Table of contents

Preface

The rewards are high!

Over the last decades, technology has evolved so much that it allows us to provide affordable home automation that is more applied to humans. The IKEA TRADFRI product line and the Philips Hue lights reflect affordable home automation that can be seen as a commodity product today. The smart assistants, on the other hand, are an entirely new range of innovative products with a human-like interface that brings the connected home closer to its residents.

Manufacturers of household appliances are releasing more and more connected versions of their products. Startups are developing new innovative 'connected' products that do not yet exist. Even suppliers you wouldn't expect, such as Rituals, active in home & body cosmetics, release a connected version of their perfume diffuser with their Genie 2.0.

Most of these 'connected' products have a good out-of-the-box user experience, are easy to install and bring some added value to our lives. What is unfortunate is that they are isolated solutions, which do not have the openness and integration possibilities to talk to each other. At best, vendors offer or refer to an IFTTT integration [1] (if-this-than-else) to link things together, but to be honest, you will quickly reach the limits of this approach. On the other hand, there is the open-source community that offers solutions such as Openhab, Hassio, Zigbee2mqtt and Homebridge to name a few and offer a lot of functionality to build your own connected home.

This is the playground for the technical skilleld people. The internet is crowded with DIY smart home implementations brewed by people making use, or even contribute to the open-source community projects. Many people invest their time in it for free, they do it for the honour and are proud when someone uses one of their contributions to the community. In most cases these products are superior to commercial solutions on many levels, such as the available features, the speed to support new products, the openness and (of course) the price. You can't beat a free meal!

The downside is that it is complicated for non-technically skilled people to work with these platforms. It is a rather long learning curve and requires a lot of patience. The reality of this DIY approach is that you will spend a lot of time looking for answers and hoping that someone has solved the problem for you. It is slow, often requires a lot of research, but the reward is high!

[1] https://ifttt.com/home

This was my motivation to write this ‘cookbook’. I try to avoid spending time reinventing the wheel. Giving an overview of what is available, free-of-use, providing ‘ready-to-use’ building plans using feature-rich open platforms where less technical skilled people would be able to successfully build a solution within a reasonable amount of time.

I hope you will enjoy it.

Peter Leemans

December 2020

Welcome to the connected house 2.0

The (non)sense of home automation

Home automation systems have been around for decades and many people, including myself, are fascinated by them. The first commercial home automation systems originated in the 1980s. At the time, the idea that all your devices could be controlled via one central hub sounded as science fiction. Some of those systems were based on the X10 protocol, which still exists today. The Pico Electronics X-10 Powerhouse is an example of that. It communicated with lights via a 120kHz signal burst sent through the home's power lines. Although technically it worked, it had its limitations. Because the signals did not stop at the doorstep, they could penetrate your neighbour's home and interfere with a similar system. Therefore, hacking would have been child's play, as the data on the power line was not encrypted. The focus was on managing lights and other appliances using a wired remote control or an MS-DOS program on your IBM computer.

X10 controllers: A simple controller (bottom left), a radio controller (top center), and an original controller (bottom right)

Next, there was HAL 2000 (Home Automated Living), adding voice control using a Windows95 computer. You could manage your home with a microphone like today's smart speakers avant-la-lettre. The technology was not at the same level of current smart speakers and generated many unwanted commands if you had a conversation in front of HAL 2000 with someone else.

HAL2000
Home Automated Living
Set the thermostat to 68 degrees.
I have set the thermostat to 68 degrees.

Hal 2000: Adding voice control using a Windows 95 computer

In the same period, you had specialized companies such as the Belgian Teletask who were pioneers in building home automation systems. They focused more on the high-end residential and professional building market. Contrary to the X10 powerline and HAL2000 DIY systems, the Teletask system used a worldwide network of certified distributors and system integrators. These systems were expensive and not flexible. Although technology shortcomings were to blame, lack of focus on customer needs by system designers was the primary shortcoming at that time.

Most of the installed home automation systems did not live up their potential. In many cases, only one person – a professional or 'dad' – had knowledge of the system. Other family members didn't know how to utilize and take advantage of the system.

Gradually, the industry began to understand its limitations and the technical evolution created new possibilities for home automation, such as wireless communication, miniaturization, the exponential growth in computing power with upcoming cloud services.

While the technical evolution was essential to the progression of home automation, the change towards a more consumer-centric thinking made the technology accessible to everyone. The idea was not so much to automate everything, but more about to help people connect to their houses and their world. This was achieved by offering people exactly what they want – whether that is valuable insights, money-saving opportunities or just convenience – and not getting carried away with futuristic utopias.

Because the connected home industry reinvented itself, it was able to create an entirely new market, one with endless potential. Comprised of many different players, it aims to offer a great user experience while making people's lives easier, safer, and more comfortable. Vendors of these new 'connected' products for the home focus on ease-of-installation, avoiding the aid of a professional to install them.

The connected house 2.0 is about this new wave of connected products that assist in creating your connected home. With the help of IoT & Cloud Services and by taking advantage of open (source) platforms, it brings value and comfort to the residents.

The '2.0' refers to the wave of 'connected' products that emerge on the market every day. According to Statica Research, the total addressable market for smart home products and technology will reach $53 billion by 2022[1]. From intelligent kitchen appliances to voice assistants, technology companies and startups are leveraging the Internet of Things (IoT) to connect everyday devices to the cloud and create new experiences for customers at a rapid pace.

This book aims to show how easy it is for a tech novice to embrace a number of these connected products and integrate them by using a best-of-breath approach.

The era of the-Internet-of-Things

During the last decade, the Internet of Things caused a rapid growth in connected products, One of the first real examples of the IoT originated from the early 1980s, when IT-students of the Carnegie Mellon University hooked up a Coca-Cola machine to the Internet to check if drinks were available and cold. However, IoT as a concept was only officially named in 1999 by the British technology pioneer Kevin Ashton in the United States.

Today, IoT includes billions of devices that collect and share data with one another. While some devices connect via wired and wireless networks, others do so over intranets and the Internet. Small cameras, sensors, monitors, and meters observe, measure, and report their physical surroundings. They detect motion, temperature, light and sound levels, energy usage, and air quality. They can track human activities, including health, mobility, eye movement, mood, location, paths of movement through space, and stress levels.

But despite all those connected devices already out there, IoT is still in his infancy.

The reason for this is the diversity of technologies that constitute IoT. Not only is it difficult to build great

[1] https://www.statista.com/statistics/682204/global-smart-home-market-size/

connected products, it is even more demanding for existing businesses to transform into a digital world and find the right connected products for their business, also referred to as products & services.

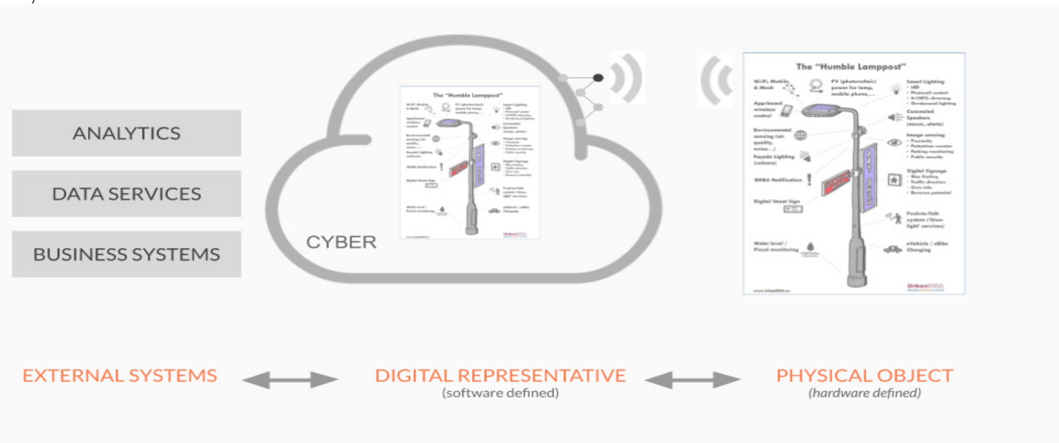
While most organizations today make decisions based on static, point-in-time data, all of that is about to change. Thanks to rapidly evolving hardware and software, we are seeing integrated solutions that promise real-time insights. Gradually, IoT is entering our homes, with smart speakers acting as home assistants as an example. Still, many objects are not connected yet, leaving a lot of opportunities unexplored at this moment.

The Internet of Things is the era where physical assets are connected to each other to feed sensor data and allow us to make actionable insights. Suddenly the IT world comes very close to our real world, where it is applied to 'simulate' human-like decisions.

It might take some time, but eventually, most 'things' and objects will become connected. With the data they produce, we will be able to receive some actionable insights. The world of the Internet-of-Things is growing. IoT is not the privilege of the tech-savvy, but for anyone who is curious to discover this domain and its endless possibilities

What is a connected Product?

To illustrate what a connected product is, let us look at the "humble lamppost", an example that was introduced by Jason Warwick. [2].



On the right the physical lamppost (hardware-defined version), which consists of a dozen sensors and actuators. In the middle the digital representation of that object (software-defined version). Between them the network and communication to exchange the asset data. This is important because this is where an asset's physical state is transformed into a digital state.

For example, a temperature with a value in °C is converted into a number value with unit '°C.' On the left, we have the IT systems that interact with the digital representative. Because all asset values are digitized, it becomes easy to bundle the sensor data or send a command to an external system. The digitized values can used for various purposes. From simple monitoring to more advanced applications such as building autonomous systems, which operate without any human intervention.

Everything connected

In the early days, home automation systems tried to connect everything. Every light bulb and wall switch got wired to a central 'monolithic' brain, which needed to be programmed by a specialist. It was everything but flexible and became outdated quite quickly because of the continuous evolution of the hardware and software industry (Moore's law, if you remember). Traditional home automation systems needed careful planning during the construction of a new home and required professionals to install that particular 'home automation' system.

The traditional home automation systems were, for that reason, expensive and suffered from a high level of gadgetry. The latest connected home products can be installed in existing homes without any special skills. These new smart home technology systems do not require you to rebuild your house. They operate wirelessly, for example by connecting to a WiFi network, which by now has become a commodity network in most of our houses.

Most importantly, these new wireless products are not built on one monolithic system. Each connected product (or product line) is managed by its own system, working independently. The fact that it does not rely on one monolithic system allows you to buy just the 'connected' products that make sense and bring value to you without having to connect everything in your home.

The Ingredients

Introduction

Now that we have an idea of what a connected house stands for, let's see what we can use to build our own. This chapter highlights various platforms, starting with the solutions from dominant players such as Amazon, Google, Apple and Samsung. Next I discuss a number of open (source) platforms. This is a selective list of platforms you can use to build a cost-effective, state-of-the-art and open connected home system.

The term "connected home" might be too narrow, as our lives do not only take place at home but also at work, in our favourite pub to meet with our friends, or travelling between those locations, which we call our places-of-interest. So maybe we should call it our life assistant system, just like the role of the butler as explained in the previous chapter.

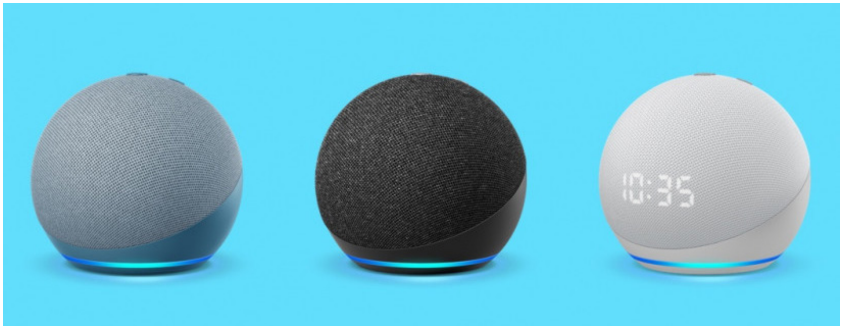
Let us start with the ecosystems of some dominant technology players and how they position themselves.

Amazon Echo

Amazon was one of the first technology players that provided a smart assistant based on the echo (DOT) hardware combined with the virtual assistant ALEXA which runs in the cloud. It is therefore not surprising that they placed their home strategy around this smart assistant.

The company has sold more than 100 million devices, making it the leading digital assistant for the smart home. Today they have different variants of the Echo product, featuring the Echo, Echo DOT, Echo show, Echo plus and Echo Spot.

Recently they announced their 4th generation of the Amazon Echo. This latest version has a built-in smart home hub that supports Zigbee devices. This latest version comes for less than 100€ which is a very competitive price including a smart assistant and a Zigbee Home HUB.



In addition to Amazon's own product line, other 'smart speaker' hardware suppliers such as Sonos, Bose and Marshall have embedded Alexa in their products.

Connected Devices

With the 4th generation of the Echo DOT a complete ecosystem of Zigbee devices is added to that portfolio of connected devices which can be used with the Amazon Alex App and it Smart Speakers.

The Alexa App supports a wide range of connected devices such as the Philips Hue product line (if you use the Hue Bridge), Fitbit Versa 3, Ecobee Smart Thermostat and Smart Camera, ...
Amazon is also investing in connected devices other than the smart assistants. in September 2019 [1], the

[1] https://www.aboutamazon.com/news/devices/amazon-devices-event-september-2019

company unveiled several proprietary consumer hardware solutions ranging from wearables including Echo Frames smart glasses, the Amazon Echo Loop, which puts Alexa on your finger and an Amazon Smart Oven, a combination microwave, convection oven, air fryer and food warmer that you can control with your voice using compatible Alexa-enabled devices.

In February 2018, Ring was acquired by Amazon for an estimated value of between $1.2 billion and $1.8 billion. Ring portfolio of products include smart doorbells and cameras. Last September Amazon announced an autonomous indoor drone that can use a map of your home to independently fly around to check out strange noises or run a patrol when you're not home. The Ring Always Home Cam will be available in 2021 for $250.

Alexa Skills

Alexa skills are essentially apps for your Echo speaker that enable Amazon's voice assistant to do everything from get specific information, play games and connect smart home devices. Whether it's turning on your Nest smart thermostat or controlling your Philips Hue lamps or your Sonos speakers, the skills enable your Amazon Echo to do just about anything. Amazon has a development programme that supports developers in creating skills to run on Alexa.

Alexa skills work across the Smart speaker product line. Whether you have the Amazon Echo, Echo Show, or even the Echo Dot, the same skills are available. Alexa smart home skills are available for all types of connected devices such as Philips Hue, Sonos, Ring, Litter thermostat/camera, Lifx, August thermostat, iRobot home,....

The Technology

The glue between a system-of-systems architecture

Now that we have dealt with different platforms, let us dive into the chapter on technology and see how these systems can interact with each other. We will look at three technologies: APIs, Webhooks and MQTT.

API's

A first technology to build a systems-of-systems architecture.

What is an API?

API stands for Application Programming Interface. In an Internet-connected world, humans use web and mobile applications, systems and applications use APIs. Websites and APIs both do the same things, like return data, content, images, video, and other information.
Web APIs are a set of rules for interacting with a web server, with the most common use case being data retrieval. APIs provide mechanisms for users to access and manipulate data stored by the API provider. The user makes a "request" to a web server, that web server accesses a database (which contains the data), and returns it to the requester in a "response".

APIs are not a specific service or tool, they are part of a system, and like the web you get something in return with every request. Instead of getting HTML back with every request, you get JSON, XML, and CSV - providing structured, machine-readable information that can be used by other systems and within other applications with very little human help.

What Are APIs used For?

While APIs are primarily used by desktop, web, mobile, and other application developers, they are also used by non-developers to work with services like IFTTT, Zapier, and the growing number of low-code or no-code solutions out there – such as Postman[1]
Every company that has a modern application out there has APIs – these APIs might not be easily found. All the big tech company names you know like Facebook, Twitter, Google, Microsoft, and others – all have APIs. Any company who uses common services like WordPress, Quickbooks, SalesForce, and other common applications, technically also have APIs.
If your website runs on WordPress, your website has an API – and if you aren't aware of it, then you

[1] https://www.postman.com/

aren't getting the opportunity to put it to use. APIs are ubiquitous, and underneath the surface of everything we do online today – you just may not have been fully aware of it until now.

As an example, let's look at OpenWeather, the API service that provides weather forecasts around the globe for any use case. It provides current weather data, forecasts and historical data to more than 2 million customers, including Fortune 500 companies and thousands of other businesses globally.

If you want to know the weather in the city of London you can execute the following API call:

http://samples.openweathermap.org/data/2.5/history/city?q=London,UK&appid=b1b15e-88fa797225412429c1c50c122a1

The API response is as follows:

```
{
  “message”: “”,
  “cod”: “200”,
  “city_id”: 2643743,
  “calctime”: 0.0875,
  “cnt”: 3,
  “list”: [
    {
      “main”: {
        “temp”: 279.946,
        “temp_min”: 279.946,
        “temp_max”: 279.946,
        “pressure”: 1016.76,
        “sea_level”: 1024.45,
        “grnd_level”: 1016.76,
        “humidity”: 100
      },
      “wind”: {
        “speed”: 4.59,
        “deg”: 163.001
      },
      “clouds”: {
        “all”: 92
      },
      “weather”: [
        {
          “id”: 500,
          “main”: “Rain”,
          “description”: “light rain”,
          “icon”: “10n”
        }
      ],
      “rain”: {
        “3h”: 2.69
      },
      “dt”: 1485717216
    }
```

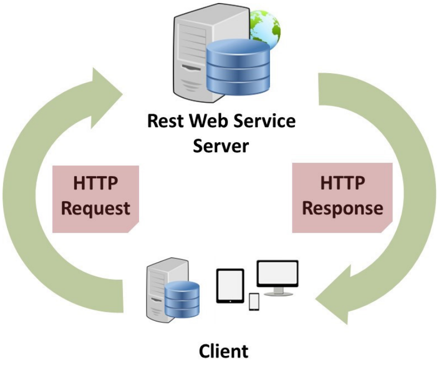
Are APIs for you?

APIs describes how data is exchanged, content is published, media are consumed and algorithms are used on the web today. APIs are how you access your social data, your photos, your accounting information and much more.

APIs are often seen as highly technical, which can be true. However, many APIs are accessible to anyone curious enough to look behind the curtain of the web. If you've ever clicked "view source" on a website, APIs are for you. If you want to know how to get Tweets into a spreadsheet so you can play with social data, APIs are for you. If you're interested in how your Nest thermostat works with your mobile phone applications, then APIs are for you.

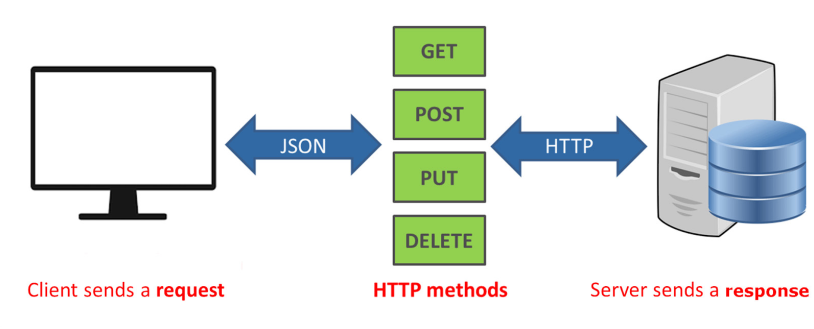
Postman is working hard to make APIs more visible and accessible to both developers and non-developers. If you want to know more about APIs, I recommend that you start by downloading the Postman application and find one or two interesting API collections in the Postman API network to start playing. You never know what you might learn along the way!

REST-API



Webservices are purpose-built web servers that support the needs of a site or other application. Client programs use Application programming interfaces (APIs) to communicate with Web services. Generally, an API exposes a range of data and functions to facilitate interaction between computer programs and enable them to exchange information. The API is the (inter)face of a web service that listens and responds to client requests.

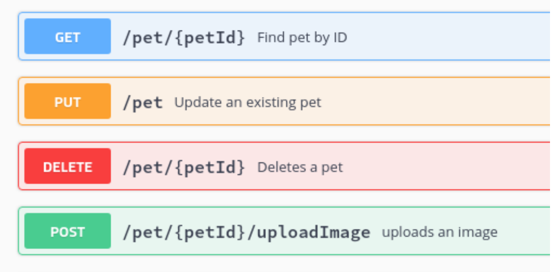
The REST architecture is commonly applied to the design of APIs for modern web services. A web API conforming to the REST architectural style is a RESTful API.



“Web resources” were first defined on the World Wide Web as documents or files identified by their URLs. However, today they have a much more generic and abstract definition that encompasses everthing or entity that can be identified, named, addressed, or handled, in any way whatsoever, on the Web. In a RESTful Web service, requests made to a resource’s URI will elicit a response with a payload formatted in HTML, XML, JSON, or some other format. The response can confirm that some alteration has been made to the stored resource, and the response can provide hypertext links to other related resources or collections of resources. When HTTP is used, as is most common, the operations (HTTP methods) available are GET, HEAD, POST, PUT, PATCH, DELETE, CONNECT, OPTIONS and TRACE

Take as an example a PET registration database web service.

With **GET /pet/{petId}**, you can retrieve the information about a pet by its ID. **PUT /pet** allows you to update an existing pet. **DELETE /pet/{petId}** will remove the pet with a given ID from the registration database. **POST /pet/{petId}/uploadImage** allows you to upload a picture for a pet with a given ID.



GET `/pet/{petId}` Find pet by ID

PUT `/pet` Update an existing pet

DELETE `/pet/{petId}` Deletes a pet

POST `/pet/{petId}/uploadImage` uploads an image

The Hardware

Now we know which platforms you can use to build your connected home based on a systems-of-systems architecture and which technology is available to integrate them, its time to look at the hardware.

Which hardware is out there we can use to build our connected home?

The Smart HUB

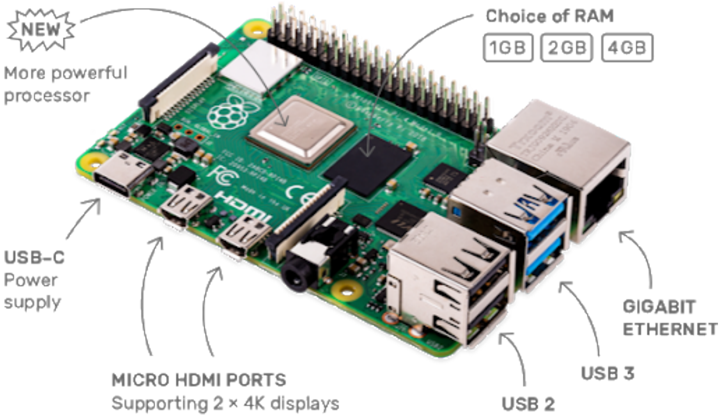
The first hardware component we are going to cover is the HUB (or gateway) which can manage the different networks and protocols. Companies such as SmartThings (acquired by Samsung), Fibaro and Philips Hue all deliver a HUB, but each of them supports only one or a limited number of networks and protocols, suited for their ecosystem. SmartThings supports Zigbee and Z-Wave [1], Fibaro focuses on Z-Wave and the Apple Homekit protocol While Philips Hue uses Zigbee to communicate with their devices.

Depending on what you want to achieve, these HUBs will all perform fine for a particular use-case, but we don't want to be constrained to one protocol or vendor and only can use their ecosystem. It would be a pity that you can't use Sonos Speakers or Ikea's Tradfri smart lights or the nice looking Fibaro Wall Plugs for that matter. You want to use them all!

The good news is, you can! As outlined in chapter 2, openHAB is the ideal software platform to build our Home HUB. Although different hardware can be used, the most popular and low-cost solution is using a single board computer such as the Raspberry Pi for our Smart HUB.

The Raspberry Pi is a single-board computer with lots of resources. It has the size of a credit card designed and manufactured in the UK with the initial intention of providing a cheap computing device for education. Since its release, however, it has grown far beyond the educational scene. The Raspberry Pi's commercial release was in February 2012 with the Raspberry Pi 1. Since then, the board has gone through several revisions. Currently, we are at revision 4. You can buy the Raspberry Pi 4 starting from 39.95€, depending on its memory size. The 1 GB version is good enough for your connected home project, but you can opt to go for more memory if you intend to run other applications aside from it.

[1] See the technology chapter for more information about Z-Wave and Zigbee.



The Raspberry Pi 4 specs:

- CPU – Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- RAM – 1GB, 2GB or 4GB LPDDR4-2400 SDRAM (depending on model)
- WiFI – 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
- Ethernet – Gigabit
- USB – 2 USB 3.0 ports; 2 USB 2.0 ports
- GPIO header – Raspberry Pi standard 40 pin
- HDMI – 2 × micro-HDMI ports (up to 4kp60 supported)
- Display port – 2-lane MIPI DSI
- Camera port – 2-lane MIPI CSI
- Audio – 4-pole stereo audio and composite video port
- Storage – Micro-SD card slot for loading operating system and data storage
- Misc – H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode), OpenGL ES 3.0 graphics
- OS – Debian Linux 10 based

Next to the Raspberry Pi, you need some additional components.

You need first of all an SD card. The SD card is required to install your operating system and software on it. In our case, we will install OpenHAB installed on it. A complete ready-to-use image is available for that purpose.

Second, you will need a Power Supply. Go for an official power supply. Most problems occur when not enough power is delivered to the board, caused by an inefficient power supply. Especially when the radio's start transmitting, the system might behave strange when using undersized or cheap power supplies.

You might also want a case to protect everything from the outside. A Keyboard and monitor is only handy during installation. Afterwards, the Raspberry Pi runs headless.

If you want to make use of Z-Wave devices, you will need a Z-Wave controller. A suitable Z-Wave controller which connects to one of the USB ports of the Raspberry Pi is the Aeon Labs Aeotec Gen 5 Z-Wave USB Interface. You can find one for less than 50€.

Aeotec Z-Wave controler model 2020

If you use a Raspberry Pi 4, make sure you order the 2020 model. This Z-Wave controller has a battery on board which allows you to unplug it from the Raspberry PI and bring it close to a new Z-Wave sensor in your home to run the discovery service. Pretty handy!

It is relatively easy to build your own (Home) HUB thanks to the excellent community software and hardware out there. It is not only an open system able to support a lot of devices, it is also fun to build your own hardware solution.

Mandatory is that the HUB can be part of a system-of-systems architecture. It will give you more flexibility as technology evolves rapidly and allow you to use new features when they get released on any of those platforms.

The first recipe further down in chapter 5 covers the installation of your Smart Home HUB. If you want, you can jump to there and try out that recipe and come back here to see what is next.

RECIPES
for the connected house

Let's get into the practical side of things. This is the **DO** part. In this chapter, you can find a collection of carefully selected recipes for the connected house based on the ingredients and technology outlined in this cookbook.

The recipes are in chronological order, but feel free to crawl through it and start executing them in the order you want. If there is a dependency on a previous recipe, this will be mentioned.

The recipes are divided into 4 categories.

The first category contains a collection of basic recipes. These are the recipes that form the basic building blocks and are typical prerequisites for other recipes.

A second category includes recipes for connecting devices. These recipes include both commercial 'off-the-shelf' devices and recipes to build your own connected devices.

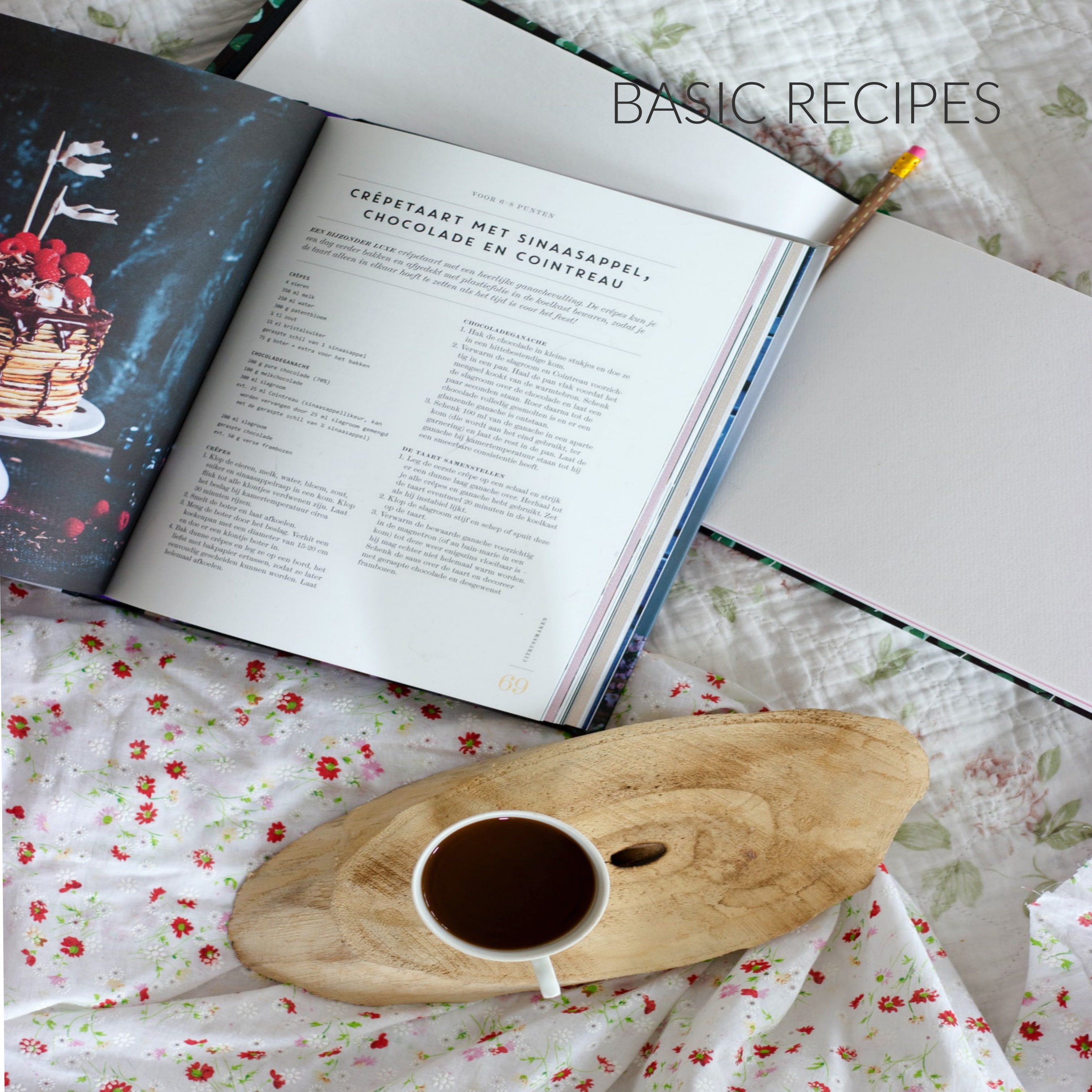
A third category includes recipes to build great user interfaces for web and mobile applications. Human interfaces, such as voice control and text-to-speech, also fall under this category.

The fourth and final category contains recipes that add automation to your home. Sample recipes include a calendar planner for your HVAC and home and away scenarios.

All recipes are accesibble on the website *theconnectedhouse.org* allow you to easily copy past commands and code. Some recipies in this cookbook include QR-codes to download Code blocks.

Let's start with the basic recipes.

BASIC RECIPES



VOOR 6-8 PUNTEN

CRÊPETAART MET SINAASAPPEL, CHOCOLADE EN COINTREAU

EEN BIJZONDER LUXE crêpetaart met een heerlijke ganachevulling. De crêpes kun je een dag eerder bakken en afgedekt met plasticfolie in de koelkast bewaren, zodat je de taart alleen in elkaar hoeft te zetten als het tijd is voor het feest!

CRÊPES
4 eieren
350 ml melk
250 ml water
300 g patentbloem
½ tl zout
1½ el kristalsuiker
geraspte schil van 1 sinaasappel
75 g boter + extra voor het bakken

CHOCOLADEGANACHE
200 g pure chocolade (70%)
100 g melkchocolade
300 ml slagroom
evt. 25 ml Cointreau (sinaasappellikeur, kan worden vervangen door 25 ml slagroom gemengd met de geraspte schil van ½ sinaasappel)

200 ml slagroom
geraspte chocolade
evt. 50 g verse frambozen

CRÊPES
1. Klop de eieren, melk, water, bloem, zout, suiker en sinaasappelrasp in een kom. Klop flink tot alle klontjes verdwenen zijn. Laat het beslag bij kamertemperatuur circa 30 minuten rijzen.
2. Smelt de boter en laat afkoelen.
3. Meng de boter door het beslag. Verhit een koekenpan met een diameter van 15-20 cm en doe er een klontje boter in.
4. Bak dunne crêpes en leg ze op een bord, het liefst met bakpapier ertussen, zodat ze later eenvoudig gescheiden kunnen worden. Laat helemaal afkoelen.

CHOCOLADEGANACHE
1. Hak de chocolade in kleine stukjes en doe ze in een hittebestendige kom.
2. Verwarm de slagroom en Cointreau voorzichtig in een pan. Haal de pan vlak voordat het mengsel kookt van de warmtebron. Schenk de slagroom over de chocolade en laat een paar seconden staan. Roer daarna tot de chocolade volledig gesmolten is en er een glanzende ganache is ontstaan.
3. Schenk 100 ml van de ganache in een aparte kom (die wordt aan het eind gebruikt, ter garnering) en laat de rest in de pan. Laat de ganache bij kamertemperatuur staan tot hij een smeerbare consistentie heeft.

DE TAART SAMENSTELLEN
1. Leg de eerste crêpe op een schaal en strijk er een dunne laag ganache over. Herhaal tot je alle crêpes en ganache hebt gebruikt. Zet de taart eventueel 20 minuten in de koelkast als hij instabiel lijkt.
2. Klop de slagroom stijf en schep of spuit deze op de taart.
3. Verwarm de bewaarde ganache voorzichtig in de magnetron (of au bain-marie in een kom) tot deze weer enigszins vloeibaar is – hij mag echter niet helemaal warm worden. Schenk de saus over de taart en decoreer met geraspte chocolade en desgewenst frambozen.

CITRUSSMAKEN

69

BASIC RECIPES

RECIPE Building a smart HUB

Time to cook 120min
Difficulty +++++

Ingredients

- Raspberry Pi 3B+ (other versions might also work)
- Power supply for your Raspberry PI
- SD card (16 GB Recommended)
- Raspberry Pi case
- OpenHABian (free to download)

Prerequisites

Tools

- Etcher (free to download)
- Putty (free to download)

Intro

If you have gone through the previous chapters, then you already know that OpenHAB is a great open source home automation platform that fits perfectly into our system-of-systems architecture for building a connected home. OpenHab can run on a low-cost hardware Single board computer (SBC) such as the popular Raspberry Pi. The software version of openHAB that runs on a Pi is called openHABian.

The smart HUB functions as an edge gateway, Besides the local processing for our home automation, it connects all types of devices over various protocols and networks, even those that do not travel the Internet such as Bluetooth and exposes them to other platforms in our home or in the cloud.

This recipe covers the installation of openHABian V1.5 on a Raspberry-Pi Model 3B+.

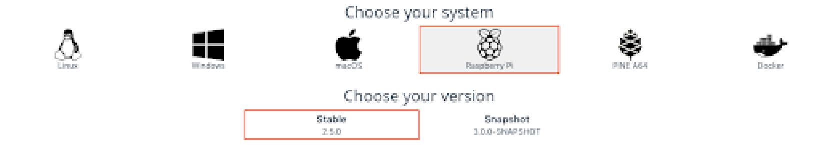
OpenHAB did a wonderful job by providing a ready to use image which includes the Operating System (Debian linux), the openHAB platform and some interesting add-ons. This makes the installation straightforward.

The Building Plan

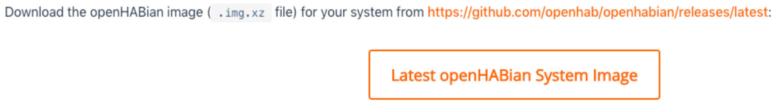
Preparing your image

let’s burn the image on our SD card:

- Go to the download section of the OpenHAB website[1]
- Select the Raspberry Pi as system on the download page.
- Select the version you want to install. The choice is between Stable or Snapshot. choose stable for your production environment.



- Download the Latest openHABian system image by clicking on the link as outlined in the image below


Download the openHABian image (`.img.xz` file) for your system from https://github.com/openhab/openhabian/releases/latest:

- Select the 32-bit version
- Once you have downloaded your openHABian image, you should flash it on an SD card for your Raspberry Pi. OpenHab proposes to use Etcher, which is a popular tool for burning your SD card. Just follow the instructions on the openHAB download page.
- Next insert the SD card in your Raspberry Pi and boot it. The installation will take approx. 45 minutes.

Best is to connect your Raspberry Pi to your home network using a LAN cable. You can use a WiFI connection if your Raspberry Pi supports it, but as the Raspberry Pi will be used as a main hub to connect all kinds of IoT devices and services, it will be more stable when it has a LAN connection and not depend on a less-stable WiFi connection.

[1] https://www.openhab.org/download/

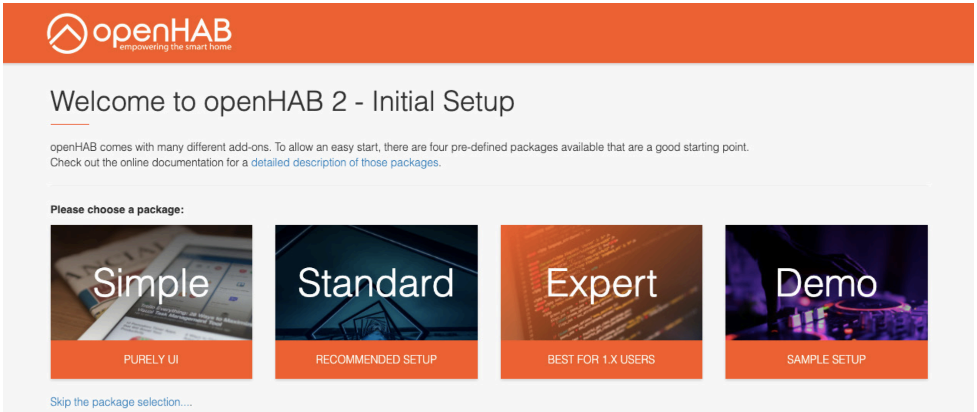
When the installation is finished, you will be able to browse to your Raspberry Pi. OpenHAB has a frontend which you can reach on port 8080 and includes one or more applications, depending how you set up your device. verify your Internet modem to see which IP address your Raspberry Pi received and use that in your browser.

Example: http://your-ip-address:8080

If your raspberry pi got the IP address 192.168.0.18. you should enter: http://192.168.0.18:8080

You see the Initial Setup screen from openHAB 2. There are 4 pre-configured packages. The Standard package is the most recommended and good for our purpose.

- Select the Standard package

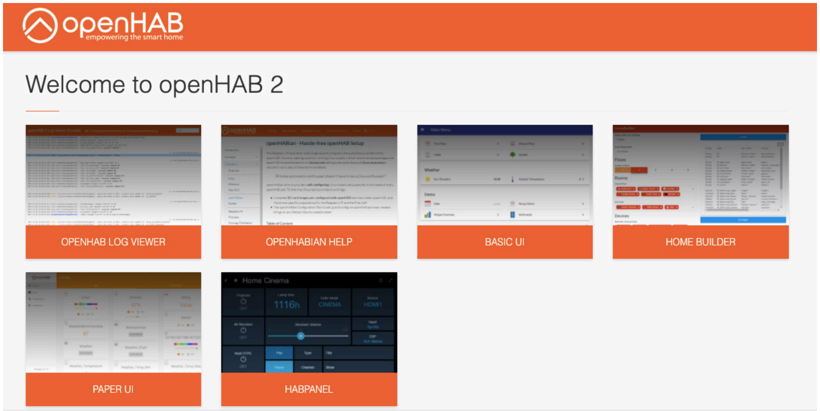


Welcome to openHAB 2 - Initial Setup

openHAB comes with many different add-ons. To allow an easy start, there are four pre-defined packages available that are a good starting point.
Check out the online documentation for a detailed description of those packages.

Please choose a package:

After selecting the standard package you should see the following applications in the OpenHAB welcome screen. You might need to refresh your browser a few times and wait a bit until all applications appear.



Welcome to openHAB 2

Change the default password

After the installation of openHAB, the minimum you should do to secure your openHAB installation is change the default password of the openhabian user.

The default username and password: **openhabian**

You can do this as follows:

- Login with the user openhabian using an SSH connection onto your Raspberry Pi. (You can use a tool like Putty to establish an SSH session to your raspberry Pi for Windows or use the Terminal program on a Mac OS)
- Type: **passwd**
- Enter your **current password**
- Enter a new password for your account

This is it. You have executed your first recipe! The outcome is that you have a system which is ready to connect different kinds of sensors using different protocols.

The next section is optional, but highly recommended if you want to control your Smart HUB remotely or want to use a voice assistant.

Installing OpenHAB Cloud Connector

The openHAB Cloud Connector allows connecting the local openHAB runtime to a remote openHAB Cloud instance [1], such as myopenHAB.org [2], which is an instance of the openHAB Cloud service hosted by the openHAB Foundation [3].

The openHAB Cloud service (and thus the connector to it) is useful for different use cases:

- It allows remote access to local openHAB instances without having to expose ports to the Internet or to require a complex VPN setup.
- It serves as a connector to Google Cloud Messaging (GCM) and Apple Push Notifications (APN) for pushing notifications to mobile phone apps.
- It brings integration possibilities with services that require an OAuth2 authentication against a web server, such as IFTTT or Amazon Alexa Skills

So, if you want to connect to your Smart HUB via the Internet, this is something you want to install. It is a secure way to connect to your Smart HUB from the Internet.

UUID and Secret

To authenticate with the openHAB Cloud service, you need to install the add-on in openHAB.

- Open PaperUI from the openHAB url.
- In the left menu, selct **Add-ons**
- Select **MISC** in the top menu
- Goto **openHAB Cloud Connector**
- Click **install**

The add-on generates two values. These values need to be entered in your account settings of the openHAB Cloud service.

File	Regular Installation	APT Installation
UUID	userdata/uuid	/var/lib/openhab2/uuid
Secret	userdata/openhabcloud/secret	/var/lib/openhab2/openhabcloud/secret

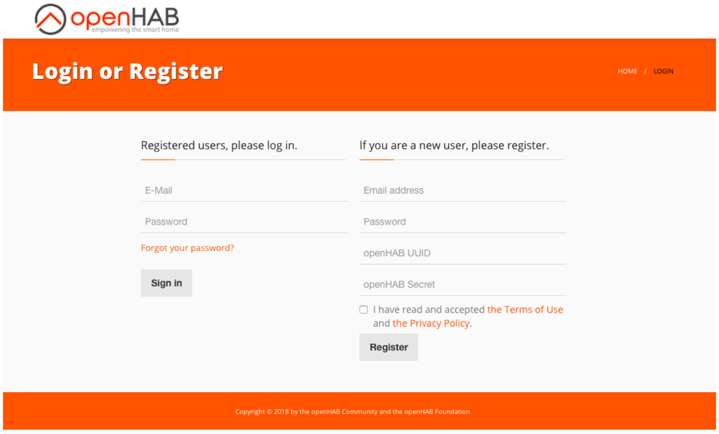
[1] ithub.com/openhab/openhab-cloud/blob/master/README.md
[2] https://www.myopenhab.org/
[3] https://www.openhabfoundation.org/

```
[10:40:09] openhabian@openhab:~$ cat /var/lib/openhab2/uuid
540da818-ac6c-4f02-a9a3-8274fa4fb0b8
```

```
[10:43:02] openhabian@openhab:~$ cat /var/lib/openhab2/openhabcloud/secret
NsluLVmZauDSgXocPJ88
```

The first one is a unique identifier, which allows you to identify your runtime.
You can think of it as something similar like a username for the cloud authentication. The second one is a random secret key which serves as a password. Both values are written to the local file system. If you lose these files for some reason, openHAB will automatically generate new ones. You will then have to reconfigure UUID and secret in the openHAB Cloud service under the My account section.

Next, Goto myopenhab.org and register/login an account.



openHAB
empowering the smart home

Login or Register

HOME / LOGIN

Registered users, please log in.

E-Mail

Password

Forgot your password?

Sign in

If you are a new user, please register.

Email address

Password

openHAB UUID

openHAB Secret

□ I have read and accepted the Terms of Use and the Privacy Policy.

Register

Copyright © 2018 by the openHAB Community and the openHAB Foundation

Configuration

When talking to people that want to set up their home automation system at home, they are very suspicious to open their systems to others, which is quite understandable. On the other hand, to build a system-of-systems approach and get the most out of it, you need to peer with other systems.

You can (and should) limit the access to third party systems. This can be done by only expose the items you want to share with the other platforms. You can do this in OpenHAB in the Paper UI under Configuration -> Services -> IO -> openHAB Cloud:

Alternatively, you can configure the settings in the file conf/services/openhabcloud.cfg.

Configuring a static IP address for your Smart HUB

Most home network routers provide a DHCP address. This means that the router will distribute an IP address to IP capable devices within a certain range. The advantages is that you don't have to set it up manually, but the disadvantage is that this IP address might change in future (when the lease expires). This behavior is something you might want to avoid on your Smart HUB especially if you would refer to it's IP address in a system-of-systems architecture.

To assign a static IP address to your Raspberry Pi, proceed as follows:

- Login with the user **openhabian** using an ssh connection onto your raspberry pi
- Edit the dhcpcd.conf file
    - **sudo nano /etc/dhcpcd.conf**
- Scroll all the way to the bottom of the file and add the following lines of code

```
interface eth0

static ip_address=192.168.0.10/24
static routers=192.168.0.1
static domain_name_servers=192.168.0.1
```

Note: You'll need to edit the numbers in the snippet so they match your network configuration

- **interface**: This defines which network interface you are setting the configuration for.
- **static ip_address**: This is the IP address that you want to set your device to. (Make sure you leave the /24 at the end)
- **static routers**: This is the IP address of your gateway (probably the IP address or your router)
- **static domain_name_servers**: This is the IP address of your DNS (probably the IP address of your router). You can add multiple IP addresses here separated with a single space

- To exit the editor, press **ctrl+x**
- To save your changes press the letter **Y** then hit enter

Now all you need to do is reboot, and everything should be set!

- **sudo reboot -h 0**

You can double check by typing

- **ifconfig**

And checking the interfaces IP address

What's next?

A good next step is to add some devices to your Smart HUB. Have a look at recipe 'Using Zwave devices' further in this book.

Recipes to connect your objects for the connected house

Connected devices

Recipes for building User Interfaces

User Interfaces

```
Installing package
Package: android-sdk 26.1.1-1 (Mon Feb 1
ng runtime dependencies...
ng buildtime dependencies...
ving sources...
oading sdk-tools-linux-4333796.zip...
  % Received % Xferd  Average Speed   Ti
                      Dload  Upload   To
100  147M    0     0  4682k      0  0:00
android-sdk.sh
android-sdk.csh
android-sdk.conf
license.html
ting source files with sha1sums...
ols-linux-4333796.zip ... Passed
d-sdk.sh ... Passed
dk.csh ... Passed
```

Recipes to add automation to your house

Automation

Security & Privacy

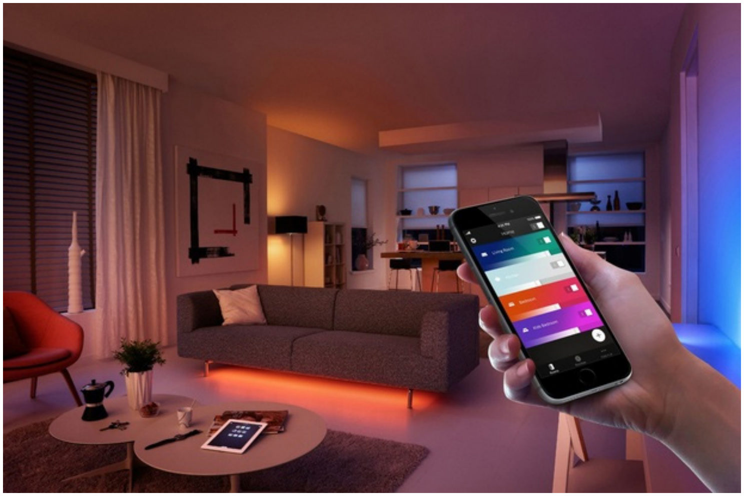
When discussing the connected house, one needs to consider security and privacy. Each connected object is a source of concern for safety. Every data stream is a concern for our privacy. It goes without saying that one product has a greater impact on safety and privacy than another.

A soil sensor in the garden is less of a security and privacy risk than a connected door lock. If someone can hack your connected door lock, he or she can gain access to your house as if they had the physical key. It's not just about the 'classic' physical security like accessing a building. It's also about gaining access to the data of various objects in a house that can reveal the whereabouts of the occupants of that house. Last but not least, connected devices can be misused for malicious purposes when they are hacked.

In 2016, on the morning of Friday 21st of October, internet users from the East Coast of the US noticed that web pages such as Twitter, Etsy, Spotify, Netflix and GitHub were unresponsive. The source of the problem was that Dyn, one of the largest internet management companies in the United States, was flooded with junk traffic. Part of the attack was caused by a "botnet" of Internet of Things devices that caused a DDoS (distributed denial of service) attack on Dyn.
Hackers used a DDoS attack to bring together swarms of Internet-connected devices such as routers, security cameras, and even connected refrigerators in a botnet.

Many security problems arise because the connected products are 'new'. Manufacturers of household appliances have good domain knowledge about their traditional product, but therefore not necessarily about how to turn it into a connected product. Developing a connected product is a complex process and requires a lot of expertise in different domains. The required expertise is often underestimated, which means they have to solve the problems later on. Ultimately it will result in a reliable and safe product, but it will take some time to get there.

During that time, vulnerabilities may arise, resulting in weaknesses that are attractive to hackers. An example of this is what happened with the Philips Hue lamps.



Signify (Philips Lighting), the company behind Philips Hue, was exposed to a vulnerability issue which was resolved later on through an update.
An issue dated from 2017 was the origin of the vulnerability in which a hacker could take control of a Hue lamp. In addition, it could adjust the brightness or color of the lamp, so that the lamp appeared to have a malfunction. Resetting the lamp while removing it from the network and reconnecting it, the hacker could generate a buffer overflow through a vulnerability in the ZigBee protocol by sending a lot of data to the Hue Bridge. That bridge is the device to control smart lamps and ZigBee is the wireless network to connect specific IoT devices.

This way, the bridge could come under the control of the hacker, and malware could be installed on it. But that bridge is also connected to the home or office network. At home this is often directly connected on the modem where the internet signal comes in, making it possible to hack other devices in the network[1]

[1] https://datanews.knack.be/ict/nieuws/philips-hue-lampen-geven-hackers-toegang-tot-netwerk/article-news-1562389.html

What can we do about it?

Connected devices are different from laptops, servers or traditional IP-based machines. Internet of Things (IoT) devices use different types of services and communicate differently on a network and therefore require their own security strategy.

Change default passwords, use strong passwords

One of the simplest security measures you can take, but often forgotten, is to change the default username and passwords such as (admin:admin).

A website called Insecam collects live images from internet cameras around the world that still use the manufacturer's default username and password. The website contains thousands of links to live video streams from cameras in Australia, including footage from homes and businesses, shopping malls, cafes, warehouses and other typical security camera locations.

Like any Internet-connected device, you will need to change the default username and password when you set it up. An unmodified default username and password gives an attacker easy access to any Internet-connected device, meaning these devices remain vulnerable to malicious activity.

When you set a new password, make sure you create a strong password.

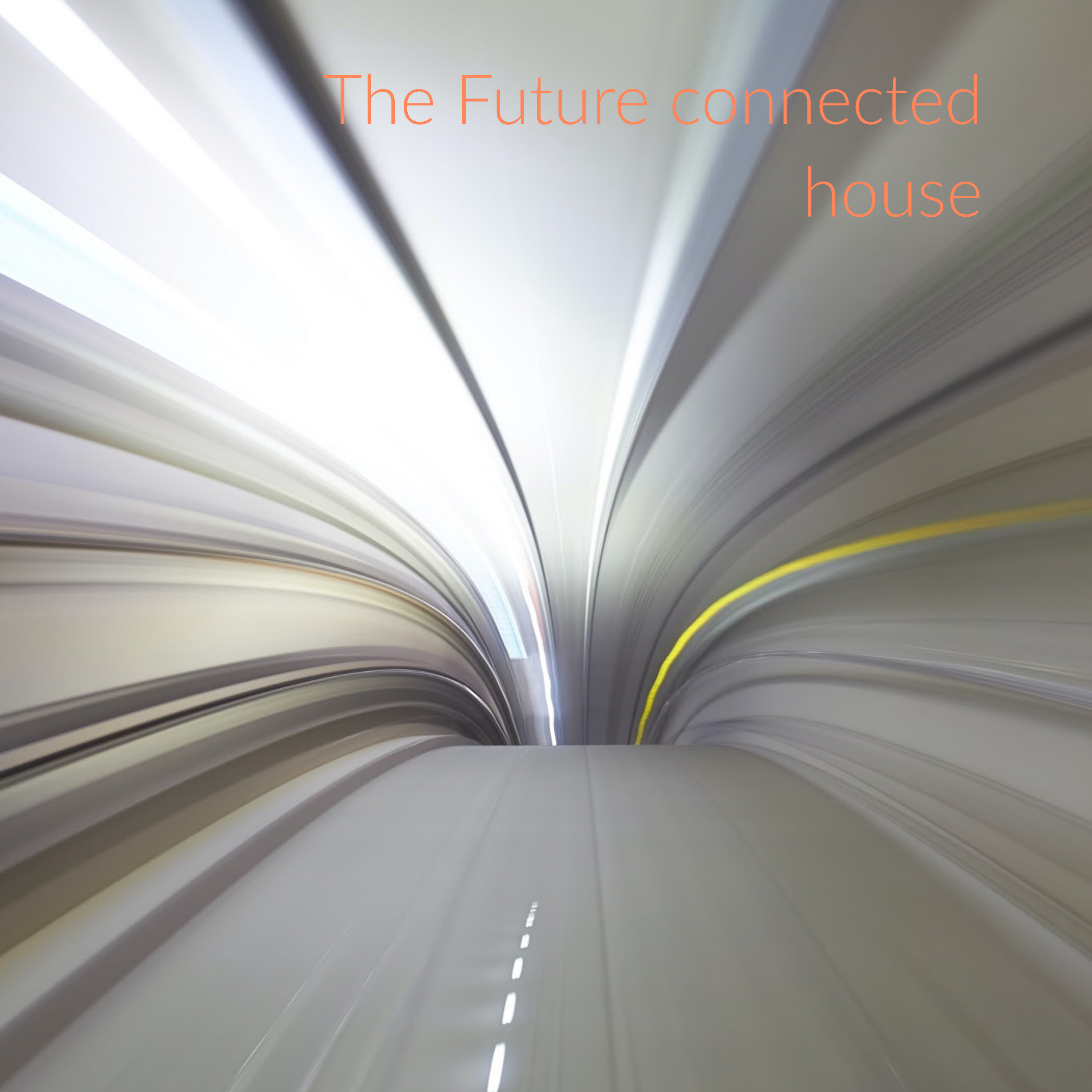
- Use a mix of alphabetical and numeric characters.
- Use a mixture of upper- and lowercase; passwords are case sensitive.
- Use symbols if the system allows

Make sure you keep your passwords safe and don't share them with anyone else.

Segregate the network

IoT traffic should be separated from an active or primary network. This is already the case when a radio network is used for the devices to communicate such as Zigbee, Z-Wave or any type of LPWAN network such as LoRaWAN, Sigfox or LTE-M. When devices share your primary LAN or WiFi network, it may be a good idea to split your network into different VLANs and apply stricter policies for your device network. For most devices, you know what types of information these devices process, what services they use, and what outdoor access is needed.

The Future connected house



Now that we are almost at the end of this cookbook, it should be clear that today's technology enables us to connect all kinds of objects in our homes. But we are still at the beginning of what is really possible. You could compare it to a child growing up. In this situation we are a 2 year old child who is starting to use its senses, while the brain still needs to develop fully.

This cookbook will help you integrate or build your own connected devices - transforming a passive home into an active living space. The result is still very basic and in most cases based on conditional logic (IF... THEN...). More intelligent devices already exist - such as the digital assistant - but they still need to be correctly integrated and programmed in order to be efficient.

Fusing sensor data with artificial intelligence and machine learning makes the connected home smarter. In this chapter I provide a glimpse into the future and see how the connected home could evolve in the coming years.

More sensors results in more insights

The digital energy meter

The traditional house will become more and more connected. Today, residents have to send their energy meter data once a year to a utility company. This process not only provides minimal information on the consumption patterns of electricity, gas and water, but is also inefficient.

Last year, the installation of smart meters started in Flanders. Fluvius, a utility company, started installing the smart meters in new and renovated houses.

Smart meters will make automatic meter registration possible. The utility company will have access to real-time data, giving it better insight into energy consumption and more accurate forecasts. Better forecasting is needed to cope with the shift towards a more decentralised energy network. This means that energy is not only produced by a handful of energy companies, but that each home acts as a potential energy producer.

The digital water meter

Just like the digital energy meters, utility companies want to gain a better insight into water consumption and detect leaks. Studies have shown that more than 25% of water intended for consumption is lost due to leaks. With the coming climate change, this is no longer justified. In more and more regions there is a shortage of water due to prolonged droughts. Over the past year, measures had to be taken by the authorities to regulate water consumption. Farmers suffered heavy losses because the crops could not grow due to the prolonged drought and it was forbidden to irrigate due to the low water level in the canals and rivers.

By continuously monitoring water consumption in homes, utility companies can detect leaks much more easily and intervene, so that valuable water is no longer wasted. In Flanders, the utility companies have started rolling out the first batch of 300,000 smart water meters. The smart water meters use the Sigfox LPWAN technology to communicate with the outside world. Every day, one payload of data is uploaded from the smart meter to the utility company to report the water consumption.

The choice of Sigfox was based on network availability, range and low power consumption. Most meters are located in hard-to-reach places for wireless communication, such as basements. SigFox uses the free sub GHz band (868Mhz in Europe) which allows deep penetration. It is also efficient in power consumption, as every water meter must be able to communicate every day on a single battery charge for 16 years [1]. Communicating daily for 16 years is not a given , It requires a unique design and batteries with almost no self-discharge specifications.

Well Monitoring

Gradually we are surrounded by a plentitude of sensors. Flanders is thinking of making the wells 'connected'. Every new home in Flanders will have to install a well to collect the rainwater used to flush toilets, washing machines, water the garden, etc.

By monitoring the water level in the well and emptying it when heavy rainfall is expected, the wells can serve as a buffer that prevents rainwater from flowing directly into rivers, thus avoiding flooding in low-lying areas. Flanders has 1.5 million residential wells, suitable for approx. 15 billion litres of water storage. The above examples show that the connected house is a fact. Similar plans are being implemented not only in Flanders, but in every country.

The Number of connected devices per person

The number of Internet-connected devices that people have is going up. On average, there will be four networked devices and connections per person globally by 2021, according to the latest annual visual networking index forecast by Cisco [1].

However, in North America, there will be 13 networked devices and connections per person, up from eight last year. This means that beyond smartphones and connected TVs, North American consumers will be adopting many more connected devices.
North America is well above the average by region when it comes to getting connected. Below are the projected number of networked devices and connection per person by region by 2021:

[1] https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html