



# **Smart Home mit Open Home Automation Bus (OpenHAB)**

Lukas Kiederle  
Dominik Ampletzer  
Daniel Böning  
Fakultät für Informatik

WS 2019/20



# Inhaltsverzeichnis

<b>1</b>	<b>Motivation</b>	<b>4</b>
<b>2</b>	<b>Was ist OpenHAB</b>	<b>4</b>
<b>3</b>	<b>OpenHAB aus technischer Sicht</b>	<b>4</b>
3.1	Bindings . . . . .	4
3.2	Things . . . . .	5
3.3	Channels . . . . .	5
3.4	Items . . . . .	5
3.5	Rules . . . . .	5
3.6	Sitemaps . . . . .	5
<b>4</b>	<b>Exemplarische Verwendung von OpenHAB</b>	<b>5</b>
<b>5</b>	<b>Fazit</b>	<b>5</b>
5.1	Stärken . . . . .	5
5.2	Schwächen . . . . .	6
<b>6</b>	<b>Infos:</b>	<b>6</b>
<b>7</b>	<b>Vorlage mit Samples</b>	<b>7</b>
<b>A</b>	<b>Erster Abschnitt des Anhangs</b>	<b>9</b>

# 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. Ein Ziel ist plattform-unabhängig zu sein. 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 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 (im 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

### 3.1 Bindings

- Typische Bindings

- Screenshot?
- Geräteerkennung

### 3.2 Things

### 3.3 Channels

### 3.4 Items

### 3.5 Rules

### 3.6 Sitemaps

## 4 Exemplarische Verwendung von OpenHAB

- Wie ist OpenHAB installiert (OpenHAB cloud oder auf raspi?)
- Welche Geräte haben wir mit OpenHAB verbunden?
- Wie haben wir die Geräte verbunden?
- 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.

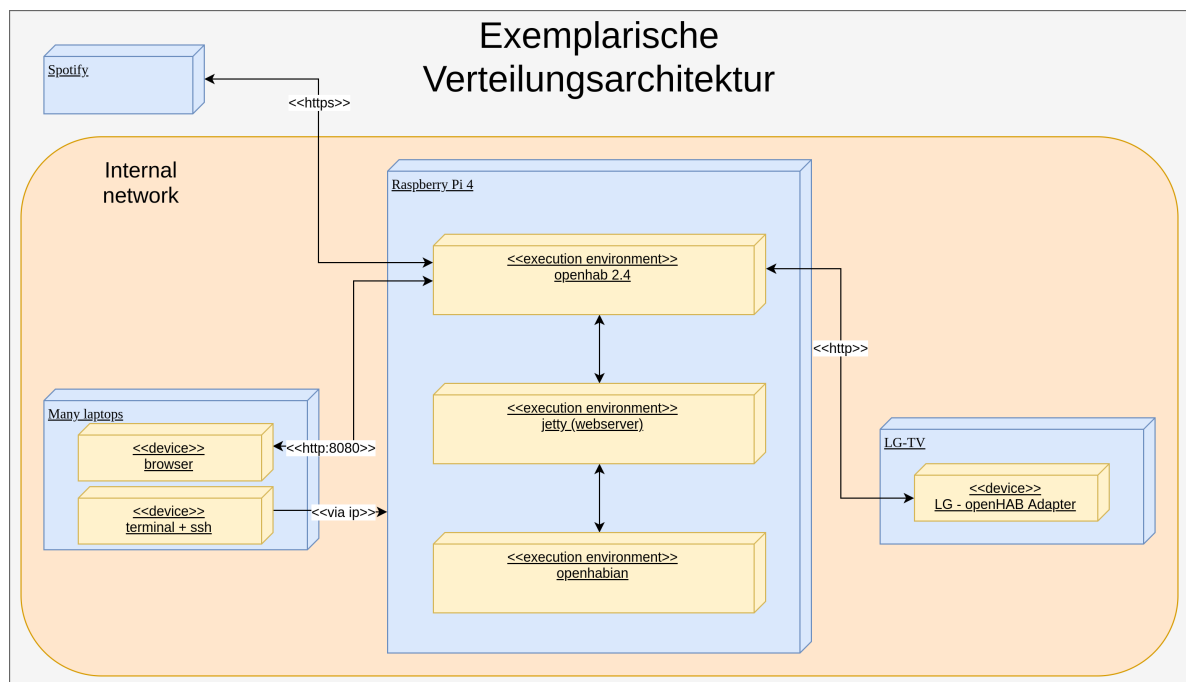


Abbildung 1: Verteilungsarchitektur

## 5 Fazit

### 5.1 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.

## 5.2 Schwächen

- 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 Gerätekennung 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
- Doku ist schwer verständlich erste schritte klappen leicht, aber danach wird es schwierig

## 6 Infos:

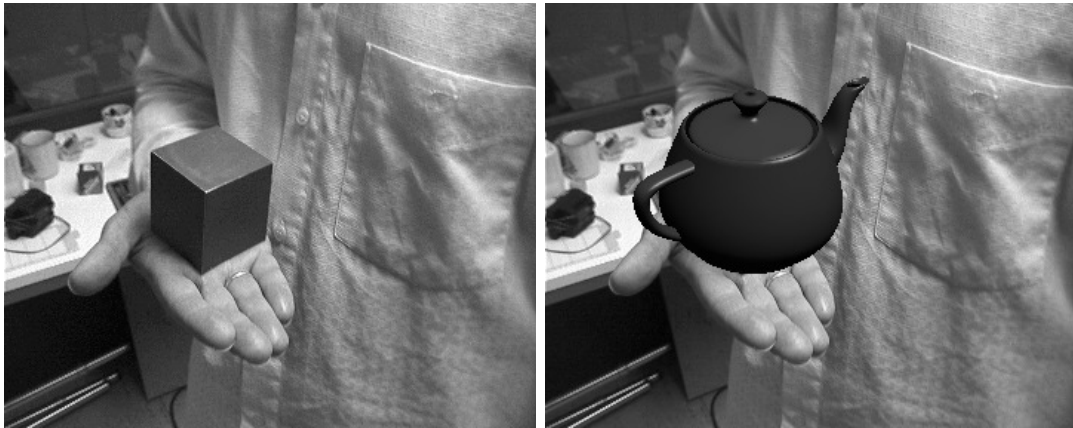
**Ausgangslage** Untersuchen Sie die Architektur und Features von OpenHAB und schreiben Sie ein Beispielanwendung. Mit myOpenHub existiert eine kostenlose Plattform die sie nutzen können.

**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>



(a) Originalbild

(b) erweitertes Bild

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

## 7 Vorlage mit Samples

Codebeispiel:

```

1 public class JavaExample
2 {
3     public static void main(String[] args)
4     {
5         int test = 5;
6         int result = 10;
7
8         result += test;
9     }
10 }

```

Codebeispiel 1: Java Beispiel

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.

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

Tabelle 2: Datenselektion für verschiedene Testdatensätze.



## **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.