11. Data Engineering

Background

One of the data science challenge: putting large troves of data into formats that can be **read** by algorithms.

Data engineering is related to processing an ever-increasing supply of data.

After processing, data scientists develop **proofs-of-concept**; AI/ML engineers translate these into **deployable models**.

Data/ML engineering have been around a while (software logs); with the rise of **cloud computing**, some argue that expertise in these fields is becoming more sought after than expertise in data analysis (at least, in some circles).

Data Roles (Reprise)

Data Engineers

- receive data from a source
- structure, distribute, and store data into data lakes and warehouses
- create tools and data models which data scientists can use to query the data

ML Engineers

- apply and deploy data models
- bridge gaps between data engineers and data scientists
- take proof-of-concept ideas to large scale

Data Scientists

- receive data procured/provided by DE
- extract value from the data
- build proof-of-concept predictive models
- measure and improve results
- build analytical models

Data Roles

In smaller organizations, data engineering and data science are typically **blended** into the same role.

Larger companies have **dedicated** data engineers on staff, who build **data pipelines** and manage **data warehouses** (populating them with data and creating table schemas to keep track of the stored data).

In general, DE $\neq$ DS.

Data Pipelines

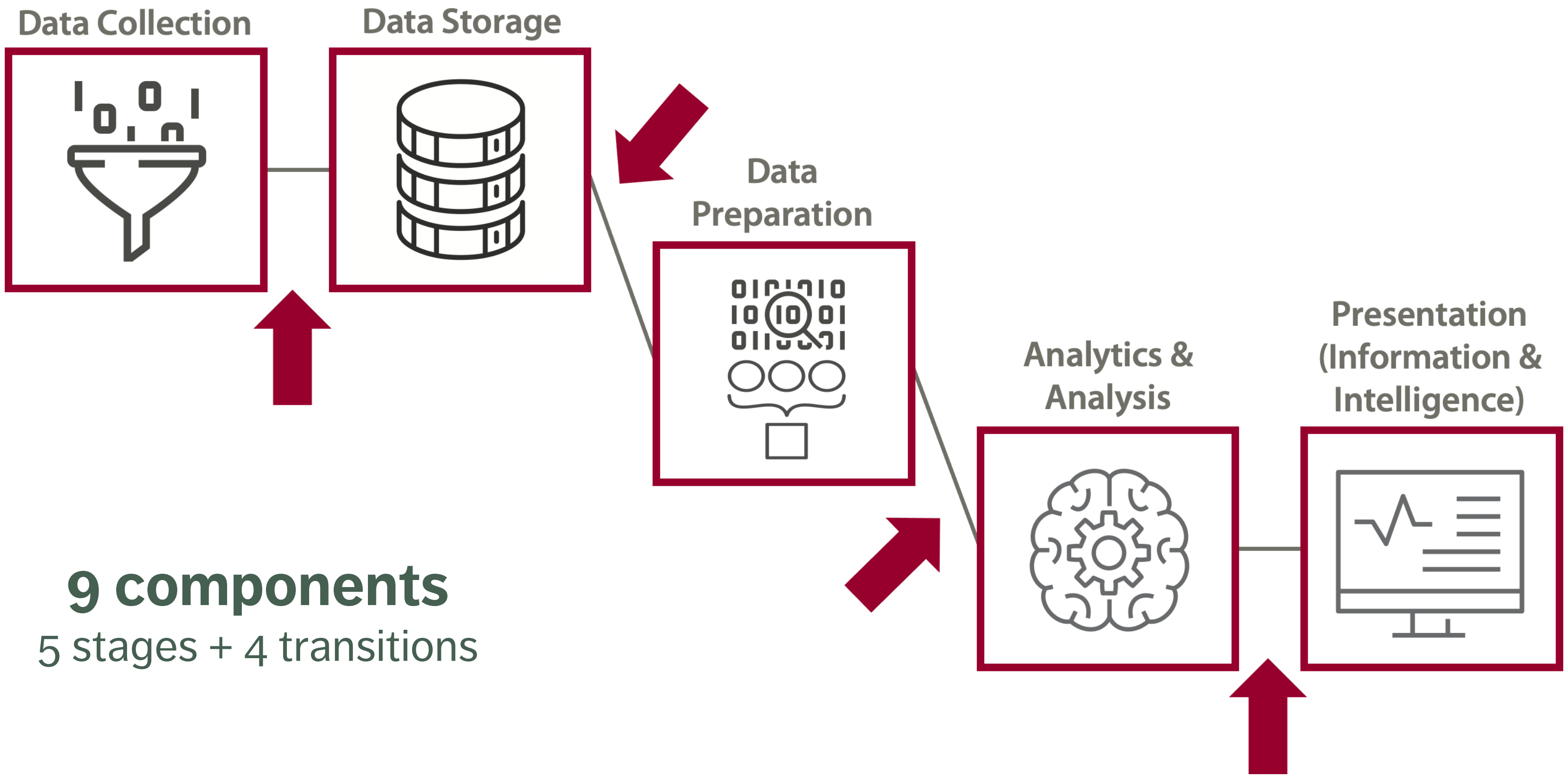
Data engineering

- operations that create **interfaces** and **mechanisms** for the flow and access of information
- setting up **data infrastructure**, preparing it for further analysis by data scientists

Data can arise from many **sources** (and types of sources), and in a variety of formats and size.

Transforming this into a process that data scientists can use and from which they can derive meaning is known as **building a data pipeline**.

Data Pipelines



9 components

5 stages + 4 transitions

Data Pipelines

Main data engineering challenge:

- building a pipeline that **runs in (close to) real-time whenever it is requested**
- so that users get **up-to-date information** from the source with **minimal delays**

Working pipeline proof-of-concept solutions are passed on to ML engineers for **deployment** and **production**. Some of the work surrounding this includes:

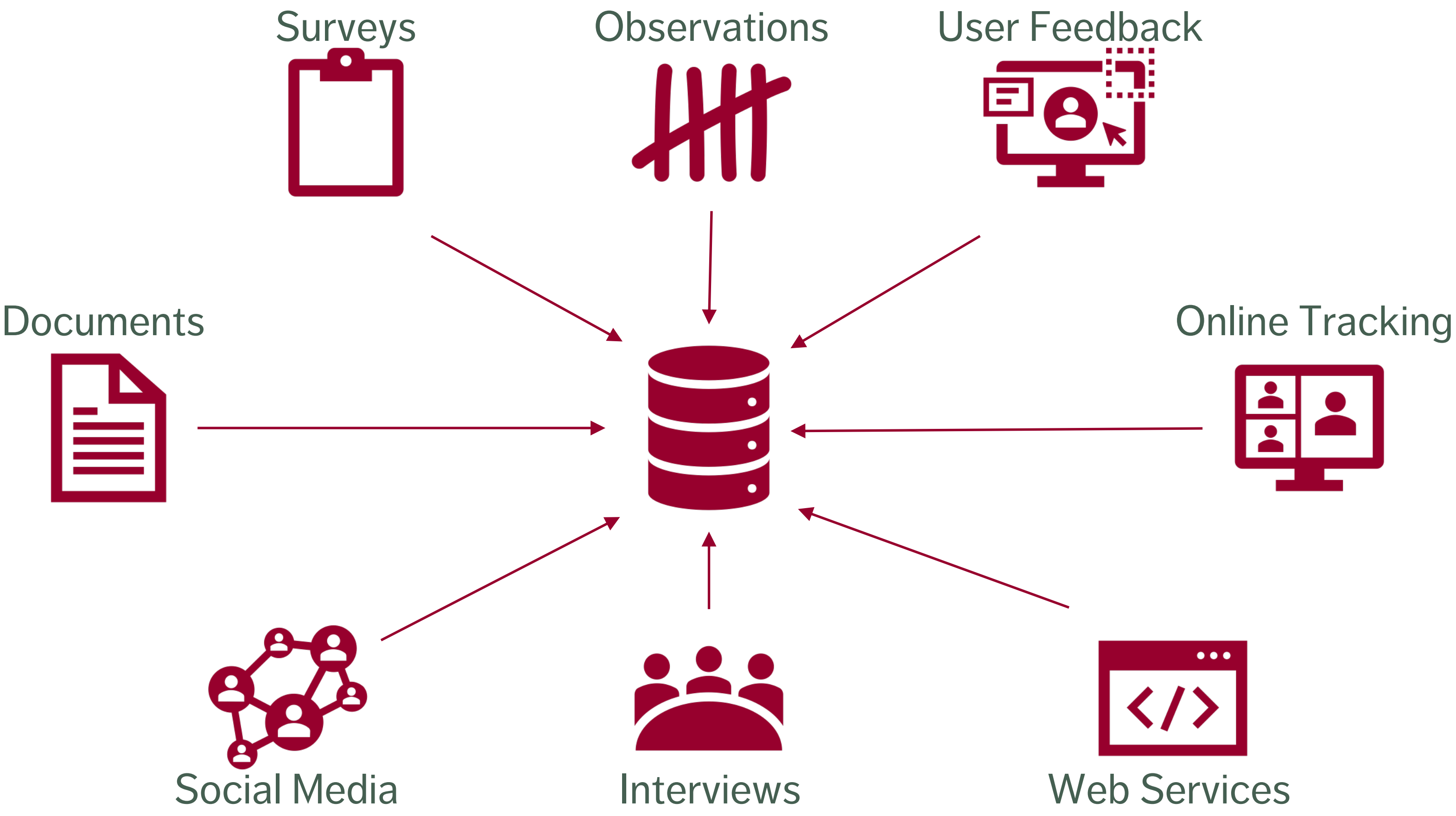
- data quality checks
- optimizing query performance
- creating a continuous integration/continuous delivery ecosystem around model changes
- ingesting data from various sources into the data model
- carrying machine learning and data science techniques to distributed systems.

Data Pipelines

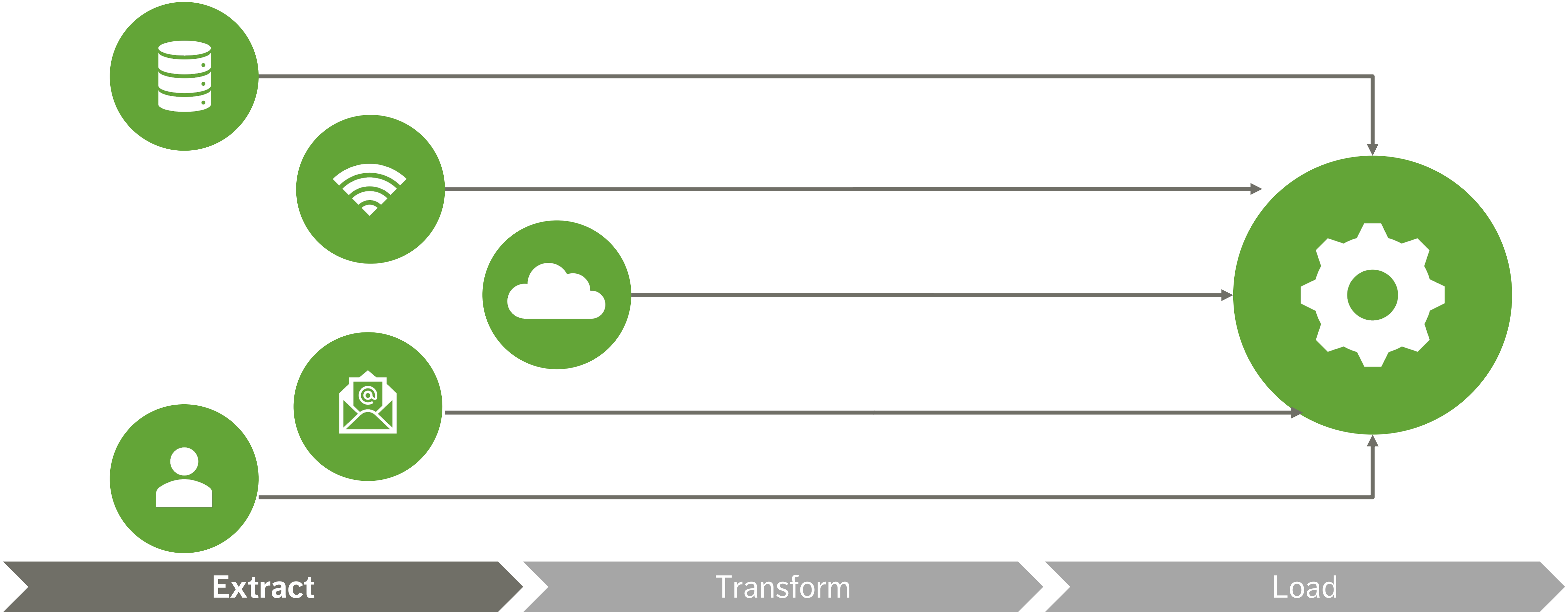
Common **themes** (operations/framework/tasks/sources) for pipeline steps :

- **data collection:** applications, mobile apps, microservices, Internet of Things (IoT) devices, websites, instrumentation, logging, sensors, external data, user generated content, etc.
- **data storage:** Master Data Management (MDM), warehouse, data lake, etc.
- **data integration/preparation:** ETL, stream data integration, etc.
- **data analysis:** machine learning, predictive analytics, A/B testing, experiments, artificial intelligence (AI), deep learning, etc.
- **delivery and presentations:** dashboards, reports, microservices, push notifications, email, SMS, etc.

Data Collection



ETL – Extract



ETL – Transform

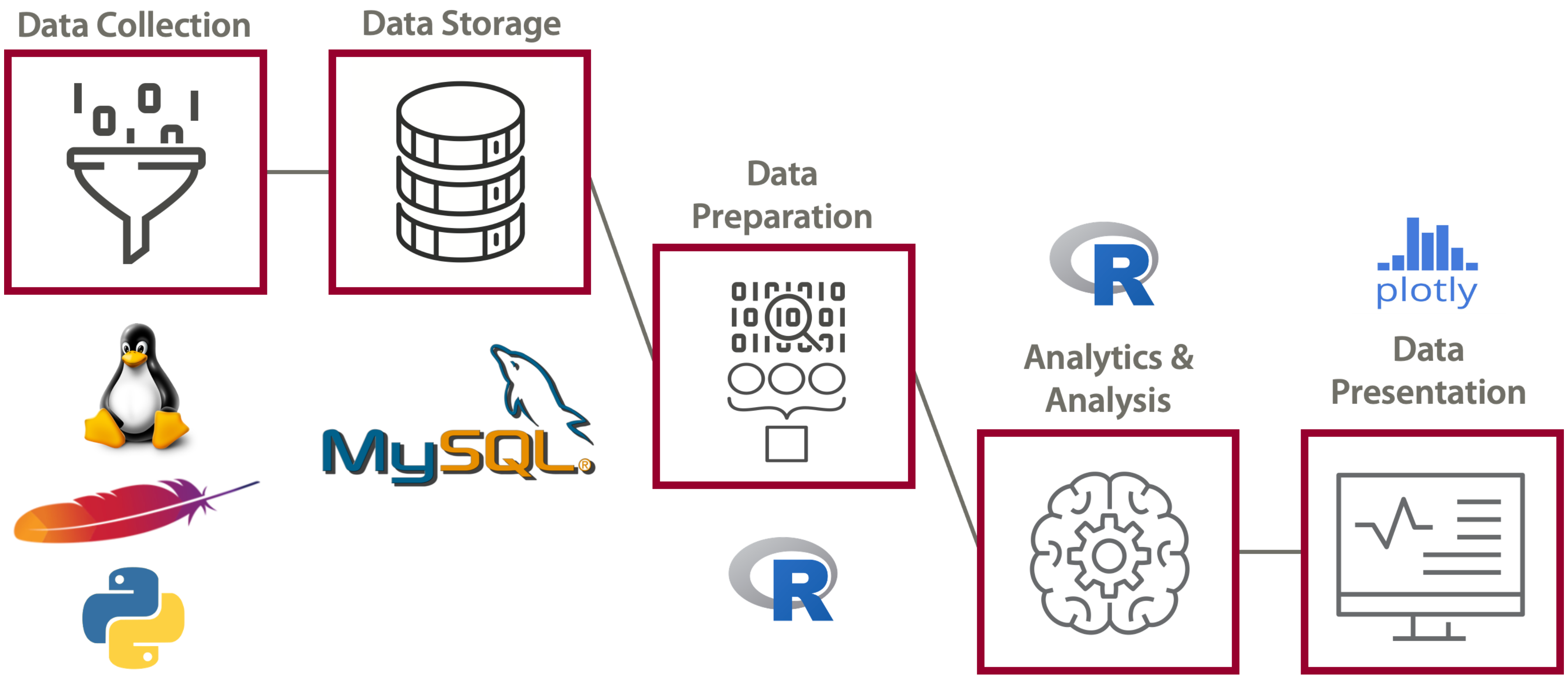
- Changing Structure
- Altering Data Types
- Aggregating Data
- Cleaning
- Joining
- Grouping

Extract → **Transform** → Load

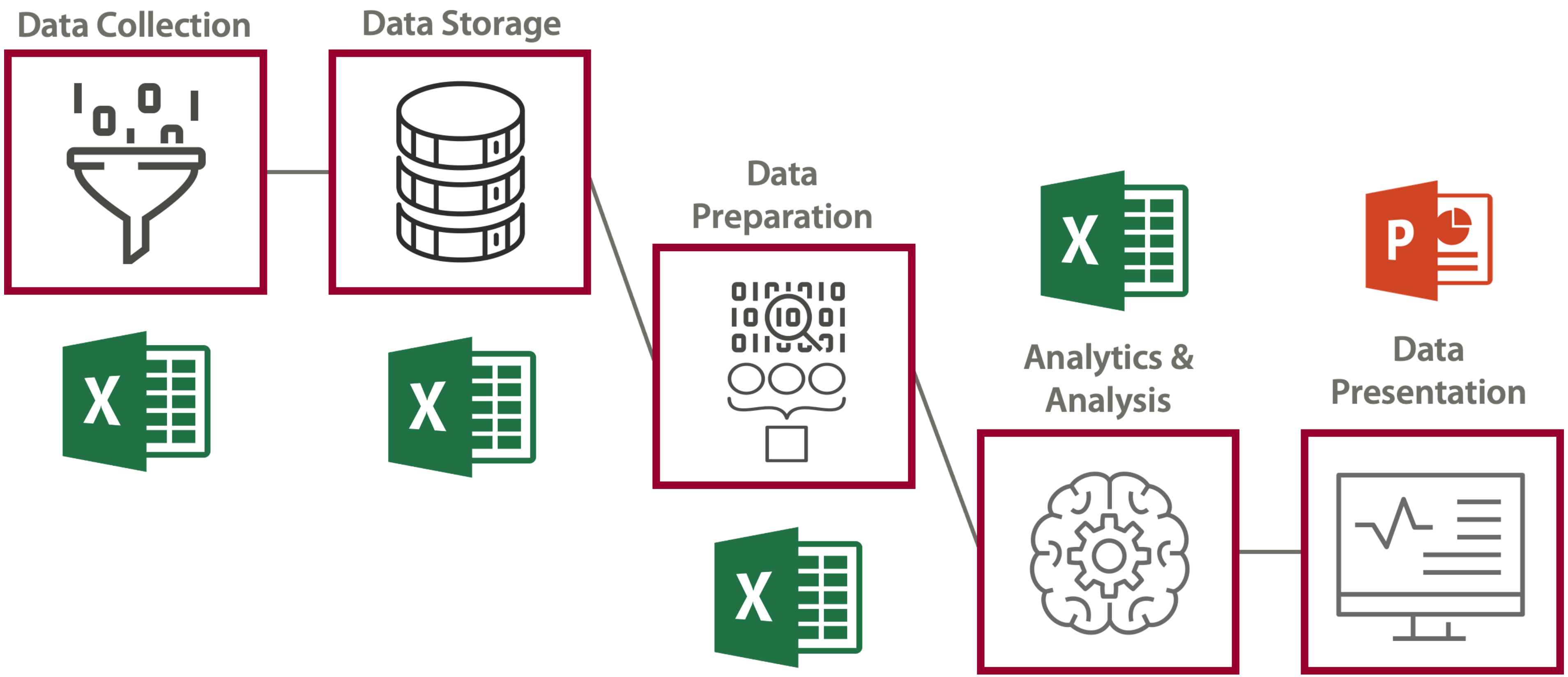
ETL – Load

Extract → Transform → **Load**

Data Pipeline: Open Source



Data Pipeline: GoC (?)



Data Pipeline Tools

Pipelines let users split large tasks into a series of smaller sequential steps, which can help **optimize** each step.

If using TensorFlow for the analysis component of a DL pipeline which consists of a single large script, then **everything** from data collection to presentation has to be done with TensorFlow; may not be optimal.

Data pipeline tools select the best framework/language for each pipeline component/task:

- Luigi (Spotify)
- Airflow (AirBnB)
- scikit-learn
- pandas/tidyverse
- etc.

Data Engineering Tools

It is unlikely that one data engineer could achieve mastery over all possible data engineering tools, but teams might get a lot of **coverage:**

- **analytical databases** (Big Query, Redshift, Synapse, etc.)
- **ETL** (Spark, Databricks, DataFlow, DataPrep, etc.)
- **scalable compute engines** (GKE, AKS, EC2, DataProc, etc.)
- **process orchestration** (AirFlow/Cloud Composer, Bat, Azure Data Factory, etc.)
- **platform deployment and scaling** (Terraform, custom tools, etc.)
- **visualization tools** (Power BI, Tableau, Google Data Studio, D3.js, ggplot2, etc.)
- **programming** (tidyverse, numpy, pandas, matplotlib, scikit-learn, scipy, Spark, Scala, Java, SQL, T-SQL, H-SQL, PL/SQL, etc.)

What is Data Governance?

Data governance encompasses:

- **people**
- **processes**
- **information technology**

It is required to create a **consistent** and **proper** handling of an organization's data across the enterprise.

It provides the foundation, strategy, and structure to ensure that data is managed as an **asset** and transformed into **meaningful** information.

Data Governance

Goals:

- create self-service data culture
- establish internal rules for data use
- implement compliance requirements
- improve internal and external comms
- increase value of data
- reduce costs
- continually manage risks
- ensure continued existence

Suggested Reading

Data Engineering

Data Understanding, Data Analysis, Data Science
Data Engineering and Management

Background and Context

Data Engineering

- Data Pipelines
- Automatic Deployment and Operations
- Scheduled Pipelines and Workflows
- Data Engineering Tools

Exercises

Data Engineering

1. What does your (or your organization's) data science pipeline look like? Could it be improved?
2. Identify instances where you have had issues due to data availability, usability, consistency, integrity, quality, security, or trustworthiness.
3. Complete any of the previous exercises you have not had the chance to finish.