IMEN891K / IDSC501 / AXGS-501 · DATA ENGINEERING · FALL 2026

1.

Overview & Logistics

Sangwoo Mo · POSTECH

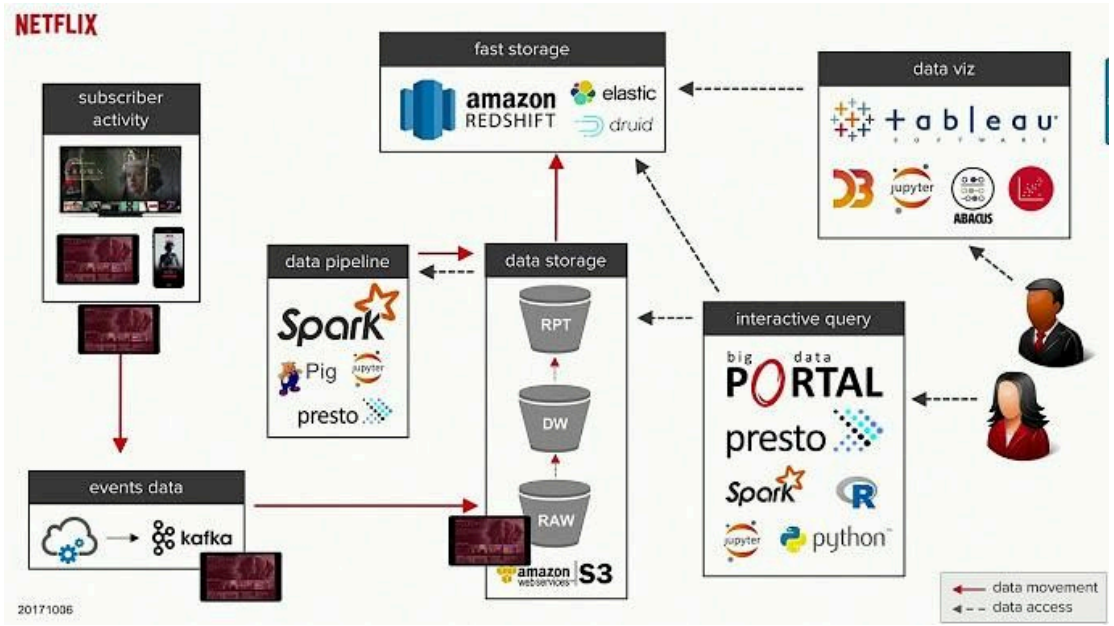
Industrial & Management Engineering

What is Data Engineering?

Before the definition: how two companies actually run on data

Case study: Netflix

One path through the platform



Follow the data

Events enter the pipeline

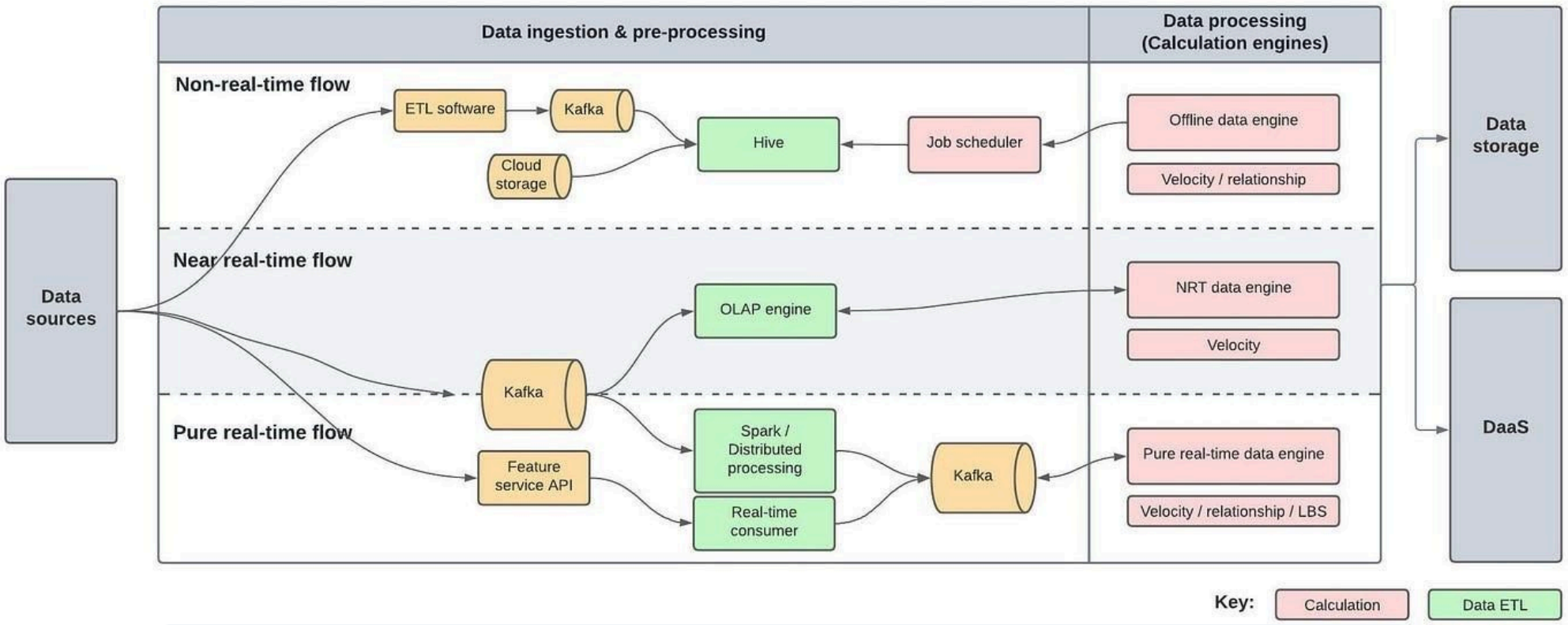
Storage keeps records available

Queries and dashboards read the prepared data

Follow events through processing and storage to the queries that use them

Case study: Coupang Eats

coupang eats



Different latency needs get **non-real-time**, **near-real-time**, and **pure real-time** flows

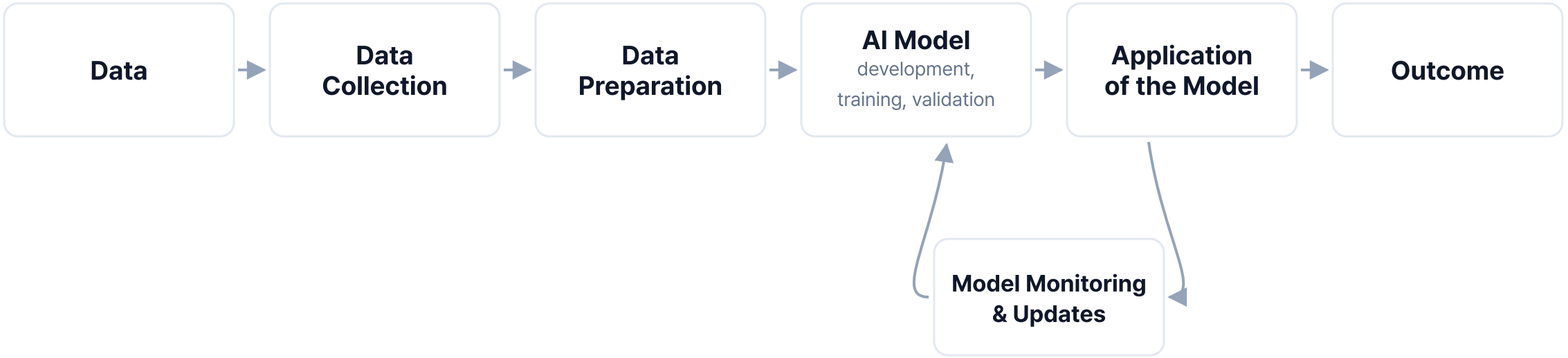
Outline: Overview and Logistics

01	**The Pipeline**	follow records from source to outcome
02	**Course Scope**	collection, storage and monitoring
03	**Course Logistics**	evaluation, project and AI policy

01 · The Pipeline

01	**The Pipeline**	follow records from source to outcome
02	**Course Scope**	collection, storage and monitoring
03	**Course Logistics**	evaluation, project and AI policy

Data Engineering Pipeline

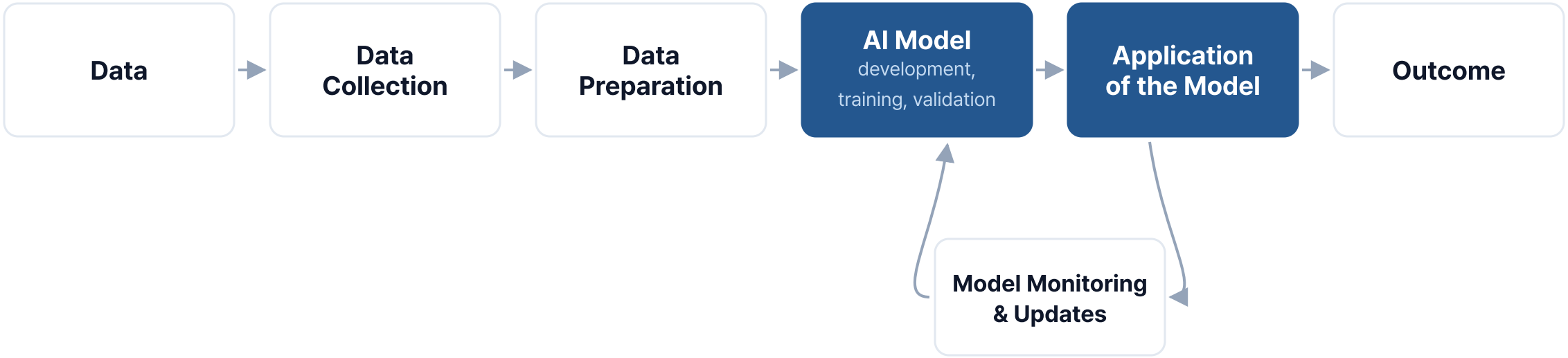


Data becomes an outcome through a **pipeline**, and the model loops back for updates

02 · Course Scope

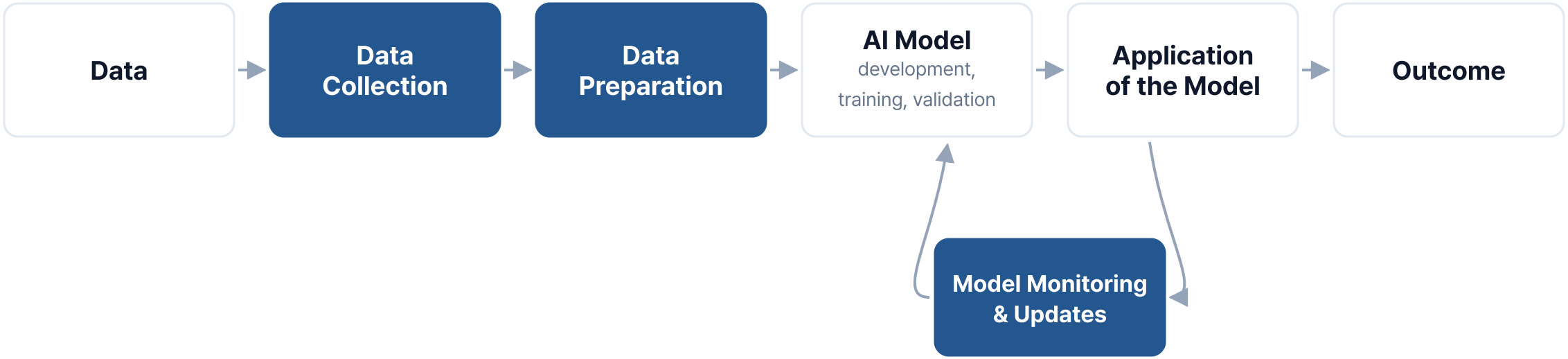
01	The Pipeline	follow records from source to outcome
02	**Course Scope**	collection, storage and monitoring
03	Course Logistics	evaluation, project and AI policy

Spring Course Scope



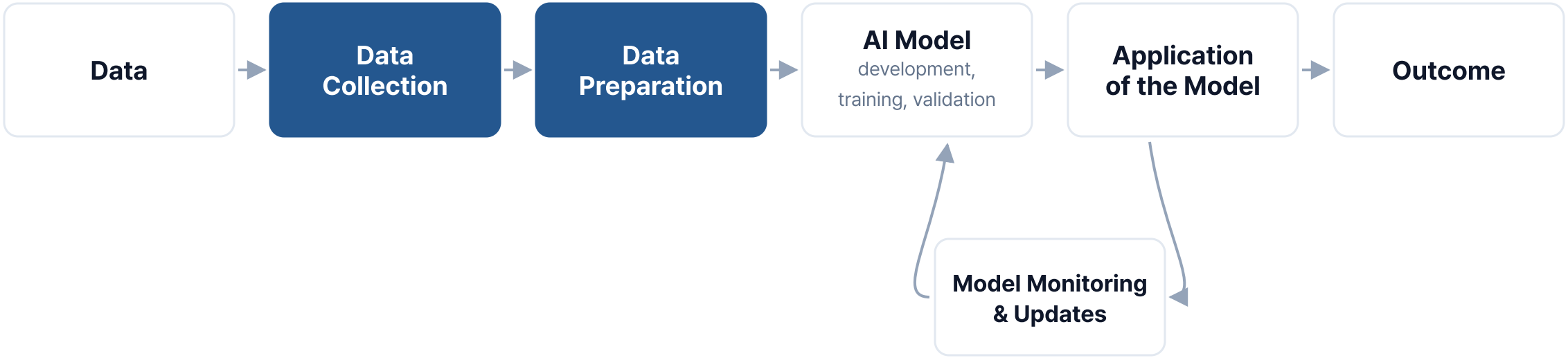
Spring covers **building and applying** the model

Fall Course Scope



Fall covers what happens **before** and **after** the model

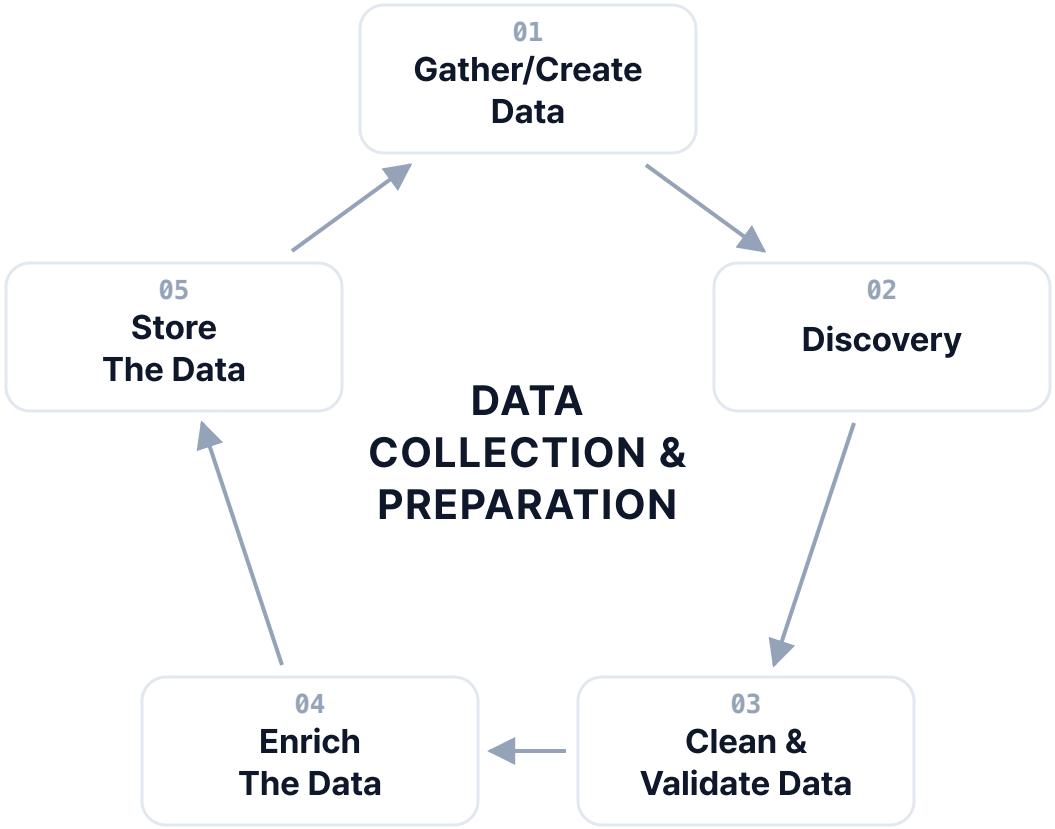
Part I & II: Before the Model



We start with **data collection** and **data preparation**

Five Steps Before the Model

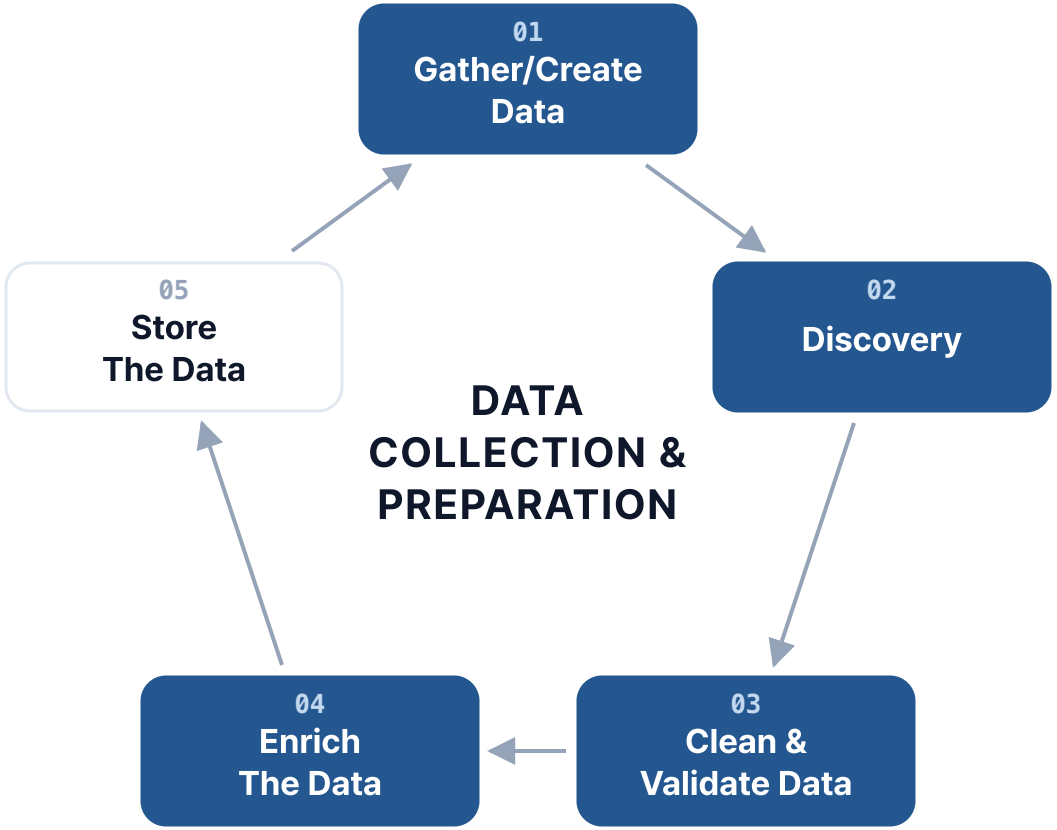
Collection, preparation and storage, as one cycle



Collection and preparation unfold into a **cycle** of five steps

Part I (Lectures 2–5)

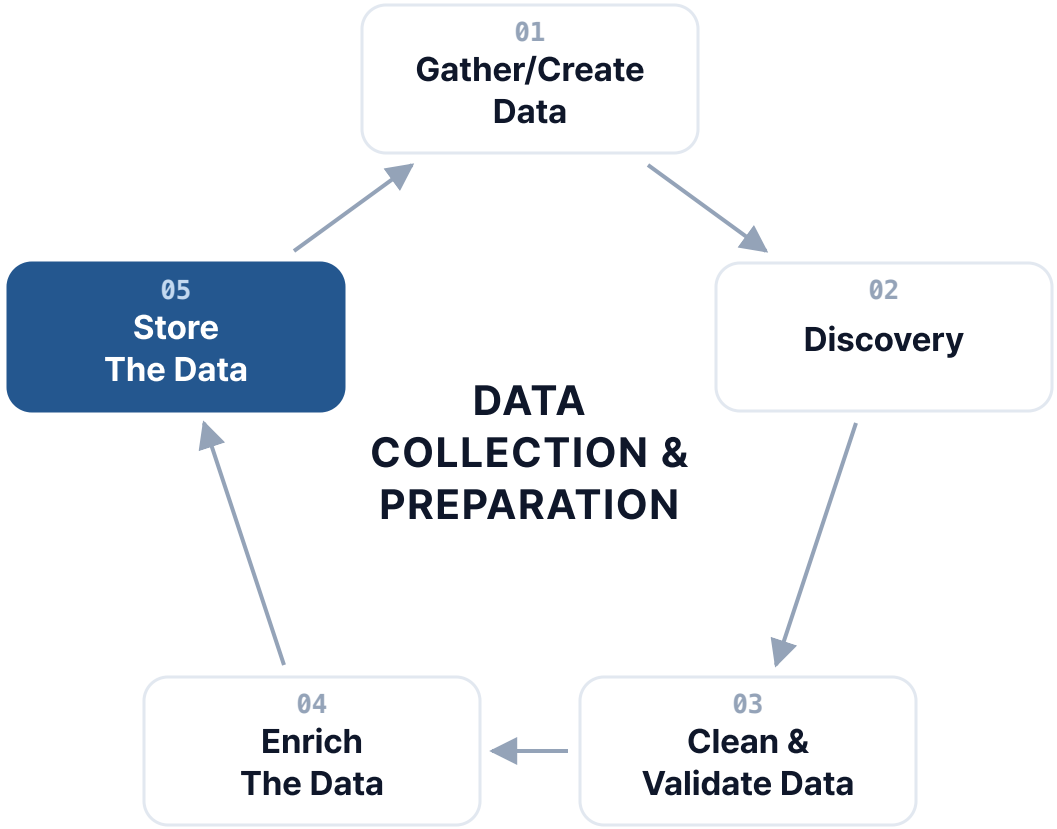
Collection and Preparation



Part I covers collection, exploration, and preprocessing

Part II (Lectures 6–13)

Storage: SQL, NoSQL, Big Data



Part II covers step 05: storing, querying and processing the data

Data Workloads and Tools

The workload determines which storage and processing tools fit

One local analysis

Read files

Explore and calculate

`pandas · DuckDB`

→

Shared data service

Repeated queries

Concurrent readers and writers

`SQL · document stores`

→

Distributed processing

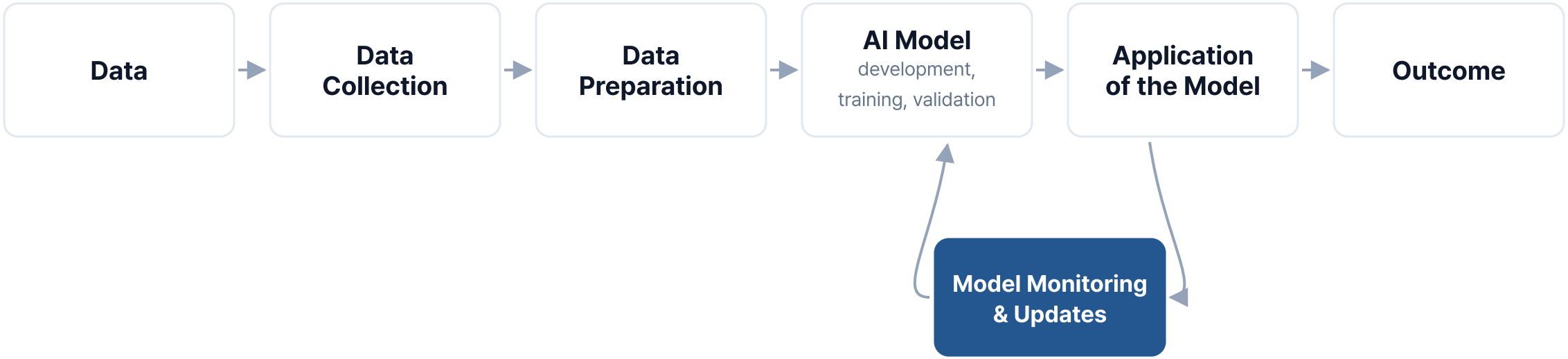
Work across machines

Large batch or event streams

`Spark · stream engines`

Choose tools for the task, shared access and processing requirements

Part III: Monitoring and Updates



After the model ships, it is **monitored and updated** when needed

From DevOps to AgentOps

	DevOps Development + Operations	MLOps Machine Learning Operations	LLMOps Large Language Model Operations	AgentOps Agent Operations
What ships	Code	+ data, model	+ prompts, context	+ tools, multi-step plans
What fails quietly	Regressions	Data drift	Hallucination, cost	Wrong actions, loops
What you watch	Uptime, latency	Accuracy over time	Evals, tokens	Traces, tool calls

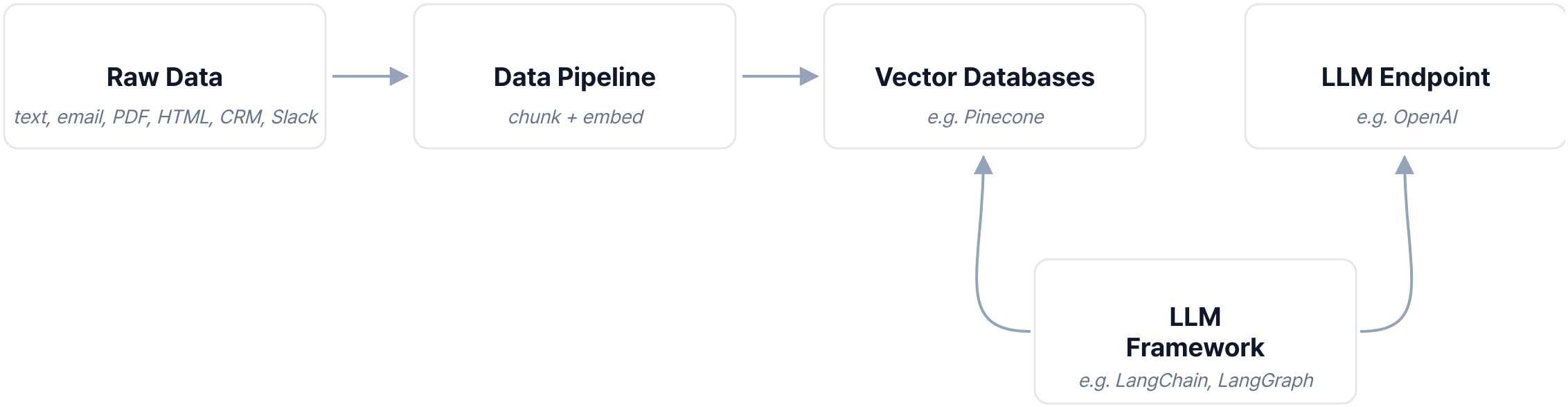
Part III walks all four: each step adds what ships and what can fail quietly

When the Model Is an LLM



The model is now **called**, and the **agent** around it is built

LLM Applications



In retrieval-augmented generation (RAG), a **pipeline** fills the **vector database** the LLM retrieves from

Part III (Lectures 14–17)

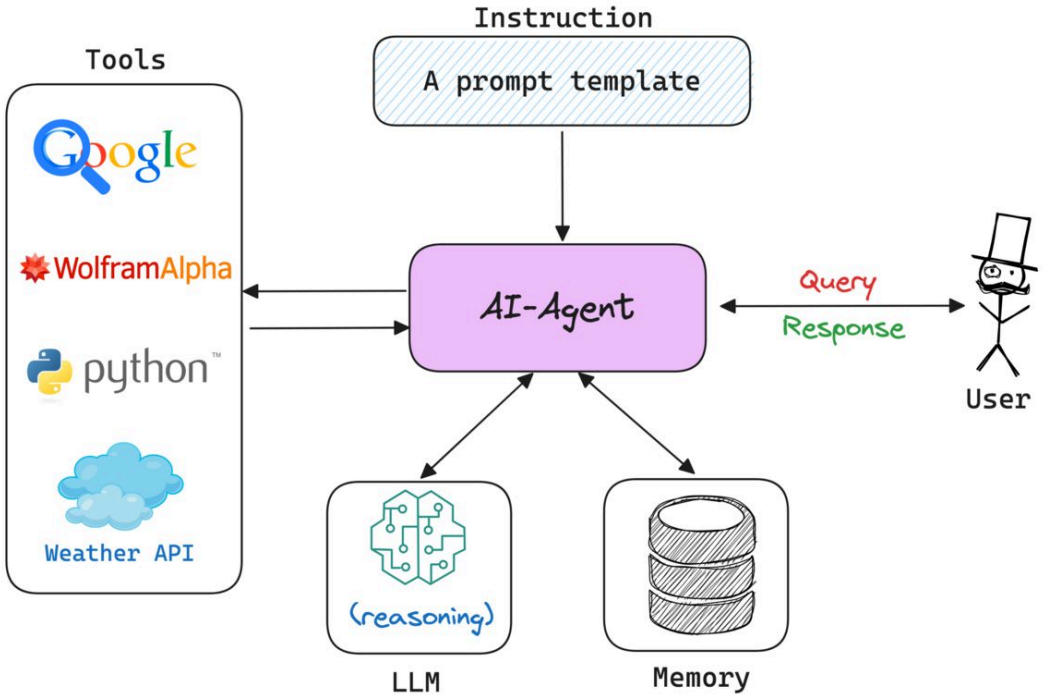
Monitoring and Updates

14 ML Systems	**DevOps vs MLOps, data drift, model monitoring**
15 LLM Systems	**LLMs, frameworks, LLMOps, data for LLMs**
16 RAG	**Retrieval-augmented generation: embeddings, vector DBs**
17 Agents	**The agent loop, workflows, and agents on your data**

Part III goes from MLOps to agents that work on **your data**

AI Agent + DB

Agent structure



The database role

Tools read and act on data

Memory retains useful information

A database can persist that memory

An agent needs access to data and a way to retain useful information

Data Engineering Lifecycle

Stages organize the work; deployed systems feed new observations back

Generate and ingest

Events become records

Collection · Lec 2

→

Store and transform

Query usable data

Preparation · Lec 3–5

Storage · Lec 6–13

→

Serve and update

Use data in a system

Monitoring and updates

Lec 14–17

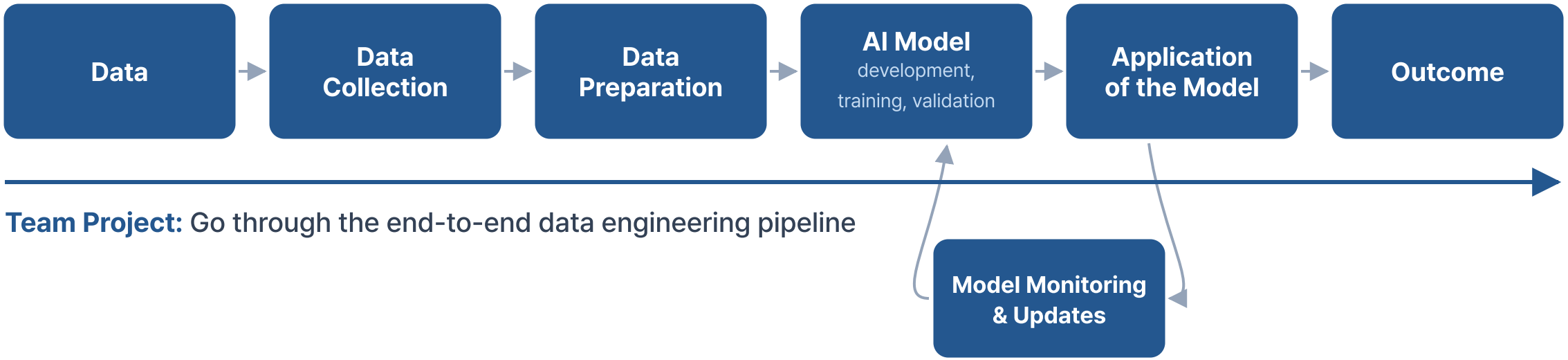
Records pass through stages and return as feedback

Requirements Across the Lifecycle

Requirement	Question across stages	Lecture examples
Security	Who may read or write these records?	Collection · Agents
Data management	What do records mean and how do they change?	Storage · Agents
DataOps and orchestration	Can a failed job be checked and rerun?	Data Warehousing · Systems
Architecture	Where should storage and processing run?	NoSQL · Big Data
Software engineering	Can changes be tested and maintained?	A requirement throughout

These requirements apply across the stages, not just at one endpoint

By the End of the Class



Team Project: Go through the end-to-end data engineering pipeline

By the end, you will have walked the **whole pipeline** once

03 · Course Logistics

01	The Pipeline	follow records from source to outcome
02	Course Scope	collection, storage and monitoring
03	**Course Logistics**	evaluation, project and AI policy

Grading Policy

40%
Midterm Exam

20%
Project (Proposal)

40%
Project (Final)

0%
Attendance (excessive absence means F)

Office Hours: **Tuesdays 11:00 am–12:00 pm, Room 319**

Team Project Evaluation Criteria

CRITERION	WHAT WE LOOK FOR
Novelty	originality and creativity of the idea
Impact	relevance and potential usefulness of the solution
Technology	effective use of the techniques and tools covered in class
Presentation	clarity and professionalism of the report and presentation

Team Size: **≤ 5**

Team Project Milestones

PROPOSAL · MID-SEMESTER · 20%

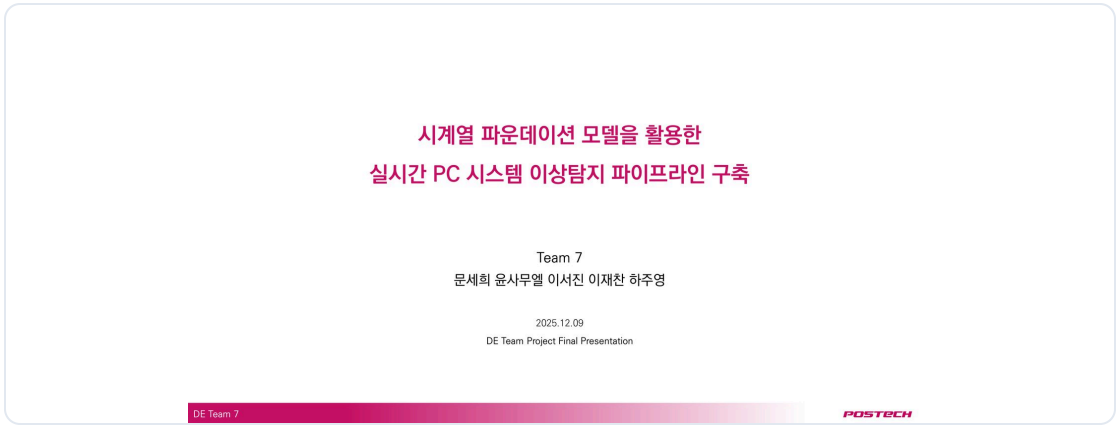
Ideation, data collection & preprocessing, DB management

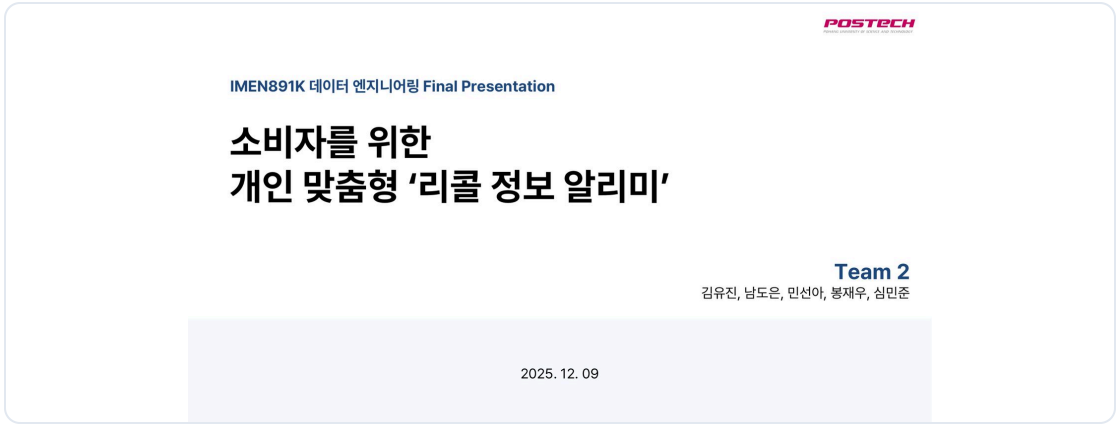
FINAL · END OF SEMESTER · 40%

Develop a product that addresses a real-world problem using a large-scale DB (with potential LLM integration)

Team Project Examples









Four final presentations from **Fall 2025**

Data Sources for Team Projects

Public data platforms

data.go.kr, Kaggle, Hugging Face datasets

Open APIs

Government, map, finance, or social platform APIs

Web crawling

Pages you collect yourself, within the site's terms

Your own data

Lab, company, sensors, surveys you run

Generated data

Synthetic or LLM-generated data, with a clear reason

Or a mix

Combine sources; joining them is part of the work

Any source is fine if you are **allowed to use it**: check the license, terms, and personal data

AI Policy



ChatGPT

Do not **compete** with AI, **utilize** it!

Takeaway: Data Engineering

- Define **what the record means** and whether it supports the task
- Choose storage and processing for **the workload and required guarantees**
- Check results and failures; use **observed outcomes** to guide updates