Project

Implementation of a (Big) Data Management Backbone

Overview and Framework

Big Data Management – FIB – UPC

Data Science Projects

Data science projects require creating **systems** that deploy **data pipelines** spanning three different areas:

- **Business understanding (domain)**
  - What do we want to analyse?
  - What is the added value of an analytical question for the organisation?
- **Data management**
  - Data discovery
  - Data modeling
  - Data storage
  - Data processing
  - Data querying
- **Data analysis**
  - Data preparation
  - Modeling
  - Validation
  - Explainability
  - Visualization

From DevOps to DataOps / MLOps

DevOps ignores a key aspect in Data Science: data and its lifecycle

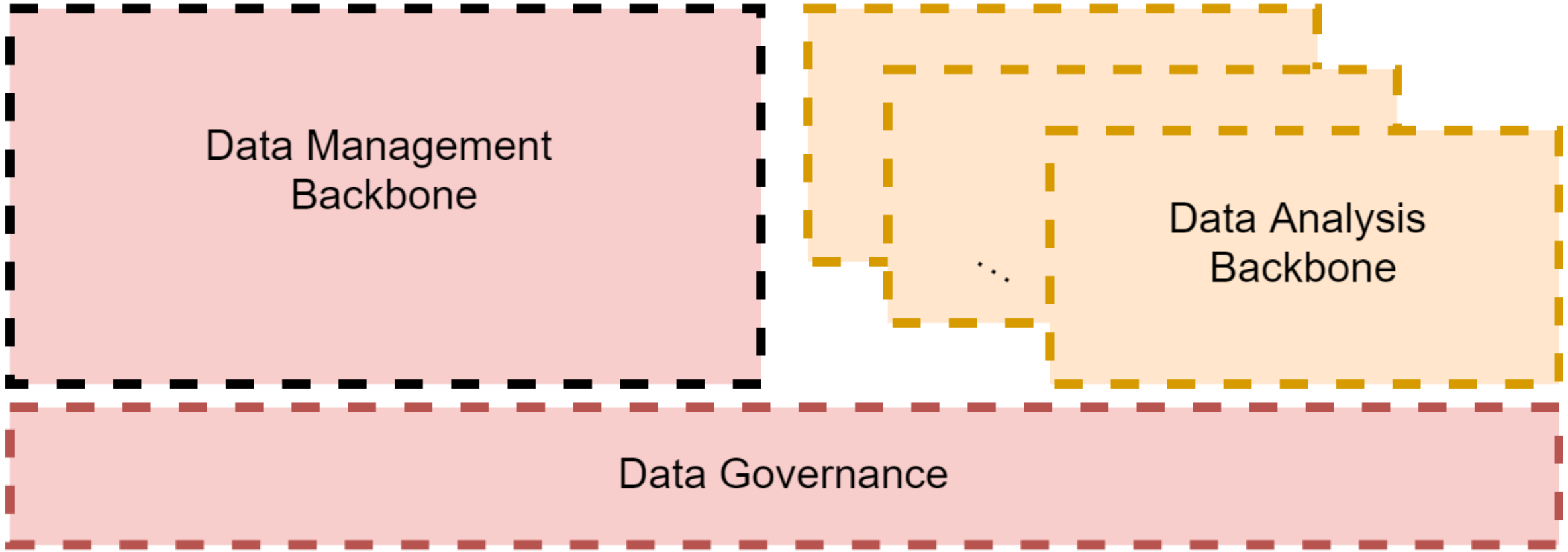
DataOps was introduced to cover this gap by *combining an integrated and process-oriented perspective on data with automation and methods from agile software engineering, like DevOps, to improve quality, speed, and collaboration and promote a culture of continuous improvement.*

Ereth, J. (2018). DataOps - Towards a Definition. LWDA 2018: 104-112.

*In short, the DataOps lifecycle is needed to manage the complexity of the data lifecycle in data science projects (**data engineering**)*

***Disclaimer**: you may read about MLOps too. However, in essence, DataOps / MLOps talk about the same problem. Simply, the latter focuses more on the ML part while the former cover the whole data lifecycle. We aim at providing an holistic view in this course and that is why we choose DataOps in front of MLOps*

DataOps in a Nutshell



DataOps in a Nutshell

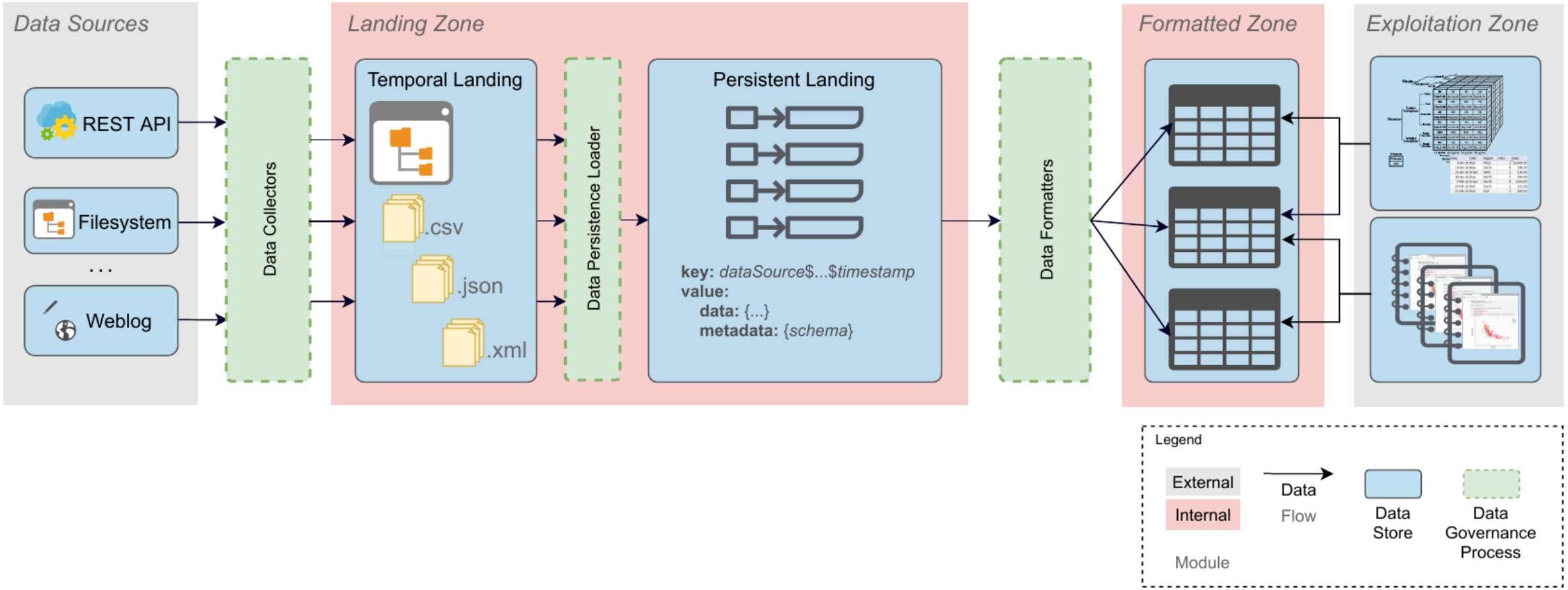
Data management backbone **(common for the whole organisation)**

- Ingest and store external data into the system
  - Data Integration (standardization and data crossing)
    - Syntactic homogenization of data
    - Semantic homogenization of data
  - Clean data / eliminate duplicates
- Expose a cleaned and centralized repository
  - Data profiling

Data analysis backbone **(repeats for every analytical pipeline)**

- Extract a data view from the centralized repository
- Feature engineering
- Specific pre-processing for the given analytical task at hand
  - Labeling
  - Data preparation specific for the algorithm chosen
- Create test and validation datasets
- Learn models (either descriptive stastitical analysis or advanced predictive models)
- Validate and interprete the model

Data Management Backbone



Data Management Backbone

- Data is ingested in the **landing zone** as it is produced (raw data)
  - The temporal landing 'stores temporarily the files' (though not necessarily deleted), until they are processed
  - The persistent landing tracks ingested files by data source and timestamp (i.e., versions)
- Data is then homogenized, according to a canonical data model in the **formatted zone** (syntactic homogenization)
  - This is where data cleaning happens. However, only generic data cleaning techniques (i.e., independent of a specific project) occur here
- The **exploitation zone** exposes data ready to be consumed / analysed either by advanced analytical pipelines or external tools. Two main kinds of tasks are conducted to generate this zone (semantic homogenization):
  - Data integration: new data views are generated by combining the instances from the trusted zone. A view may serve one or several data analysis. Relevantly, data integration spans data discovery, entity resolution, ad-hoc transformations and data loading into a target schema in a potentially different data model (not necessarily in the form of the canonical data model)
  - Data quality: data quality requires to have a wider view (e.g., instances from different sources or requiring to happen after data Integration is conducted)

Exploitation Zone

It is the zone where data is exposed either to in-house data scientists (to conduct ad-hoc advanced data analysis tasks) or to external tools

For analytical purposes, the three most extended data models are **tensors**, **relations** and **dataframes**:

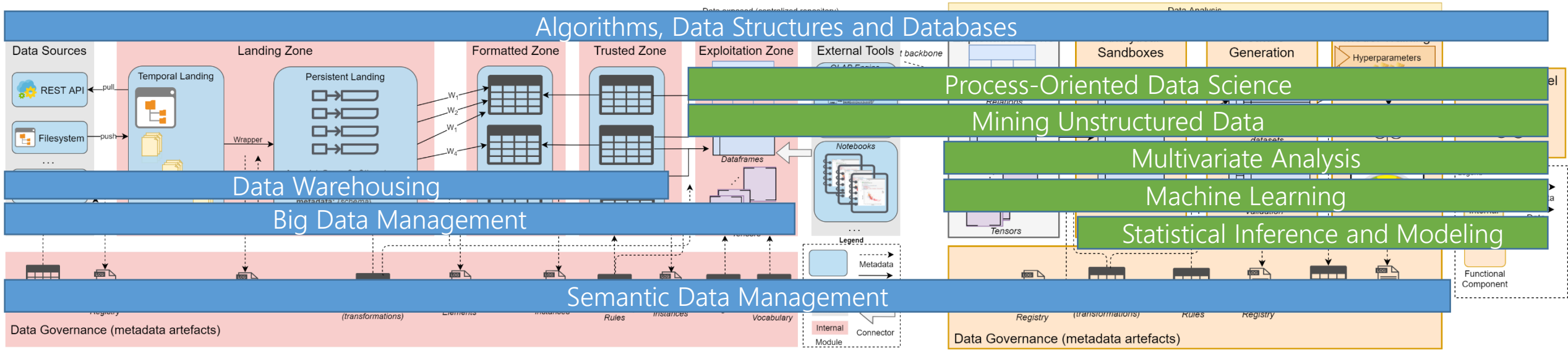
- Query & reporting (descriptive data analysis): relations (traditional OLAP engines over relational databases) or dataframes (newer engines built on top of Data Lakes)
- Machine Learning and Data Mining (predictive data analysis): typically require dataframes (e.g., R, Data Science Python libraries / frameworks, SAS, etc.). Also, distributed Machine Learning and Data Mining (e.g., MLlib) typically require dataframes. Deep Learning frameworks, instead, they require tensors to work.

***Disclaimer**: other data models may be required at this stage to connect the exploitation zone to other tools (e.g., graph data modeling). However, we focus on those models required for traditional ML / DM*

Scope of the project

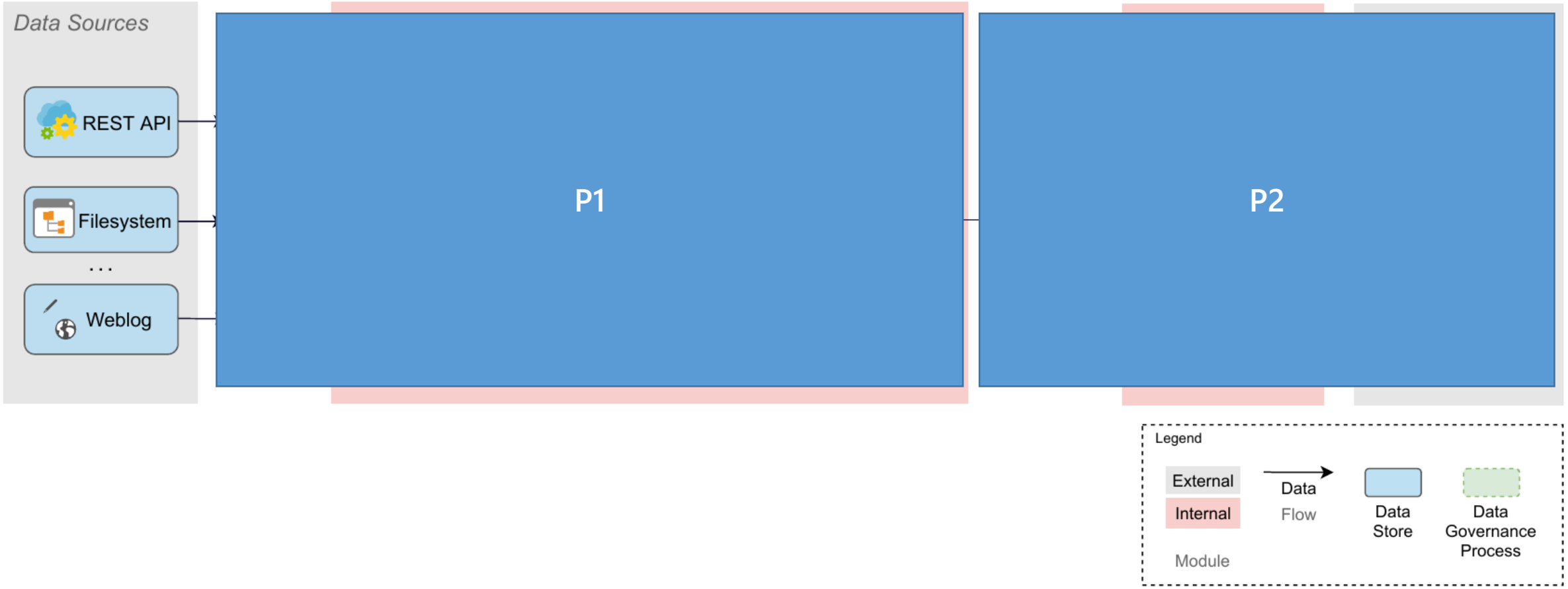
Implementation of a (Big) Data Management backbone

Overview


Algorithms, Data Structures and Databases
Process-Oriented Data Science
Mining Unstructured Data
Multivariate Analysis
Machine Learning
Statistical Inference and Modeling
Data Warehousing
Big Data Management
Semantic Data Management
Data Sources
Landing Zone
Formatted Zone
Trusted Zone
Exploitation Zone
External Tools
Sandboxes
Generation
Hyperparameters
REST API
Filesystem
Temporal Landing
Persistent Landing
Wrapper
Dataframes
Notebooks
Tensors
Legend
Metadata
Internal Module
Connector
Functional Component
Data Governance (metadata artefacts)
Data Governance (metadata artefacts)

Objective of the project

- Model, deploy and implement a Big Data Management backbone



Part 1 – Landing Zone

- Identify data sources
- Implement data collectors
- Decide on the structure and deploy the temporal landing zone
  - This is required when pulling data from external sources
- Implement the Data Persistence Loaders per source
- Decide on the structure and deploy the Persistent Landing Zone

Part 2 – Formatted and Exploitation Zone

- Decide on the kind of analytics/queries to be applied
  - Descriptive analytics
  - Predictive analytics

this will impact the choice of technology for the Formatted Zone

- Decide on the structure and deploy the Formatted Zone
- Implement the Data Formatters
  - Data cleaning
  - Data integration
  - Data reconciliation
- Decide on the structure and deploy the Exploitation Zone

Closing