Technische Hochschule Rosenheim

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 Bewertung OpenHAB **4**

4 Datenintegriertät und Sicherheit **6**

5 OpenHAB aus technischer Sicht **6**
5.1 Add-ons 9
5.2 Bindings 10
5.3 Things 10
5.4 Channels 11
5.5 Items 11
5.6 Rules 11
5.7 Sitemaps 12
5.8 Api 12

6 Verwendung von OpenHAB **12**
6.1 Integration der Big Player 12
6.2 Beispiel Aufbau eines OpenHAB Smart-Homes 13
6.3 Umgang mit OpenHAB 15

7 Fazit **16**
7.1 Stärken 16
7.2 Schwächen 16

8 Infos: **16**

A Erster Abschnitt des Anhangs **17**

1 Motivation

Die Ausgaben für das Internet der Dinge (IoT) wird weltweit, laut Statista, bis zum Jahr 2022 auf 1000 Milliarden US-Dollar steigen. Im Vergleich zum Jahr 2019 bedeutet dies, eine Steigerung von über 40%.[Ten] Bei einem solch starken Trend in der Informatik-Branche sollten sowohl Studenten, als auch Professoren dessen Grundlagen kennen. Deshalb ist im Rahmen des Faches Softwarearchitektur an der technischen Hochschule Rosenheim diese Ausarbeitung geschrieben worden. Das Ziel ist es OpenHAB, ein Heimautomatisierungs-Tool, aus praktischer und technischer Sicht zu untersuchen. Außerdem wird dabei auf die Aspekte Markttauglichkeit und Benutzbarkeit in der Praxis eingegangen.

2 Was ist OpenHAB?

OpenHAB ist eine technologie-unabhängige Open-Source-Automatisierungssoftware für Smart-Homes. Das Projekt wurde von Kai Kreuzer 2010 erstmals initiiert und wird mittlerweile durch die Community weiterentwickelt. Die Software ist hauptsächlich in Java und der Auszeichnungssprache XML geschrieben. Seit dem 16. Dezember 2019 ist die Version 2.5 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 weißt auf die vielen verschiedenen Automatisierungsmöglichkeiten hin, die OpenHAB zu bieten hat. Dabei werden Auslöser (in englisch: Trigger), Aktionen, Skripte und auch Voice-Kontrolle genannt.

3 Bewertung OpenHAB

In diesem Kapitel wird das open-source Projekt OpenHAB untersucht und bewertet. Das Projekt OpenHAB befindet sich auf Github unter `https://github.com/openhab/openhab-core`. Die hierfür verwendeten Kriterien sind orientiert an der QAware Open Source Quick-Check Liste. Dies beinhaltet:

Kriterium	**Beschreibung**
Lizenz	Für welche Zwecke kann das Projekt genutzt werden?
Reifegrad	Existiert eine stabile Version > 1.0?
Support	Existiert Issue-Tracking und ein Antwortzeit auf Tickets unter 24h?
Dokumentation	Existiert? • Api-Dokumentation • Getting Started Tutorial • Aktuelle Dokumentation

Projekt Aktivität	Gibt es? • Mindestens 1 Release im letzten Jahr • Mindestens 3 aktive Contributer • Stetige Commit-Aktivität von mindestens 1x pro Monat
Bekanntheitsgrad	Vergleich von öffentlichem Interesse in Blockposts, auf Stackoverflow et cetera.

Tabelle 1: OpenHAB Projektbewertungskriterien

- `https://www.openhub.net/p/openhab`
- Was ist Openhub: Website zur Katalogisierung von open-source Softwareprojekten
- Dabei werden Daten wie Projektname, Beschreibung und Sourcecode erfasst. Basierend auf diesen Daten erstellt Open Hub eine Statistik, die es ermöglicht, Codeanalyse, Projektmitarbeiter, Aktivitäten und eine Übersicht zu erhalten. Dabei werden auch viele weitere open source Projekte miteinander verglichen um aussagekräftige Statistiken und Aussagen treffen zu können.
  - openhub bewertet das Projekt mit sehr hoher Aktivitätsrate
  - über 1.5 Millionen lines of code Hauptsächlich Java und XML
  - 31% des Codes ist dokumentiert (was soviel wie andere durchschnittliche Java-Projekte ist)
  - insgesamt 1140 Contributers und 20k commits
  - Basierend auf dem Vergleich von Commits des Vorjahres und des jetzigen Jahres, steigt das Interesse in OpenHAB
  - Im letzten Jahr waren es 343 neue Contributer. Das macht OpenHAB zu einem der größten open source-teams der Welt. Sie sind unter den top 2% von allen Projektteams auf Openhub.
- Lizenz: freie Software Lizenz EPL-2.0: ermöglicht kommerzielle und private Nutzung, Modifizierung, Weiterverbreitung. `https://www.eclipse.org/legal/epl-2.0/`
- Stable Release 2.5 > 1
- Durchschnittliche Antwortzeit in der Community > 24h? ja: wie beweisen?
- OpenHAB wurde bereits in der Bachelorarbeit von Pirmin Gersbacher vom Jahr 2017/2018 anhand von usecases untersucht und verglichen. Im Vergleich stellte sich folgende Ergebnistabelle 1[Ger] heraus.

	OpenHAB	ioBroker	Home Assistant	Node-RED
Installation	+	+	+	++
Oberfläche	+	O	O	++
Technologien	+	-	++	++
Einfachheit	O	+	++	-
Visualisierung	+	++	O	O
Erweiterbarkeit	++	++	++	++
Automation	++	++	O	O
Verbreitung	+	-	++	O

Abbildung 1: Vergleich OpenHAB und anderen Heimautomatisierungstools von 2017/2018

4 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.

5 OpenHAB aus technischer Sicht

In diesem Kapitel sind die grundlegenden Komponenten, die OpenHAB verwendet, tabellarisch dargestellt. Anschließend wird detaillierter auf die einzelnen Elemente eingegangen und wie diese miteinander in Beziehung stehen. Ein Weiterer Aspekt von OpenHAB kommt in diesem Kapitel immer wieder zum Vorschein. OpenHAB sieht nicht den einen Weg vor etwas zu installieren/einzubinden. So kann z.B. je nach vorliebe des Nutzers eine Geräte über die Web-Oberfläche integriert werden, als auch programmatisch in dafür vorgesehen Datei-ORdner als Code. Dieses Konzept findet sich immer wieder z.B. für Bindings, Things,

Items, Rules usw. OpenHAB liefert für das Entwickeln von Add-Ons Skelleton-Skripte, welche die von openHAB bentigte Struktur erzeugen. Für manuelles Hinzufügen von Things, Items, Rules usw. bietet openHAB bereits eine ORdnerstruktur. Diese ist so offensichtlich, dass es trivial ist zu entscheiden, wo welche Komponente hinzugefügt werde soll. Sollte dies dennoch unklar sein, wird der Nutzer noch weiter durch readme-dateien unterstützt. Siehe

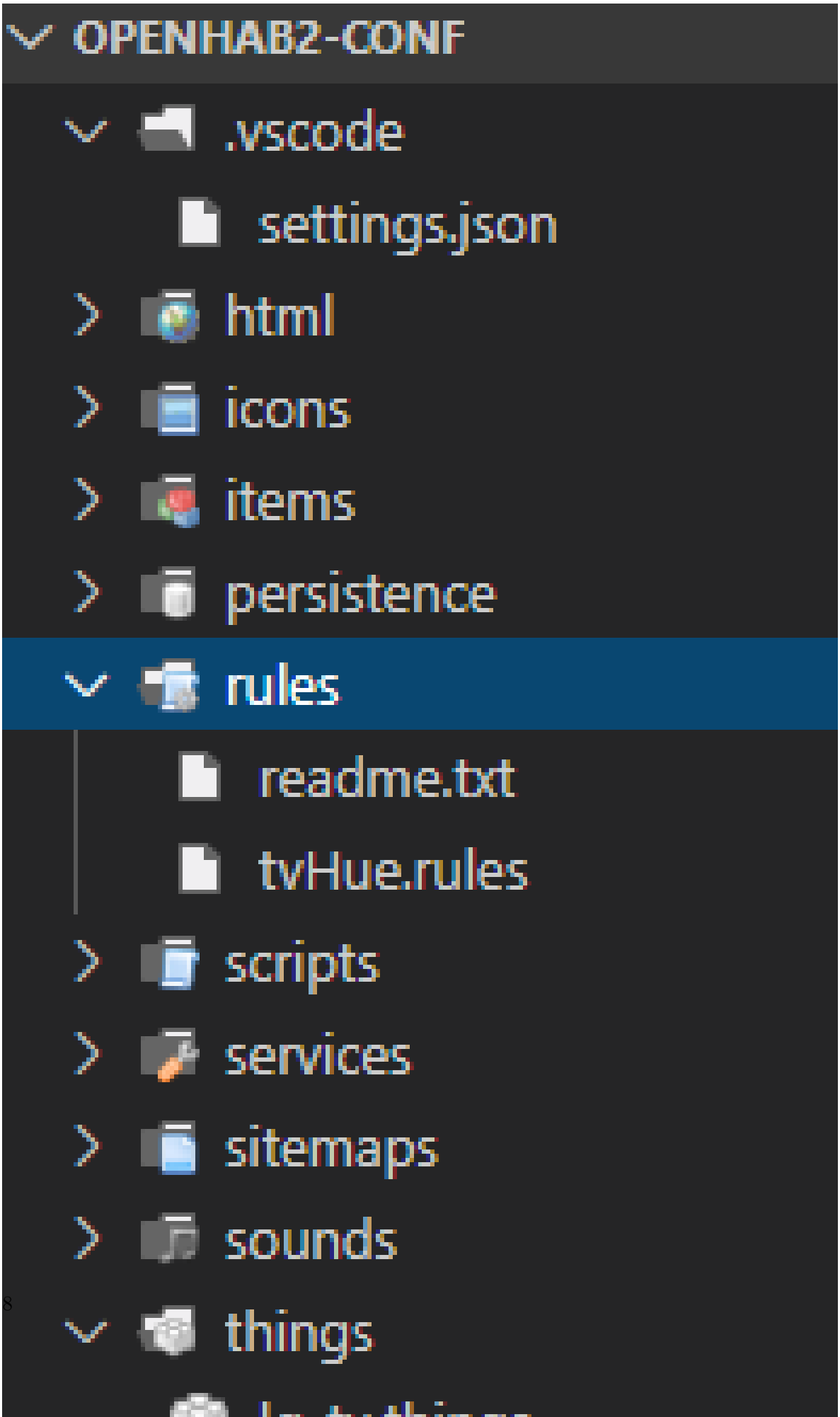
OPENHAB2-CONF
.vscode
settings.json
html
icons
items
persistence
rules
readme.txt
tvHue.rules
scripts
services
sitemaps
sounds
things

Abbildung 2: openHAB-conf Ordnerstruktur

Konzept	Beschreibung
Add-ons	Erweiterung welche von openHab ermöglicht werden.
Bindings	openHAB-Komponenten, welche die Schnittstelle zu fremd Systemen bereitstellen.
Things	erste von openHAB (Software) generierte Darstellung von Geräten.
Channels	openHAB (Software)-Verbindung zwischen „Dingen“ und „Gegenständen“.
Items	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 2: OpenHAB Komponenten

5.1 Add-ons

openHAB sich selbst als System bezeichnet welches alles Integriert, dies wird dadruch erreicht, dass es von der so konzipiert ist, dass bestimmte bereiche jederzeit erweitert werden können. OpenHAB gibt hier als verschiedene Module/Komponenten, welche neu hinzugefügt werden können vor.

- Bindings
- Automation Engine Modules
- Transformations / Profiles
- IO Services
- Persistence Services
- Audio & Voice

Um nicht den Rahmen dieser ARbeit zu übersteigen wird Detailierter auf Bindings eingegangen. Was unter anderem daran liegt, dass Die Dokumentation von openHAB, welche die entwicklung von IO-, Persistence Services und Audio/Voice mit einem TODO beschreibt eher dürftig ist. Bei den Transforamtions / Profiles handelt es sich um transformationen welche genutzt werden können um zu übertragenede Werte durch einen Channel zu manipulieren/tranformieren. Add-ons sind in openHab das Herzstück der pluggable Architecture. Da Nutzer des Systems ihre individuellen Bedürfnisse haben und sich demnach nciht alle verfügbaren Add-ons installieren müssen, sondern nur diejenigen welche für Ihre Bedürfnisse nötig sind. Sollte es allerdings keine passendes Add-on für ein gewünschtes Gerät geben, steht der erweituerng und dem entwicklen eines eigenenen Add-ons durch das Add-on System nichts im Weg. Mehr noch, durch den OpenSource gedanken, welcher openHAB vorantreibt könnte das Gesamt-System durch ein weiteres Add-on erweitert werden.

5.2 Bindings

Bindings sind die Schnittstelle bzw. die Erweiterung, welche es erst ermöglicht andere Systeme mit openHAB zu integrieren. Mit einem Binding kann sowohl ein Pysisches Geräte z.B. eine LG Fernseher als auch eine Service z.B. Spotify angebunden werden. Die Bindings sind dabei soweit abstrahiert, dass nicht jedes einzelne Model bzw. Version eine eigenes Binding benötigt. Ein Binding ermöglicht es dem System erst neue Geräte im System zu erkennen. Und diese als Thing im openHAB System zur weiteren Interaktion zur Verfügung zu stellen. Interessant ist hierbei, dass Bridges oder Schnittstellen wie Bluetooth es erst ermöglichen Weitere Geräte zu erkennen. So kann durch das Hue-Binding die Hue-Bridge integriert werden, welche es wiederum ermöglicht einzelne Komponenten des Hue-Systems anzusprechen (Lampen), welche nicht direkt erkannt und angesporchen werden konnten. Laut openHAB gibt es aktuell über 300 Bindings und dadurch können über 2000 Things angesprochen werden.

5.3 Things

Things sind die repräsentation von Geräten/Services innerhalb des Systems. Berachtet man die Objektorierntierung ist es ein sehr schönes Beispiel wie im einem Computersystem ein realer gegenstand beschreiben werden kann. Things bieten verschiedene so gennante Channels an. Diese entsprechen die Funktionen/Aktionen welche das Gerät ausführen kann. Things können auf 3 verschiedene arten openHAB hinzugefügt werden

- per Web-Oberfläche per Button click in der Inbox, nachdem das entsprechende Binding installiert wurde. Hierbei werden alle nötigen Einstellungen für das Thing automatisch getroffen. Diese können nachträglich noch bearbeitet werden z.B. Bezeichnung.

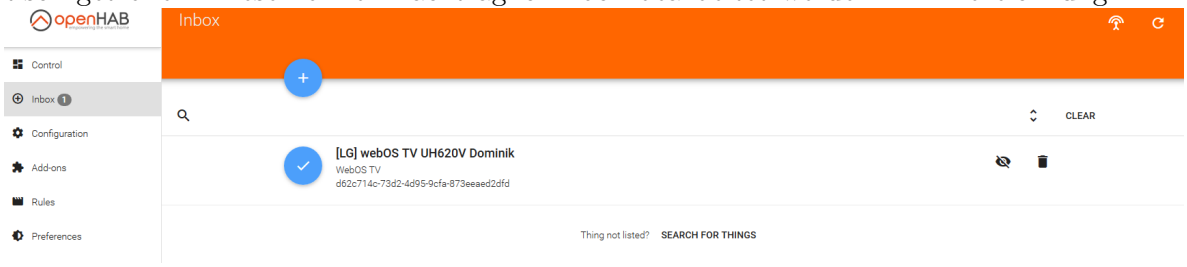

Abbildung 3: Thing per Paper-UI

- per Web-Oberfläche Manuell. Erlaubt völlige Einstellung im Vorhinein. Allerdings ist auf Statische IP-Adressen zu achten.

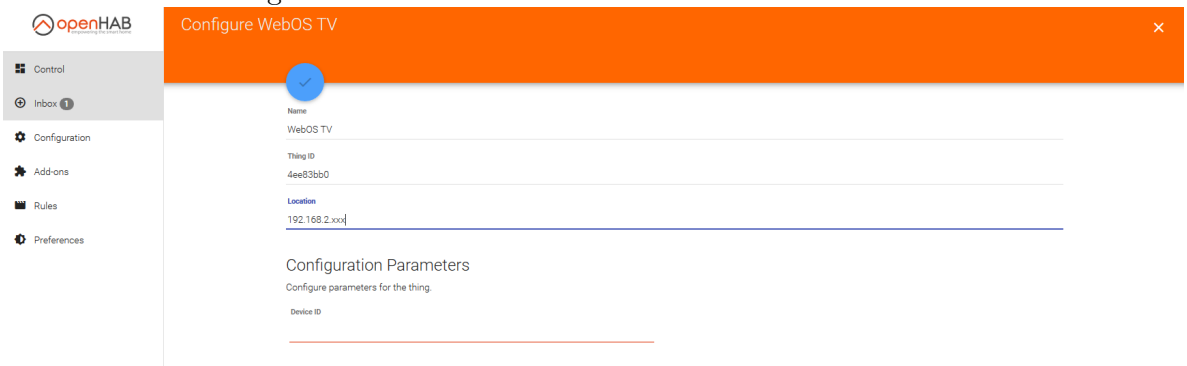

Abbildung 4: Thing per Paper-UI Manuell

- per Code. Ermöglicht und benötigte komplette manuel-

le eingabe aller nötiger Infomrmationen wie IP-Adresse

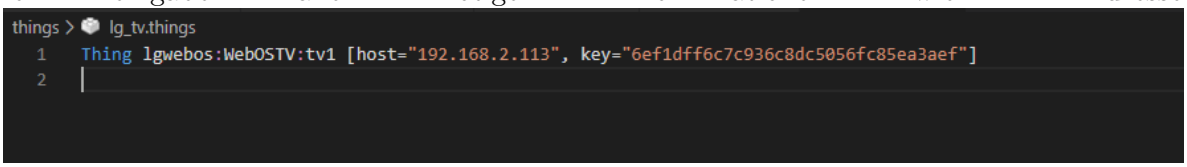

Abbildung 5: Thing per Code

5.4 Channels

sind Kommunikationswege welche von den Things angeboten werden. Channels verbinden Things und Items. Und könne in openHab, bzw. den Rules genutzt werden. Sie stellen die Verbindung von externen Gerät zu openHAB aktionen dar. Über die Channels können Informationen durch PAramerter ausgetauscht werden. So wäre z.B. die Helligkeit einer Lampe ein Parameter, welcher durch einen Channel übertragen werden kann. nicht jedoch die Informationen, dass sich die Helligkeit ändern soll. Hierfür wird das Item benötigt, welches durch den Channel an das Thing verbunden wrude.

5.5 Items

Der Begriff Item wirkt unter umständen falsch. Ein Item könnte auch als Aktion beschreiben werden. Ein Item selbst hat sehr wenig Informationen. Das Item besteht neben eine Titel noch aus eine Kategorie und eine Gruppe um es zuzuordnen und um gleichzeitig mehrere Items gleichzeitig anzusteueren.. Außerdem hat ein Item noch einen Typ, dieser Typ entspricht einer Menge von Basistypen, welche von openHAB angeboten werden. Diese sind unter anderem String, Numnber, DateTime, Location, Player usw. `https://www.openhab.org/docs/configuration/items.html#type`. Wichtig ist auch der Status. Wodoruch es doch wieder ein Item anstatt einer Aktion ist. (könnte man item als object bezeichnen mit label, kategorie, status und nur einer funktion??? memo an mich DAM) Items sind essentielle bestandteile um das Panel zu bauen, da die Widgets über Items mit den Geräten agieren.

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

Codebeispiel 1: Item-Gruppierung Beispiel

5.6 Rules

- Rules stelln 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-Bedinung und einem then Abschnitt.

- Name dient als Zuordnung
- When ist der Trigger bzgw. 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" +
               LGWebOSTVUH620VDominik_Volume.state.toString)
6  if ( LGWebOSTVUH620VDominik_Volume.state >= 20 ) {
7          HueWhiteLamp2_Brightness.sendCommand(80)
8  }
9  else {
10         HueWhiteLamp2_Brightness.sendCommand(5)
11 }
12 end
```

Codebeispiel 2: Beispiele Rule Beispiel

5.7 Sitemaps

5.8 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

6 Verwendung von OpenHAB

6.1 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/`

6.2 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.

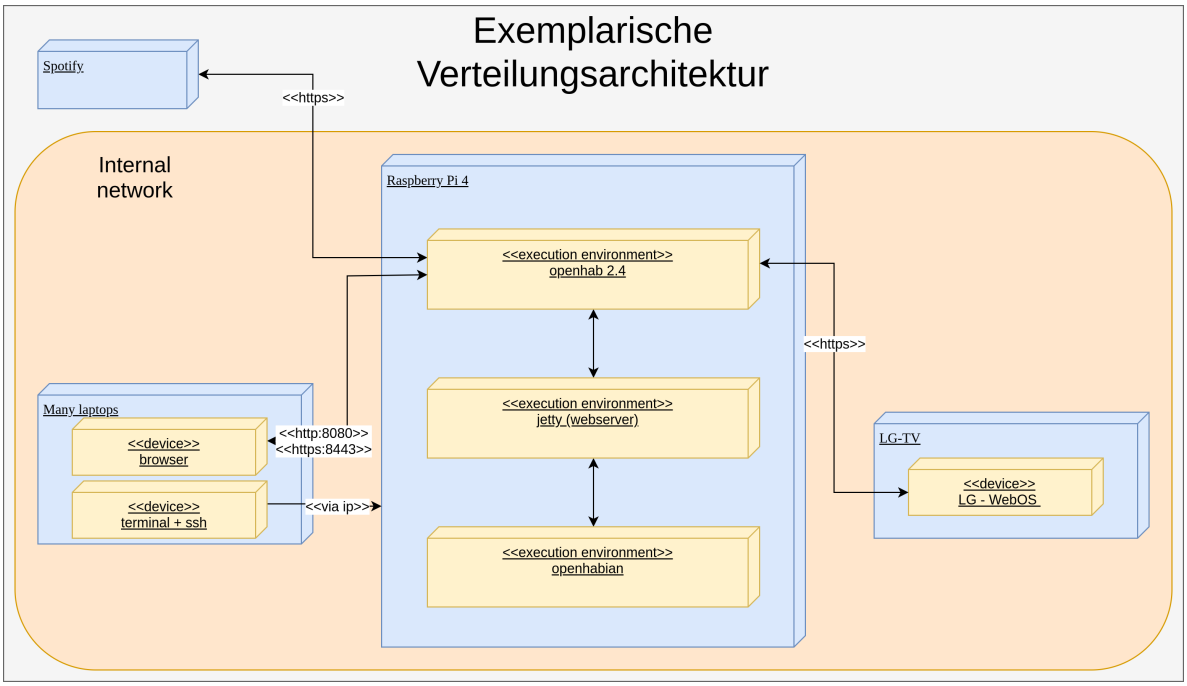

Abbildung 6: Verteilungsarchitektur

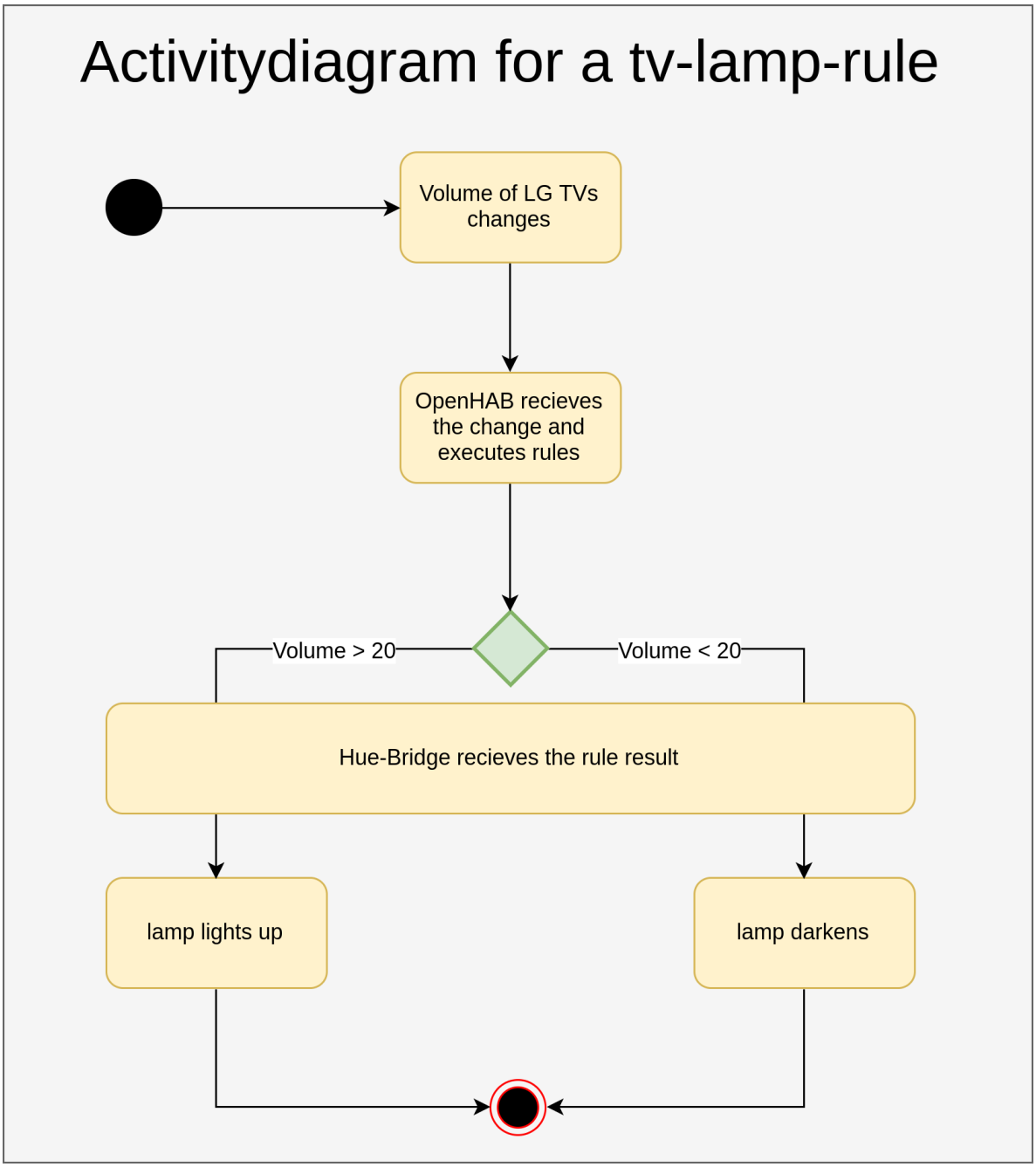

Abbildung 7: Aktivitätsdiagram für eine Rule

6.3 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

7 Fazit

7.1 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.

7.2 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

8 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 Erster Abschnitt des Anhangs

In diesem Anhang wird ...

Literatur

[Ger] P. Gersbacher. Untersuchung und Vergleich von Open Source Plattformen für das Smart Home . https://opus.hs-offenburg.de/frontdoor/deliver/index/docId/2805/file/Abschlussarbeit_P_Gersbacher_178004.pdf. Last visit: 21 Dez 2019.

[Ten] F. Tenzer. Prognose zu den Ausgaben für das Internet der Dinge (IoT) weltweit in den Jahren 2018 bis 2022. https://de.statista.com/statistik/daten/studie/537226/umfrage/prognose-zu-den-ausgaben-fuer-das-internet-der-dinge/. Last visit: 23 Dez 2019.