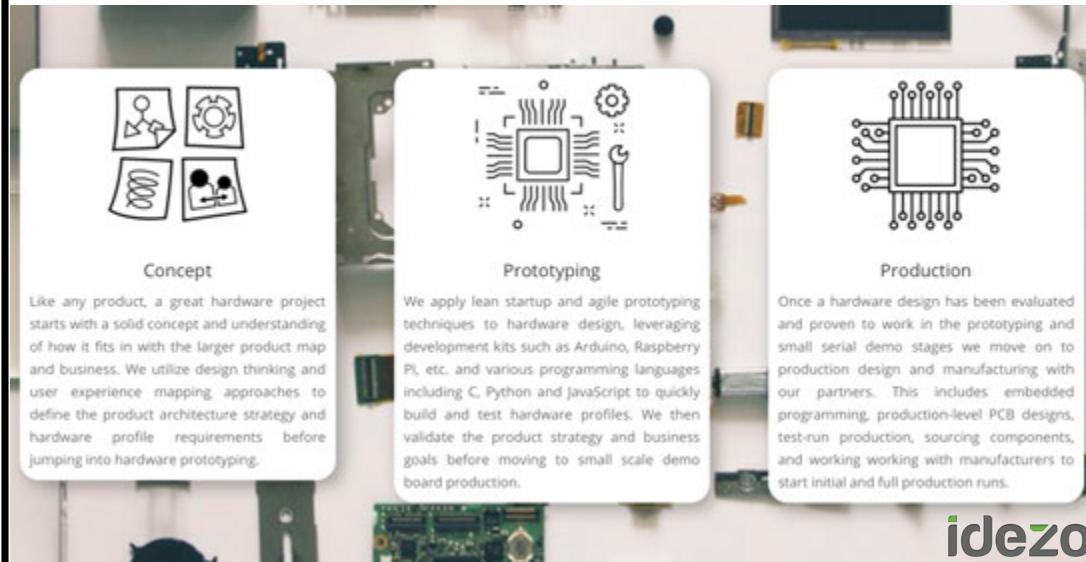


HANDS ON APPROACH TO

# Data Science

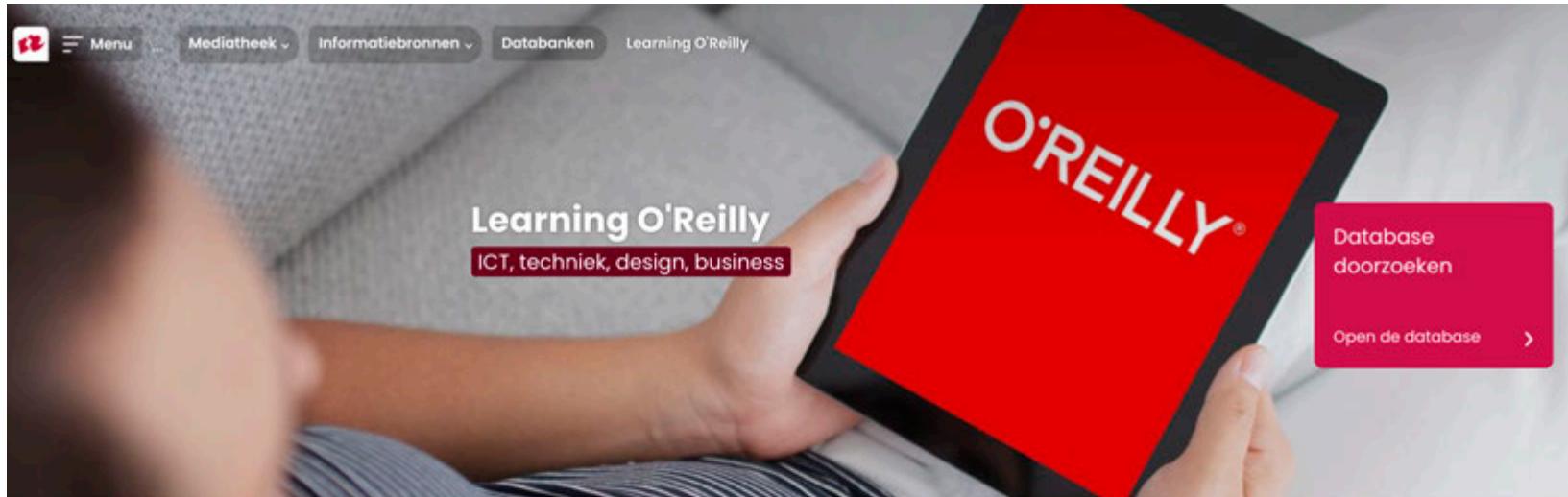
## for (the)

## IoT



**Rob van der Willigen**

<https://github.com/robvdw/CMIDAT01K-DATA-SCIENCE-for-IOT>



Learning O'Reilly is een Engelstalig online leerplatform gericht op ICT vaardigheden. Het biedt tienduizenden e-books en vele video's, case studies en "learning paths", waarbij je zelf je voortgang kunt monitoren.

## Toegang

### Via de website

De Learning O'Reilly website is gekoppeld aan de Hogeschool Rotterdam login. Kom je toch een O'Reilly inlogscherm tegen, vul dan alleen je @hr.nl e-mailadres in en klik op Sign In with Single Sign On. Krijg je een blanco pagina te zien? Schakel eventuele adblockers uit!

### Via de O'Reilly app

Leer wat je wil, waar je wil. Met de O'Reilly app download je e-books en ga je verder met een playlist waar je op een ander apparaat gebleven was.

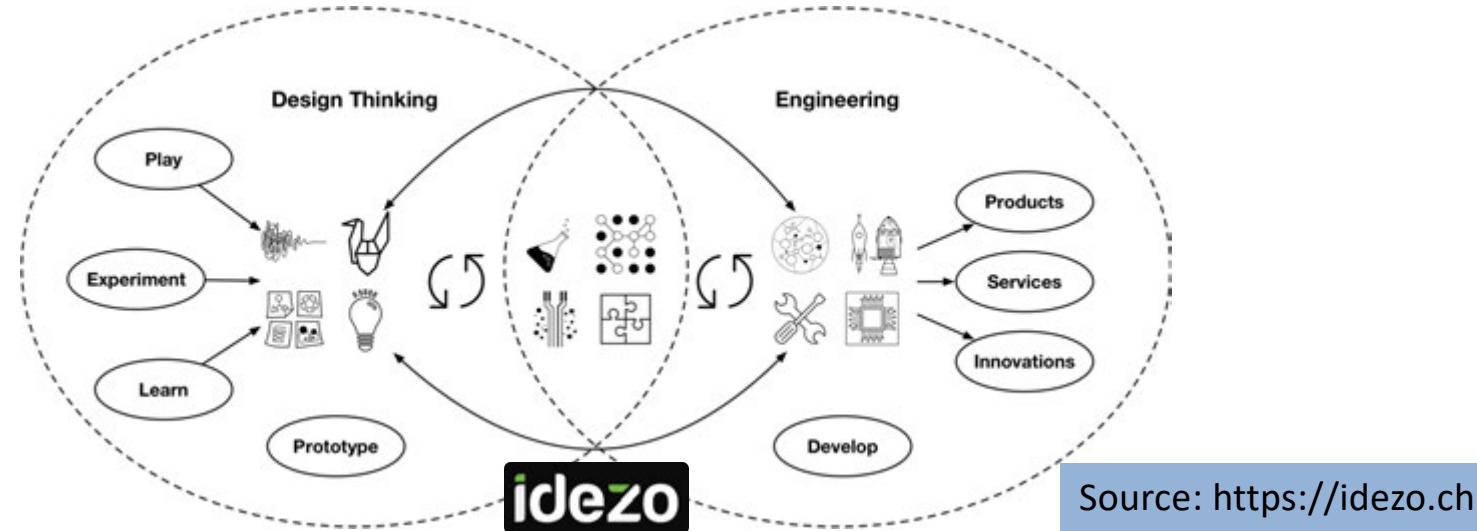


- 1 Gebruik je @hr.nl e-mailadres om in te loggen.

## Inhoud

- Meer dan 8000 instructievideo's en 42.000 (hoofdzakelijk Engelstalige) e-books.

# HANDS ON APPROACH TO DATA SCIENCE for (the) IoT



Source: <https://idezo.ch>

This Course material is distributed under the Creative Commons Attribution- NonCommercial- ShareAlike 3.0 license. You are free to copy, distribute, and transmit this work. You are free to add or adapt the work. You must attribute the work to the author(s) listed above.

You may not use this work or derivative works for commercial purposes. If you alter, transform, or build upon this work you may distribute the resulting work only under the same or similar license.

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](mailto: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>

# Course Setup

**Lesson 01:** week 02    **Discovering the IoT Data Science Domain**

**Lesson 02:** week 03    **Defining project requirements**  
                          + Cost 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    **Explaining Grading + Summary + Q & A**

**Week 09 / 10: FEEDBACK**

# About this course

# Data Science for IoT is about ...

---

Experimenting, prototyping & Iterating

Concepting, Testing and Debugging

Reusing and Remixing

Abstracting and Documentation

Keeping a (GitHub) Journal + code commenting

# Documentation

---

Domain knowledge

Prototype-Iteration

Concept + Design Scenarios

Code / Coding (algorithms)

Feedback

# IoT-project questions that need to be answered

---

Show me what you are working on?

What is it about?

What makes it a Data Science for IoT project?

What steps did you take to plan the project?

What coding IDE did you use?

What hardware did you use and how is it connected to the internet?

Which IoT platform + IoT data pipeline did you choose & why?

Did the plan change at all over time?

What did you do when you had to change the plan and how did the program then work?

# Define for yourself

---

If someone new came to our class and asked you about data science for IoT:

**what it is, how it works and  
what you can do with it -  
what would you tell them?**



## Hardware

---

Microcontroller technology  
Sensor technology  
Hardware integration  
Physical design (3d)  
Testing

## Software

---

Web application development  
Microcontroller programming  
Software integration

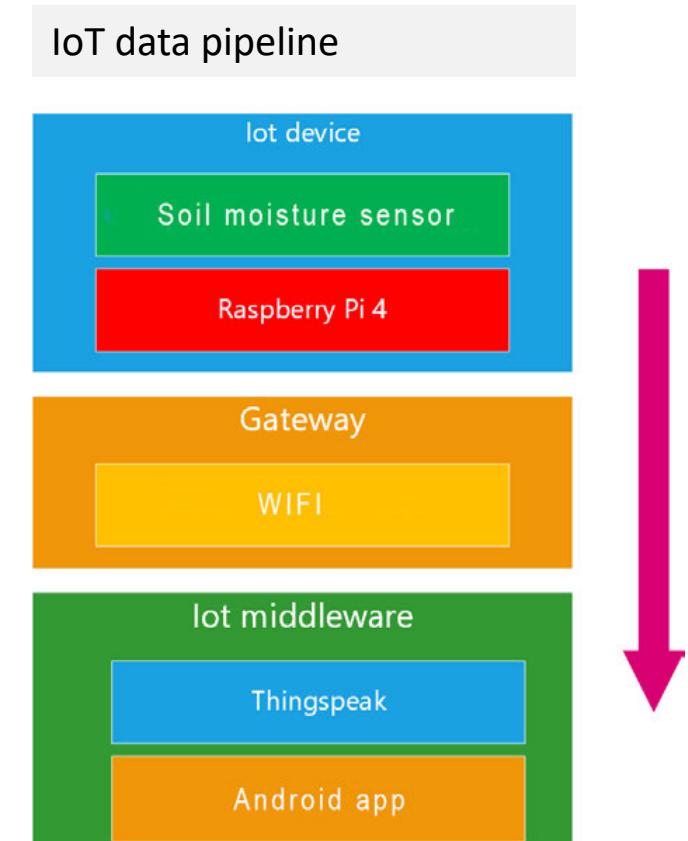
# IoT-project == IoT data pipeline

Imagine you would like to know the weather outside in your garden.

In an ideal IoT world, you would have a range of sensors, for example, a temperature sensor that can talk directly to the Internet.

This would be achieved by sending the temperature data from the sensor directly to a backend service via the Internet.

The backend service would probably display the data on a web page and would ideally offer the ability to display and analyze the data on your phone.



# Learning Objective

During the course, students will:

- Create + Describe a Internet of Things (IoT) concept/scenario
- Become familiar with the concepts surrounding IoT applications by building them such as Data Science & Computational Thinking
- Learn to design and develop simple lightweight IoT prototypes
- Learn to work independently to prepare a interactive prototype of an IoT product

# Deliverables:

- A working prototype that includes software (code), hardware and electronics elements.
- Code for the application
- Documentation + Evidence of the IoT project (contributed to GitHub).
- Demonstration of the completed project (annotated video)

# Choosing an IoT-project

## **Think Big / Act Small**

The main objective is to choose something you **care** about.

Having a project that's genuinely interesting to you is going to be best.

Keep in mind that it should be reasonably well scoped.

Choose something small, discrete and easy to implement

Identify a small solvable problem that illustrate your key IoT idea/concept.

Keep it constrained but conceptually interesting.

Provide a IoT data pipeline

# Submitting your IoT-project

Documentation should be posted to GitHub as a new Repository.

Hands-on guide for working with this platform can be [found here](#).

You should document your project thoroughly including:

- *A description of the concept and goals (include video and/or diagrams as needed.)*
- *A description of relevant prior projects, approaches or methods you researched and that inspired the project. Be thorough and show what informed the project.*
- *The process you underwent to reach the outcome (experiments, hacks, tests, refinements, iterations, failures)*
- *The outcome itself and how it works. Include supporting images, a video of the working prototype, circuit diagrams, etc.)*
- *Outline next steps and future directions.*
- *Add/upload code (with comments) and any supporting documentation and files.*

# Grading

The most important thing for this project is to come up with a **compelling concept** - something interesting, informed, aware or critical.

The breakdown of the grades favors your ability to come up with an idea like this. Focus on the idea before implementation!

***This IoT course is a way for you to showcase:***

- Your engagement and understanding of the Data Science for IoT domain
- Your creative capacity + explanation of the IoT-concept/scenario
- Your own development in technical (coding-)skills

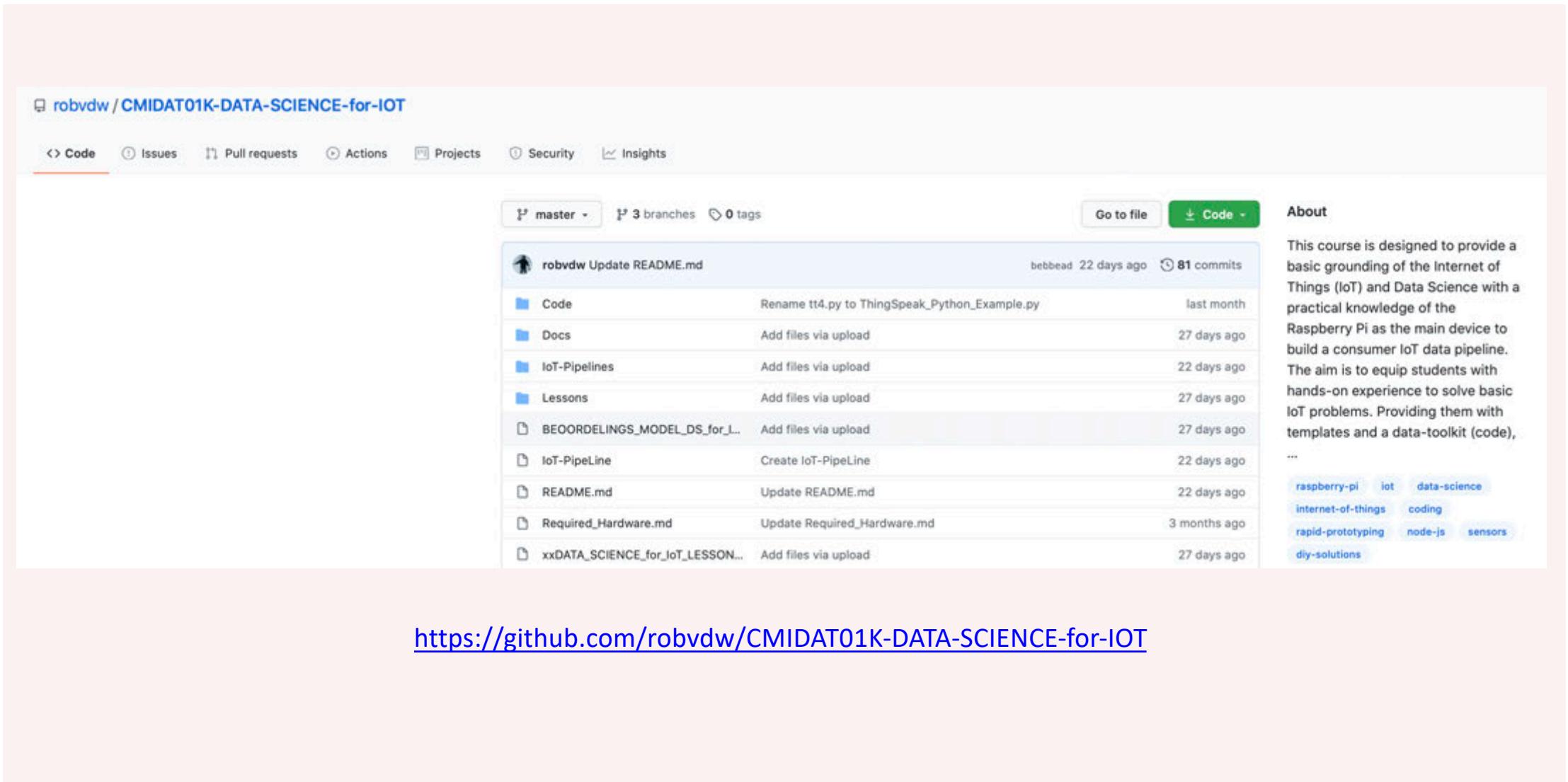
# Grading

A strong grade will result by creating interesting, well-crafted and well documented projects.

As such, each creative project will be graded as follows:

- 30% - Technical Implementation - Quality of code and execution of the outcome
- 20% - Creativity of Approach and Topic - Merit, Coding, and Context for the outcome/proposal
- 20% - Outcome Documentation - Well illustrated with appropriate use of code, video, diagrams, repeatability, etc.
- 20% - Git-Hub Repository and/or Maker Meet-Up Demonstration - high quality and well narrated demonstration of the solution
- 10% - Process - Description of process (ideation, iteration, etc.) and personal reflection on learning outcomes of technical skills

# Materials & Guidelines



The screenshot shows a GitHub repository page for the user robvdw. The repository name is CMIDAT01K-DATA-SCIENCE-for-IOT. The page has a light pink header bar.

**Code** tab is selected. Other tabs include Issues, Pull requests, Actions, Projects, Security, and Insights.

Branch: master (dropdown), 3 branches, 0 tags.

Commit history:

Author	Message	Date	Commits
robvdw	Update README.md	bebbead 22 days ago	81 commits
	Rename tt4.py to ThingSpeak_Python_Example.py	last month	
	Add files via upload	27 days ago	
	Add files via upload	22 days ago	
	Add files via upload	27 days ago	
	Add files via upload	27 days ago	
	Create IoT-PipeLine	22 days ago	
	Update README.md	22 days ago	
	Update Required_Hardware.md	3 months ago	
	Add files via upload	27 days ago	

**About** section:

This course is designed to provide a basic grounding of the Internet of Things (IoT) and Data Science with a practical knowledge of the Raspberry Pi as the main device to build a consumer IoT data pipeline. The aim is to equip students with hands-on experience to solve basic IoT problems. Providing them with templates and a data-toolkit (code), ...

Tags:

- raspberry-pi
- iot
- data-science
- internet-of-things
- coding
- rapid-prototyping
- node-js
- sensors
- diy-solutions

<https://github.com/robvdw/CMIDAT01K-DATA-SCIENCE-for-IOT>

# Materials & Guidelines

README.md

## Goal & Aims

This Do-it-Yourself (DIY) course is designed to provide a basic grounding of the Internet of Things (IoT).

- The aim is to equip students with the Data Science skills in order to complete basic IoT-scenario's.
- Providing students with templates and a data-toolkit (code), which can be implemented through popular single board microcontrollers (Arduino / Raspberry Pi).

## Required Hardware

See: [https://github.com/robvdw/CMIDAT01K-DATA-SCIENCE-for-IOT/blob/master/Required\\_Hardware.md](https://github.com/robvdw/CMIDAT01K-DATA-SCIENCE-for-IOT/blob/master/Required_Hardware.md)

## Scope

A hands-on introductory course exploring the Internet of Things from a (Human-Centered) Data Science point of view.

Creating your own IoT data pipeline [https://github.com/robvdw/CMIDAT01K-DATA-SCIENCE-for-IOT/blob/master/IoT-Pipelines/data\\_pipeline%20example01.png](https://github.com/robvdw/CMIDAT01K-DATA-SCIENCE-for-IOT/blob/master/IoT-Pipelines/data_pipeline%20example01.png)

## Learning objectives

- Become familiar with Data Science concepts surrounding IoT applications
- Prepare a (Rapid) prototype for an Internet of Things (IoT) data-pipeline scenario's
- Learn to design and develop simple lightweight IoT-prototypes
- Learn to report on IoT projects (inclusive prototypes + data-pipelines).

## Lessons

The content of each lesson will become available each week as a PDF document via:

<https://github.com/robvdw/CMIDAT01K-DATA-SCIENCE-for-IOT/tree/master/Lessons>

## Deliverables

- A working prototype that includes software (code), hardware and electronics elements.
- Code (commented) for the application
- Documentation of the project (contributed to a personal GitHub Repository).
- An online demo (video or IoT Platform implementation) of the completed IoT data-pipeline/project.

## Grading

This course will be graded according to the rules and definitions as outlined in:

[https://github.com/robvdw/CMIDAT01K-DATA-SCIENCE-for-IOT/blob/master/Docs/BEOORDELINGS\\_MODEL\\_DS\\_for\\_IoT\\_V24\\_npv\\_2020.pdf](https://github.com/robvdw/CMIDAT01K-DATA-SCIENCE-for-IOT/blob/master/Docs/BEOORDELINGS_MODEL_DS_for_IoT_V24_npv_2020.pdf)

## Data Science Beyond the IoT

For those who want to keep up with novel developments within the Data Science domain see my Medium.com Blog: <https://robfvdw.medium.com>

<https://github.com/robvdw/CMIDAT01K-DATA-SCIENCE-for-IOT>

# **lesson one**

# LES: 01

**IoT + DataScience Domain explained**

**Raspberry Pi**

**Assignment**

**Preview Les 02**

# Data Science

# Fundamentals

## DATA SCIENCE (ANALYTICS) VERSUS DATA ENGINEERING (PROBLEM SOLVING)

is a complex domain that requires a lot of skillful effort to perform adequately

[0] Collection	Big Data (Acquisition/Aggregation) Empirical (Sensor/IoT Measuring/Sampling)	Gathering
[1] Access + Retrieval	Ownership (Open/Closed) Storage (Cloud/Database)	Ingesting
[2] Preparation + Wrangling (Munging)	Loading Feature Extraction/Reduction Normalization Transformation Conversion	Processing
[3] Exploration	Graphical (spatial) Ontological (Language) Semantic (text) Rule-based/Algorithmic Quantitative/Qualitative Numerical/Categorical/Symbolic	Discovering
[4] Analysis + Machine-Learning	Mining (Heuristics/Statistics/Descriptive/Prescriptive) Construct Useful Insights/Trends/Patterns/Diagnosis(Information)	Conceptualizing
[5] Abstraction	Parameter Selection + Representation Summarization Problem Solving Diagnostic Prediction Encryption	Modelling
[6] Organization + Managing	Visualization Virtualization	Presenting
[7] Automation + Reporting	Performance (Measure/Monitor) Evaluation & Review Decision & Advise or Prescription (Interactive/Passive) Story Telling Prototyping	Applying

# Waarom Data Science?

Veel van de huidige  
**maatschappelijke en wetenschappelijke**  
**vraagstukken zijn dermate complex**  
dat zij niet zonder de hulp van **computertechnologie**  
opgelost kunnen worden.

Bij deze vraagstukken is de **rekenkracht van de**  
**computer** nodig om tot een **oplossing** te komen.

# Fundamentals

## Data Deluge: too much data

1 kilobyte	1,000,000,000,000,000,000
1 megabyte	1,000,000 000,000,000,000
1 gigabyte	1,000,000,000 000,000,000
1 terabyte	1,000,000,000,000,000,000
1 petabyte	1,000,000,000,000,000,000
1 exabyte	1,000,000,000,000,000,000
1 zettabyte	1,000,000,000,000,000,000

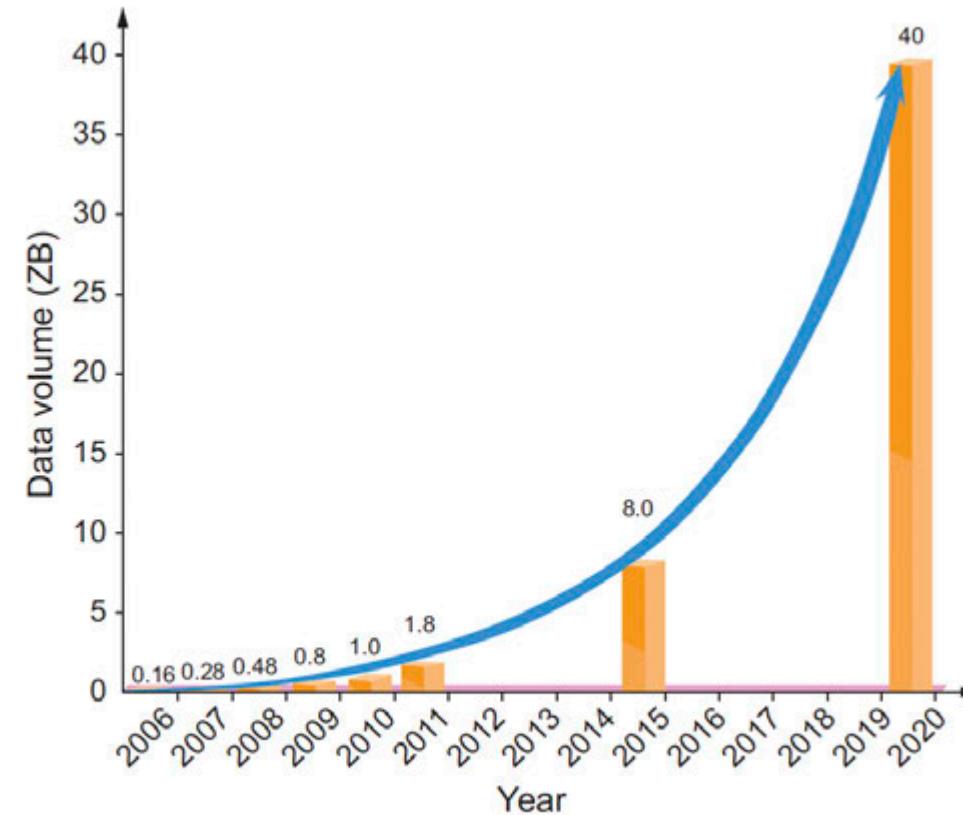
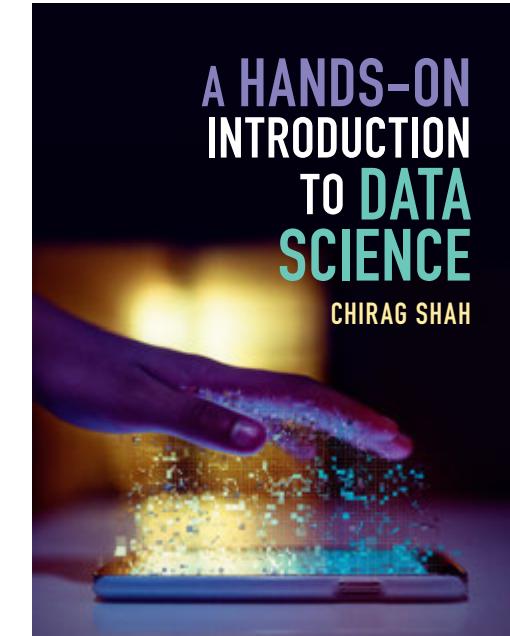


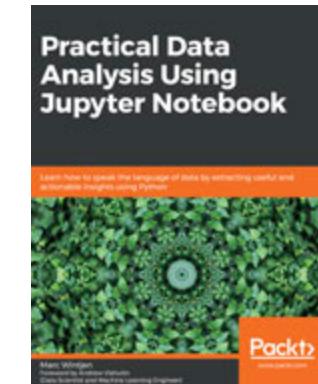
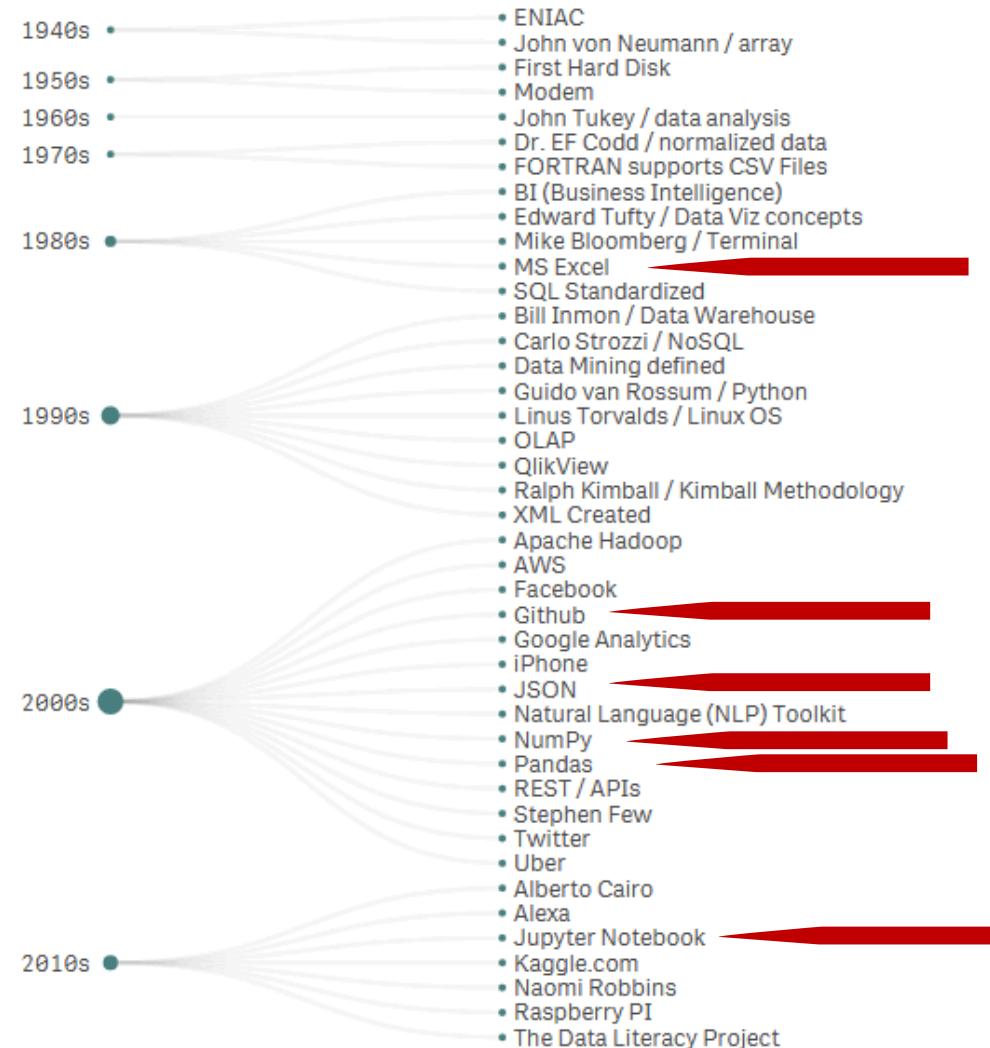
Figure 1.1

Increase of data volume in last 15 years. (Source: IDC's Digital Universe Study, December 2012.<sup>5</sup>)



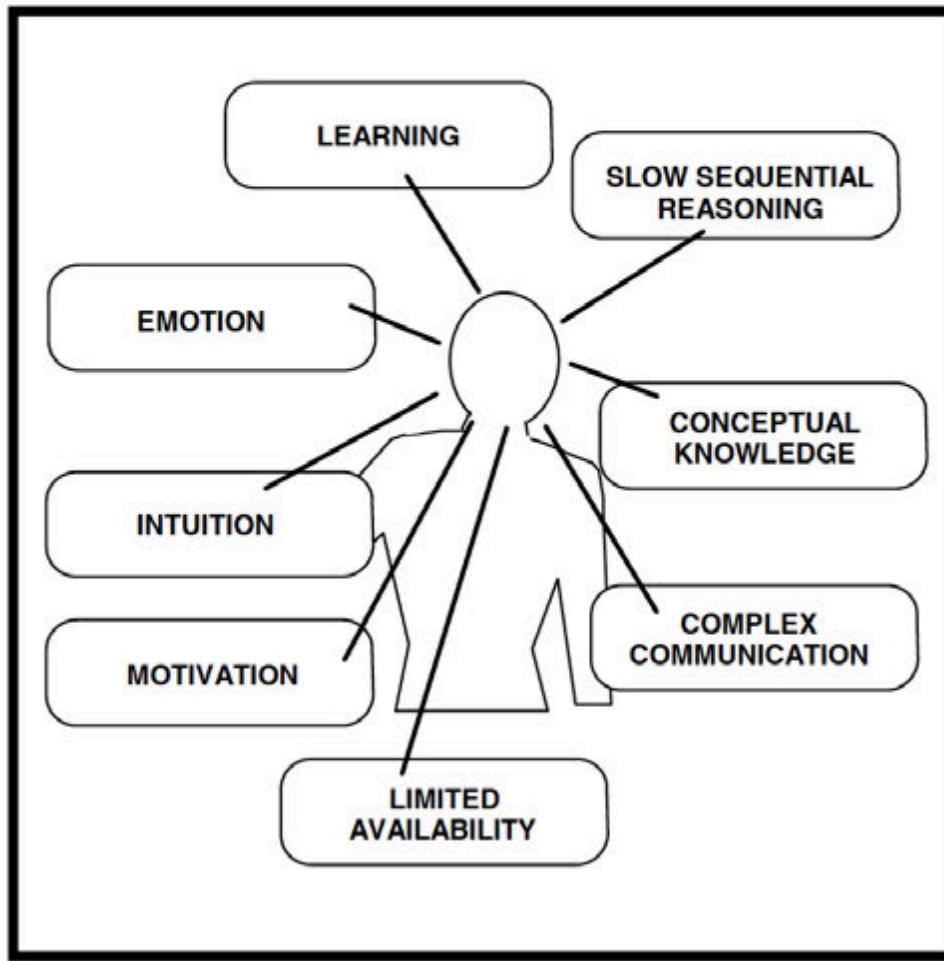
# Fundamentals

DATA Science Timeline  
gives  
Historical Context

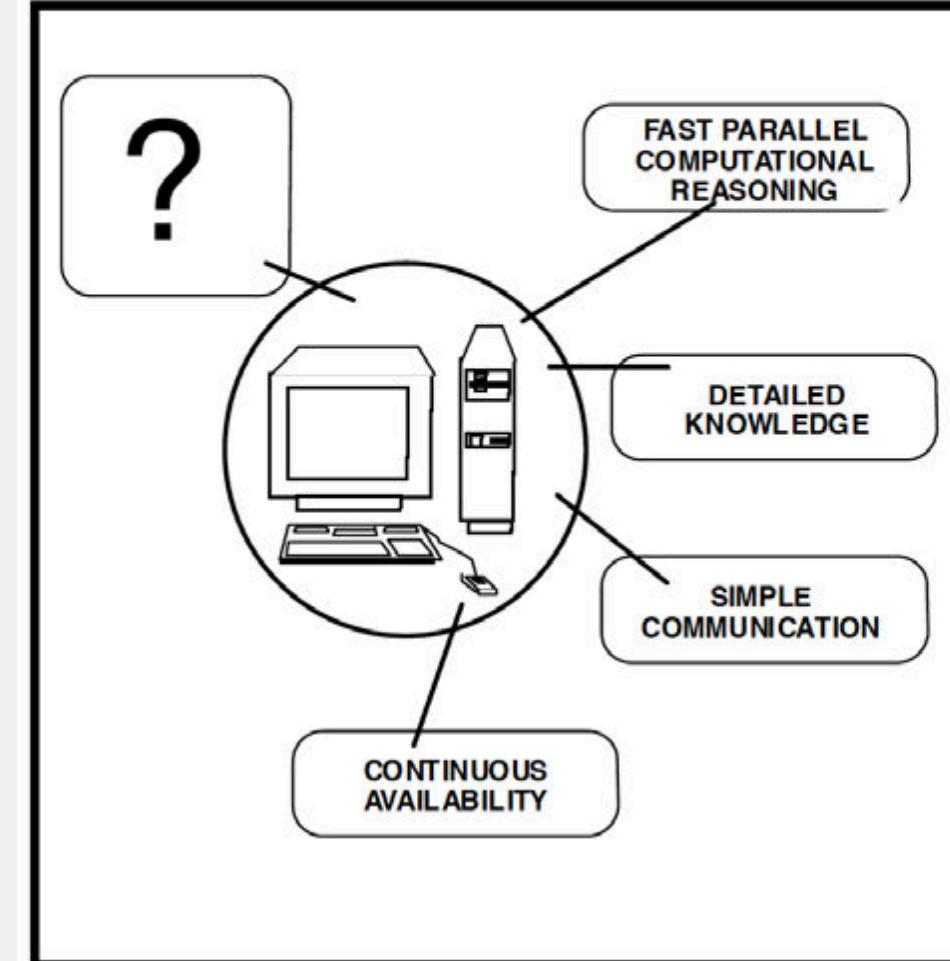


# Fundamentals

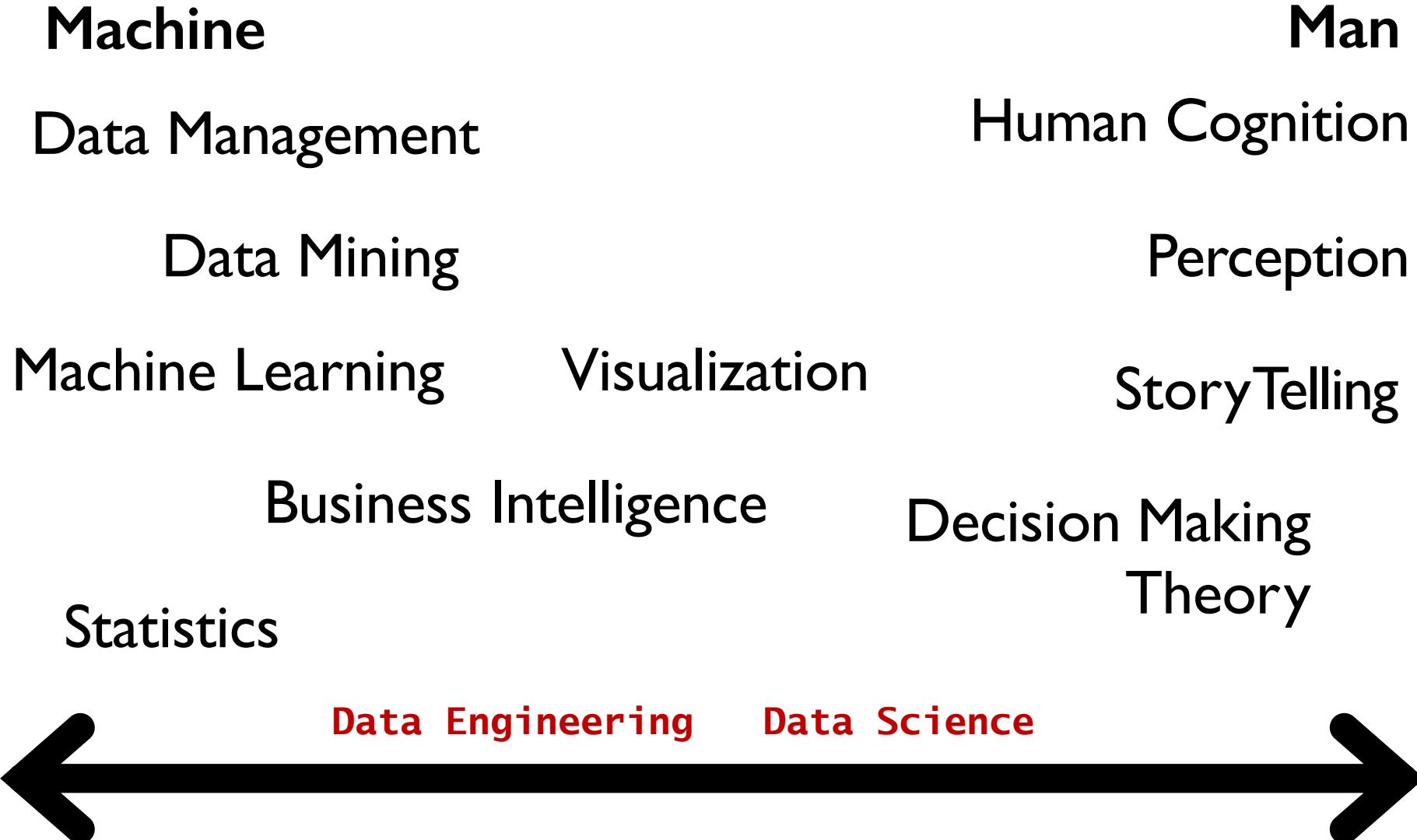
Humans are pattern-driven



Computers are data-driven



# Fundamentals



Inspired by Daniel Keim, "Visual Analytics: Definition, Process, and Challenges"

# WHAT IS DATA?

## Data [gegevens]

Raw Facts

No Context

Numbers

Symbols

Data comes from the Latin word, "datum," meaning a "thing given."

Although the term "data" has been used since as early as the 1500s, modern usage started in the 1940s and 1950s as practical electronic computers began to input, process, and output data.

<https://robfvdw.medium.com/a-generic-approach-to-data-driven-activities-e54144a509a6>

98734975471894614398734578

20875980542158009258202908

12349823094823048002343423

98734975471894614398734578

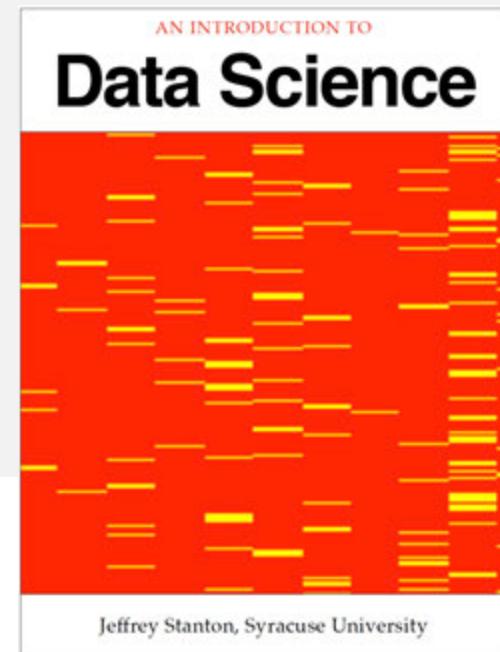
20875980542158009258202908

12349823094823048002343423

# Wat is DATA?

## Data [gegevens]

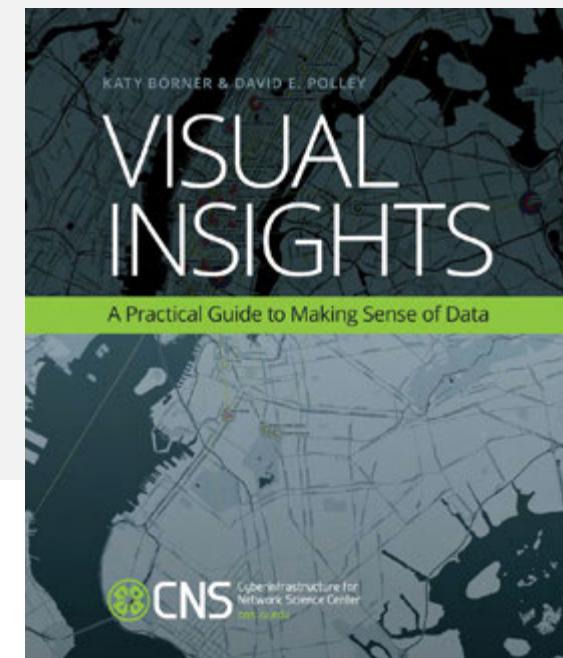
Raw Facts  
No Context  
Numbers  
Symbols



## Information [informatie]

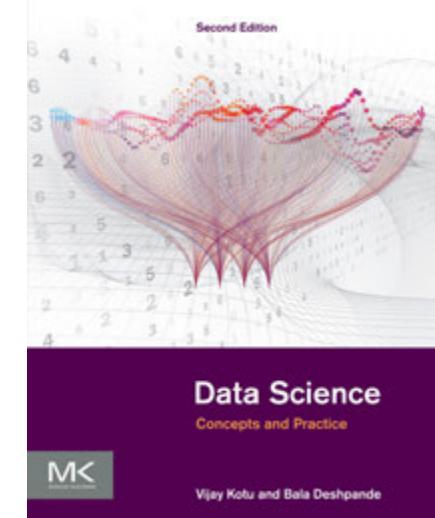
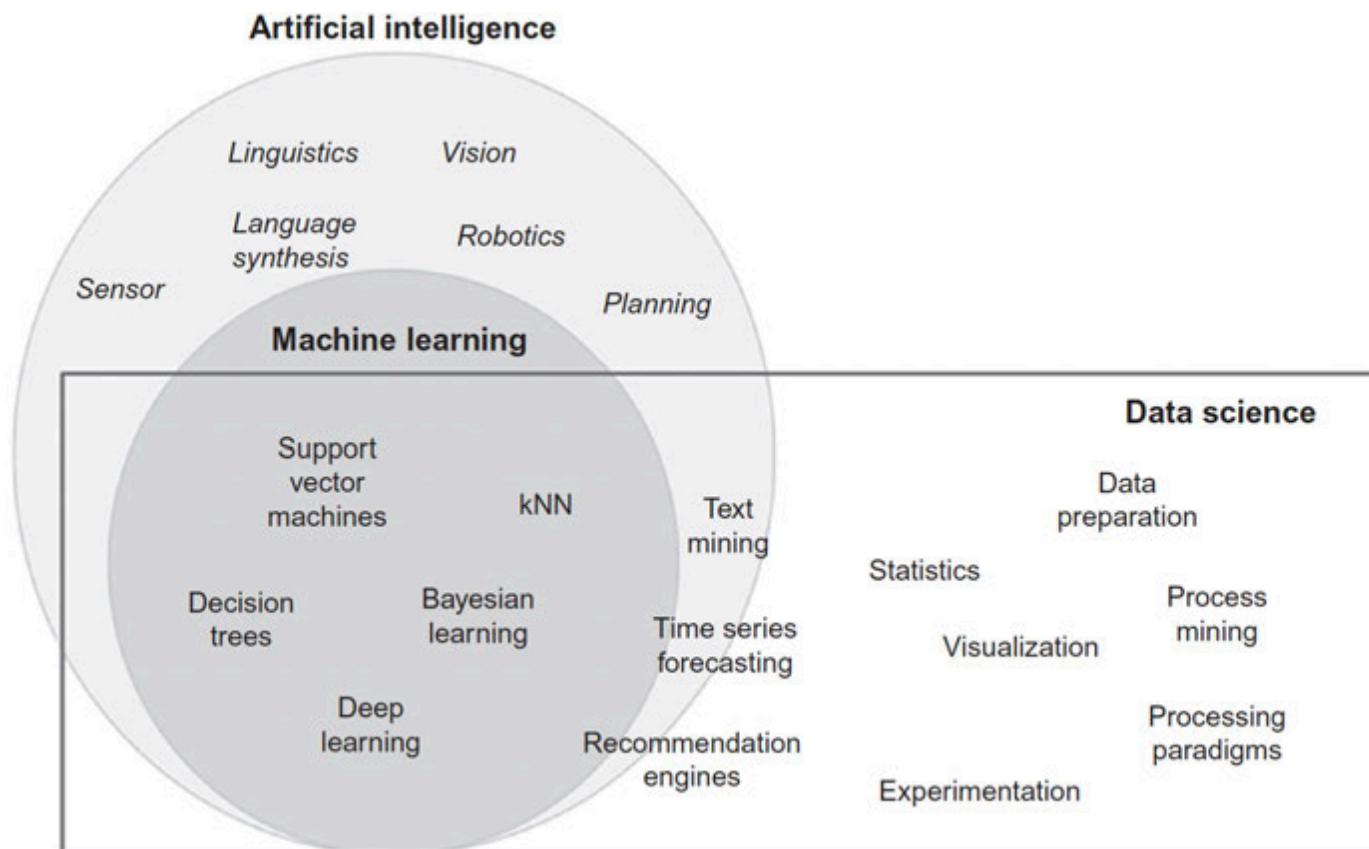
Data with structure = processed data  
Reduces Uncertainty about Data

- Summarised
- Organised
- Analysed

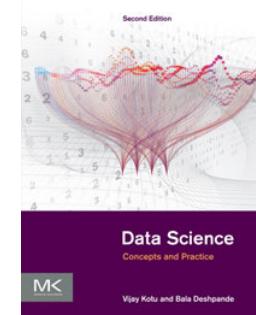
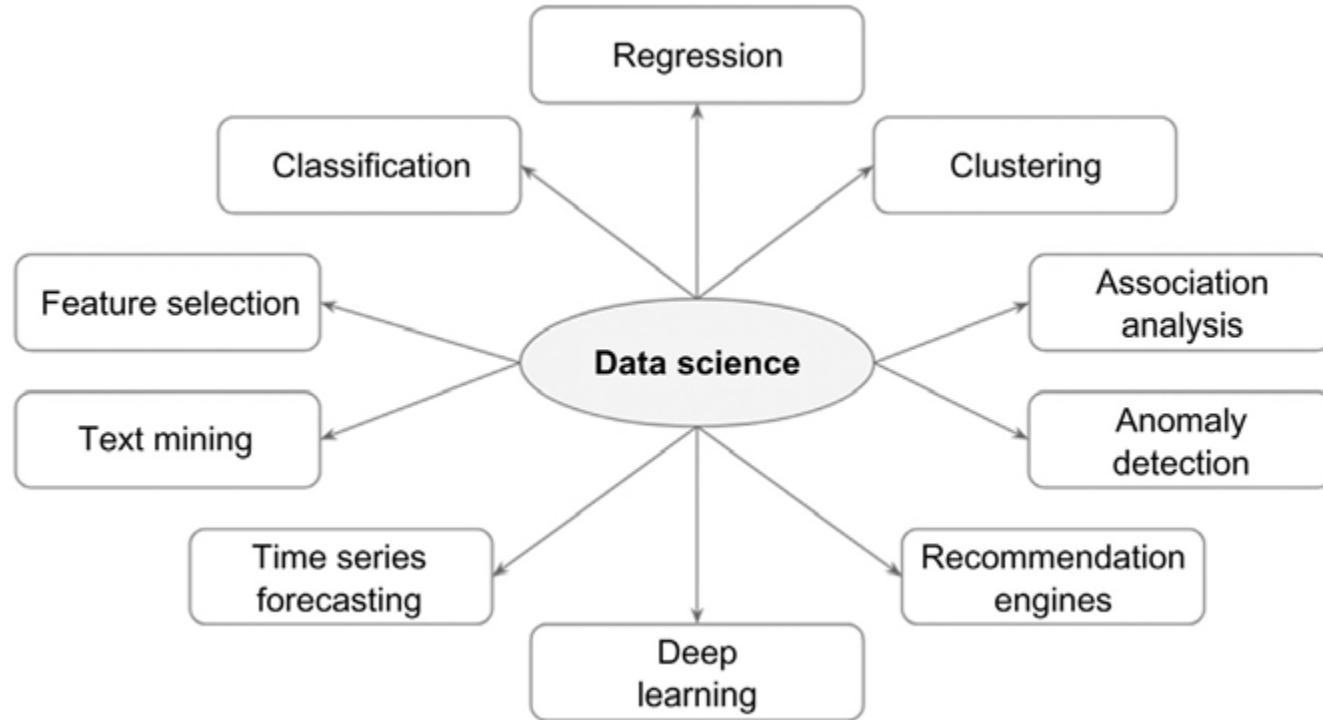


# Data Science:

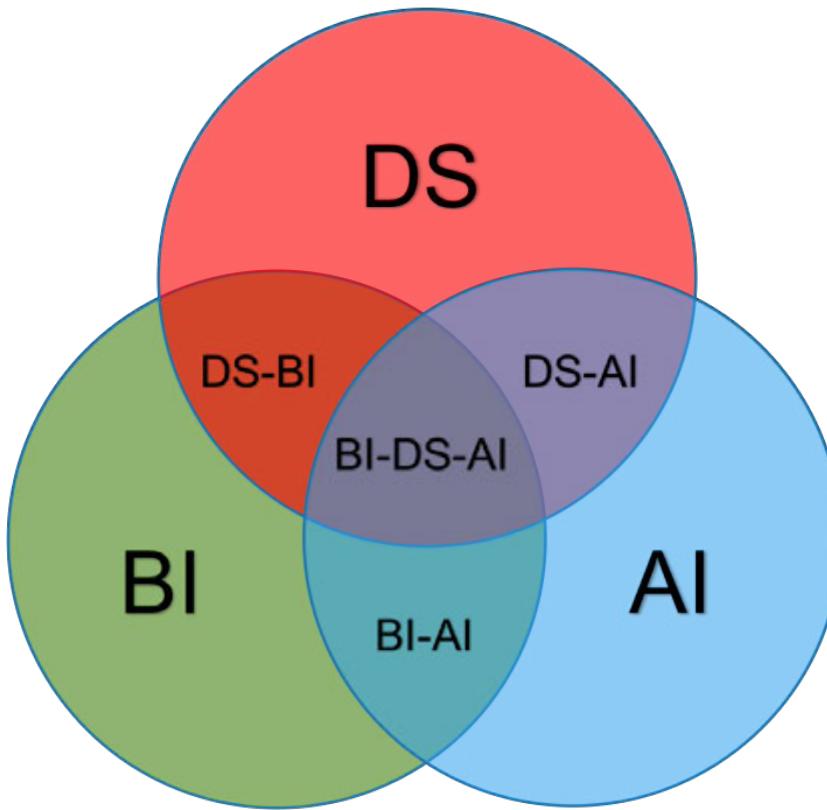
## Comprises many Techniques that extract Value from Data



# Data Science: Many Skills



Data Science refers to an emerging area of work concerned with the collection, preparation, analysis, visualization, management, and preservation of large collections of information. Although the name Data Science seems to connect most strongly with areas such as databases and computer science, many different kinds of skills - including non-mathematical skills - are needed.



- Data Science (DS)
- Business Intelligence (BI)
- Artificial Intelligence (AI)

2016 – Chuck Russell @cichuck

Methods/Techniques	Discipline
Unsupervised Machine Learning	AI
Reinforcement Learning	AI
Reporting/Decision Support	BI
Multi-Dimensional Analysis	BI
Natural Language (Query)	BI-AI
Data Management	BI-DS-AI
Big Data Management	BI-DS-AI
Inferential Statistics	DS
Predictive Modeling	DS
Supervised Machine Learning	DS-AI
Descriptive Statistics	DS-BI
Data Discover/Query	DS-BI
Data Visualization	DS-BI

Thinking like a **data scientist**  
means more than being able to  
program a computer.

It requires thinking at multiple  
levels of abstraction called  
**Computational Thinking**

**Viewpoint** | Jeannette M. Wing

## Computational Thinking

*It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use.*



Computational thinking builds on the power and limits of computing processes, whether they are executed by a human or by a machine. Computational methods and models give us the courage to solve problems and design systems that no one of us would be capable of tackling alone. Computational thinking confronts the niddle of machine intelligence: What can humans do better than computers? and What can computers do better than humans? Most fundamentally it addresses the question: What is computable? Today, we know only parts of the answers to such questions.

Computational thinking is a fundamental skill for everyone, not just for computer scientists. To reading, writing, and arithmetic, we should add computational thinking to every child's analytical ability. Just as the printing press facilitated the spread of the three Rs, what is appropriately incutious about this vision is that computing and computers facilitate the spread of computational thinking.

Computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science. Computational thinking includes a range of mental tools that reflect the breadth of the field of computer science.

Having to solve a particular problem, we might ask: How difficult is it to solve? and What's the best way to solve it? Computer science rests on solid theoretical underpinnings to answer such questions pre-

cisely. Stating the difficulty of a problem accounts for the underlying power of the machine—the computing device that will run the solution. We must consider the machine's instruction set, its resource constraints, and its operating environment.

In solving a problem efficiently, we might further ask whether an approximate solution is good enough, whether we can use randomization to our advantage, and whether false positives or false negatives are allowed. Computational thinking is reformulating a seemingly difficult problem into one we know how to solve, perhaps by reduction, embedding, transformation, or simulation.

Computational thinking is thinking recursively. It is parallel processing. It is interpreting code as data and data as code. It is type checking as the generalization of dimensional analysis. It is recognizing both the virtues and the dangers of aliasing, or giving someone or something more than one name. It is recognizing both the cost and power of indirect addressing and procedure call. It is judging a program not just for correctness and efficiency but for aesthetics, and a system's design for simplicity and elegance.

Computational thinking is using abstraction and decomposition when attacking a large complex task or designing a large complex system. It is separation of concerns. It is choosing an appropriate representation for a problem or modeling the relevant aspects of a problem to make it tractable. It is using invariants to describe a system's behavior succinctly and declaratively. It is having the confidence we can safely use, modify, and influence a large complex system without understanding its every detail. It is

Computational thinking is a way of solving problems, designing systems, & understanding human behavior that draws on concepts fundamental to data science.

Viewpoint | Jeannette M. Wing

## Computational Thinking

*It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use.*



Computational thinking builds on the power and limits of computing processes, whether they are executed by a human or by a machine. Computational methods and models give us the courage to solve problems and design systems that no one of us would be capable of tackling alone. Computational thinking confronts the niddle of machine intelligence: What can humans do better than computers? and What can computers do better than humans? Most fundamentally it addresses the question: What is computable? Today, we know only parts of the answers to such questions.

Computational thinking is a fundamental skill for everyone, not just for computer scientists. To reading, writing, and arithmetic, we should add computational thinking to every child's analytical ability. Just as the printing press facilitated the spread of the three Rs, what is appropriately incutious about this vision is that computing and computers facilitate the spread of computational thinking.

Computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science. Computational thinking includes a range of mental tools that reflect the breadth of the field of computer science.

Having to solve a particular problem, we might ask: How difficult is it to solve? and What's the best way to solve it? Computer science rests on solid theoretical underpinnings to answer such questions pre-

cisely. Stating the difficulty of a problem accounts for the underlying power of the machine—the computing device that will run the solution. We must consider the machine's instruction set, its resource constraints, and its operating environment.

In solving a problem efficiently, we might further ask whether an approximate solution is good enough, whether we can use randomization to our advantage, and whether false positives or false negatives are allowed. Computational thinking is reformulating a seemingly difficult problem into one we know how to solve, perhaps by reduction, embedding, transformation, or simulation.

Computational thinking is thinking recursively. It is parallel processing. It is interpreting code as data and data as code. It is type checking as the generalization of dimensional analysis. It is recognizing both the virtues and the dangers of aliasing, or giving someone or something more than one name. It is recognizing both the cost and power of indirect addressing and procedure call. It is judging a program not just for correctness and efficiency but for aesthetics, and a system's design for simplicity and elegance.

Computational thinking is using abstraction and decomposition when attacking a large complex task or designing a large complex system. It is separation of concerns. It is choosing an appropriate representation for a problem or modeling the relevant aspects of a problem to make it tractable. It is using invariants to describe a system's behavior succinctly and declaratively. It is having the confidence we can safely use, modify, and influence a large complex system without understanding its every detail. It is

LIBRARY

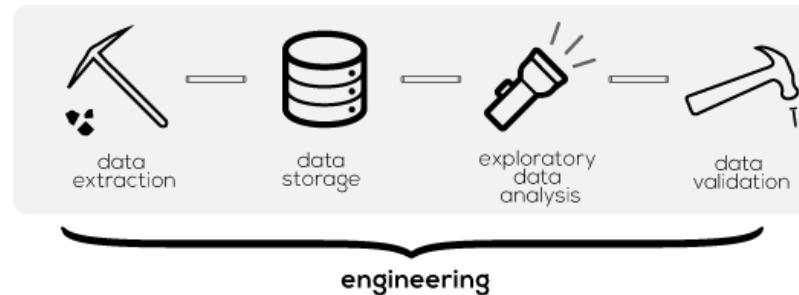
# Computational Thinking

Computational thinking richt zich op de vaardigheden die essentieel zijn om problemen op te lossen waarbij veel informatie, variabelen en rekenkracht nodig zijn.

Het is daarbij belangrijk om te begrijpen hoe informatie tot stand komt zodat je computersystemen kan benutten voor probleemoplossen, voor het denken in stappen en daarmee in voorwaardelijkheden voor volgorde van de benodigde gegevens.

Computertechnologie gebruiken bij het zoeken naar oplossingen ---*Data Science*--- betekent inzicht krijgen in algoritmes (een reeks instructies om vanaf een beginpunt een bepaald doel te bereiken) en procedures (een verzameling activiteiten die in een bepaalde volgorde moet worden uitgevoerd).

# Engineering a IoT Data Pipeline



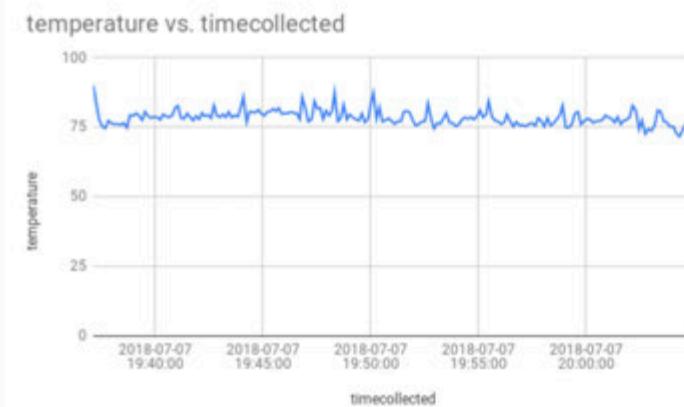
<https://alexpertalia.com/posts/2016/6/22/the-data-pipeline-part-2>

## Building and Running a complete data pipeline

It starts with an internet of things (IoT) device which captures temperature, leverages IoT Core to securely publish the data to a message queue where it will then be transported into a data warehouse.

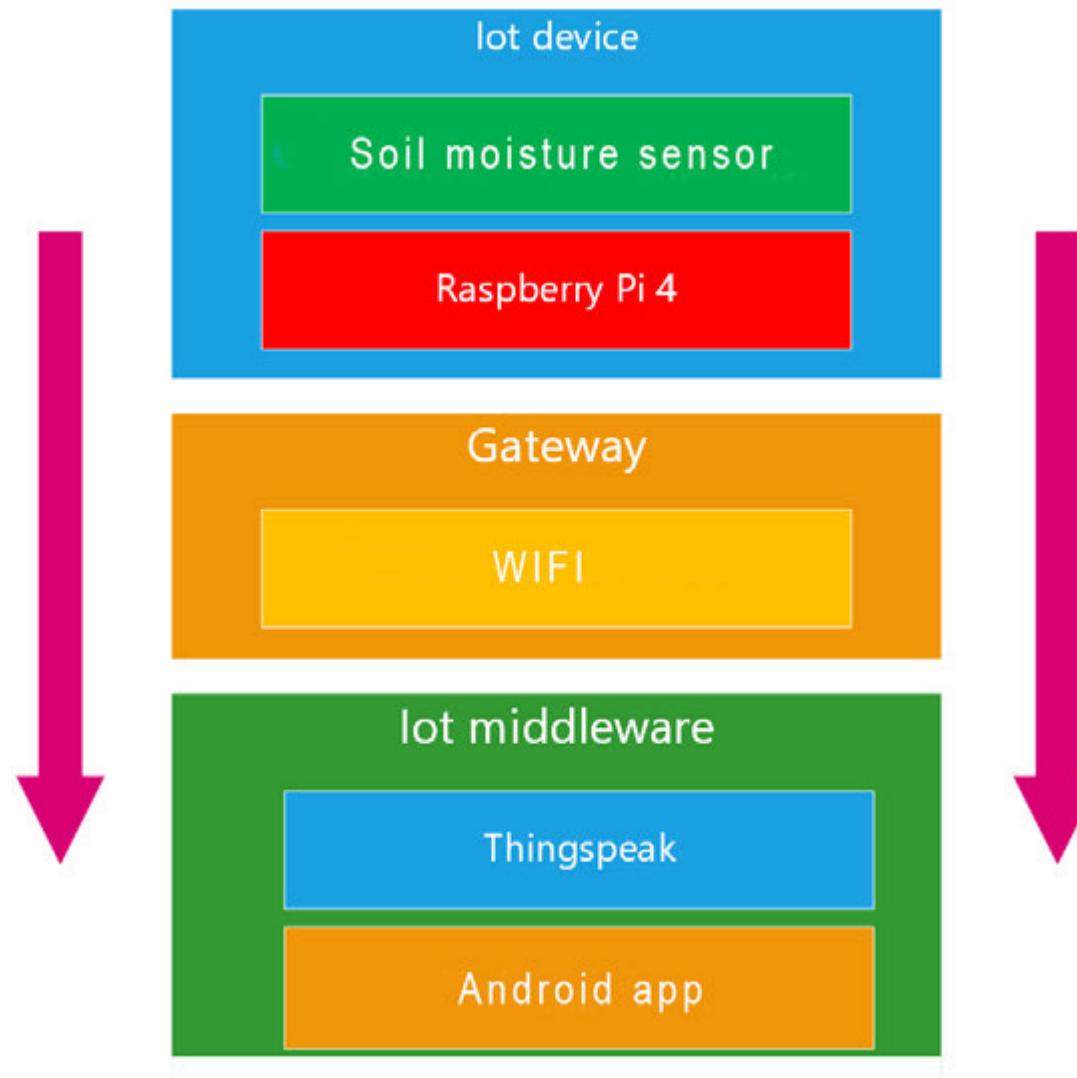


At the end of this blog we will be able to visualise our data using bigquery and charts in google sheets as shown in the following picture:



[https://uplua.github.io/jekyll/update/2019/06/23/  
Fourth-Week\\_report.html](https://uplua.github.io/jekyll/update/2019/06/23/Fourth-Week_report.html)

# Engineering a IoT Data Pipeline



IoT

## Surface Web

YAHOO!

Google

reddit

CNN.com

bing

## Deep Web

Academic databases  
Medical records  
Financial records  
Legal documents  
Some scientific reports  
Some government reports  
Subscription only information  
Some organization-specific repositories

IoT

**96%**

of content on the  
Web (estimated)

## Dark Web

TOR  
Political protest  
Drug trafficking  
and other illegal activities

# World Wide Web (WWW) vs IoT

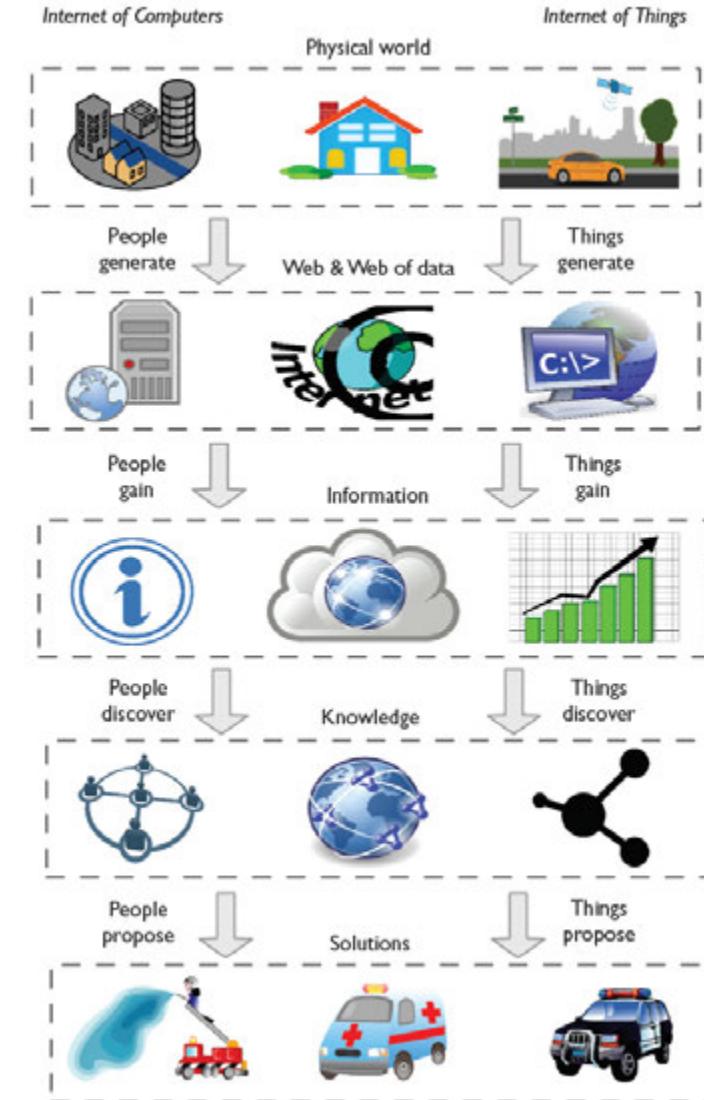
## WWW:

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

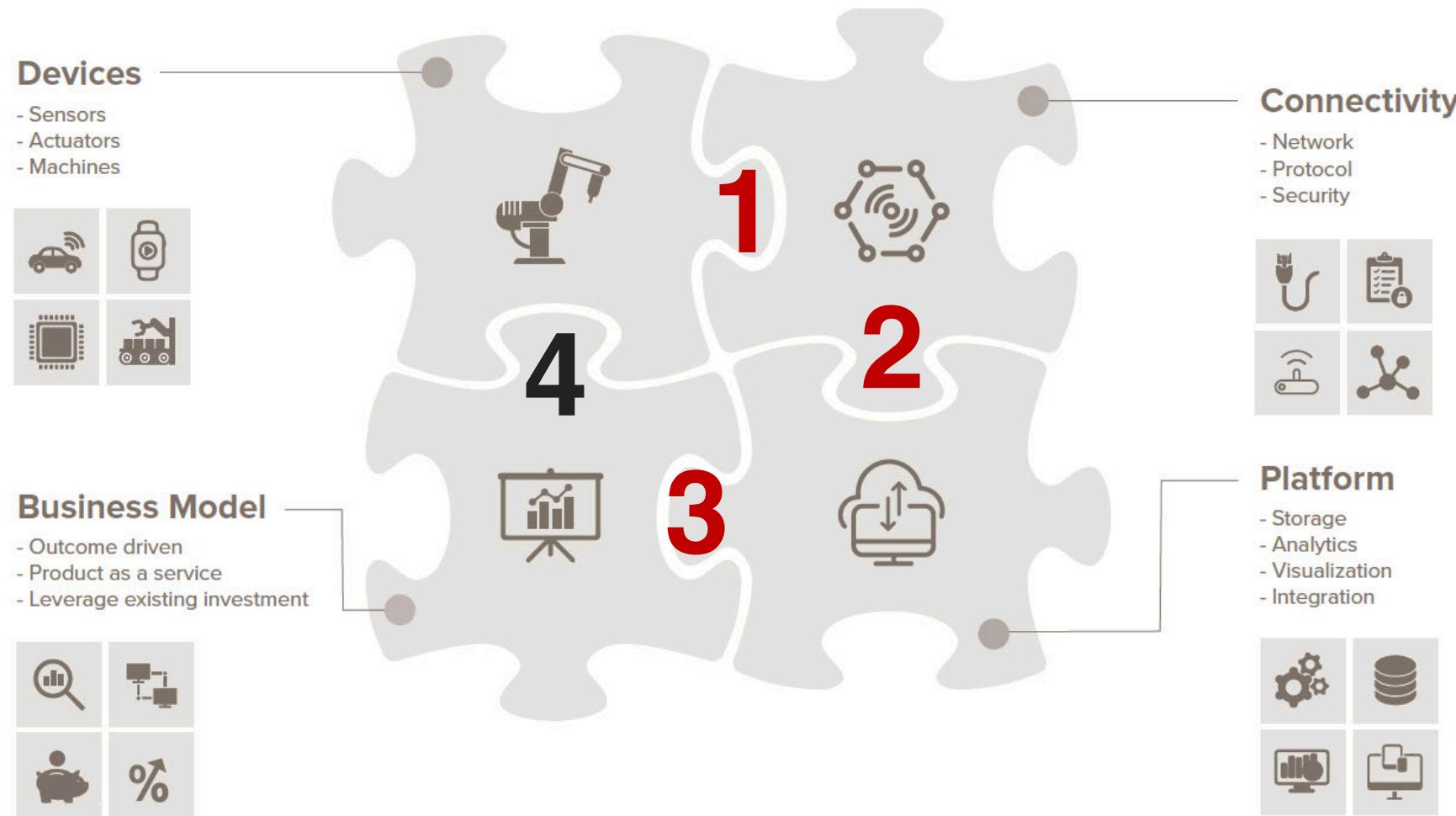
## IoT:

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.



# 3 Pillars of the IoT



**Smart Systems and the Internet of Things  
are driven by a combination of:**

**1** SENSORS  
& ACTUATORS

**2** CONNECTIVITY

**3** PEOPLE &  
PROCESSES

<http://postscapes.com/what-exactly-is-the-internet-of-things-infographic>

# ACTUATORS



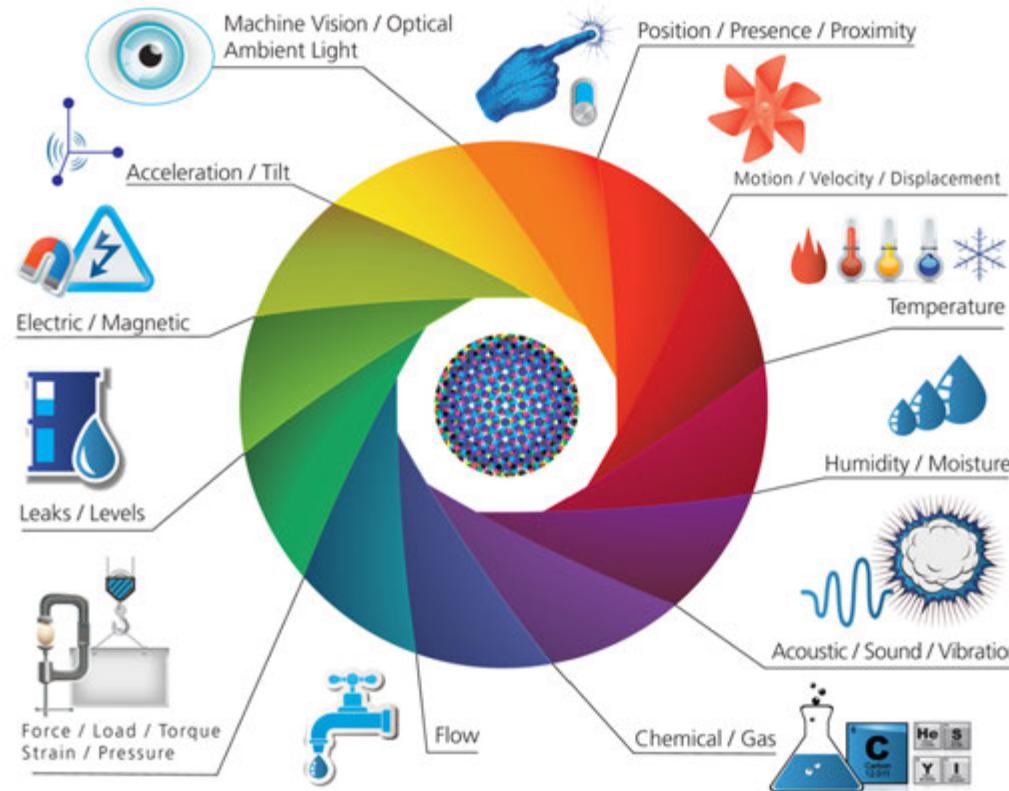
An actuator is a component of a machine that is responsible for moving or controlling a mechanism or system.

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.

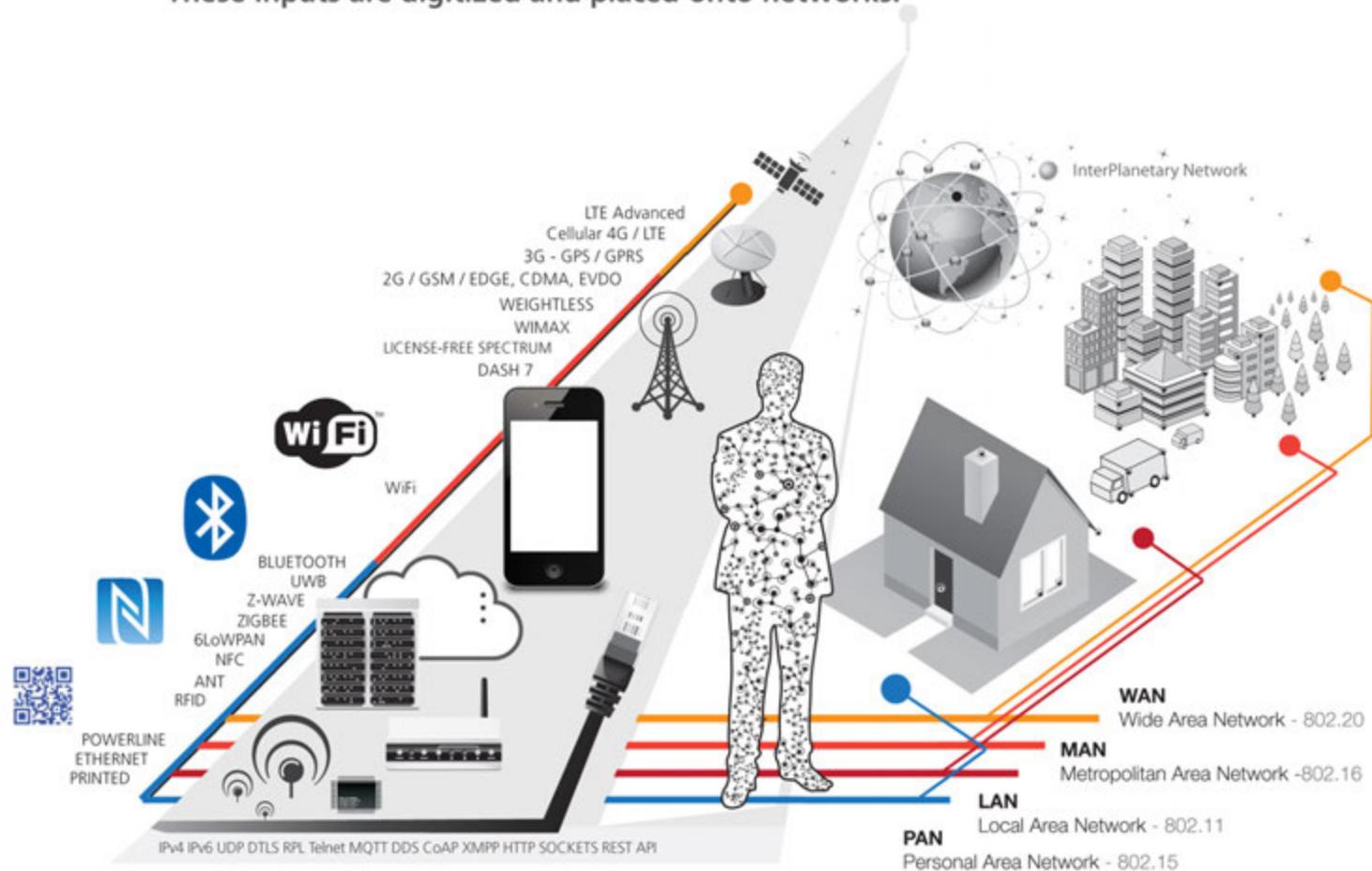
## 1 SENSORS & ACTUATORS

We are giving our world a **digital nervous system**. Location data using GPS sensors. Eyes and ears using cameras and microphones, along with sensory organs that can measure everything from temperature to pressure changes.



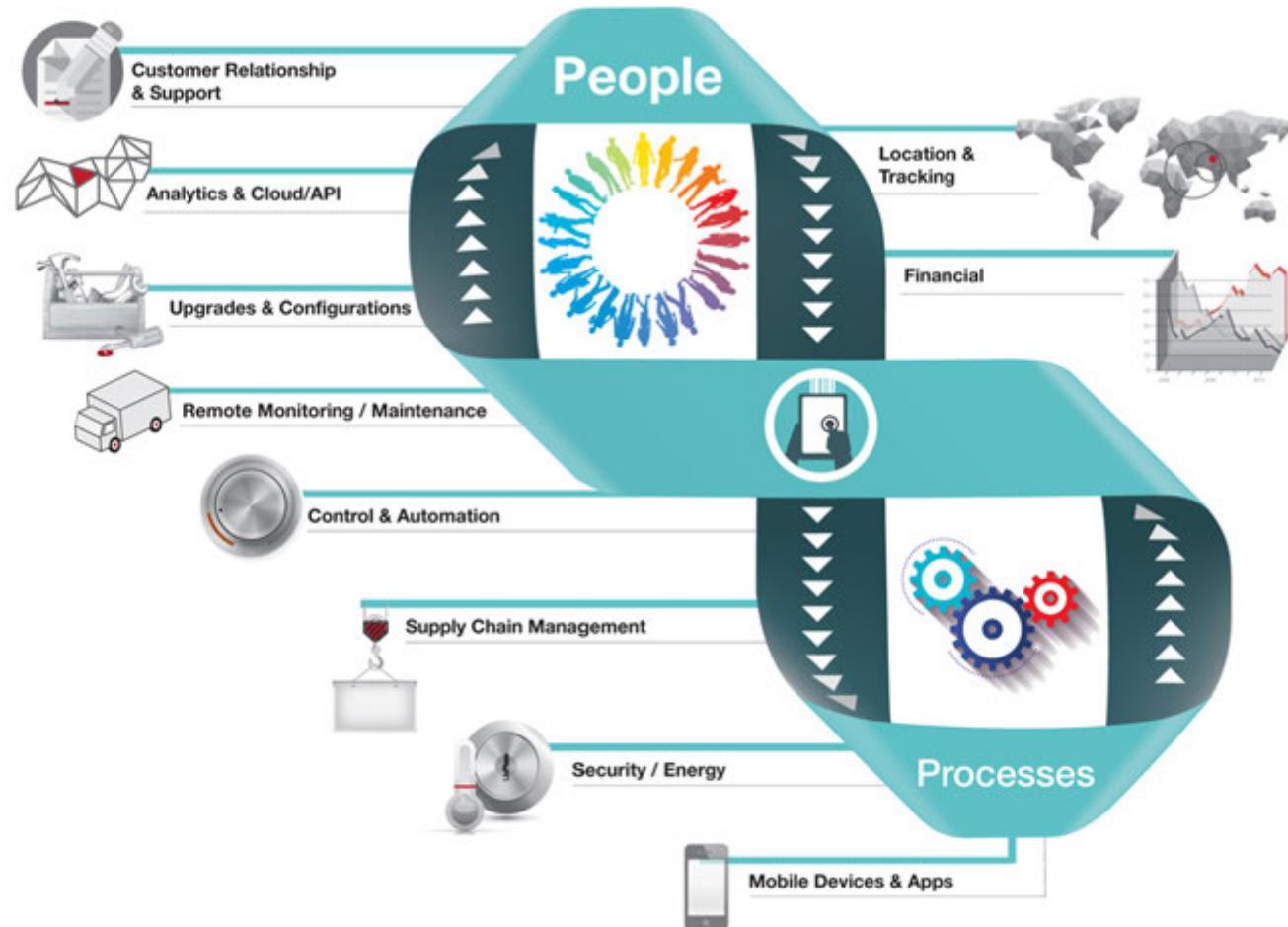
## 2 CONNECTIVITY

These inputs are digitized and placed onto networks.



## 3 PEOPLE & PROCESSES

These networked inputs can then be combined into bi-directional systems that integrate data, people, processes and systems for better decision making.



# IoT Platforms + User-Types



Hobbyists



Consumer



Industrial



Industry

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.

# 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



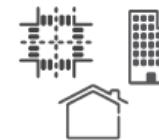
- Micro payments
- Retail logistics
- Product life-cycle info
- Shopping assistance

## 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

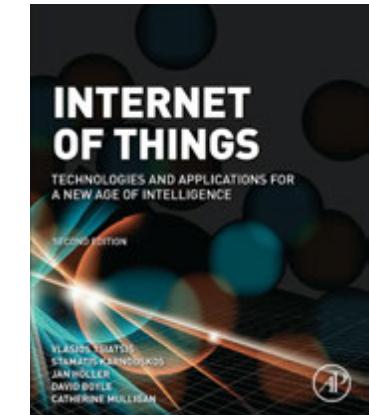


- Robotics
- Manufacturing
- Natural resources
- Remote operations
- Automation
- Heavy machinery

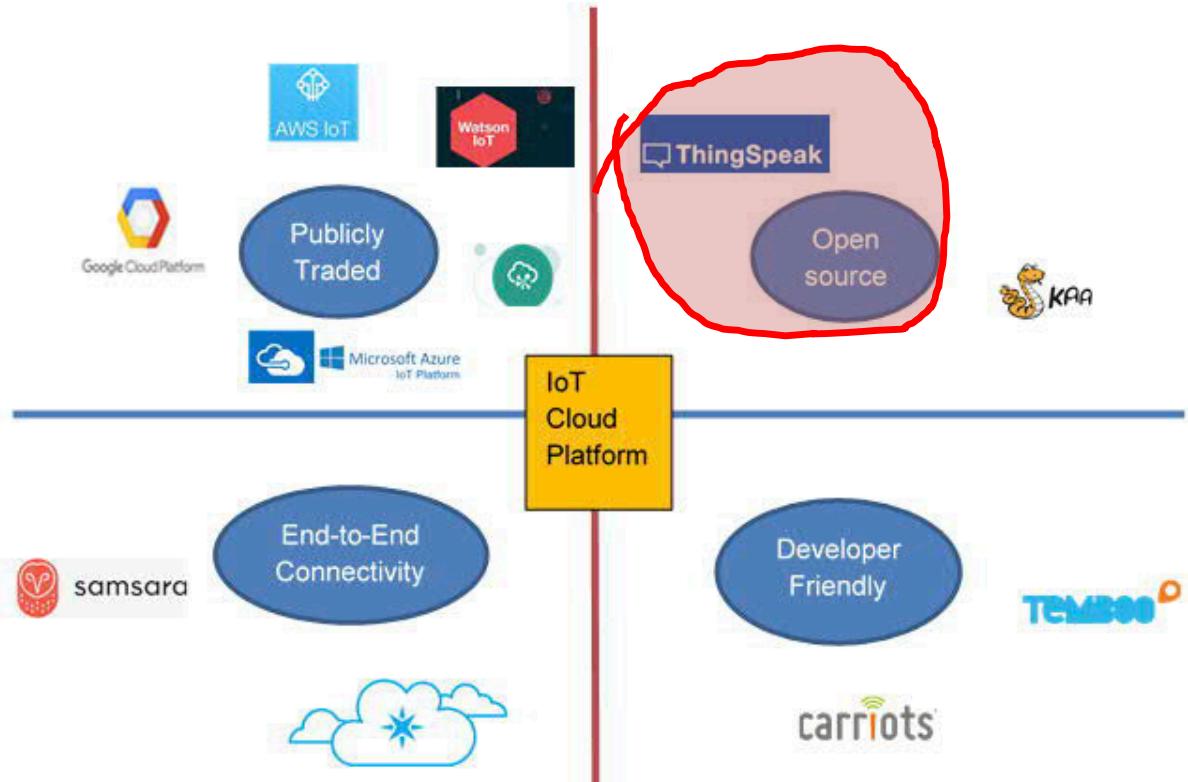
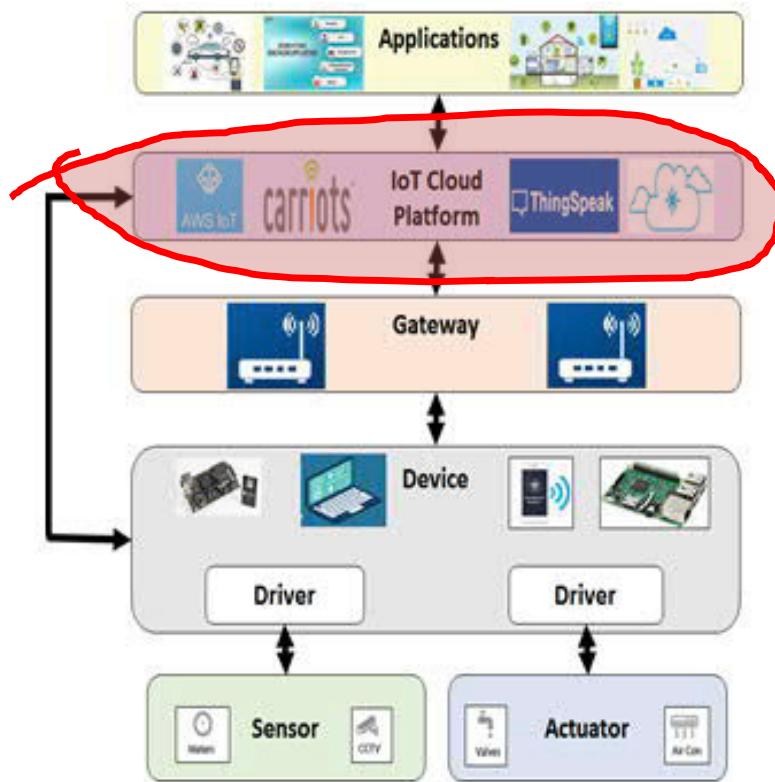
## Agriculture



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



# IoT Platforms == MiddleWare



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

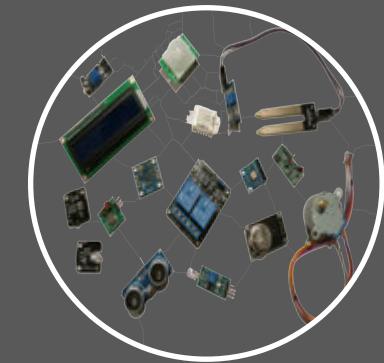
# Technologies enabling IoT

WIRELESS TECHNOLOGIES	TCP/IP PROTOCOLS	AUTHORIZATION & AUTHENTICATION TECHNOLOGIES
802.11, 802.15.4, Zigbee, Bluetooth, BLE, CDMA, GSM, LTE	Sockets, IPv4, IPv6, TCP, UDP, ICMP, QoS, SNMP, IMAP, POP3, IPMI, etc.	Oauth2, PAM, LDAP
NETWORK TECHNOLOGIES	MOBILE TECHNOLOGIES	PLATFORMS AND CPU ARCHITECTURES
Ethernet, CAN, rs485/rs422/rs232, 1-wire, i2c, SPI, ModBus/MudBusRTU, IPMI, iSCSI	Android SDK, Qt, iOS SDK, Objective C, Java, Swift	ARM, X86, PowerPC, AVR, PIC
WEB SERVICE TECHNOLOGIES	PROTOCOL TECHNOLOGIES	PROGRAM / SCRIPTING LANGUAGES
SOAP, REST, WSDL, XML, JSON, UDDI, WebSockets	HTTP, JMS, AMQP, D-Bus	Java, C/C++, C#, JavaScript, Ruby, Groovy, Python, Tcl/Tk, ASM, Bash

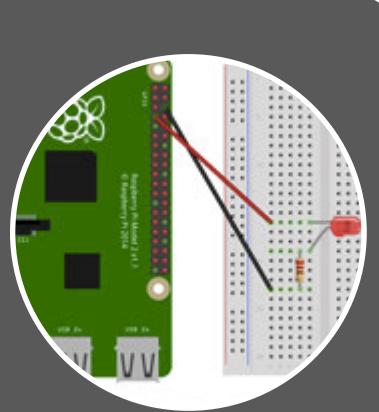
<https://doi.org/10.15779/Z38WW0V>

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

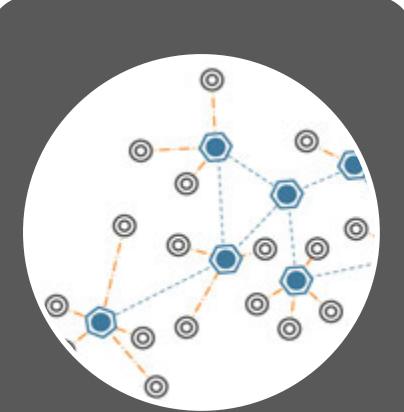
# Engineering IoT Data Pipelines



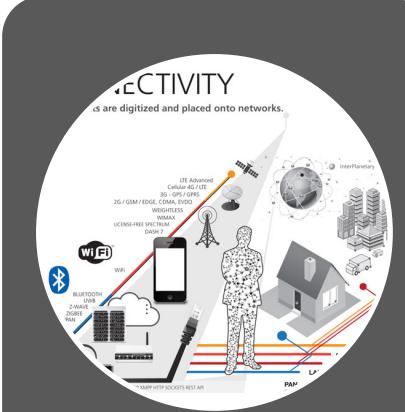
Sensors &  
Actuators



Micro-  
Processors/  
Controllers  
(Development  
Boards)



Network /  
Protocols



Platforms,  
Cloud &  
Data Analytics

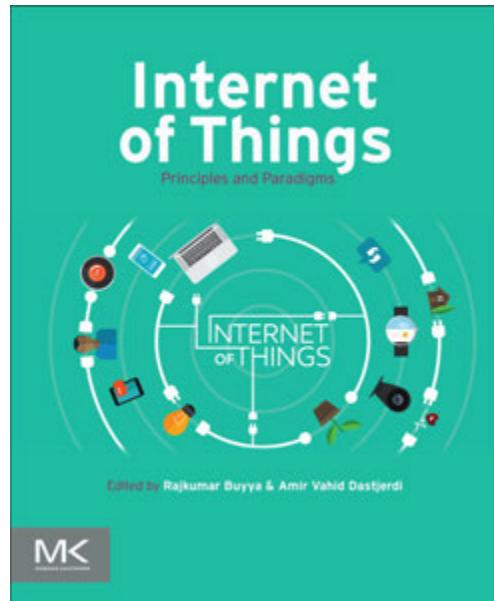
# IoT Communication

***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*.

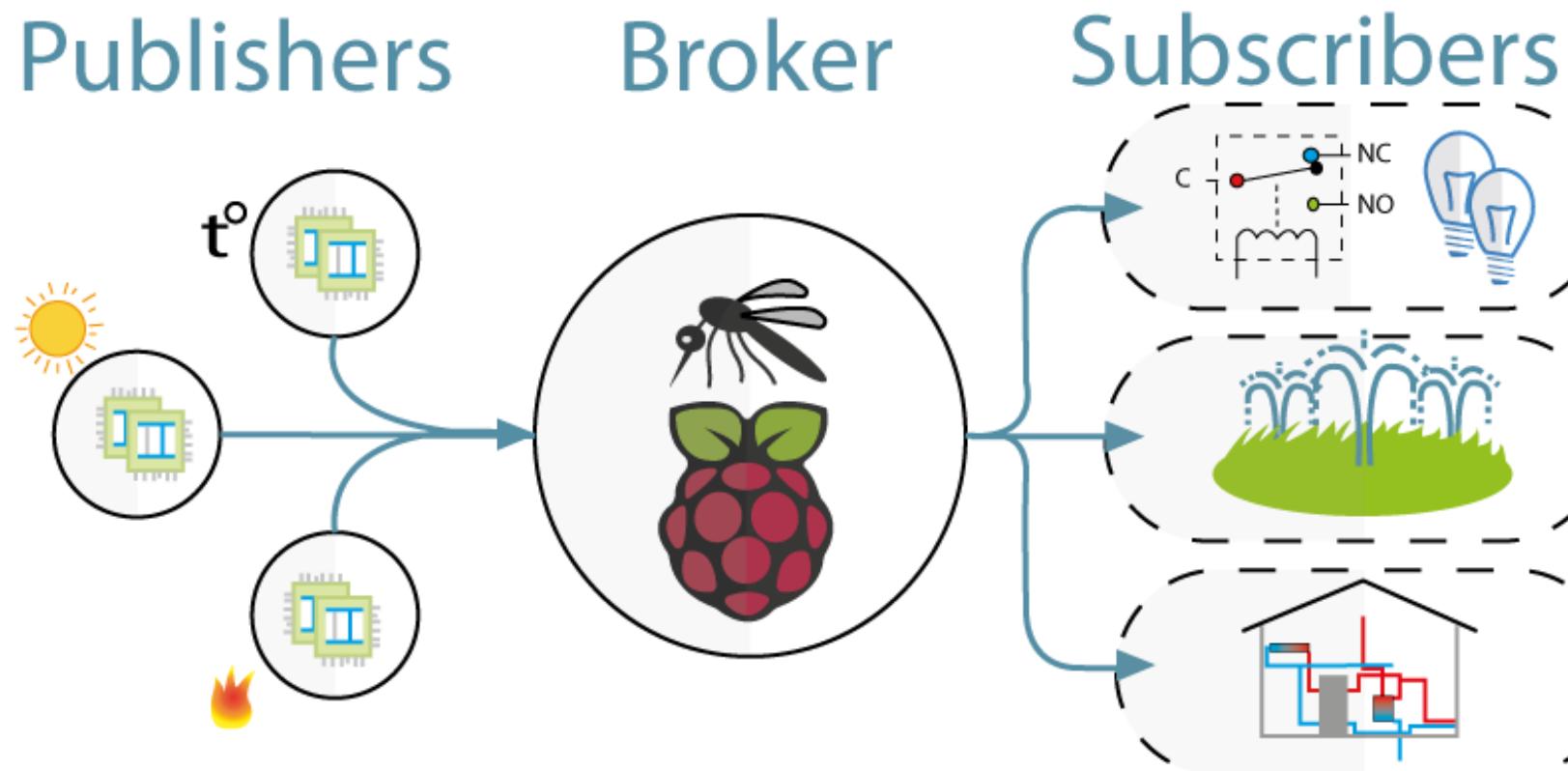
***The Subscriber:*** The subscriber declares its topics of interest to the broker, and the broker sends messages published to those topics.

***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.

# IoT-device communication protocols

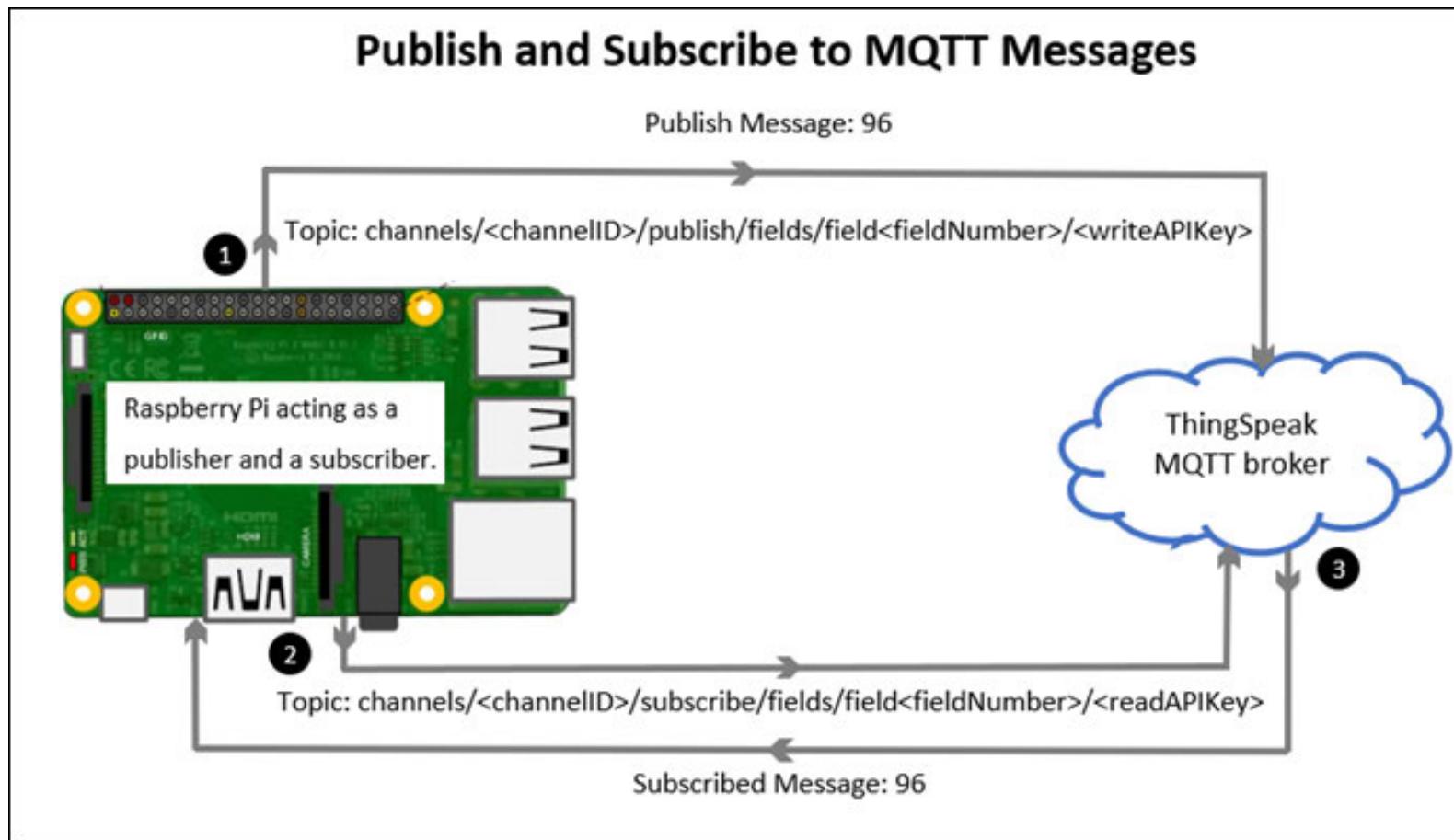


Protocol Name	Transport Protocol	Messaging Model	Security	Best-Use Cases	Architecture
AMPQ	TCP	Publish/Subscribe	High-Optional	Enterprise integration	P2P
CoAP	UDP	Request/Response	Medium-Optional	Utility field	Tree
DDS	UDP	Publish/Subscribe and Request/Response	High-Optional	Military	Bus
MQTT	TCP	Publish/Subscribe 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



# Setting Up Your IoT Device

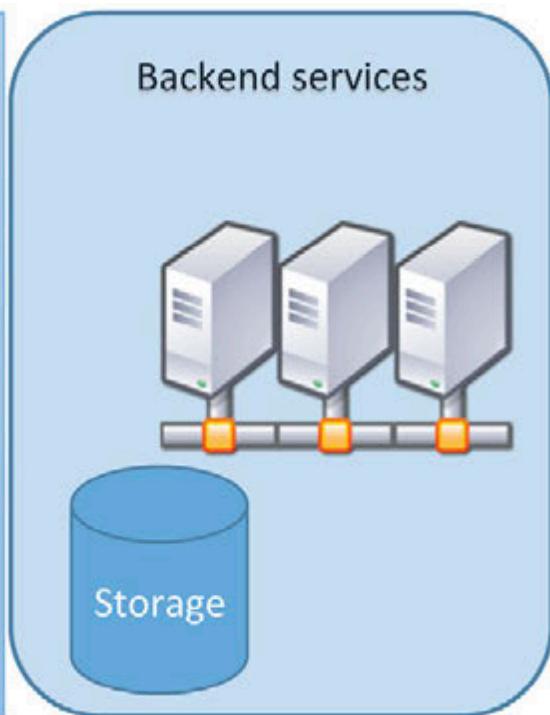
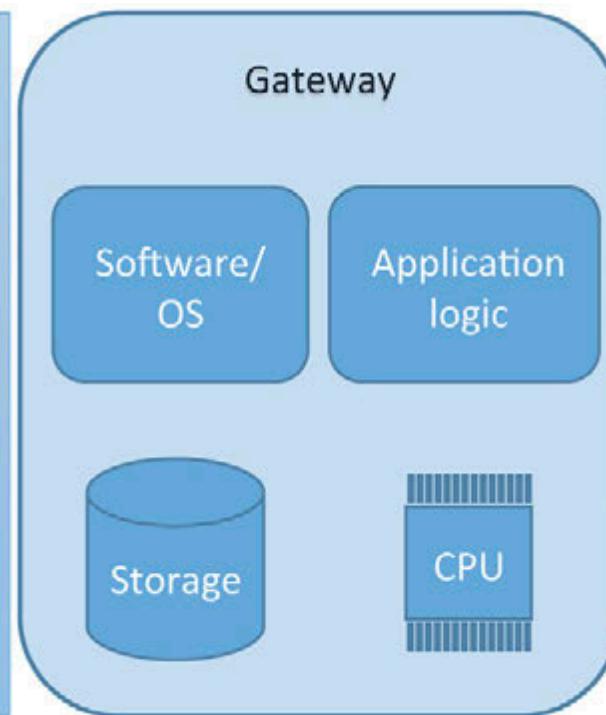
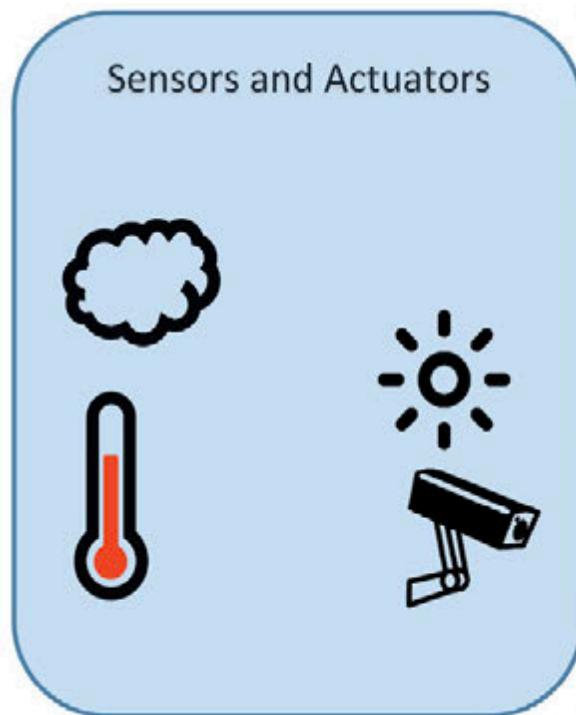
**{IoT device == micro-controller== Raspberry Pi}**



# IoT communication

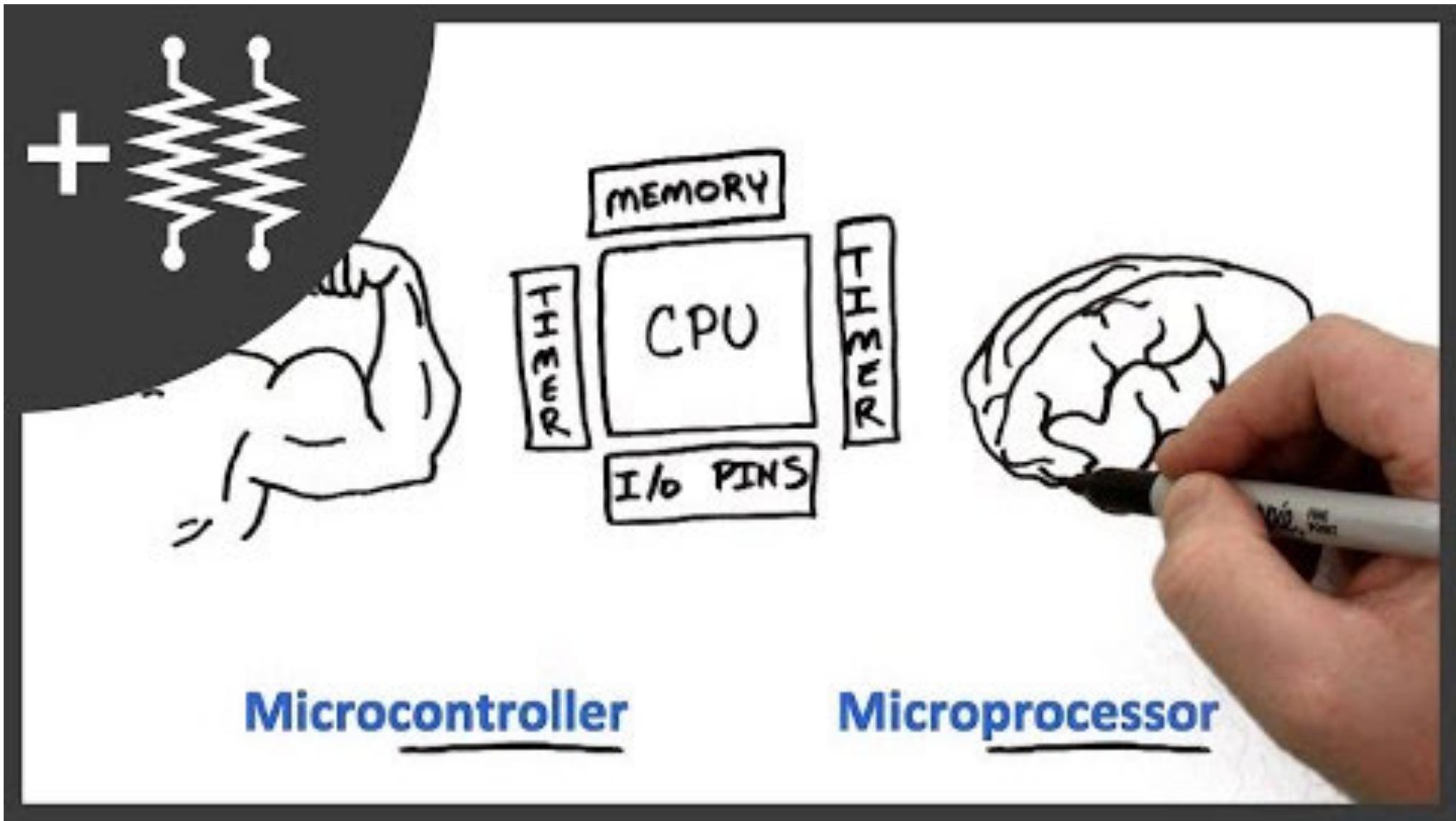
*Publisher*

*Broker*



SENSOR TO GATEWAY COMMUNICATION

# Getting to Know Your IoT Device



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

# Raspberry Pi

<https://projects.raspberrypi.org/en/projects/raspberry-pi-setting-up>

## Introduction

- What you will need
- Set up your SD card
- Connect your Pi
- Start up your Pi
- Finish the setup
- Where to find help
- What next?

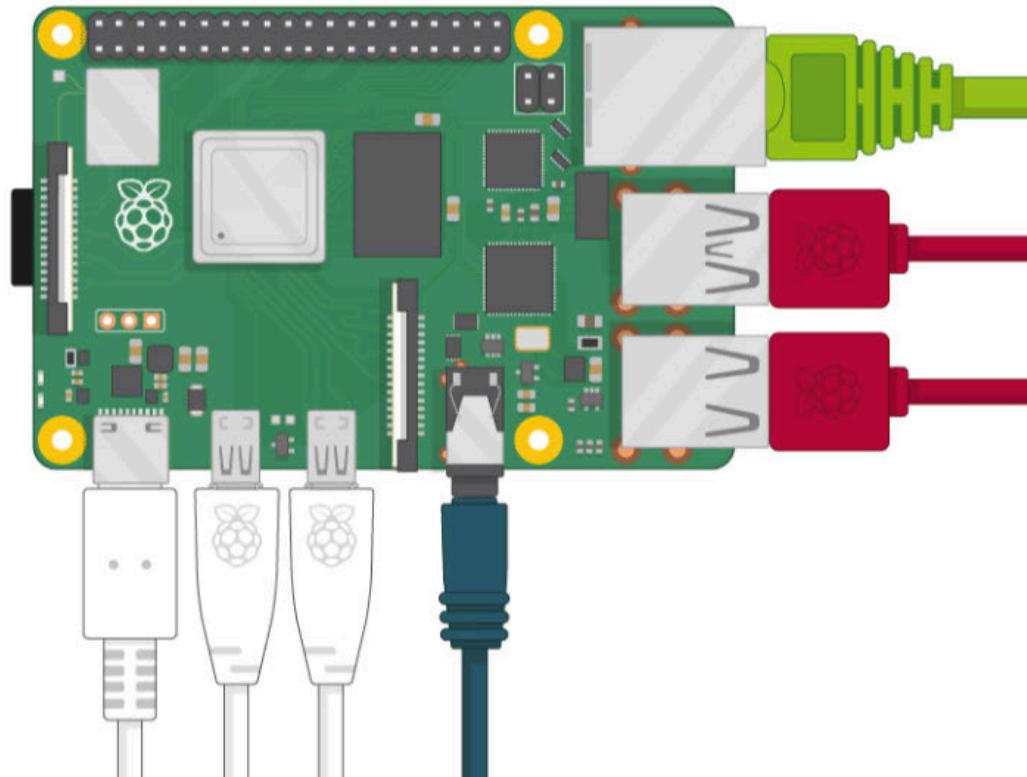
## What is a Raspberry Pi?

The Raspberry Pi is a low cost, **credit-card sized computer** that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It's capable of doing everything you'd expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.

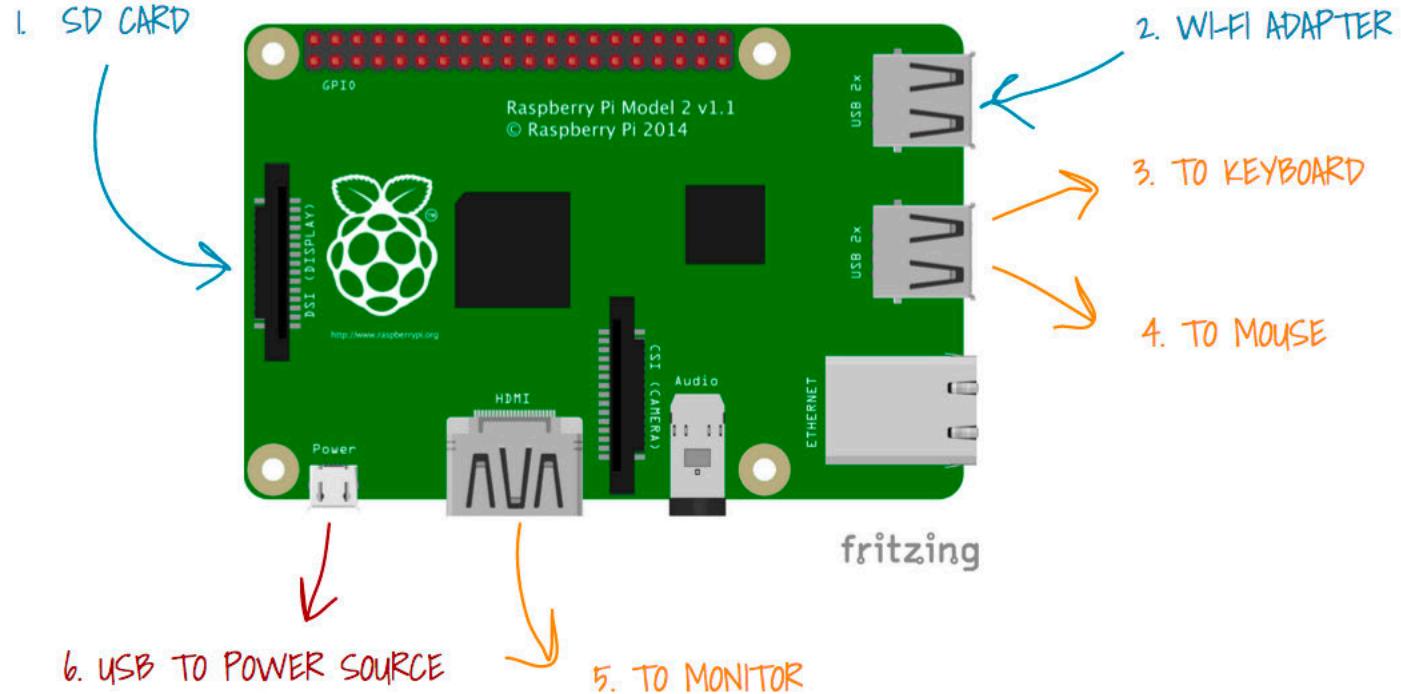
What's more, the Raspberry Pi has the ability to interact with the outside world, and has been used in a wide array of digital maker projects, from music machines and parent detectors to weather stations and tweeting birdhouses with infra-red cameras. We want to see the Raspberry Pi being used by kids all over the world to learn to program and understand how computers work.

## Introduction

Here you'll learn about your Raspberry Pi, what things you need to use it, and how to set it up.



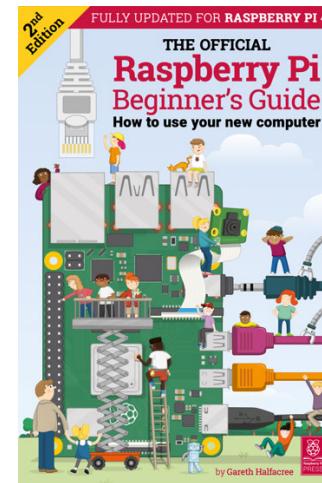
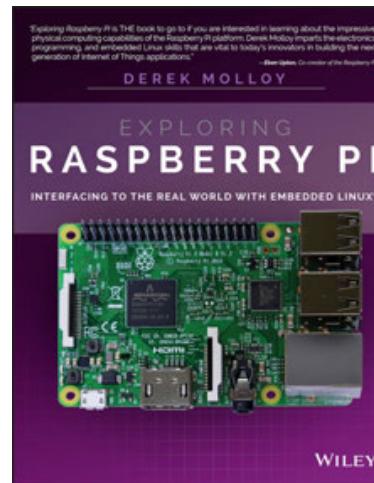
# Main Components Raspberry Pi



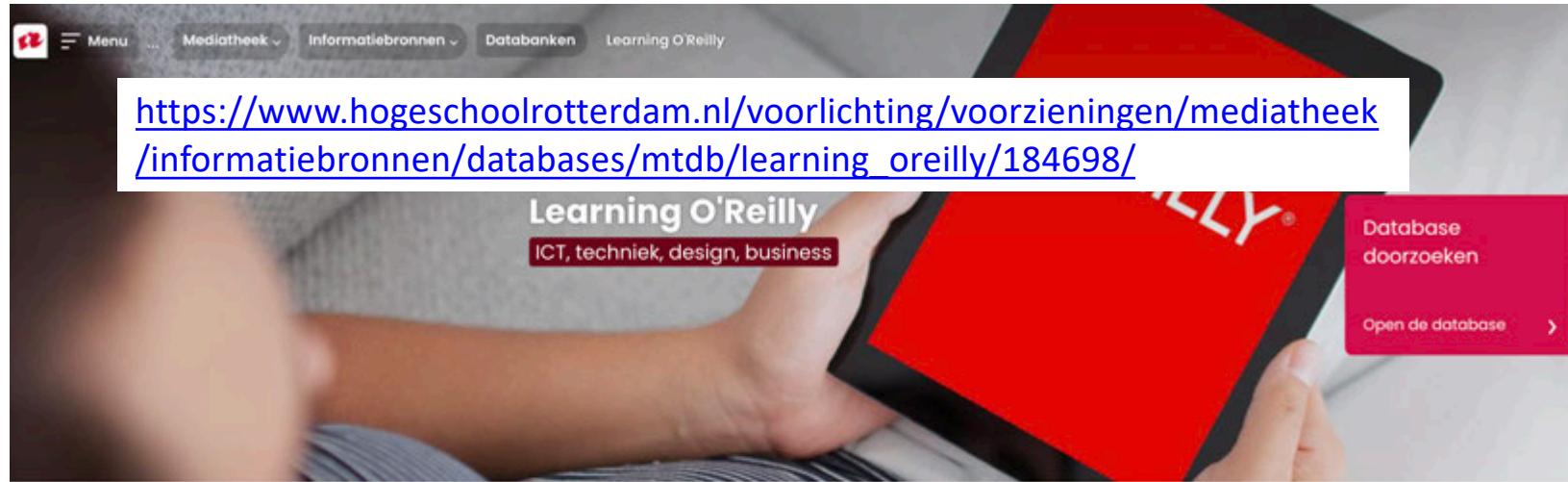
<https://www.pubnub.com/blog/internet-of-things-101-getting-started-w-raspberry-pi/>

**Things you  
should  
Do**

# Start Reading TEXTBOOKS!



# How to get Access to TEXTBOOKS!



Learning O'Reilly is een Engelstalig online leerplatform gericht op ICT vaardigheden. Het biedt tienduizenden e-books en vele video's, case studies en 'learning paths', waarbij je zelf je voortgang kunt monitoren.

## Toegang

### Via de website

De Learning O'Reilly website is gekoppeld aan de Hogeschool Rotterdam login. Kom je toch een O'Reilly inlogscherm tegen, vul dan alleen je @hr.nl e-mailadres in en klik op **Sign In with Single Sign On**. Krijg je een blanco pagina te zien? Schakel eventuele adblockers uit!

### Via de O'Reilly app

Leer wat je wil, waar je wil. Met de O'Reilly app download je e-books en ga je verder met een playlist waar je op een ander apparaat gebleven was.



- 2 Gebruik je @hr.nl e-mailadres om in te loggen.

# Hardware YOU NEED:

## Raspberry Pi 3 Model A+ or B+

De **Raspberry Pi 3 Model A+** is het nieuwste board in de Raspberry Pi 3 reeks en is een flinke update sinds de eerste A+ die in november 2014 werd geïntroduceerd. Het board heeft 512MB RAM en beschikt net als zijn grote broer over een 64-bit quad core processor op 1.4GHz, dual-band WiFi (2.4GHz en 5GHz) en Bluetooth 4.2/BLE.

### Specificaties

**Processor:** Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz

**Geheugen:** 512MB LPDDR2 SDRAM

### Connectiviteit:

2.4GHz en 5GHz 802.11b/g/n/ac WiFi

Bluetooth 4.2 / BLE

1x USB 2.0 poort

**Uitbreiding:** 40-pin GPIO header

### Video & Geluid:

1x full-size HDMI

DSI display poort

DSI camera poort

4-polige 3.5mm jack voor stereo audio en composiet video

**Multimedia:** H.264, MPEG-4 decode (1080p30); H.264 encode (1080p30); OpenGL ES 1.1, 2.0 gr

### Stroomvoorziening:

5V / 2.5A DC via microUSB

5V DC via GPIO header

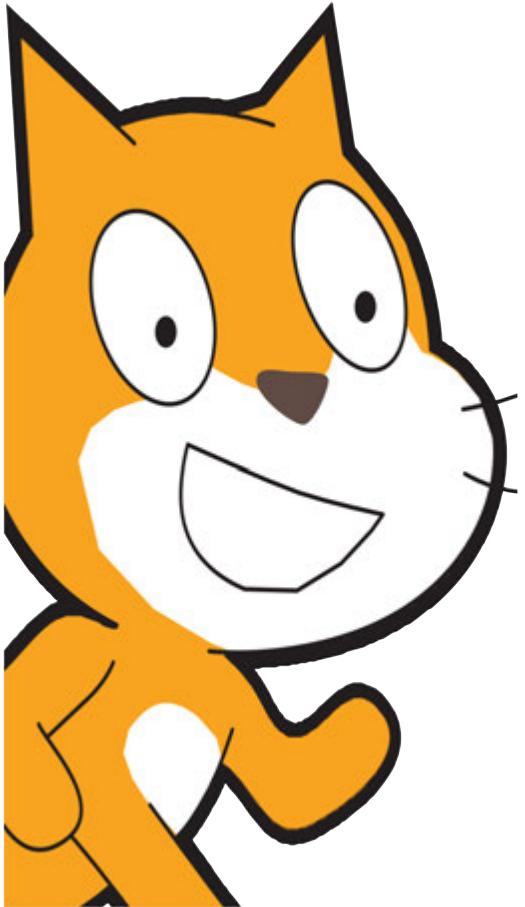
**Afmetingen:** 65x56x8.5mm

**Gewicht:** 29g



# 1<sup>st</sup> STEP Managing Data Science for IoT Projects

## →→→ Getting started with GitHub ←←←



A Computational Thinking culture has an intellectual dimension, engaging with a set of creative concepts and practices. It has a physical dimension, encouraging interactions with others through the placement of desks, chairs, and computers. Most importantly, it has an affective dimension, cultivating a sense of confidence and fearlessness.

### LEARNING OBJECTIVES

---

Students will:

- + be introduced to the concept of computational Thinking, in the context of GitHub
- + be able to imagine possibilities for their own GitHub-based computational thinking
- + become familiar with resources that support their computational thinking
- + prepare for creating GitHub projects by establishing a GitHub account, exploring GitHub, creating design journals

### Tip

Programming involves working with many different file types, each of which is indicated by its **extension** (the letters after the . in the file name, such as .pdf). It is useful to specify that your computer should show these extensions in File Explorer or Finder; see instructions for Windows<sup>a</sup> or for Mac<sup>b</sup> to enable this.

<sup>a</sup><https://helpx.adobe.com/x-productkb/global/show-hidden-files-folders-extensions.html>

<sup>b</sup>[https://support.apple.com/kb/PH25381?locale=en\\_US](https://support.apple.com/kb/PH25381?locale=en_US)

### KEY WORDS, CONCEPTS, & PRACTICES

---

- + IDE
- + GitHub
- + Visual Studio Code

## 1 INSTALLING GIT

One of the most important aspects of doing data science is keeping track of the changes that you (and others) make to your code. `git` is a *version control system* that provides a set of commands that you can use to manage changes to written code, particularly when collaborating with other programmers (version control is described in more detail in Chapter 3).

`git` comes pre-installed on Macs, though it is possible that the first time you try to use the tool you will be prompted to install the *Xcode command line developer tools* via a dialog box. You can choose to install these tools, or download the latest version of `git` online.

On Windows, you will need to download<sup>2</sup> the `git` software. Once you have downloaded the installer, double-click on your downloaded file, and follow the instructions to complete installation. This will also install a program called *Git Bash*, which provides a command line (text-based) interface for executing commands on your computer. See Section 1.1.2 for alternative and additional Windows command line tools.

On Linux, you can install `git` using `apt-get` or a similar command. For more information, see the download page for Linux.<sup>3</sup>

<sup>2</sup> `git` downloads: <https://git-scm.com/downloads>

<sup>3</sup> `git` download for Linux and Unix: <https://git-scm.com/download/linux>

## 2 CREATING A GITHUB ACCOUNT

GitHub<sup>4</sup> is a website that is used to store copies of computer code that are being managed with `git`. To use GitHub, you will need to create a free GitHub account.<sup>5</sup> When you register, remember that your profile is public, and future collaborators or employers may review your GitHub account to assess your background and ongoing projects. Because GitHub leverages the `git` software package, you don't need to install anything else on your machine to use GitHub.

<sup>4</sup> GitHub: <https://github.com>

<sup>5</sup> Join GitHub: <https://github.com/join>

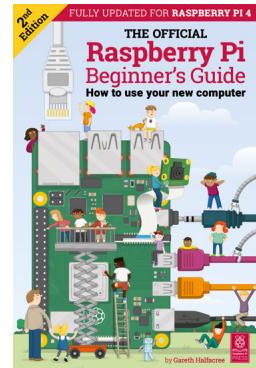
## 3 SELECTING A TEXT EDITOR

**Visual Studio Code**<sup>9</sup> (or VS Code; not to be confused with Visual Studio) is a free, open source editor developed by Microsoft—yes, really. While it focuses on web programming and JavaScript, it readily supports lots of languages, including Markdown and R, and provides a number of extensions for adding even more features. It has a similar command palette to Atom, but isn't quite as nice for editing Markdown specifically. Although fairly new, it is updated regularly and has become one of the authors' main editors for programming.

<sup>9</sup> Visual Studio Code: <https://code.visualstudio.com>

# 2<sup>nd</sup> STEP Managing Data Science for IoT Projects

→→→ Getting to know the **Raspberry Pi** ←←←



Use the **Raspberry Pi Beginner's Guide** +  
<https://www.raspberrypi.org/downloads/raspbian/> to install Raspbian on your own device.

## LEARNING OBJECTIVES

---

Students will:

- + be introduced to the concept of computational Thinking, in the context of **Raspberry Pi**
- + be able to imagine possibilities for their own **Raspberry Pi**-based data science for IoT project
- + become familiar with Raspbian
- + Programming with Python using the Thonny Pyton IDE

**Raspbian**

Raspbian is the Foundation's official supported operating system. You can install it with [NOOBS](#) or download the image below and follow our [installation guide](#).

Raspbian comes pre-installed with plenty of software for education, programming and general use. It has Python, Scratch, Sonic Pi, Java and more.

The Raspbian with Desktop image contained in the ZIP archive is over 4GB in size, which means that these archives use features which are not supported by older unzip tools on some platforms. If you find that the download appears to be corrupt or the file is not unzipping correctly, please try using [7-Zip](#) (Windows) or [The Unarchiver](#) (Macintosh). Both are free of charge and have been tested to unzip the image correctly.



Raspbian Buster with desktop and recommended software  
Image with desktop and recommended software based on Debian Buster

Version: September 2019  
Release date: 2019-09-24  
Kernel version: 4.19  
Size: 2541 MB  
[Release Notes](#)

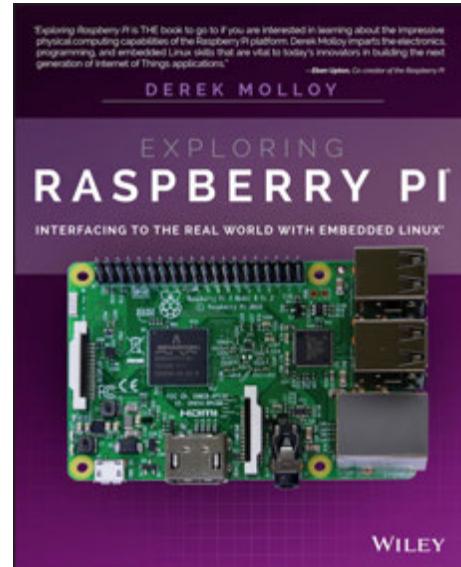
[Download Torrent](#) [Download ZIP](#)



Raspbian Buster with desktop  
Image with desktop-based on Debian Buster

Version: September 2019  
Release date: 2019-09-24  
Kernel version: 4.19  
Size: 1123 MB  
[Release Notes](#)

[Download Torrent](#) [Download ZIP](#)



"Exploring Raspberry Pi is THE book to go to if you are interested in learning about the impressive physical computing capabilities of the Raspberry Pi platform. Derek Molloy imparts the electronics, programming, and embedded Linux skills that are vital to today's innovators in building the next generation of Internet of Things applications."

DEREK MOLLOY

**EXPLORING RASPBERRY PI**

INTERFACING TO THE REAL WORLD WITH EMBEDDED LINUX™

WILEY

## KEY WORDS, CONCEPTS, & PRACTICES

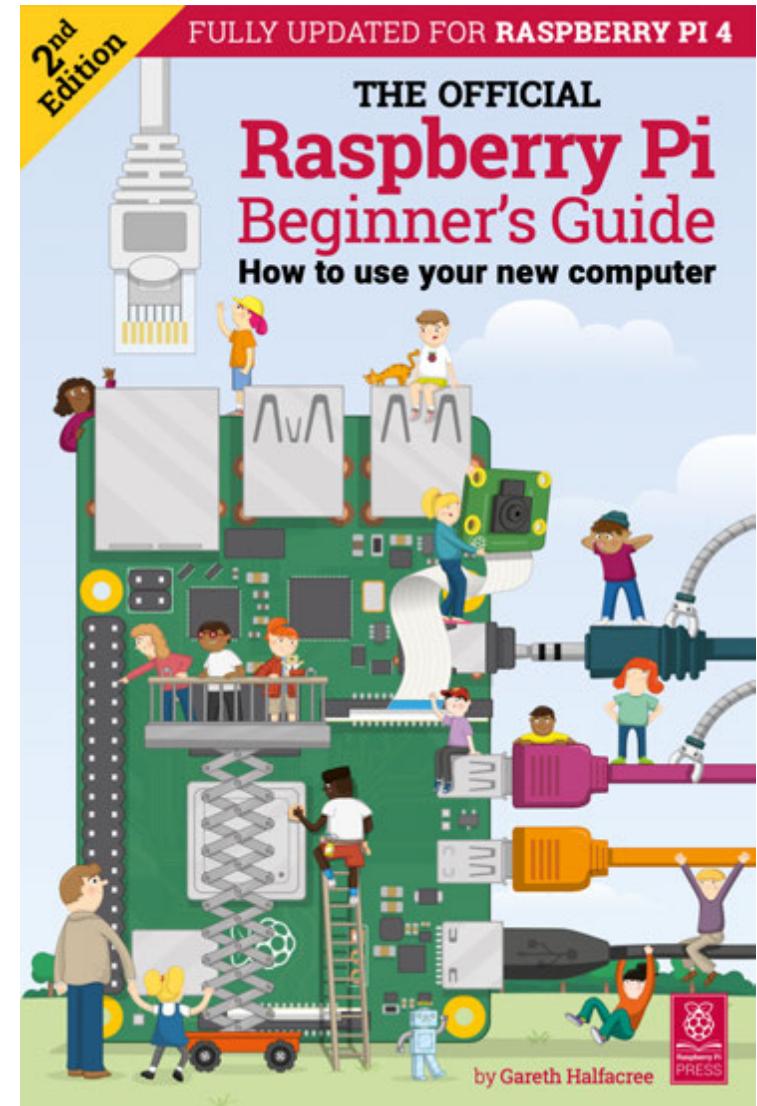
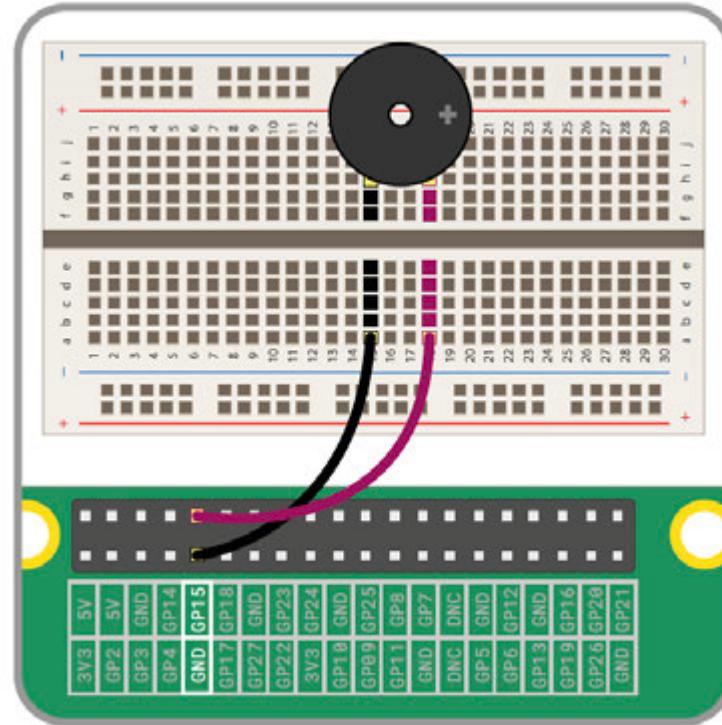
---

- + IDE
- + Python
- + Raspbian

# **Preview Lesson Two**

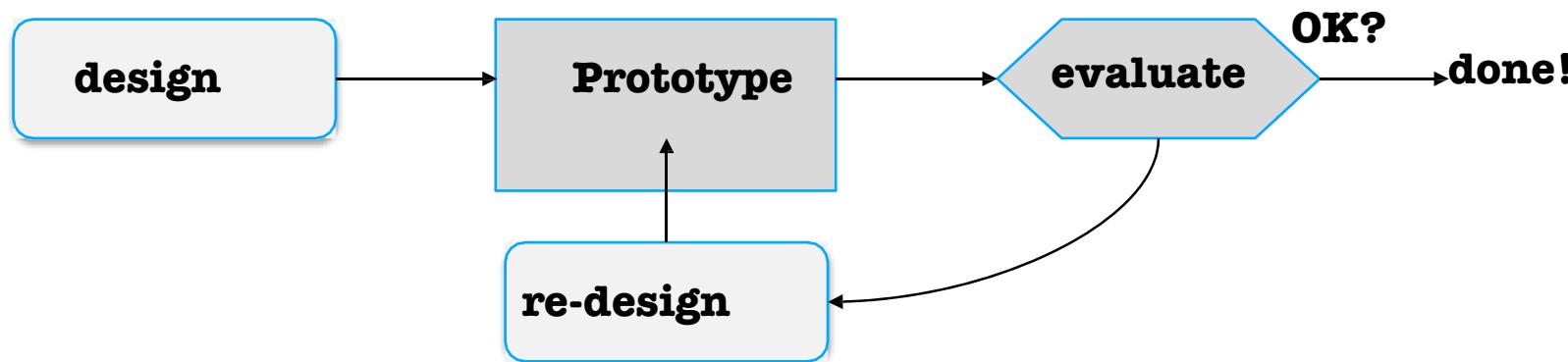
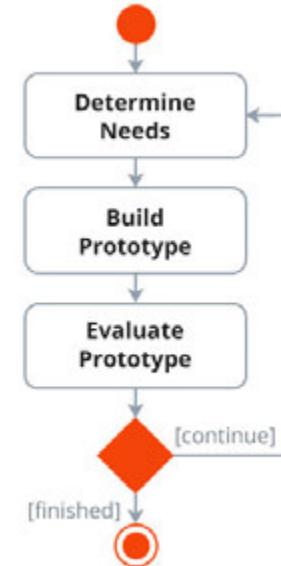
# NEXT TIME

# Rapid Prototyping



# PROTOTYPING

- You never get it right first time
- If at first you don't succeed ...



# PITFALLS OF PROTOTYPING

- Moving little by little ... but to where



1. need a good start point
2. need to understand what is wrong

**Google**  
for Entrepreneurs

# Rapid Prototyping

## Part 1: Paper Prototyping



# Prototyping Internet of Things



# Top 10 Technology Blogs/Magazines

#1 **The Verge** – Founded in 2011 The Verge (always in italics!) covers broad spectrum of topics with features, product reviews and podcasts across tech and science and culture, including the arts.

#2 **Mashable** – Founded in 2005 in New York Mashable provides plenty of news and resources for the “Connected Generation.” It’s got perhaps one of the most engaged audiences of all tech communities with 20m monthly unique visitors and over 6m followers on social media platforms.

#3 **TechCrunch** – Like Mashable founded in 2005. TechCrunch enjoys over 12m unique visitors a month, with its community numbering over 2m followers on social media networks. As well as often breaking stories on tech businesses – acquisition and fundraising – its CrunchBase database has become the place to go for information on tech companies, funding and major stories.

#4 **The Next Web** – Founded in 2008 The Next Web has more than 7.2m monthly unique visitors. The Next Web prides itself on giving an international angle to Internet and technology news and culture, and expands its readership through adding new channels and content partnerships. Like TechCrunch it runs events in Europe and North America.

#5 **LifeHacker** – Launched in 2005 Lifehacker bills itself as the place for “Tips, tricks, and downloads for getting things done.” Covering topics as diverse as “From the Tips Box”, anything Windows, Mac, Linux, Android or iOS related, careers, health and wine. Lifehacker also has two international editions – Lifehacker Japan & Lifehacker Australia.

#6 **Wired** – Wired.com (home to WIRED magazine first published in 1993) is part of the massive Conde Nast publishing group – which also owns Reddit (see below) as well as other great blogs such as Ars Technica. Wired’s angle is to look at how ideas and innovation are changing the world. Chris Anderson (writer of The Long Tail, Free) set up his blog GeekDad which was later to become Wired.com. WIRED and Wired.com reach more than 14m readers a month.

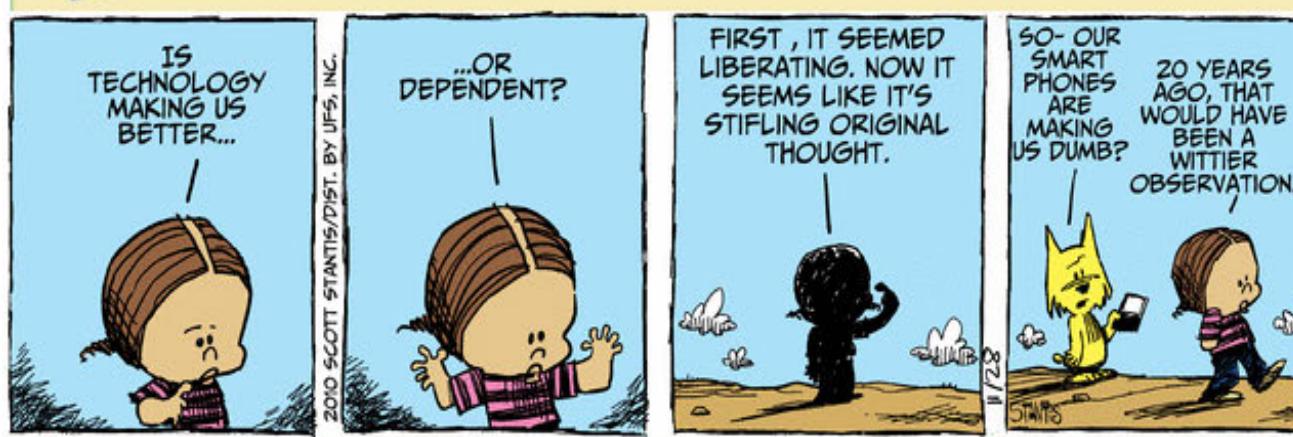
#7 **Reddit** – If you’ve not come across Reddit where have you been? A massive community with a simple bulletin board, users post links or self-posts – other members of the community ranking the post or links up or down. The most popular links making the site’s front page. Users build up kudos through the amount of ups/downs their posts and links get – “karma”. Although part of Conde Nast, feels far from commercial.

#8 **Geek** – One of the oldest blogs on our list bust still going strong. Founded in 1996 Geek.com features cover anything from buying guides and review for mobile, gaming, gadgets and computer hardware and software.

#9 **Forbes** – More of a business and lifestyle website than the rest on the list, but Forbes.com has a vast network of writers meaning that there’s always new content – news and commentary. And given Forbes’ size and reputation often with CEOs and senior execs of many of the major technology companies.

#10 **Hongkiat** – The creation of Hongkiat Lim, since 2007 this website has been giving its readers hints, tips and ideas on technology and design. Based in Malaysia this website has grown considerably and is often mentioned itself by many of the bigger sites – such as Lifehacker and TheNextWeb above.

<https://www.statuscake.com/statuscakes-top-10-technology-blogs-how-do-you-start-your-day/>



### Creative Commons License Types

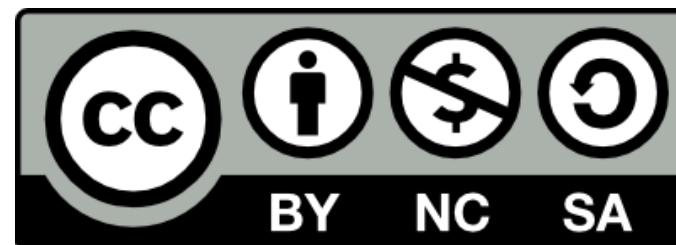
	Can someone use it commercially?	Can someone create new versions of it?
Attribution		
Share Alike		 Yup, AND they must license the new work under a Share Alike license.
No Derivatives		
Non-Commercial		 Yup, AND the new work must be non-commercial, but it can be under any non-commercial license.
Non-Commercial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

This lesson was developed by:

Rob van der Willigen  
CMD, Hogeschool Rotterdam  
NOV 2020



<http://creativecommons.org/licenses/by-nc-sa/3.0/>

This lesson is licensed under a Creative Commons Attribution-Share-Alike license. You can change it, transmit it, show it to other people. Just always give credit to RFvdW.

<http://creativecommons.org/licenses/>

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

