



Smart Home mit Open Home Automation Bus (OpenHAB)

Lukas Kiederle
Dominik Ampletzer
Daniel Böning
Fakultät für Informatik

WS 2019/20

Inhaltsverzeichnis

1	Motivation	4
2	Was ist OpenHAB	4
3	Datenintegriertät und Sicherheit	4
4	OpenHAB aus technischer Sicht	4
4.1	Bindings	5
4.2	Things	5
4.3	Channels	5
4.4	Items	5
4.5	Rules	6
4.6	Sitemaps	6
4.7	Api	6
5	Verwendung von OpenHAB	7
5.1	Integration der Big Player	7
5.2	Beispiel Aufbau eines OpenHAB Smart-Homes	7
5.3	Umgang mit OpenHAB	8
6	Fazit	8
6.1	Stärken	8
6.2	Schwächen	9
7	Infos:	9
8	Vorlage mit Samples	10
A	Erster Abschnitt des Anhangs	11

1 Motivation

Diese Ausarbeitung wird für das Fach Softwarearchitektur an der technischen Hochschule Rosenheim geschrieben. Das Ziel ist es OpenHAB, ein Heimautomatisierungs-Tool, aus praktischer und technischer Sicht zu untersuchen. Dabei liegt der Fokus vor allem auf der Softwarearchitektur von OpenHAB.

2 Was ist OpenHAB

OpenHAB ist eine technologie-unabhängige Open-Source-Automatisierungssoftware für Smart-Homes. Sie wurde von Kai Kreuzer 2010 erstmals initiiert und wird mittlerweile durch die Community weiterentwickelt. OpenHAB ist in Java geschrieben und aktuell in der Version 2.4 erhältlich.

Auf der offiziellen Website von OpenHAB <https://www.openhab.org/> sind drei klare Hauptziele definiert, die diese Software erreichen soll. Dabei ist ein Ziel die Plattformunabhängigkeit. Somit kann OpenHAB sowohl auf Linux, MacOS oder Windows betrieben werden. Auch das hosten mit Docker oder einem Raspberry Pi wird unterstützt. Weiterhin soll es durch die Plugin-Architektur möglich sein, fast jedes Gerät zu integrieren. Es werden über 200 Technologien und mehrere tausende verschiedene Geräte unterstützt. Das dritte Ziel weist auf die vielen verschiedenen Automatisierungsmöglichkeiten hin, die OpenHAB zu bieten hat. Dabei werden Auslöser, Aktionen, Skripte und auch Voice-Kontrolle genannt.

3 Datenintegriertät und Sicherheit

<https://www.openhab.org/docs/installation/security.html>

- Through the command line console, which is done through SSH and thus always authenticated and encrypted. You will find all details about this in the Console documentation.
- Through HTTP(S) over `https://<ip>:8443`
- Options for Secure Remote Access
 - VPN: The most secure option is probably to create a VPN connection to your home network
 - myopenHAB Cloud Service with a tunnel that forwards all requests to the openHAB instance
 - Running openHAB Behind a Reverse Proxy: A reverse proxy simply directs client requests to the appropriate server. This means you can proxy connections to `http://mydomain_or_myip` to your openHAB runtime.

4 OpenHAB aus technischer Sicht

In diesem Kapitel sind die grundlegenden Komponenten, die OpenHAB verwendet, tabellarisch dargestellt. Anschließend wird detaillierter auf die einzelnen Elemente eingegangen.

Konzept	Beschreibung
---------	--------------

Bindings	sind die openHAB-Komponenten, die die Schnittstelle zur elektronischen Interaktion mit Geräten bereitstellt.
Things	sind die erste von openHAB (Software) generierte Darstellung von Geräten.
Channels	sind die openHAB (Software)-Verbindung zwischen „Dingen“ und „Gegenständen“.
Items	sind die von openHAB (Software) generierte Darstellung von Informationen über die Geräte.
Rules	führen automatische Aktionen durch (in einfachster Form: wenn „dies“ passiert, wird openHAB „das“ tun).
Sitemaps	ist die von openHAB (Software) generierte Benutzeroberfläche (Website), die Informationen präsentiert und Interaktionen ermöglicht.

Tabelle 1: OpenHAB Komponenten

4.1 Bindings

- Typische Bindings
- Screenshot?
- Geräteerkennung

4.2 Things

-
-

4.3 Channels

- Dient als Kommunikationsweg für von Openhab zu things.
- Channel sind beispielsweise bei einer Lampe die Brightness.

4.4 Items

- Item bedeutet nicht Gerät oder Service
- Items können gruppiert werden
- Ein Item stellt einer der Basistypen dar: String, Number, Color oder Group.
- Items werden über mit Hilfe von Bindings mit der Außenwelt verbunden
- <https://www.openhab.org/docs/configuration/items.html#items>

```

1      Group groundFloor
2      Switch kitchenLight (groundFloor)
3      Switch livingroomLight (groundFloor)

```

Codebeispiel 1: Item-Gruppierung Beispiel

4.5 Rules

- Rules stellen wenn dann Beziehungen dar
- Diese können sowohl über zusammenklicken, als auch programmatisch erstellt werden.
- Das Zusammenklicken basiert auf einem noch nicht fertigen Feature namens experimental rules. Zum Zeitpunkt dieser Arbeit können dadurch schon einige Standardrules definiert werden. Allerdings fehlt beispielsweise noch die Vergleichsoption größer oder größer-gleich
- Codeblock 2 zeigt ein programmatisch erstellte Rule.
 - Eine Rule besteht immer aus einem Namen, einer when-Bedingung und einem then Abschnitt.
 - Name dient als Zuordnung
 - When ist der Trigger bzw. Auslöser der Aktion, welche im then Block definiert ist.
 - Diese Rule prüft, ob die Lautstärke des Items (TV) Dominik_volumen sich verändert hat. Wenn das der Fall ist, wird eine geprüft, wiehoch denn die aktuelle Lautstärke ist. Folglich wird bei unter 20 die Lampe gedimmt und bei über 20 die Lampe erhellt.
 - Falls etwas nicht klappen sollte, können auch Debug-Ausgaben mit dem Kommando logDebug geschrieben werden.

```
1 rule "React on Volume (LGWebOSTVUH620VDominik_Volume) change"
2 when
3     Item LGWebOSTVUH620VDominik_Volume changed
4 then
5     logDebug("React some changes on Volume", "some Message" +
6         LGWebOSTVUH620VDominik_Volume.state.toString)
7 if ( LGWebOSTVUH620VDominik_Volume.state >= 20 ) {
8     HueWhiteLamp2_Brightness.sendCommand(80)
9 }
10 else {
11     HueWhiteLamp2_Brightness.sendCommand(5)
12 }
13 end
```

Codebeispiel 2: Beispiele Rule Beispiel

4.6 Sitemaps

4.7 Api

<https://www.openhab.org/docs/configuration/restdocs.html>

- Item ein-/ausschalten
- Eine List von allen Items, Sitemaps ausgeben lassen
- Mit einem Editor auf die ganzen Komponenten zugreifen:

- Visual Studio Code installieren
- Openhab Extension installieren
- Geteiltes Openhab Laufwerk als Ordner öffnen
- Openhab Extension konfigurieren
- Es werden auch andere Editoren unterstützt

5 Verwendung von OpenHAB

5.1 Integration der Big Player

- Amazon Alexa und Echo Dot Integration möglich
 - Alexa:
 - * This certified Amazon Smart Home Skill allows users to control their openHAB powered smart home with natural voice commands. Lights, locks, thermostats, AV devices, sensors and many other device types can be controlled through a user's Alexa powered device like the Echo or Dot.
 - * <https://www.openhab.org/docs/ecosystem/alexa/>
 - * <https://www.openhab.org/addons/bindings/amazonechocontrol/>
 - Google Home
 - * Google Home Integration möglich
 - * With the Action you can voice control your openHAB items and it supports lights, plugs, switches and thermostats. The openHAB Action comes with multiple language support like English, German or French language.
 - * The openHAB Action links your openHAB setup through the myopenHAB.org cloud service to the Google Assistant platform
 - * openHAB Cloud Connector configured using myopenHAB.org . (Items DO NOT need to be exposed to and will not show up on myopenHAB.org , this is only needed for the IFTTT service!) Google account. Google Home or Google Home mini.
 - * <https://www.openhab.org/docs/ecosystem/google-assistant/>

5.2 Beispiel Aufbau eines OpenHAB Smart-Homes

- OpenHAB auf Raspberry Pi 3/4 installiert
- Welche Geräte haben wir mit OpenHAB verbunden?
 - Spotify
 - * Lautstärkeregler
 - * Aktueller Song Display
 - LG Smart TV
 - * Lautstärkeregler
 - * An- und ausschalten

- * One-Way-Chat
- Lampen
- Wie haben wir die Geräte verbunden?
 - **Verschiedene Binding:**
 - Spotify Binding
 - LG Smart TV Binding
- On the server the configuration is stored somewhere in userdata (/var/lib/openhab2 for apt-get installs). In an upgrade the userdata folder is preserved when using apt-get.

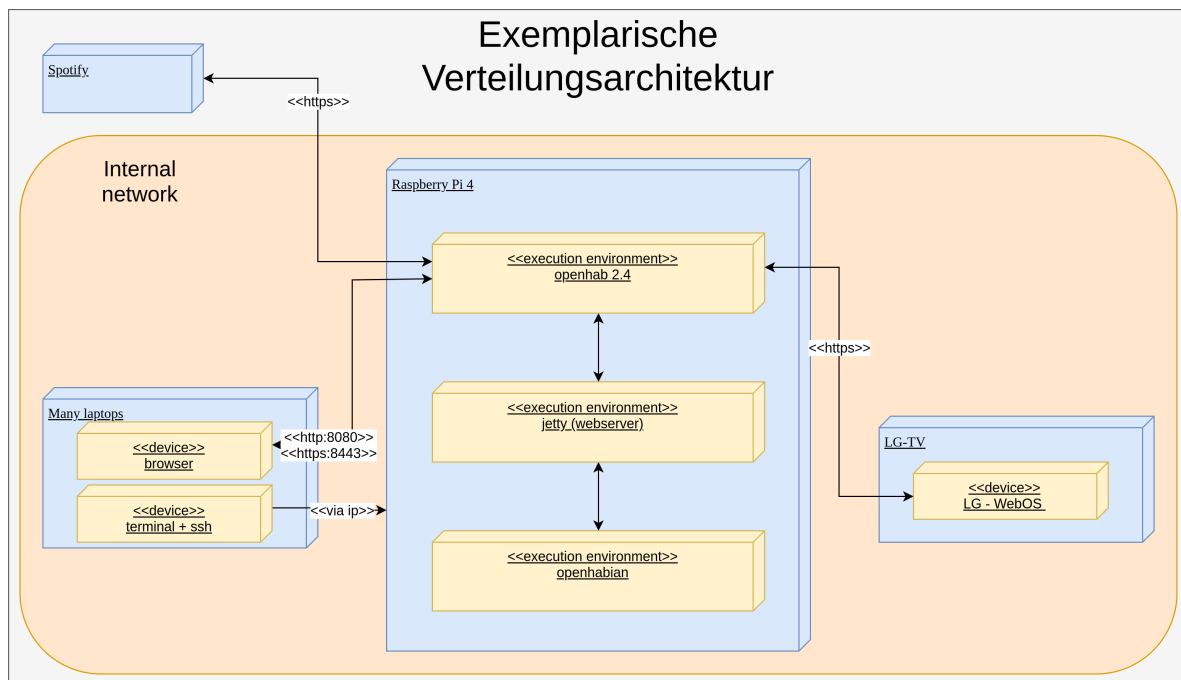


Abbildung 1: Verteilungsarchitektur

5.3 Umgang mit OpenHAB

- Das meiste klickt mans ich zusammen: Bindings, Rules, Channels, Items, Things
- Implementierung von rules scheint idiotensicher, weil:
 - einfacher Syntax
 - Abhängigkeiten managed Openhab
- Bindings schreiben scheint eher schwieriger

6 Fazit

6.1 Stärken

Some of openHAB's strengths are:

Its ability to integrate a multitude of other devices and systems. openHAB includes other home automation systems, (smart) devices and other technologies into a single solution To provide a uniform user interface and a common approach to automation rules across the entire system, regardless of the number of manufacturers and sub-systems involved Giving you the most flexible tool available to make almost any home automation wish come true; if you can think it, odds are that you can implement it with openHAB.

6.2 Schwächen

Wollen wir das hier als SWOT Analyse aufziehen?

- Integration von USB-Geräten scheint eher kompliziert. Vor allem auf Raspberry Pi
- Serial Binding wird nicht angezeigt
 - Mikrofon an Raspberry Pi oder anderes Geräte verbinden
 - Input des Mikrofons über OpenHAB an ein Ausgabegerät, wie zum Beispiel eine Bluetooth Box, senden und abspielen
 - Raspberry hat da auch für große Probleme bei der Geräteerkennung gesorgt - USB gerät wurde nicht im devices Verzeichnis aufgeführt und somit konnte auch keine Verbindung mit OpenHAB aufgebaut werden
 - OpenHAB Serial Device Binding wurde auch nicht angezeigt, um Geräte darüber zu suchen

7 Infos:

Ausgangslage Untersuchen Sie die Architektur und Features von OpenHAB und schreiben Sie ein Beispielanwendung. Mit myOpenHub existiert eine kostenlose Plattform die sie nutzen können.

Beantworten Sie dabei

- Aktueller Status des Projekts und
- Integration der Big Player wie Alexa und Google Home
- Welche Tools und Konzepte und APIs gibt es
- Welche Deployment Modi und Betriebsmodi existieren
- Untersuchen Sie auch Aspekte wie Datenintegriertät und Sicherheit

Unterlagen Linkes

- <https://www.myopenhab.org/>
- <https://www.openhab.org/>
- <https://jaxenter.de/openhab-2-4-78711>

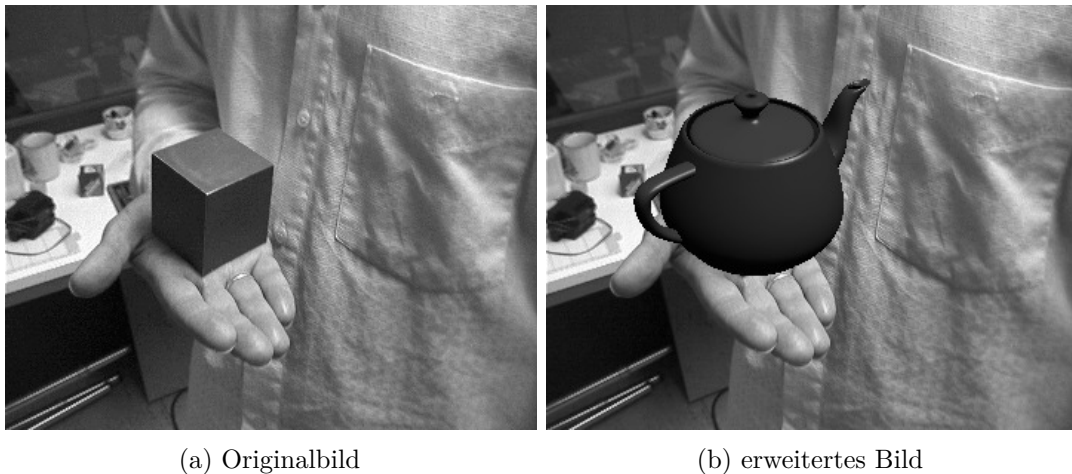


Abbildung 2: Beispiel eines Augmented Reality Systems: es folgt eine Beschreibung (Bilder aus [Sch01])

Sequence	ARTS	wman	stcams	ARTVZ	ARTSUZ
# Frames	190	40	400	270	190
# relative movements	17955	780	79800	36315	17955
# movements after pre-sel.	14336	623	37915	21788	14343
min. angle in seq.	0.233°	5.95°	0.154°	0.00000171°	0.0388°
max. angle in seq.	81.7°	180°	47.3°	80.3°	80.9°
min. angle after pre-sel.	12.9°	21.1°	17.3°	16.3°	12.9°
max. angle after pre-sel.	81.7°	161°	47.3°	80.3°	80.9°

Tabelle 2: Datenselektion für verschiedene Testdatensätze.

8 Vorlage mit Samples

Einen Überblick findet man z. B. in [Aue00].

Ein Beispiel wird in Abb. 2 gezeigt. Das verwendete Objekt ist in Abb. 2a dargestellt, das Ergebnis in Abb. 2b.

Eine Formel

$$f(x) = \frac{1}{3}x + 5, \quad x \in \mathbb{R}. \quad (1)$$

Und noch eine:

$$\mathbf{M} = \mathbf{A}\mathbf{x}\pi, \quad \mathbf{A} \in \mathbb{R}^{2 \times 2}, \mathbf{x} \in \mathbb{R}^2. \quad (2)$$

Tabelle 2 gibt einen Überblick über XYZ.

A Erster Abschnitt des Anhangs

In diesem Anhang wird . . .

Literatur

- [Aue00] T. Auer. *Hybrid Tracking for Augmented Reality*. Dissertation, Technische Universität Graz, Graz, Austria, 2000.
- [Sch01] J. Schmidt, I. Scholz und H. Niemann. Placing Arbitrary Objects in a Real Scene Using a Color Cube for Pose Estimation. In B. Radig und S. Florczyk, Hg., *Pattern Recognition, 23rd DAGM Symposium*, Bd. 2191 von *Lecture Notes in Computer Science*, S. 421–428. Springer-Verlag, Berlin, Heidelberg, New York, 2001.