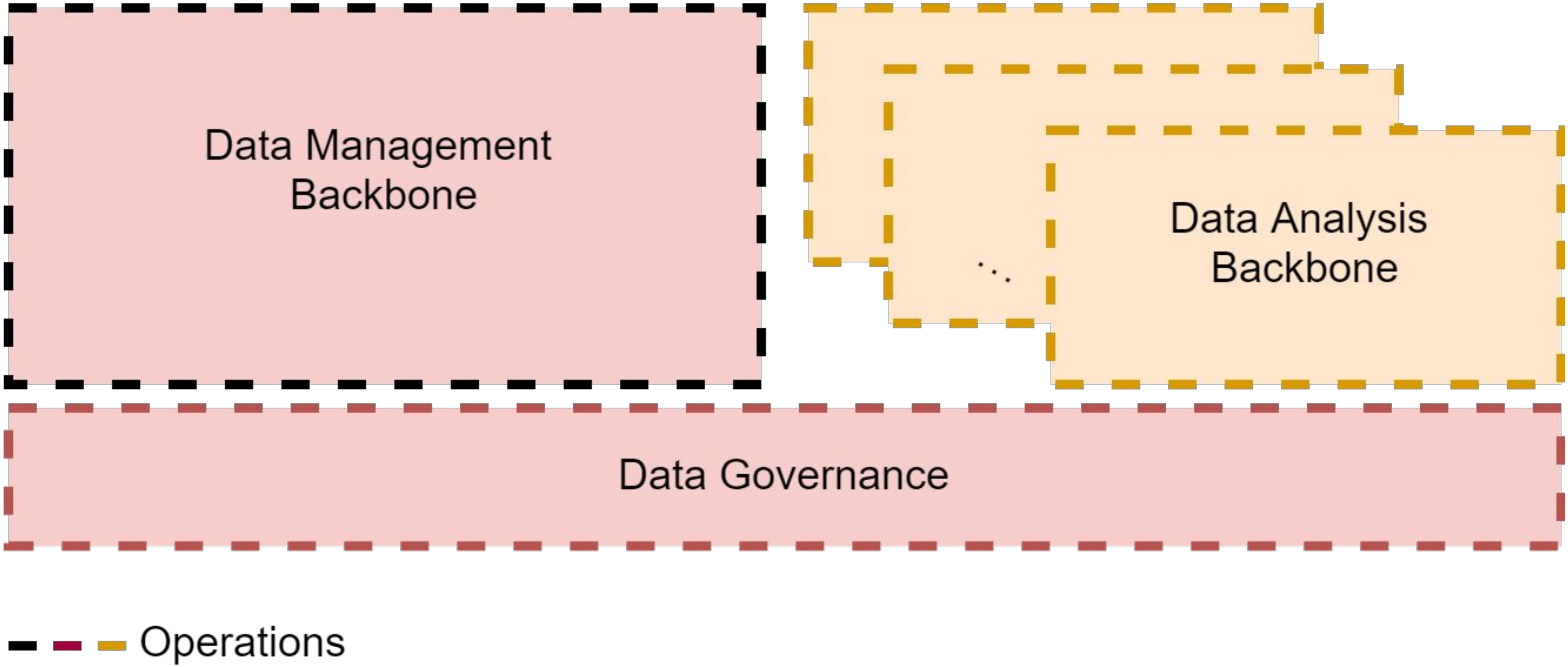
Data Science End-to-End Project

Oscar Romero, Anna Queralt and Marc Maynou

DTIM RESEARCH GROUP (http://www.essi.upc.edu/dtim/)

UNIVERSITAT POLITÈCNICA DE CATALUNYA – BARCELONATECH

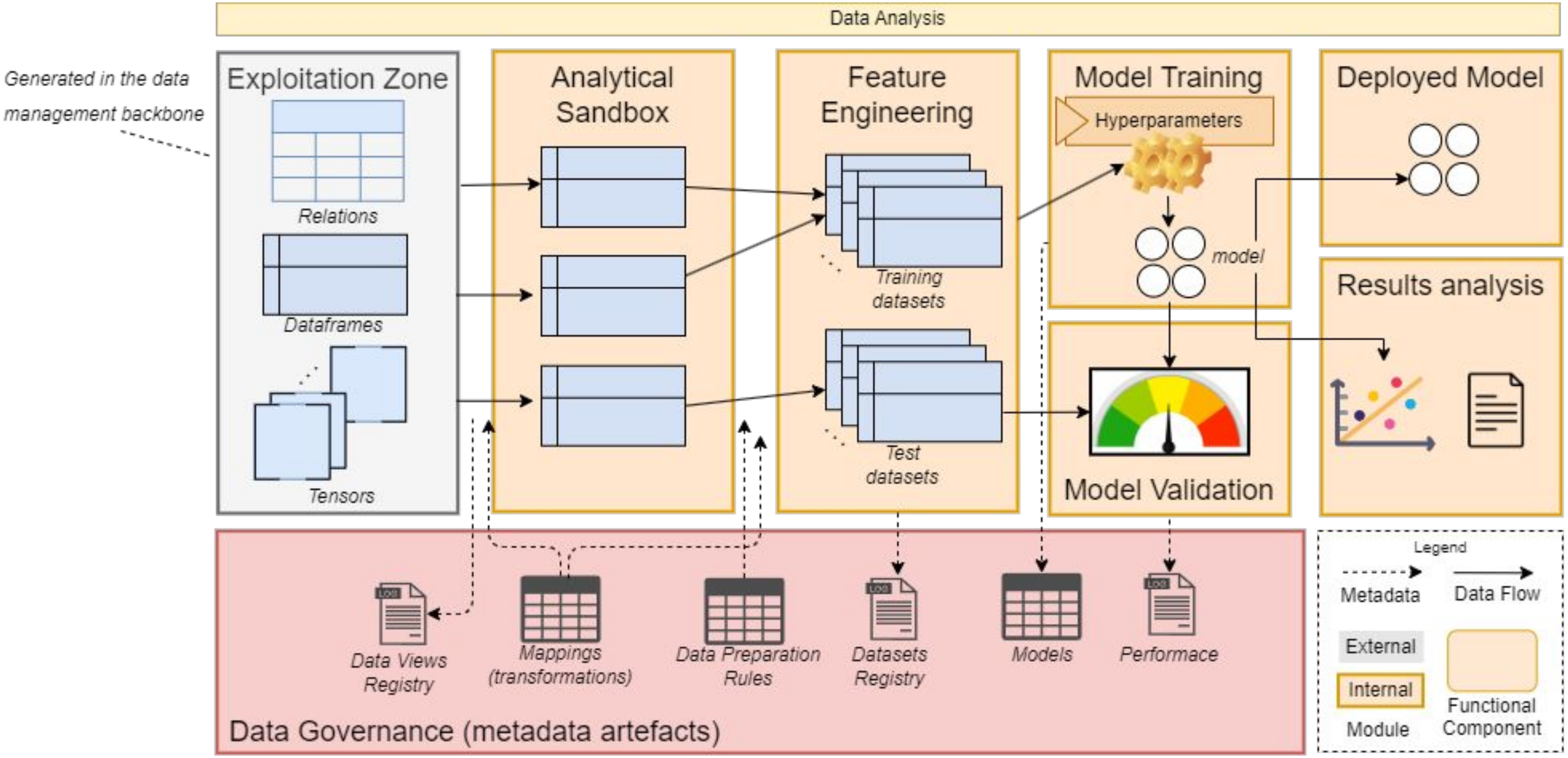
Recap: DataOps in a Nutshell



The Data Analysis Backbone

And its Governance

The Data Analysis Backbone



The Data Analysis Backbone

Unlike the data management backbone, there is an analytical pipeline per analysis needs. Thus, a project may define several analytical pipelines

- The analytical sandboxes capture subsets of the elements created in the exploitation zone and of relevance for the analysis at hand
- During feature generation, features are generated from the data in the analytical sandboxes. The following tasks take place:
    - Feature generation: creating new features from the ones available.
    - Data preparation rules, specific for the algorithm and kind of analysis. For example: discretizing values, one hot encoding for categorical data, value normalization, etc.
    - Labeling, for supervised methods, is also conducted here (if needed).
    - Two corpuses are generated: the training and test datasets.
- Model training requires choosing an algorithm and specifying the required algorithm hyperparameters and outputs a model
- Then, the generated model is validated according to some criteria (e.g., precision, recall, etc.)
- Out of the models generated, one is chosen to be deployed

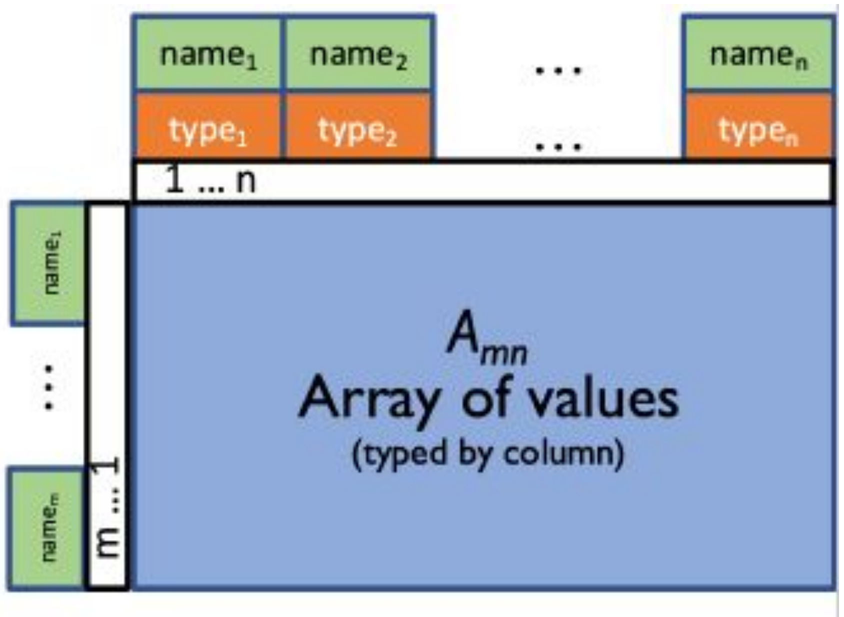
Data Models

- In the data analysis backbone we usually talk about three different and predominant data models:
  - Dataframes: two-dimensional, labeled data structure commonly used in data analysis and manipulation. It is essentially a table with rows and columns, where each column can have different data types.
  - Matrices and tensors: multi-dimensional array used primarily in mathematical computations, particularly in machine learning and deep learning. It is a generalization of scalars (0D), vectors (1D), and matrices (2D) to n-dimensional space.
  - Relational: structured representation of data used in relational databases. It follows a schema and stores data in rows and columns with relationships defined between tables.

Dataframes

No standard definition. They were born blendin ideas from relations and matrices

- Originally defined in the S language, later originating R
- Both, rows and columns, are ordered and named
- Only columns can have types

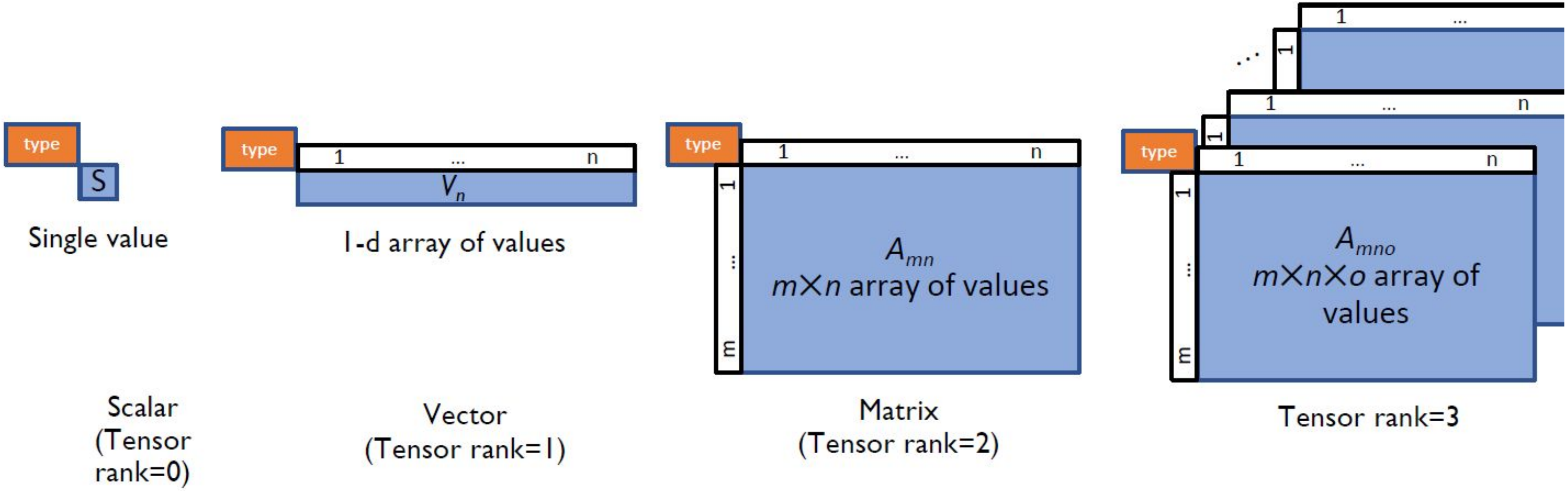


Manipulation language: depends on the tool (Spark, R, etc.)

Popular tools: R, Spark and most traditional ML / DM tools (e.g., SAS)

Extracted from Joe Hellerstein's materials (Berkeley): Three Data models: Relations, Tensors and Dataframes

Matrices and Tensors



Manipulation language: linear algebra

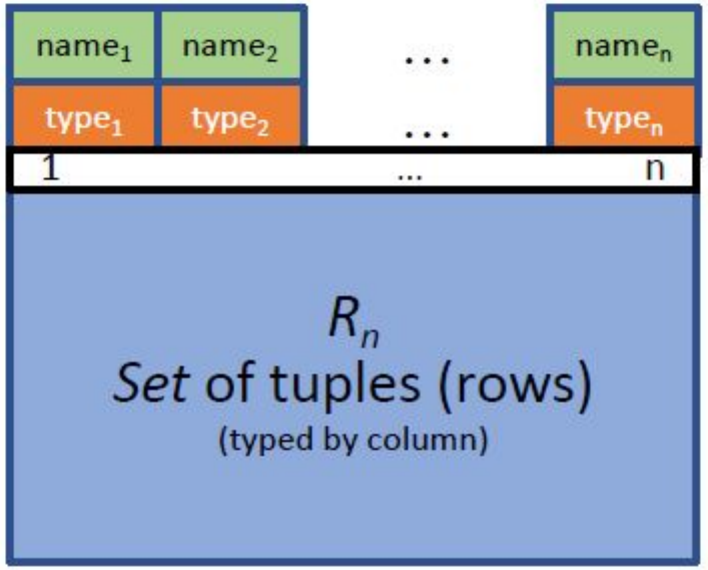
Popular tools: Keras, TensorFlow, and in general Deep Learning tools / frameworks

Extracted from Joe Hellerstein's materials (Berkeley): Three Data models: Relations, Tensors and Dataframes

Relational

Follow the relational model data structure

- Schema of the relation
  - Name of the relation
  - Set of attributes: each attribute has a name and a predefined domain (datatype)
- Range of the relation
  - A tuple is an element of the range meeting the schema (NULL values allowed)
- Integrity constraints:
  - Entity constraint, PKs, FKs, CHECKs, etc.



Relation
(Table)

Manipulation language: relational algebra (SQL)

Extracted from Joe Hellerstein's materials (Berkeley): Three Data models: Relations, Tensors and Dataframes

Comparison

Aspect	Dataframes	Tensors	Relational Tables
Structure	Tabular (rows/columns)	Multi-dimensional array	Tabular (rows/columns)
Data Type	Heterogeneous per column	Homogeneous	Heterogeneous per column
Primary use	Data analysis, ETL	Deep learning	Database storage & querying
Operations	Filtering, joins, groupby	Linear algebra, tensor ops	SQL queries, relational algebra
Persistence	In-memory	In-memory or GPU memory	Disk-based (databases)
Relationships	No explicit relationships	Not relational	Supports relationships (e.g. PK, FK)

Comparison

Aspect	Dataframes	Tensors	Relational Tables
Structure	Tabular (rows/columns)	Multi-dimensional array	Tabular (rows/columns)
Data Type	Heterogeneous per column	Homogeneous	Heterogeneous per column
Primary use	Data analysis, ETL	Deep learning	Database storage & querying
Operations	Filtering, joins, groupby	Linear algebra, tensor ops	SQL queries, relational algebra
Persistence	In-memory	In-memory or GPU memory	Disk-based (databases)
Relationships	No explicit relationships	Not relational	Supports relationships (e.g. PK, FK)

In practice:

- Use dataframes for exploratory data analysis or preprocessing (in-memory).
- Use tensors for advanced numerical computation and Deep Learning.
- Use relational tables to store and query structured data in a database system.

Tools for Data Science

Methodology (Remains the Same)

According to good practices, remember we must differentiate two types of environments: development and operations

- Development: during development, the most popular IDE nowadays are notebooks (e.g., Jupyter, Zeppelin, RStudio). Their dynamicity is key to enable trial and error and fast prototyping (e.g., decide the right system architecture, database schemas, etc.)
- Operations: in operations, notebooks are not used. Instead, continuous integration and deployment tools are used (e.g., Gitlab, Jerkins, etc.), following the good practices set by DevOps. In this backbone, it is relevant to monitor the model to detect **concept drift**

Therefore, realise that in most Data Science projects development (the glue code putting all the previously mentioned software pieces together) is coded in Python (or any other language enabling fast prototyping and coding iterations) via notebooks. However, unlike the data management backbone, operations is typically kept in R/ Python or the specific tool for the analysis (e.g., TensorFlow).

Tools for the Data Analysis Backbone

The data analysis backbone tends to be executed within one tool. There are two (three) main trends: programmatic solutions or off-the-shelf tools.

- Data repositories: There are databases specifically meant to store data meant to be analytically processed. A very popular tool nowadays is DuckDB. Another popular family of databases are the so-called vector databases that are store, natively, multi-dimensional data (such as vector embeddings)
  - DuckDB: SQL-native engine prepared for advanced data analysis
  - Vector databases: Pinecone, Milvus, etc. and many solutions provided by Cloud and AI services.
- Programmatic solutions: R and Python are nowadays the way-to-go programming languages for data analysis given its powerful libraries. In distributed environments, Spark and its MLlib library are the most prominent solution. Additionally, Deep Learning has its own, such as TensorFlow and Pytorch.
- Off-the-shelf tools: these tools provide data repositories and analytical solutions
  - Cloud services nowadays provide powerful tools for data analysis: Microsoft Azure advanced analytics, Google AI, IBM Watsonx, etc.
  - Stand-alone Tools: SAS, Weka, KNIME, etc.

Tools for the Data Analysis Backbone

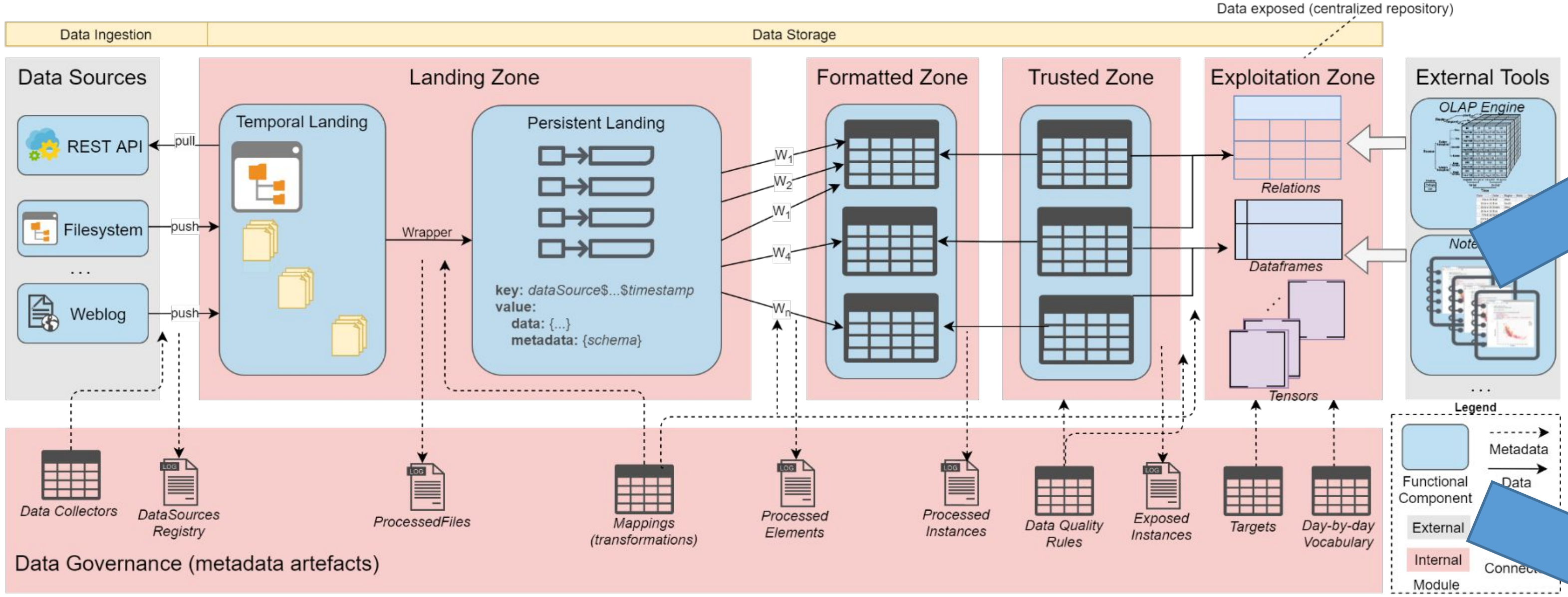
Besides, there are some tasks in these flows that are capitalizing attention and new specific tools are appearing:

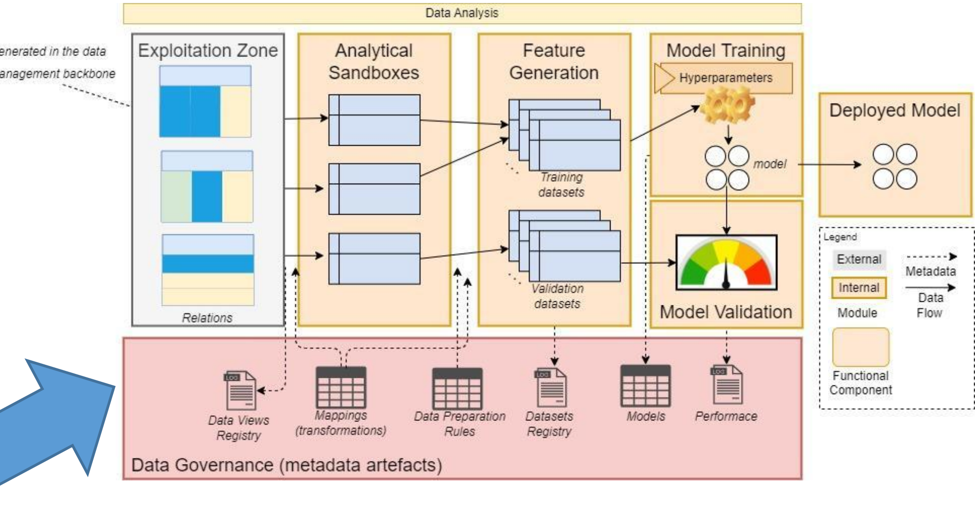
- Data preparation: automating data preparation is a current hot topic. Some of the tools already presented for data quality in the data management backbone propose solutions to this regard: Trifacta, Tamr, Paxata and other such as Tableau Prep. Also, plenty of libraries for R, Python or Deep Learning solutions provide functionalities for data preparation (e.g., encoding)
- Feature Stores: feature stores won a lot of attention in the past years. This are dedicated databases that store and trace features from the data in the sandboxes. Nowadays, these tools have evolved to what we call MLOps later in this list
- Modeling: automating modeling (algorithm selection and hyperparameterization) is also a hot topic and some tools present solutions for this: e.g., AutoML, auto-sklearn, AutoKeras, Auto-WEKA, H2O AutoML
- MLOps: Some tools cover (and that includes governance) the whole data analysis backbone. These are very new tools getting a lot of momentum: e.g., Hopsworks.ai, Databricks (solid connection with Spark), Hugging Face (allows to share models too), etc.

Evolving the Backbones

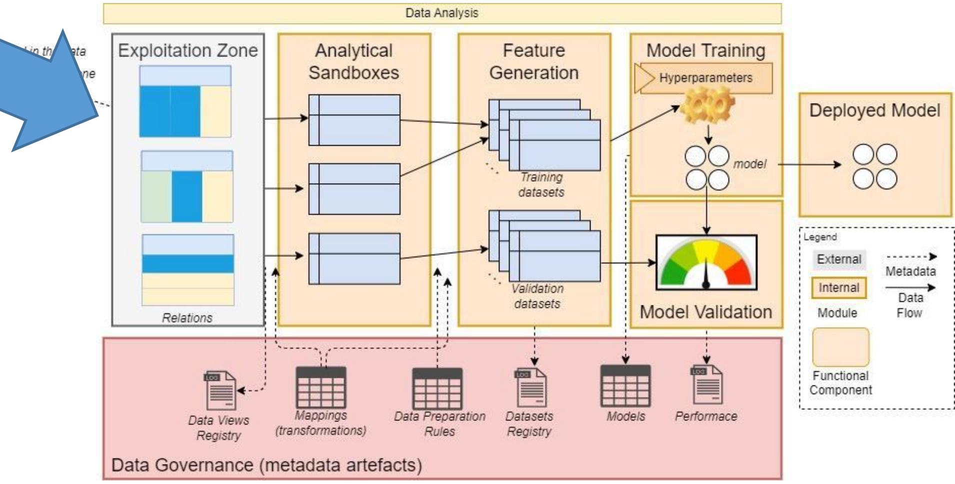
Evolution and maintenance

Evolving the Backbones



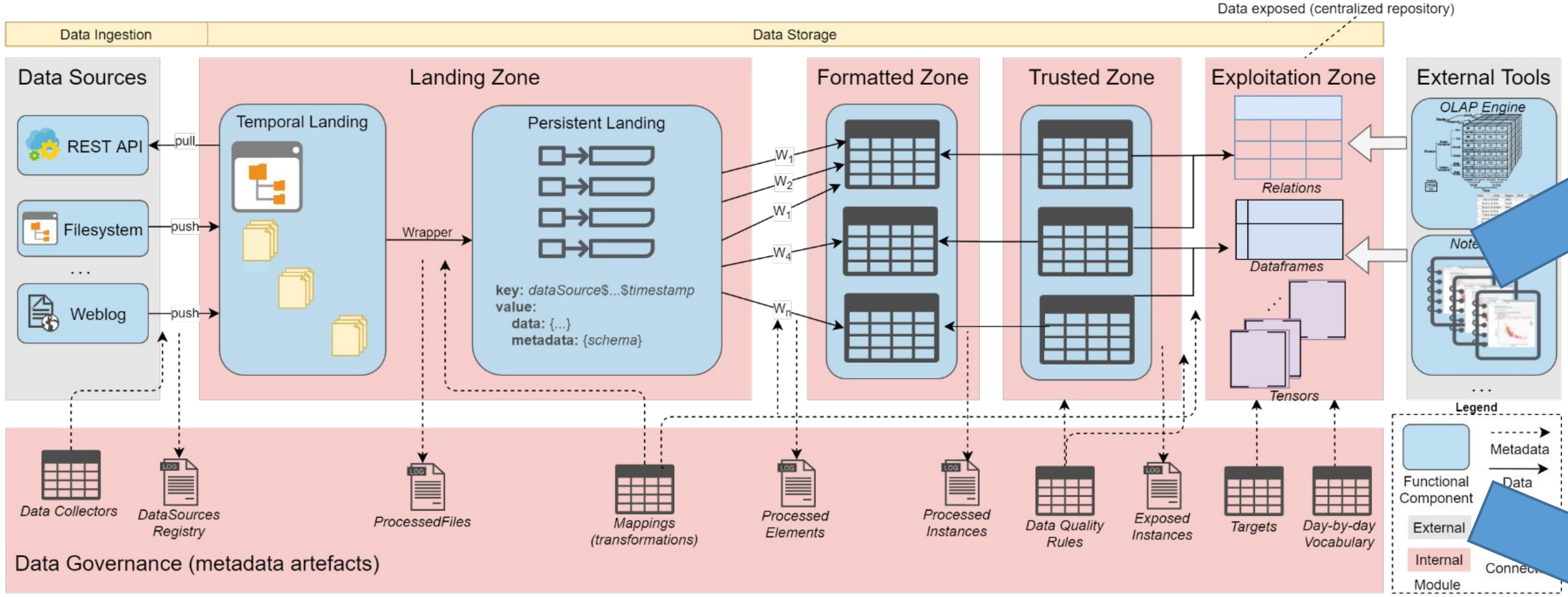


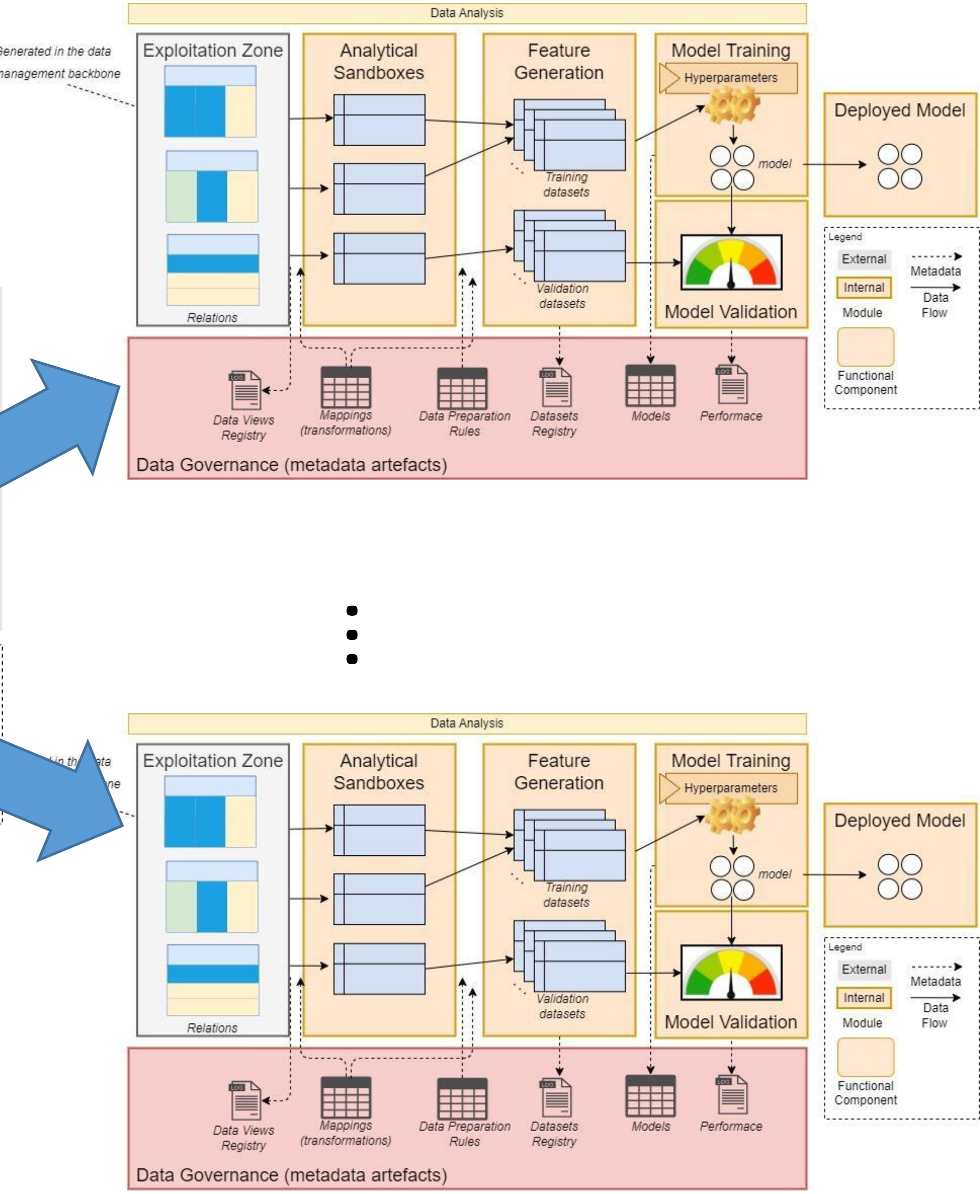
⋮



Evolve the project with a topic of your interest

Evolving the Backbones





- Data Discovery
- Data Augmentation
- Data Quality
- Entity Resolution
- Feature Selection