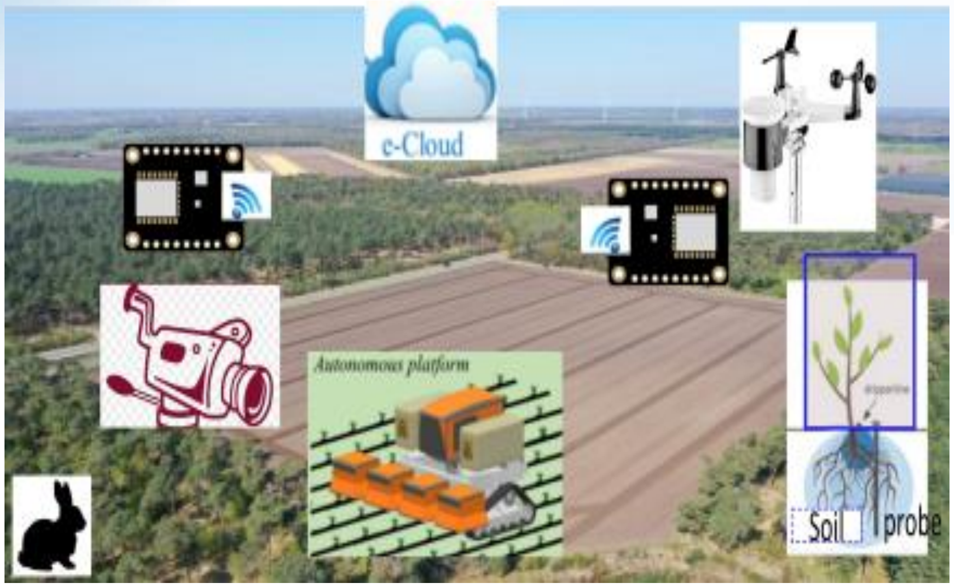
# AI-Autonomous Robots for Agriculture – Weeding with Laser

WeLASER Summer School




## Cloud computing for agriculture

Matteo Francia

2023-07-10

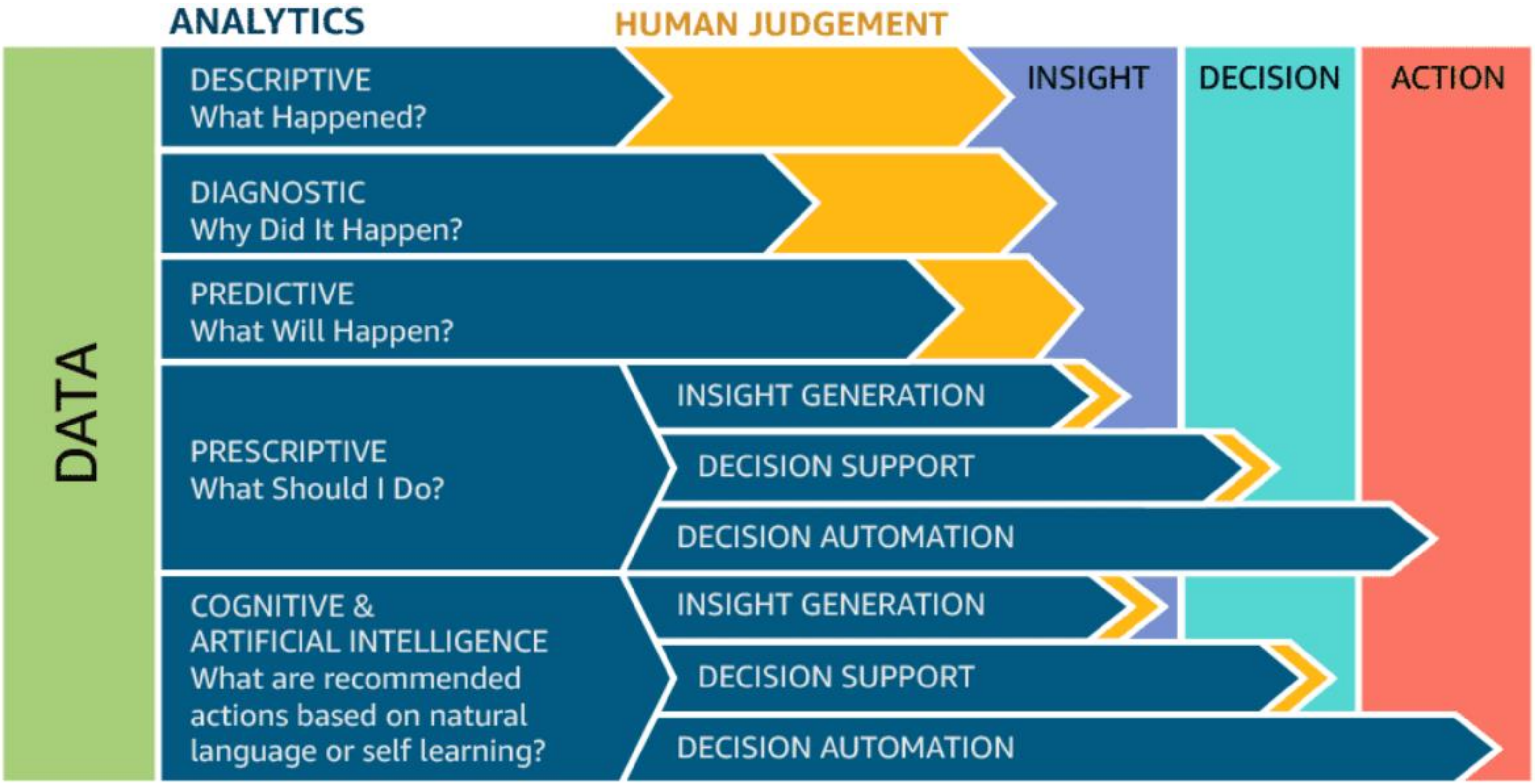
Funded by the Horizon 2020 programme
of the European Union

# Background

# How did we get here?

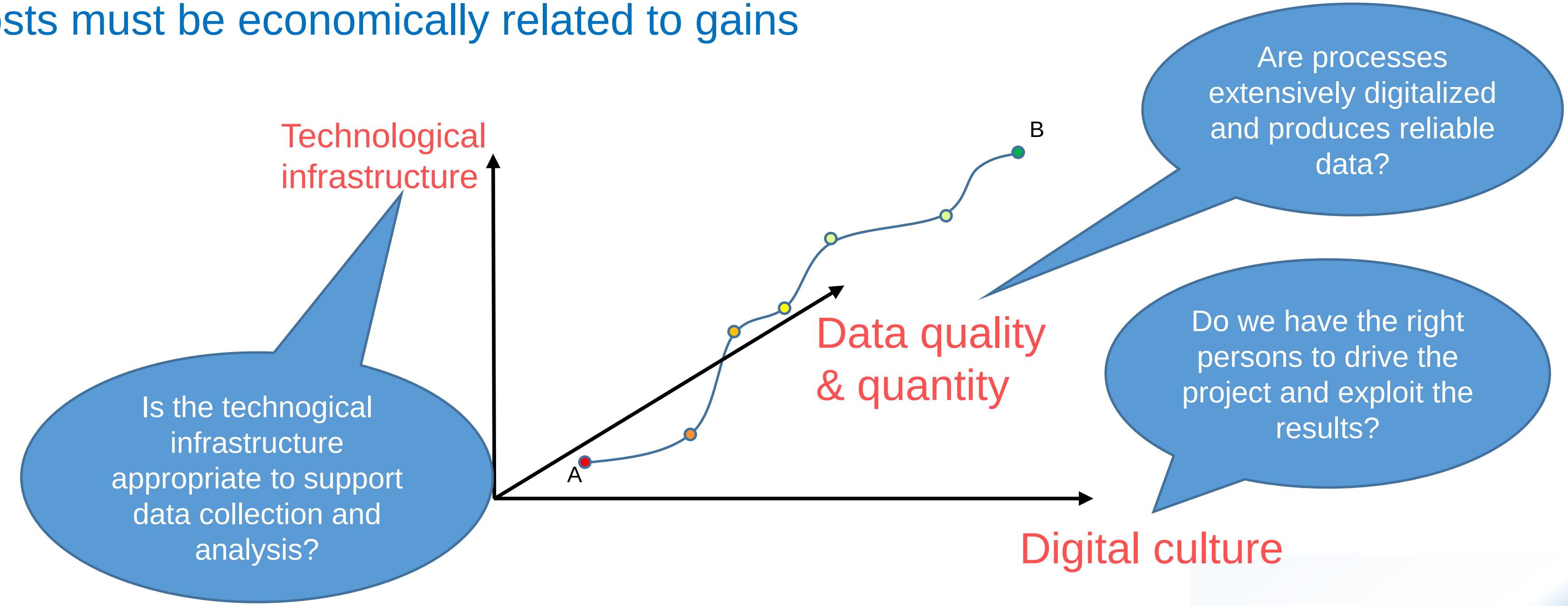
- **Data-Driven Innovation**
  - ❑ Use of data and analytics to foster new products, processes and markets
  - ❑ Drive discovery and execution of innovation, achieving new services with a business value
- **Analytics**
  - ❑ A catch-all term for different business intelligence (BI)- and application-related initiatives
    - ❖ E.g., of analyzing information from a particular domain
    - ❖ E.g., applying BI capabilities to a specific content area (e.g., sales, service, supply chain)
- **Advanced Analytics**
  - ❑ (Semi-)Autonomous examination of data to discover deeper insights, make predictions, or generate recommendations (e.g., through data/text mining and machine learning)
- **Augmented Analytics**
  - ❑ Use of technologies such as machine learning and AI to assist with data preparation, insight generation and insight explanation to augment how people explore and analyze data

# How did we get here?



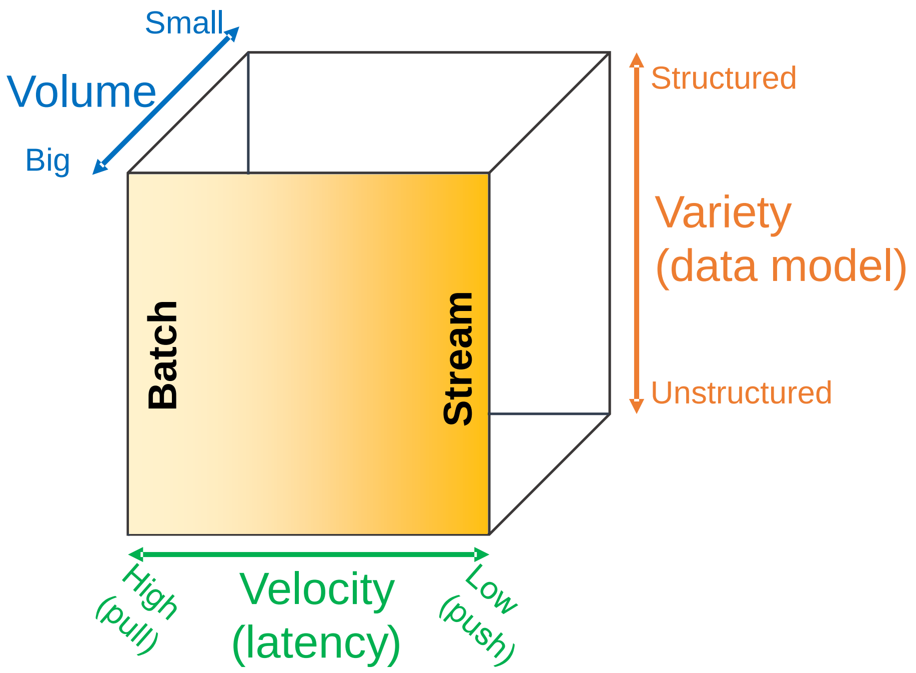
# Why going cloud?

- **Digitalization is a journey that involves three main dimensions**
  - ❑ Moving from A to B is a multi-year process made of intermediate goals
  - ❑ Each of which must be feasible
    - ❖ Solves a company pain and brings value
    - ❖ Can be accomplished in a limited time range (typically less than one year)
    - ❖ Costs must be economically related to gains



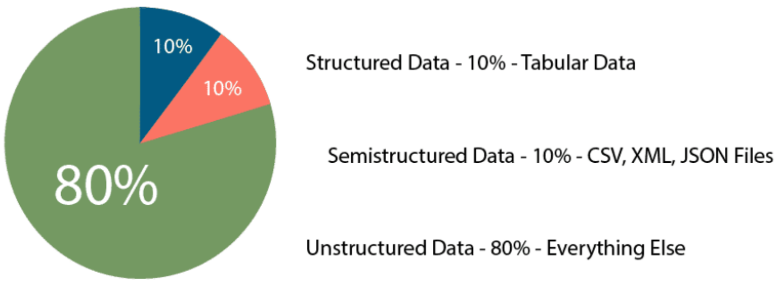
# Big data as a noun – The V's

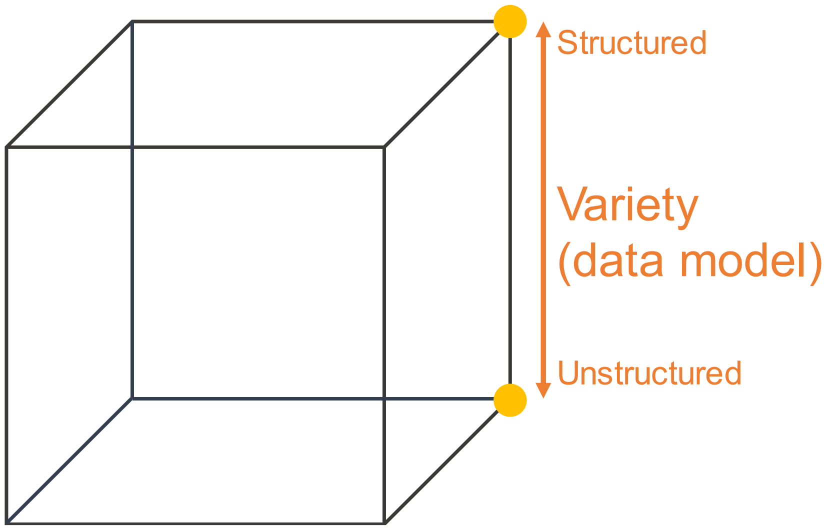
- **The big-data cube**
  - ❑ Volume: small to big
  - ❑ Variety: structure to unstructured
  - ❑ Velocity: pull to push



# Reference scenario

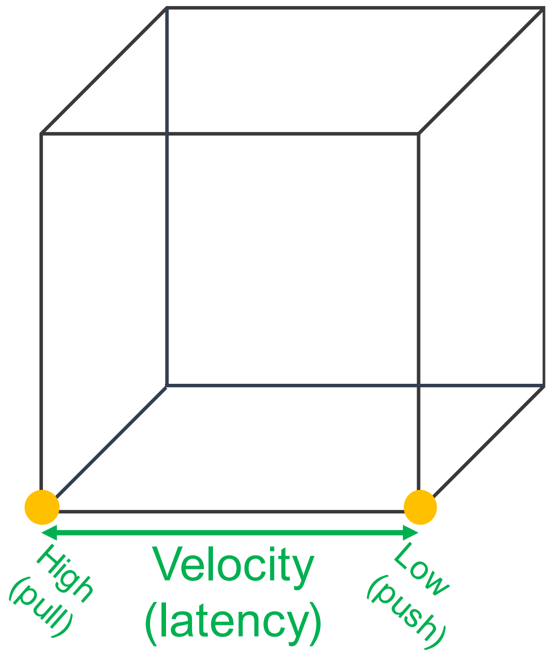
- **Variety**
  - ❑ **Structured**
    - ❖ Relational tuples with FK/PK relationships
  - ❑ **Unstructured**
    - ❖ Key-value
    - ❖ Columnar
    - ❖ Document-based
    - ❖ Graph
    - ❖ ...



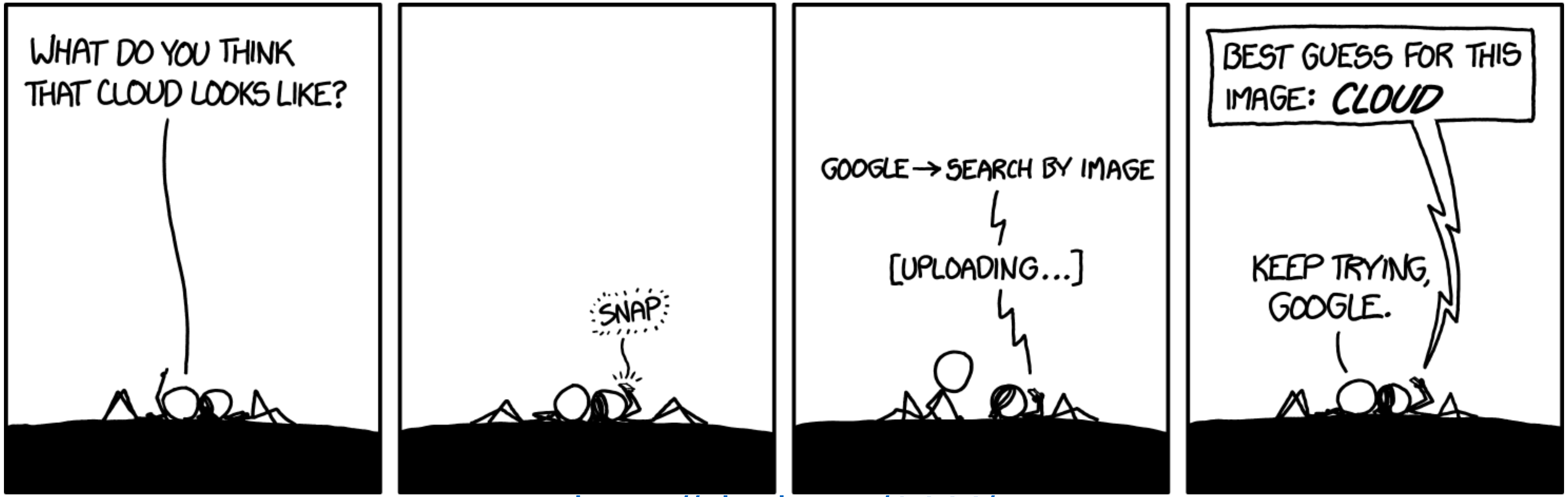



# Reference scenario

- **Velocity (latency)**
  - **High**: clients synchronously pulling data from sources
  - **Low**: sources asynchronously pushing data to clients
- **Velocity (speed; dual to latency)**
  - **High**: processing in real-time (milliseconds) or near-real time (minutes)
  - **Low**: processing can take hours
- **Acceleration**
  - Velocity is not constant, data comes in bursts
  - Take Twitter as an example
    - Hashtags can become hugely popular and appear hundreds of times in just seconds
    - … or slow down to one tag an hour
  - Your system must be able to efficiently handle the peak as well as the lows



# Cloud Computing




https://xkcd.com/1444/

# Why going cloud?

- **Cloud computing (National Institute of Standards and Technology)**

  *"A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."*

  - ❑ On-demand self-service (consume services when you want)
  - ❑ Broad network access (consume services from anywhere)
  - ❑ Resource pooling (infrastructure, virtual platforms, and applications)
  - ❑ Rapid elasticity (enable horizontal scalability)
  - ❑ Measured service (pay for the service you consume as you consume)

- **Digital transformation involves the cloud to create/change business flows**
  - ❑ Often involves changing the company culture to adapt to this new way of doing business
  - ❑ One of the end goal is to meet ever-changing business and market demand

# Why going cloud?

- **Goal: adjusts capacity to have predictable performance at the lowest cost**

- **Scalability that is not possible on premises**
  - ❑ Scale from one to thousands of servers
- **Elasticity**
  - ❑ Automatically scale resources in response to run-time conditions
  - ❑ Adapt to changes in workload by turning on/off resources to match the necessary capacity
  - ❑ Core justification for the cloud adoption

# Why going cloud?

- **Hardware scalability**
  - ❑ No longer think about rack space, switches, and power supplies, etc.
- **Grow storage from GBs to PBs**
  - ❑ 1PB: one hundred 10TB Enterprise Capacity 3.5 HDD hard drives


- https://blog.seagate.com/business/linus-tech-tips-want-petabyte-system/

# Why going cloud?

- **Resource pooling**
  - ❑ Enable cost-sharing, a resource to serve different consumers
  - ❑ Resources are dynamically reassigned according to demands
  - ❑ Based on virtualization, running multiple virtual instances on top of a physical computer system
  - ❑ Economy of scale for physical resources
- **Reliability**
  - ❑ Built to handle failures
  - ❑ Fault-tolerant or highly available

# Why going cloud?

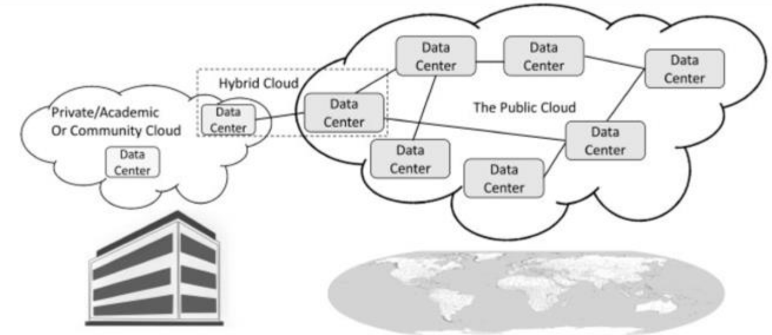
- **Worldwide deployment**
  - ❑ Deploy applications as close to customers as possible
    - ❖ E.g., to reduce network latency
  - ❑ Improve data locality
  - ❑ Compliant to privacy regulations (e.g., GDPR)
- **Measured quality of service**
  - ❑ Services leverage a quantitative qualitative metering capability making pay-as-you-go (or pay-per-use) billing and validation of the service quality available

# Why going cloud?

- **Service integration**
  - Do not reinvent the wheel, eliminate repetitive tasks
    - Use services that solve common problems (e.g., load balancing, queuing)
  - Abstract and automatically adapt the architecture to requirements
    - E.g., create (test) environments on demand
- **Integration and abstraction are drivers of change**
  - From databases to data platforms
  - From on-premises to serverless architectures
  - From custom to standardized data pipelines

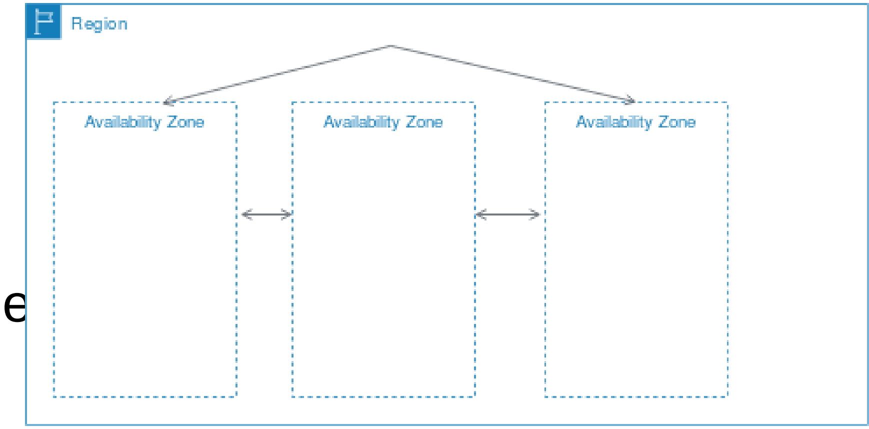
# Cloud computing: types of cloud

- **There are different types of cloud**
  - **Public**: accessible to anyone willing to pay (e.g., Microsoft, AWS, Google)
  - **Private**: accessible by individuals within an institution
    - In public cloud, any resources that you are not using can be used by other
    - Users share the costs
    - Cost-sharing disappears in private clouds
  - **Hybrid**: a mix of the previous



# Cloud computing: types of cloud

- **Cloud services are hosted in separate geographic areas**
  - ❑ Locations are composed of **regions** and **availability zones**
- **Region (e.g., us-east-1)**
  - ❑ Is an independent geographical area that groups data centers
  - ❑ Has availability zones
- **Availability zones in a region**
  - ❑ A data center
  - ❑ Connected through low-latency links
  - ❑ Resources are usually replicated across zone




- https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html

# Cloud computing: principal vendors

- **Gartner Magic Quadrant**
  - ❑ Understanding the technology providers to consider for an investment
  - ❑ **Leaders** execute well and are well positioned for tomorrow
  - ❑ **Visionaries** understand where the market is going but do not yet execute well
  - ❑ **Niche Players** focus successfully on a small segment, or are unfocused and do not out-innovate or outperform others
  - ❑ **Challengers** execute well but do not demonstrate an understanding of market direction
  - ❑ Focusing on leaders isn't always the best
    - ❖ A niche player may support needs better than a market leader. It depends on how the provider aligns with business goals

Figure 1. Magic Quadrant for Cloud Infrastructure and Platform Services

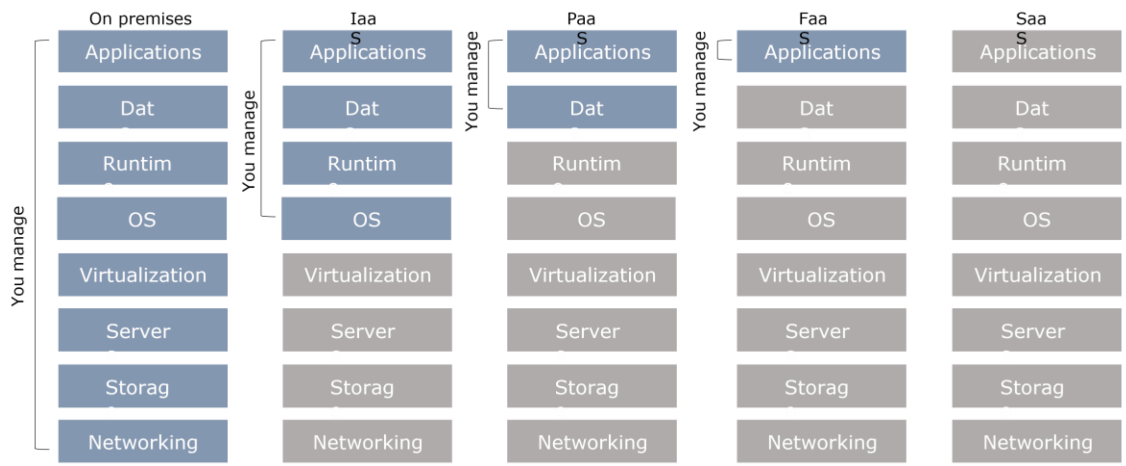
# Cloud computing: deployment models

- **On a cloud architecture, you can rely on serverless or managed services**
- **Serverless**
  - ❑ Standalone independent services built for a specific purpose and integrated by cloud service provider
  - ❑ No visibility into the machines
    - ❖ There are still servers in serverless, but they are abstracted away
    - ❖ No server management, do not have to manage any servers or scale them
    - ❖ E.g., when you run a query on BigQuery you do not know how many machines were used
  - ❑ Pay for what your application uses, usually per request or per usage
- **(Fully) Managed**
  - ❑ Visibility and control of machines
    - ❖ You can choose the number of machines that are being used to run your application
  - ❑ Do not have to set up any machines, the management and backup are taken care for you
  - ❑ Pay for machine runtime, however long you run the machines and resources that your application uses

- https://cloud.google.com/blog/topics/developers-practitioners/serverless-vs-fully-managed-whats-difference (accessed 2020-08-01)

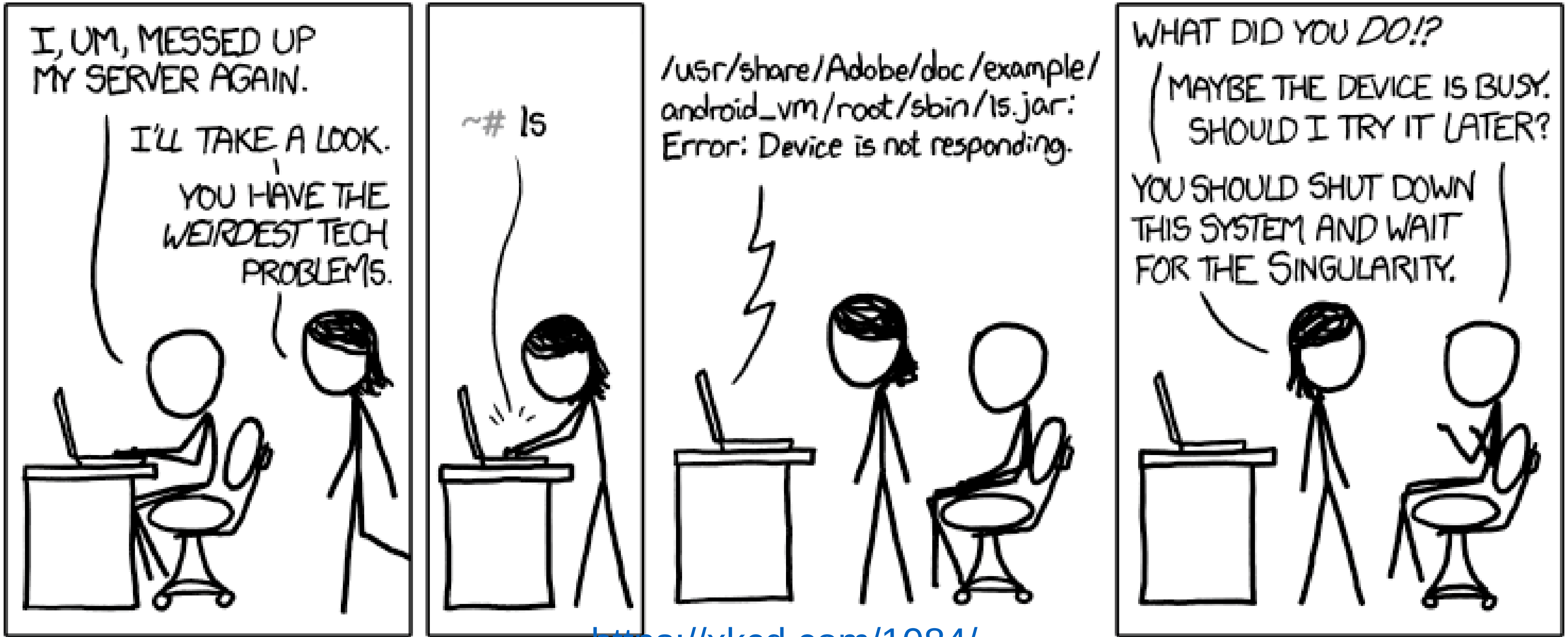
# Cloud computing: deployment models

- **Understanding architectures is paramount to successful systems**
  - ❑ Good architectures help to scale
  - ❑ Poor architectures cause issues that necessitate a costly rewrite
- **XaaS (anything as a service)**
  - ❑ A collective term that refers to the delivery of anything as a service
  - ❑ It encompasses the products, tools and technologies that vendors deliver to users



# Cloud computing: deployment models

- **Principles of FaaS architectures**
  - ❑ FaaS is based on a serverless approach, use a compute service to execute code on demand
  - ❑ Every function could be considered as a standalone service
  - ❑ Write single-purpose stateless functions
- **Functions react to events**
  - ❑ Design push-based, event-driven pipelines
  - ❑ Create thicker, more powerful front ends
  - ❑ Embrace third-party services (e.g., security)
- **FaaS is not a silver bullet**
  - ❑ Not appropriate for latency-sensitive applications
  - ❑ Strict specific service-level agreements
  - ❑ Migration costs
  - ❑ Vendor lock-in can be an issue

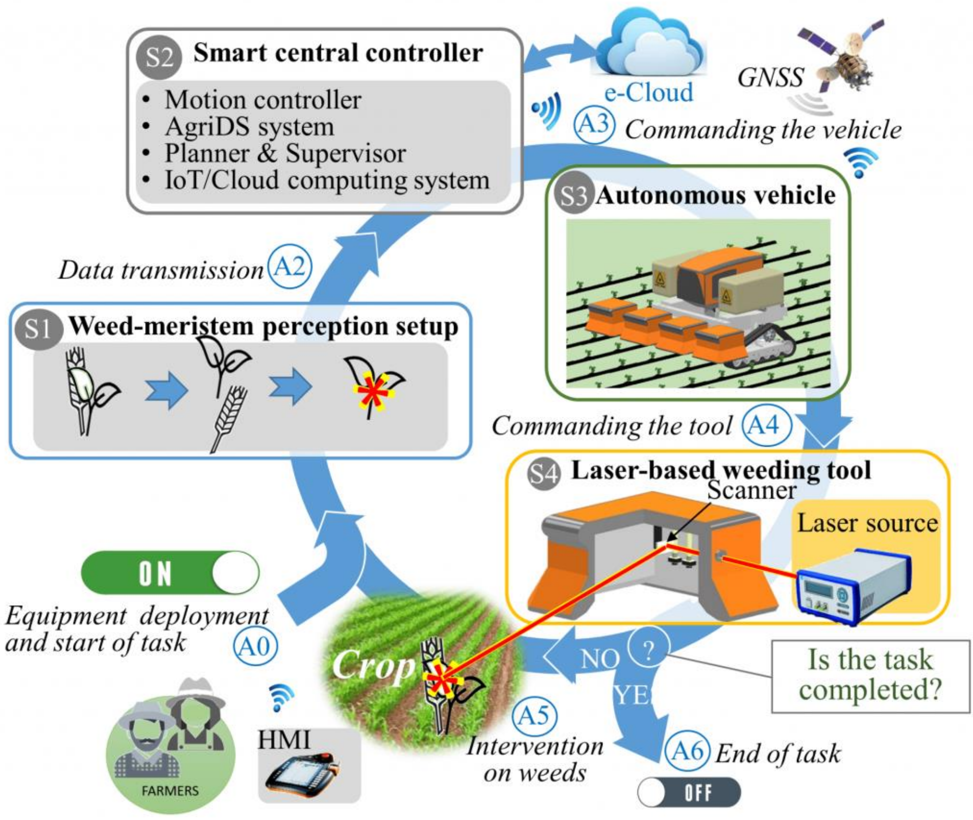



https://xkcd.com/1084/

# Cloud in WeLASER

# WeLASER

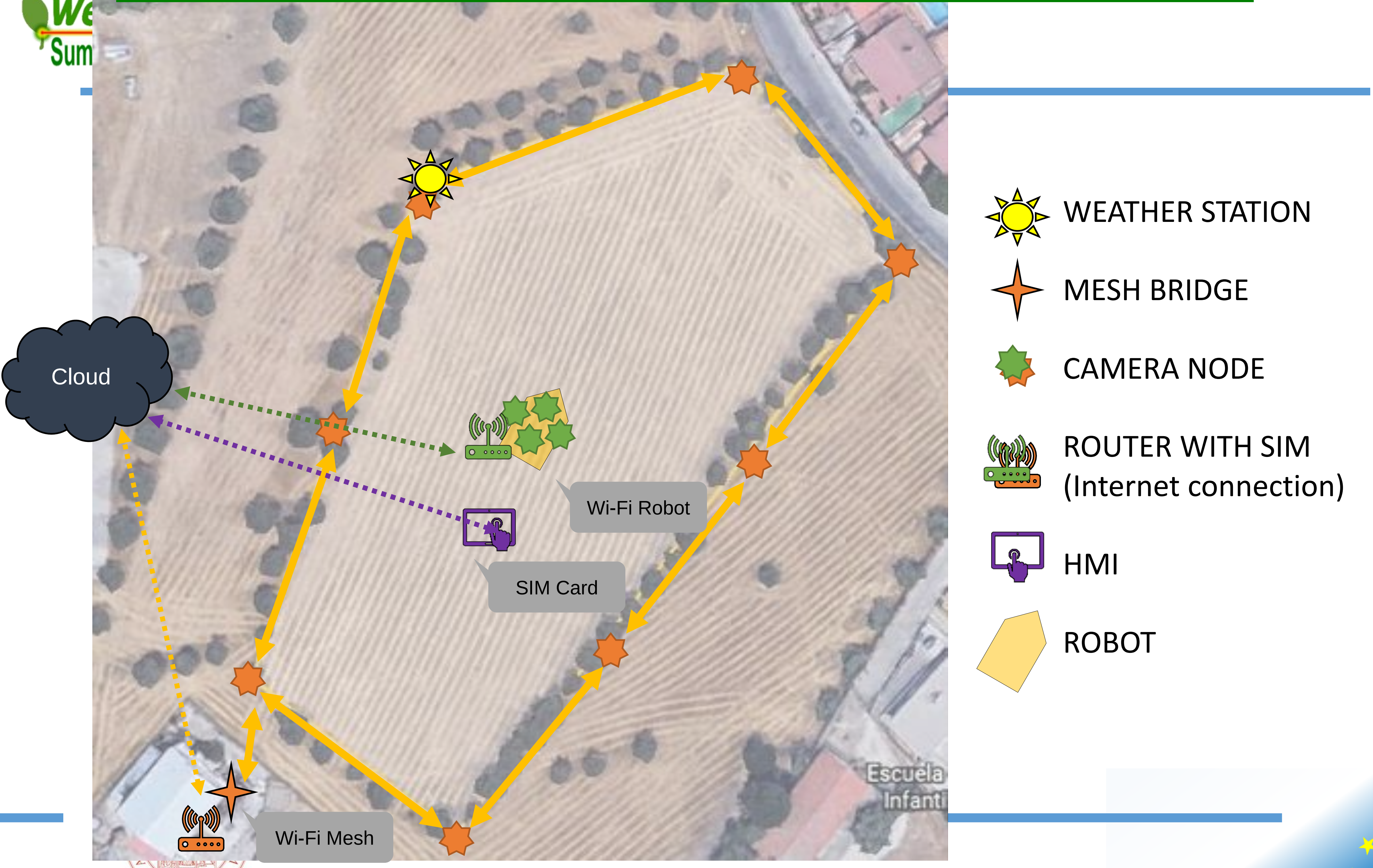
The EU-funded WeLASER project will develop **a non-chemical solution for weed management** based on pioneering technology consisting of the application of lethal doses of energy on the weed meristems **through a high-power laser source**. **An AI-vision system separates crops from weeds**, identifying the weed meristems and pointing the laser at them. **A smart controller based on IoT and cloud computing techniques coordinates the system**, which is transferred all over the field by an autonomous vehicle.



# Network and Data Collection

- **The architecture entails three networks**
  - A wi-fi mesh necessary for exchanging data between the field nodes
    - Wi-fi mesh wakes up after receiving events from the e-fence
  - A wi-fi router on the robot connected to the internet
    - Always turned on
  - A SIM card on the HMI

# Network and Data Collection

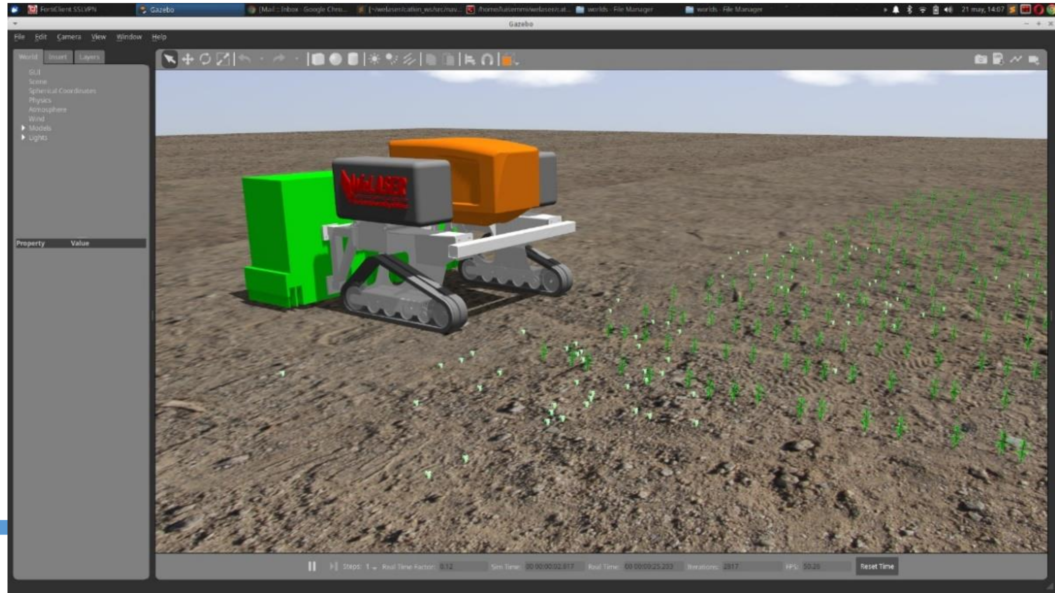


# Requirements

- **A modular data platform for precision agriculture**
  - Collecting data from heterogeneous sources (e.g., MQTT devices and robots)
  - Augmenting data with externally available knowledge (e.g., Open Street Map open data)
  - Building a unifying data model supporting precision agriculture
  - Managing meta-data supporting effective data access and management (e.g., access data depending on the geographical location)
  - Enabling heterogenous data fruition
    - Querying the collected data through well-known languages and APIs
    - Dashboards of KPIs
  - Maintaining digital twins
  - …

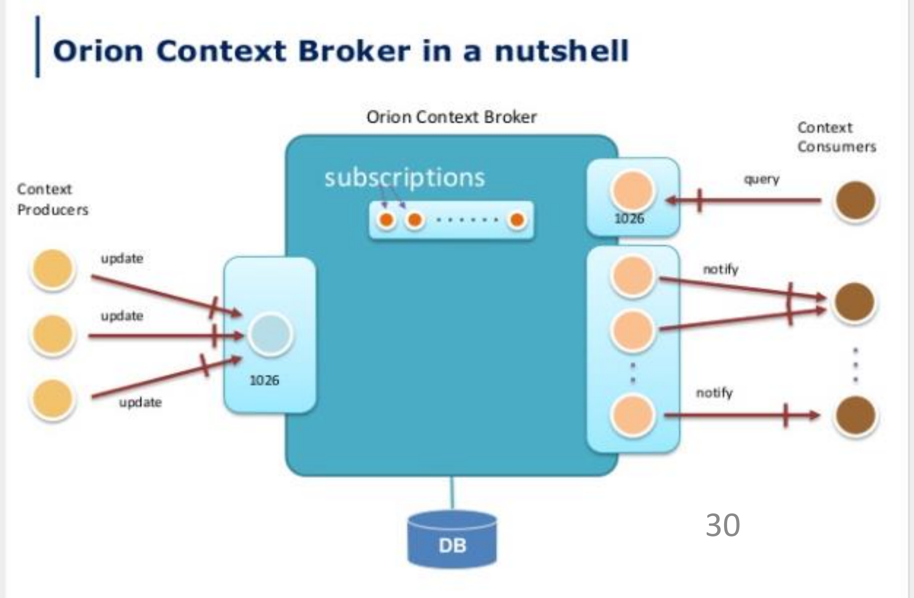
# Digital Twins and Simulation

- **Digital twin**
  - A virtual representation of an object or system that spans its lifecycle, is updated from real-time data, and uses simulation, machine learning and reasoning to help decision making (IBM)
- **Component twins**
  - Basic unit of digital twin, the smallest example of a functioning component
- **Asset twins**
  - Study the interaction of components
- **System twins**
  - How different assets come together to form an entire functioning system
- **Process twins**
  - How systems work together to create an entire production facility



# FIWARE

- An open-source framework of modular software components
  - **Goal**: Developing digital twins and data spaces in several domains
- Only mandatory component of any “Powered by FIWARE” platform is the Context Broker
  - Manage context information
  - Enabling to perform updates
  - Bring access to context
- Smart Data Model includes three elements:
  - Schema defining data types and structure
  - Specification of a written document for human readers
  - Examples of the payloads



# FIWARE

- **GET /v2/entities/{entityID}/attrs/{attrName}**
  - retrieve an attribute’s data
- **PUT /v2/entities/{entityID}/attrs/{attrName}**
  - updates an attribute’s data
- **DELETE /v2/entities/{entityID}/attrs/{attrName}**
  - deletes an attribute
- **GET /v2/entities/{entityID}/attrs/{attrName}/value**
  - retrieves an attribute’s value
- **PUT /v2/entities/{entityID}/attrs/{attrName}/value**
  - updates an attribute’s value
- **GET /v2/entities/{entityID}/attrs/{attrName}**
  - retrieve an attribute’s data
- **PUT /v2/entities/{entityID}/attrs/{attrName}**
  - updates an attribute’s data
- **DELETE /v2/entities/{entityID}/attrs/{attrName}**
  - deletes an attribute
- **GET /v2/entities/{entityID}/attrs/{attrName}/value**
  - retrieves an attribute’s value
- **PUT /v2/entities/{entityID}/attrs/{attrName}/value**
  - updates an attribute’s value

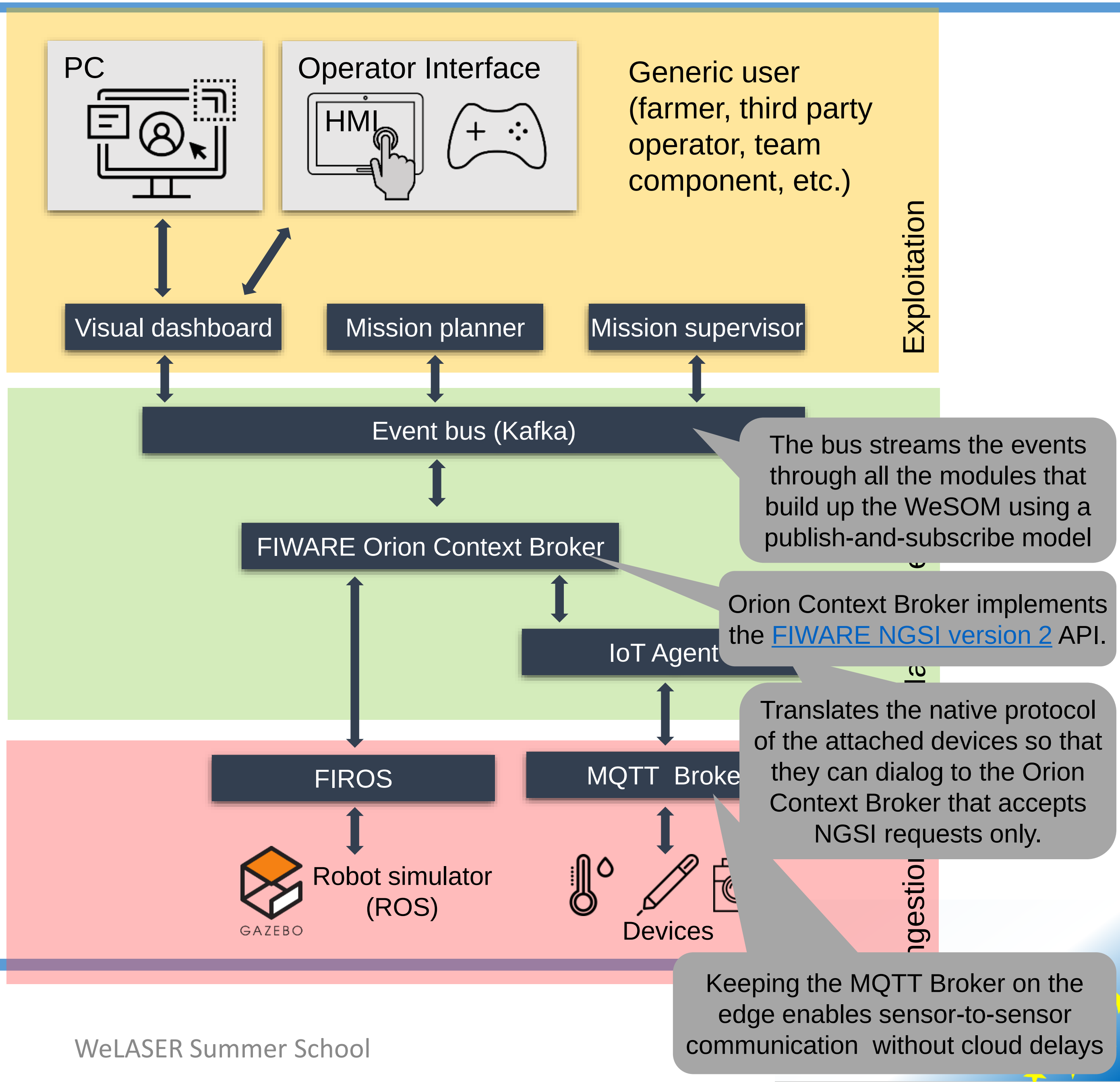
# Smart Data Models

### Example of FIWARE entity and Weeding Heatmap


```
{
    "type": "Device",
    "name": “Laser-123",
    "id": "urn:ngsi-ld:Device:Laser123",
    "areaServed": "urn:ngsi-ld:AgriFarm:123",
    "location": { "type": "Point", "coordinates": [-3.481, 40.312 ] },
    "timestamp": “2023-04-10T08:30:59”
    "controlledProperty": [ “detectedWeeds”, “treatedWeeds” ],
    "value": [ 123, 100 ]
}
```

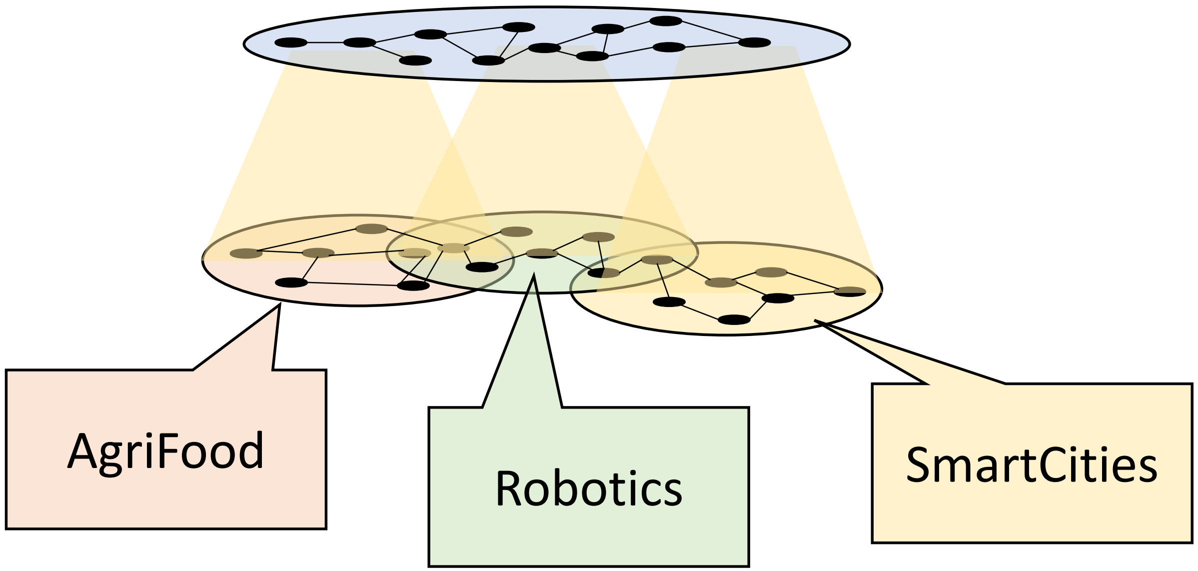

# Innovation: Data Platform for Precision Agriculture

- Visual dashboard
  - A graphical user interface which provides at-a-glance views of key performance indicators (KPIs)
  - Real-time and historical data
- Operator interface = HMI + Controller
- HMI
  - A **mobile tablet** (e.g., iPad) running a web-based application
  - As of now, requires internet connection
- Controller
  - A "**joystick**" used both for safety and remote manual control



# Smart Data Models

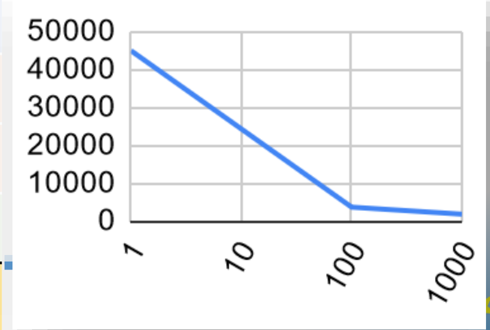
- **Many smart data models from the FIWARE ecosystem exist**
  - ❑ E.g., AgriFood, Robotics, SmartCities
  - ❑ DMs are overlapping and sometimes inconsistent between each other
    - ❖ E.g., AgriOperation and BuildingOperations have different semantics
  - ❑ DMs are good for representing data, but «bad» for analytics
- **Create a unifying meta-model to integrate them**



# Cloud and Economy of Scale

- Cloud costs (on AWS) per 1/100/1000 licenses
  - ❑ Hardware
    - ❖ Computing:
      - **Fixed**: machines that must always be up and running
      - **Spot**: machines that are turned only when concurrent missions are running
    - ❖ Storage:
      - **MongoDB**: data from FIWARE Context Broker (i.e., current context) and historic data
      - **S3**: raw mission images
      - **No HDFS**: more expensive than switching to S3 unless HW for storage is already available
  - ❑ People (8h/day): customer care (answering tickets) + developers (maintenance)
  - ❑ Software: cost of the single software license

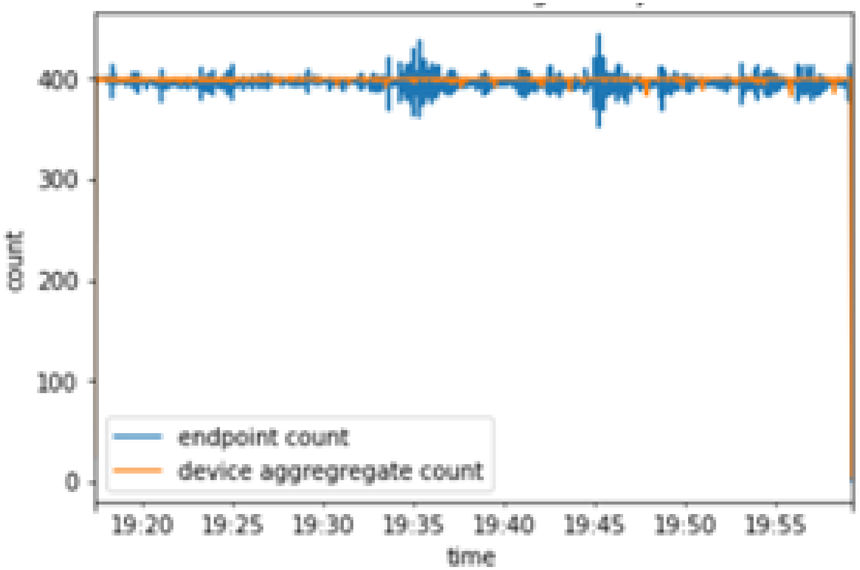
| Cost type | #Licenses | 1 | 100 | 1000 |
|---|---|---|---|---|
| Hardware | Computing (Fixed + Spot) | 3 000€ | 15 000€ | 120 000€ |
| | Storage (MongoDB + S3) | 1 000€ | 15 000€ | 130 000€ |
| People | Customer care | 0€ | 120 000€ (3 people) | 400 000€ (10 people) |
| | Developer | 40 000€ (1person) | 120 000€ (3 people) | 200 000€ (5 people) |
| Software | License | 1 000€ | 100 000€ | 1 000 000€ |
| | Total cost | 45 000€ | 370 000€ | 1 850 000€ |
| | **Per license cost** | **45 000€** | **3 700€** | **1 850€** |



# Performance

- We implemented a FIWARE architecture to get practiced with it and to stess it performances and limitations
- We handled up to 600 messages per second

200 entities, 1 message every 500ms




200 entities, 1 message every 100ms