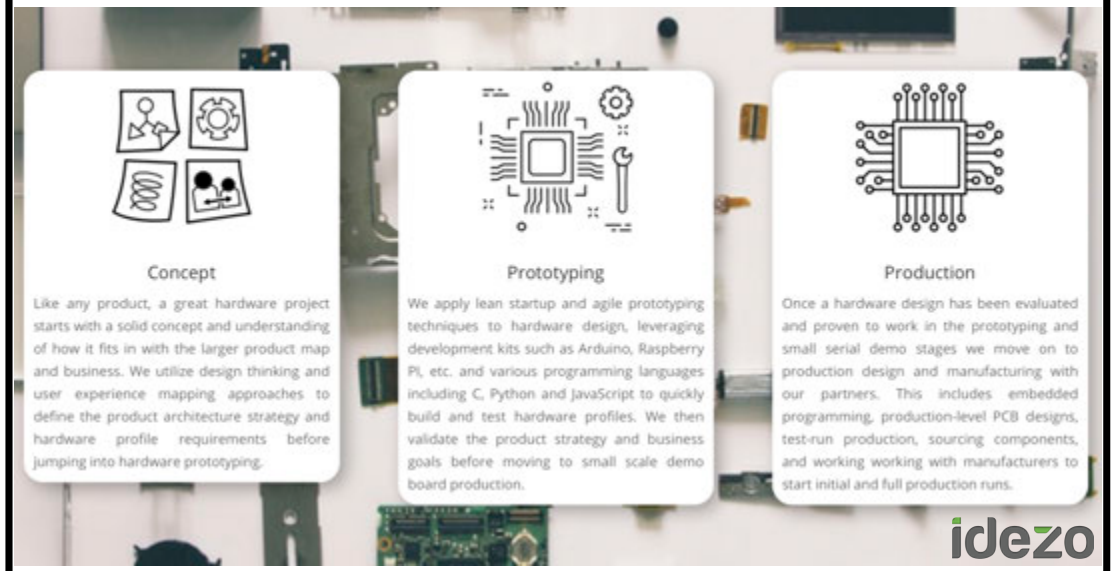
HANDS ON APPROACH TO

Data Science

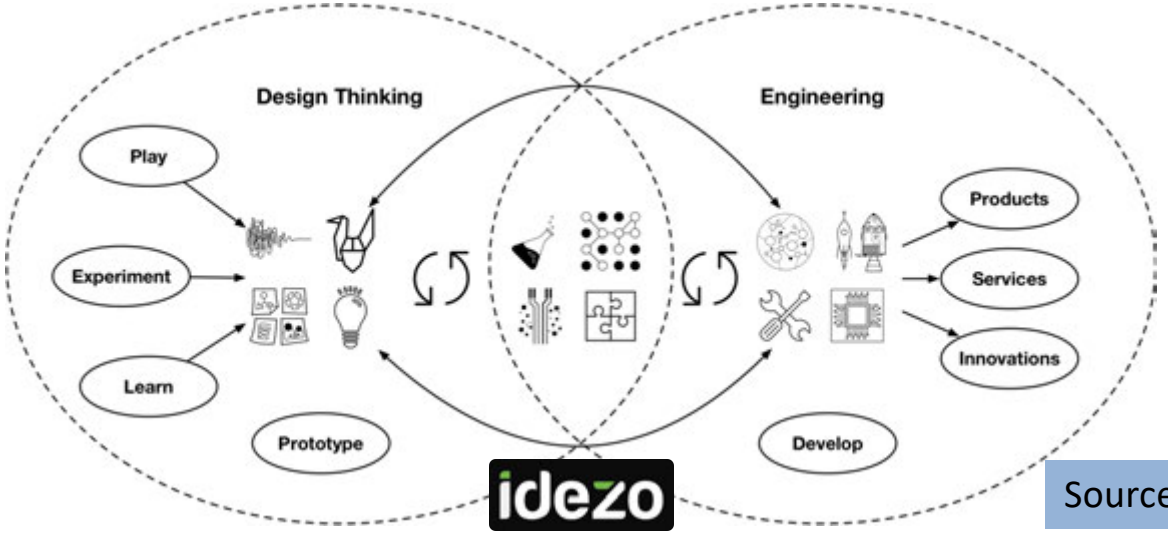
for (the)

IoT



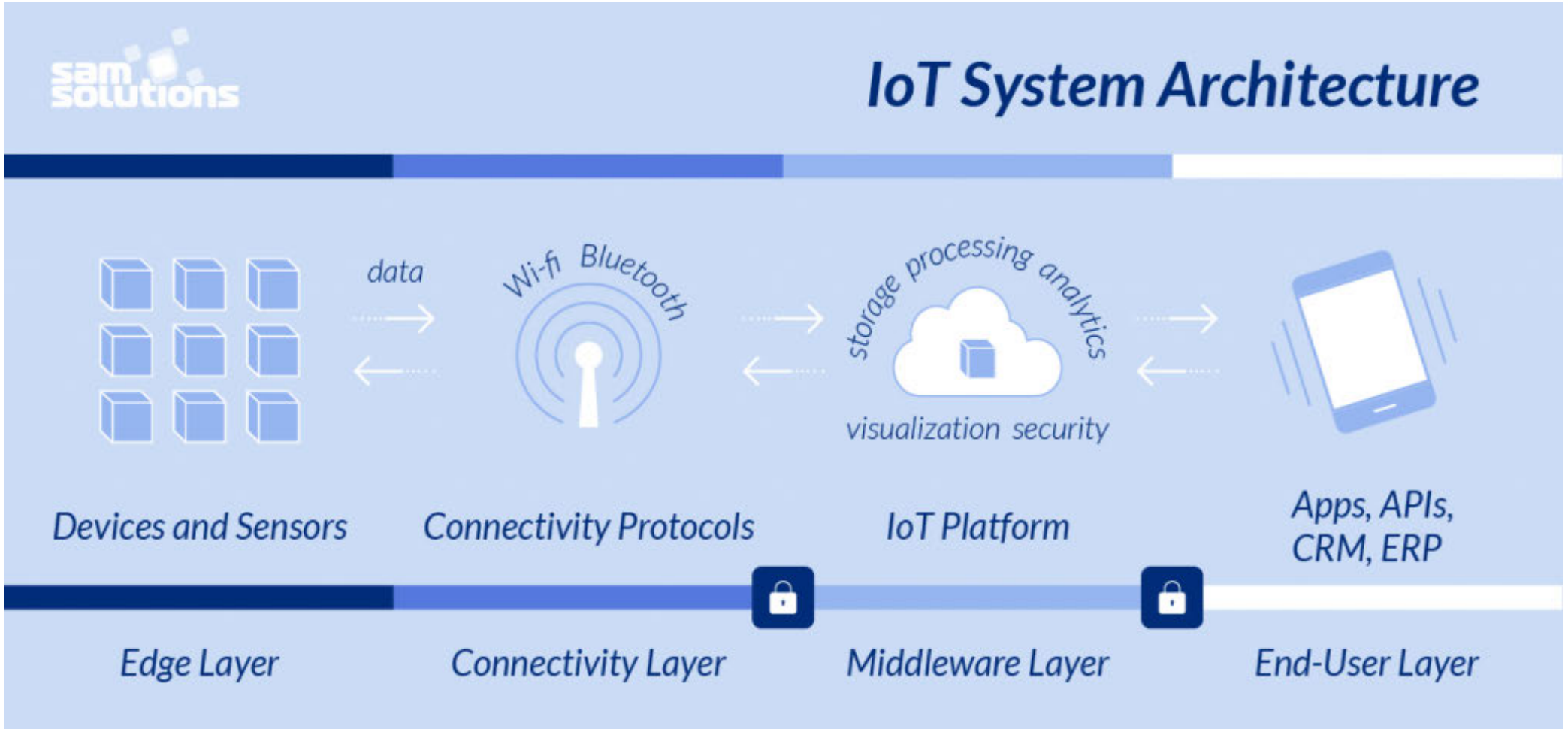
Rob van der Willigen

HANDS ON APPROACH TO DATA SCIENCE for (the) IoT



Source: https://idezo.ch

This Data Science Course was developed for keuzevak- program of the School of Communication, Media and Information Technology (CMI, at the Hogeschool Rotterdam **(Rotterdam University of Applied Sciences, RUAS)**.
If you find errors or omissions, please contact the author, Rob van der Willigen, at r.f.van.der.willigen@hr.nl. Materials of this course and code examples used will become available at:
https://github.com/robvdw/CMIDAT01K-DATA-SCIENCE-for-IOT



https://www.sam-solutions.com/blog/top-eight-internet-of-things-platforms/

The Internet of Things: Five Myths and Realities

Exploring the technology's true nature and capabilities

The Internet of Things (IoT) is a concept that describes a totally interconnected world. It's a world where devices of every shape and size are manufactured with "smart" capabilities that allow them to communicate and interact with other devices, exchange data, make autonomous decisions and perform useful tasks based on preset conditions.

It's a world where technology will make life richer, easier, safer and more comfortable. Or is it?

Like many emerging technologies, the Internet of Things is vulnerable to hype and exaggeration. To some people, it sounds like a vague and distant prospect; to others, it sounds threatening and dangerous.

This executive brief will explore five of the most common myths associated with the Internet of Things and shed some light on the true nature of the technology and its capabilities.

freescale

Any new technology involves a certain amount of uncertainty and business risk. In the case of the Internet of Things, however, many of the risks have been exaggerated or misrepresented.

3

Will the Internet of Things create serious privacy and security challenges?

Security and privacy are major concerns—and addressing these concerns is a top priority.

Inevitably, discussions of an Internet of Things evoke images of a future dystopia where personal privacy—and personal freedom—is a thing of the past. In addition, as a seemingly endless stream of data-breach incidents and hacker attacks proves, stolen or misused digital data is already a very real, and rapidly growing, concern.

These are legitimate concerns. New technology often carries the potential for misuse and mischief, and it's vital to address the problem before it hinders personal privacy and security, innovation or economic growth.

Manufacturers, standards organizations and policy-makers are already responding on several levels:

- Existing privacy enhancing technology (PET) infrastructure standards, such as virtual private networks and DNS Security Extensions, that can be easily adapted to protect sensitive data as it moves between devices or through the cloud.
- Government regulations such as the U.S. Health Insurance Portability and Accountability Act (HIPAA) that mandate the protection of sensitive personal information—no matter where it is stored or how it is transmitted.
- Other security frameworks proposing DTLS secured light CoAP running on UDP, which has also been adopted by ETSI.
- As well as many other activities by multiple members of the ecosystem.

Course Setup

HOGESCHOOL ROTTERDAM

Lesson 01: week 02	**Discovering the IoT Data Science Domain**
Lesson 02: week 03	**Defining project requirements**
	+ Cost calculation/estimate
Lesson 03: week 04	**Learn to write code**
Lesson 04: week 05	**Data Science: How to start your own IoT Project**
Lesson 05: week 06	**IoT Platforms & MiddleWare**
Lesson 06: week 07	**Core IoT Concepts + setting-up your IoT-device**
Lesson 07: week 08	**Summary + Q & A**

Week 09/10: FEEDBACK + GRADING

lesson six

HOGESCHOOL ROTTERDAM

HOGESCHOOL ROTTERDAM

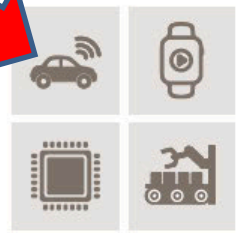
IoT Core-Concepts

IoT Core Concepts

HOGESCHOOL ROTTERDAM

Devices

- Sensors
- Actuators
- Machines



Business Model

- Outcome driven
- Product as a service
- Leverage existing investment

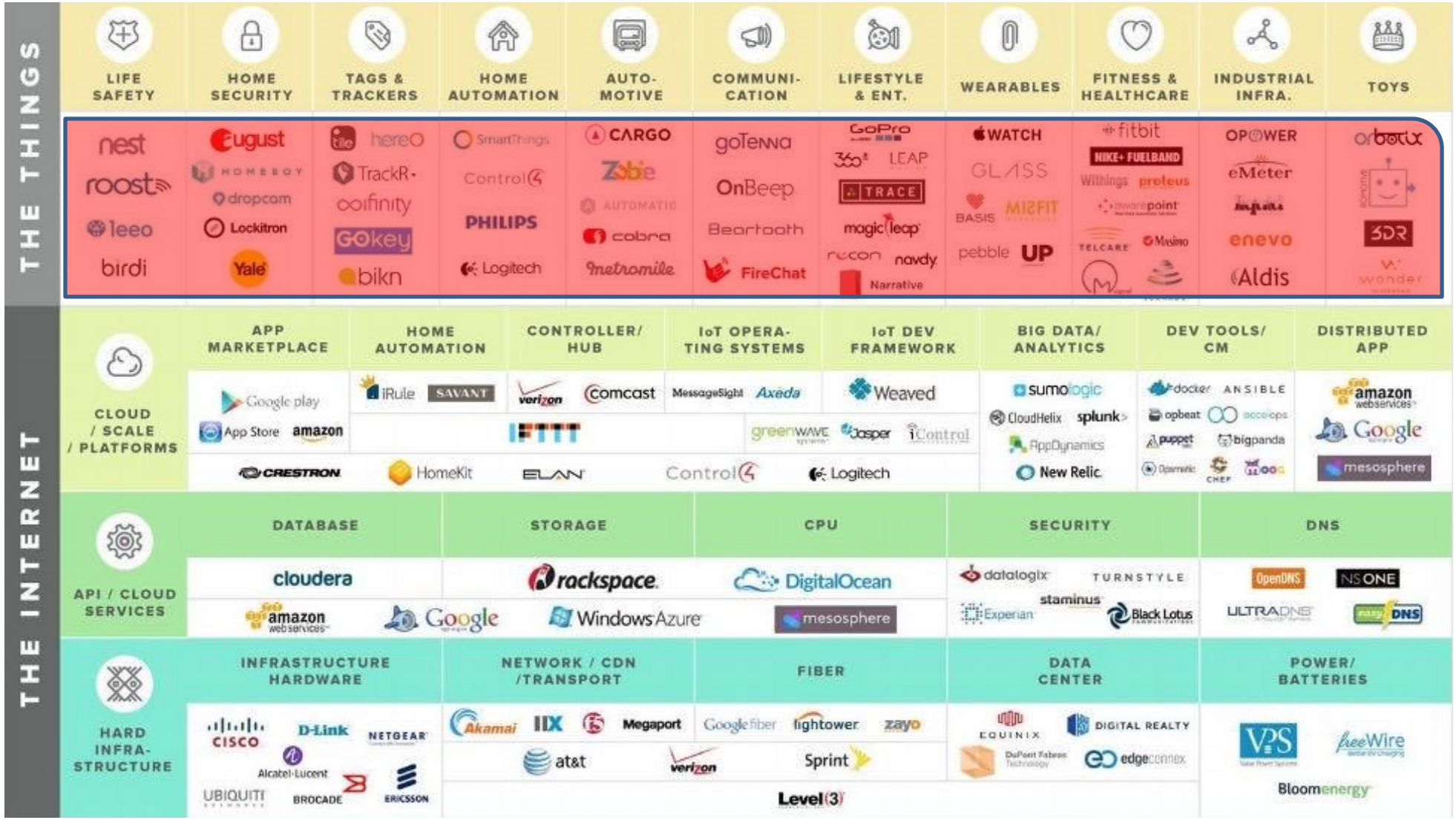
Connectivity

- Network
- Protocol
- Security

Platform

- Storage
- Analytics
- Visualization
- Integration

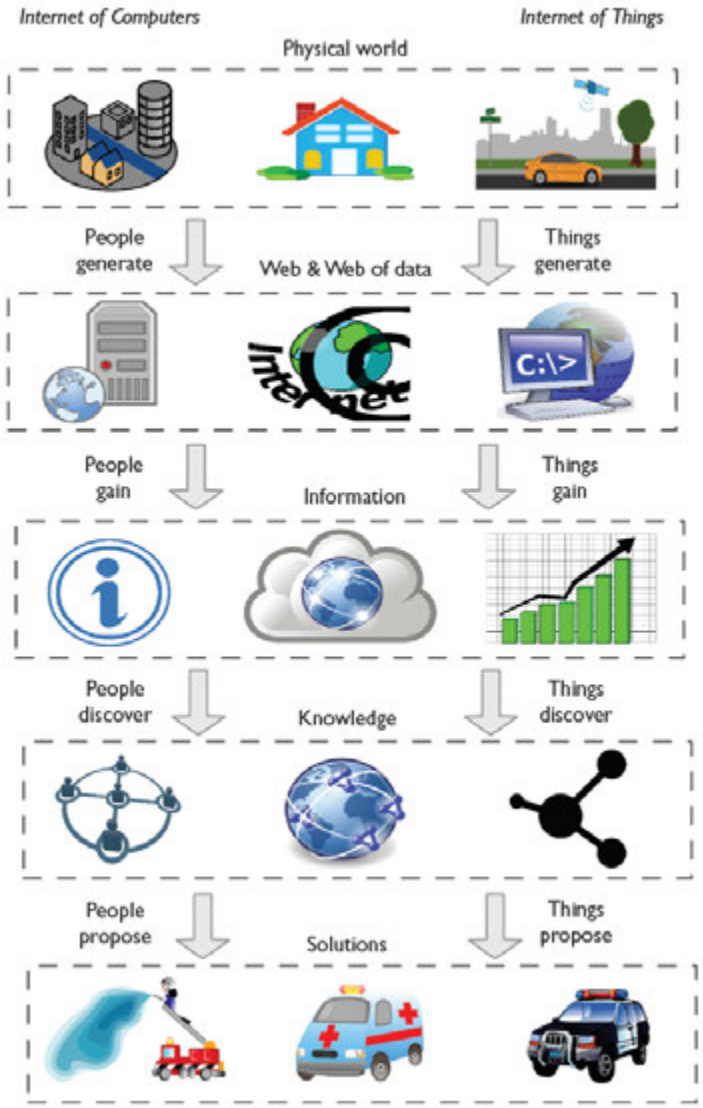
IoT Landscape



World Wide Web (WWW) vs IoT

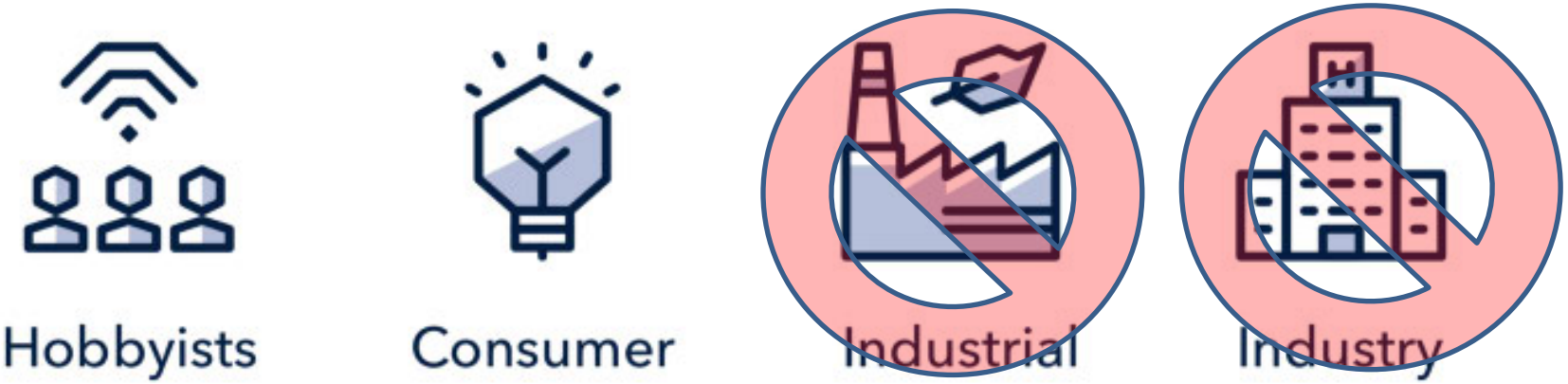
WWW:
In the Internet of Computers (WWW), the main data producers and consumers are human beings.

IoT:
The main actors become things, where things are the majority of data producers and consumers.



Qin, Y., Sheng, Q. Z., & Curry, E. (2015). Matching over linked data streams in the internet of things. *IEEE Internet Computing*, *19*(3), 21-27.

IoT Platform + User-Types

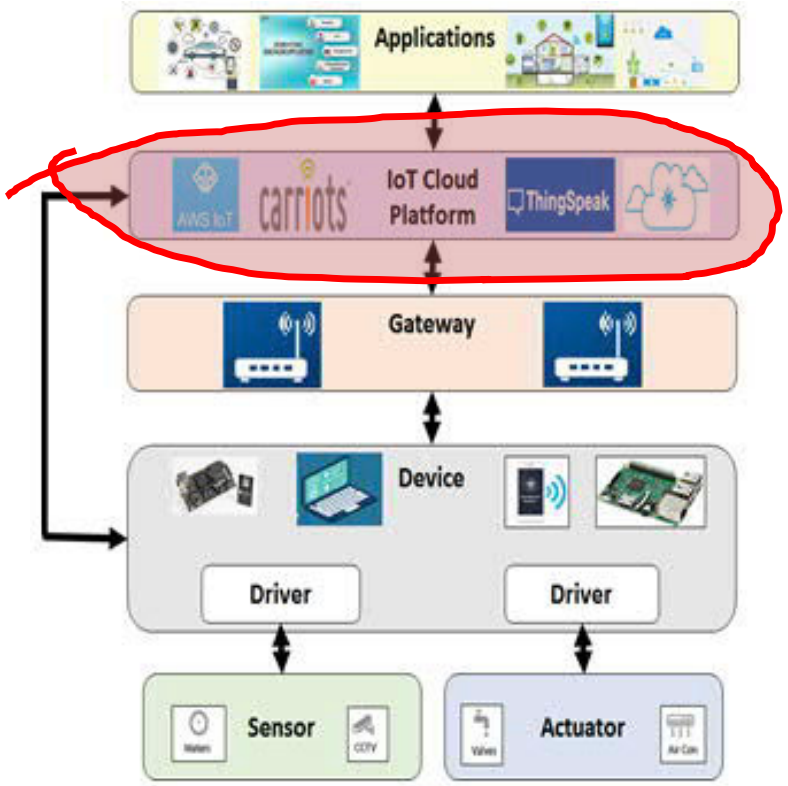


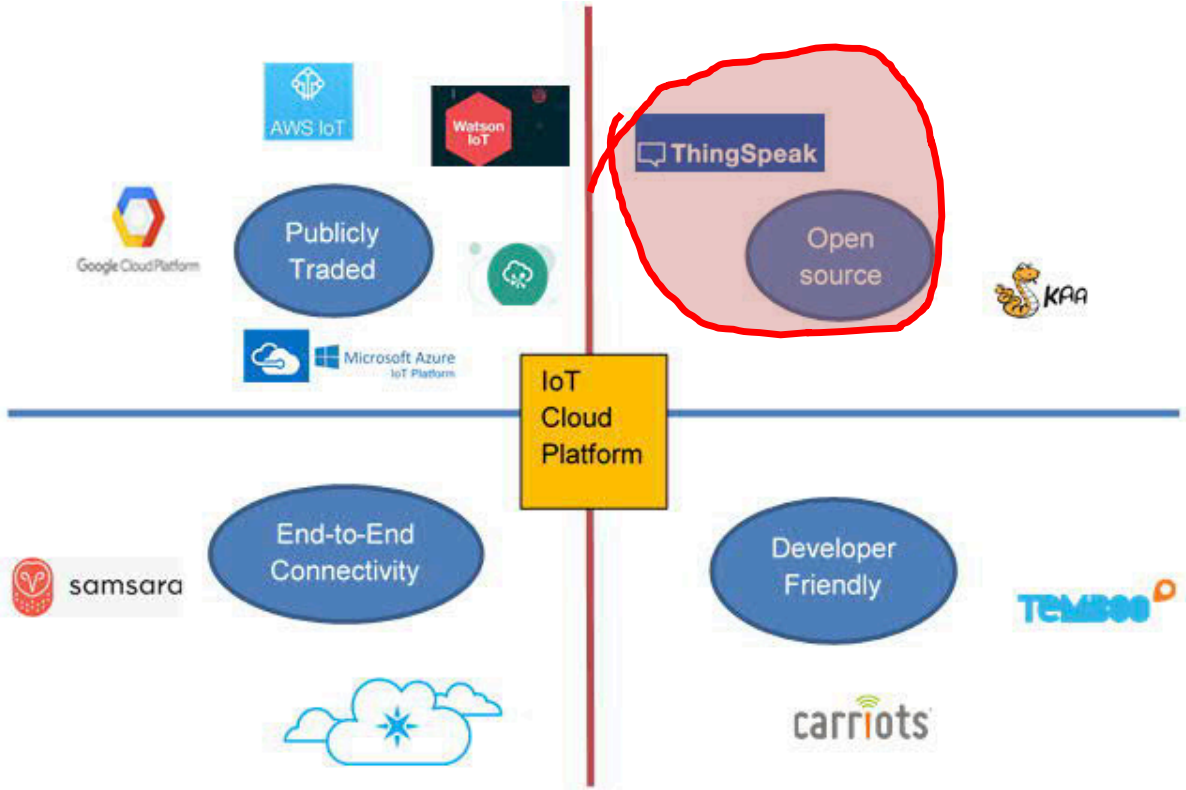
The Internet of Things is a state where the things on the face of the earth connect to the internet and start talking to each other. Things here can be electronic, electrical, mechanical, or electro-mechanical objects.

The number of connected things is projected to grow at an annual compound rate of 23.1% between 2014 to 2020, reaching 50.1 billion things in 2020. [1]

IoT Platforms == MiddleWare

HOGESCHOOL ROTTERDAM





http://www.thetips4you.com/iot-best-open-source-applications-open-source-industrial-iot-platform

IoT Application Domains

Consumers

- Connected gadgets
- Appliances
- Wearables
- Domestic robots
- Participatory sensing
- Social Web of Things

Automotive Transport

- Autonomous vehicles
- Multimodal transport
- Logistics
- Traffic management

Retail Banking

- Micro payments
- Retail logistics
- Product life-cycle info
- Shopping assistance

Environmental

- Pollution
- Air, Water, Soil
- Weather, Climate
- Noise
- Erosion, fires

Infrastructures

- Buildings, Homes
- Roads, Rail

Utilities

- Smart Grid
- Water management
- Gas, Oil, Renewables
- Waste management
- Heating, Cooling

Health Well-being

- Remote monitoring
- Assisted living
- Behavioral change
- Treatment compliance
- Sports, Fitness

Smart Cities

- Integrated environments
- Optimized operations
- Convenience
- Socioeconomics
- Sustainability
- Inclusive living

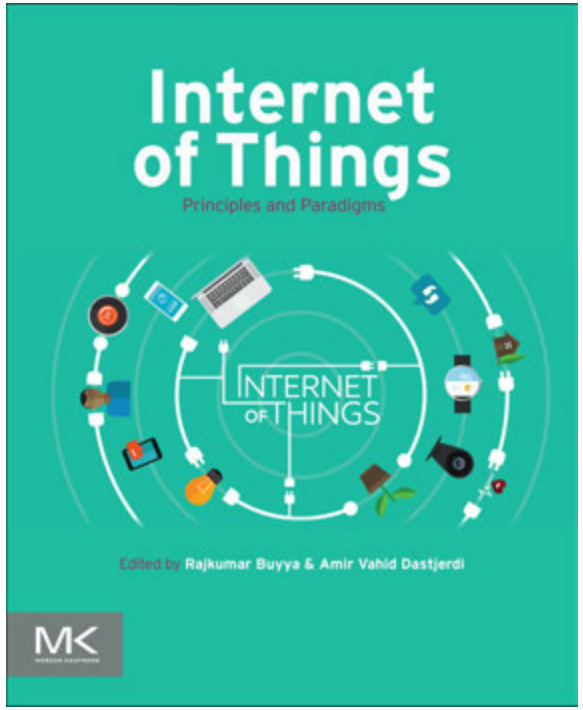
Process industries

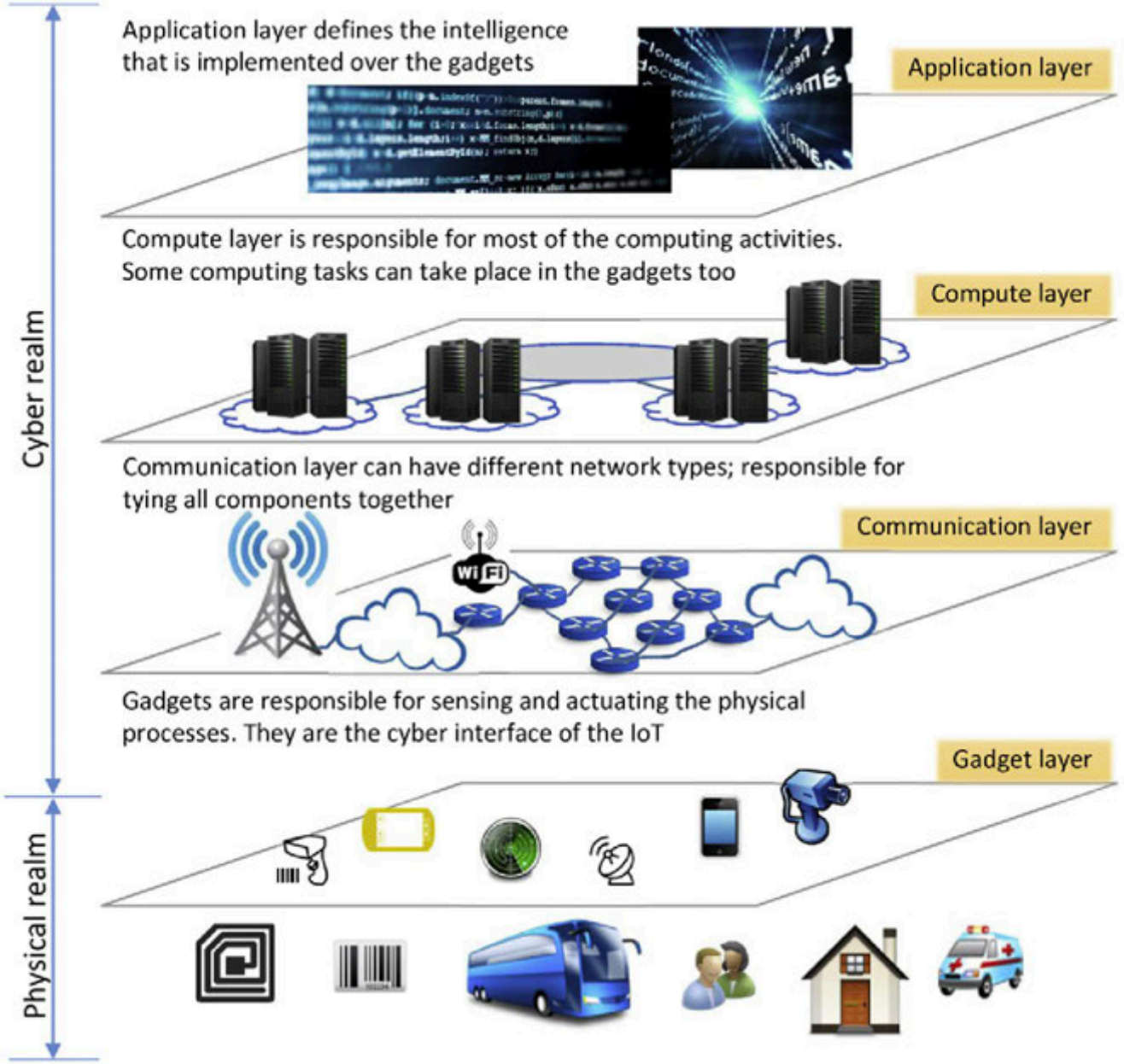
- Robotics
- Manufacturing
- Natural resources
- Remote operations
- Automation
- Heavy machinery

Agriculture

- Forestry
- Crops and farming
- Urban agriculture
- Livestock, Fisheries

IoT Layers


Internet
of Things
Principles and Paradigms
INTERNET OF THINGS
Edited by Rajkumar Buyya & Amir Vahid Dastjerdi
MK


Cyber realm
Physical realm
Application layer defines the intelligence that is implemented over the gadgets
Application layer
Compute layer is responsible for most of the computing activities. Some computing tasks can take place in the gadgets too
Compute layer
Communication layer can have different network types; responsible for tying all components together
Communication layer
Wi Fi
Gadgets are responsible for sensing and actuating the physical processes. They are the cyber interface of the IoT
Gadget layer

Technologies enabling IoT

WIRELESS TECHNOLOGIES

802.11, 802.15.4, Zigbee, Bluetooth, BLE, CDMA, GSM, LTE

TCP/IP PROTOCOLS

Sockets, IPv4, IPv6, TCP, UDP, ICMP, QOS, SNMP, IMAP, POP3, IPMI, etc.

AUTHORIZATION & AUTHENTICATION TECHNOLOGIES

Oauth2, PAM, LDAP

NETWORK TECHNOLOGIES

Ethernet, CAN, rs485/rs422/rs232, 1-wire, i2c, SPI, ModBus/MudBusRTU, IPMI, iSCSI

MOBILE TECHNOLOGIES

Android SDK, Qt, iOS SDK, Objective C, Java, Swift

PLATFORMS AND CPU ARCHITECTURES

ARM, X86, PowerPC, AVR, PIC

WEB SERVICE TECHNOLOGIES

SOAP, REST, WSDL, XML, JSON, UDDI, WebSockets

PROTOCOL TECHNOLOGIES

HTTP, JMS, AMQP, D-Bus

PROGRAM / SCRIPTING LANGUAGES

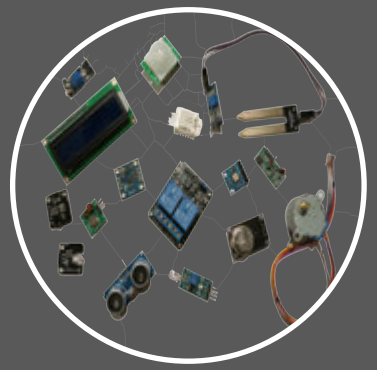
Java, C/C++, C#, JavaScript, Ruby, Groovy, Python, Tcl/Tk, ASM, Bash

https://doi.org/10.15779/Z38WW0V

https://doi.org/10.1016/j.eswa.2019.05.014

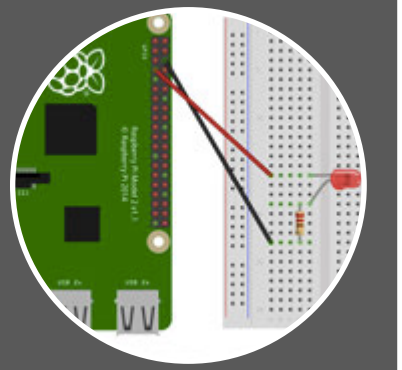
Engineering a IoT Data Pipeline

HOGESCHOOL ROTTERDAM



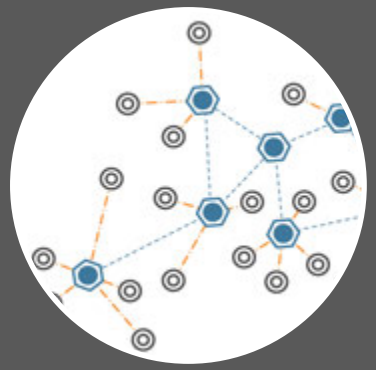
Sensors &

Actuators



Micro-Processors/ Controllers

(Development Boards)



Network / Protocols

Platforms,

Cloud &

Data Analytics

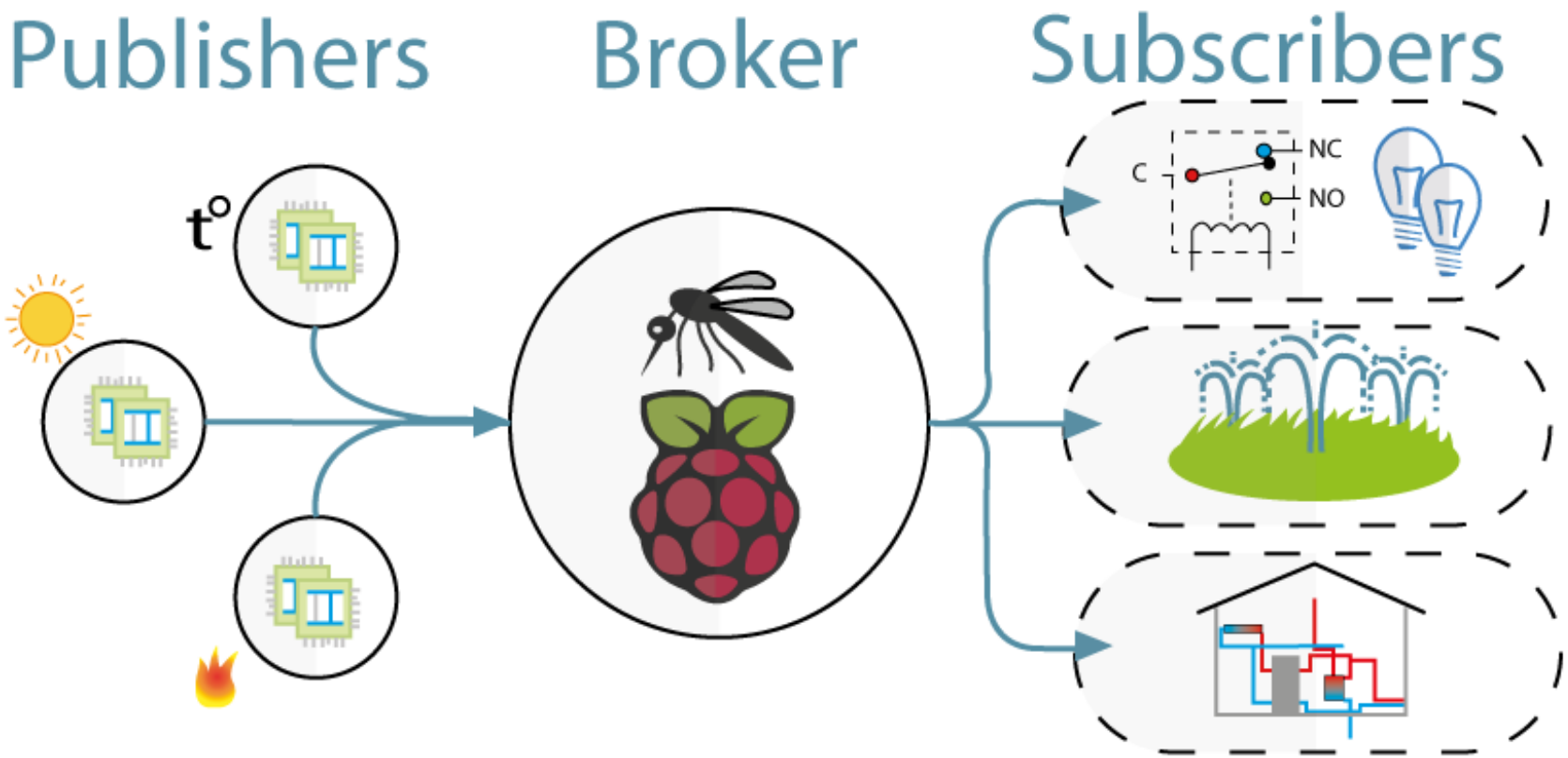
IoT Communication

HOGESCHOOL ROTTERDAM

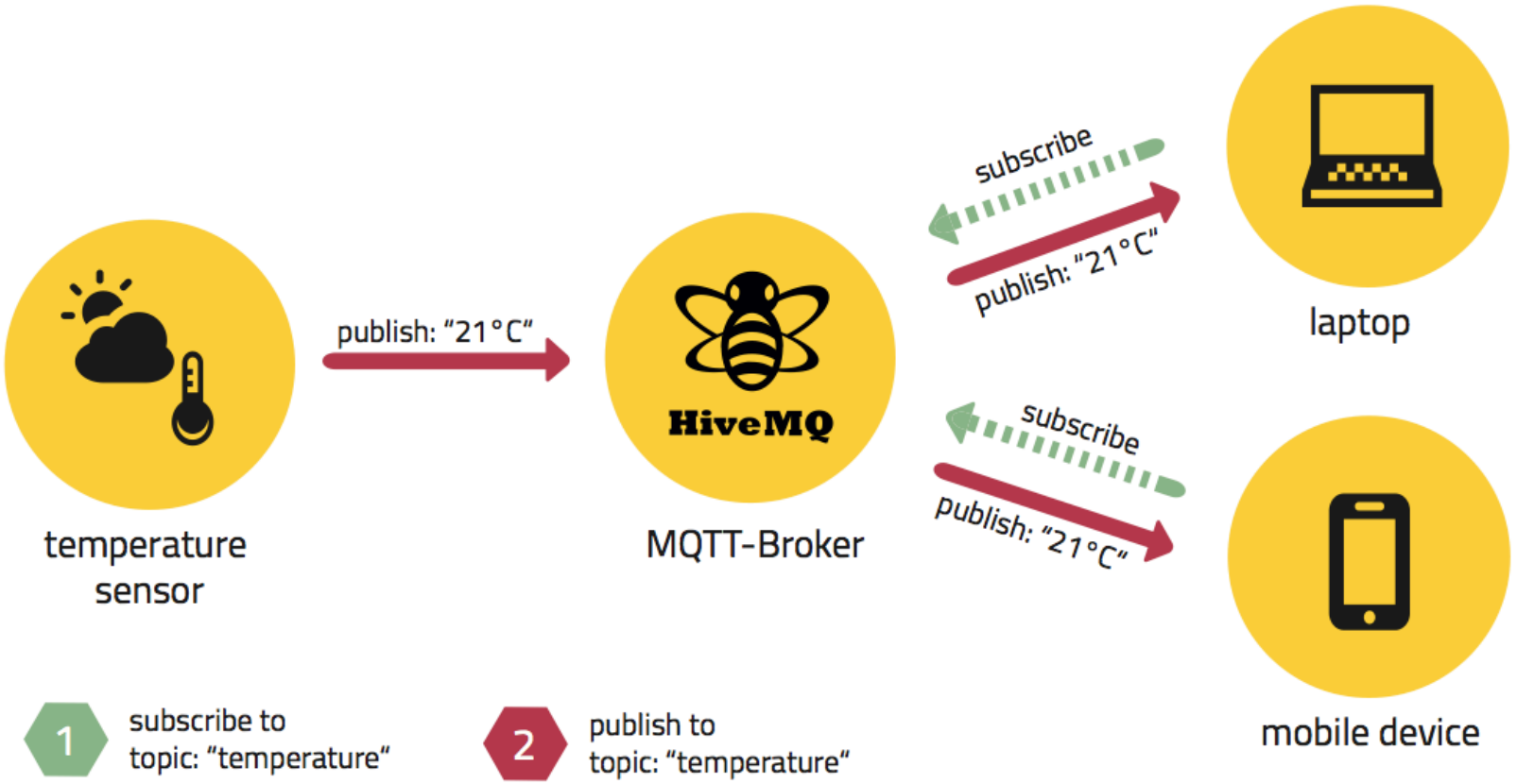
The Broker. The broker acts as a gateway; it receives messages from a publisher (a client) and delivers the messages to a subscriber (another client). Brokers are sometimes referred to as *servers*.

The Subscriber. The subscriber declares its topics of interest to the broker, and the broker sends messages published to those topics.

The Publisher. The publisher sends messages to the broker using a name- space or a topic name, and the broker forwards the messages to the respective subscribers.



https://raspberry-valley.azurewebsites.net/Mosquitto/



https://www.eclipse.org/community/eclipse_newsletter/2014/october/article2.php

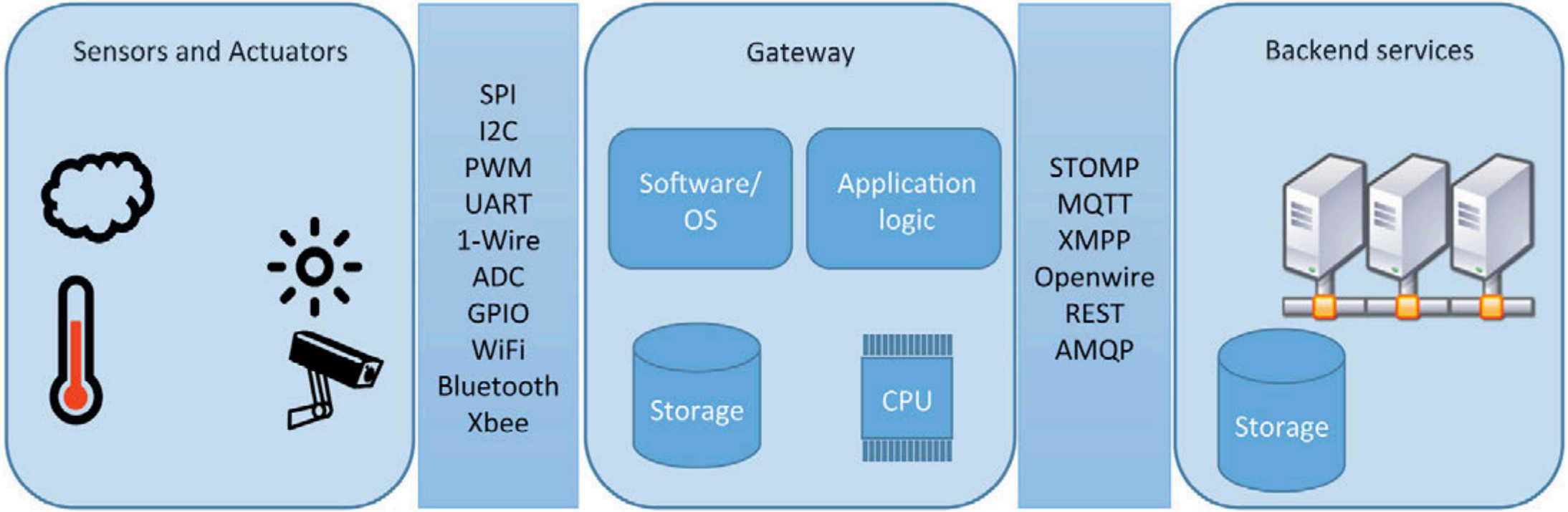
IoT-device communication protocols

Protocol Name	Transport Protocol	Messaging Model	Security	Best-Use Cases	Architecture
AMPQ	TCP	Publish/Subscribe	High-Optional	Enterprise integration	P2P
CoAP	UDP	Request/Response	Medium-Optional	Utility field	Tree
DDS	UDP	Publish/Subscribe and Request/Response	High-Optional	Military	Bus
MQTT	TCP	Publish/Subscribe and Request/Response	Medium-Optional	IoT messaging	Tree
UPnP	—	Publish/Subscribe and Request/Response	None	Consumer	P2P
XMPP	TCP	Publish/Subscribe and Request/Response	High-Compulsory	Remote management	Client server
ZeroMQ	UDP	Publish/Subscribe and Request/Response	High-Optional	CERN	P2P

IoT communication Protocols

HOGESCHOOL ROTTERDAM

Publisher ***Broker***

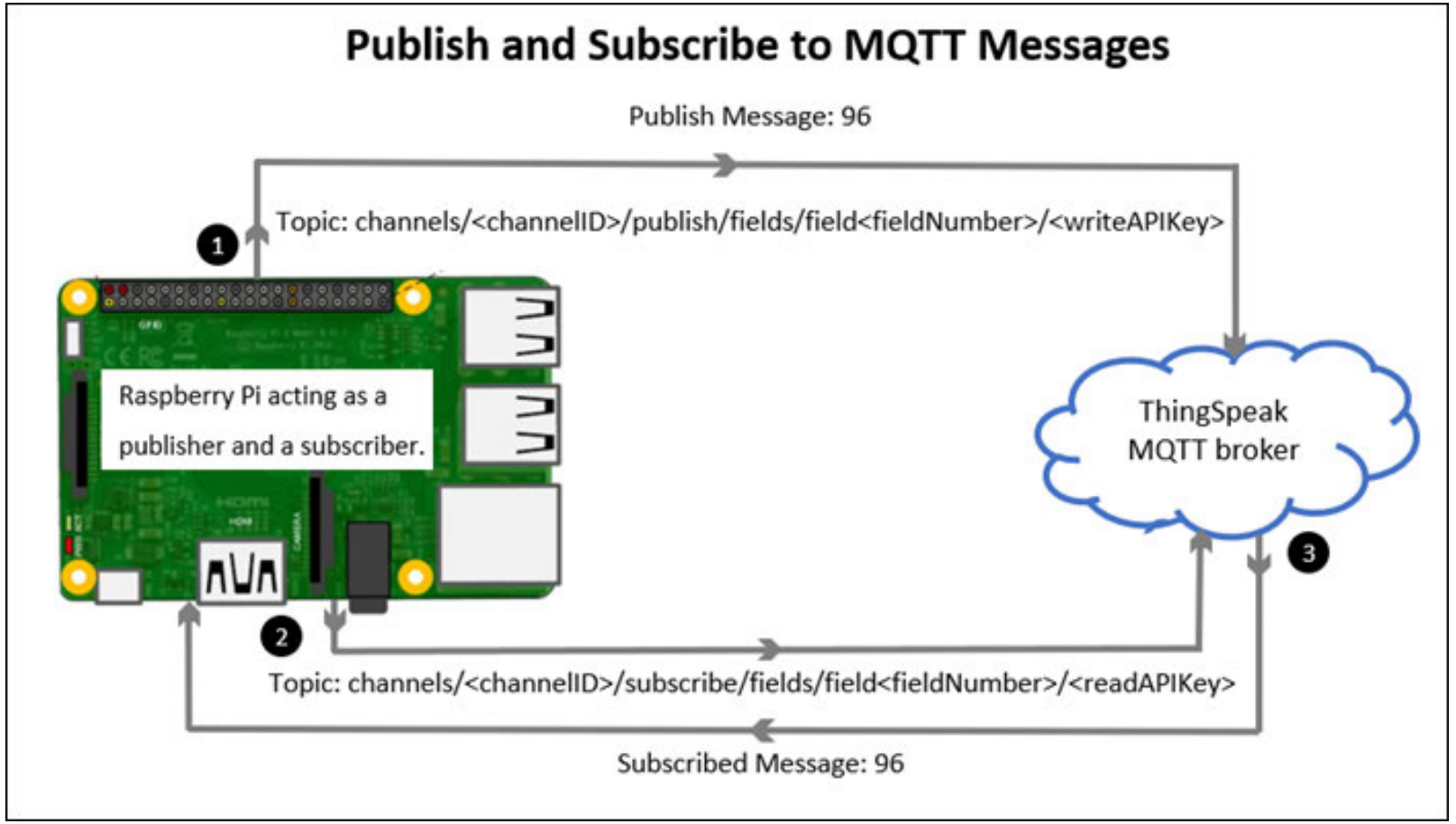


SENSOR TO GATEWAY COMMUNICATION

Setting Up Your IoT Device

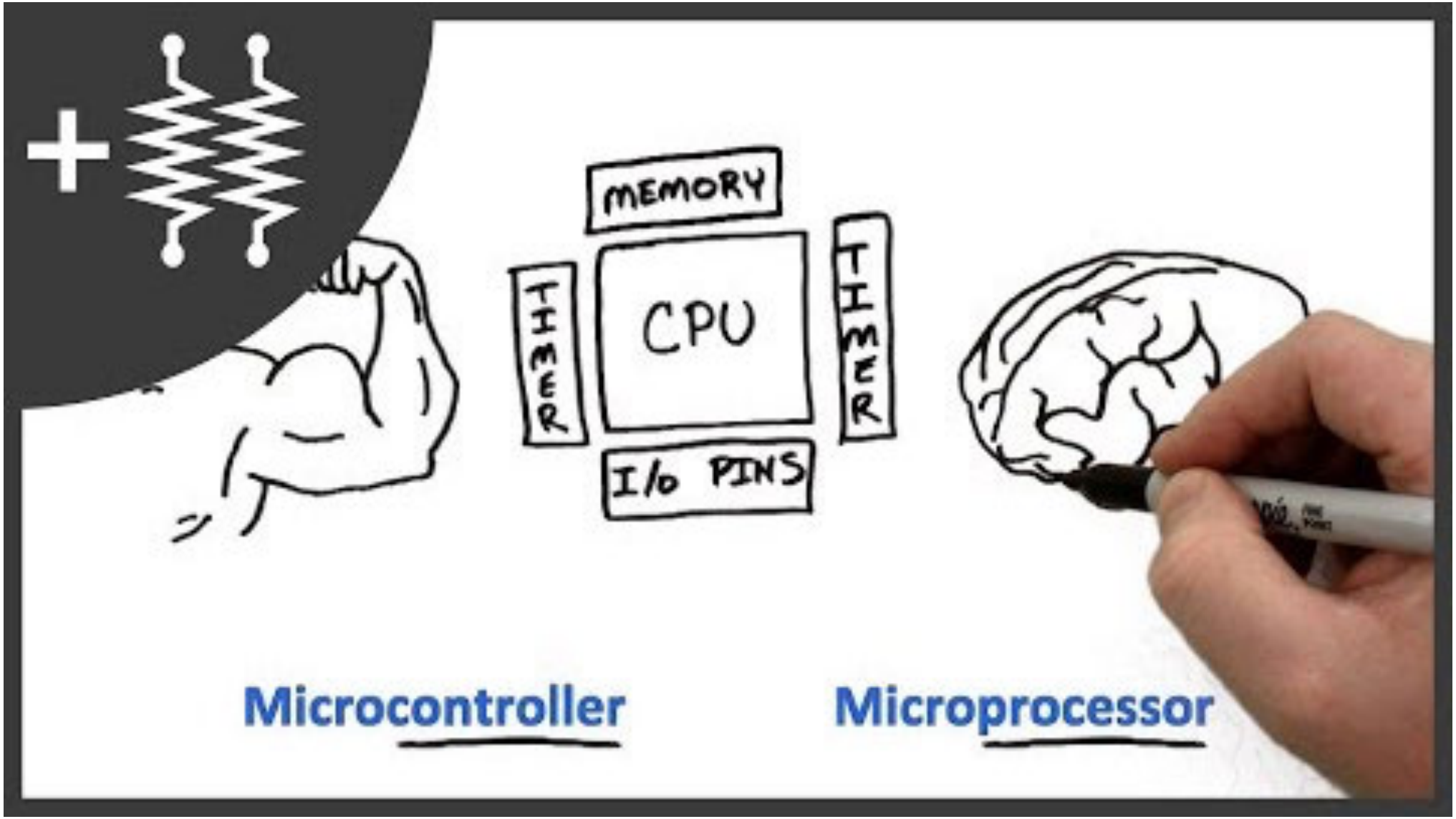
HOGESCHOOL ROTTERDAM

{IoT device == micro-controller== Rasperry Pi}



HOGESCHOOL ROTTERDAM

Micro-controller/processor Core-Concepts

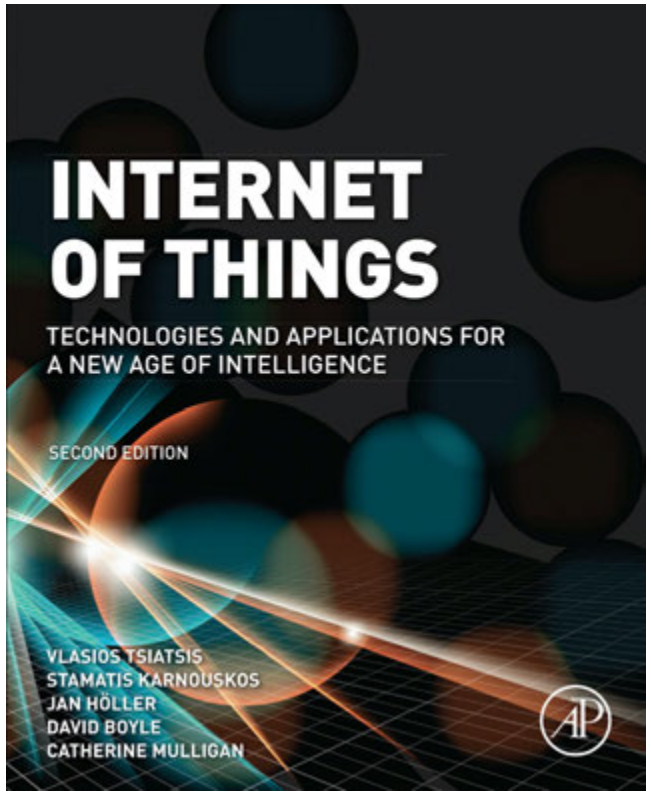


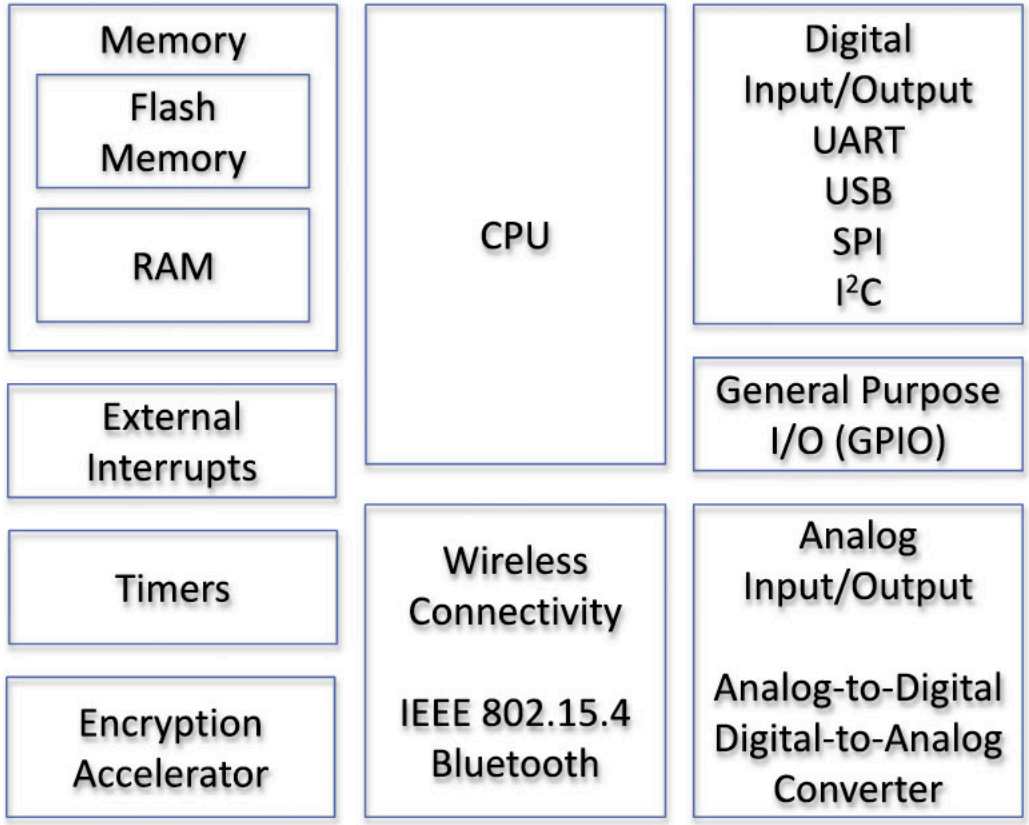
https://www.youtube.com/watch?v=7vhvnaWUZjE

HOGESCHOOL ROTTERDAM

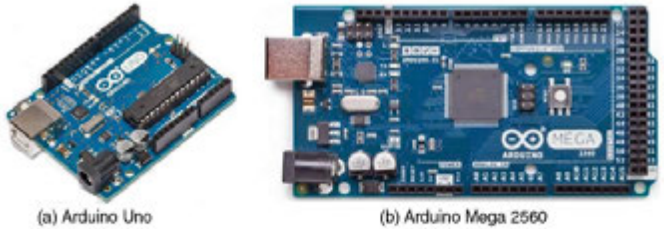
Microprocessor	Microcontroller
Only CPU; other peripherals are interfaced via the CPU bus	CPU + (Flash) Memory + RAM + ROM + I/O + Timers + UARTS + ADC + DAC and so on on a single chip
General purpose	Single purpose
High processing power	Low processing power
Power hungry	Can work on a battery

Micro-controller/processor Internal Architecture

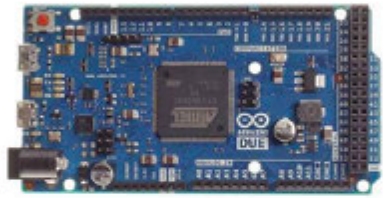




(micro)Controlers (hardware requirements)


(a) Arduino Uno

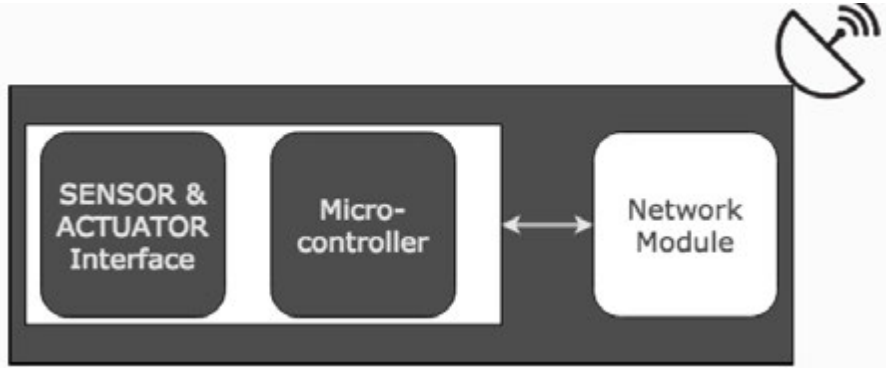
(b) Arduino Mega 2560


(c) Arduino Due


(d) Arduino Yún

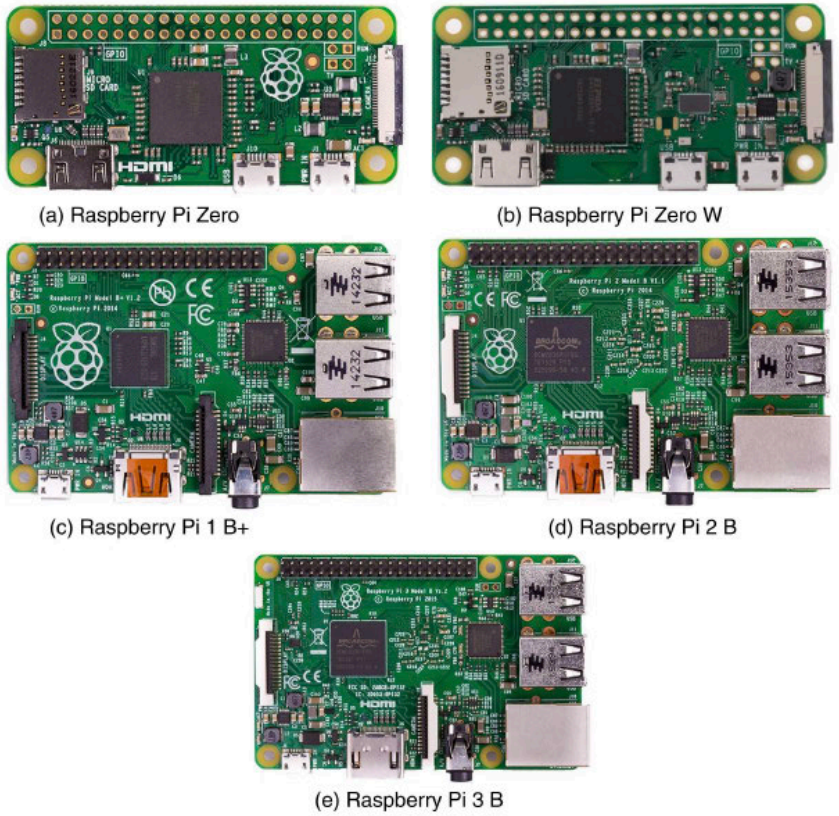

(e) Arduino 101


(f) Arduino MKR1000



	Microcontroller			Memory/storage		
Board name	MCU chip	Processor bit	Clock speed	RAM	EEPROM (kb)	Flash memory (kb)
Uno	ATmega328	8	16 MHz	2 KB	1	32
Mega 2560	ATmega2560	8	16 MHZ	8 KB	4	250
Due	AT91SAM3X8E	32	84 MHz	96 KB	–	512
Yún	ATmega32U4	8	16 MHz	2.5 KB	1	32
	AR9331	8	16 MHZ	64 MB, 2.5 KB	1	16
101	Intel Quark SE SoC	32	32 MHz	24 KB	–	196
MKR1000	ATSAMW25 SoC	32	48 MHz, 32.768 KHz	32 KB	–	256

(micro)Controlers (hardware requirements)



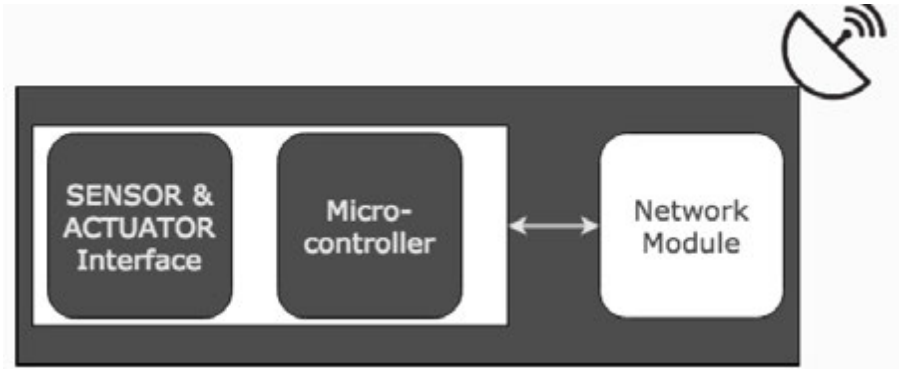
(a) Raspberry Pi Zero

(b) Raspberry Pi Zero W

(c) Raspberry Pi 1 B+

(d) Raspberry Pi 2 B

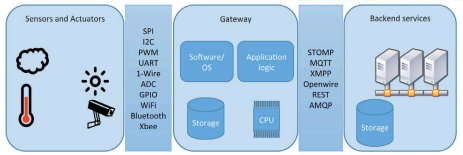
(e) Raspberry Pi 3 B



Raspberry Pi	Onboard connectivity	GPIO pins	USB ports	Display ports/ interfaces	Camera port	Other hardware interfaces
Zero	–	40	1 mini	Mini-HDMI	CSI	UART, SPI, I2C
Zero W	Wi-Fi, Bluetooth 4.1, BLE	40	1 mini	Mini-HDMI	CSI	UART, SPI, I2C
1 B+	Ethernet	40	4	HDMI, DSI, 3.5 mm Video Jack	CSI	UART, SPI, I2C
2 B	Ethernet	40	4	HDMI, DSI, 3.5 mm Video Jack	CSI	UART, SPI, I2C
3 B	Ethernet, Wi-Fi, Bluetooth 4.1, BLE	40	4	HDMI, DSI, 3.5 mm Video Jack	CSI	UART, SPI, I2C

(connecting) SENSORS

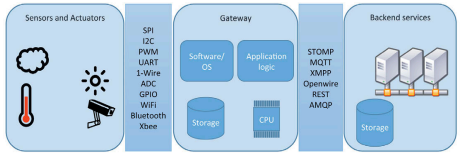
Sensor	Interface	Processor/Hardware	Power
Wind vane	Analog	Switched resistors	—
Anemometer	GPIO interrupt	Reed switch	—
Rain gauge	GPIO interrupt	Reed switch	—
UV intensity	Analog	ML8511	1,000 µW
Humidity and temperature	2-wire serial	SHT15	80 µW
Barometric pressure	I^2C	BMP180	30 µW
Luminosity	I^2C	TSL2561	720 µW
Lightning sensor	SPI + GPIO interrupt	AS3935	100 µW
Geiger counter	RS-235	LND712	147,000 µW
Weather board	I^2C	Si7020	540 µW
Weather board	I^2C	BMP180	30 µW
Weather board	I^2C	Si1132	1,419 µW



Sensor/Actuator IoT
Microcontroller interfacing protocols

There are multiple protocols with which communications can happen. The most commonly used ones are:

- GPIO
- Analog/voltage reading
- UART
- SPI
- I²C

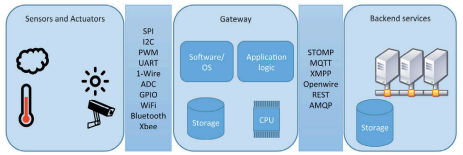


Sensor/Actuator IoT
Microcontroller interfacing protocols

GPIO

General Purpose Input Output (GPIO) is a simple communication mechanism that involves one pin between the controller and the sensor/actuator. This protocol is pretty good for sending a few bits of data.

A relay can be controlled via a GPIO pin. DHT11's temperature can be read using the digital pin.



I2C

Inter-Integrated Circuit, pronounced as I-Squared-C or I-two-C, uses a single data line and a clock to communicate between multiple sensors connected to the controller. This protocol is very good for on-board or short-range communication.

For example, the BH1750 from the ROHM semiconductor is an ambient light sensor that measures the ambient light around the sensor in Lux. This sensor exposes its data to the controller over the I^2C bus. Read more about BH1750 here: `htt p://www.mouser.com/ds/2/348/bh1750fvi-e-186247.pdf`.

This protocol can be used for controller-to-controller communication as well.

Micro-controller to Cloud Communication Protocol

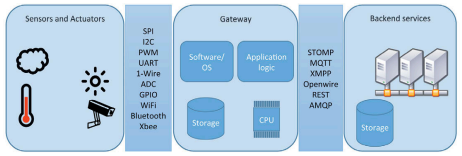
Message Queuing Telemetry Transport

The **Message Queue Telemetry Transport** (**MQTT**) protocol is a battery-friendly, small footprint message broker that implements an extremely lightweight publish/subscribe protocol. MQTT is suitable when working with power- and computation-constrained devices.

MQTT for Sensor Networks

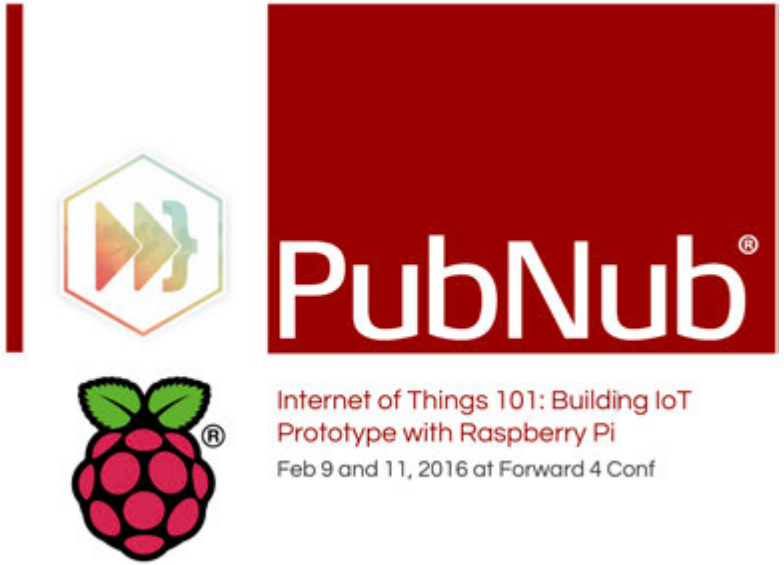
MQTT for Sensor Networks (**MQTT-SN**) caters for more non-TCP/IP embedded devices, where MQTT runs using a TCP/IP stack.

MQTTS and MQTT-SN are different. MQTTS is MQTT over SSL.



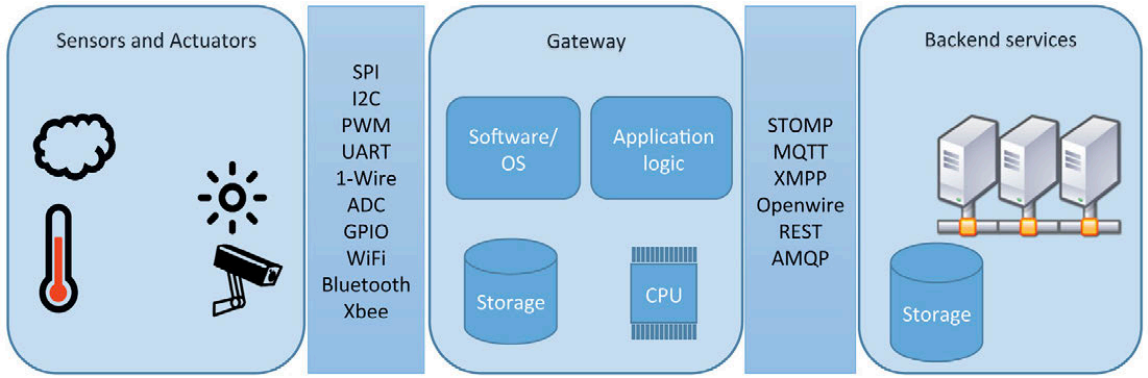
HOGESCHOOL ROTTERDAM

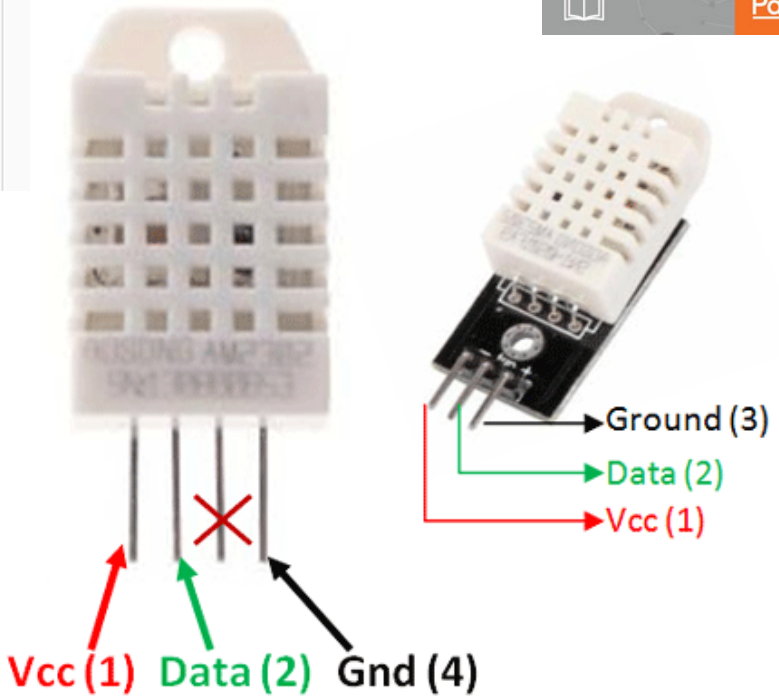
Sensor / Actuator Core-Concepts



https://github.com/pubnub/workshop-raspberrypi

Sensor name	Datasheet link
Temperature and humidity sensor (DHT11)	https://akizukidenshi.com/download/ds/aosong/DHT11.pdf





ACTUATORS

An actuator is a component of a machine that is responsible for moving or controlling a mechanism or system. An actuator requires a control signal and a source of energy.

The control signal is relatively low energy and may be electric voltage or current, pneumatic or hydraulic pressure, or even human power.

ACTUATORS

Actuators

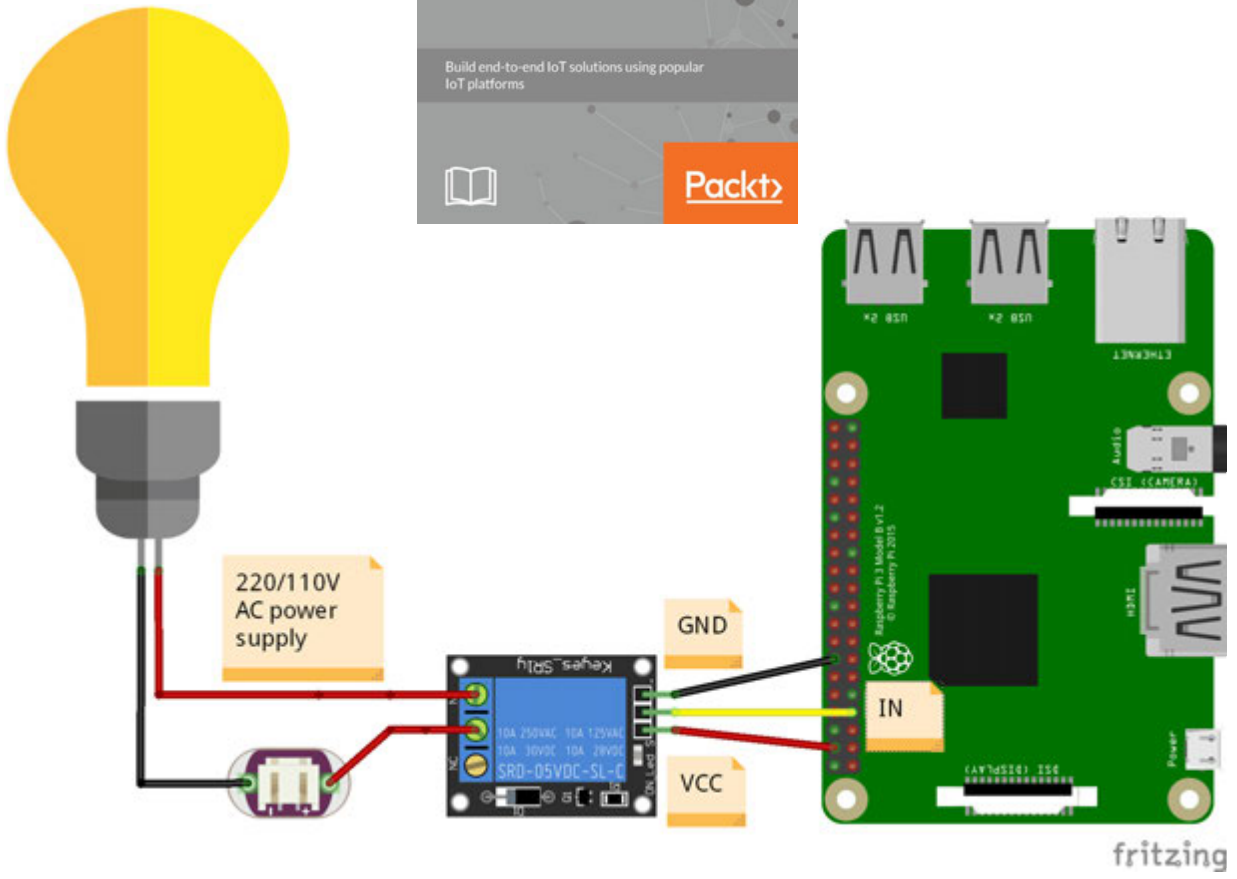
While sensors sense the environment around them, an actuator is responsible for controlling a system by actuating (open/close or on/off). For example, a switch that can be used to turn a light on or off can be controlled via an actuator.

An actuator generally takes a signal which indicates the nature of an operation. For example, if we need to turn a switch on or off, we use a relay.

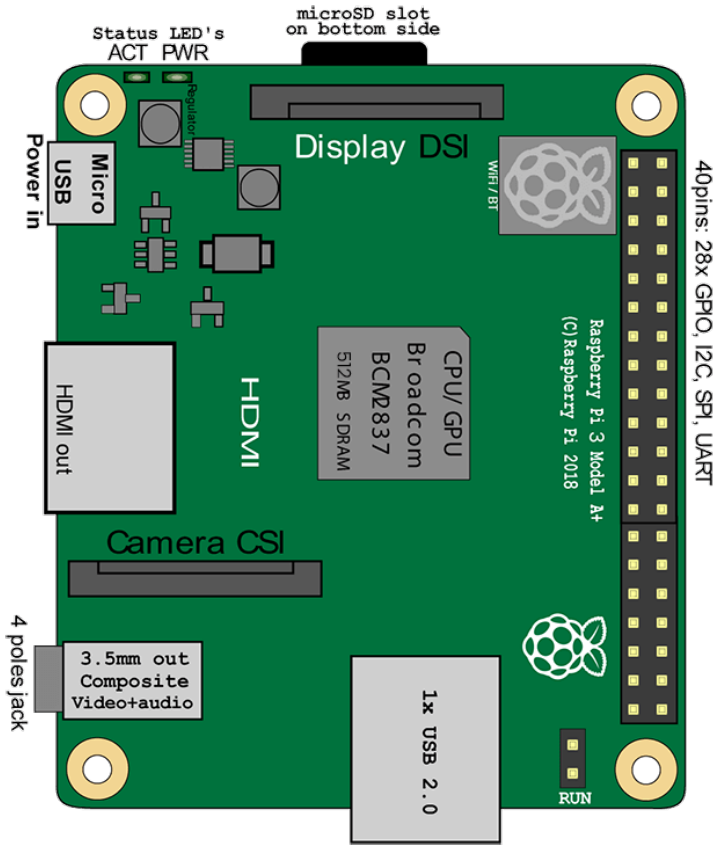
A relay is an electromagnetic switch that takes in the AC mains power supply at one end and a digital signal at other. The general state of the relay is open and that means there is no power supply flowing through from the mains to our load. Once there is a digital high or a digital low signal (depending on the type of relay), the relay moves to a closed state where the load and mains are short and the power flows through.

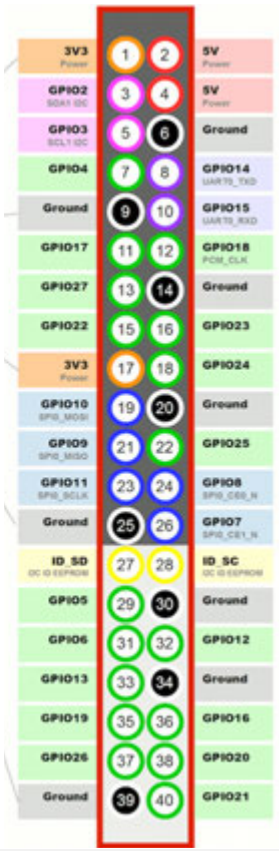
To get a better understanding of how a relay works and how to use it, take a look at *How to use a relay, the easy way*: https://www.youtube.com/watch?v=T1fNQjelojs.

A relay is one of several core actuator components, which are then reused in other actuators such as a solenoid valve.



Raspberry Py 3A+





GPIO schematics of the Raspberry Pi 3 A Plus hardware.

Shown als are general positioning of all the vital circuitry on the board.
The Raspberry Pi 3 A+ is a cut down version of the Pi 3B. As you can tell by its diagram, it features a single USB 2.0 Port. It's only means of network connectivity is the inbuilt Wi-Fi.

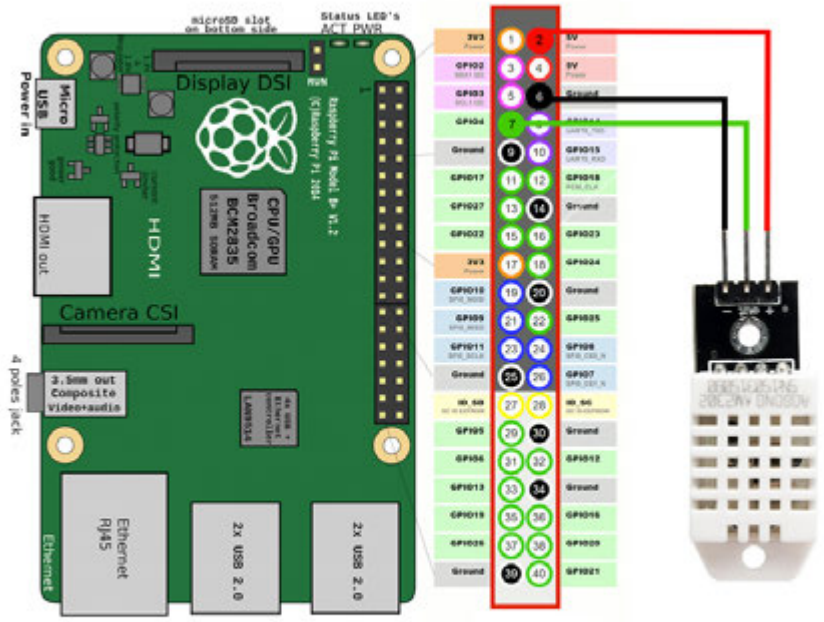
CPU: 1.4 GHz quad core ARM Cortex-A53
GPU: 250MHz Broadcom VideoCore IV
RAM: 512mb (Shared with GPU)
Storage: Micro SD
USB 2.0 Ports: 1
USB 3.0 Ports: 0
Networking: 802.11b/g/n/ac dual band 2.4/5 GHz wireless, Bluetooth 4.2 LS BLE
Video Input: 15-pin MIPI camera interface (CSI) connector
Video Ouputs: HDMI 1.3, MIPI display interface, DSI
Audio Inputs: Audio over I2S
Audio Outputs: 3.5mm phone jack, Digital Audio via HDMI
Low-Level peripherals: 17 x GPIO, +3.3v, +5v, ground, Plus the following that can be used as GPIO: UART, I2C Bus, SPI bus with two chip select, I2S audio
Power Source: 5v via MicroUSB or GPIO header
Size: 65.00mm x 56.50mm x 17mm
Weight: 23 g (0.81 oz)

Raspberry Pi Sensor to ThingsBoard using Constrained Application Protocol

HOGESCHOOL ROTTERDAM

Installation

Connecting the sensor



To connect the DHT22 sensor to the RPi:

1. Connect the `-` output on the sensor the pin 6 on the RPi.
2. Connect the `+` input on the sensor to pin 2 on the RPi.
3. Connect the `output` pin on the sensor to pin 7 on the RPi.

https://github.com/Silver292/rpi-coap

Introduction

This script is designed to be run on Raspberry Pi 3 hardware running the Raspbian 9.6 operating system.
It is also assumed that a DHT22 temperature and humidity sensor is connected to the Raspberry Pi (RPi) using GPIO 4 pin.
The process of connecting the sensor is described here.

The script creates a Constrained Application Protocol (CoAP) client on the RPi and repeatedly polls the attached DHT22 sensor for temperature and humidity data.

This data is then formatted to JavaScript Object Notation (JSON) and is sent to a CoAP endpoint provided by the ThingsBoard Cloud Platform.

This data is then displayed on the ThingsBoard dashboard that shows the current temperature and humidity as well as historical readings in a graph form.

IoT & Node-RED

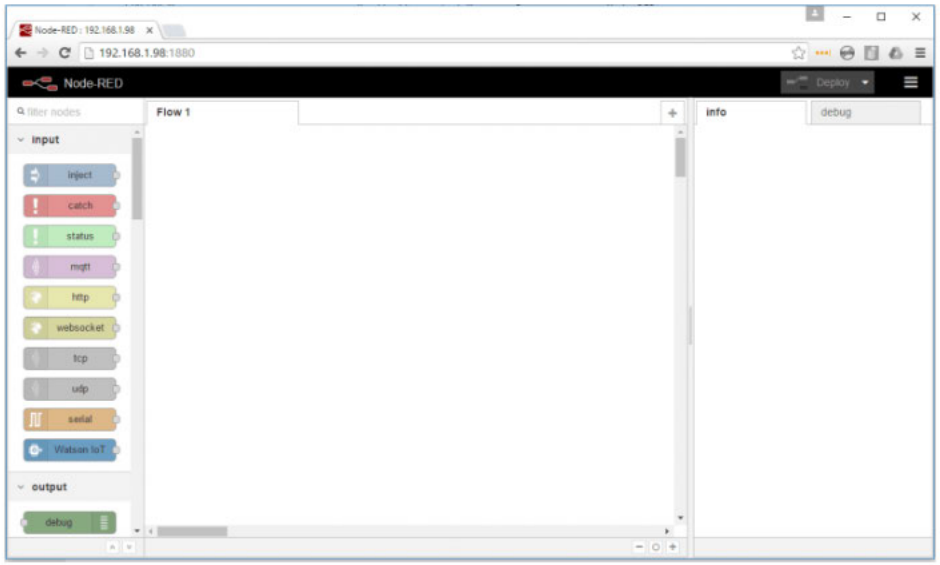
Tools that make it easier for developers & designers at all levels to bring together the different streams of events, both Physical and Digital that make up the IoT

What's Node-RED?

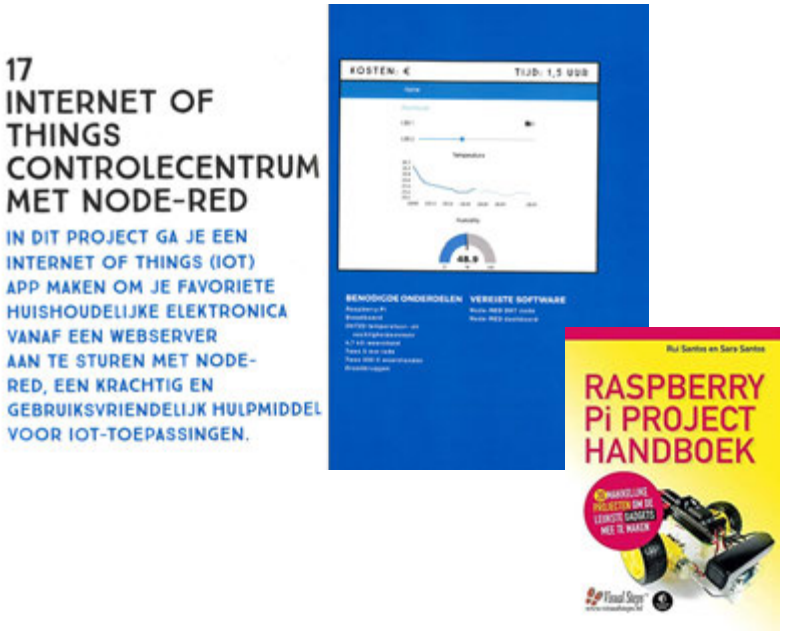
Node-RED is a powerful open source tool for building Internet of Things (IoT) applications with the goal of simplifying the programming component.

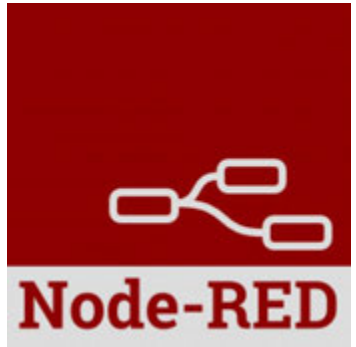
It uses a visual programming that allows you to connect code blocks, known as **nodes**, together to perform a task.

The nodes when wired together are called **flows**.



https://randomnerdtutorials.com/getting-started-with-node-red-on-raspberry-pi/





IoT & Node-RED

What can you do with Node-RED?

Node-RED makes it easy to:

- Access your RPi GPIOs
- Establish an MQTT connection with other boards (Arduino, ESP8266, etc)
- Create a responsive graphical user interface for your projects
- Communicate with third-party services (IFTTT.com, Adafruit.io, Thing Speak, etc)
- Retrieve data from the web (weather forecast, stock prices, emails. etc)
- Create time triggered events
- Store and retrieve data from a database

Here's a library with some examples of flows and nodes for Node-RED.

https://randomnerdtutorials.com/getting-started-with-node-red-on-raspberry-pi/

Setting up Node.js on the Raspberry Pi

Open a new Terminal and run the following commands:

$ sudo apt update
$ sudo apt full-upgrade

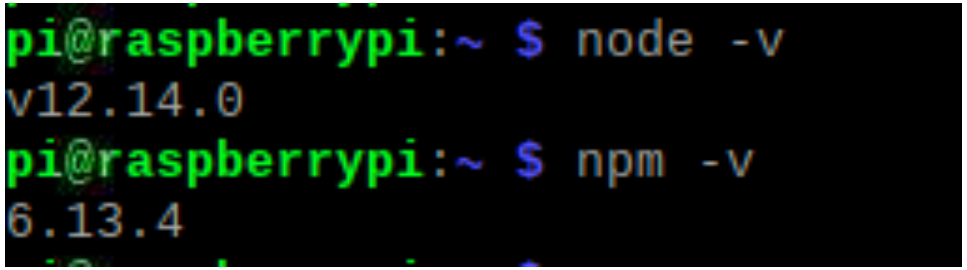
This will upgrade all the packages that need upgrades. Next, we will install the latest version of Node.js.

We will be using the Node 7.x version:

$ curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash -
$ sudo apt install nodejs

This will take a moment to install and once your installation is done you should be able to run the following commands to see the version of Node.js and npm:

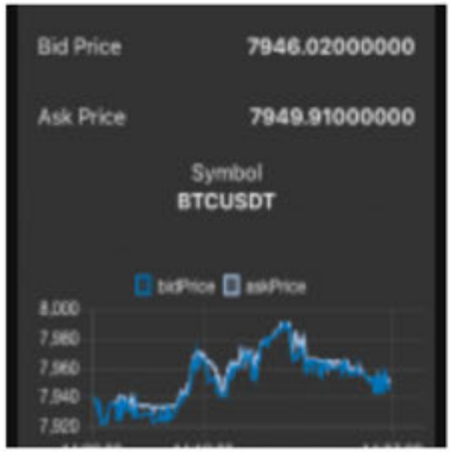
$ node -v
$ npm -v



Original author(s)	Isaac Z. Schlueter
Developer(s)	Rebecca Turner, Kat Marchán, others
Initial release	12 January 2010; 9 years ago[1]
Stable release	6.12.0 / 8 October 2019; 2 months ago[2]
Repository	github.com/npm/cli
Written in	JavaScript
License	Artistic License 2.0
Website	www.npmjs.com

Node-RED Programming Guide to the IoT

http://noderedguide.com/



Node-RED: Lecture 7 – Dashboards and UI techniques. In this lecture you will take a look a few techniques to allow you to visualize data passing through flows. We'll focus on three approaches;

- PART 1: the use of a third party dashboard tool – FreeBoard,
- PART 2: using the standard Node-RED default UI nodes
- PART 3: using a standard external JavaScript charting library (Morris.JS).

http://noderedguide.com/lecture-7-dashboards-and-ui-techniques-for-node-red/

What you need to do (lesson 6)

- Make a new Mini Project Repository on your GitHub Account
  by providing insights on Core IoT Concepts you learned in this lesson

- Studie *README.md*
  on https://github.com/robvdw/CMIDAT01K-DATA-SCIENCE-for-IOT

- Studie *GRADING*
  on https://github.com/robvdw/CMIDAT01K-DATA-SCIENCE-for-IOT/blob/master/Docs/BEOORDELINGS_MODEL_DS_for_IoT_V24_npv_2020.pdf

DEADLINE OPLEVERING

HOGESCHOOL ROTTERDAM

Om voor beoordeling in aanmerking te komen lever je een ascii text file (.txt) aan met de link (URL) van jouw Git-Hub account met de IoT-repository.

De repository bevat een beschrijving van het IoT-Project waarin bewijzen
---zoals video-materiaal / code / design concept / data organisatie---
zijn opgenomen van op basis van de door jouw bedachte/geteste IoT data-pipeline/Prototype
zoals beschreven op https://github.com/robvdw/CMIDAT01K-DATA-SCIENCE-for-IOT

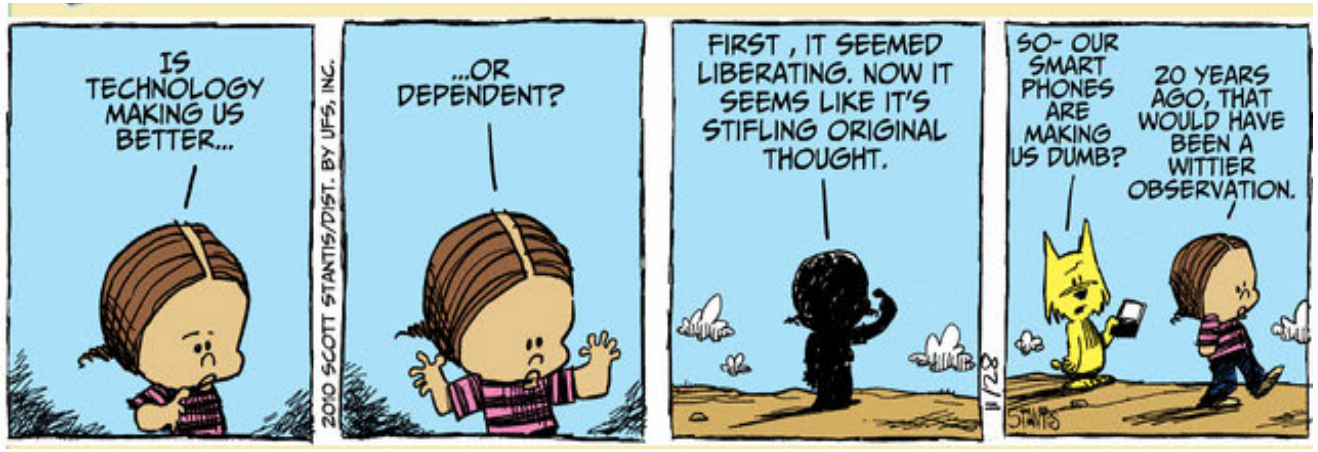
Deze text-file moet je uploaden via de LMS **INLEVER_MAP_OP2**
van de CURSUS CMIDAT01k Data Science for IoT (2020-2021).
Deadline is: 9 februari (vanaf 10 februari kun je niet meer inleveren).

Beoordeling vindt plaats conform het beoordelingsmodel:
https://github.com/robvdw/CMIDAT01K-DATA-SCIENCE-for-IOT/blob/master/Docs/BEOORDELINGS_MODEL_DS_for_IoT_V24_npv_2020.pdf

Deze mededeling wordt als E-mail bericht wordt verstuurd (2 februari)
in week 10 OP2 collegejaar 2020-2021

Nota Bene! Stuur een Reply op deze E-Mail
als je voor beoordeling van de cursus: CMIDAT01k Data Science for IoT (2020-2021) **in aanmerking wilt komen.**

HOGESCHOOL ROTTERDAM



This lesson was developed by:
Rob van der Willigen
CMD, Hogeschool Rotterdam
NOV 2019

Creative Commons License Types

	Can someone use it commercially?	Can someone create new versions of it?
Attribution	👍	👍
Share Alike	👍	Yup, AND they must license the new work under a Share Alike license.
No Derivatives	👍	👎
Non-Commercial	👎	Yup, AND the new work must be non-commercial, but it can be under any non-commercial license.
Non-Commercial Share Alike	👎	Yup, AND they must license the new work under a Non-Commercial Share Alike license.
Non-Commercial No Derivatives	👎	👎

SOURCE
http://www.masternewmedia.org/how-to-publish-a-book-under-a-creative-commons-license/

http://creativecommons.org/licenses/

http://empoweringthenatives.edublogs.org/2012/03/15/creative-commons-licenses/