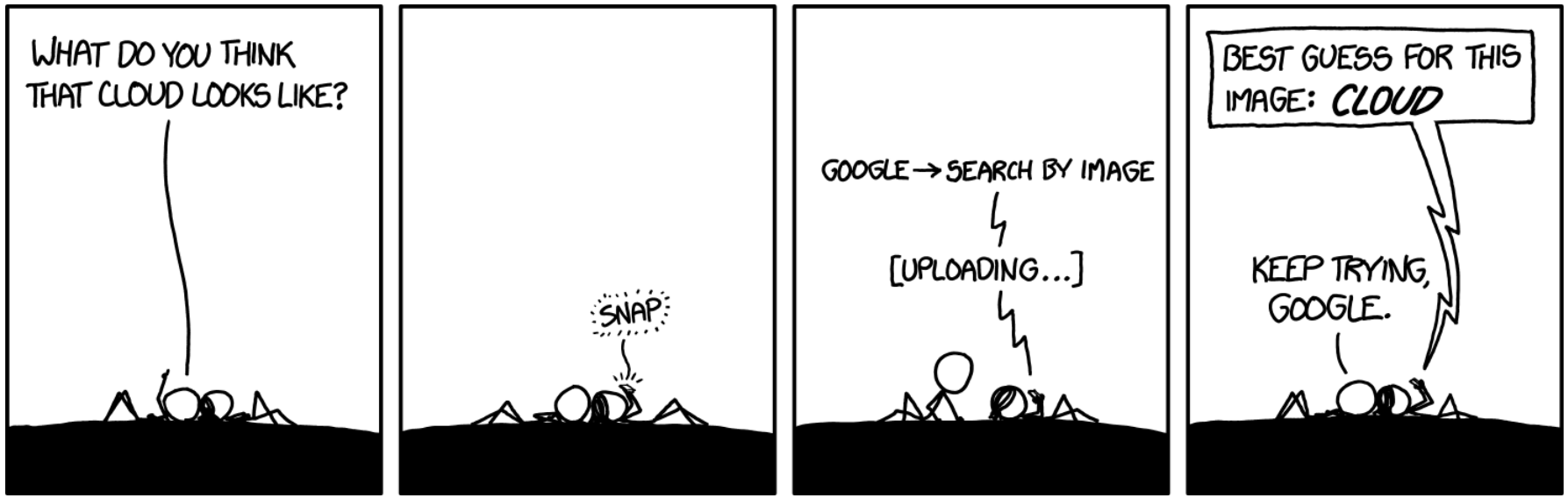
BIG DATA AND CLOUD PLATFORMS

Cloud computing

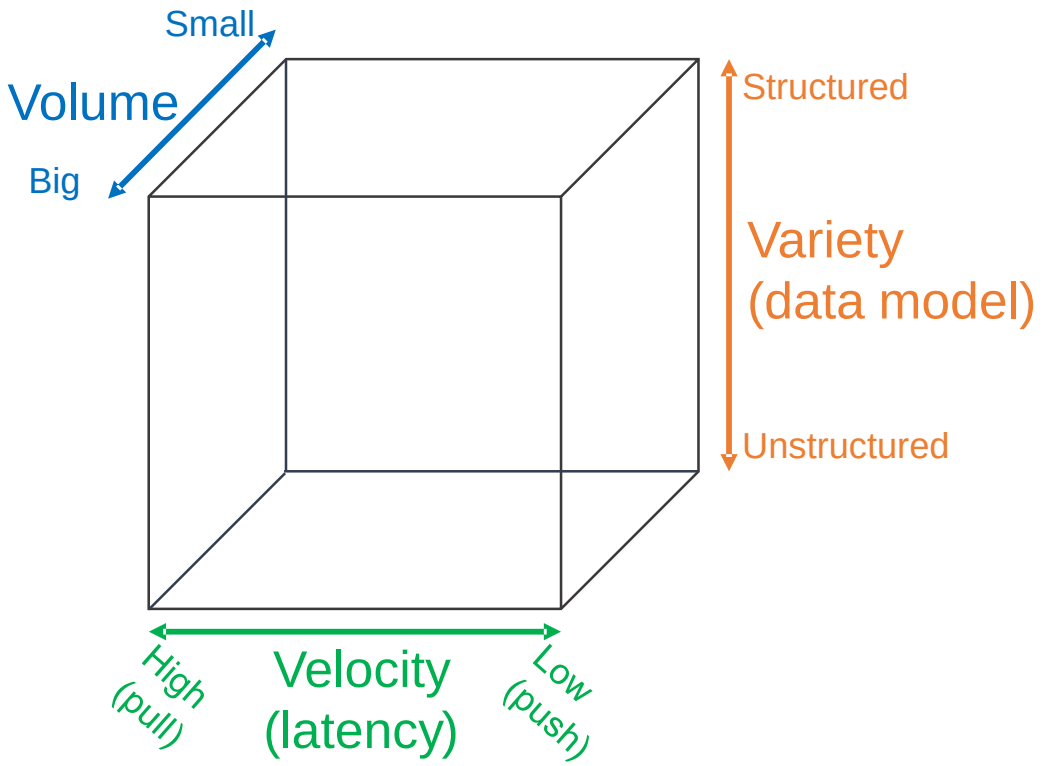


https://xkcd.com/1444/

Reference scenario

The big-data cube

- Volume: small to big
- Variety: structure to unstructured
- Velocity: pull to push

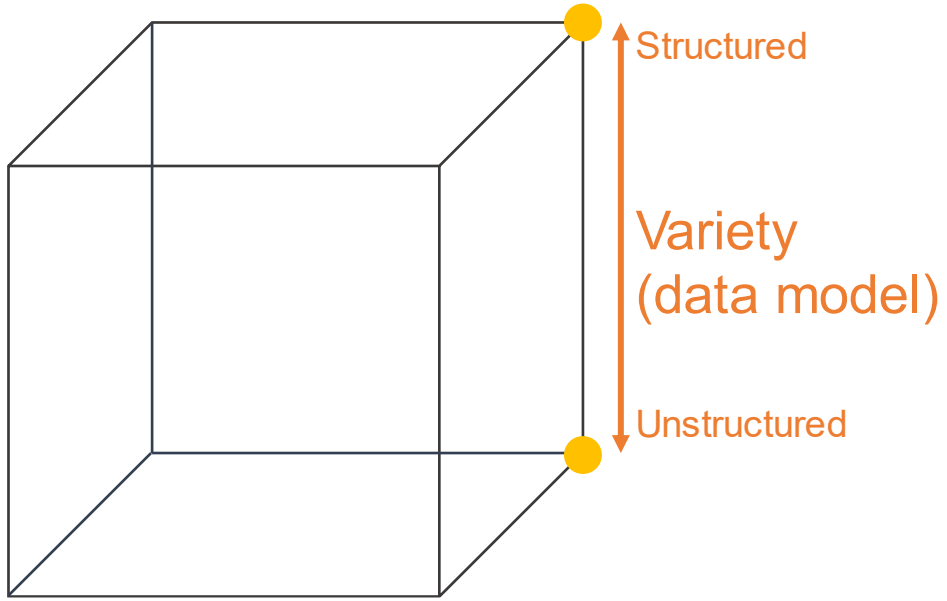


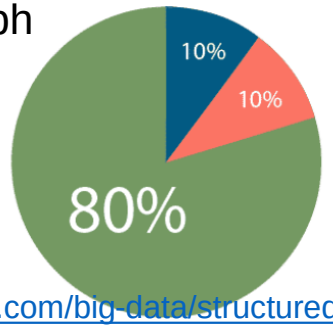
Meijer, Erik. "Your mouse is a database." *Communications of the ACM* 55.5 (2012): 66-73.

Reference scenario

Variety

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





Structured Data - 10% - Tabular Data

Semistructured Data - 10% - CSV, XML, JSON Files

Unstructured Data - 80% - Everything Else

https://www.datamation.com/big-data/structured-vs-unstructured-data/ (accessed 2022-08-01)

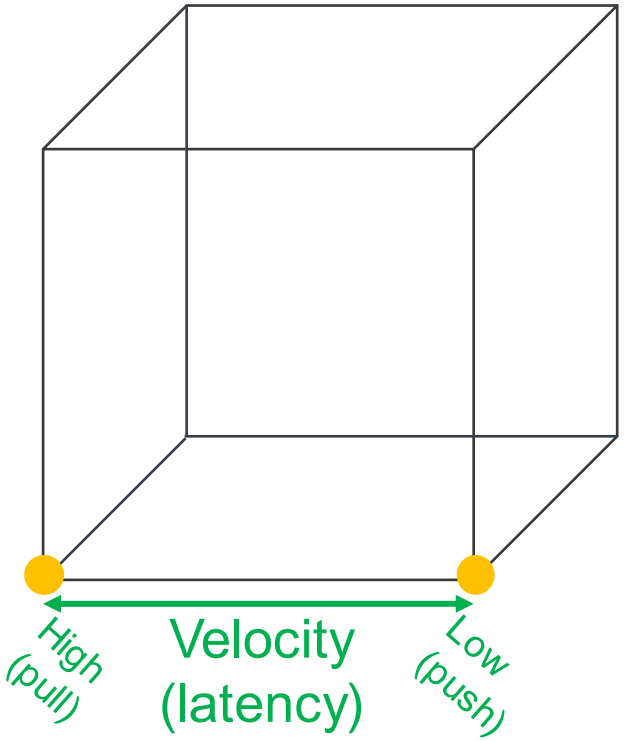
Reference scenario

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

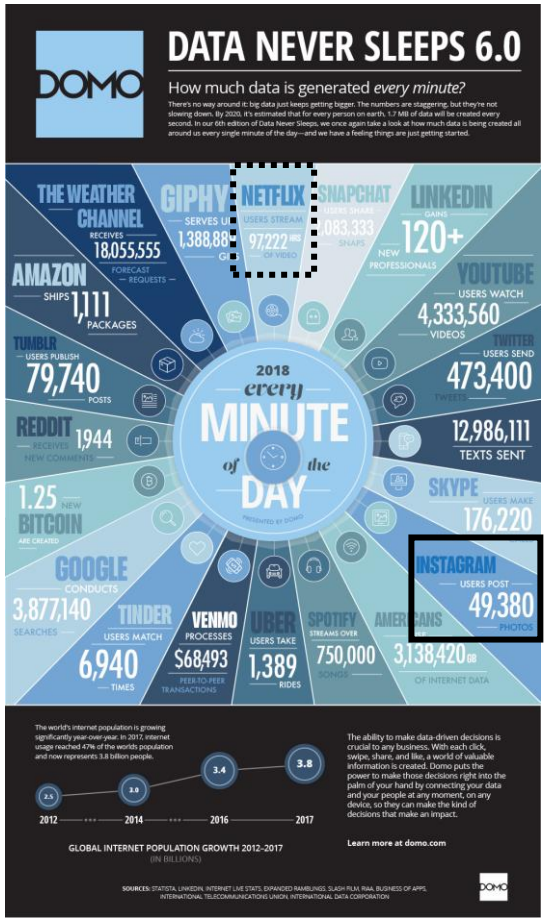


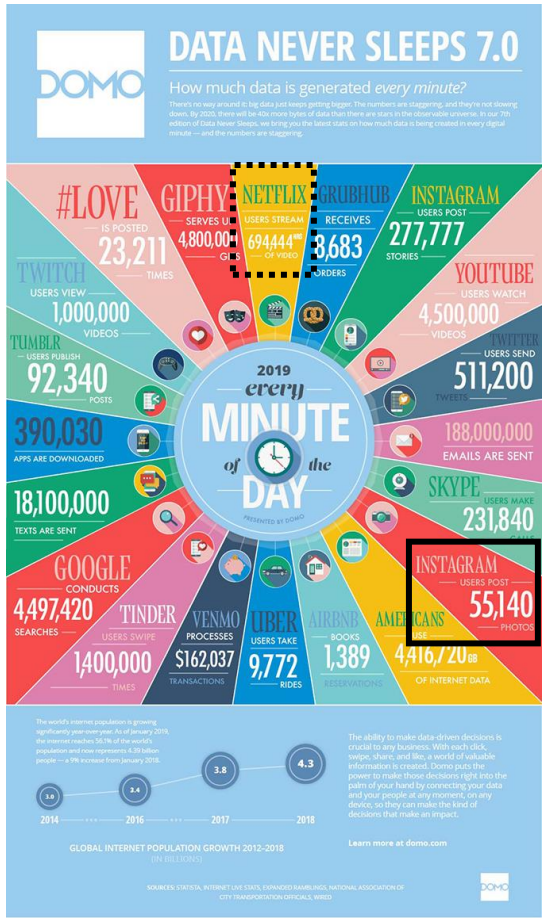
Reference scenario

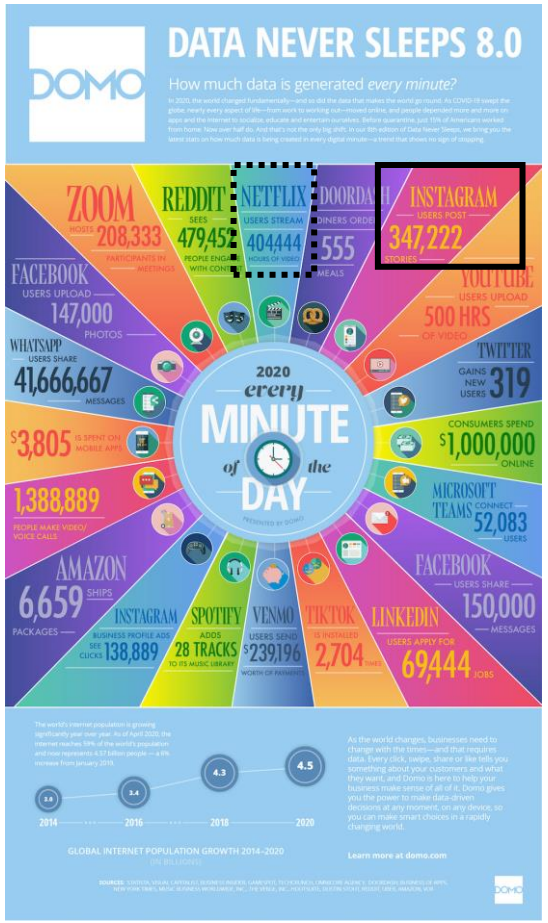
Acceleration

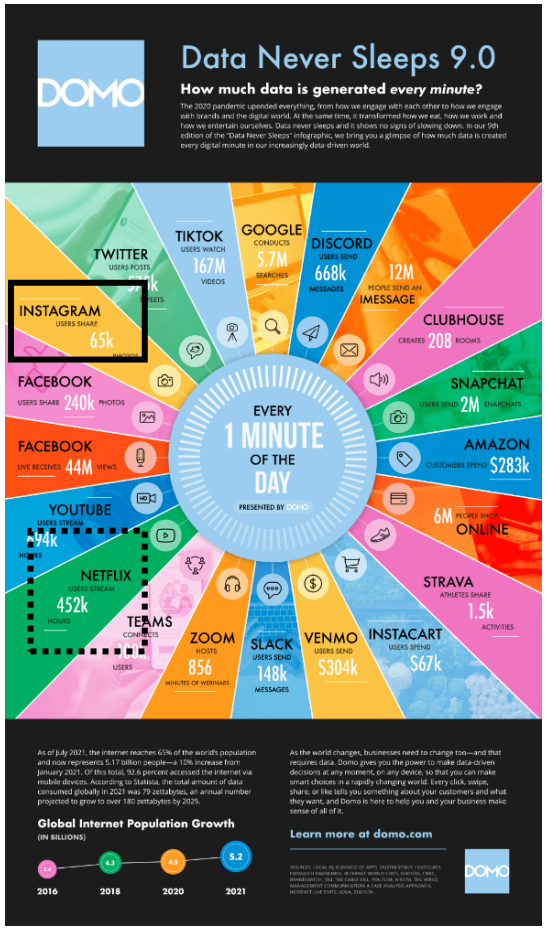
- Velocity is not constant, data comes in bursts
- Take Twitter as an example
  - Hashtags can become hugely popular and appear hundreds of times in just seconds
  - … or slow down to one tag an hour
- Your system must be able to efficiently handle the peak as well as the lows

Reference scenario





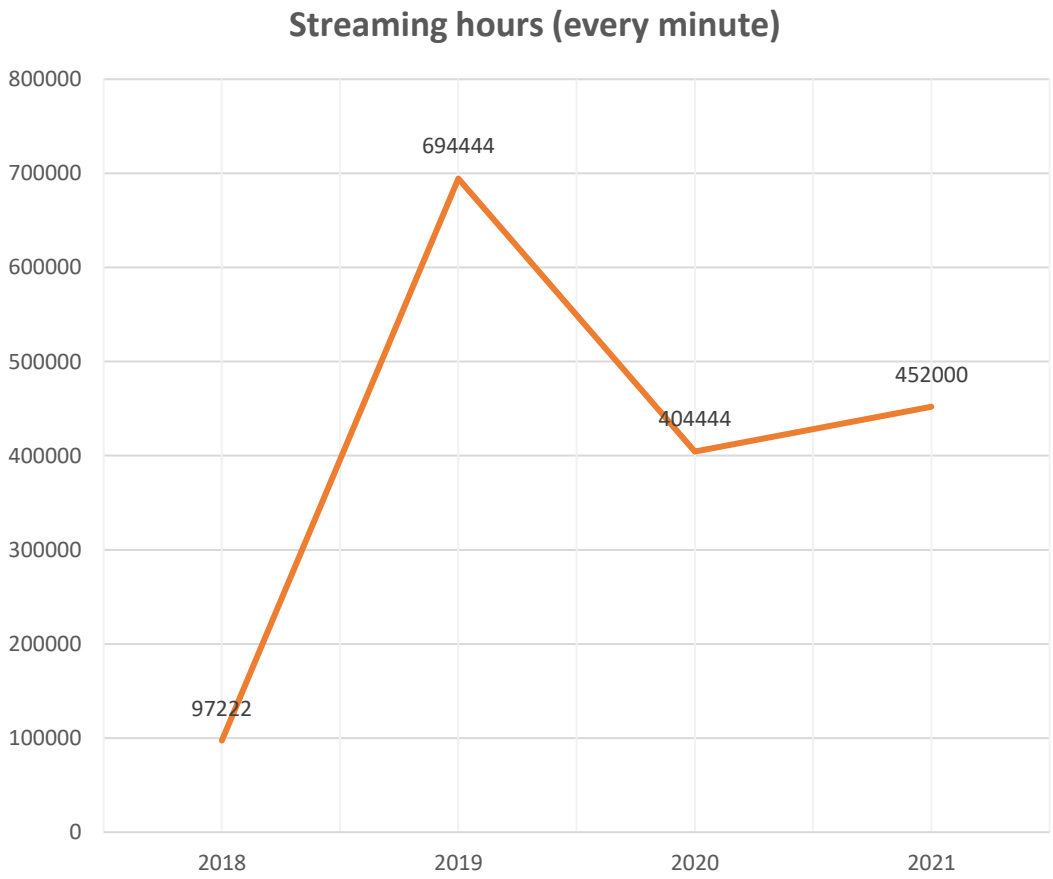




https://www.domo.com/learn/data-never-sleeps-9

Reference scenario

The Netflix scenario



https://www.domo.com/learn/data-never-sleeps-9

Reference scenario

Collecting data

- **Scheduled Batch**
  - Large volume of data processed on a regular scheduled basis
  - Velocity is very predictable
- **Periodic**:
  - Data processed at irregular times (e.g., after collecting a certain ---large--- amount of data)
  - Velocity is less predictable
- **Near real-time**
  - Streaming data processed in small individual batches collected and processed within minutes
  - Velocity is a huge concern
- **Real-time**
  - Streaming data collected and processed in very small individual batches within milliseconds
  - Velocity is the paramount concern

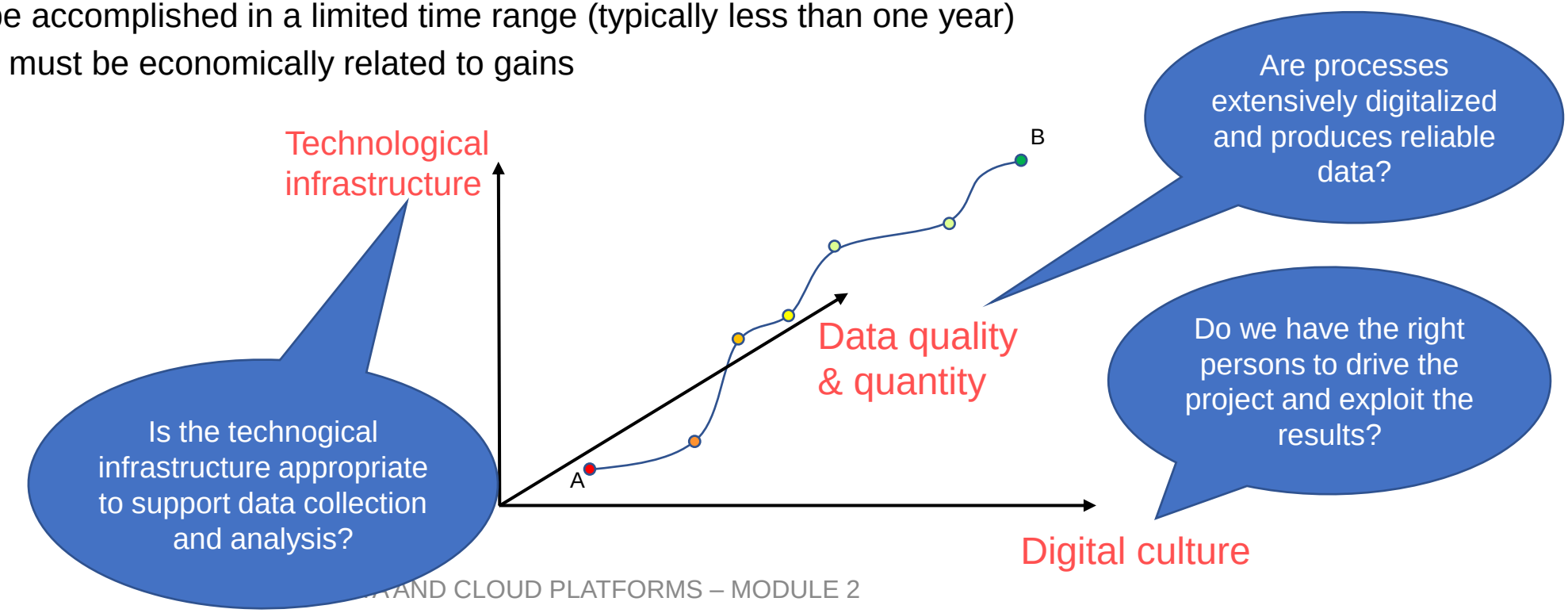
Processing data

- **Batch and periodic**
  - Once data has been collected, processing can be done in a controlled environment
  - There is time to plan for the appropriate resources
- **Near real-time and real-time**
  - Collection of the data leads to an immediate need for processing
  - Depending on the complexity of the processing (cleansing, scrubbing, curation), this can slow down the velocity of the solution significantly
  - Plan accordingly

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



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

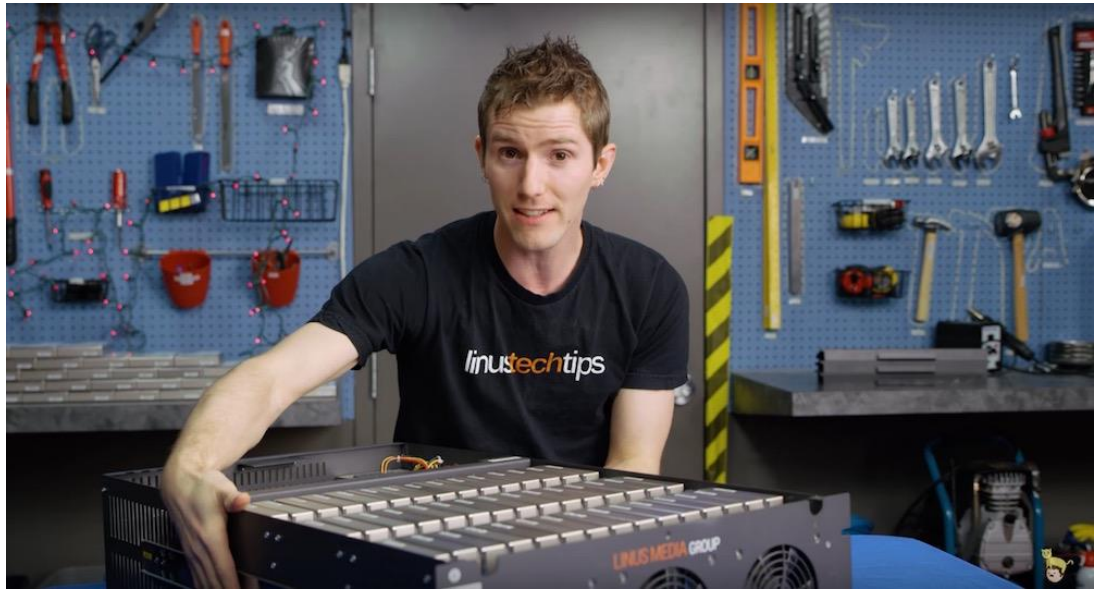
Why going cloud?

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

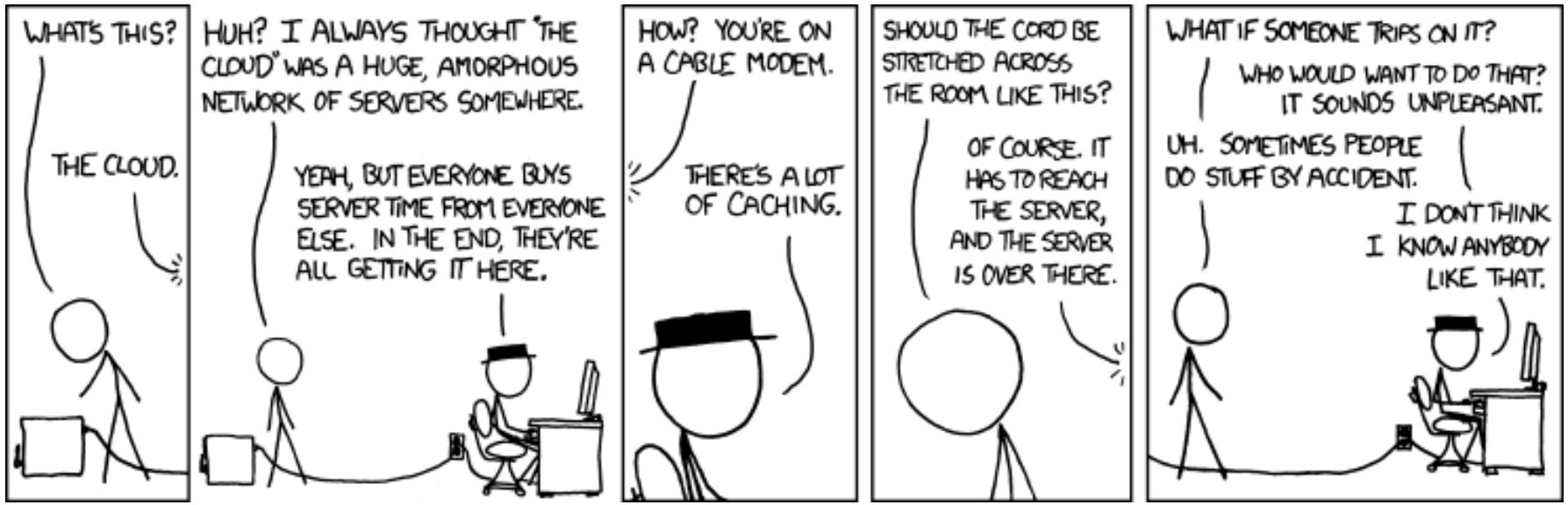
Integration and abstraction are drivers of change

- From databases to data platforms
- From on-premises to serverless architectures
- From custom to standardized data pipelines

Is cloud a silver bullet?

Cloud computing is the outsourcing of a company's hardware and software architecture

- Which are the risks and issues?

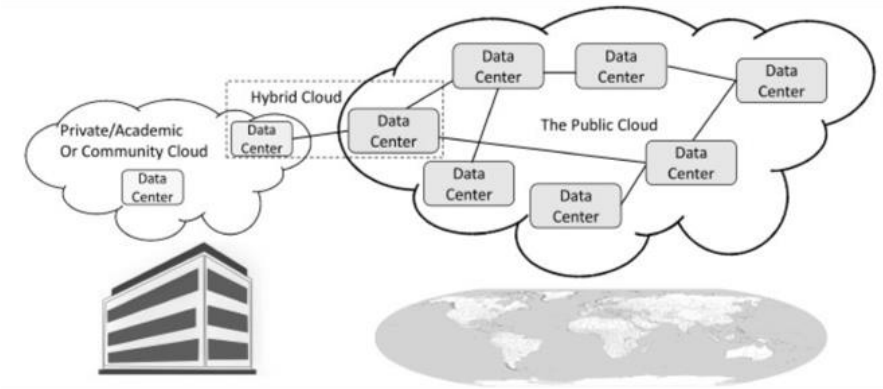


https://xkcd.com/908/

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

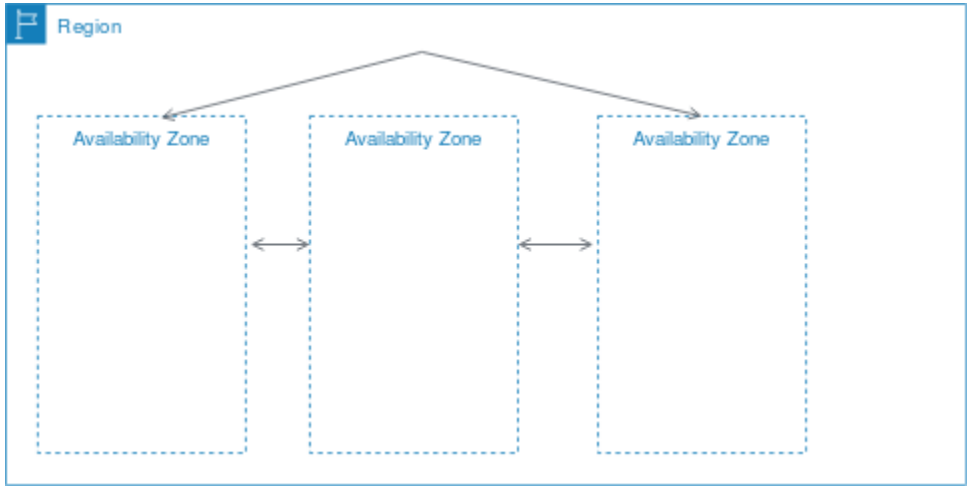
- Locations are composed of **regions** and **availability zones**

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 zones but not regions



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

Cloud computing: principal vendors

Figure 1. Magic Quadrant for Cloud Infrastructure and Platform Services

https://www.gartner.com/en/research/methodologies/magic-quadrants-research

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

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)

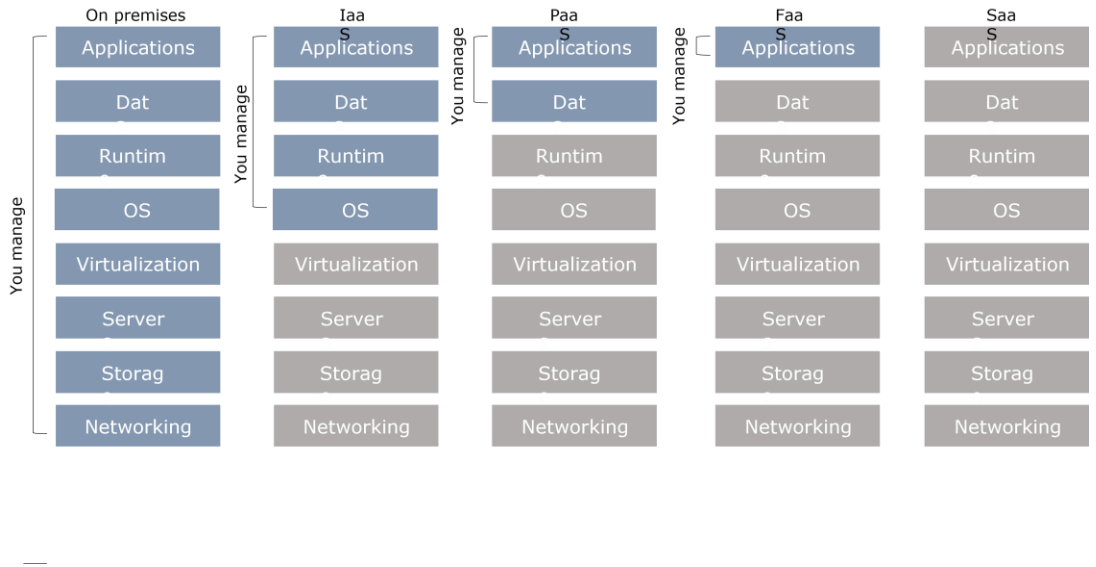
Cloud computing: deployment models

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)

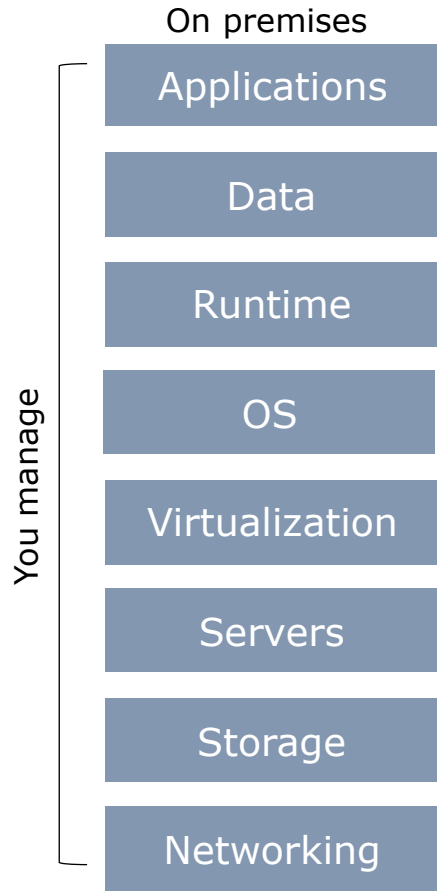
- A collective term that refers to the delivery of anything as a service
- It encompasses the products, tools and technologies that vendors deliver to users



Cloud computing: deployment models

On-premises

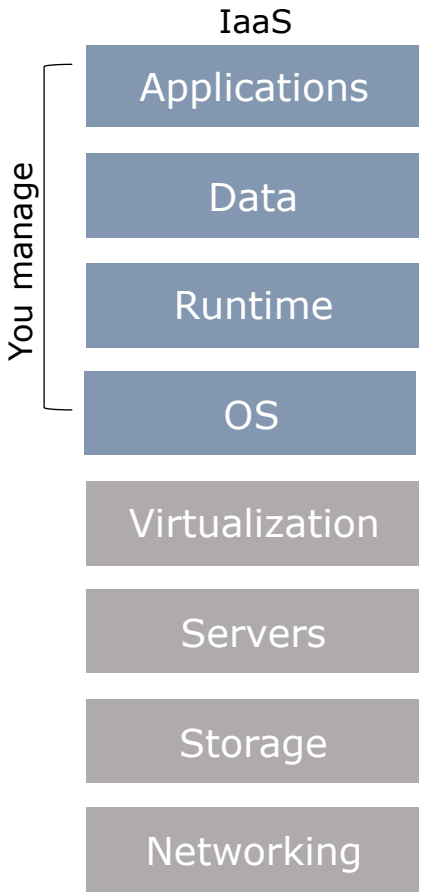
- Provisioning servers is time-consuming
  - A non-trivial environment is hard to set up
- Require dedicated operations people
- Often a distraction from strategic tasks



Cloud computing: deployment models

Infrastructure as a service (IaaS)

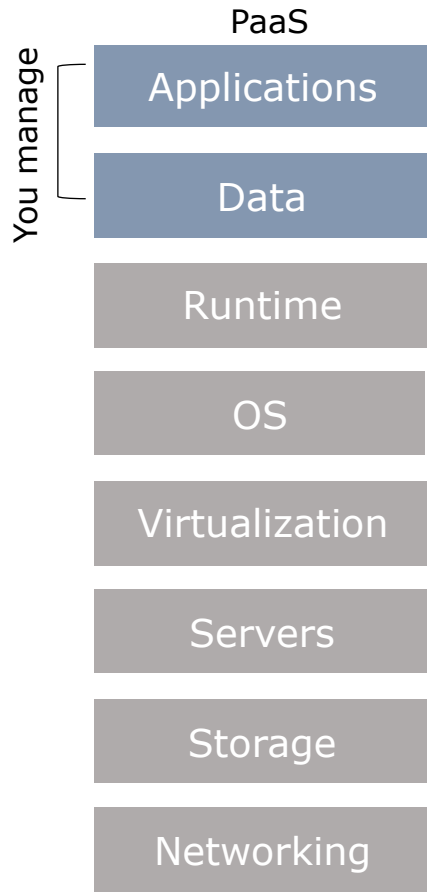
- A computing infrastructure provisioned and managed over the internet (e.g., AWS EC2)
- Avoid expense/complexity of buying/managing physical servers/data-centers
- IaaS overcomes issues on-premises
- Possibly requires to manage many environments



Cloud computing: deployment models

Platform as a Service (PaaS)

- A development and deployment environment in the cloud (e.g., AWS Elastic Beanstalk)
- Support complete application life-cycle: building, testing, deploying, etc.
- Avoid expense/complexity of managing licenses and application infrastructure



Cloud computing: deployment models

PaaS and **containers** are potential solutions to inconsistent infrastructures

PaaS provides a platform for users to run their software

- Developers write software targeting features/capabilities of the platform

Containerization isolates an application with its own environment

- Lightweight alternative to full virtualization
- Containers are isolated but need to be deployed to (public/private) server
- Excellent solution when dependencies are in play
- Housekeeping challenges and complexities

Cloud computing: deployment models

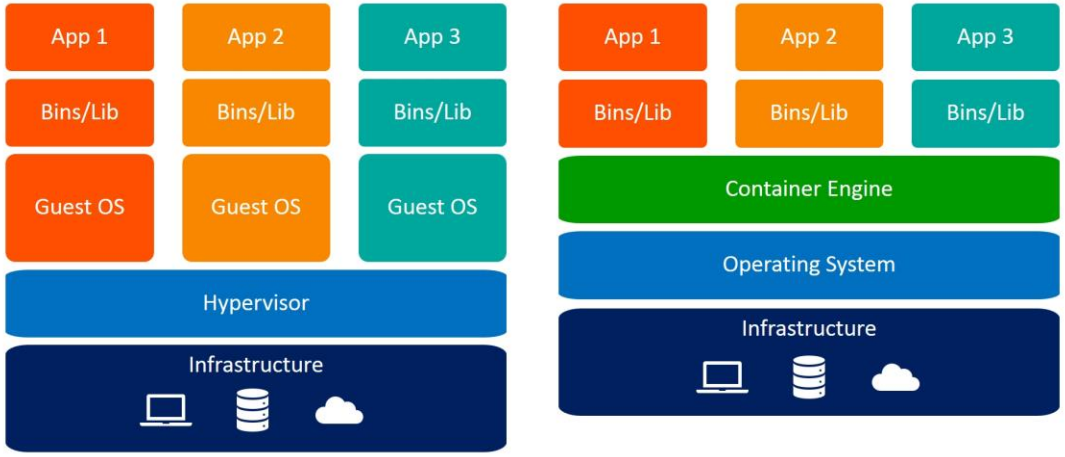
Containers and virtual machines are packaged computing environments

Containers

- On top of physical server and its host OS
- Share the host OS kernel
- Shared components are read-only
- “Light”, take seconds to start

Virtual machines

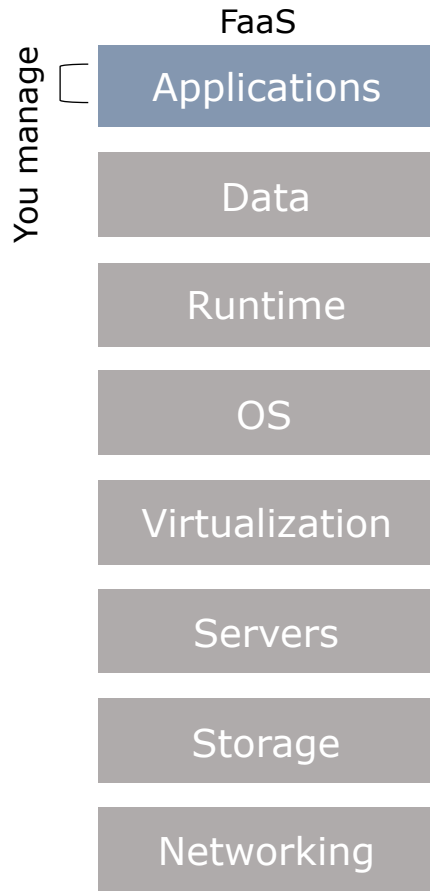
- Emulate a hardware/software system
- On top of a hypervisor (VM monitor)



Cloud computing: deployment models

Function as a Service (FaaS)

- A coding environment, cloud provider provisions platform to run the code (e.g., AWS Lambda)
- Infrastructure provisioning and management are invisible to the developer



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)

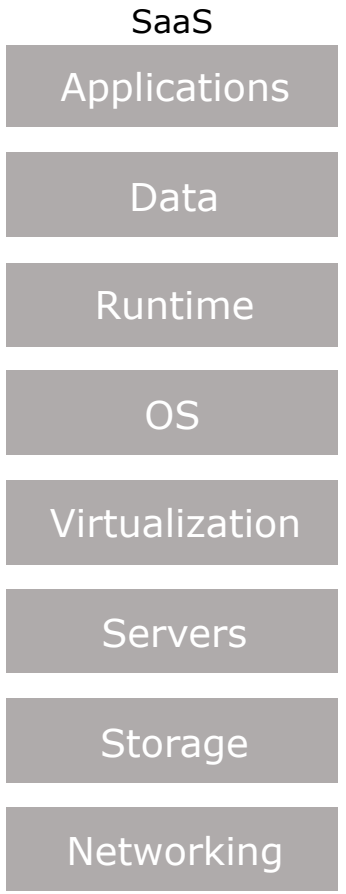
FaaS is not a silver bullet

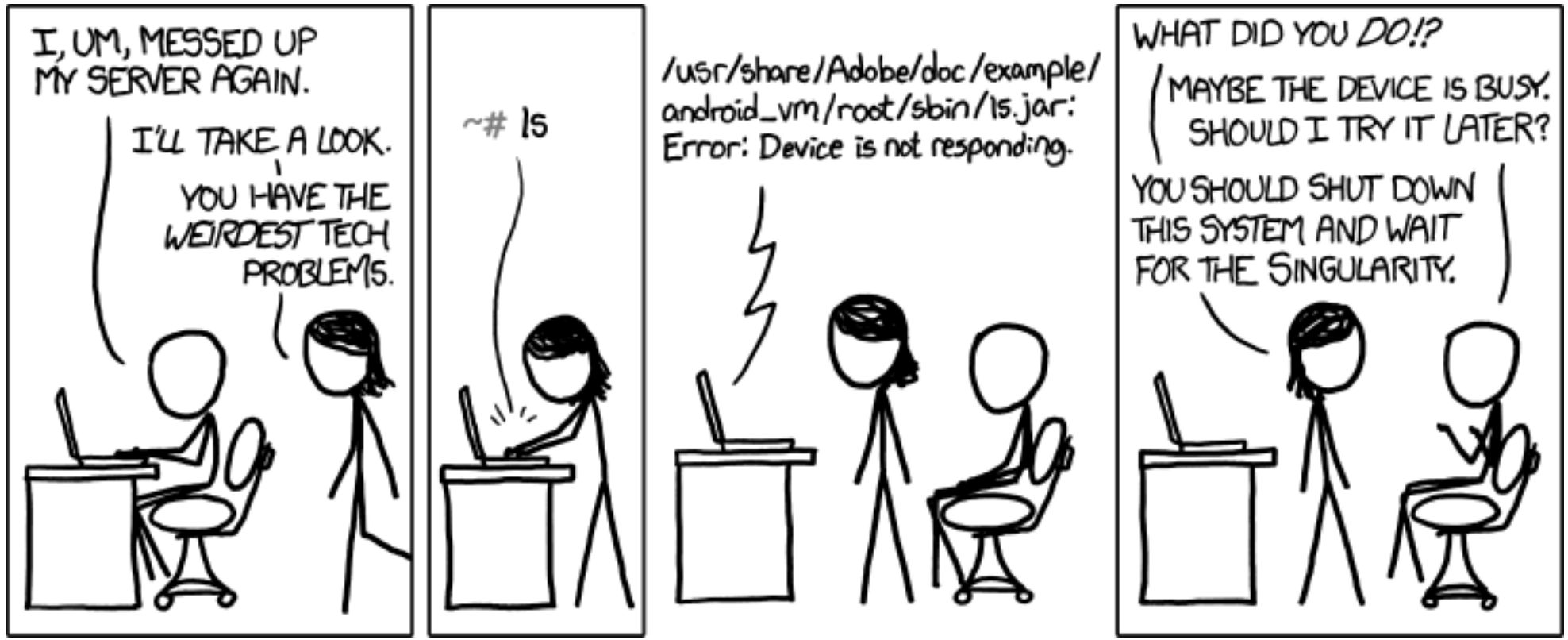
- Not appropriate for latency-sensitive applications
- Strict specific service-level agreements
- Migration costs
- Vendor lock-in can be an issue

Cloud computing: deployment models

Software as a service (SaaS)

- An application environment
- Access cloud-based apps over the Internet (e.g., email, Microsoft Office 365, Github)





https://xkcd.com/1084/

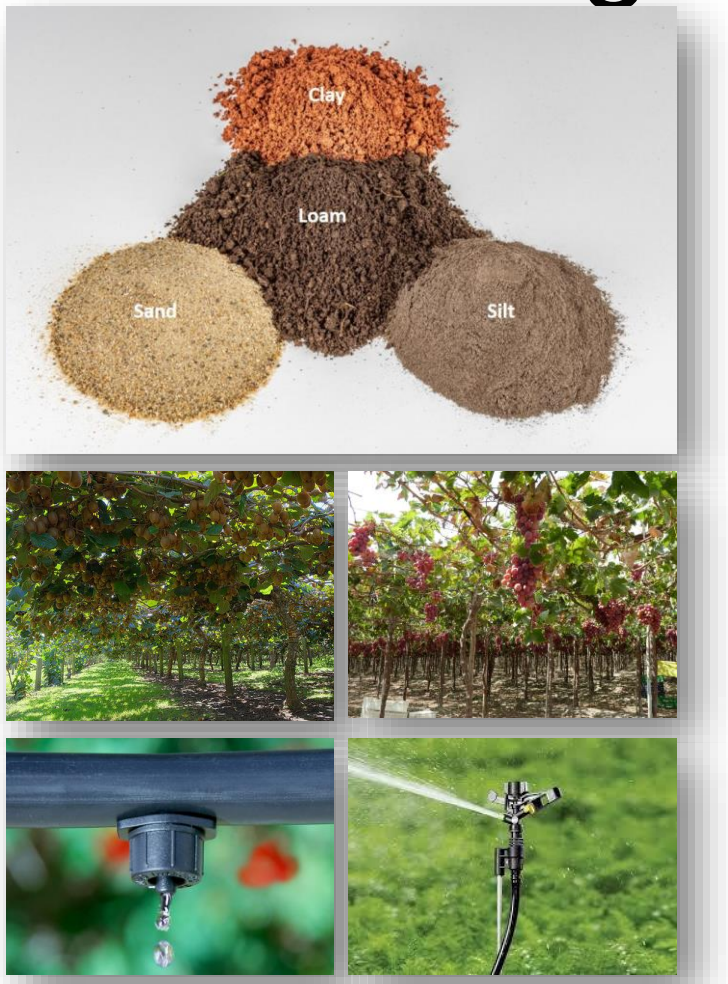
BIG DATA AND CLOUD PLATFORMS

From data lake to data warehouse

Context: Soil moisture monitoring

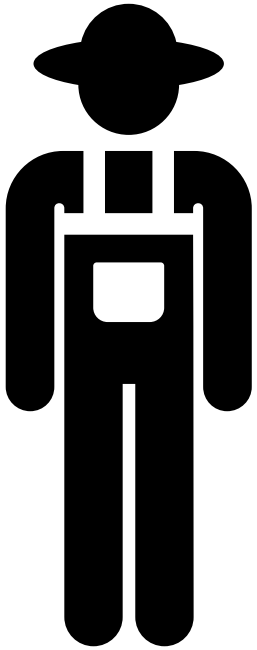
Optimizing soil moisture is crucial for watering and crop performance [1]

- **GOAL**: build an expert system to save water while improving fruit quality (i.e., provide a recommendation of the optimal amount of water)
- Soils have different water retention
- Watering systems have different behaviors (e.g., drippers and sprinklers)
- Plants have different water demand (e.g., Kiwi [2] vs Grapes)
- Sensors produce different measurements with different precisions



[1] Turkeltaub et al., Real-time monitoring of nitrate transport in the deep vadose zone under a crop field–implications for groundwater protection, Hydrology and Earth System Sciences 20 (8) (2016) 3099–3108.
[2] M. Judd, et al., Water use by sheltered kiwifruit under advective conditions, New Zealand journal of agricultural research 29 (1) (1986) 83–92.

Context: Soil moisture monitoring



(Example) Scenarios of digital transformation in agriculture

Scenario #1

- The farmer/technician controls the watering system based only on the experience
- No digital data/KPIs/automation

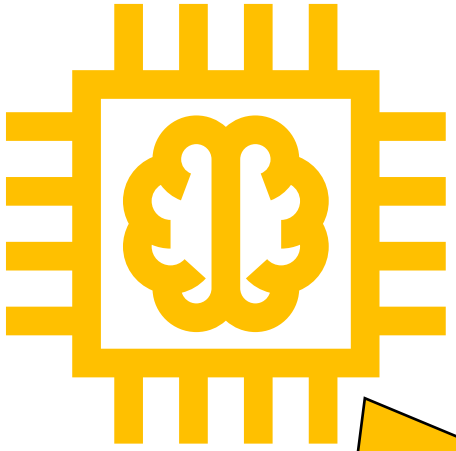
Scenario #2

- The control of the watering system is refined by observing sensor data
- Sensor data is digitalized, no KPIs/automatic

Scenario #3

- Sensor data feeds a decision support system that, knowing how to optimize KPIs, controls the watering system

Context: Soil moisture monitoring



(Example) Scenarios of digital transformation in agriculture

Scenario #1

- The farmer/technician controls the watering system based only on the experience
- No digital data/KPIs/automation

Scenario #2

- The control of the watering system is refined by observing sensor data
- Sensor data is digitalized, no KPIs/automatic

Artificial intelligence (AI) is intelligence demonstrated by machines. AI research has been defined as the field of study of intelligent agents, which refers to any system that perceives its environment and takes actions that maximize its chance of achieving its goals.

a decision support system that, knowing how to
rols the watering system

Context: Soil moisture monitoring

We need to understand how the soil behaves

Simulate [1, 2] the soil behavior according to physical models [3]

- However, a fine tuning is required
- We need to know/parametrize everything
  - Soil (e.g., retention curve, hysteresis [3])
  - Plant (e.g., roots, LAI)
  - Weather conditions (temperature, humidity, wind, precipitations)
  - Watering system (e.g., capacity, distance between drippers)

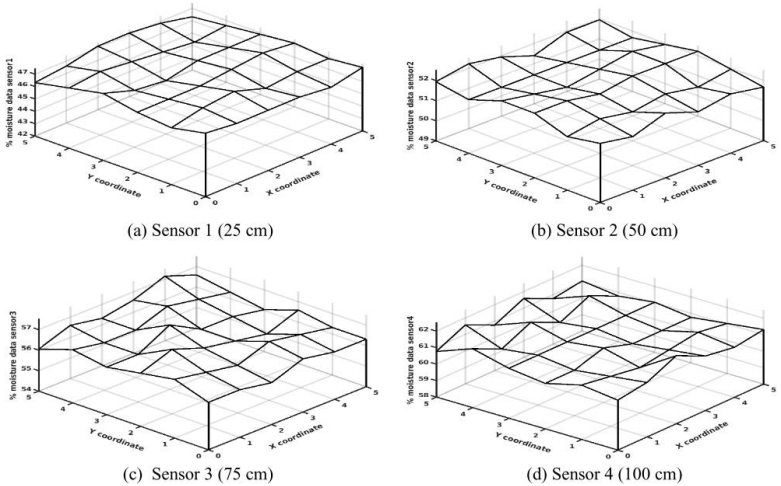
Tuning can take months (of human interactions)!

- Need to collect samples from the field… if parameters are incorrect, trace back
- Need to implement/code all these features into the simulator [1, 2]
- Hyper-parameter tuning with machine learning can help, but it is not a silver bullet

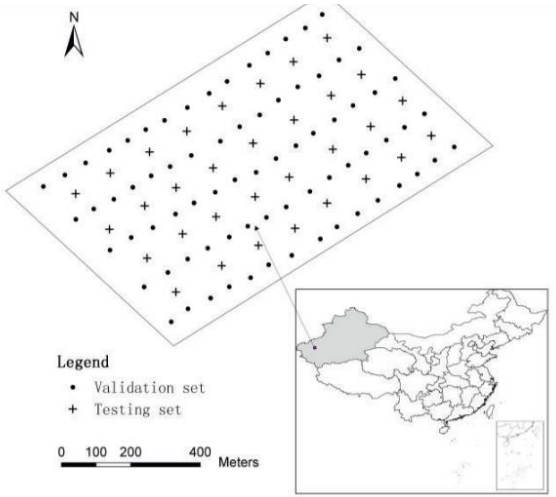
[1] Šimunek, J., et al. "HYDRUS: Model use, calibration, and validation." Transactions of the ASABE 55.4 (2012): 1263-1274.
[2] Bittelli, Marco, et al. Soil physics with Python: transport in the soil-plant-atmosphere system. OUP Oxford, 2015.
[3] Van Genuchten, M. Th. "A closed-form equation for predicting the hydraulic conductivity of unsaturated soils." Soil science society of America journal 44.5 (1980): 892-898.
[4] Pham, Hung Q., Delwyn G. Fredlund, and S. Lee Barbour. "A study of hysteresis models for soil-water characteristic curves." Canadian Geotechnical Journal 42.6 (2005): 1548-1568.

Context: Soil moisture monitoring

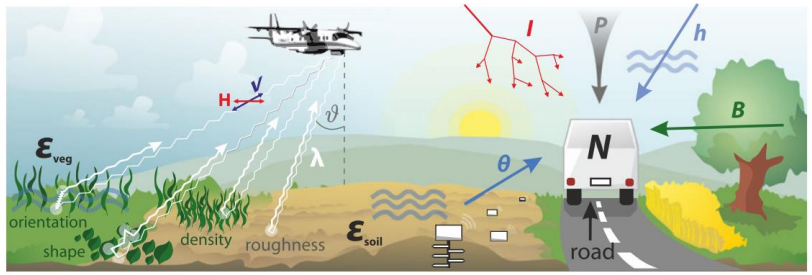
But… we have sensors!



[1]



[2]

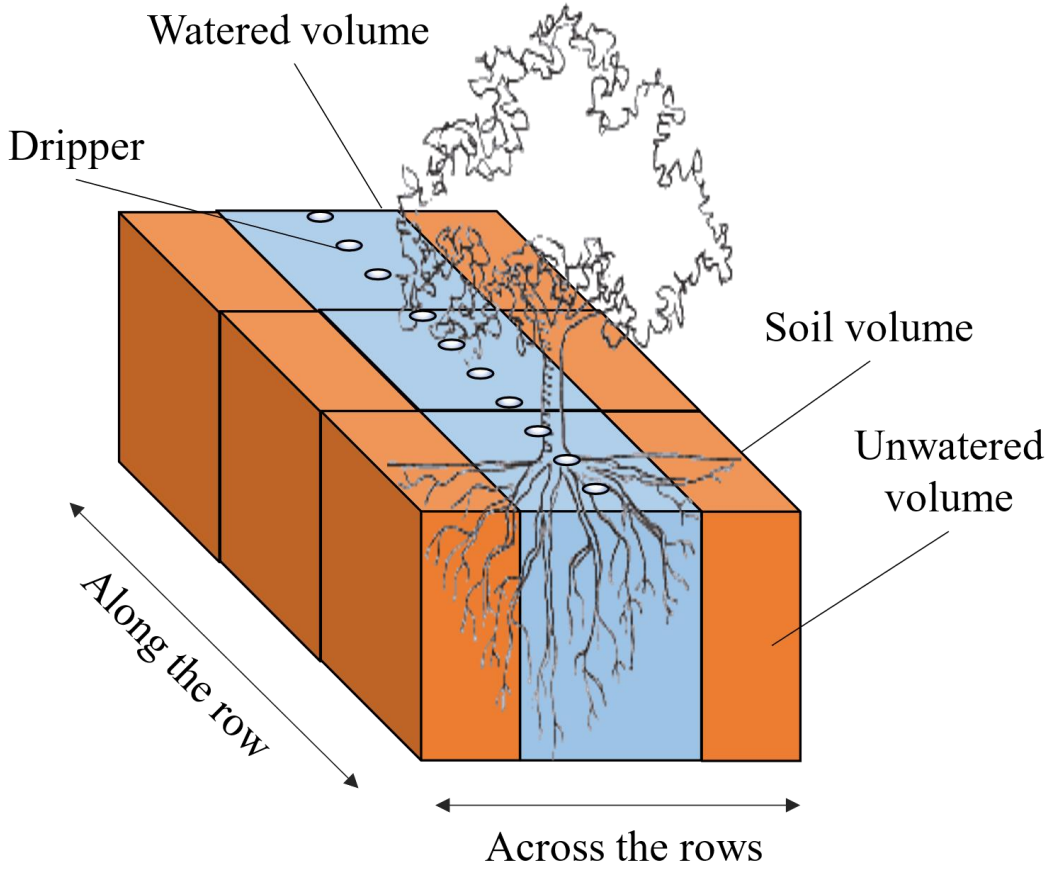


[3]

- These settings are too coarse to monitor soil moisture with precision
- They require many sensors

[1] Koyuncu, Hakan, et al. "Construction of 3D soil moisture maps in agricultural fields by using wireless sensor communication." Gazi University Journal of Science 34.1 (2021): 84-98.
[2] Zheng, Zhong, et al. "Spatial estimation of soil moisture and salinity with neural kriging." International Conference on Computer and Computing Technologies in Agriculture. Springer, Boston, MA, 2008.
[3] Fersch, Benjamin, et al. "Synergies for soil moisture retrieval across scales from airborne polarimetric SAR, cosmic ray neutron roving, and an in situ sensor network." Water Resources Research 54.11 (2018): 9364-9383.

Reference scenario



We consider an orchard where

- Kiwi plants are aligned along rows
- Each row has many drippers (e.g., 1 every meter)
- Drippers can water a limited soil volume

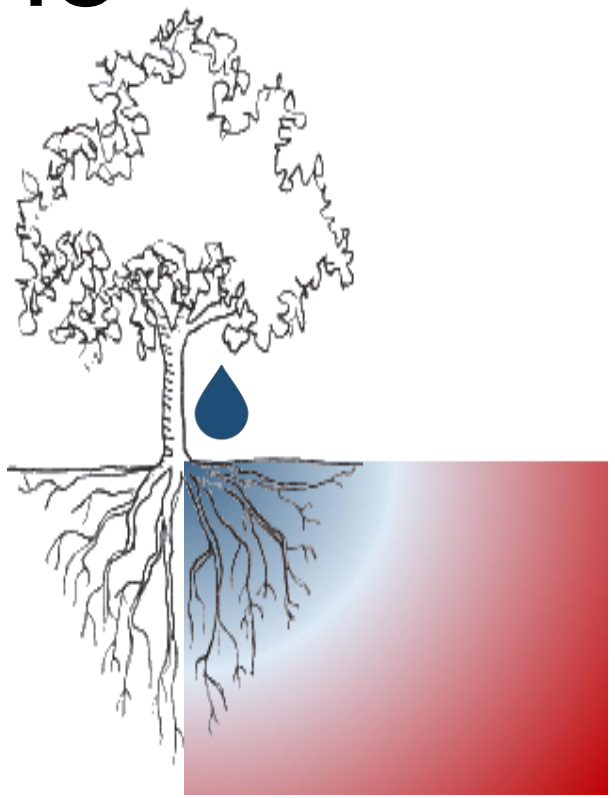
Francia, Matteo, et al. "Multi-sensor profiling for precision soil-moisture monitoring." Computers and Electronics in Agriculture 197 (2022): 106924.

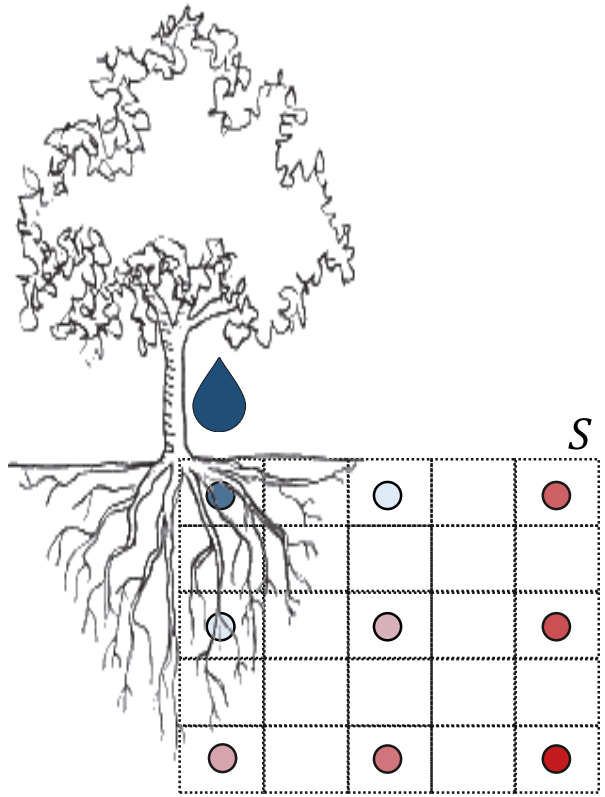
Reference scenario

(a) Soil moisture is a continuum

(b) Sensors return a discretized representation of soil moisture

- The monitoring accuracy changes
- depending on the sensor layout





Francia, Matteo, et al. "Multi-sensor profiling for precision soil-moisture monitoring." Computers and Electronics in Agriculture 197 (2022): 106924.

Reference scenario

We consider a 2D grid of 3 x 4 gypsum block sensors

- Sample soil moisture-sensor data every 15 minutes
- Collect dripper and weather data (humidity, temperature, solar radiation, wind) every hour

How many data does each monitored field produces every season?

- $\left(12 \cdot 4 \frac{samples}{hour} + 5 \frac{samples}{hour}\right) \cdot 24 \frac{hour}{day} \cdot 30 \frac{day}{month} \cdot 5 \frac{month}{year} \cong 200 \cdot 10^3 \frac{samples}{year}$

We monitored 6 fields for 2 years

- $200 \cdot 10^3 \frac{samples}{year} \cdot 2\ years \cdot 6 = 2.4 \cdot 10^6\ samples$

We should consider accessory data for storage and optimization structures

- In two years, we collected/generated 16GB data (as of 2022-08-30)

Francia, Matteo, et al. "Multi-sensor profiling for precision soil-moisture monitoring." Computers and Electronics in Agriculture 197 (2022): 106924.