Alma Mater Studiorum – University of Bologna
CdS Laurea Magistrale (MSc) in
Computer Science Engineering

Mobile Systems M course (8 ECTS)
II Term – Academic Year 2019/2020

04 – Internet of Things (IoT): Definitions and Application Scenarios

Paolo Bellavista
paolo.bellavista@unibo.it

Luca Foschini
luca.foschini@unibo.it

http://lia.disi.unibo.it/Courses/sm1920-info/

Internet of Things (IoT)

- Old term, probably its first appearance in 1999 by Kevin Ashton
- IoT term applied to RFID, supply chain, and the Internet

- Then, in 2009: "Conventional diagrams of the Internet include servers and routers and so on, but they leave out the most numerous and important routers of all: people. The problem is, **people have limited time, attention and accuracy** […] If we had **computers that knew everything** there was to know **about things** […] we would be able to […] greatly **reduce waste, loss and cost.** We would know when things needed replacing, repairing or recalling, and whether they were fresh or past their best

Internet of Things - Scenarios

- Since 1999 the concept evolved far beyond RFID
- Everything can now be connected to the Internet

- The term IoT now refers to different scenarios:
  - Wireless Sensor Networks (WSN)
  - Near Field Communications (NFC)
  - Biotechnology and Body Area Networks (BAN)
  - Machine-to-Machine communications (M2M)
  - Personal Area Networks (PAN)
  - ...

IoT - How Big?

- Possibly every single device and object will be connected to the Internet
- About 50-100 billions devices in 2020 (data from SAP/Intel and Ericsson)

- IBM Smarter Planet vision:
  - instrumented
  - interconnected
  - intelligent

- Several real world examples: industrial control systems, health monitoring, smart metering, home automation, …

Internet of Things - Use Cases

Smart Wearables

Smart Home

Smart City

Smart Agriculture

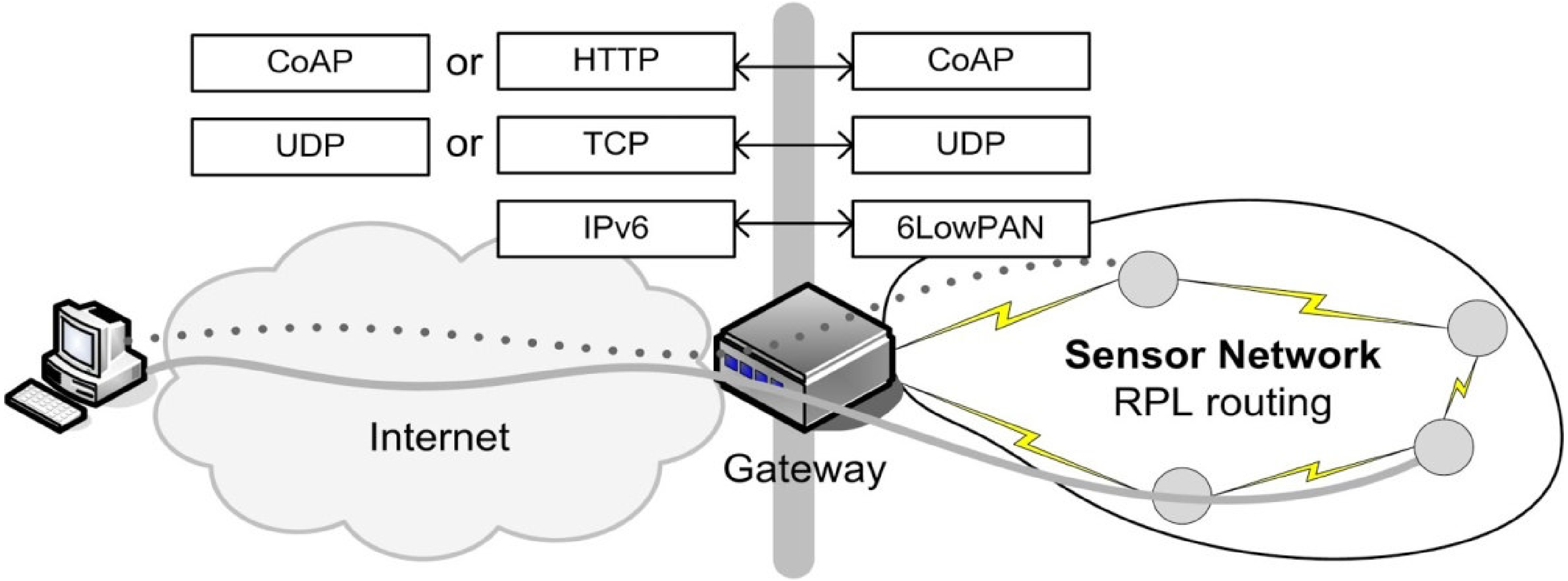
Connected Car

Health Care

Industry Automation

Smart Energy

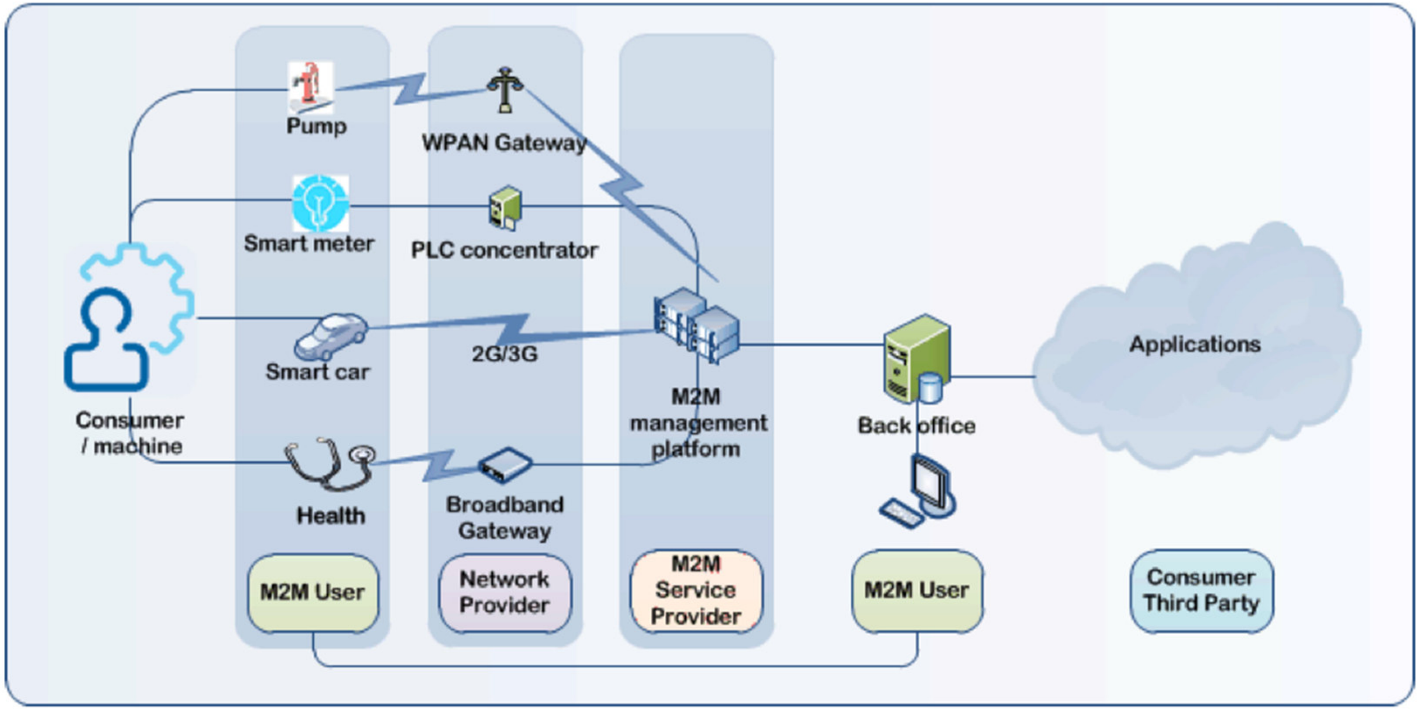
IoT - the General Idea



- The general idea behind the (commonly accepted) vision of IoT consists in the extension of Internet protocols to Wireless Sensors Networks (WSNs), composed of sensors as well as actuators

IoT for Manufacturing - Machine to Machine

- The term "Machine to Machine" (M2M) describes devices connected to each other, by using a variety of fixed/wireless networks and through the Internet to the wider world. They are active communication devices

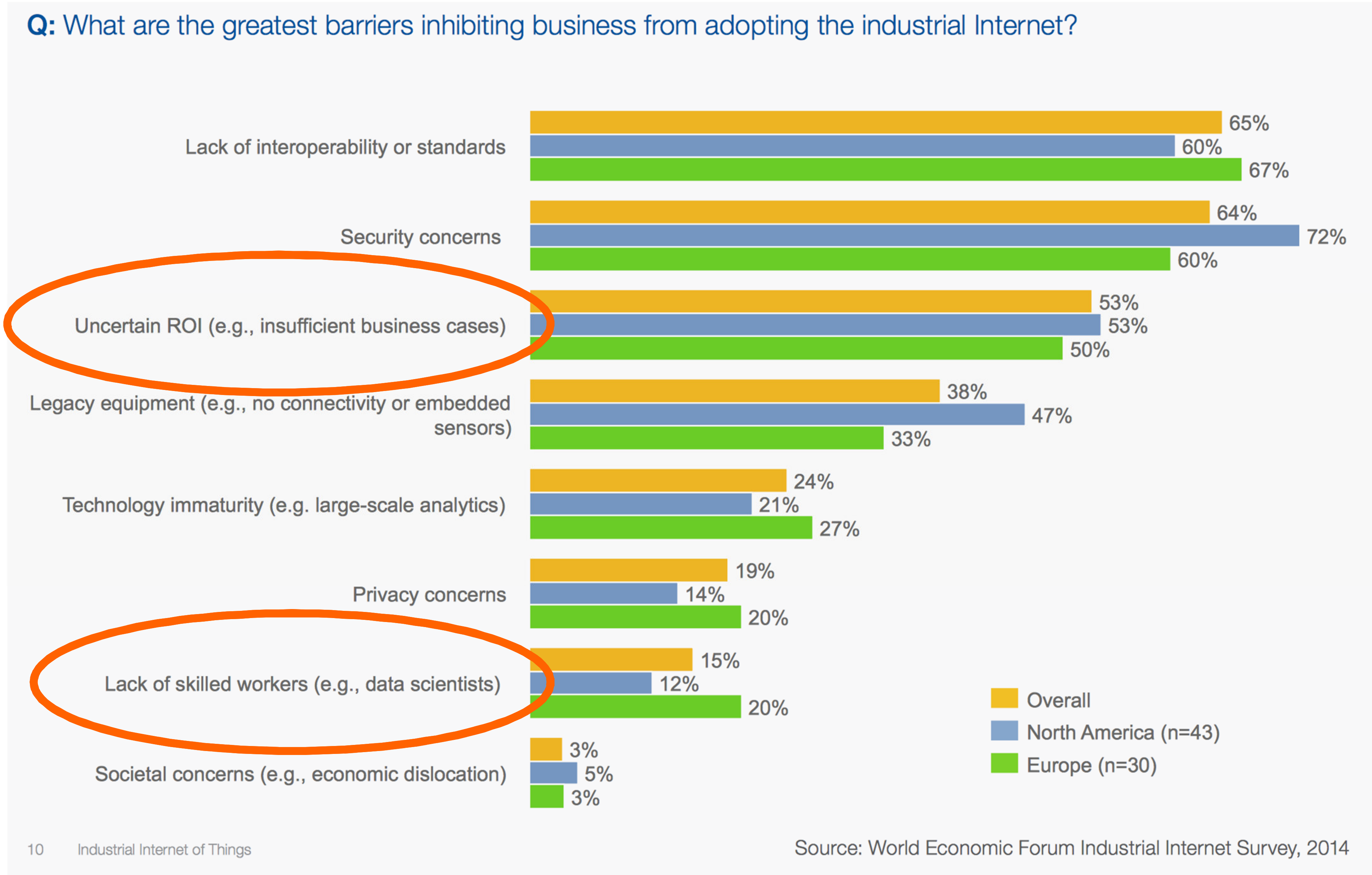


- Note the wide variety of communication infrastructures and services and massive number of nodes

IoT - Main Tasks

- Gather information from things and send commands to things
  - monitoring: state information
  - control: command enforcement
- Send information back and forth remote locations (private/public cloud)
- Store and aggregate information
- Analyze information to improve system knowledge
- Take decisions, in a human-assisted or autonomous manner

IoT - Barriers and Fears



Outline (1)

1. Introduction to IoT
   - definition, enabling technologies and concepts to better understand the general framework of IoT solutions
   - layering architecture, cloud computing vs. fog computing

2. Most relevant components of IoT solutions
   - devices, wireless communication protocols, data exchange protocols, IoT platforms, and data analysis

Outline (2)

3. Industry 4.0
   - introduction and definition
   - Industry 4.0 and the digital twin

4. Examples of IoT and Industry 4.0 solutions
   - from e-Maintenance to i-Maintenance
   - IoT-enabled use cases
   - experience with a business case

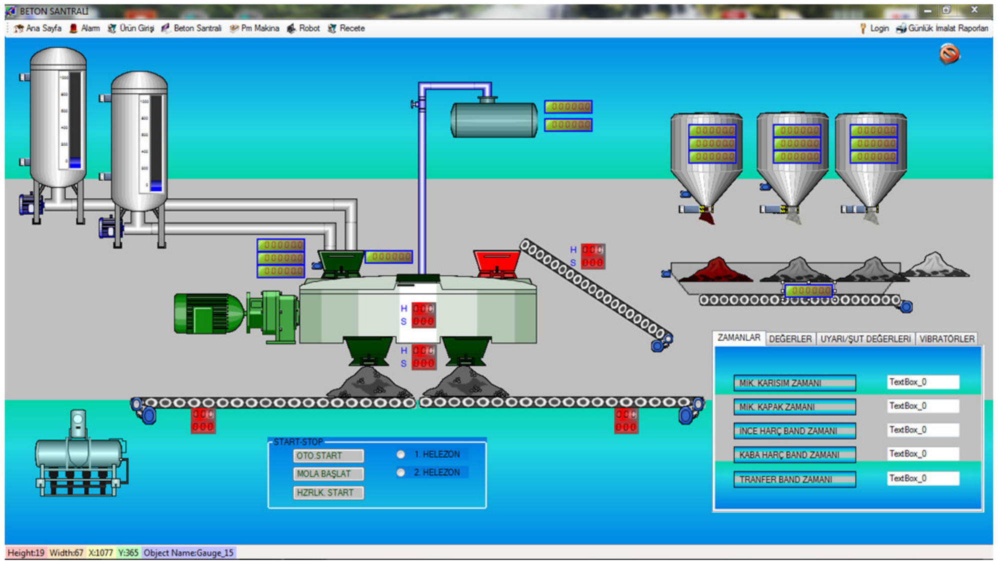
Outline

1. Introduction to IoT
   - definition, enabling technologies and concepts to better understand the general framework of IoT solutions
   - layering architecture, cloud computing vs. fog computing

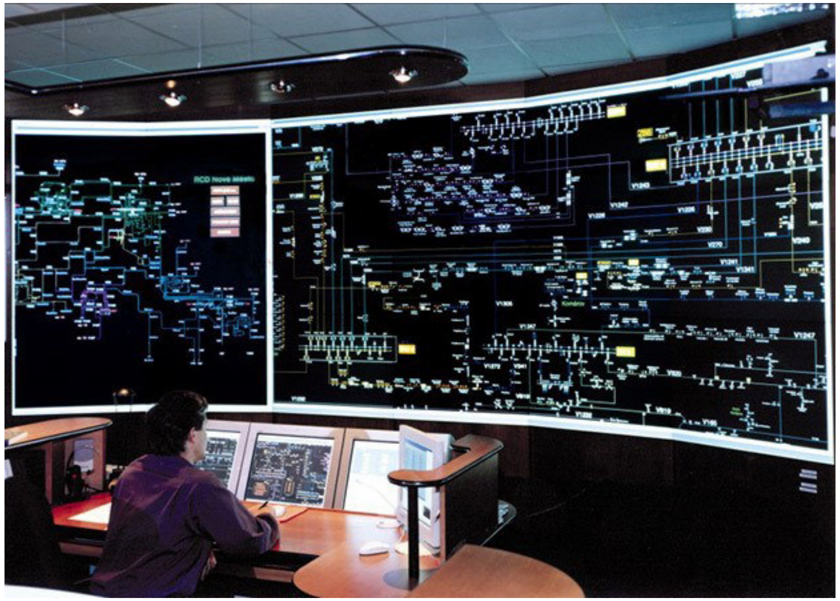
2. Most relevant components of IoT solutions
   - devices, wireless communication protocols, data exchange protocols, IoT platforms, and data analysis

SCADA - Supervisory Control And Data Acquisition

- Before IoT, remote monitoring and control already available for years
- Central control system with sensors/actuators, controllers, communication equipment, and software
- Periodic reading of values and status of sensors to require minimal intervention of human personnel
- Typical deployments: gas and oil distribution, electrical power, water distribution, bus traffic system, airport



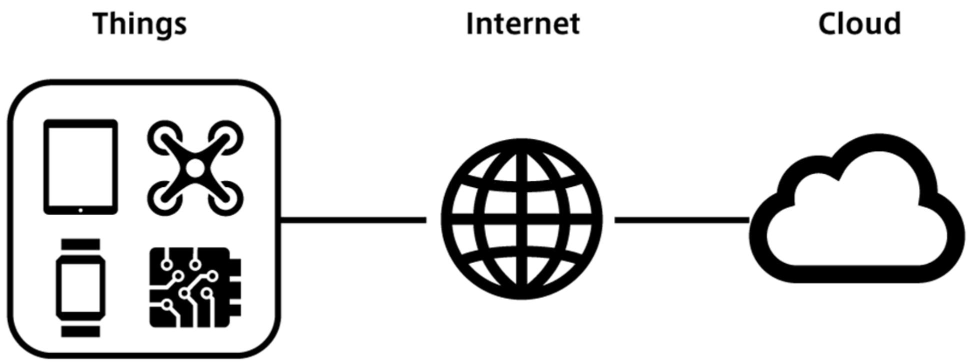
SCADA



- Pros
  – widely tested solutions
  – real-time and reliable
  – tailored to operation technicians

- Cons
  – expensive, but worth for mission critical tasks
  – extensible, but within the same scope, e.g., an additional sensor in the same machine or an additional machine in the same plant
  – lack of standards, difficult to maintain/evolve
  – usually no interoperability (horizontal/vertical)

IoT - Main Goals,
in a nutshell...



- IoT can be seen as a novel business approach
  - based on a set of (already available) technology tools
  - exploited in novel (industrial) scenarios
- The Internet (and its standards) exploited to transfer data about things in an *efficient*, *interoperable,* and *secure* manner

- Things can be
  - physical: Cyber Physical System (CPS) solutions to create a bridge between the physical and digital worlds
  - digital: information are already in binary format, but typically not in a standard format

- Gathered data analyzed to acquire a more in-depth knowledge of physical/digital systems, typically composed of geographically distributed and heterogeneous things

IoT Definition based on List of Features (1)

IEEE Internet of Things

- Interconnection of Things: it is a system that deals with the interconnection of "Things" (similar to M2M)
- Connection of Things to the Internet: "Things" are connected to the Internet, thus the system is not an Intranet or Extranet of Things
- Uniquely Identifiable Things: an IoT system is composed of things that are uniquely identifiable
- Ubiquity: a major feature of an IoT system, indicating a network which is available anywhere and anytime, i.e., where and when it is needed

IoT Definition based on List of Features (2)

- Sensing/Actuation capability: sensors/actuators are connected to the "Things" and perform the sensing/actuation which bring the smartness of the "Things"

- Embedded intelligence: smart and dynamic objects, with emergent behavior, embed intelligence and knowledge functions as tools and become an (external) extension to the human body and mind.

- Interoperable Communication Capability: the IoT system has a communication capability based on standard and interoperable communication protocols

IoT Definition based on List of Features (3)

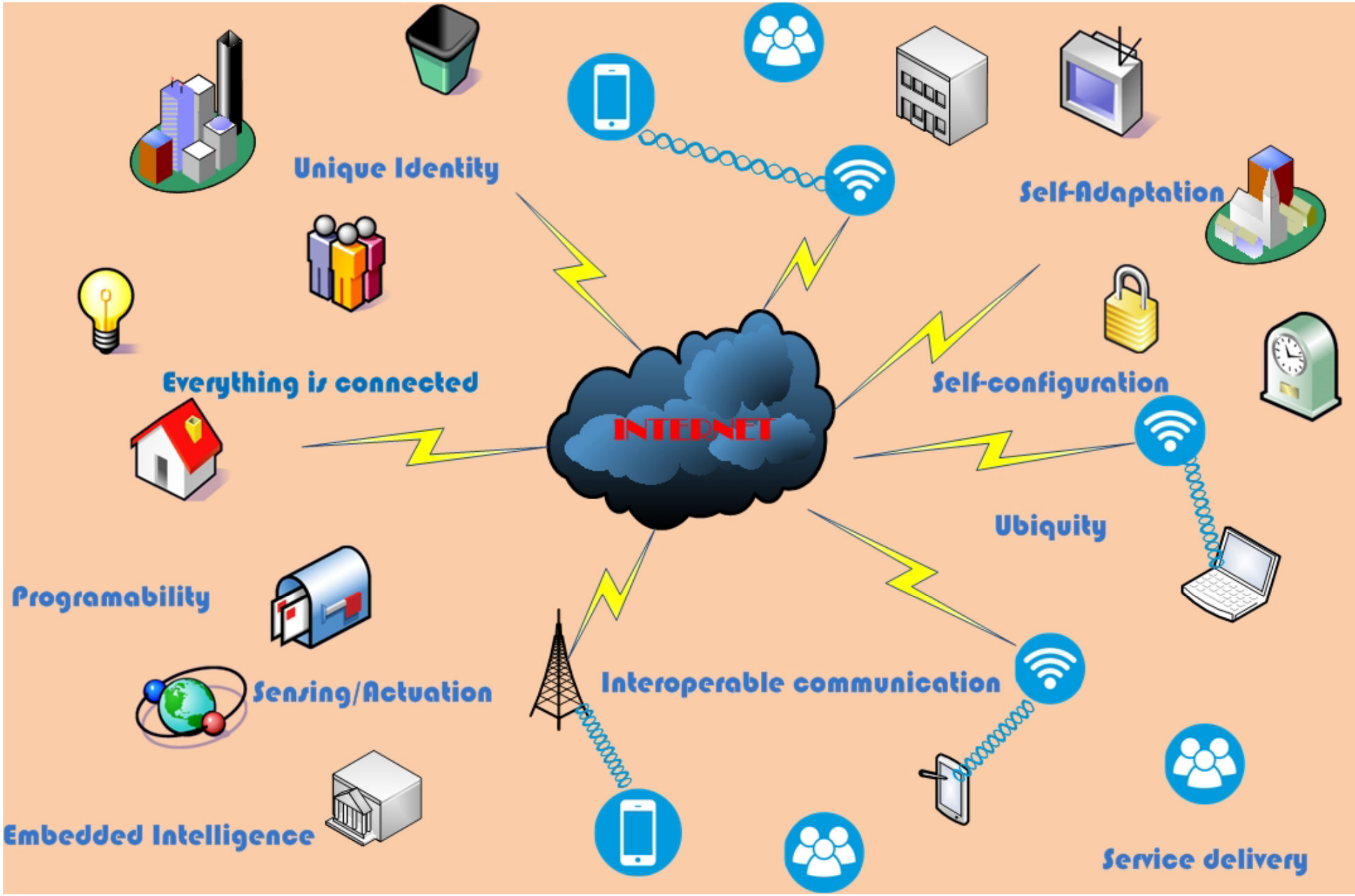
- Self-configurability: due to the amount and the heterogeneity of devices, the natural direction for IoT devices is to manage themselves, both in terms of their software/hardware configuration and their resource utilization (energy, communication bandwidth, medium access, etc.)
  - neighbor and service discovery, network organization and resource provisioning, etc.

- Programmability: at the simplest level, a programmable device is one that can take on a variety of behaviors at a user's command without requiring physical changes

IoT Definition –
Small Environment Scenario

- An IoT
  - is a network that
  - connects uniquely identifiable "Things"
  - to the Internet
- The Things
  - have sensing/actuation and
  - potential programmability capabilities
- Through the exploitation of unique identification and sensing
  - information about the Thing can be collected
  - and the state of the Thing can be changed
  - from anywhere, anytime, by anything

IoT Definition –
Large Environment Scenario (1)



IoT Definition – Large Environment Scenario (2)

- Internet of Things envisions a self-configuring, adaptive, complex network that interconnects Things to the Internet through the use of standard communication protocols
- The interconnected things have physical or virtual representation in the digital world, sensing/actuation capability, a programmability feature and are uniquely identifiable
- The representation contains information including the thing's identity, status, location or any other business, social or privately relevant information
- The things offer services, with or without human intervention, through the exploitation of unique identification, data capture and communication, and actuation capability
- The service is exploited through the use of intelligent interfaces and is made available anywhere, anytime, and for anything, taking security into consideration

IoT Enabling Technologies

- Reduced **hardware** cost and size
  - from special-purpose to Commercial Off-The-Shelf (COTS)
- Pervasive and cheap **wireless communication**
  - from cables to large-bandwidth and/or wide-coverage wireless communication

actual game changers, IMHO

- Consolidated and emerging **Web-based communication**
  - from close protocols to open standards, also applied in constrained devices
- **Standards**, to achieve interoperability
  - e.g., communication standards and data representation
- General purpose **horizontal solutions**
  - from SCADA to IoT platforms
- Automatic tools to **infer knowledge**
  - wide application of AI (Artificial Intelligence) techniques

IoT: Horizontal Integration vs. Vertical Market

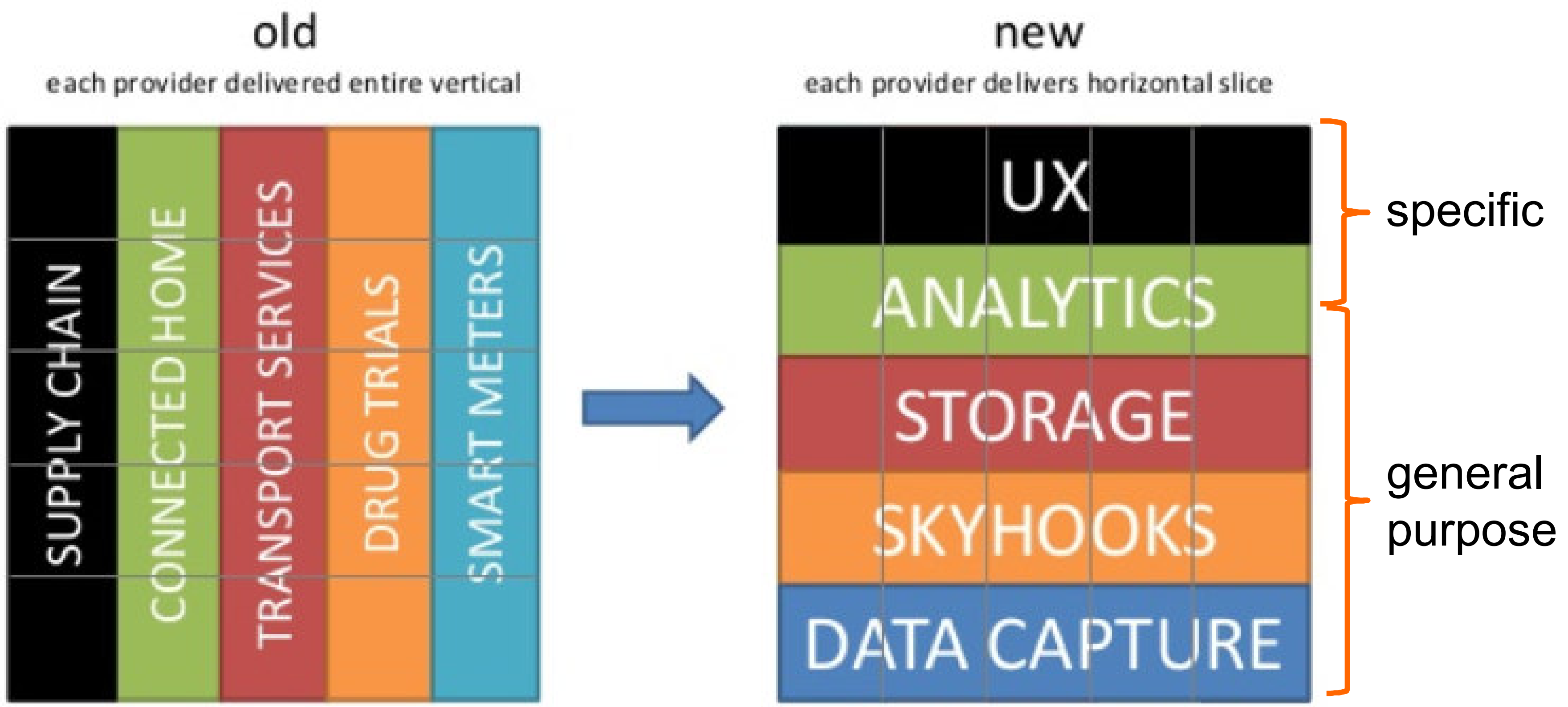
- One horizontal layer/platform managing heterogeneous information in an efficient manner
- Several vertical applications to provide specific information in a market-tailored manner

Horizontal General-purpose Layer

- Main building blocks are agnostic in relation to the vertical market
  - can contain data of different markets at the same time
  - enables interaction of different environments
- Same horizontal solution reused in several heterogeneous projects
  - reduced cost and time-to-market
  - widely tested components, thus increased reliability and security

Vertical Per-market Specializations

- Graphical User Interface (GUI) tailored to the vertical market
- Can apply domain-specific analytical tools
- Industry-specific expertise to provide added value



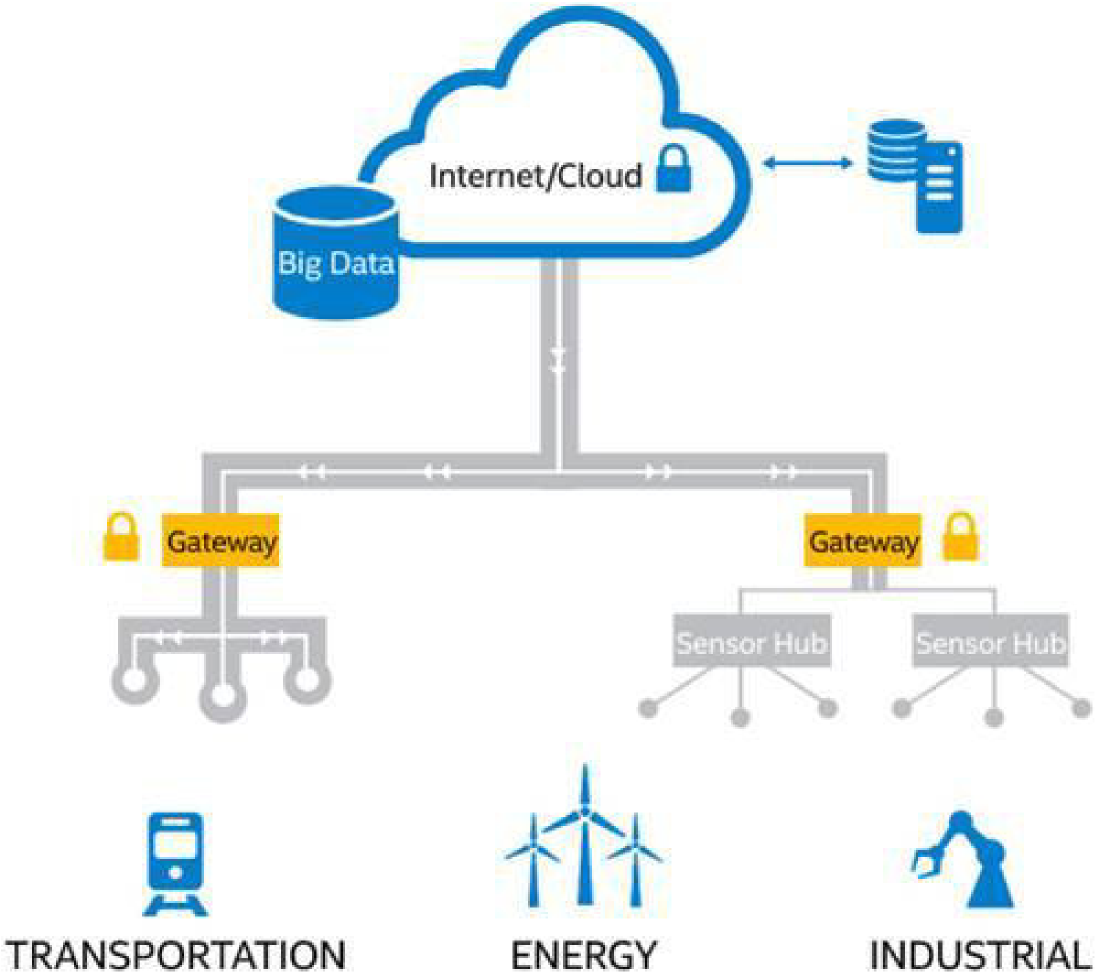
Outline

1. Introduction to IoT
   - definition, enabling technologies and concepts to better understand the general framework of IoT solutions
   - layering architecture, cloud computing vs. fog computing

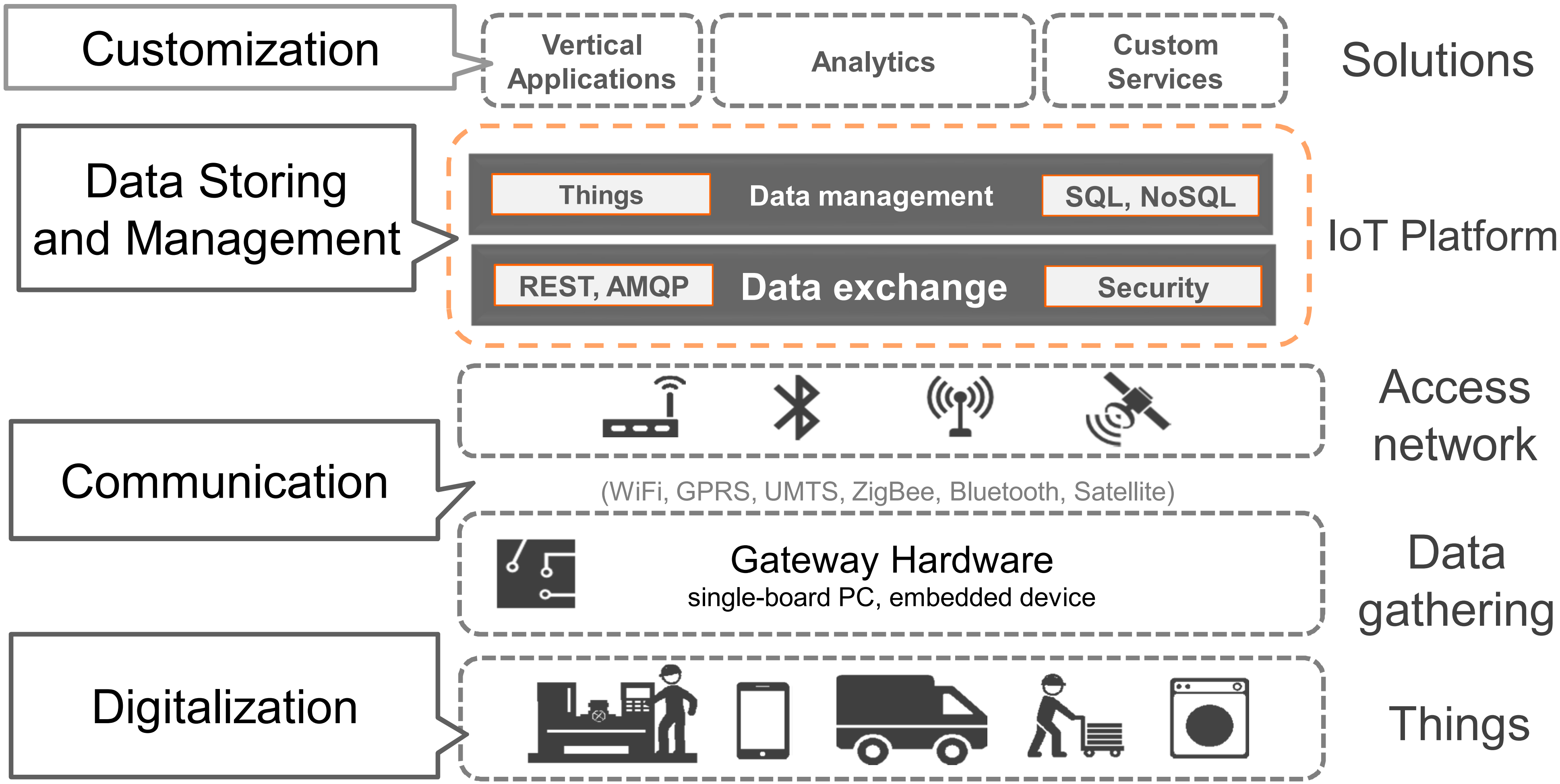
2. Most relevant components of IoT solutions
   - devices, wireless communication protocols, data exchange protocols, IoT platforms, and data analysis

Typical Cloud-based IoT Architecture

- Several heterogeneous things, e.g., sensors and actuators
- Multiple gateways geographically close to sensors/actuators
  - directly interact with things
  - dispatch data to/from the Internet
- Server-side remote applications stored in the Cloud and managing data



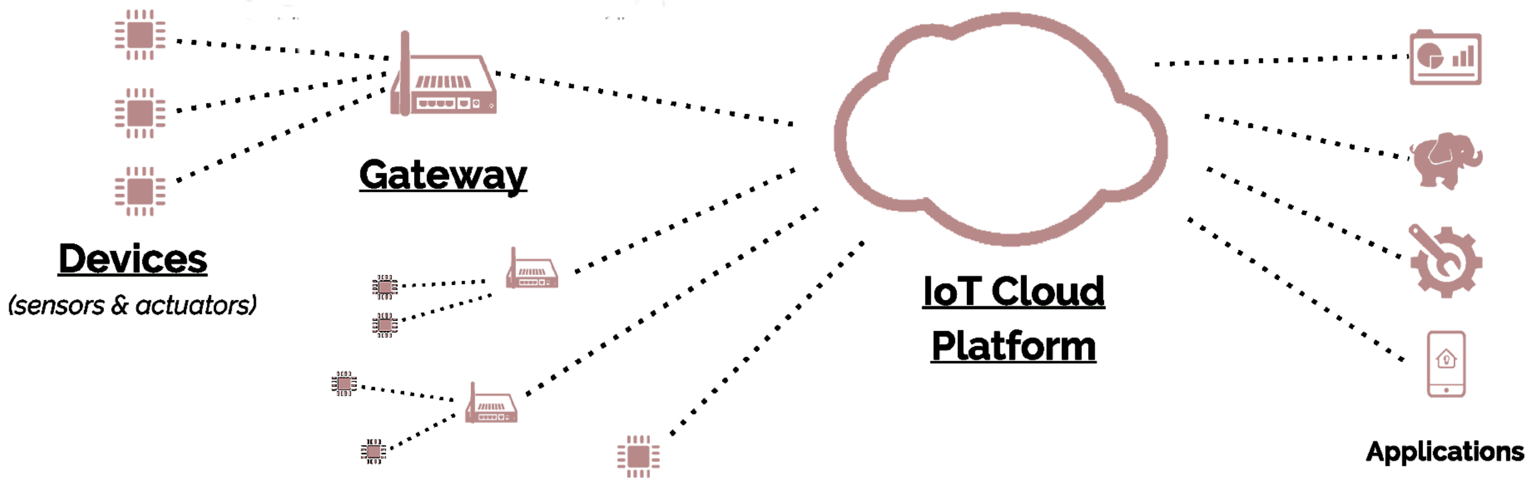
IoT: One Solution, Many Layers



Layering to Simplify Complexity

- Things: any physical or digital objects that should be monitored or controlled
  - physical objects must be digitalized
  - virtual objects must be standardized
- Gateway: close to one or multiple things to interact with them and send data and command back and forth the Internet
- Communication protocol: wired/wireless technology to actually send bytes
- Data exchange protocol: software protocol to standardize how information are transported (and eventually also represented)
- IoT Platform: data storing and management, application of (simple) aggregation/processing functions on data
- Analytics: complex analysis on data to infer new knowledge

Gateway



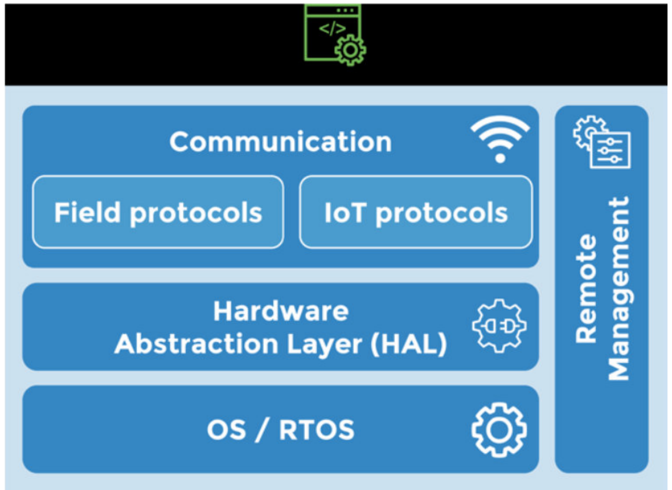
- **Protocol translation** between peripheral trunks of the IoT, eventually provided with lower parts of the communication stacks
- Gateways can also provide: pre-processing, **security**, scalability, service discovery, **geo-localization**, **billing**, etc.

- Pre-processing:
  - data **buffering**: temporarily store data to wait for connectivity or to increase efficiency
  - data **efficiency**: temperature read every 1s, but only per-minute average sent
  - data **aggregation**: water level from different silos, but only the sum is sent
  - data **filtering**: send temperature values only if grater than 25°C

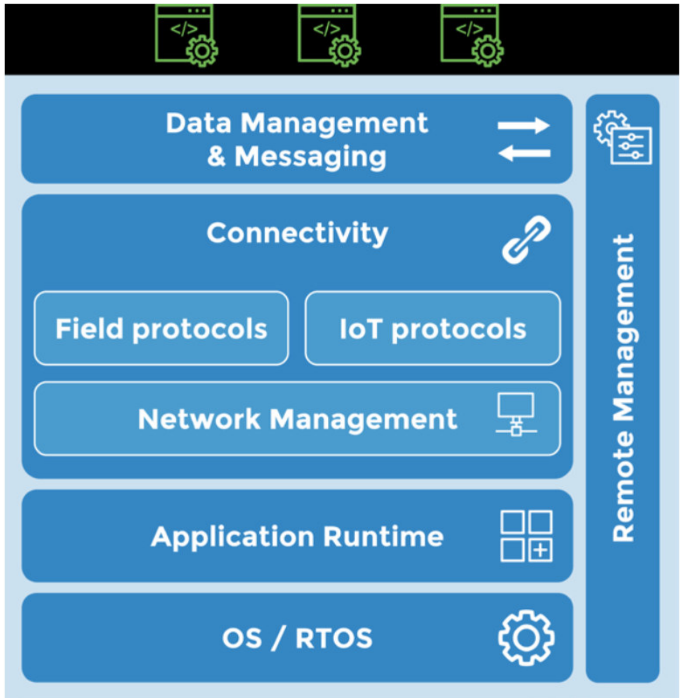
With/Without Gateway

- If there is no gateway, things have to send/receive data on their own
- In case of constrained devices, reduced set of capabilities, e.g.,
  - no security since cryptography is CPU-intensive
  - no data buffering, filtering aggregation
  - no programmability

Stack for constrained devices



Stack for gateways

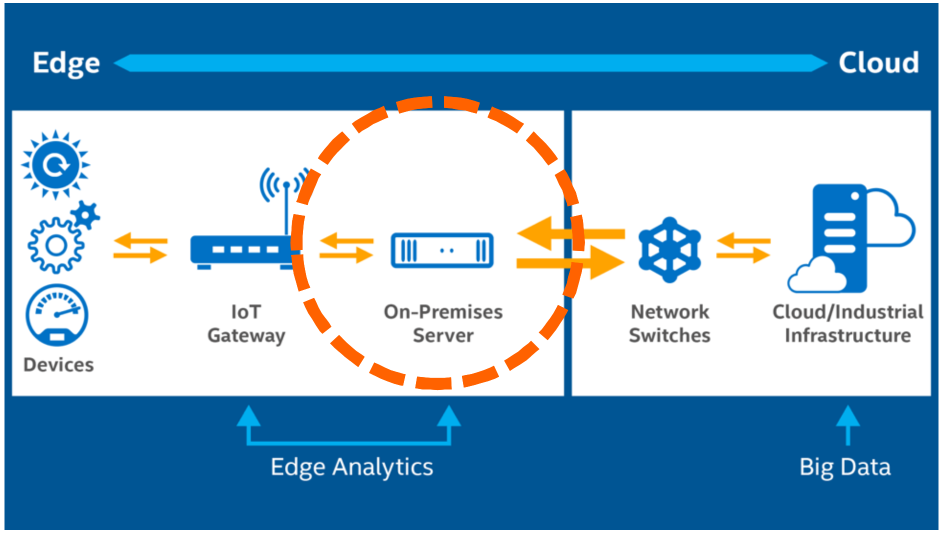


Cloud Computing: few words

- Primary concepts
  - IT on demand pricing
  - best benefits in a reliable context
  - pool of virtualized computer resourc
  - rapid live providing while demanding
  - systems on scaling architecture

- What is a Cloud
  - one Cloud is capable of providing IT resources “as a service”
  - one Cloud is an IT service delivered to users that have:
    - reduced incremental management costs when additional IT resources are added
    - services oriented management architecture
    - massive scalability

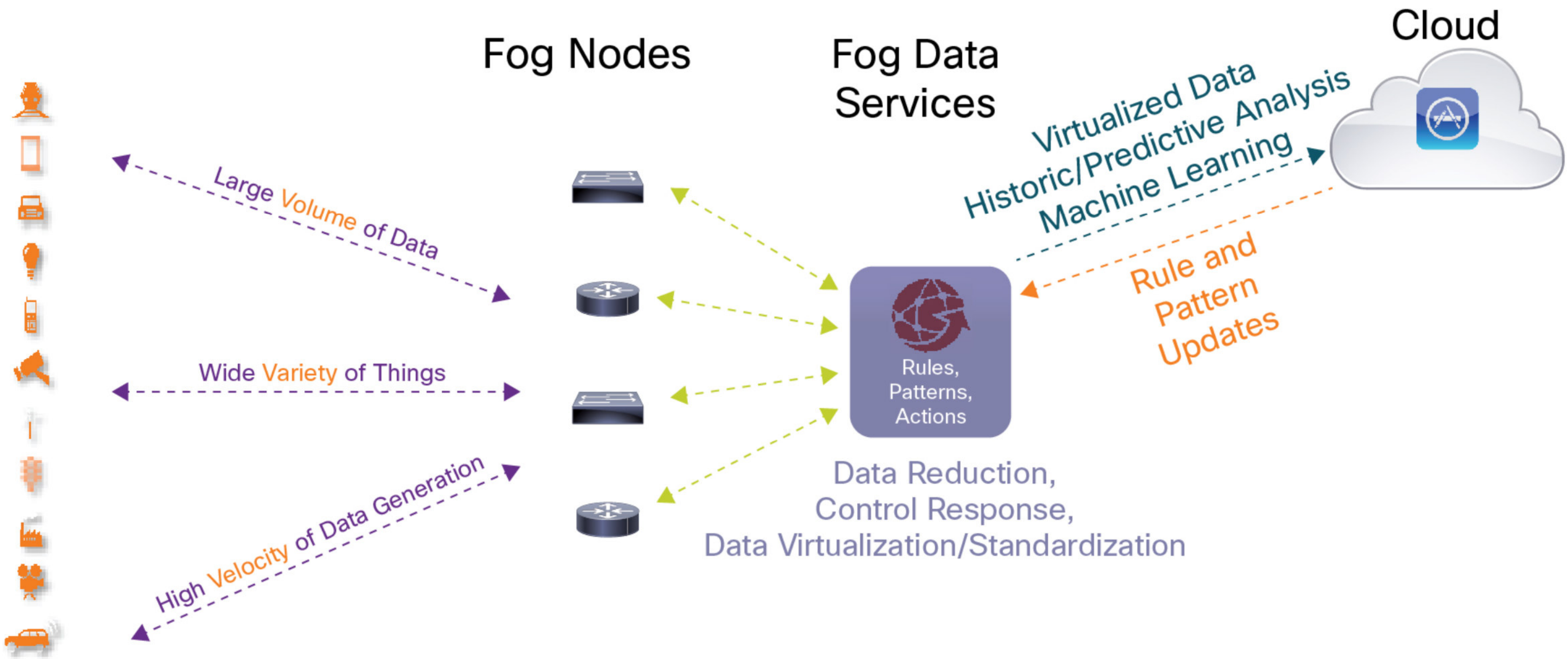
IoT: From Cloud Computing to Fog/Edge Computing



- IoT Cloud Computing architecture
  - most of the computation on the Cloud
  - only gateways are deployed close to things
  - gateways perform few and simple tasks

- IoT Fog/Edge Computing architecture
  - additional relatively powerful devices
  - close to things, but between gateways and the Cloud
  - complex analytical tasks on the client-side, before sending data to the Cloud

Fog Computing



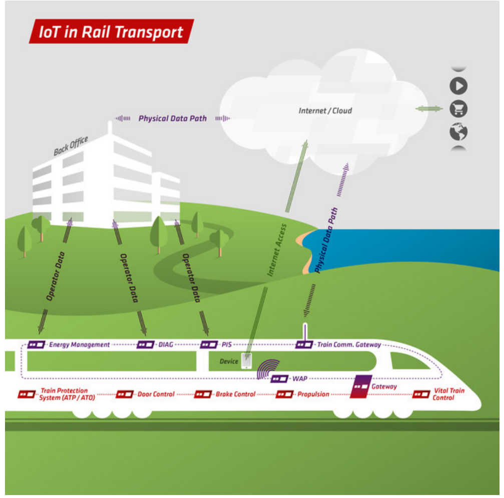
- Cisco: the fog extends the cloud to be closer to the things that produce and act on IoT data

Fog Computing for IoT

- Cloud models are not designed for the volume, variety, and velocity of data that the IoT generates
- Fog Computing allows to
  - minimize latency
  - conserve network bandwidth
  - address security concerns in transit and at rest
  - move data to the best place for processing

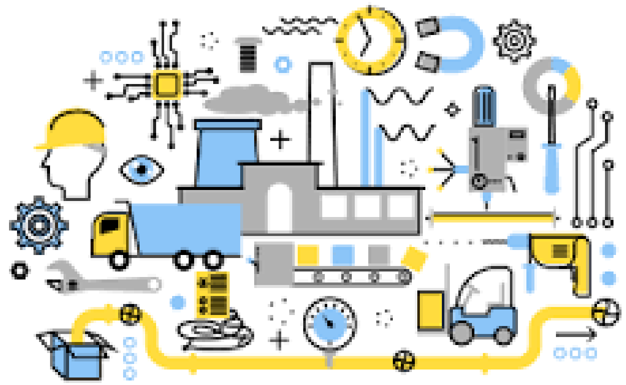
- When to consider Fog Computing
  - data is collected at the extreme edge: vehicles, ships, factory floors, roadways, railways, etc.
  - thousands or millions of things across a large geographic area are generating data
  - it is necessary to analyze and act on data promptly, in less than a second

Fog Computing for IoT use case: Rails



- Improve passenger safety
  - analyze and correlate data from cameras on the trains and at stations
  - monitor sensors on wheels and brakes to determine when parts need service before failure causes an accident
- Prevent cybersecurity attacks
  - take automated actions such as suspending operations or transferring control to a failover system
- Alert drivers to treacherous conditions ahead
  - Fog nodes gather sensor data on tracks and trains to detect unsafe conditions

Fog Computing for IoT use case: Manufacturing



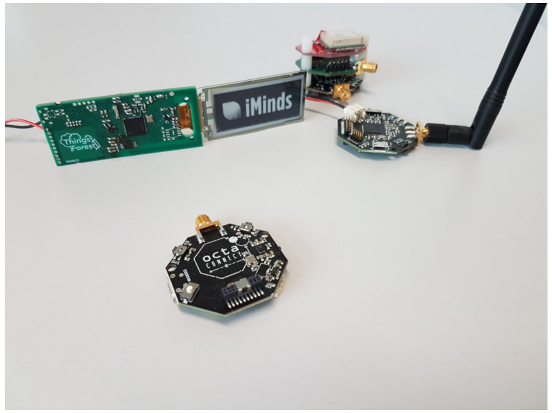
- Increase agility
  - quickly change production lines and introduce new products
- Reduce downtime
  - predictive maintenance to avoid costly equipment downtime
  - Fog nodes collect machine data and report early signs of problems
- Continually confirm that safety systems are intact
  - analyze machine data in real-time
  - promptly shut down compromised equipment automatically, without waiting for a human to respond to an alert

Outline

1. Introduction to IoT
   - definition, enabling technologies and concepts to better understand the general framework of IoT solutions
   - layering architecture, cloud computing vs. fog computing

2. Most relevant components of IoT solutions
   - devices, wireless communication protocols, data exchange protocols, IoT platforms, and data analysis

Constrained Devices



- “A node where some of the characteristics that are otherwise pretty much taken for granted for Internet nodes at the time of writing are not attainable, […] due to cost, size, and energy constraints”

- Significant constraints on:
  - maximum code complexity (ROM/Flash)
  - size of state and buffers (RAM)
  - available computational power
  - connectivity

Constrained Networks

- "A network where some of the characteristics pretty much taken for granted with link layers in common use in the Internet at the time of writing are not attainable"
- Significant constraints on:
  - low achievable throughput
  - high packet loss
  - highly asymmetric links
  - severe penalties for using larger packets
  - limits on reachability over time

Classes of Constrained Nodes

Name	Data size (RAM)	Code size (Flash)
Class 0	<< 10 KiB	<< 100 KiB
Class 1	~ 10 KiB	~ 100 KiB
Class 2	~ 50 KiB	~ 250 KiB

Values in 2014

- C0 Devices
  - no direct secure Internet connection
  - use larger devices as gateways/proxies
  - preconfigured and rarely reconfigured
- C1 Devices
  - can use environment specific protocols, e.g., CoAP
  - no access to standard Internet protocols, e.g., HTTP, TLS
  - can be integrated into an IP network
- C2 Devices
  - can use most of protocols

Constrained Devices: Class 0

- Very constrained sensor-like
- They need the help of larger devices acting as proxies, gateways, or servers to participate in Internet communications
- They cannot be secured or managed in a traditional way and are most likely preconfigured

- In other words, they cannot be used without a gateway

Constrained Devices: Class 1

- Cannot easily talk to other Internet nodes employing a full stack, e.g., no HTTP, TLS, and XML
- However, they are capable enough to
  - use a protocol stack specifically designed for constrained nodes (CoAP over UDP)
  - participate in conversations without the help of a gateway node
- Class 1 devices can provide support for the security functions required on a large network

- They still need to be parsimonious with state memory, code space, and often power expenditure for protocol and application usage

Constrained Devices: Class 2

- Less constrained and fundamentally capable of supporting most of the same protocol stacks as used on notebooks or servers

- However, even class 2 devices can benefit from lightweight and energy-efficient protocols and from consuming less bandwidth, e.g., using the protocol stacks defined for more constrained devices

Limitations based on energy constrains

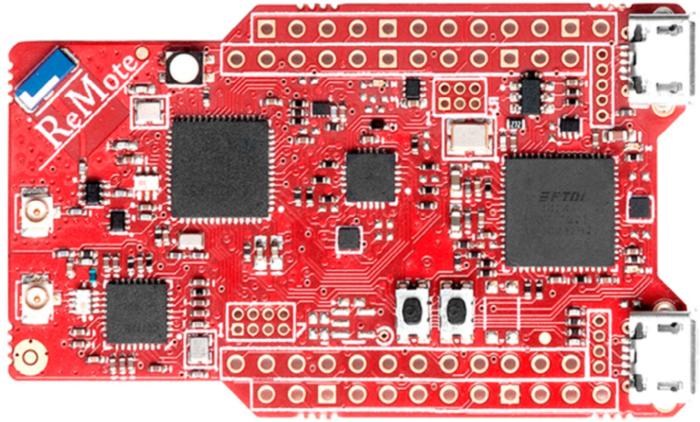
- Devices are classified also based on their energy capabilities

Name	Type of limitation	Example power source
E0	Event energy-limited	Event-based harvesting
E1	Period energy-limited	Battery periodically recharged/replaced
E2	Lifetime energy-limited	Non-replaceable primary battery
E9	No energy limitation	Mains-powered

IoT Device Example: TelosB

- Legacy example of IoT device
  - TelosB Mote for Wireless Sensors Networks
  - 8 MHz 16-bit TI MSP430
  - 10kB RAM, 16kB ROM, 1MB EEPROM
  - IEEE 802.15.4 2.4 GHz radio with antenna
  - temperature, humidity, and light sensors

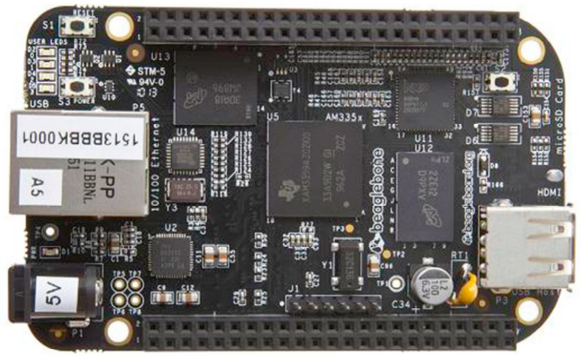
IoT Device Example: Zolertia Re



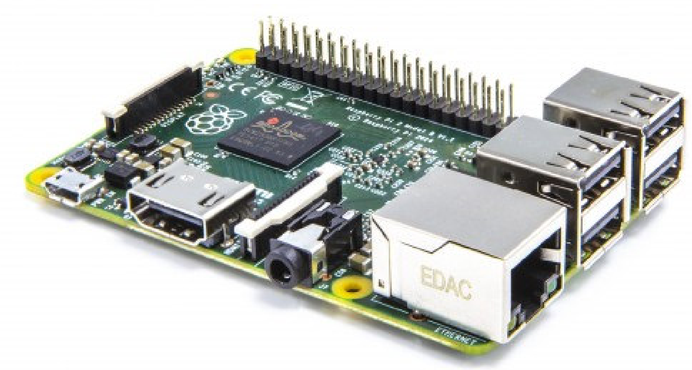
- Newer generation board
  - **double radio interface** to enable short (100 meters) and long range (up to 40 Kms) communication
  - powerful 32Mhz ARM Cortex-M3 (32 KiB RAM, 512KiB ROM)
  - **low power** operation down to 0.4uA to operate on batteries for years
  - **several sensing capabilities** (via additional hardware)

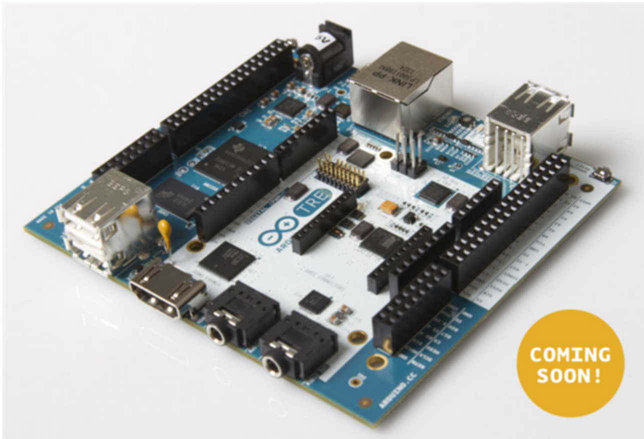
Beyond Class 2: Single Board Computers

- In the last 5/10 years
  - tremendous improvement in CPU/memory capabilities
  - dramatically reduced costs



- Modern Single-Board Computers (SBCs) are very cheap (~100$) and powerful
  - can host complete operating systems
  - extensions via daughterboards
  - mostly designed to be mains-powered





The Evolution of Constrained Devices

- Phase 1: Smart Tags to uniquely identify physical things
  - textual identifiers: bar code, QR code
  - digital identifiers: passive/active RFID, i.e., small chips transmitting their unique identification number via wireless at small range
- Phase 2: Automated Sensing to transmit sensed data to remote devices
  - periodic or on-demand transmission
  - read-only and battery-powered
- Phase 3: Smart Devices allowing to develop modern complex IoT applications
  - sensors and actuators
  - gateways and fog nodes

Outline

1. Introduction to IoT
   - definition, enabling technologies and concepts to better understand the general framework of IoT solutions
   - layering architecture, cloud computing vs. fog computing

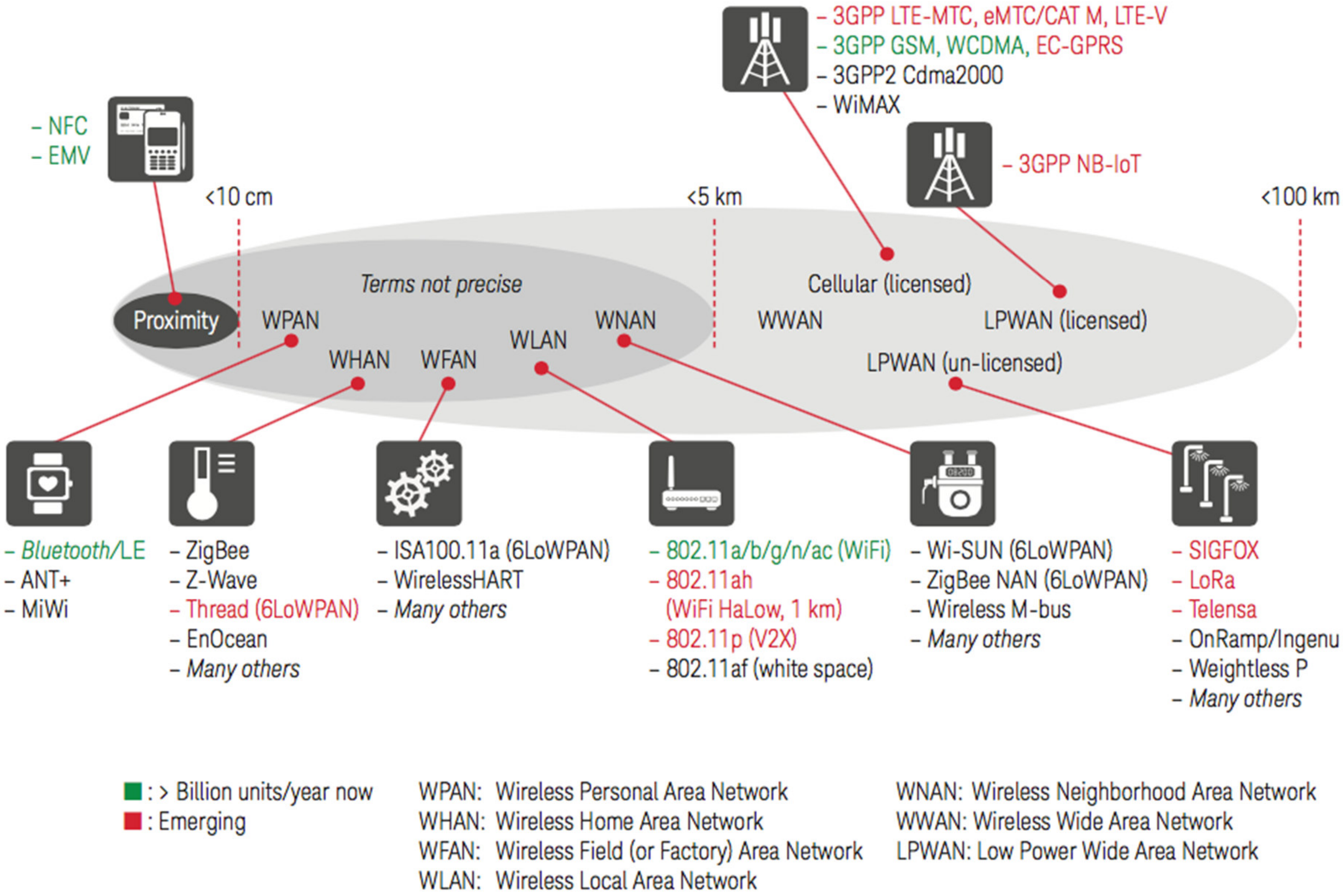
2. Most relevant components of IoT solutions
   - devices, wireless communication protocols, data exchange protocols, IoT platforms, and data analysis

Wireless Communication Protocols for the IoT

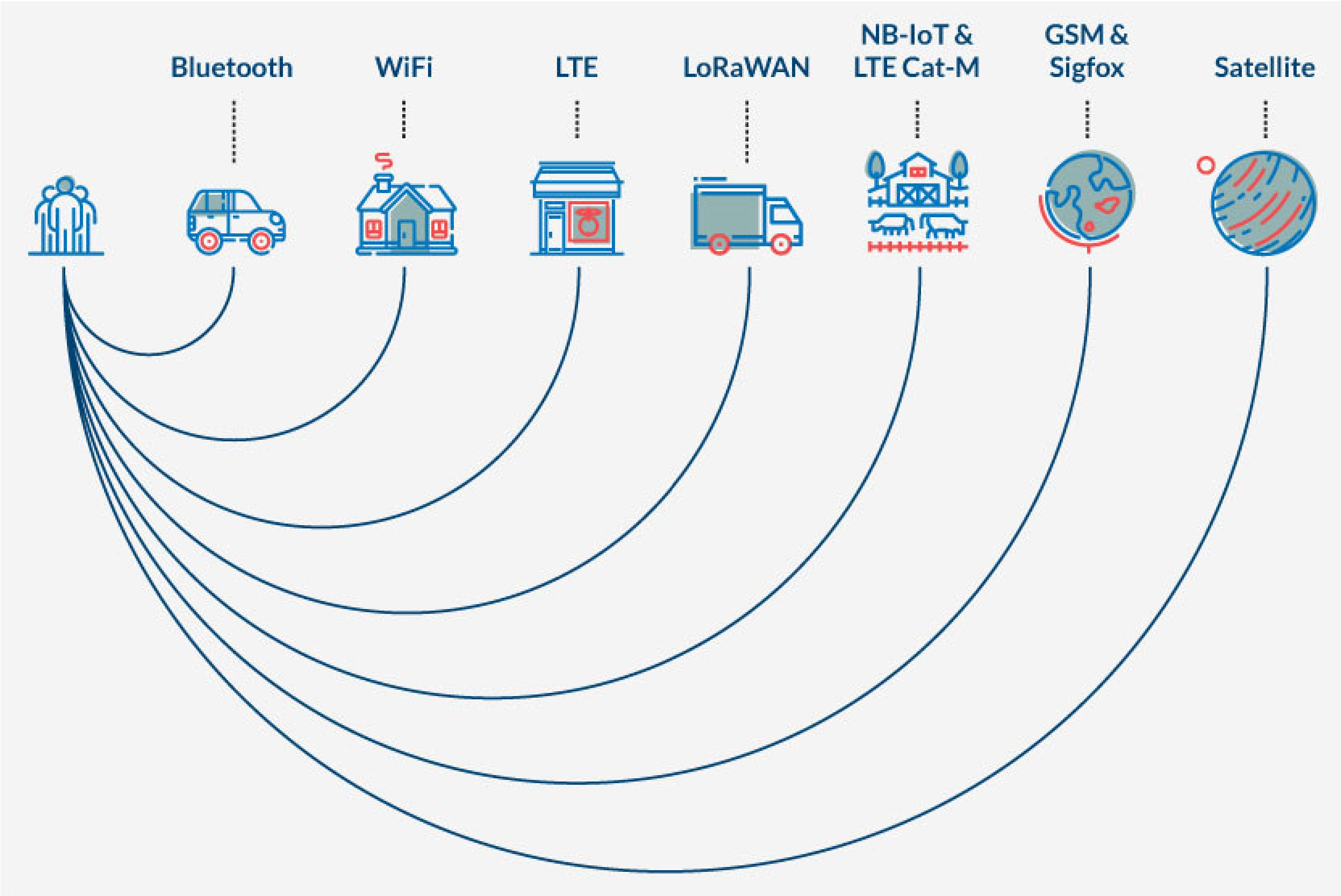
- The capability of connecting to things in a seamless, ubiquitous, and cheap manner have pushed the spread of IoT solutions

- IoT wireless communication protocols primarily differs in relation to
  - coverage range: from few cm to several km
  - power consumption: from few mW to several W
  - bandwidth: from few bytes per day to hundreds of MB/s
  - security: from plain data to strong encryption
  - cost: greatly varying, both for equipment and data transmission

Several Wireless Communication Protocols



Wireless Protocols per Coverage Range



Bluetooth

- Bluetooth and BLE are radio protocols for Personal Area Networks (PAN)
- Mostly these are on a person's body or in close proximity to them

- Typical range: very short, 20m (or less)
- Max output power: very limited, 0.003 W
- Bandwidth: limited, 0.7–2.1 Mbit/s
- Security: pairing task to exchange encryption keys
- Cost: cheap equipment, no transmission costs

- Good for: devices that stay in close proximity of each other, like between a smartphone and a headset, heart rate monitor, bicycle speedometer

WiFi - IEEE 802.11

- WiFi is meant for broadband network connections in a confined area
- Normally less than 100 square meter per access point

- Typical range: short, 50m
- Max output power: medium/high, 0.1 W
- Bandwidth: large, up to hundreds of MB/s
- Security: suffered recent exploits, work in progress
- Cost: cheap equipment, no transmission costs

- Good for: security cameras, power meters or anything that is installed in a fixed location, has power, and needs bandwidth

LTE, the 4th generation mobile network system

- LTE supports voice calls, but designed for scalability and wireless broadband
- Range normally less than GSM, but data rate is orders of magnitude greater

- Typical range: long, 2Km
- Max output power: medium/high, 0.2 W
- Bandwidth: large, up to hundreds of MB/s
- Security: multiple unique identifiers, static keys, and encryption methods
- Cost: depends on the telco operator

- Good for: broadband wireless internet connection, also for streaming security camera videos

LoRaWAN

- Low speed, but long range and low power communication protocol
- Open specification so anyone is free to implement the protocol

- Typical range: long, 5-10 km
- Max output power: low, 0.025 W
- Bandwidth: very low, between 290 bps and 50 kbps
- Security: protocol to exchange to set unique set of AES keys
- Cost: cheap client equipment, needs access point or service provider

- Good for: isolated or private network on a farm or in a city, ideal for sensors that only seldomly send a value, like a soil moisture sensor sending its measurements every 10 minutes or a water trough alarming that it is empty

SigFox

- Low speed and low power, but also long range
- Meant for remote meter reading, also used for any remote data uplink
- Proprietary network and protocol

- Typical range: long, 24 km, global (partial) coverage https://www.sigfox.com/en/coverage
- Max output power: low, 0.025 W
- Bandwidth: very low, up to 140 messages per object per day, 12 bytes payload, 100 bps
- Security: no encryption
- Cost: cheap client equipment, for transmission about 1/12€ per device per year

- Good for: remote electricity or water meter reading, mostly useful if you do not need to send data to the device

Other Wireless Protocols

- NB-IoT and LTE Cat-M
  - use cellular channels for long range, limited bandwidth communication
  - still in development or with limited coverage
- Satellite
  - huge coverage
  - extremely expensive

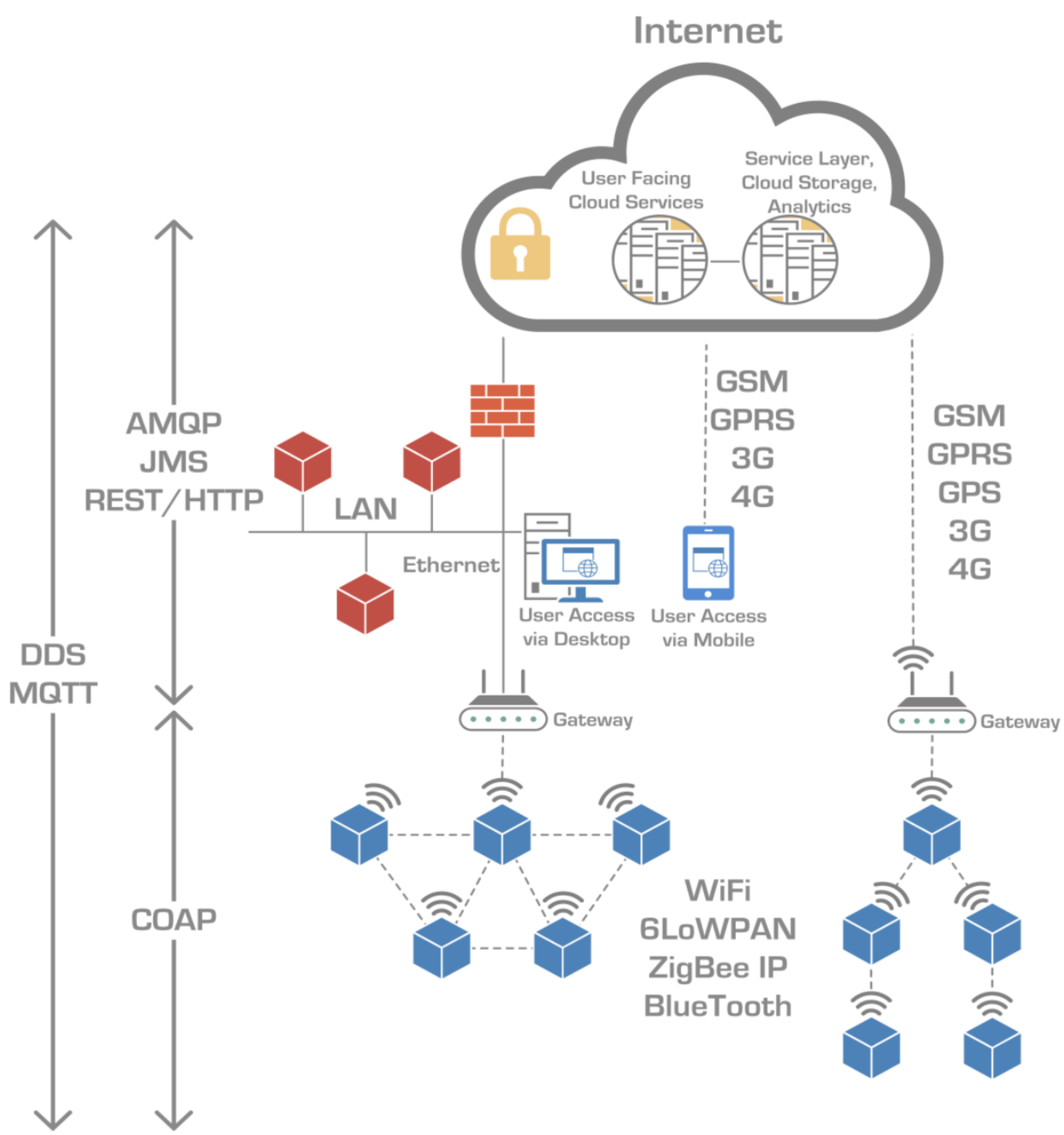
Outline

1. Introduction to IoT
   - definition, enabling technologies and concepts to better understand the general framework of IoT solutions
   - layering architecture, cloud computing vs. fog computing

2. Most relevant components of IoT solutions
   - devices, wireless communication protocols, data exchange protocols, IoT platforms, and data analysis

IoT Connectivity Problem Space

- Wired/wireless protocols to transport bits
- Messaging protocols to transfer data
- Data: information and commands described following a given syntax and semantic
- Messaging protocols to exchange data: encapsulate data and transmit it via wired/wireless protocols



Data Exchange Protocols

- How to encapsulate information and commands in messages?
- Data from gateways to servers: who should initiate the communication?
  - push: gateways autonomously decide to send messages to servers, e.g., when new sensed values are available
  - pull: servers ask to gateways to send messages, e.g., only when servers actually require sensed values

- Two primary models
  - request/response, can be push or pull
  - publish/subscribe, usually only push

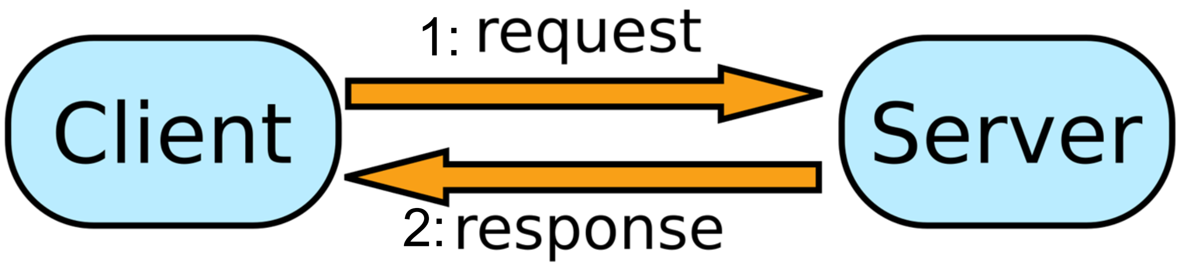
- Several protocols:
  - request/response: REST/HTTP, CoAP, …
  - publish/subscribe: MQTT, AMQP, DDS, …

Multiple Protocols supporting Different Features

- Several protocols thriving to become a de facto standard
- It could be difficult to pick up the best solution, since it depends on requirements of a given application domain

- Some request/response protocols:
  - REST architectural principles (REpresentational State Transfer), 2000
  - CoAP (Constrained Application Protocol), 2014 (also pub/sub)
- Some publish/subscribe protocols:
  - MQTT (Message Queue Telemetry Transport), 1999
  - AMQP (Advanced Message Queuing Protocol), 2003
  - DDS (Data Distribution Service), v1.0 2003; v1.2 2007
- At different level of abstractions, LonWorks (1999), Modbus (1979)…

Request/Response model



- A client application requests services (e.g., send data, require data, perform an addition) and a server application responds to the service request (e.g., by providing the data or the addition results)
- Direct communication between client and server
- Data exchange only if the client starts the communication
  - the server is not able to contact the client autonomously
- Typical interactions in the IoT scenario:
  - push: sensors send data to servers
  - pull: actuators request to servers new configurations

REST - REpresentational State Transfer

- REST is not an actual protocol, but substantially a solution architectural style
- Promote client/server and stateless interaction, oriented to the usage of caching opportunities, also with possibility of code-on-demand to clients
- Usually based on HTTP, the protocol used to surf the Web
- Very simple, but each time the client has to start the communication from the beginning

REST: Identifying and Interacting with Resources

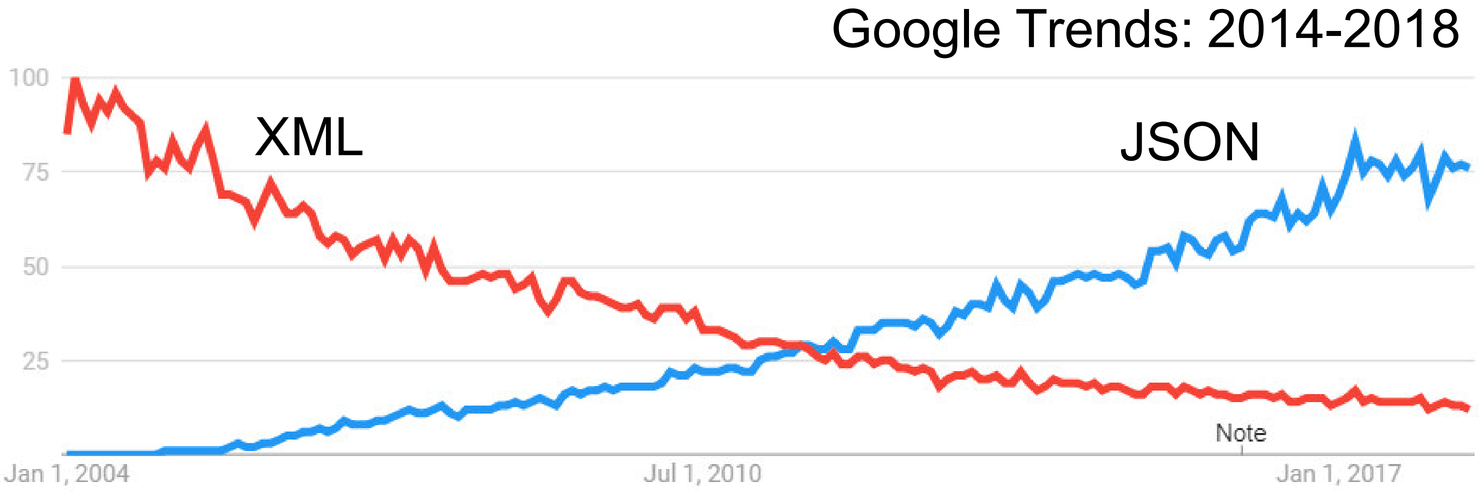
- URI (Uniform Resource Identifier) to identify the remote resource
  - any resource has a persistent identifier
  - do not transfer resources but their representations via HTTP
  - endpoint for a service managing user information: www.examples.com/resources/users
- HTTP method to interact with remote resources in a standard manner
  - GET, retrieve a specific resource (by id) or a collection of resources
  - PUT, create a new resource
  - POST, update a specific resource (by id)
  - DELETE, remove a specific resource by id

REST: Endpoint Examples

- Remote resource: a software component managing users

HTTP Method	URI	Operation
GET	www.examples.com/resources/**users**	Get list of users
GET	www.examples.com/resources/**users/1**	Get the user with id 1
DELETE	www.examples.com/resources/**users/1**	Delete the user with id 1
POST	www.examples.com/resources/**users/2**	Update the user with id 2
PUT	www.examples.com/resources/**users/1**	Insert the user with id 1

REST: Formatting data



- JavaScript Object Notation (JSON) most spread syntax for formatting/serializing data, e.g., objects, arrays, numbers, strings, booleans, and null
- Resource: a group of users, each one with a first name and a last name

```
{“users":
   [
      { "firstName":"John", "lastName":"Doe" },
      { "firstName":"Anna", "lastName":"Smith" },
      { "firstName":"Peter", "lastName":"Jones" }
   ]
}
```

REST: AWS IoT example

- Adds a thing to a thing group

```
PUT /thing-groups/addThingToThingGroup
HTTP/1.1
Content-type: application/json

{
   "thingArn": "string",
   "thingGroupArn": "string",
   "thingGroupName": "string",
   "thingName": "string"
}
```

- Manage “shadows” of things

```
POST endpoint/things/thingName/shadow
HTTP/1.1
Content-type: application/json

{
   "state":
   {
     "desired" :
      {
         "color" : { "r" : 10 },
         "engine" : "ON"
      }
   }
}
```

CoAP - Constrained Application Protocol

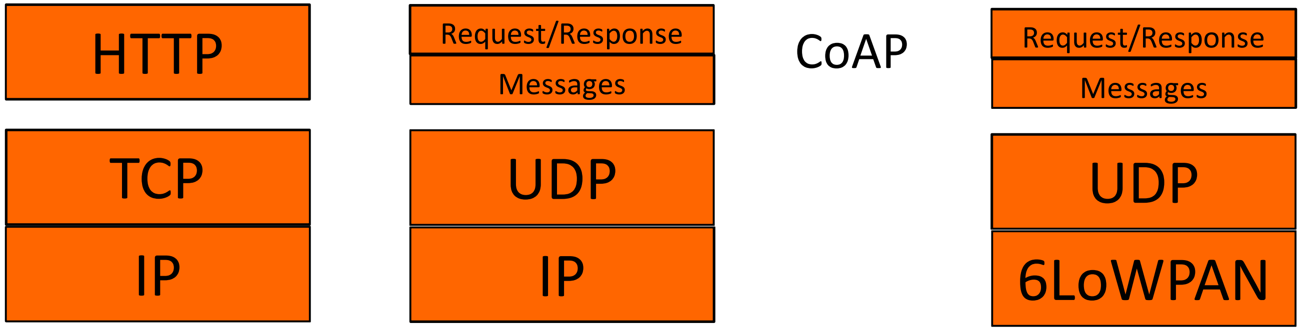
- Designed for M2M/IoT applications such as smart-metering, e-health, building and home automation with:
  - constrained nodes, e.g., 8-bit microcontrollers with 16KB RAM and battery-operated
  - constrained networks, e.g., Wireless Sensor Networks
- Low-overhead request/response protocol that also supports discovery of services/resources
- Based on UDP communications (with multicast)
- Inspired by (and compatible with) the HTTP protocol and REST architectures: CoAP is a specialized Web transfer protocol

CoAP Features

- **Web RESTful protocol** fulfilling M2M requirements in constrained environments
- **Simple request/response HTTP mapping**, to access CoAP resources via HTTP
- **URI** and Content-type support (a sensor is identified by an URI)
- **Low header overhead** and parsing complexity
- **Security** binding to Datagram Transport Layer Security (DTLS)
- UDP binding with **optional reliability**, supporting unicast and multicast
- **Asynchronous** message exchanges
- Services and resources **discovery**
- Also **publish/subscribe** (and push notifications)
- Simple **caching** (max-age parameter)

- CoAP implementations in many programming language, e.g., C, C++, and Java

CoAP Messaging



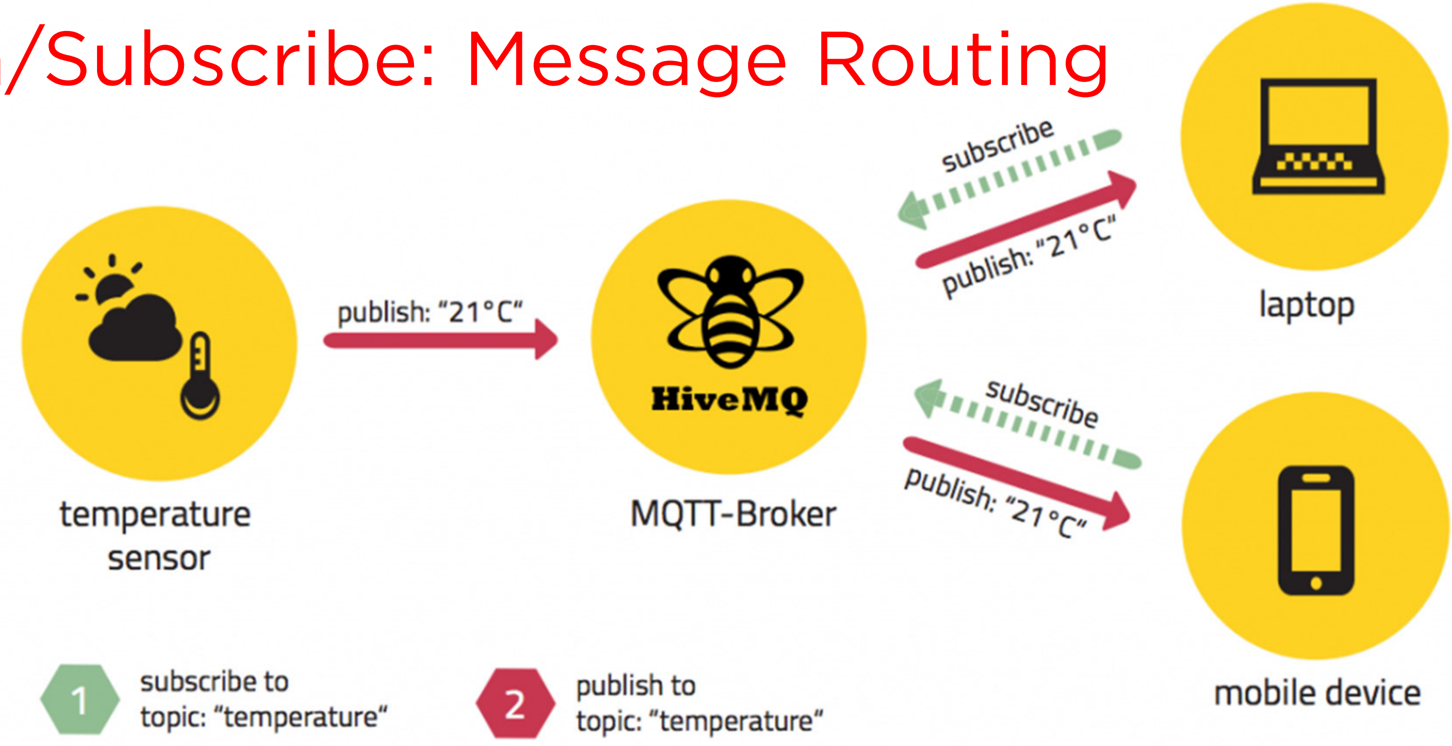
- Duplicate detection and optional reliability
- Possible messages:
  - confirmable message (CON), wait for ACK
  - acknowledgement message (ACK), in response to CON
  - non-confirmable message (NON), no need to wait for ACK
- CoAP on top of UDP allows for multicast requests

Publish/Subscribe model

- Publish/subscribe (pub/sub) pattern alternative to traditional client-server model, where a client communicates directly with an endpoint

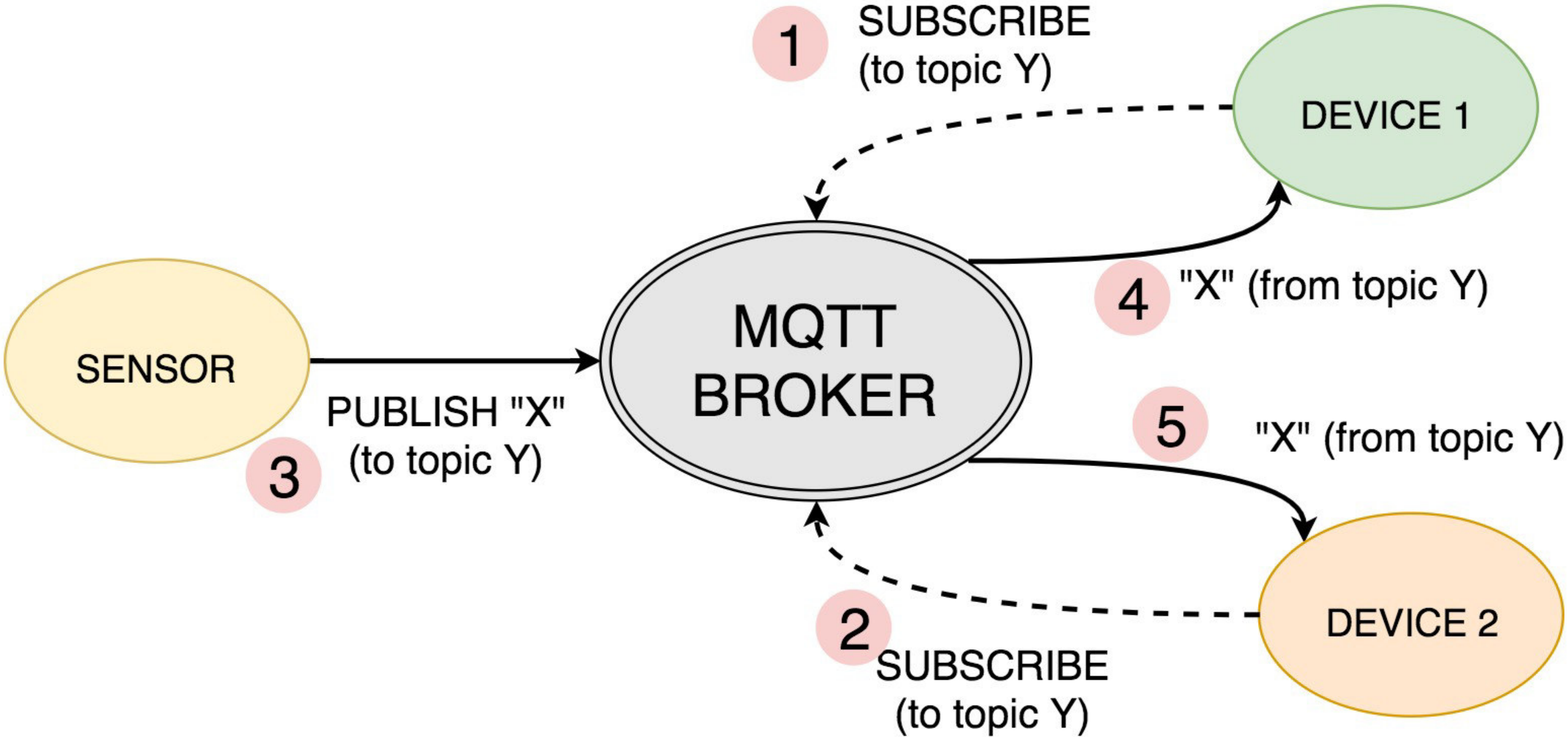
- Roles:
  - publisher: a client sending a message
  - subscriber: one or more receivers waiting for the message
  - broker: a central component receiving and distributing messages to interested receivers

Publish/Subscribe: Message Routing



- The broker **routes messages** (i.e., selects receivers of a message) based on:
  - Message Topic: a subject, part of each message. Receiving clients subscribe on the topics they are interested in with the broker and from there on they get all message based on the subscribed topics
  - Message Type: depending on the type of the message
  - Message Header: depending on a set of fields of the message
  - Message Content: possibly depending on the whole message content (expressive but expensive)

Publish/Subscribe: Typical Sequence



Publish/Subscribe: Decoupling

- Space decoupling: publisher and subscriber do not need to know each other (by ip address and port for example)
- Time decoupling: publisher and subscriber do not need to run at the same time
- Synchronization decoupling: operations on both components are not halted during publish or receiving

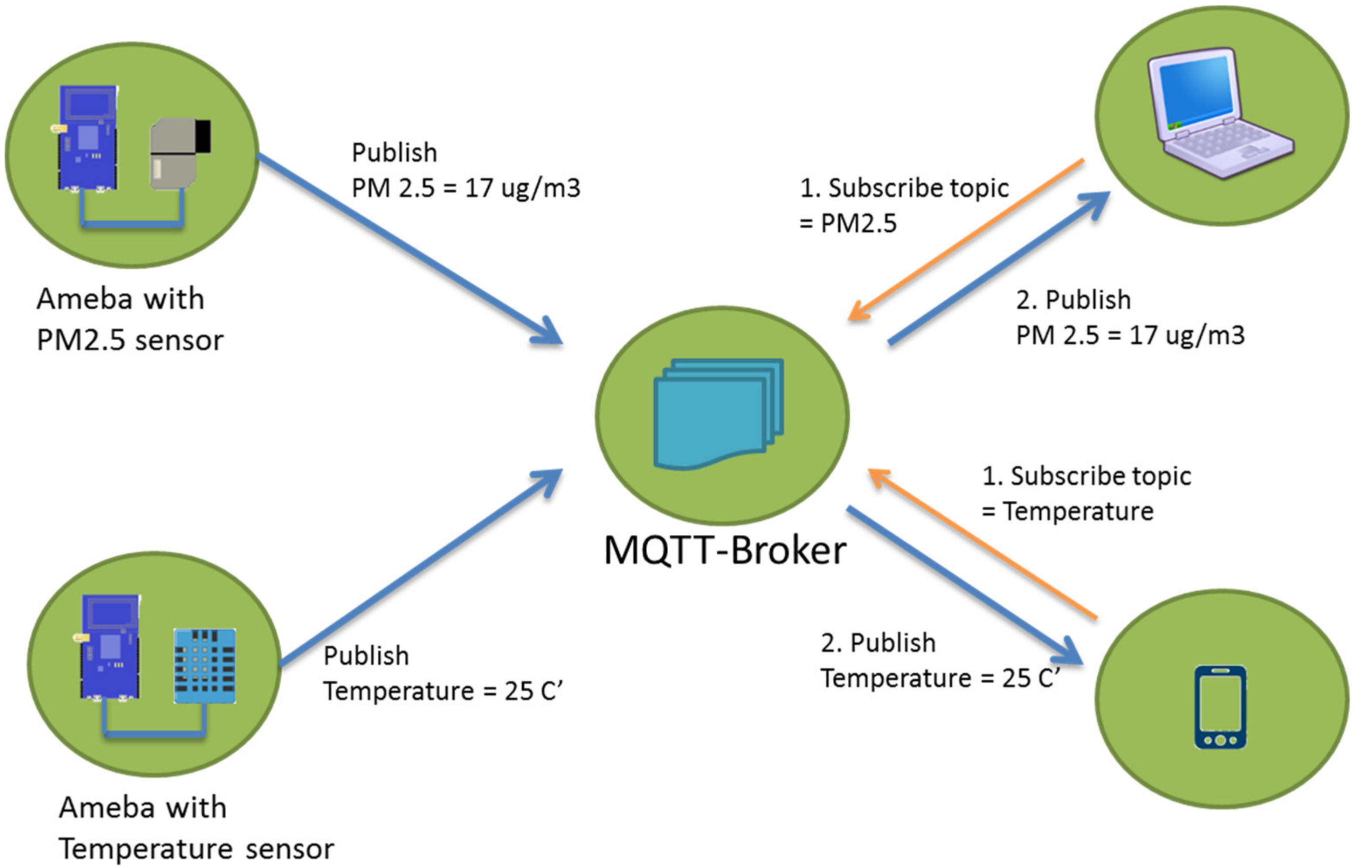
- Event system as logically centralized system
  - anonymous communication
  - possibility to use filters (on headers or entire messages)
  - basic primitives: subscribe, unsubscribe, publish

MQTT - Message Queue Telemetry Transport

- IBM developed WebSphere MQ as a message-based backbone mainly for Enterprise Application Integration (EAI), MQTT is its evolution:
  - an open standard but it is strongly supported by IBM
  - designed to permit WebSphere MQ to talk with constrained (smart) devices at the edge of the network

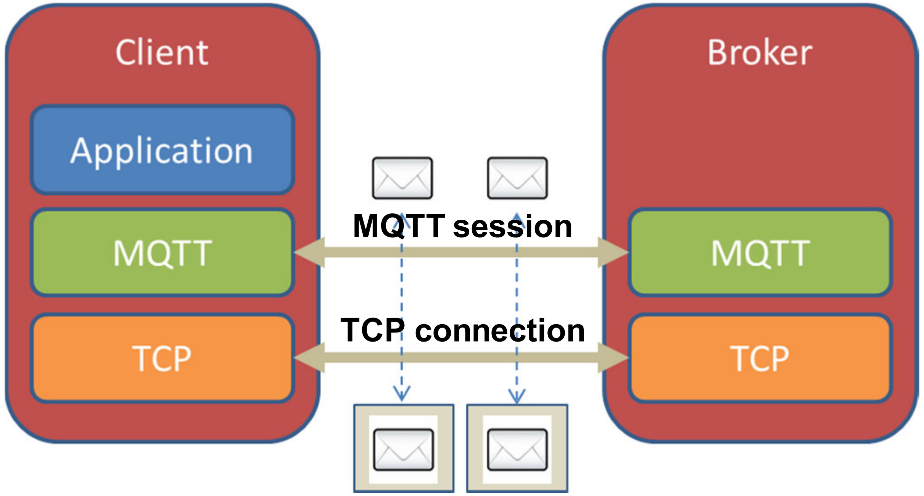
- MQTT: simple, lightweight, broker-based, publish/subscribe, open messaging protocol
- Ideal for use in constrained nodes and networks
  - on embedded devices with limited processor or memory resources
  - where the network is expensive, has low bandwidth or is unreliable
- MQTT-SN for wireless sensor networks, aimed at embedded devices on non-TCP/IP networks (such as Zigbee).

MQTT: Typical Use Case



1. Subscribe to one ore more topic
2. Publish to a topic
3. Receive messages related to subscribed topics

MQTT: Features (1)



- **Publish/subscribe** message pattern to provide
  - one-to-many message distribution
  - decoupling of applications
- Use of TCP/IP to provide basic network connectivity
- **Small transport overhead** (minimal header length just 2 bytes) and protocol exchanges minimised to reduce network traffic
- **Easy to use** with few commands: connect, subscribe, publish, disconnect
- **Retained messages**: an MQTT broker can retain a message that can be sent to newly subscribing clients

MQTT: Features (2)

- **Three message delivery semantics** with increasing reliability and cost:
  - Quality of Service (QoS): at least once, at most once, exactly once
- **Durable connection**: client subscriptions remain in effect even in case of disconnection
  - subsequent messages with high QoS are stored for delivery after connection reestablishment
- **Wills**: a client can setup a will, a message to be published in case of unexpected disconnection, e.g., an alarm

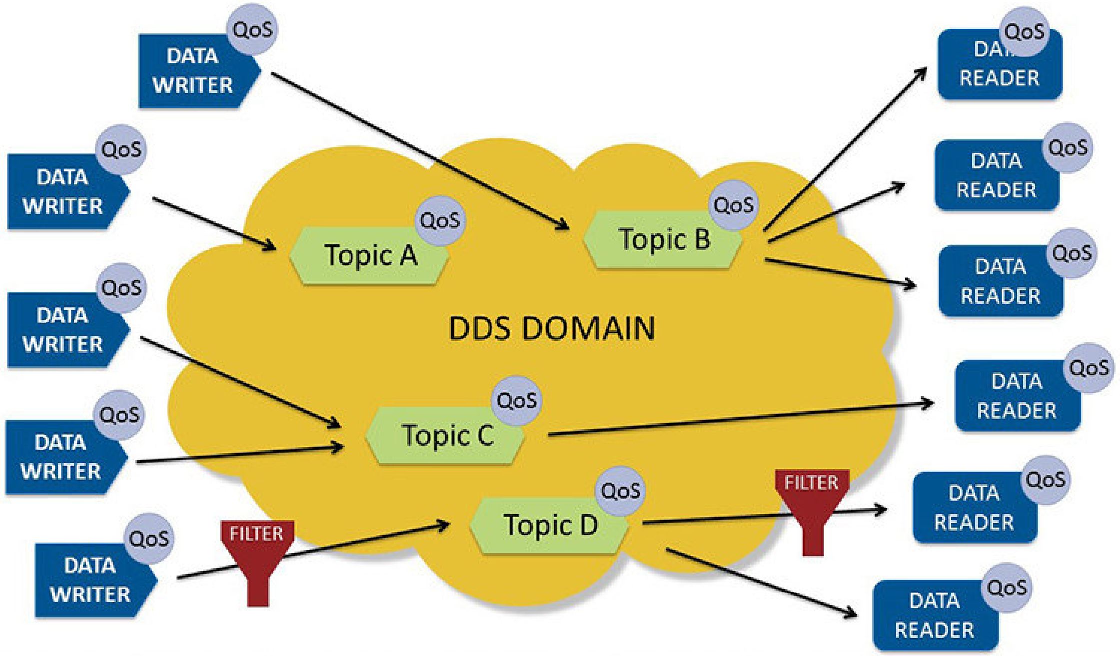
- MQTT is a protocol adopted in several platforms: there are MQTT brokers in WebSphere, Mosquitto, RabbitMQ, etc. and clients in several languages

AMQP –
Advanced Message Queuing Protocol

- **Richer semantic** than MQTT, e.g., supports topics and queues, but also **heavier**, e.g., the broker is much more complex
- AMQP implementations, e.g., Apache Qpid, focus on providing **several features**: queuing, message distribution, security, management, clustering, federation, heterogeneous multi-platform support
  - most of them (possibly) not essential in the IoT scenario

- Originally developed at JPMorgan Chase in London, AMQP was designed as a message-oriented protocol for the integration of enterprise IT components (Enterprise Message Bus)

DDS -
Data Distribution Servi



- Publish/subscribe, but broker-less (based on multicast)
  – scalable, real-time, dependable, high performance and interoperable

- Proposed in several mission-critical environments where performance and reliability are essential, e.g., industrial automation, financial applications, air traffic control and management, and even military environments
  – http://twinoakscomputing.com/datasheets/DDS-Brochure.pdf

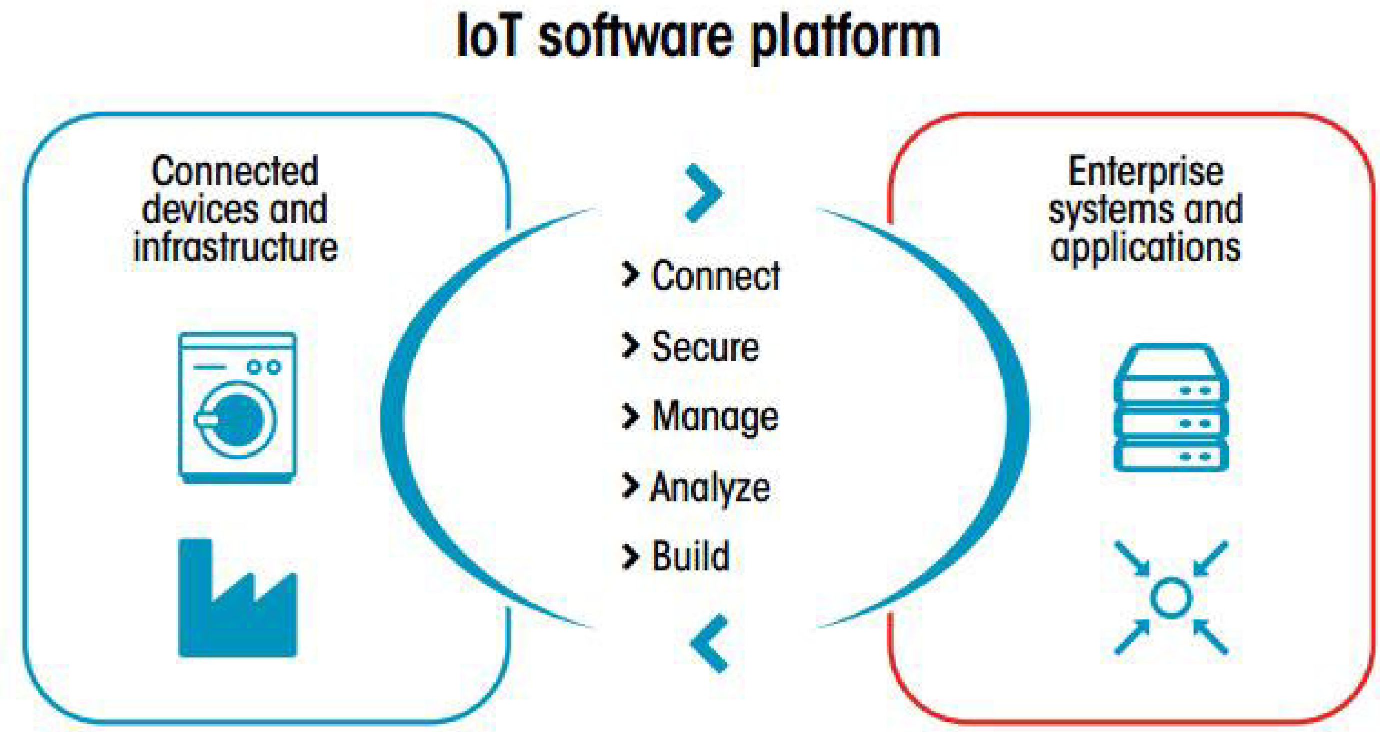
- Standard developed by the Object Management Group

Outline

1. Introduction to IoT
   - definition, enabling technologies and concepts to better understand the general framework of IoT solutions
   - layering architecture, cloud computing vs. fog computing

2. Most relevant components of IoT solutions
   - devices, wireless communication protocols, data exchange protocols, IoT platforms, and data analysis

IoT Platforms

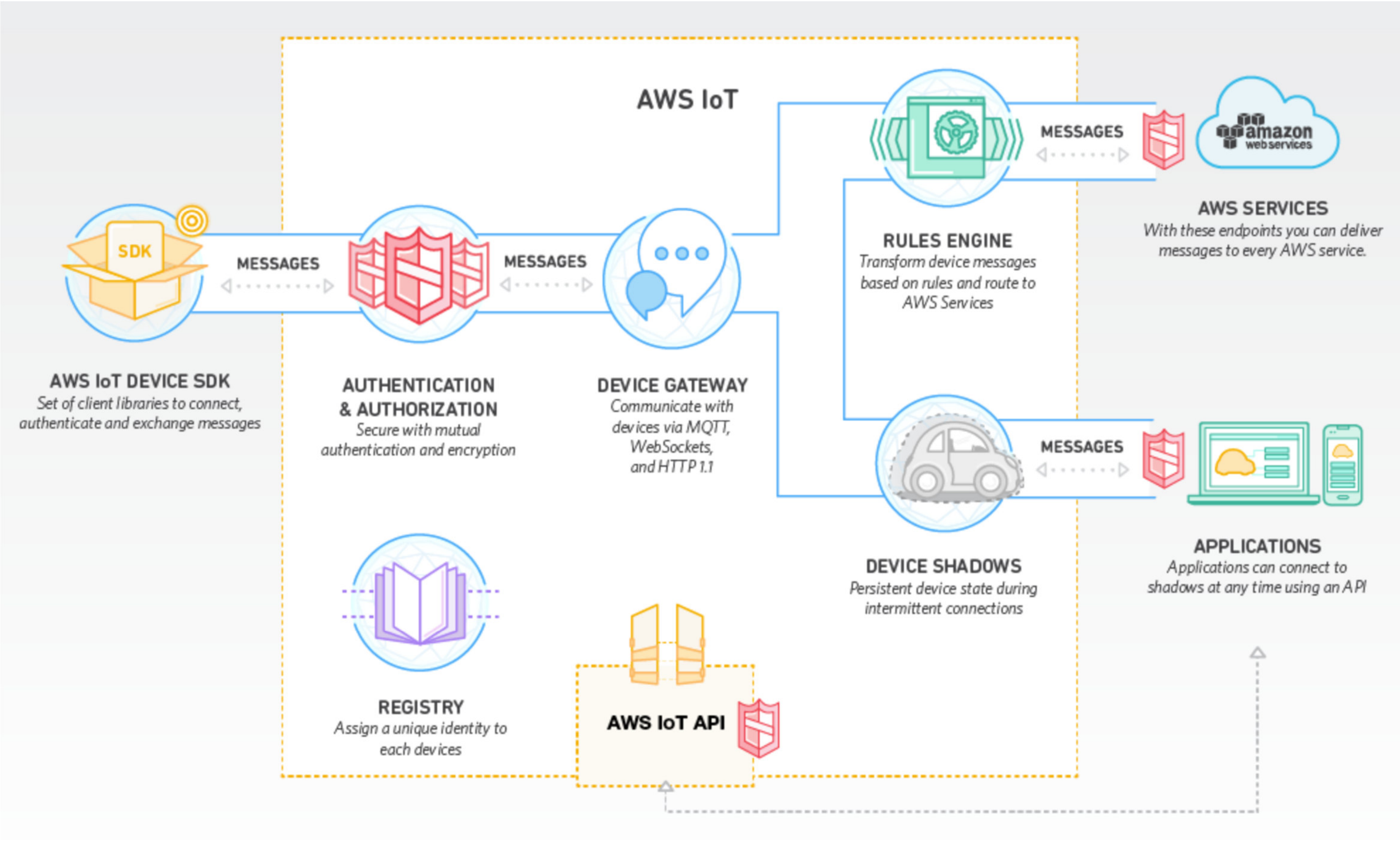


- IoT Platform between things/devices and business applications
  - connect devices to gather information
  - secure communication with devices and applications
  - manage devices to control their behavhior
  - analyze data, e.g., with AI
  - build applications, also interacting with CRM/ERP/…

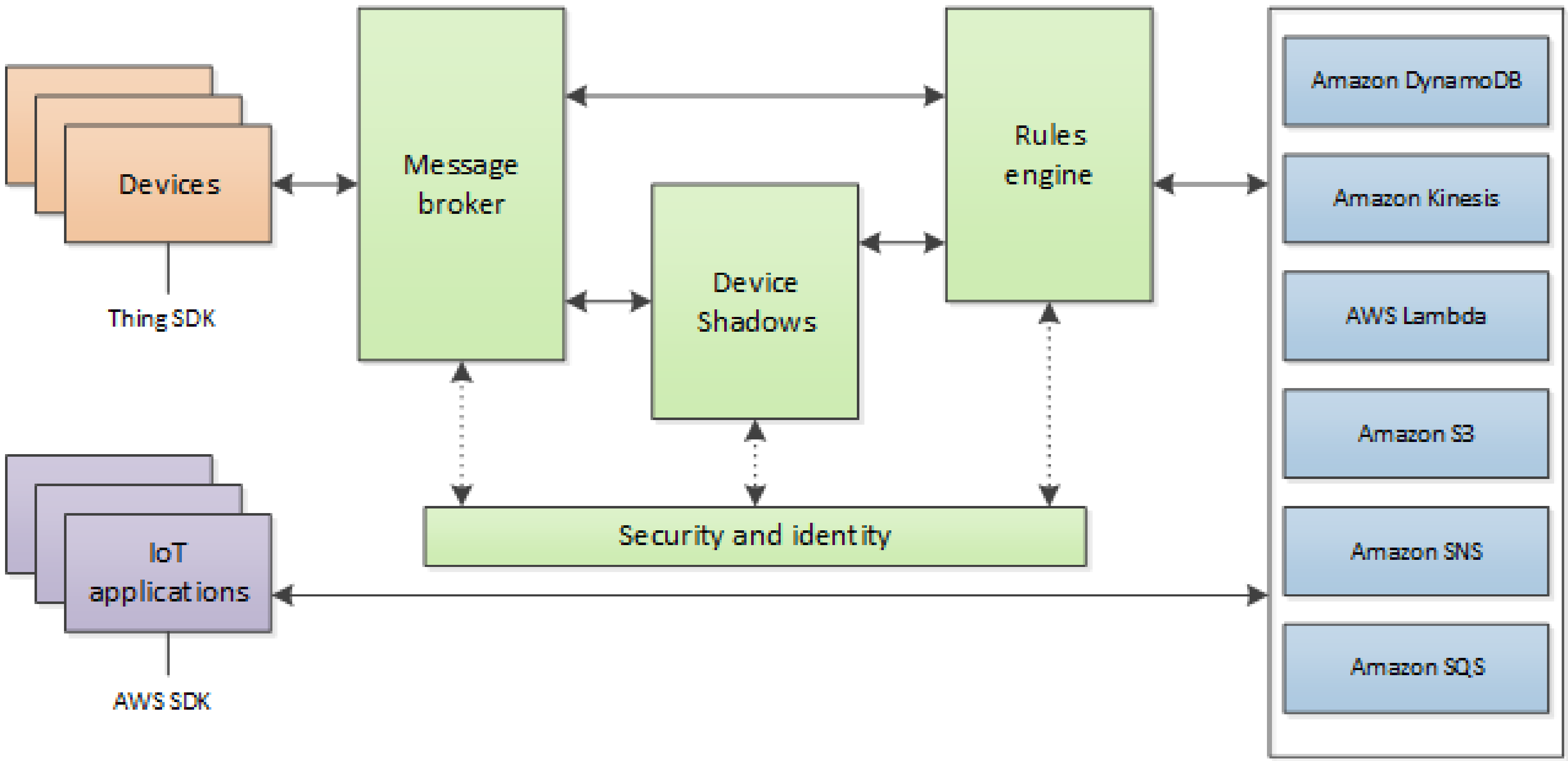
Several IoT Platforms

- Amazon Web Services (AWS) IoT
- Microsoft Azure IoT
- Google Cloud Platform
- ThingWorx IoT Platform
- IBM Watson
- Mindsphere by Siemens
- Carriots
- Kaa
- ...
- also local SMEs/startups like Stoorm5

Amazon Web Services (AWS) IoT



AWS IoT: Architecture



- https://docs.aws.amazon.com/iot/latest/developerguide/aws-iot-how-it-works.html

AWS IoT: Main Components (1)

- AWS IoT Device SDK connects devices to AWS IoT by using the MQTT, HTTP, or WebSockets protocols
  - the Device SDK supports C, Java, JavaScript, Arduino, iOS, Android,…
- Device Gateway supports (secure) communication with devices, by using a (1:1 or 1:n) publish/subscribe model
  - supports MQTT, WebSockets, and HTTP 1.1 protocols
- Authentication and Authorization with mutual authentication and encryption
  - authentication with native AWS (called 'SigV4') as well as X.509 certificate based authentication
  - AWS facilitates the whole certificate process management
- Registry assigns a unique identity to each device
  - also supports metadata describing capabilities of devices, e.g., whether a sensor reports temperature and if data in Fahrenheit or Celsius

AWS IoT: Main Components (2)

- Device Shadows, i.e., persistent, virtual version of devices
  - applications or other devices exchange messages and interact through the Shadow Device
  - Device Shadows persist the last reported state and desired future state of each device even when the device is offline
  - exploit it to retrieve the last reported state of a device or set a desired future state through the API or using the rules engine

- Rules Engine, supporting to build IoT applications that gather, process, analyse, and act on data generated by connected devices
  - evaluates inbound messages published into AWS IoT and transforms and delivers them to another device or a cloud service, based on business rules you define
  - a rule can apply to data from one or many devices, and it can take one or many actions in parallel

RabbitMQ

- Not actually an IoT platform, but an interesting **messaging infrastructure**

- **Robust** messaging for applications
- **Reliable** message delivery
- Distributed deployment as **clusters** for high availability and throughput
- **Federate** across multiple availability zones and regions.
- **Multi-Protocol:** AMQP, MQTT, HTTP, etc.
- **Managing and monitoring** via HTTP-API, command line tool, and UI

- Runs on all major operating systems
- Supports a huge number of developer platforms
- Open source and commercially supported

RabbitMQ: Durable, Persistent, and Synchronized

- Exchanges (Topics) and Queues can be durable
  - capable of surviving to a broker restart (as opposed to transient)
- Messages can be defined as persistent
  - only persistent messages survive exchange/queue/brokers/channel failures
- Synchronization
  - automatic acknowledgement model, i.e., verify that the message is actually delivered to the application waiting for it
  - explicit acknowledgement model, i.e., wait for explicit acknowledgment sent back by the application
    - immediately at message reception, after processing, …

MTConnect

- Protocol/platform specialized for data exchange between shop floor equipment and software applications, over networks using the Internet Protocol (IP)
- Lightweight, open, extensible, and read-only
  - introduced only for the monitoring of numerically controlled machine tools
- MTConnect exploits several Internet open standards
  - data (from shop floor devices) is presented in XML format
  - communication via HTTP with a RESTful interface
  - Lightweight Directory Access Protocol (LDAP) for discovery services
- MTConnect was developed at UCB (University of California Berkeley) and GeorgiaTech, now managed by the MTConnect Institute, a forum of companies and organizations

MTConnect Dictionary

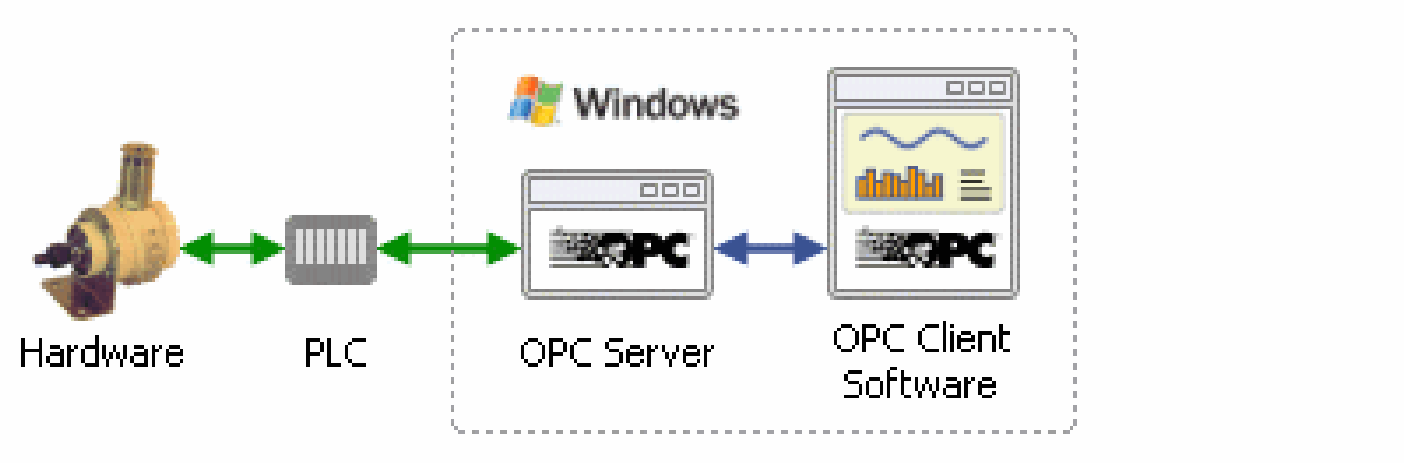
- Standard dictionary for manufacturing data, e.g.,

Data Item type/subtype	Description
ACCELERATION	Rate of change of velocity
ACCUMULATED_TIME	The measurement of accumulated time associated with a Component
ANGULAR_ACCELERATION	Rate of change of angular velocity.
ANGULAR_VELOCITY	Rate of change of angular position.
AMPERAGE	The measurement of AC Current or a DC current
ALTERNATING	The measurement of alternating current. If not specified further in statistic, defaults to RMS Current
DIRECT	The measurement of DC current
ANGLE	The angular position of a component relative to the parent.
ACTUAL	The angular position as read from the physical component.
COMMANDED	The angular position computed by the Controller.
AXIS_FEEDRATE	The feedrate of a linear axis.
ACTUAL	The actual federate of a linear axis.
COMMANDED	The feedrate as specified in the program.
OVERRIDE	The operator's overridden value. Percent of commanded.
CLOCK_TIME	The reading of a timing device at a specific point in time. Clock time **MUST** be reported in W3C ISO 8601 format.
CONCENTRATION	Percentage of one component within a mixture of components
CONDUCTIVITY	The ability of a material to conduct electricity

OPC Foundation

- OPC = OLE (Object Linking and Embedding) for Process Control
- Microsoft software architecture for industrial control systems (1996)
- Designed for connecting Windows based PC with PLC and SCADA systems in industrial automation
- Based on COM ar ... ntly related to Microsoft technol

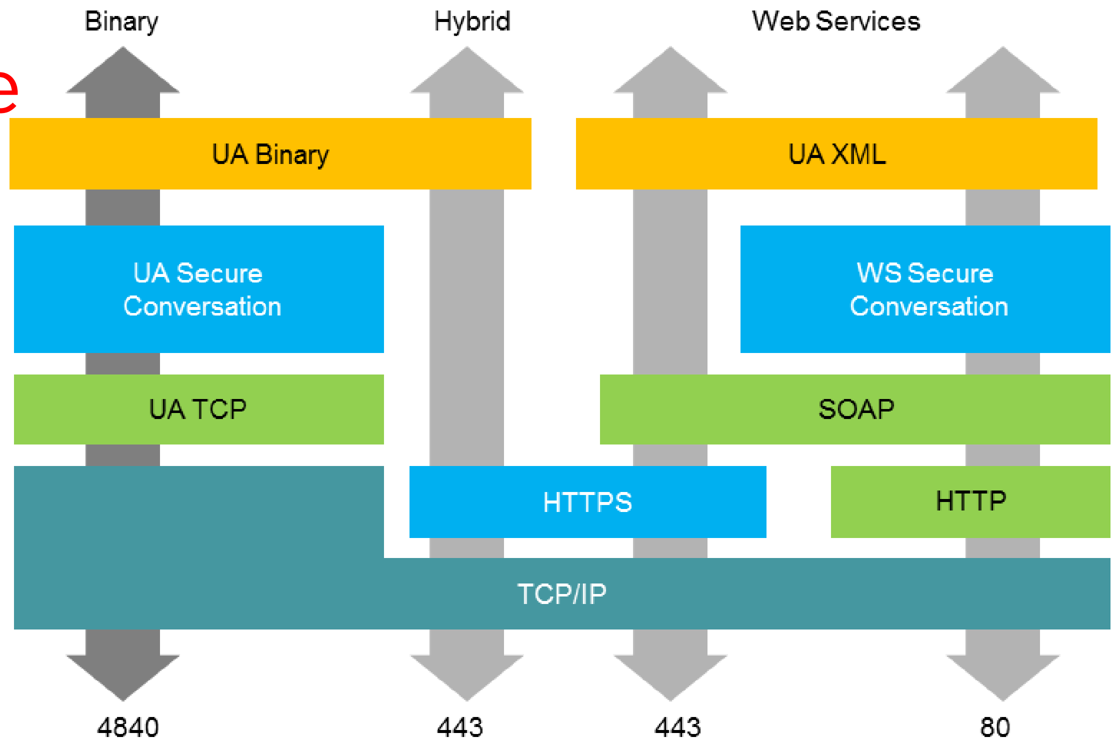
OPC Architecture



- Client/Server architecture
  - the client and the server can be on the same or different machines

- Interface between OPC Server and a specific PLC always vendor-specific
  - typically each PLC producer sells also its specific OPC Server component

OPC UA – Unified Architecture



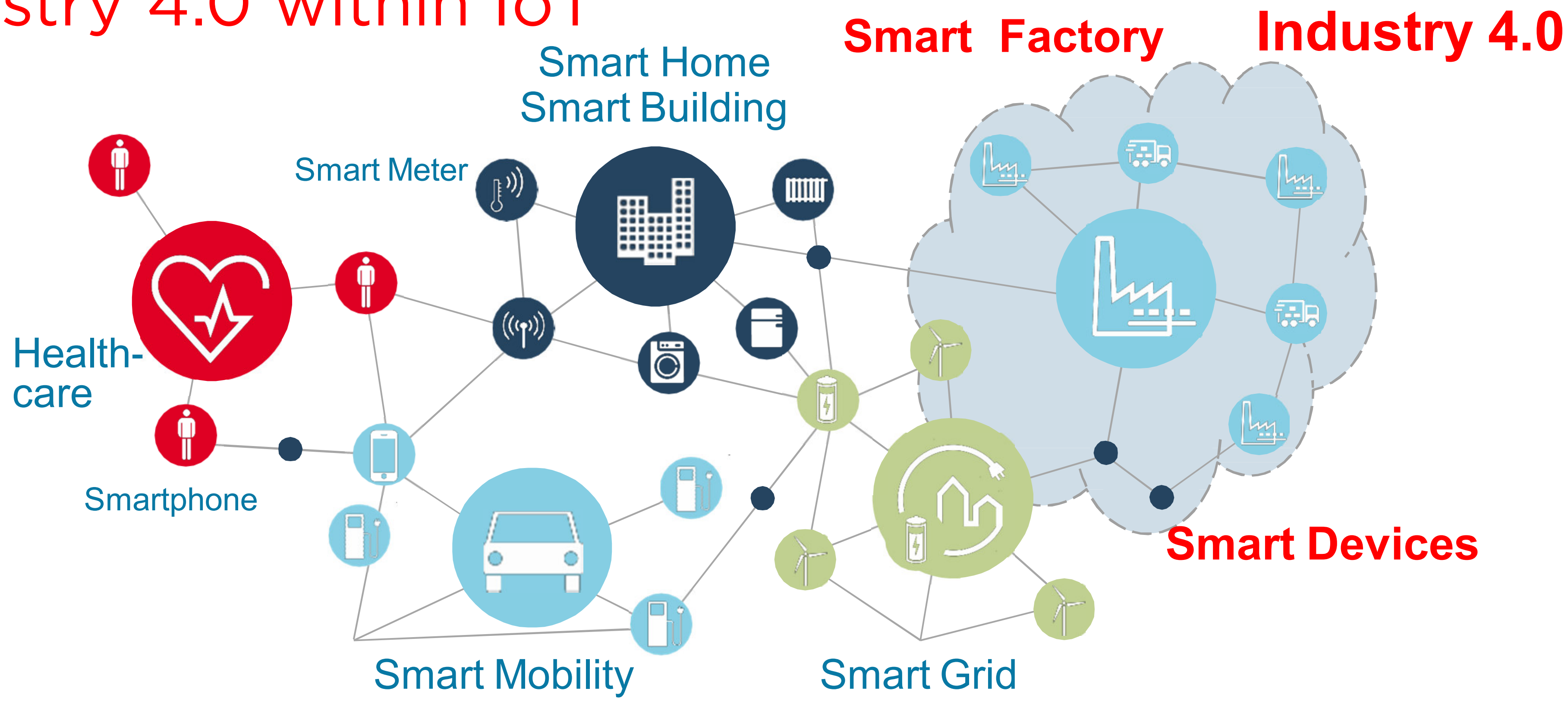
- Modern release of OPC (2006), greatly differing from the previous one
- Platform independent: PC, ARM, Cloud, Windows, Linux, Android…
- Secure: authentication, encryption, auditing, fault tolerance…
- Extensible: multi-layered architecture of OPC UA provides a "future proof" framework

Outline

3. Industry 4.0
   - Introduction and definition
   - Industry 4.0 and the digital twin

4. Examples of IoT and Industry 4.0 solutions
   - From e-Maintenance to i-Maintenance
   - IoT-enabled use cases
   - experience with a business case

Industry 4.0 within IoT



- Industry 4.0 is typically enabled and part of the IoT trend
- IoT has enabled Smart City environments
  - Smart mobility, Smart Grid, Smart mobility, Smart people, …
- And now it expands to industries scenarios

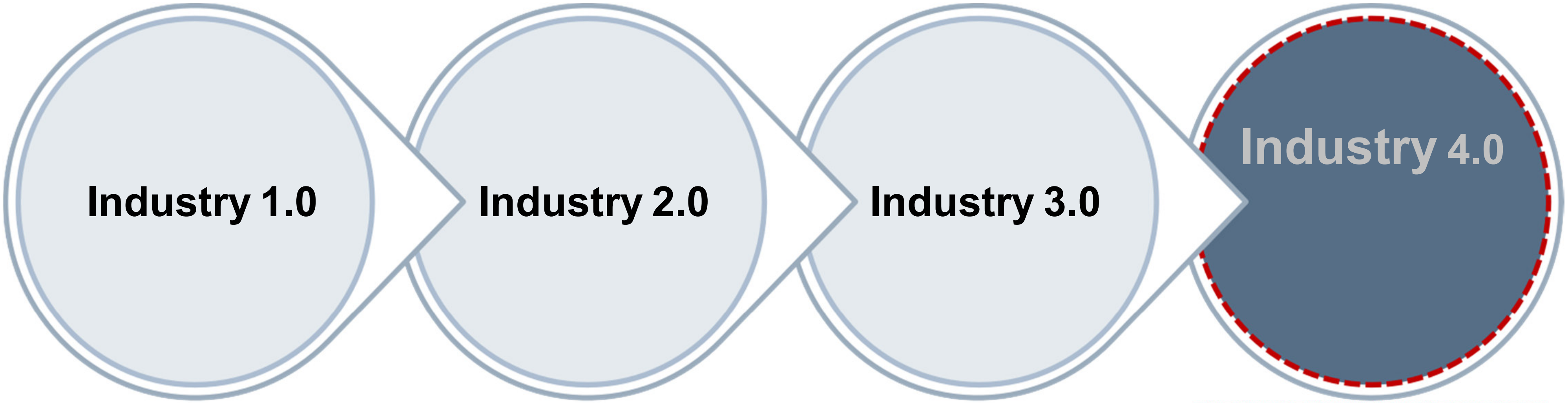
Industry 4.0 - I4.0

- Industry 4.0 is a large spread phenomenon and trend to consider an evolution of traditional industrial processes

- Industry 4.0 (I4.0) has multiple meanings
  - connects/merges production with ICT
  - merges customer data with machine data
  - goes M2M: machines communicate with machines
  - components and machines autonomously manage production in a flexible, efficient, and resource-saving manner

Industry 4.0 - I4.0

- Industry 4.0 is in the trends of the industrial revolutions

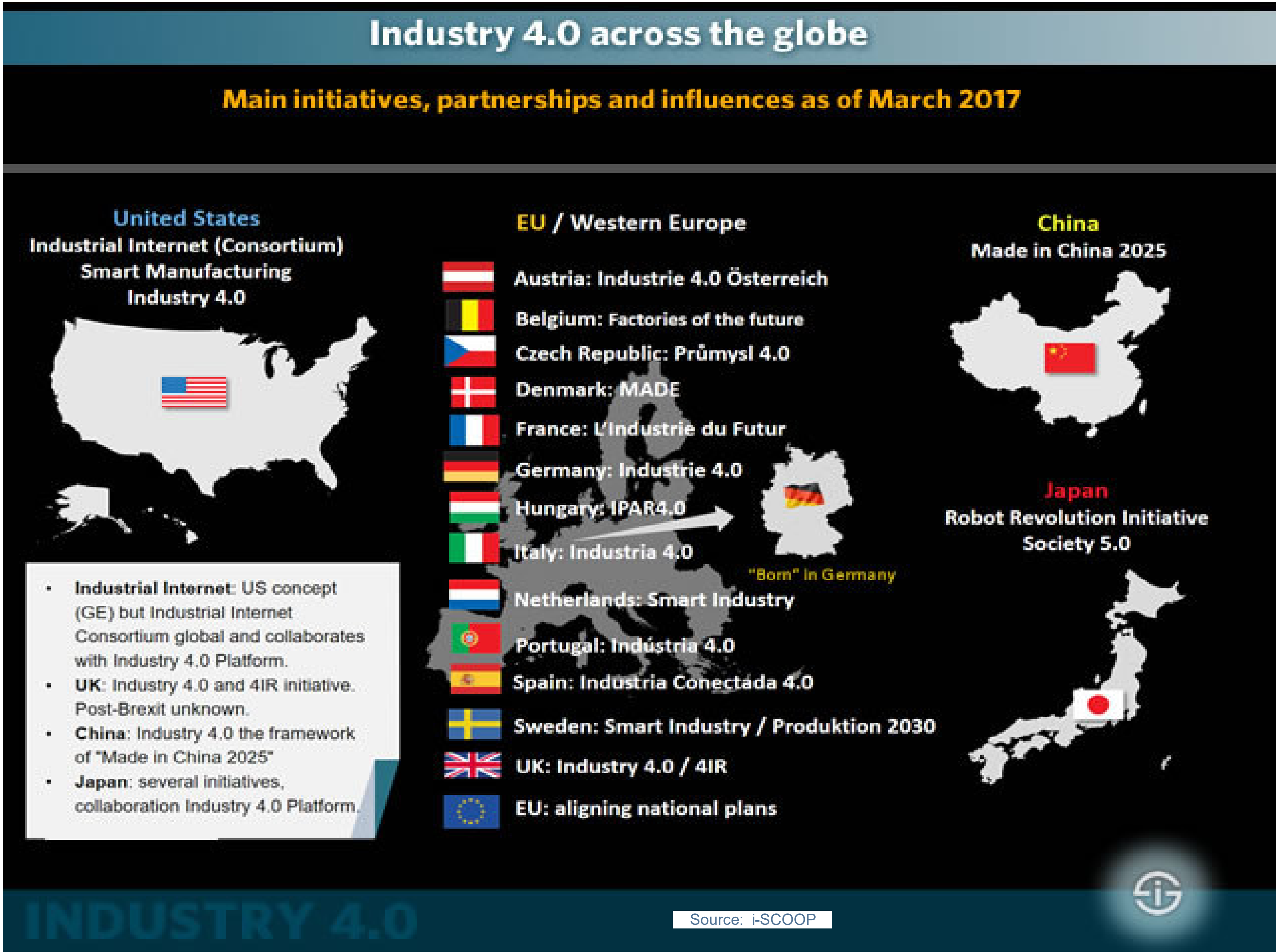


- **1760s–1900**
- Use of **steam** and mechanically-driven production facilities

- **1900–1970s**
- **Electric** power-driven mass production based on division of labor

- **1970s to date**
- Extensive use of **controls**, IT, and electronics for an **automated and high productivity** environment

- **Future**
- **Smart**: based on integration of virtual and physical production systems


Industry 4.0 across the globe
Main initiatives, partnerships and influences as of March 2017
United States
Industrial Internet (Consortium)
Smart Manufacturing
Industry 4.0
EU / Western Europe
Austria: Industrie 4.0 Österreich
Belgium: Factories of the future
Czech Republic: Průmysl 4.0
Denmark: MADE
France: L'Industrie du Futur
Germany: Industrie 4.0
Hungary: IPAR4.0
Italy: Industria 4.0
"Born" in Germany
Netherlands: Smart Industry
Portugal: Indústria 4.0
Spain: Industria Conectada 4.0
Sweden: Smart Industry / Produktion 2030
UK: Industry 4.0 / 4IR
EU: aligning national plans
China
Made in China 2025
Japan
Robot Revolution Initiative
Society 5.0
Industrial Internet: US concept (GE) but Industrial Internet Consortium global and collaborates with Industry 4.0 Platform.
UK: Industry 4.0 and 4IR initiative. Post-Brexit unknown.
China: Industry 4.0 the framework of "Made in China 2025"
Japan: several initiatives, collaboration Industry 4.0 Platform.
INDUSTRY 4.0
Source: i-SCOOP

Industry 4.0

- Industry revolution 1 – 3
  - mechanized processes
  - mass production
  - production automation

- Industry revolution 4.0 – Goals
  - product innovation
  - increased collaboration
  - operational process enhancement
  - cyber-physical production

- Strategic Trends
  - convergence of applications will form the core of new advances
  - energy efficiency and sustainability toward greater business focus
  - greater presence of mobility and Web-based information systems

Definition of Industry 4.0

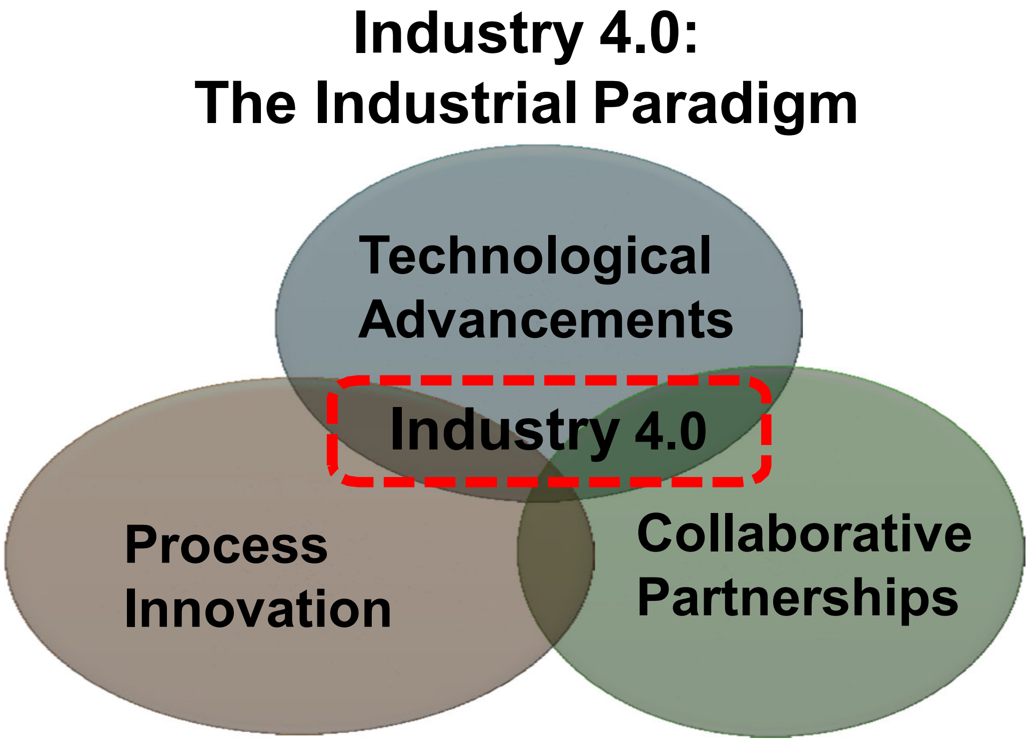
- The information-intensive transformation of manufacturing
  - in a connected environment of data, people, processes, services, systems and production assets
  - with the generation, leverage and utilization of actionable information
  - as a way and means to realize the smart factory and new manufacturing ecosystems

Industry 4.0 Vision

- In Industry 4.0 vision, Internet of Things is the enabler of the integration across:
  - the horizontal areas of product design, engineering, development, and disposal, creating new value networks in the supply chain
  - the vertical segments within the manufacturing enterprise: from the shop floor to the top management integration

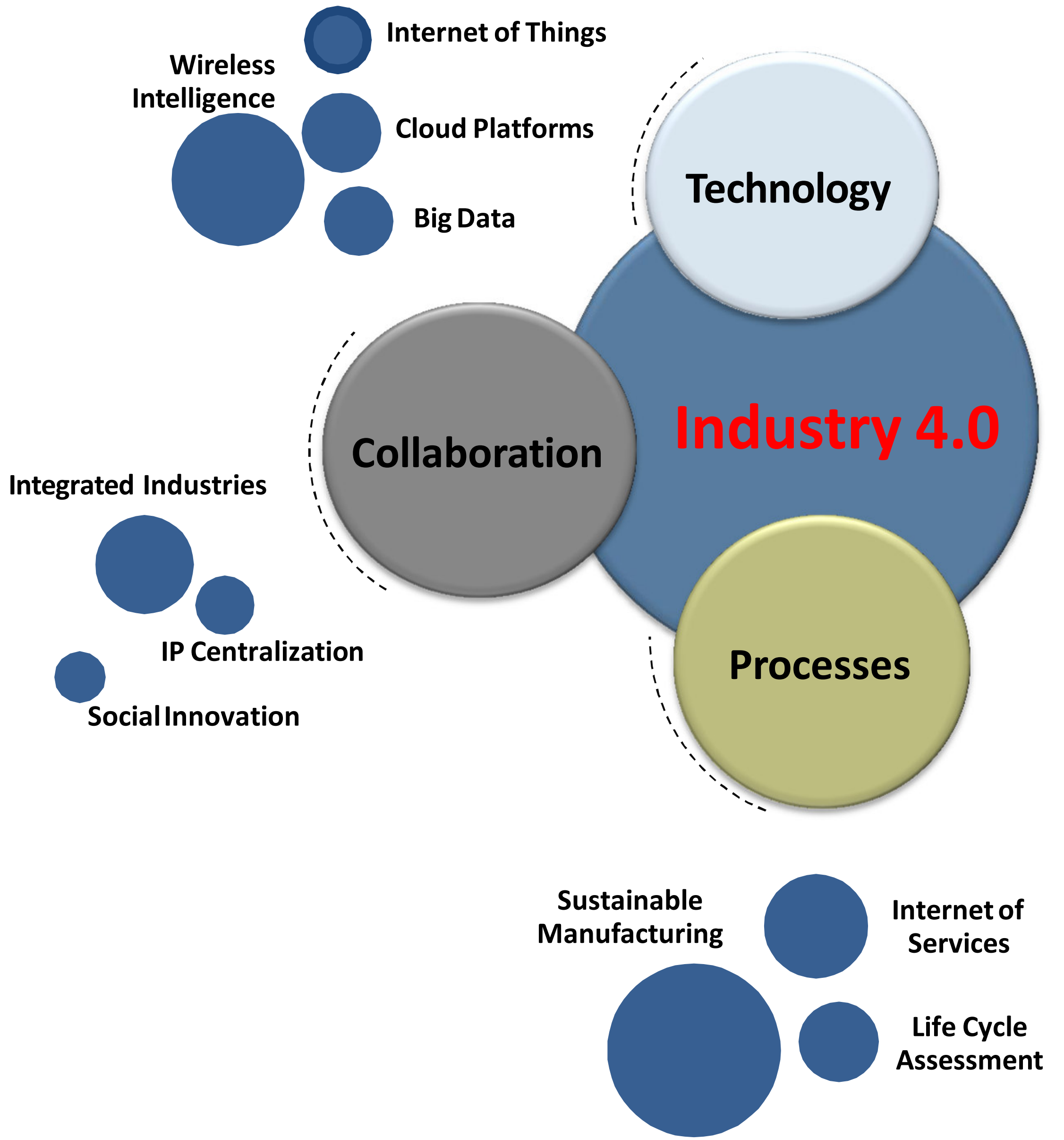
Industry 4.0: Pillars and Main Aspects

- Four functional pillars or the enablers of Industry 4.0:
  - Big Data
  - Internet of Things
  - Internet of Services
  - Integrated Industries
- They will converge into three main aspects:
  - Technology
  - Collaboration
  - Processes



Industry 4.0

- Industry 4.0 is in the sense of product innovation in manufacturing as an effort in three areas
  - technology
  - collaboration
  - processes



Industry 4.0: Potential

- Industry 4.0 has been spoken about since some years ago and there is a large expectat
- Considering any vertical production are
- 78 Billion euros by 2025!
- Connects/merges production with ICT
- Merges customer data with machine data

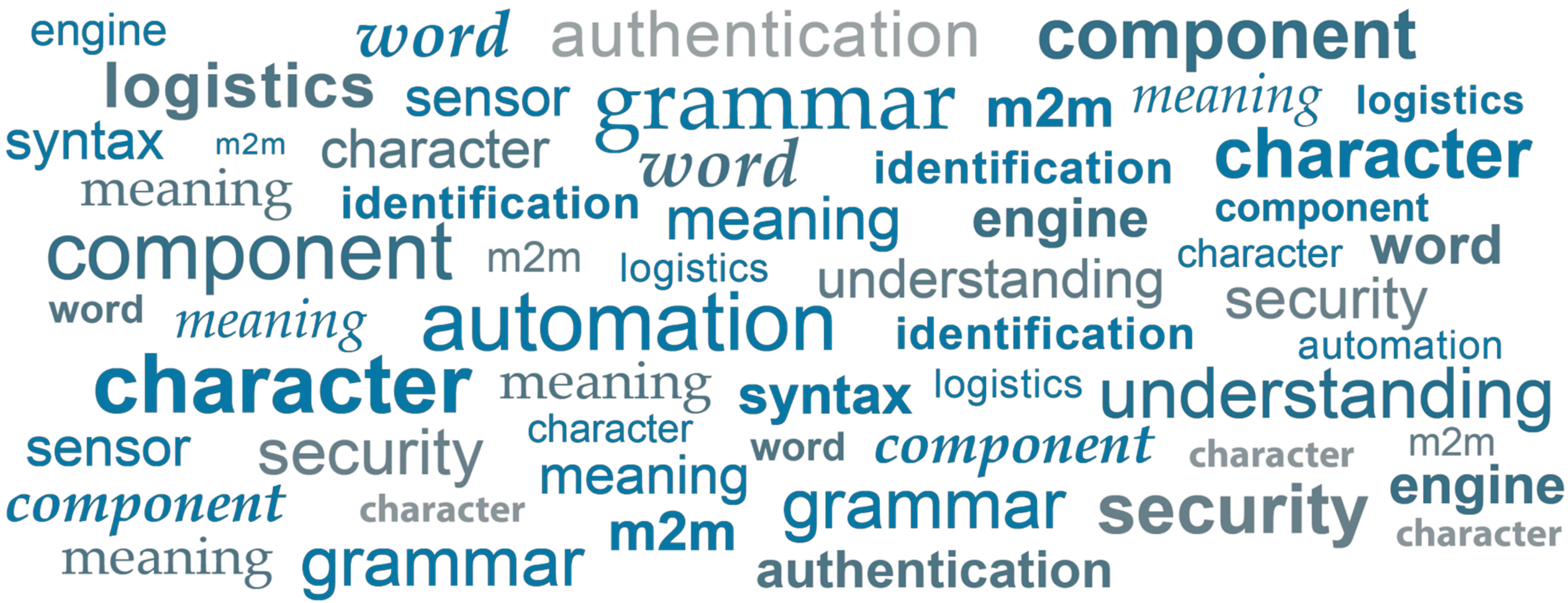
Industry 4.0: Benefits

- Industry 4.0 has created a large momentum so to make people wait for:
  - higher quality
  - more flexibility
  - higher productivity
  - earlier launch of products
  - standardization in development
  - continuous benchmarking and improvement

- Global competition among strong businesses
- New labor market opportunities
- Creation of appealing jobs at the intersection of mechanical engineering, automation, and IT
- New services and business models

Industry 4.0: prerequisites

- Defining communication structures
- Development of a common language with its own signs, alphabet, vocabulary, syntax, grammar, semantics, pragmatics and culture

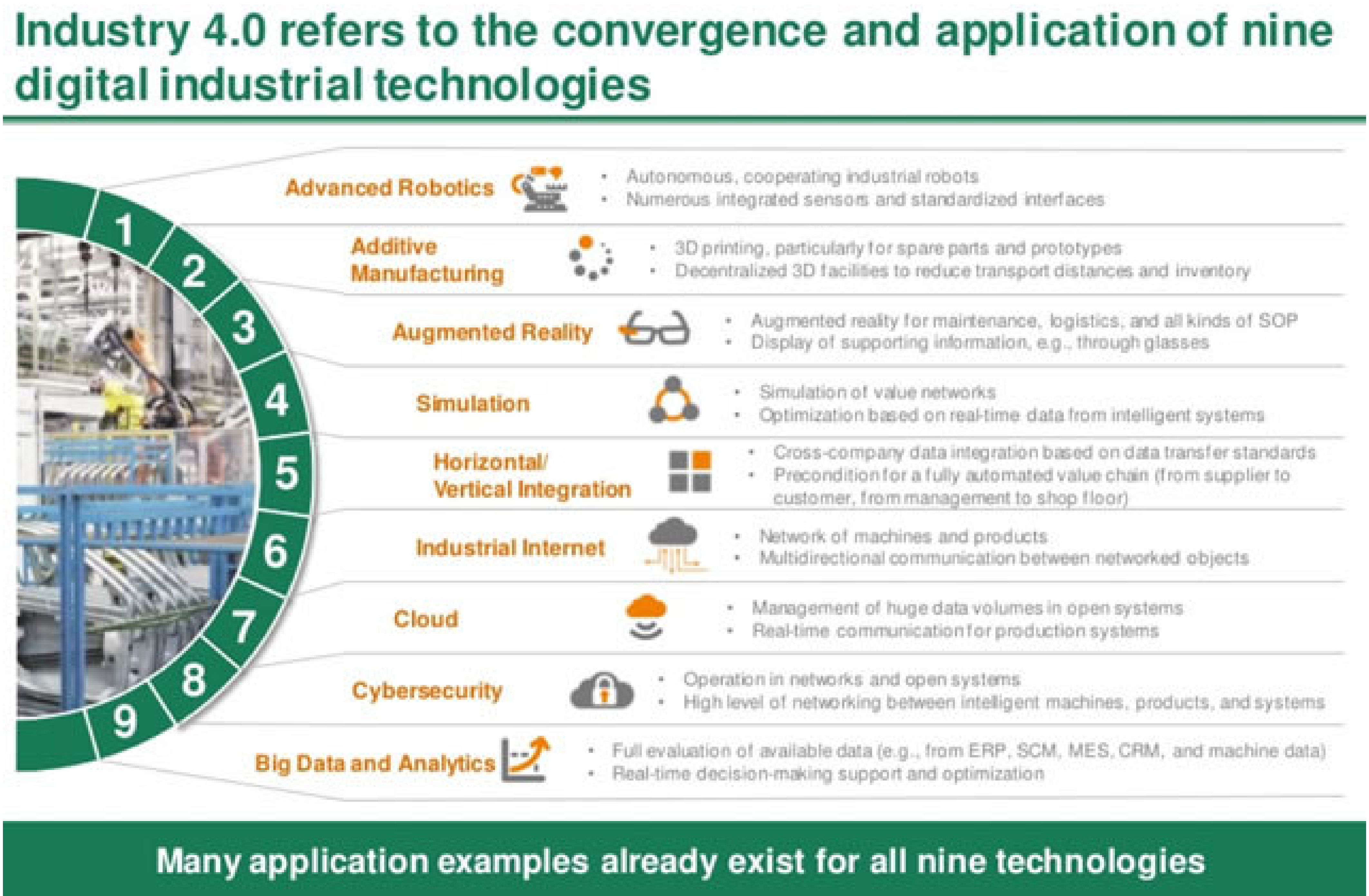


Industrie 4.0

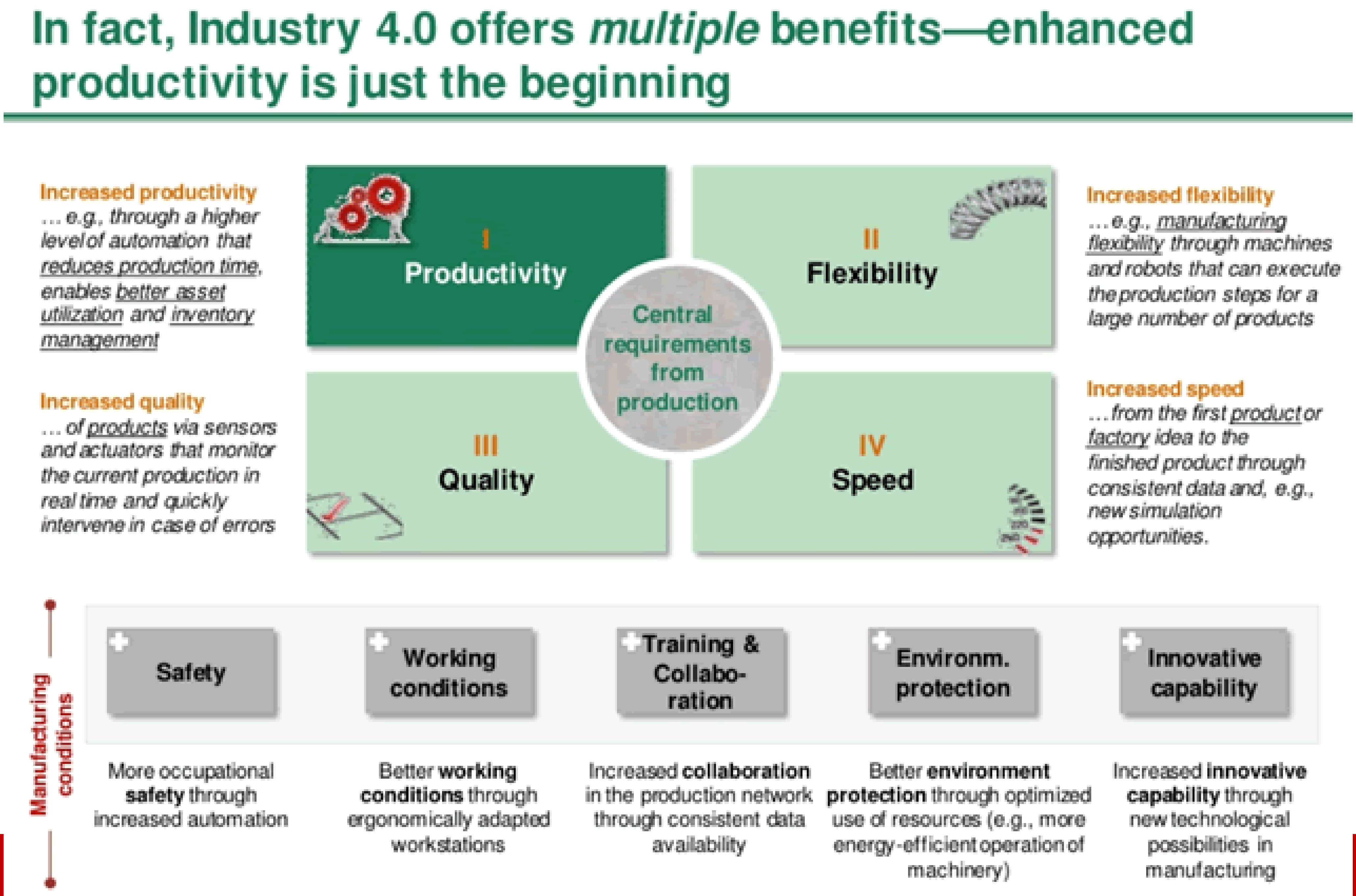
- Industry 4.0 is closely related to the Germany effort for the modernization of the entire Industry there, Industrie 4.0
- German Industrial associations and research organizations (ZVEI, VDMA, BITKOM, VDI, etc.) realize the vision of "Smart factory"
  - emphasizing the vertical integration in a factory setting
  - proposing an horizontal integration that goes behind factory floor to include the whole value chain

- The ultimate goal is to accompany the production into a modern view with a reference architecture on whose goals all stakeholders have agreed: the Reference Architecture Model for Industry (RAMI 4.0)

Industrie 4.0 Digital Industrial Technologies

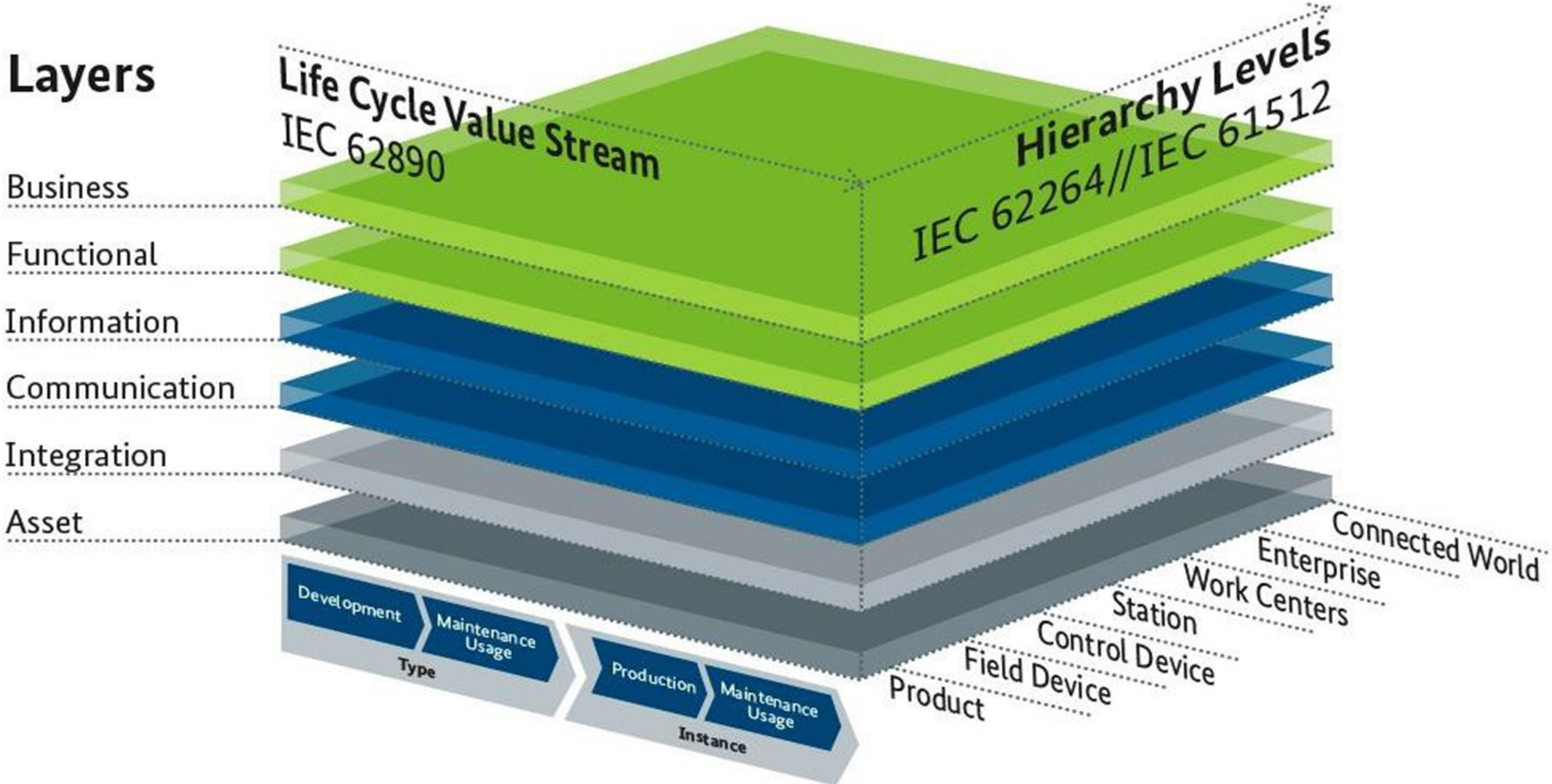


Industrie 4.0: Not Only Productivity



RAMI 4.0 as a standard

- The Reference Architectural Model for Industry 4.0 (RAMI 4.0) is a comprehensive model to collocate any effort in the area and to make stakeholder sure that all understand the organization and the structure



RAMI 4.0: Main Aspects

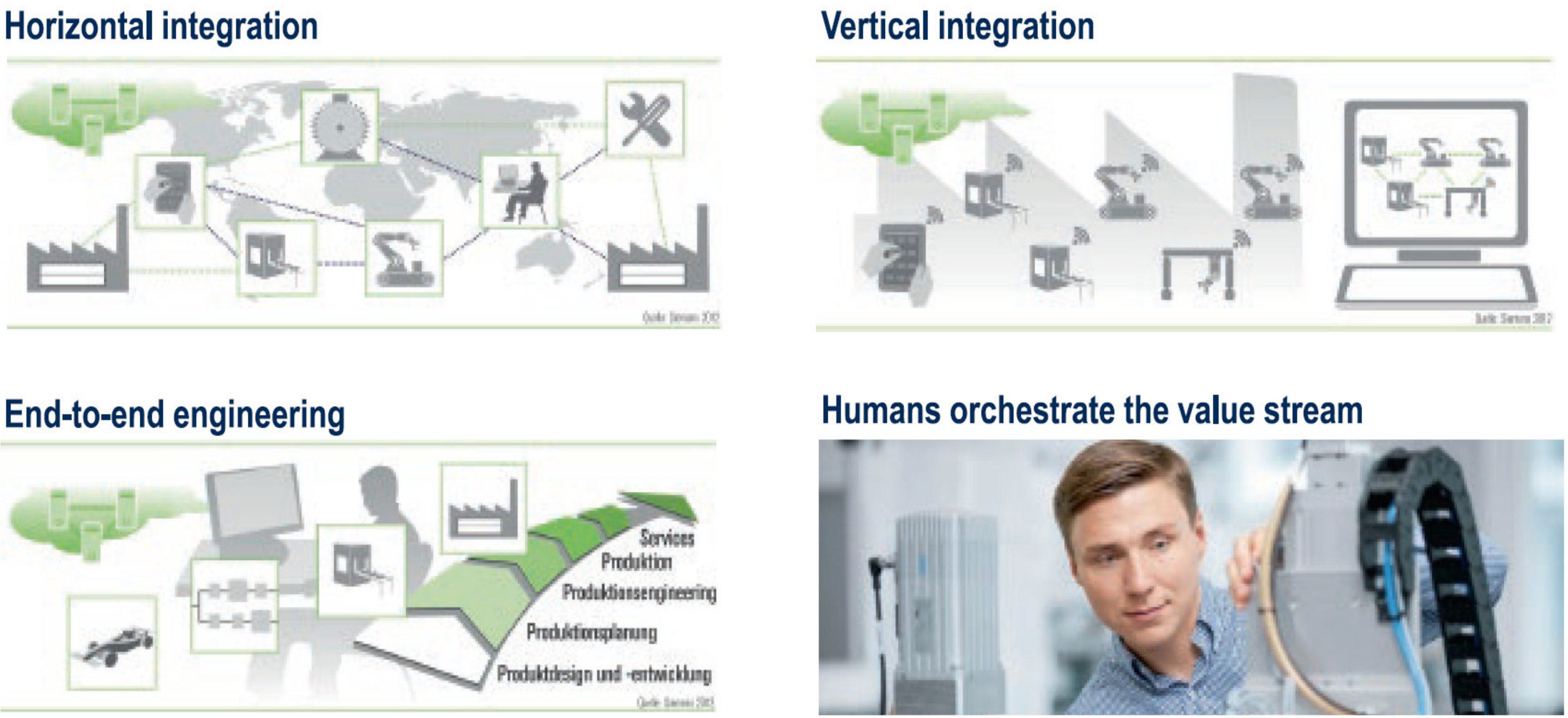
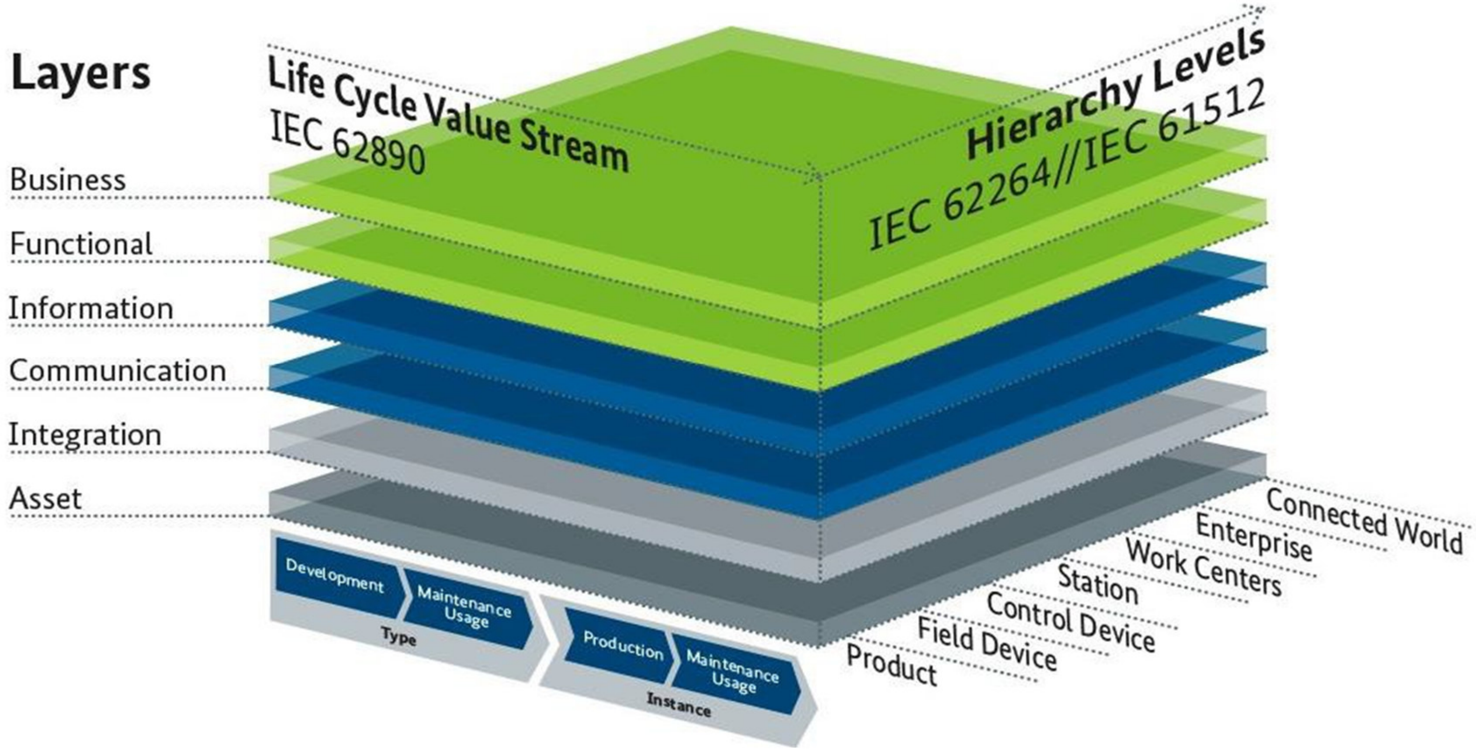


Figure 3. Four important aspects of Industrie 4.0 (Sources: Siemens, Festo)
Copyright „Umsetzungsstrategie Industrie 4.0 - Ergebnisbericht, Berlin, April 2015"

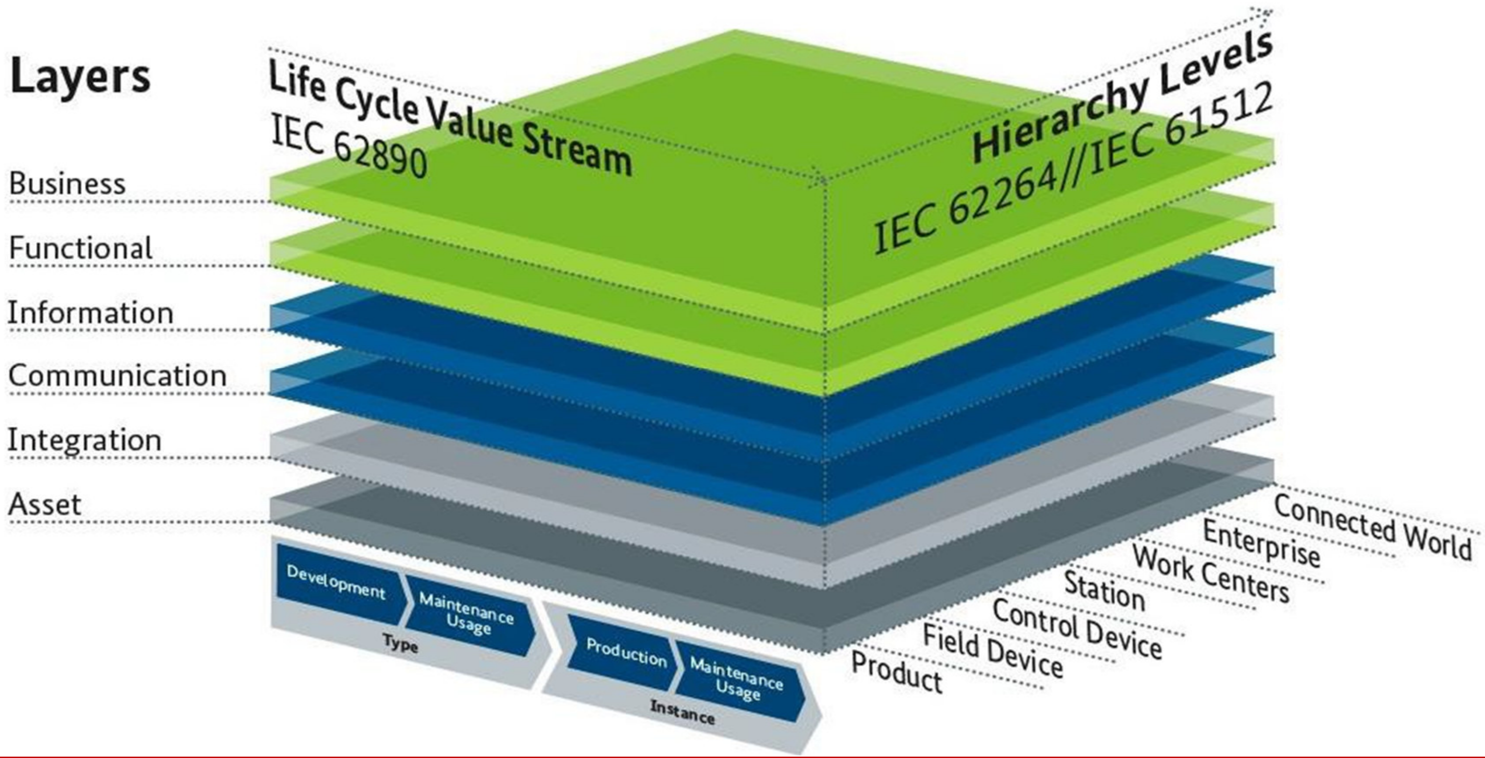
RAMI 4.0

- RAMI 4.0 is a Service-Oriented Architecture (SOA) and combines all items and IT components in a layered and life cycle model
- RAMI 4.0 goal is to break down complex processes into easy-to-grasp packages, including data privacy and IT security



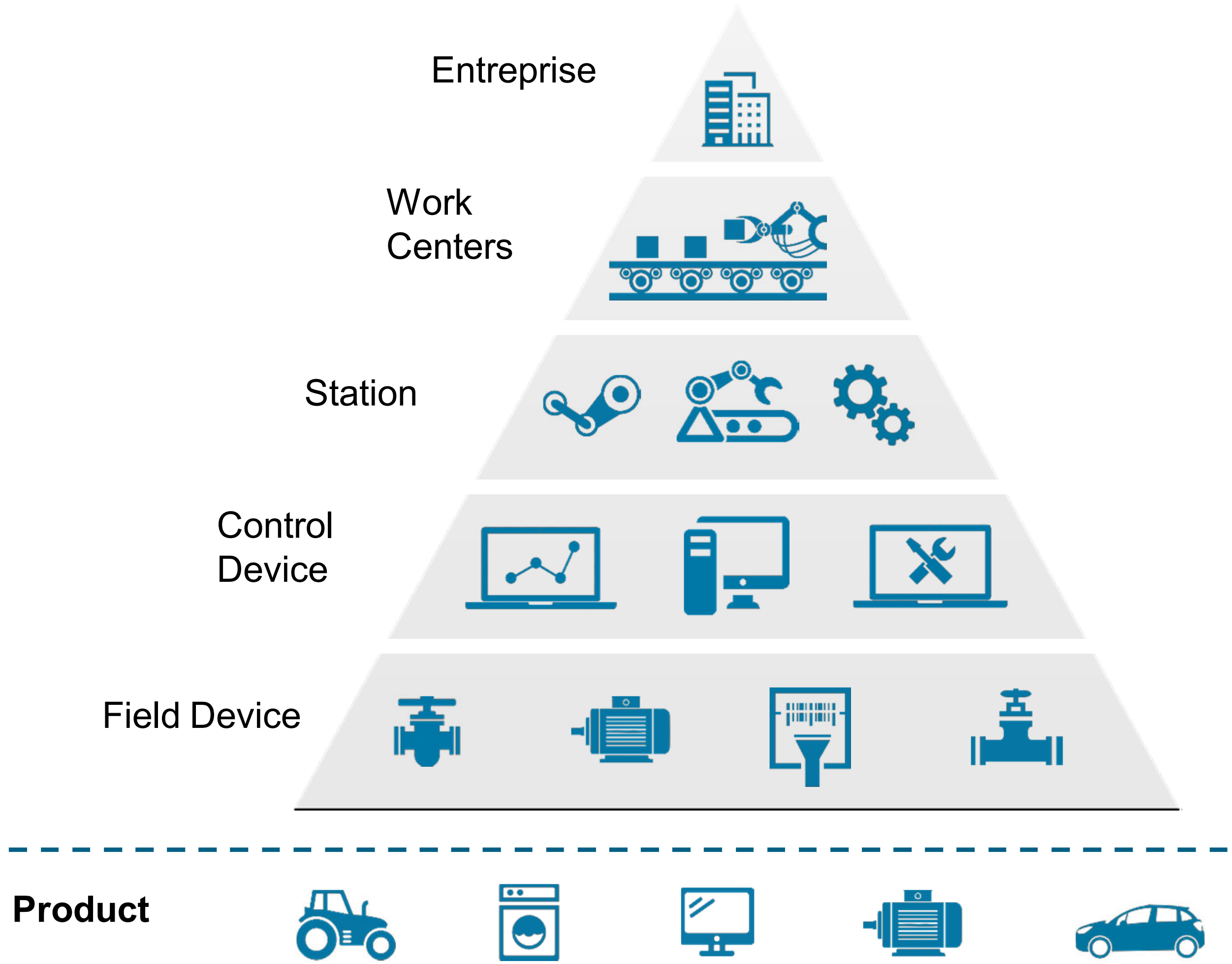
RAMI 4.0: Three Dimensions

- RAMI 4.0 is a three dimensional architecture:
  - layers in the vertical map: Business, Functional, Information, Communication, Integration, Asset
  - value streams for life cycle (left horiz.): type and instance
  - hierarchy levels for the functional assignments (right horiz.)



Old World: Industry 3.0

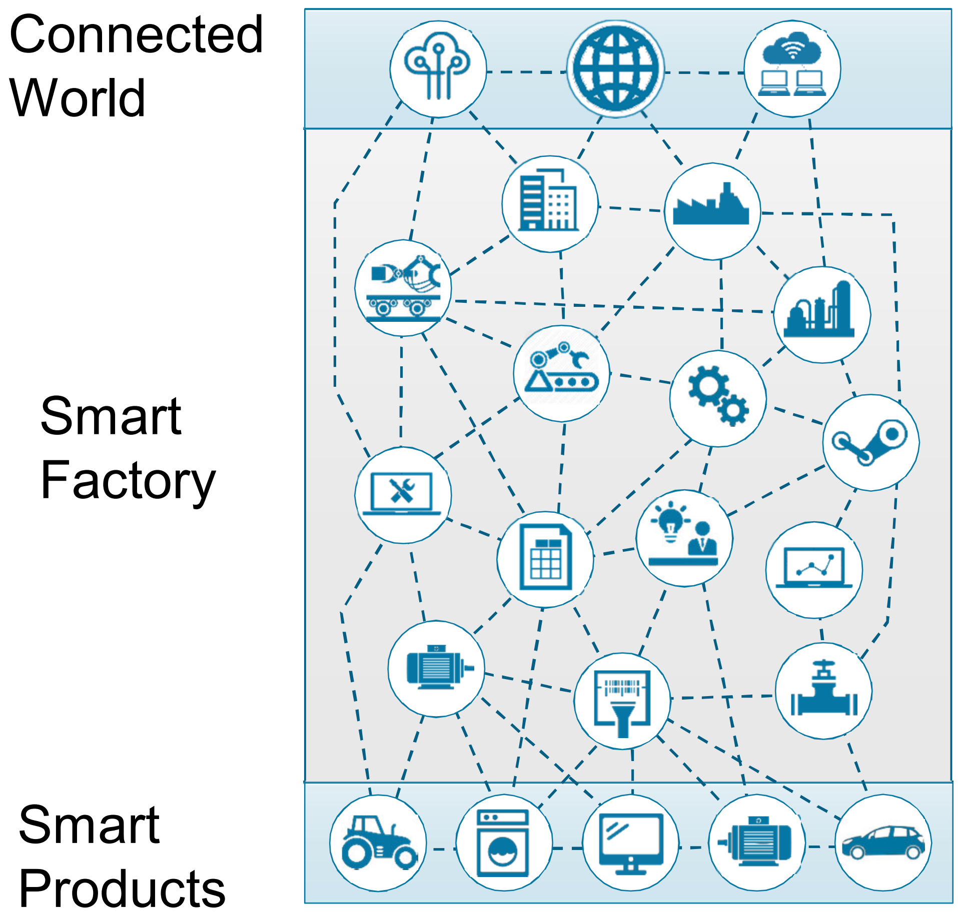
- Hardware-based structure
- Functions are bound to hardware
- Hierarchy-based communication
- Products are isolated



Graphics © Plattform Industrie 4.0

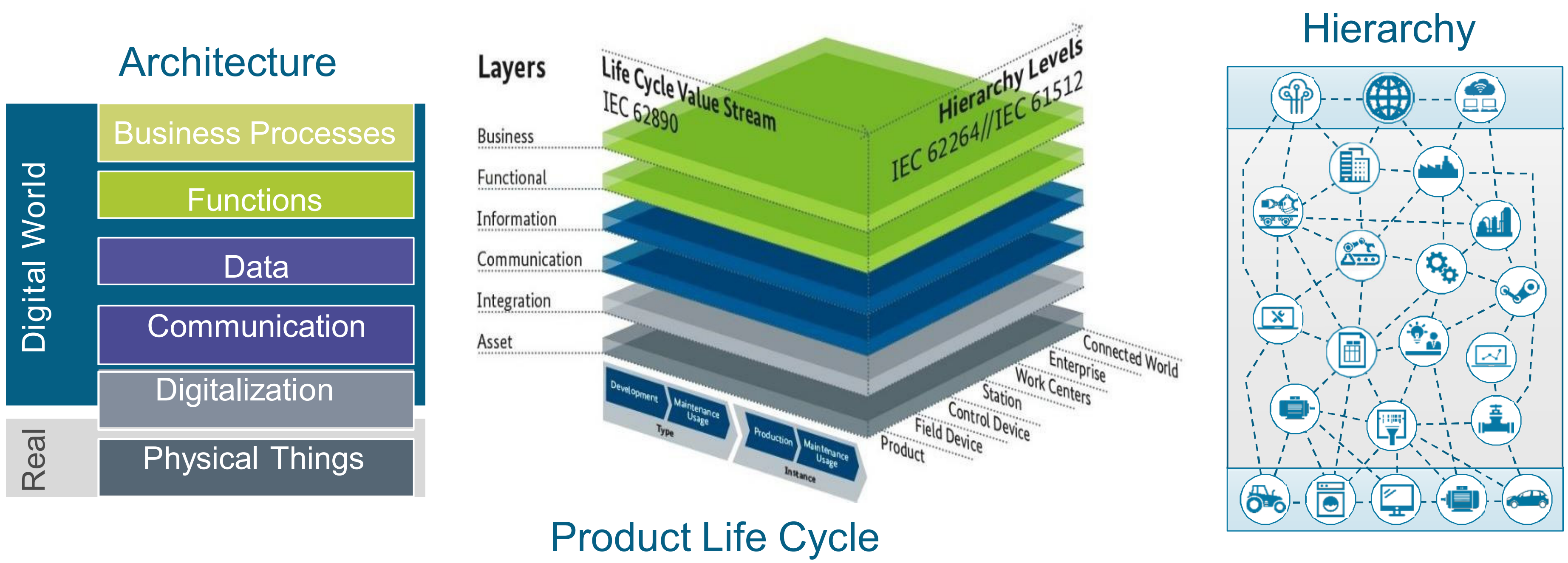
New World: Industry 4.0

- Centrality of products:
  - flexible systems and machines
  - functions are distributed throughout the network
  - participants interact across hierarchy levels
  - communication among all participants
  - product is part of the network



Graphics © Plattform Industrie 4.0

RAMI 4.0 Architecture

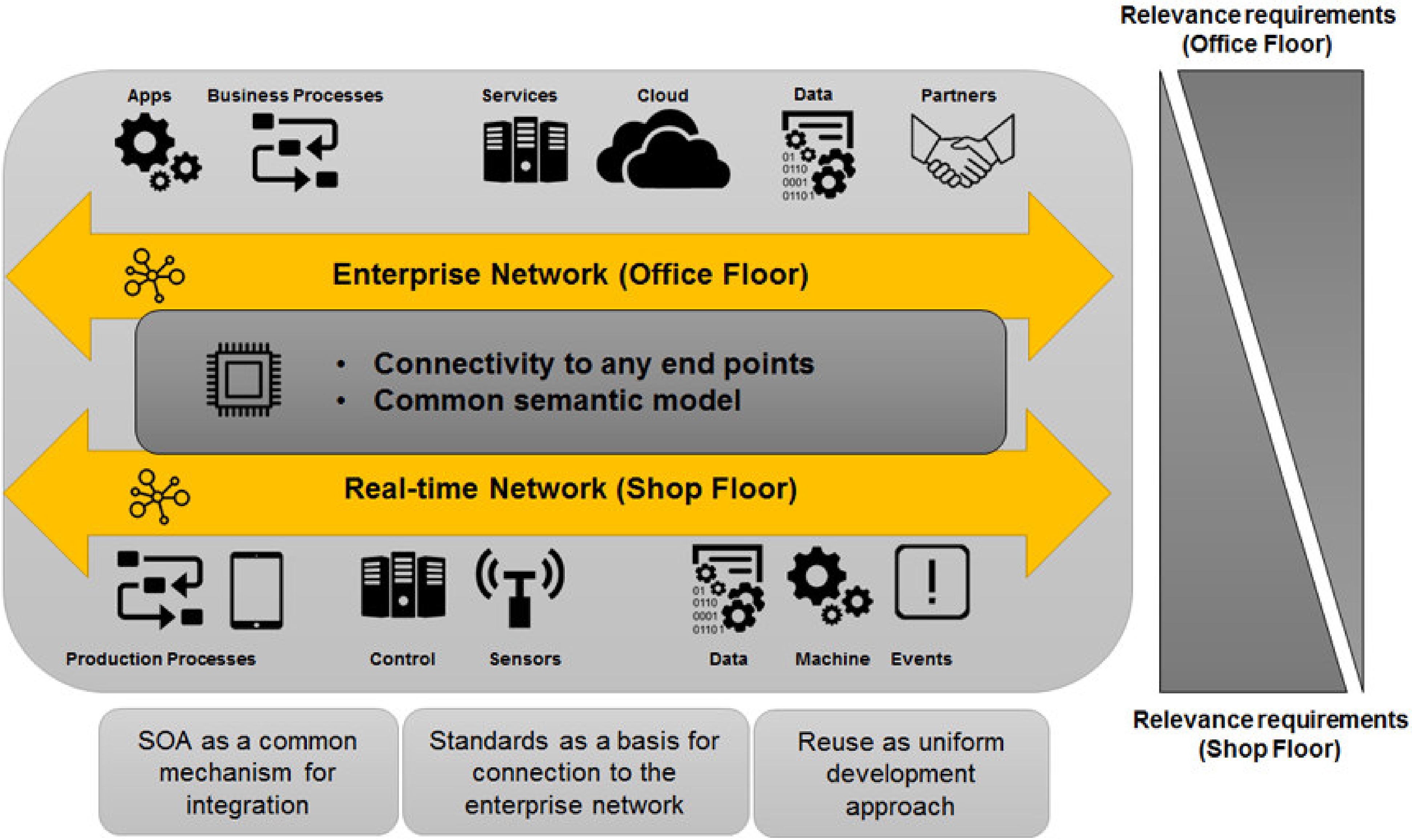


- A Solution Space with a Coordinate System for Industrie 4.0

I4.0 Components in RAMI 4.0

- To define the industry scenario, we traditionally refer to "office floor" and "shop floor", where entities were in one or the other
- Modern I4.0 blurs the distinction: an I4.0 component can be a production system, an individual machine or station, or an assembly inside a machine
- I4.0 components
  - have certain common properties independently of the level
  - obey requirements for connectivity to any time
  - exhibit a common semantic model
- Each I4.0 component moves along the life cycle of the factory from the office floor and shop floor and in contact with such central and significant factory systems as the PLM (Product Lifecycle Management), ERP (Enterprise Resource Planning), and Industrial Control and Logistics systems

I4.0 Components and Areas



I4.0 Components Connections

- First basic requirement
  - a network of I4.0 components must be structured in such a way that **connections between all I4.0 components must be possible**
  - all I4.0 components and their contents must follow a **common semantic model**
- Second basic requirement
  - the concept of an I4.0 component must be capable of be expressed so that it can **meet requirements with different focal areas**, i.e., “office floor” or “shop floor”
- Third basic requirement
  - any I4.0 component must be **associated with a virtual representation** that can be kept inside or in a higher level IT system

I4.0 Components Virtual Representation

- The virtual representation is the container of the data on the object
- Data can be
  - kept in the I4.0 component itself and made available to the outside world by I4.0 compliant communication, or
  - stored in a (higher level) IT system which makes them available to the outside world by I4.0 compliant communication

- In the reference architecture model RAMI4.0, the virtual representation takes place in the Information Layer
- I4.0 compliant communication is thus of great importance

Outline

3. Industry 4.0
   - introduction and definition
   - Industry 4.0 and the digital twin

4. Examples of IoT and Industry 4.0 solutions
   - from e-Maintenance to i-Maintenance
   - IoT-enabled use cases
   - experience with a business case

Industry 4.0 and the Digital Twin

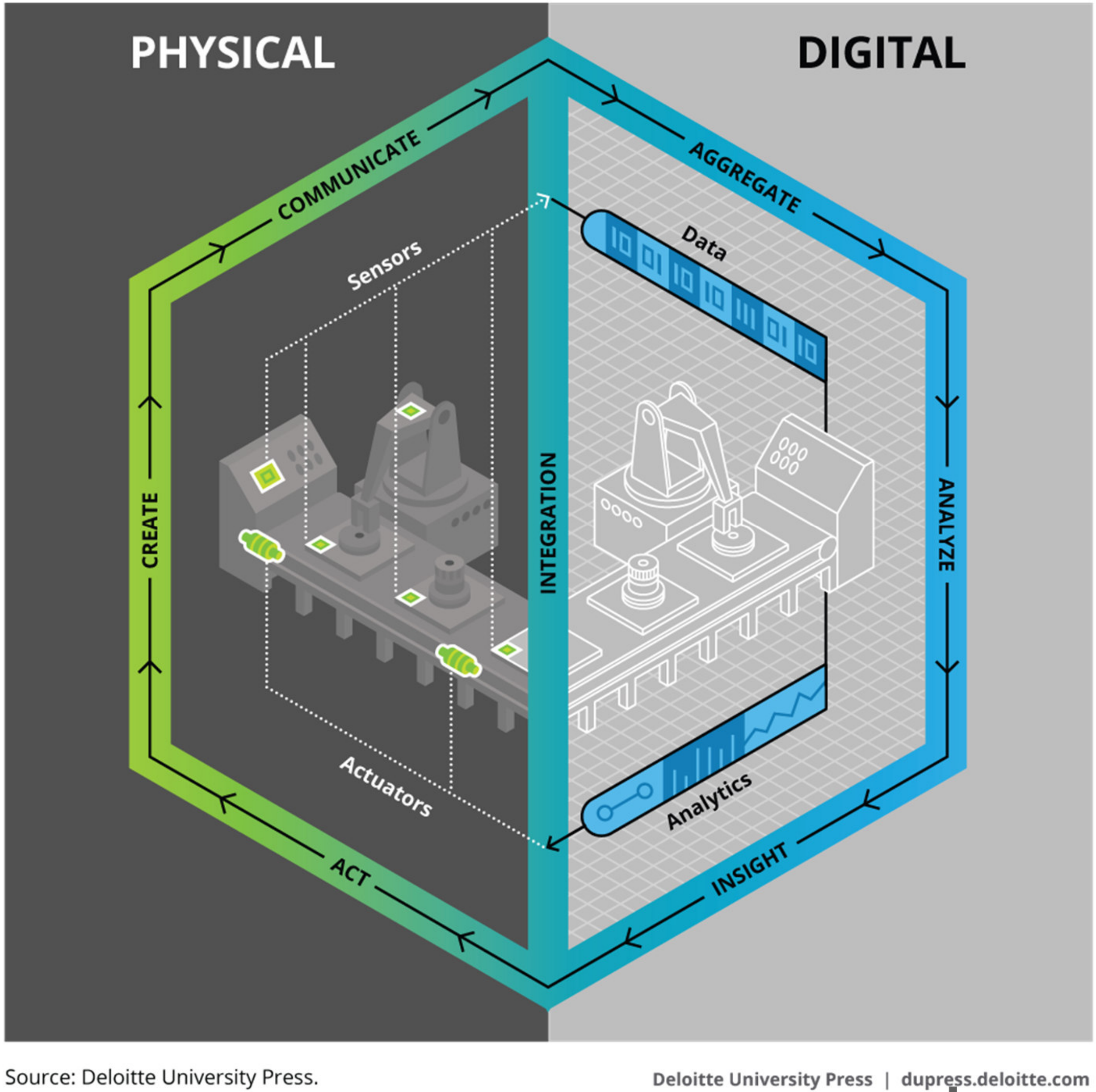
- A near-real-time **evolving digital profile** of the historical and current behavior of a **physical object or process** that helps
  - **optimize business performance**
  - identify **intolerable deviations** from optimal conditions

- A complete digital footprint of products allows to
  - **detect** physical issues sooner
  - **predict** outcomes more accurately
  - **build** better products

- Based on
  - the **tight integration** of Information Technology (IT) and Operations Technology (OT)
  - **massive, cumulative, real-time, real-world data** measurements across an array of dimensions

Model of a Digital Twin

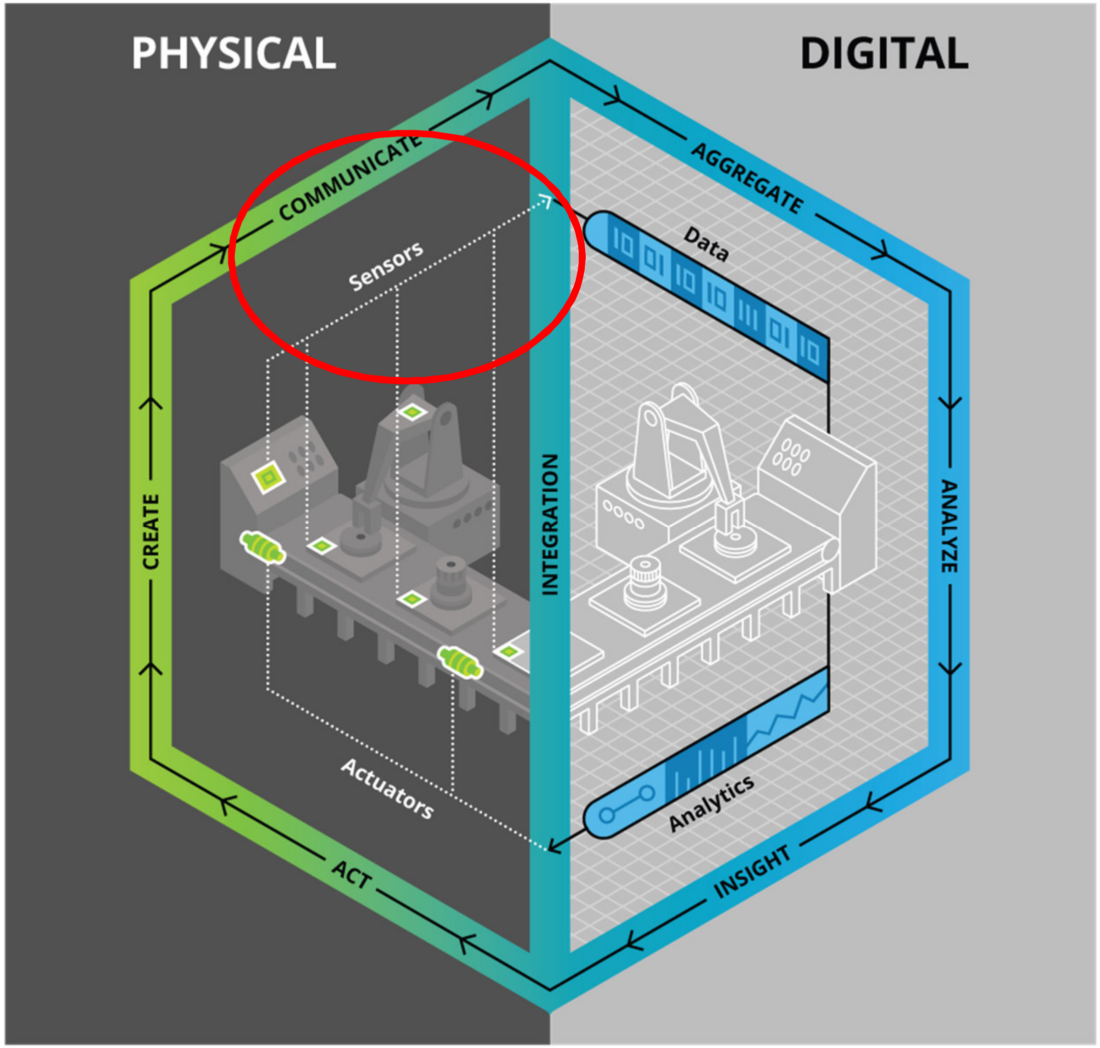
- Physical vs. Digital
- Constituent elements
  - sensors
  - data
  - integration
  - analytics
  - actuators
- Six steps
  - create
  - communicate
  - aggregate
  - analyze
  - insight
  - act



Source: Deloitte University Press.

Constituent elements of Digital Twin: Sensors

- Distributed throughout the manufacturing process
- Create signals that enable the twin to capture operational and environmental data pertaining to the physical process in the real world



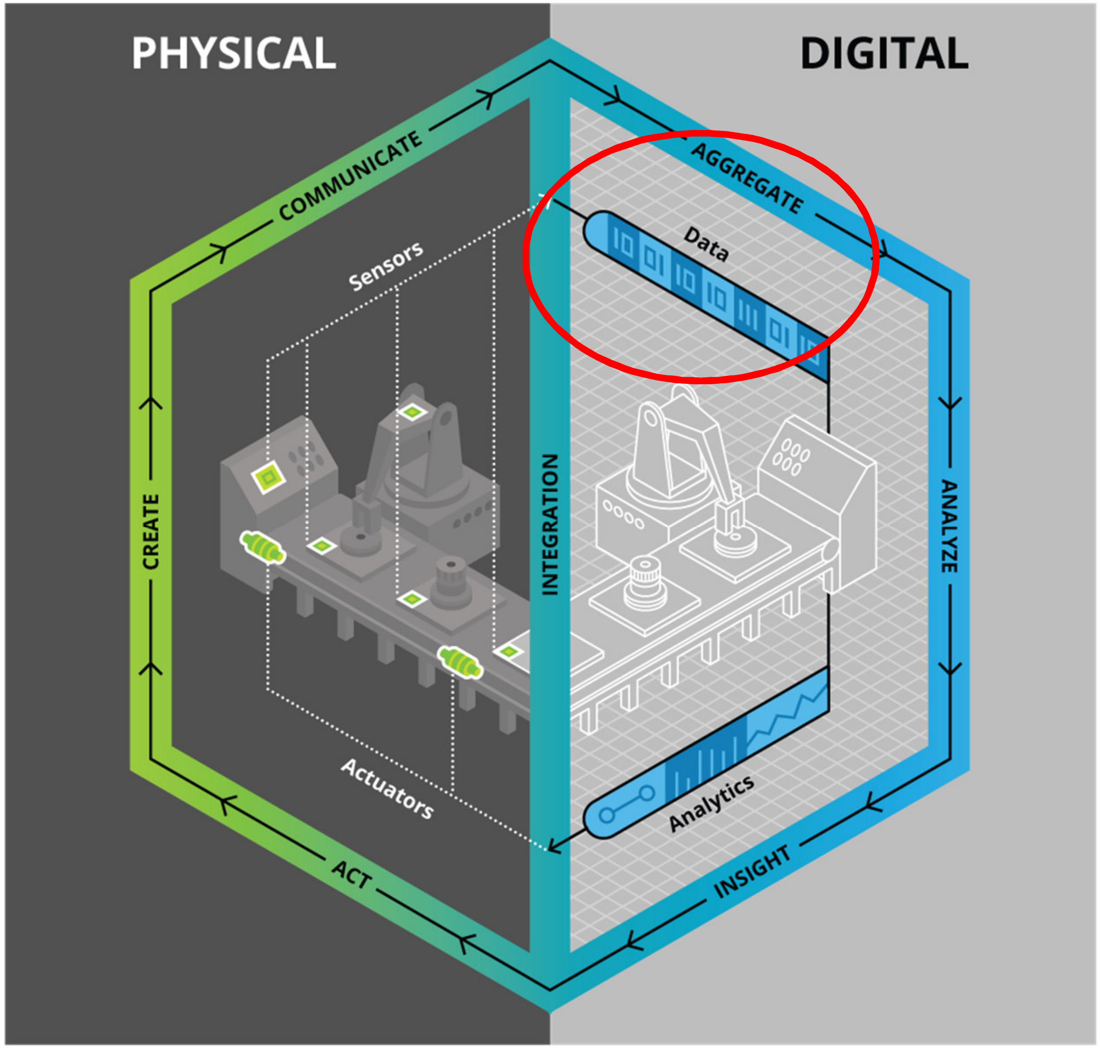
Source: Deloitte University Press.

Deloitte University Press | dupress.deloitte.com

Constituent elements of Digital Twin: Data

- Real-world operational and environmental data from the sensors are aggregated and combined with data from the enterprise, such as the Bill Of Materials (BOM), enterprise systems, and design specifications

- Data may also contain other items such as engineering drawings, connections to external data feeds, and customer complaint logs



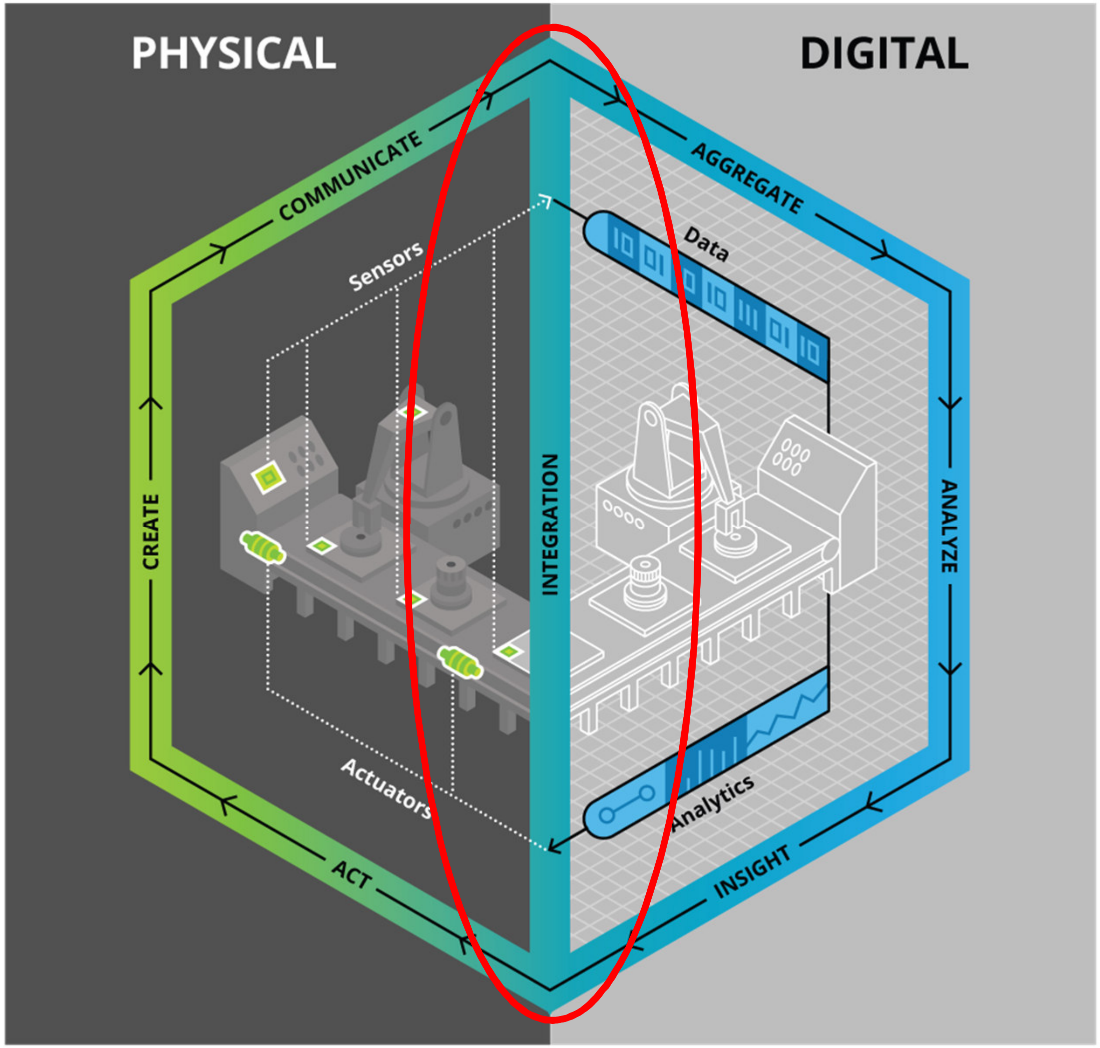
Source: Deloitte University Press.

Deloitte University Press | dupress.deloitte.com

Constituent elements of Digital Twin: Integration

- Sensors communicate the data to the digital world through integration technology between the physical world and the digital world, and vice versa

- Integration technology: edge devices, communication interfaces, security, … more in general, IoT!

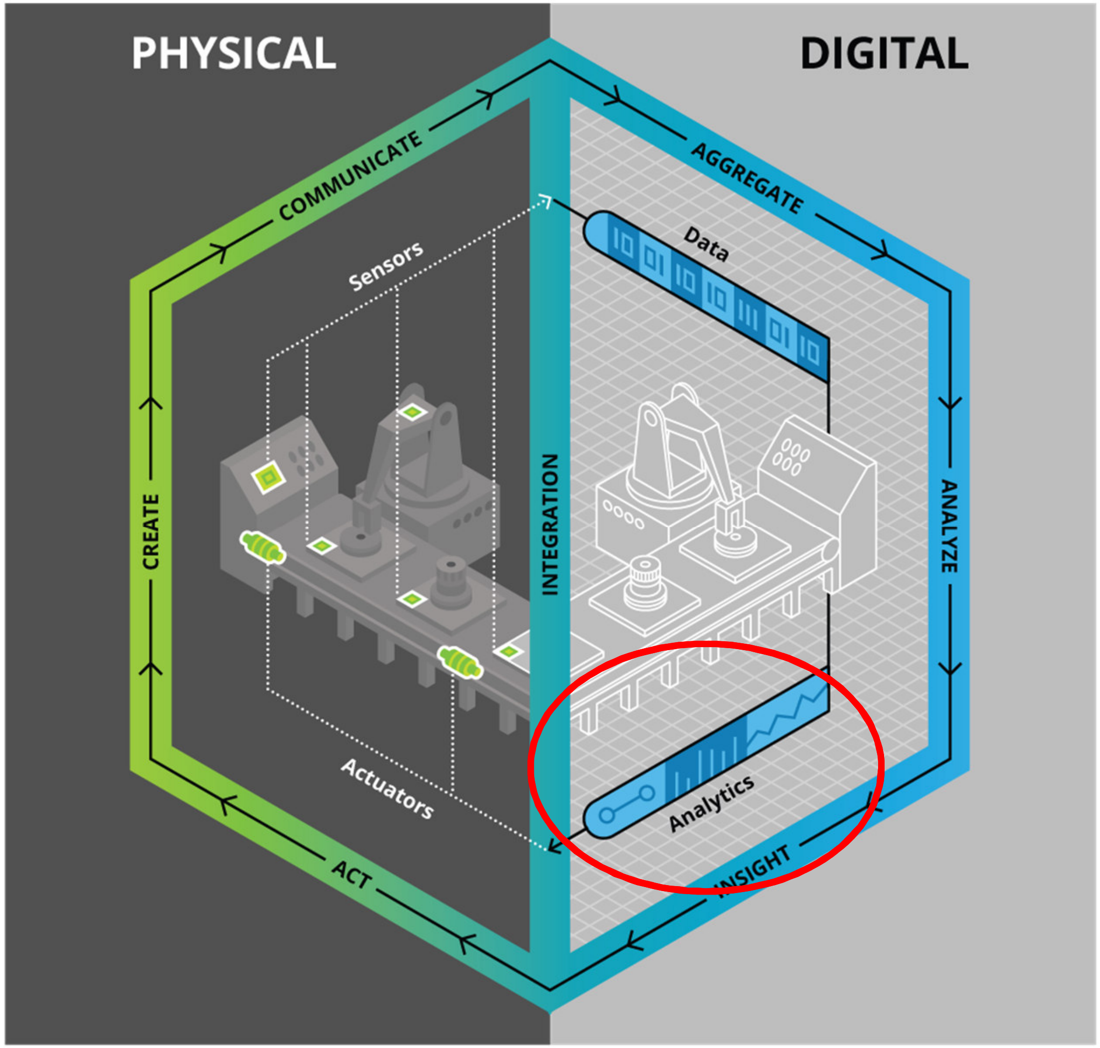


Source: Deloitte University Press.

Deloitte University Press | dupress.deloitte.com

Constituent elements of Digital Twin: Analytics

- Analytic techniques used to analyze the data through algorithmic simulations and visualization routines that are used by the digital twin to produce insights

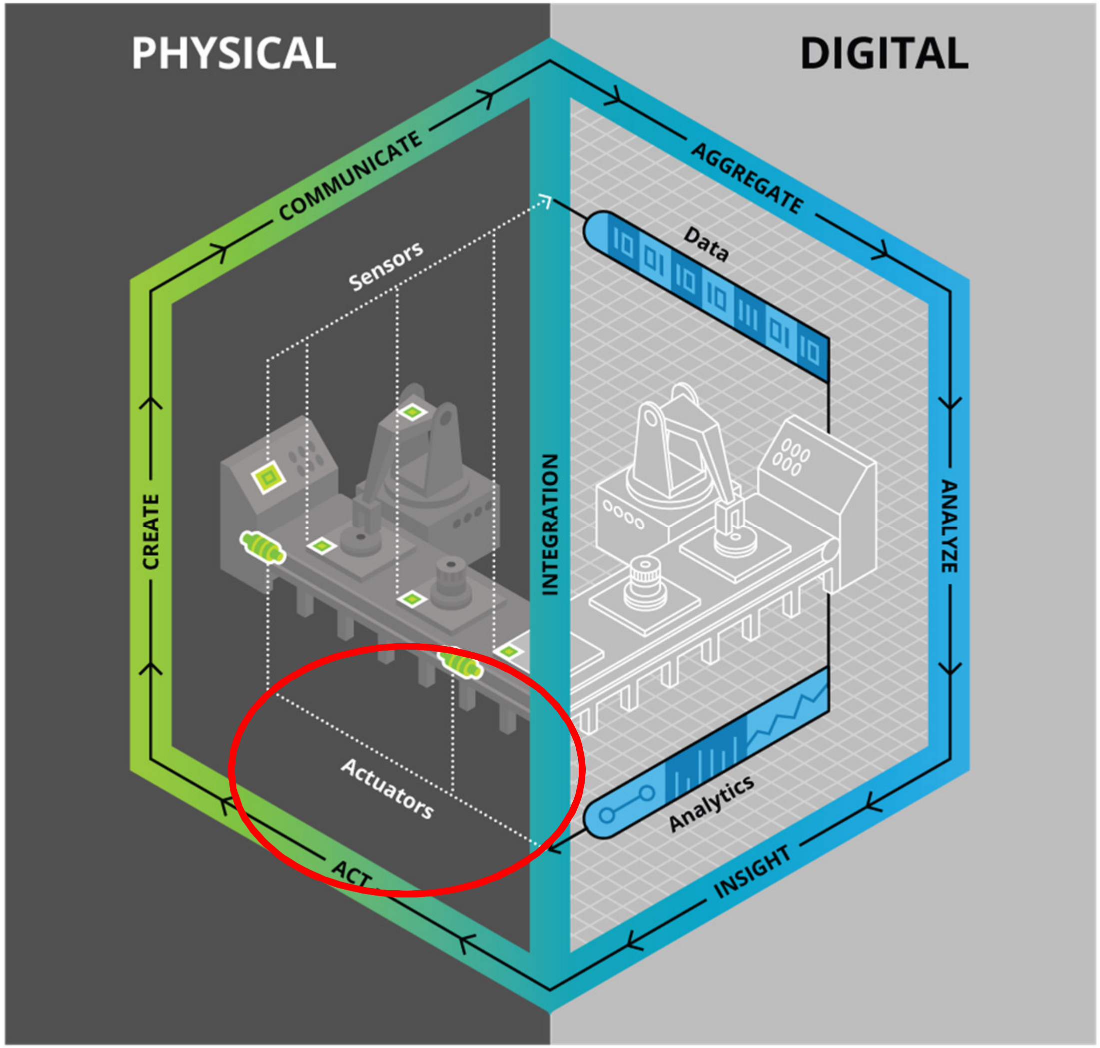


Source: Deloitte University Press.

Deloitte University Press | dupress.deloitte.com

Constituent elements of Digital Twin: Actuators

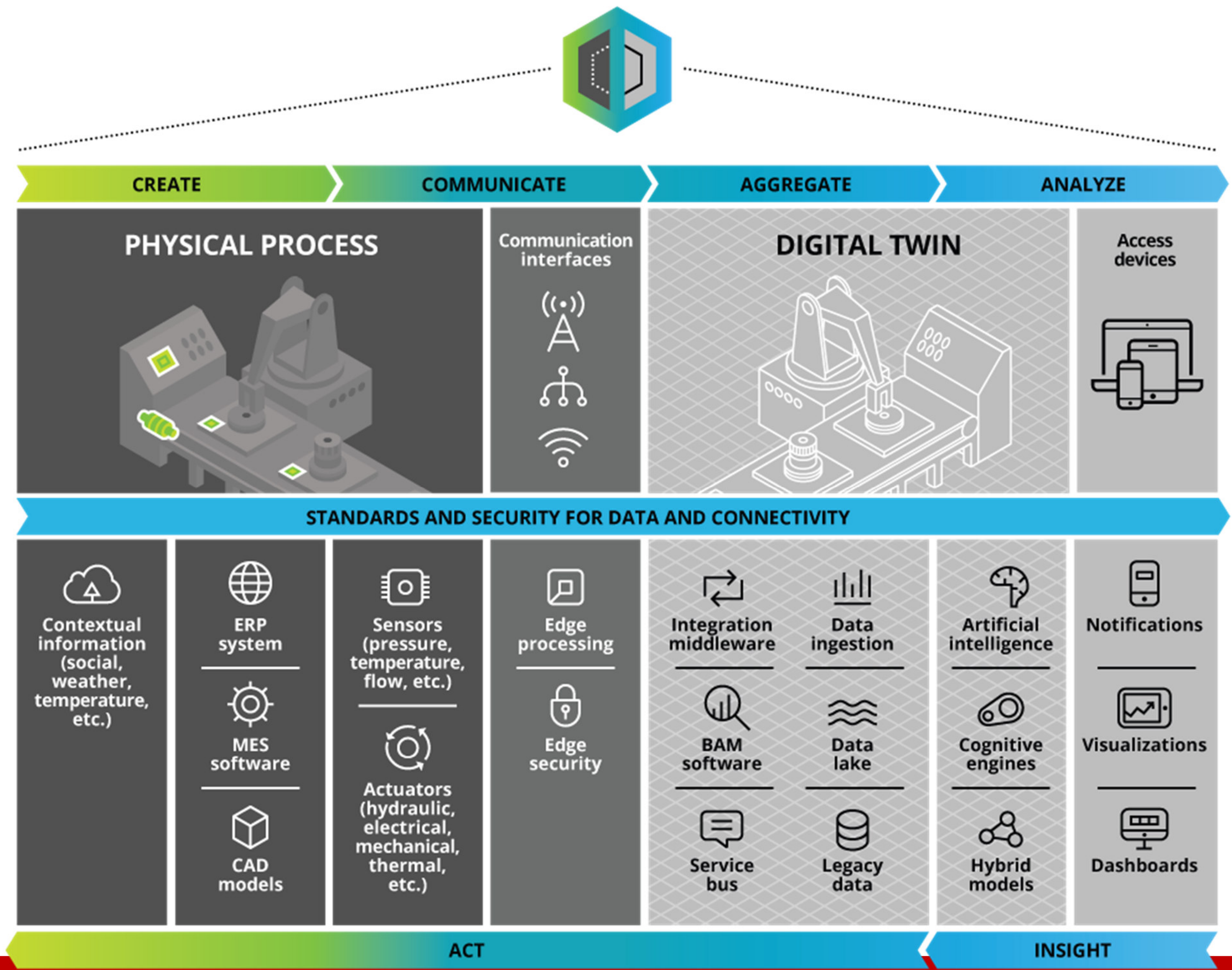
- Should an action be warranted in the real world, the digital twin produces the action by way of actuators, subject to human intervention, which trigger the physical process



Source: Deloitte University Press.

Deloitte University Press | dupress.deloitte.com

Digital Twin Conceptual Architecture

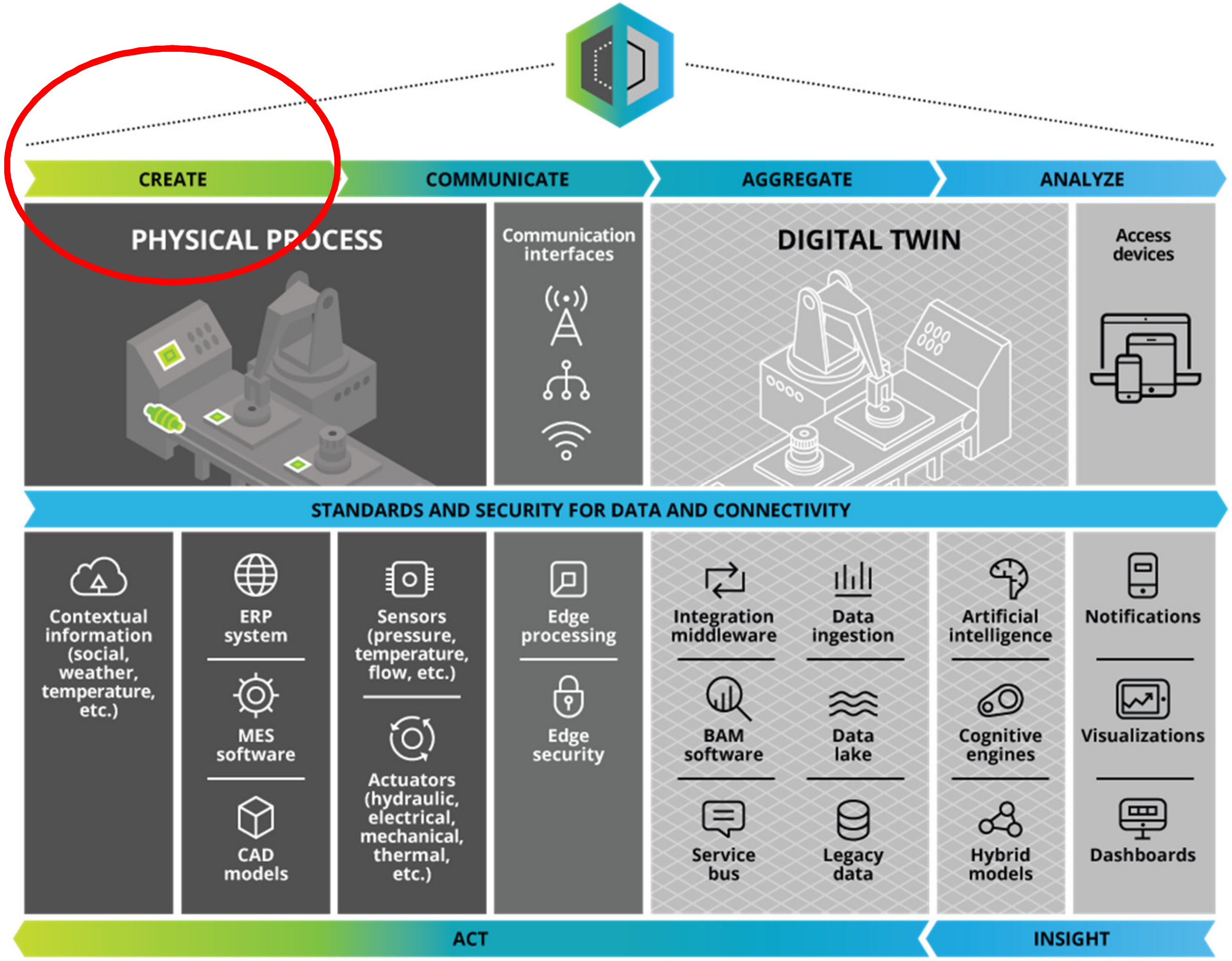


Source: Deloitte University Press.

Deloitte University Press | dupress.deloitte.com

Steps of Digital Twin: 1) Create

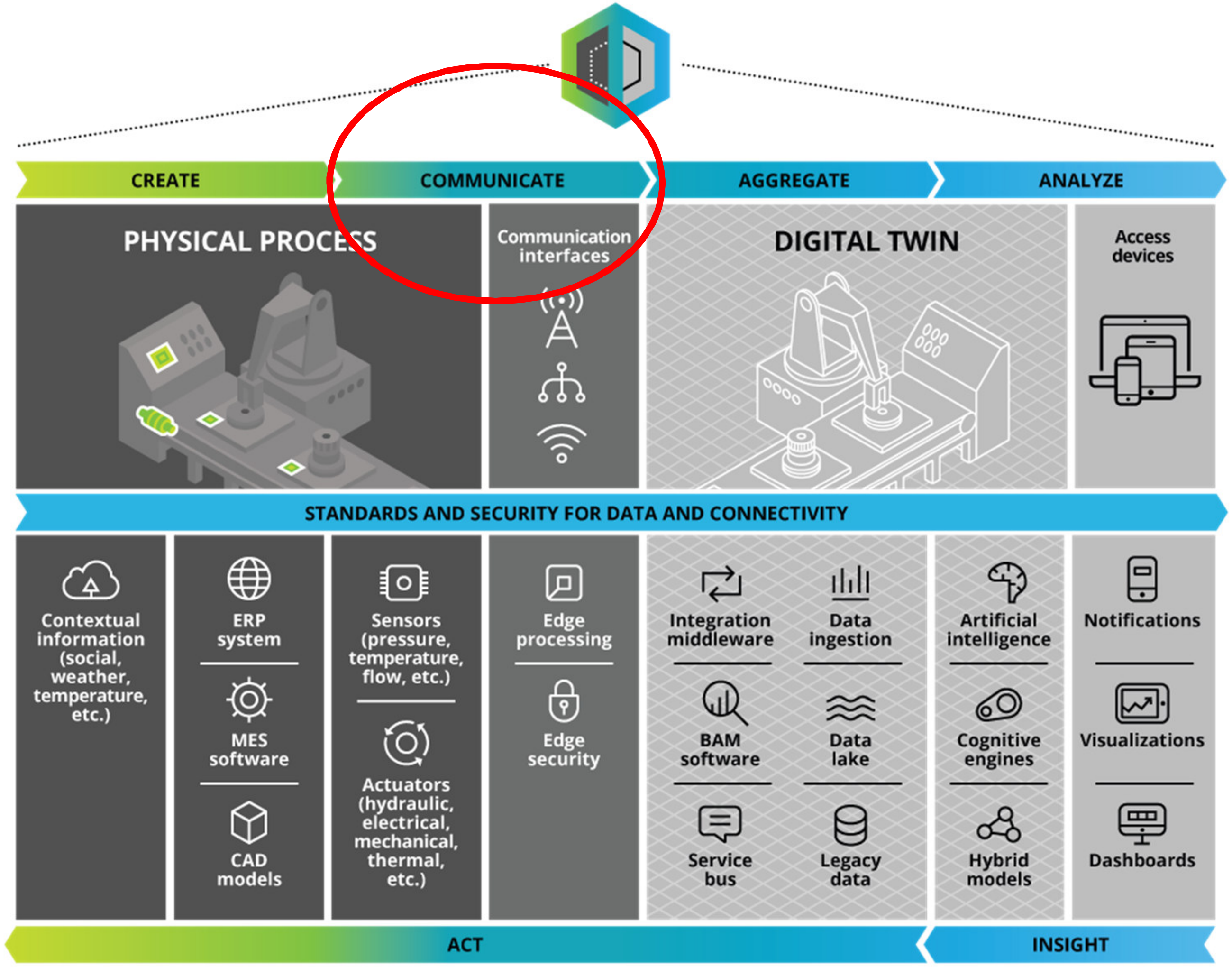
- The create step encompasses outfitting the physical process with myriad of sensors that measure critical inputs from the physical process and its surroundings.



Source: Deloitte University Press.

Steps of Digital Twin: 2) Communicate

- The communicate step helps the seamless, real-time, bidirectional integration and connectivity between the physical process and the digital platform



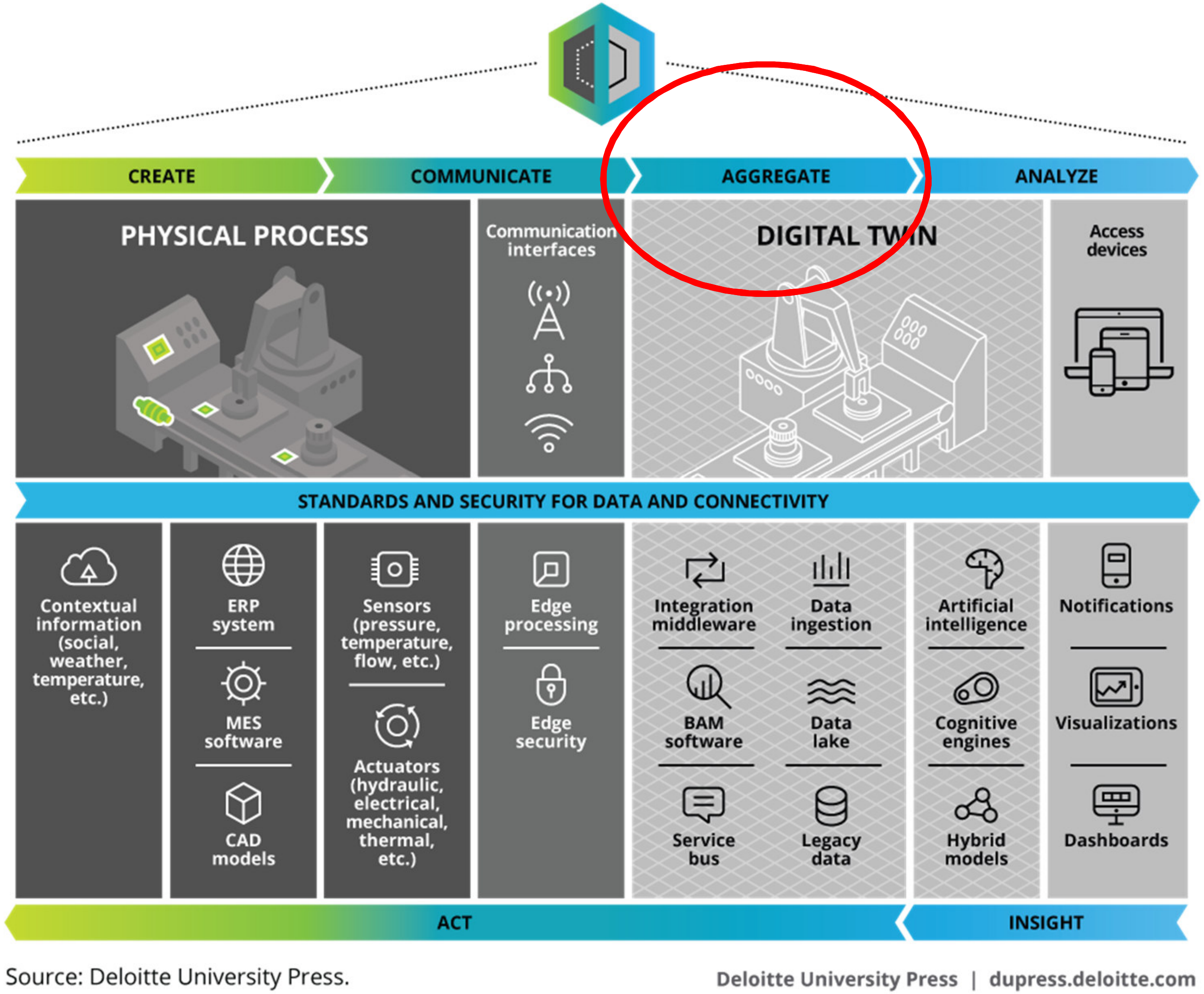
Source: Deloitte University Press.

Deloitte University Press | dupress.deloitte.com

Steps of Digital Twin: 3) Aggregate

- The aggregate step can support data ingestion into a data repository, processed and prepared for analytics

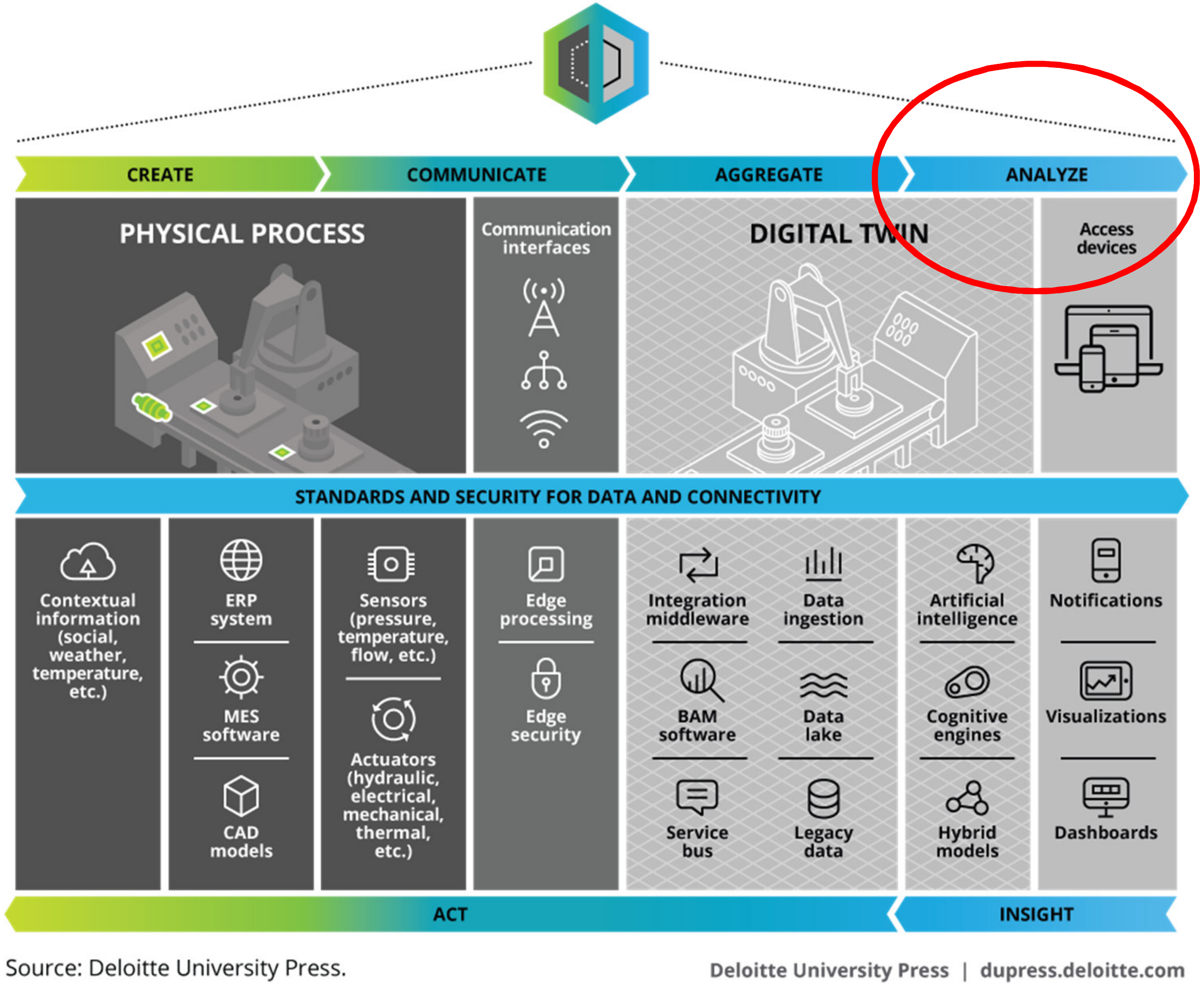
- The data aggregation and processing may be done either on the premises or in the cloud



Source: Deloitte University Press.

Steps of Digital Twin: 4) Analyze

- In the analyze step, data is analyzed and visualized
- Data scientists and analysts can utilize advanced analytics platforms and technologies to develop iterative models that generate insights and recommendations and guide decision making

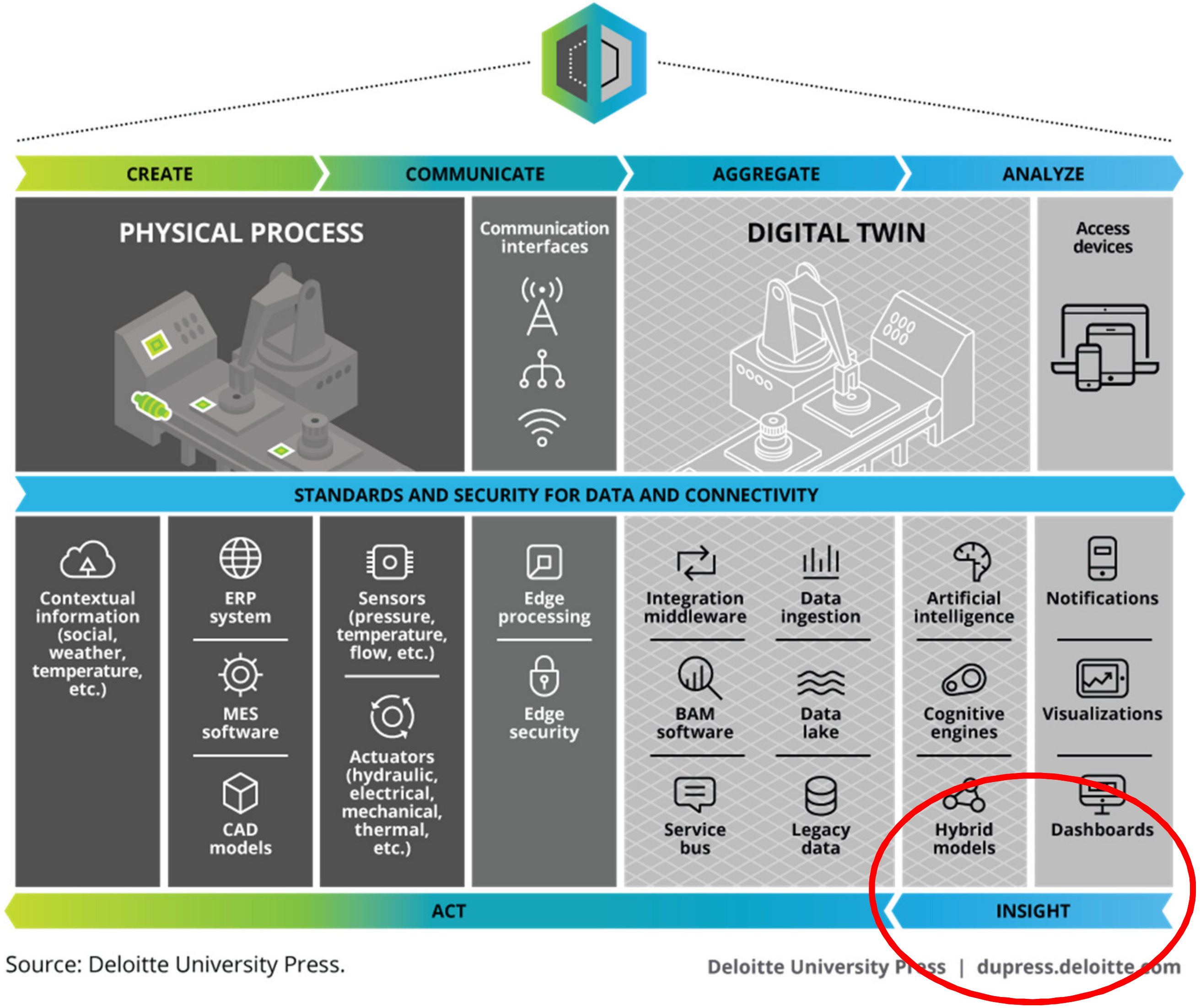


Source: Deloitte University Press.

Steps of Digital Twin: 5) Insight

- In the insight step, insights from the analytics are presented through dashboards with visualizations

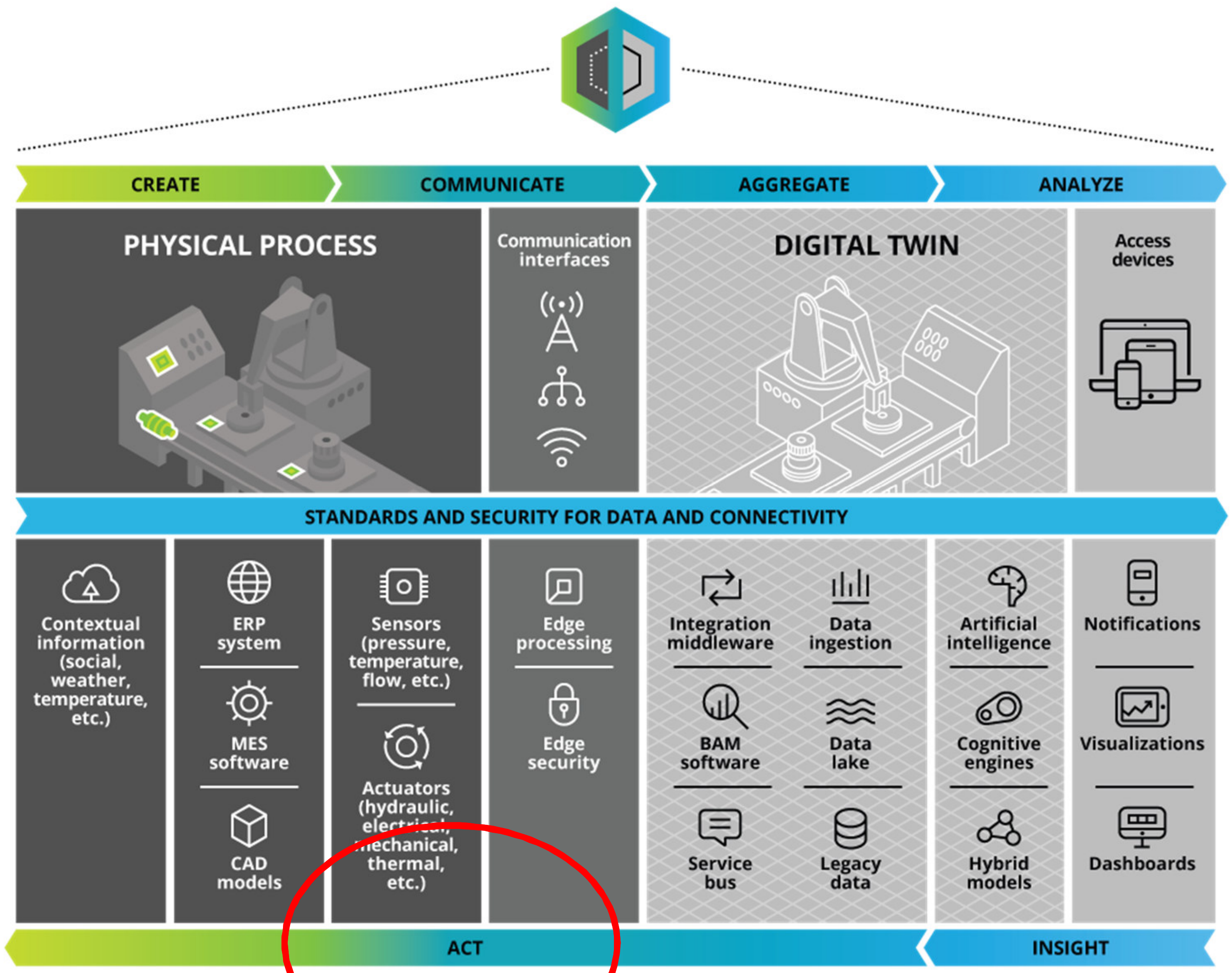
- Goal of highlighting unacceptable differences in the performance of the digital twin model and the physical world analogue, indicating areas that need investigation



Source: Deloitte University Press.

Steps of Digital Twin: 6) Act

- The act step is where actionable insights from the previous steps can be fed back to the physical asset and digital process to achieve the impact of the digital twin



Source: Deloitte University Press.

Digital Twins: Use Cases

- Digital twins help manufacturers and engineers by:
  - visualizing products in use, by real users, in real-time
  - building a digital thread, connecting disparate systems and promoting traceability
  - refining assumptions with predictive analytics
  - troubleshooting far away equipment
  - managing complexities and linkage within systems-of-systems

Outline

3. Industry 4.0
   - introduction and definition
   - IIRA and RAMI 4.0 reference architectures
   - Industry 4.0 and the digital twin

4. Examples of IoT and Industry 4.0 solutions
   - from e-Maintenance to i-Maintenance
   - IoT-enabled use cases
   - experience with a business case

From e-Maintenance to i-Maintenance

- From Maintenance to e-Maintenance
- Some e-Maintenance Products (and Services – Servitization)
- ICT technologies and e-Maintenance
- From e-Maintenance to i-Maintenance
- The Carpigiani i-Maintenance Solution

From Maintenance to e-Maintenance

- Maintenance
  - the process that preserve the correct working conditions of factories, vehicles, automated machines, appliances…
  - Maintenance requires skills, techniques, methods and theories to develop technical and organizational solutions to ensure the correct performing of the required functions in industrial equipment

- The evolution of Maintenance
  - Total Productive Maintenance (TPM) (Nakajima, 1988)
  - Reliability Centred Maintenance (RCM) (Moubray, 1997)
  - Condition Based Maintenance (CBM) (CrespoMarquez, 2007)
  - implemented in industry and process plants with excellent results

The evolution of Maintenance

- RCM is a technology centric process, a highly structured method for the planning of maintenance, originated from a critical analysis of maintenance requirements

- TPM (related to Total Quality Management) applies a continuous improvement Plan-Do-Check-Act cycle (Deming Cycle)
  - machine operators are involved in the maintenance process
  - operators take over some of the maintenance day-to-day tasks, for example, cleaning, lubricating…

- CBM permits to plan maintenance interventions when needed, by measuring physical (temperature, pressure, vibrations, etc.) and functional working parameters to determine the current status of an equipment
  - CBM moves from preventive maintenance to predictive maintenance

What is e-Maintenance?

- In the Condition Based Maintenance (CBM) area, e-Maintenance refers to the introduction of Information and Communication Technologies (ICT) in the maintenance process

e-Maintenance = CBM + ICT

- It is a technology-led (ICT) innovation, that opens up to new maintenance solutions
- E-maintenance platforms provides technicians with powerful maintenance tools, such as monitoring, diagnosis, and prognosis
- Towards the proactive maintenance in the digital factory

e-Maintenance: Functions and Goals

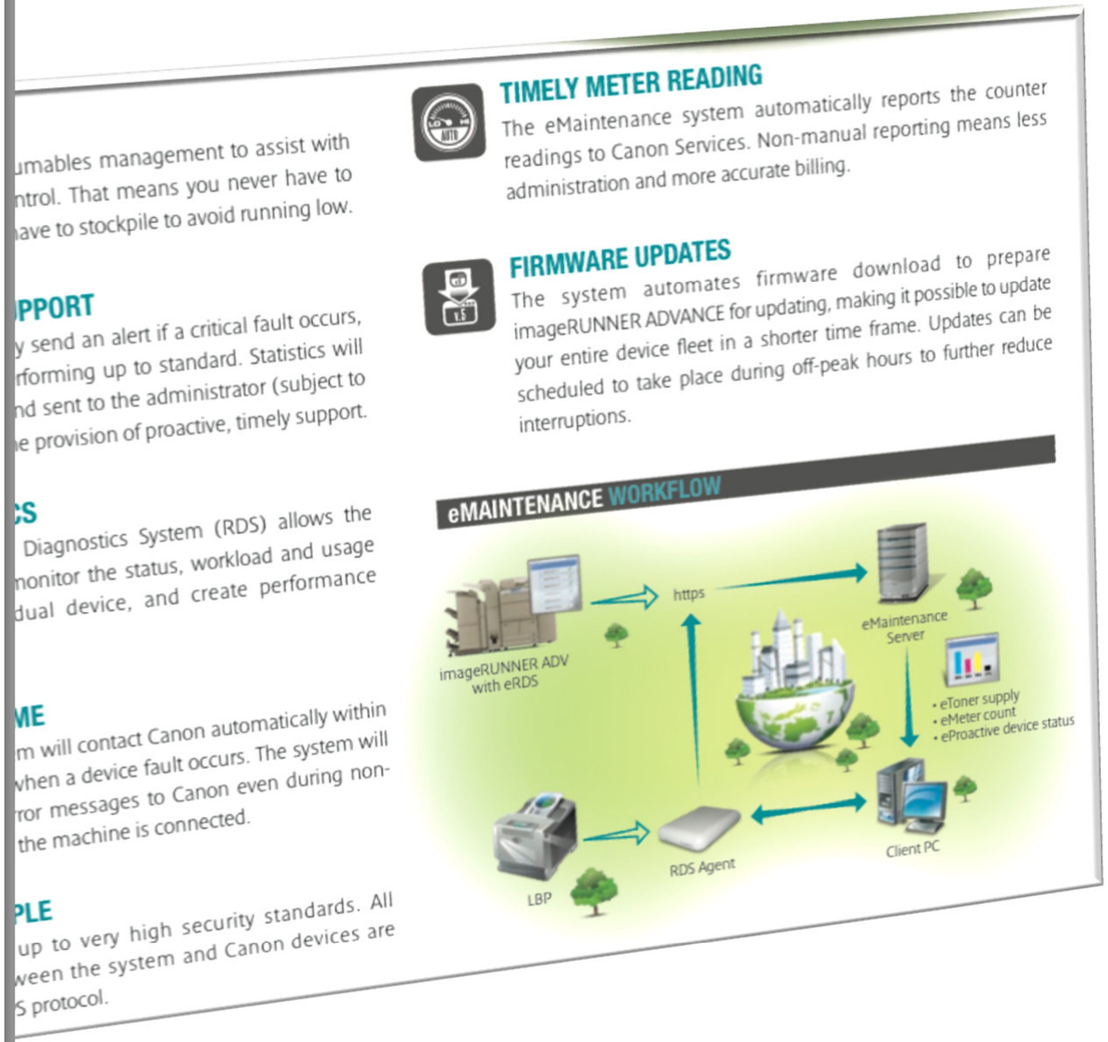
- e-Maintenance **functions**
  - remote monitoring
  - remote configuration and management
  - diagnosis and prognosis

- e-Maintenance **goals**
  - **reduce cost** of maintenance intervention (first-time fix, phone fix)
  - improve equipment **uptime**
  - supply chain **optimization**
  - better management of (highly skilled) technicians
  - technical know-how sharing
  - maintenance-data integration in enterprise IT systems (ERP, CRM,...)
  - new business models toward servitization

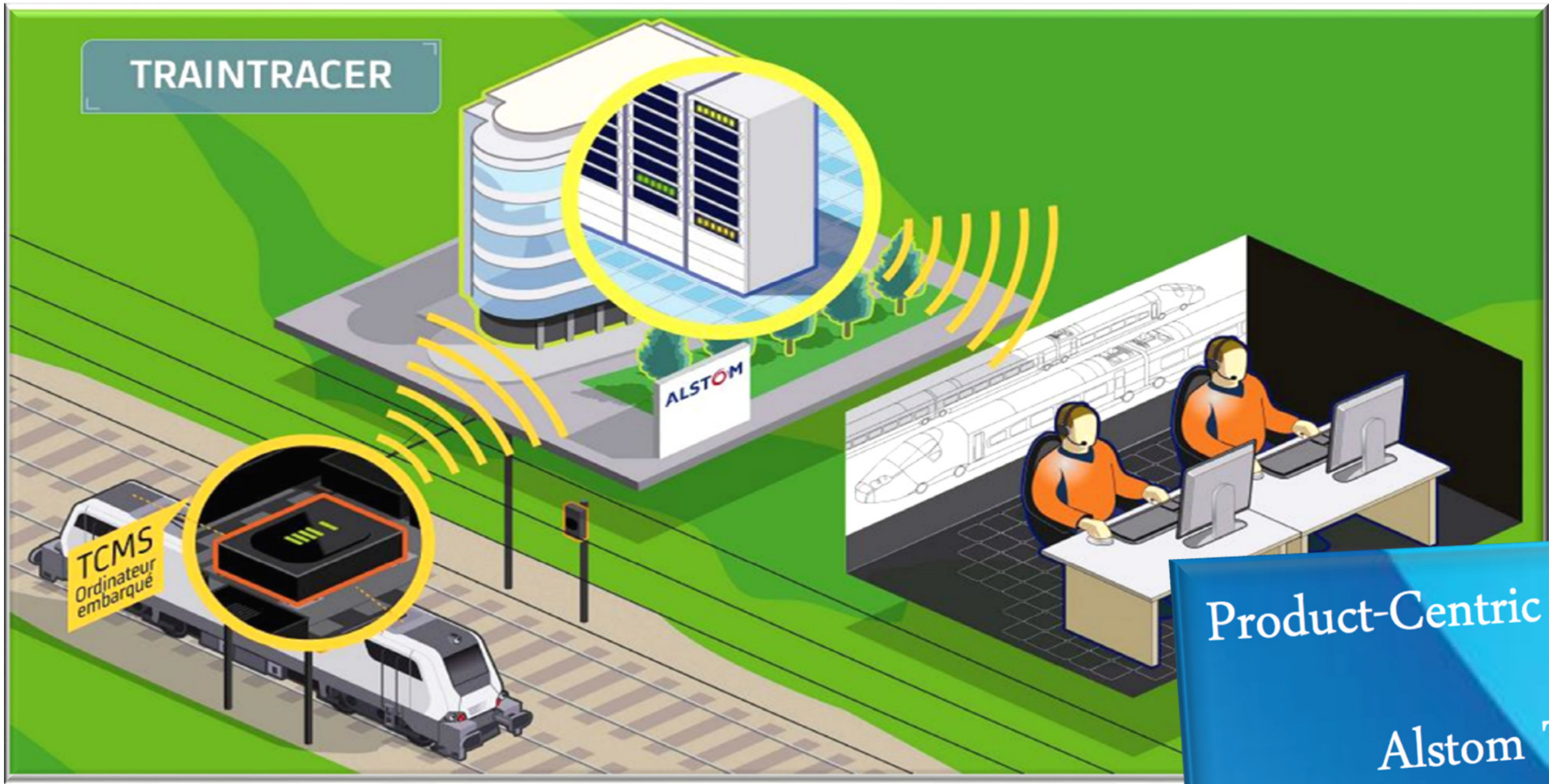
e-Maintenance: where does it come from?

- e-Maintenance evolves from the CBM of large plants, mainly for heavy industries and power plants, but also train
- Those solutions usually were based on proprietary (and very costly) Supervisory Control And Data Acquisition (SCADA), with:
  - ad-hoc and proprietary communication stack (very expensive)
  - specific-purpose hardware (very expensive)
  - proprietary software solution (often built from scratch, with no reuse allowed, no platform and technologies to build upon, very expensive)
  - with poor security features

The Canon e-Maintenance Solution



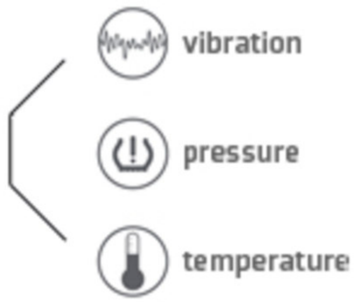
The Alstom e-Maintenance Solution



The Caterpillar e-Maintenance Solution

- CAT monitors vehicles: position, fuel, fluids

The Nespresso e-Maintenance Solution

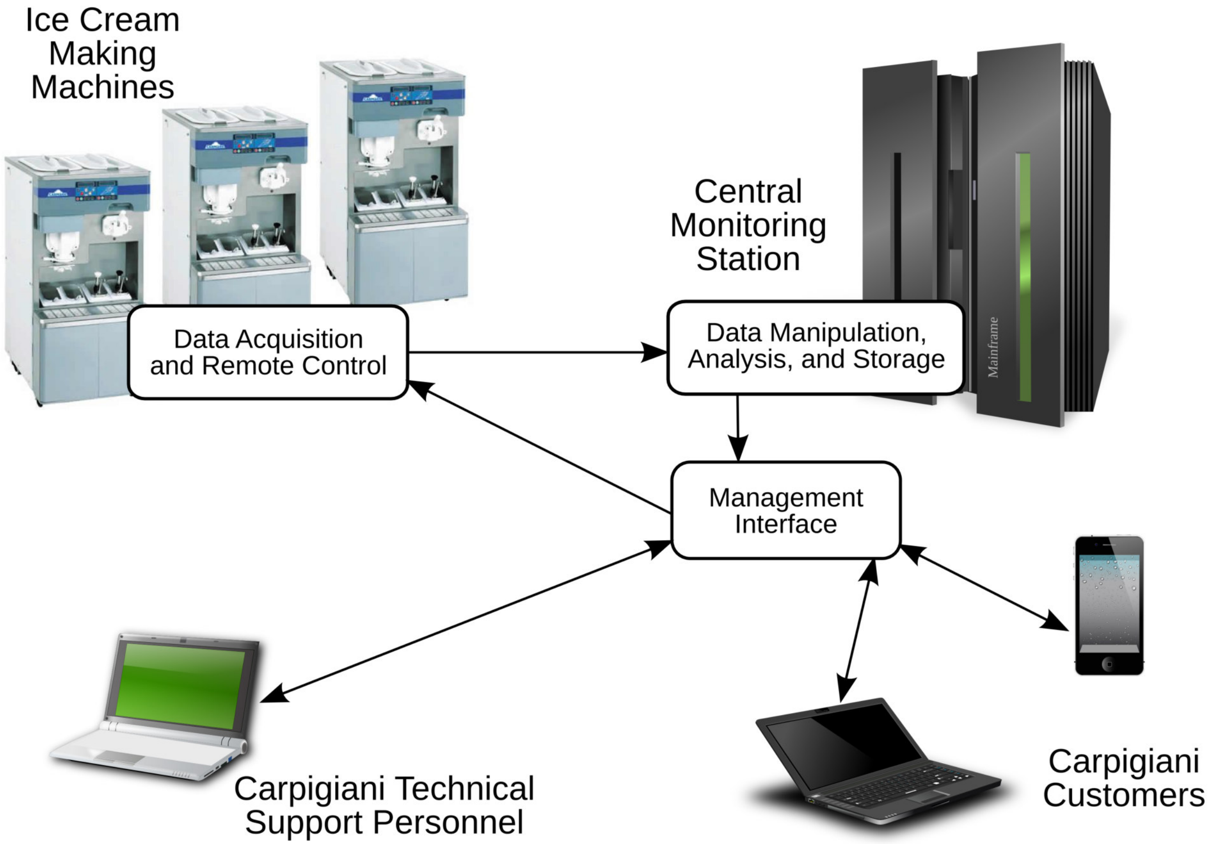


SIERRA WIRELESS

Nespresso + Sierra Wireless

- The digital kit inside the Nespresso machine

The Carpigiani e-Maintenance Solution



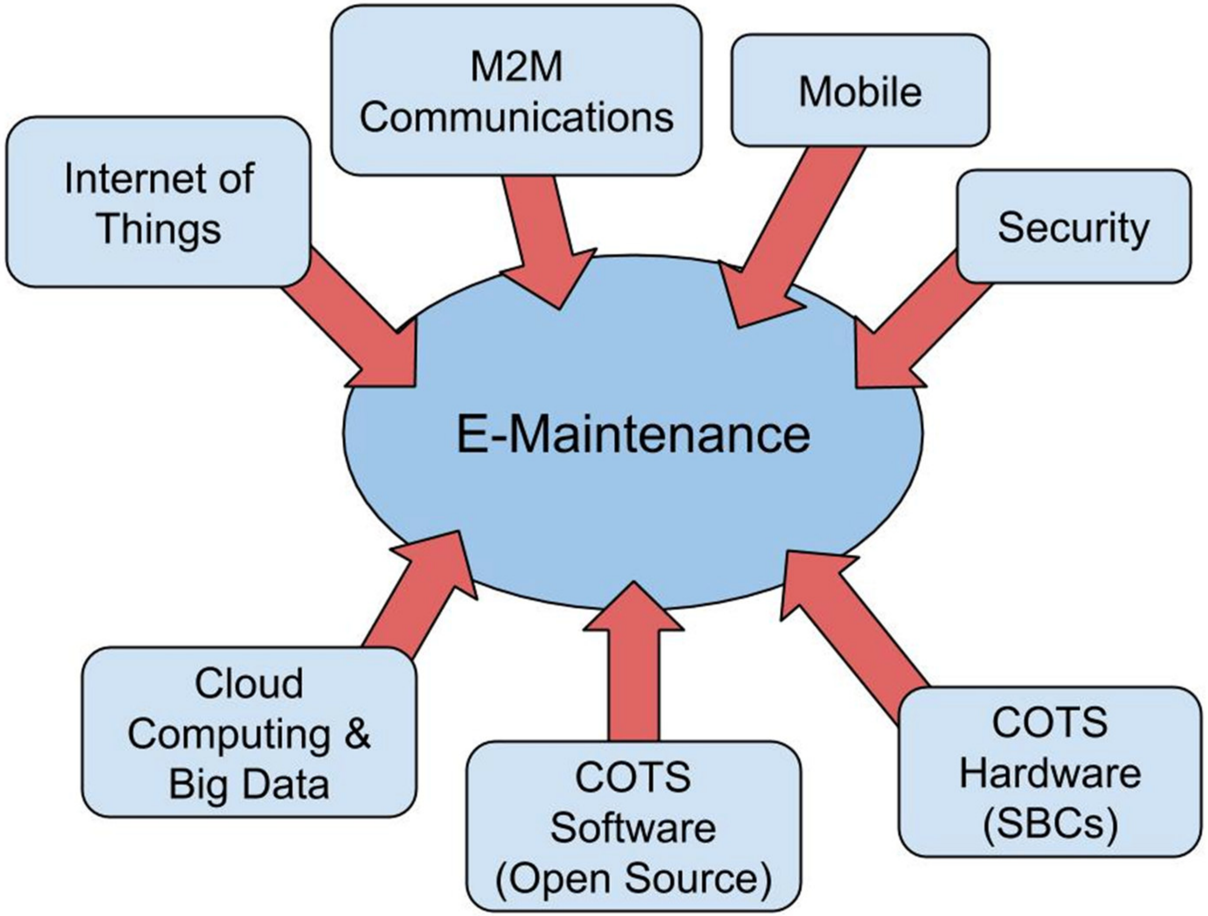
e-Maintenance Motivation: from Products to Services (Servitization)

- E-Maintenance platforms that permit **real-time remote monitoring and control** of machine operational state are paving the way to **servitization**
- **Servitization**: a new business model
  - instead of selling machines, sell a service.

- Examples:
  - **Rolls Royce** used to sell engines and replacement parts. Now it provides **long-term repair and maintenance** - or "power by the hour" (the "total care" service). It is Rolls's job to make sure the engine keeps running. Margins are typically higher than on hardware
  - several **printer producers** (Canon, HP, Xerox, etc.) are selling **number of printed pages**, no longer the printer machine itself

e-Maintenance and the Internet Ecosystem

- The digital revolution, in particular the Internet ecosystem, is forcing a rapid (disruptive) evolution of e-Maintenance platform

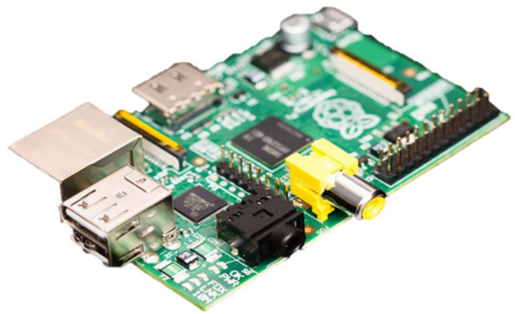


Commercial Off-The-Shelf (COTS) Hardware

- Modern Single-Board Computers (SBC) are very cheap (<100€) and permit a large scale adoption.
- SBC hosts complete operating systems, i.e. Linux, thus offering:
  - high level programming models and tools
  - easy integration and interoperability
  - re-use of existing software components

- SBC examples: BeagleBone, Arduino, Raspberry Pi
- Linux SBC as fundamental basic block for modern e-Maintenance platforms

- Cultural problem: SBC can be the future machine control platform? Often there is a strong “PLC-based” culture opposing the change

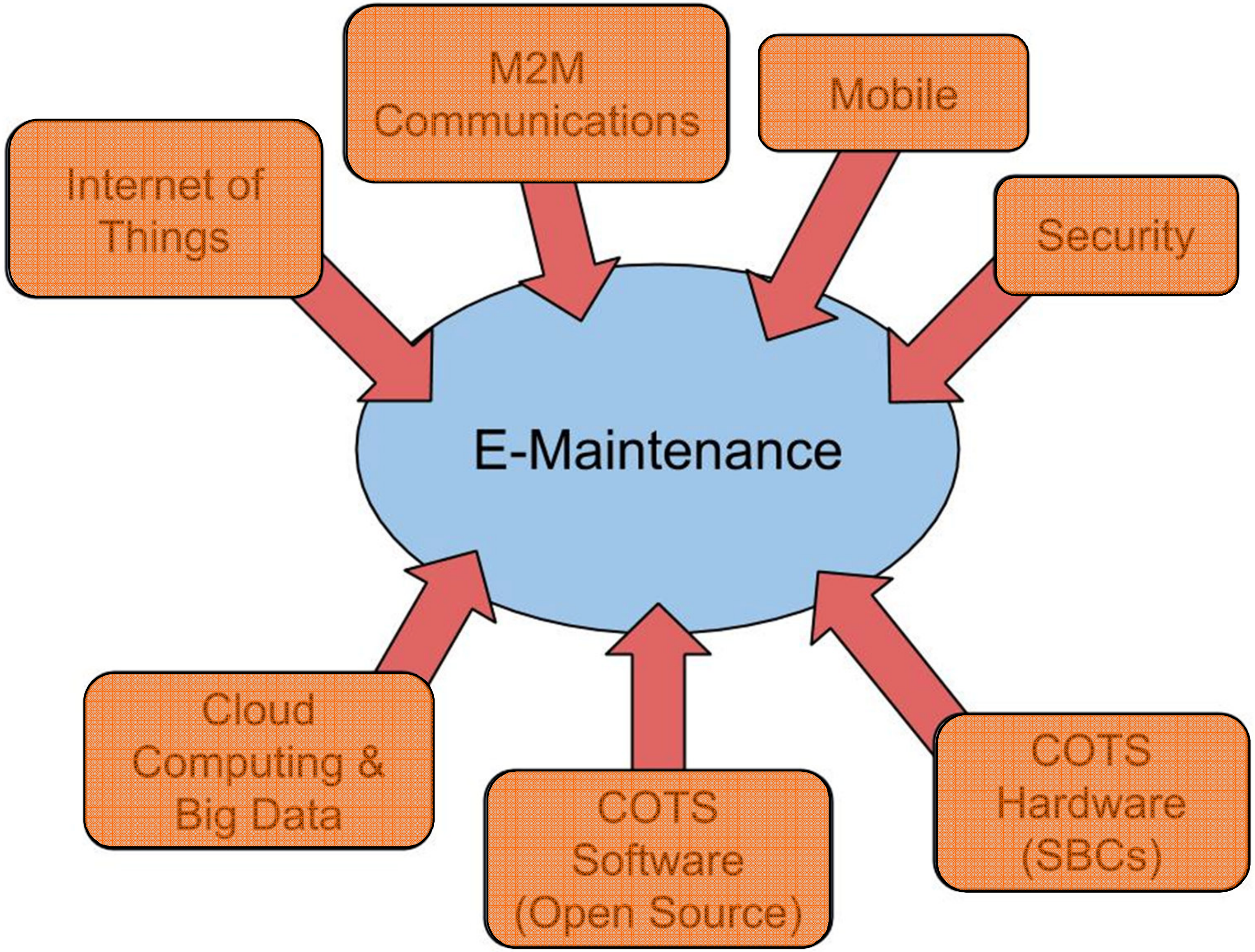


COTS Software

- Novel e-Maintenance platform (low cost, high efficiency) leverage on the recent evolution in the Internet ecosystem, mostly Open Source software components:
  - development and deployment environments
  - management tools
  - communication bus management library
  - Web-based communication architecture
  - security solutions
- In addition, this permits to escape the typical lock-in of large electronic component producers (Siemens, Rockwell,…)

- Cultural problem: the “proprietary” culture is usually very entrenched, as if “proprietary” could significate (not correctly) “stable” and “reliable”
  - is there anything more reliable of the Internet global infrastructure, almost completely dominated by Open Source software components

e-Maintenance and the Internet Ecosystem

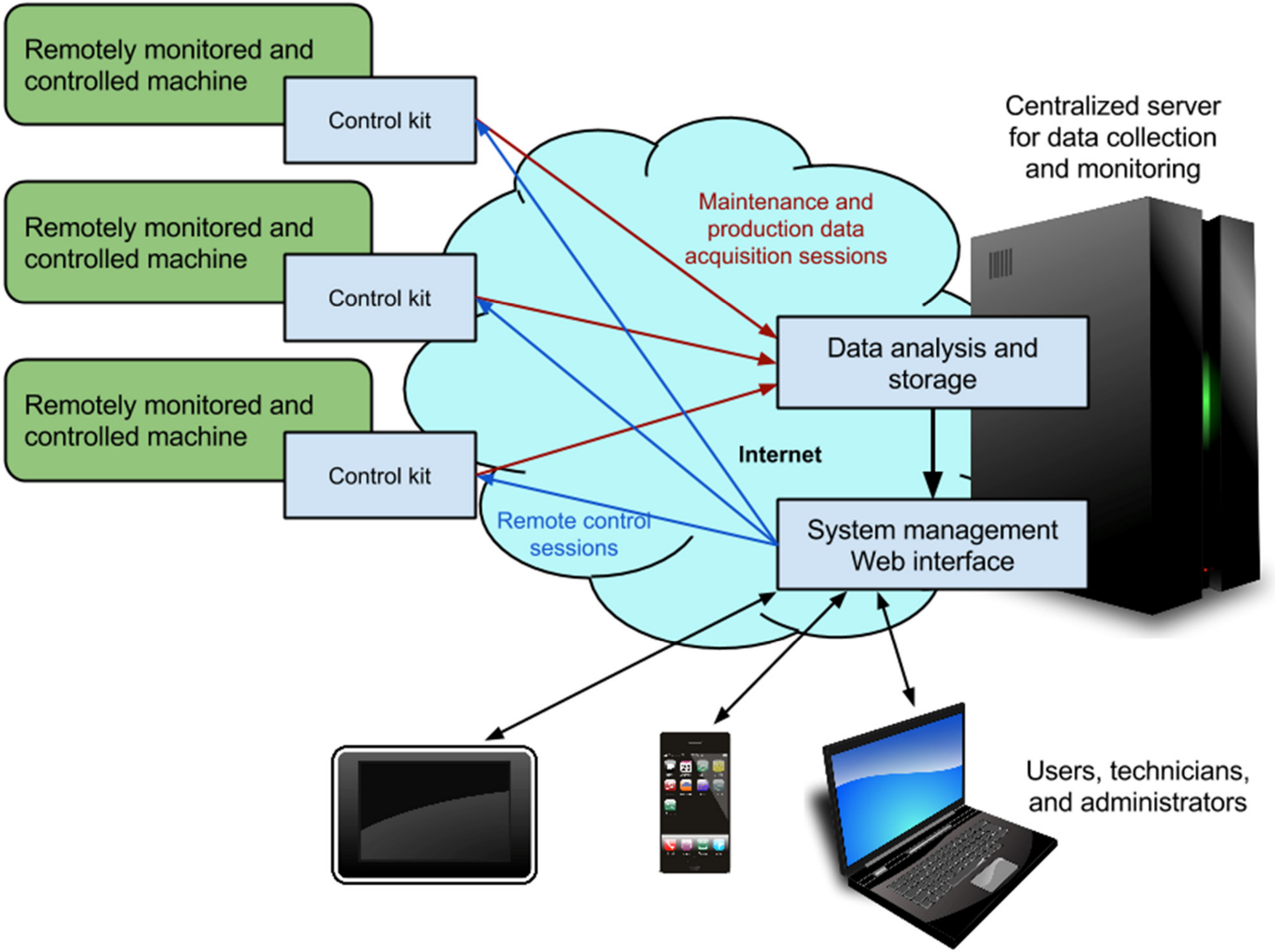


“Large-scale” e-Maintenance (1)

- COTS, Cloud, Big Data, M2M, IoT, Mobile… are all technologies that can contribute to a widespread adoption of e-Maintenance platform, more and more cost effective, efficient, etc.

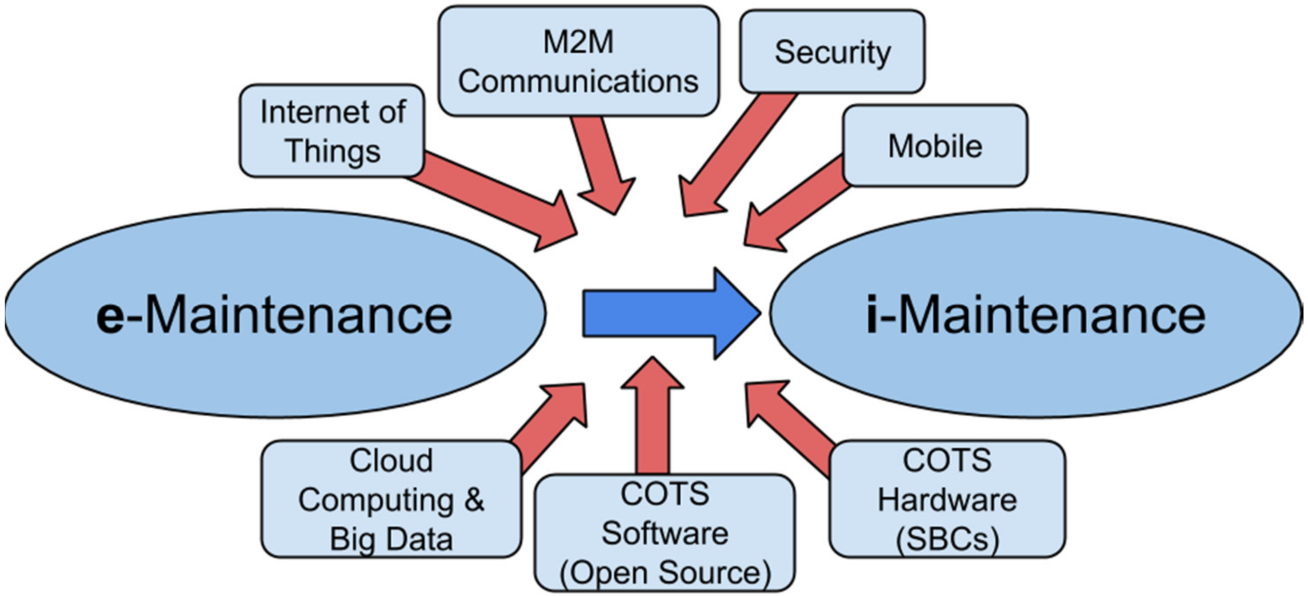
- Towards large scale e-Maintenance platforms, that could be adopted in several areas, such as:
  - machine automation
  - industrial plant
  - household & similar appliances
  - environmental monitoring
  - agricultural soil condition monitoring
  - etc.

“Large-scale” e-Maintenance (2)



From e-Maintenance to i-Maintenance (1)

- The term e-Maintenance was born with “traditional” ICT technologies in mind: custom electronic, proprietary software, etc.
- New ICT technologies (IoT, Cloud, Big Data, COTS, …) should probably suggest us to move from e-Maintenance to i-Maintenance, thus taking into account the strong contribution of the Internet world



From e-Maintenance to i-Maintenance (2)

- Warning: for an i-Maintenance project to be successful in the manufacturing arena, it is important to take into proper account **the whole Internet ecosystem**

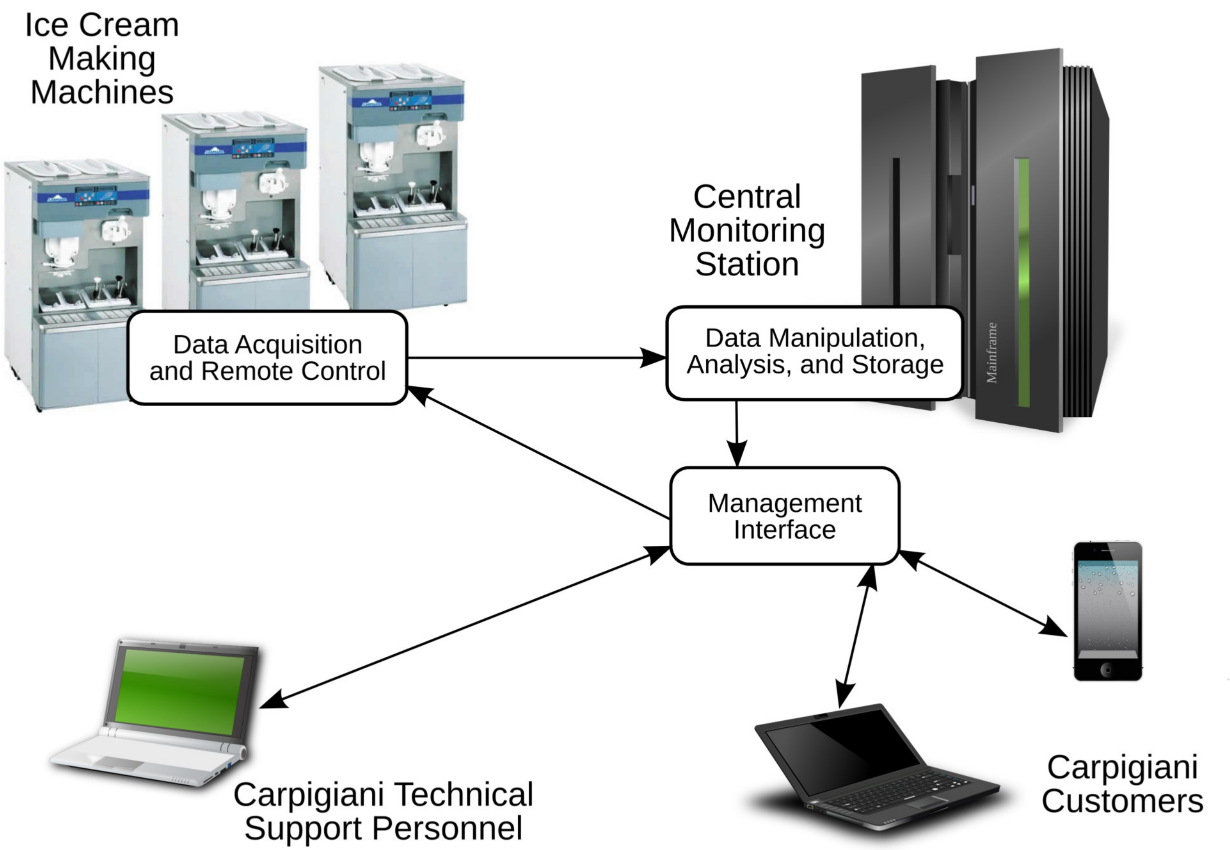
- Internet ecosystem driving forces:
  - **standard, standard, standard**, for heterogeneous systems interoperability
  - “**consumerization**” push
  - languages, environment, more and more powerful **middleware** solutions (in rapid evolution)
  - **open source**, not only intended as software, but also as a collaborative software development process
  - there is the need of a **tight integration of IT and OT research** in the manufacturing arena

From e-Maintenance to i-Maintenance (3)

- i-Maintenance opens up to new interesting evolutions, goals:
  - (self) diagnosis and (self) prognosis, self-healing systems, self-maintenance
  - adaptive design, i.e., designing systems that can evolve and adapt to the environment
  - better management of the expert knowledge base
  - client in-the-loop: CRM systems tightly integrated in the production process
  - Enterprise Information System (EIS) more and more integrated (even SAP is moving toward light and open…)

- New production models: from mass production to mass customization or individualized production
- New business models: performance-based contracting, power by the hour, servitization

The Carpigiani e-Maintenance Solution

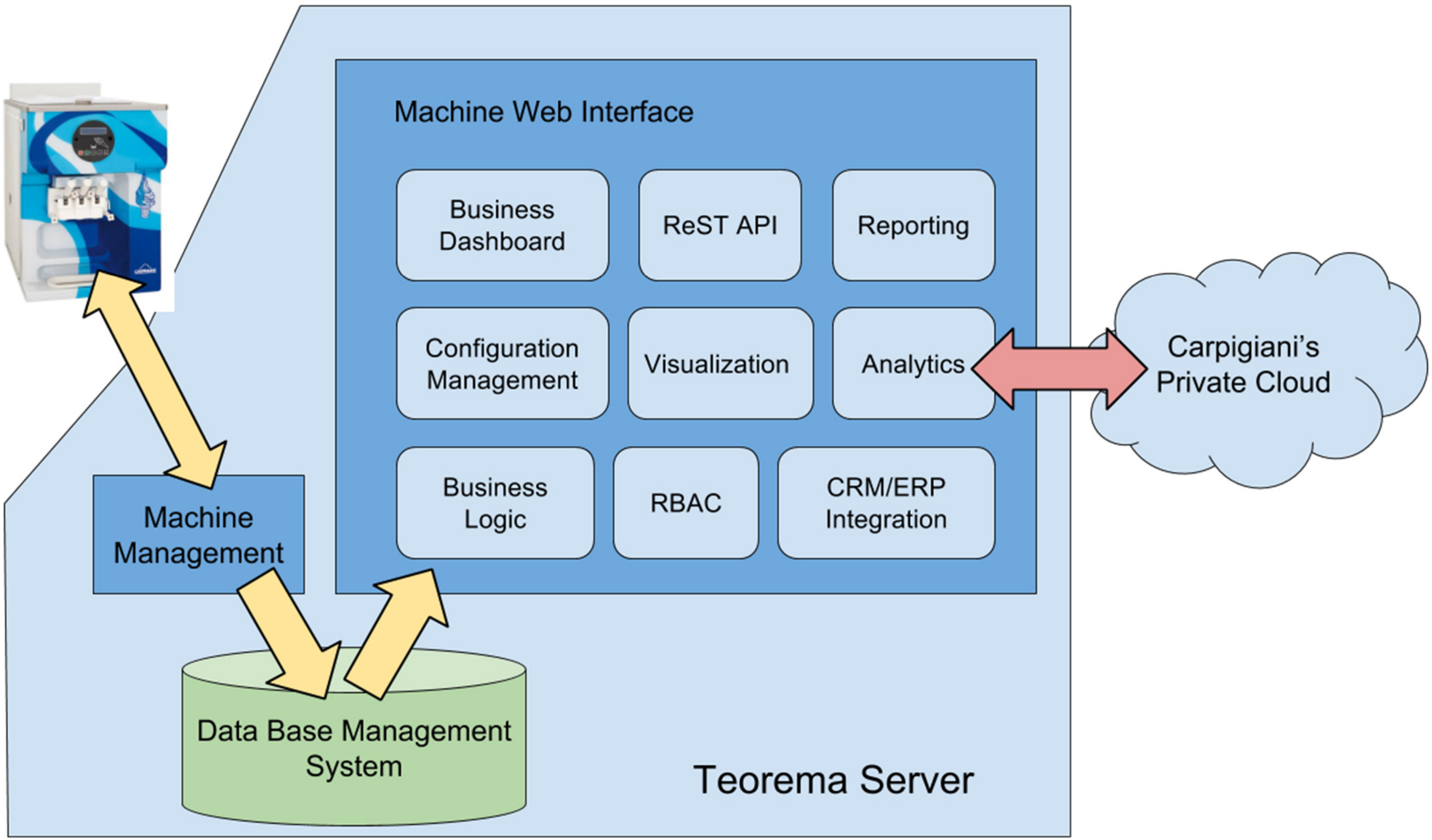


Teorema: the Carpigiani e-Maintenance Solution

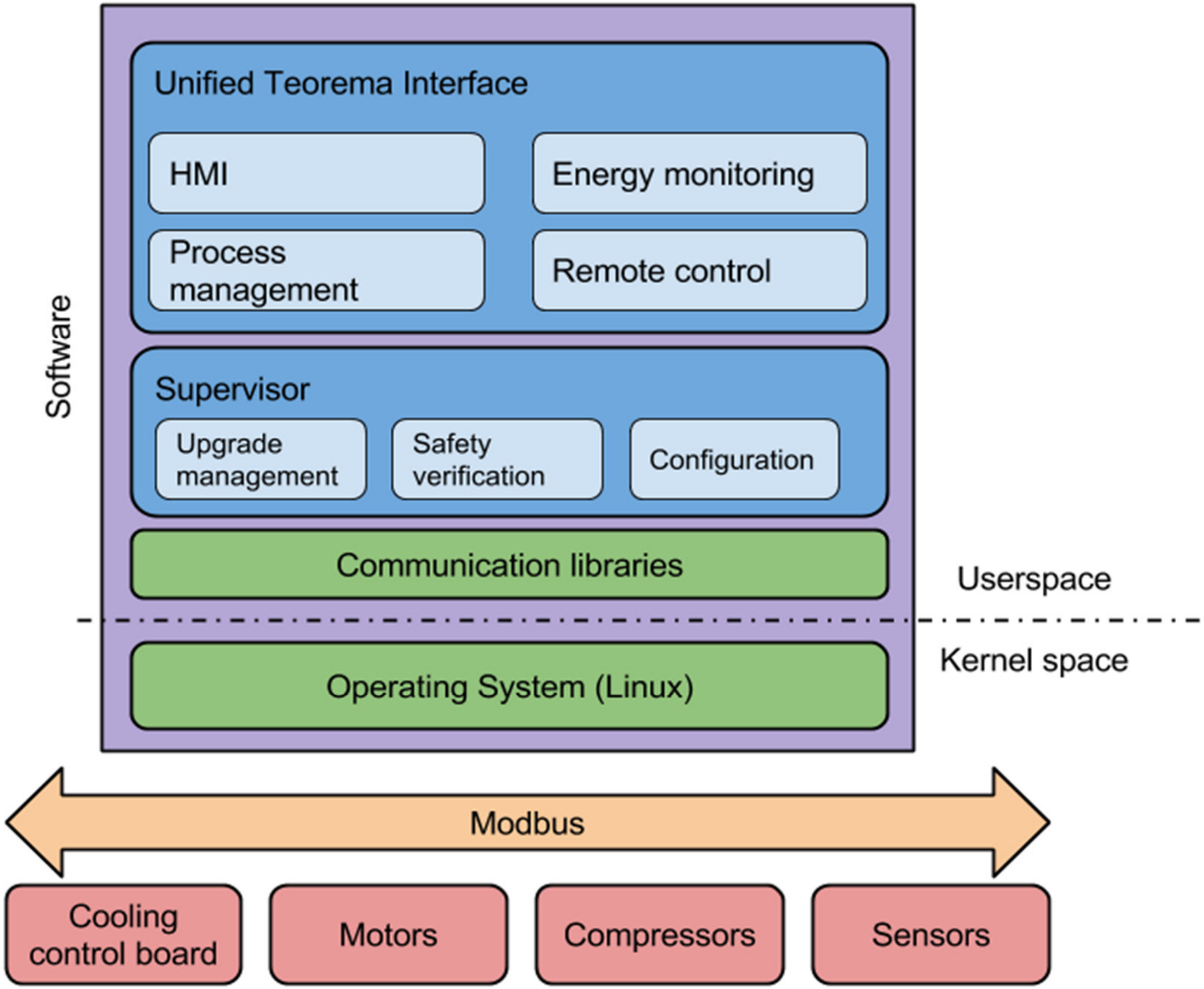
- Teorema goals:
  - reduce maintenance costs
  - innovative after-sales service to customers

- Teorema functions:
  - remote monitoring of each machine operational state: number of ice creams served, engine hours of work, temperatures, ...
  - automatic notification of malfunctions
  - interactive diagnostics analysis, some prognostics functions
  - reporting service for technicians, for the real time access to the machine operational state: temperature, pressures, engine state, alarms, ...
  - reporting service for customers, for the real time access to the production data of their machines: number of served ice creams, ingredient exhaustion, ...

Carpigiani Overall Architecture



Carpigiani Machine Software Architecture (project Idea)



Carpigiani e-Maintenance Platform (project Teorema)

- Teorema is the distributed application that connects Carpigiani machines and represents the core of the e-Maintenance platform.
- It is a client/server Web-based application composed of a machine-installed remote control hardware and a centralized monitoring station.
- The remote control hardware is packaged in a kit that comes pre-installed in new ice cream machine models and can be installed in older models

- Customers can access Teorema information and services anywhere, anytime, and from a wide range of devices with different screen sizes, CPU capabilities, and input interfaces, e.g., laptops and smartphones

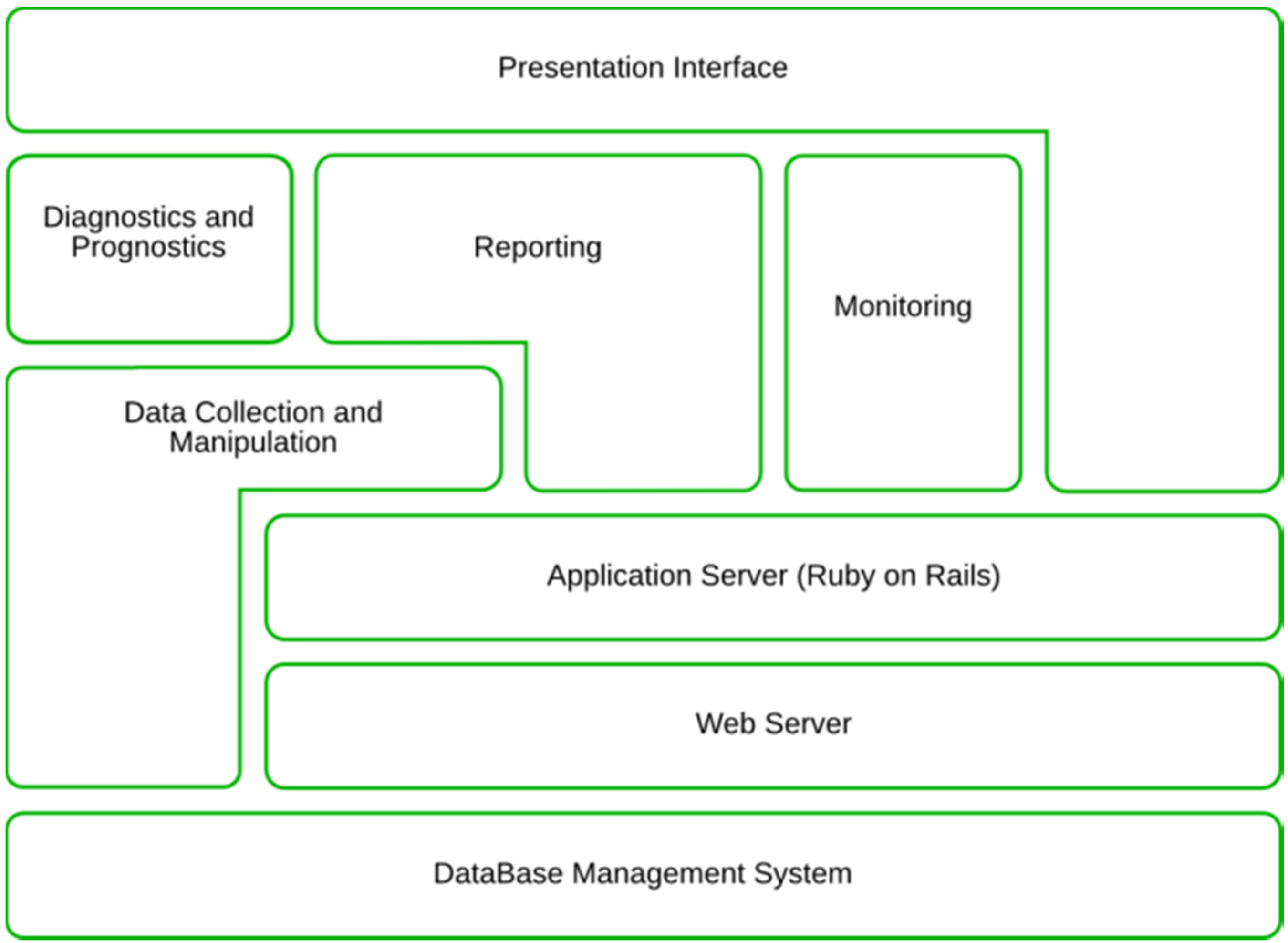
Carpigiani e-Maintenance Communication Protocol

- Remote control kits and the Central Monitoring Station communicate with a simple and flexible native protocol
- The **native protocol** choice was made because:
  - **no mature protocols were available** when Teorema was designed (2007)
  - protocols are complex: the adoption of a standard protocol would have required an upfront (very careful) study and design of message formats, that would have slowed down (at the time) the initial adoption of Teorema

- Carpigiani ice cream making machines will increasingly operate in a **smart and interconnected world**, e.g., smart kitchens
- It is possible that future versions of Teorema will **switch to standard protocols**

Carpigiani e-Maintenance Platform (project Teorema)

- Teorema Central Monitoring Station (Server) architecture



Teorema Mobile Interface

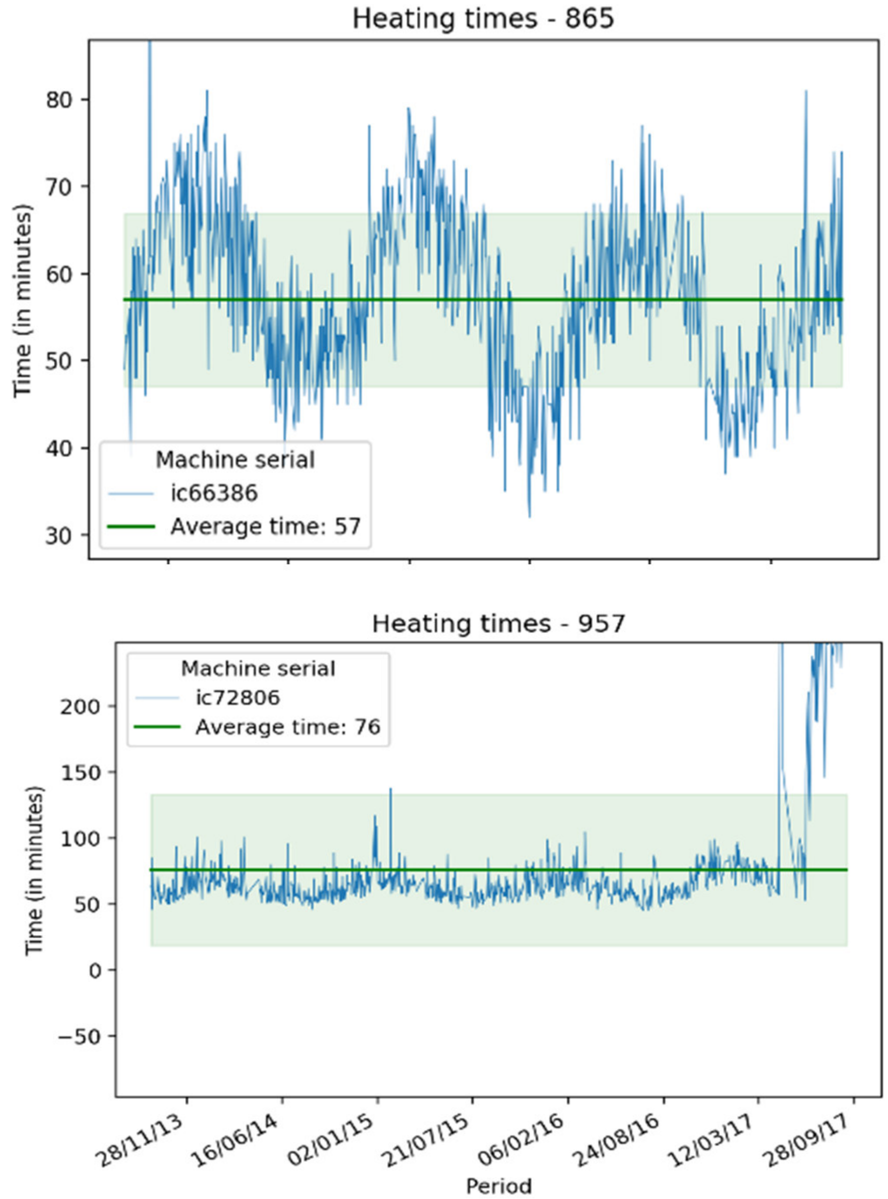
Teorema and Open Source

- The development of Teorema is based on Open Source technologies, that present several advantages:
  - rapid application development, thanks to many components already available (and to a collaborative Internet community)
  - highly innovative, albeit already mature for the industrial use and very well supported
  - reduce development costs and protect the investments by not restricting the software platform to a single proprietary vendor.
- Example: Web-based interface, realized as a Ruby on Rails application, a framework for the development of Web 2.0 applications based on the Ruby programming language and designed to enable agile programming practices, thus allowing rapid development and minimizing the time-to-deployment

Teorema-enabled Analytics (1)

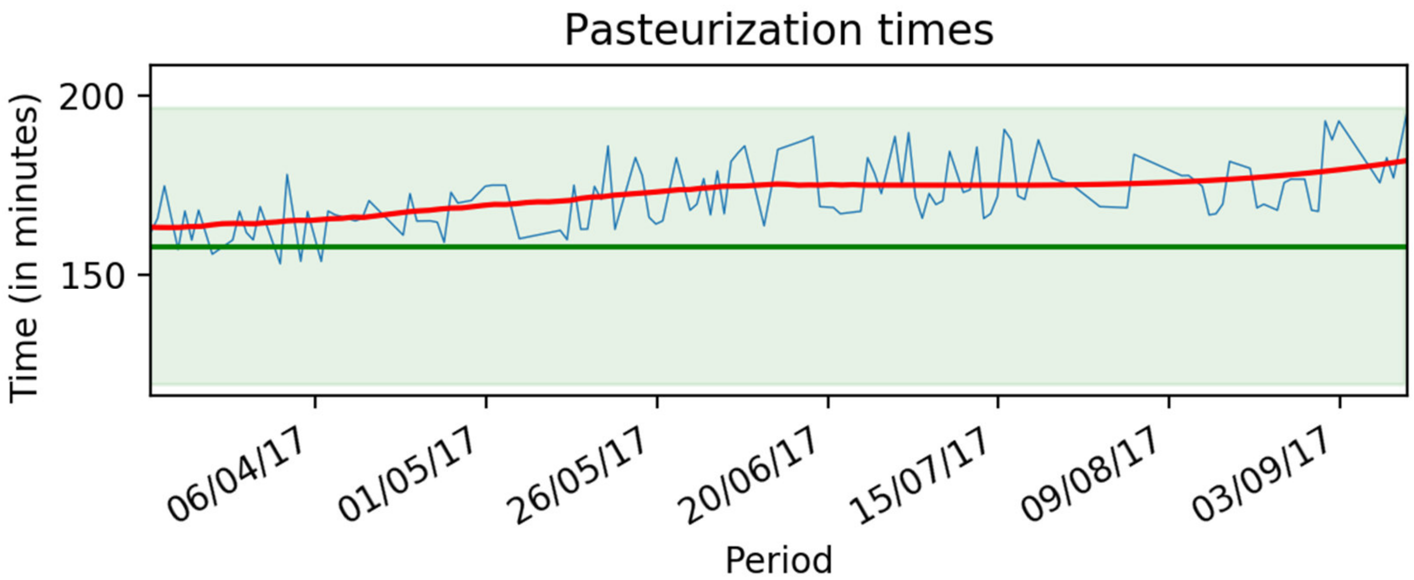
- Up: heating times with (regular) seasonal shifts
- Down: heating times showing an abrupt and unpredictable breakdown

- Useful for **prompt** intervention after machine failure



Teorema-enabled Analytics (2)

- Pasteurization time slowly increasing, despite the season
- Performance drift of machines along their lifecycle allows to forecast the likelihood of breakdowns and proactively plan technician operations



Teorema and RAMI 4.0

- Design and development of Teorema started in 2007, well before the RAMI 4.0 reference architecture and also before IoT and Industry 4.0
- By anticipating the industrial state-of-the-art, Teorema presents many interesting similarities with the RAMI 4.0 of almost one decade afterward

Teorema and RAMI 4.0: Layers

- Teorema nicely fits the RAMI 4.0 Layers vertical dimension by adopting an architecture that neatly separates the different layers

- Assets, ice cream machine themselves
- Integration and Communication, based on cellular enabled off-the-shelf boards
- Information and Functional, storing gathered data in relational databases and allowing to access them for advanced data processing
- Business, providing differentiated high-level information to several stakeholders
  - technical details of running machines to R&D engineers
  - smart maintenance services to service providers
  - business support services to ice cream shop owners to investigate their usage

Teorema and RAMI 4.0: Hierarchy Levels

- Teorema follows only partially the RAMI 4.0 Hierarchy Levels direction in relation to communication.

- Ice cream machines outside the factory environment (actually enlarging factory virtual borders) connected to the Carpigiani factory, as suggested by RAMI 4.0
- However, RAMI 4.0 adopts a cross-layer approach allowing (smart) appliances to interact with any smart factory component
- Instead, Teorema server-side components act as intermediary between ice cream machines and other Carpigiani enterprise-wide tools (such as CRM and ERP)

- A cross-layer approach increases the flexibility of the adopted solution, but its costs in terms of development and management complexity in the smart appliance scenario may overcome its benefits

i-Maintenance, conclusive remarks

- i-Maintenance is definitely a disruptive innovation that can open up new Production, Maintenance, and Business models.
- i-Maintenance is breeding a new generation of solutions that will be user-centered, intelligent, and tightly integrated in the IT corporate system
- Warning: when talking about "Manufacturing 2.0", beware that the "2.0" term is much more than a set of ICT technologies. It refers to methodologies, work organization, technology culture, change management, eco-systems, …

Outline

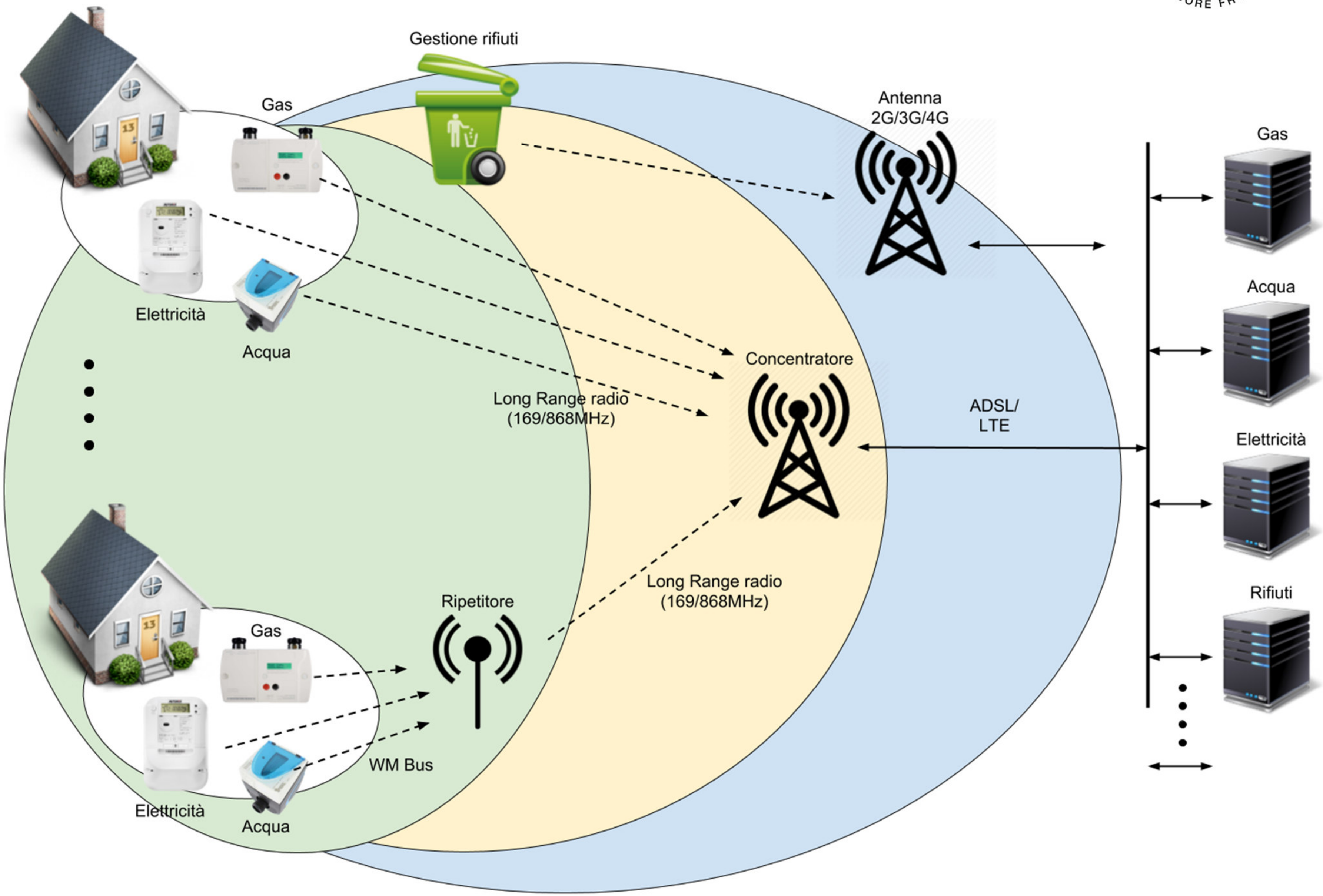
3. Industry 4.0
   - introduction and definition
   - IIRA and RAMI 4.0 reference architectures
   - Industry 4.0 and the digital twin

4. Examples of IoT and Industry 4.0 solutions
   - from e-Maintenance to i-Maintenance
   - IoT-enabled use cases
   - experience with a business case

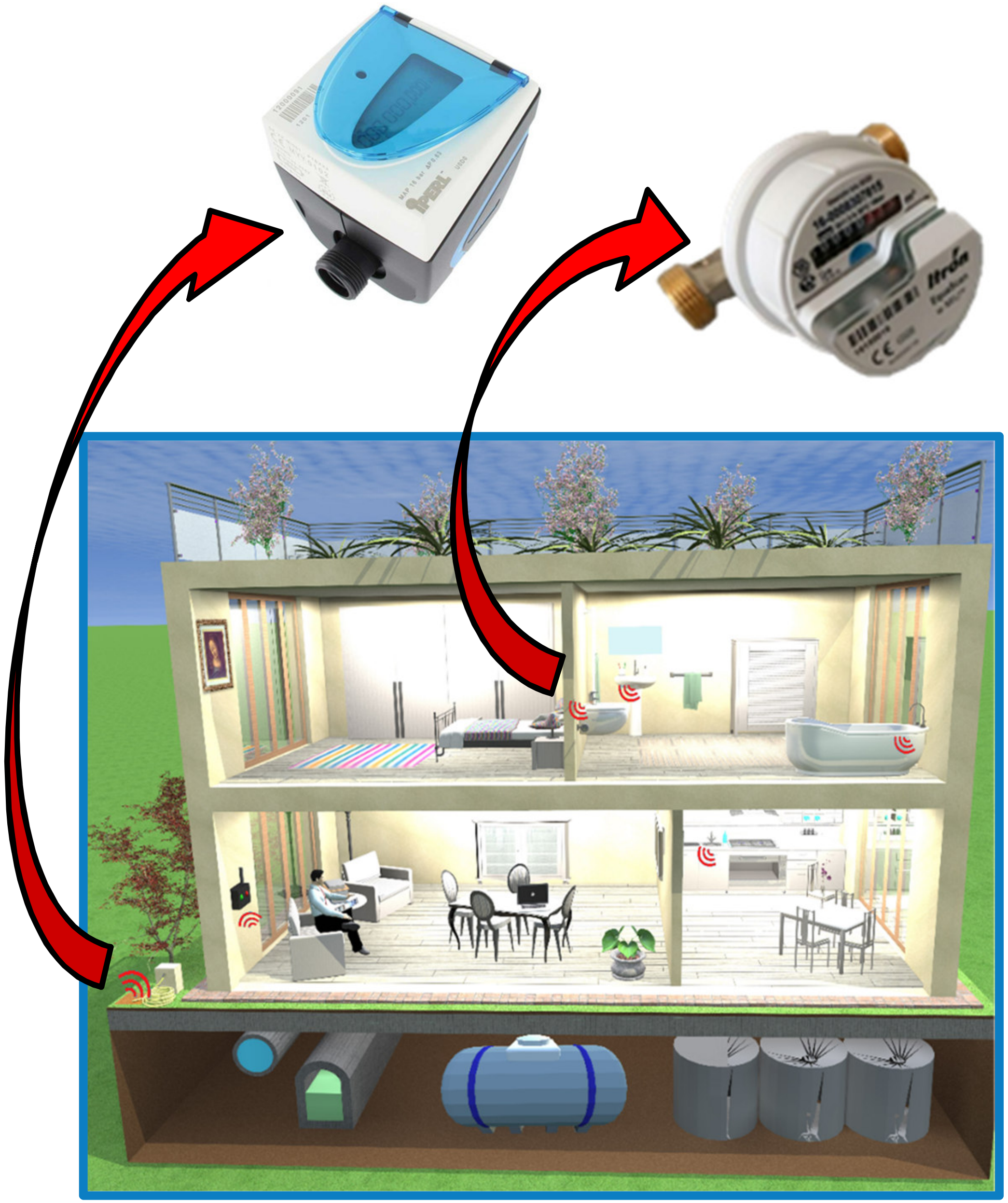
Smart Metering

- Centralized and autonomous metering system to
  - maximize measurement performance, e.g., with frequent value reads
  - support energy efficiency, e.g., promptly identifying water leaks
  - provide benefits for end-users, e.g., with real-time energy consumption
- Based on
  - smart meters, digitalized and connected
  - smart (wireless) communication network
  - smart centralized control system

Smart Metering: Typical Architecture



Smart Meters

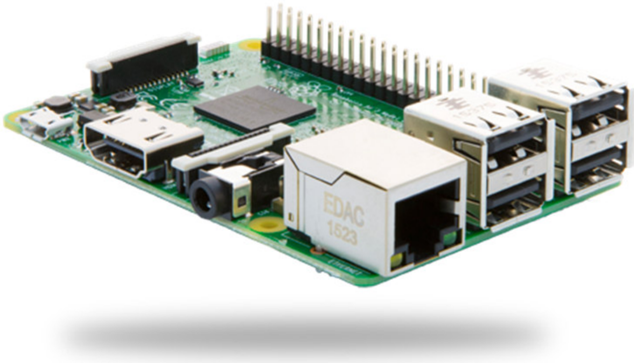


- Sensus iPERL meter
  - open protocol (OMS) based on Wireless M-Bus
  - periodic reading ≥ 15min
  - https://sensus.com/products/iperl-international/
- iTron meter
  - open protocol (OMS) based on Wireless M-Bus
  - periodic reading ≥ 5min
  - https://www.mwatechnology.com/brands/itron/

Data Gateway: Hardware

Università degli Studi di Ferrara

- Based on Rasperyy Pi 3 Single Board Computer (SBC)
- Linux operating system
- Low-cost: < 30€
- Low-power: < 4 Wh



Raspberry Pi 3
Single Board Computer

Data Gateway: Wireless Communication

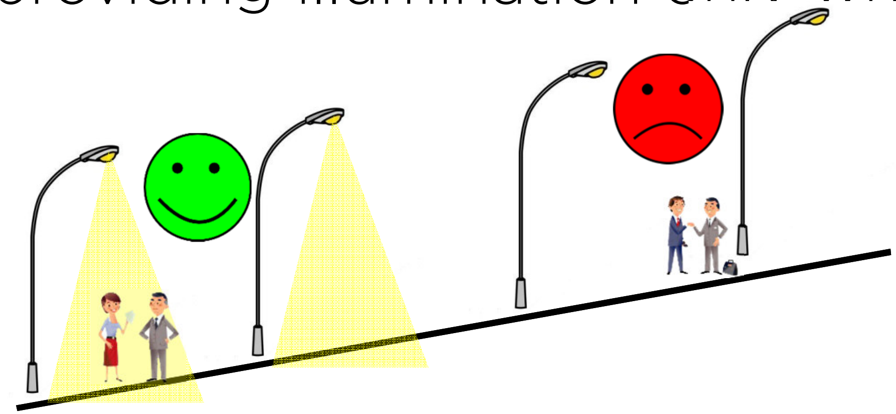
Università degli Studi di Ferrara

- Towards the cloud:
  - embedded WiFi & Bluetooth to send data to nearby access points
  - 2G/3G/4G modules to extend the connectivity

- Towards smart meters: wireless modules working on ISM 868MHz band
  - Wireless Meter Bus
  - LoRa - Long range radio

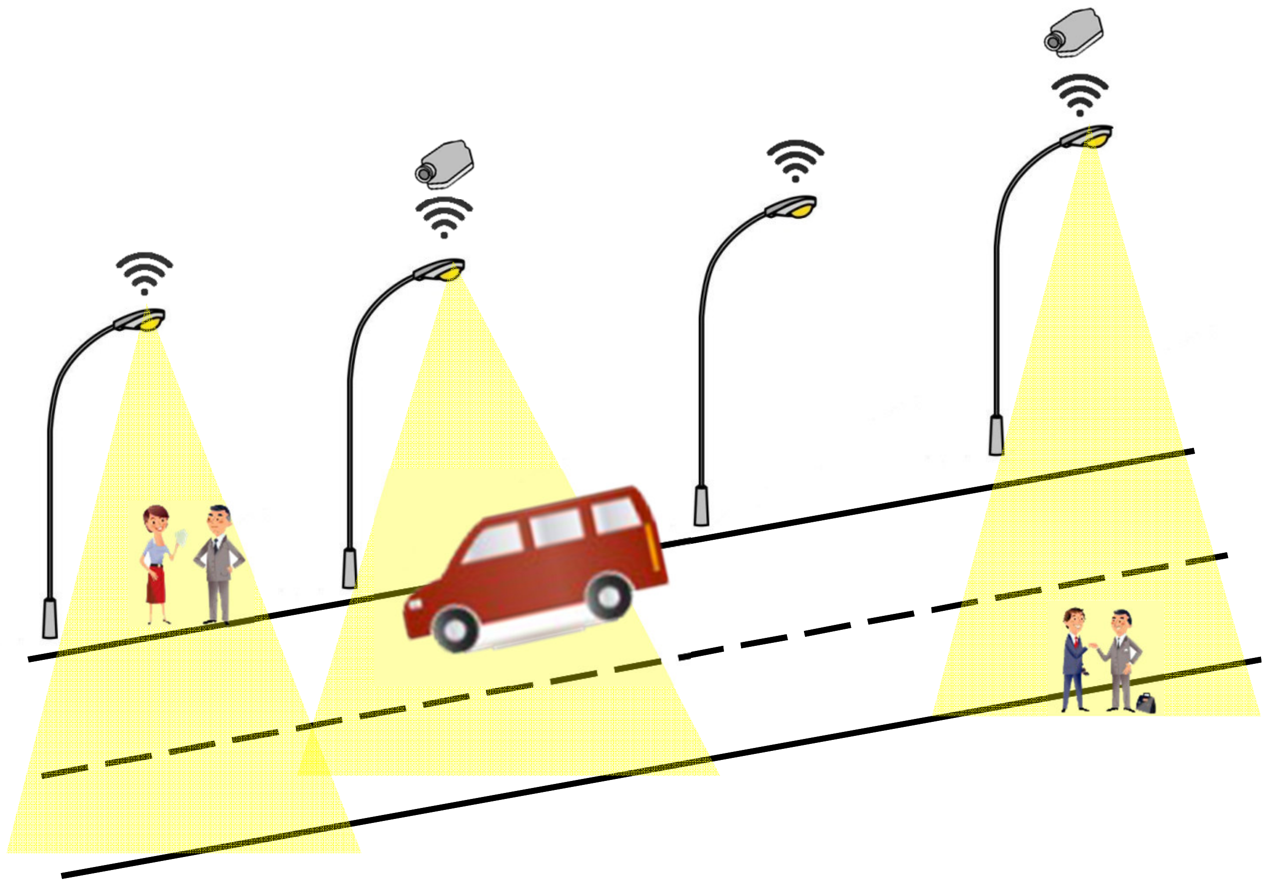
IoT-enabled Smart Spotlights

- Outdoor spotlights always on during nighttime
  - impose huge energy wasting
  - but there is no need of illumination without human activities
- Evolve controllable spotlights towards smart lighting systems, providing illumination only when and where required



Dynamic detection of activities

- Evolving controllable spotlights towards smart **real-time lighting** products…
- …by detecting **human activities**, e.g., vehicles, bicycles, pedestrians…
- …based on **cameras** deployed on roads



- Human activity → more illumination
  – save energy and thus money

Main components

AXIS M1113-E

AXIS M1114-E

- Cameras with trap algorithms
  - embedded programmable modules
  - otherwise, additional Single Board Computer
- Communication module
  - multi-hop short-range wireless protocol + gateway
  - otherwise, cellular communication
- Server-side application

PRESS Project –
Pervasive Environment Sensing and Sharing

Real-time information about air pollution by PRESS box

Aggregated view of sensed data

Boost awareness about the quality of air

Citizens and SMEs more aware of the natural environment they live in

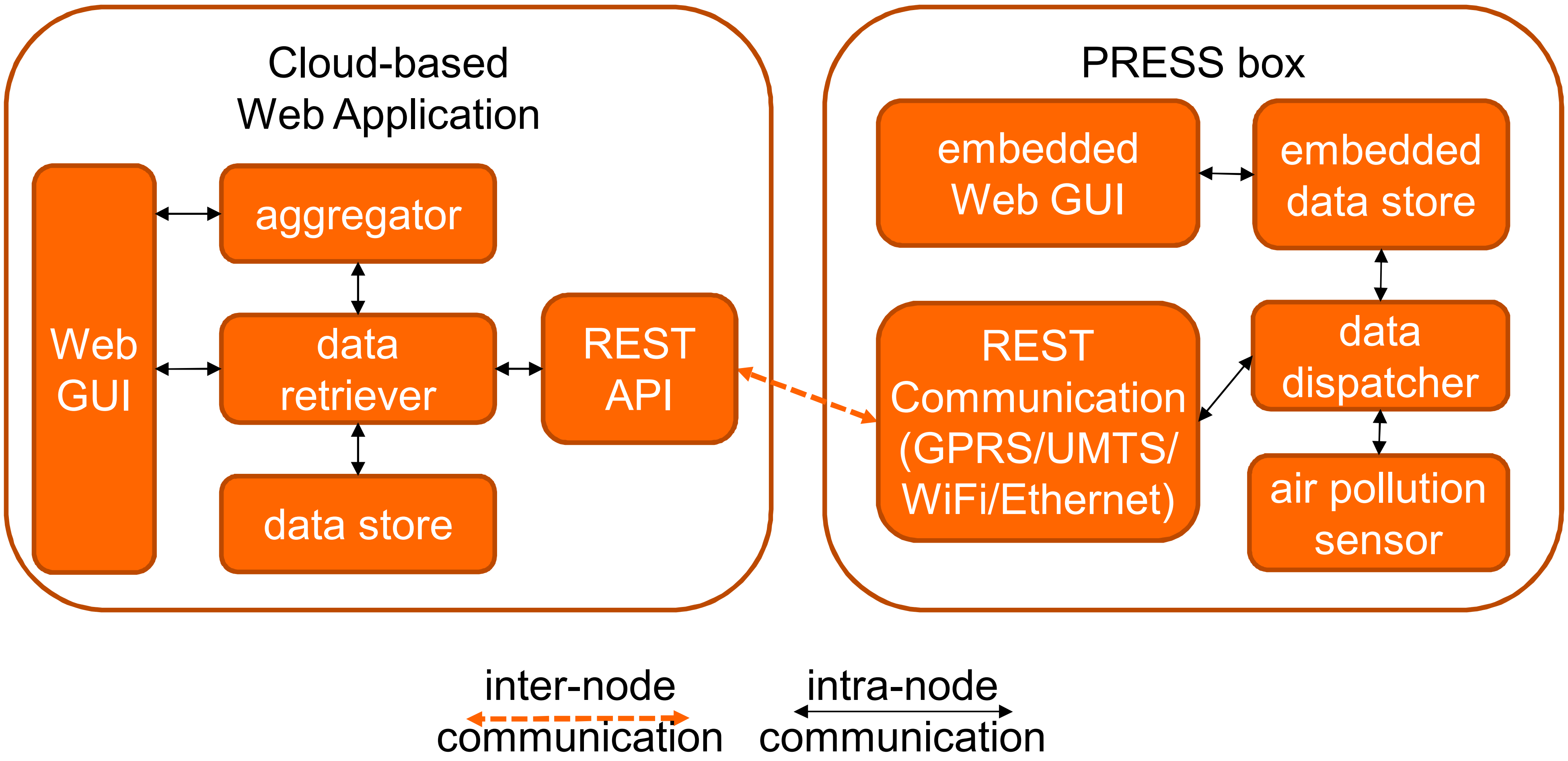
Libelium Air Quality Sensors

Challenges and Solution

- Challenges
  - information about the quality of air is typically very coarse-grained
  - usually only few sensing devices deployed across wide geographical areas
  - pervasive air quality information is still missing!

- Solution:
  - provide citizens and SMEs with pervasive and personalized information about the quality of air
  - Information directly gathered by citizens
  - assessing the quality of air related to and locations citizens spend most of their time and point of interests for SME

Overall Architecture



- PRESS box gathers pervasive air pollution information and send data to the cloud
- Cloud-based application store and share the status of the quality of air
- Similar solution: EveryAware www.everyaware.eu/activities/case-studies/air-quality

PRESS Potential Customers

- Citizens interested in measuring the quality of the air they breathe
- Runners and bikers to select healthy paths during their sport activities
- SMEs in the agriculture and cattle fields to monitor and certify the quality of the environment cultivation/cattle has grown
- Public bodies to exploit a cheaper way to detect air pollution measured on wider areas and not only in selected spots

PRESS Business Model

- Sale of PRESS boxes, each box will include
  - perpetual access to data directly interacting with an embedded Web server
  - access to the cloud Web app to visualize personalized data
  - access to aggregated data related to a specific neighborhood
- Recurring revenues for premium services based on a monthly subscription to
  - access personalized data via the cloud-based Web application
  - access aggregated data via the cloud-based Web application
  - access aggregated data for wider geographical locations
- On-demand further data elaboration
  - based on customers needs

IoT-enhanced Farming

- Toward the ICT-enabled farm of the future

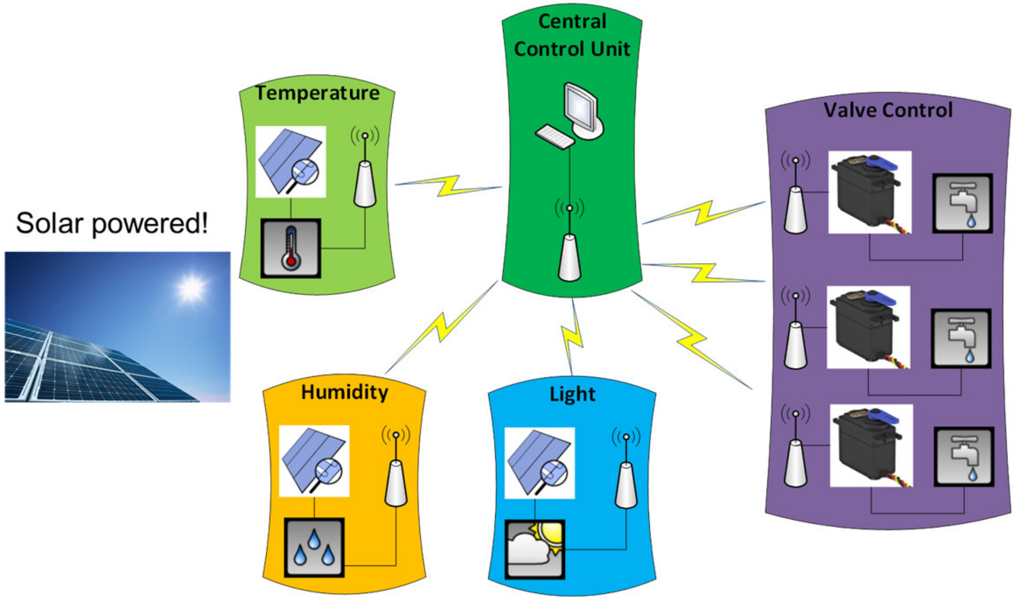
- Irrigation is mainly performed either in a manual manner or in very coarse-grained programmed manner
- Current market landscape has a variety of expensive solutions tailored on big farms
- Small farms are waiting for accessing current state-of-the-art ICT technology in a cost-effective manner

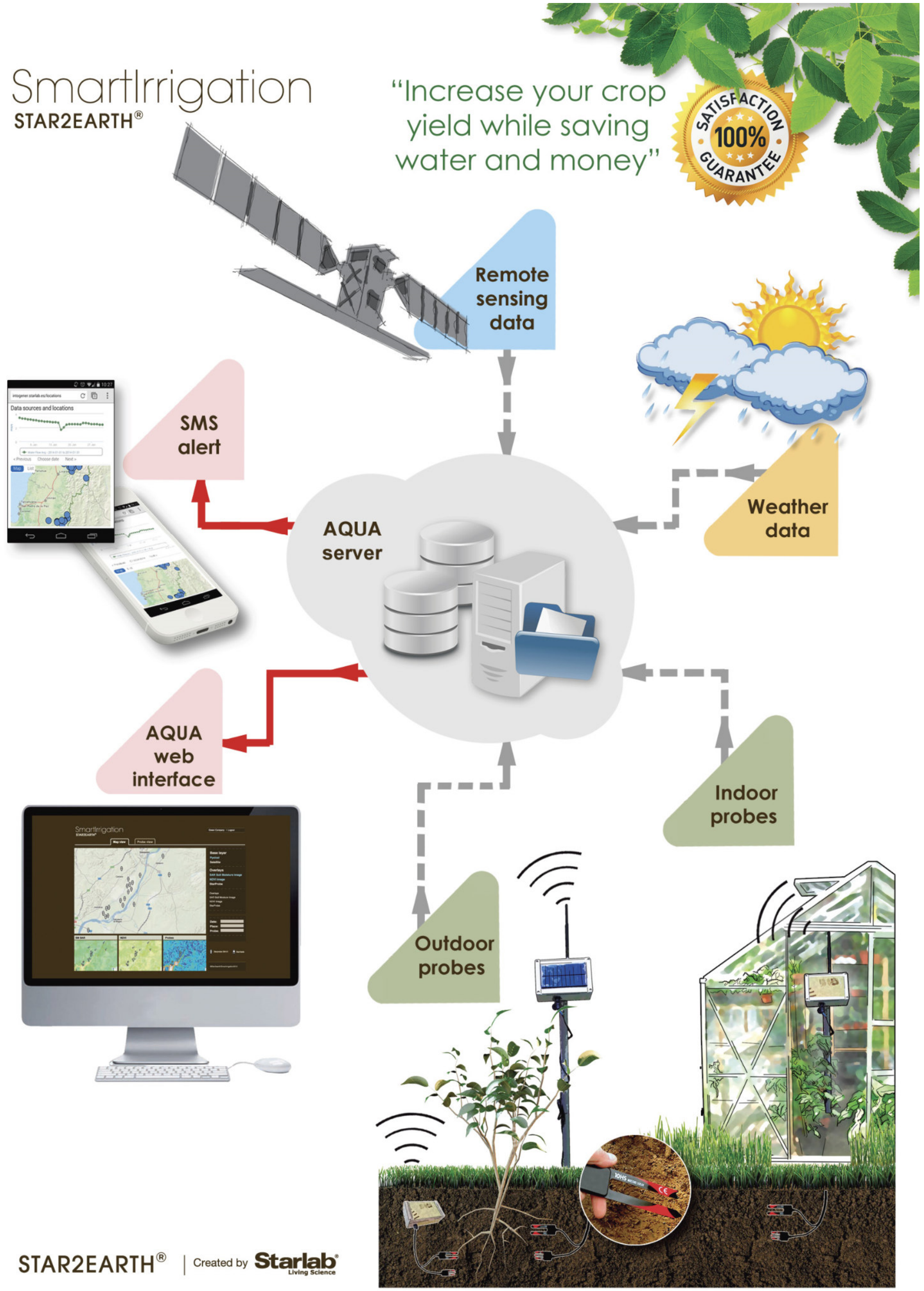
Smart Water Management

- Current solution
  - back and forth to the fields to manage irrigator systems
  - little or no information about water/energy consumption
  - delayed identification of valves/pumps inefficiencies and pipes leakages

- Smart irrigation system
  - remote management, no need to periodically move farmers/equipment
  - Many useful weather information: real-time, historical, forecast
  - easy-to-use Web app to monitor water/energy consumption in real-time

Examples


Solar powered!
Temperature
Central Control Unit
Valve Control
Humidity
Light


SmartIrrigation
STAR2EARTH®
"Increase your crop yield while saving water and money"
SATISFACTION 100% GUARANTEE
Remote sensing data
SMS alert
AQUA server
Weather data
AQUA web interface
Indoor probes
Outdoor probes
STAR2EARTH® | Created by Starlab

Main steps and components

- Connect sensors and actuators via Web based on standard API
- Control via Web valves/pumps in real-time as if you were in the field
- Manage periodic events and unusual behaviors remotely
- Analyze in the cloud data gathered from the field to generate new knowledge

- Based on
  - terrain humidity/temperature sensors, controllable pumps/valves, local weather station, weather forecast Internet service, information from domain expert (eventually, AI-based)
  - short/long-range wireless communication, Single Board Computers, Web/mobile app...

More than a product: a change of perspective

	Value	Traditional	IoT-enabled
Creation	Farmers' needs	Specific solution tailored to irrigation	Cross-domain solution**, also considering future requirements**
	Offering	Standalone products, likely to quickly become obsolete	**Continuous improvements** of products and software
	Role of data	Data gather by equipment for the only purpose of its management	Heterogeneous data correlated to generate **new knowledge**
Remuneration	Profits	Selling of products	**Recurring revenue** based on services
	Control	Traditional after-market maintenance	Remote h24 maintenance
	Value chain	Vertical expertise on specific products and processes	Identification of **novel remunerable services and processes**

Monitoring of Fire Extinguishers

- Information from physical sensors
  - pressure
  - weight
  - accelerometer
- Alarm in case of use and/or movement

- Information from digital information sources
  - ERP integration to gather the date of the last maintenance
- Alarm in case of expiration

A Wireless Sensor Network of Fire Extinguishers

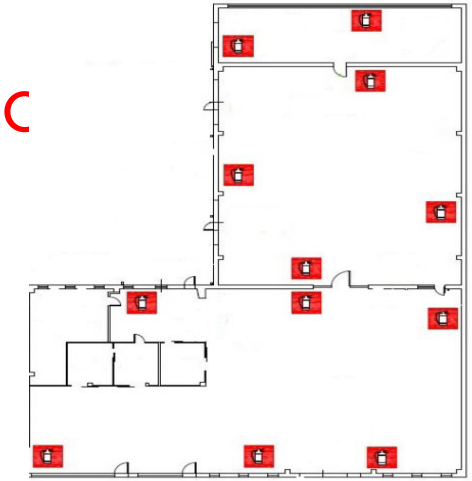
- Fire extinguisher equipped with short-range wireless technology
  - ZigBee, based on IEEE 802.15.4
  - limited power consumption
  - multi-hop paths to dispatch information to a gateway

- Gateway equipped with long-range wireless technology
  - UMTS (or even Ethernet, if available)
  - send data/alarms to the central server
  - verify the availability of fire extinguishers

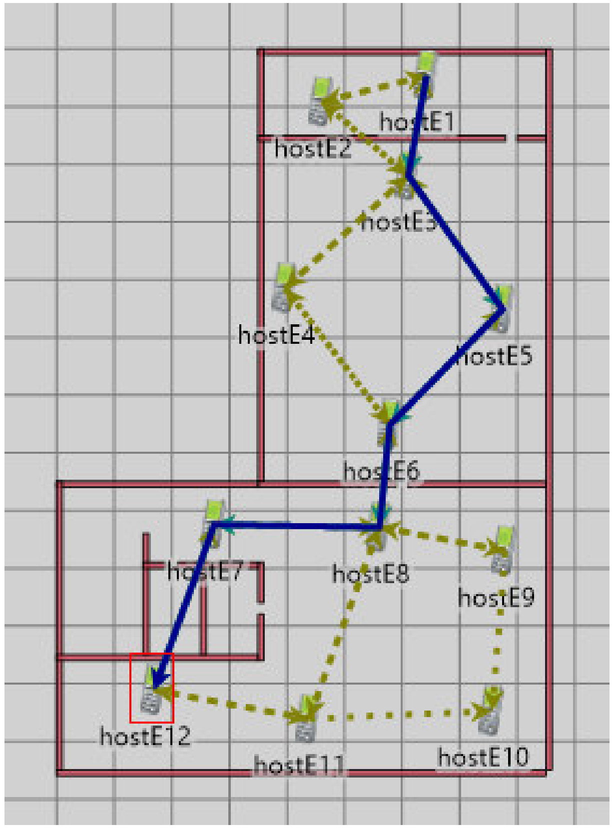
Coverage of large areas

- Goal: optimal deployment of fire extinguisher to ensure connectivity
  - how many fire extinguisher?
  - best locations?
  - how many missing fire extinguisher before splitting the network?
- Traditional solutions:
  - deploy and verify (complexity)
  - additional devices (cost)

Simulation of environments with obstac



- OMNeT++ simulator: Objective Modular Network Testbed in C++
- INET module for OMNeT++
  - Internet protocols: TCP, UDP, IPv4, IPv6
  - Data Link protocols: Ethernet, IEEE 802.11
  - Obstacle Loss Model to simulate signal attenuation due to walls: brick, wood, glass



- Identification of single point of failures
  - e.g., hostE3 and hostE6

Zume Pizza

- Founded by former Zynga Studio head Alex Garden in Mountain View, California
- From the production line assembly of the pizza to the eventual delivery, robots are the primary labour ingredient

- Backing on the way!
  - delivery trucks, each with 56 remotely controlled ovens pre-loaded with pizzas
  - drive around like an ice cream truck and cook the pizza moments before it is delivered

- https://www.youtube.com/watch?v=TkhWonFm-Lw

Zume Pizza and Industry 4.0

- A small plant with robotic and visual recognition, with connected and mobile production units

- Most of Industry 4.0 digital industrial technologies
  - advanced robotics
  - horizontal/vertical integration
  - industrial Internet
  - cloud
  - cybersecurity
  - big data and analytics

Outline

3. Industry 4.0
   - introduction and definition
   - IIRA and RAMI 4.0 reference architectures
   - Industry 4.0 and the digital twin

4. Examples of IoT and Industry 4.0 solutions
   - from e-Maintenance to i-Maintenance
   - IoT-enabled use cases
   - experience with a business case

Apply IoT and Industry 4.0 to a Proof of Value (1)

- How to use IoT and Industry 4.0 to generate value
- What
  - identify a real-world scenario: clients, potential market, strength and weak points, …
  - eventually, exploit one of the previous use cases
- Why
  - identify the "added value"
  - exploiting IoT and Industry 4.0 it is possible to enable novel products and services…

Apply IoT and Industry 4.0 to a Proof of Value (2)

- How
  - describe how to show the "added value" to a customer/investor/decision maker
  - describe how IoT and Industry 4.0 generate new value: novel products, efficiency, servitization, other business innovations, …
  - identify IoT components: sensors/actuators, wireless requirements, suitable data exchange protocols, IoT platforms, AI
  - identify costs/complexity (in a very coarse manner)

- Final questions:
  - which Industry 4.0 digital industrial technologies?
  - relationships with IIRA and RAMI 4.0?

References (1)

- J. Campos, “Development in the application of ICT in condition monitoring and maintenance”, Computers in Industry, Vol. 60, No. 1, pp. 1-20, 2009.
- M. Carrero, “Innovation for the Web 2.0 Era”, IEEE Computer, Vol. 42, No 11, pp. 96-98, 2009.
- CrespoMar̀quez, A. (2007), "The Maintenance Management Framework", SpringerVerlag, London, UK.
- Chryssolouris, G., Mavrikios, D., Papakostas, N., Mourtzis, D., Michalos, G., and Georgoulias, K. (2009), “Digital manufacturing: history, perspectives, and outlook”, *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, Vol. 223, No. 5, pp. 451462.
- Holmberg, K., Adgar, A., Arnaiz, A., Jantunen, E., Mascolo, J., Mekid, S. (Eds.), "E-maintenance", Springer-Verlag, London, United Kingdom, 2010.
- R. Lazzarini, C. Stefanelli, M. Tortonesi, G. Virgilli, "E-Maintenance for Household and Similar Appliances", International Journal of Productivity and Quality Management, Inderscience, ISSN 1746-6474, Vol. 12, No. 2, pp. 141-160, 2013.
- R. Lazzarini, C. Stefanelli, M. Tortonesi, "Large-Scale E-maintenance: A New Frontier of Management?", Proceedings of the 13th IFIP/IEEE Integrated Network Management Symposium (IM 2013). 2013, Ghent, Belgium.

References (2)

- Lee J., Ni J., Djurdjanovic D., Qiu H. and Liao H, (2006), Intelligent prognostics tools and e-maintenance, Computers in Industry, 57 pp. 476–489
- Levrat, E., Iung, B., CrespoMar̀quez, A. (2008), "Emaintenance: review and conceptual framework", *Production Planning & Control*, Vol. 9, No. 4, pp. 408–429.
- Moubray J. (1997), "ReliabilityCentered Maintenance", 2nd Edition, Industrial Press Inc., New York, NY, USA.
- Muller, A., CrespoMar̀quez, A. and Iung, B. (2008), "On the concept of e–maintenance: Review and current research", *Reliability Engineering and System Safety*, Vol. 93, No. 8, pp. 11651187.
- Nakajima, S. (1988), "Introduction to TPM: Total Productive Maintenance", Productivity Press, Cambridge, MA, USA.
- A. Sehgal, V. Perelman, S. Kuryla, J. Schönwälder, "Management of Resource Constrained Devices in the Internet of Things", IEEE Communications Magazine Vol. 50, No. 12, pp. 144-149, December 2012.
- Z. Shelby, "Embedded Web Services", IEEE Wireless Communications, Vol. 17, No. 6, pp. 52-57, 2010.
- S. Vaughan-Nichols, "The Mobile Web Comes of Age", IEEE Computer, Vol. 41, No. 11, pp. 15-17, 2008.
- Galloway, B.; Hancke, G.P., "Introduction to Industrial Control Networks," *Communications Surveys & Tutorials, IEEE* , vol.15, no.2, pp.860,880, Second Quarter 2013.
- Forbes: Zume's Robot-Made Pizza Asks: You Want A Piece Of This?

References (3)

- http://www.rfidjournal.com/articles/view?4986
- http://www.stoorm5.com/
- https://iot.ieee.org/definition.html
- Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications
- IEEE Internet of Things: Towards a definition of the Internet of Things (IoT)
- Eclipse IoT White Paper – The Three Software Stacks Required for IoT Architectures
- Cisco Fog Computing Solutions: Unleash the Power of the Internet of Things
- Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are
- RFC 7228: Terminology for Constrained-Node Networks
- https://www.polymorph.co.za/iot-connectivity-comparison-gsm-vs-lora-vs-sigfox-vs-nb-iot/
- https://www.hivemq.com/blog/mqtt-essentials-part2-publish-subscribe
- Messaging Technologies for the Industrial Internet and the Internet of Things
- https://www.hivemq.com
- https://www.smart-industry.net/iot-platforms-foundation-success/

Backup introductory material on AI and ML

In the following slides, without any pretense of being exhaustive, you can find some introductory material on AI and ML

Artificial Intelligence

- Long-term goal of developing intelligent entities
- Started in 1956, more than 60 years ago!
  - Minsky, McCarthy, Shannon, Newell, Simon

- How to define/verify if an entity is actually "intelligent"?
  - transitory definition: Artificial Intelligence (AI) goal is to allow computers to do things that humans do better, so far (chess, Go)

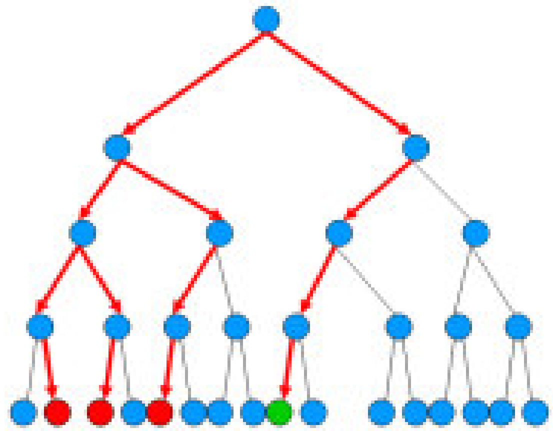
AI categories

- Regular activities
  - natural language
    - comprehension
    - production
    - translate
  - perception
    - vision
    - spoken language
    - common sense reasoning
    - robot control
- Formal activities
  - games: chess, go
  - mathematatic and logic
    - automatic theorem validation
    - geometry
    - differential calculus
- Specialized activities
  - engineering
  - financial
  - programming
  - planning

AI Scope

- Problems without a fixed resolution procedure

- Branching: at each step there are many possible choices, each producing several and different possible outcomes



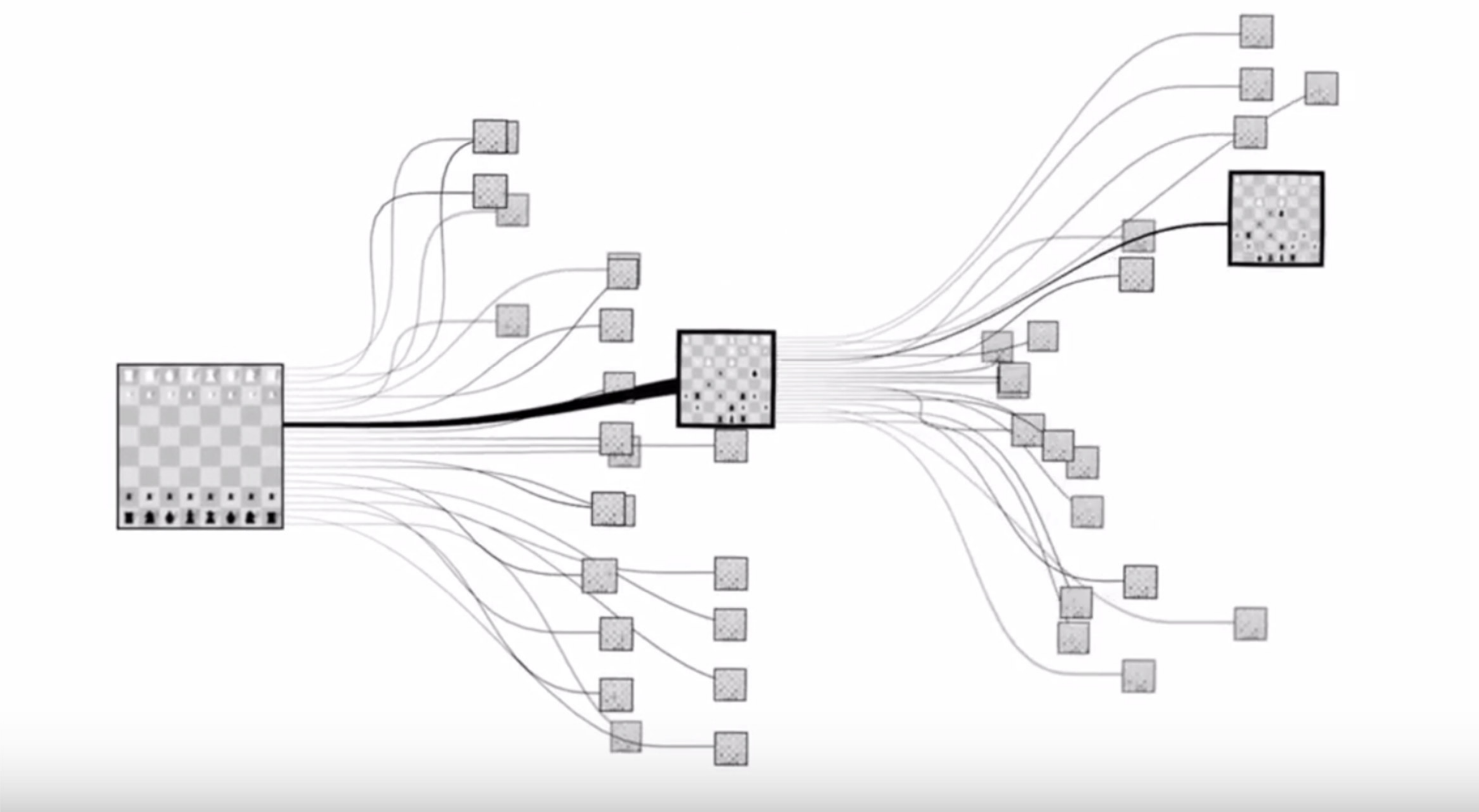
Chess

- In 1997 DeepBlue beats Kasparov

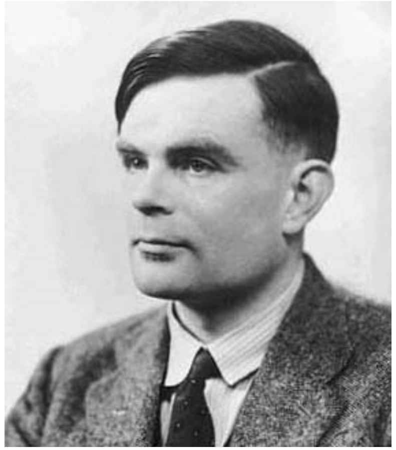
Chess problem size

- Possible legal combination of 32 chess pieces on 64 squares is beteen 10^{43} and 10^{50}
- Unfeasable to compute every possible combination

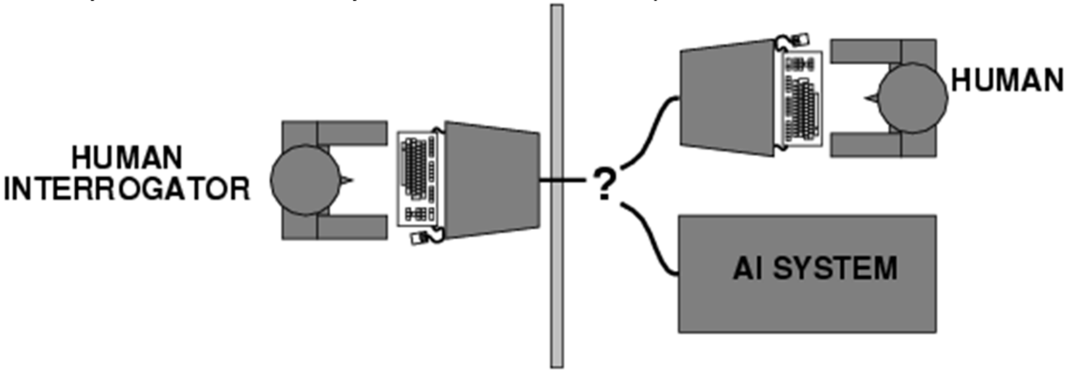
Chess Branching



AI Definition: Turing Test



- Alan Turing (1950): “computing machinery and intelligence”
- Imitation Game
  - a human operator interacts with a person and an AI system (written questions and answers)
  - after a while, the human interrogator should be able to distinguish among the human and the AI system
  - otherwise, the AI system has passed the Turing Test



Turing Test

- AI system skills to pass the Turing Test:
  - natural language
  - knowledge representation
  - automatic reasoning
  - automatic learning

- Global Turing Test:
  - robotic
  - artificial vision

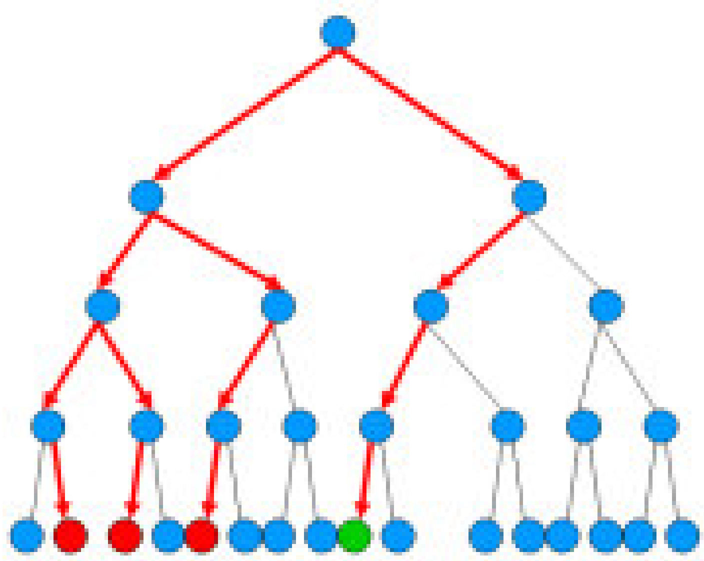
AI Techniques

- Knowledge Based (KB) system
  - rules provided by domain experts

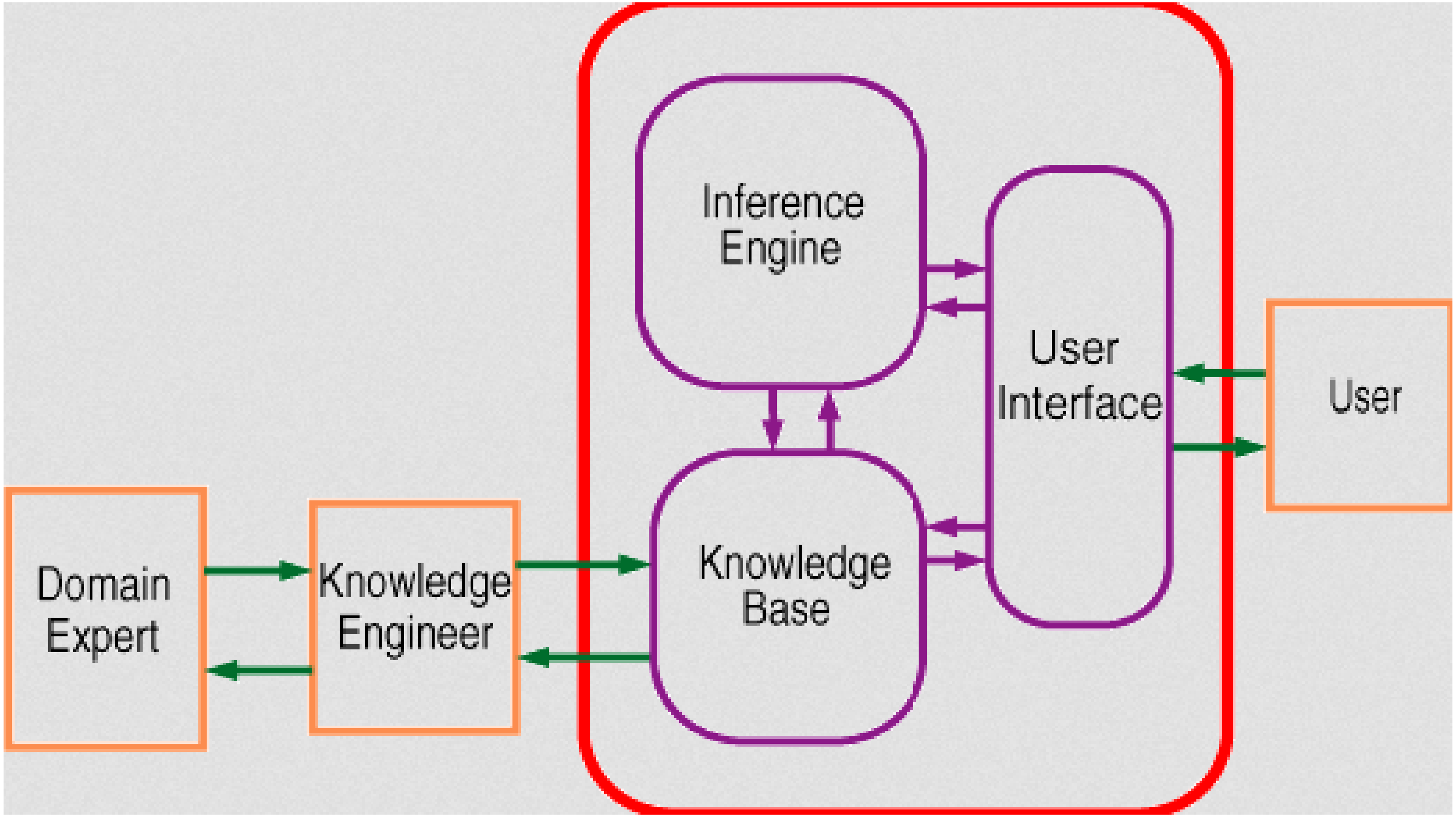
- Machine Learning
  - supervised: based on a training set
  - unsupervised: not based on a training set
  - Artificial Neural Networks based on models of actual neurons

Knowledge Based (KB) System

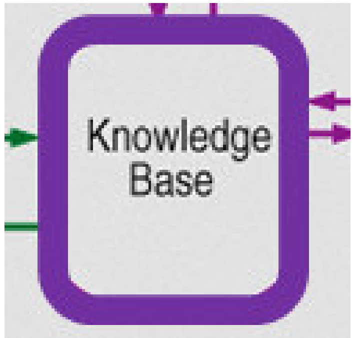
- Rules: knowledge representation of a domain provided by a human expert
- Control module to consider possibilities and select the best solution
- Capabilities of solving problems with performance close to a human expert in that specific domain



Architecture of a KB system



KB Example



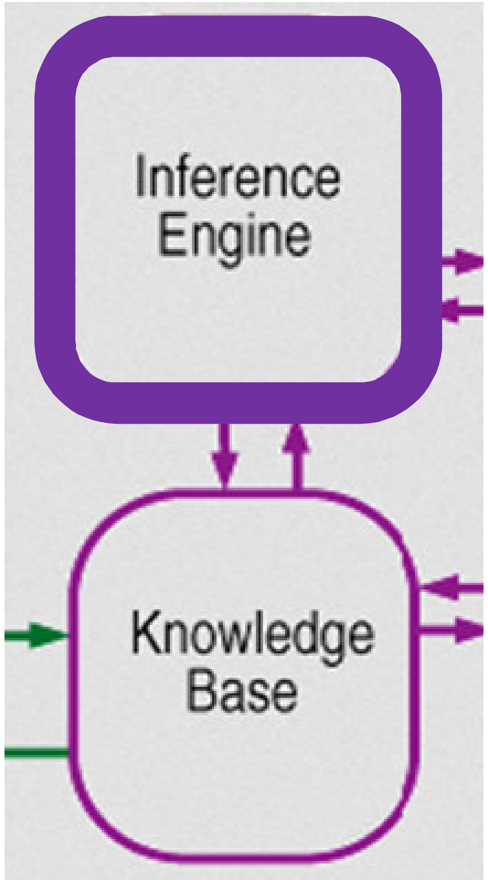
- Maximum loan based on customer characteristics

```
?- give(pete,MaxLoan)

unemployed(pete).
not(insolvent(pete)).

R1: unemployed(P) → highRisk(P).
R2: highRisk(P) → tunableLoan(P,1000).
R3: tunableLoan(P,Amount)AND not(insolvent(P))
        → give(Amount*1.30)
```

Inference Engine

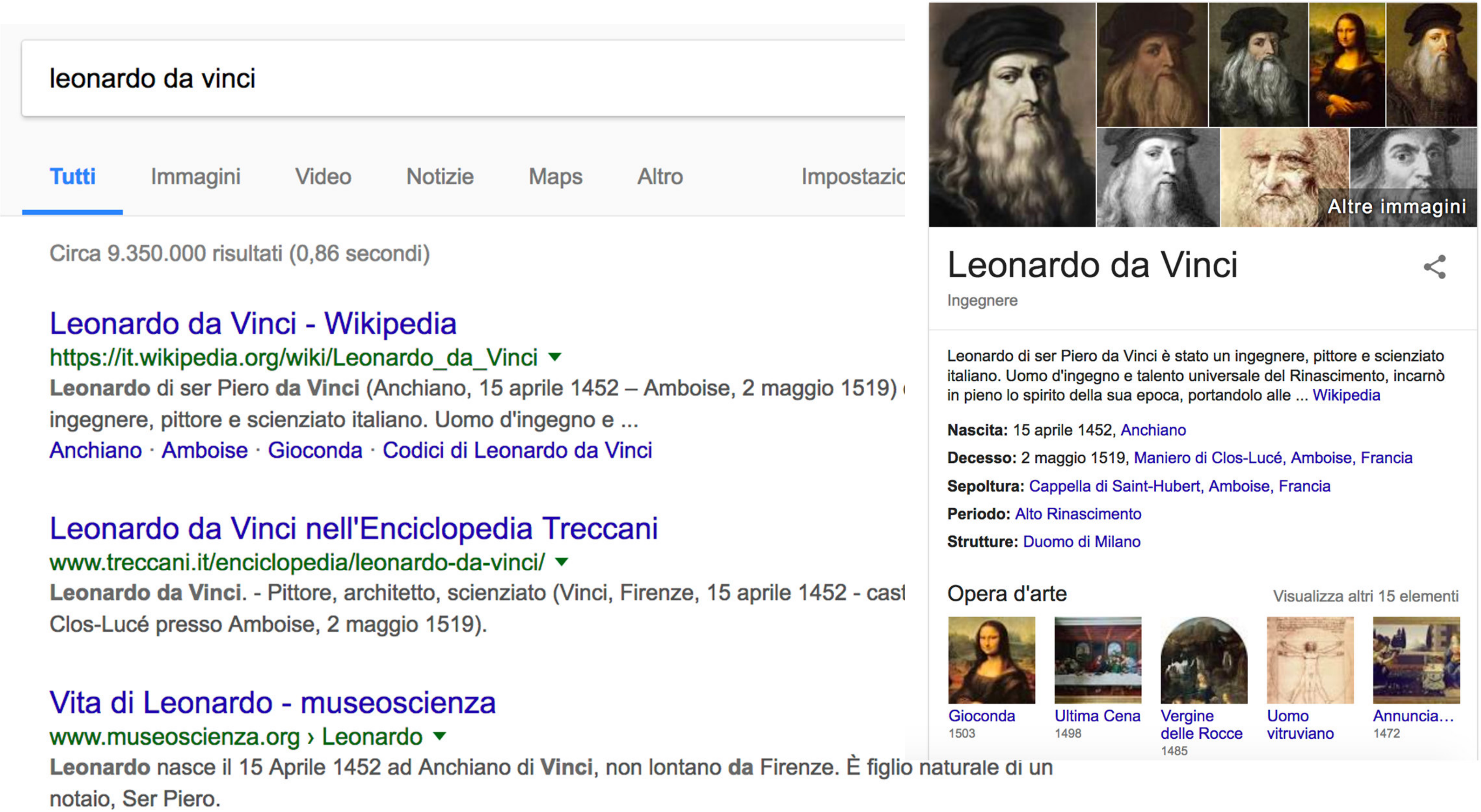
- Independent control module to apply rules
  - forward chaining
  - backward chaining

- Consider possible alternatives in case more than one rule can apply
- Already available (does not require a domain expert)



Google Knowledge Graph

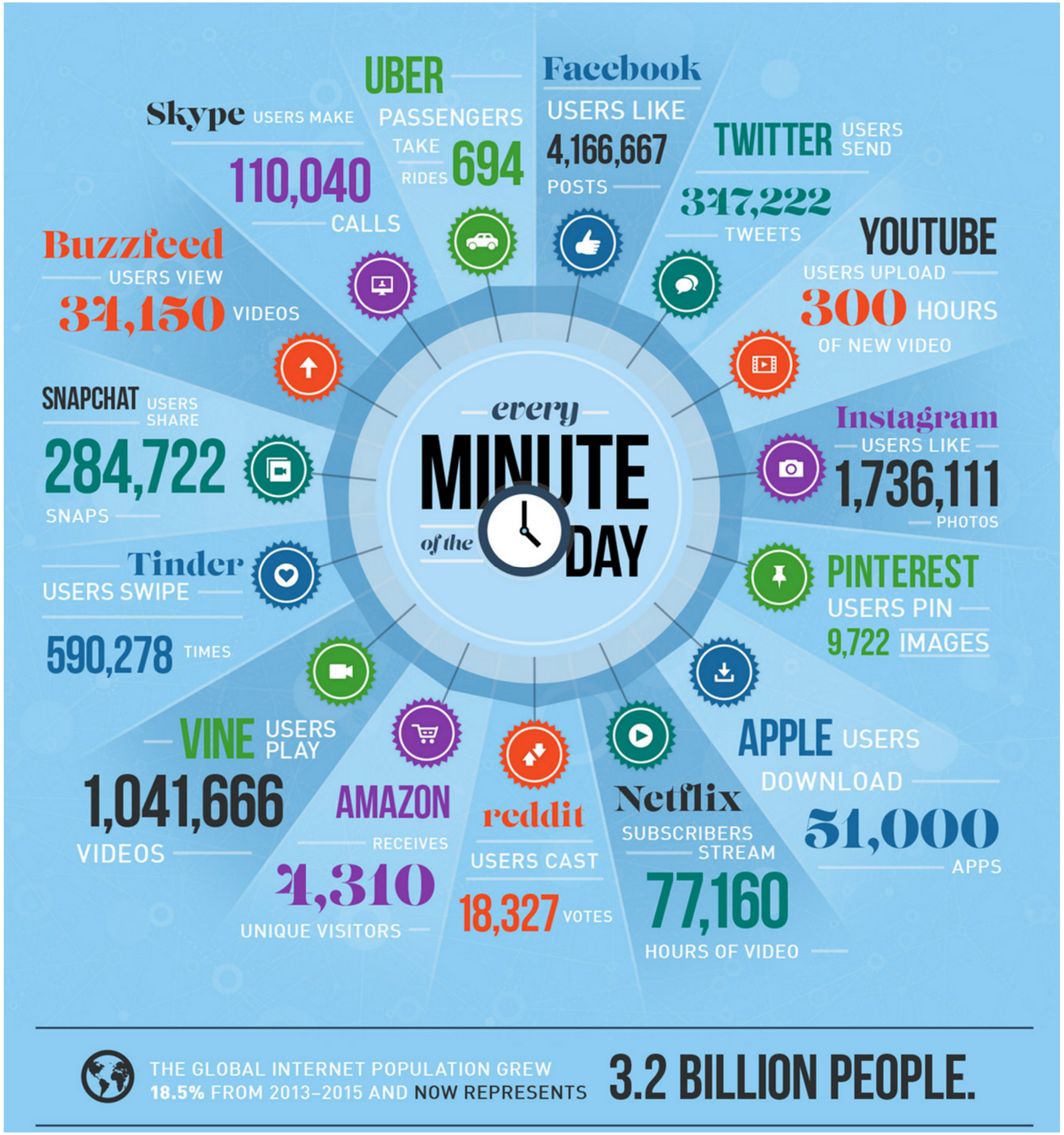
https://developers.google.com/knowledge-graph



Automatic Learning for Big Data

- Huge amount of data generated continuously
- Difficult to explicitly define rules to manage a wide and heterogeneous set of information

- Machine Learning as a tool to get value from Big Data

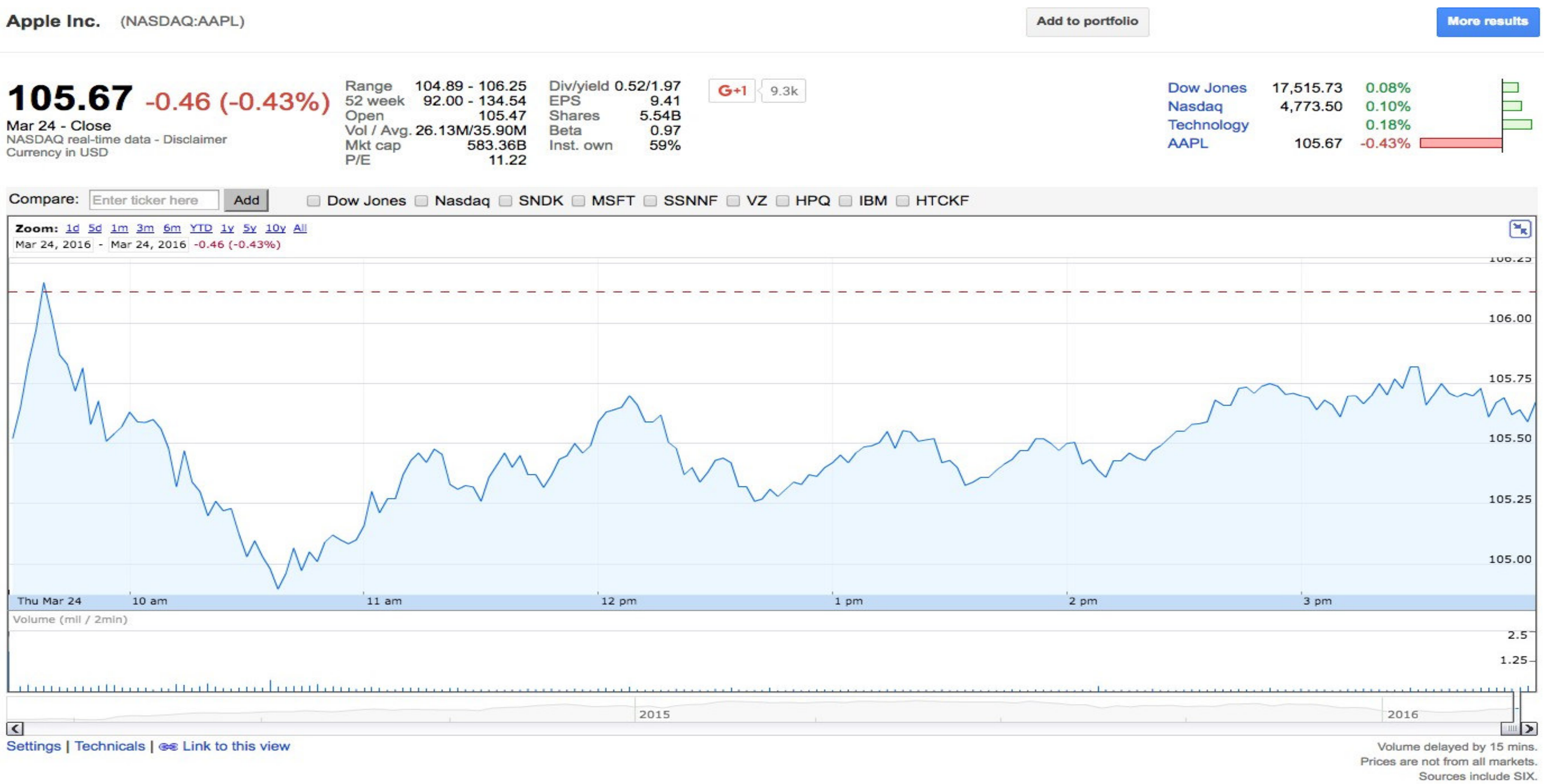


Machine Learning - ML

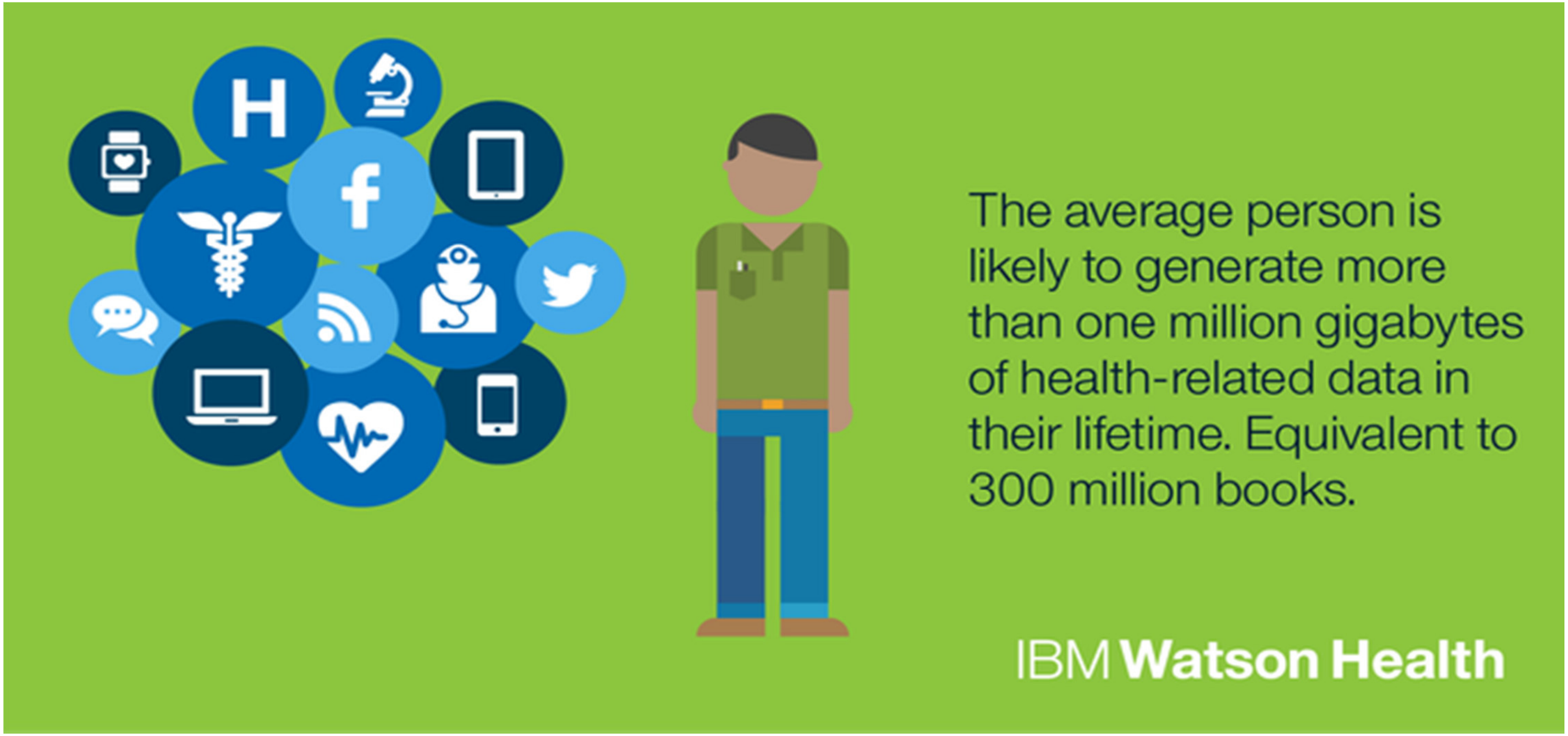
- It is not based on programs or rules describing the model
- It is based on a "program"
  - exploiting actual data of the domain
  - to learn (infer) a model of the domain in an automatic manner
- The model can vary dynamically

- Example: classify emails as spam (

ML: three hour prediction of Apple stock



ML: medical diagnosis



ML: teach a robot how to handle a mug

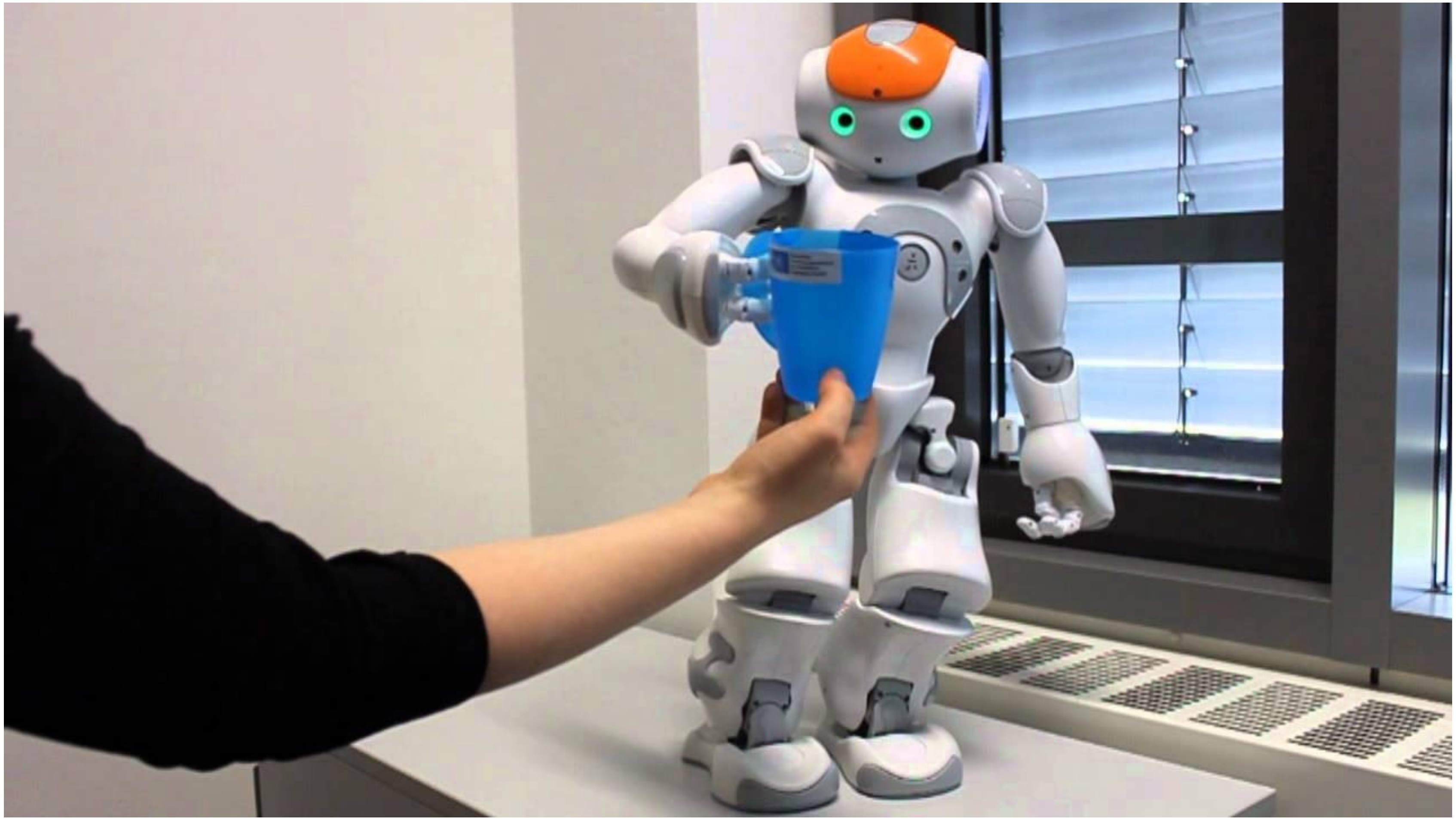
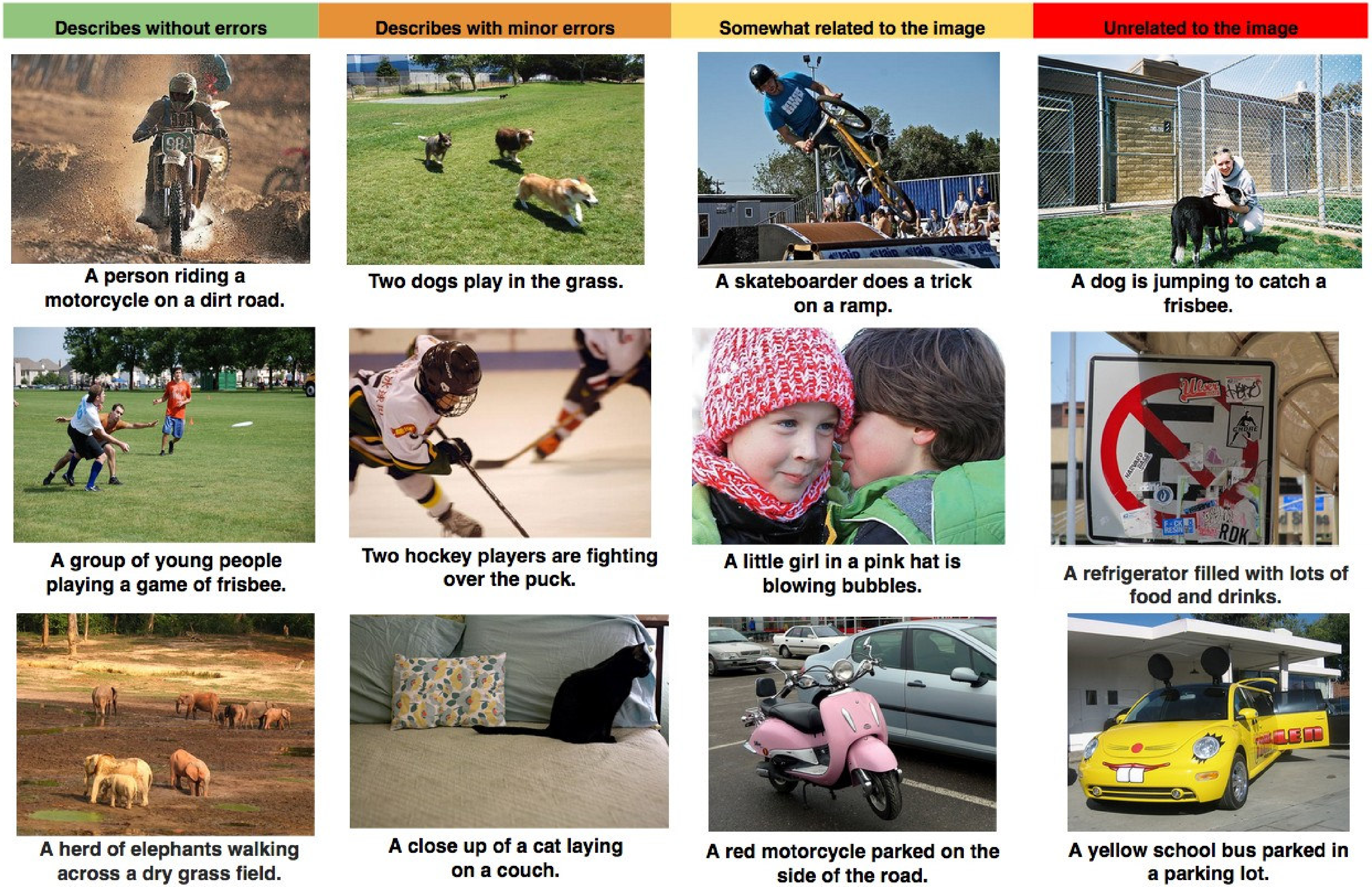


Figure source: http://www.informatik.uni-bremen.de/

ML: automatic labeling of pictures



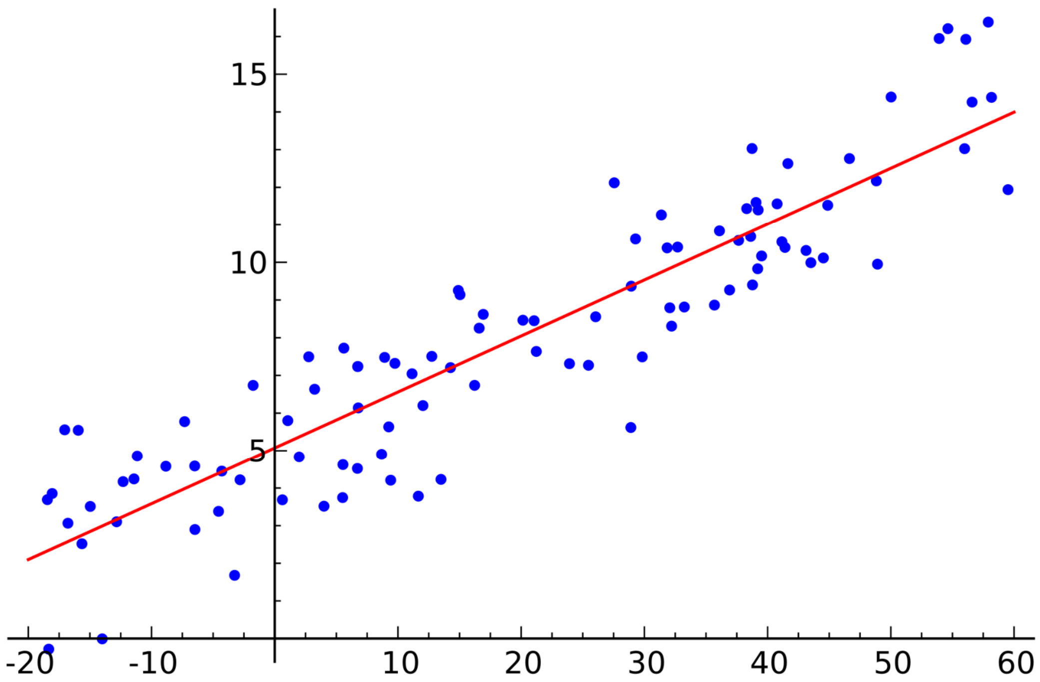
Machine Learning: Supervised Learning

- Training set: previous data exploited to train the model to perform
  - prediction, e.g., linear regression
  - classification of novel data (learning by example)

- Test set: pre-classified data to verify the performance od the system

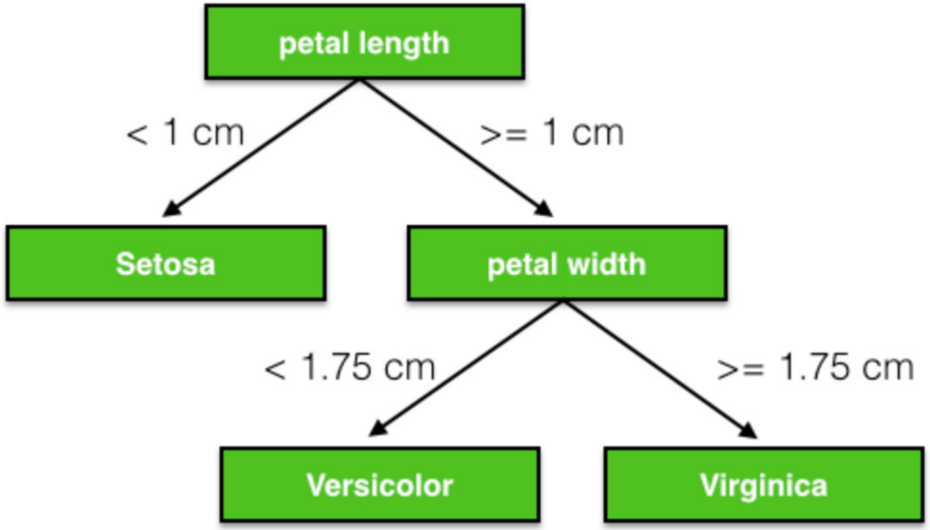
ML: Supervised Learning and Prediction

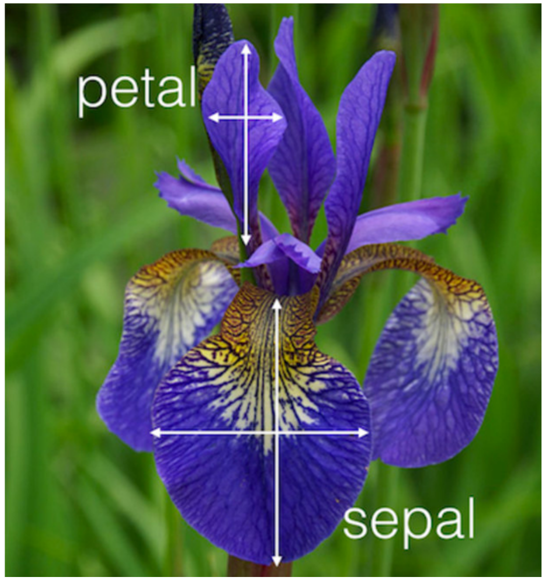
- Least-squares linear regression to predict the amount of yogurt sold in relation to weather/temperature forecast



ML: Supervised Learning and Classification

- Classifying type of Iris based on petal and sepal
- Values of the decision tree automatically inferred exploiting a training set

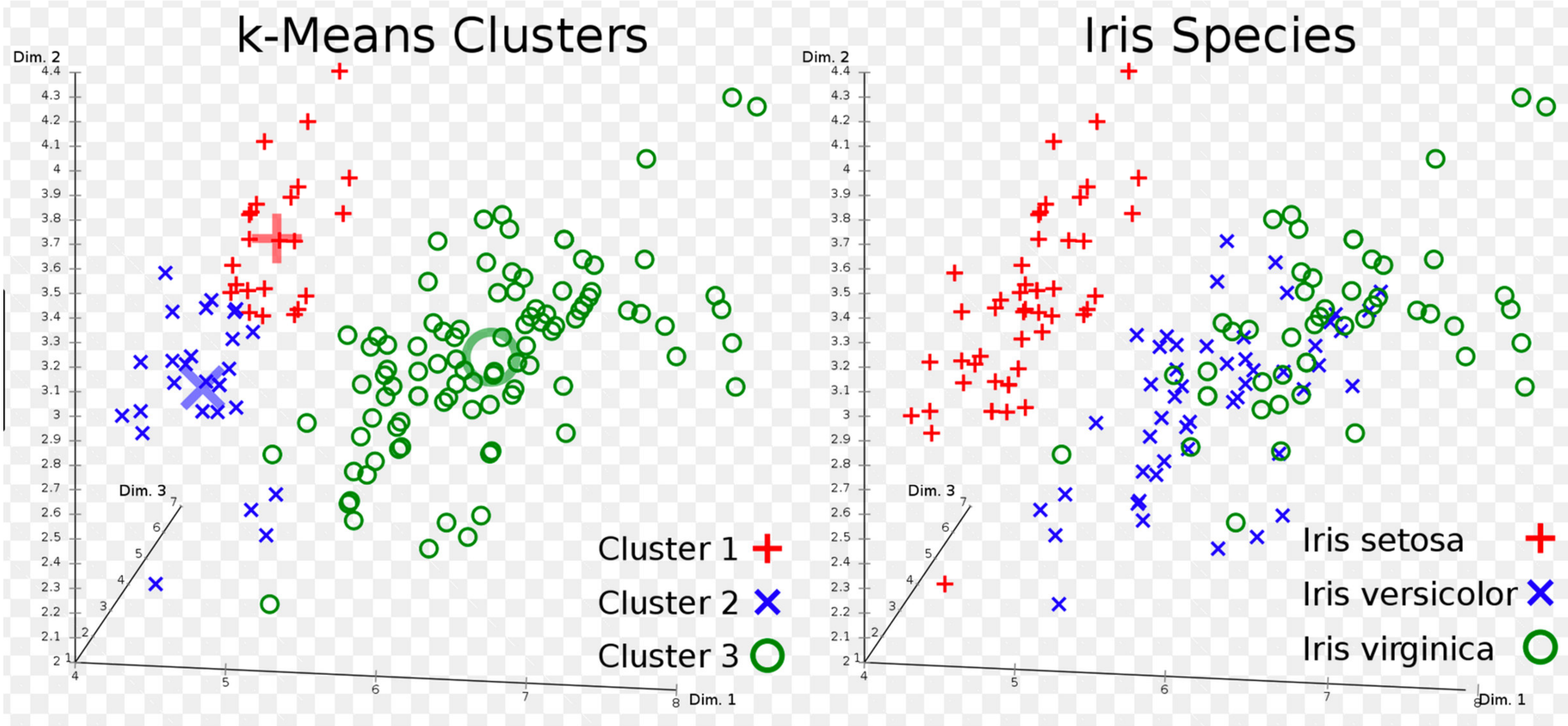




Machine Learning: Unsupervised Learning

- We do not known what to expect!
- Available data automatically split in clusters based on some metric
- Objectives:
  - find some pattern
  - identifying most relevant characteristics
  - reducing the size of data (encoding/compression)

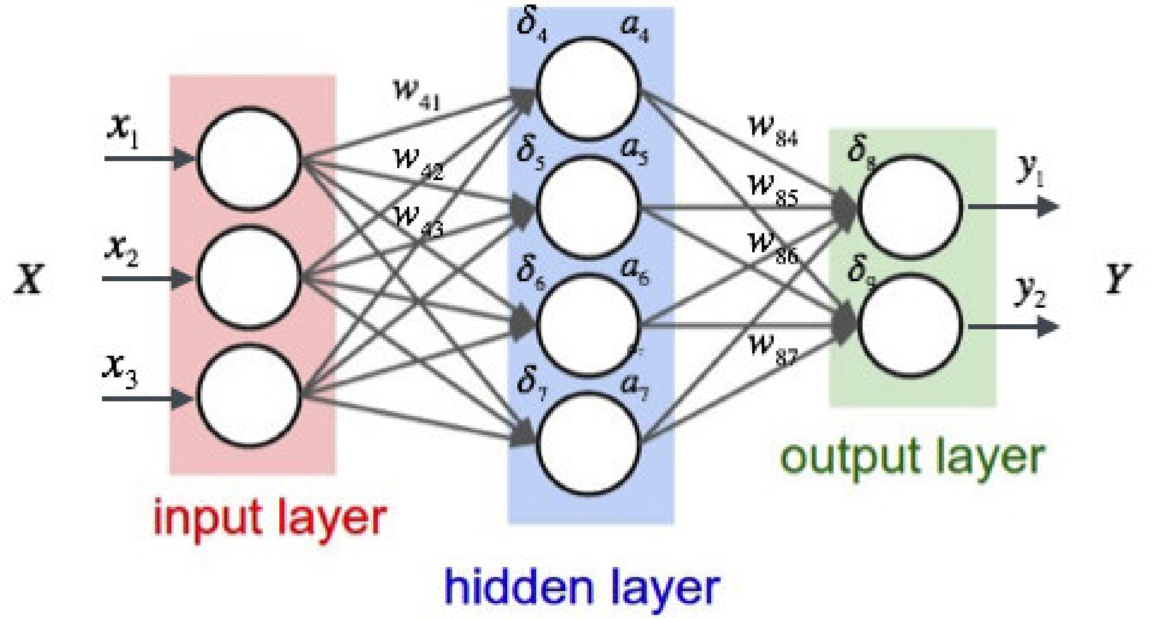
Machine Learning: Unsupervised Learning Example



Machine Learning: Artificial Neural Networks

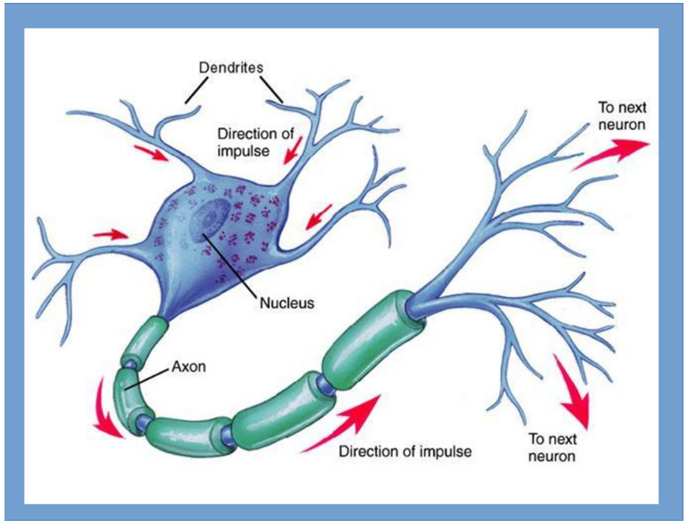
- Trying to **emulate the brain** in a computer
- Building intelligence based on **artificial neurons**

- Three layers:
  - **input**, with sensed data
  - **hidden**, computation from input to output
  - **output**, with final activated neurons

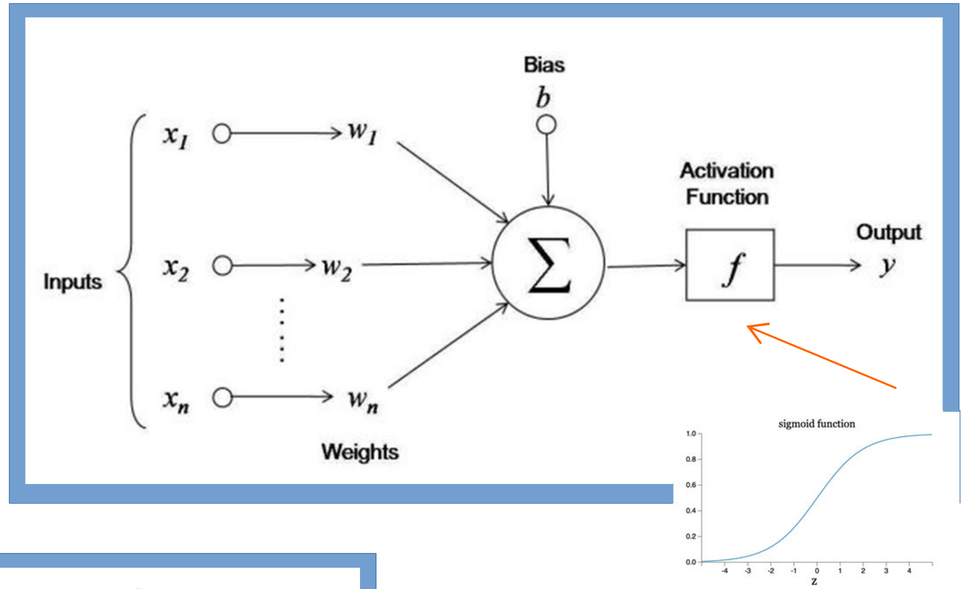


Very simple model of actual neurons and neuron networks

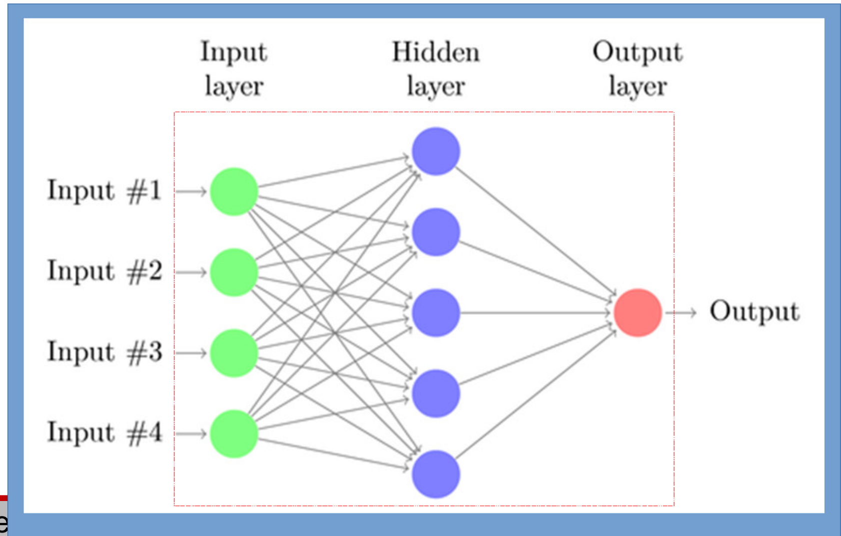
From a biological neuron…



… to an artificial neuron…

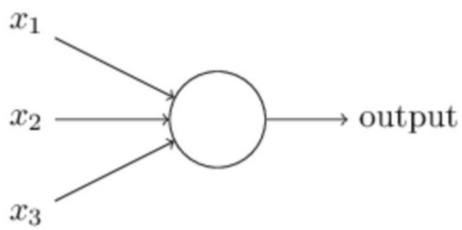


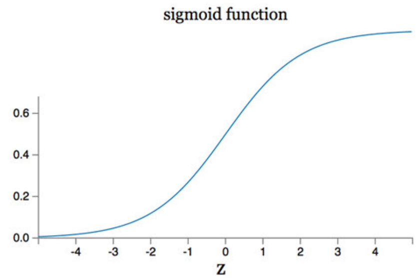
… to create a complex network of neurons.

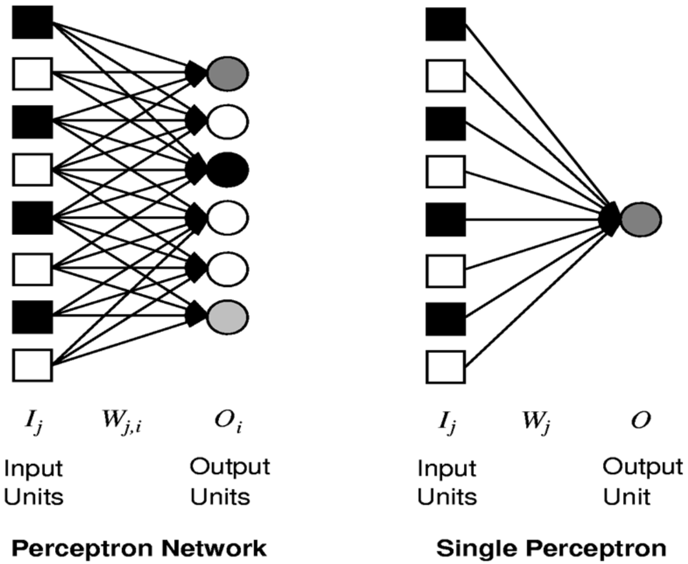


Neural Network: main concepts

- Function to learn, classify, perceive
  - output value based on some formula
  - eventually activating the neuron
- Back-propagation to tune weights with reinforcement learning
  - is the final outcome positive/negative?
  - e.g., winning/loosing a chess game?





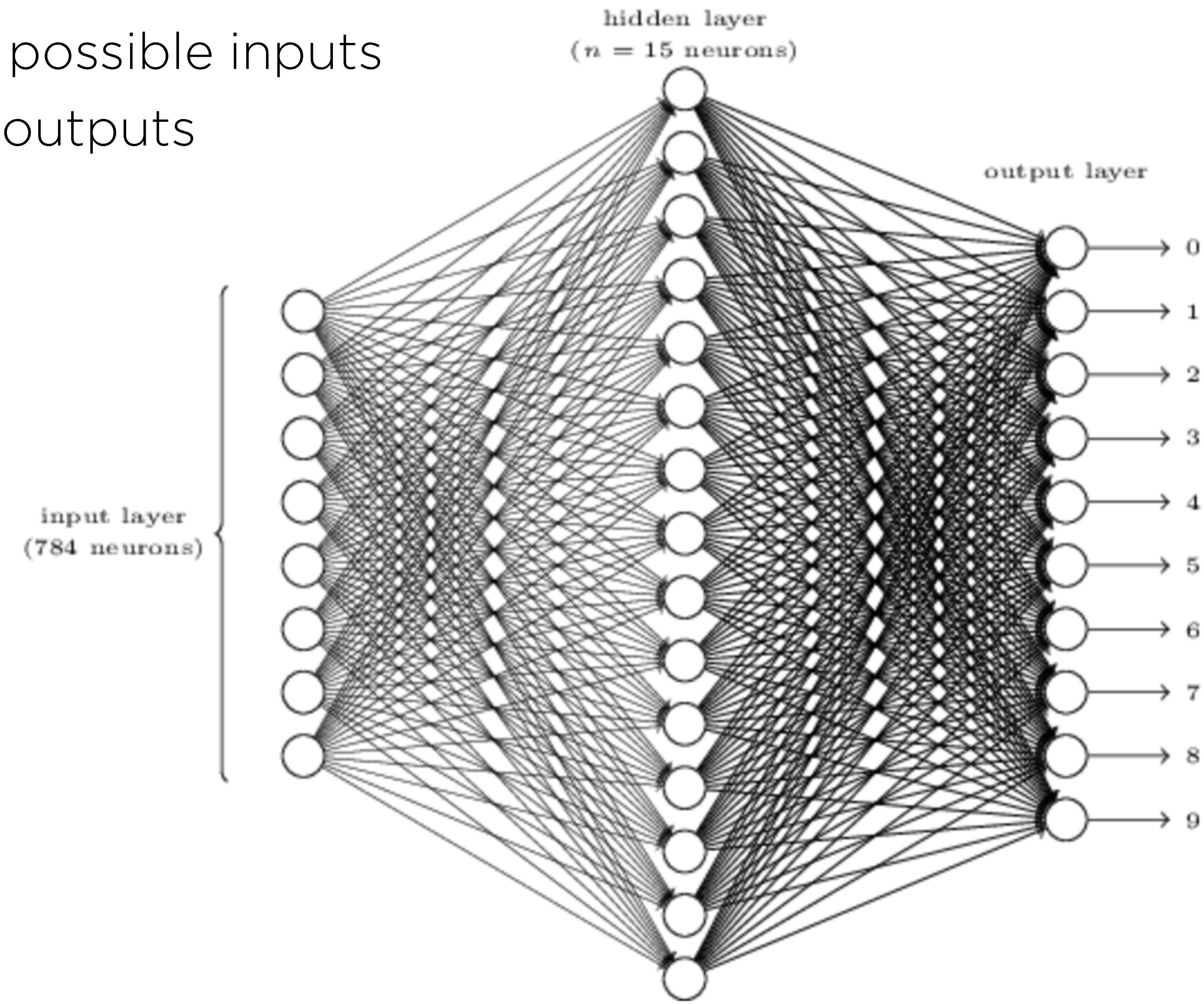


Neural Network: training set of 28X28 digits

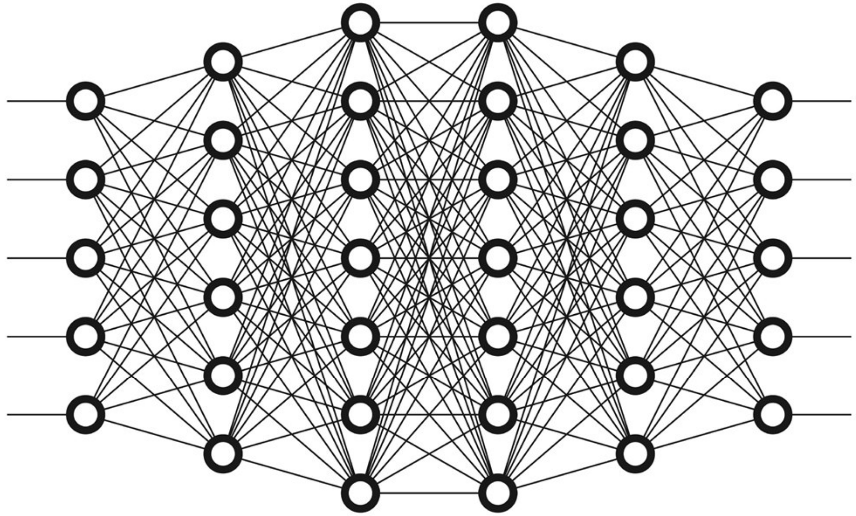
- Identifying hand-written digits

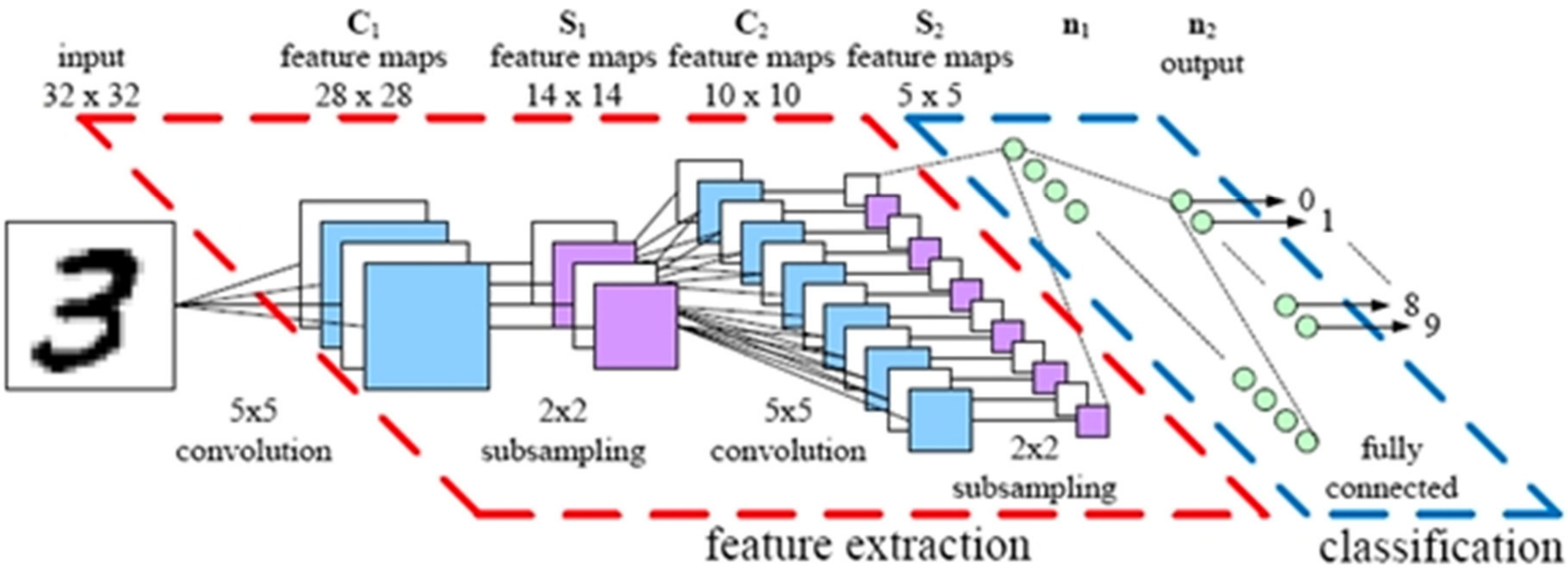
Neural Network: back-propagation to tune weight

- 28x28=784 possible inputs
- 10 possible outputs



Deep Learning: many hidden layers





AI and IoT

- Automated vacuum cleaners, e.g., iRobot Roomba

- Smart thermostat solutions, e.g., Nest Labs

- Self-driving vehicles, such as that of Tesla Motors

Credits

- Thanks to:
  - Paolo Bellavista, University of Bologna
  - Aldo Campi, Stoorm5
  - Antonio Corradi, University of Bologna
  - Carlo Giannelli, University of Ferrara
  - Evelina Lamma, University of Ferrara
  - Fabrizio Riguzzi, University of Ferrara
  - Cesare Stefanelli, University of Ferrara
  - Mauro Tortonesi, University of Ferrara