Witekio

EMBEDDING SUCCESS

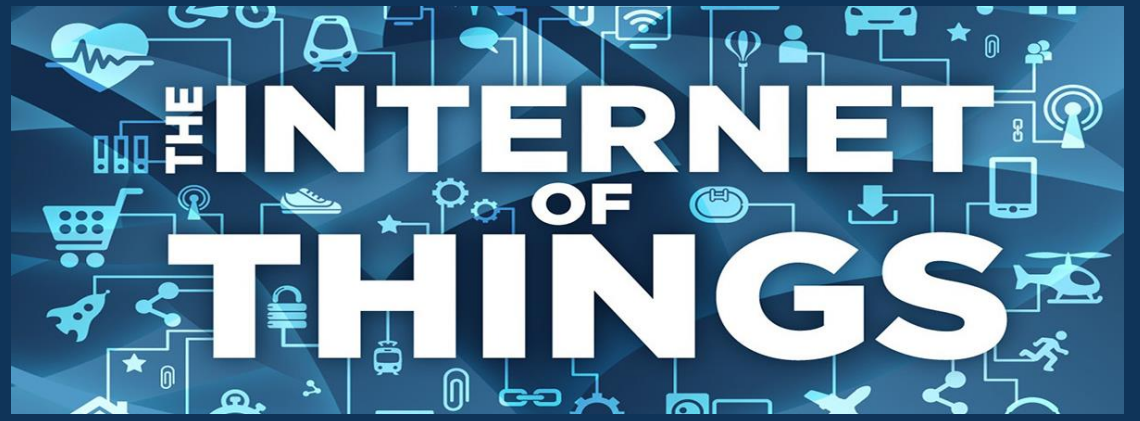
Internet of Things

Internet of Things

Introduction

Witekio

EMBEDDING SUCCESS



Definition

An IoT Device is any internet-connected, non-browser device that isn't a piece of network infrastructure.

Internet of Things

Use case

Witekio

EMBEDDING SUCCESS

Mass Market

Wearable

- SmartWatch
- Heartbeat sensor

House Automation

- Temperature sensors
- Smart Power Grid
- Thermostat (Nest ….)

Connected Car

- Trafic regulation
- Autonomous cars

Smart Cities

- Street Light Management
- Trafic management

….

Industry

Production line

- Sensors on a production line
- Production speed figures
- Product defaults tracking

Sensor devices

- Water meters
- Gas flavor measure in petroleum industry
- Earthquake connected detector

Predictive Maintenance

- Lift default anticipations
- Defibrillator battery level

Goals ...

Connect brand new developed devices, or make old device connectable.

Avoid on-site only data processing

Manage Business Intelligence and provide tools for decisions :

- Alarms (levels, ….)
- Identify faulty devices
- Anticipate maintenance on devices
- Remotely track production lines
- Automate operations

.... And limits

Requires 24/7 connectivity
Battery operated devices are nightmares
Heterogeneous communication protocols
No on the shelves solution

Make money with IoT

Products

- Connected devices development
- Energy harvesting for battery operated devices
- Custom sensors development

Services

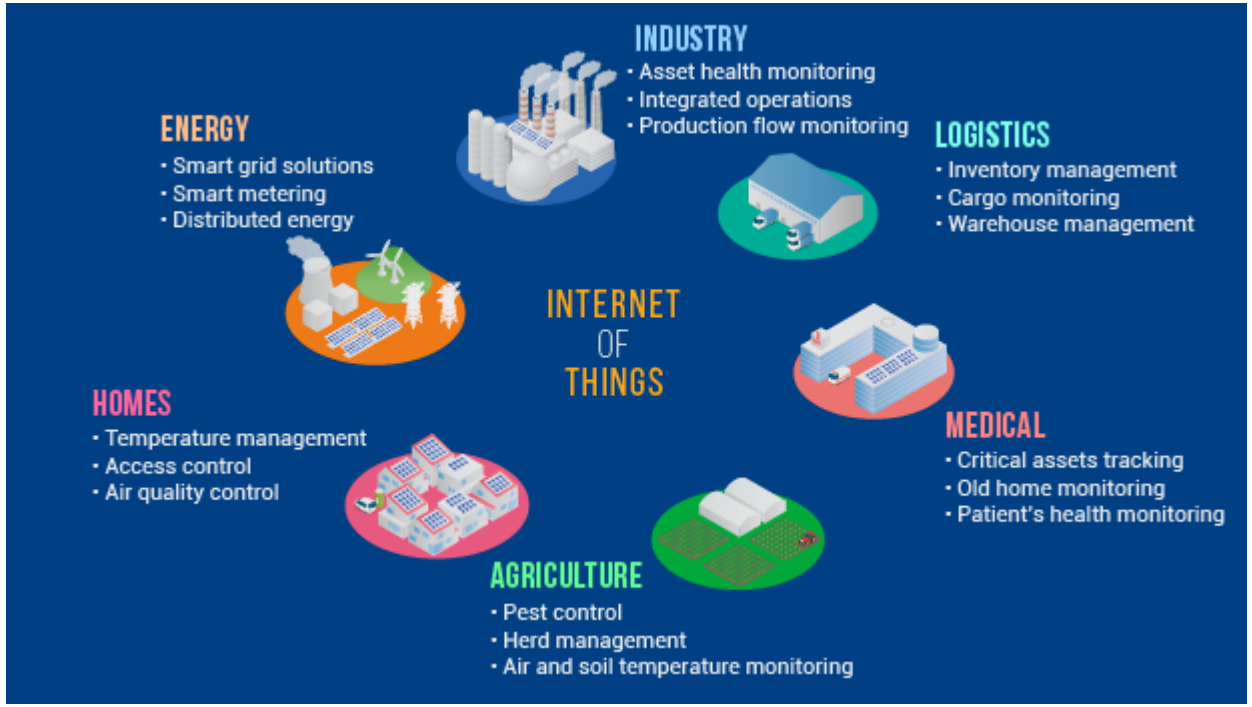
- Cloud service hosting
- IoT solution deployment

Software

- Phone applications (mobility)
- Dashboard applications

Technology

- Data exchange technology
- System communication unifications



Internet of Things

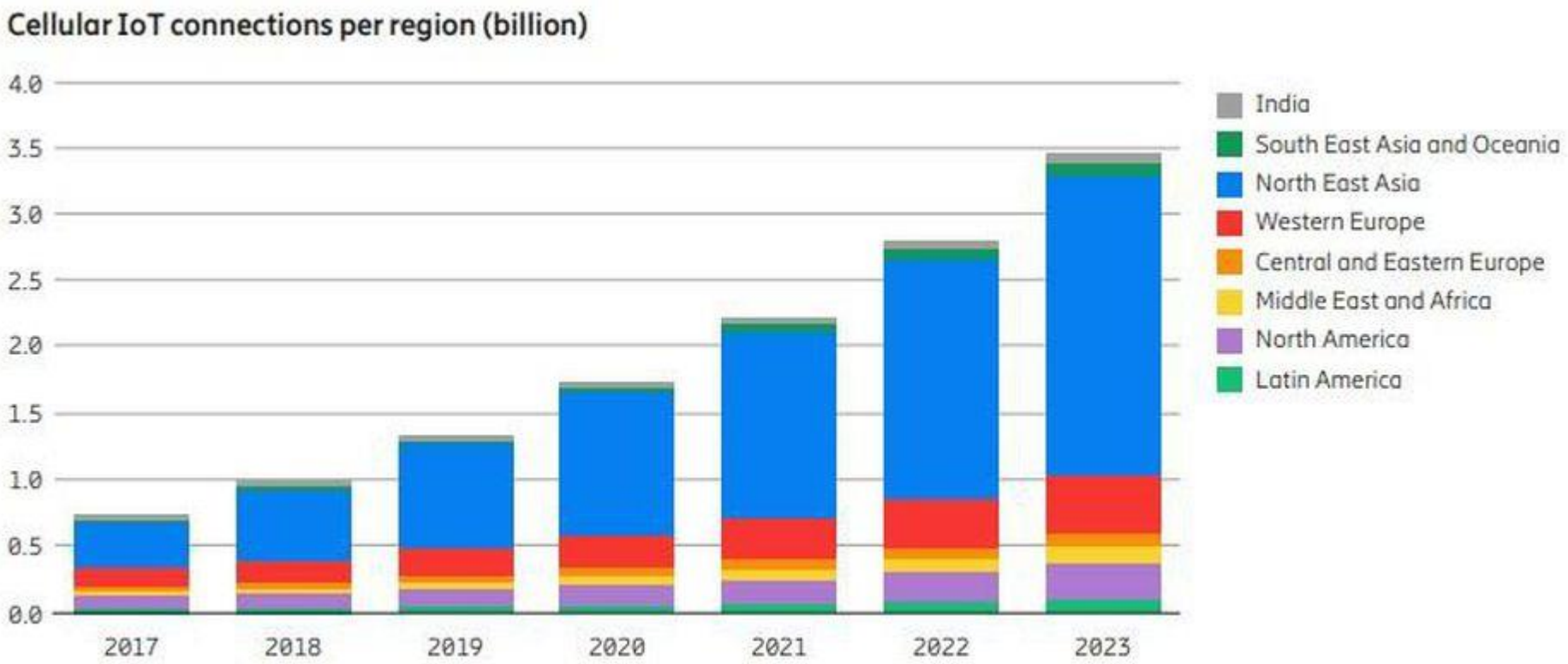
Market

Witekio

EMBEDDING SUCCESS

IoT Trends – Network connections

Ericcson is forecasting the number of cellular IoT connections is expected to reach 3.5B in 2023

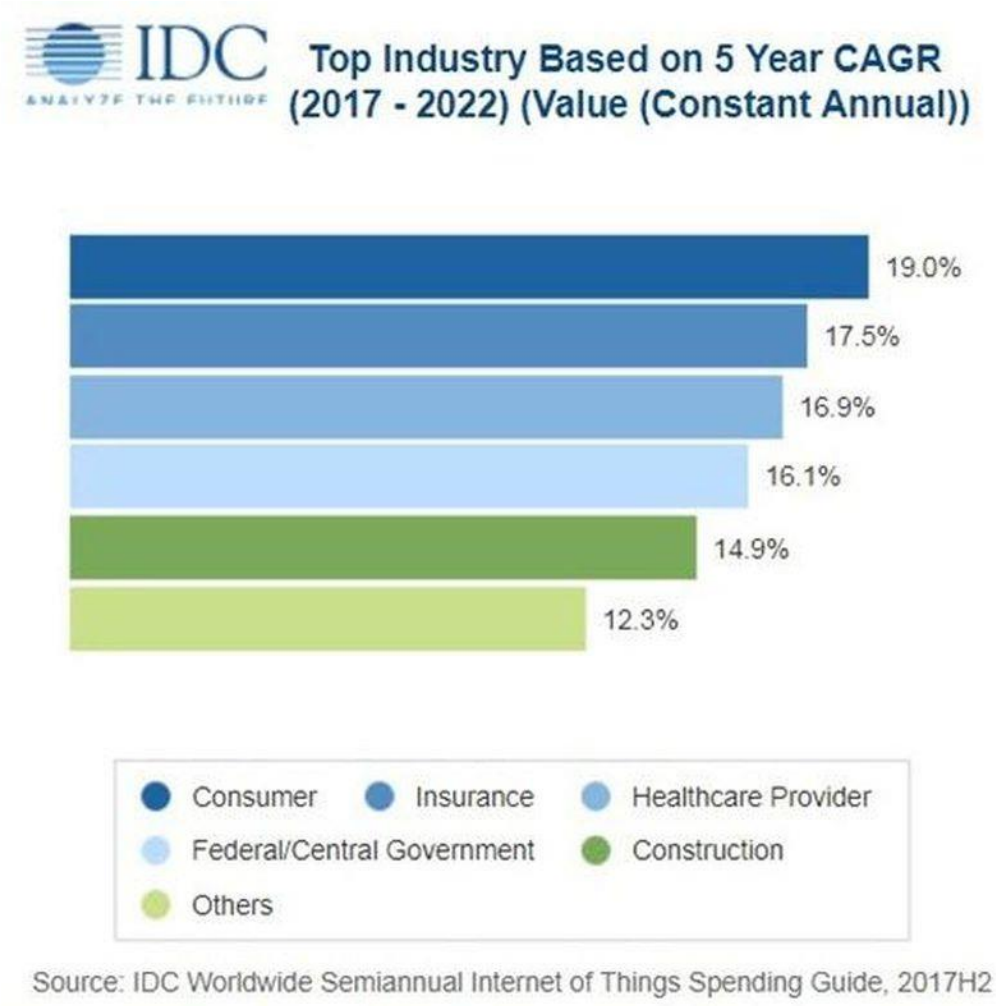


SOURCE: ERICSSON MOBILITY REPORT, JUNE 2018 (PDF, 36 PP., NO OPT-IN).

IoT Trends – Investment

Worldwide technology spending on the Internet of Things to reach $1.2T in 2022

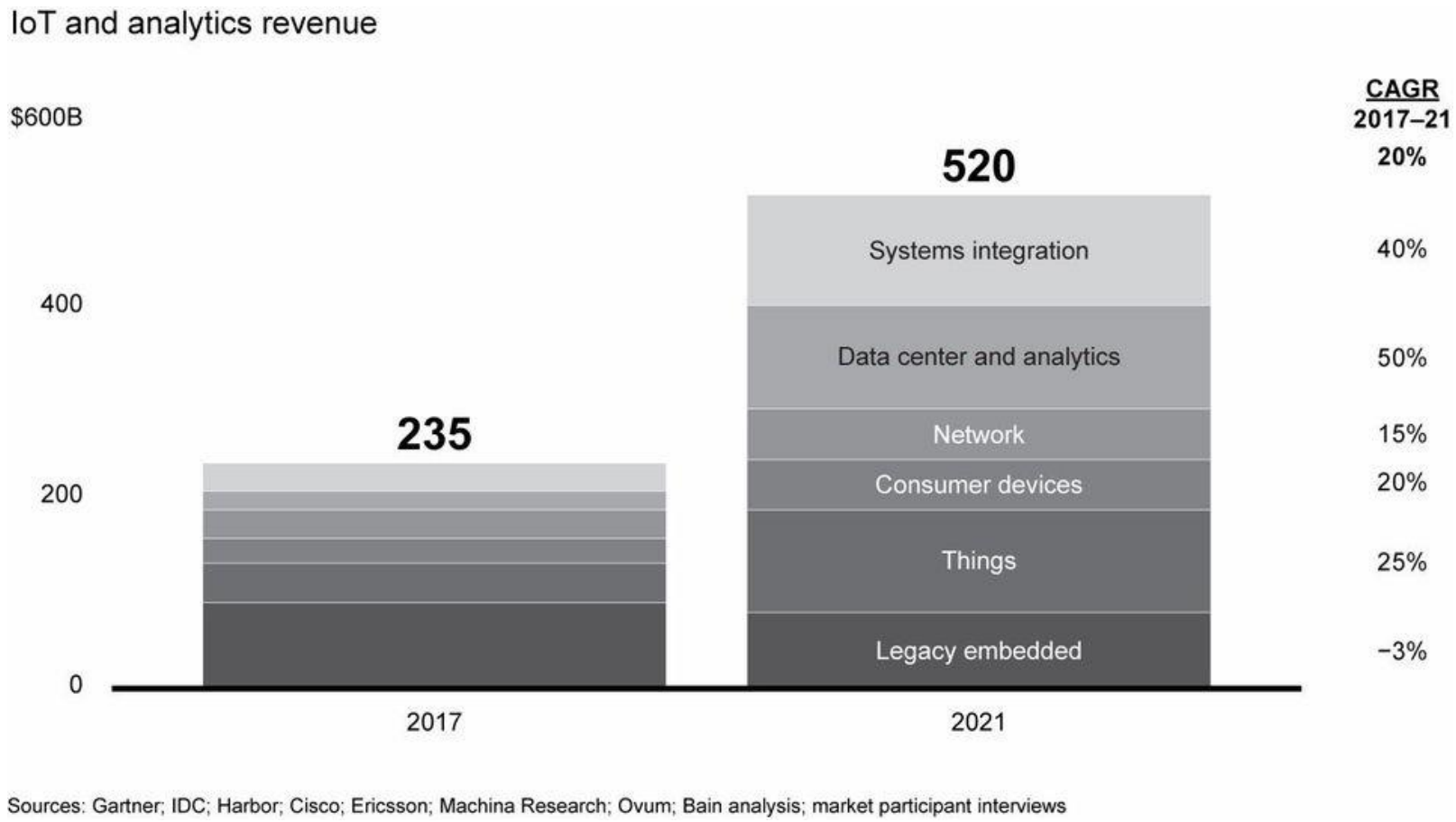
CAGR. Compound annual growth rate



Source: IDC Worldwide Semiannual Internet of Things Spending Guide, 2017H2

IoT Trends – Revenue

Bain predicts the combined markets of the Internet of Things will grow to about $520B in 2021, more than double the $235B spent in 2017

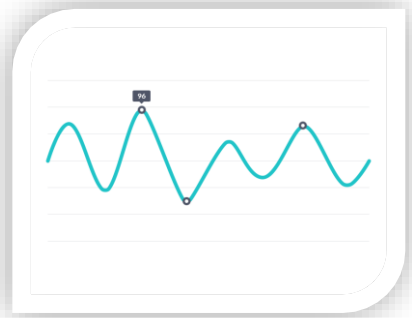


Sources: Gartner; IDC; Harbor; Cisco; Ericsson; Machina Research; Ovum; Bain analysis; market participant interviews

Internet of Things

Architecture

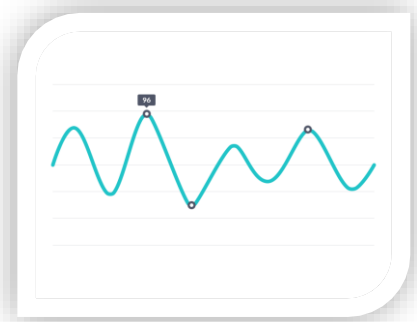
Wearable



Wearable

Phone is connect to the Internet and act as a gateway.
Phone is connected to the wearable device through Bluetooth Low Energy (BLE)
Phone host the business application

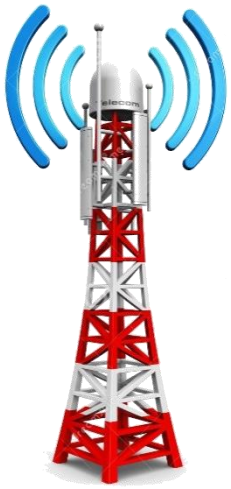
Typical Architecture

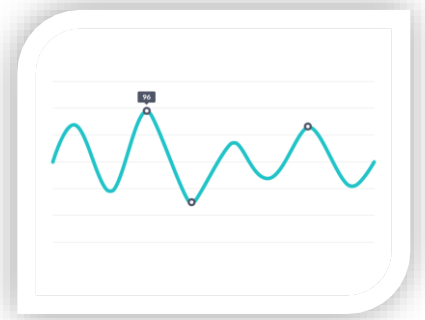


Typical Architecture

Sensors and actuators are connected over Radio to the gateway.
The Gateway is connected to the Internet, and relay data to the Cloud application.

WAN Architecture





Sensors and actuators are connected to a WAN public network (GSM, LoRa, SigFox, other).
The WAN Network provider provides a proprietary solution to aggregate the data.

Hardware parts

Sensor/Actuator with micro processor
Gateway
Server

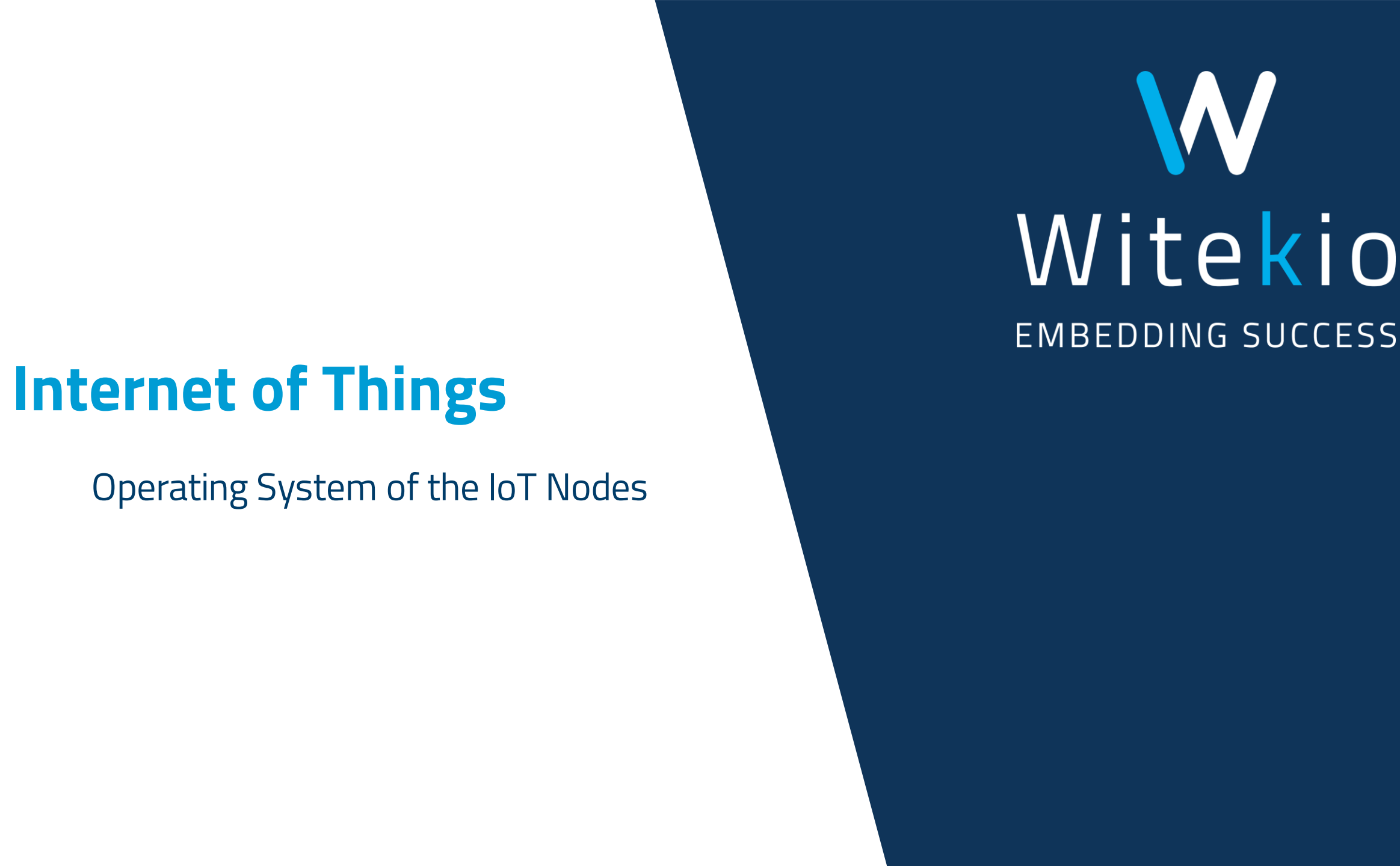
Software parts

Sensors/Actuators
- OS/ OSLess
- Communication stack

Gateway
- Operating system
- Communication stack
- Network stack (IPv4 or IPv6)

Server
- Operating system
- Web Server
- Database

Customer
- Web browser
- Custom application



Internet of Things

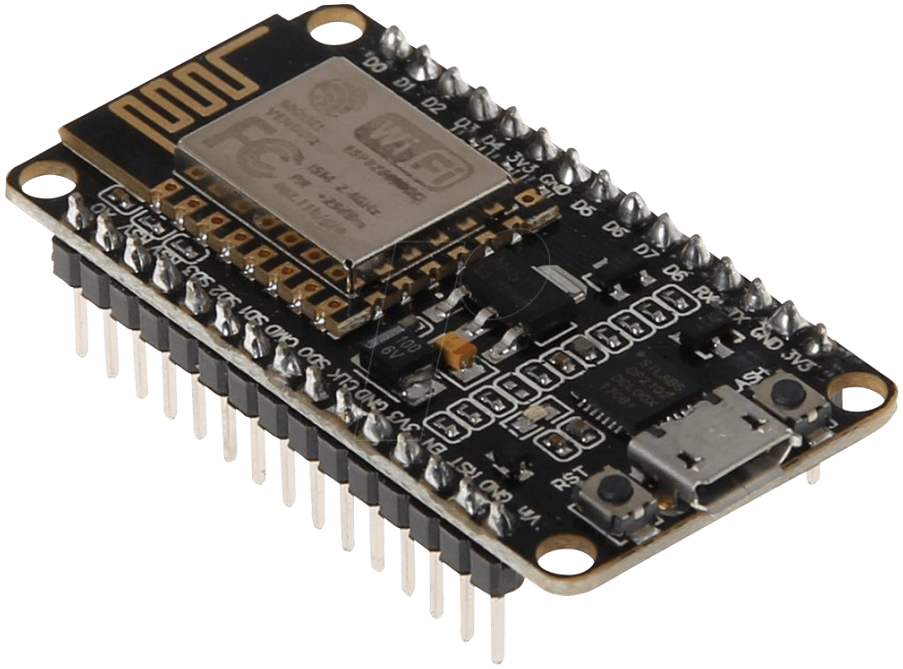
Operating System of the IoT Nodes

Sensor/Actuator

OSLess : bare metal development

OS:

- Linux
- Contiki
- TinyOS
- RIOT
- Mbed
- FreeRTOS
- .Net MicroFramework
- …



Linux

Embedded version of the Linux desktop version
Kernel Image size reduced to target device application purpose
Target ARM and x86 CPUs

Contiki

Contiki is an operating system for networked, memory-constrained systems with a focus on low-power wireless Internet of Things devices. Extant uses for Contiki include systems for street lighting, sound monitoring for smart cities, radiation monitoring, and alarms. It is open-source software released under a BSD license.

TinyOS

TinyOS is an open source, BSD-licensed operating system designed for low-power wireless devices, such as those used in sensor networks, ubiquitous computing, personal area networks, smart buildings, and smart meters. A worldwide community from academia and industry use, develop, and support the operating system as well as its associated tools, averaging 35,000 downloads a year.

RIOT

Program like you are used to. Do not waste time with complex or new environments.

- Standard programming in C or C++
- Standard tools such as gcc, gdb, valgrind
- Minimized hardware dependent code
- Zero learning curve for embedded programming
- Code once, run on 8-bit platforms (e.g. Arduino Mega 2560), 16-bit platforms (e.g. MSP430), and on 32-bit platforms (e.g. ARM)
- Partial POSIX compliance. Towards full POSIX compliance.
- Develop under Linux or Mac OS using the native port, deploy on embedded device

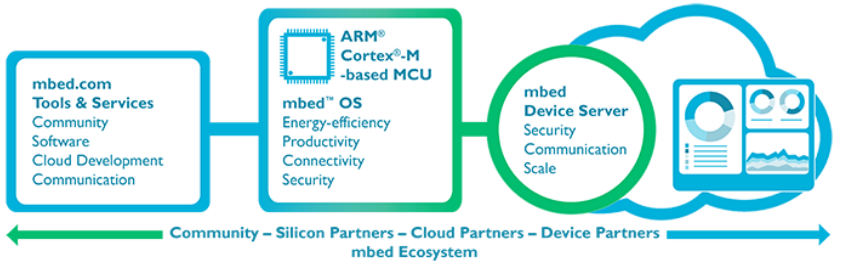
FreeRTOS

Real Time Operating System supporting a large set of MCU's.

- Used in wireless MCU (ESP8266,…)
- Used on Real Time devices but not only

ARM mbed

Developed by ARM and optimized for ARM Cortex based MCU
Online firmware development



.Net Microframework

Originally edited by Microsoft, now Open Source

Reduce set of API of the Microsoft .Net Framework

- C# and VB.Net support
- Multithread support
- Graphical Library

Wear OS by Google

Developped by Google for SmartWatches

Conclusion

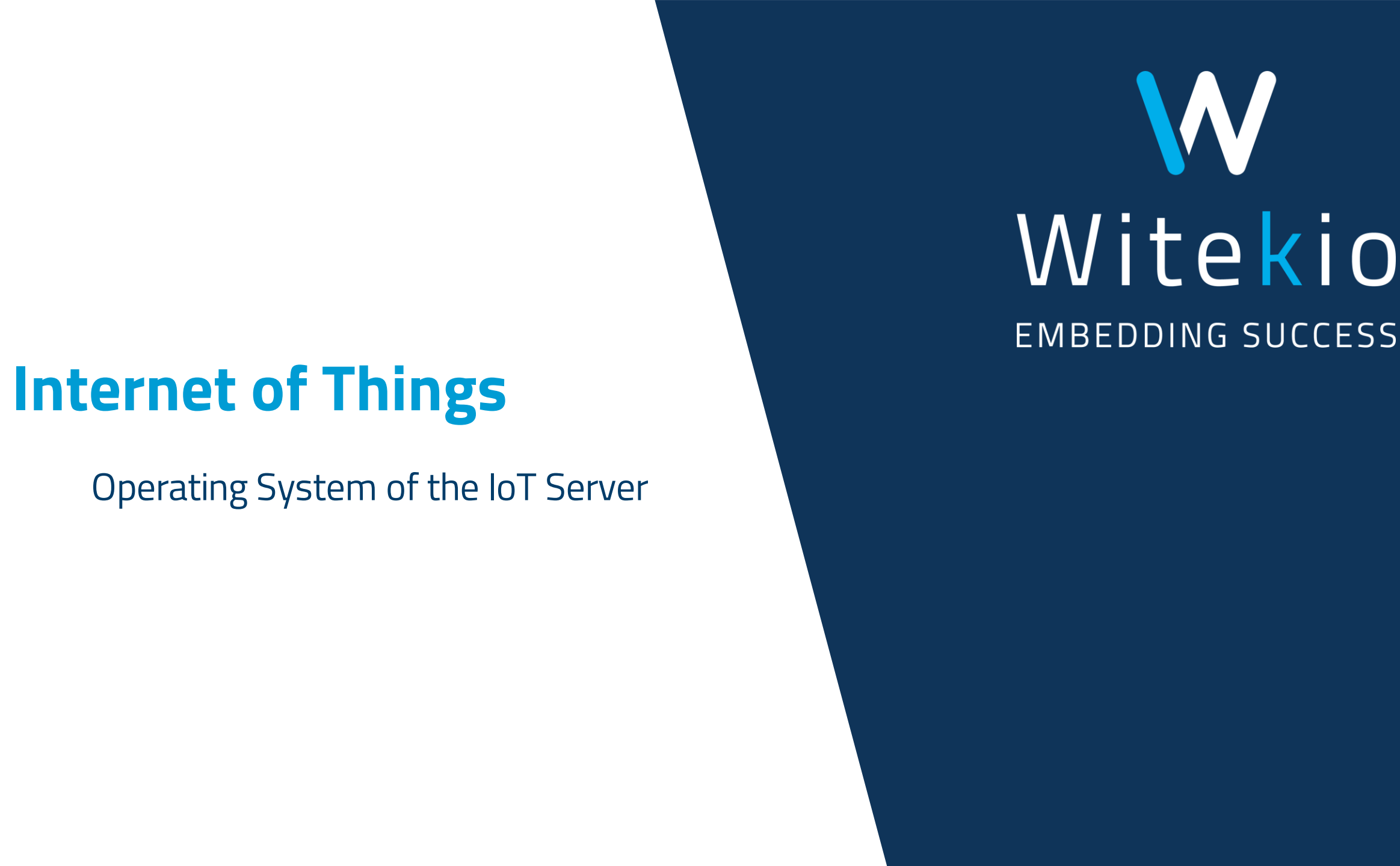
OS	Min RAM	Min ROM	C Support	C++ Support	Multi-Threading	MCU w/o MMU	Modularity	Real-Time
Contiki	< 2kB	< 30kB	●	✗	●	✓	●	●
Tiny OS	< 1kB	< 4KB	✗	✗	●	✓	✗	✗
Linux	~ 1MB	~ 1MB	✓	✓	✓	✗	●	●
RIOT	~ 1.5kB	~ 5kB	✓	✓	✓	✓	✓	✓
.Net MF	100 kB	800kB	✗	✗	✓	✓	✓	✗

Internet of Things

Operating System of the IoT Gateway

Gateway

Mostly based on Linux
Some based on Windows 10 IoT Core



Internet of Things

Operating System of the IoT Server

Server

Web Servers are mainly running Linux

Internet of Things

Radio connection

Network topology

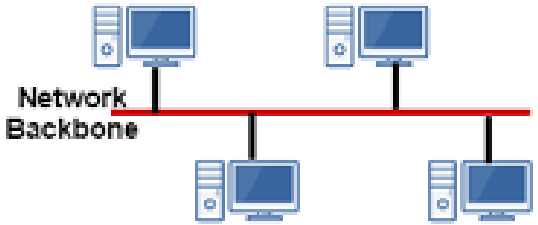
Point-to-point: The simplest topology with a dedicated link between two endpoints. This is the topology for Bluetooth and Bluetooth Smart (BLE).

Star: In local area networks with a star topology, each network host is connected to a central hub with a point-to-point connection. So it can be said that every device is indirectly connected to every other node with the help of the hub.

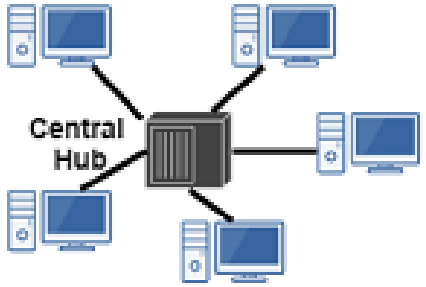
Mesh: mesh network is a topology where a node is as a Roudeclared as coordinator and other nodes ter

Cluster tree: A tree topology is essentially a combination of bus topology and star topology. The nodes of bus topology are replaced with standalone star topology networks. This results in both disadvantages of bus topology and advantages of star topology.

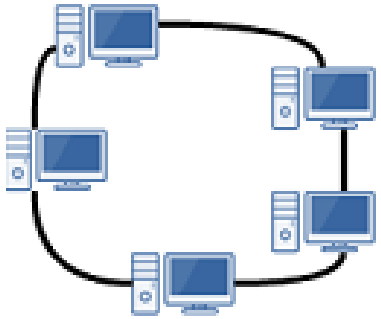
Network topology



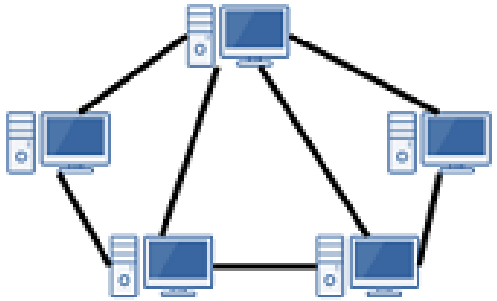
Bus



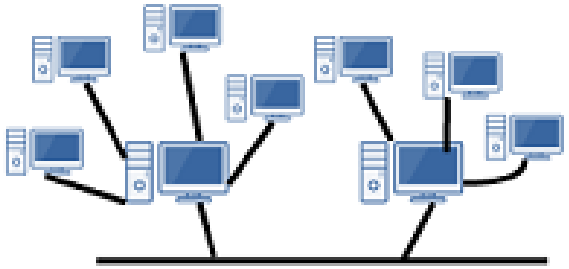
Star



Ring



Mesh



Tree

Point-to-Point

SubGhz

The « Sub-1 GHz » technology covers the wireless communications using a frequency below than 1 GHz (mostly used with 433, 868 et 915 MHz bands) depending on the local wireless regulation rules.
The technology is not constrained to any protocol. Every transmitter and receiver modulate the data and data transmission is linked to manufacturer datasheet.
Advantages of SubGhz compared to higher frequency resides in:

- Better distance coverage (at same transmission power)
- Less expensive (cost, power consumption)
- Less sensitive to disturbances (RF environment)

Bluetooth Low Energy

“Bluetooth low energy (Bluetooth LE, BLE, marketed as Bluetooth Smart) is a wireless personal area network technology designed and marketed by the Bluetooth Special Interest Group aimed at novel applications in the healthcare, fitness, beacons, security, and home entertainment industries. Compared to Classic Bluetooth (V3.0), Bluetooth Smart (4.x) is intended to provide considerably reduced power consumption and cost while maintaining a similar communication range.” - wikipedia

Wifi

Wi-Fi is a set of wireless communications protocols covered by the standards of the IEEE 802.11 group. This protocol has the advantage of offering high given rate, but the disadvantage of high consumption (relative to ZigBee or thread, for example), which makes it not very suitable for battery operated products requiring greater autonomy.

ZigBee

“ZigBee is an IEEE 802.15.4-based specification for a suite of high-level communication protocols used to create personal area networks with small, low-power digital radios.
The technology defined by the ZigBee specification is intended to be simpler and less expensive than other wireless personal area networks (WPANs), such as Bluetooth or Wi-Fi. Applications include wireless light switches, electrical meters with in-home-displays, traffic management systems, and other consumer and industrial equipment that requires short-range low-rate wireless data transfer.” - wikipedia
ZigBee allows creation of « star » ou « mesh » network

Z-Wave

Z-Wave is a wireless communications specification for home automation. It is used by more than 325 manufacturers in the Z-Wave Alliance.
The Z-Wave wireless protocol is optimized to trade with low bandwidth (between 9 and 40 kbit/s) and devices on battery or electrically powered, as opposed to Wi-Fi, for example, which is intended to exchange high speed and on electric appliances only.
Z-Wave operates in the frequency range of sub-gigahertz, which depends on the regions (868 MHz in Europe, 908 MHz in US, and other frequencies following ISM bands or regions). The range is about 50 m. The technology uses the technology of mesh (mesh) to increase the range and reliability.
Z-Wave is designed to be easily integrated in consumer electronics products, including battery-operated devices such as remote controls, smoke detectors and safety sensors.

Thread

Thread is a low power network protocol, based on IPV6 and created especially for the communication of connected objects.
Thread based on the wireless communication protocol IEEE 802.15.4 as the ZigBee. This technology enables the creation of secure networks of type "mesh" (up to 250 devices).


LoRa™
VS
SIGFOX

LoRa

LoRa (Long Range) is a long-range communication dedicated to M2M and developed by the Grenoble firm Cycleo (Semtech Group). This communication ensures high immunity to interference, is very sensitive in reception, is ultra low power and of course is long range. An ideal communication between connected objects, to avoid radio mesh architecture deployment.

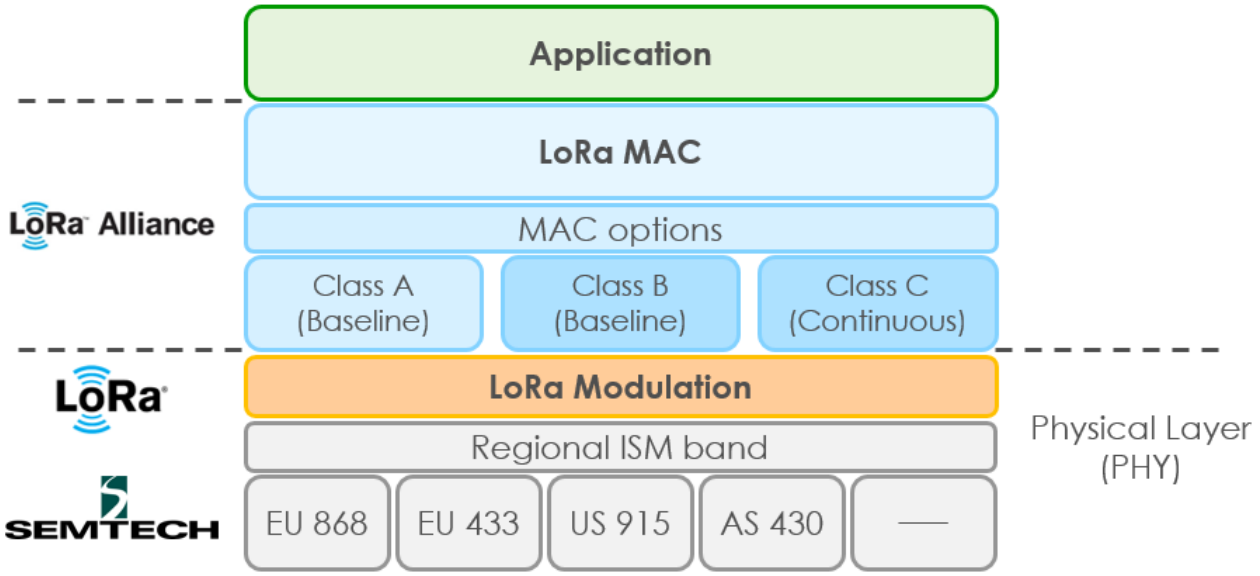
LoRa

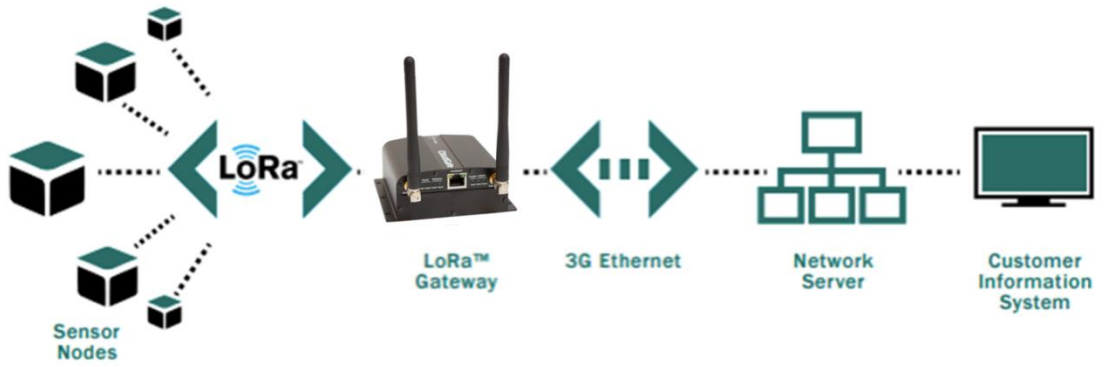
Transmission range

LoRaWAN

LoRaWAN is the protocol that relies on LoRa radio network to ensure the integrity and confidentiality of the transmitted data.

LoRaWAN Application uses network keys to encrypt the data on your private network, and application keys to encrypt data of the specific application.





Sigfox

SIGFOX uses UNB (Ultra Narrow Band) based on a wireless technology for connecting peripherals to its global network. The use of UNB is essential to providing a high-capacity network, scalable, very low energy consumption, while maintaining a simple and easy cellular infrastructure deployment as a star.

The network operates in the ISM bands available globally (unlicensed frequency bands) and coexist on these frequencies with other radio technologies, but without any risk of collision or capacity problems. SIGFOX currently using the European ISL most popular band of 868 MHz (as defined by ETSI and CEPT) and the 902 MHz in the US (as defined by the FCC), depending on specific regional regulations.

Sigfox

Objects

Uplink payload: 0 to 12 bytes
Downlink payload: 0 to 8 bytes

0 to 140 messages per day per device

Sigfox Stations

Over 1 million messages can be processed per base station per day

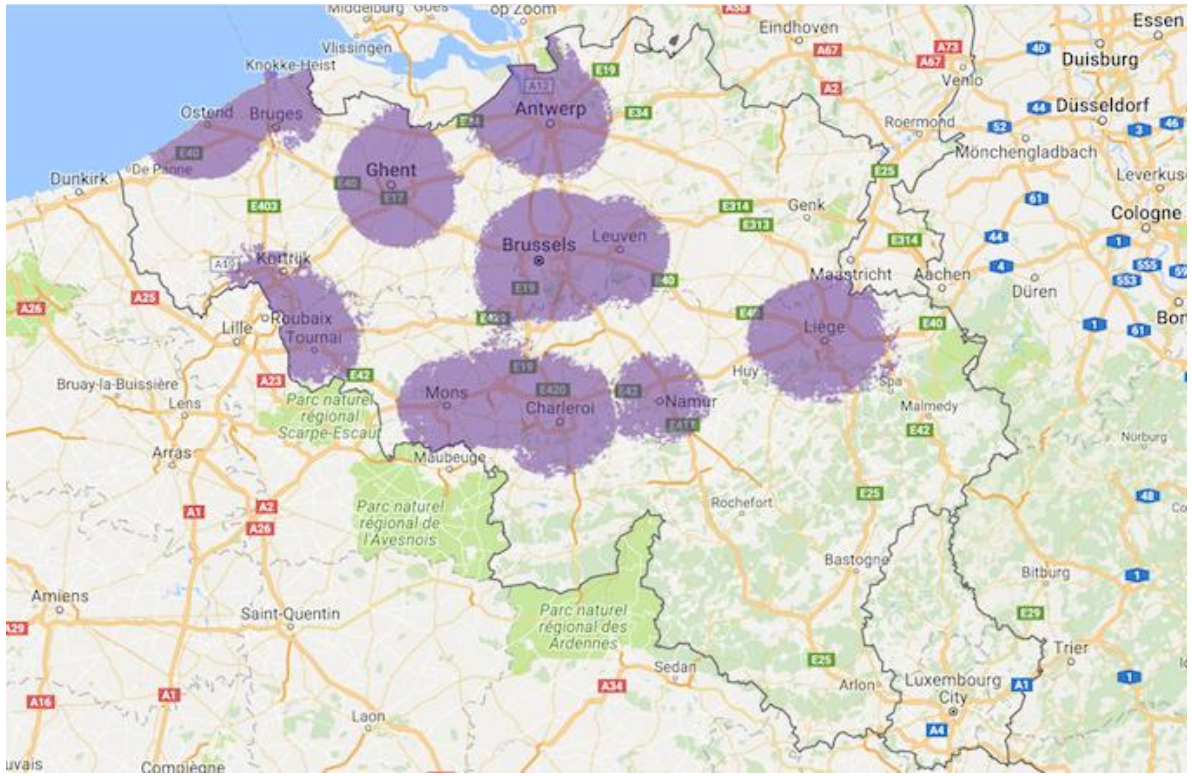
Sigfox Cloud™

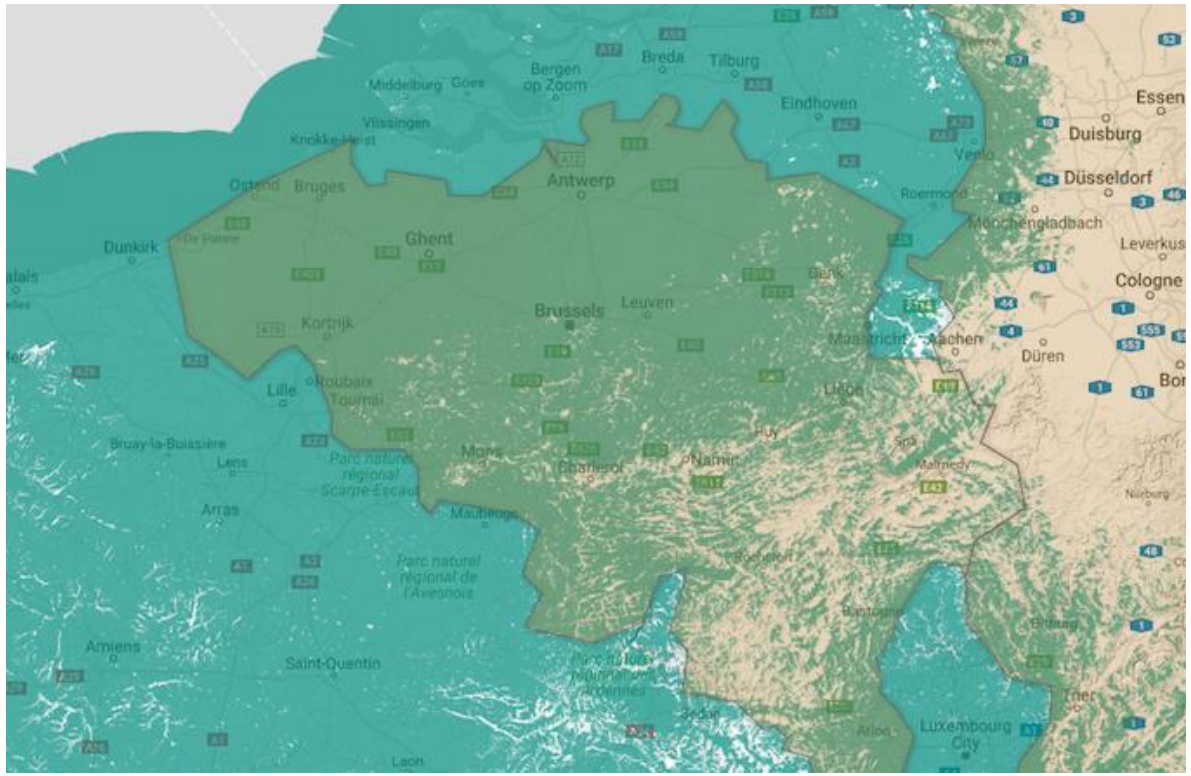
Web interface
API
Callbacks

Customer IT

LoRa vs Sigfox

Network Coverage





LoRa vs Sigfox

Network Coverage

LoRa Coverage is :

- Ensured by private operators
- Everybody can become an operator

Sigfox coverage is:

- Ensured by Sigfox only
- Sigfox is providing the infrastructure

Recap

Technology	Distance	Gateway	Subscription
SubGhz	**5 cm à 1 km**	√	
Bluetooth LE	**5 cm à 30 m**	√	
Wifi	**50 cm à 100 m**	Option	
ZigBee	**5 cm à 30 m**	√	
Zwave	**5 cm à 30 m**	√	
LoRa	**30 m à 15 km**		√
Sigfox	**30 m à 15 km**		√

Witekio

EMBEDDING SUCCESS

Internet of Things

Gateway Introduction

Gateway Introduction

Gateway between smart objects and the cloud

Connectivity: IP, Bluetooth, Wifi, Sub-1 GHz, ZigBee…

Bidirectional communication

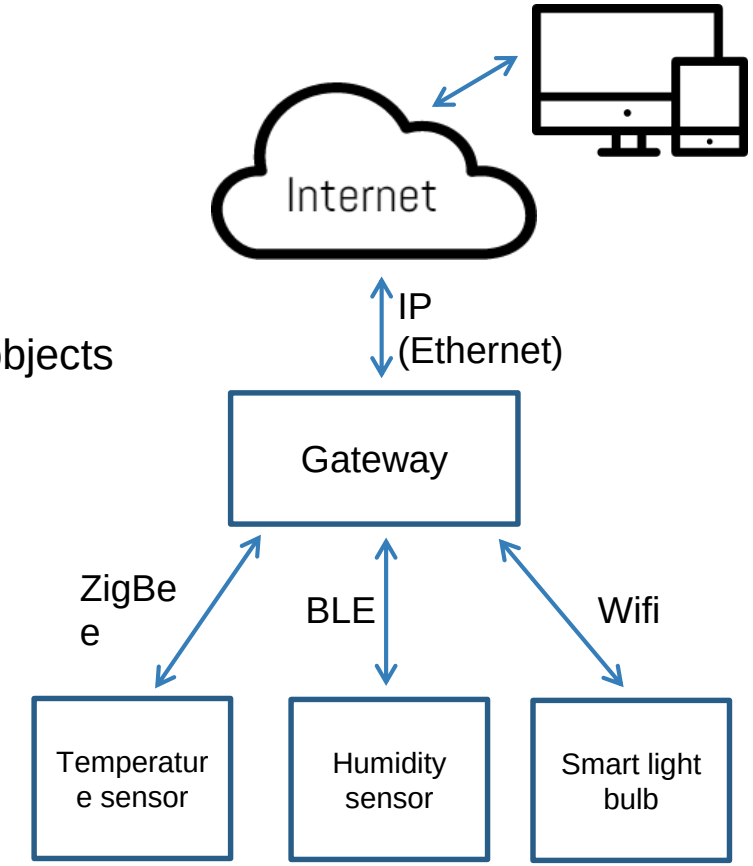
- Retrieve data from objects and forward them to the Cloud
- Receive requests from the Cloud and forward them to the objects

Smart

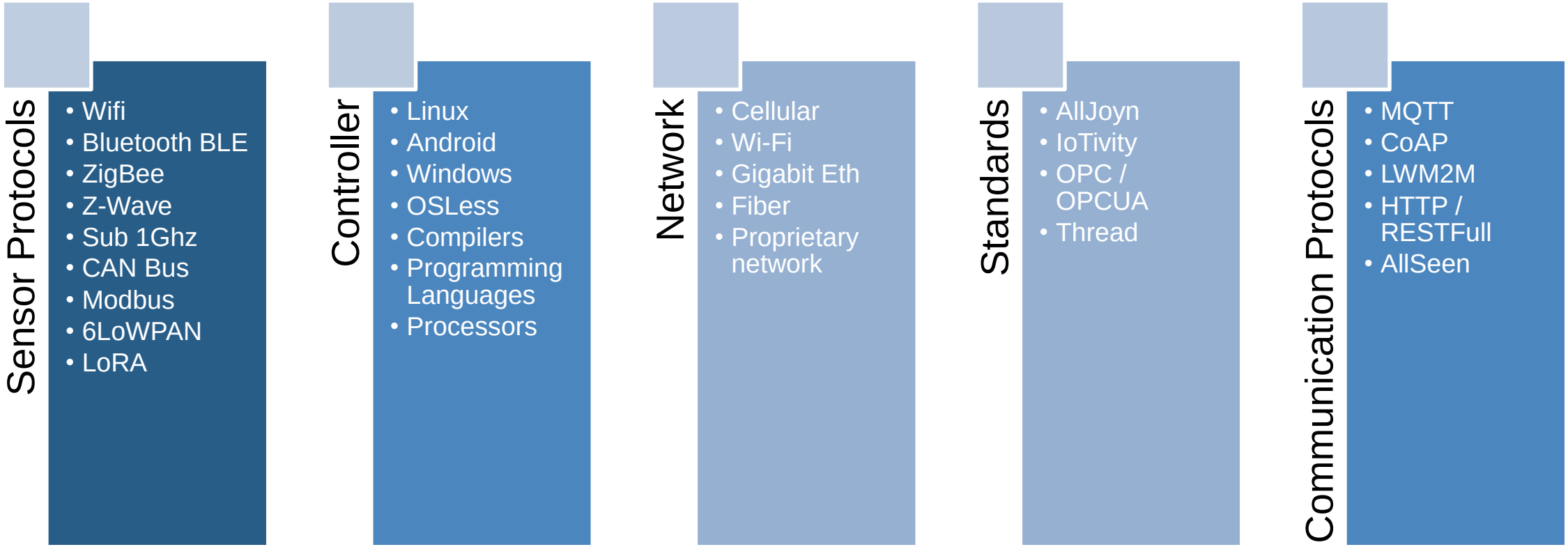
- Process data locally
- Analyze and normalize data
- Local storage

Secure

- User data protection
- Secures access to the different sensors and actuators



A Segmented market



Witekio

EMBEDDING SUCCESS

Internet of Things

Gateway Solutions

Possible Gateway Solutions

« Turnkey » solutions

- There are Hardware and Software turnkey solutions available on the market.
- For instance some manufacturers offer home automation solutions which contain a dedicated gateway

Pros

- Fast and easy installation
- User-friendly interface

Cons

- General use products (not dedicated for industrial applications)
- Not much modular
- Supports a limited number of protocols to communicate with the objects (mainly Z-Wave)
- Cloud solution from the vendor

Hardware Solutions

Here are some possible hardware that can host an IoT Gateway

- Smartphone
- PC
- Custom device
- Dedicated platform

Smartphone

Smartphone

Example: Smart watch connected to an Android smartphone

- The watch sends the following data to the smartphone by using BLE protocol
  - Number of steps made by the wearer (pedometer)
  - Cardiac frequency
- The smartphone sends data to the Cloud by using Wifi or 3G

Pros

- Low cost
- Connectivity (Wifi, Bluetooth Low Energy, USB…)
- Many users already own a smartphone

Cons

- Many different hardware and software
- Powered on battery (limited battery life)
- Gateway software (application or service) must coexist with other software running on the smartphone
- Not compatible with some critical use cases (ex: medical applications)

PC/Embedded Computer

Example

- Kontron, Advantech, Torradex, … embedded computer

Pros

- IP connectivity
- Many different OS available (x86 and ARM)
- Modular

Cons

- Bulky
- High power consumption

On the Shelf Platforms

Existing boards that are suitable for IoT applications are already available on the market

Pros

- Many boards available. Ex:
  - Raspberry Pi, DragonBoard, BeagleBone …
- Many different OS
  - Linux, Windows 10 IoT Core, Android, …
- Turnkey software solutions available for those boards
- Low power consumption
- Expandable connectivity (via USB, SPI, I²C, …)
- Can be used for prototyping a custom solution
- Time to market reduced

On the Shelf Platforms

Cons

- Need to pay attention to the board characteristics to make sure it’s suitable
  - Performances, battery life, storage…
  - Connectivity
  - OS compatibility

Custom device

Pros

- Design oriented for gateway application
- Many microcontrollers dedicated for IoT:
  - TI AM335x, NXP iMX6x, NXP LS1024A, ...
- Can choose the OS
  - Linux, Android, Windows 10 IoT, ...
- Can be tailored to the needs

Cons

- Complex hardware and software development
- High development cost
- Time to Market increased

IoT Gateway Market

Postscapes

IoT GATEWAY MARKET

Software / Edge Analytics

kura
AGILE
ubiworx
FOGHORN
BITSTEW SYSTEMS
ProSyst Bosch Group
BOSCH
ParStream
CISCO
WIND
intel

Hardware Vendors

ADVANTECH
kontron
FlexaWare
DELL
ADLINK TECHNOLOGY INC.
NXP
AAEON an ASUS company
OPTION
libelium
MultiTech Systems
CISCO
EUROTECH Imagine. Build. Succeed.
NEXCOM
Hewlett Packard Enterprise
HUAWEI
LANTRONIX CONNECT SMART. DO MORE.

End-to-End Providers

EUROTECH Imagine. Build. Succeed.
SIERRA WIRELESS
samsara
B-Scada
ADVANTECH B+B SmartWorx

Witekio

EMBEDDING SUCCESS

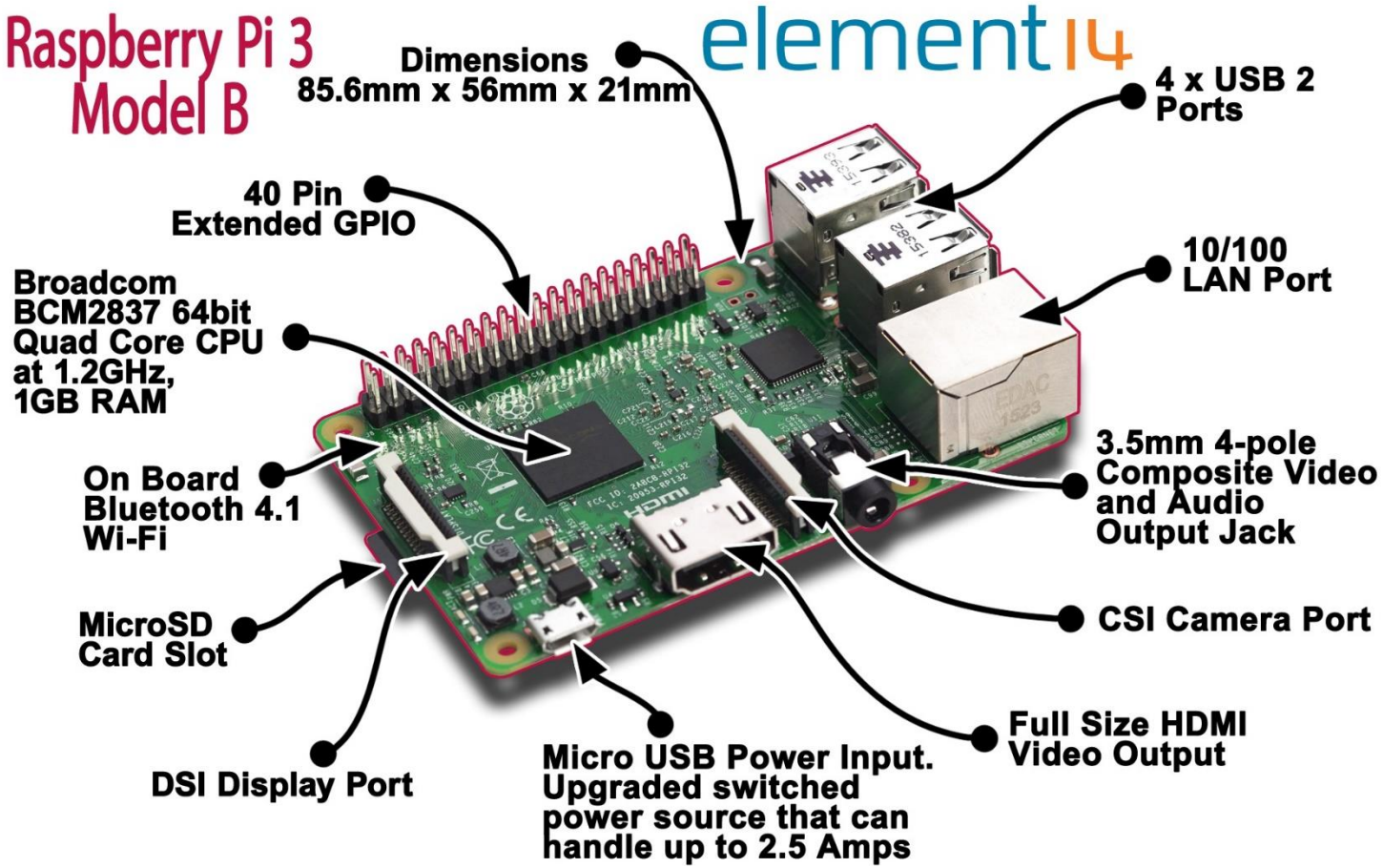
Internet of Things

Gateway Solutions

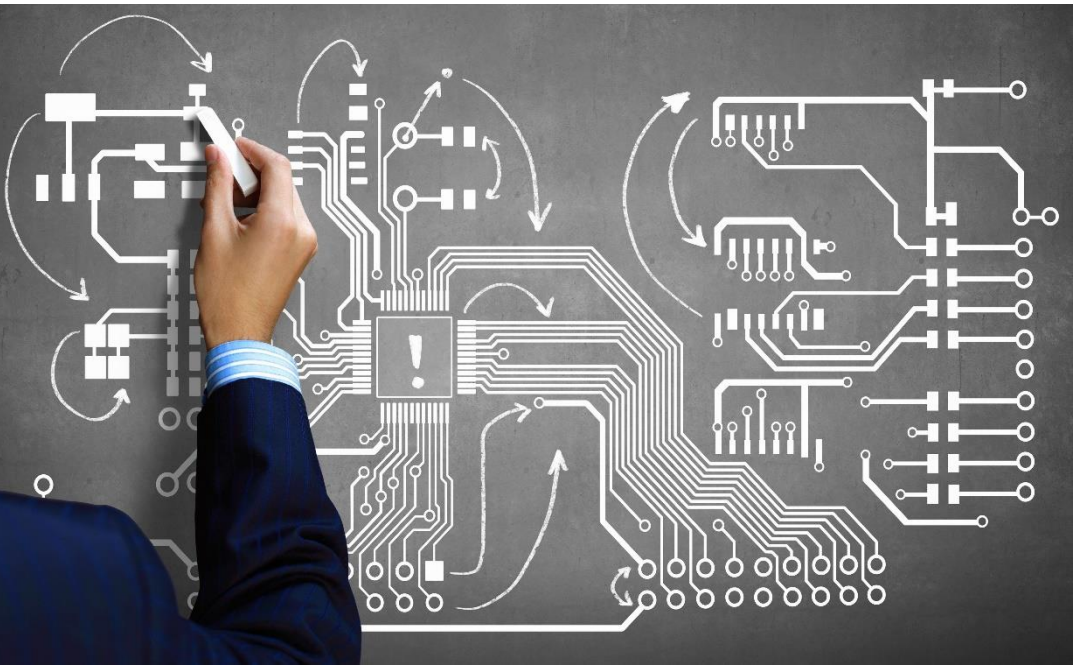
Raspberry Pi 3 Board

Chosen Hardware solution: Raspberry Pi 3

- Broadcom BCM2837
- 1 GB DDR3
- Connectivity
  - Micro SD card slot
  - 10/100BT Ethernet port
  - Wifi
  - Bluetooth 4.1
  - 4x high–speed USB Host port
  - HDMI connector



Raspberry Pi 3 Board



Technical characteristics of the Raspberry Pi make this board a good choice for developing an IoT Gateway.

- ARM processor offering sufficient computing power for gateway application
- IP connectivity: 100Mbps Ethernet and Wifi
  - For communication with the Cloud
  - For remote administration
- Connectivity extensible using USB Host port
  - Ex: ZigBee, Zwave, …
- Small size
- Several OS available for this board
  - Linux
  - Windows 10 IoT Core
- Ultra Low Cost

But industrials are looking for long term support.

- Customisation services available
- Limited set of customisations

Internet of Things

Gateway Software

Software Solutions

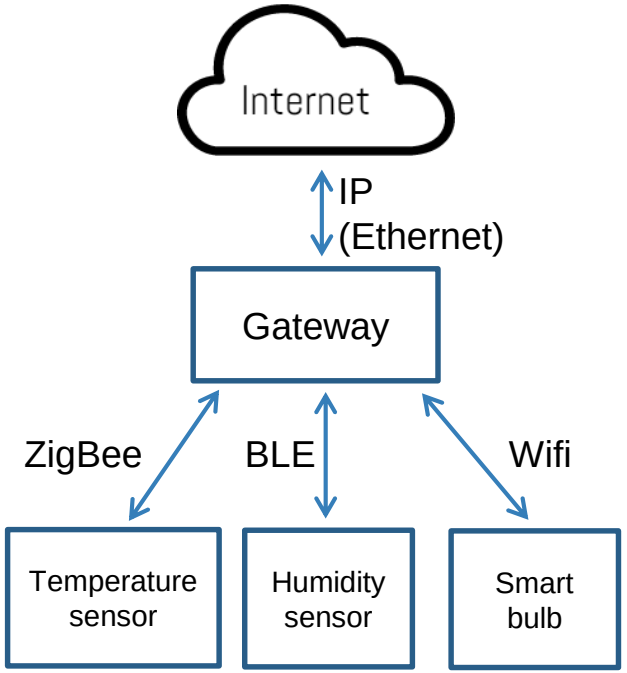
The gateway software commonly performs the following tasks

- Configure the sensors and the actuator that are connected to the gateway
- Retrieve data coming from the sensors
  - Ex: Temperature, humidity…
- Send commands to the actuators
  - Ex: Turn a light On, Turn air cooling system On, …
- Analyze, format and transmit data to the Cloud
- Receive and process requests sent by the Cloud
- Provide an interface for Gateway configuration
  - Config file
  - Web interface
  - …

Software components

Main components of an IoT Gateway

- IP Connection for communication with the Cloud
  - Ethernet, Wifi, 3G, GPRS…
- Interface allowing to configure and monitor the Gateway
- Software stacks allowing to communicate with the devices over several protocols
  - Wifi, BLE, Thread, Z-Wave…
- Communication protocol for unifying messages routed through the Gateway
  - MQTT, AMQT



Witekio

EMBEDDING SUCCESS

Internet of Things

Security

Gateway security

The Gateway is a central component of an IoT system. It important to secure the access to the gateway to protect the data.

- Data exchanged between the gateway and the Cloud must by encrypted so they can’t be intercepted by a 3rd party
- An IoT system can potentially contain some components whose access needs to be protected, for instance a connected door latch. In this case, the protocol used for communicating with the devices must be secured.
- If the gateway provides an interface for remote administration, it needs to be protected to grant access only to authenticated users.

Custom Software Solutions

Possible to develop a custom software solution

- From scratch, or by using existing components

Pros

- Choice among various OS
  - Windows 10 IoT Core, Windows Embedded Compact, Linux…
- Choice of programming language
  - C, C++, C#, Java, Python, JavaScript…
- Software development oriented according to the final needs

Cons

- Increased development time
- Complexity
- Reliability, robustness
- Security

Witekio

EMBEDDING SUCCESS

Internet of Things

Communication Protocols

IoT Protocols

Here are some protocols used by IoT applications

- HTTP
- MQTT
- AMQP
- CoAP
- XMPP
- STOMP

HTTP

HTTP: Hypertext Transfer Protocol

Client-server protocol

Uses TCP protocol

Functions as Request-Response protocol

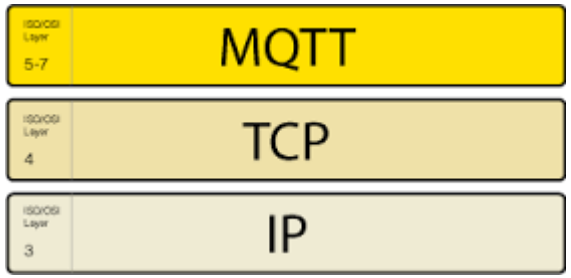
Not dedicated to IoT

- No QoS (Quality of Service)
- HTTP headers causes traffic overhead
- ASCII only communication, no binary
- If the TCP connection is broken, the message are not delivered

MQTT

MQTT : Message Queue Telemetry Transport
Open protocol developed by IBM and Eurotech
Light and small memory footprint
Reliable with 3 QoS level
Simple

- TCP based
- Asynchronous
- Publish/Subscribe architecture
- Not verbose
- Payload can be anything



MQTT Topology

Each client connects to the broker (server) MQTT with a unique ID

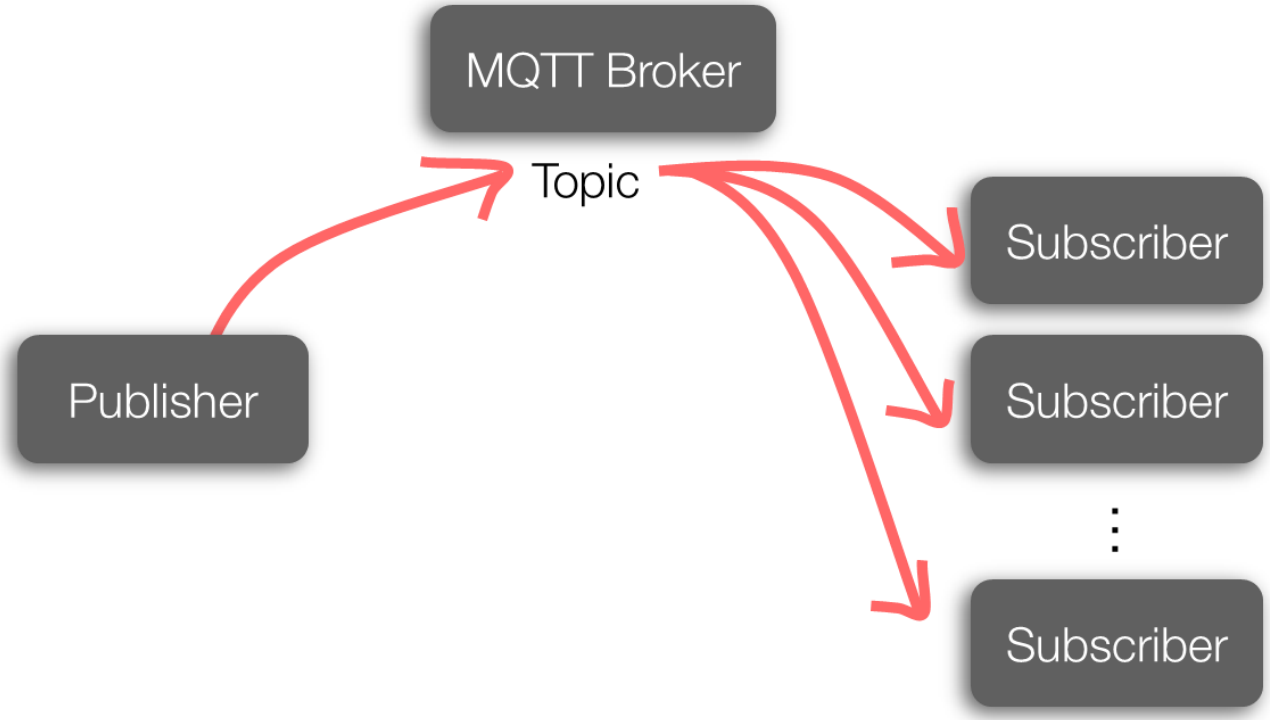
The broker manages the connection between all the clients and transfer the messages between them

Messages retention

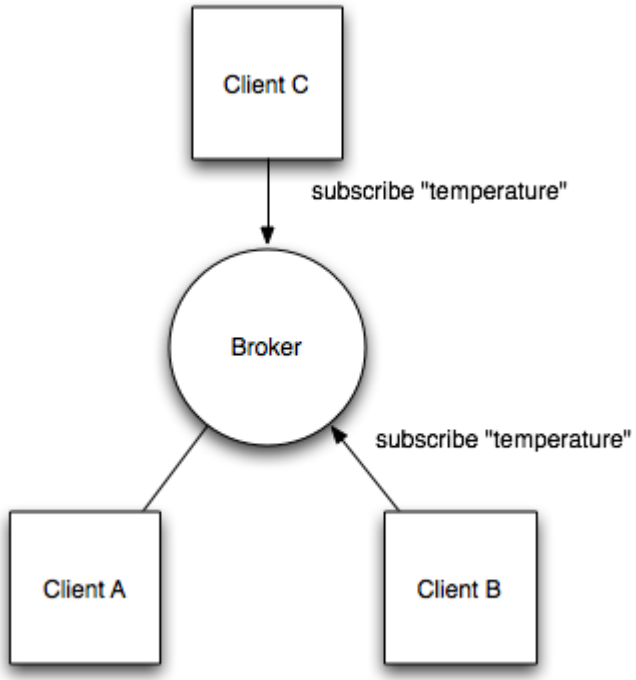
- If a subscribing client is not connected when a message has to be sent, the broker saves the message (typically in a data base)
- Once the clients are online again, the messages are delivered to them
- Useful when network connection are not reliable

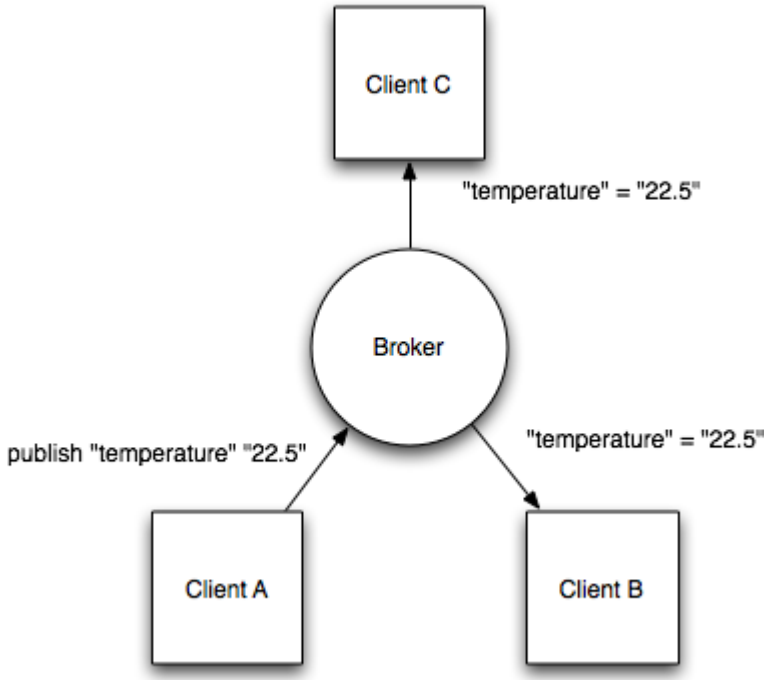
MQTT Topolgy

MQTT Architecure



MQTT Publish / Subscribe





MQTT Publish Subscribe

Publish / Subscribe protocol

- A client can be a publisher, a subscriber or both

« topic » concept

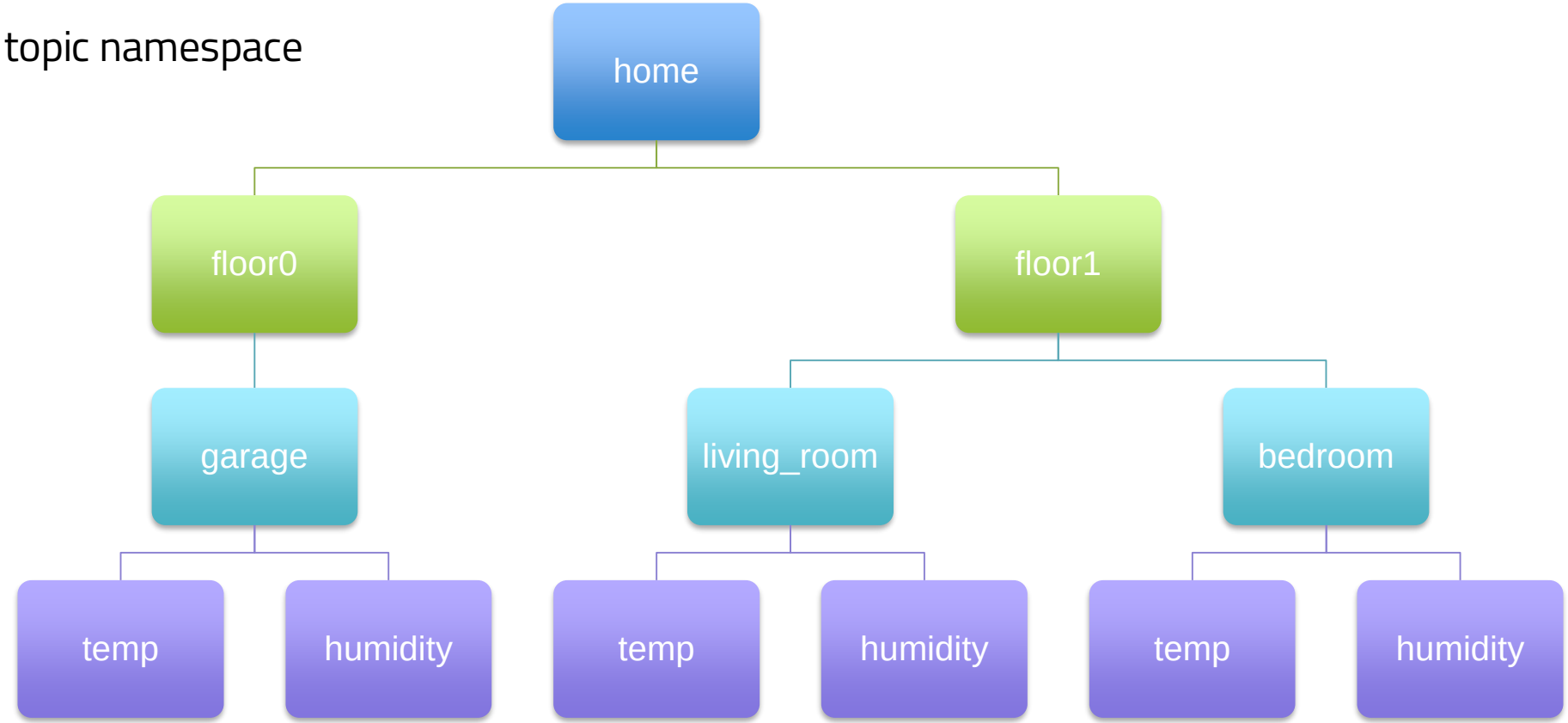
- Publishers publish messages over topics
- Subscribers subscribe to topics
- When a client publish a message on a topic, this message is delivered to all the other clients who subscribed to this topic
- Allow one-to-one and one-to-many communication

Topic subscription can be durable or non-durable

- Durable: If the subscriber is not connecter, the broker saves the message and delivers it once the client is connected again
- Non durable: The message is delivered only if the subscriber is connected, otherwise the message is lost

MQTT Topics

Hierarchical topic namespace



MQTT Topics

A topic is a UTF-8 string, which is used by the broker to filter messages for each connected client

Hierarchical namespace

- Each topic level is separated by a forward slash (topic level separator).

Example of topics for a home automation system

- home/floor1/living_room/temperature
- home/floor0/garage/humidity

A publisher can only publish on an « absolute » topic

A subcriber can subscribe to a topic using absolute path or with wildcards

- A '+' character represents a single level of the hierarchy and is used between delimiters.
  - Ex: home/floor1/+/temperature
- A '#' character represents a complete sub-tree of the hierarchy and thus must be the last character in a subscription topic string
  - Ex: home/floor1/#

MQTT Quality of Service

MQTT is based on TCP/IP

- Messages can be lost if the TCP connection is broken

MQTT protocol defines 3 levels of QoS

- QoS Level 0 (At most once)
  - MQTT only depends of the quality of the TCP connection
  - Messages are not guaranteed to reach their destinations
- QoS Level 1 (At least once)
  - Message is guaranteed to be delivered at least once to the receiver
  - A client can receive the same message more than once
- QoS Level 2 (Exaclty once)
  - Each message is guaranteed to be received once and only once by the subscriber

The more the QoS level is high, the more ressources and bandwidth will be used.

MQTT Authentication and Security

There is no security layer implemented in the MQTT protocol itself

Possibility to use a username and a password to allow a client to connect to the network

MQTT is based on TCP

- Possibility to encrypt data by using SSL/TLS
- Use certificates for authentication

Possibility to encrypt the messages directly at the application level

MQTT Last Will

Last Will allows to notify the other clients when client lose the network connection

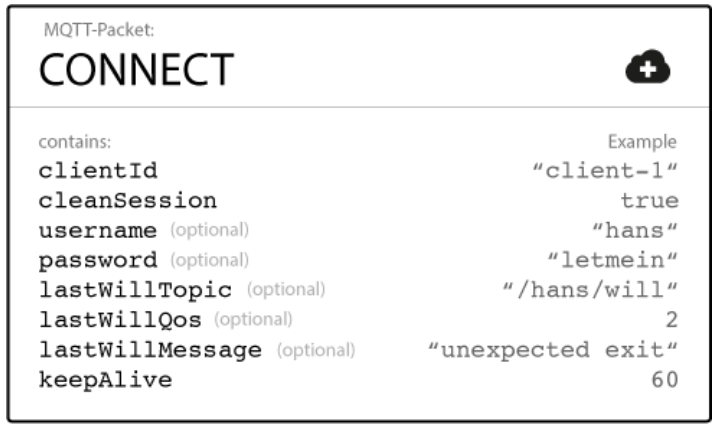
Each client can specify its own Last Will message

- Classic MQTT message (payload, QoS…)

The last will message is sent in the following situations

- The server has detected a problem on the network
- The client is not answering anymore after a certain amount of time (keepAlive)
- The client closes the connection without sending the DISCONNECT packet
- The server close the connection because of an error

- The last will message characteristics are passed to the broker in the connection message



MQTT Brokers

List of MQTT brokers (non exhaustive)

- Mosquitto
- HiveMQ
  - Professional solution, Java
- GnatMQ
  - Open Source source project, .Net
- RabbitMQ
- Moquette
  - Open source solution, Java
- Apollo
  - Plugin Apache

Mosquitto (MQTT Broker)

Open Source Project

- Eclipse Public License
- Written in C

Support many OS

- Windows
- Mac
- Linux (Arch Linux, Debian, Fedora…)
- FreeBSD
- OpenWRT
- QNX

MQTT Clients

There are several MQTT clients available for different OS

Here are some graphical MQTT clients

- MQTT.fx
  - Windows, MacOSX, Linux
- mqtt-spy
  - Based on Java 8.
- MQTT Inspector
  - iOS only
- MyMQTT
  - Android
- HiveMQ Websocket Client
  - Web Service
- MQTT Lens
  - Google Chrome application

MQTT Clients: Paho

Paho is an Open Source project

Provides a set of tools and librairies for creating MQTT clients

Available for many languages

- C/C++
  - C Posix and Windows
  - C++ Posix and Windows
  - C/C++ for embedded systems
- Java
  - J2SE
  - Android
- JavaScript
- Python
- Go
- C# .Net and WinRT

MQTT Clients: Paho

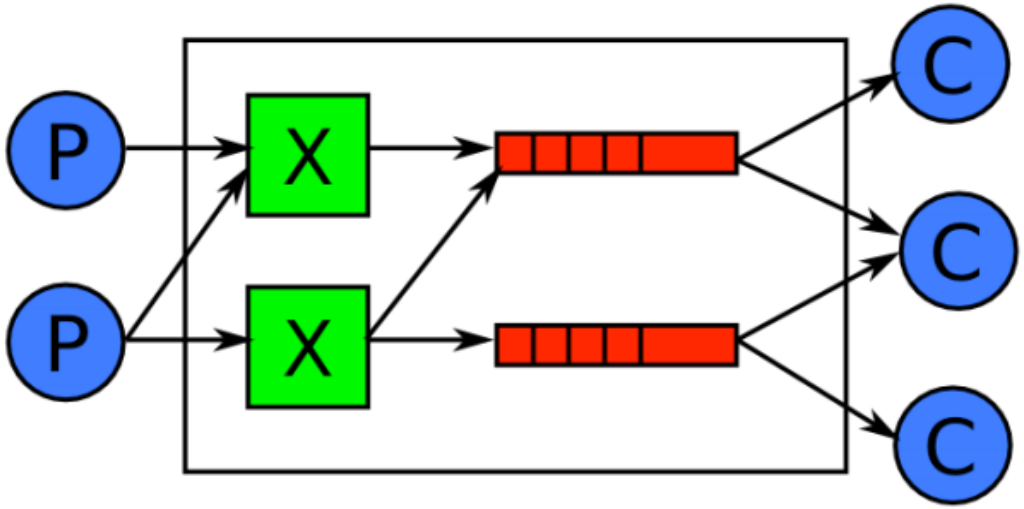
The following example shows how easily a client can be implemented with Paho using JavaScript

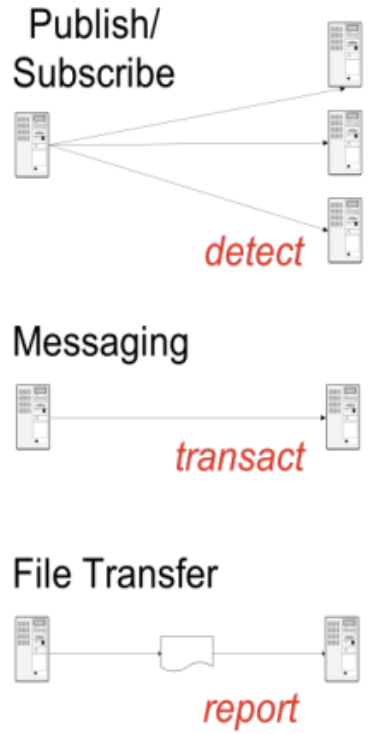
```
// Create a client instance
client = new Paho.MQTT.Client(location.hostname, Number(location.port), "clientId");

// set callback handlers
client.onConnectionLost = onConnectionLost;
client.onMessageArrived = onMessageArrived;

// connect the client
client.connect({onSuccess:onConnect});

// called when the client connects
function onConnect() {
  // Once a connection has been made, make a subscription and send a message.
  console.log("onConnect");
  client.subscribe("/World");
  message = new Paho.MQTT.Message("Hello");
  message.destinationName = "/World";
  client.send(message);
}

// called when the client loses its connection
function onConnectionLost(responseObject) {
}

// called when a message arrives
function onMessageArrived(message) {
  console.log("onMessageArrived:"+message.payloadString);
}
```

AMQP

Advanced Message Queuing Protocol
Open protocol originally developed by JPMorgan Chase bank
Asynchronous publish / subscribe protocol (like MQTT)
Reliable
Interoperable
Supports most of the programming languages
Supports most of the OS

AMQP Topology

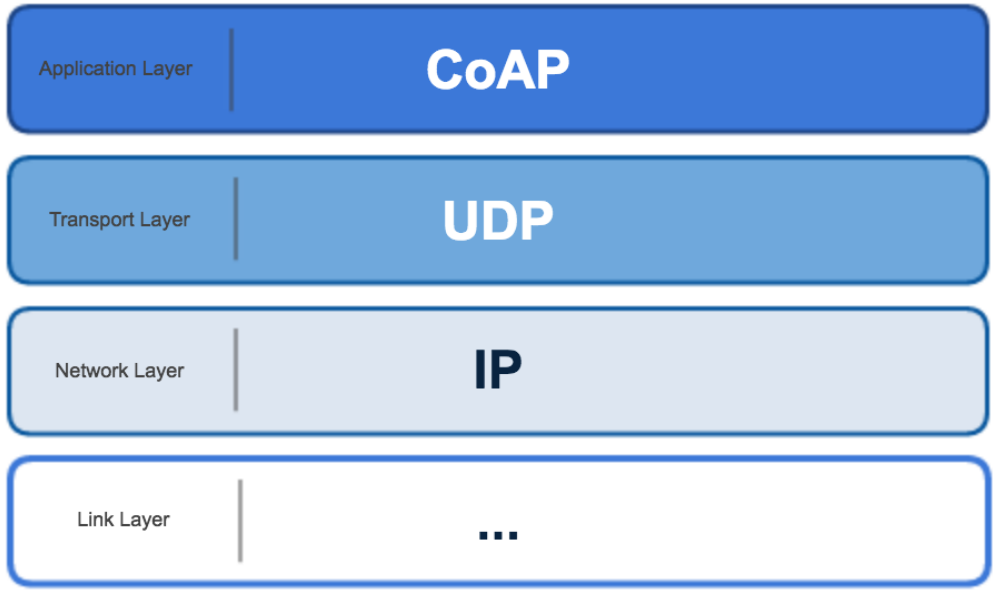




CoAP

Constrained Application Protocol

Constrained machine-to-machine web protocol
Representational State Transfer (REST) architecture
Simple proxy and caching capabilities
Asynchronous transaction support
Low header overhead and parsing complexity
URI and content-type support
UDP binding (may use IPsec or DTLS)
Reliable unicast and best-effort multicast support
Built-in resource discovery



Witekio

EMBEDDING SUCCESS

Internet of Things

Gateway Application

Existing software solutions

Node-red - IBM
Kura – Eclypse Fundation
Ubiworx - Ubiworx
Many others….

Node-RED

Based on Node JS, it is flow-based development tool for visual programming

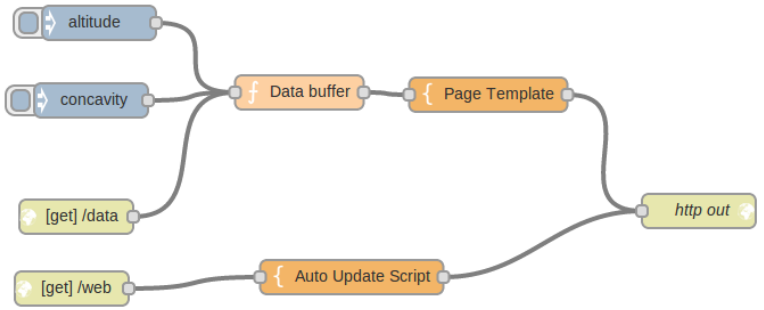
Developed in JavaScript

Describes flow between node-red modules

- MQTT
- HTTP
- JSON

Simple and intuitive

Can build web services

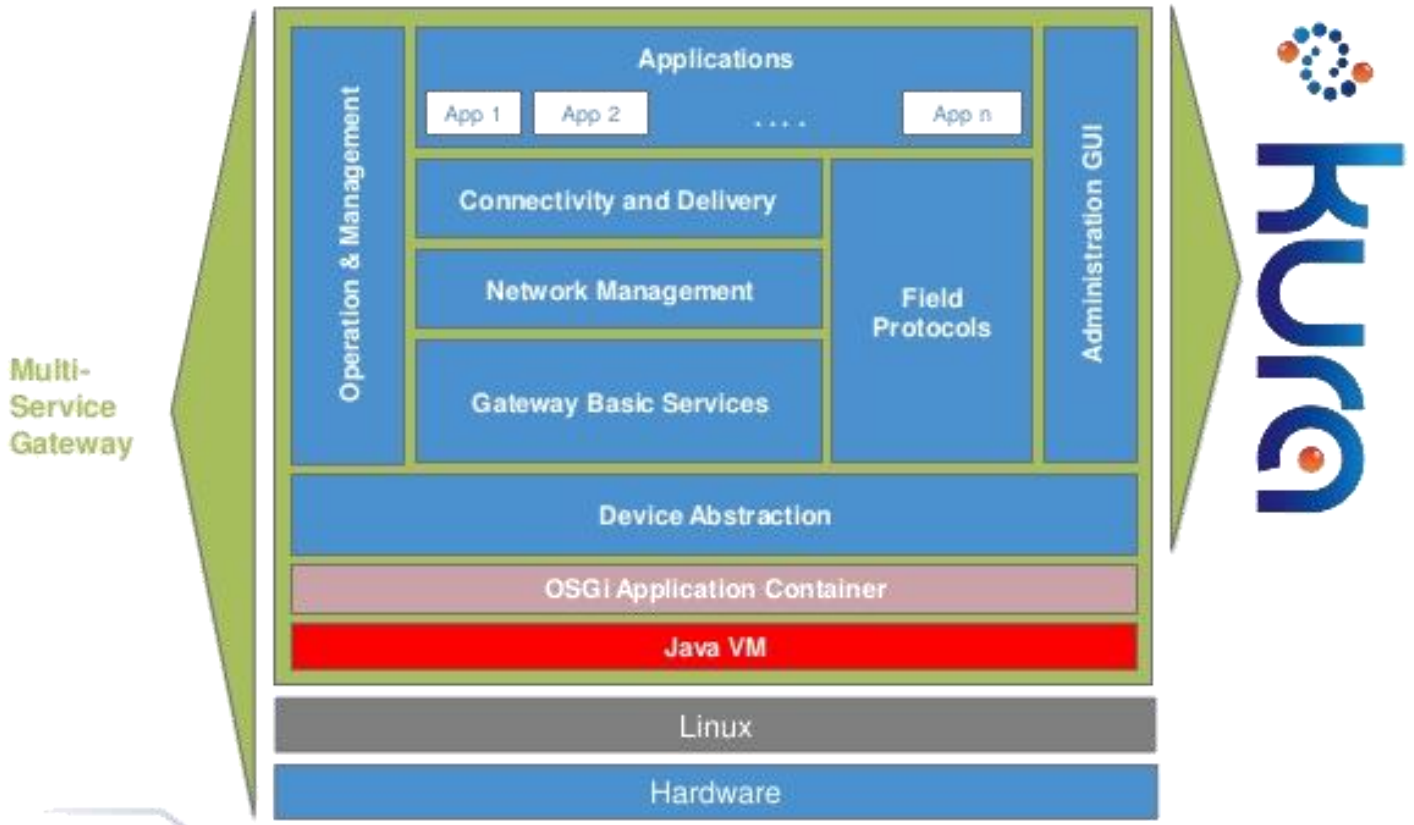


Kura - Eurotech

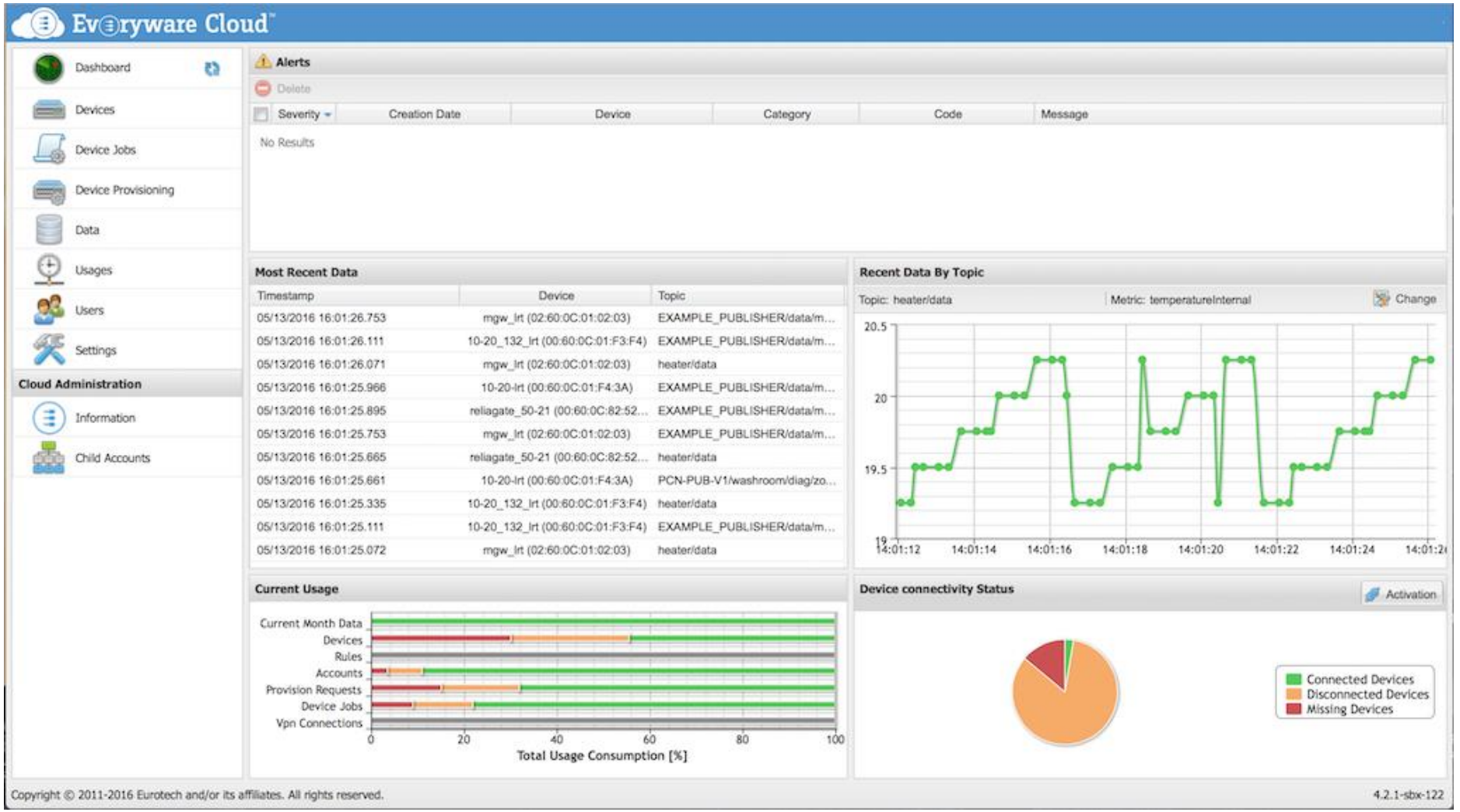
Java based gateway application, that act as a node in the Internet of Things
Applications can be remotely deployed and managed
The Kura package is completed with a web front end which allows the developer or administrator to remotely log in and configure

Functional Architecture

Decoupling functional layers

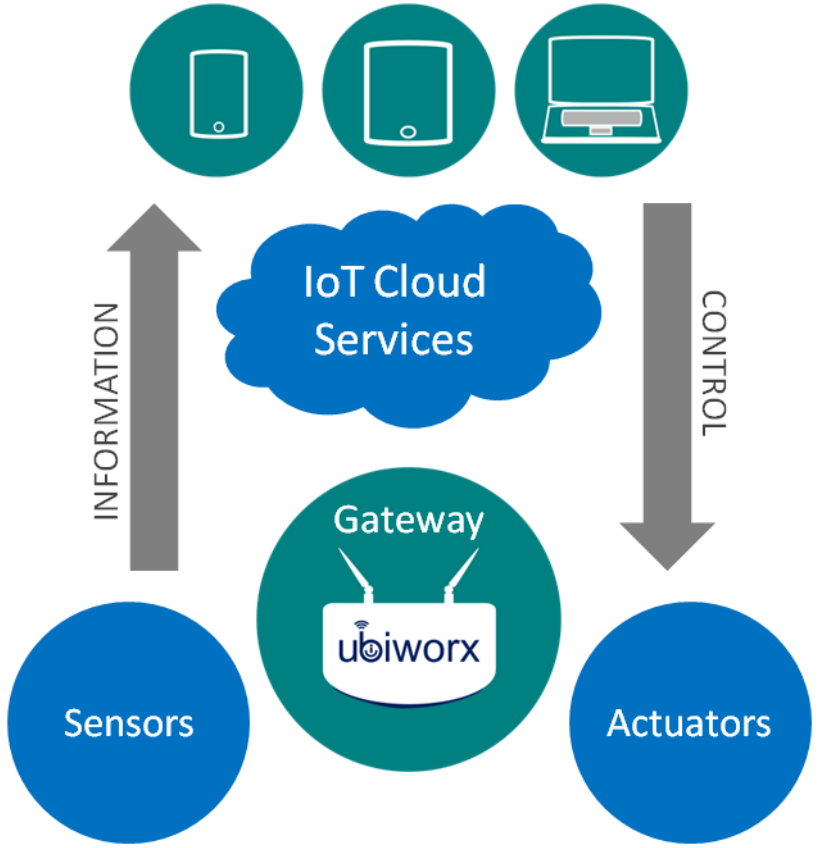


Solution Cloud Eurotech

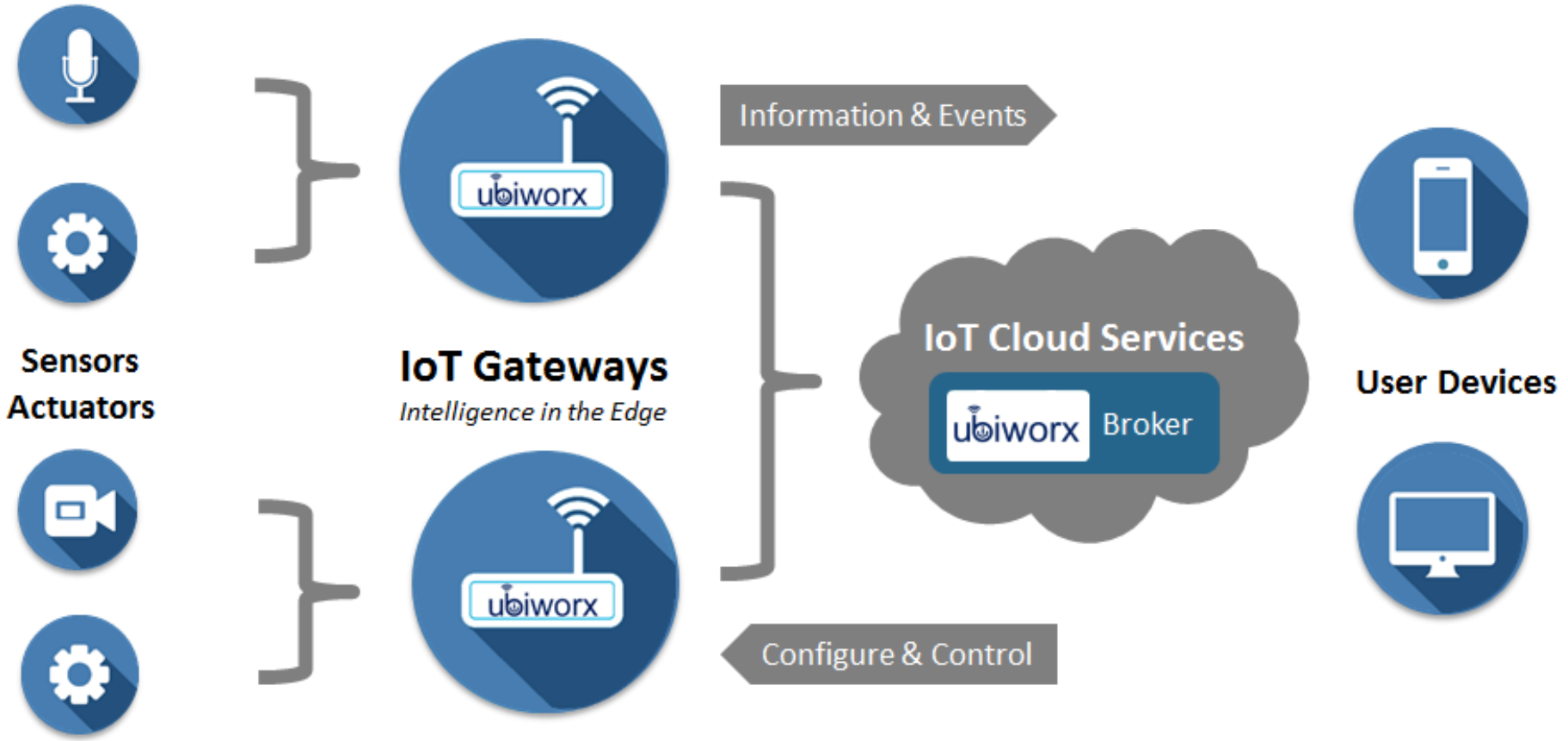


ubiworx

ubiworx is an IoT software framework for embedded systems running Linux. ubiworx enables embedded systems to bridge sensors, actuators and machines with data analytics systems and smart phone applications to form complete IoT Gateways in Internet of Things solutions.



ubiworx


Sensors
Actuators
ubiworx
IoT Gateways
Intelligence in the Edge
ubiworx
Information & Events
Configure & Control
IoT Cloud Services
ubiworx Broker
User Devices

Witekio

EMBEDDING SUCCESS

Internet of Things

Cloud

Definition

The word "cloud" is commonly used in science to describe a large agglomeration of objects that visually appear from a distance as a cloud and describes any set of things whose details are not inspected further in a given context.

In IoT context “Cloud” is used for all server/system hosting by a external/general service provider

- Typical example is Microsoft Azure, Amazon AWS or IBM Bluemix

Covers large set of feature, from simple file/data storage to advanced web site hosting and management

More and more IoT broker based system

- Microsoft : IoT Hub, Amazon : AWS IoT …

Definition

In a cloud solution we have

- Data storage (file or database)
- Web server for web site or web service hosting
- Message broker
- Data analyze and calculation system
- Cron/batch type job
- Virtual Machine for custom usage
- Virtual network management
- … and so much more

Definition

Cloud isn't necessarily hosted by a 3rd party company
Cloud means connected to the Internet network
Cloud means be accessible from everywhere
Cloud means a new way of thinking : do the job on server side !
Cloud means security management !

Where to host

Cloud can be a external provider (Microsoft, Amazon, IBM, Oracle …)

- All type of infrastructure can be found
- Your data is in the Cloud !
- Very high scalability

Cloud can be private and manage internal

- Need infrastructure knowledge (installation, update, security, up time supervising …)
- Strong server configuration knowledge
- Strong application server knowledge
- Your data stays private ! But not necessarily better protected

Where to host

It depends of your needs and your internal knowledge

- Each IoT scenario has a well adapted solution : you have to defined it

Pricing calculation has to taking account all parameters

- Yes, 3rd party cloud solution has a price BUT:
  - 100% up time has a cost in a private cloud solution !
  - Get the team with the good knowledge is complex

Cloud infrastructure solutions

IaaS : Infrastructure as a Service

- Manage physical platform/server and network in the cloud
- Manage Operating system and no more

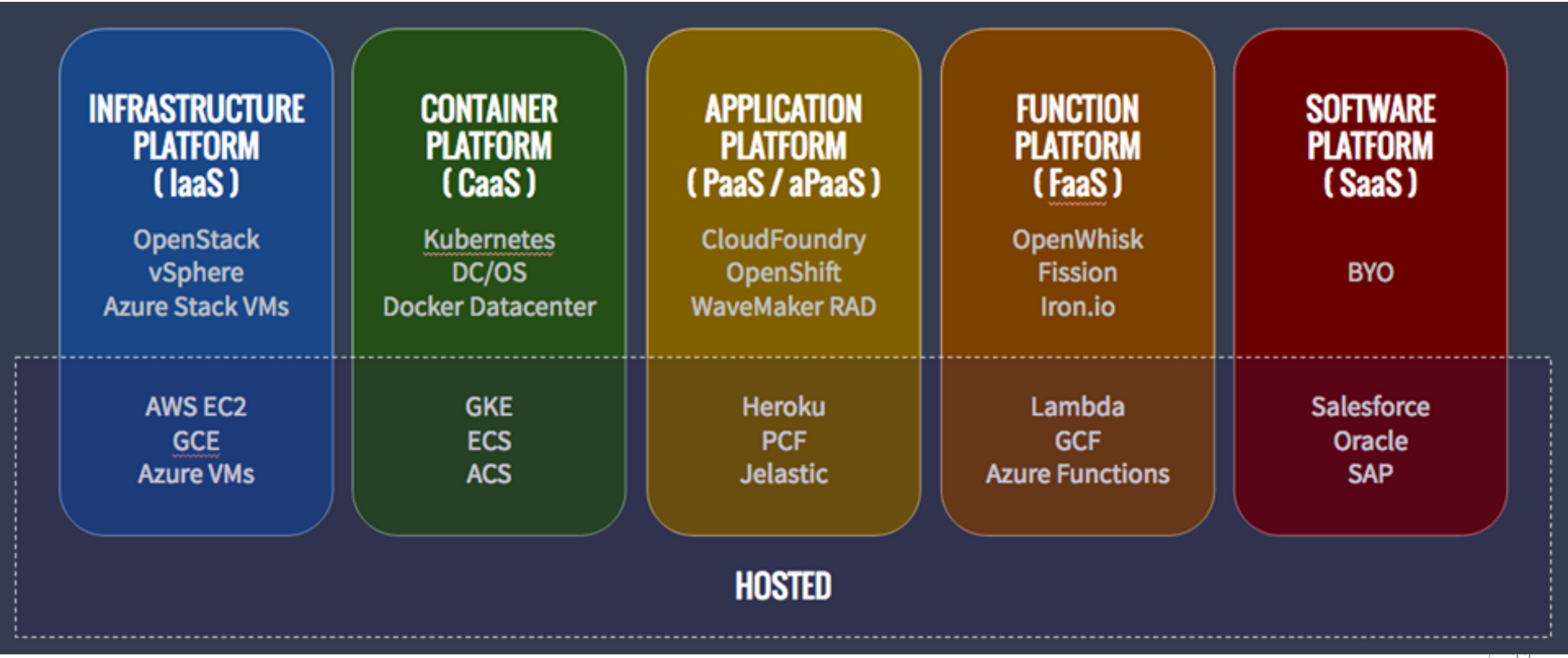
PaaS : Platform as a Service

- High level operating system management
- Middleware hosting
- Application server hosting (NOT application hosting)
- Decades Paas providers : http://www.paasify.it/vendors

SaaS : Software as a Service

- The highest level
- Application is managed by the cloud provider

Datacenter Architecture Solution



Amazon AWS

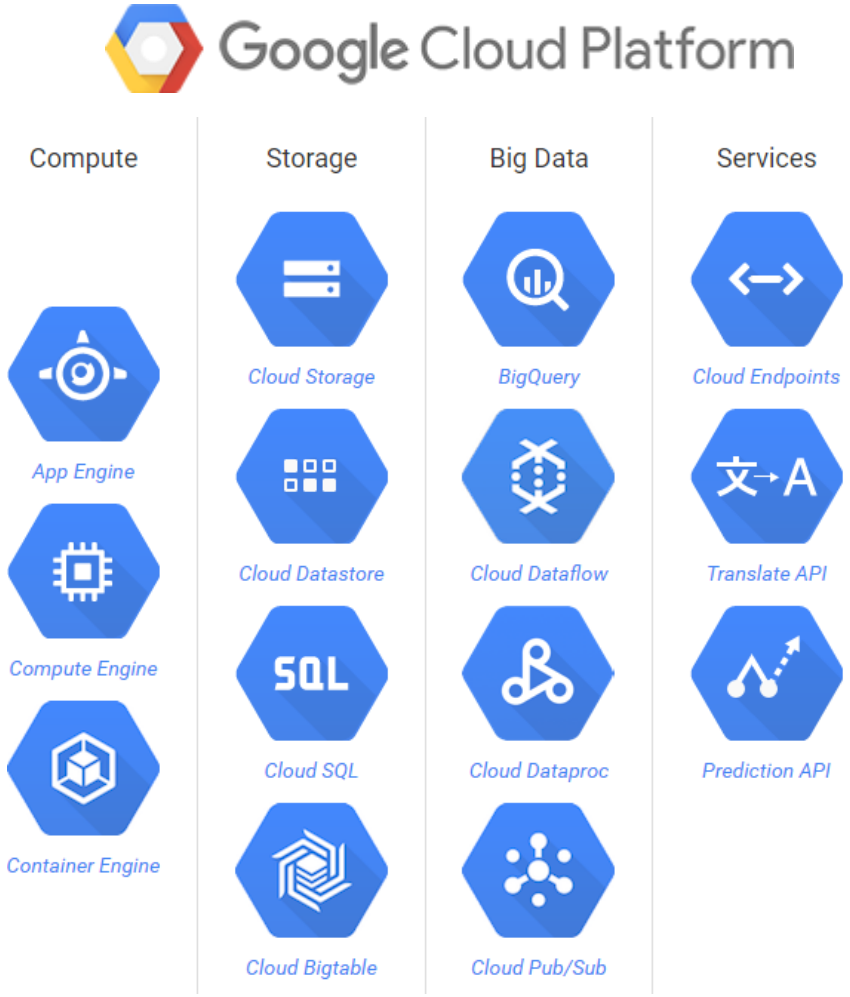
Cloud virtual machine hosting service (PaSS) with ready to use applications like git, wordpress, …
Firewall and Security solutions
https://aws.amazon.com

Microsoft Azure

Cloud hosting solution for web sites and web applications
Azure IoT Suite for IoT applications with front end AMQP broker
Azure Power BI for Business Intelligence
Azure Analytics for data processing and storage

Google Cloud

IaaS, PaaS, SaaS solution



IBM Cloud platform

IaaS, PaaS, SaaS solution

IBM Bluemix : Docker support to speed up environment bringup.

Web page hosting

Webserver hosting, is much less expensive than cloud solution
Various type of languages supported (php, ...)
Database support.

Private hosting

In house web hosting solutions, with web server, database, …

Pros:

- The flexible solution
- Buid your own custom scaled solution

Cons:

- Installation, configuration … done in house
- Maintenance, update … done in house

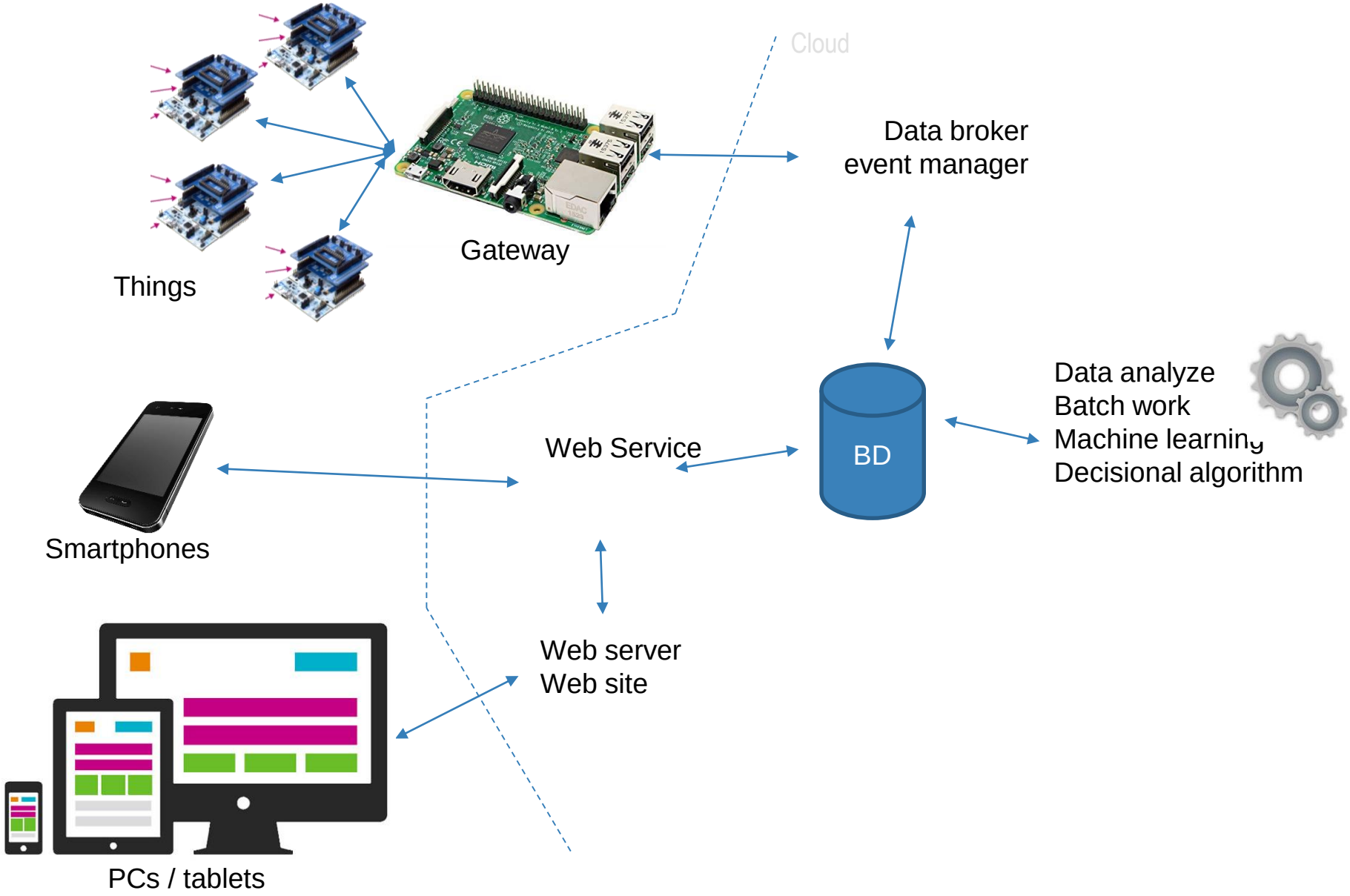
Witekio

EMBEDDING SUCCESS

Internet of Things

Architecture

Cloud sample architecture



Architecture

Things objects

- As seen previously
- OS or OS less system
- Sensor and actuator provider

Gateway

- A bridge between connected objects and the ‘server’ side world
- Unique entry point of your network
- Permit not IP capable device to communicate with the server

Data broker / event manager

- Lot of kind of MQ technology
- Lot of commercial solution available

Architecture

Database

- Store your data
- Lot of technology provider (MySQL, SQLServer, Oracle, SQLite, PostgreSQL, MongoDb, Cassandra …)
- SQL vs NoSQL solution

Data analyze / Batch work / Machine learning / Decisional algorithm

- Run job on server side to consume the available data and output added value
- Can be real time analysis
- Can be complex/long term analysis

Architecture

Web Service

- Manage API accessible through internet
- Follows standardization (REST Json …)
- To be secure ! It is a entry point to your system / to your data

Web site

- A specific version of your IoT system application
- Classical internet development system
- LOT OF technology available (ASP.NET, PHP, J2EE, NodeJS …)
- LOT OF framework available (Symfony, ASP MVC, Strust …)

Web server

- Host web service and web site

Architecture

Mobile Application

- Can be a responsive Web Site
- Multiple OS to support (Android, iOS, Windows …)
  - Cross platform Framework available
- Smartphone and tablet support
- HMI is important ! Use designer knowledge for the waaoo effect.
- HMI reflects the entire system quality for the user.

Witekio

EMBEDDING SUCCESS

Internet of Things

Databases

Definition

A database is an organized collection of data. It is the collection of schemas, tables, queries, reports, views and other objects. The data are typically organized to model aspects of reality in a way that supports processes requiring information, such as modelling the availability of rooms in hotels in a way that supports finding a hotel with vacancies.

Database ranking

345 systems in ranking, March 2019

Rank Mar 2019	Rank Feb 2019	Rank Mar 2018	DBMS	Database Model	Score Mar 2019	Score Feb 2019	Score Mar 2018
1.	1.	1.	Oracle	Relational, Multi-model	1279.14	+15.12	-10.47
2.	2.	2.	MySQL	Relational, Multi-model	1198.25	+30.96	-30.62
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	1047.85	+7.79	-56.94
4.	4.	4.	PostgreSQL	Relational, Multi-model	469.81	-3.75	+70.46
5.	5.	5.	MongoDB	Document	401.34	+6.24	+60.82

337 systems in ranking, November 2017

Rank Nov 2017	Rank Oct 2017	Rank Nov 2016	DBMS	Database Model	Score Nov 2017	Score Oct 2017	Score Nov 2016
1.	1.	1.	Oracle	Relational DBMS	1360.05	+11.25	-52.96
2.	2.	2.	MySQL	Relational DBMS	1322.03	+23.20	-51.53
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1215.08	+4.76	+1.27
4.	4.	4.	PostgreSQL	Relational DBMS	379.92	+6.64	+54.10
5.	5.	5.	MongoDB	Document store	330.47	+1.07	+5.00
6.	6.	6.	DB2	Relational DBMS	194.06	-0.53	+12.61
7.	7.	↑ 8.	Microsoft Access	Relational DBMS	133.31	+3.86	+7.34
8.	8.	↓ 7.	Cassandra	Wide column store	124.21	-0.58	-9.76
9.	9.	9.	Redis	Key-value store	121.18	-0.87	+5.64
10.	10.	↑ 11.	Elasticsearch	Search engine	119.41	-0.82	+16.84

Source : http://db-engines.com/

SQL vs NoSQL

SQL

- SQL or relational database has a static model defined at system configuration
- Store data in tables (with predefined columns)
- Classic in the data storage world / lot of competency available in administration and usage/development

NoSQL

- Not relational database does not have a static model.
- Store data in documents (Schemaless)
- Looks like flat file organize with description language
  - MongoDb use BSON (similar to JSON)

SQL vs NoSQL (Myth)

NoSQL is not a replacement of SQL but an alternative
NoSQL is not better or worse than SQL but it has different usage
NoSQL is not so much different than SQL
NoSQL and SQL database can be use from any languages/framework

Data representation

Sample board table

Name	SerialNumber
Nucleo1	123456789
Nucleo2	ABCDEF

Sample sensor table

Name	Size	Board
Sensor1	12.0	Nucleo1
Sensor2	16.5	Nucleo1

Data representation (mongodb)

Sample board information (one document needed)

```
{
        name: “nucleo1”
        serialNumber: ”123456789”
        sensors: [
                {
                        name: “sensor1”,
                        size: 12.0
                },
                {
                        name: “sensor2”,
                        size: 16.5
                },
        ]
},
{
        name: “nucleo1”
        serialNumber: ”ABCDEF”
        sensors: []
}
```

Data query

SQL	NoSQL
insert a new book record	
INSERT INTO book (`ISBN`, `title`, `author`) VALUES ('9780992461256', 'Full Stack JavaScript', 'Colin Ihrig & Adam Bretz');	db.book.insert({ ISBN: "9780992461256", title: "Full Stack JavaScript", author: "Colin Ihrig & Adam Bretz" });
update a book record	
UPDATE book SET price = 19.99 WHERE ISBN = '9780992461256'	db.book.update({ ISBN: '9780992461256' }, { $set: { price: 19.99 } });
return all book titles over $10	
SELECT title FROM book WHERE price > 10;	db.book.find({ price: { >: 10 } }, { _id: 0, title: 1 }); The second JSON object is known as a **projection**: it sets which fields are returned (_id is returned by default so it needs to be unset).

Data query

count the number of SitePoint books	
SELECT COUNT(1) FROM book WHERE publisher_id = 'SP001';	db.book.count({ "publisher.name": "SitePoint" }); This presumes denormalized documents are used.
return the number of book format types	
SELECT format, COUNT(1) AS `total` FROM book GROUP BY format;	db.book.aggregate([{ $group: { _id: "$format", total: { $sum: 1 } } }]); This is known as **aggregation**: a new set of documents is computed from an original set.
delete all SitePoint books	
DELETE FROM book WHERE publisher_id = 'SP001'; Alternatively, it's possible to delete the publisherrecord and have this cascade to associated bookrecords if foreign keys are specified appropriately.	db.book.remove({ "publisher.name": "SitePoint" });

Join

Join data from multiple table is in SQL dna

Join data in NoSQL model is « not possible »

- Latest Mongodb version 3.2 integrate new aggregate function to support Join like feature

NoSql (to go further)

Performance : it depends of the use case.

- For sure less performant for Join type operation
- But can manage huge amount of data

Scalability

- NoSql is easy to scale

Support / existing knowledge

- Hardest to find the good NoSQL dba !

MapReduce and advance feature

- NoSql solution offer feature not available in standard DB
- MapReduce permit huge volume treatment (on parallel machine)

Sql vs NoSql conclusion

Projects where SQL is ideal:

- logical related discrete data requirements which can be identified up-front
- data integrity is essential
- standards-based proven technology with good developer experience and support.

Projects where NoSQL is ideal:

- unrelated, indeterminate or evolving data requirements
- simpler or looser project objectives, able to start coding immediately
- speed and scalability is imperative.

Internet of Things

Web site

Web Site

A web site is a set of page interpreted by a web browser

Base on 3 major technologies

- HTML description language
- CSS for style
- JavaScript (JS) for a browser interpreted script execution

Page are generated and sent by a Web server

- Exist a lot of technology for
  - Generate HTML page on server side (can also be static)
  - Serve page through the internet network

Web Site

Available server side technology

- PHP
- ASP.NET
- Java J2EE
- Node.JS
- Python
- CGI
- …

Web Site

Available client/browser side technology

- Native JS
- JQuery
- Angluar JS

Witekio

EMBEDDING SUCCESS

Internet of Things

HTTP and server

HTTP Protocol

Goal of HTTP protocol is to permit file transfer (essentially HTML format) identify by a string named URL between a web browser (client) and a web server
It also apply for Web Service request that generate a file and send this response to the Web service caller.
It is a application layer (OSI model) which use TCP/IP protocol on port 80 that can be secured with SSL/TLS stack on port 443.

- 1. browser send a request
- 2. Server handle the request
- 3. Server send the response to the browser

HTTP Request

GET : Client want a resource at a specific URI. (Parameters are included in the l'URI)

POST : Client send data a specific URI (parameter are included into the request core)

```
telnet www.adeneo-embedded.com 80
GET http://www.adeneo-
embedded.com/index.html
<HTML>
<HEAD>
<TITLE> … … … </TITLE>
</HEAD>
<BODY>
… … …
</BODY>
</HTML>
Connection closed by foreign host
> 
```

HTTP Response code

Code	Message	Meaning
200	*OK*	Success
301	*Moved Permanently*	Document has been moved permanently
302	*Moved Temporarily*	Document has been moved temporary
400	*Bad Request*	Request syntax is bad
401	*Unauthorized*	You are not autorized to access this resource
403	*Forbidden*	Authentication refused
404	*Not Found*	Resource not found
500	Internal error	An error occurs during the request handling

Apache Tomcat

Apache Tomcat™ is an open source software
Supports : Java Servlet, JavaServer Pages,
Java Expression Language and Java WebSocket technologies
Web applications are WAR packages containing instructions for application installation.

Microsoft IIS 7

Web Server (IIS) role in Windows Server 2008 R2, and the Web server in Windows 7
Supports : ASP.NET

NodeJS

Node.js® is a JavaScript runtime built
on Chrome's V8 JavaScript engine.
Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem, npm, is the largest ecosystem of open source libraries in the world.

Nginx

nginx [engine x] is an HTTP and reverse proxy server, a mail proxy server, and a generic TCP proxy server

Security and Risks

Define your enemy

Script kiddies/Hacker (White hat/Grey hat,Black hat) / NSA...

Different Goals

- Just for the glory
- Destroy the device and make it unusable
- Change the device behaving and make it dangerous
- Data theft
- Inject malicious code to run massive attacks against services.

According to Proofpoint, more than 25 percent of the botnet was made up of devices other than computers, including smart TVs, baby monitors and other household appliances.

Supervisory Control And Data Acquisition

Stuxnet: a worm targeting SCADA system

2010

Stuxnet is a malicious computer worm, first uncovered in 2010 by Kaspersky Labs, the antivirus company. Thought to have been in development since at least 2005, stuxnet targets SCADA systems and was responsible for causing substantial damage to Iran's nuclear program. Although neither country has admitted responsibility, since 2012 the worm is frequently described as a jointly built American/Israeli cyberweapon

Smart Grid

Ukraine's power outage was a cyber attack: Ukrenergo

Jan 2017

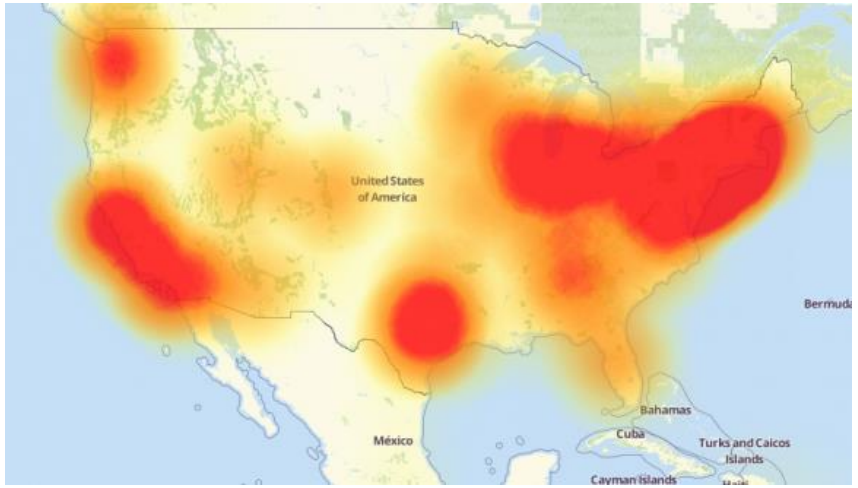
KIEV/MILAN (Reuters) - A power blackout in Ukraine's capital Kiev last month was caused by a cyber attack and investigators are trying to trace other potentially infected computers and establish the source of the breach, utility Ukrenergo told Reuters on Wednesday.

Camera

Hacked Cameras, DVRs Powered Today's Massive Internet Outage

Oct 2016

A massive and sustained Internet attack that has caused outages and network congestion today for a large number of Web sites was launched with the help of hacked "Internet of Things" (IoT) devices, such as CCTV video cameras and digital video recorders, new data suggests.
Earlier today cyber criminals began training their attack cannons on **Dyn**, an Internet infrastructure company that provides critical technology services to some of the Internet's top destinations. The attack began creating problems for Internet users reaching an array of sites, including Twitter, Amazon, Tumblr, Reddit, Spotify and Netflix.



Cars

THE JEEP HACKERS ARE BACK TO PROVE CAR HACKING CAN GET MUCH WORSE

ALMOST EXACTLY two years ago, Chrysler announced a recall for 1.4 million vehicles after a pair of hackers demonstrated to WIRED that they could remotely hijack a Jeep's digital systems over the Internet. For Chrysler, the fix was embarrassing and costly.

Countermeasures

Protect the devices from attacks

- Avoid hackers to enter the device (physically or remotely)

Protect the data from attacks

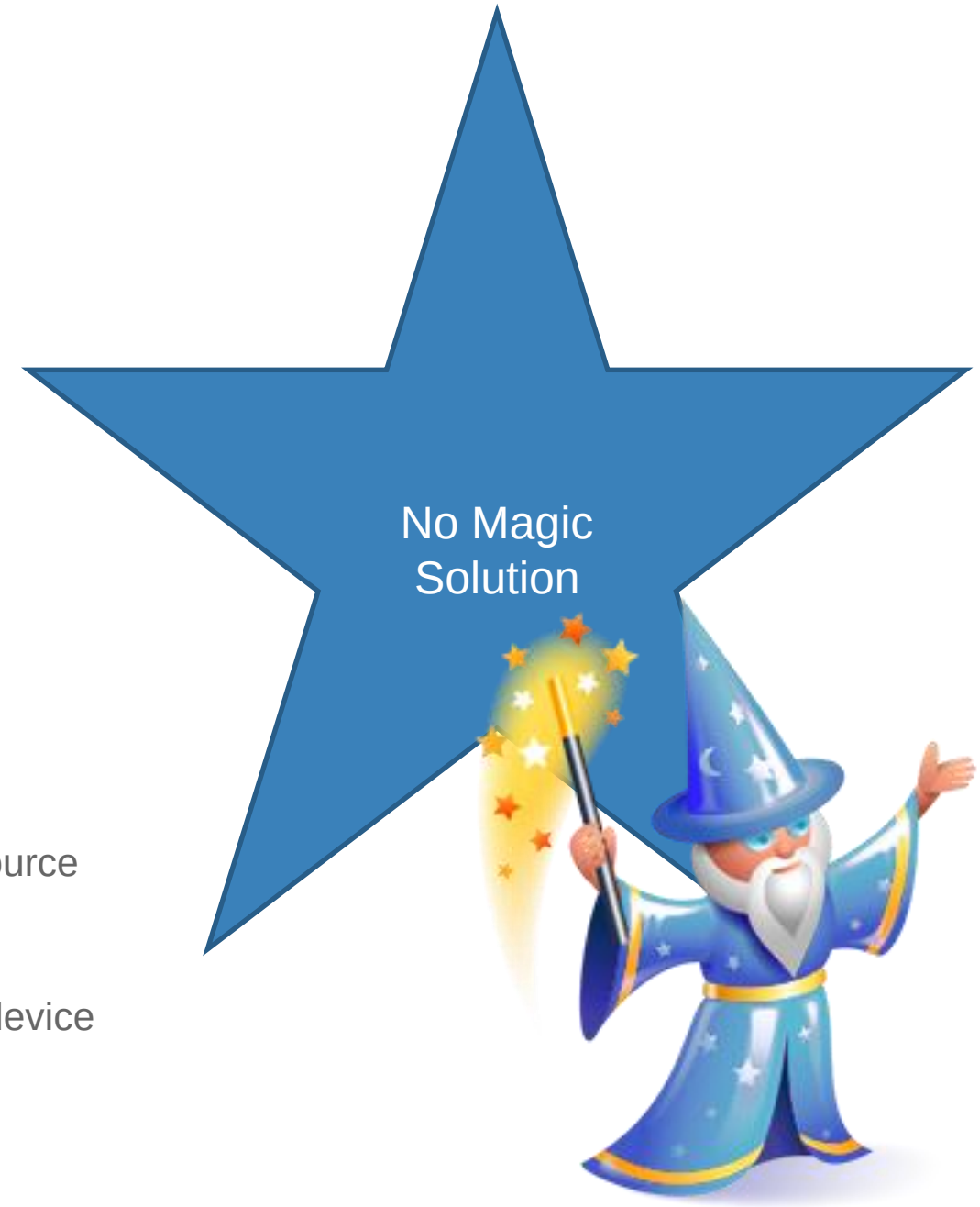
- Data encryptions
- Secure communication

Protect the device update from attacks

- Avoid hackers to force a system update from a compromise source

Protect devices during production

- Avoid injection of malicious code during the production of the device



Secure Gateway

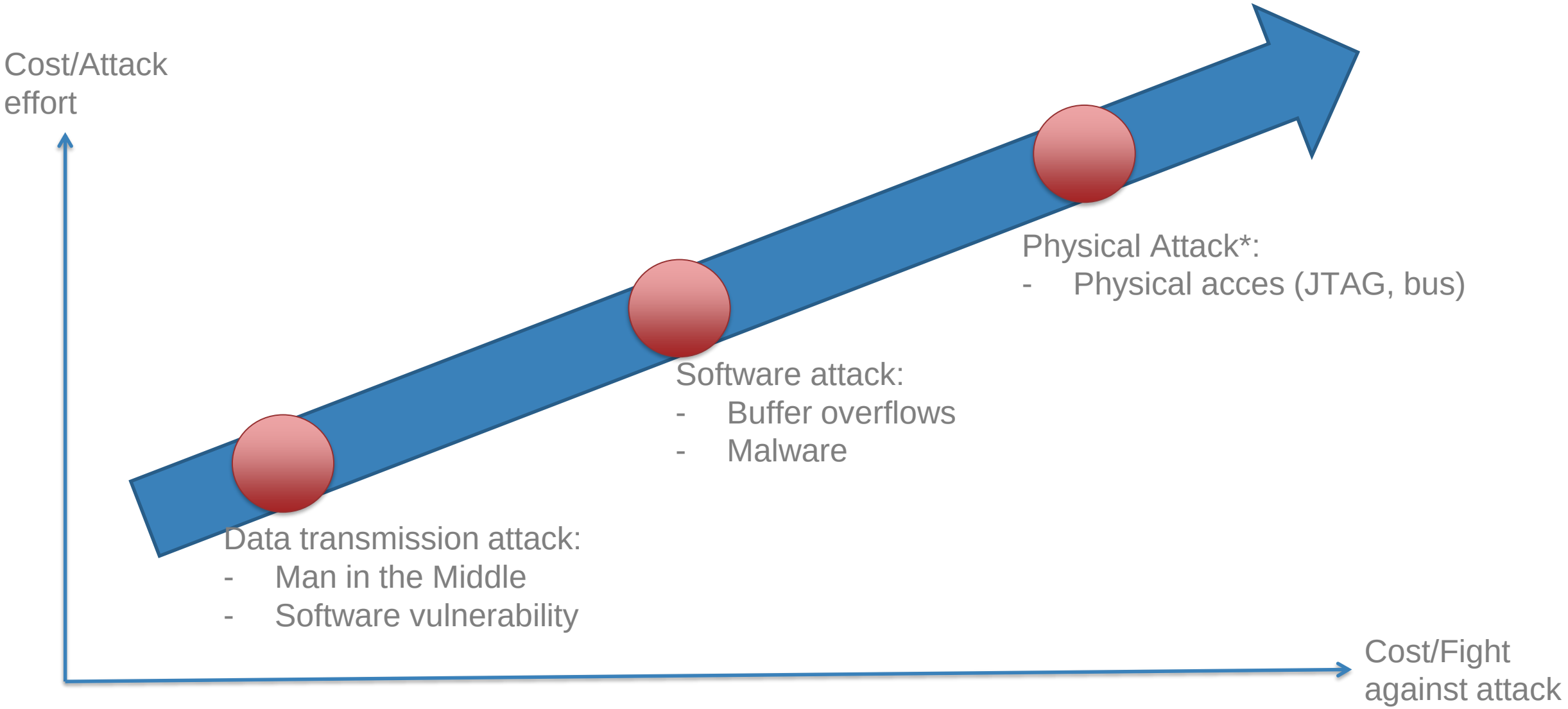
Risk Analysis / Threat Model

First starting point is the risk analysis of the platform based on :

- Use cases of the devices
- Physical interfaces (UART/JTAG)
- Communication medium
- Physical access to the device

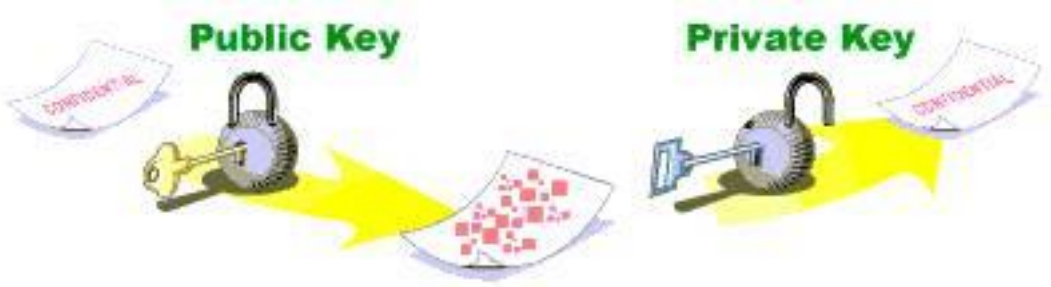
But not only the device itself, you have to focus on the criticity of the data that the hackers can get access to.

Ratio Cost/Effort



Cryptography

In cryptography, a private key (secret key) is a variable that is used with an algorithm to encrypt and decrypt code. Quality encryption always follows a fundamental rule: the algorithm doesn't need to be kept secret, but the key does. Private keys play important roles in both symmetric and asymmetric cryptography.



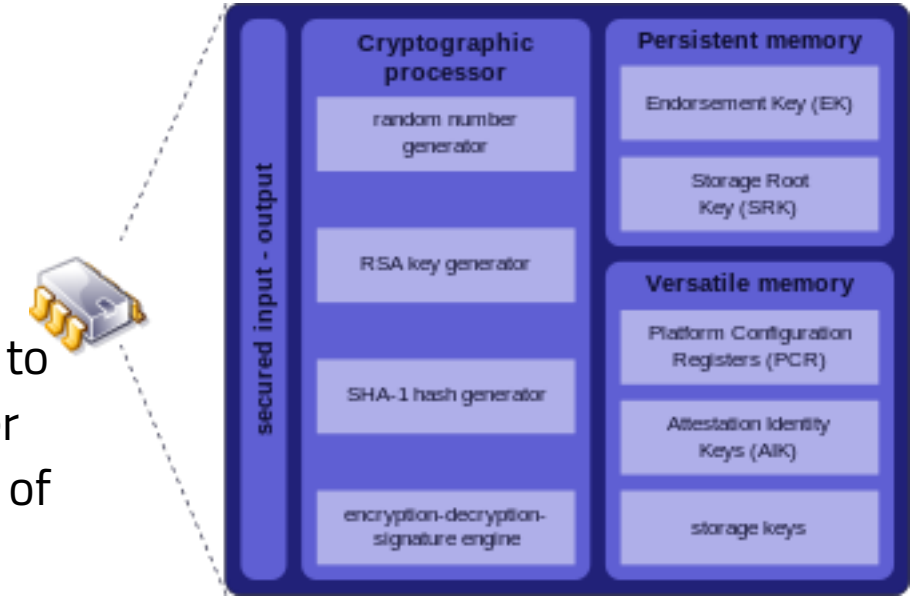
Secure Elements

Secure Gateway

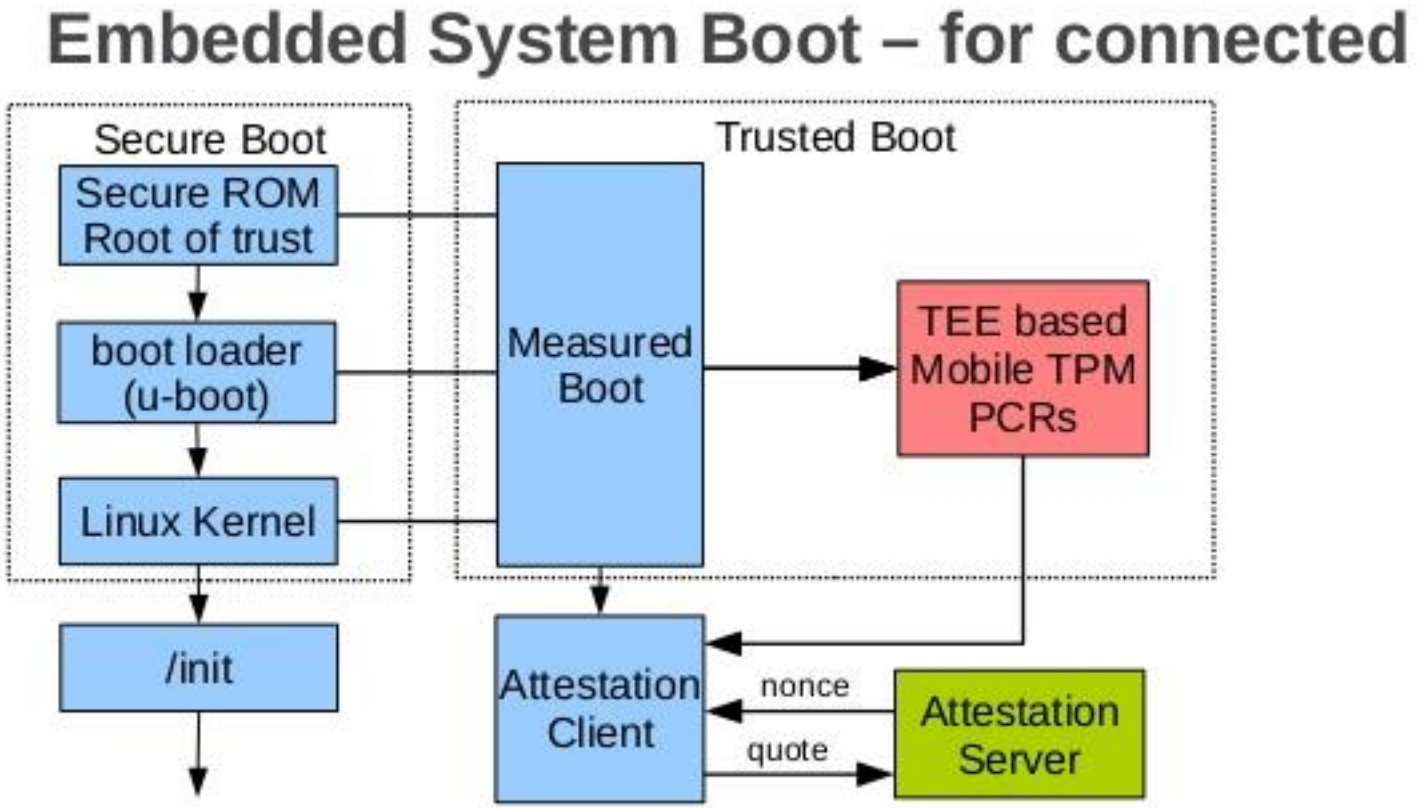
TPM : Trusted Platform Module

Trusted Platform Module (TPM) technology is designed to provide hardware-based, security-related functions. A TPM chip is a secure crypto-processor that is designed to carry out cryptographic operations. The chip includes multiple physical security mechanisms to make it tamper resistant, and malicious software is unable to tamper with the security functions of the TPM. Some of the key advantages of using TPM technology are that you can:

- Generate, store, and limit the use of cryptographic keys.
- Use TPM technology for platform device authentication by using the TPM's unique RSA key, which is burned into itself.
- Help ensure platform integrity by taking and storing security measurements.



Trusted Boot



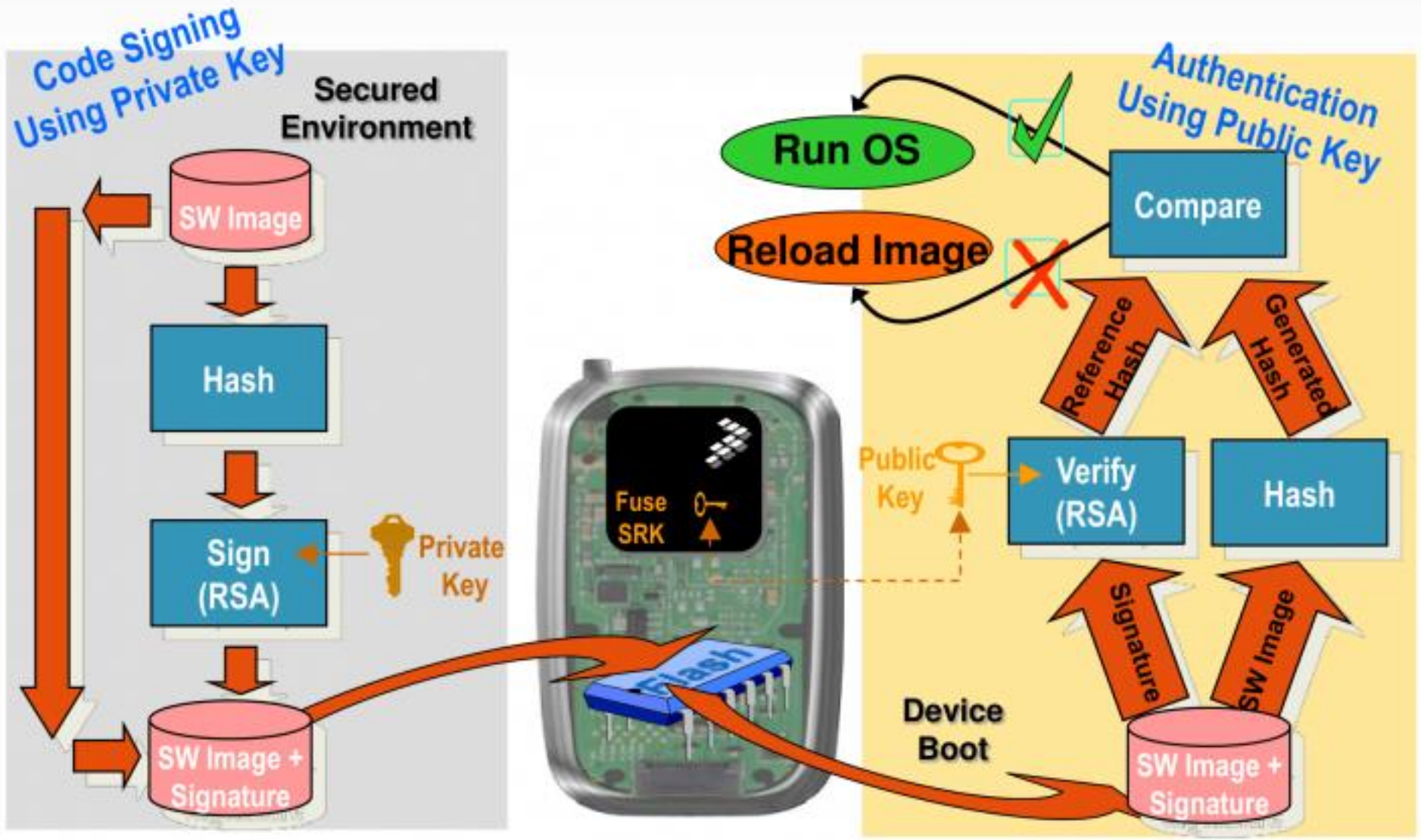
High Assurance Boot (HAB)

NXP – ~~Freescale~~ i.MX processors have optional feature, which allows you to make sure only software **images signed by you can be executed** on the SOC.

Silicon incorporates boot ROM level security which cannot be altered after programming the appropriate one-time electrically programmable fuses (eFuses). The boot ROM is responsible for loading the initial software image from the boot medium (usually this initial software is a bootloader such as SPL/U-Boot. HAB enables the boot ROM to authenticate the initial software image by using digital signatures. It also provides a mechanism to establish a chain of trust for the remaining software components (such as the kernel image) and thus to establish a secure state of the system.

Buffer overflow in the I.MX bootrom : https://blog.quarkslab.com/vulnerabilities-in-high-assurance-boot-of-nxp-imx-microprocessors.html

High Assurance Boot (HAB)



Protect against malware

Secure Coding :

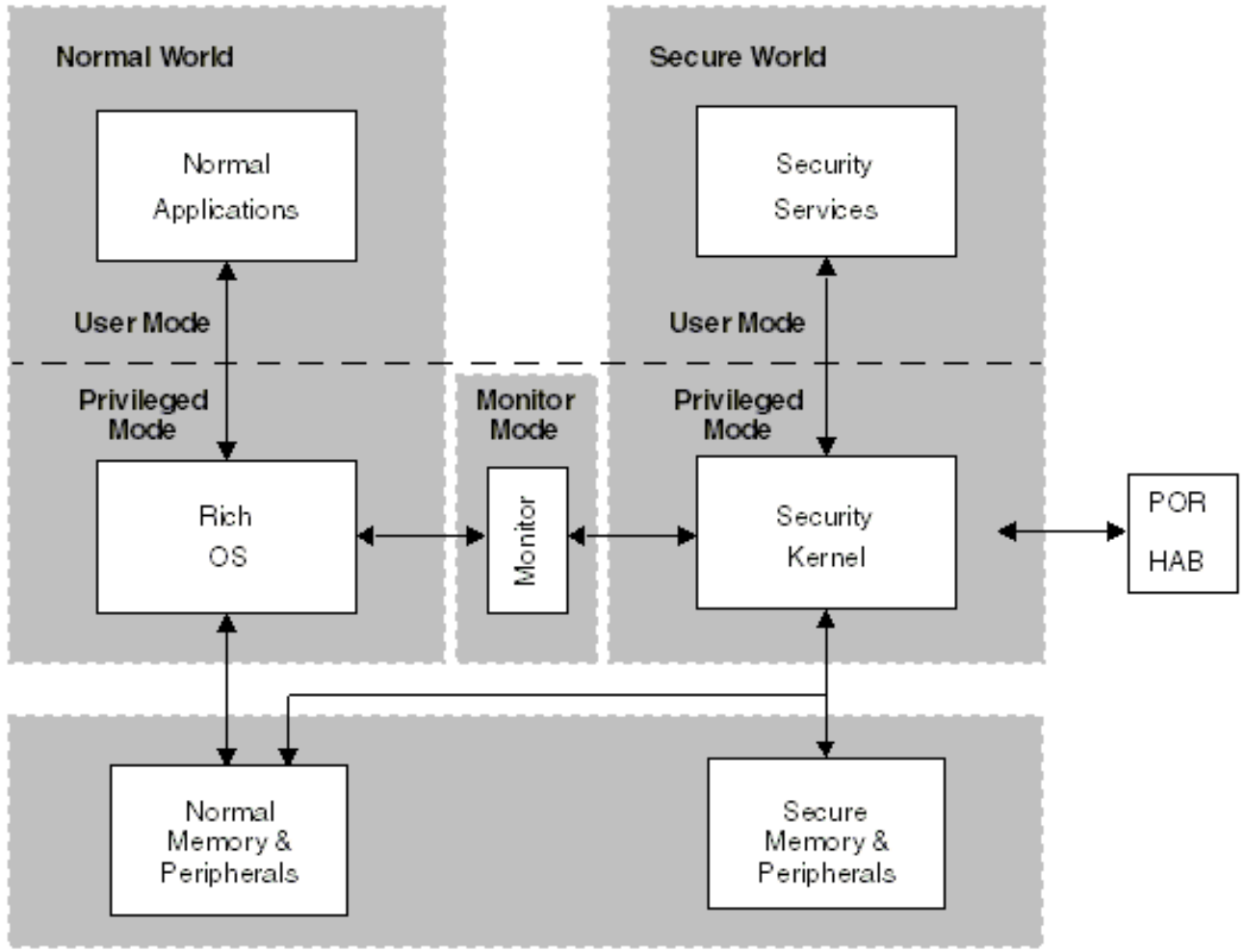
- Static code analysis
- Code coverage
- Hardening Kernel/Code
- Stack Protection (Canary)
- Fuzzing / PenTesting

klocwork
a Rogue Wave Company

coverity®
A Synopsys Company

sonarqube

Trust Zone ARM Cortex A and M



Protect the devices from attacks

Reduce as much a possible the attack surface

Wired Network

- Close all the port that could

Sub GigaHertz network

- use encrypted data transmission protocol

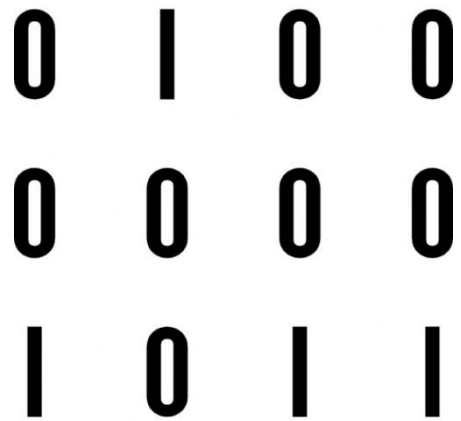
Bluetooth & BLE

- enable encryption and pairing

Wifi

- WPA2 …. No more secure, so encrypt your data even on an encrypted medium

Protect the data from attacks



Data encryption

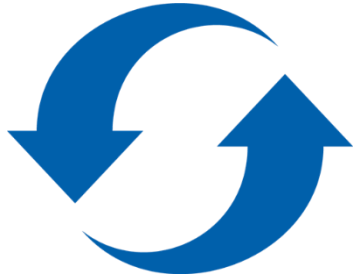
- Local data can be encrypted on the storage : use encrypted file system
- Transmission of the data need to be secured over TLS

Use secure connections

- Need trusted certificate, signed by a Certificate Authority
- For better compatibility, don’t self sign your certificate
- Let’s Encrypt as a free, automated, and open Certificate Authority.

Let’s Encrypt

Protect the device update from attacks



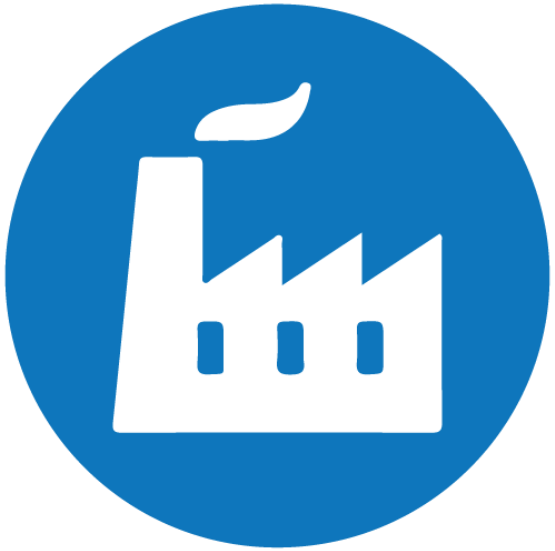
Firmware integrity

- Use digitally signed package to avoid corruption between source and destination

Firmware update reliability

- Ensure that the system resist to power outage during Firmware Update

Protect devices during production



Production should be subcontracted only to trusted companies

Never provide the private keys to those OEM

Digitaly Sign firmwares running on the platform

Secure Boot

Security Process

Security Team / Red Team & Pentesting

Bug bounty program

Disclosure policy

Maintain and update your product during all its life.

Follow recommendation from ANSSI, alert from CERT-FR,…

The ultimate solution

No perfect solution or recipe, only elements that you have to care about to secure your devices and gateways. Always start from an audit of the device use case and ensure that only those cases are handled.

Secure Cloud

Secure access to Cloud Service

History of Secure Socket Layer & Transport Layer Secure

Developed by Netscape in 1994 to secure Web communication

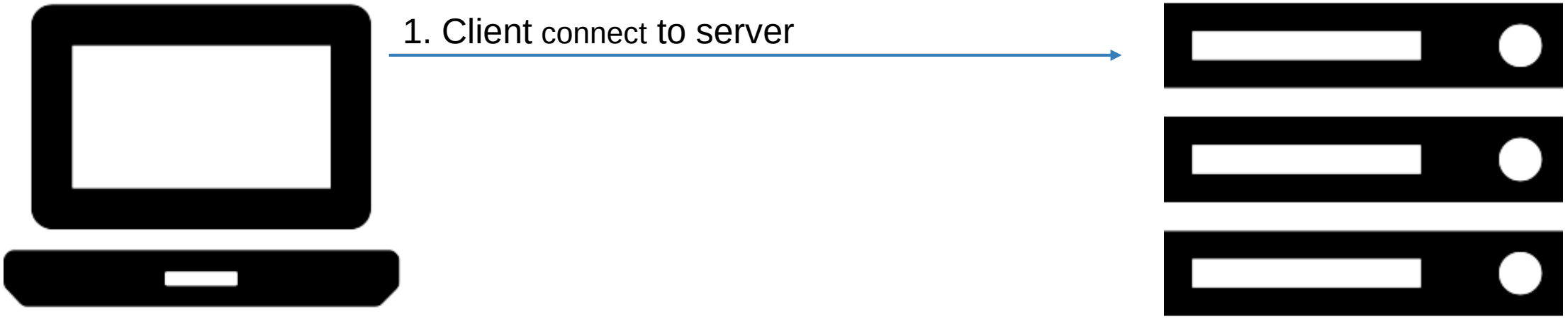
SSL v3.0 was designed as a IETF standard in 1996

TLS is the successor of SSL, TLSv1 have been introduce in 1999, TLSv1.1 on 2006 and TLSv1.2 in 2008

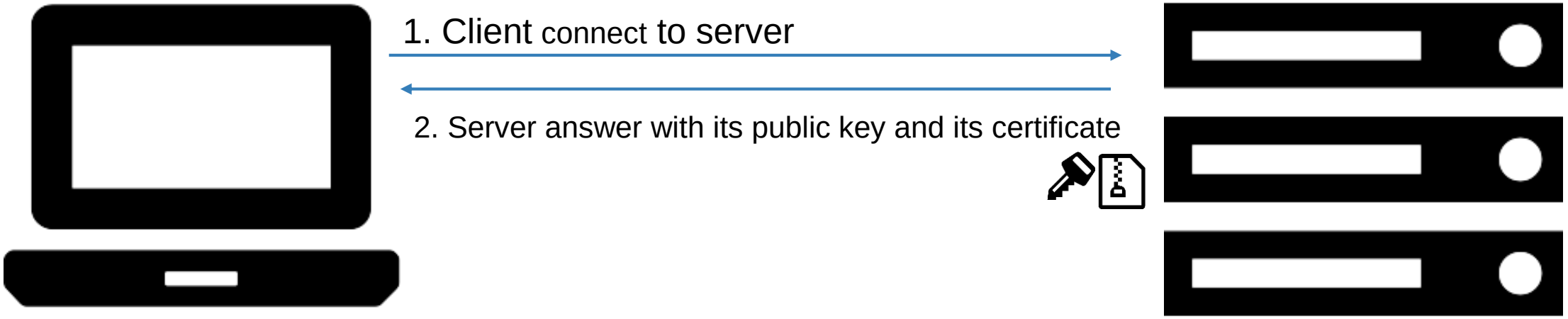
SSLv2 have been disapproved by IETF in 2011, SSLv3 have been disapproved by IETF in 2015

ANSSI recommendation : https://www.ssi.gouv.fr/uploads/2016/09/guide_tls_v1.1.pdf

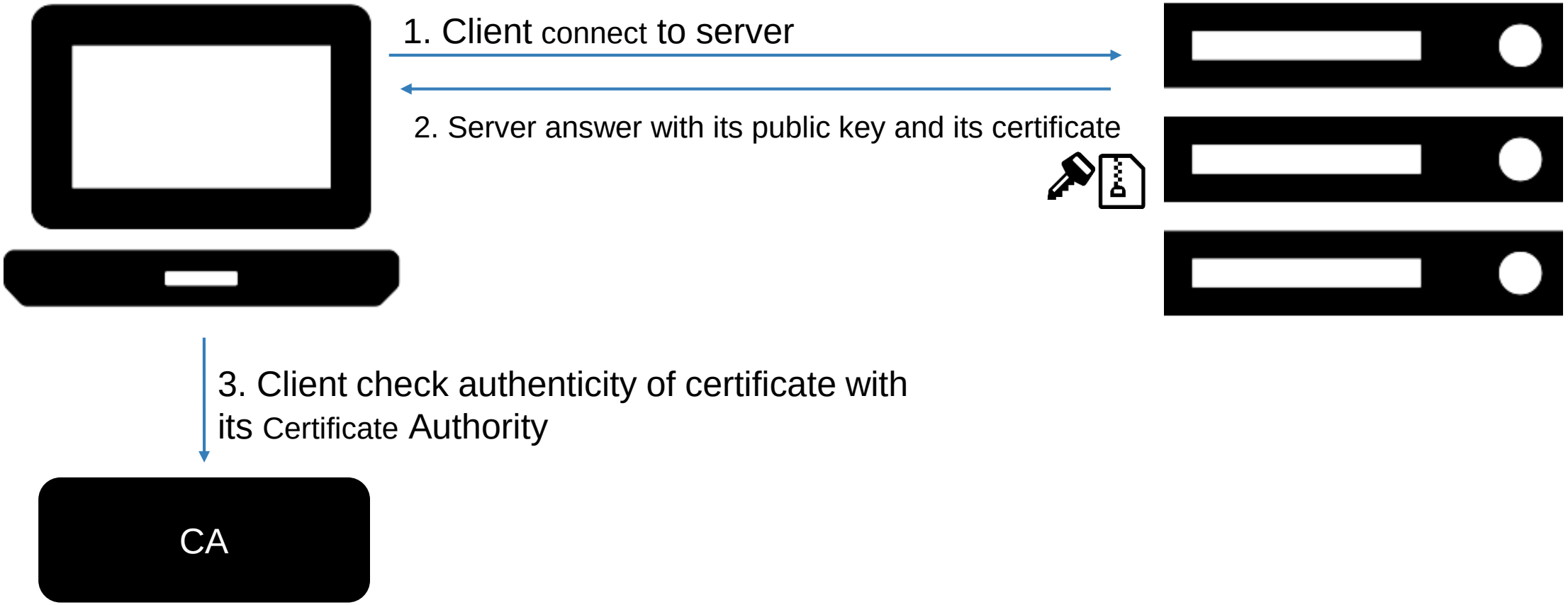
TLS principle 1/5



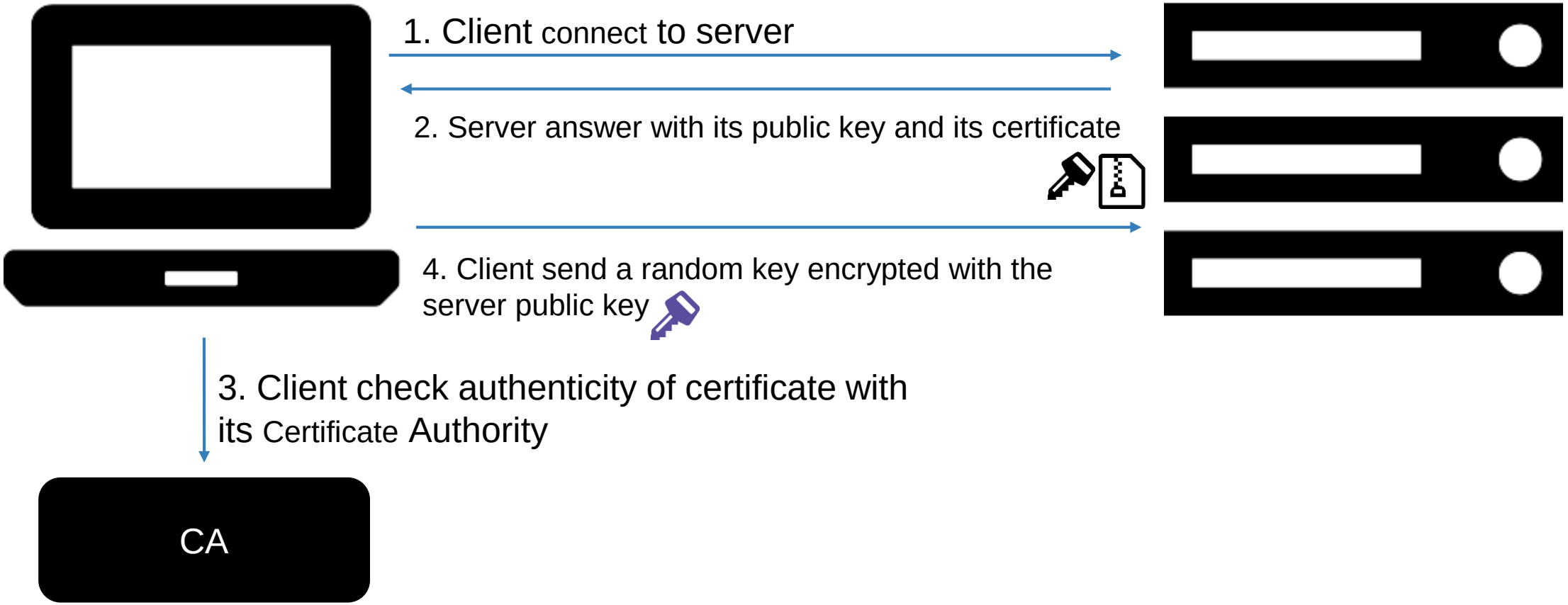
TLS principle 2/5



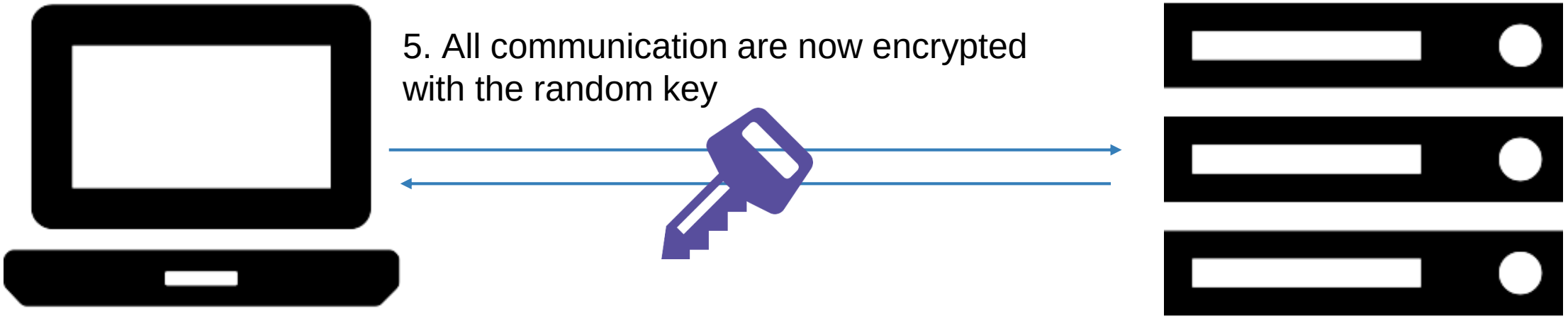
TLS principle 3/5



TLS principle 4/5



TLS principle 5/5



Certificate Transparency

The successor of HTTP Public Key Pinning

Launched by Google in 2013 to solve the trust problems with the Web's PKI. (IETF RFC 6962)

Log certificate in a Meerkle-tree (blockchain-like)

Chrome will only accept “CT trusted” certificate after April 2018.

Alterting solution on false certificate : https://sslmate.com/certspotter/

The DNS-Based Authentication of Named Entities (DANE RFC6698) is a solution that add certificate information in DNS TLV section (require DNSSEC)

Know SSL attack

2009 Renegotiation attack

Renegotiation used when attempting to authenticate while in SSL session

Attacker initiates session with server

Hijacks connection between client and server by holding handshake packets sent by client

Requests authentication from server

Presents the clients handshake instead

Server now believes the attacker is the client

Mitigation : Use TLS1.1 or TLS1.2

Know SSL attack

2009 Renegotiation attack

Renegotiation used when attempting to authenticate while in SSL session

Attacker initiates session with server

Hijacks connection between client and server by holding handshake packets sent by client

Requests authentication from server

Presents the clients handshake instead

Server now believes the attacker is the client

Mitigation : Use TLS1.1 or TLS1.2

Know SSL attack

2011 BEAST (Browser Exploit Against SSL/TLS)
2012 CRIME (Compression Ratio Infoleak Made Easy)
2013 BREACH (Browser Reconnaisance and Exfiltration via Adpative Compression of Hypertex)

BEAST is a vulnerability in the way cipher block chaining is implemented in pre TLS 1.1 versions
Mitigation : TLSv1.1 or TLSv1.2

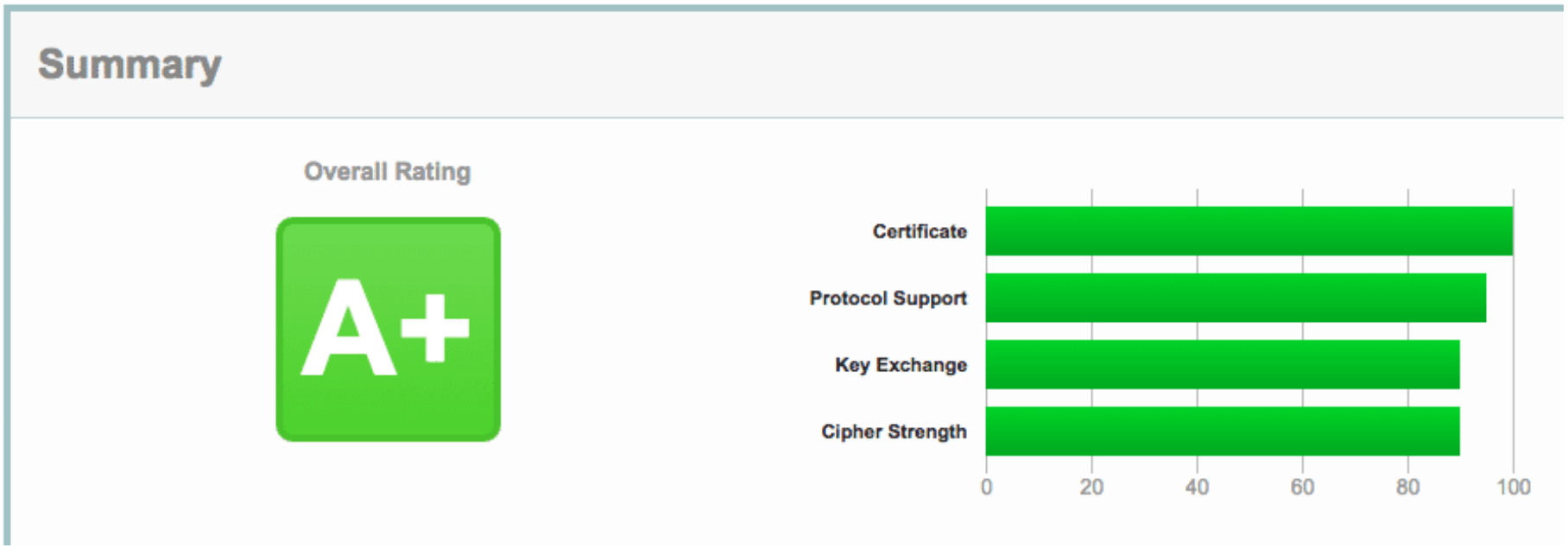
CRIME is based on a compression based attack.
Mitigation : Turn compression off

BREACH was the same compression based attack against the HTTP protocol itself
Mitigation: Difficult to mitigate because HTTP compression is widely used

Verify our TLS configuration

Lots of free tool exist for checking TLS compliance

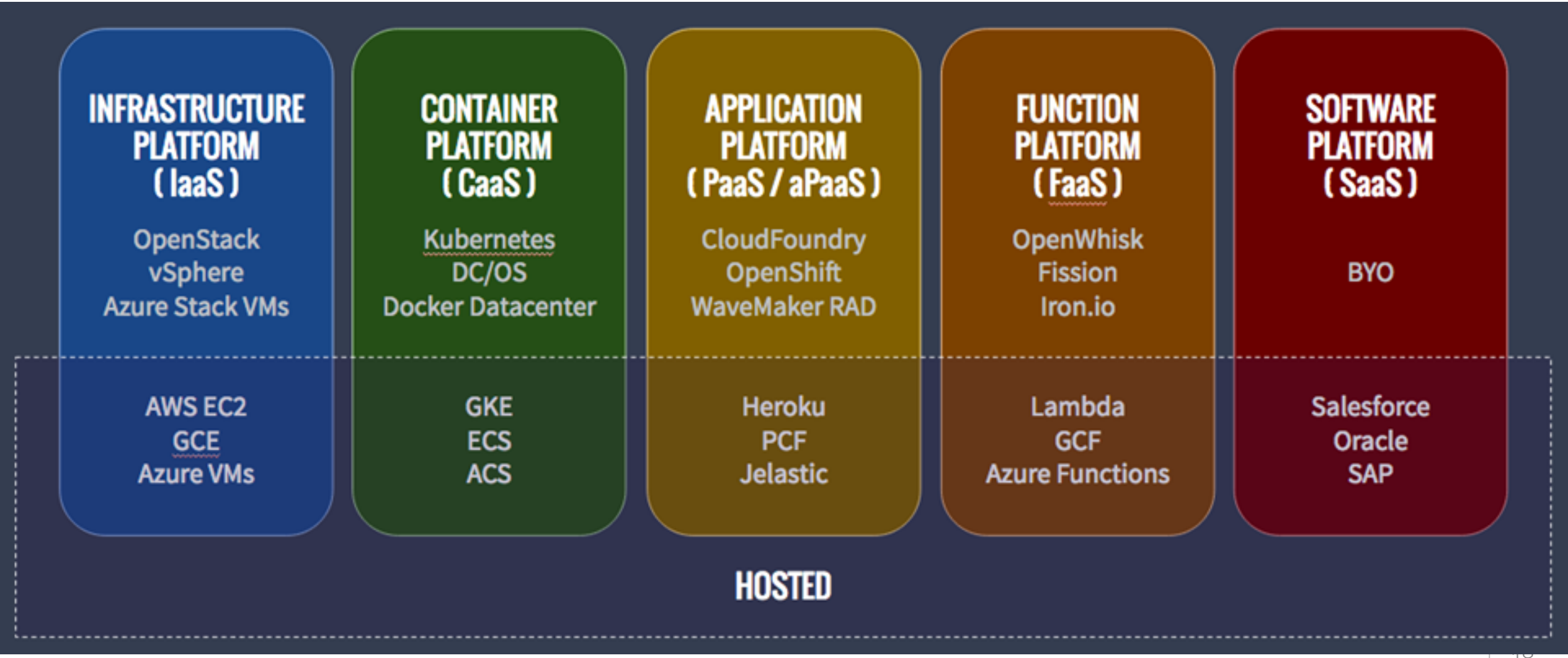
Eg : SSL Labs



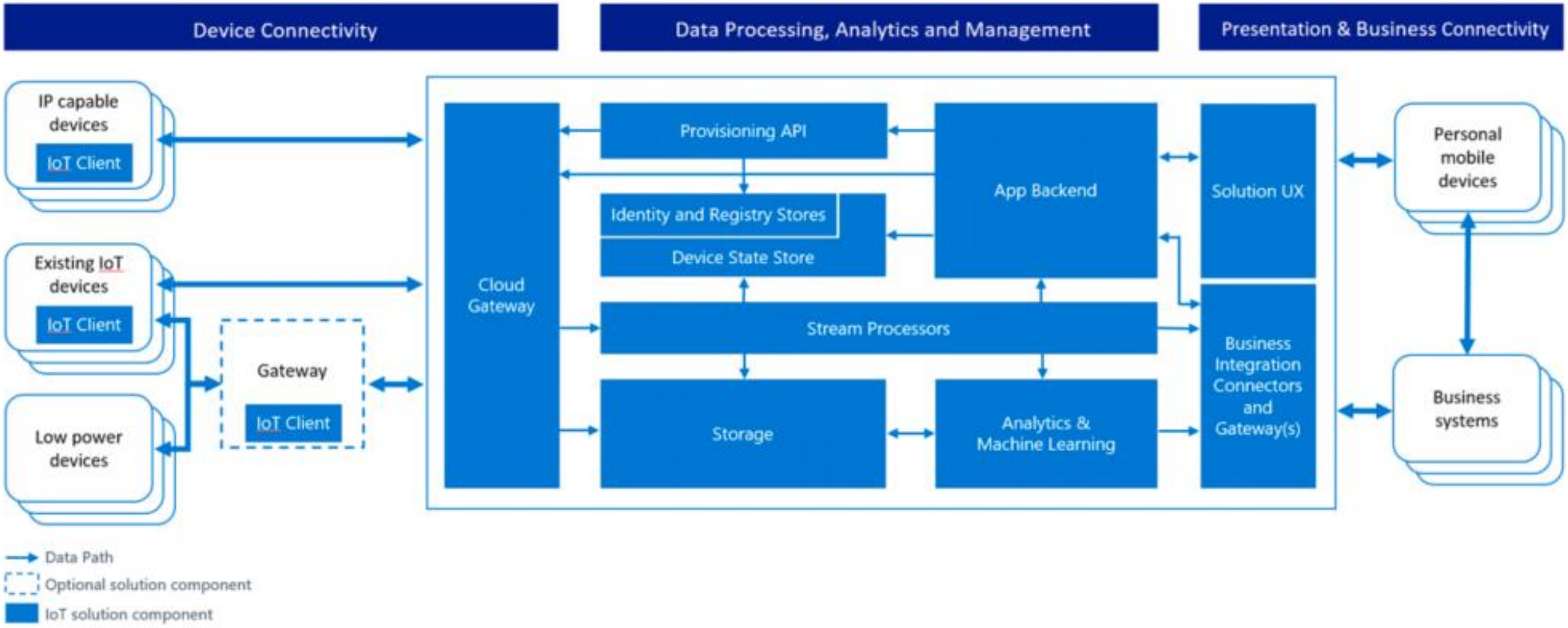
Cloud Scalability

Cloud Solutions and performances

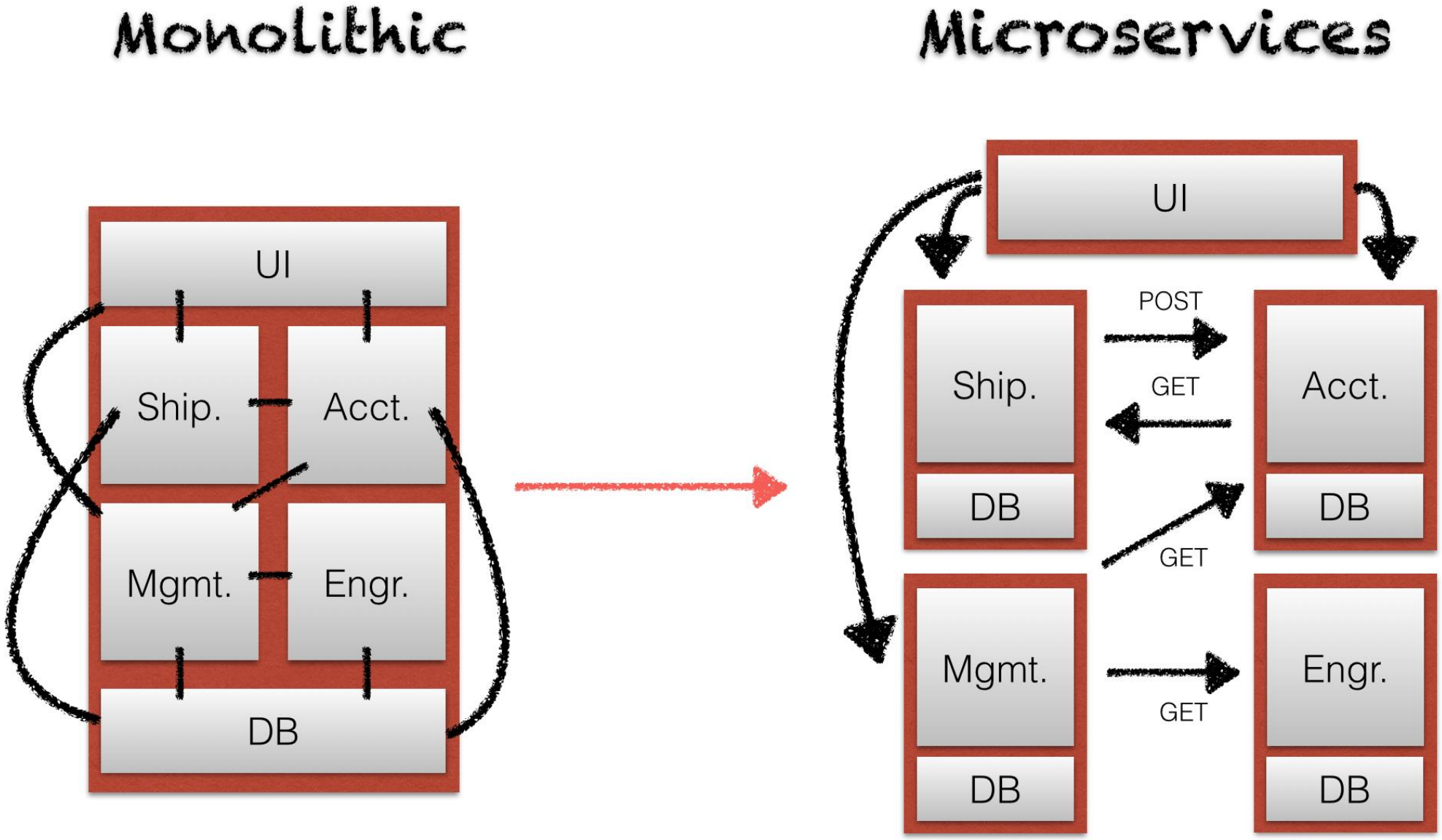
Datacenter Architecture Solution



Datacenter Architecture Solution



Micro-services architecture



Accountability & Commitment

Up to date expertise

Agility & Scalability **Reliability**

Local Teams **Worldwide presence**

Global Actions

Compliance with customers teams

Thank you

Witekio France
4, chemin du ruisseau
69134, Ecully France
Phone : + 33(0) 4 26 49 25 39
sales.emea@witekio.com

Witekio USA
3150 Richards Roads Suite 210 Bellevue,
WA, 98005, USA
Phone : + 1 425 749 4335
sales.amer@witekio.com

Witekio Germany
Am Wartfeld- 61169
Friedberg, Germany
Phone : + 49 6031 693 7070
sales.dach@witekio.com

Witekio UK
Hollywood Mansion, Hollywood Lane,
Bristol BS10 7TW, UK
Phone : + 44(0) 117 369 0930
sales.uk@witekio.com