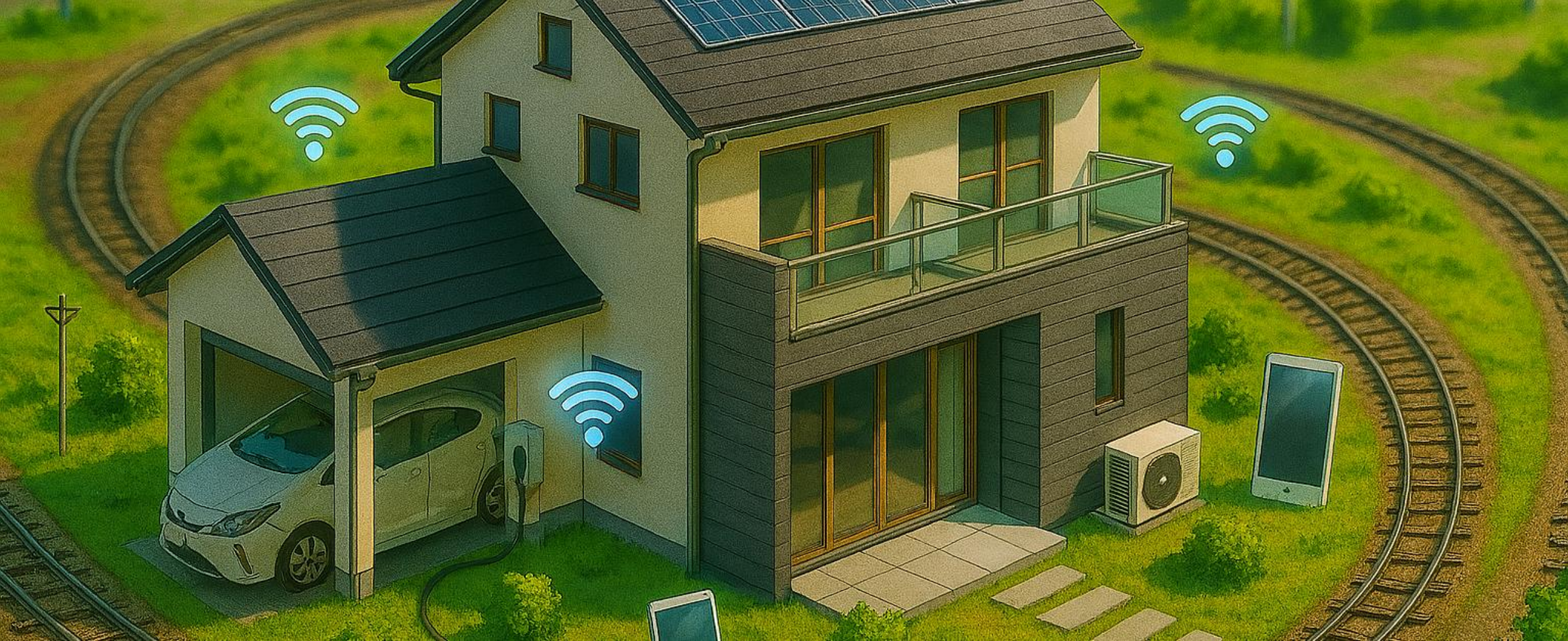




# Smart Home für Anfänger

Viele Spielereien für Zuhause mit openHAB

# ? Wer bin ich und weshalb dieser Vortrag von mir?



- Philipp Schneider
- Seit 2011 bei der Firma tracetronic GmbH in Dresden
- Als Softwareentwickler angefangen und dann in Product-Owner Rolle hineingewachsen



- Wenn dem Entwicklerherz dann die Entwicklung fehlt, dann passiert Folgendes ...

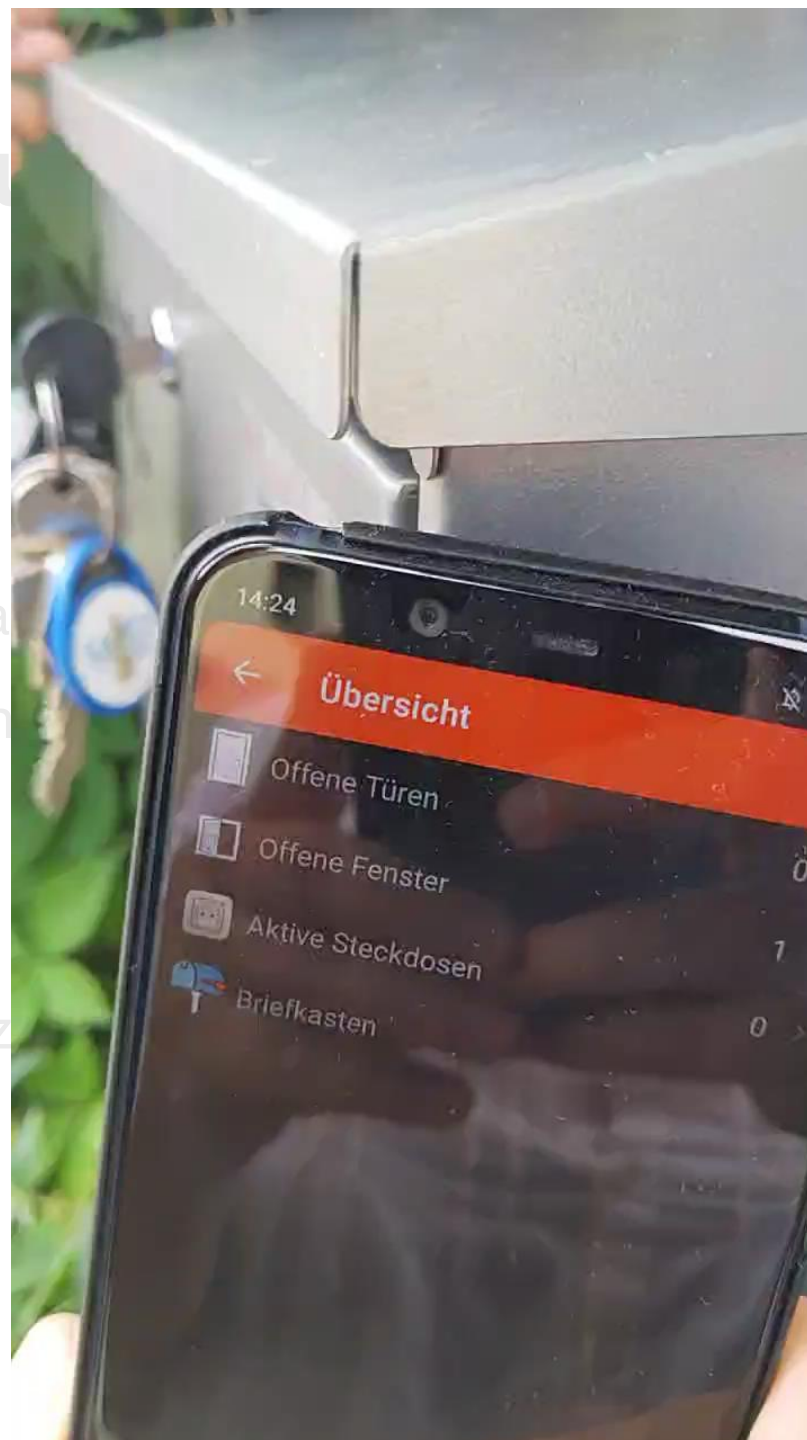
# ? Wer bin ich u... er Vortrag von mir?



- Philipp Schneider
- Seit 2011 bei der Firma tra
- Als Softwareentwickler an



- Wenn dem Entwicklerherz

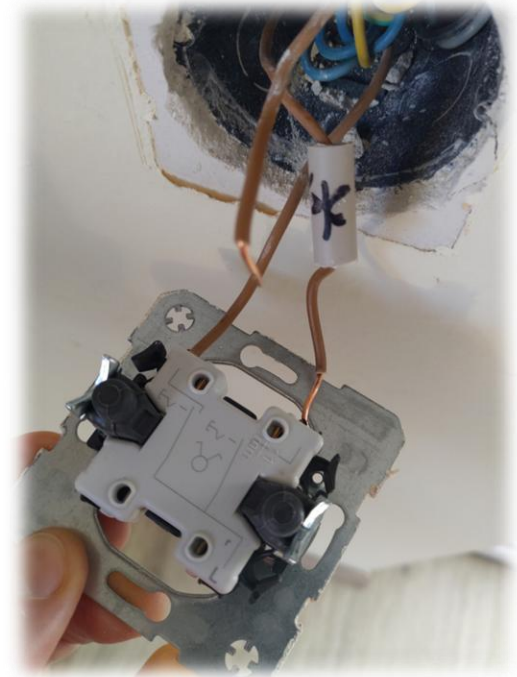


owner Rolle hineingewachsen

passiert Folgendes ...

# ! Disclaimer

- 📁 Wie eine Modelleisenbahn kann auch dieses Hobby schnell ins Geld gehen 🤖
- 🛠️ Mein Chef, der Elektrotechniker, zu mir:
  - „Wenn Informatiker mit mehr als Niedervolt-Sachen spielen, mache ich mir Sorgen. 😊“
  - ⚡ Er hat Recht: Wenn man keine Expertise hat, muss man sich fachmännische Hilfe holen!
- 🧑
  - Bei Mietwohnung sollten Änderungen vorab geklärt werden
  - Ggf. Hausratversicherung anpassen
  - Es ist auf sichere Netzwerkzugriffe und sichere Passwörter zu achten



# Agenda

1. Was ist openHAB?
2. Vorteile von openHAB
3. Installation / Inbetriebnahme
4. Architektur & Konzepte
5. Praxisbeispiele
6. Zusammenfassung
7. Ausblick





# Was ist openHAB?

- openHAB kurz für **open Home Automation Bus** – zentrale Steuerung für Gebäude
- openHAB wurde von Kai Kreuzer **2010** initiiert
- **Open-Source-Softwareplattform** zur Gebäude- und Heimautomatisierung
- Unterstützt **über 400 Anbindungen** und Geräte (z. B. Homematic, Hue, KNX, Zigbee, ...)
- Technologie- und hersteller**unabhängig**
- openHAB und die Entwicklung werden von der **openHAB Foundation**, einer eingetragenen gemeinnützigen Organisation, getragen.



# Vorteile von openHAB

- Datenschutzfreundlich: **Keine Cloud erforderlich** – alle Daten bleiben lokal, sofern gewünscht.



# Kurze Recherche

WEITERBETRIEB VERURSACHT WEITERE KOSTEN

## Devolos macht Smart-Home-System zum Großteil unbrauchbar

Wer Devolos Smart-Home-Geräte auch nächstes Jahr noch vollwertig nutzen will, muss mit zusätzlichen Kosten rechnen.

21. August 2025 um 14:14 Uhr / Ingo Pakalski

SPEXOR

## Bosch macht Alarmanlage für Camper zu Elektroschrott

Die Server von Spexor werden abgeschaltet. Laut Bosch lässt sich das einmal mehr als 200 Euro teure Gerät bald gar nicht mehr nutzen.

26. Mai 2025 um 12:15 Uhr / Oliver Nickel

ABSCHALTUNG WEMO-CLOUD

## Belkin macht Smart-Home-Geräte für viele zu Elektroschrott

Nur wer die passende Infrastruktur besitzt, kann die Smart-Home-Geräte von Belkin auch noch ab Februar 2026 verwenden.

11. Juli 2025 um 13:45 Uhr / Ingo Pakalski

# Vorteile von openHAB

- Datenschutzfreundlich: **Keine Cloud erforderlich** – alle Daten bleiben lokal, sofern gewünscht.
- **Zentrale Steuerung**: Verbindet verschiedenste Smart-Home-Geräte und -Systeme in einer einzigen Plattform  schauen wir uns dann gleich alles Live mal an 
- **Flexibles Regelwerk**: Als Nutzer kann ich komplexe Automatisierungen mit zeit- und ereignisbasierten Auslösern, Skripten und Aktionen erstellen.
- **Plattformunabhängig**: Läuft auf Windows, macOS, Linux, Raspberry Pi, Docker, Synology, ...
- **Sprachsteuerung** & Cloud-Integration: Unterstützt Google Assistant, Alexa, Apple HomeKit...
- **Einheitliche Benutzeroberfläche**: Zugriff über Webbrowser, Android- und iOS-Apps.

# Installation / Inbetriebnahme openHABian - Hassle-free openHAB Setup

## Install Raspberry Pi OS using Raspberry Pi Imager

Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi.

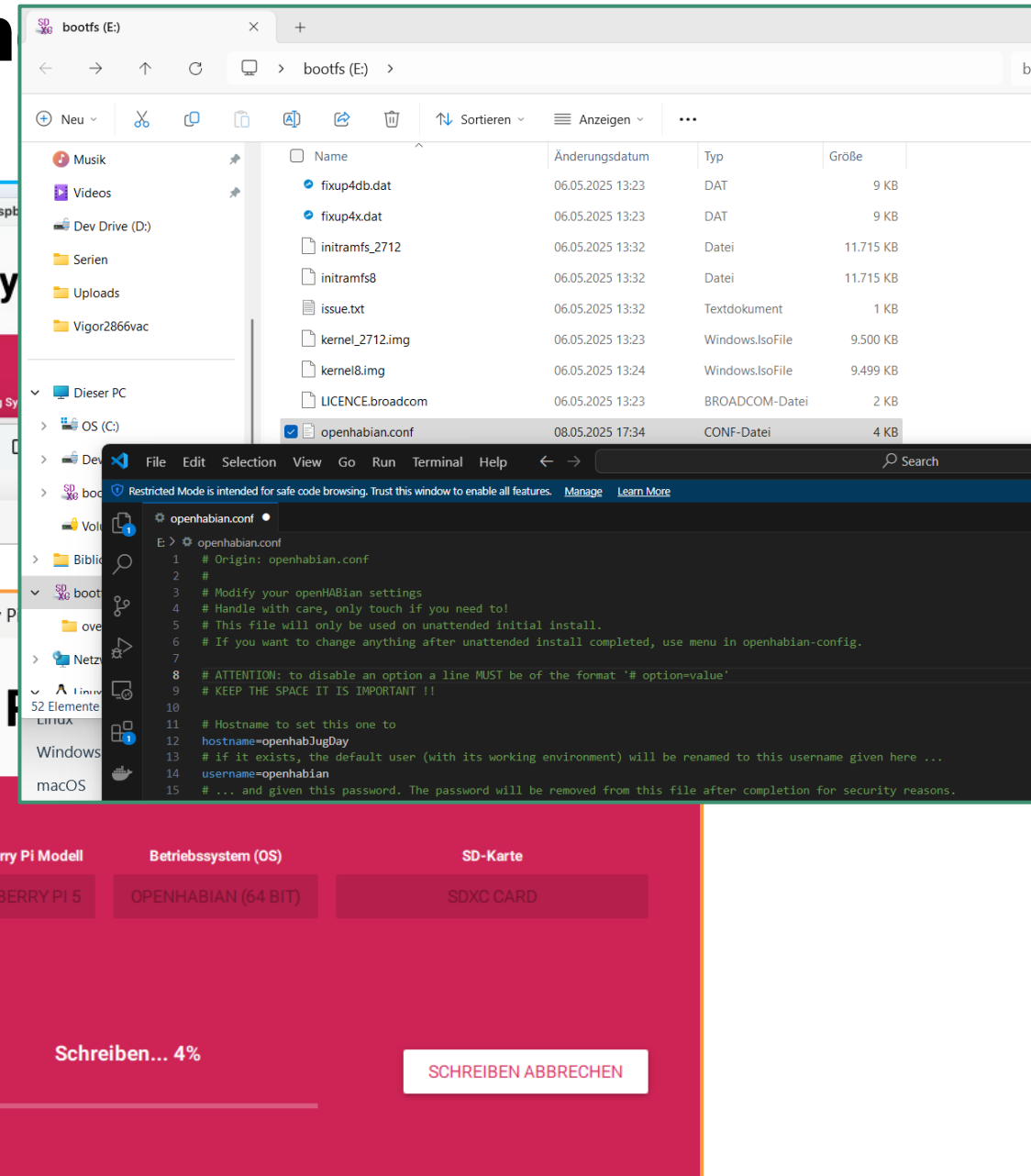
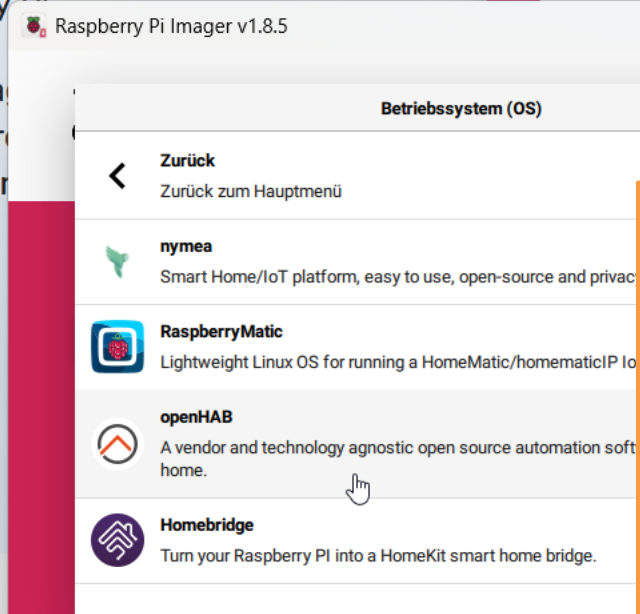
Download and install Raspberry Pi Imager with an SD card reader. Put the SD card in your Raspberry Pi into the reader and use Imager.

[Download for Windows](#)

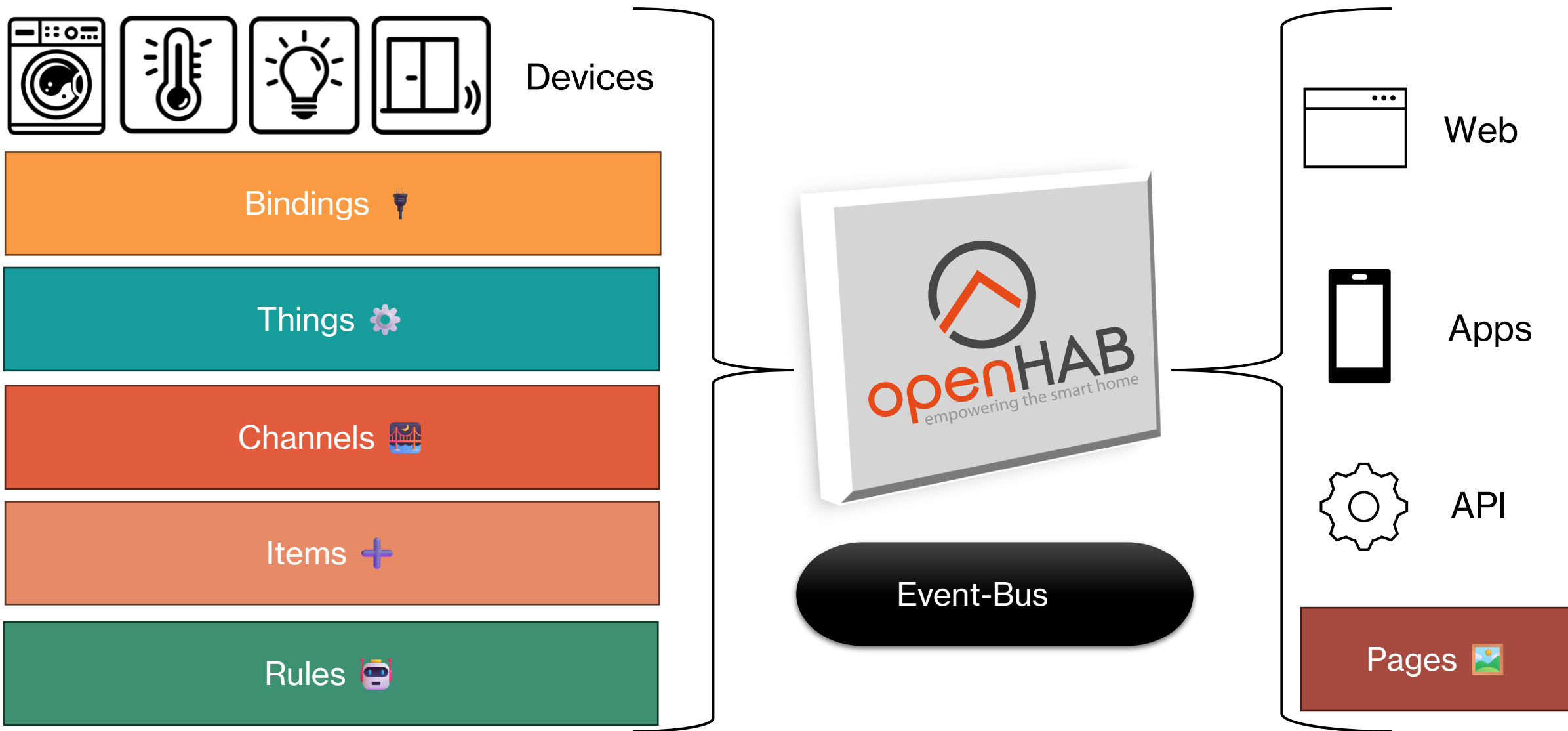
[Download for macOS](#)

[Download for Ubuntu for x86](#)

To install on **Raspberry Pi OS**, type  
`sudo apt install rpi-imager`  
in a Terminal window.

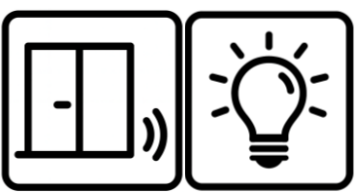


# Architektur & Konzepte





# Konzept



Bindings

Things

Channels

Items

Rules

Pages

Homematic

Schalt-Mess-Steckdose

Schalter

Flurlicht

Steckdose

Verbrauchsmessung

Flurverbrauch

Flurlicht anschalten

Verbrauch

Türkontakt

Status

Haustür

Eingangstür



astro

Sonnenstand

Sonnen-aufgang

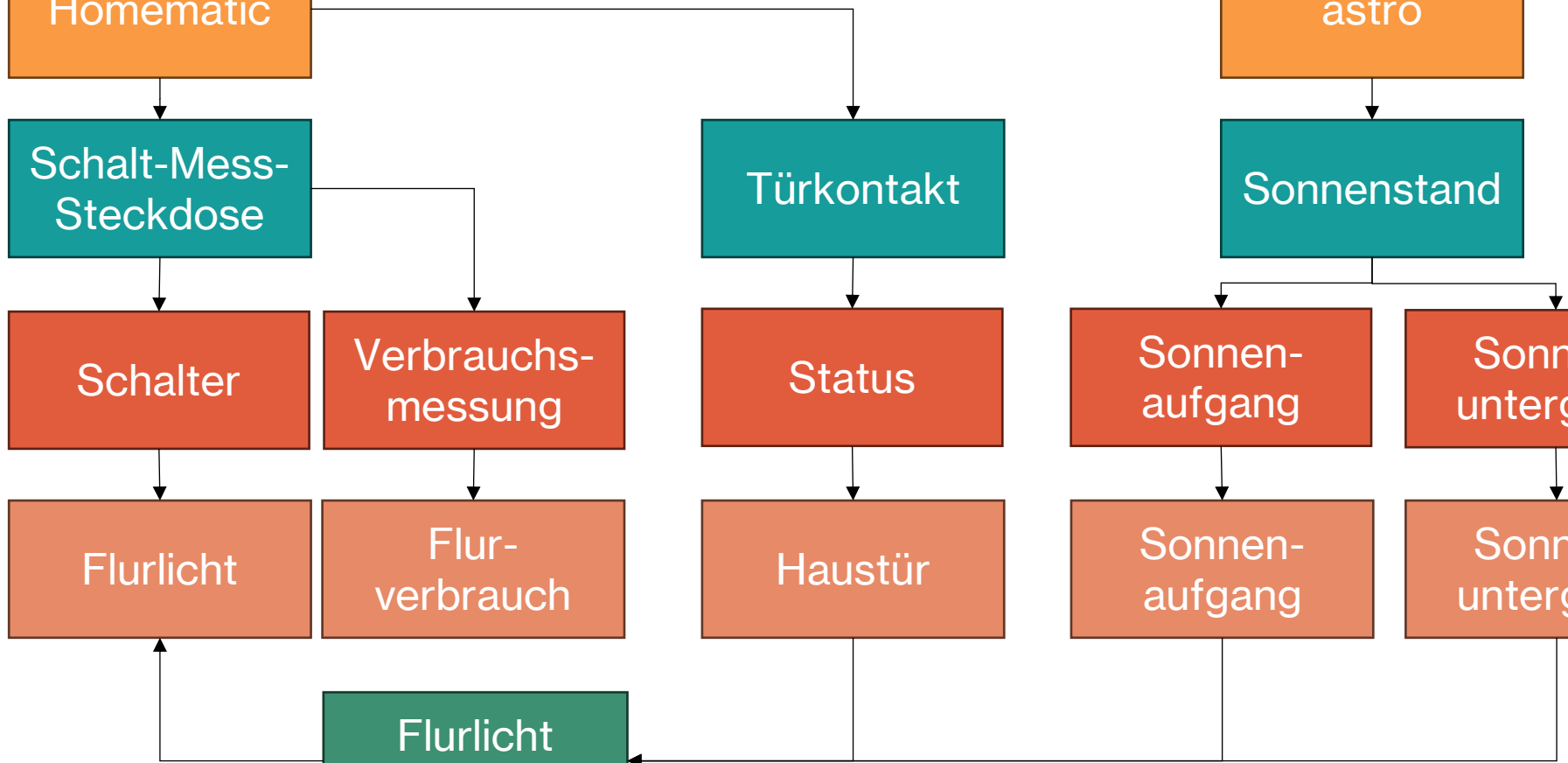
Sonnen-aufgang

Sonnen-aufgang

Sonnen-untergang

Sonnen-untergang

Sonnen-untergang





# Konzept - UI-driven Configuration vs. File-based



empowering the smart home

[< Back](#)

Keine Seiten

Administration

- Einstellungen
- Add-on Store**
- Bindings
- Automation
- Transformations
- Persistence
- User Interfaces
- System Integrations
- Voice & Speech

Entwickler Tools

Über & Hilfe

**Add-on Store**

Filters

**Installed Bindings**

 **Astro Binding**  
openHAB 

REMOVE

 **MQTT Binding**  
openHAB 

REMOVE

 **Homematic Binding**  
openHAB 

REMOVE

 **Hue Binding**  
openHAB 

REMOVE

**Installed Automation**

 **JavaScript Scripting**  
openHAB 

REMOVE

**Installed Persistence**

 **RRD4J Persistence**  
openHAB 

REMOVE

**Installed User Interfaces**

 **Basic UI**  
openHAB 

REMOVE



# Konzept - File-based vs. UI-driven Configuration

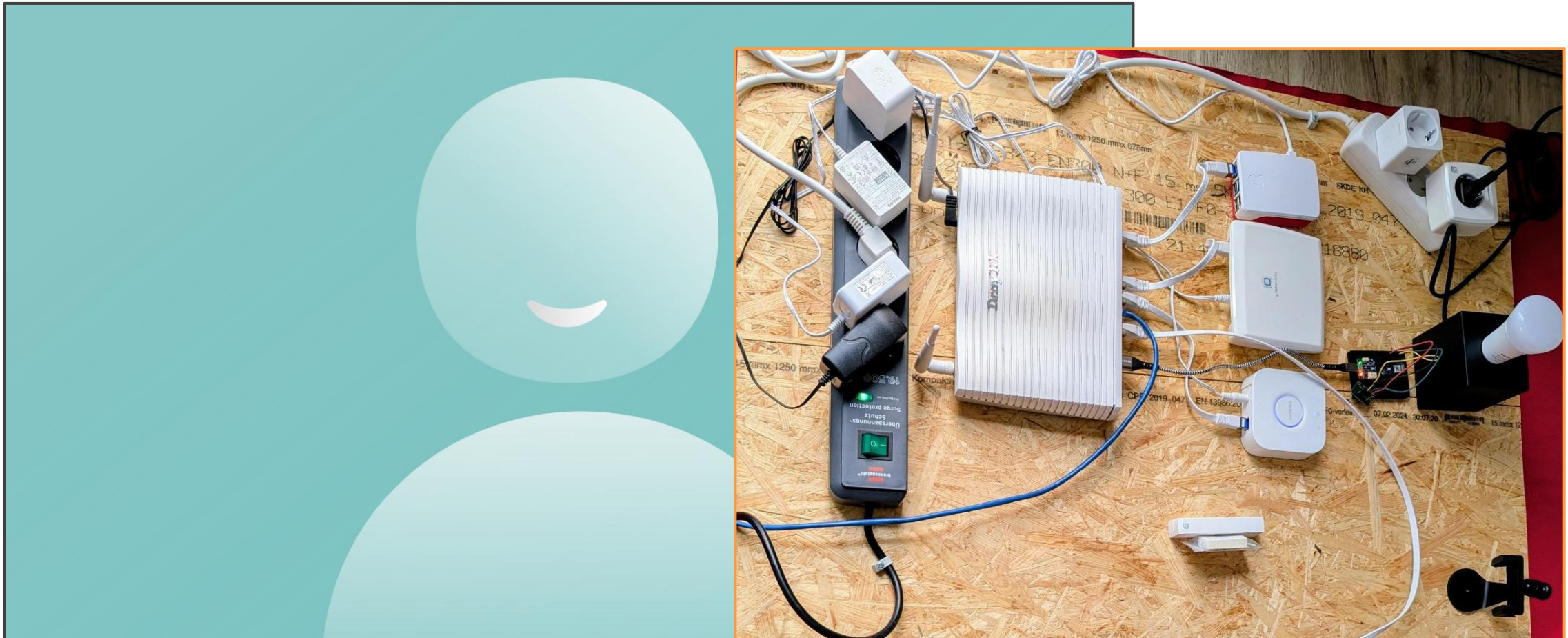
The screenshot displays the openHAB configuration interface. On the left, the 'EXPLORER' sidebar shows the project structure under 'OPENHAB-CONF (WORKSPACE)'. The 'services' folder is expanded, and 'addons.cfg' is selected. The main editor shows the contents of 'addons.cfg', which is a file-based configuration. The configuration includes comments and settings for various services, bindings, persistence, transformation, and UIs. Several lines are highlighted with red boxes to illustrate file-based configuration:

- Line 30: `# A comma-separated list of bindings to install (e.g. "binding = knx,sonos,zwave")`
- Line 31: `binding = homematic,mqtt,hue,astro`
- Line 37: `persistence = rrd4j`
- Line 43: `ui = basic`

The background features a large, semi-transparent 'git' logo and a red diamond-shaped icon with a black branching diagram, symbolizing version control and configuration management.

# Praxisbeispiele

- openHABian, Astro, Homematic, Philips Hue und Arduino via MQTT





# Praxisbeispiele - Astro-Binding anlegen

<https://www.openhab.org/addons/bindings/astro/>

Bindings

astro

openHAB-conf\services\addons.cfg  
Binding hinzufügen

Things

Sonnenstand

openHAB-conf\things\astro.things  
Thing anlegen

Channels

Sonnen-  
aufgang

Sonnen-  
untergang

openHAB-conf\items\astro.items

Items

Sonnen-  
aufgang

Sonnen-  
untergang

Sonnenstand-Items anlegen

Pages

Sonnen-  
aufgang

Sonnen-  
untergang

Sonnenaufgang und -untergang  
in einer Sitemap (Basic UI) anzeigen



# Praxisbeispiele - Homematic

Rules

✓ Homematic



Steckdose



Türkontakt



Schalter



Status



Flurlicht



Haustür



Flurlicht  
anschalten



Steckdose



Tür

## Unsere erste „Rule“:

- **Beim Tür Öffnen das Licht automatisch anschalten**
- Prinzip:
  - einfachsten Form: Wenn „dies“ passiert, führe „das“ aus
  - Eigene DSL für die Erstellung von Regeln

```
rule "Beim Haustür öffnen das Licht einschalten"
when
    Item Homematic_Door changed to OPEN
then
    Homematic_Switch.sendCommand(ON)

    // Flurlicht nach xyz automatisch wieder ausschalten
    createTimer(now.plusMinutes(1), [ |
        Homematic_Switch.sendCommand(OFF)
    ])
end
```



# Praxisbeispiele – Homematic + Astro

Rules



## 2. Rule: Beim Tür Öffnen das Licht automatisch anschalten aber nur nach

openHAB-conf > rules > door\_astro.rules

```
1 rule "Beim Haustür öffnen das Licht nur nach Sonnenuntergang einschalten"
2 when
3     Item Homematic_Door changed to OPEN
4 then
5     var timeNow = now()
6
7     if(timeNow.isAfter((Sunset_Time.state as DateTimeType).getZonedDateTime()) ||
8         timeNow.isBefore((Sunrise_Time.state as DateTimeType).getZonedDateTime())){
9
10        Homematic_Switch.sendCommand(ON)
11
12        // Flurlicht nach xyz automatisch wieder ausschalten
13        createTimer(now.plusMinutes(1), [ |
14            Homematic_Switch.sendCommand(OFF)
15        ])
16    } else {
17        logInfo("door_astro.rules", "Licht nicht eingeschaltet, da es noch Tag ist.")
18    }
19 end
```



# Praxisbeispiele – Philips Hue + Gruppierungen

Items +

Wie kann ich meinen Flur in einer anderen Farbe erstrahlen lassen?

- Hue-Lampe als Item anlegen:

```
Color Hue_Lamp "Flurlampe" <colorpicker>
```

- Hue-Item in Sitemap aufnehmen:

```
sitemap hue label="Hue Flurfarbspiel" {  
  
  Frame label="Sonnenstand" {  
    Text item=Sunrise_Time  
    Text item=Sunset_Time  
  }  
  
  Frame label="Flur" {  
    Switch item=Homematic_Switch  
    Switch item=Homematic_Door  
    Text item=Homematic_Switch_Meas  
    Switch item=Hue Switch  
    Colorpicker item=Hue_Lamp  
  
    Group item=Overview_Poweroutlets  
    Switch item=Overview_Poweroutlets  
  }  
}
```



# Praxisbeispiele – Philips Hue + Gruppierungen

Rules

- Lichtfarbe beim Öffnen der Tür ändern:

```
openHAB-conf > rules > ≡ doors_hue.rules
```

```
1 rule "Beim Haustür öffnen das Licht einschalten und grün leuchten lassen"
2 when
3     Item Homematic_Door changed to OPEN
4 then
5     Homematic_Switch.sendCommand(ON)
6
7     // Kurze Verzögerung, damit die Lampe auch wirklich an ist, wenn die Farbe gesetzt wird
8     createTimer(now.plusSeconds(6), [ |
9         var DecimalType hue = new DecimalType(125)    // 0-360; 0=red, 120=green, 240=blue, 360=red
10        var PercentType sat = new PercentType(90)      // 0-100
11        var PercentType bright = new PercentType(80)   // 0-100
12
13        // Farbton, Sättigung, Helligkeit
14        var HSBType hsb = new HSBType(hue, sat, bright)
15
16        Hue_Lamp.sendCommand(hsb)
17    ])
18
19 end
```



# Praxisbeispiele – Philips Hue + Gruppierungen

Items +

- Wenn ich mehr als zwei Steckdosen habe, wie kann ich alle auf einmal prüfen und deaktivieren?
- Eine „Group“ als Item anlegen:

```
openHAB-conf > items > ≡ systemvars.items
1 | Group:Switch:OR(ON,OFF) Overview_Poweroutlets "Alle Steckdosen [%d]" <poweroutlet>
```

- Diese erstellte Gruppe den „Steckdosen“-Items zuweisen:

```
Switch Hue_Switch "Flursteckdose" <poweroutlet> (Overview_Poweroutlets) {ch
```

```
Switch Homematic_Switch "Steckdosenlicht" <poweroutlet> (Overview_Poweroutlets)
```



# Praxisbeispiele – Philips Hue + Gruppierungen

Pages

- Die neue „Group“ Overview\_Poweroutlets als Item in die Sitemap übernehmen:

```
openHAB-conf > sitemaps > hue.sitemap
1  sitemap hue label="Hue Flurfarbspiel" {
2
3      Frame label="Sonnenstand" {
4          Text item=Sunrise_Time
5          Text item=Sunset_Time
6      }
7
8      Frame label="Flur" {
9          Switch item=Homematic_Switch
10         Switch item=Homematic_Door
11         Text item=Homematic_Switch_Meas
12         Switch item=Hue_Switch
13         Colorpicker item=Hue_Lamp
14
15         Group item=Overview_Poweroutlets
16         Switch item=Overview_Poweroutlets
17     }
18 }
19
```

Flur

- Steckdosenlicht ☐
- Haustür ☐
- Steckdose-Netzfrequenz 50.01 Hz
- Flursteckdose ☐
- Flurlampe
- Alle Steckdosen 0 >
- Alle Steckdosen 0 ☐



# Praxisbeispiele – Arduino / MQTT + Persistence



Wenn ich Sensoren habe, wie kann ich mir diese Werte über den Zeitverlauf anschauen?

Arduino sendet

 Temperatur- und  
 Luftfeuchtigkeitsdaten  
via **MQTT-Topics** an einen Broker

```
259 void loop() {
260
261     if (bme1Detected){
262         float temp = bme1.readTempC();
263         float humidity = bme1.readHumidity();
264
265         Serial.print(F("Temperatur [°C]:\t"));
266         Serial.println(temp);
267         Serial.print(F("Luftfeuchtigkeit [%]:\t"));
268         Serial.println(humidity);
269
270         mqttClient.loop();
271
272         if (!mqttClient.connected()) {
273             connectToMqtt();
274         }
275
276         mqttClient.publish("WeatherStation/Temp", String(temp,2), false, 2);
277         mqttClient.publish("WeatherStation/Hum", String(humidity,2), false, 2);
278     }
279
280     Serial.println();
281     delay(5000);
282 }
```



# Praxisbeispiele – Arduino / MQTT + Persistence



Items +

Channels 

- MQTT-Broker als Thing + Channels (**MQTT-Topics**) einbinden:

```
openHAB-conf > things > ≡ weathermqtt.things
1  Bridge mqtt:broker:mosquitto [ host="localhost", secure=false, username=" ", password=" " ]
2
3  Thing mqtt:topic:mosquitto:Weather (mqtt:broker:mosquitto) {
4      Channels:
5          Type number : temperature "Temperature" [ stateTopic="WeatherStation/Temp" ]
6          Type number : humidity "Humidity" [ stateTopic="WeatherStation/Hum" ]
7  }
```

- Items anlegen:

```
openHAB-conf > items > ≡ mqtt.items > ⚙ WeatherStationHum
1  Number:Temperature    WeatherStationTemp "Temperatur im Flur [%.1f °C]"    <temperature>    { channel="mqtt:topic:mosquitto:Weather:temperature" }
2  Number:Dimensionless  WeatherStationHum  "Luftfeuchtigkeit im Flur [%.0f %]"    <humidity>       { unit="%", channel="mqtt:topic:mosquitto:Weather:humidity" }
```



# Praxisbeispiele – Arduino / MQTT + Persistence



- Persistence-Strategie für (numerische) Items festlegen → rrd4j.persist

```
openHAB-conf > persistence > ≡ rrd4j.persist > ...
```

```
1 Strategies {
2     everyMinute : "0 * * * * ?"
3     default = everyChange
4 }
5
6 Items {
7     WeatherStationTemp : strategy = everyUpdate, everyMinute, restoreOnStartup
8     WeatherStationHum : strategy = everyUpdate, everyMinute, restoreOnStartup
9 }
```

```
openHAB-conf > items > ≡ mqtt.items > ⚙ WeatherStationHum
```

```
1 Number:Temperature WeatherStationTemp "Temperatur im Flur [%.1f °C]" <temperature> { channel="mqtt:topic:mosquitto:Weather:temperature" }
2 Number:Dimensionless WeatherStationHum "Luftfeuchtigkeit im Flur [%.0f %]" <humidity> { unit="%", channel="mqtt:topic:mosquitto:Weather:humidity" }
```



# Praxisbeispiele – Arduino / MQTT + Persistence

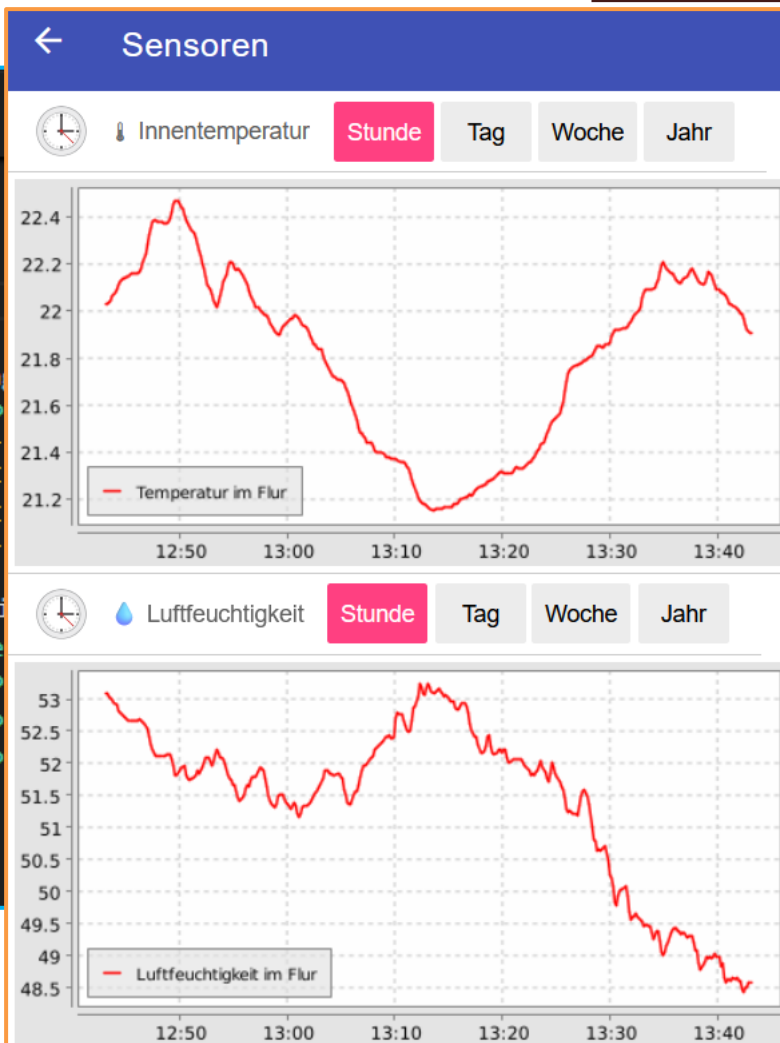


Pages



- Sensordaten als Diagramm (Chart) in Basic UI einbinden

```
openHAB-conf > sitemaps > ≡ arduino.sitemap > 🔑 arduino
1  sitemap arduino label="Unser Smart Home Flur" {
18      Default item=WeatherStationTemp
19      Default item=WeatherStationHum
20
21      Text label="Sensoren" icon="line" {
22
23          Switch item=PeriodWeatherChoice label="🕒 Innentemperatur" icon="time" mapping=
24          Chart item=WeatherStationTemp legend=true period=h refresh=7200 visibility=[P
25          Chart item=WeatherStationTemp legend=true period=D refresh=14400 visibility=[P
26          Chart item=WeatherStationTemp legend=true period=W refresh=14400 visibility=[P
27          Chart item=WeatherStationTemp legend=true period=Y refresh=14400 visibility=[P
28
29          Switch item=PeriodWeatherChoice label="💧 Luftfeuchtigkeit" icon="time" mapping=
30          Chart item=WeatherStationHum legend=true period=h refresh=7200 visibility=[Pe
31          Chart item=WeatherStationHum legend=true period=D refresh=14400 visibility=[P
32          Chart item=WeatherStationHum legend=true period=W refresh=14400 visibility=[P
33          Chart item=WeatherStationHum legend=true period=Y refresh=14400 visibility=[P
34      }
35  }
36 }
```

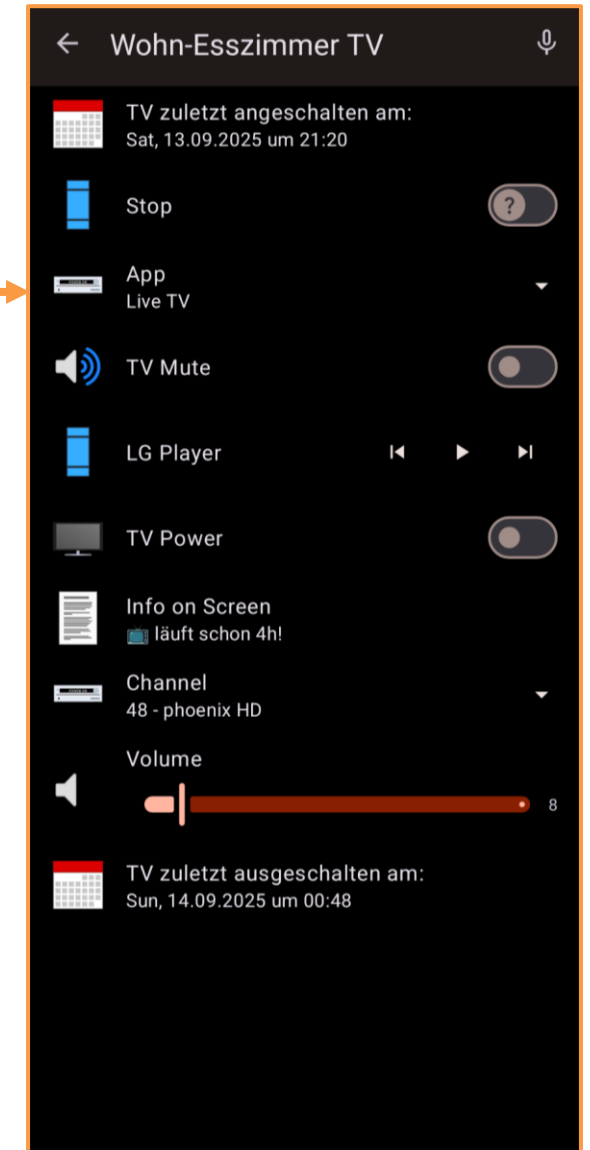
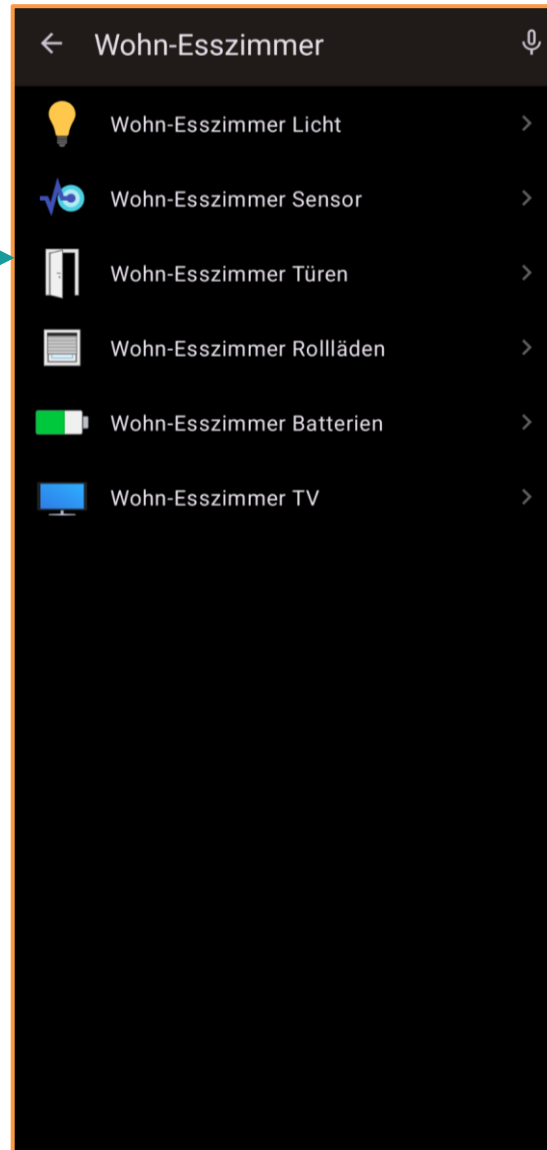
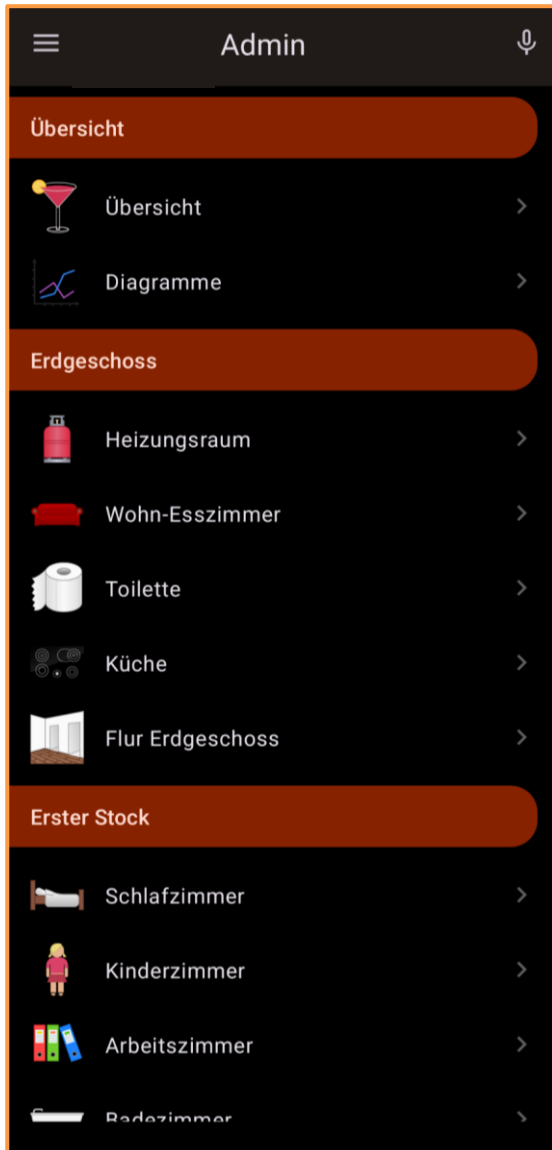




# Praxisbeispiele -



# Basic UI auf allen Geräten

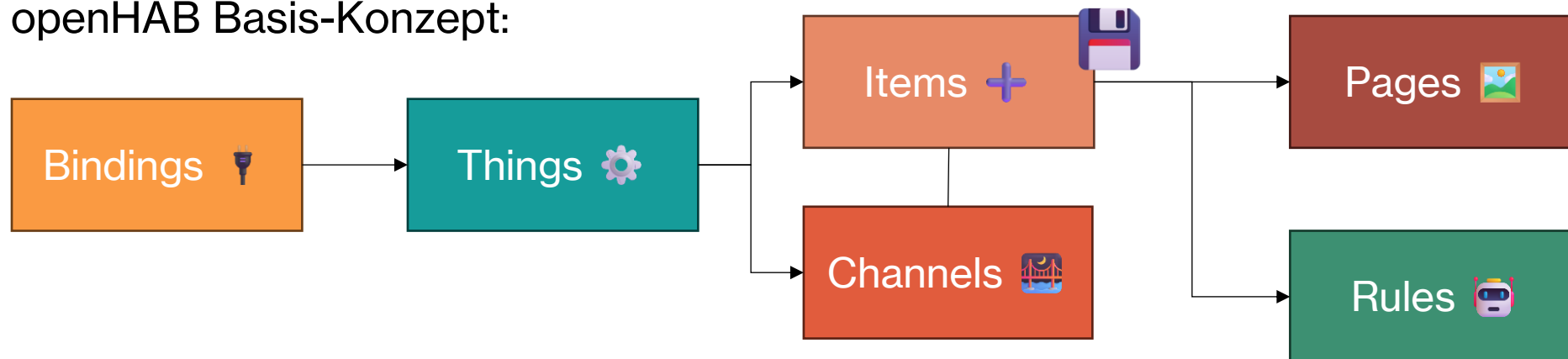




# Zusammenfassung



- **openHAB** eine Technologie- & herstellerunabhängig **Open-Source-Softwareplattform**
- **Einfache Installation** mit „openHABian – Hassle-free openHAB Setup“ auf dem Raspberry Pi
- openHAB Basis-Konzept:



- Git freundliche „**File-based**“ Konfiguration
- Basic **UI** läuft auf allen Geräten
- Viele Spielmöglichkeiten mit den **über 400 gut dokumentierten Bindings** und über „**3000 Things**“ zum **Ansteuern**



# Ausblick



- openHAB Cloud Connector – um von unterwegs auf sein Zuhause zugreifen zu können  
<https://www.openhab.org/addons/integrations/openhabcloud/>  
<https://www.myopenhab.org/>
- Selbst mitmachen und ein Binding entwickeln:  
<https://www.openhab.org/docs/developer/>
- Uis mit Semantic Model oder Dashboards erstellen mit z. B. HABPanel  
<https://www.openhab.org/docs/ui/habpanel/habpanel.html>
- LoRaWAN (Long Range Wide Area Network) ermöglicht mit Hilfe von MQTT die Einbindung weit entfernter Sensoren  
<https://www.thethingsnetwork.org/>
- ... es gibt noch soviel mehr zu entdecken 😊  
<https://demo.openhab.org/overview>

# Quellen



[https://upload.wikimedia.org/wikipedia/commons/thumb/0/07/OpenHAB\\_logo\\_2.svg/1200px-OpenHAB\\_logo\\_2.svg.png](https://upload.wikimedia.org/wikipedia/commons/thumb/0/07/OpenHAB_logo_2.svg/1200px-OpenHAB_logo_2.svg.png)



<https://upload.wikimedia.org/wikipedia/commons/thumb/e/e0/Git-logo.svg/768px-Git-logo.svg.png?20160811101906>