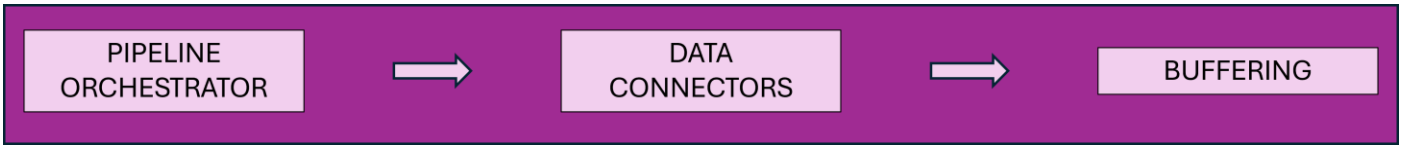
PART TWO – PLATFORM DESIGN

DATA INGESTION:

This layer is responsible for collecting data from multiple sources, its initial processing if necessary or desired, and loading it into the storage system. We will avoid processing the data in this step, as we prefer to store the raw data.

- Pipeline Orchestrator (Apache Airflow or Prefect): allows us to orchestrate, schedule and coordinate data ingestion pipelines to the storage system
- Data connectors (Apache NiFi or Logstash): make connecting to different data sources easy. They extract data from RESTful APIs, relational databases, file systems, etc., and transmit it to storage systems.
- Buffering (Apache Kafka or AWS Kinesis): act as a messaging and buffering system for data in transit. Receives data from connectors and transmits it to storage or distributes it for real-time processing.
- Data Transformations (Apache Spark, AWS Glue): Perform transformations on data before storing it. Processes incoming data from Kafka or Kinesis and loads it transformed into the Data Lake or Data Warehouse. We will avoid this.



DATA STORAGE / STORAGE COMPONENTS:

This layer is responsible for storing and managing data in various forms, allowing efficient and secure access.

We beggin with raw data in a low-cost storage, with little or no modifications or filtering made. Data is not organized nor have an harmonized format, so it allows more flexibility to grow as we previously don't know yet the future analysis we will need for our use case.

Later on, we get data derivatives but still in a low-cost storage. It is based on the business usage of the analysis. Each use case should have its own independent and dedicated flow of data, even if the information is duplicated somewhere else. In this step we can optimize the data, obtaining mostly aggregated, filtered and transformed data to fit a specific business need.

We will mantain cache datastores to allow user interactions with the result of data analytics in terms of query capabilities and speed. This might be a more expensive solution, so it is focused only on use cases for the users.

All of the orchestration will be made having automatization in mind, based again in the use cases, data sources, technical capabilities... (StepFunctions, Apache AirFlow, Upsolver...)

- Data Lake (Amazon S3, Azure Data Lake Storage): Massive storage of raw data in a variety of formats. It receives data from the ingestion pipelines and allows it access for analysis and further processing. It provides a highly effective solution that can process large amounts of untransformed information and can store raw data that does not yet have a set purpose.
- Data Warehouse (Amazon Redshift, Google BigQuery, Snowflake): Structured data warehouse for fast queries and analysis. Receive transformed, processed, and purpose-driven data structured from transformation pipelines and provide SQL query interfaces.
- Data Catalog (AWS Glue Catalog, Apache Atlas): Maintains an inventory of all available data, making it easier to find and govern data. It integrates with the Data Lake and Data Warehouse, cataloguing all data and providing metadata for queries and audits.
- Data Lakehouse (Delta Lake, Apache Hudi): Combine capabilities of a Data Lake and Data Warehouse, delivering high-performance transactions and queries on stored data. It acts as a unifying layer that enables analysis of both structured and unstructured data. It allows you to implement similar data structures and data management

features but with low-cost storage. By combining them, data can move faster without having to access multiple systems.

DATA PROCESSING:

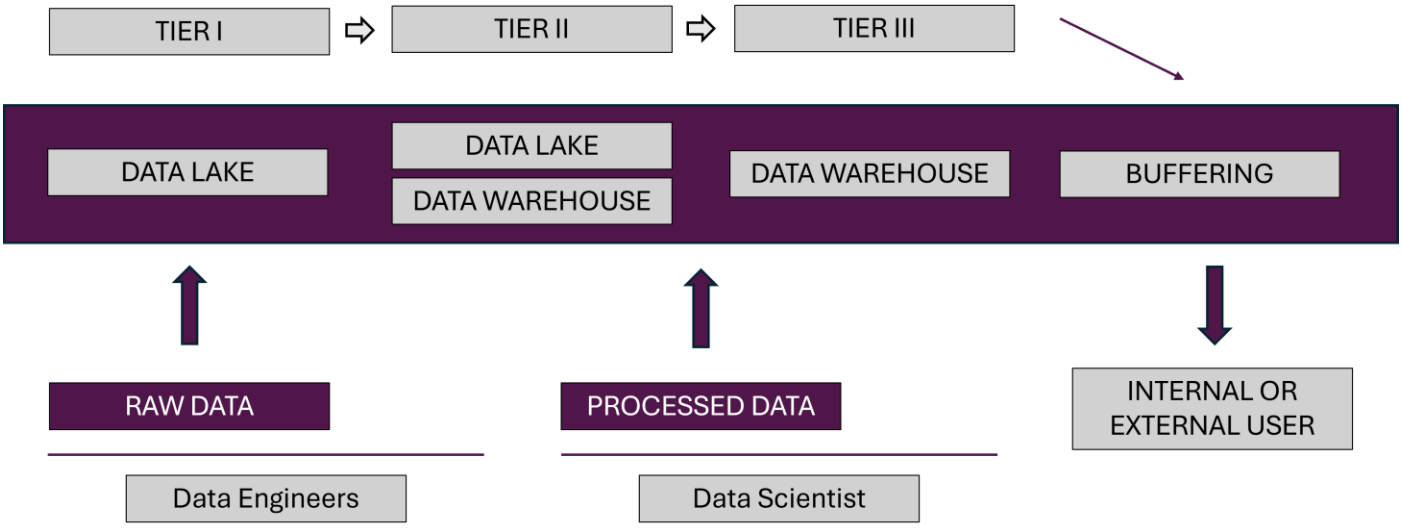
This layer is responsible for executing data processing, real-time analysis, and advanced modelling tasks.

- Batch Processing Engine (Apache Spark, Hadoop MapReduce): Processing large volumes of data in batch mode. Process data from the Data Lake or Data Warehouse and write results to these same systems or other destinations.
- Real-Time Processing Engine (Apache Flink, AWS Kinesis Data Analytics): Process and analyze data in real-time. It consumes data from Kafka or Kinesis and produces real-time results that can be stored or visualized directly.
- Machine Learning Environments (Amazon SageMaker, Databricks): Provide an environment for training and deploying machine learning models. You access data from the Data Lake or Data Warehouse for model training and can write results to a Data Warehouse or send them to APIs.
- ETL workflows (Apache NiFi, AWS Glue Jobs): Execute and automate data extraction, transformation, and loading tasks. They process data from various sources and prepare it for analysis or storage.

DATA ANALYSIS:

This layer provides interfaces and tools so that users can interact with data, perform

- Notebook Platform (JupyterHub, Amazon SageMaker): Provides an interactive environment for analysis and development in Python and R. Users access data stored in the Data Lake and Data Warehouse via notebooks, performing analysis and visualizations.
- Query Engine (Presto, Trino, Google BigQuery): Allows queries on data in the Data Lake and Data Warehouse. It makes it easy to run data analytics in large volumes.
- Dashboards and Reporting (Tableau, Power BI, Apache Superset): Create and visualize interactive dashboards for data monitoring and analysis. Connects with the Data Warehouse to extract data and visualize it interactively for end users.
- Data Access APIs (RESTful APIs, GraphQL): Provide programmatic access to data for external applications or custom scripts. They allow users to access specific subsets of data from the Data Lake or Data Warehouse.



DATA ACCESS / SECURITY LAYER:

This layer ensures the protection, availability and continuous operability of the platform.

- Authentication and authorization systems (OAuth 2.0, AWS IAM, LDAP): Manage access to the platform and data, ensuring that only authorized users can access specific resources. They integrate with all components that require controlled access, from notebooks to the Data Warehouse.

- Monitoring and alerting systems (Prometheus, Grafana, Amazon CloudWatch): Monitor the health and performance of the platform, and generate alerts in case of incidents. They connect with all the components of the platform, monitoring their performance and generating alerts for the operations team.
- Logging and auditing systems (ELK Stack, Splunk): Record and analyze events and activities on the platform for auditing and troubleshooting. They collect logs of all operations and access to data, enabling audits and regulatory compliance.
- Data encryption and security systems (AWS KMS, HashiCorp Vault): Ensure that data is encrypted both in transit and at rest. They implement in all data storage and communication systems to ensure confidentiality.
- Data Governance (Collibra, Apache Atlas): They manage policies and rules to ensure the correct governance of data. They integrate with the data catalogue and other components to ensure that data access and use comply with corporate policies and regulations.