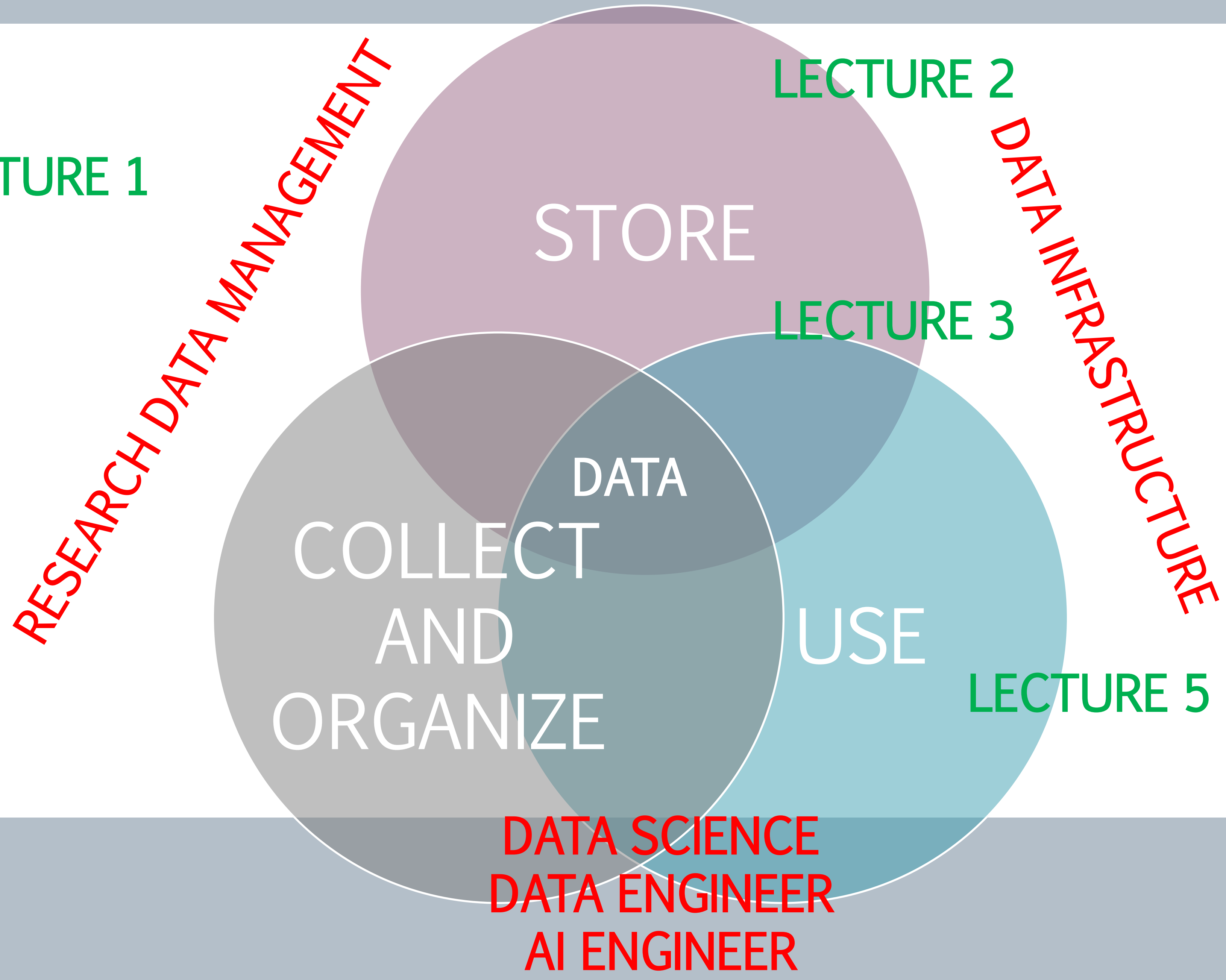
Piano Nazionale di Ripresa e Resilienza

Data Infrastructure

Lecture 5 : Data Stack Architecture

Federica Bazzocchi
14/4/2025


LECTURE 1
RESEARCH DATA MANAGEMENT
LECTURE 2
STORE
DATA INFRASTRUCTURE
LECTURE 3
DATA
COLLECT AND ORGANIZE
USE
LECTURE 5
DATA SCIENCE
DATA ENGINEER
AI ENGINEER

LECTURE 5 OUTLINE:

- Data Architecture vs Data Infrastructure
- Two words on HPC vs Data Infrastructure
- Data Infrastructure software layer

DATA ARCHITECTURE VS DATA INFRASTRUCTURE

WHAT IS DATA ARCHITECTURE?

- ❑ the set of rules that defines within an infrastructure how data is gathered, kept, managed, and utilized.
- ❑ Includes the toolset, policies, and standards that help in managing the handling of data assets properly.
- ❑ the framework that regulates how an organization's IT infrastructure enables its data strategy
- ❑ provides a framework for and is a relevant part of data management.

What is Data Architecture? | GeeksforGeeks

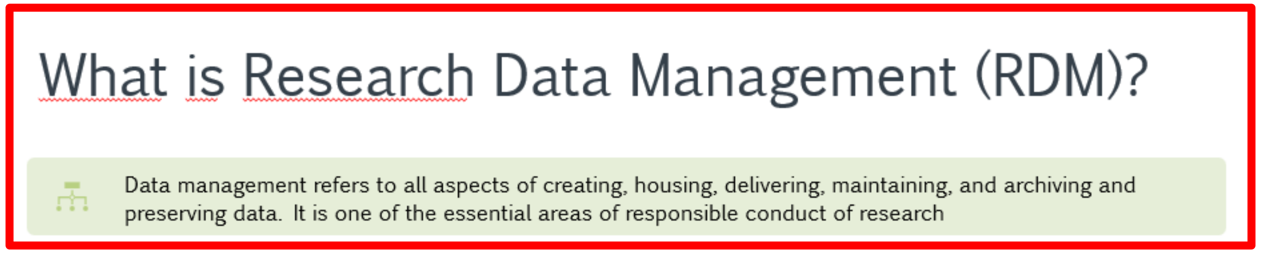
COMPONENTS OF DATA ARCHITECTURE

- **Data Models** : data models are abstract representations of data objects and their relationships. They provide a structured framework for organizing data
- **Data Governance**: involves the policies, procedures, and standards that ensure data quality, security, and compliance
- **Data Integration**: Data integration involves combining data from different sources to provide a unified view.
- **Data Storage**: storage solutions are critical for maintaining and managing data.
- **Data Access and Analytics** : Data access and analytics tools enable users to retrieve, analyze, and visualize data.

TYPES OF DATA ARCHITECTURE

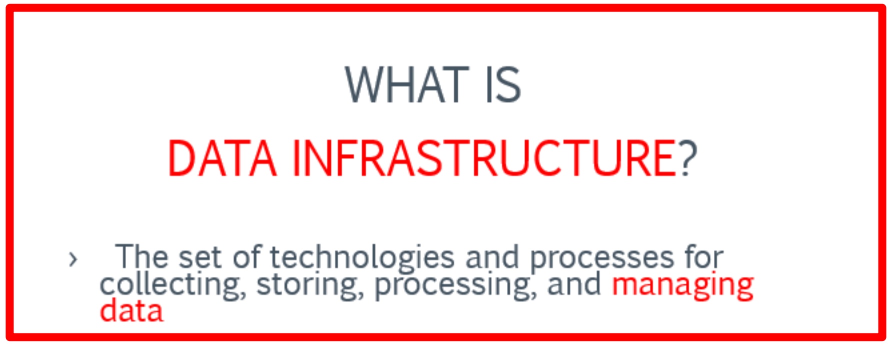
- ❑ **Centralized Data Architecture**: In this framework, all data (being stored and managed) are done in a central repository, which might be a data warehouse. It combines data from many sources into one place, attempting to facilitate data management, analysis, and integrity maintenance. A common association of centralized data architecture is with conventional monolithic data infrastructure,.
- ❑ **Decentralized Data Architecture**: In this framework data processing and storage are distributed among many nodes or systems, enabling each domain to handle its own. Data is spread all over different servers or databases.

Lecture 1


What is Research Data Management (RDM)?

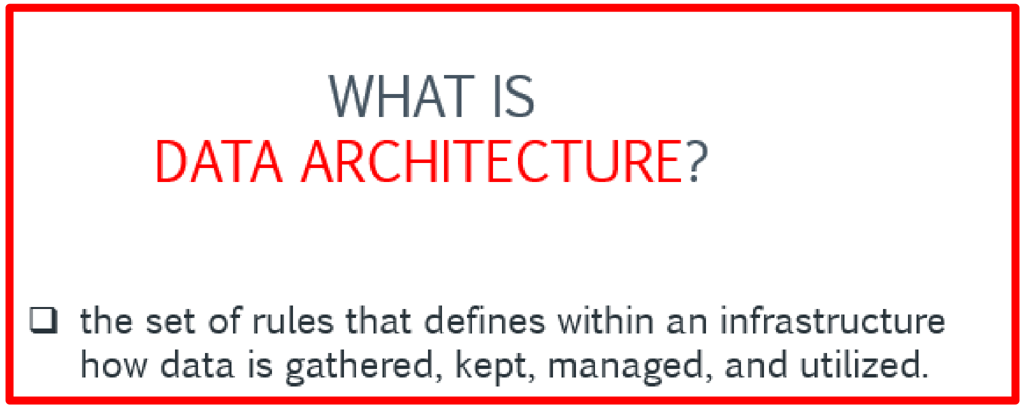
Data management refers to all aspects of creating, housing, delivering, maintaining, and archiving and preserving data. It is one of the essential areas of responsible conduct of research

Lecture 2


WHAT IS DATA INFRASTRUCTURE?

› The set of technologies and processes for collecting, storing, processing, and managing data

Lecture 5 (Today)


WHAT IS DATA ARCHITECTURE?

❑ the set of rules that defines within an infrastructure how data is gathered, kept, managed, and utilized.

ALWAYS THE SAME CONCEPTS?

A BUILDING METAPHOR (FOR A DATA ECOSYSTEM)

Data Infrastructure:

The 'shell' of the system:

- Includes servers, storage, networks, orchestrato
- Provides the physical and technical backbone

shell structure, electrical and plumbing installation

Data Architecture:

The 'essence' inside:

- Includes data models, flows, governance, access logic
- Gives meaning, shape, and function to the infrastructure

Room division and completed house

Data Management:

The ‘rule' inside:

- Includes all the policy to according to which data are organized
- includes all the facility operations

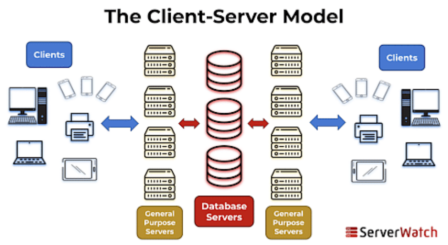
Furniture and space planning, ‘house’ policy

ARCHITECTURAL MODELS

From Lecture 2

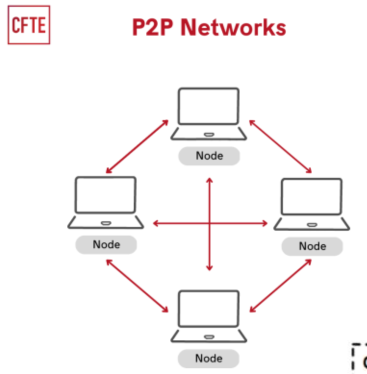
Comparison Overview

- Client-Server:
  - Centralized
  - Easier to manage
  - Single point of failure

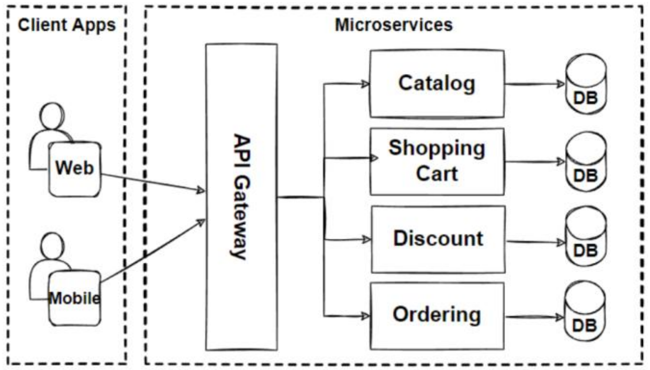


Are connected to the type of data architecture

- Peer-to-Peer:
  - Fully decentralized
  - Fault-tolerant
  - Complex coordination



- Microservices:
  - Modular, loosely coupled
  - Scalable and flexible
  - Requires DevOps and orchestration



From Lecture 2

COMBINING ARCHITECTURES/MODELS

Architectural Model	Only Distributed ?	Typical of Distributed System
Client-Server	NO	YES
P2P	YES	YES
Microservices	NO, BUT ALMOST ALWAYS	YES

Not all the architecture sare compatible with all data architectures

→

A building may become a private house or a B&B, another building may become a bank or a hotel (see Hilton Hotel in TS) but not all the buildings may become a bank!

TWO WORDS ON HPC VS DATA INFRASTRUCTURE

HPC IS NOT EASY TO BE DEFINED

- ❑ High performance computing (HPC), also known as supercomputing, refers to computing systems with extremely high computational power that are able to solve hugely complex and demanding problems.

  https://ec.europa.eu/digital-single-market/en/high-performance-computing

- ❑ "CRUCIAL PROBLEMS that we can only hope to address computationally REQUIRE US TO DELIVER **EFFECTIVE COMPUTING POWER ORDERS-OF-MAGNITUDE GREATER THAN WE CAN DEPLOY TODAY**."

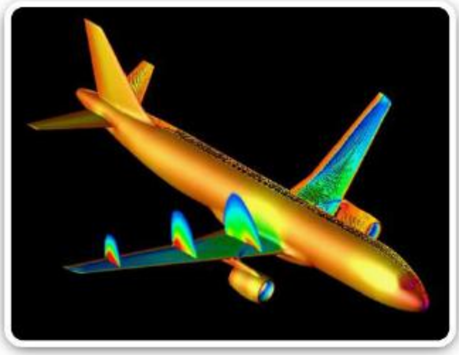
*DOE's Office of Science, **2012***

What do they have in common ?

Finding the supermassive black hole in the heart of Milky Way

Looking for a cure for cancer and extend life expectations for patients

Improving the aerodynamic of an aircraft to make safer and more efficient



Pricing shares and winning trading by millisecond arbitrage

Forecasting an hurricane and its impact with increased precision

Finding oil under a salt crust saving billions in exploration and drilling

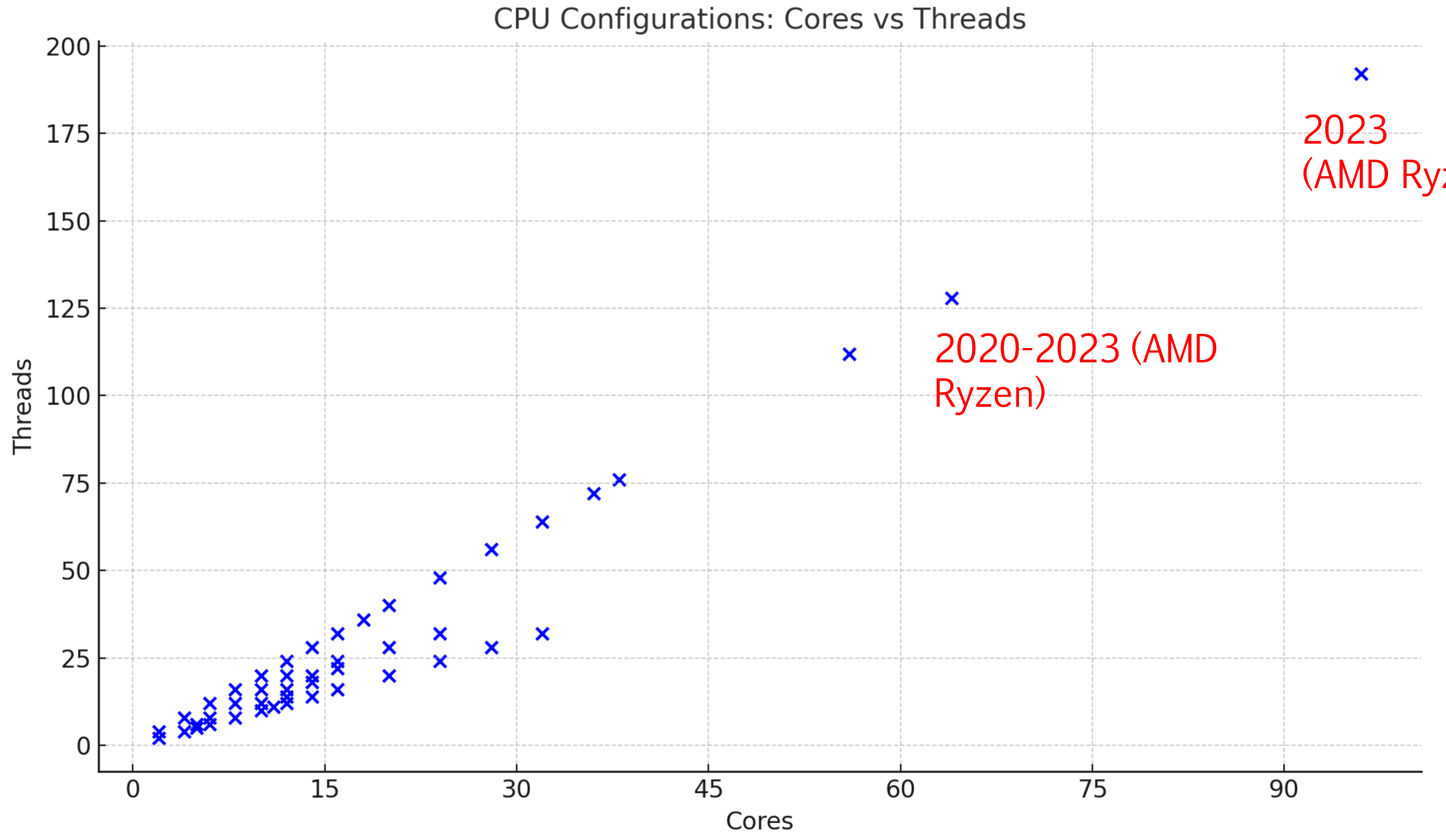
THEY ALL NEED COMPUTATIONAL POWER AND USE HIGH PERFORMANCE COMPUTING TO DELIVER BETTER RESULTS FASTER

Complex problems solved by simulations

- Simulation has become the way to research and develop new scientific and engineering solutions.
- Used nowadays in leading science domains like aerospace industry, astrophysics, etc.
- Challenges related to the complexity, scalability and data production of the simulators arise.
- Impact on the relaying IT infrastructure.

Central role of parallel computing

From Lecture 3



2023 (AMD Ryzen)

2020-2023 (AMD Ryzen)

8(8): 2019 (Intel Core i7) 2020-2024 (Apple M1-M4)

EatYourBytes

BUT ….HPC is a Subset of Data Infrastructure

Data Infrastructure includes all components required to store, process, manage, and move data:

- Storage (block, object, file systems)
- Networking (LAN, WAN, cloud, high-speed fabrics)
- Compute (general-purpose CPUs, accelerators, distributed systems)
- Software layers (data orchestration, management, security)

HPC (High-Performance Computing) is a specialized layer within this ecosystem:

- Focused on massively parallel compute workloads
- Requires high-throughput storage and low-latency interconnects
- Used in simulations, modeling, genomics, physics, etc.

Not all data infrastructure is HPC, but all HPC depends on a strong data infrastructure foundation

DATA STACK

WHAT IS A DATA STACK?

- A data stack is a collection of technologies and tools used to collect, store, process, analyze, and visualize data
- It forms the core of modern data infrastructure
- Enables scalable, reliable, and efficient data pipelines
- Supports collaboration across engineering, analytics, and business teams
- Data Stack is the ‘soul’ of Data Architecture

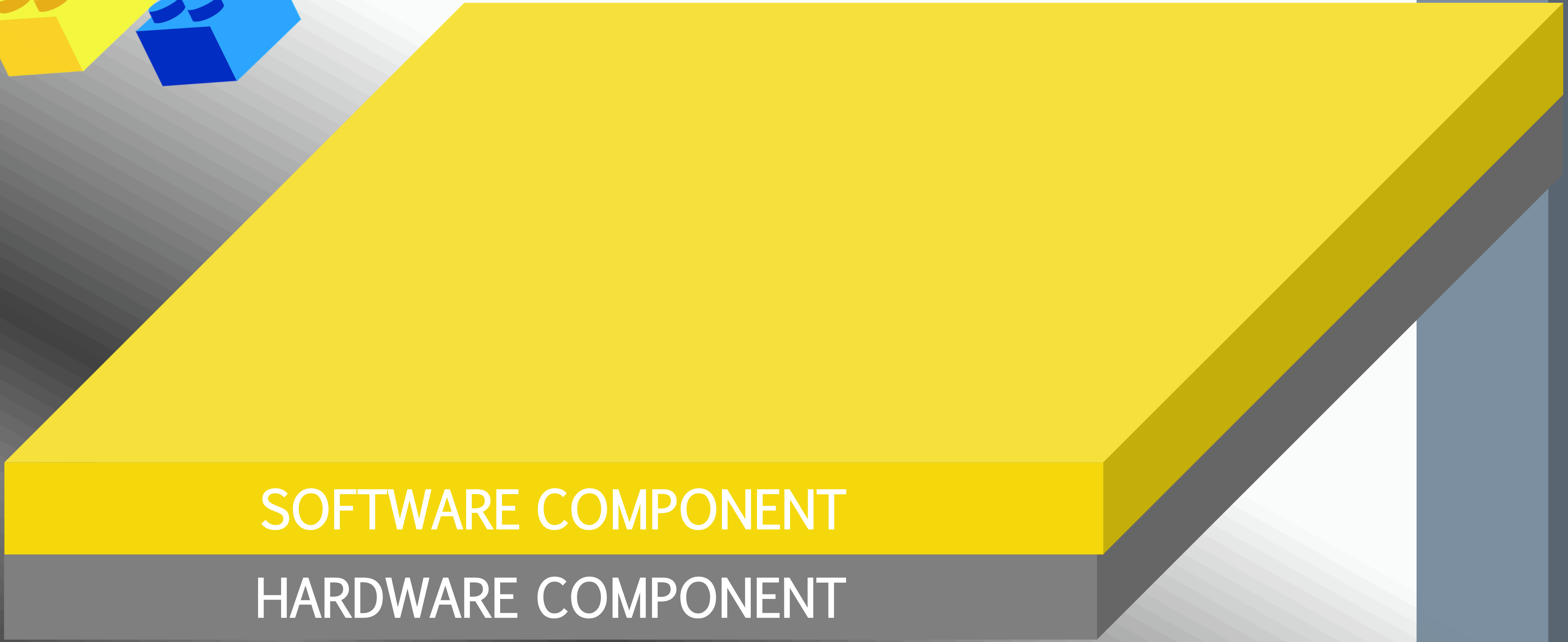
DATA STACK TOOLS

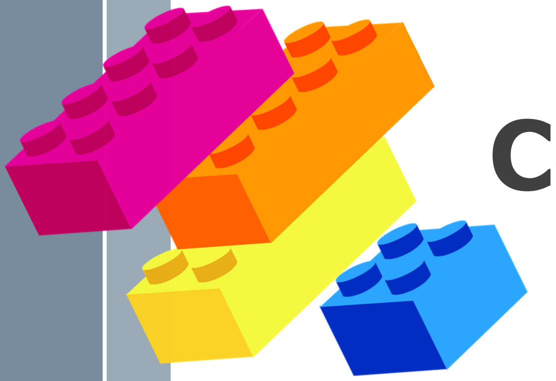
These tools are essential for turning data from 'inedible data' (data that cannot be worked with) to 'edible data' (data that can be worked with).

The faster that data can be accessed, prepared, and analyzed, the faster they can be used (decision-making or research)

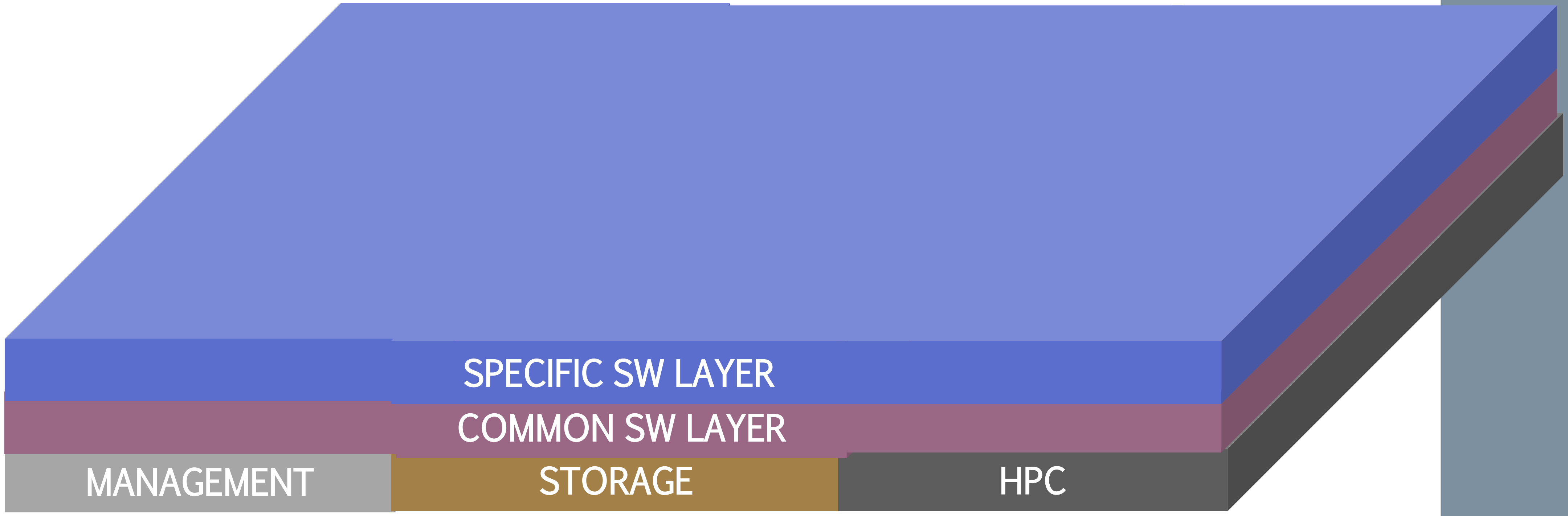
An effective modern data stack architecture is therefore crucial for any data infrastructure   hoping to extract value from their data.

Generic Data Infrastructure





Common + Specific SW Layers



Common Layers in a Modern Data Stack

1. Ingestion: Data collection from various sources
2. Storage: Raw and structured data storage
3. Processing: ETL/ELT and transformation logic
4. Analytics: Query engines and data modeling
5. Visualization: Business Intelligence tools
6. Governance: Data quality, lineage, and security

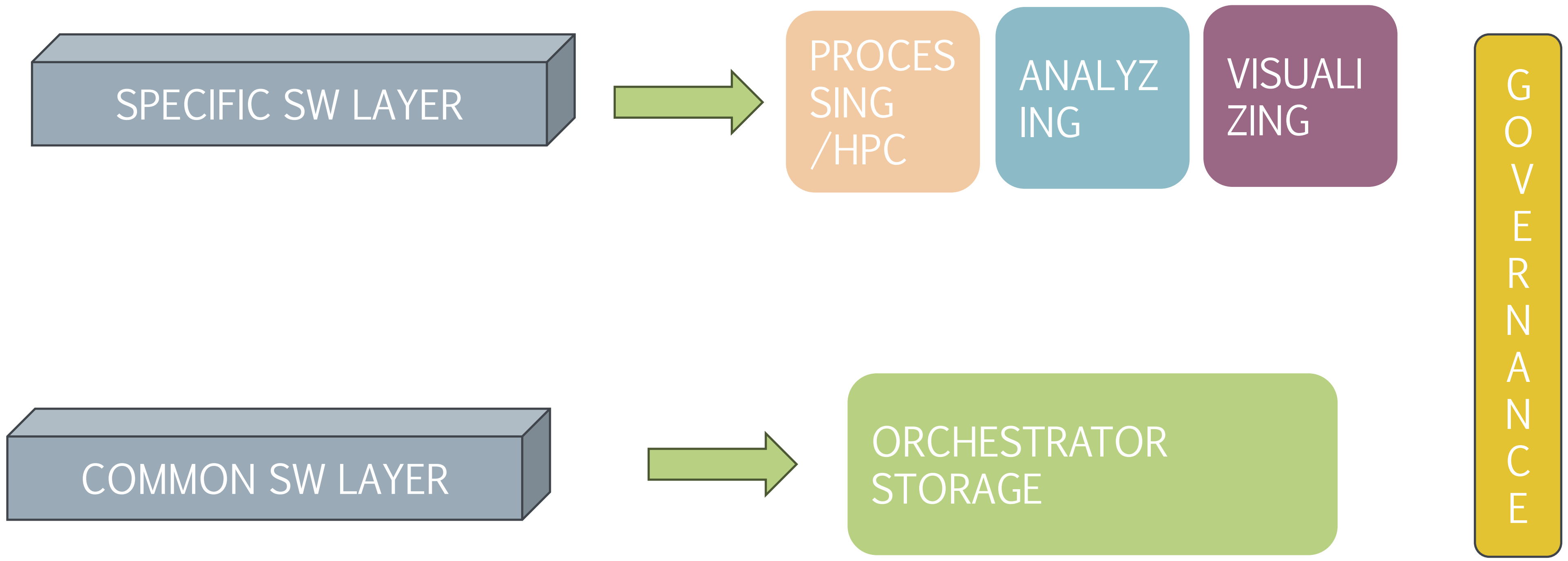
Enterprise vs Scientific Data Stack Layers

Layer	Enterprise	Research
Ingestion & ETL	ETL/ELT pipelines (Kafka, Airbyte, dbt)	FAIR-by-design acquisition workflows
Storage	Data Lakes, Data Warehouse, S3	Open Source Storage SW
Processing/Analysing	Spark, SQL engines, Jupyter workflows	HPC clusters, Jupyter workflows, scientific libraries (NumPy, SciPy)
Governance & Access	Internal Data Governance	Provenance tracking, persistent identifiers (DOI), open standards
Goal	Business intelligence and performance (enhance profit)	Reproducibility, transparency, data reusability--FAIR principles– publication (enhance scientific discoveries)

Traditional vs Modern Data Stack

- Traditional:
  - On-prem databases
  - Batch ETL
  - Static reports

- Modern:
  - Cloud data lakes and warehouses
  - Streaming ingestion and ELT
  - Interactive BI dashboards
  - Integrated monitoring and governance


SPECIFIC SW LAYER
PROCESSING/HPC
ANALYZING
VISUALIZING
COMMON SW LAYER
ORCHESTRATOR STORAGE
GOVERNANCE

COMMON SOFTWARE LAYER

Container: very short history

CONTAINER: DALLE ORIGINI FINO A DOCKER - Criticalcase

- The idea behind containers was born in 1979 con chroot UNIX, a system that was able to give process isolated space inside storage
- Around 2000 FreeBSD Jail allowed to divide the system in smaller pieces, called jail.
- In 2001 it is developed a solution similar to FreeBSD, Linux Vserver.
- In 2008 the LinuX Container (LXC) project was born in 2008: the most comprehensive container management solution of those years.
- In 2013 Docker arrived, the most used Linux container system in the IT field. Docker is an open-source project developed by the company Dotcloud (later renamed Docker) based on LXC,
- Since 2014, Docker no longer uses LXC as the default execution environment, replaced by its own libcontainer library written in the Go programming language.

Container: what they are

- A container is a standard package of software that bundles an application’s code together with the related configuration files and libraries, and with the dependencies required for the app to run. This allows developers and IT pros to deploy applications seamlessly across environments.
- Containers solve the problem of an application failing to run correctly when moved from one environment to another. They ensure consistency across environments.
- Containers provide a lightweight, immutable infrastructure for application packaging and deployment. An application or service, its dependencies, and its configuration are packaged together as a container image. It is used for reproducible data pipelines.
- This way, containers enable developers and IT professionals to deploy applications across environments with little or no modification.

What is a container? | Microsoft Azure

What is a Container Orchestrator?

Container orchestration is the process of automating the deployment, management, scaling, and networking of containers throughout their lifecycle, making it possible to deploy software consistently across many different environments at scale.

Container orchestration makes it possible to build continuous integration and continuous deployment (CI/CD) pipelines, which improve software delivery throughout the software development lifecycle via automation. Container orchestration also connects to a DevOps approach, which aims to accelerate the processes of bringing an idea from development to deployment.

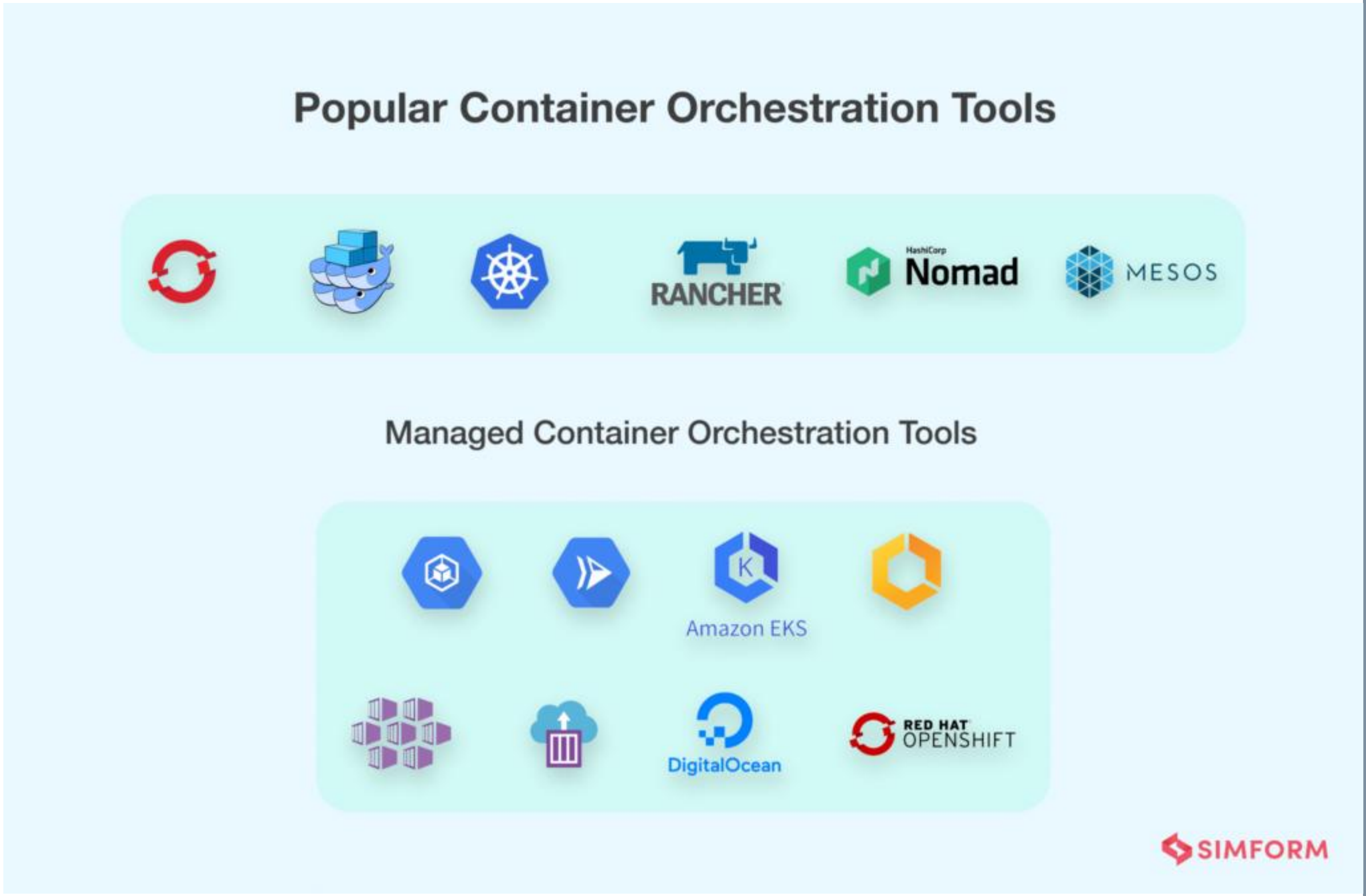
Running containers at scale requires orchestration and management of distributed, containerized applications via an orchestration platform such as Kubernetes.

Example of Container Orchestrator

Kubernetes: **a portable, extensible, open source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. Originally developed by Google**

Docker Swarm: a native clustering and orchestration tool for Docker containers. It enables you to manage a cluster of Docker hosts (also called nodes) as a single virtual host.

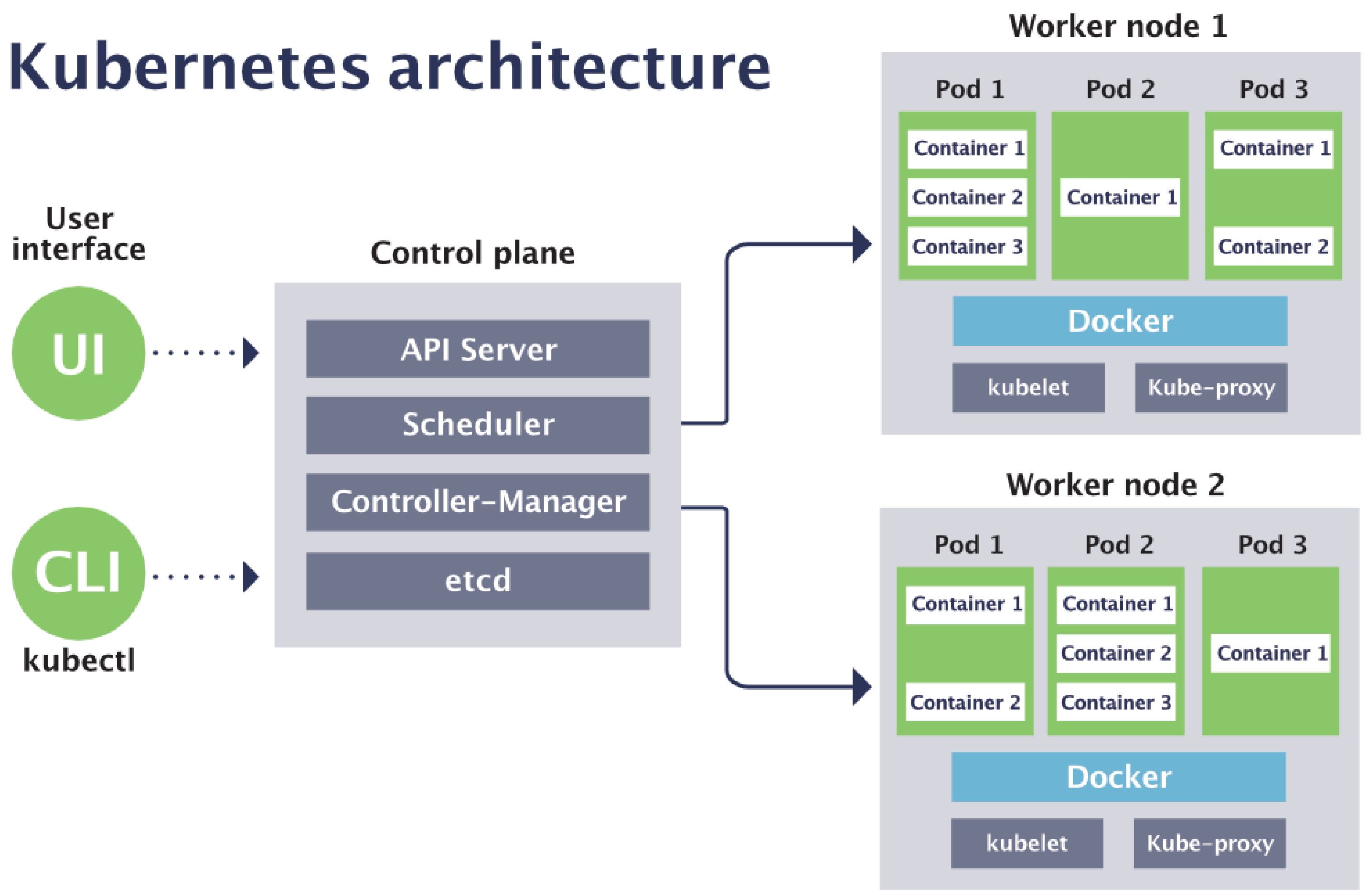
Apache Mesos: is built using the same principles as the Linux kernel, provides applications with API's for resource management. Native support for launching containers with Docker and AppC images.

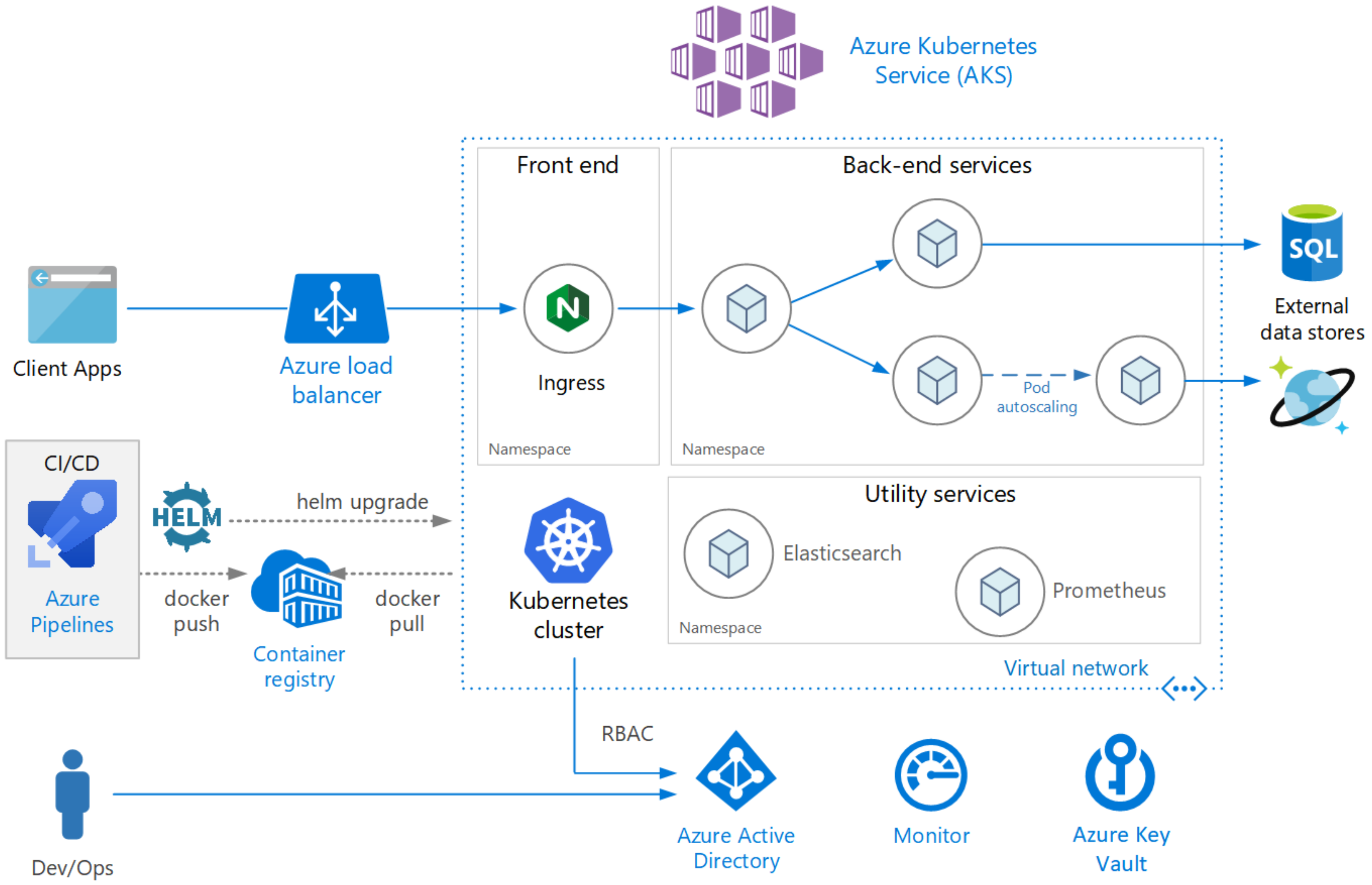


Popular Container Orchestration Tools

Managed Container Orchestration Tools

Kubernetes architecture




Azure Kubernetes Service (AKS)
Front end
Back-end services
Client Apps
Azure load balancer
Ingress
Namespace
Namespace
Pod autoscaling
External data stores
SQL
CI/CD
HELM
helm upgrade
Azure Pipelines
docker push
Container registry
docker pull
Kubernetes cluster
Utility services
Elasticsearch
Prometheus
Namespace
Virtual network
RBAC
Dev/Ops
Azure Active Directory
Monitor
Azure Key Vault

What is a Data Orchestrator?

- ❑ A data orchestrator is a tool used to automate, schedule, and manage data workflows
- ❑ Helps coordinate multiple tasks across ingestion, transformation, and storage
- ❑ Commonly used in both batch and streaming data pipelines

- ❑ Ensures dependencies are respected and failures are tracked
- ❑ Supports alerting, retries, and monitoring for robust data operations

Few examples of Data Orchestrator

General purpose Orchestrator:

Apache Airflow is an open-source platform for developing, scheduling, and monitoring batch-oriented workflows in Python code strong integration with cloud tool

Prefect is an **open-source orchestration engine** that turns your Python functions into production-grade data pipelines with minimal friction.

ML and Kubernetes native   Orchestrator:

Argo Workflows is an open source container-native workflow engine for orchestrating parallel jobs on Kubernetes. Define workflows where each step is a container.

Kubeflow an open-source project that aims to make it easy to deploy and manage ML workflows on Kubernetes. This platform provides a set of tools to develop and maintain the machine learning lifecycle on Kubernetes.

SPECIFIC SOFTWARE LAYER

Layer 1: Data Ingestion

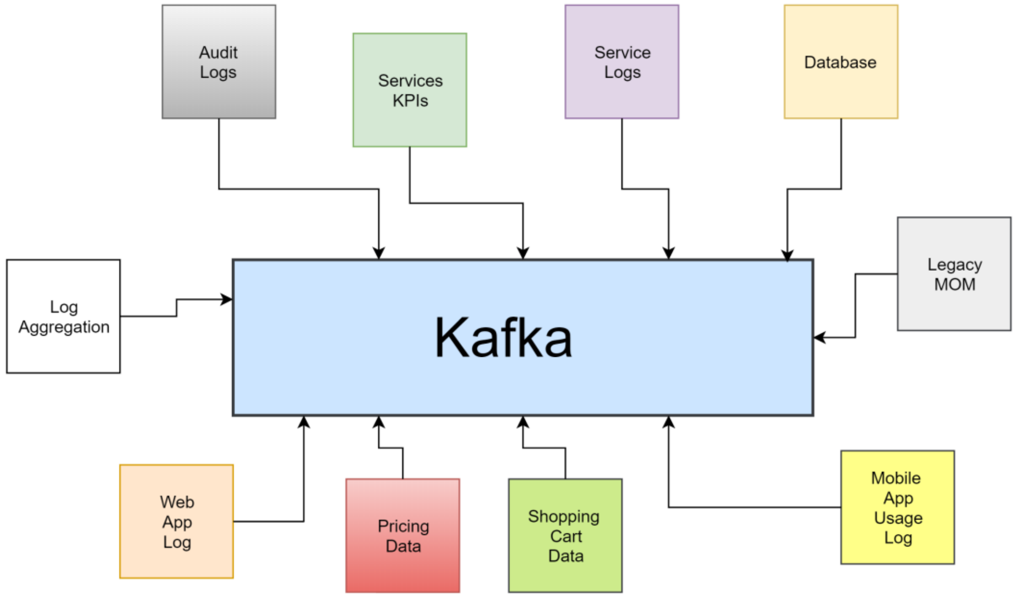
Tools that collect and bring data into the platform

- Real-time and batch ingestion
- Examples: Apache Kafka, Fivetran, Airbyte

Notice that for FAIR-by-design data pipelines, data ingestion depends on instrumentations (they may expose API or it is necessary realizing ad hoc solutions). In general we have custom solutions.

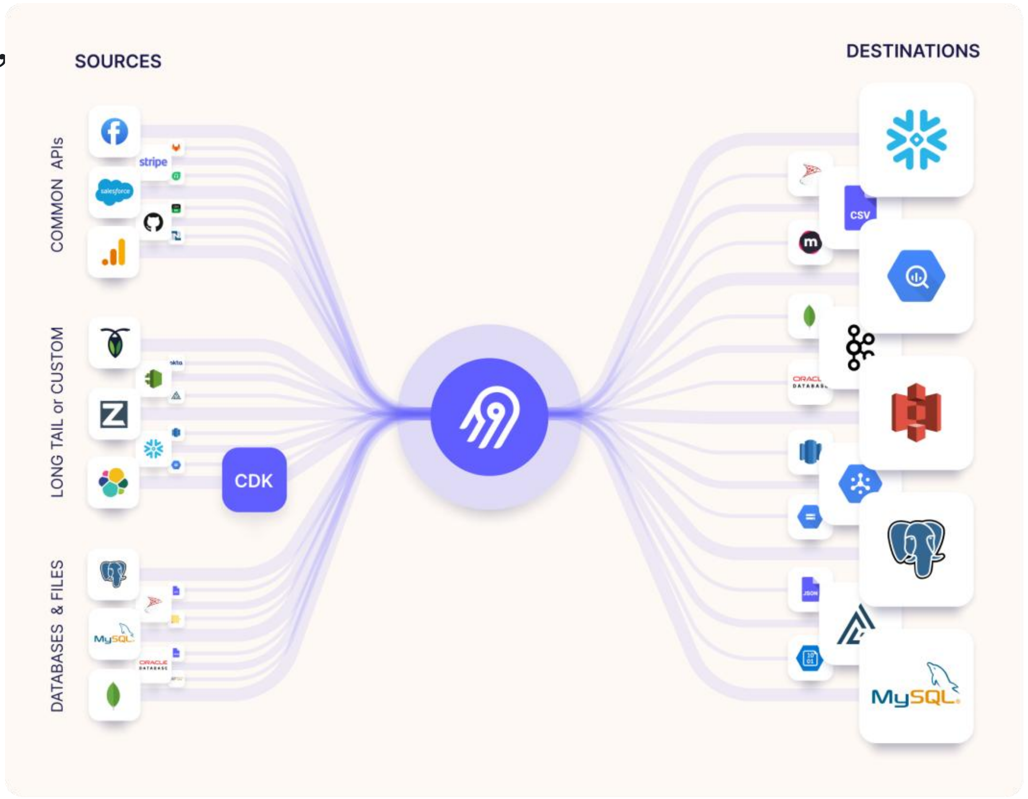
Apache Kafka

Apache Kafka is an **open-source platform** designed for real-time data streaming. It was developed by LinkedIn and later became part of the Apache Software Foundation[1]. Kafka is used to decouple data streams and systems, allowing data to flow smoothly from source systems to target systems without direct integration between them



Airbyte

Airbyte is an open-source data integration platform. It helps consolidate data from various sources into data warehouses, lakes, and databases.



Layer 2: Data Storage (I/II)

- Centralized location for raw and structured data
- Durable, scalable, and secure

Technically it belongs to the common sw layer, but we preferred privileging the data stack horizontal conceptual layers

File Storage (file= finite-length stream of bytes):

The FS storage files into directory tree. At each file/directory are associated metadata and when we loo kfor a file in a given path, starting from the root '/' the FS look sat metadata at each hierarchy level.

Object Storage (object=a specialized file-like construct, any type of file):

an object store is a key-value store for immutable data objects. We loose writing flexibility (no append operation, we must rewrite) but they allow data storage across massive disk clusters (support parallel stream writes and reads). There is no hierarchical structure (the 'folder-tree' is fictitious)

Layer 2: Data Storage (II/II)

Data Warehouse:

central data hub used for reported and analysis. Data is highly formattedand structured for analytics use cases

Cloud Data Warehouse:

Amazon redshifts kicked off such revolution. Data is hosted in object storage allowing virtually limitless storage.

Data Marts: subset of data warehouse designed to serve departments

Data Lake:

emerged in big data era, the idea is dumping data without rigid structure (any size, any tipe). Data lake 1.0 started with HDFS. Processing data is challenging, even common operation in SQL.

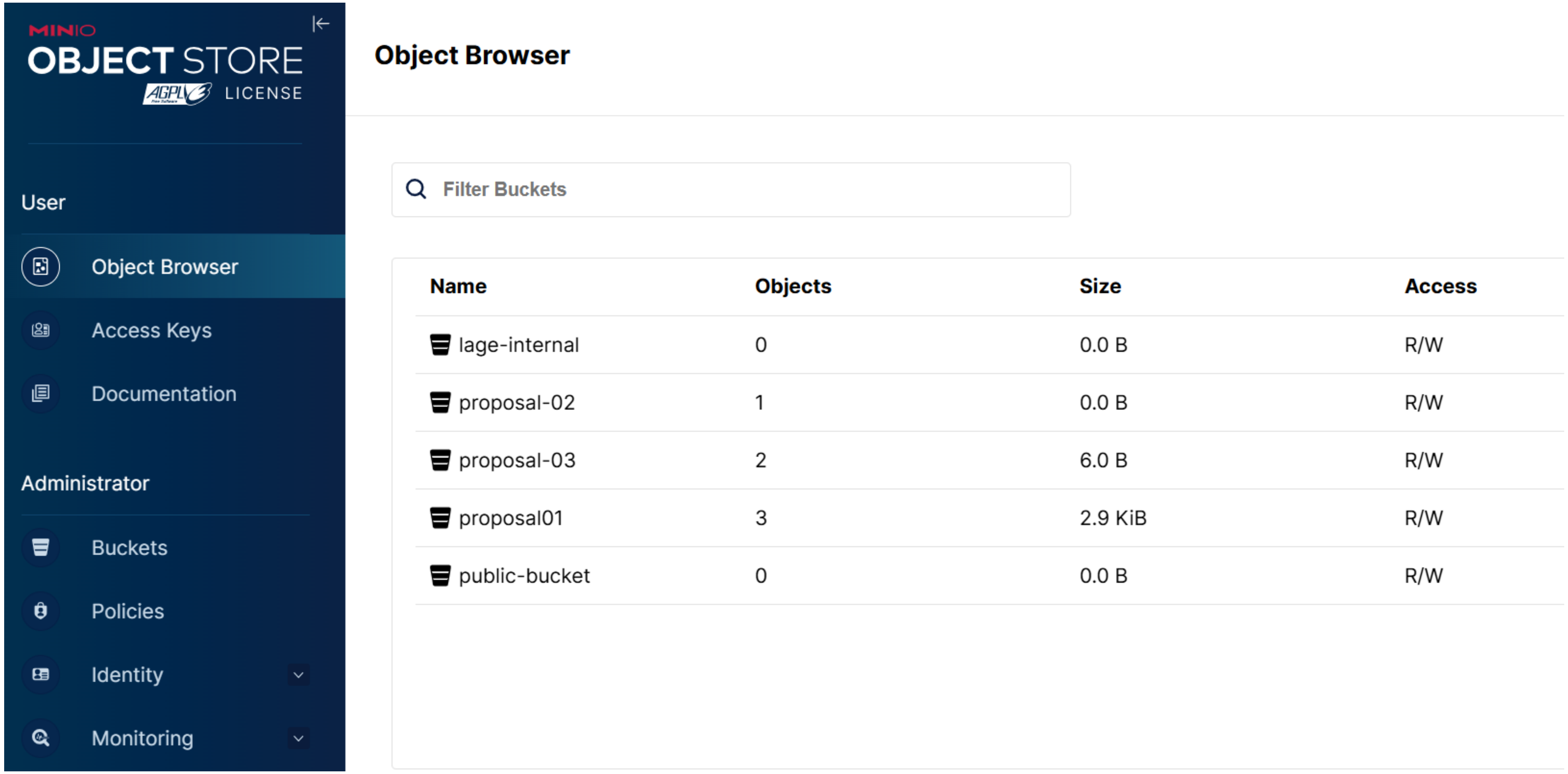
Amazon S3

- Cloud object storage from AWS
- Highly available and durable
- Ideal for raw data, backups, data lake storage
- Pay-as-you-go model

MinIO

- Fully compatible with Amazon S3 API
- Kubernetes Native (in ORFEO is delpoyed as a Kubernetes POD)
- High Performance (speed!)
- distributed under the GNU AGPL v3 license, supported by a company and with a large community

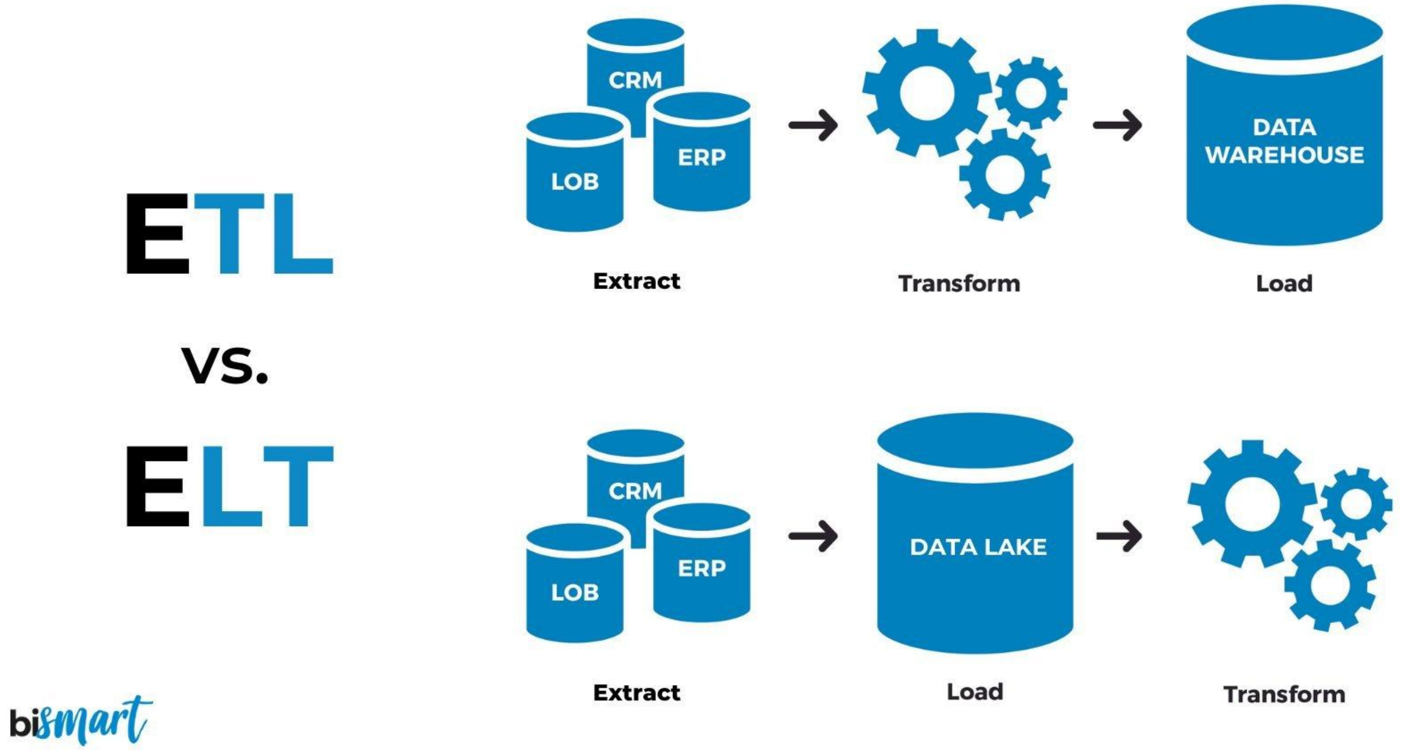
MinIO



Name	Objects	Size	Access
lage-internal	0	0.0 B	R/W
proposal-02	1	0.0 B	R/W
proposal-03	2	6.0 B	R/W
proposal01	3	2.9 KiB	R/W
public-bucket	0	0.0 B	R/W

Layer 3: Data Processing

Transforming, cleaning, and modeling data

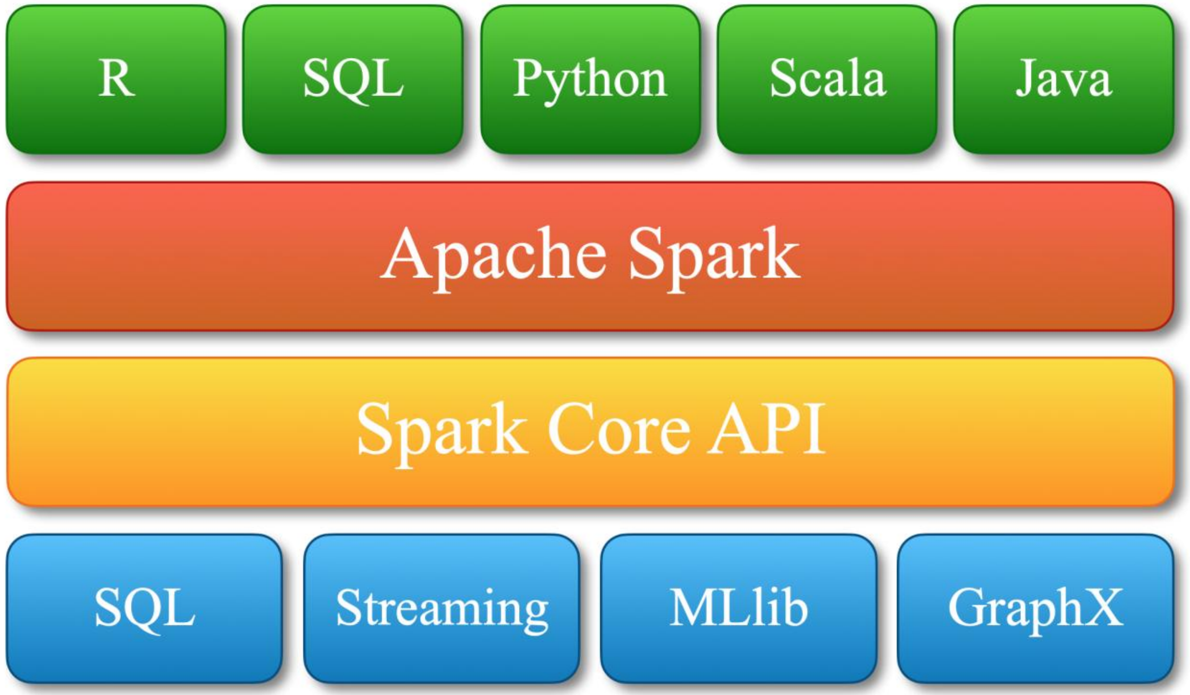


Apache Spark

Apache Spark™ is a multi-language engine for executing data engineering, data science, and machine learning on single-node machines or clusters.

Unify the processing of data in batches and real-time streaming,

Perform Exploratory Data Analysis (EDA) on petabyte-scale data without having to resort to downsampling



Apache Spark™ - Unified Engine for large-scale data analytics

Apache Spark

APACHE Spark

Data science and Machine learning

scikit learn
pandas
TensorFlow
PyTorch
mlflow
R
NumPy

SQL analytics and BI

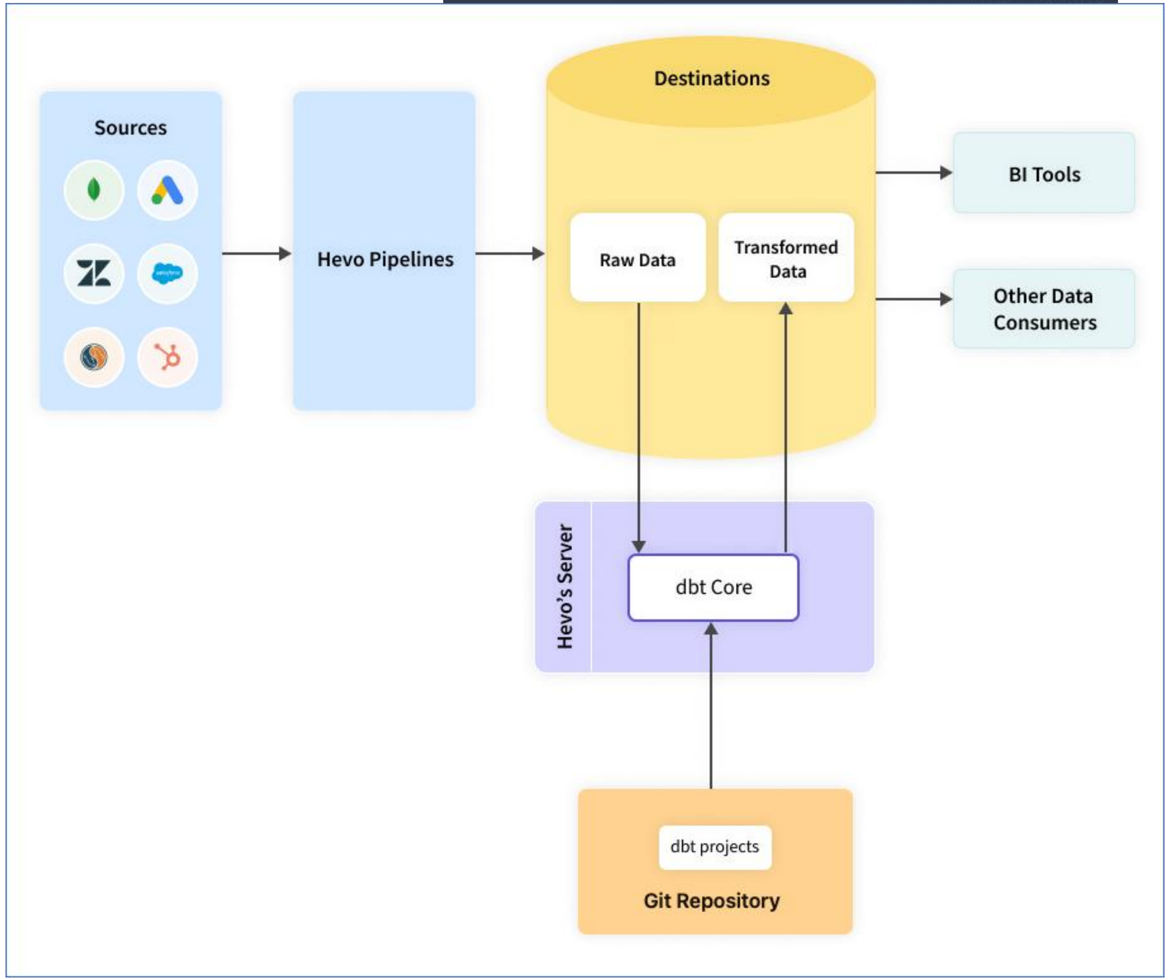
Apache Superset
Power BI
Looker
redash
tableau SOFTWARE
dbt

Storage and Infrastructure

elasticsearch
mongoDB.
APACHE kafka.
DELTA LAKE
kubernetes
Apache Airflow
Parquet
Microsoft SQL Server
cassandra
Apache orc

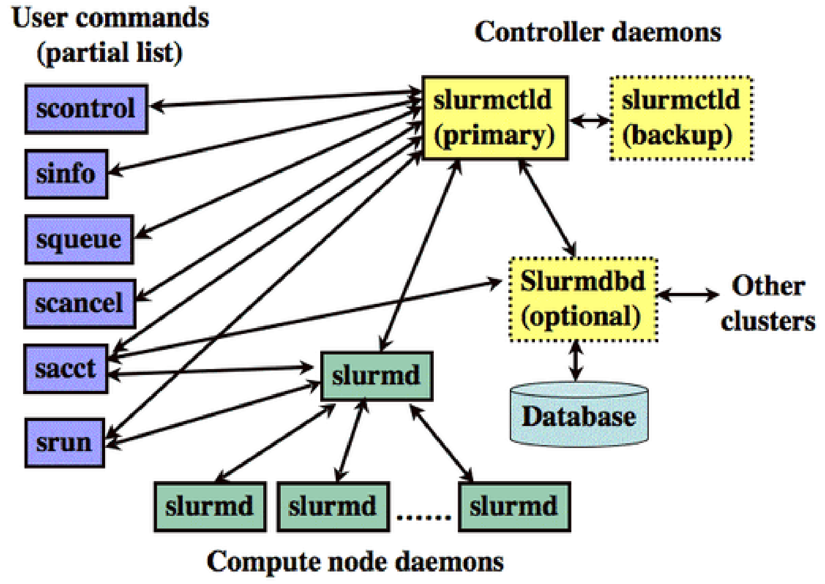
Data Build Tool

dbt (Data Build Tool) is an open-source tool that enables data analysts and engineers to transform data in their warehouse more effectively. It is a workflow tool that allows to write SQL code to transform data and then it handles the workflow of running queries in the correct order, testing results, and documenting datasets.



Slurm

Slurm is an open source, fault-tolerant, and highly scalable cluster management and job scheduling system l Linux clusters



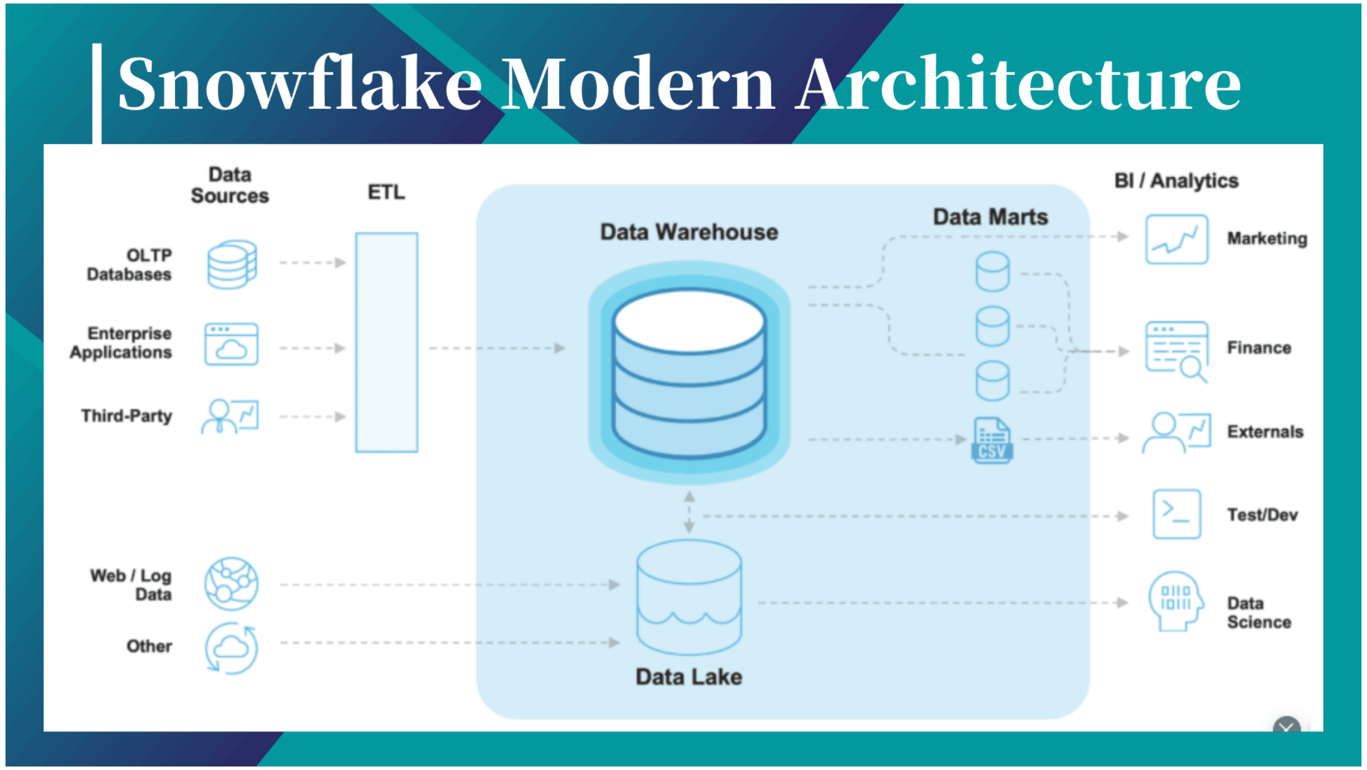
Other workload manager: Stonebranch (free) Red Hat OpenShift (pay)

Layer 4: Data Analytics

- Query engines
- Interactive exploration

Snowflake

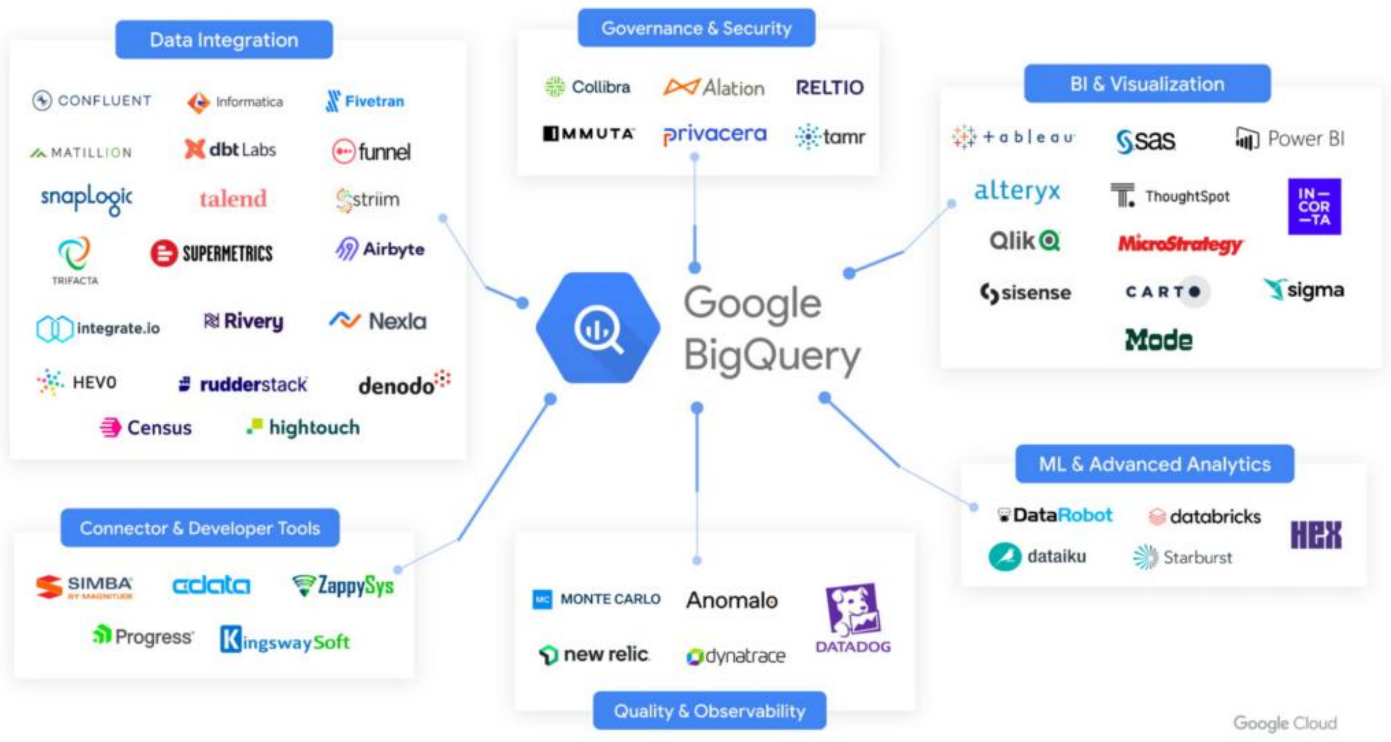
Snowflake is a cloud-based data warehousing platform known for its scalability, flexibility, and ease of use. It is designed to handle large volumes of data efficiently and cost-effectively by leveraging the benefits of cloud computing platforms. It is built specifically for the cloud, allowing it to scale resources up or down based on demand.



BigQuery

BigQuery

BigQuery is a data warehouse that many companies.
BigQuery's serverless architecture allows to quickly execute standard SQL queries and analyze millions of data rows in seconds.
Data may be stored both in Google Cloud Storage in files and buckets or in BigQuery storage.



Layer 5: Data Visualization and Business Intelligence

- Reporting,   extract final value from the big data
- May be descriptive, diagnostic, predictive

Tableau

powerful, secure, and flexible end-to-end analytics platform.
Both in desktop and cloud version

Power BI

Collection of sw, services, connectors created by Microsoft for BI. Compatible (of course!) with all Microsoft services and storage system (Azure, Access,..)

Notice that   in a research context Layers 3-4-5 ‘collapse’ in 1 or 2 layers according to the domain, and are done by the same researcher/group.

Layer 6: Data Governance

- ❑ Data governance tools are essential for managing the availability, integrity, and confidentiality of enterprise data.
- ❑ These tools help organizations ensure that data is handled consistently and not misused, addressing data privacy and security issues effectively[1].

Key Data Governance Tools

- Alation Data Catalog: it offers a collaborative data catalog with automated metadata ingestion.
- Ataccama: A Platform-as-a-Service for self-driven data management and governance, uses AI-driven automated capabilities to analyze data quality and classify data
- Collibra: it automates data operations and keeps cross-functional teams on the same page. It provides features like natural language search, data governance automation, and data stewardship..

Cloud Platform Overview

- Infrastructure as a Service (IaaS) : provides **virtualized computing resources** over the internet—like servers, storage, and networking.
  Customer manages the operating system and software; the provider manages the hardware.
- Platform as a Service (PaaS): PaaS provides a **ready-to-use platform** for developing, running, and managing applications without dealing with the underlying infrastructure.
  Customer focus on coding; the provider handles servers, OS, and middleware.
- Major providers: AWS (amazon), GCP, Azure
- Offer scalable compute, storage, and networking

Google Cloud Platform (GCP)

- Strong in analytics and AI/ML tools
- Services: BigQuery, Dataflow, Vertex AI
- Deep integration with open-source tools

Microsoft Azure

- Enterprise-friendly cloud provider
- Services: Azure Synapse, Blob Storage, ML Studio
- Strong integration with Microsoft stack

Identity Providers

- ❑ An IdP service manages and verifies user identities,
- ❑ It allows secure access to applications, systems, and networks.
- ❑ IdP acts as a central authentication authority towards services or different platform

- ❑ IdP authentication is a key component of identity and access management (IAM) frameworks, letting users securely access resources.
- ❑ IdP systems also support single sign-on (SSO) and multi-factor authentication (MFA).

- ❑ IdPs verify user identities by checking them against a centralized database. When authorized users attempt to log in, the IdP provider authenticates them and allows users to access multiple connected applications

What Is an Identity Provider (IdP) and How Does It Work? - Security Boulevard

Identity Provider Examples: FreeIPA vs Authentik

❑ FreeIPA:

- Open source identity management system by Red Hat
- Includes LDAP, Kerberos, DNS, and certificate authority
- Suited for centralized authentication in Linux environments

❑ Authentik:

- Modern, open-source identity provider and access portal
- Supports OAuth2, SAML, LDAP, and OpenID Connect
- Focus on UI/UX, self-service flows, and external integrations
- Good for cloud-native, containerized applications

LEFT FOR NEXT TIME …

WE WILL START FROM SOME CONSIDERATIONS

- ❑ From last lesson, I wanted to investigate a bit more on Eni HPC (how it is used etc)

- ❑ Big hype regarding AI data center energy consumption (fair) but how many data centers are in the world? How many are AI oriented?