DeepLearning.AI

Data Engineering

Welcome!

Data Engineering Professional Certificate Program

Course 1 Introduction to Data Engineering

Course 2 Source Systems, Data Ingestion, and Pipelines

Course 3 Data Storage and Queries

Course 4 Data Modeling, Transformation, and Serving

DeepLearning.AI

Introduction to Data Engineering

Course 1 Overview

Scenario

Data Engineer

Wasting time & resources!

System Requirements

Functional Requirements	Nonfunctional Requirements

Technology Choices

- Ingestion tool
- Storage system
- Transformation tool

Plan for Course 1

Week 1 **High level look at the field of data engineering**

- Data Engineering lifecycle
- History of Data Engineering
- The Data Engineer among other stakeholders
- Business value
- Translation of stakeholder needs into requirements

Week 2 **Data engineering lifecycle and undercurrents**

Week 3 **Principles of good data architecture**

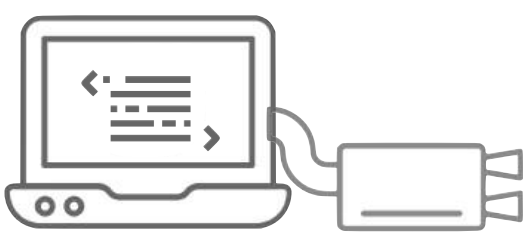
Week 4 **Design and build out a data architecture**

DeepLearning.AI

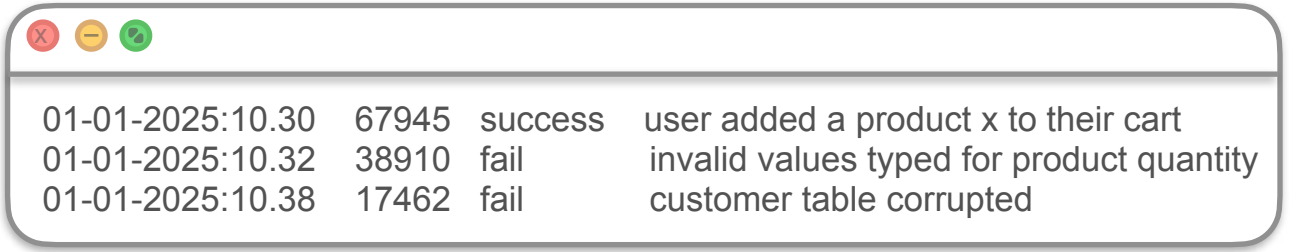
Introduction to Data Engineering

Data Engineering Defined

Data Engineering

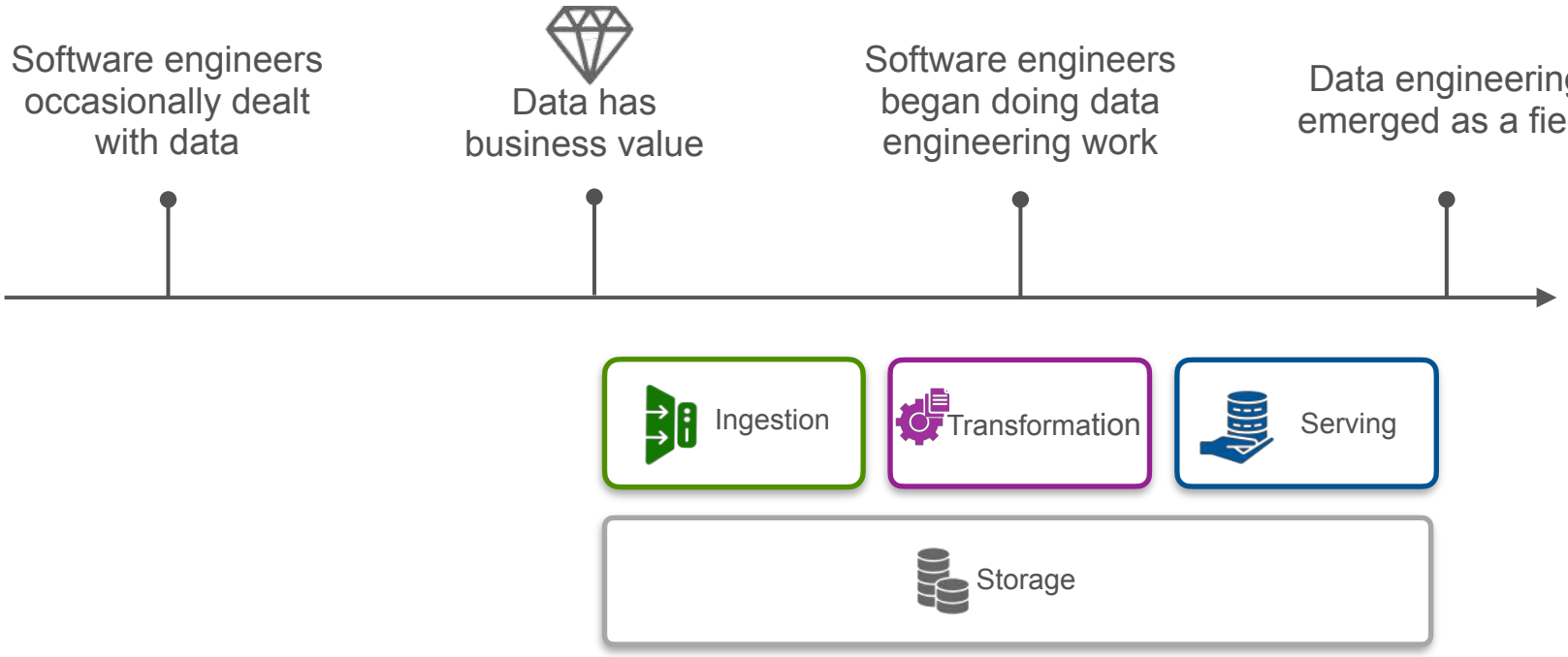


Software Application

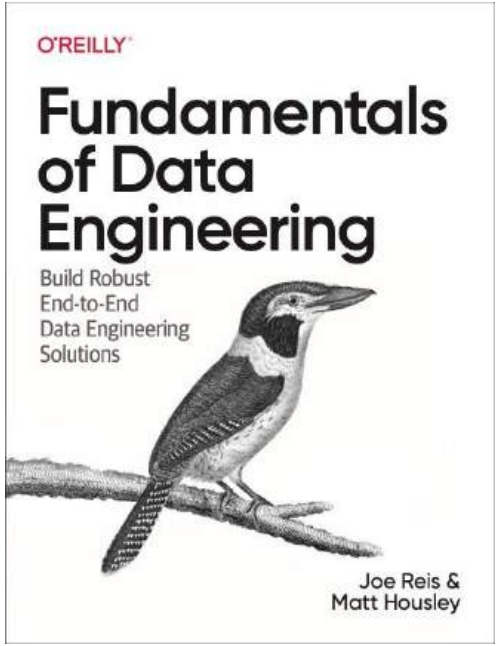


Exhaust / Byproduct

Software Engineering

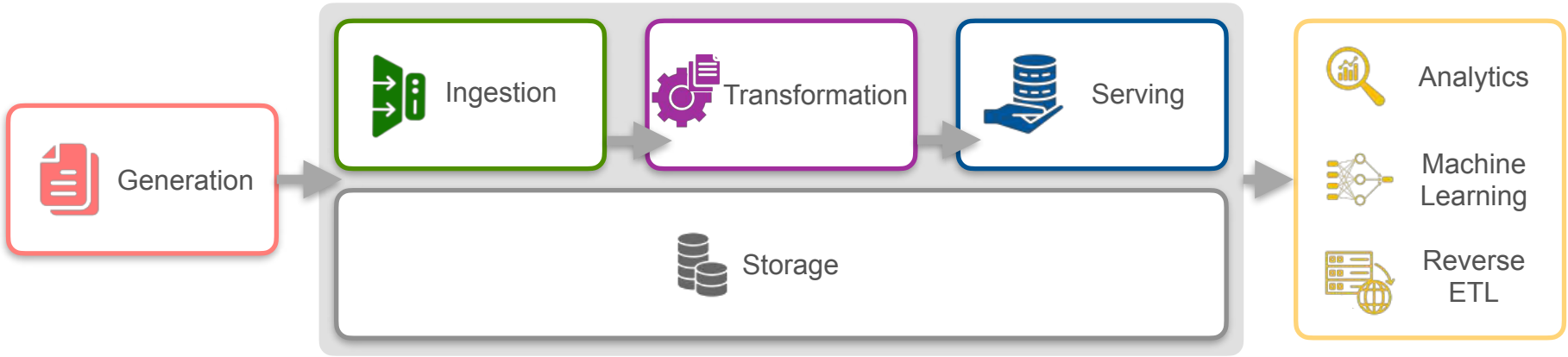


Data Engineering

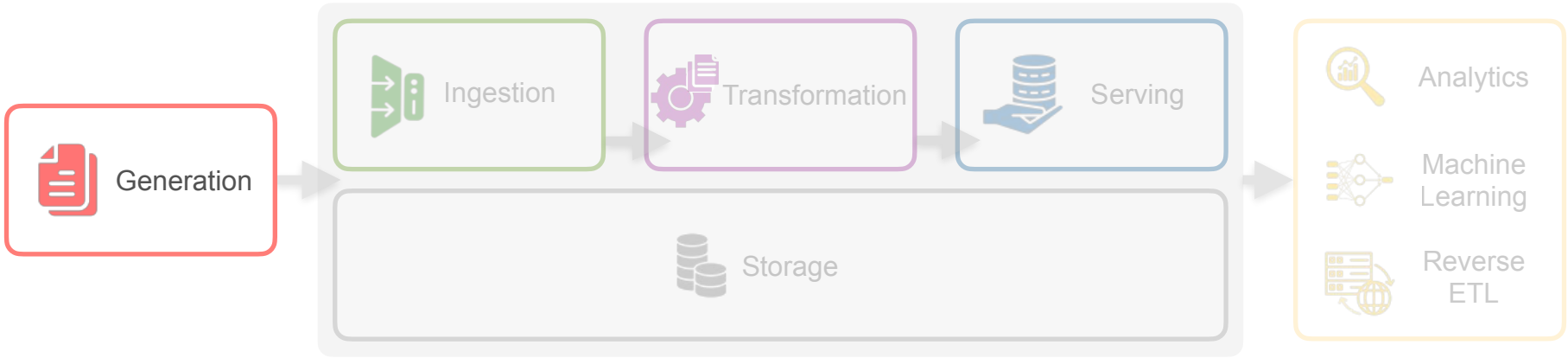


"Data engineering is the development, implementation, and maintenance of systems and processes that take in raw data and produce high-quality, consistent information that supports downstream use cases, such as analysis and machine learning. Data engineering is the intersection of security, data management, DataOps, data architecture, orchestration, and software engineering."

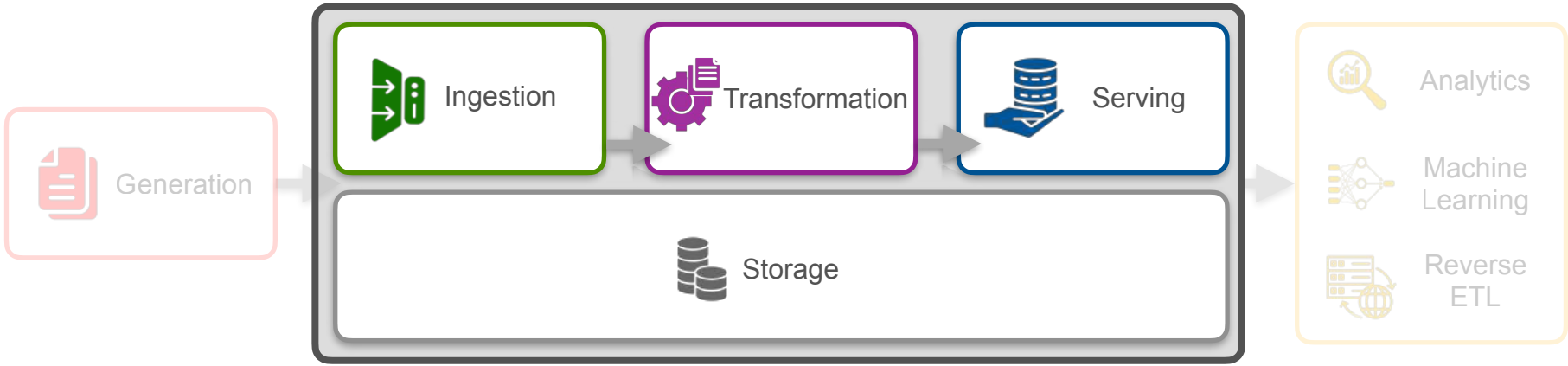
Data Engineering Lifecycle



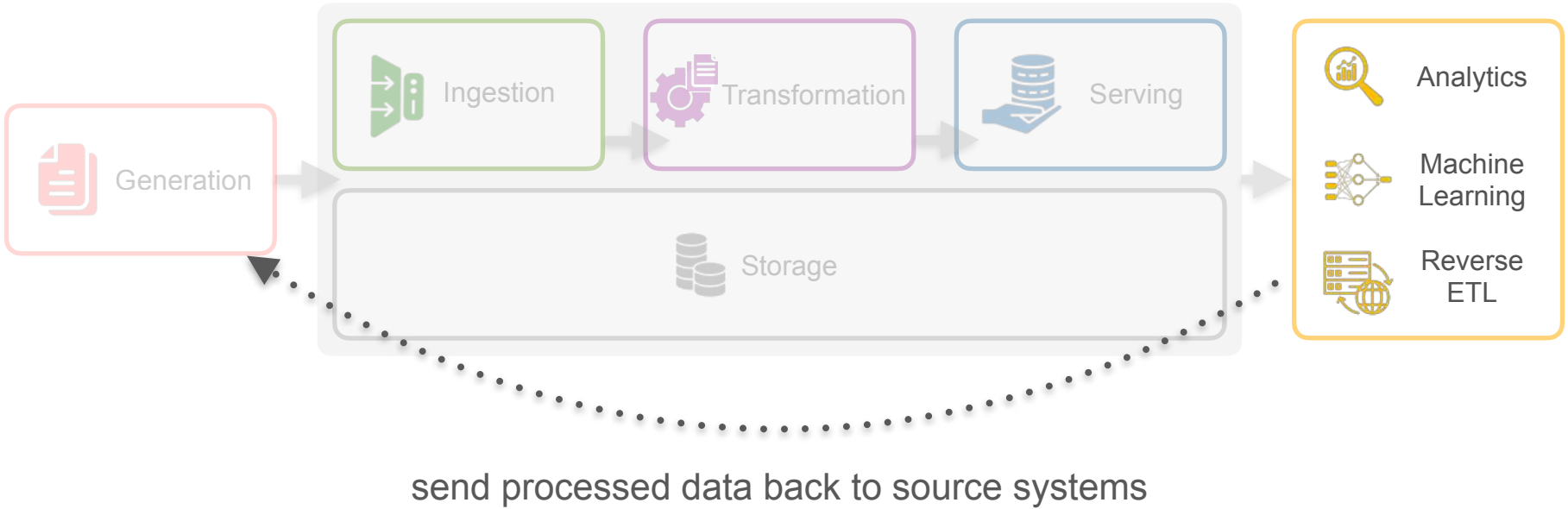
Data Engineering Lifecycle



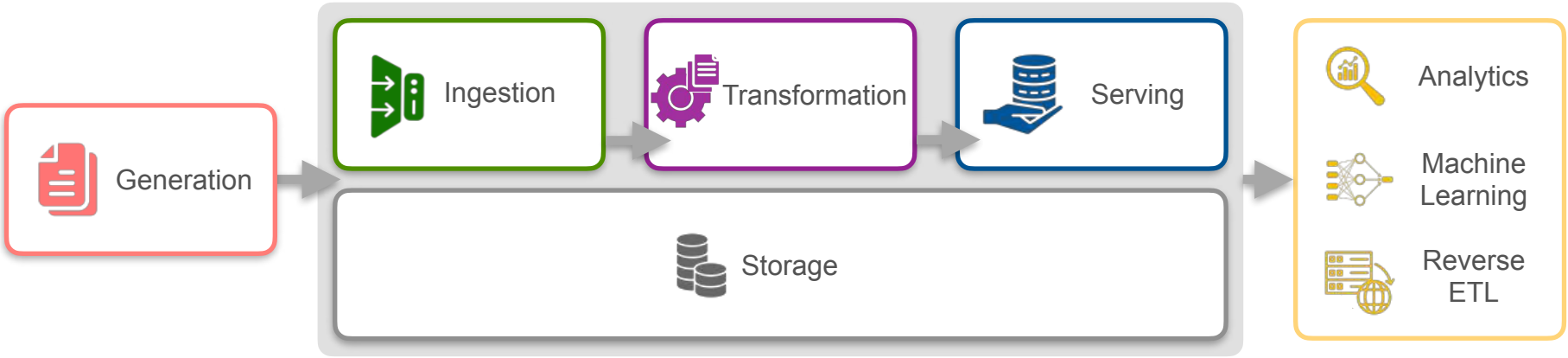
Data Engineering Lifecycle



Data Engineering Lifecycle

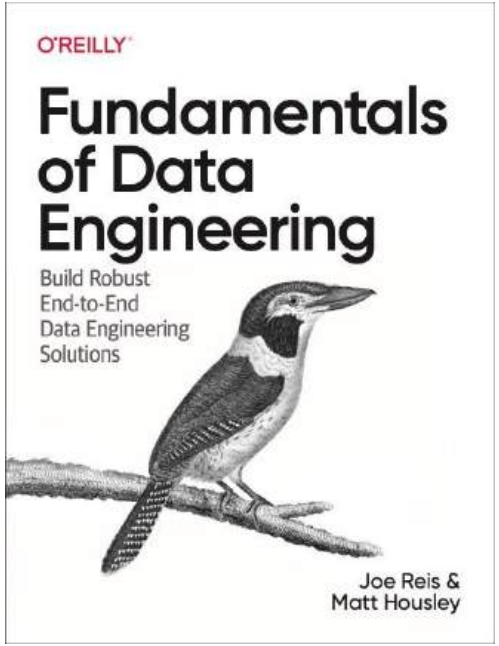


Data Pipeline



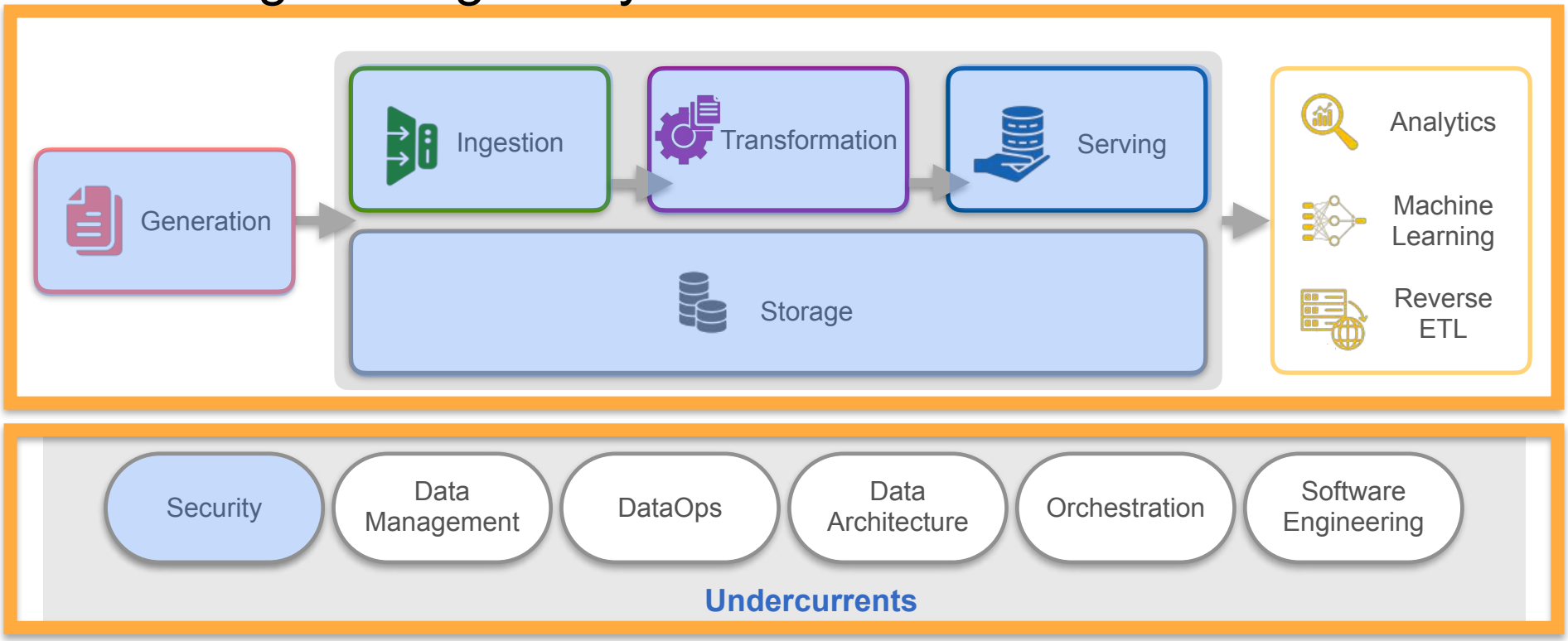
Data Pipeline The combination of architecture, systems, and processes that move data through the stages of the data engineering lifecycle.

Data Engineering



"Data engineering is the development, implementation, and maintenance of systems and processes that take in raw data and produce high-quality, consistent information that supports downstream use cases, such as analysis and machine learning. Data engineering is the intersection of Security, Data Management, DataOps, Data Architecture, Orchestration, and Software Engineering."

Data Engineering Lifecycle & Undercurrents



DeepLearning.AI

Introduction to Data Engineering

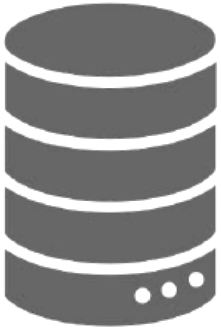
A Brief History of Data Engineering

History of Data Engineering

1960s

Computers

Computerized Database

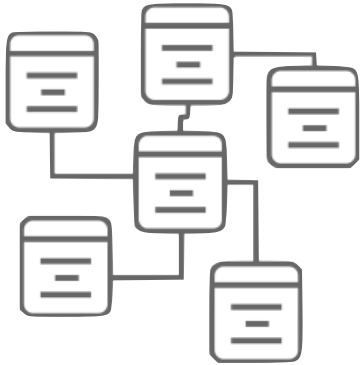


History of Data Engineering

1960s

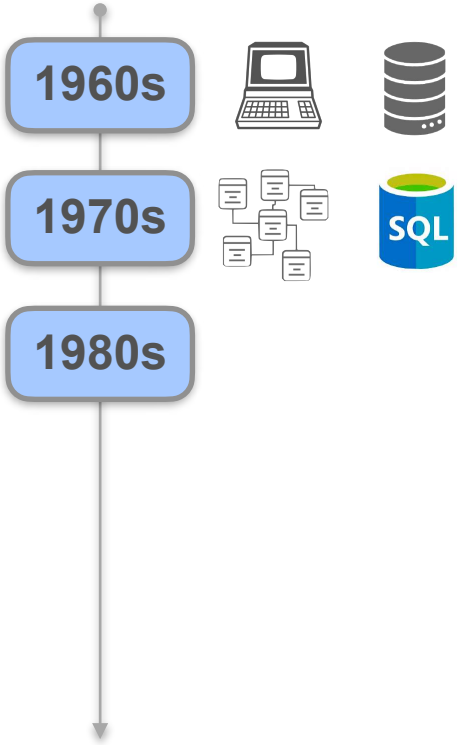
1970s

Relational Databases



Structured Query Language

History of Data Engineering

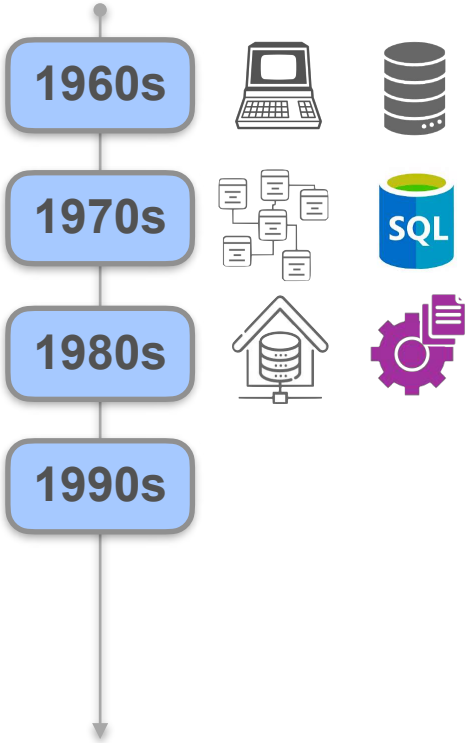


Bill Inmon

Data Warehouse

Transforming Data

History of Data Engineering

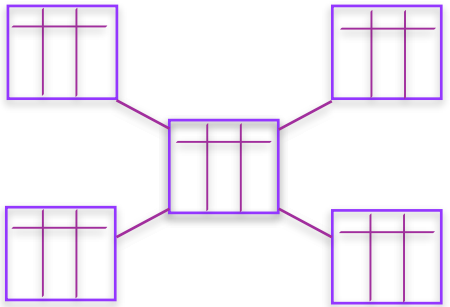


Business Intelligence

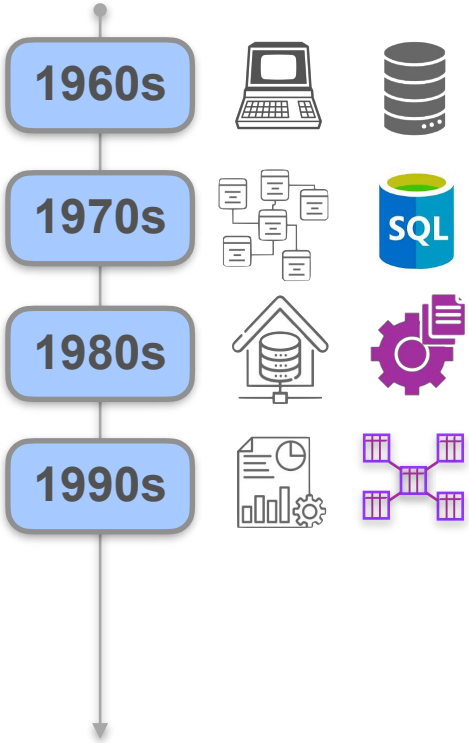
Bill Inmon

Ralph Kimball

Data Modeling



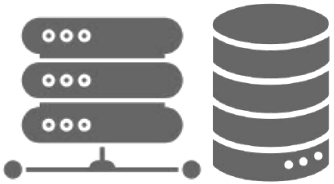
History of Data Engineering



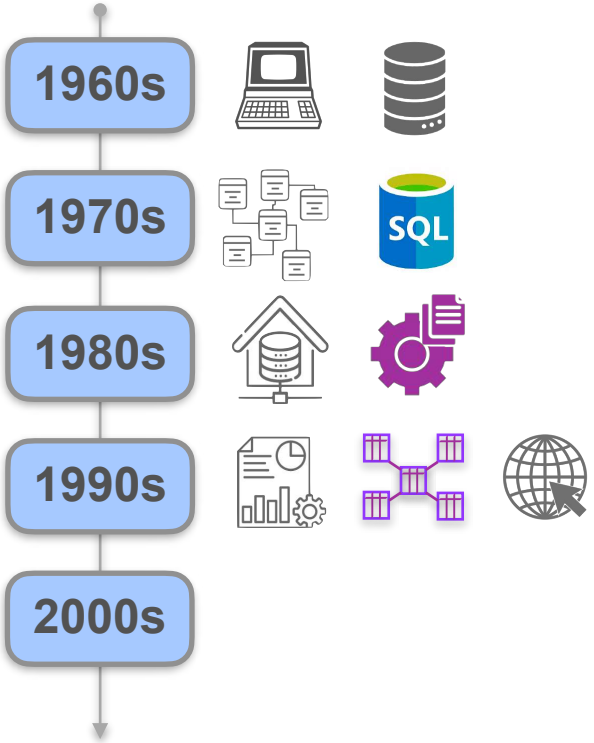
Internet

Web-first Companies

Backend Systems



History of Data Engineering



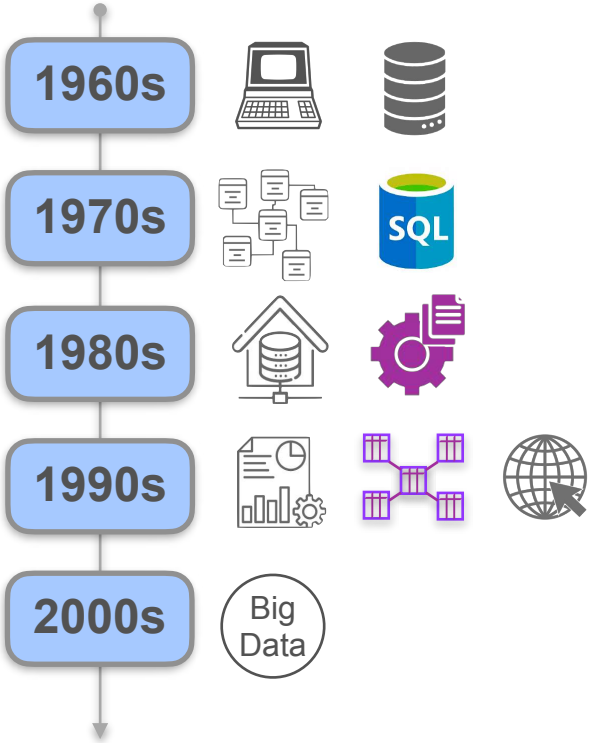
yahoo!

Google

amazon

Big Data

History of Data Engineering



The "Big Data" Era

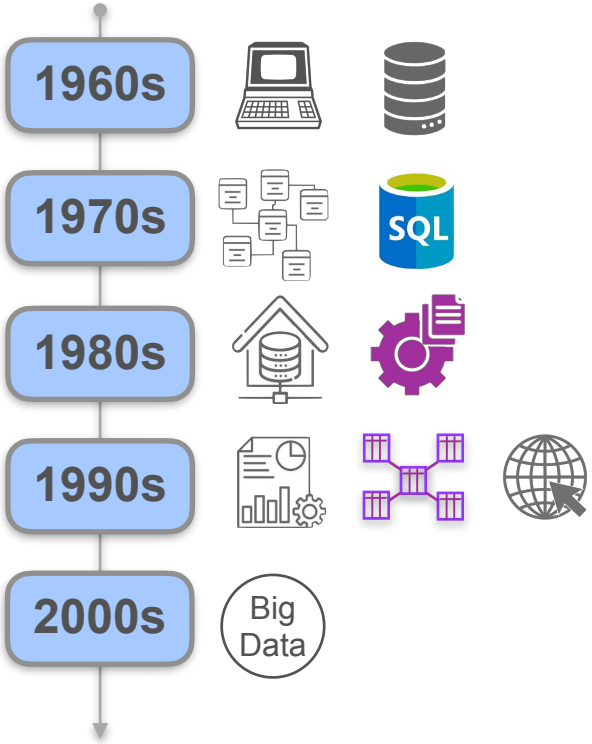
"extremely large data sets that may be analyzed computationally to reveal patterns, trends, and associations, especially relating to human behavior and interactions."

Velocity

Variety

Volume

History of Data Engineering



2004

Google

MapReduce: Simplified Data Processing on Large Clusters

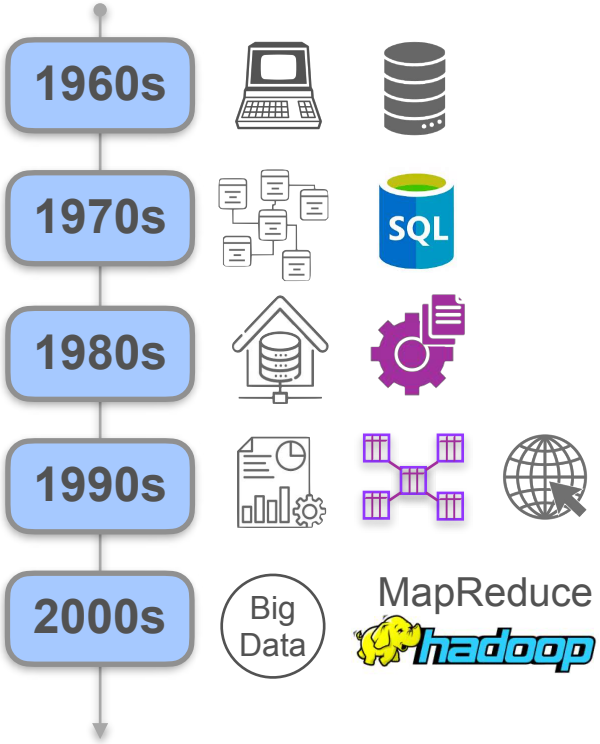
2006

yahoo!

hadoop

The “Big Data Engineer” Era

History of Data Engineering



Pay-as-you-go resource marketplace

Amazon EC2

Amazon S3

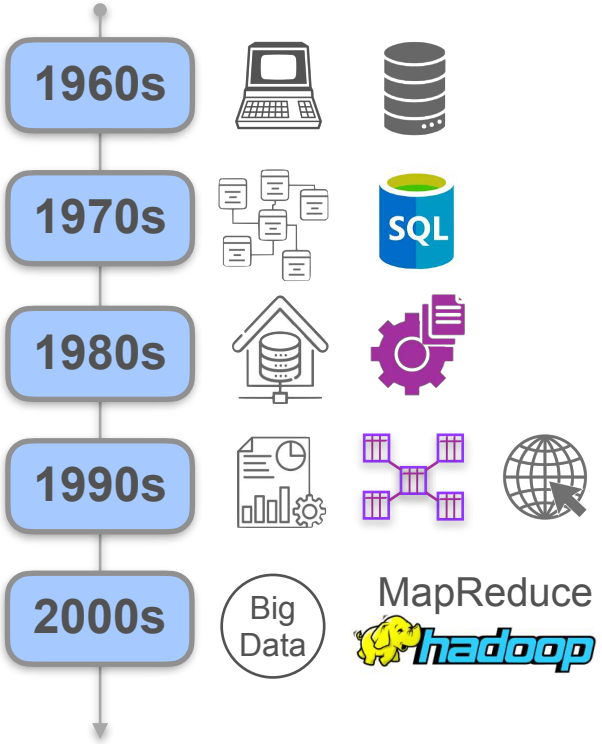
Amazon DynamoDB

aws

Amazon Web Services

The first popular public cloud

History of Data Engineering



Public Cloud

aws

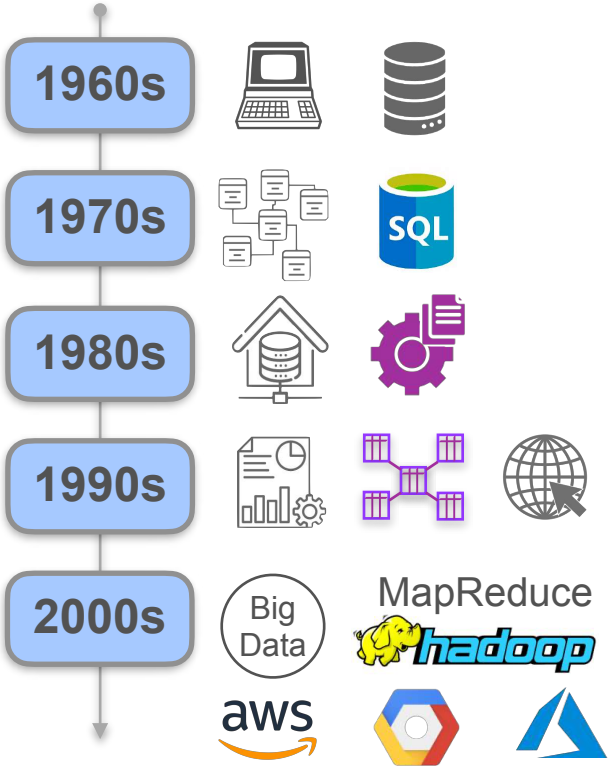
Google Cloud Platform

Azure

Public Cloud & Early big data tools :

Foundation for today's data ecosystem

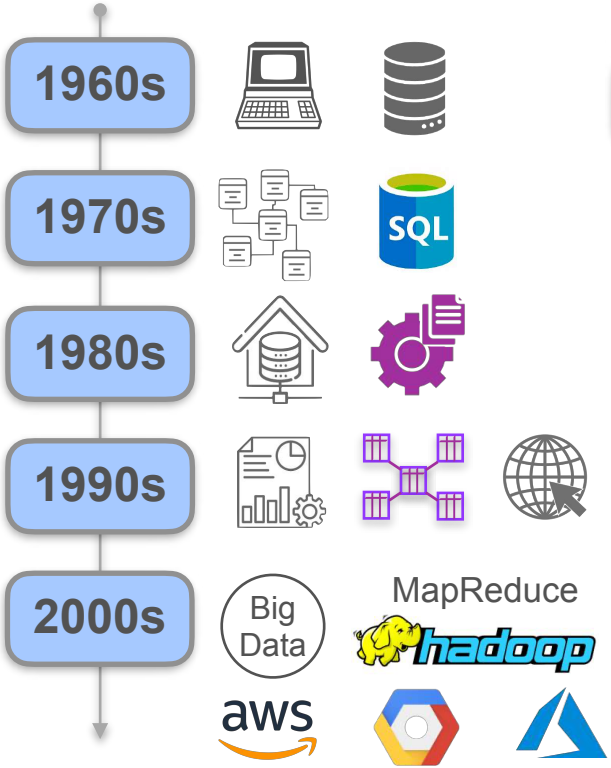
History of Data Engineering



Late 2000s and 2010s (big data tools)

- Access to bleeding-edge data tools
- Transition from batch computing to event streaming

History of Data Engineering



2010s - present

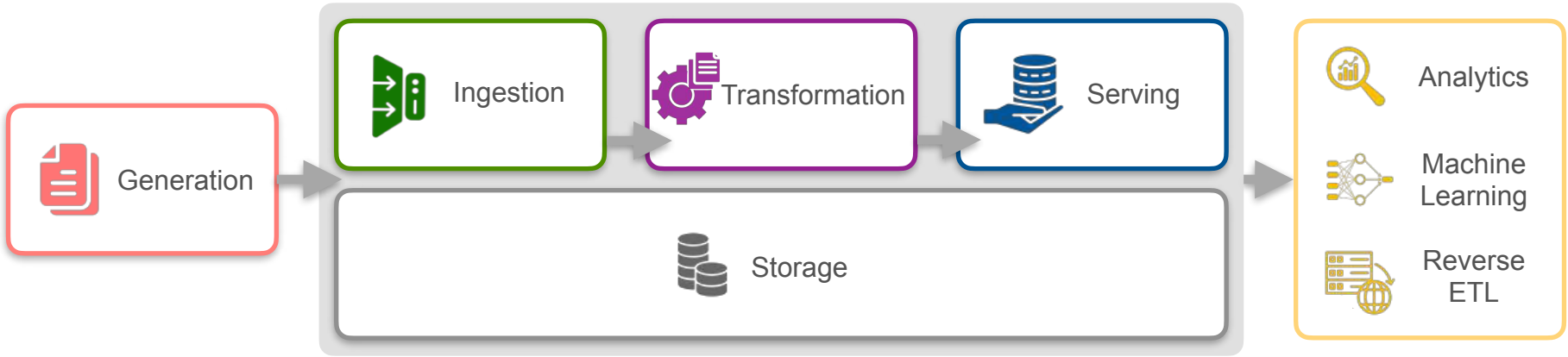
Abstract Simplified Tools

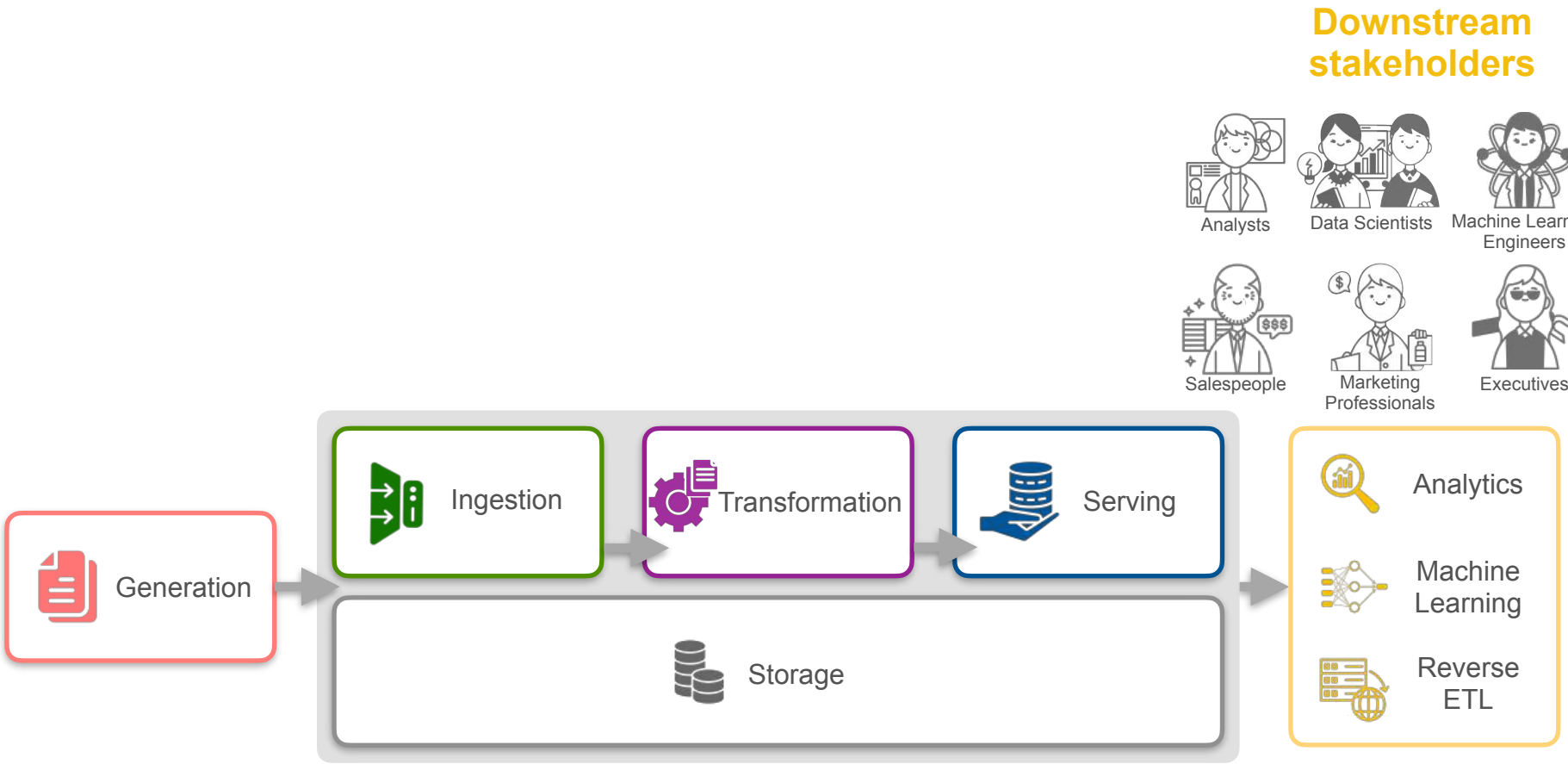
DeepLearning.AI

Introduction to Data Engineering

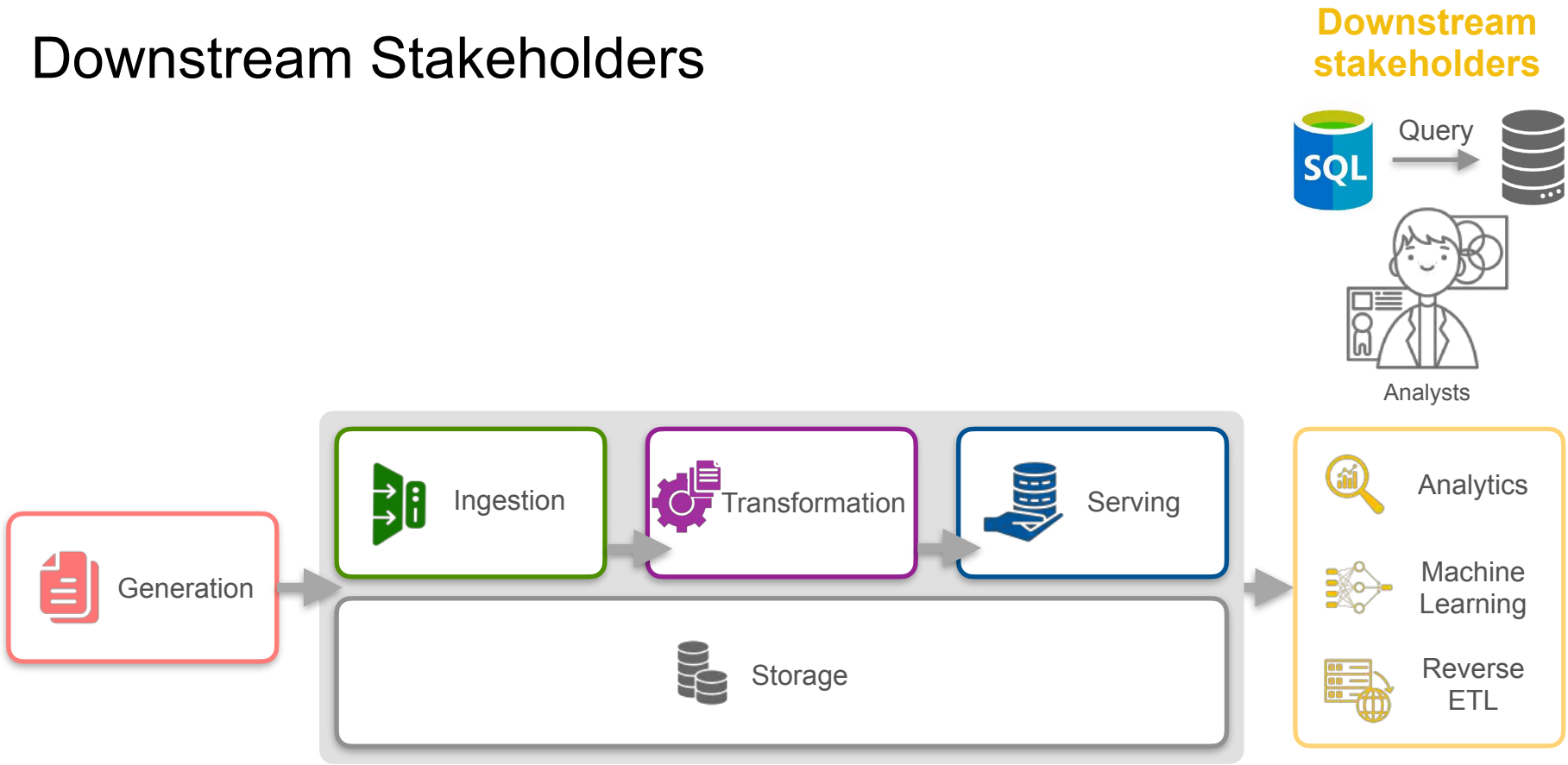
The Data Engineer Among Other Stakeholders

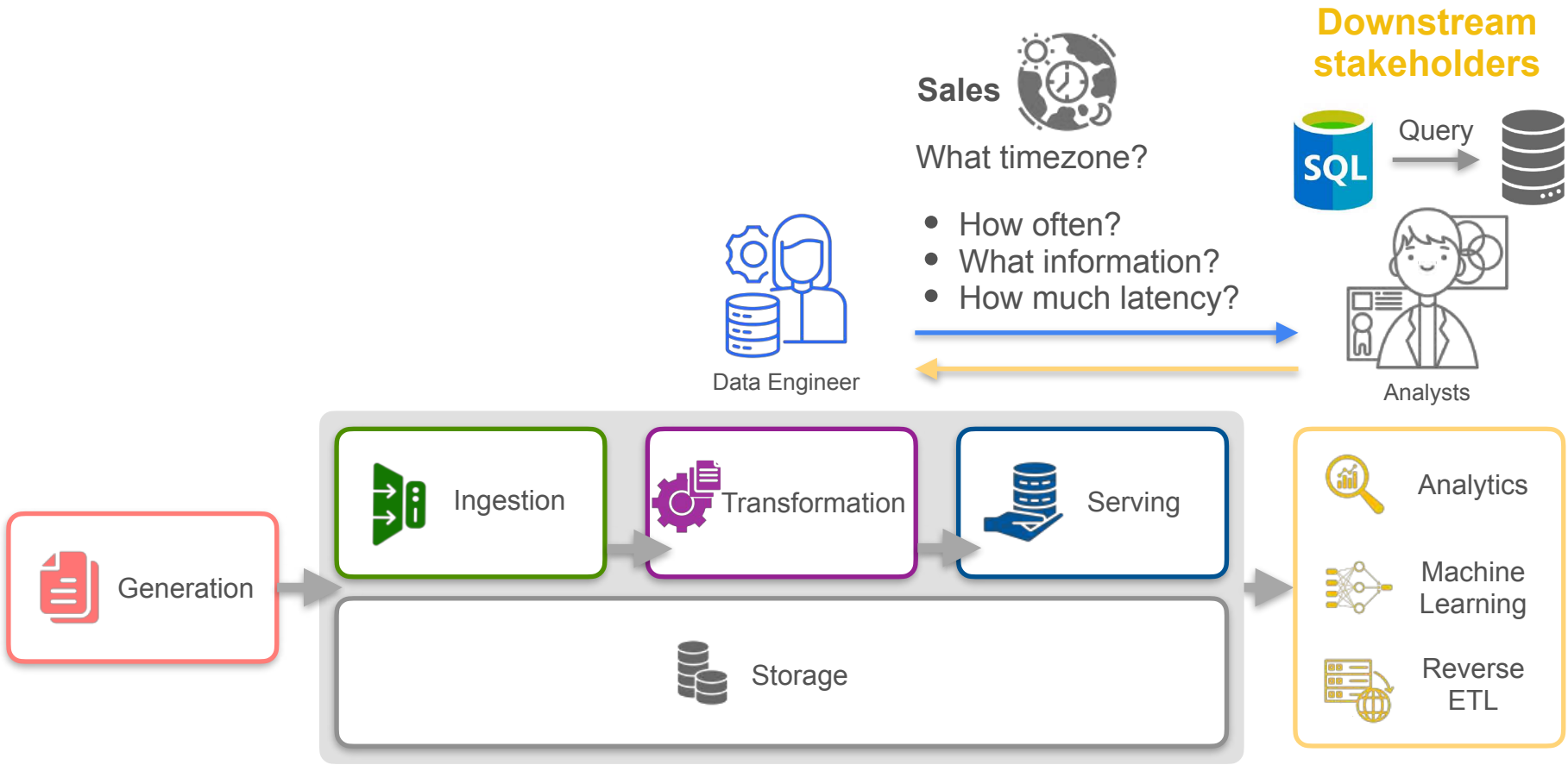
Downstream Use Cases



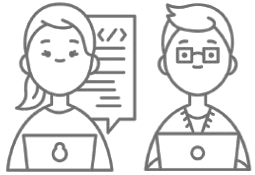

Downstream stakeholders
Analysts
Data Scientists
Machine Learning Engineers
Salespeople
Marketing Professionals
Executives
Generation
Ingestion
Transformation
Serving
Storage
Analytics
Machine Learning
Reverse ETL

Downstream Stakeholders




Downstream stakeholders
Sales
What timezone?
Query
SQL
How often?
What information?
How much latency?
Data Engineer
Analysts
Generation
Ingestion
Transformation
Serving
Storage
Analytics
Machine Learning
Reverse ETL

Upstream stakeholders



Software Engineers

- Volume
- Frequency
- Format
- Data Security
- Regulatory compliance

Data Consumer

Data Engineer

Sales

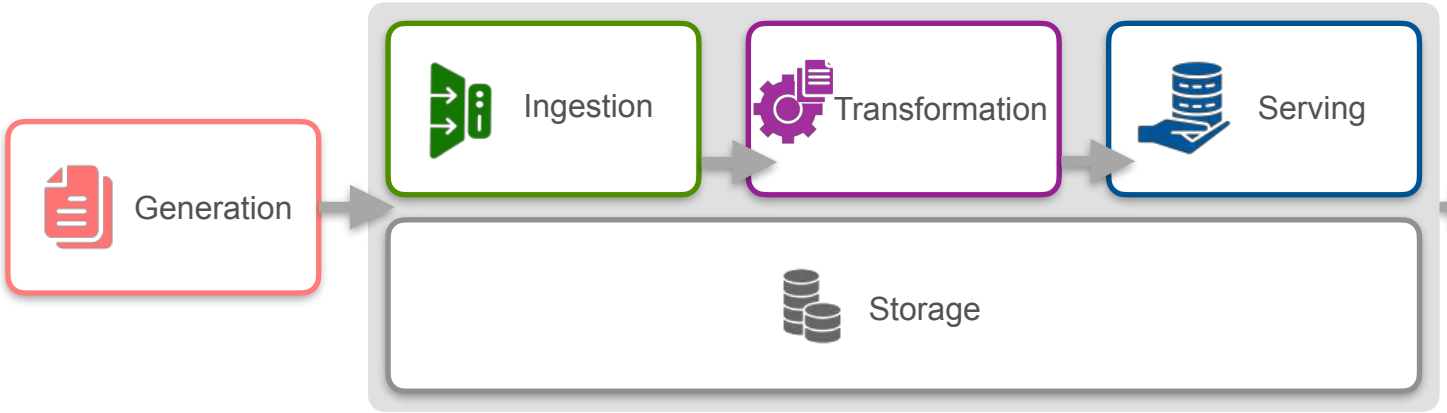
What timezone?

- How often?
- What information?
- How much latency?

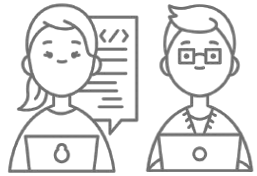
Downstream stakeholders

Query

Analysts



Upstream stakeholders



Software Engineers

- Volume
- Frequency
- Format
- Data Security
- Regulatory compliance

Data Engineer

- How often?
- What information?
- How much latency?

Downstream stakeholders

Analysts

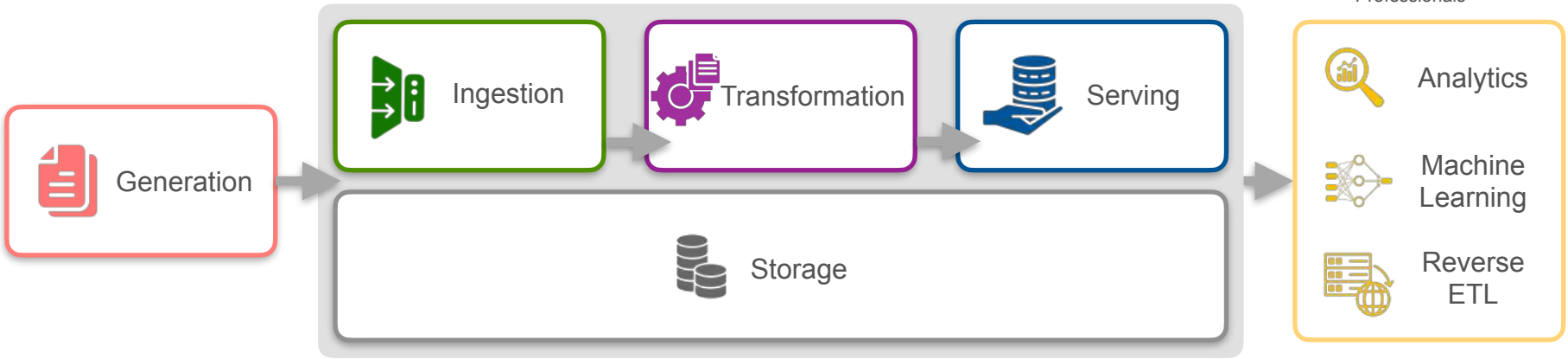
Data Scientists

Machine Learning Engineers

Salespeople

Marketing Professionals

Executives



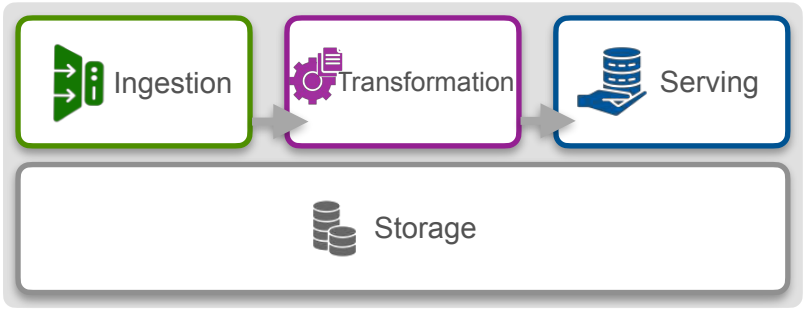
DeepLearning.AI

Introduction to Data Engineering

Business Value

Business Value

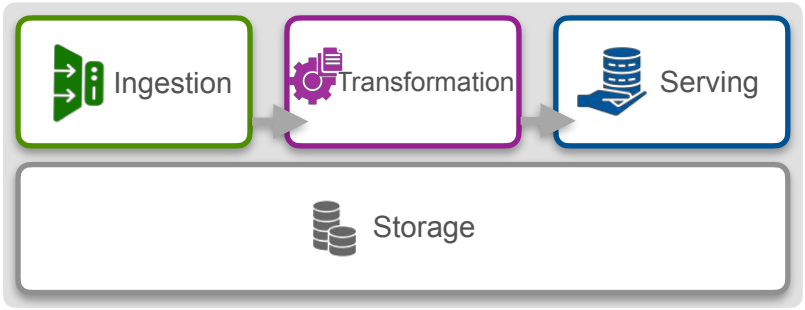
Goal: Revenue Growth



Value Created!

Business Value

Goal: Revenue Growth

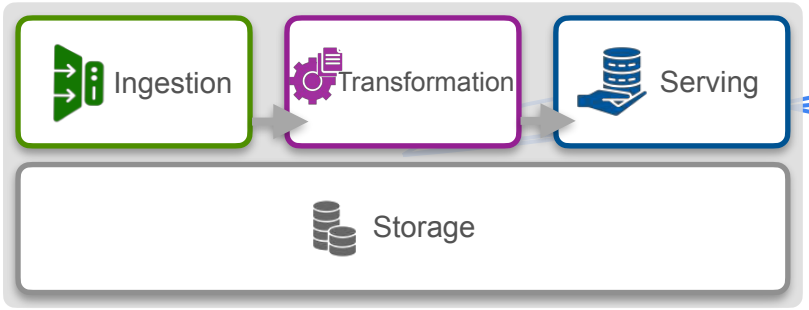


No Value!

Business Value

Is your work helping them achieve their goals?

Multiple forms for business value



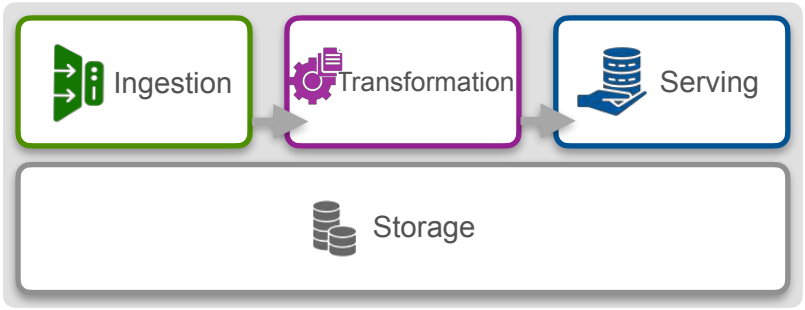
Analysts

Machine Learning Engineers

Marketing Professionals

Business Value

Multiple forms for business value



- Increased Revenue
- Cost Savings
- Improved efficiency
- Launch a product

DeepLearning.AI

Introduction to Data Engineering

System Requirements

Requirements

Business Requirements

High level goals of the business

For example: grow revenue, increase user base

Requirements

Business Requirements

High level goals of the business

For example: grow revenue, increase user base

Stakeholder Requirements

Needs of individuals within the organization

Things they need to get their job done well

Requirements

Business Requirements

High level goals of the business

For example: grow revenue, increase user base

Stakeholder Requirements

Needs of individuals within the organization

Things they need to get their job done well

System Requirements

- **Functional Requirements** — **The "WHAT"**
- **Non-Functional Requirements** — **The "HOW"**

Requirements

Functional Requirements

What the system needs to be able to do

- Provide regular updates to a database
- Alert a user about an anomaly in the data

Non-Functional Requirements

How the system accomplishes what it needs to do

- Technical specifications of an ingestion or orchestration or storage approach
- How you'll meet the end user's needs

Requirements Gathering

Business & Stakeholder Requirements

Features & Attributes

Memory & Storage Capacity

Cost & Security Constraints

Gather your system requirements

Translate

Data Engineer

High-level goals & needs

Analysts

ML Engineers

Marketing Professionals

DeepLearning.AI

Introduction to Data Engineering

Translate Stakeholder Needs into Specific Requirements

Key Elements of Requirements Gathering

Learn what existing data systems or solutions are in place

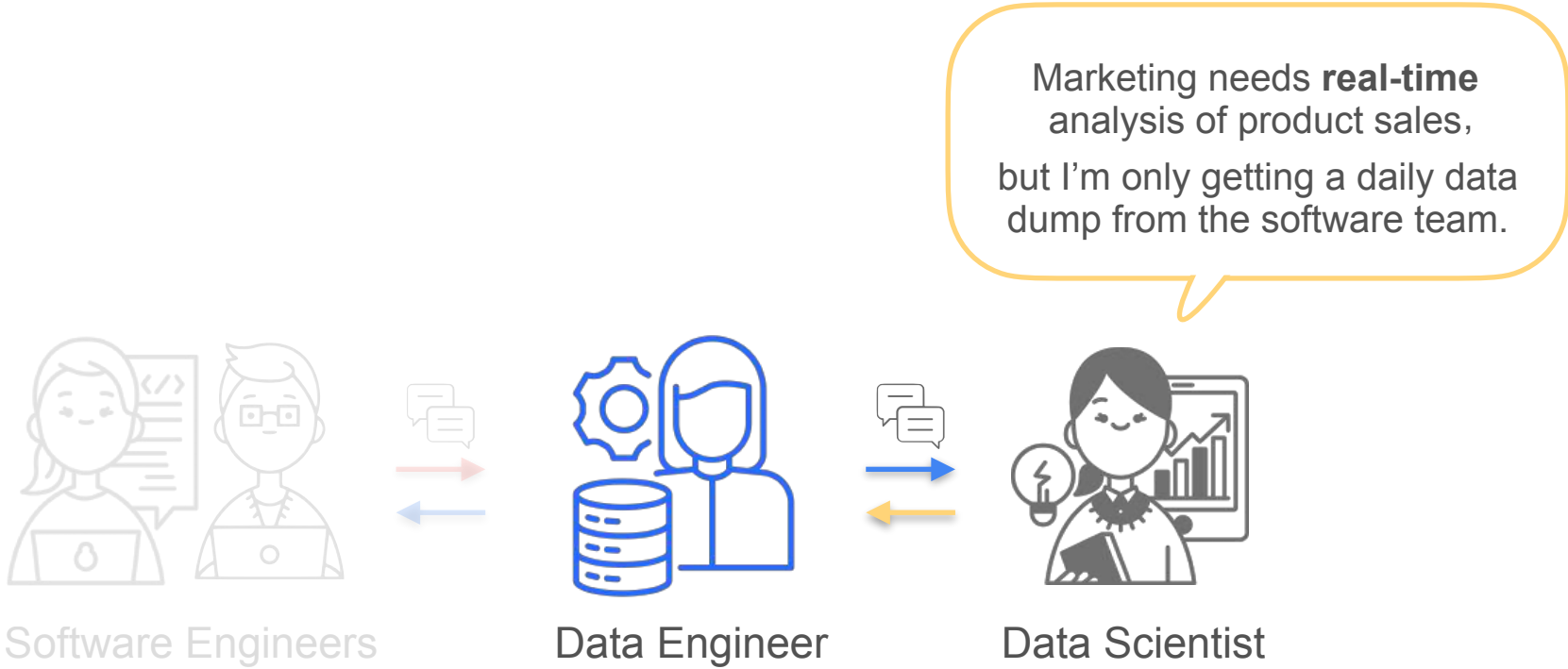
Learn what pain points or problems there are with the existing solutions

Learn what actions stakeholders plan to take with the data

Tip: Repeat what you learned back to your stakeholders.

Identify any other stakeholders you'll need to talk to if you're still missing information

Conversation with Data Scientist



Conversation with Data Scientist

- Build in automatic checks on the ingested data
- Know about changes or disruptions before they happen

Problems with schema changes & other anomalies in the data

Software Engineers

Data Engineer

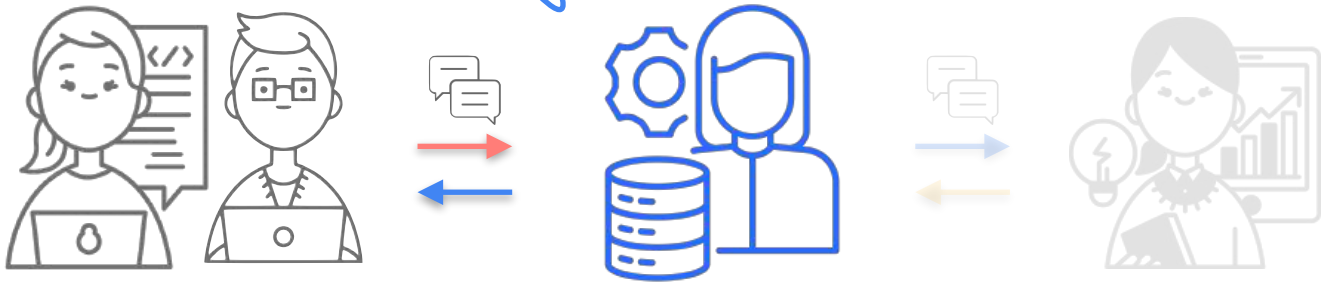
Data Scientist

Conversation with Source Owners

What's a better ingestion solution?

What kind of disruptions or changes can I expect?

How can I anticipate these changes?



Software Engineers

Data Engineer

Data Scientist

Conversation with Data Scientist

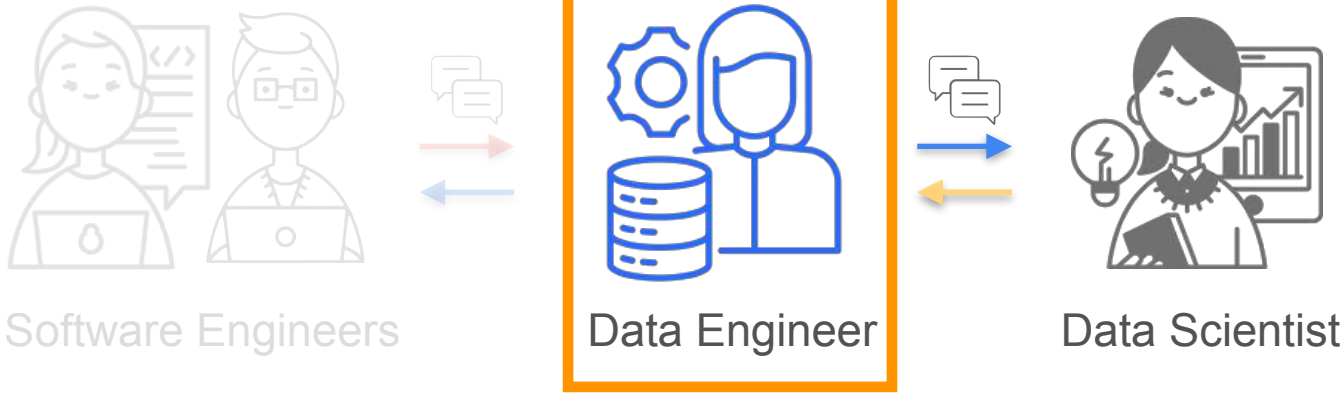
Functional Requirement

Ingest, transform, & serve data in the format required

Non-Functional Requirement

Make data available some time after it is recorded

Lots of data cleaning & processing



Software Engineers

Data Engineer

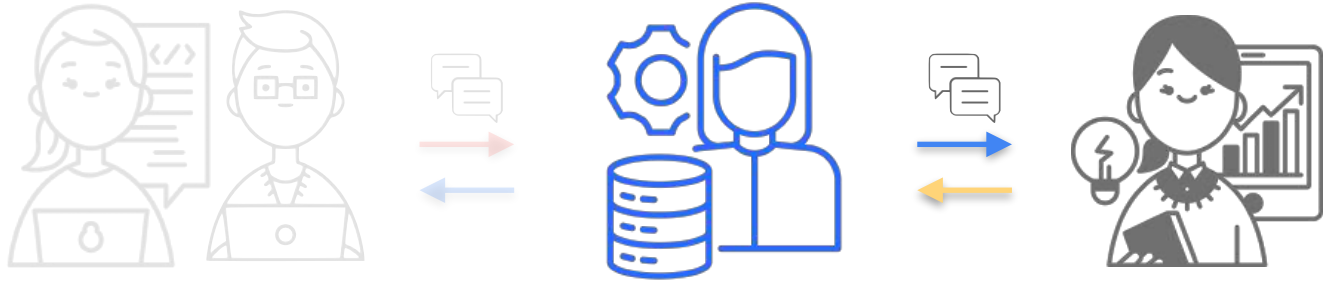
Data Scientist

Conversation with Data Scientist

What is “real-time”?

- monthly basis
- daily, hourly, minutes, seconds, ….

The marketing team needs the data in real-time



Software Engineers

Data Engineer

Data Scientist

Conversation with Data Scientist

Key Tactic:

Ask stakeholder what action they plan to take with the data

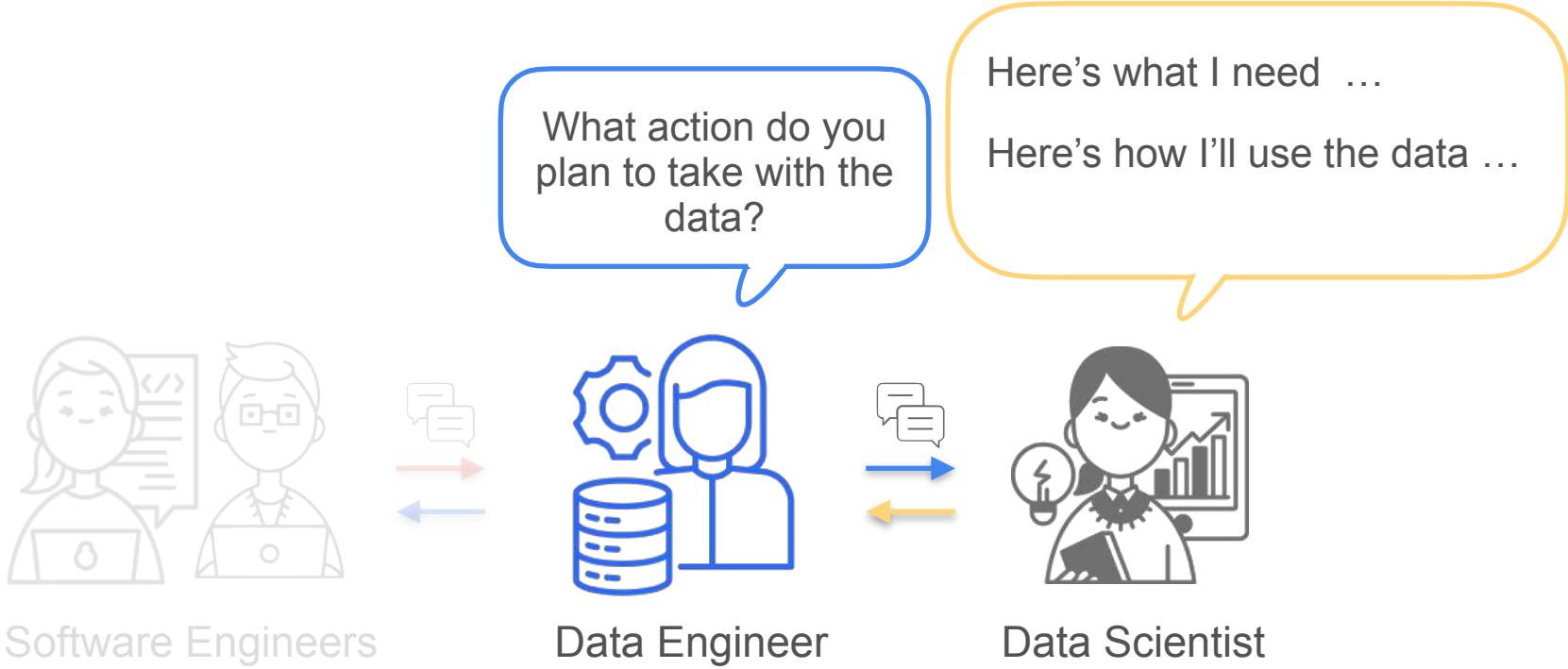
Not the same as asking what they need!

Software Engineers

Data Engineer

Data Scientist

Conversation with Data Scientist



Conversation with Data Scientist

A recommender system

Provide product recommendations in near real-time

An analytics dashboard

?

Software Engineers

Data Engineer

Data Scientist

Conversation with Data Scientist

1. Learned about existing solutions and pain points
2. Started to identify some of your system requirements
3. Identified stakeholders to talk to:
   - Marketing team
   - Software engineering team

Software Engineers

Data Engineer

Data Scientist

DeepLearning.AI

Introduction to Data Engineering

Thinking Like a Data Engineer

Thinking Like a Data Engineer

1 2 3 4

Identify business goals & stakeholder needs

Define system requirements

Choose tools & technologies

Build, evaluate, iterate & evolve

1. Identify business goals & stakeholders you will serve
2. Explore existing systems and stakeholder needs
3. Ask stakeholders what actions they will take with the data product

Business Goals

Stakeholders' Needs

Thinking Like a Data Engineer

1 Identify business goals & stakeholder needs

2 **Define system requirements**

3 Choose tools & technologies

4 Build, evaluate, iterate & evolve

1. Identify business goals & stakeholders you will serve
2. Explore existing systems and stakeholder needs
3. Ask stakeholders what actions they will take with the data product

1. Translate stakeholder needs to functional requirements
2. Define non-functional requirements
3. Document and confirm requirements with stakeholders

Confirm with Stakeholders

Thinking Like a Data Engineer

1

Identify business goals & stakeholder needs

1. Identify business goals & stakeholders you will serve
2. Explore existing systems and stakeholder needs
3. Ask stakeholders what actions they will take with the data product

2

Define system requirements

1. Translate stakeholder needs to functional requirements
2. Define non-functional requirements
3. Document and confirm requirements with stakeholders

3

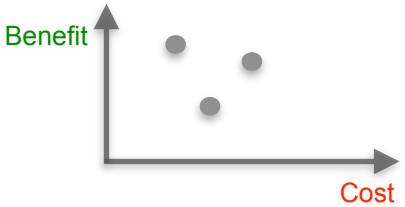
Choose tools & technologies

1. Identify tools & tech to meet non-functional requirements
2. Perform cost / benefit analysis and choose between comparable tools & tech
3. Prototype and test your system, align with stakeholder needs

4

Build, evaluate, iterate & evolve

Cost-Benefit Analysis



Thinking Like a Data Engineer

Identify business goals & stakeholder needs

1. Identify business goals & stakeholders you will serve
2. Explore existing systems and stakeholder needs
3. Ask stakeholders what actions they will take with the data product

Define system requirements

1. Translate stakeholder needs to functional requirements
2. Define non-functional requirements
3. Document and confirm requirements with stakeholders

Choose tools & technologies

1. Identify tools & tech to meet non-functional requirements
2. Perform cost / benefit analysis and choose between comparable tools & tech
3. Prototype and test your system, align with stakeholder needs

Build, evaluate, iterate & evolve

1. Build & deploy your production data system
2. Monitor, evaluate, and iterate on your system to improve it
3. Evolve your system based on stakeholder needs

Thinking Like a Data Engineer

Identify business goals & stakeholder needs

1. Identify business goals & stakeholders you will serve
2. Explore existing systems and stakeholder needs
3. Ask stakeholders what actions they will take with the data product

Define system requirements

1. Translate stakeholder needs to functional requirements
2. Define non-functional requirements
3. Document and confirm requirements with stakeholders

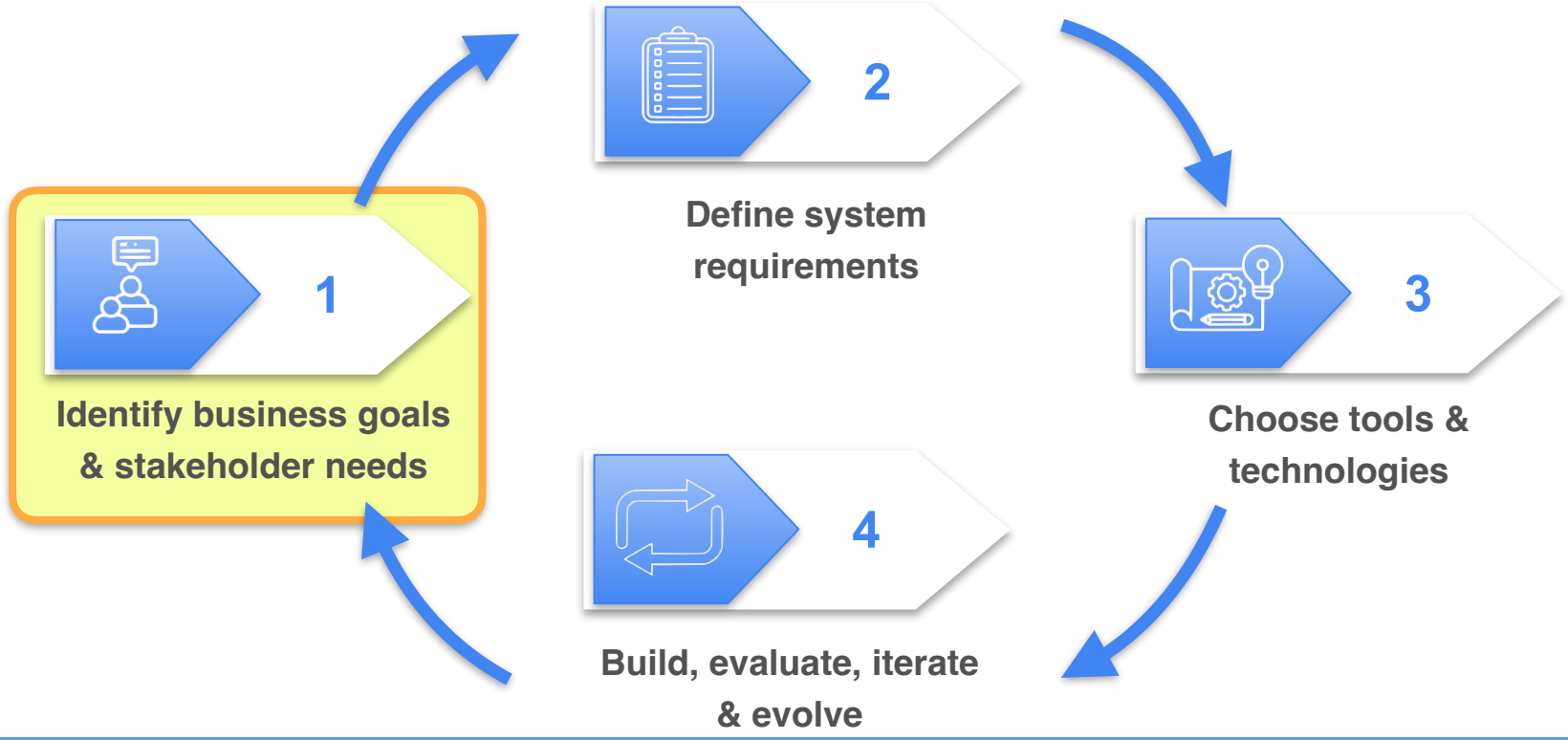
Choose tools & technologies

1. Identify tools & tech to meet non-functional requirements
2. Perform cost / benefit analysis and choose between comparable tools & tech
3. Prototype and test your system, align with stakeholder needs

Build, evaluate, iterate & evolve

1. Build & deploy your production data system
2. Monitor, evaluate, and iterate on your system to improve it
3. Evolve your system based on stakeholder needs

Thinking Like a Data Engineer



Thinking Like a Data Engineer

Identify business goals & stakeholder needs

1. Identify business goals & stakeholders you will serve
2. Explore existing systems and stakeholder needs
3. Ask stakeholders what actions they will take with the data product

Define system requirements

1. Translate stakeholder needs to functional requirements
2. Define non-functional requirements
3. Document and confirm requirements with stakeholders

Choose tools & technologies

1. Identify tools & tech to meet non-functional requirements
2. Perform cost / benefit analysis and choose between comparable tools & tech
3. Prototype and test your system, align with stakeholder needs

Build, evaluate, iterate & evolve

1. Build & deploy your production data system
2. Monitor, evaluate, and iterate on your system to improve it
3. Evolve your system based on stakeholder needs

DeepLearning.AI

Data Engineering in Practice

Data Engineering on the Cloud

Location

On-Premises

Migrating

Cloud

- Regulatory concerns
- Legacy systems

Hybrid

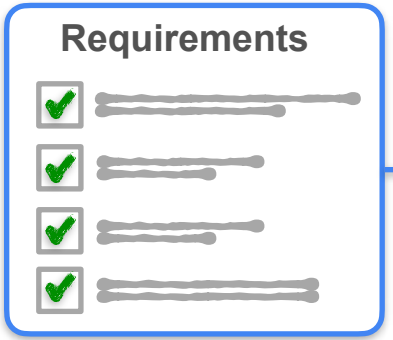
Public Cloud

Google Cloud Platform

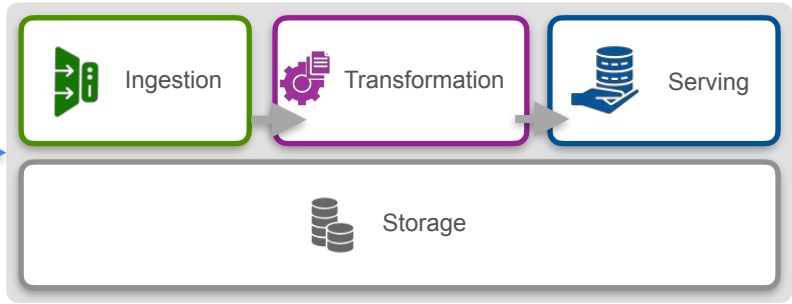
Microsoft Azure

Specialization Approach

Cloud-first approach



A Data Pipeline



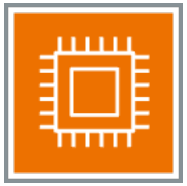
Just in time approach

DeepLearning.AI

Data Engineering in Practice

Intro to the AWS Cloud

IT Resources



Compute

Places to run code

Virtual Machine

Container hosting services

Serverless functions

Storage

Places to store data

Amazon Simple Storage Service (S3)

Amazon Elastic Block Store (EBS)

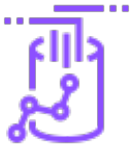
Database Services

Networking

Connect other resources to each other

Amazon Virtual Private Cloud (VPC)

Security



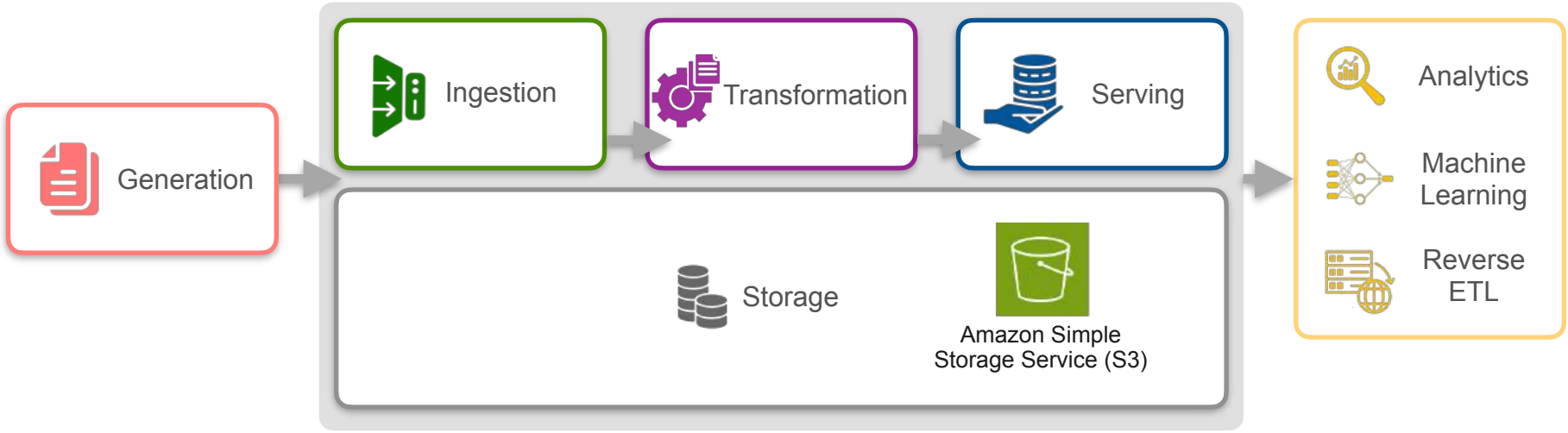
Data Streaming

Ingestion

Transformation

Advantage of Building on Cloud

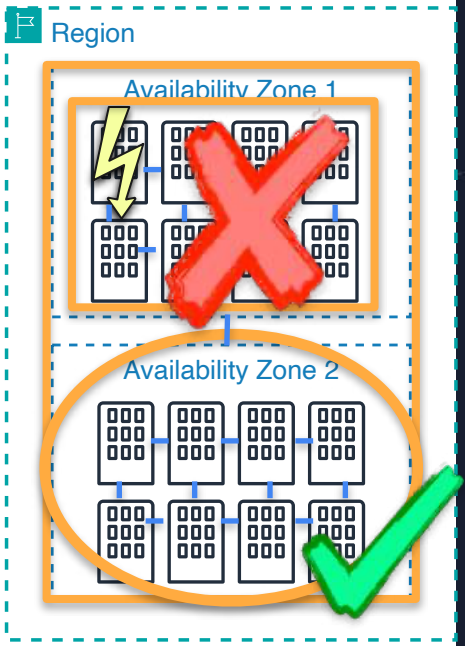
- Cloud resources are scalable and elastic.



- No need to worry about the exact storage capacity needed
- No need to manage the scaling operations

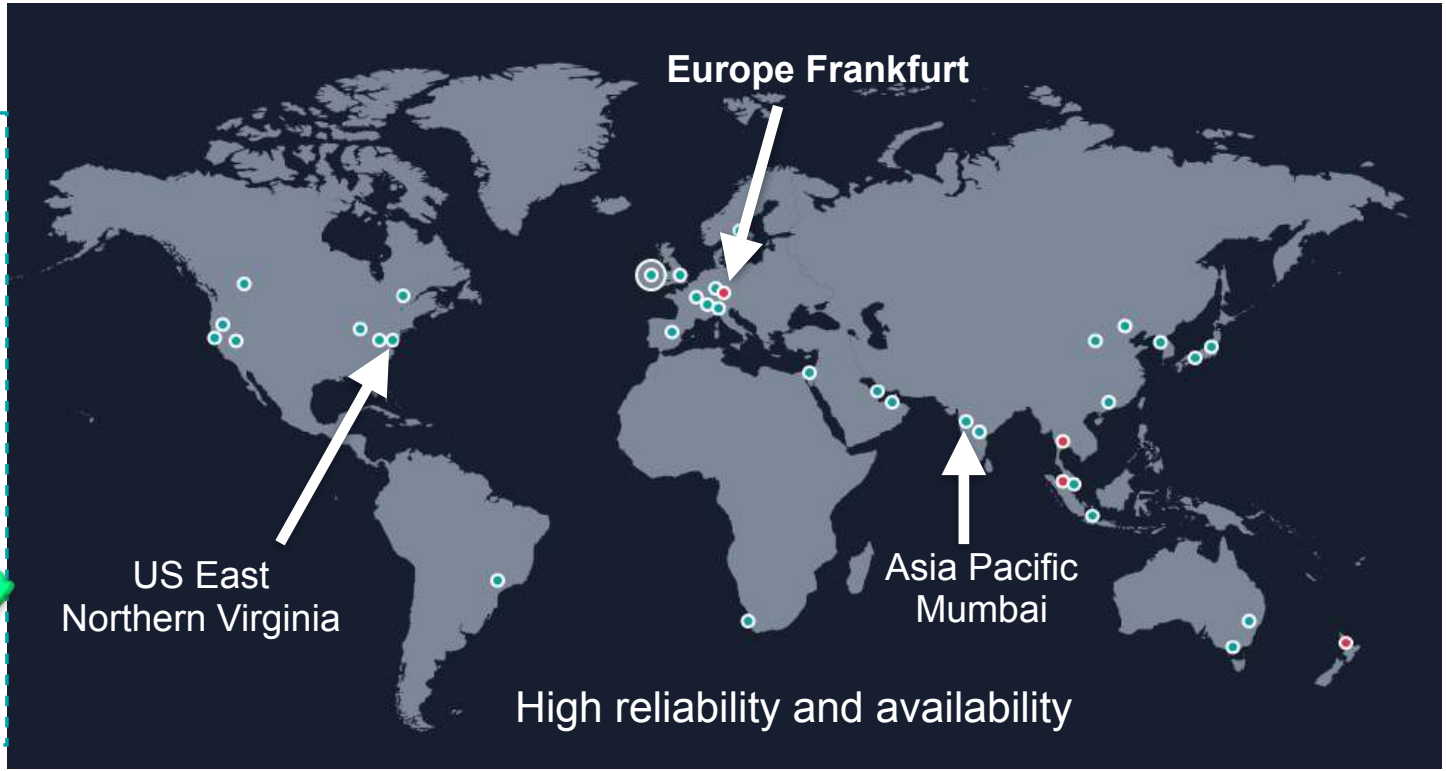
Data Center

Availability Zones



AWS Regions

Collections of data centers within geographical areas, where you can use AWS services

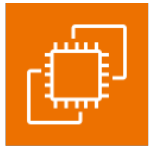


DeepLearning.AI

Data Engineering in Practice

Intro to AWS Core Services

Compute



Amazon Elastic Compute Cloud (EC2)

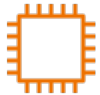
The service that provides virtual machines, or VMs, on AWS

Virtual Machines

Virtual computers or servers, where you can run any operating system and applications.

EC2 Instance

EC2 Instance



EC2 Instance

- You have complete control over an EC2 instance
- EC2 is a very flexible option for your workloads:
  - Use as a development machine for programming
  - Use to run a web server, container, or machine learning workload

AWS Lambda

Amazon Elastic Container Service (ECS)

Amazon Elastic Kubernetes Service (EKS)

Networking

Amazon Virtual Private Network (VPC)

The private network you can create and place resources into.

- VPCs are isolated from other networks.
- You choose the size of the private IP space.
- Partition space into smaller networks called **subnetworks or subnets**.
- Your data and resources don't leave the region unless you specifically build your solutions to behave that way

Storage

Object Storage — Most often used for storing unstructured data

Amazon Simple Storage Service (S3)

Block Storage — Used for database storage, virtual machine file systems, and other low-latency environments

Amazon Elastic Block Store (EBS)

File Storage — Data is organized into files and directories in a hierarchical structure

Amazon Elastic File System (EFS)

Storage

Amazon Relational Database Service (RDS)

A cloud-based relational database service

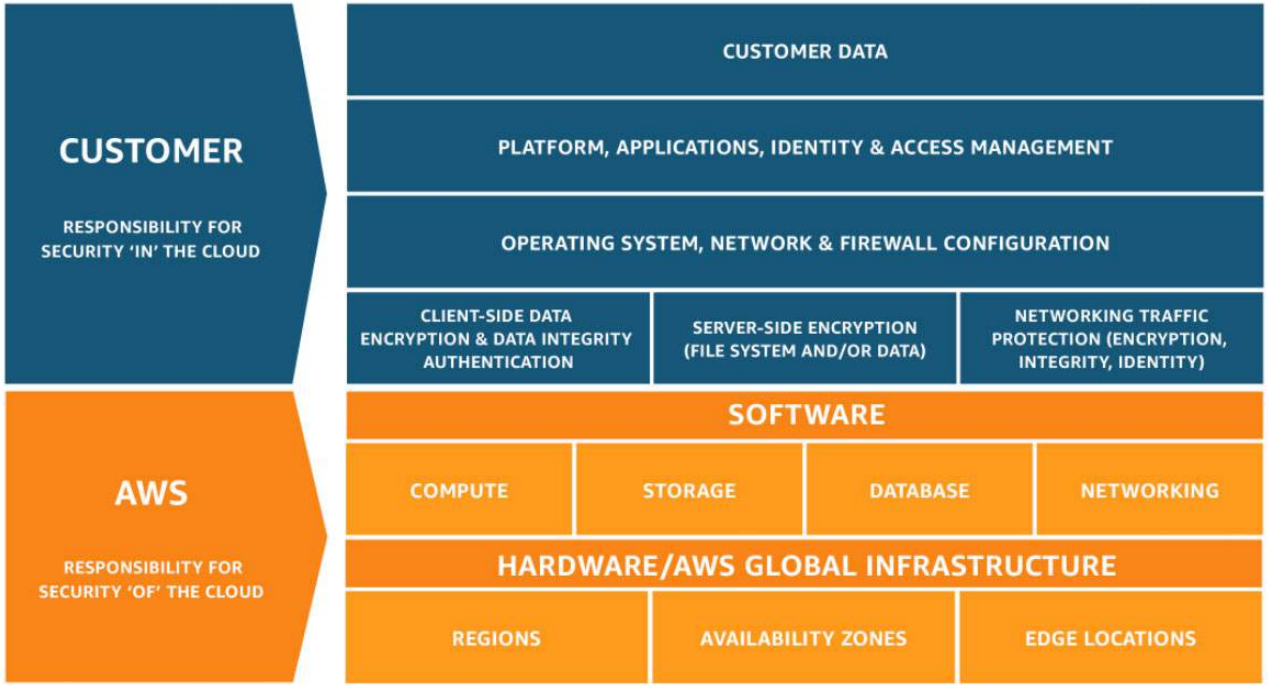
Amazon Redshift

A data warehouse service that allows you store, transform, and serve data for end use cases

Security

Shared Responsibility Model

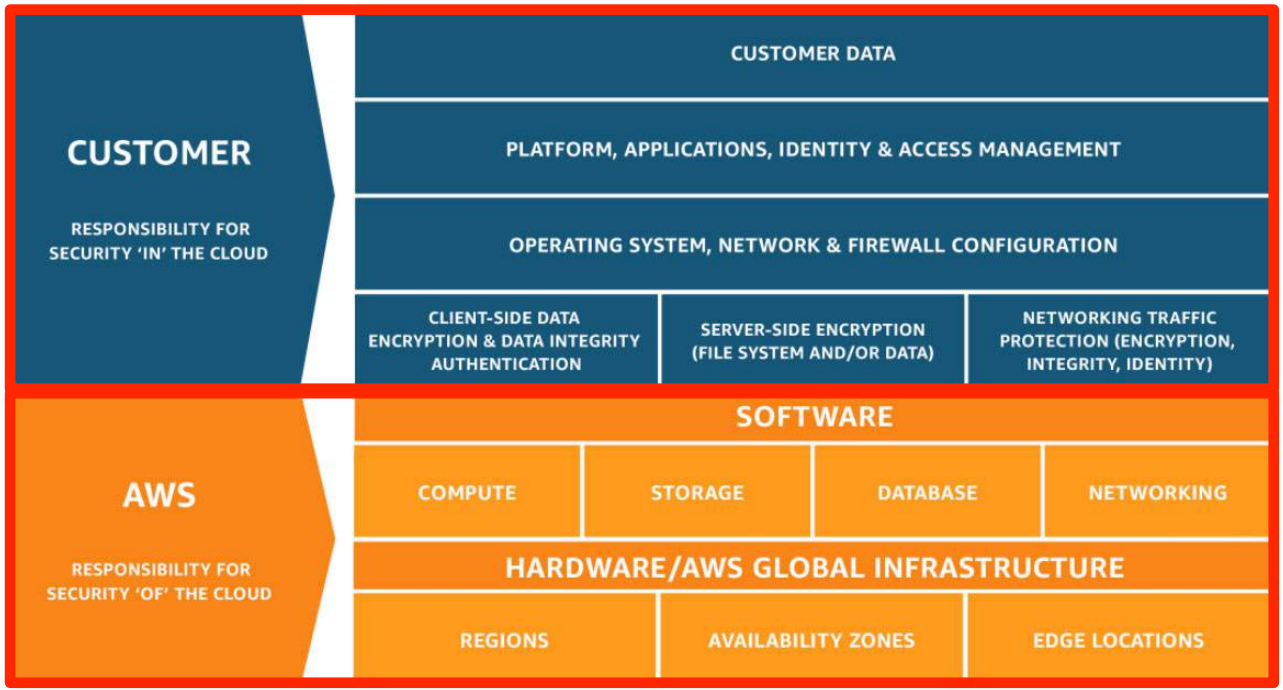
AWS is responsible for security **OF** the cloud, and you are responsible for security **IN** the cloud



Security

Shared Responsibility Model

AWS is responsible for security **OF** the cloud, and you are responsible for security **IN** the cloud



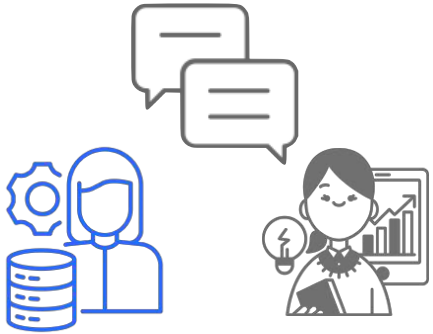
DeepLearning.AI

Introduction to Data Engineering

Week 1 Summary

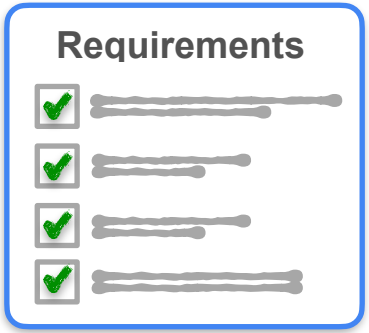
Week 1 Summary

1. Understand the needs of your stakeholders

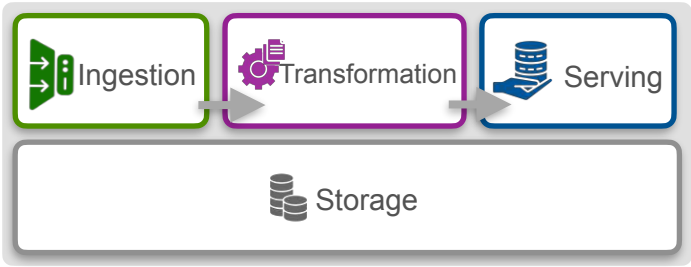


Data Engineer Data Scientist

2. Translate needs into system requirements



3. Choose appropriate tools & technologies



Value Created!

Thinking Like a Data Engineer

Identify business goals & stakeholder needs

1. Identify business goals & stakeholders you will serve
2. Explore existing systems and stakeholder needs
3. Ask stakeholders what actions they will take with the data product

Define system requirements

1. Translate stakeholder needs to functional requirements
2. Define non-functional requirements
3. Document and confirm requirements with stakeholders

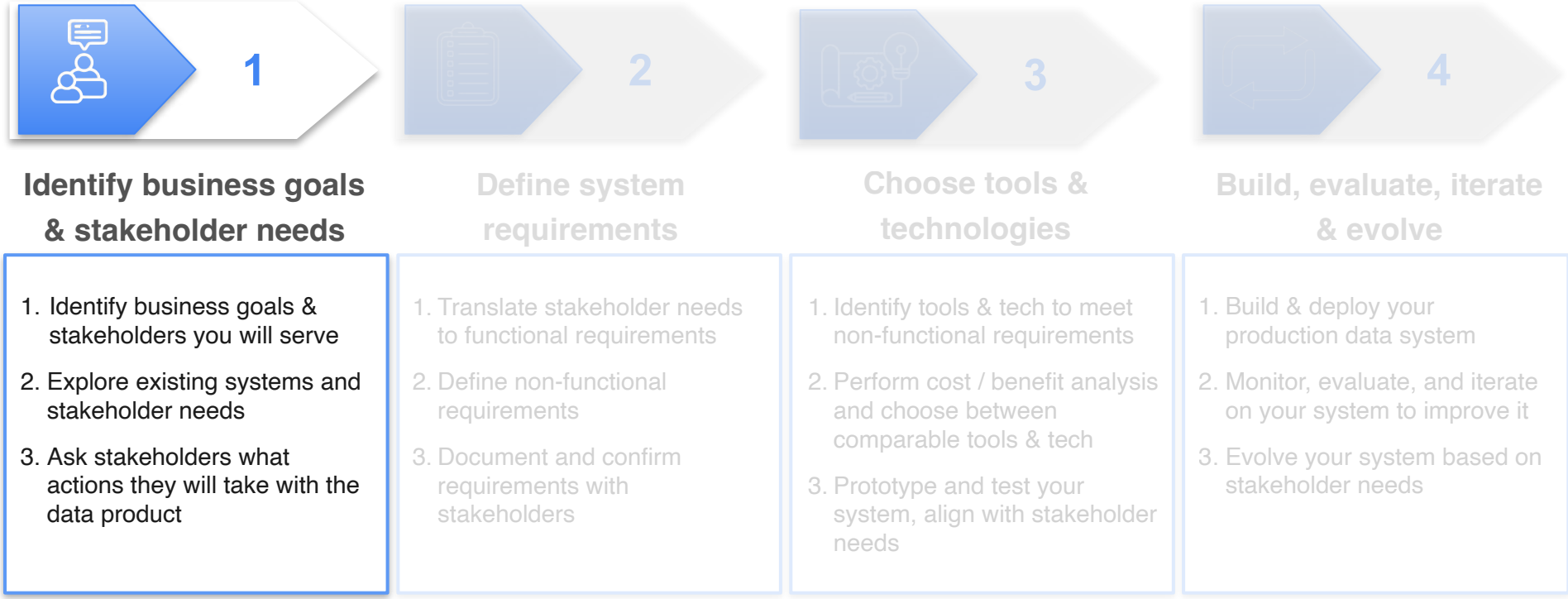
Choose tools & technologies

1. Identify tools & tech to meet non-functional requirements
2. Perform cost / benefit analysis and choose between comparable tools & tech
3. Prototype and test your system, align with stakeholder needs

Build, evaluate, iterate & evolve

1. Build & deploy your production data system
2. Monitor, evaluate, and iterate on your system to improve it
3. Evolve your system based on stakeholder needs

Thinking Like a Data Engineer



Thinking Like a Data Engineer

Identify business goals & stakeholder needs

1. Identify business goals & stakeholders you will serve
2. Explore existing systems and stakeholder needs
3. Ask stakeholders what actions they will take with the data product

Define system requirements

1. Translate stakeholder needs to functional requirements
2. Define non-functional requirements
3. Document and confirm requirements with stakeholders

Choose tools & technologies

1. Identify tools & tech to meet non-functional requirements
2. Perform cost / benefit analysis and choose between comparable tools & tech
3. Prototype and test your system, align with stakeholder needs

Build, evaluate, iterate & evolve

1. Build & deploy your production data system
2. Monitor, evaluate, and iterate on your system to improve it
3. Evolve your system based on stakeholder needs

Thinking Like a Data Engineer

Identify business goals & stakeholder needs

1. Identify business goals & stakeholders you will serve
2. Explore existing systems and stakeholder needs
3. Ask stakeholders what actions they will take with the data product

Define system requirements

1. Translate stakeholder needs to functional requirements
2. Define non-functional requirements
3. Document and confirm requirements with stakeholders

Choose tools & technologies

1. Identify tools & tech to meet non-functional requirements
2. Perform cost / benefit analysis and choose between comparable tools & tech
3. Prototype and test your system, align with stakeholder needs

Build, evaluate, iterate & evolve

1. Build & deploy your production data system
2. Monitor, evaluate, and iterate on your system to improve it
3. Evolve your system based on stakeholder needs

Thinking Like a Data Engineer

Identify business goals & stakeholder needs

1. Identify business goals & stakeholders you will serve
2. Explore existing systems and stakeholder needs
3. Ask stakeholders what actions they will take with the data product

Define system requirements

1. Translate stakeholder needs to functional requirements
2. Define non-functional requirements
3. Document and confirm requirements with stakeholders

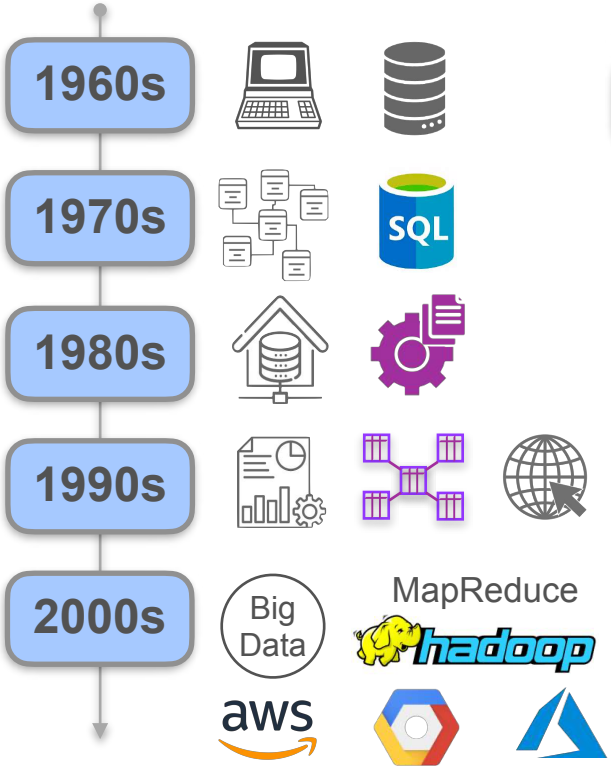
Choose tools & technologies

1. Identify tools & tech to meet non-functional requirements
2. Perform cost / benefit analysis and choose between comparable tools & tech
3. Prototype and test your system, align with stakeholder needs

Build, evaluate, iterate & evolve

1. Build & deploy your production data system
2. Monitor, evaluate, and iterate on your system to improve it
3. Evolve your system based on stakeholder needs

History of Data Engineering

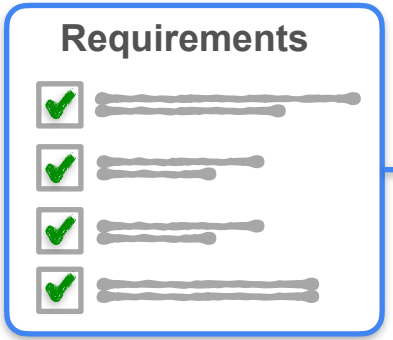


2010s - present

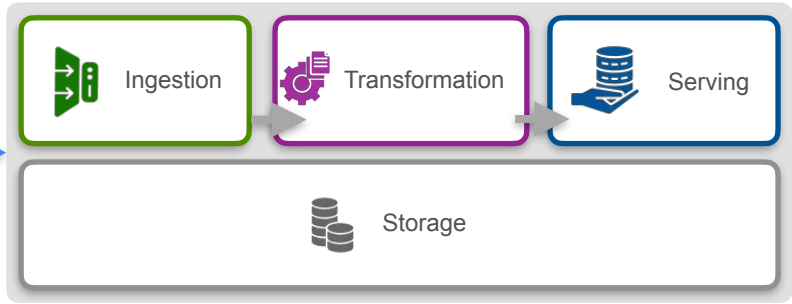
Abstract Simplified Tools

Specialization Approach

Cloud-first approach



A Data Pipeline



Just in time approach