An introduction to data engineering

Mingjun Zhong

Department of Computing Science

University of Aberdeen

Content

- What is data engineering?
- Concepts in data engineering

What is data engineering?

- **Data engineering (**Wikipedia**)** refers to ***the building of systems*** to enable the collection and usage of data. This data is usually used to enable ***subsequent analysis*** and ***data science***, which often involves machine learning. Making the data usable usually involves substantial compute and storage, as well as data processing.

What is data engineer?

- **Data Engineer** (Wikipedia): A data engineer is a type of ***software engineer*** who creates big data ETL (Extract, Transform, and Load) pipelines to manage the flow of data through the organisation.

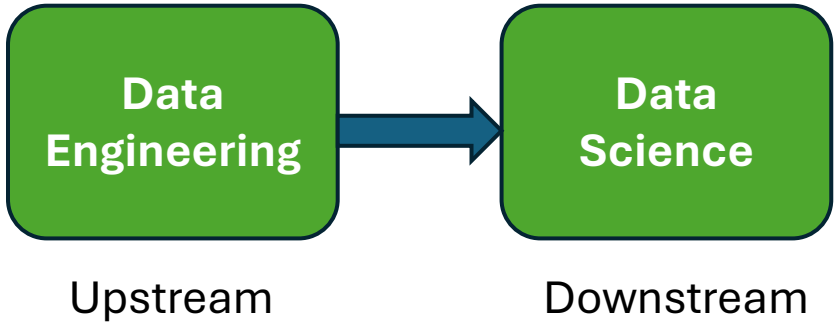
- Take huge amounts of data and translate it into insights.
- Focusing on the production readiness of data and things like formats, resilience, scaling, and security.
- Hail from a software engineering background and are proficient in programming languages like Java, Python, Scala, and Rust.
- Familiar with databases, architecture, cloud computing, and Agile software development.

What is data science?

- **Data science** (Wikipedia): is an interdisciplinary academic field that uses statistics, scientific computing, scientific methods, processing, scientific visualization, algorithms and systems to extract or extrapolate knowledge from potentially noisy, structured, or unstructured data.
- **Data scientists** (Wikipedia): are more focused on the analysis of the data, they will be more familiar with mathematics, algorithms, statistics, and machine learning.

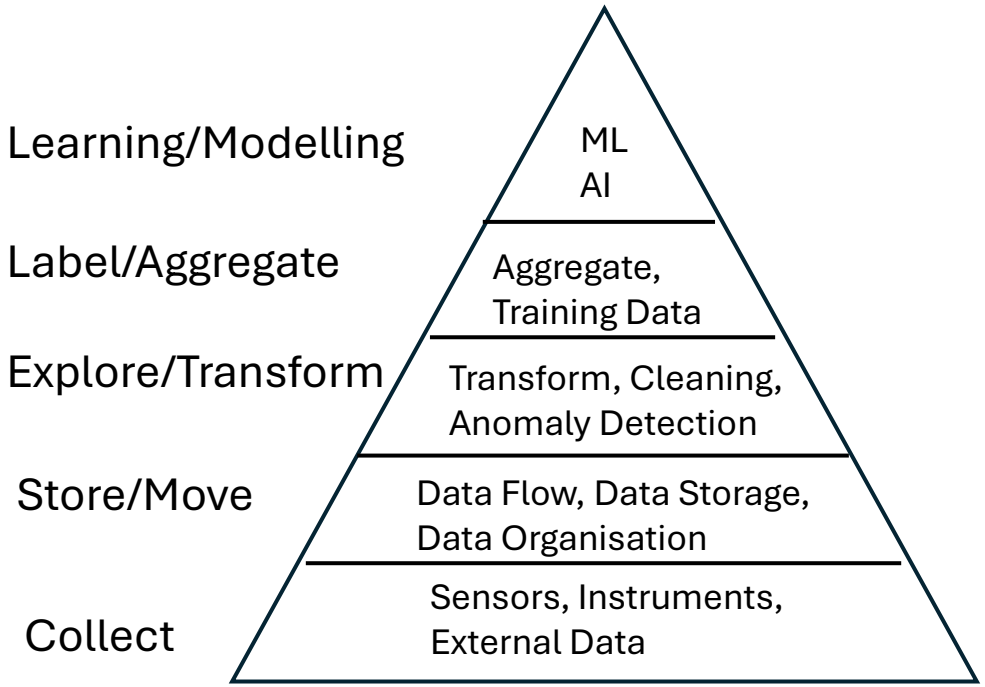
Data engineering vs. data science

- Data engineering: prepare the raw data to use
- Data science: extract knowledge from the prepared data
- Data scientists to build ML/AI models:
  - 70%-80% time doing DE – gathering, cleaning, and preprocessing data
  - 20%-30% time on ML and analysis



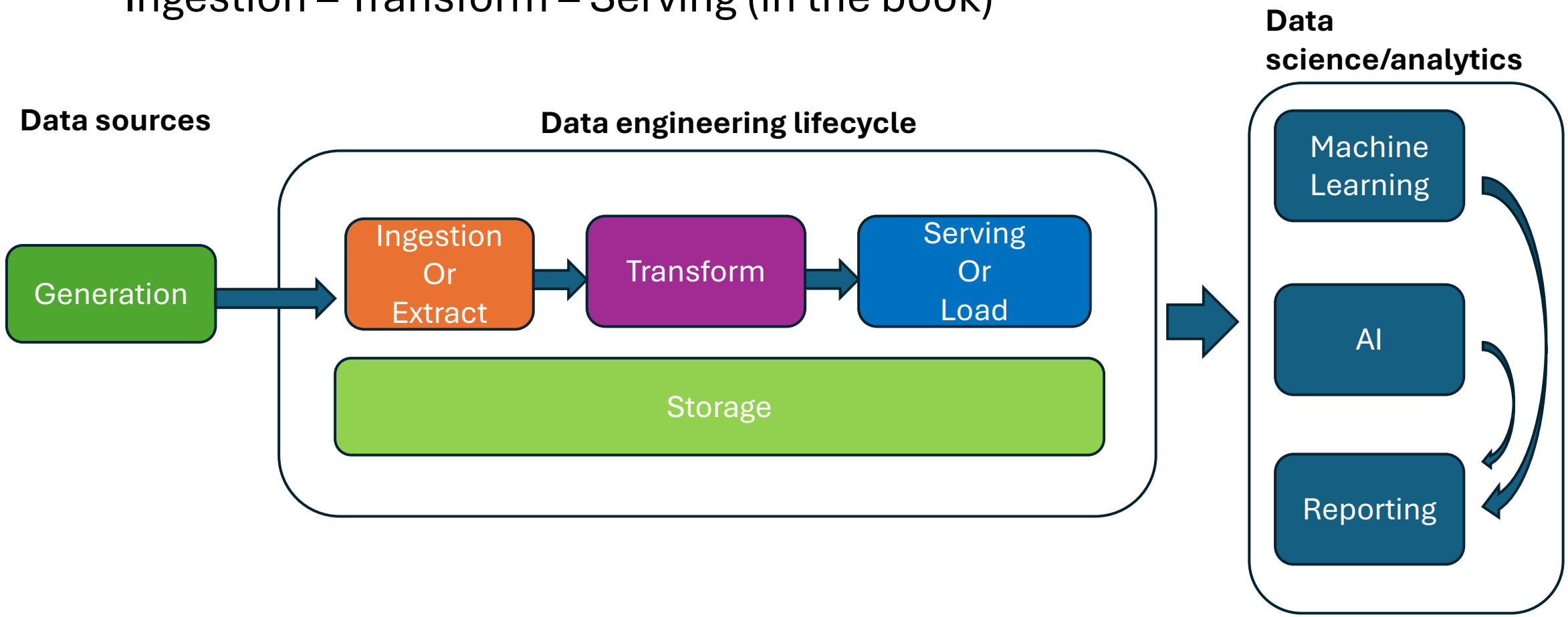
The data science hierarchy of needs

- Progression from raw data collection at bottom to AI/ML at top
- Upper levels are difficult or impossible without foundation of data collection and engineering
- Organizations may not understand the hierarchy and hire data scientists
- As a result, highly qualified data scientists, who are often familiar with AI/ML, find themselves spending most of the time on data engineering



Data engineering lifecycle

- **Data engineering lifecycle**: Extract – Transform – Load or Ingestion – Transform – Serving (in the book)



Adapted from the book: **Fundamentals of Data Engineering: Plan and Build Robust Data Systems, by Joe Reis & Matt Housley, 2022 O'Reilly**

Skills needed in data engineering

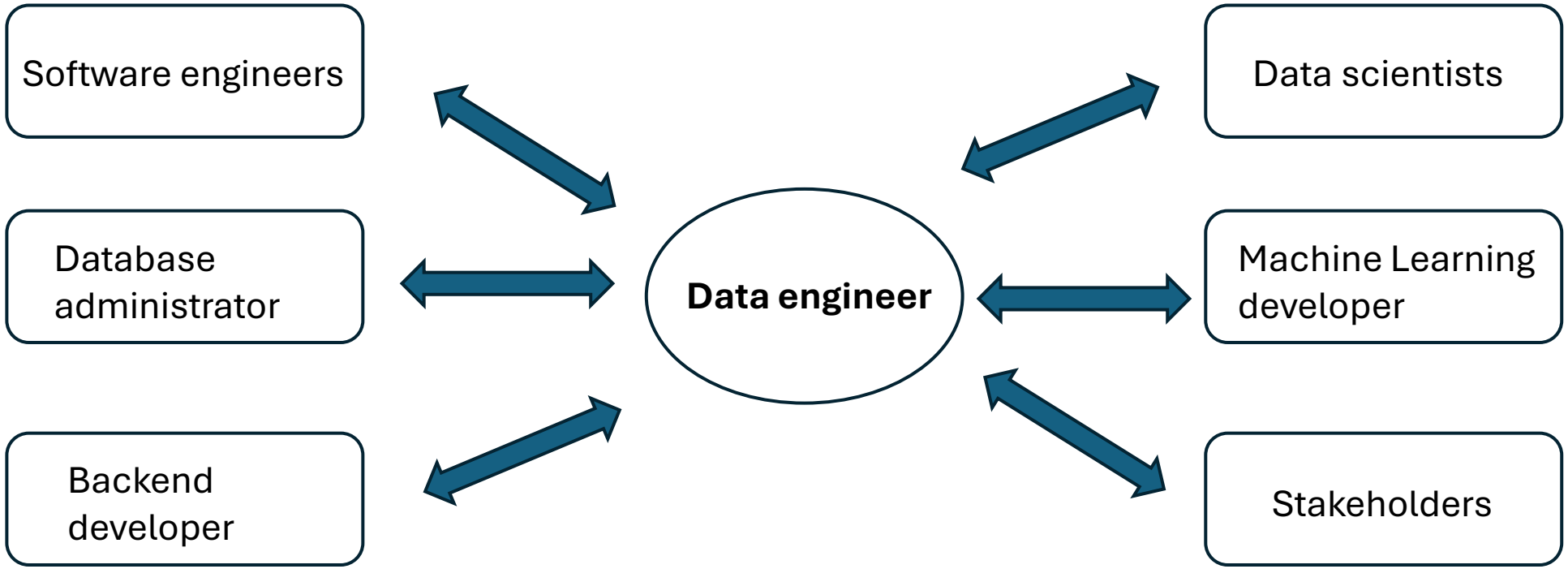
- Data management, data operations (DataOps), data architecture, data security, software engineering, business requirements
- Know how data is produced, how data scientists will consume the data
- Know how to use powerful tools: Apache Airflow, TensorFlow's TFX, Scikit-learn, Apache Kafka (distributed system), etc
- Utilizing these techs requires software engineering, distributed systems, databases, etc
- Many tools make DE easier to hand on: data engineering pipeline – Airflow + TFX, scikit-learn pipelines, Kafka distributed systems
- Any programming languages: bash, SQL, Python, Java, R, ...

Business responsibilities

Successful data engineers need to understand the big picture and the value of the business.

- Know how to communicate with nontechnical and technical people – comm. is key to DE
- Understand how to scope and gather business and product requirements – understand stakeholders' needs, business impact
- Understand the cultural foundations of Agile, DevOps, and DataOps – needs to combine with software engineering methods
- Control costs – business like low costs, high value of data
- Learn continuously – DE is fast development, new DE pipelines emerge

The role of data engineer in an organisation



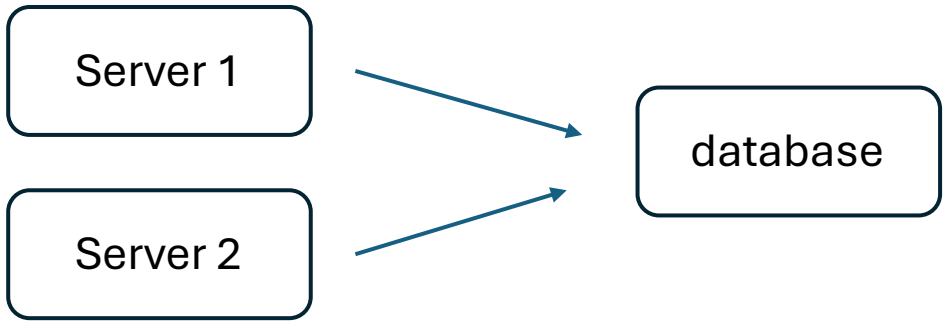
Data engineering lifecycle

The data engineering lifecycle is to turn raw data into a useful end data product with specific data structures. Five components may be needed:

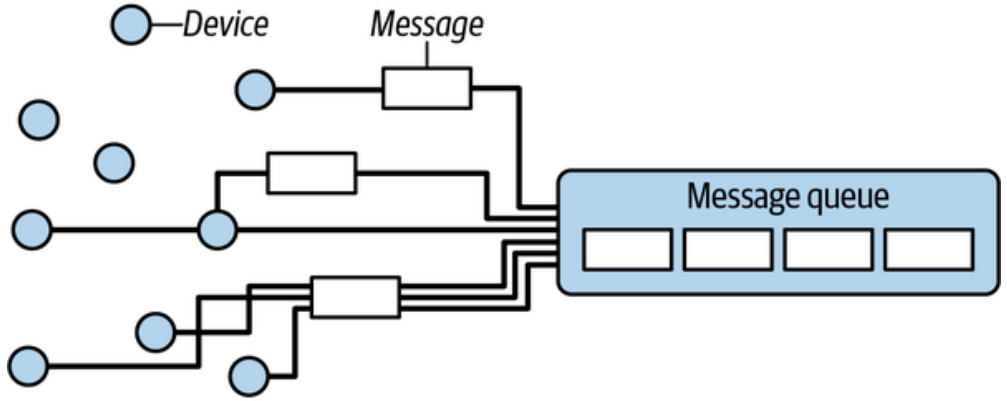
- Generation
- Storage
- Extract/Ingestion
- Transform
- Load/Serve

Generation – source of data

- Data source to be understood
  - IoT devices – home temperature, smart homes
  - Message system – email, X, Reddit, etc.
  - Transactional databases – data in databases
- To understand how data was generated, how the data source system is working
  - Frequency, velocity, volume
  - Types of data: numeric, categorical, text
  - The data structure: structured (tables), unstructured (text)
- To proceed DE pipelines



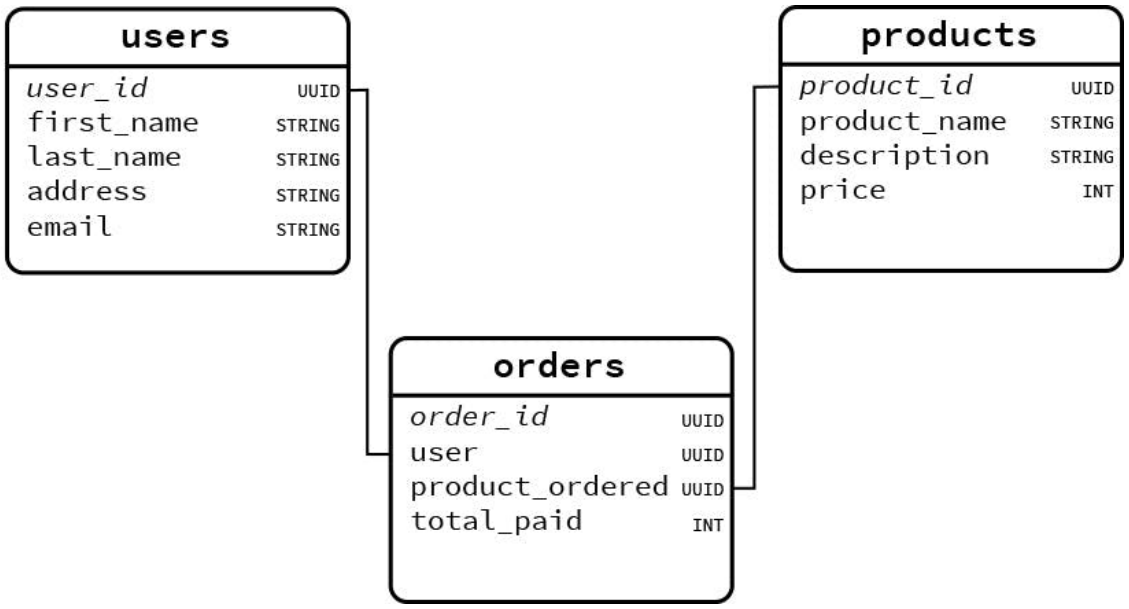
Application database



An IoT system and message system

Generation – source of data

- **Schema** of the data:
  - The blueprint or structure defining how data is organised and stored
  - The relationships between data elements
  - Type of data, constraints, rules
- Absence of schema: need to define the schema
- Schema may change over time



Storage

- Storage happens during the ETL pipeline – can be at every stage of the pipeline
- Storage system for various data sytems: data warehouse, data lakehouse, database, objects:
  - Storage compatible with DE pipelines?
  - Read and write speeds?
  - Compatible with downstream tasks including speeds?
  - Scalability: Can it satisfy future scale?
  - Pure storage or it supports complex queries?
  - How to handle regulatory compliance and data security?

Extract data or Ingestion

- After you understand data source, you will extract the data from sources
- Ingestion may be the most significant bottlenecks of the DE lifecycle:
  - The data source system is out of your control
  - May be unresponsive
  - Poor quality data provided
  - Data flow may be interrupted

Extract data or Ingestion

- Questions you need to ask:
  - What are the use case for the data?
  - Is data available when I need it?
  - What is the data destination after ingestion?
  - How often will I need to access the data?
  - In what volume will the data arrive?
  - What is the format of the data?
  - What is the schema of the data?

Transform

- After the data is ingested or extracted, you need to transform the data into the required format or structure
- Transformation is driven by downstream tasks or generic use
- Consider:
  - What format is needed – for ML tasks, or generic use?
  - What transformations are needed: missing values, normalisation, etc?
  - What business requirements does the transformation support?
  - Can I minimise data movement between transformation and storage during transformation?
  - Shall the raw data be kept?
  - Do you need to do preprocessing of the data for ML?
    - Preprocessing is different to DE: a type of feature engineering for ML modelling

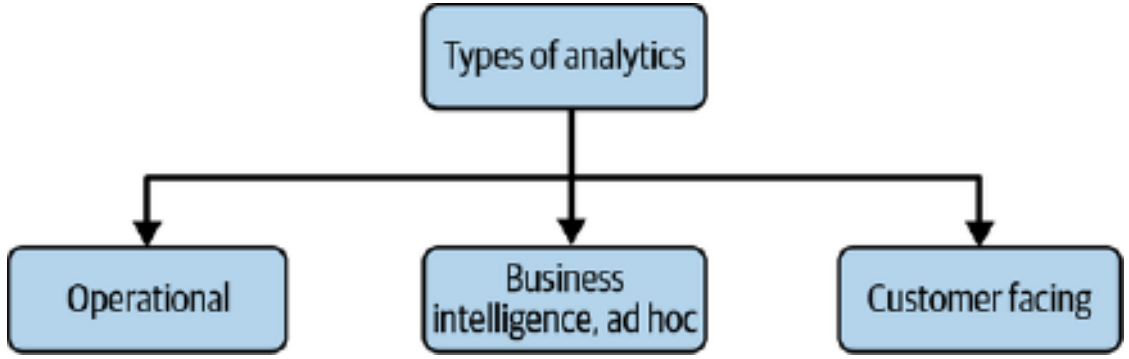
Serving/loading data

- After the data is extracted and transformed, it requires to get value from your data for downstream tasks
- Various types of use of data:
  - Machine Learning
  - Data Analytics
  - Reverse ETL

Machine learning

- ML is the core technology of Artificial Intelligence
- Knowledge extracted using ML – industrial applications
- Data engineer may help to train ML models
- DE may be combined with ML engineering – e.g., feature engineering, visualisation, etc
- Data engineer should be familiar with ML – helps to engineering data

Analytics



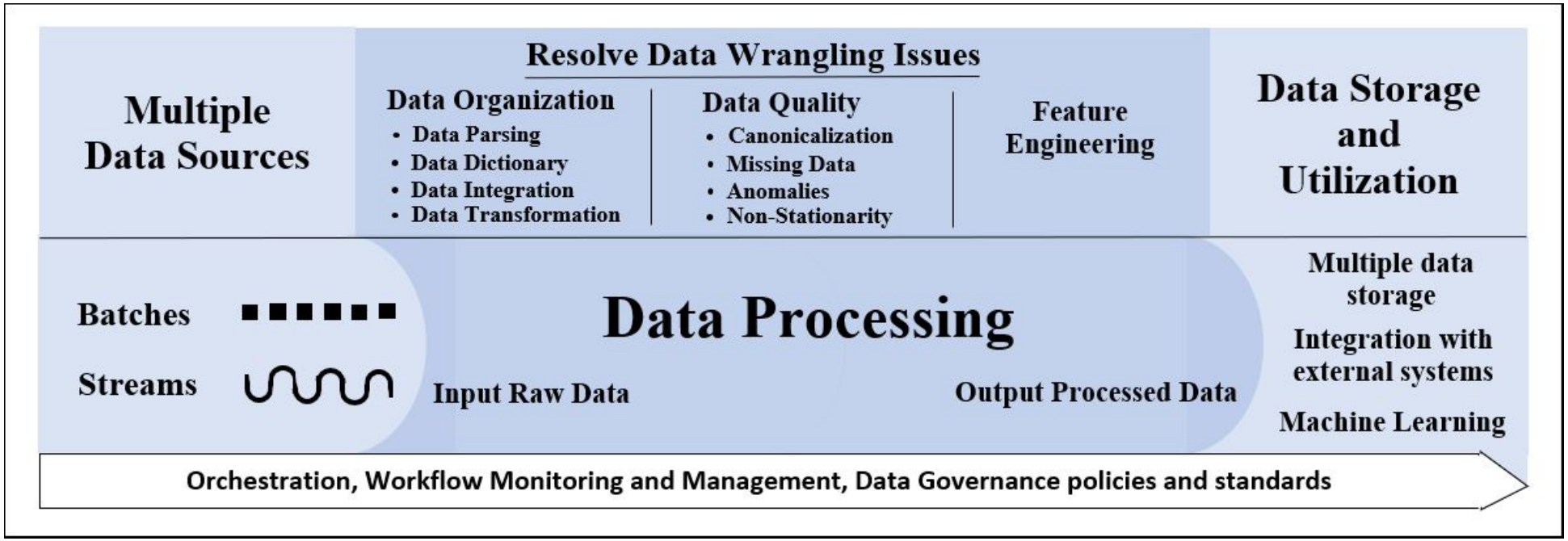
- Generate reports, dashboards, visualisations
- Types of analytics:
  - Customer-facing analytics
    - ✓ Accessing data control – requests from customers
    - ✓ Thousands of customers
    - ✓ Security
  - Operational analytics
    - ✓ Data consumed in teal time
    - ✓ Focus on fine-grained details of operation, actions immediately on requirements
  - Business intelligence
    - ✓ Maintains business logic and definitions
    - ✓ Extract important information to inform business value
    - ✓ Move from ad hoc data analysis to self-service analytics

Reverse ETL

- Reverse ETL takes processed data from output of DE and feeds it back into source systems
- Marketing and sales need real-time data for their daily activities
- Can be viewed as an operational analytics

Data engineering pipeline tools

- Data engineering pipeline: The combination of some steps and processes from the data source to the destination where the data is utilised.



An ideal pipeline for data engineering

Classification of data engineering pipelines

Data pipeline:

- Encapsulates all the tasks for data processing
- Tasks include reading in data, transforming the data, output data

Pipelines can be classified into the following clusters depending on their focus:

- ETL data pipelines
- Integration, Ingestion, and Transformation
- Pipeline orchestration & workflow management
- Machine learning and model deployment

ETL data pipelines

- Designed to perform an aspect of data integration through the extraction of data from sources
- Apache Spark:
  - Open-source, multiple languages – Python, Java, SQL, Scala, and R
  - Handling large data
- AWS Glue:
  - a serverless data integration service that simplifies the process of discovering, preparing, and integrating data from various sources for analytics, machine learning, and application development.
  - It offers a fully managed ETL (Extract, Transform, Load) service that helps automate data preparation and loading for analytical tasks.

Integration, Ingestion, and Transformation

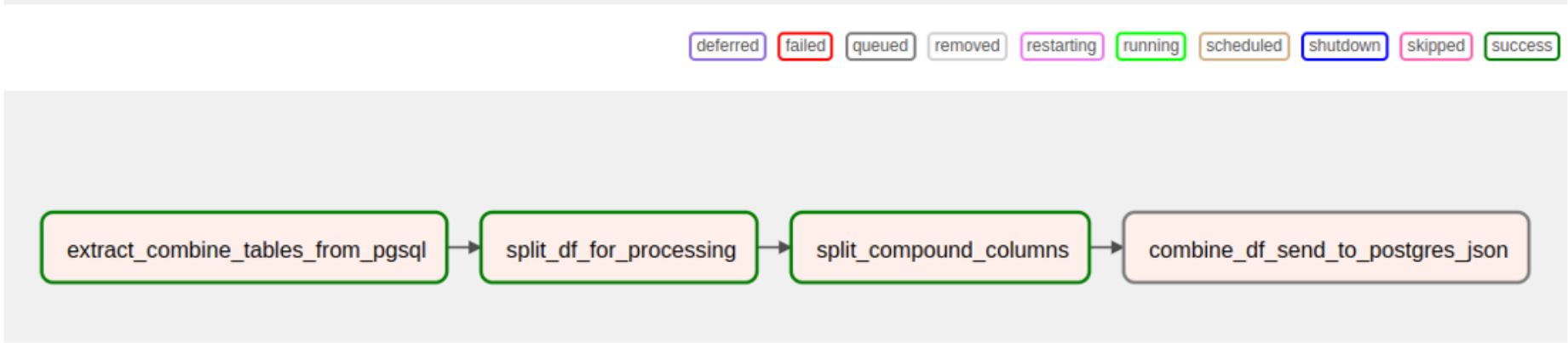
Aims to provide solutions to the data engineering issues associated with data organisation for example, working with data from multiple sources, heterogeneous data, integrating aspects of data to suite specific requirements

- Apache Kafka
  - Ingest multiple data sources and transform data during processing
  - A distributed system: producer – consumer pattern
  - Integrate with other tools, e.g., Apache Airflow

Pipeline orchestration & workflow management

- Are designed to centralise the automation, execution, monitoring, and management of workflow through the entire data pipelines from data sourcing through to utilization.
- These pipelines ensure that other interconnected pipelines or phases of a pipeline are executed in an orderly fashion to process data end-to-end.
- Apache Airflow:
  - is an open-source web-interfaced data pipeline orchestration and workflow management tool for ETL and data integration
  - Python language
  - Perform automated workflow orchestration, scheduling, and monitoring of processes
  - Monitoring processes with graphs
  - Can work with Apache Kafka to provide streaming data processing

Airflow example

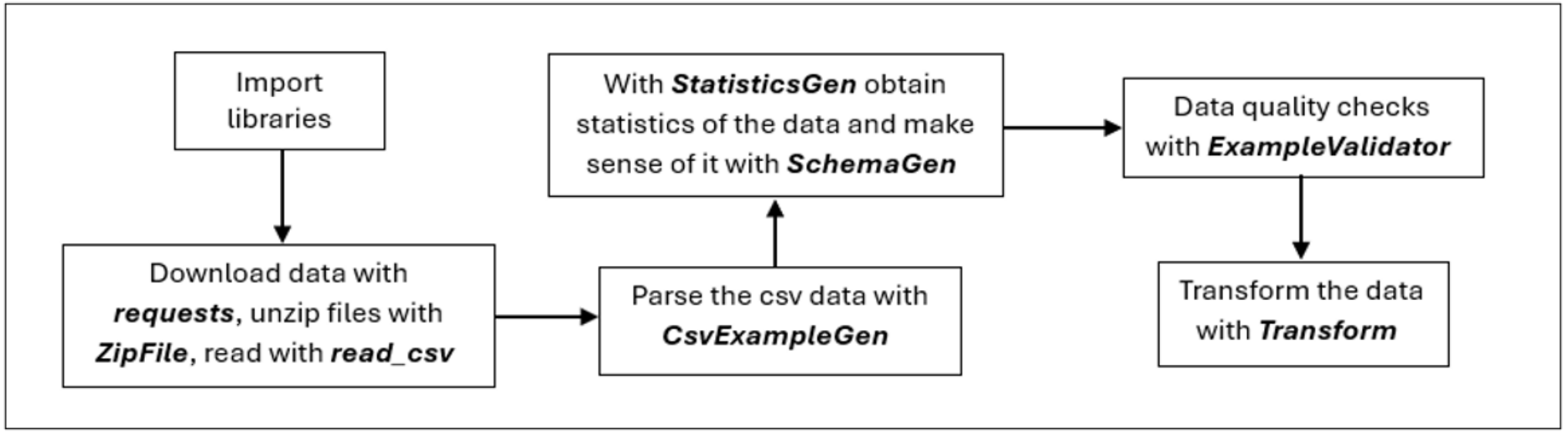


Machine learning and model development

- The downstream of data engineering pipelines
- TensorFlow Extended (TFX):
  - Extended from TensorFlow
  - A basis for building automatic machine learning following DE pipelines
  - Works with Apache Airflow and Kubeflow to provide orchestration functions.
- AutoKeras:
  - Automatic ML framework based on Keras and TensorFlow
  - Raw data processing, ML model building, deployment
  - Useful for deep learning

- Scikit-learn pipeline:
  - Library of transformations
  - Clean, reduce, expand, and/or generate feature reps.
  - Purpose for ML modelling

TFX example



References

Materials were taken from the following references:

- Fundamentals of Data Engineering: Plan and Build Robust Data Systems, by Joe Reis & Matt Housley, 2022 O'Reilly
- Mbata, Anthony, Yaji Sripada, and Mingjun Zhong. "A survey of pipeline tools for data engineering." *arXiv preprint arXiv:2406.08335* (2024).
- Nazabal, Alfredo, Christopher KI Williams, Giovanni Colavizza, Camila Rangel Smith, and Angus Williams. "Data engineering for data analytics: A classification of the issues, and case studies." *arXiv preprint arXiv:2004.12929* (2020).