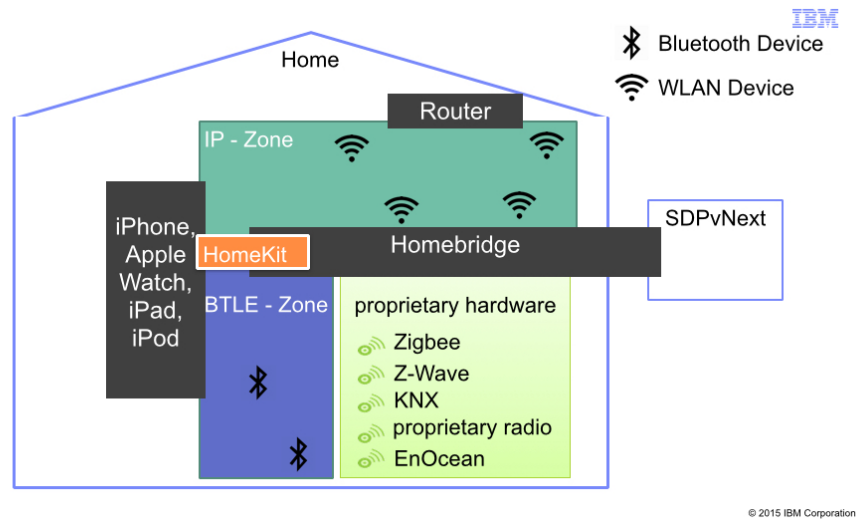


Figure 13: Overview BridgeSolution

SDPvNext Rest API + iOS App The iOS App makes use of the SDPvNext Rest Api to get information about connected devices and stores them in the HomeKit database.

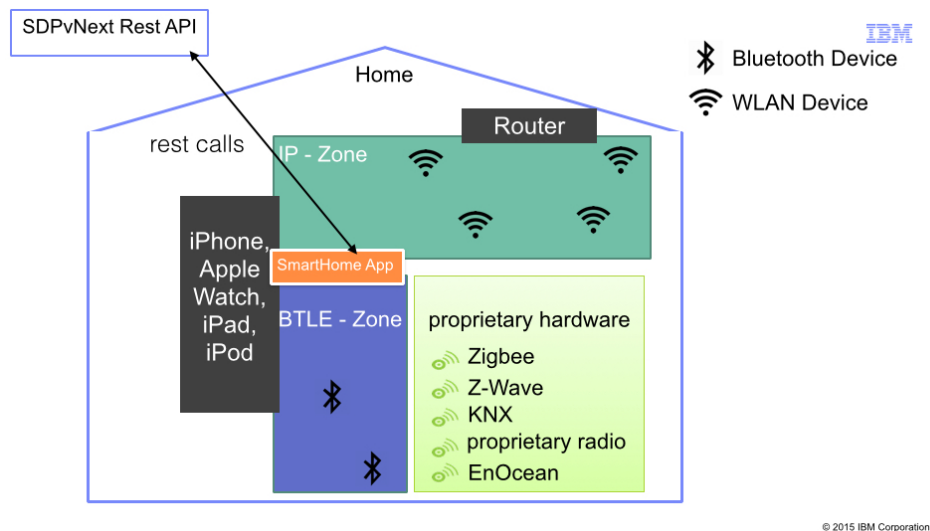


Figure 14: Overview iOS Solution

3.2.2 Advantages and disadvantages

Bridge solution The advantages and disadvantages of the bridge solution are provided in the following table:

Table 4: Advantages and disadvantages of the bridge solution

Advantages	Disadvantages
Proprietary radios can be connected directly to the home gateway without the workaround of SDPvNext.	<p>The bridge solution requires extra hardware in terms of a home gateway that provides server functions in order to represent non HomeKit hardware.</p> <p>Access to the Apple MFI program is necessary in order to be able to officially use the HAP protocol and therefore the bridge solution</p>

SDPvNext Rest API + iOS App

Table 5: Advantages and disadvantages of the iOS solution

Advantages	Disadvantages
The iOS App solution is independent of extra hardware	The iOS solution has to implement the HomeKit api and extra logic has to be added to the SDPvNext environment to cover meta data necessary to operate the HomeKit database.

3.3 Solution

The middleware's task is to connect to SDPvNext, load a json configuration file with all the information about the connected devices and store them into the HomeKit database. Further implementation includes enabling an http post request to set device states and to allow interaction. Apple provides a sample app where the whole spectrum of the HomeKit api is implemented ("Apple HomeKit sample app," 2015). This sample code can be accessed by anyone that registers to Apples developer program. To not exceed the frame of this thesis the sample code is not explained. Further the iOS solution explanation will not cover how devices will be connected to HomeKit and written to the database. There is plenty documentation in the web that explains how this is done. Lastly a WatchKit interface with a shared datastore is covered in order to view and manage your devices from the Apple Watch.

3.3.1 Design

The iOS app has four core functions that have to be implemented in order to show devices from SDPvNext.

3.3.2 Implementation

To keep the implementation simple the solution will be divided into five milestones:

- SDPvNext rest api
- Json parsing
- Presentation
- Apple WatchKit
- Data exchange

SDPvNext rest api First of all a connection has to be build up to the SDPvNext rest api. In order to accomplish this minor tasks have to be fulfilled.

The header for the connection has to be filled with authentication and request method information:

```
1      let username = "userName"
2      let password = "password"
3      let loginString = NSString(format: "%@:%@", username,
                                password)
4      let loginData: NSData = loginString.dataUsingEncoding
                                (NSUTF8StringEncoding)!
5      let base64LoginString = loginData.
                                base64EncodedStringWithOptions(nil)
```

Listing 1: Authentication Setup

Credentials have to be base64 encoded.

```
1      var Error: NSError?
2      var Url: NSURL = NSURL(string: "here comes the url"
                                )!
3      let Request = NSMutableURLRequest(URL: Url)
4      Request.HTTPMethod = "GET"
```

```
5         Request.setValue("Basic \$(base64LoginString)",  
                           forHTTPHeaderField: "Authorization")  
6     var Response: NSURLConnection?
```

Listing 2: Request Setup

Further the connection is started and received data is stored.

```
1     var Data = NSURLConnection.sendSynchronousRequest(  
        Request, returningResponse: &Response, error: &  
        Error) as NSData?
```

Listing 3: Create connection to receive json

Json parsing For further procedures the status code is checked whether or not the request was successful.

```

1      if let httpResponse = Response as?
        NSHTTPURLResponse {
2          if(httpResponse.statusCode == 200){
3              let parsedObject: AnyObject? =
                NSJSONSerialization.JSONObjectWithData(
                    Data!, options: NSJSONReadingOptions.
                        AllowFragments, error:nil)
4              ...

```

Listing 4: Check whether request was successful

Json data is parsed into an NSDictionary to get access to the inquired values. A dummy object is created to store the values.

```

1      ...
2      if let json = parsedObject as? NSDictionary {
3          for (key, value) in json{
4              if key as! String == "children"{
5                  let children = value as! NSArray
6                  for child in children{
7                      var device = [
8                          "no_location",
9                          "no_tenant",
10                         "no_name",
11                         "no_href",
12                         "no_type",
13                         "no_switch"
14                     ]
15                 ...

```

Listing 5: Json parsing

This is an example of how iteration through the different children is done.

```

1      ...
2      if let location = child["location"] as? NSString{

```

```
3                                     device[0] = location as
4                                     String
5                                     }
```

Listing 6: Json iteration

Send change requests SDPvNext in its current stage is not able to process any interaction over the rest api, however to make the middleware interactive a shaspa bridge rest api is used to send json in order to control connected devices. Where the connecting part is the same, the type of request has to be POST. Further a json string has to be build:

```
1      let jsonString = "{\"obix\":\"obj\",\"name\":\"(\"e
      .g. the name)\",\"tenant\":\"(\"e.g the user")
      \",\"children\": [{\"obix\":\"bool\",\"name\":\"
      Status\",\"val\":\"(\"e.g. the status which is
      set (e.g. either on or off)\")\"}]}"
```

Listing 7: Json build

In addition the header information has to be updated to consist *application/json* data:

```
1      Request.HTTPBody = jsonString.dataUsingEncoding(
      NSUTF8StringEncoding, allowLossyConversion:
      false)
2      Request.setValue("application/json",
      forHTTPHeaderField: "Content-Type")
```

Listing 8: Additional header information

Presentation Devices will be presented in a table view with a foldout submenu featuring an on/off button as well as a status label. This label houses many different values ranging from temperature to operating states:

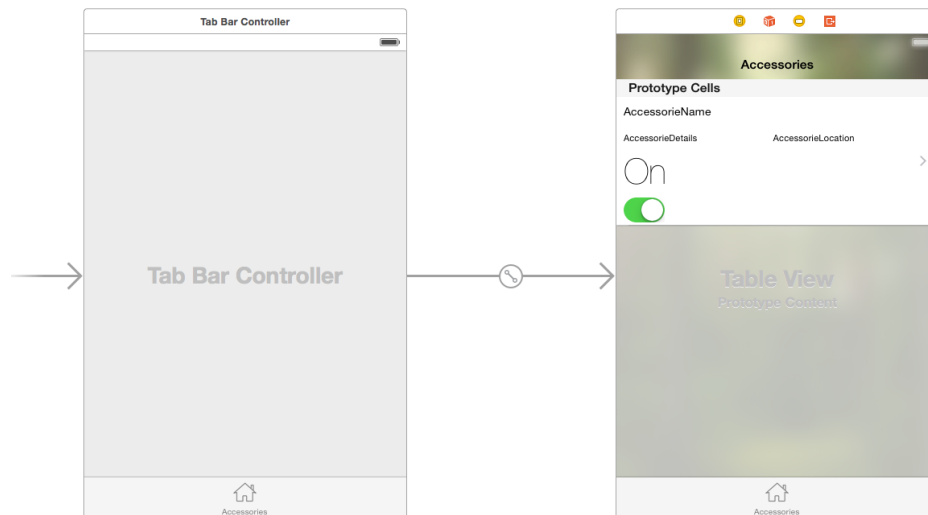


Figure 15: Device presentation

Apple WatchKit The interface running on the Apple Watch looks somewhat similar to the iOS app. Only difference is that the app is only displaying a filtered list of devices that can be managed directly, not including devices like servers or home gateways. This is necessary because of the small form factor of the watch's display. In addition to the iOS app property information is realized by a segue to a new detail view.

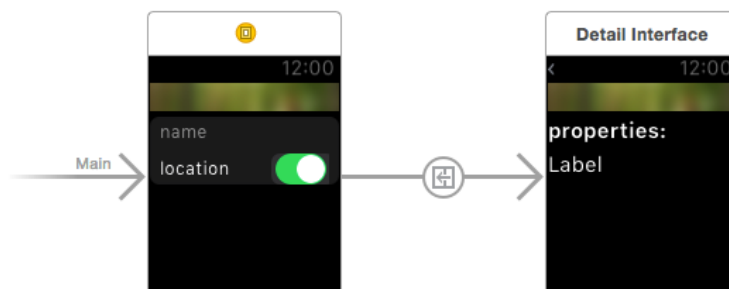


Figure 16: Watchkit interface

Data exchange

Exchange of data is handled by a datastore between the iOS and WatchKit app. This feature has to be enabled in the iOS apps capabilities. The Datastore-Protocol stores the keys that are used to identify the values in the datastore.

```

1      let kaccessorieName      = "kaccessorieName"
2      let kaccessorieType      = "kaccessorieType"
3      let kaccessorieLoc       = "kaccessorieLoc"
4      let kaccessorieProp      = "kaccessorieProp"
5      let kaccessorieStat      = "kaccessorieStat"
6      let kaccessorieHref      = "kaccessorieHref"
7      let kaccessorieTenant    = "kaccessorieTenant"
8
9      protocol DatastoreProtocol {
10         func save(#key: String, value: NSArray)
11         func load<T>(#key: String) -> T?
12         func commitToDisk()
13     }

```

Listing 9: DataStore keys

The default datastore has to be specified as well as the type of data that is stored. Additionally methods to write and load from the datastore are provided.

```

1      //created custom datastore: "here comes your
2      datastore"
3      class SharedUserDefaultsDatastore: NSObject,
4          DatastoreProtocol {
5          let userDefaults = UserDefaults(suiteName: "
6              datastore url")!
7
8          func save(#key: String, value: NSArray) {
9              userDefaults.setObject(value, forKey: key)
10         }
11
12         func load<T>(#key: String) -> T? {

```

```
10         let obj: AnyObject? = userDefaults.objectForKey
11           (key)
12
13         if let validObj = obj as? T {
14             return validObj
15         }
16         else {
17             return nil
18         }
19     }
20
21     func commitToDisk() {
22         userDefaults.synchronize()
23     }
24 }
```

Listing 10: Custom dataStore

3.3.3 Additional work

The given implementation shows how devices can be loaded from SDPvNext and how the received json is parsed to access information. As already mentioned the HomeKit api is not explicitly covered due to the fact that this topic would fill an entire thesis on its own. This does not imply that further research on this theme is obsolete. To add devices to the HomeKit database additional meta data is required as well as a pairing code for every single device.

Problem The SDPvNext's backend has to store these additional information to provide full cross compatibility coverage. Changes in the backend of SDPvNext are out of the scope of this thesis but will be recommend to the developers. At least the possibility to load simple prototypes that match all the necessary requirements for a HomeKit device should be implemented to enable a cross compatibility of 100%.

4 Summary

Reflecting the path of this thesis the definition of a home successfully led to requirements for smart home technology to actually improve everyday life. Moreover in the scope of this work the Sullivans rule implies that smart home solutions do not have to be fancy. Although some vendors like Samsung rely on feature-ism to maintain sales. This thesis highlights the core needs a smart home solution has to fulfill. Among others Apple HomeKit complies best with these measurements and is therefore chosen as part of the middleware solution.

The superior solution out of two possible middleware solutions is chosen and implemented. Although the design covers the whole HomeKit api implementation does not. Due to the fact that HomeKit itself is documented well enough and an extra documentation would blast the scope of this thesis. Changes in the device data model in SDPvNext and the IoT Foundation have to be implemented since HomeKit needs meta data like a pairing code which must be stored alongside regular data. The goal of this thesis is to develop a middleware however changes in the two systems that are supposed to be connected are out of scope.

4.1 Reflection

The elaborated smart home measurements security, smart home functions and cross compatibility are all based on evaluating definitions of smart home and iot as well as basic home functions that nearly everyone can rely to as common sense. By applying these measurements an adequate smart home solution is encountered as part of the middleware solution. To choose the right solutions out of two was tricky business but overall the homebridge solution lost because there is no documentation on the hap protocol which makes implementation nearly impossible.

4.2 Result

Where the middleware solution fulfills its supposed task considering the given limitations of SDPvNext. Future work includes implementing the HomeKit api into the existing middleware but is obsolete until SDPvNext expands its interactivity possibilities to actually control devices over rest. Further tasks include considering large corporation solutions and redefining measurements for their domain.

A big task still remains in the topic of smart home, namely defining official standards. New thesises should cover this topic to actually improve the overall situation inside the world of iot. Another good thread to keep an eye on is the buzzword fog. I guarantee that this as soon as people understand the benefits of this kind of paradigm, big data will get really handy. Further an analysis of user or company driven development trend is predictable. Thank you for reading this thesis!

List of Figures

1	Social functions for rooms	5
2	Evaluation Overview	11
3	Overview of Apple HomeKit functionality	12
4	Common database provided by HomeKit (McLaughlin, 2014) . .	19
5	Zones (McLaughlin, 2014)	21
6	Service Groups (McLaughlin, 2014)	22
7	Actionsets (McLaughlin, 2014)	22
8	IoT Foundation overview (“IoT Foundation Ceoncept,” 2015) . .	27
9	Evaluation Overview	32
10	Overview of Samsung Smart Home functionality	34
11	Concept idea	42
12	HomeKit middleware possibilities	43
13	Overview BridgeSolution	44
14	Overview iOS Solution	44
15	Device presentation	52
16	Watchkit interface	52

List of Tables

1	Homekit Accessory Profiles	7
2	Mapping of smart home services to basic home functions	9
3	Evaluation of smart home solutions	41
4	Advantages and disadvantages of the bridge solution	45
5	Advantages and disadvantages of the iOS solution	45

Listings

1	Authentication Setup	47
2	Request Setup	47
3	Create connection to receive json	48
4	Check whether request was successfull	49
5	Json parsing	49
6	Json iteration	49
7	Json build	51
8	Additional header information	51
9	DataStore keys	53
10	Custom dataStore	53

References

- Ashton, K. (1999). Iot definition.
- Carey, B. (2015). *A renter's guide to a smart home*.
- Chacos, B. (2013). *Why moore's law, not mobility, is killing the pc*. Retrieved from <http://www.pcworld.com/article/2030005/why-moores-law-not-mobility-is-killing-the-pc.html>
- Knuth, D. E. (1998). *The art of computer programming, volume 3: (2nd ed.) sorting and searching*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc.
- Lohmann, R. F. (2014). *Samsung smart home: den haushalt per app steuern*. Retrieved from <http://www.computerbild.de/artikel/cb-News-PC-Hardware-Samsung-Smart-Home-App-Haushalt-steuern-9963297.html>
- Machine, C. V. (2012). *The "only" coke machine on the internet*. Retrieved from http://www.cs.cmu.edu/~coke/history_long.txt
- Magrassi, P. (2001). *E-tagging: from niches to the supranet*.
- Magrassi, P. (2002). *A world of smart objects*.
- Martellaro, J. (1990). *That thing you do with apple? don't do it*. Retrieved from <http://www.macobserver.com/tmo/article/that-thing-you-do-with-apple-dont-do-it>
- McLaughlin, K. (2014). Introduction to homekit. [Online; accessed 10 June 2015]. Apple WWDC14. Retrieved from <https://developer.apple.com/videos/wwdc/2014/#213>
- Mielke, F. L. (2015). personal communication. Frank Leo Mielke, IBM Employee.
- Mottl, J. (2015). *Samsung smart tv doesn't encrypt voice data: hackers' delight?* Retrieved from <http://www.techtimes.com/articles/33805/20150219/samsung-smart-tv-doesnt-encrypt-voice-data-hackers-delight.htm>
- Oliver Sack, C. R. (2014). *"like a family member who takes care of me" users' anthropomorphic representations and trustworthiness of smart home environments*. Retrieved from <http://vwhci.avestia.com/2014/004.html>

Webster's new world college dictionary. (n.d.). Retrieved from <http://websters.yourdictionary.com>

Raising young refugees' voices in europe and beyond. (2008). Retrieved from https://www.coe.int/t/dg4/youth/Resources/Publications/Presentations/069_Young_Refugees_Voices_en.asp

SmartThings: Security Issue. (2012). Retrieved from <https://community.smartthings.com/t/security-announcement-from-smartthings/10950>

Apple HomeKit sample app. (2015). Retrieved from <https://developer.apple.com/library/ios/samplecode/HomeKitCatalog/Introduction/Intro.html>

Future Proofing Your Smart Home for Apple HomeKit Compatibility. (2015). Retrieved from <http://www.makeuseof.com/tag/future-proofing-smart-home-apple-homekit-compatibility/>

Google Brillo und Weave: Ein Android fürs Internet der Dinge. (2015). Retrieved from <http://www.heise.de/newsticker/meldung/Google-Brillo-und-Weave-Ein-Android-fuers-Internet-der-Dinge-2670592.html>

Google Developers - Brillo. (2015). CNET Magazine. Retrieved from <https://developers.google.com/brillo/>

In The U.S. Smart Home Market, Don't Rule Out China. (2015). Retrieved from <http://techcrunch.com/2015/06/11/in-the-u-s-smart-home-market-dont-rule-out-china/#.dpvbs4:xPfK>

Internet of Things - Techcrunch. (2015). Retrieved from <http://techcrunch.com/topic/subject/internet-of-things/>

IoT Foundation Ceoncept. (2015). Retrieved from <https://docs.internetofthings.ibmcloud.com/reference/concepts.html>

Lutron History. (2015). Retrieved from <http://www.lutron.com/en-US/Company-Info/Pages/AboutUS/OurStory.aspx>

Nest: Erstmals neue Smart-Home-Geräte seit Übernahme durch Google. (2015). Retrieved from <http://www.heise.de/newsticker/meldung/Nest-Erstmals->

neue-Smart-Home-Geraete-seit-Uebernahme-durch-Google-2714910.
html

Smart Home: SmartThings wendet sich von der Cloud ab. (2015). Retrieved from <http://www.heise.de/newsticker/meldung/Smart-Home-SmartThings-wendet-sich-von-der-Cloud-ab-2594477.html>

SmartThings: Make Your World Smarter. (2015). Retrieved from <https://www.kickstarter.com/projects/smartthings/smartthings-make-your-world-smarter>

Why Is My Smart Home So Fucking Dumb? (2015). Retrieved from <http://gizmodo.com/why-is-my-smart-home-so-fucking-dumb-1684949715>

PatriziaGufler. (2015). personal communication. PatriziaGufler, IBM Employee.

Pochanke, S. (2014). *Ces 2014: samsung zeigt neues smart home-konzept.*
Retrieved from <http://www.giga.de/events/ces/news/ces-2014-samsung-zeigt-neues-smart-home-konzept/>

Raji, R. (n.d.). Iot definition.

Rossignol, J. (2015). *Apple's strict bluetooth le security requirements slowing rollout of homekit accessories.* Retrieved from <http://www.macrumors.com/2015/07/22/apple-bluetooth-le-security-slowing-homekit/>

Sullivan, L. (1930). Form follows function. rediscovered in 1930, originally between 1805-1852.

Appendix