# Big Data Management
## Building a Big Data Architecture

DTIM Research Group
Universitat Politècnica de Catalunya (UPC) - BarcelonaTech

# Introduction and general framework

# Introduction - Early 2000s

- Relational, expensive and monolithic data management systems buckled under the demands of **modern data processing**.
  - New systems had to be cost-effective, scalable, available, and reliable.

# Introduction - Early 2000s

- Relational, expensive and monolithic data management systems buckled under the demands of **modern data processing**.
  - New systems had to be cost-effective, scalable, available, and reliable.

- Several innovations allowed **distributed** computation and storage on massive computing clusters at a **vast scale**.

# Introduction - Early 2000s

- Relational, expensive and monolithic data management systems buckled under the demands of **modern data processing**.
  - New systems had to be cost-effective, scalable, available, and reliable.

- Several innovations allowed **distributed** computation and storage on massive computing clusters at a **vast scale**.

- The **Big Data** era had begun, and the profile of the **Big Data engineer** emerged.

# Introduction - 2000s & 2010s

- Explosion of tools.

# Introduction - 2000s & 2010s

- Explosion of tools.

- Engineers had to be proficient in **both** low-level infrastructure hacking and software development.
  - Managing these tools was a lot of work and required constant attention to install, maintain, configure and upgrade.

# Introduction - 2000s & 2010s

- Explosion of tools.

- Engineers had to be proficient in **both** low-level infrastructure hacking and software development.
  - Managing these tools was a lot of work and required constant attention to install, maintain, configure and upgrade.

- Open source developers and third parties started looking for ways to **abstract, simplify,** and make big data **available** without the high administrative overhead and cost of managing their clusters

# Introduction - 2020s

- Data management trends are moving towards decentralized, modularized, managed, and **highly abstracted** tools.

# Introduction - 2020s

- Data management trends are moving towards decentralized, modularized, managed, and **highly abstracted** tools.

- Data engineers increasingly find their role focused on things **higher in the value chain**: data management, data architecture, orchestration, and general data lifecycle management.

# Introduction - 2020s

- Data management trends are moving towards decentralized, modularized, managed, and **highly abstracted** tools.

- Data engineers increasingly find their role focused on things **higher in the value chain**: data management, data architecture, orchestration, and general data lifecycle management.

Data engineering is increasingly a discipline of interoperation; connecting various technologies in streamlined processing workflows to serve ultimate business goals.

# Introduction - Project Goal

To build a Big Data Management Architecture

# Introduction - Project Goal

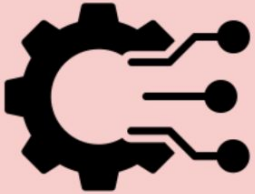
To build a Big Data Management Architecture

- Understanding the different stages of data management
- Defining the execution pipeline
- Selecting the appropriate tools/technologies
- Deploying and orchestrating the execution

# The Framework

# The Framework

**Data Ingestion**



Get data from
the **sources**

# The Framework

| Data Ingestion | Landing Zone |
| --- | --- |
| Get data from the **sources** | Store the **raw** data |

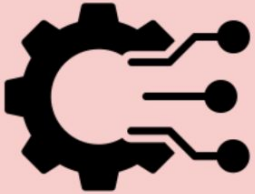
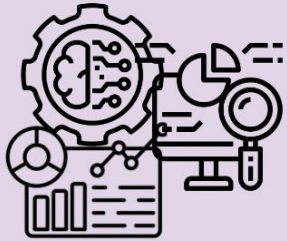
# The Framework

| Data Ingestion | Landing Zone | Trusted Zone |
| --- | --- | --- |
| Get data from the **sources** | Store the **raw** data | Apply **generic** data quality |

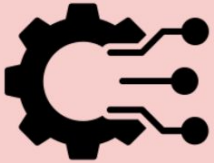
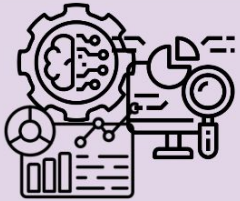
# The Framework

| Data Ingestion | Landing Zone | Trusted Zone | Exploitation Zone |
|---|---|---|---|
| Get data from the **sources** | Store the **raw** data | Apply **generic** data quality | **Arrange** data to be exploited |

# The Framework

| Data Ingestion | Landing Zone | Trusted Zone | Exploitation Zone | Data Consumption |
|---|---|---|---|---|
|  |  |  |  |  |
| Get data from the **sources** | Store the **raw** data | Apply **generic** data quality | **Arrange** data to be exploited | Get the data and generate **value** |

# The Framework

| Context | Data Ingestion | Landing Zone | Trusted Zone | Exploitation Zone | Data Consumption |
|---|---|---|---|---|---|
|  |  |  |  |  |  |
| **Define** domain, value and datasets | Get data from the **sources** | Store the **raw** data | Apply **generic** data quality | **Arrange** data to be exploited | Get the data and generate **value** |

# The Framework - Considerations

- This framework is meant to **guide** the development of the project, but in any case defines a hard set of rules to follow

| Context | Data Ingestion | Landing Zone | Trusted Zone | Exploitation Zone | Data Consumption |
|---|---|---|---|---|---|
| **Define** domain, value and datasets | Get data from the **sources** | Store the **raw** data | Apply **generic** data quality | **Arrange** data to be exploited | Get the data and generate **value** |

# The Framework - Considerations

- This framework is meant to **guide** the development of the project, but in any case defines a hard set of rules to follow.

- You can deviate from this structure if it fits your needs.

| Context | Data Ingestion | Landing Zone | Trusted Zone | Exploitation Zone | Data Consumption |
|---|---|---|---|---|---|
| **Define** domain, value and datasets | Get data from the **sources** | Store the **raw** data | Apply **generic** data quality | **Arrange** data to be exploited | Get the data and generate **value** |

# The Framework - Considerations

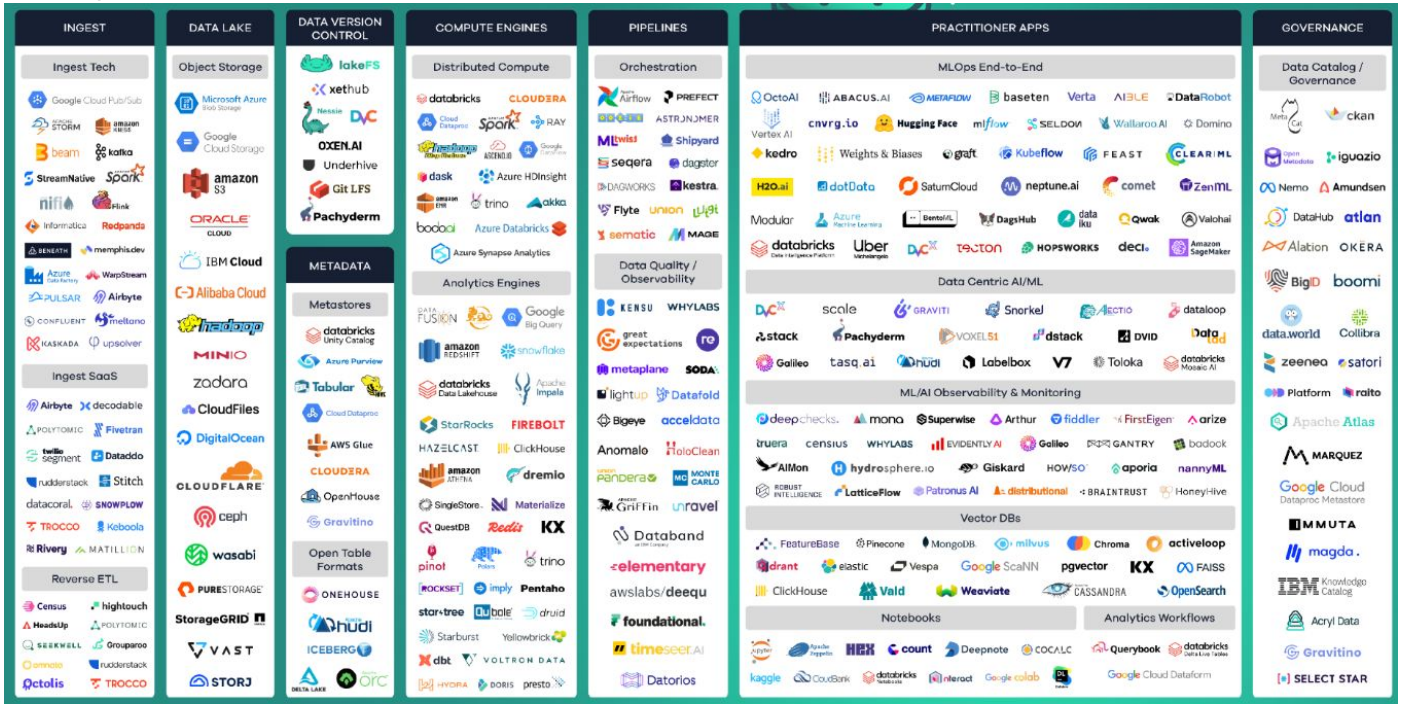
- This framework is meant to **guide** the development of the project, but in any case defines a hard set of rules to follow.

- You can deviate from this structure if it fits your needs.

- Always make sure to **justify** your decisions.

| Context | Data Ingestion | Landing Zone | Trusted Zone | Exploitation Zone | Data Consumption |
|---|---|---|---|---|---|
| **Define** domain, value and datasets | Get data from the **sources** | Store the **raw** data | Apply **generic** data quality | **Arrange** data to be exploited | Get the data and generate **value** |

# Too many tools




Data management landscape in 2024 ([lakefs])

# Organization, deliverables and guidelines

(You have both this and the previous information in LearnSQL)

# Organization

- Groups of **3 people** (4 if someone is left alone).

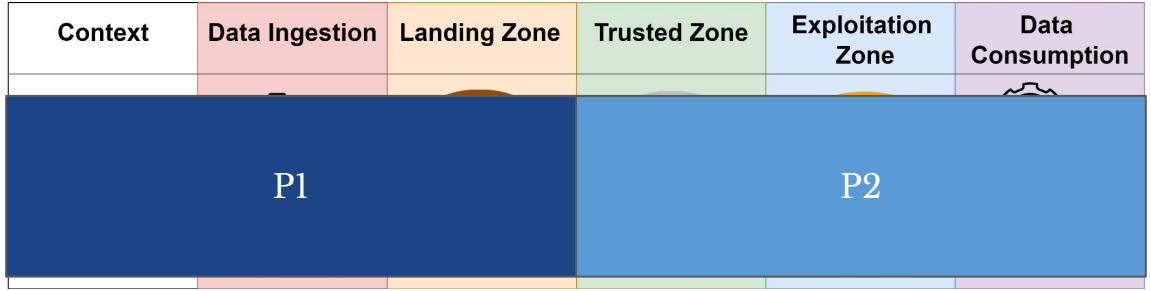
# Organization

- Groups of **3 people** (4 if someone is left alone).
- Evaluation is divided into two main deliverables

# Organization

- Groups of **3 people** (4 if someone is left alone).

- Evaluation is divided into two main deliverables



# Organization

- Groups of **3 people** (4 if someone is left alone).
- Evaluation is divided into two main deliverables


- In each deliverable we demand **two artifacts**
  - A explanatory **document** of the project.
  - A project **repository** with the developed code.

# Organization

- Groups of **3 people** (4 if someone is left alone).
- Evaluation is divided into two main deliverables


- In each deliverable we demand **two artifacts**
  - A explanatory **document** of the project.
  - A project **repository** with the developed code.
- The final grade of the project will be:

$$P = 0.5P1 + 0.5P2$$

# Follow-up deliverables

- You will have a deliverable **before** each follow-up session

# Follow-up deliverables

- You will have a deliverable **before** each follow-up session

| | Week (Monday) | Project/Problems (Wednesday) | Theory (Thursday) |
|---|---|---|---|
| 1 | 2/10/2025 | (No class) | Introduction + Big Data Design |
| 2 | 2/17/2025 | P1-Presentation | Distributed Data Management |
| 3 | 2/24/2025 | Problems | Distributed Data Processing |
| 4 | 3/3/2025 | P1-Followup | Distributed File Systems |
| 5 | 3/10/2025 | Problems | Key-Value Stores |
| 6 | 3/17/2025 | P1-Followup | Document Stores |
| 7 | 3/24/2025 | Problems | Autonomous Learning (no class) |
| 8 | 3/31/2025 | Partials | P1-Delivery |
| 9 | 4/7/2025 | (No class) | Big Data Architectures |
| 10 | 4/14/2025 | Easter Week | |
| 11 | 4/21/2025 | P2-Presentation | MapReduce I |
| 12 | 4/28/2025 | Problems | May 1st |
| 13 | 5/5/2025 | P2-Followup | MapReduce II |
| 14 | 5/12/2025 | Problems | Spark |
| 15 | 5/19/2025 | P2-Followup | Stream Management |
| 16 | 5/26/2025 | Problems | Data Engineering |
| 17 | 6/2/2025 | P2-Delivery + Final Exam | |

# Follow-up deliverables

- You will have a deliverable **before** each follow-up session
- In each, you will have to include parts of the architecture

# Follow-up deliverables

- You will have a deliverable **before** each follow-up session

- In each, you will have to include parts of the architecture

- **These will not directly count towards the grade**
  - The goal is to structure the development and provide better feedback
  - If significant effort is show, it will be taken into consideration for the final evaluation

# Evaluation guidelines

- The document is a fundamental part of the project
  - Showcase rigorous thinking and soundness

# Evaluation guidelines

- The document is a fundamental part of the project
  - Showcase rigorous thinking and soundness

- The implementation has to present the desirable properties of software development
  - That is, dynamicity, reusability, openness, reproducibility, etc.

# Evaluation guidelines

We want you to be ambitious and develop a complex project