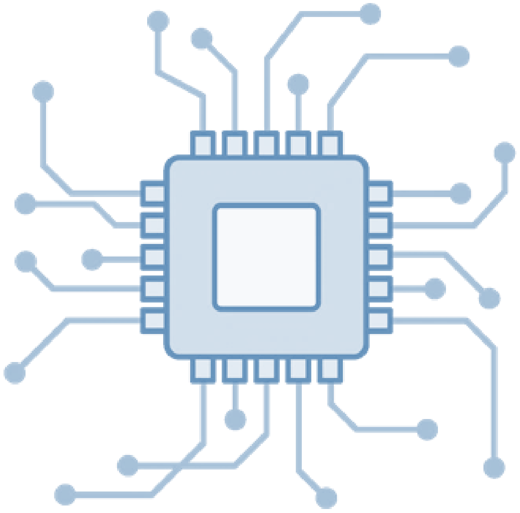
CodeWeek.EU

IEEE
University of Porto - Student Branch

A Workshop on the Internet-of-Things

From the **Cloud** to the *Edge* and the *Mist*



João Pedro Dias
23 October 2020

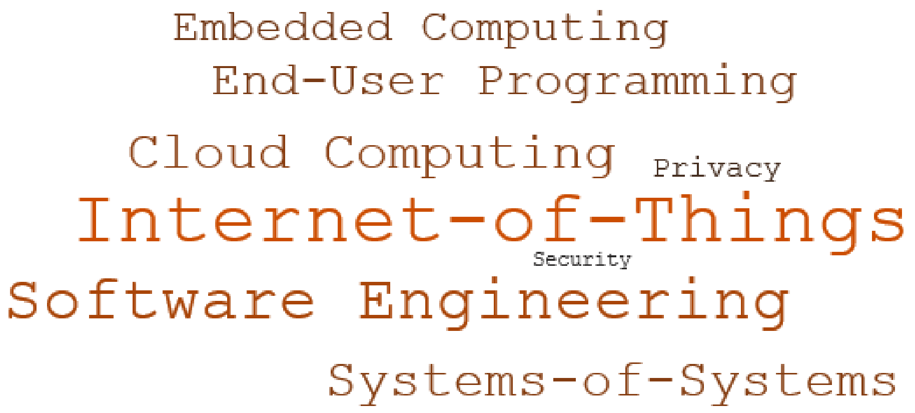
$ whoami

- From Porto, PT

- Researcher @ INESC TEC
- PhD Student @ FEUP/DEI

- Previous talks/workshops
  - PixelsCamp 2017 and 2019
  - 0xOPOSEC meetups
  - …

- Me around the web:
  - https://jpdias.me
  - https://twitter.com/jpd1as/
  - jpmdias@fe.up.pt



Internet-of-Things *by the standards*

- *"An infrastructure of **interconnected objects**, people, systems and information resources together with intelligent services to allow them to **process information of the physical and the virtual world and react**."*

ISO/IEC JTC 1 Internet of Things (IoT)

What *is* IoT *really*?

- *A network of physical objects —* ***things*** *— that are embedded with* ***sensors, actuators, software, and other*** *technologies for the purpose of* ***connecting and exchanging data*** *with other devices and systems* ***over the Internet****.*

From Wikipedia, the free encyclopedia

CodeWeek.

PHILIPS
hue

0,65 kW
edp re:dy
edp

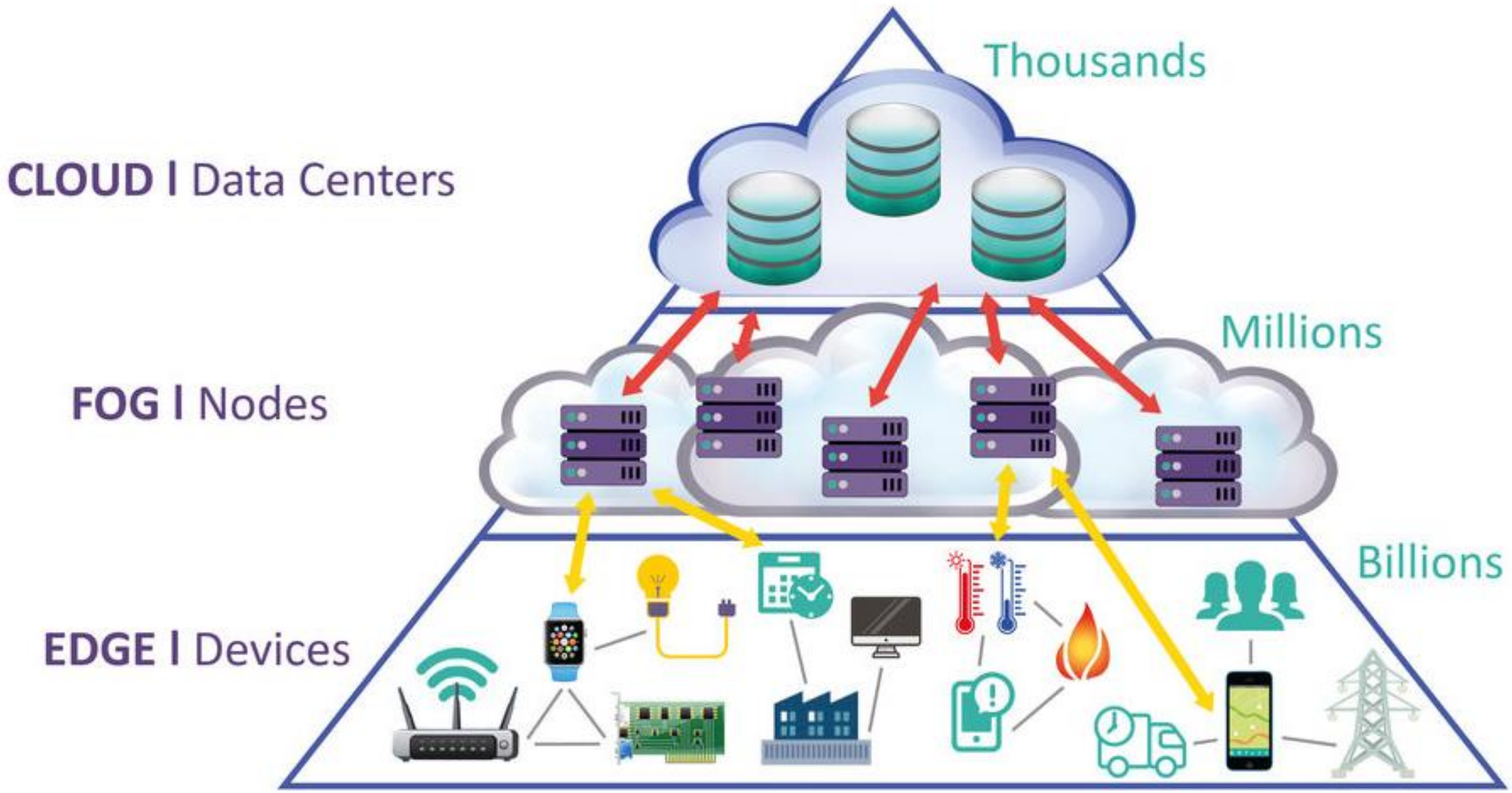
mi

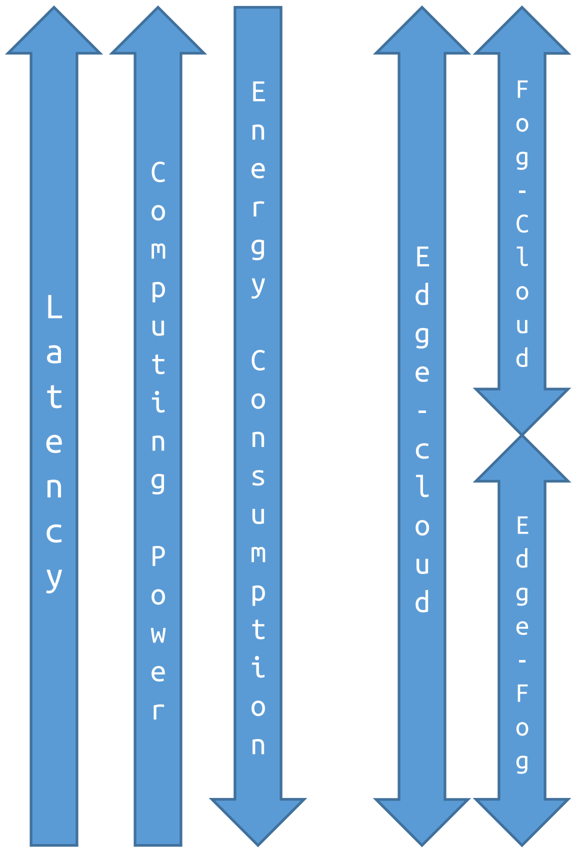
nest
70
nest
Google Home
Hands-free smart speaker
G

ring
ring

IKEA

Cloud, Fog and Mist (Edge)





Latency

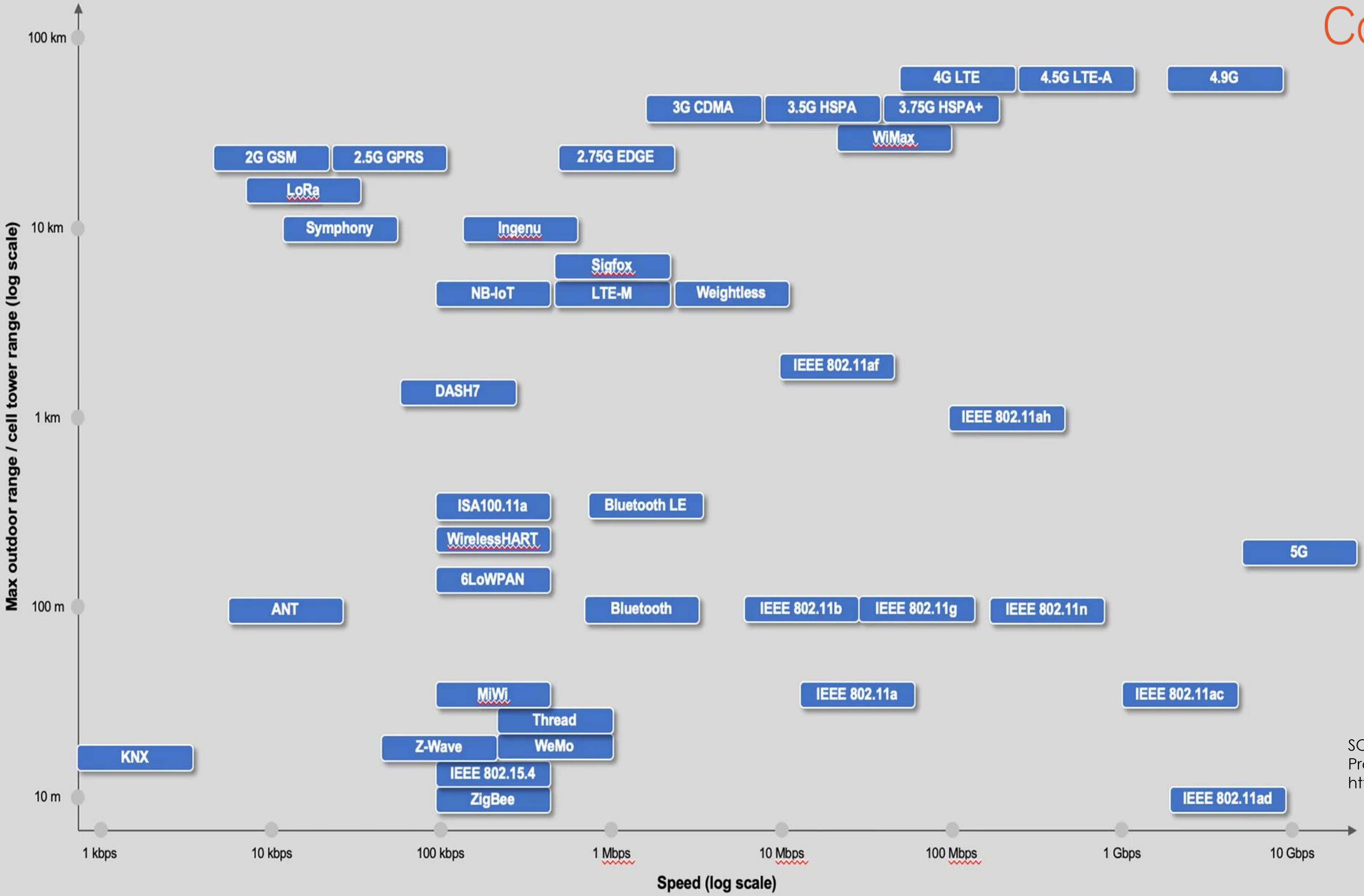
Computing Power

Energy Consumption

Edge-cloud

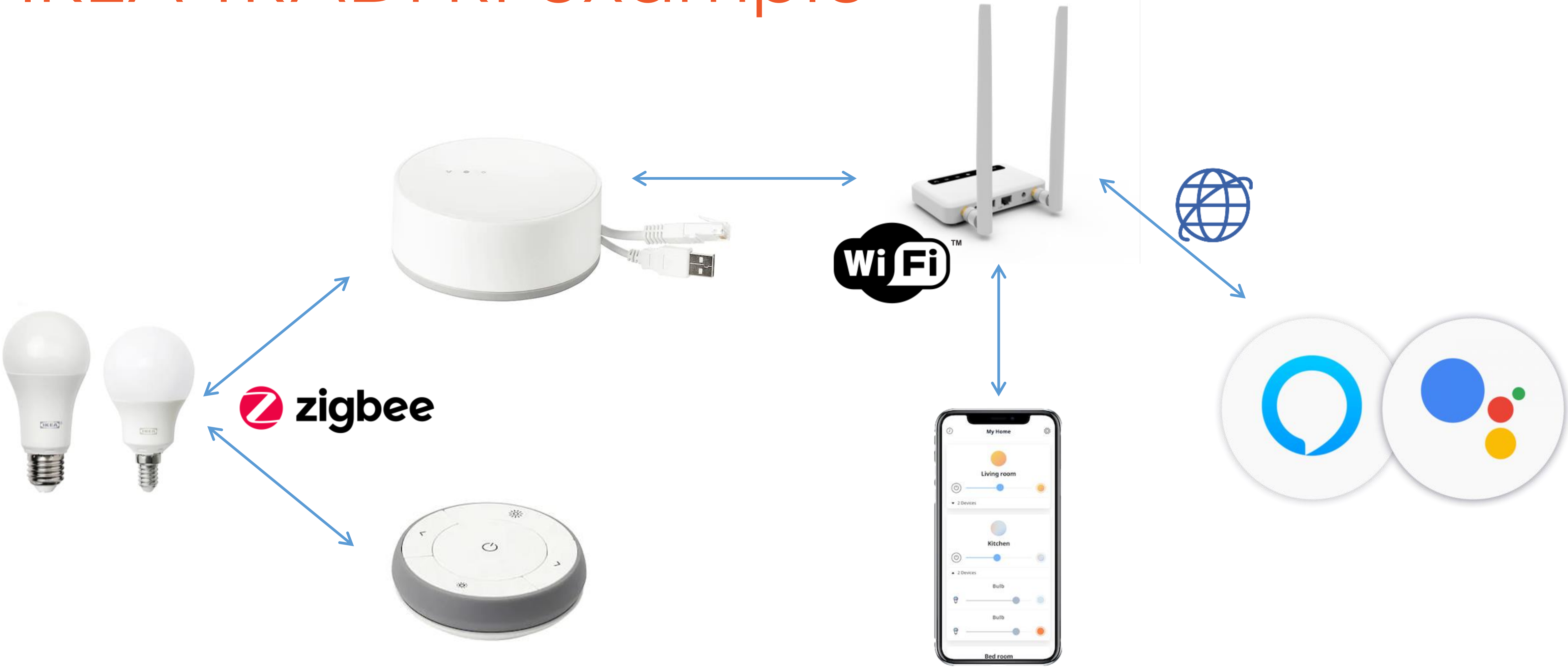
Fog-Cloud

Edge-Fog



SOURCE: Internet of Things (IoT) Wireless Protocols, https://5g.security/iot-wireless-protocols/

IKEA TRÅDFRI example



The DIY path

Because buying things is expensive... and there's no fun in that.

Also, personalize your system, after all, it's yours.

Plan of attack

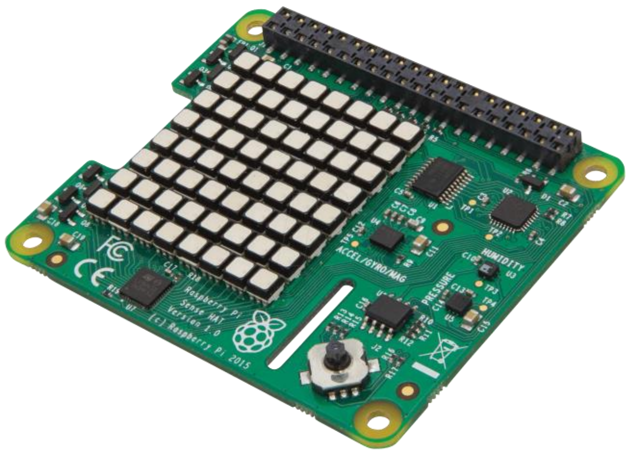
1. Define your architecture
   - Edge-Cloud, Edge-Fog-Cloud, Edge-Fog (local only)
2. Pick one or more communication protocols
   - ZigBee, Bluetooth LE, RF433, WiFi (REST, MQTT, CoAP, etc.), …
3. Choose your microcontrollers (Mist)
   - ESP8266, ESP32, Atmel, Nordic, …
   - Sometimes, a Operating System can be used: FreeRTOS, Zephyr,…
4. Get a protocol-compliant gateway (Fog)
   1. Raspberry Pi (or other SBC), Full-fledge server,…
5. Pick a cloud provider
   - AWS, Azure,…
   - Out-of-the-shelf solution: Google Assistant, IFTTT, …

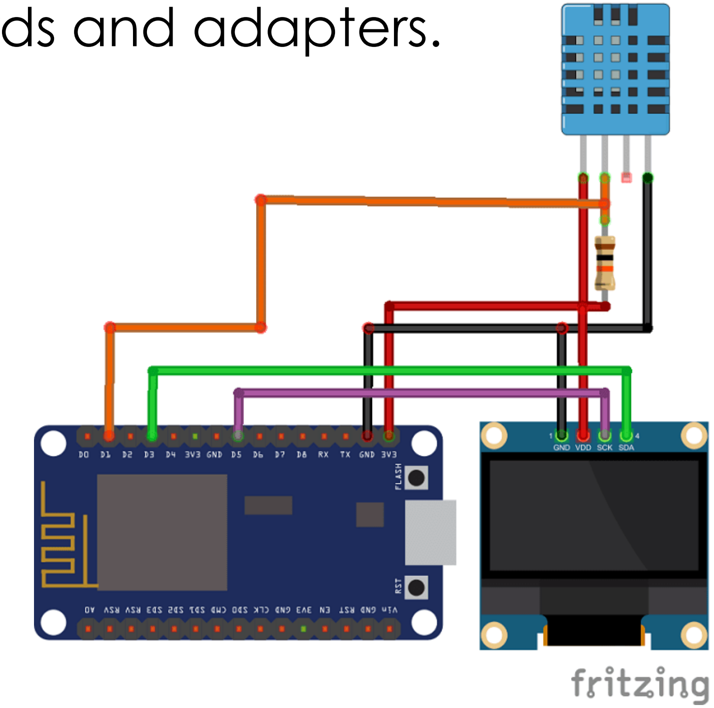
The Software

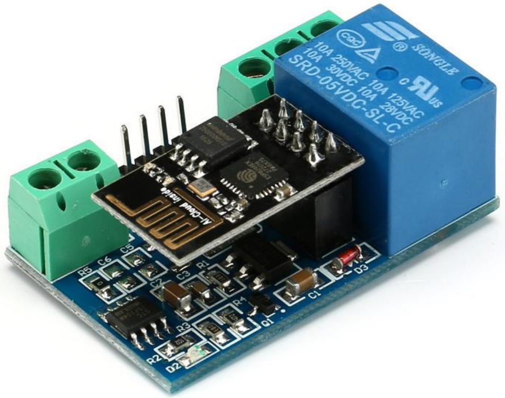
- Edge devices:
  - C, Arduino, microPython
- Fog devices:
  - Full-fledge Linux
  - Node-RED, Domoticz, Home Assistant, OpenHab…
- Cloud:
  - Anything
  - Out-of-the-shelf services

The Hardware

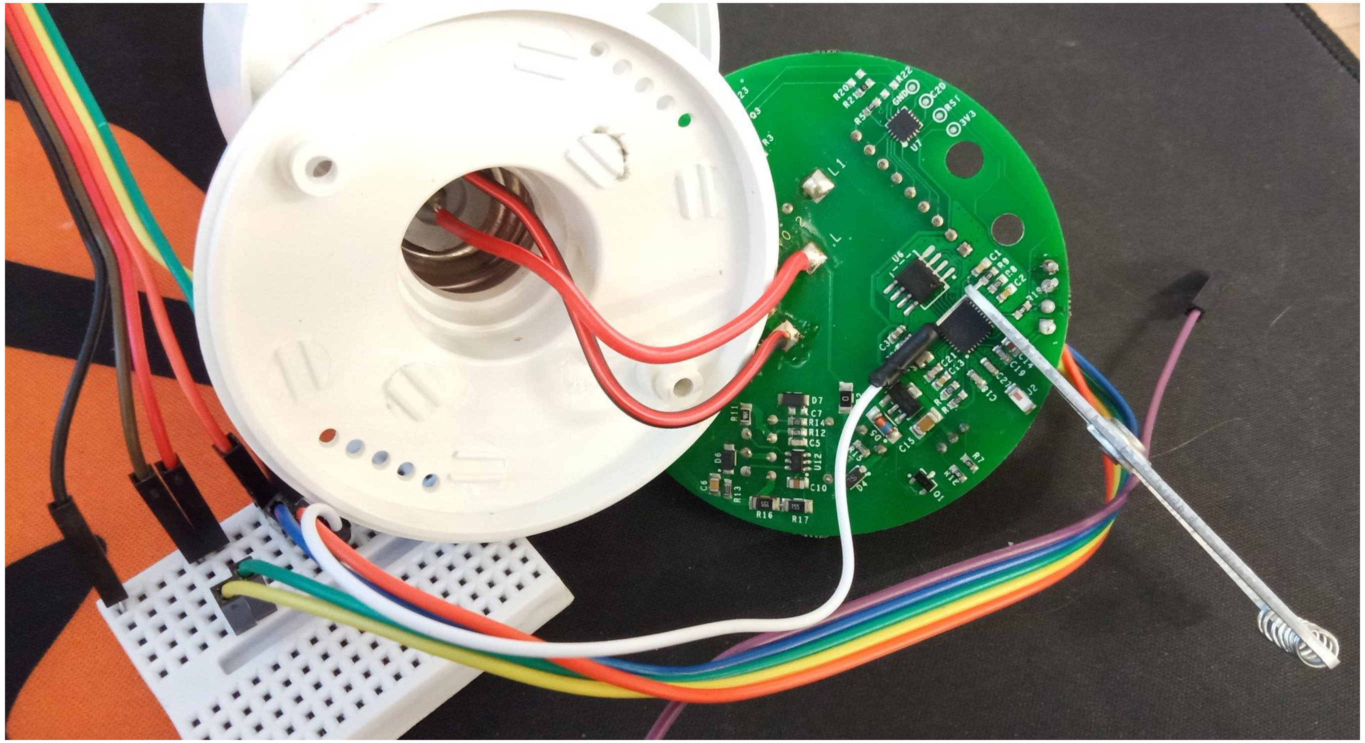
- Flash existent hardware with your software.
  - Serial port is your friend.
- Make your own circuits.
- Buy rapid development boards and adapters.





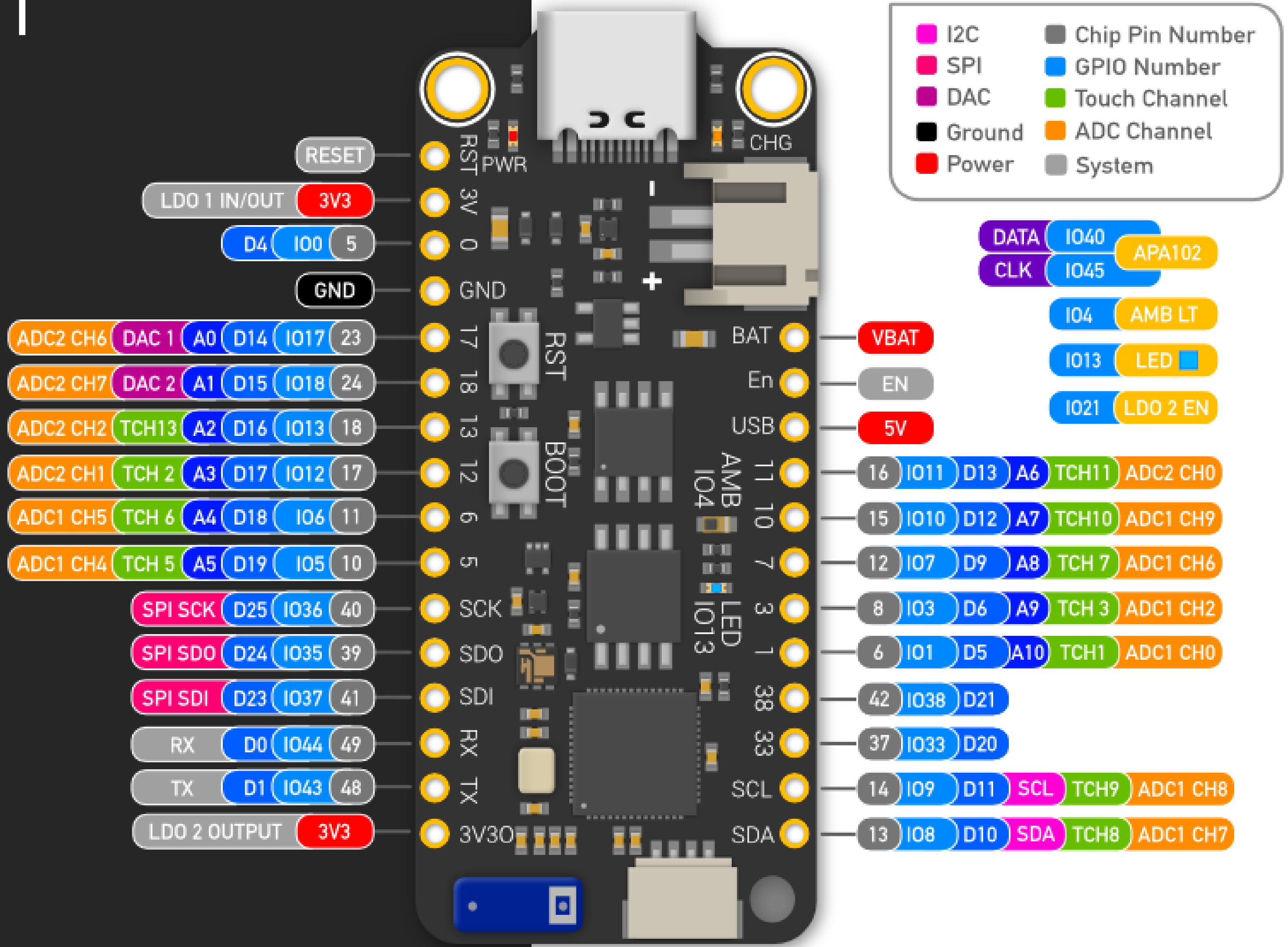


Flashing a Sonoff Slampher with Tasmota



How To: https://tasmota.github.io/docs/devices/Sonoff-Slampher/
Tasmota: https://github.com/arendst/Tasmota

ESP32S2 Pinout


I2C
SPI
DAC
Ground
Power
Chip Pin Number
GPIO Number
Touch Channel
ADC Channel
System
RESET — RST
LDO 1 IN/OUT 3V3 — 3V
D4 IO0 5 — 0
GND — GND
ADC2 CH6 DAC 1 A0 D14 IO17 23 — 17
ADC2 CH7 DAC 2 A1 D15 IO18 24 — 18
ADC2 CH2 TCH13 A2 D16 IO13 18 — 13
ADC2 CH1 TCH 2 A3 D17 IO12 17 — 12
ADC1 CH5 TCH 6 A4 D18 IO6 11 — 6
ADC1 CH4 TCH 5 A5 D19 IO5 10 — 5
SPI SCK D25 IO36 40 — SCK
SPI SDO D24 IO35 39 — SDO
SPI SDI D23 IO37 41 — SDI
RX D0 IO44 49 — RX
TX D1 IO43 48 — TX
LDO 2 OUTPUT 3V3 — 3V3O
PWR
CHG
RST
BOOT
AMB IO4
LED IO13
DATA IO40
CLK IO45
APA102
IO4 AMB LT
IO13 LED
IO21 LDO 2 EN
BAT — VBAT
En — EN
USB — 5V
11 — 16 IO11 D13 A6 TCH11 ADC2 CH0
10 — 15 IO10 D12 A7 TCH10 ADC1 CH9
7 — 12 IO7 D9 A8 TCH 7 ADC1 CH6
3 — 8 IO3 D6 A9 TCH 3 ADC1 CH2
1 — 6 IO1 D5 A10 TCH1 ADC1 CH0
38 — 42 IO38 D21
33 — 37 IO33 D20
SCL — 14 IO9 D11 SCL TCH9 ADC1 CH8
SDA — 13 IO8 D10 SDA TCH8 ADC1 CH7

I already have some smart things...

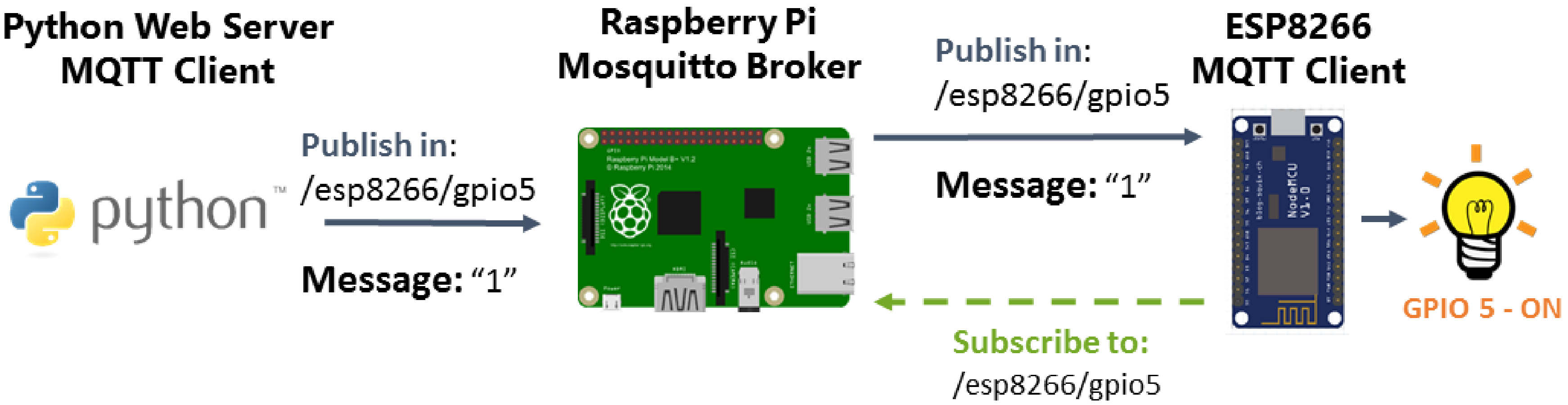
Now what?

- Maybe you can *flash* it!
- Welcome to the **protocol dongle jungle.**
  - Make bridges for existent protocols.
  - Typically you will need some specific hardware and software to convert between protocols.
  - This is one of the core features of the **fog** tier.

- Zigbee to MQTT bridge
  - https://www.zigbee2mqtt.io/
- IR blaster
  - https://github.com/mdhiggins/ESP8266-HTTP-IR-Blaster
- RF433, IR, BLE broker
  - https://docs.openmqttgateway.com/

How it all comes together

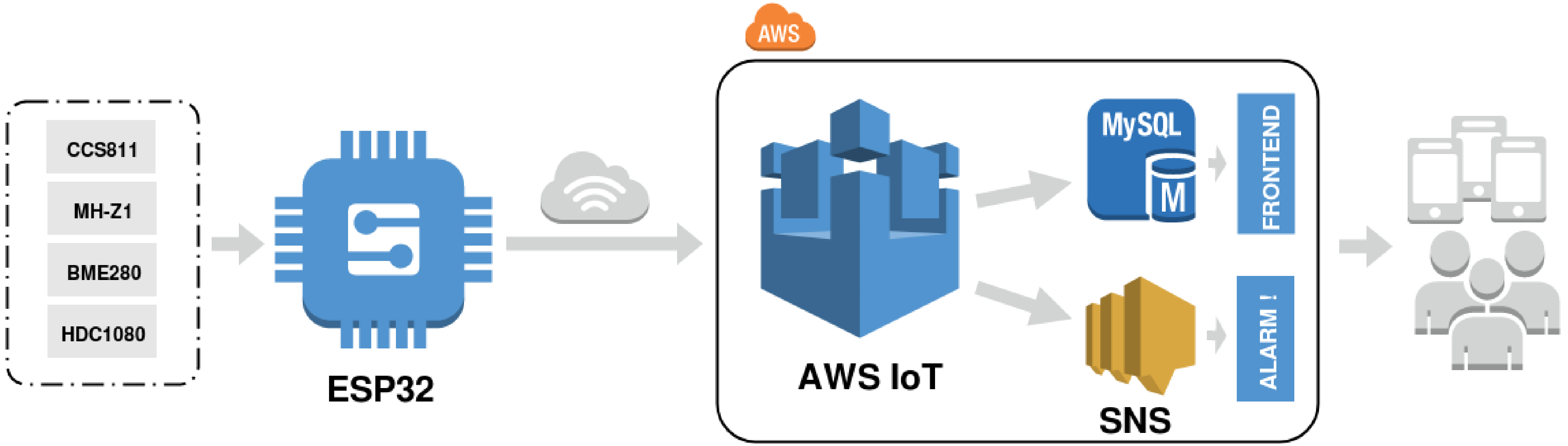
The MQTT + WebServer Way



Source: https://randomnerdtutorials.com/raspberry-pi-publishing-mqtt-messages-to-esp8266/

How it all comes together

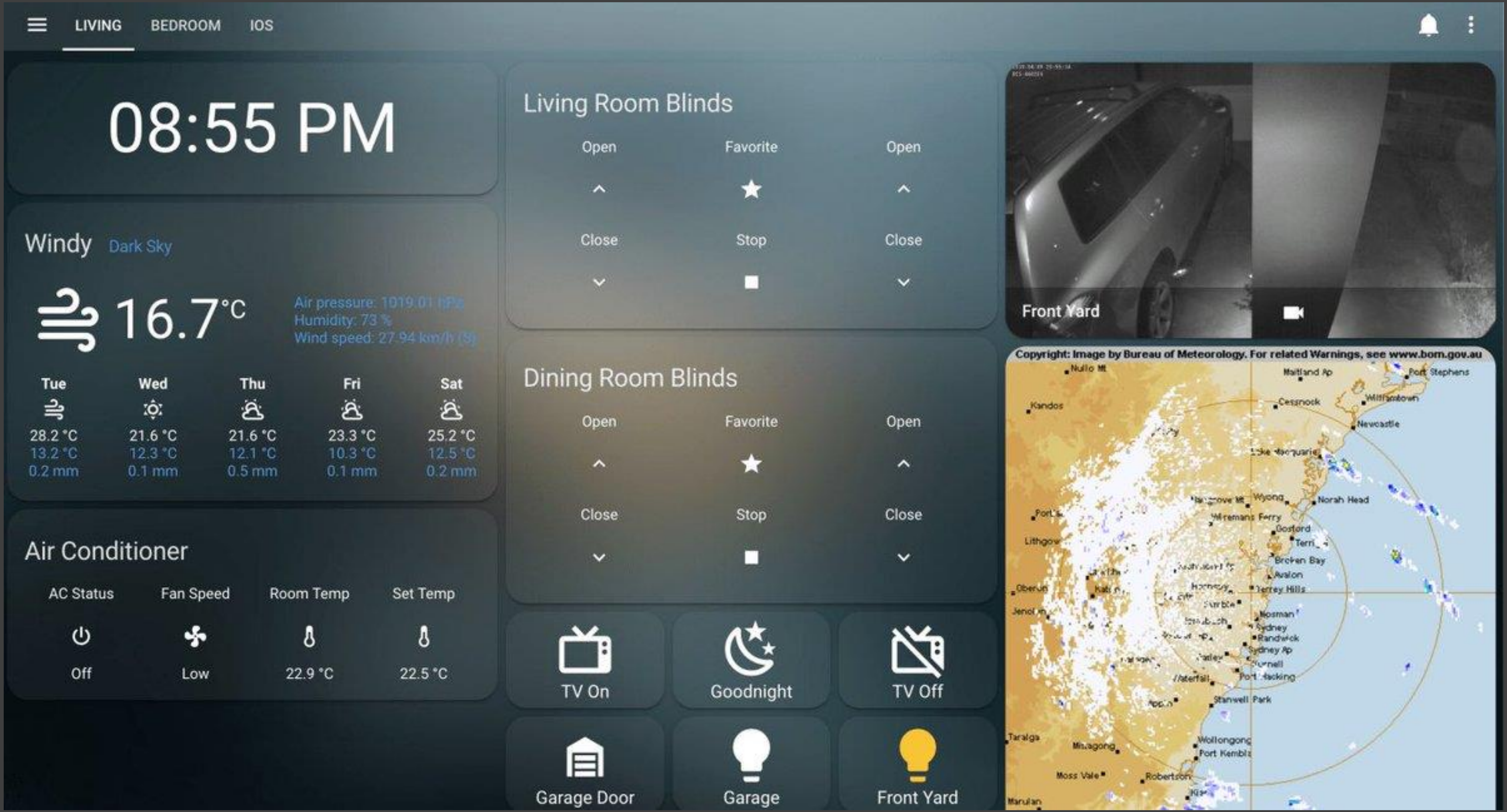
The Cloud All-in



Source: https://sudonull.com/post/3560-Configuring-data-transfer-from-the-device-to-AWS-IoT-Core

Home Assistant example

CodeWeek.



LIVING BEDROOM IOS

08:55 PM

Windy Dark Sky

16.7°C

Air pressure: 1019.01 hPa
Humidity: 73 %
Wind speed: 27.94 km/h (S)

Tue	Wed	Thu	Fri	Sat
28.2 °C	21.6 °C	21.6 °C	23.3 °C	25.2 °C
13.2 °C	12.3 °C	12.1 °C	10.3 °C	12.5 °C
0.2 mm	0.1 mm	0.5 mm	0.1 mm	0.2 mm

Air Conditioner

AC Status	Fan Speed	Room Temp	Set Temp
Off	Low	22.9 °C	22.5 °C

Living Room Blinds

Open Favorite Open

Close Stop Close

Dining Room Blinds

Open Favorite Open

Close Stop Close

Copyright: Image by Bureau of Meteorology. For related Warnings, see www.bom.gov.au

Virtual hands-on workshop

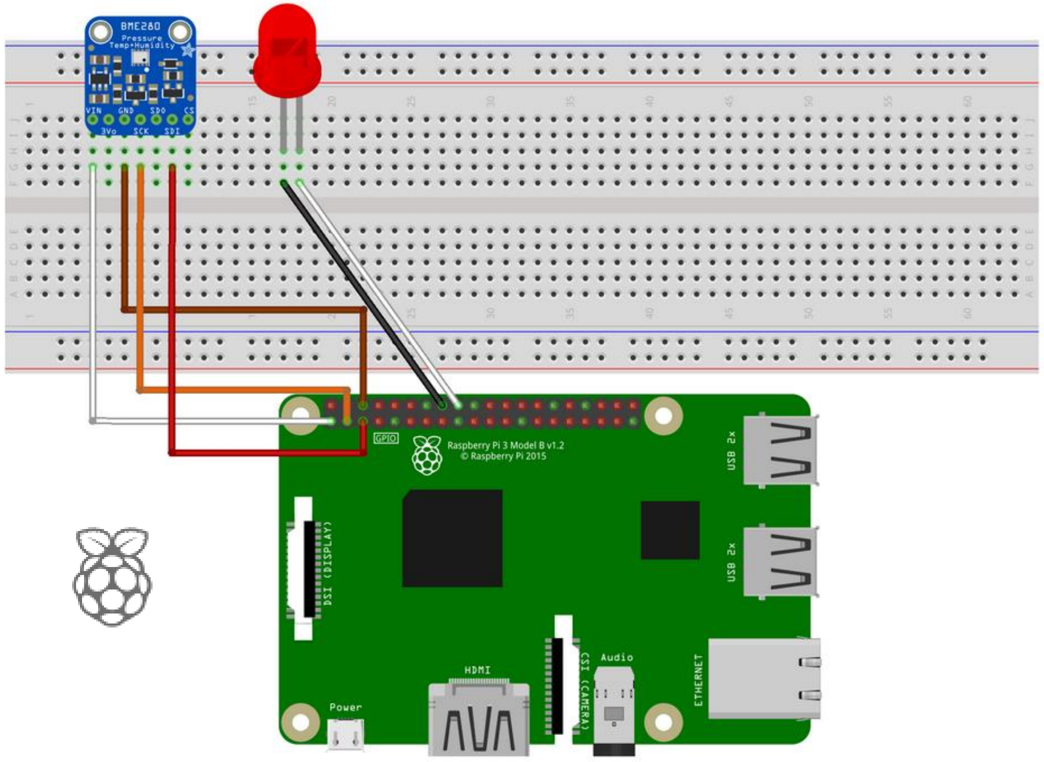
Hammer time!

Goals

1. Toggle a LED.
2. Read data from a sensor.
3. Toggle the LED depending on sensing data.
4. Send sensing data over the web.
5. Request weather data and act upon it.
6. Try different conditions and change things around.

The virtual way

- Raspberry Pi Azure IoT Online Simulator
  - https://azure-samples.github.io/raspberry-pi-web-simulator/
- Coded in JavaScript with WiringPi
  - https://github.com/WiringPi/WiringPi-Node
  - **Delete all the existent code**! It's for Azure related stuff.
- The Raspberry Pi will be our "edge" device
  - But, typically, that's not the case.
- Hardware available:
  - BME280: humidity, barometric pressure and ambient temperature sensor
  - Red LED



1. Toggle a LED

```
//Import wiringPi
const wpi = require('wiring-pi');
//Set pin to which the LED is connected
const LEDPin = 4;

//wiringPi setup
wpi.setup('wpi’);

//set LEDPin as an OUPUT (we will change its status)
//set LEDPin default status to off
wpi.pinMode(LEDPin, wpi.OUTPUT);
wpi.digitalWrite(LEDPin, 0);

//write to LEDPin the ON status
//Set the voltage to 5V (or 3.3V on 3.3V boards) for 1 (HIGH), 0V (ground) for 0 (LOW)
wpi.digitalWrite(LEDPin, 1);

//wait for 0.5 seconds and then turn off the LED
blinkLEDTimeout = setTimeout(function () {
  wpi.digitalWrite(LEDPin, 0);
}, 500);
```

2. Read data from a sensor (1/2)

```
//Import wiringPi
const wpi = require('wiring-pi');
//Import sensor lib
const BME280 = require('bme280-sensor');

//wiringPi setup
wpi.setup('wpi');

//device configurations
const BME280_OPTION = {
  i2cBusNo: 1, // defaults to 1
  i2cAddress: BME280.BME280_DEFAULT_I2C_ADDRESS() // defaults to 0x77
 };
```

- What is I2C?
  - Is a synchronous, multi-master, multi-slave, packet switched, single-ended, serial communication bus.
  - https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/

2. Read data from a sensor (2/2)

```
//instantiate BME sensor
sensor = new BME280(BME280_OPTION);
sensor.init()
 .catch(function (err) {
  console.error(err.message || err);
 });

//Read sensor data and log
sensor.readSensorData().then(function (data) {
  console.log(data)
});
```

- Expected output:

```
{
    "temperature_C":26.090723800037097,
    "humidity":67.46105997902815,
    "pressure_hPa":10.687267861684184
}
```

3. Toggle the LED depending on sensing data

- Based on the previous code:

```
sensor.readSensorData().then(function (data) {
  if(data.humidity > 50){
    wpi.digitalWrite(LEDPin, 1);
  }
});
```

4. Send sensing data over the Internet

- Edit the previous code and add a fetch POST request.

```
sensor.readSensorData().then(function (data) {
  console.log(data)
  fetch('https://hookb.in/<provided_during_workshop>', {
    method: 'POST', // or 'PUT'
    mode: 'no-cors',
    headers: {
      'Accept': 'text/plain',
      'Content-Type': 'text/plain'
    },
    body: JSON.stringify(data),
  })
})
```

5. Request weather data and act upon it.

```
fetch("https://api.jsonbin.it/bins/el0gfqit")
 .then(resp => resp.json())
 .then(data => {
   if(data.co2ppm > 300){
     wpi.digitalWrite(LEDPin, 1);
   }
 })
```

- The API endpoint is a mock weather data endpoint.
  - In real life, real weather services are used. There are several free.

Danger Zone

Some recommendations.

Vendor lock-in

- “*vendor lock-in*, also known as proprietary *lock-in* or customer *lock-in*, **makes a customer dependent on a *vendor* for products and services**, **unable to use another** *vendor* **without substantial** switching **costs.** ”

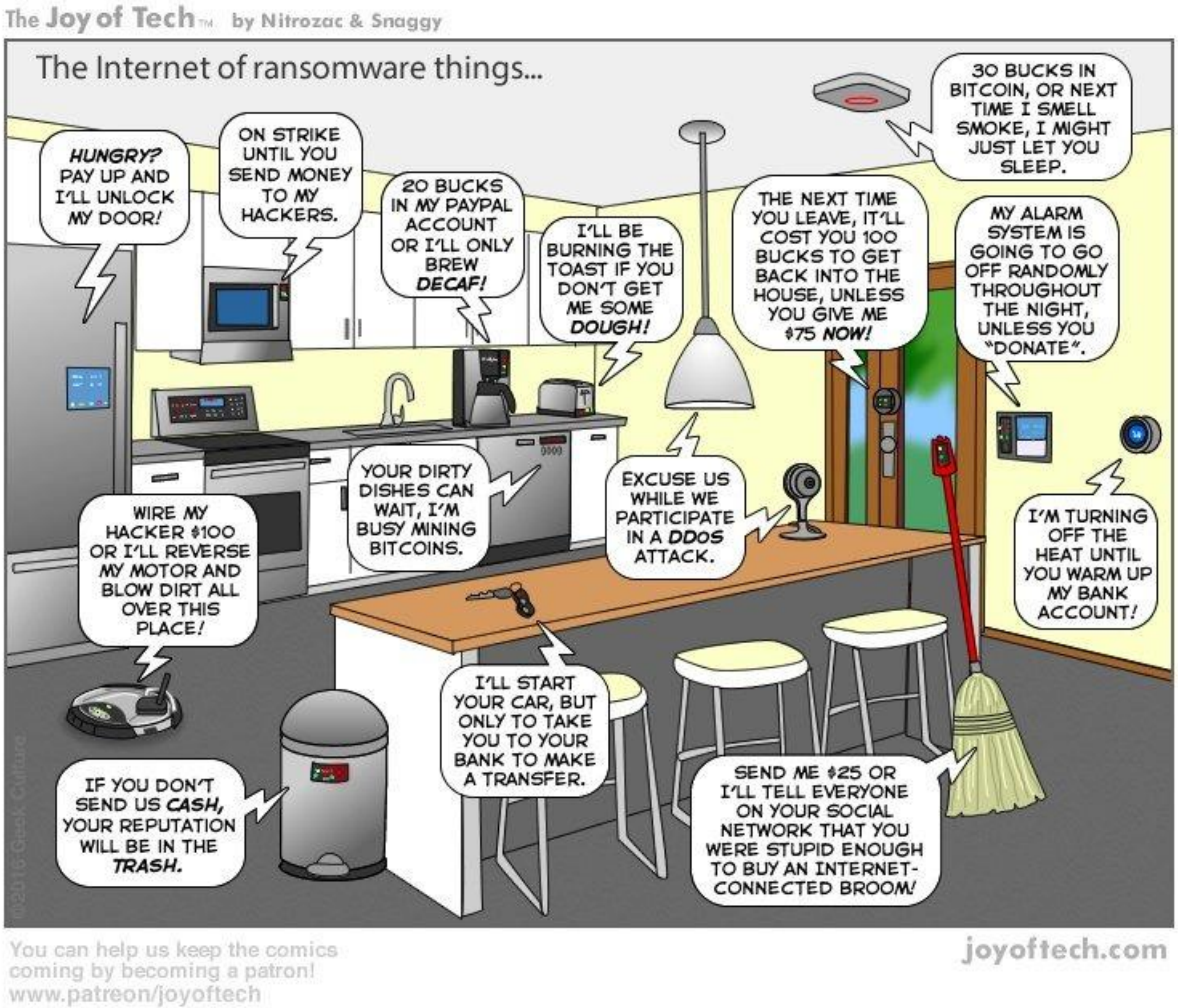
  From Wikipedia, the free encyclopedia

- *Sometimes there are workarounds:*
  - https://github.com/homebridge/**homebridge** for Apple HomeKit
  - Flash, root, and other solutions also exist for some devices.

The **security** side

- IoT systems are more sensible than most software-only things, because **things** can **affect the real-world**.
- Think before you expose your infrastructure over the web.
  - And, when you do it, **do it securely** (e.g. over VPN).
  - Try to not end on **Shodan**: https://2000.shodan.io
- IoT devices are not made to be long-lived.
  - Eventually, vulnerabilities will eventually appear, and no patch will be made.

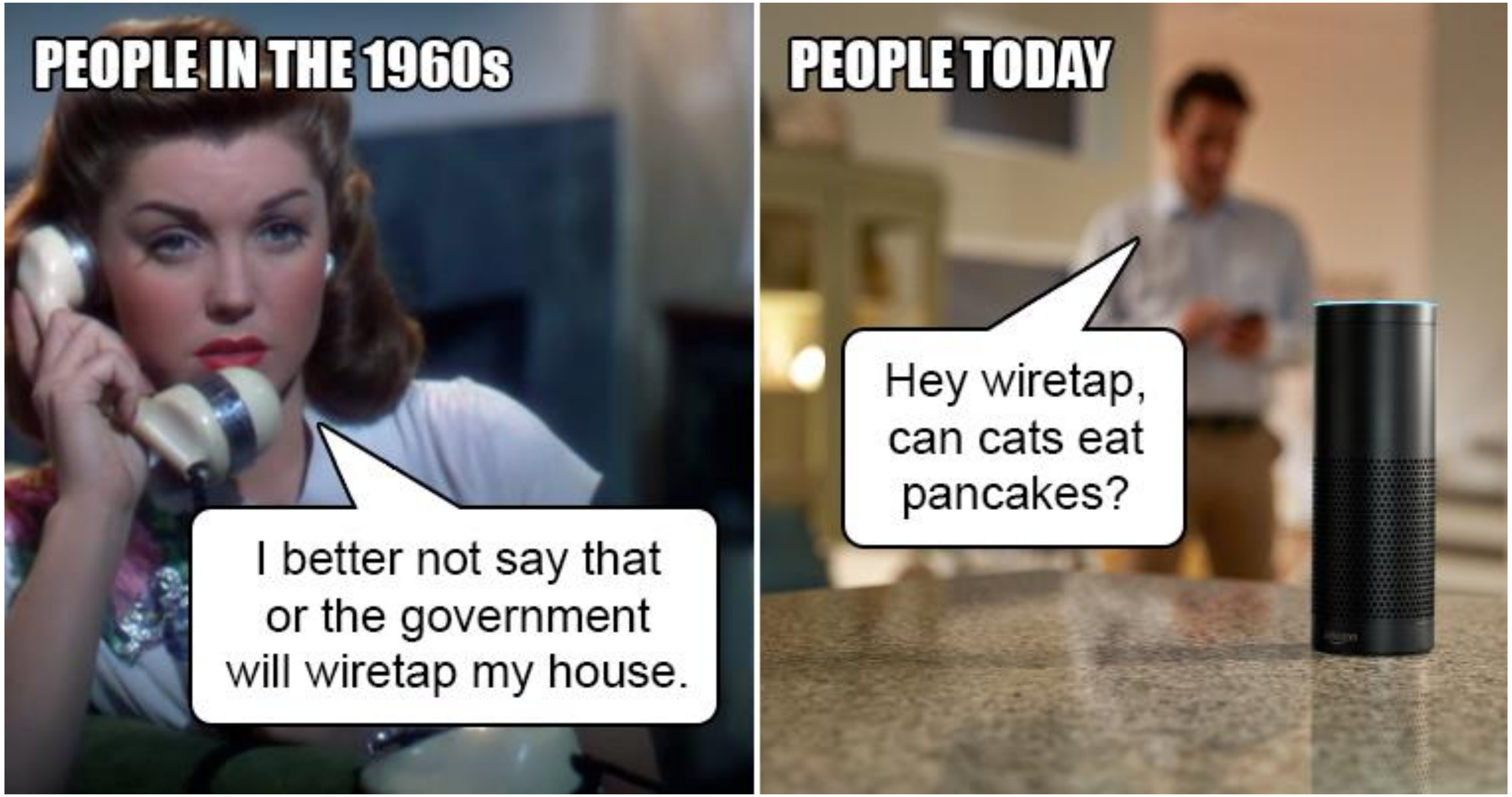
CodeWeek.


The Joy of Tech™ by Nitrozac & Snaggy
The Internet of ransomware things...
HUNGRY? PAY UP AND I'LL UNLOCK MY DOOR!
ON STRIKE UNTIL YOU SEND MONEY TO MY HACKERS.
20 BUCKS IN MY PAYPAL ACCOUNT OR I'LL ONLY BREW DECAF!
I'LL BE BURNING THE TOAST IF YOU DON'T GET ME SOME DOUGH!
THE NEXT TIME YOU LEAVE, IT'LL COST YOU 100 BUCKS TO GET BACK INTO THE HOUSE, UNLESS YOU GIVE ME $75 NOW!
30 BUCKS IN BITCOIN, OR NEXT TIME I SMELL SMOKE, I MIGHT JUST LET YOU SLEEP.
MY ALARM SYSTEM IS GOING TO GO OFF RANDOMLY THROUGHOUT THE NIGHT, UNLESS YOU "DONATE".
YOUR DIRTY DISHES CAN WAIT, I'M BUSY MINING BITCOINS.
EXCUSE US WHILE WE PARTICIPATE IN A DDoS ATTACK.
I'M TURNING OFF THE HEAT UNTIL YOU WARM UP MY BANK ACCOUNT!
WIRE MY HACKER $100 OR I'LL REVERSE MY MOTOR AND BLOW DIRT ALL OVER THIS PLACE!
I'LL START YOUR CAR, BUT ONLY TO TAKE YOU TO YOUR BANK TO MAKE A TRANSFER.
IF YOU DON'T SEND US CASH, YOUR REPUTATION WILL BE IN THE TRASH.
SEND ME $25 OR I'LL TELL EVERYONE ON YOUR SOCIAL NETWORK THAT YOU WERE STUPID ENOUGH TO BUY AN INTERNET-CONNECTED BROOM!
©2016 Geek Culture
You can help us keep the comics coming by becoming a patron! www.patreon/joyoftech
joyoftech.com

The **privacy** side

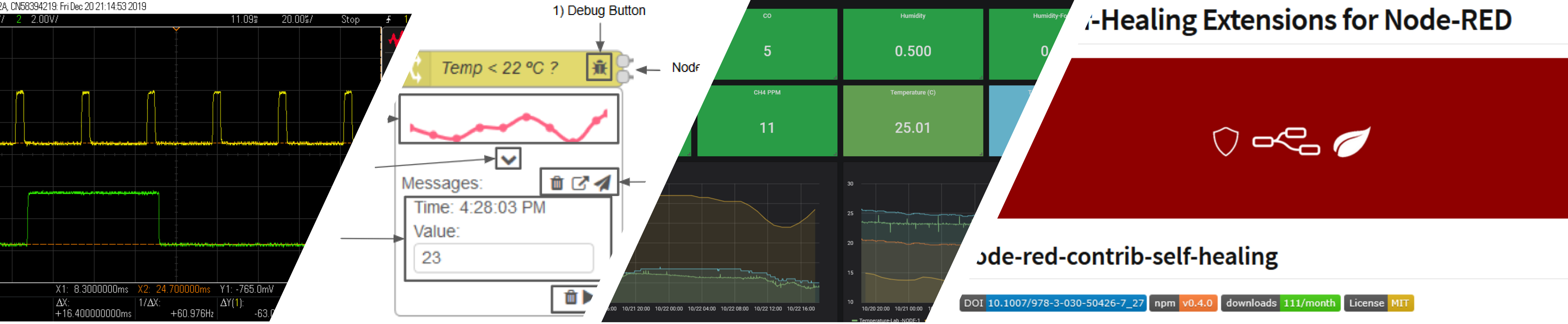
- When you buy a device, you can buy more than you want to.
  - Identification
  - Localization and Tracking
  - Profiling
  - Privacy-violating interaction and presentation
  - Lifecycle transition
  - Inventory attack
  - Linkage

- *Privacy in the Internet of Things: Threats and Challenges*
  - https://arxiv.org/pdf/1505.07683


PEOPLE IN THE 1960s
I better not say that or the government will wiretap my house.
PEOPLE TODAY
Hey wiretap, can cats eat pancakes?

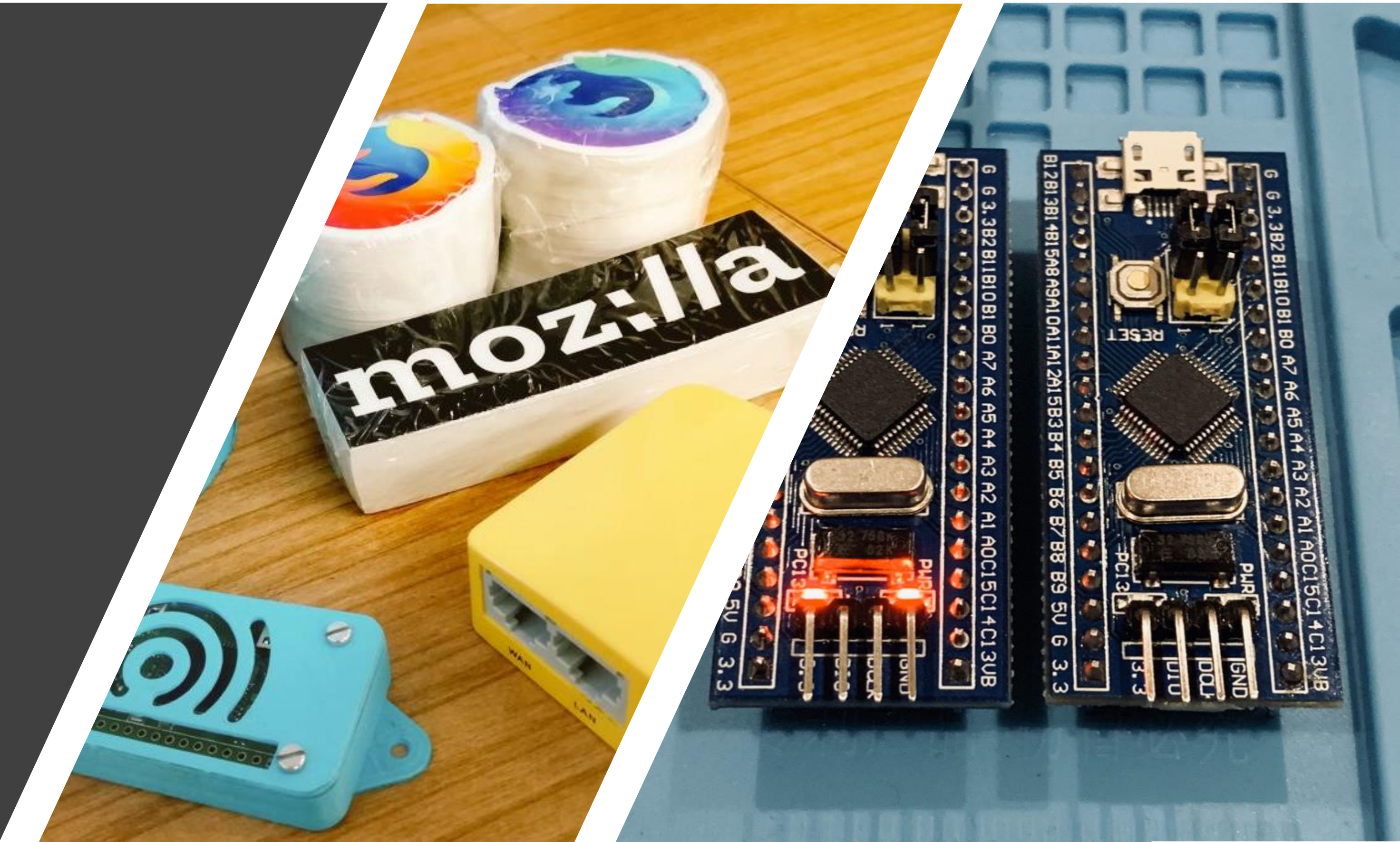
Read it *later*

- An IDE for programmable things: https://platformio.org/
- *The Internet of Risky Things: Trusting the Devices That Surround Us*
  - Book by Sean Smith
- awesome-iot list: https://github.com/HQarroum/awesome-iot
- WebThings for an open standard IoT: https://webthings.io/
- OWASP IoT Project: https://owasp.org/www-project-internet-of-things/
- Fun++:
  - https://twitter.com/internetofshit
  - https://www.shodan.io
  - https://www.iotvillage.org



Call for interest

- IoT research lines:
  - Software Engineering
  - Visual programming and low-code
  - Orchestration heterogeneous systems
  - Autonomic Computing (self-healing)
  - Fault-tolerance
  - Privacy and security
  - Embedded and retro computing



Find me @ http://jpdias.me , Twitter and jpmdias@fe.up.pt