

Copyright Notice

These slides are distributed under the Creative Commons License.

[DeepLearning.AI](#) makes these slides available for educational purposes. You may not use or distribute these slides for commercial purposes. You may make copies of these slides and use or distribute them for educational purposes as long as you cite [DeepLearning.AI](#) as the source of the slides.

For the rest of the details of the license, see <https://creativecommons.org/licenses/by-sa/2.0/legalcode>



DeepLearning.AI

Introduction to Data Engineering

Week 4



DeepLearning.AI

Stakeholder Management and Gathering Requirements

Week 4 Overview

Requirements Gathering



Software Engineers



Data Engineer



Data Scientist



Marketing Team

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



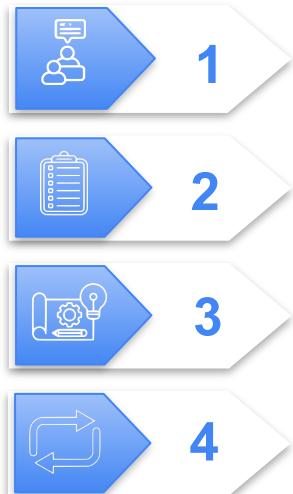
4

Build, evaluate, iterate & evolve

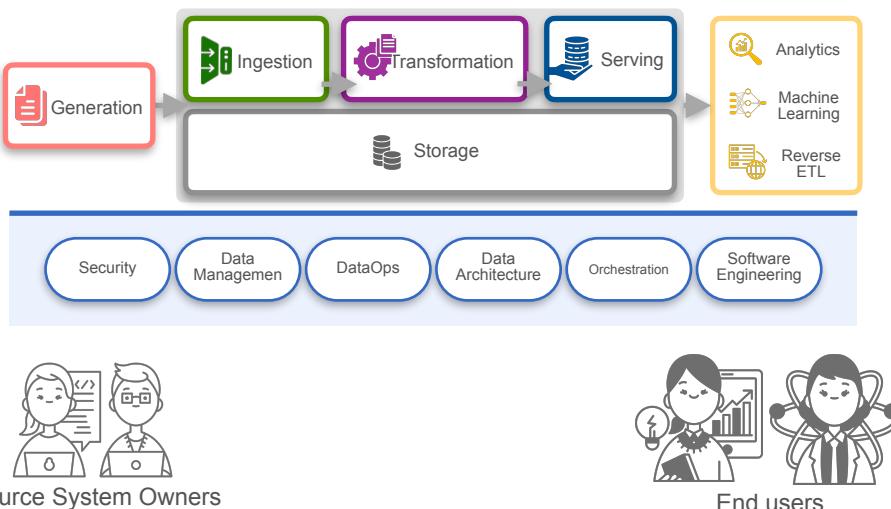
1. Build & deploy your production data system
2. Monitor, evaluate, and iterate on your system to improve it
3. Evolve your system based on stakeholder needs

The Complete Mental Model

Week 1: Thinking Like a Data Engineer



Week 2 : Data Engineering Lifecycle & Undercurrents



Week 3: Principles of Good Data Architecture

1. Choose common components wisely
2. Plan for failure!
3. Architect for Scalability
4. Architecture is leadership
5. Always be Architecting
6. Build loosely coupled systems
7. Make reversible decisions
8. Prioritize Security
9. Embrace FinOps

The Complete Mental Model

Week 4: Practical on-the-job scenario



Data Engineer

Week 1:
**Thinking Like a Data
Engineer**

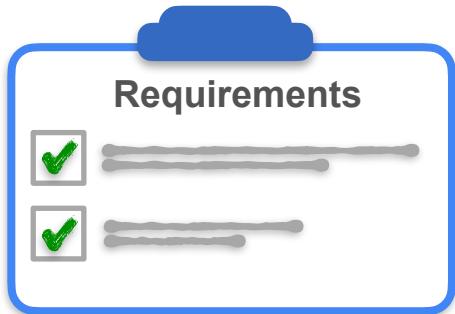
Week 2 :
**Data Engineering Lifecycle
& Undercurrents**

Week 3:
**Principles of Good
Data Architecture**

The Complete Mental Model

Week 4: Practical on-the-job scenario

Requirements Gathering



Functional requirements

Nonfunctional requirements

Choosing tools & technologies



Translate requirements into an architecture design

System Implementation



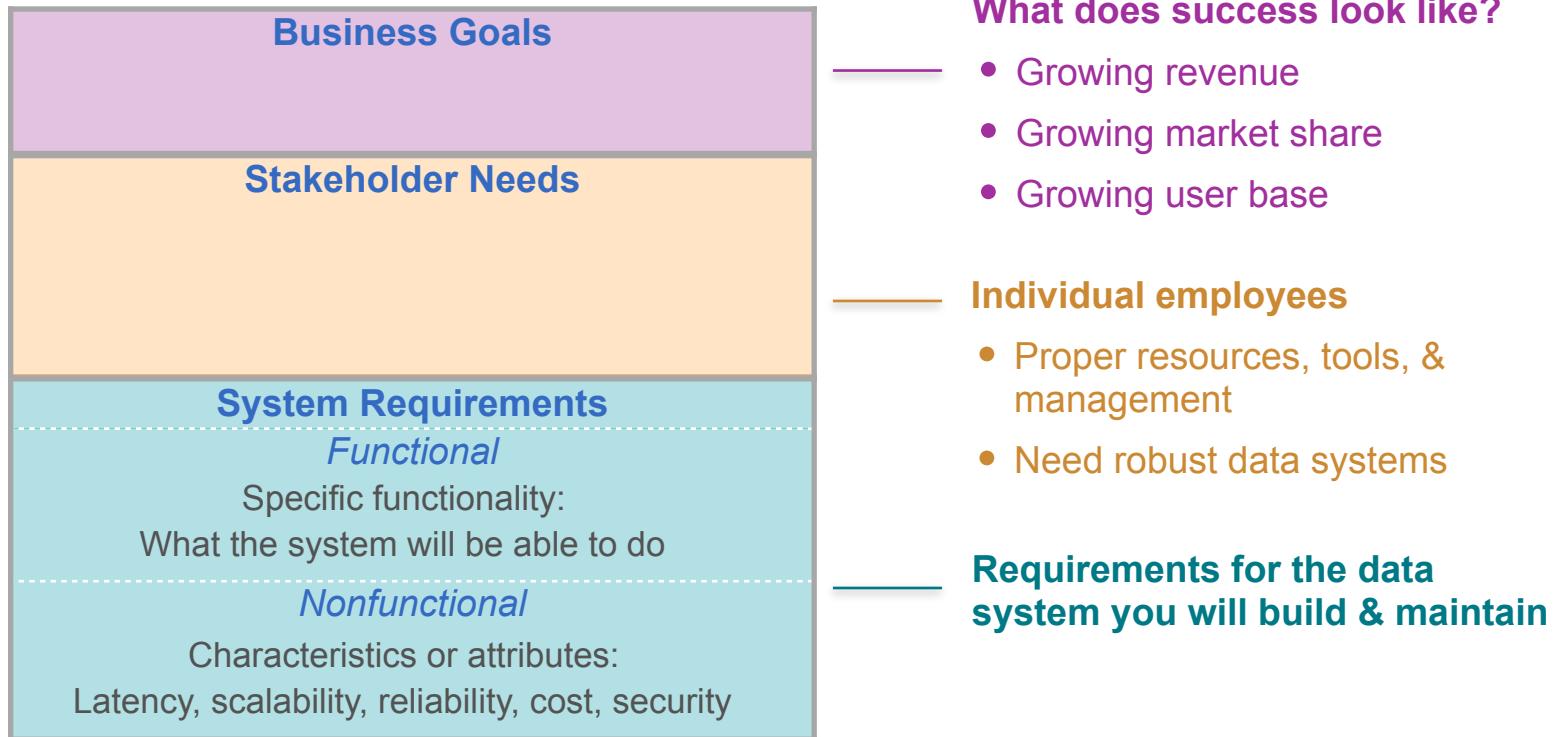


DeepLearning.AI

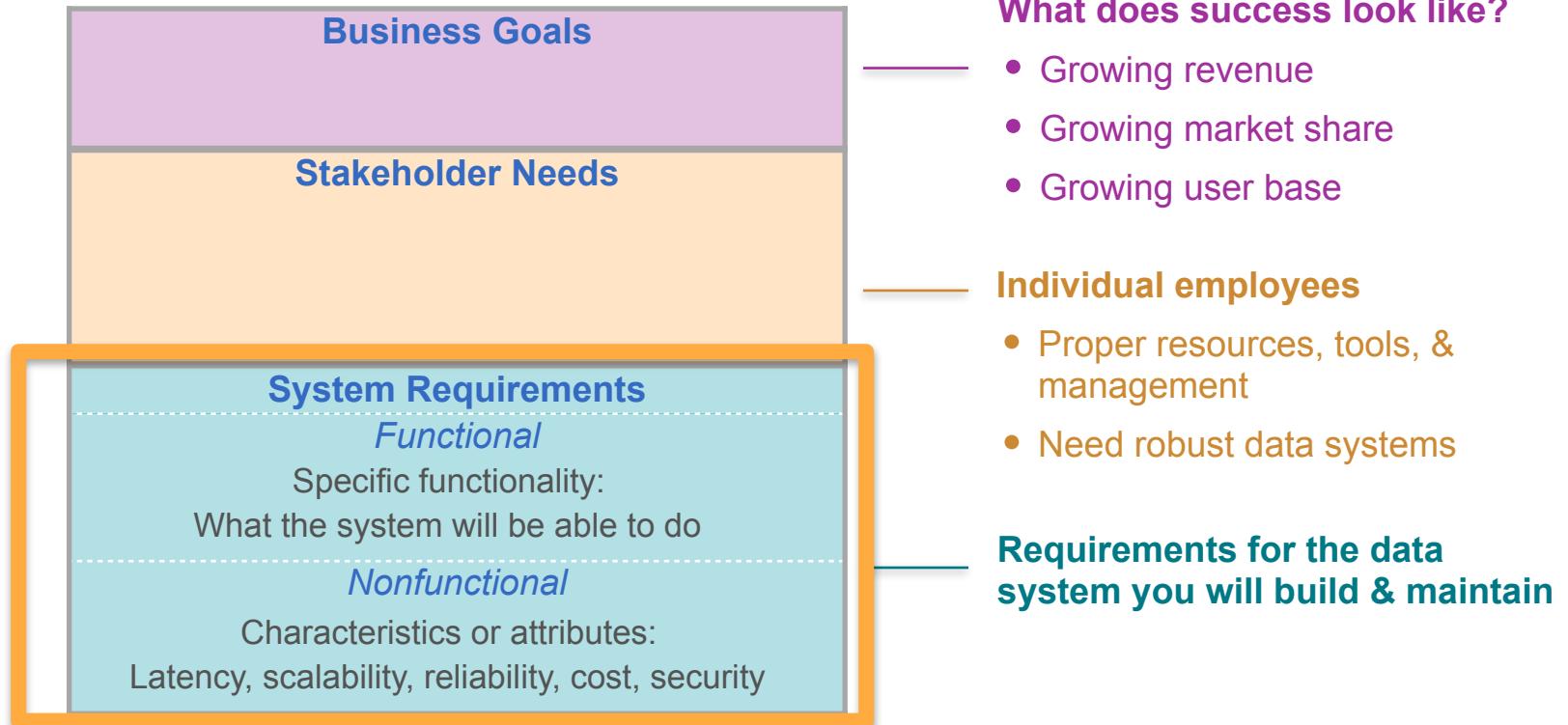
Stakeholder Management and Gathering Requirements

Requirements

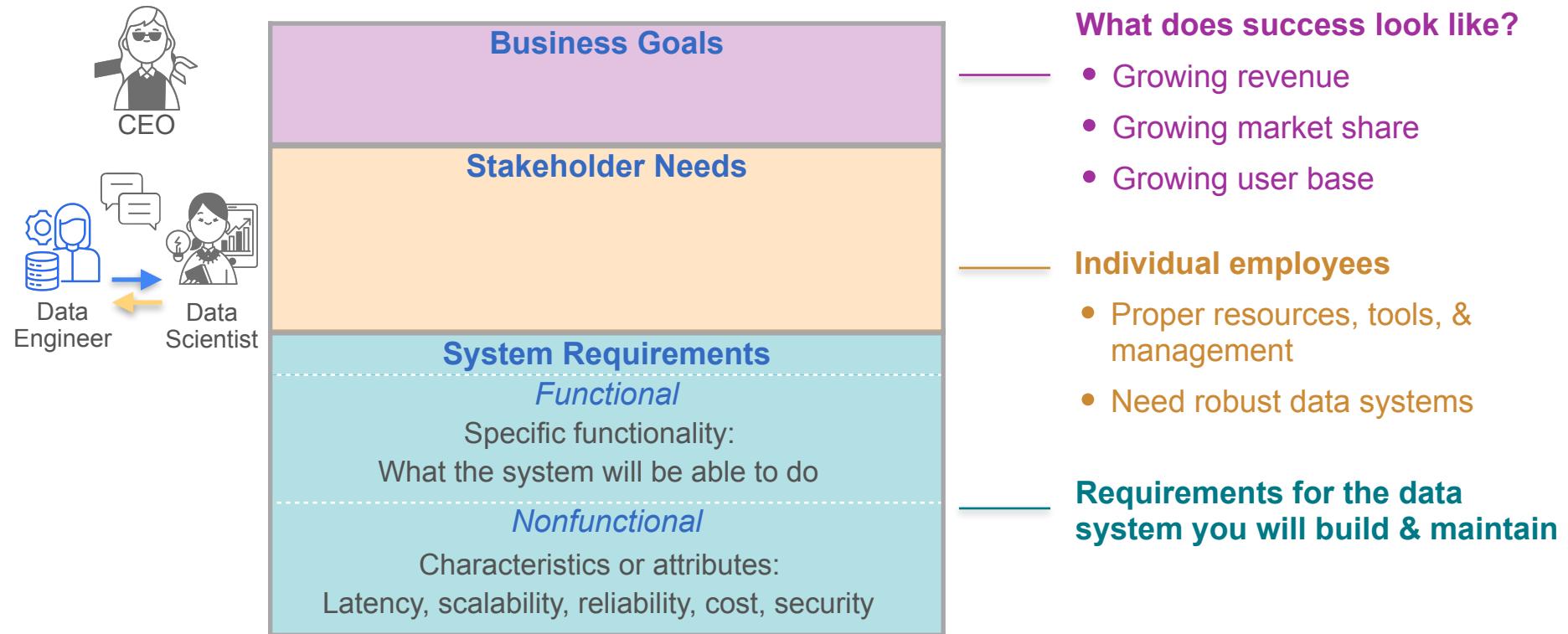
Hierarchy of Needs



Hierarchy of Needs



Requirements Gathering



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 based on the data you serve them.



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



DeepLearning.AI

Stakeholder Management and Gathering Requirements

Breaking Down the Conversation with Marketing

Documenting Requirements

Business Goals

Stakeholder Needs

System Requirements

Functional

Nonfunctional

Documenting Requirements

Business Goals

Continue on the growth trajectory:

Focus on customer retention and loyalty, expand to new markets and new product offerings

Stakeholder Needs

System Requirements

Functional

Nonfunctional

Documenting Requirements

Business Goals

Continue on the growth trajectory:

Focus on customer retention and loyalty, expand to new markets and new product offerings

Stakeholder Needs



System Requirements

Functional

Nonfunctional

Documenting Requirements

Business Goals

Continue on the growth trajectory:

Focus on customer retention and loyalty, expand to new markets and new product offerings

Stakeholder Needs

Analytics dashboard

Recommender System

System Requirements

Functional

Functional

Nonfunctional

Nonfunctional

Documenting Requirements

Business Goals

Continue on the growth trajectory:

Focus on customer retention and loyalty, expand to new markets and new product offerings



Stakeholder Needs

Analytics dashboard

Marketing needs to know about demand spikes, with hourly dashboard updates

Recommender System

System Requirements

Functional

The data system needs to serve transformed data that is no more than one hour old

Functional

Nonfunctional

Nonfunctional

Documenting Requirements

Business Goals

Continue on the growth trajectory:

Focus on customer retention and loyalty, expand to new markets and new product offerings



Stakeholder Needs

Analytics dashboard

Marketing needs to know about demand spikes, with hourly dashboard updates

Recommender System

Marketing needs a system that recommends products based on browsing or purchase history and current cart contents

System Requirements

Functional

The data system needs to serve transformed data that is no more than one hour old

Functional

The system needs to:

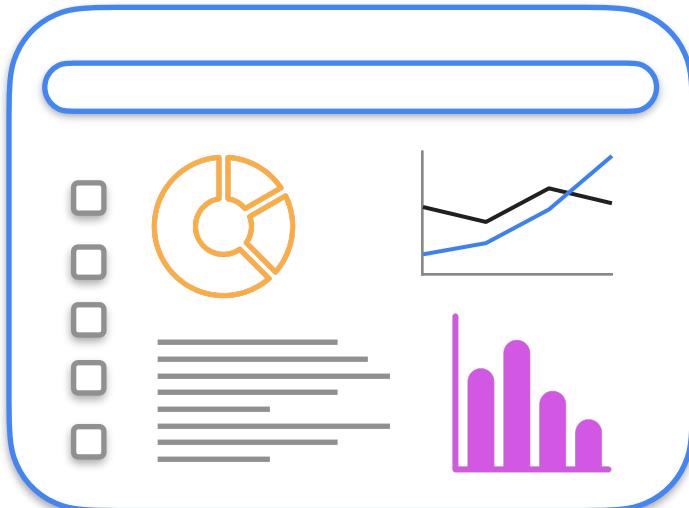
- Provide training data for recommender
- Ingest, transform, & serve user data to recommender
- Return model outputs back to sales platform

Nonfunctional

Nonfunctional

Functional Requirements - Analytics Dashboard

Functional Requirements: System serves needed data in a timely manner



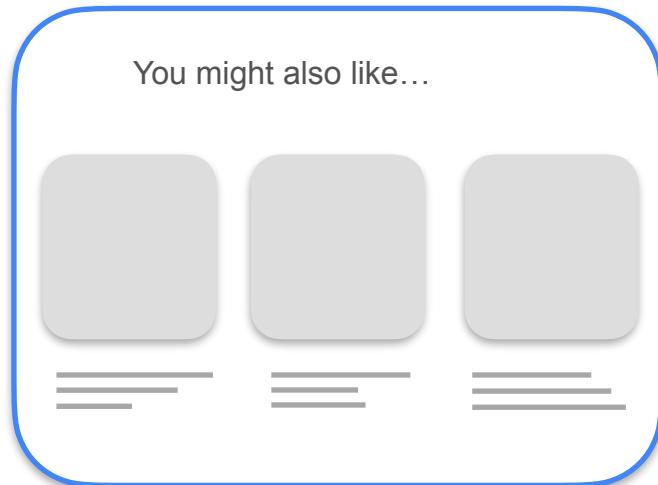
An analytics dashboard

Dashboard features or metrics to display:

- Responsibility of the data scientist
- Not functional requirements of your data system

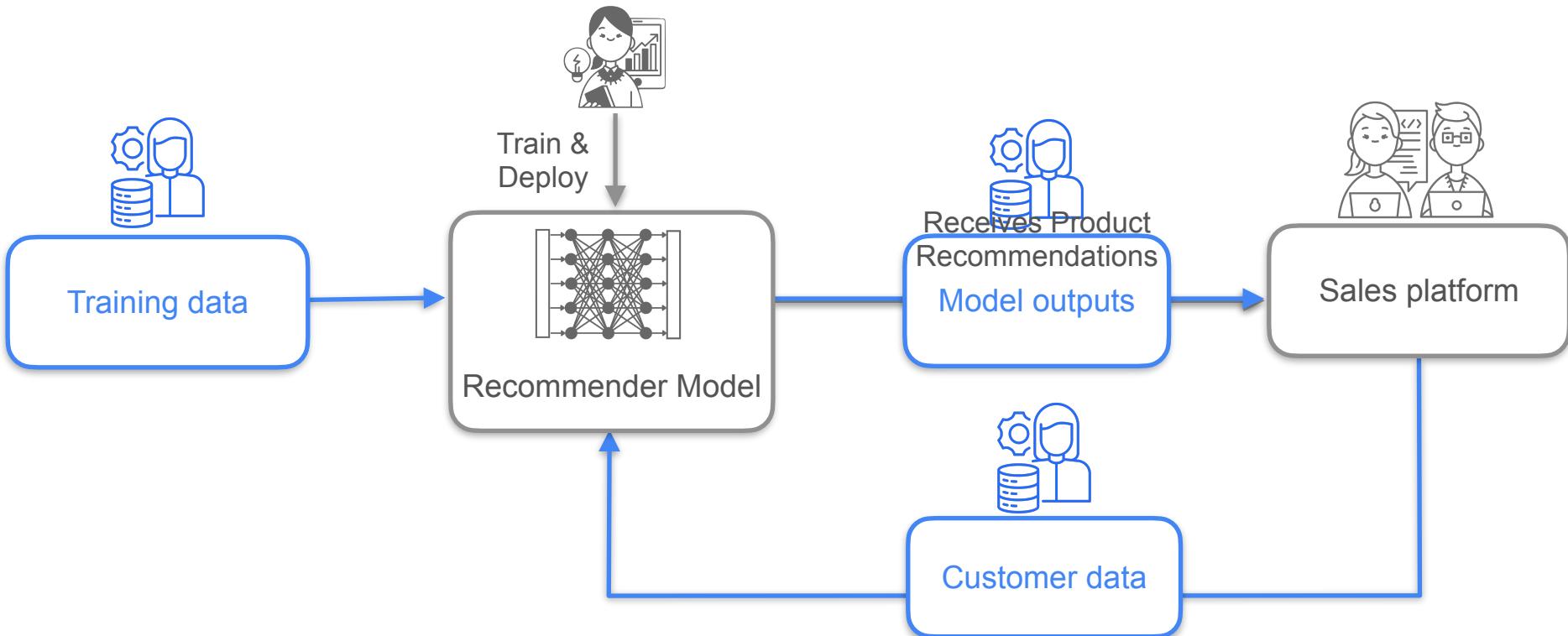
Functional Requirements - Recommender System

Functional Requirements



A recommender system

Functional Requirements - Recommender System



Documenting Requirements

Business Goals

Continue on the growth trajectory:

Focus on customer retention and loyalty, expand to new markets and new product offerings



Stakeholder Needs

Analytics dashboard

Marketing needs to know about demand spikes, with hourly dashboard updates

Recommender System

Marketing needs a system that recommends products based on browsing or purchase history and current cart contents

System Requirements

Functional

The data system needs to serve transformed data that is no more than one hour old

Functional

The system needs to:

- Provide training data for recommender
- Ingest, transform, & serve user data to recommender
- Return model outputs back to sales platform

Nonfunctional

Nonfunctional



DeepLearning.AI

Stakeholder Management and Gathering Requirements

Conversation with Software Engineering

When Talking to Source System Owners



Learn about existing data systems or solutions



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



Learn what actions stakeholders plan to take based on the data you serve them.



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



DeepLearning.AI

Stakeholder Management and Gathering Requirements

Documenting Nonfunctional Requirements

Conversation Takeaways



- Ensure a degree of stability for read replica
- Provide notifications for system outages or changes to the database schema

Documenting Requirements

Business Goals

Continue on the growth trajectory:

Focus on customer retention and loyalty, expand to new markets and new product offerings

Stakeholder Needs



Analytics dashboard

Marketing needs to know about demand spikes, with hourly dashboard updates



Recommender System

Marketing needs a system that recommends products based on browsing or purchase history and current cart contents

System Requirements

Functional

The data system needs to serve transformed data that is no more than one hour old

Functional

The system needs to:

- Provide training data for recommender
- Ingest, transform, & serve user data to recommender
- Return model outputs back to sales platform

Nonfunctional

Nonfunctional

An analytics dashboard



System Requirements

Functional

The data system needs to serve transformed data that is no more than one hour old

Nonfunctional

Scalability and Latency

System will be able to scale up to ingest, transform and serve the data volume expected with the maximum level of user activity, while staying within the latency requirements

Reliability

System will perform data quality checks to ensure data is conformant

Maintainability

The ingestion and transformation stages must be easily adaptable to accommodate any changes in the data schema

A recommender system

You might also like...



“Real-time”?

System Requirements

Functional

The system needs to:

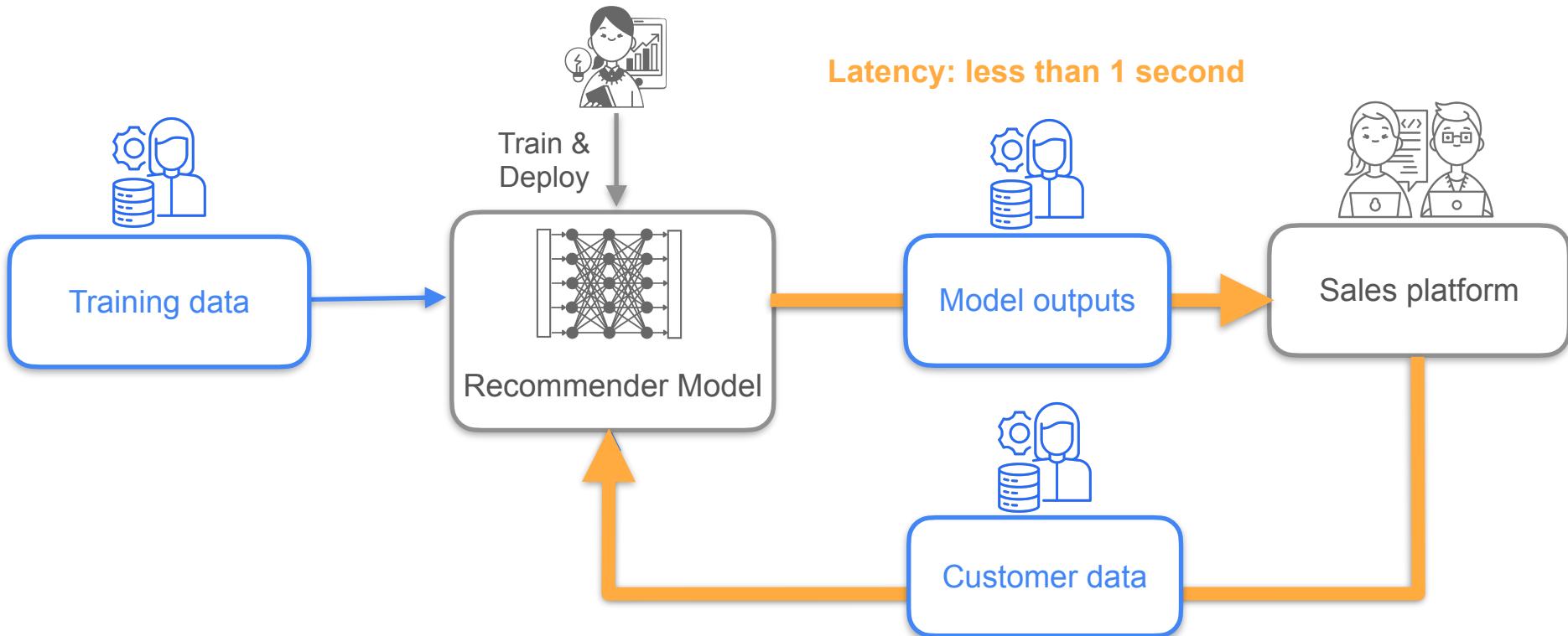
- Provide training data for recommender
- Ingest, transform, & serve user data to recommender
- Return model outputs back to sales platform

Nonfunctional

Latency

System must have a latency of less than 1 second from ingestion of user data to serving of recommendation data

Functional Requirements - Recommender System



A recommender system

You might also like...



System Requirements

Functional

The system needs to:

- Provide training data for recommender
- Ingest, transform, & serve user data to recommender
- Return model outputs back to sales platform

Nonfunctional

Latency

System must have a latency of less than 1 second from ingestion of user data to serving of recommendation data

Scalability

System must be able to scale up to the maximum number of concurrent users on the platform

Reliability

- System must return a set of recommendations within one second
- If the recommender pipeline fails it should default to serving a selection of the most popular products



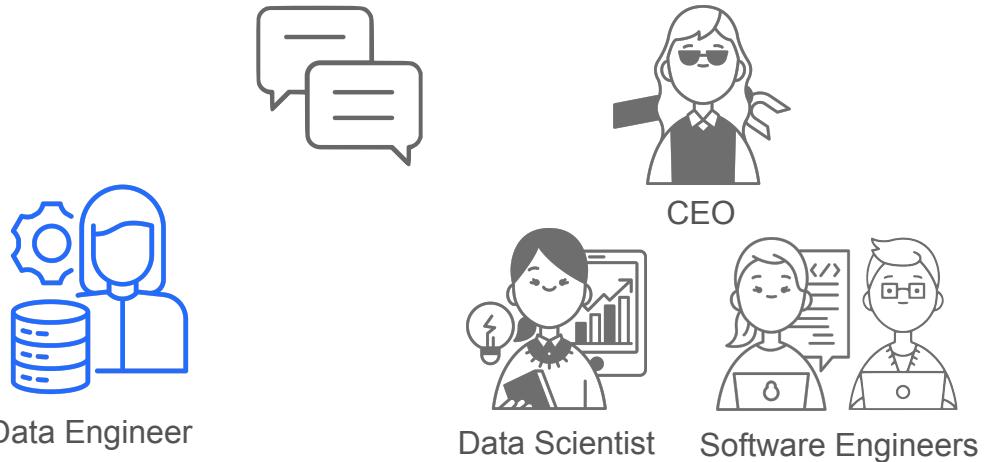
DeepLearning.AI

Stakeholder Management and Gathering Requirements

Requirements Gathering Summary

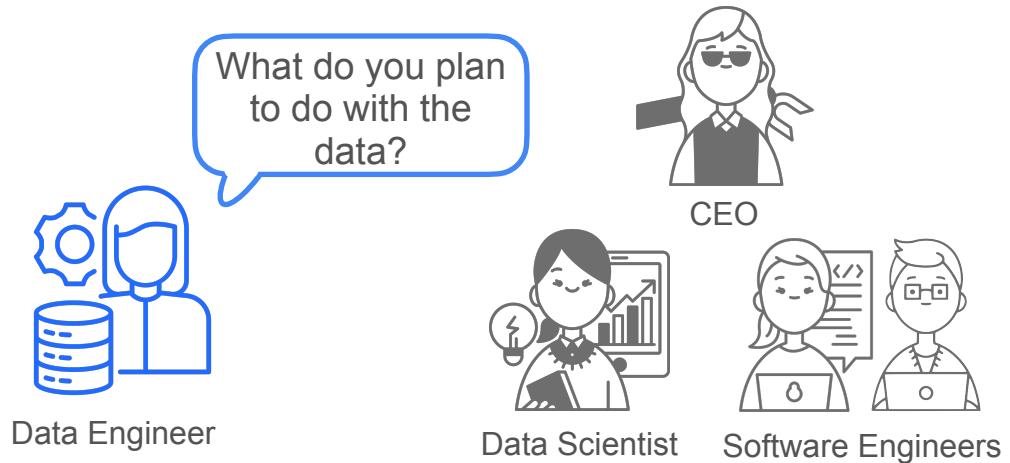
Main Takeaways

1. Identify the stakeholders, understand their needs and the broader goals of the business



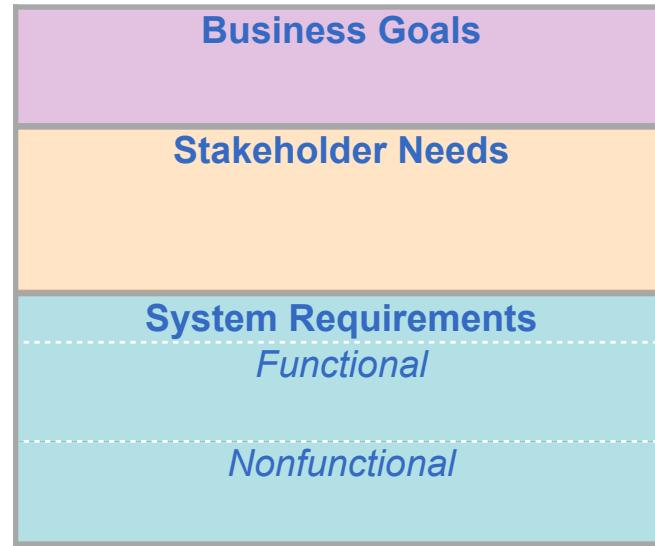
Main Takeaways

1. Identify the stakeholders, understand their needs and the broader goals of the business
2. Ask open-ended questions



Main Takeaways

1. Identify the stakeholders, understand their needs and the broader goals of the business
2. Ask open-ended questions
3. Document all of your findings



Evaluation of Trade Offs

Timeline

Stakeholders want you to build them a data system as quickly as possible

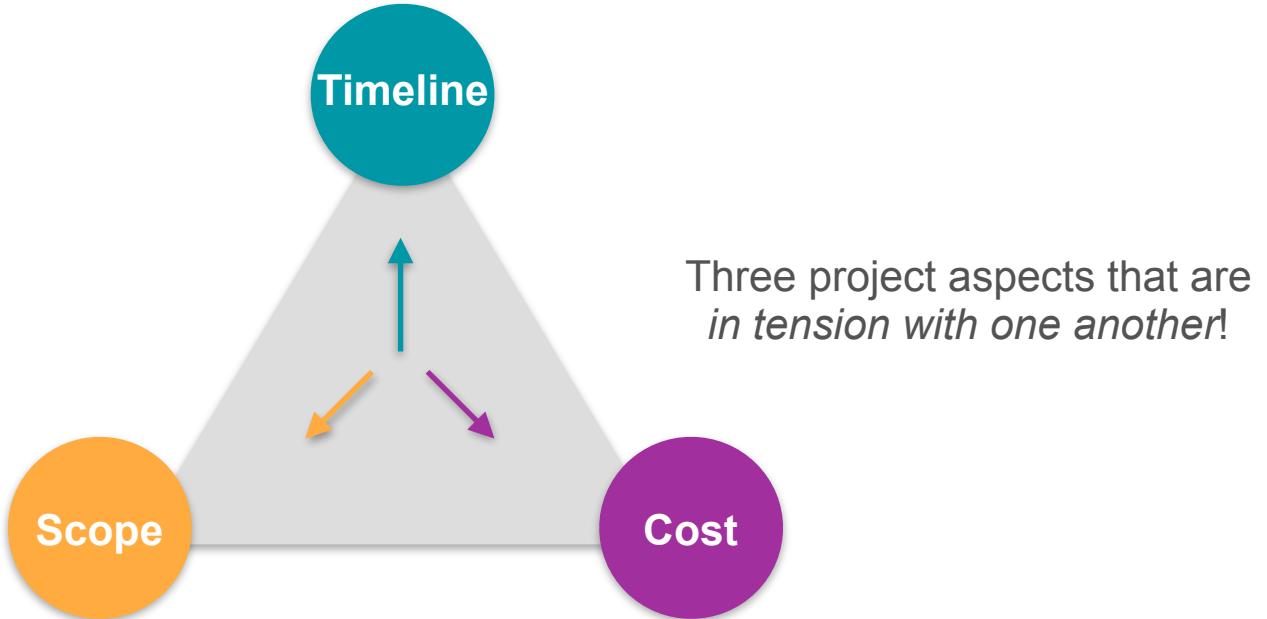
Cost

You might be working within a limited budget

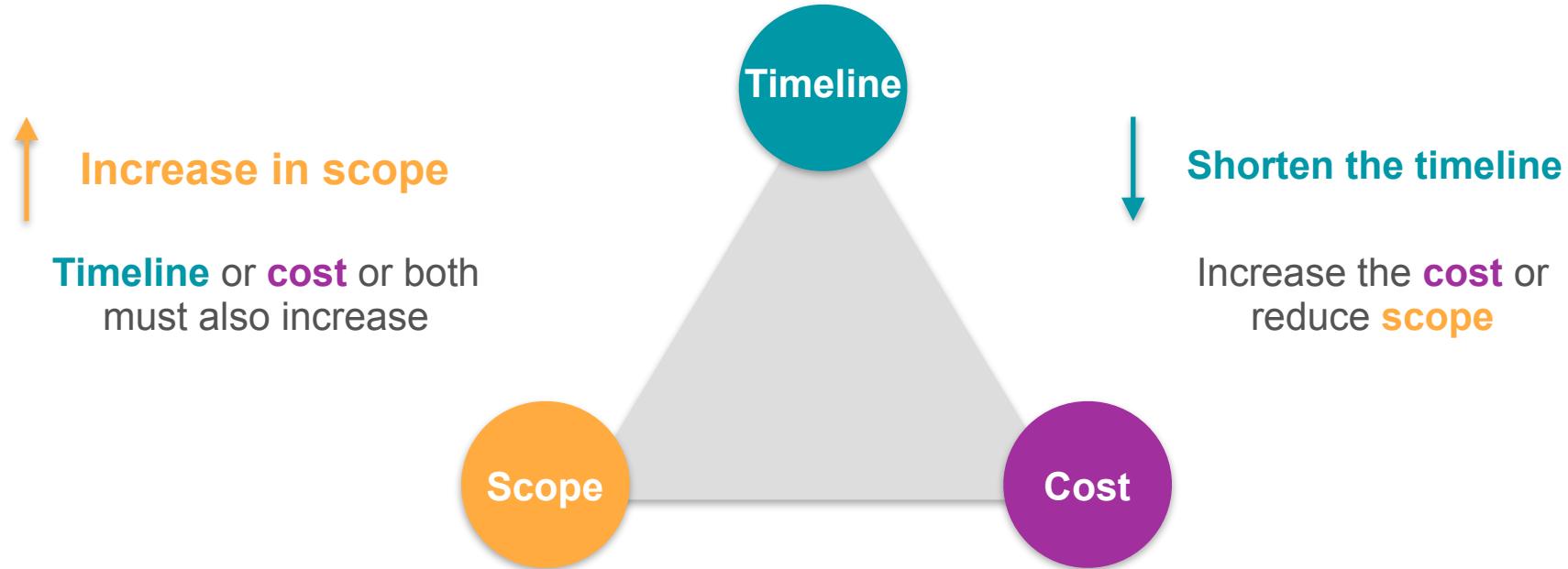
Scope

Features of the system

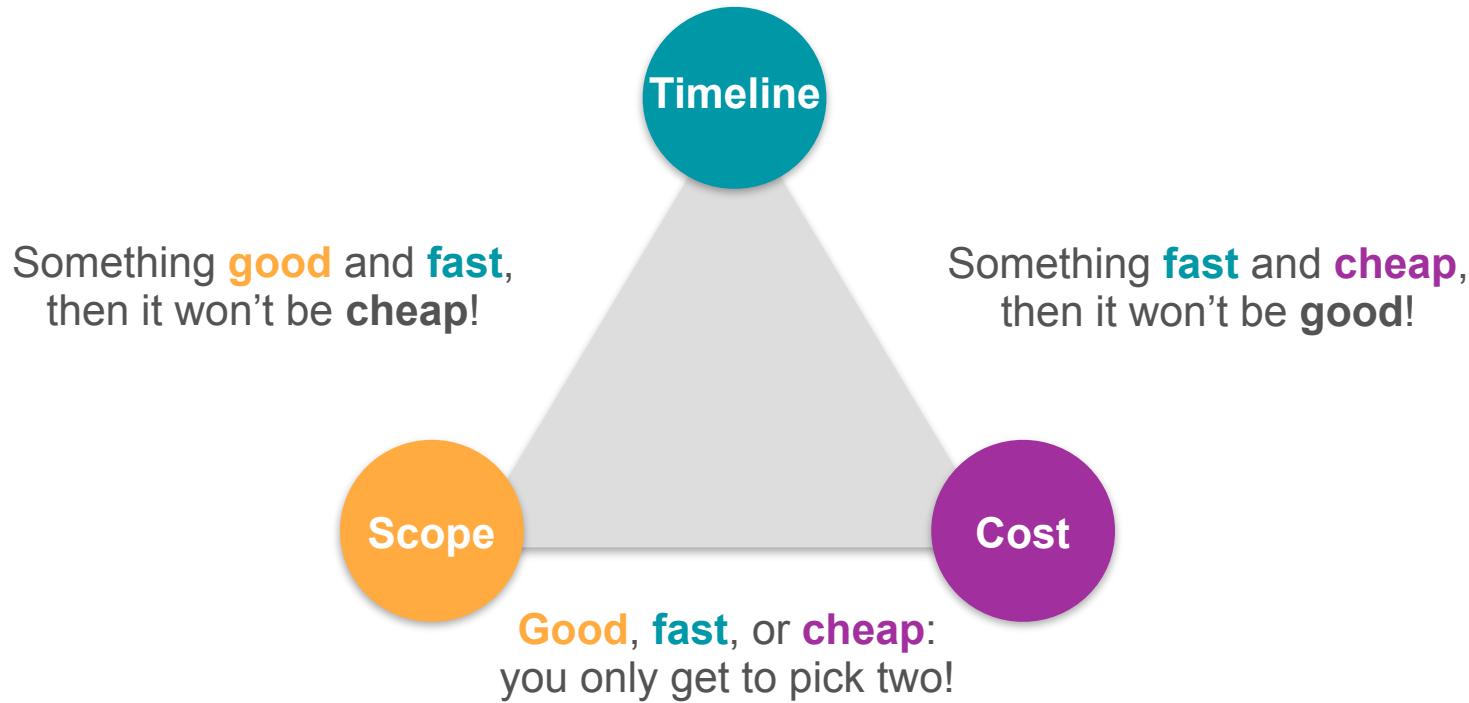
Iron Triangle



Iron Triangle

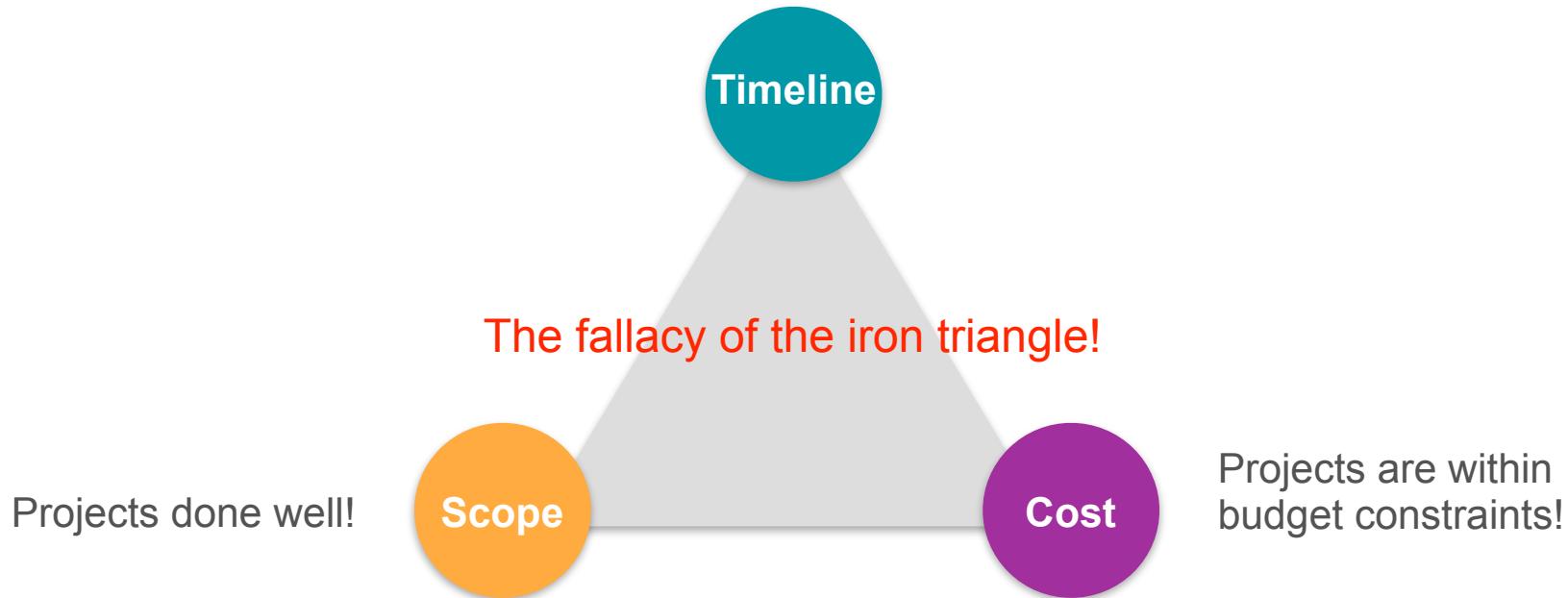


Iron Triangle



Iron Triangle

Projects done as quickly as possible!



Iron Triangle

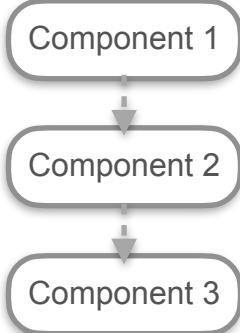
How to break the iron triangle?

Application of principles & processes

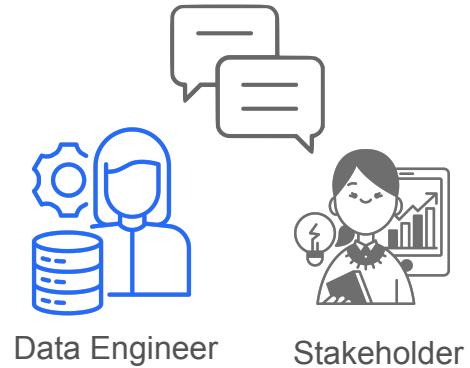
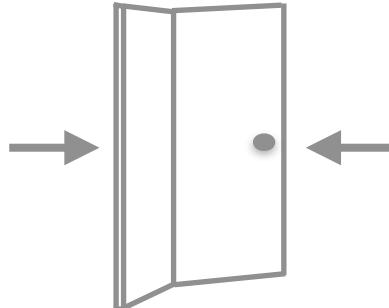
Loosely Coupled Components

Update:

New Component 2



Two-way





DeepLearning.AI

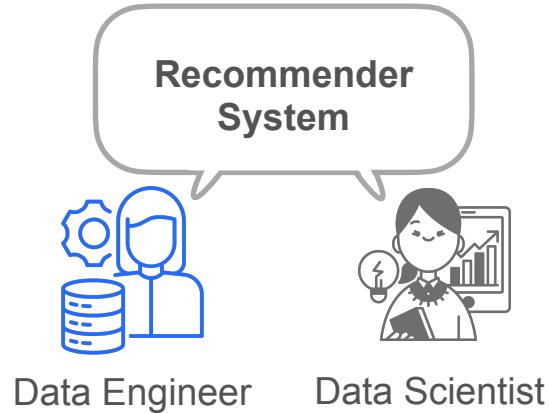
Translating Requirements to Architecture

Requirements Gathering Exercise

Lesson's Plan

1. Requirements gathering video

Conversation: data engineer & data scientist



Review Conversation

- Functional requirements
- Nonfunctional requirements

Lesson's Plan

1. Requirements gathering video

Conversation: data engineer & data scientist

2. Practice Quiz

Extract functional & nonfunctional requirements

3. Video with Morgan Willis

AWS tools & services

4. Quiz

Identify tools and services to meet requirements

5. Lab

- Recommender system has been set up for you
- You'll take steps to customize the data system



DeepLearning.AI

Translating Requirements to Architecture

AWS Services for Batch Pipelines

AWS Services for Batch Pipelines

Transport your team



Range	10,000 nmi
Speed	1100 km/hr
Capacity	250+ passengers
Cost	\$225,000,000

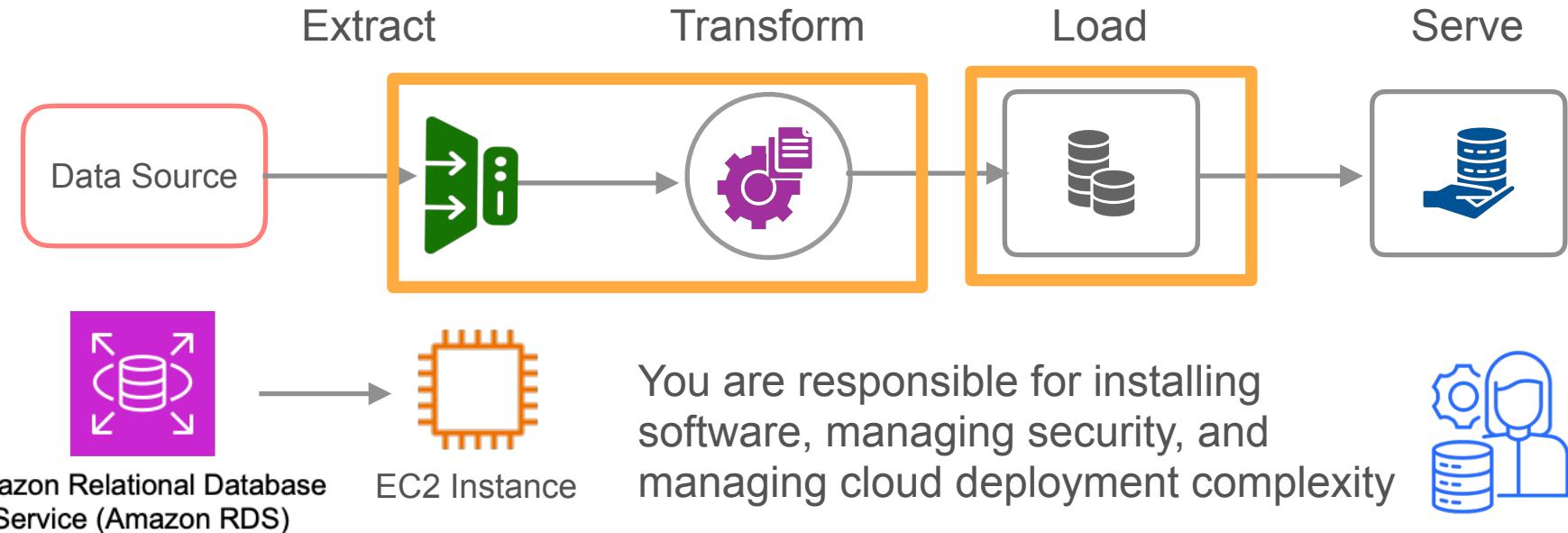


Range	360 km
Speed	320 km/hr
Capacity	4 passengers
Cost	\$70,000

- Where do you need to go?
- How big is your team?
- Is speed the most important?
- Are you on a tight budget?

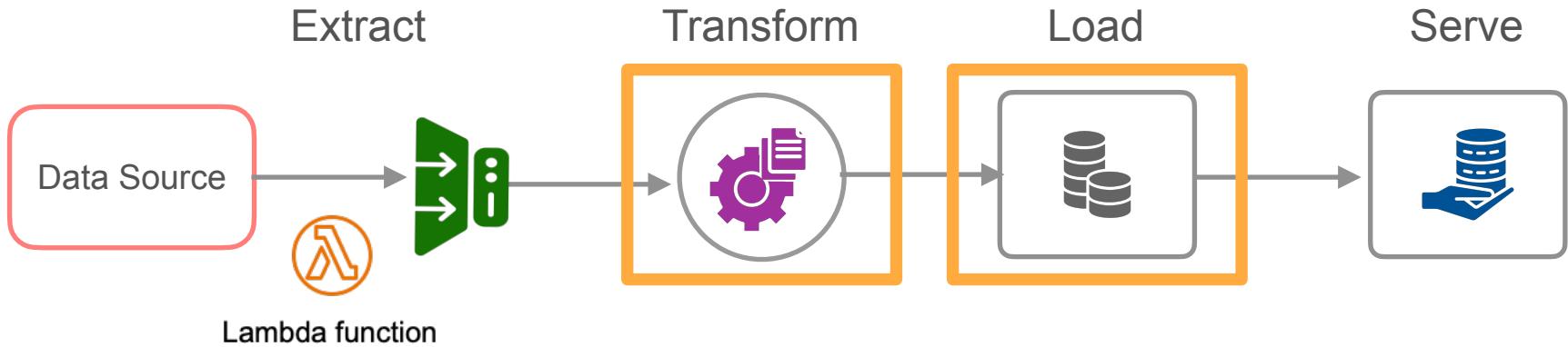
AWS Services for Batch Pipelines

Extract-Transform-Load (ETL) Pipeline



AWS Services for Batch Pipelines

Extract-Transform-Load (ETL) Pipeline



Limitations

- 15-minute timeout for each function call
- Memory and CPU allocation for each function
- Requires you to write custom code for your use case

Serverless Tools for Batch Processing

Difference: Tradeoffs between control vs convenience



Amazon EMR



AWS Glue ETL

More control

- Designed as a big data tool



More convenience

- Can handle big data with additional features



AWS Glue
Crawler

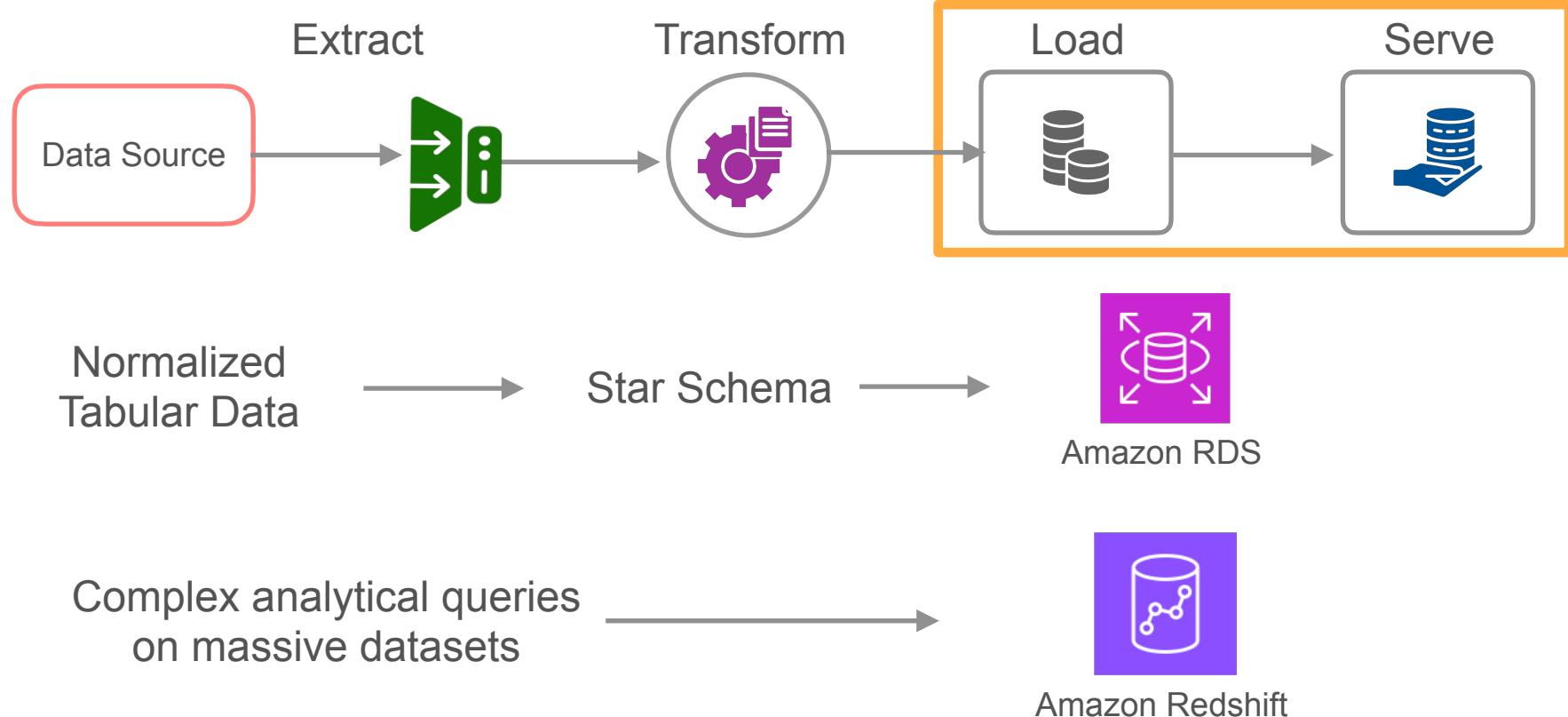
- Automatically discover & classify data
- Create metadata



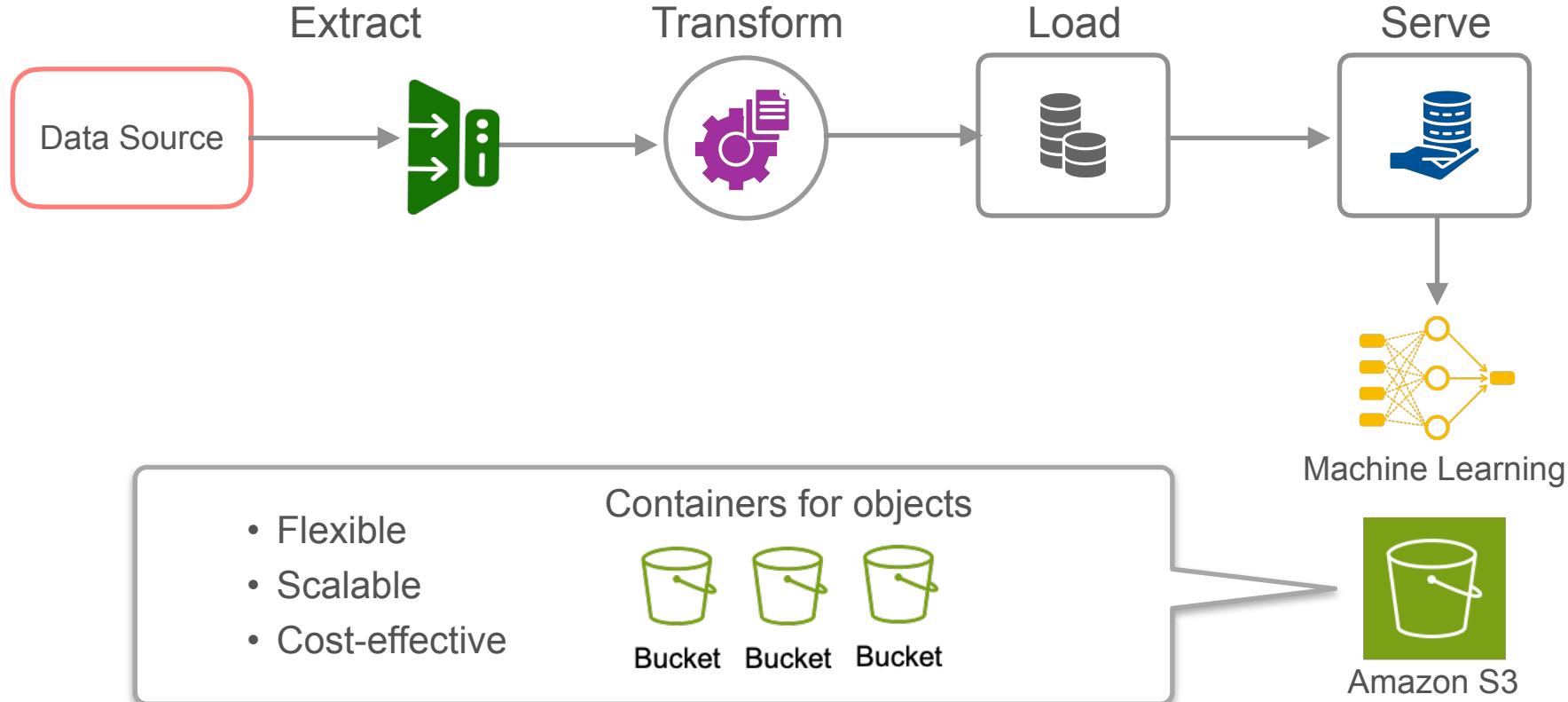
AWS Glue
Data Catalog

- Can use Glue visual ETL tool to design your pipeline

Serverless Tools for Batch Processing



Serverless Tools for Batch Processing



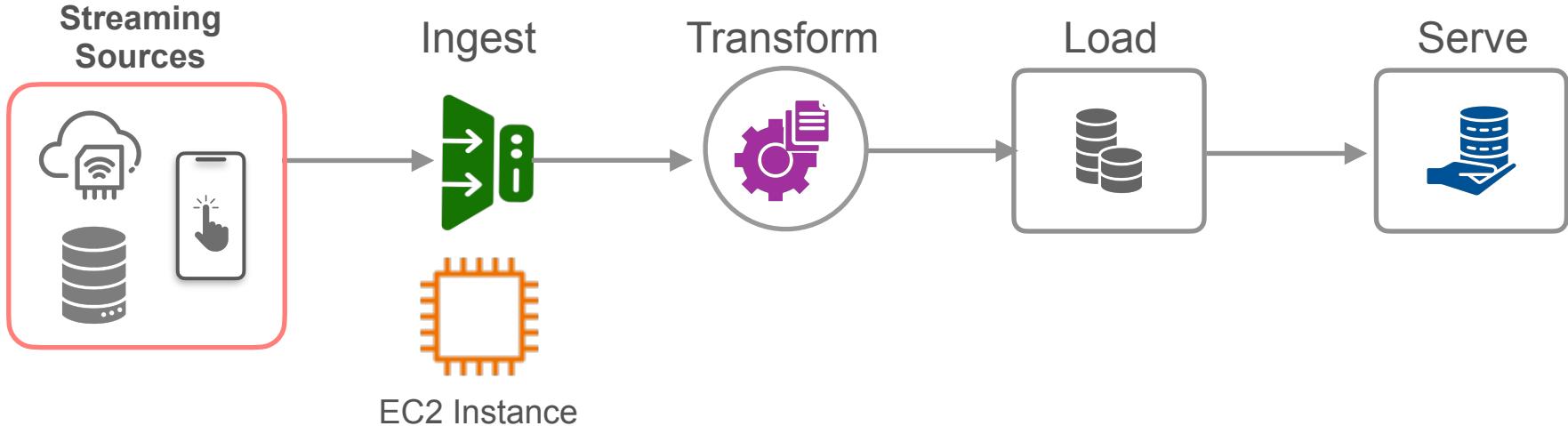


DeepLearning.AI

Translating Requirements to Architecture

AWS Services for Streaming Pipelines

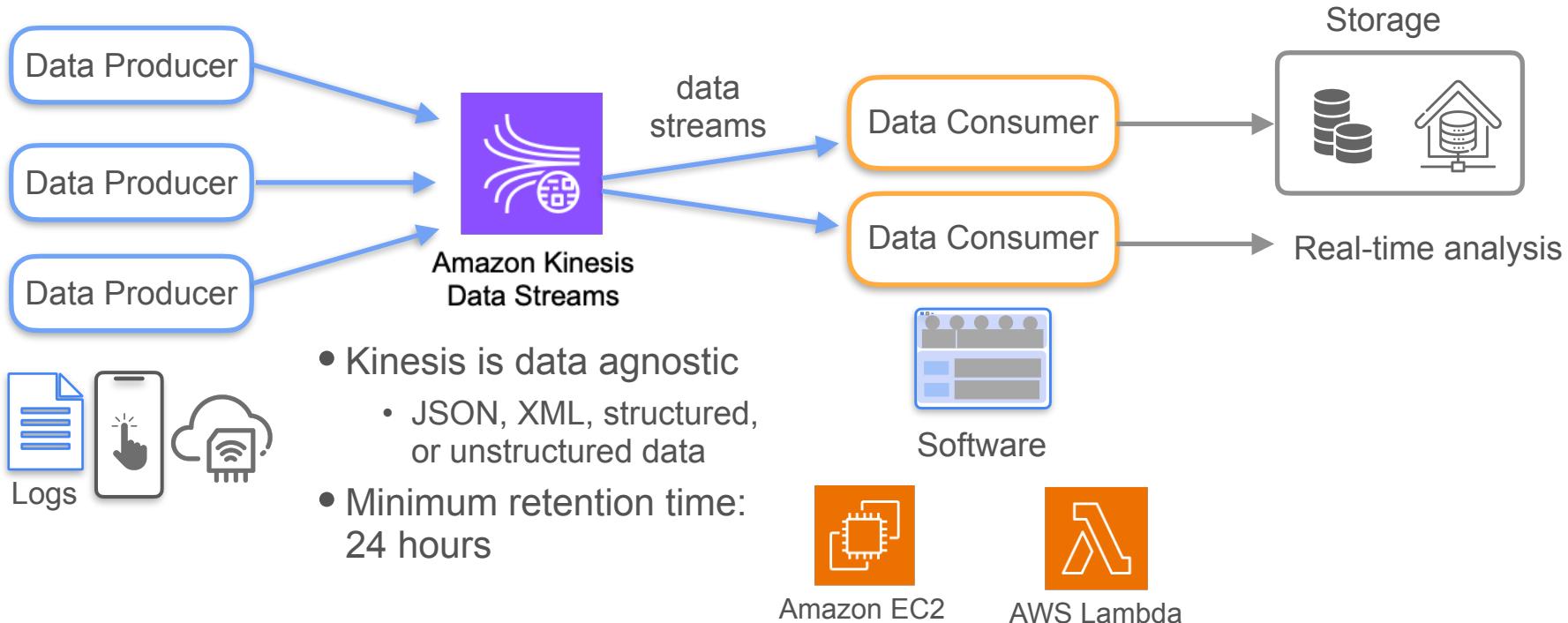
AWS Services for Streaming Pipelines



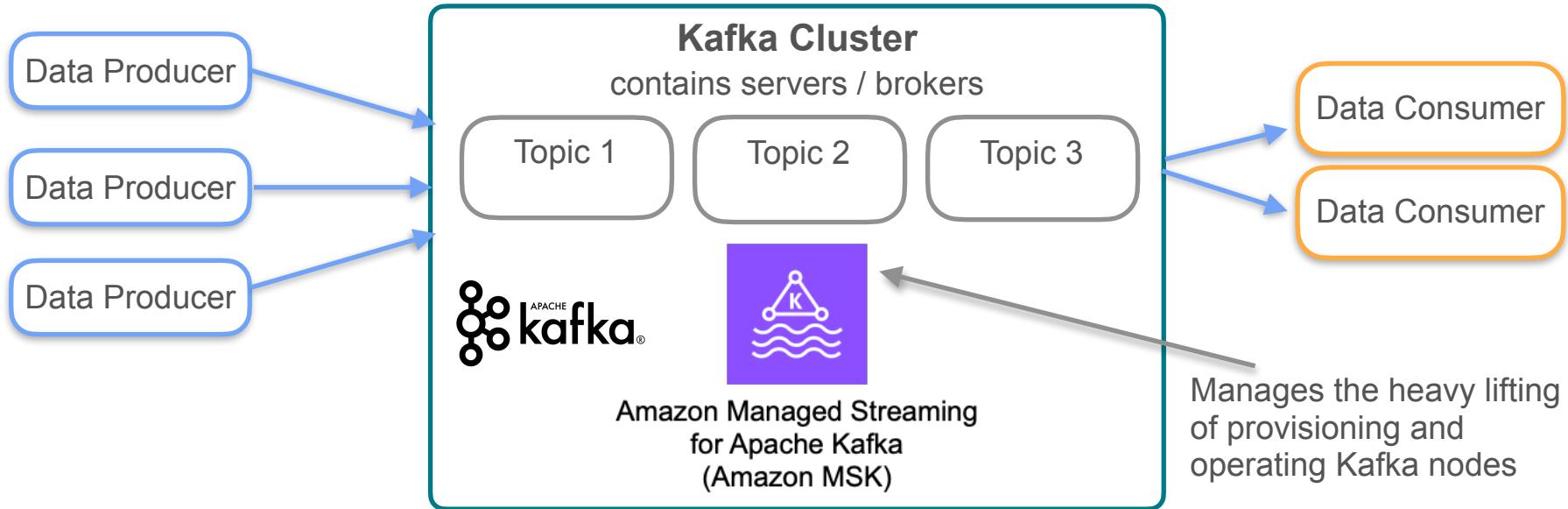
You are responsible for installing software, managing security, and managing cloud deployment complexity



AWS Services for Streaming Pipelines



AWS Services for Streaming Pipelines

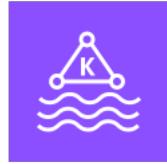


Kafka Data Plane

An interface you interact with that MSK manages for you to create topics, produce, and consume data

AWS Services for Streaming Pipelines

Both can scale up to handle petabyte-level data volumes with millisecond latency



Amazon Managed Streaming
for Apache Kafka
(Amazon MSK)



Amazon Kinesis
Data Streams

More control

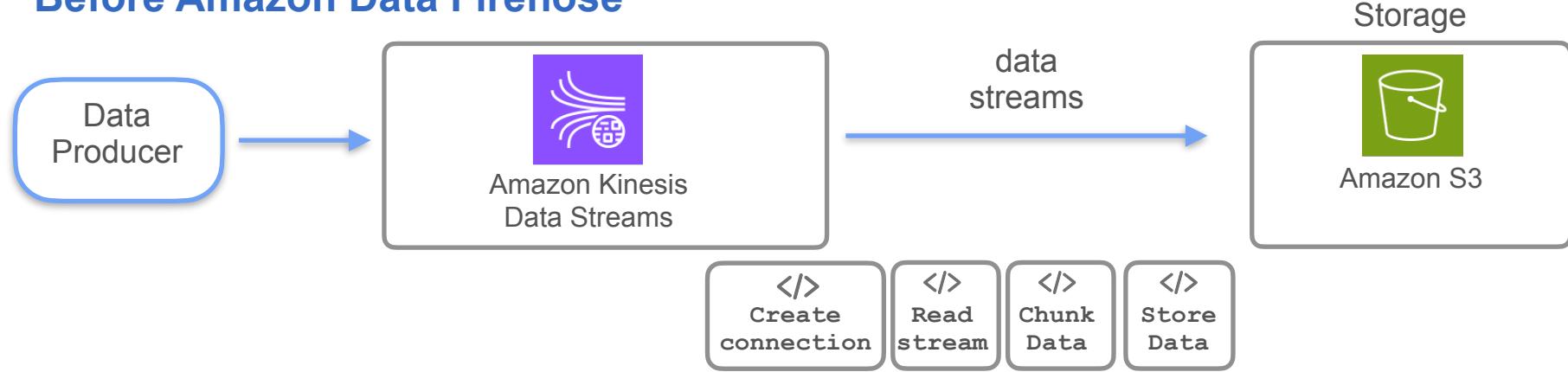
- Used for Kafka clusters
- High degree of flexibility and control

More convenience

- User-friendly
- Reduced operational overhead

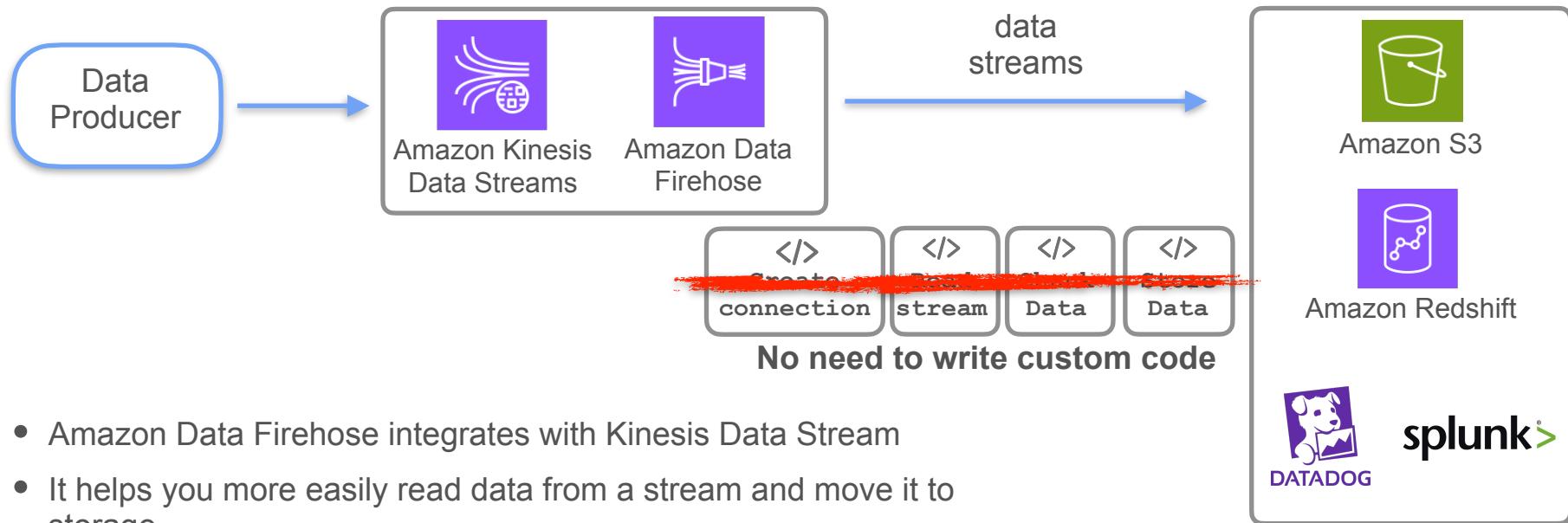
Amazon Data Firehose

Before Amazon Data Firehose



Amazon Data Firehose

With Amazon Data Firehose



- Amazon Data Firehose integrates with Kinesis Data Stream
- It helps you more easily read data from a stream and move it to storage
- Integrates with more than 20 other AWS sources



DeepLearning.AI

Lab Walkthrough

Implementing the batch pipeline

Lab Overview

1. Requirements gathering video

Conversation: data engineer & data scientist

2. Practice Quiz

Extract functional & nonfunctional requirements

3. Video with Morgan Willis

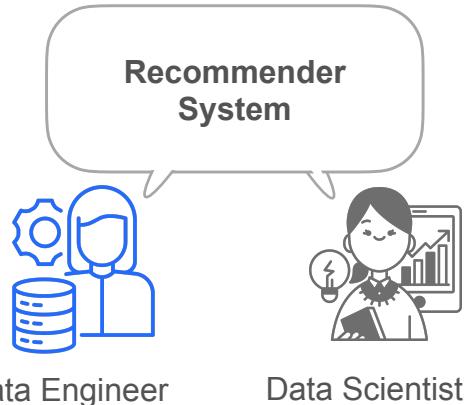
AWS tools & services

4. Quiz

Identify tools and services to meet requirements

5. Lab

- Implement the batch and streaming architectures for the recommender system



Lab Overview

1. Implement the batch pipeline to serve training data to the data scientist
2. Set up a vector database to store the output embeddings of the recommender system
3. Implement the streaming pipeline to output product recommendations for a user



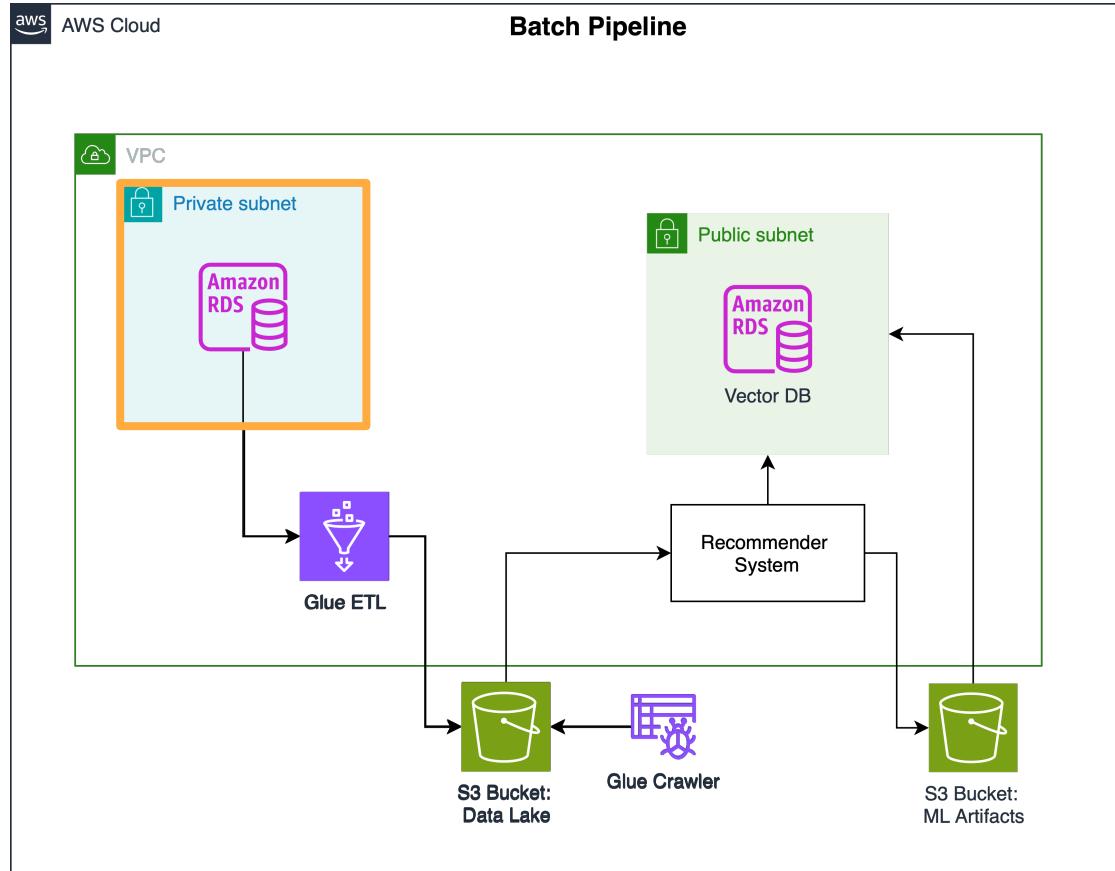
Terraform

Batch Pipeline Overview

RDS MySQL database

Source system that contains:

- classicmodels dataset
- ratings table (labels for the training set)



Batch Pipeline Overview

Glue ETL

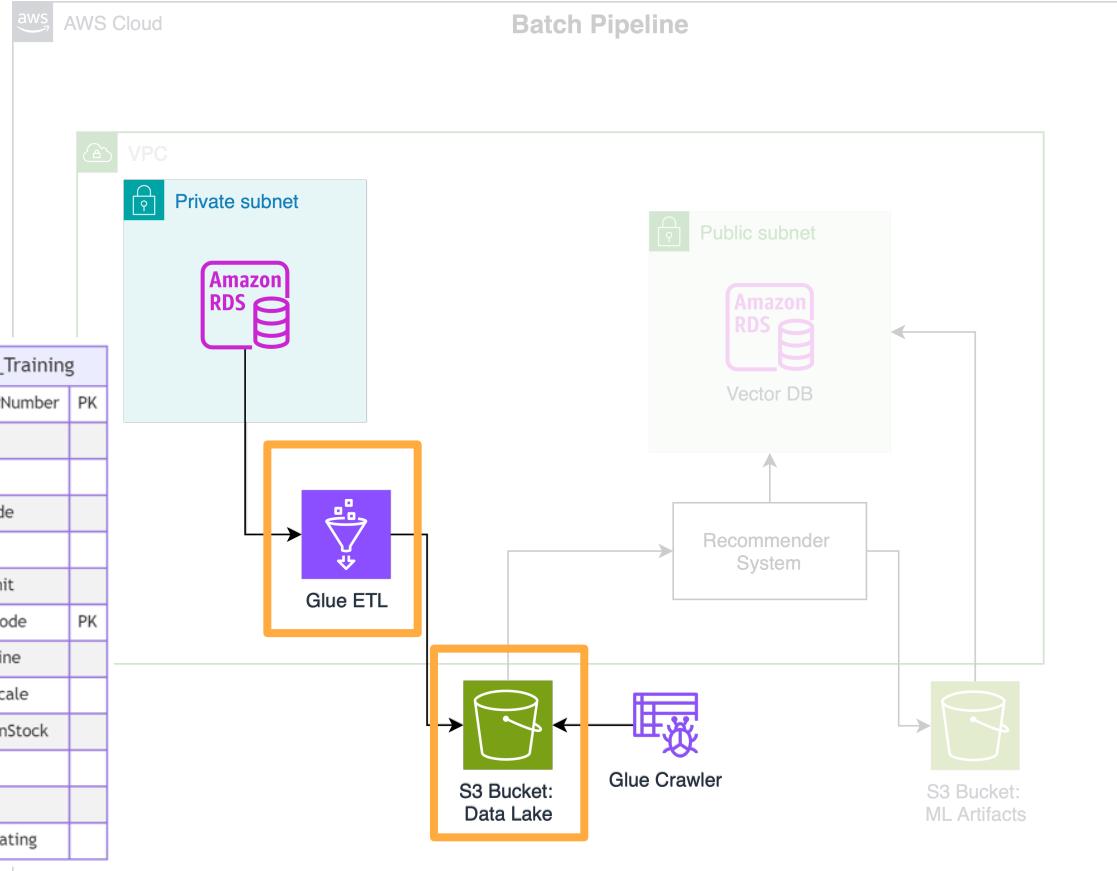
- Ingest data from the MySQL database
- Transform the data
- Load the data into the S3 bucket “Data Lake”

Create Glue ETL
and S3 using

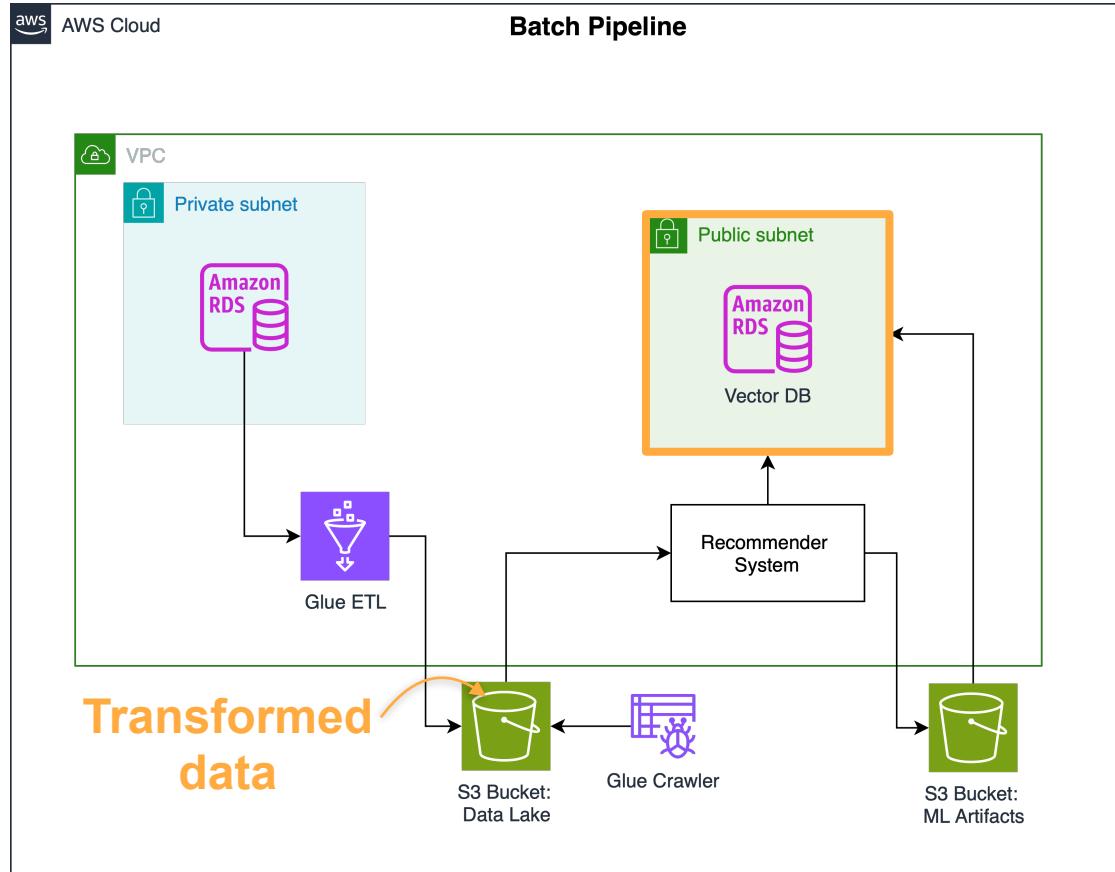


Terraform

Ratings_ML_Training		
		PK
int	customerNumber	PK
string	city	
string	state	
string	postalCode	
string	country	
float	creditLimit	
string	productCode	PK
string	productLine	
string	productScale	
int	quantityInStock	
int	buyPrice	
int	MSRP	
int	productRating	



Batch Pipeline





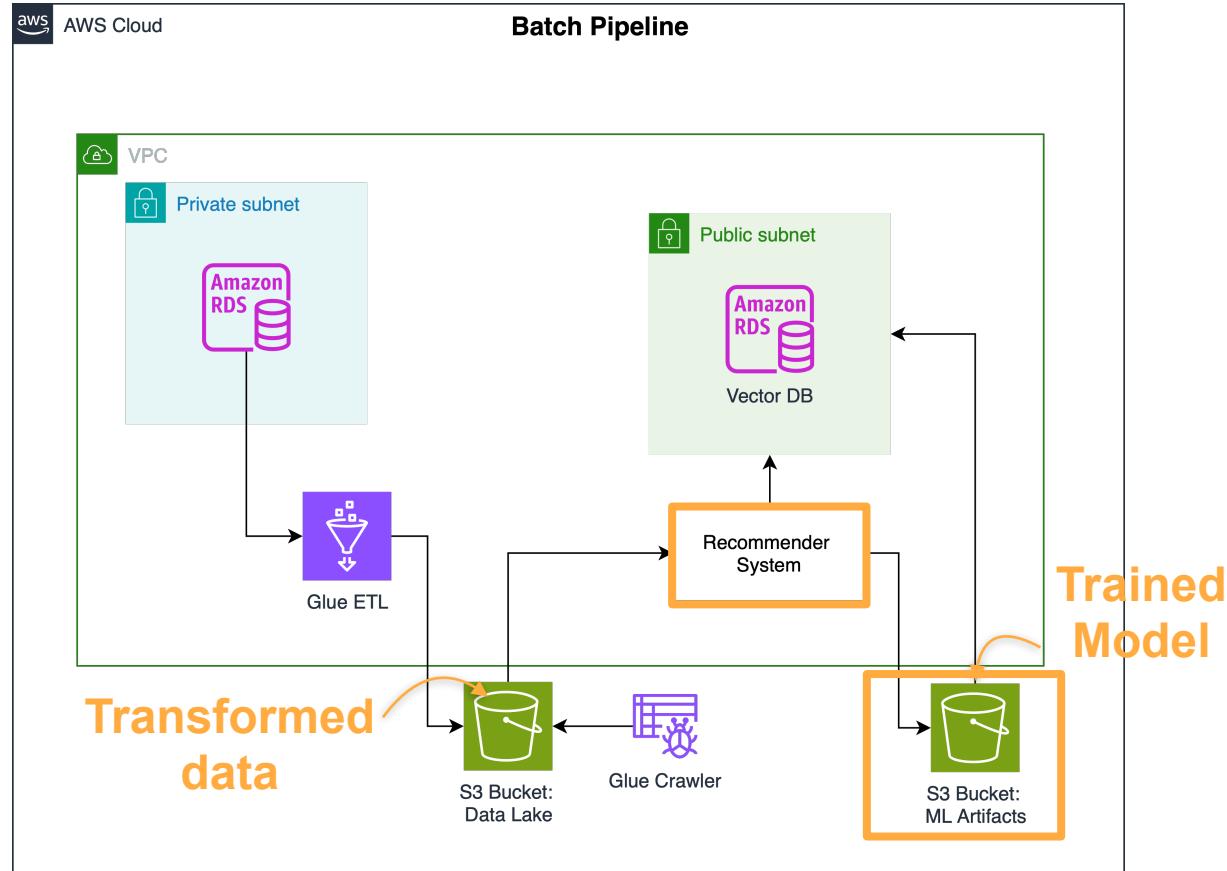
DeepLearning.AI

Lab Walkthrough

Setting up the vector database

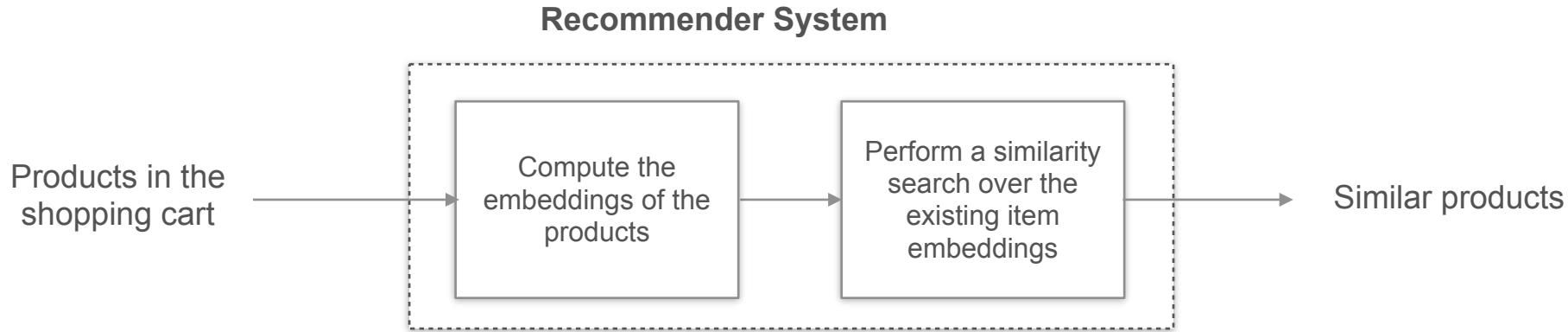
Vector Database Overview

Section 2: Creating and Setting up the Vector Database

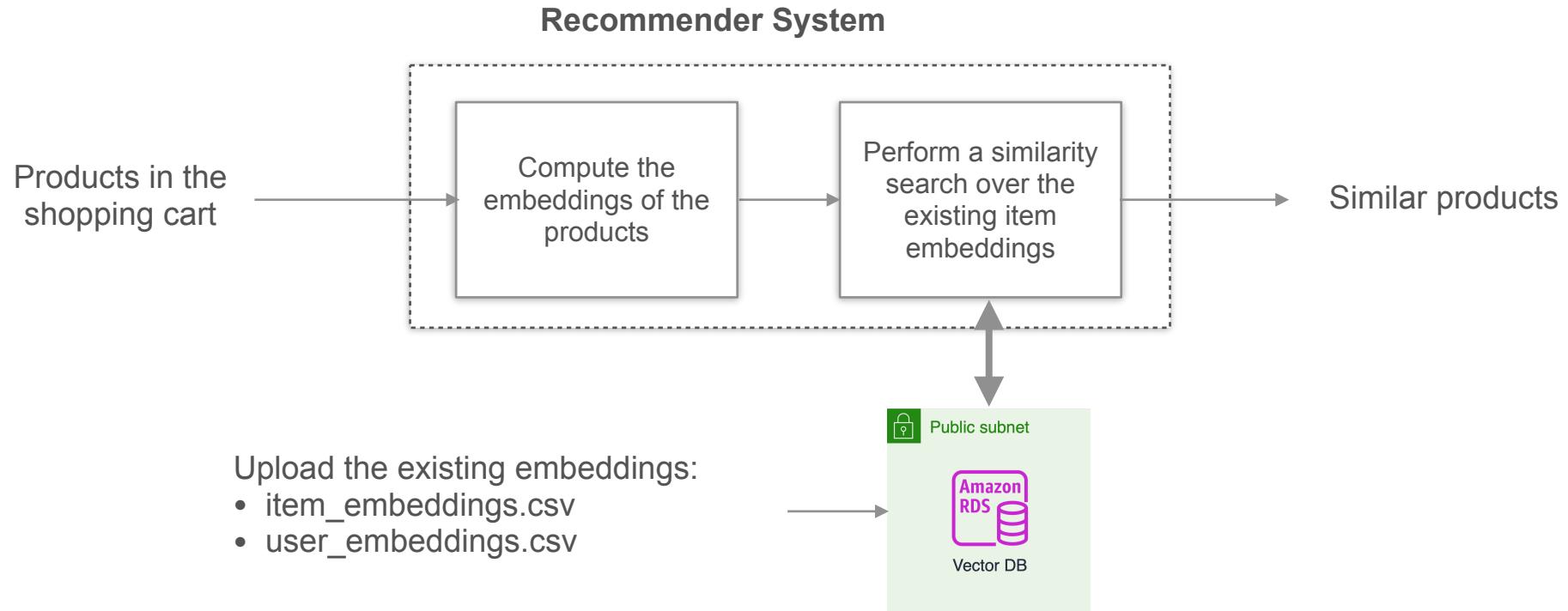


Vector Database

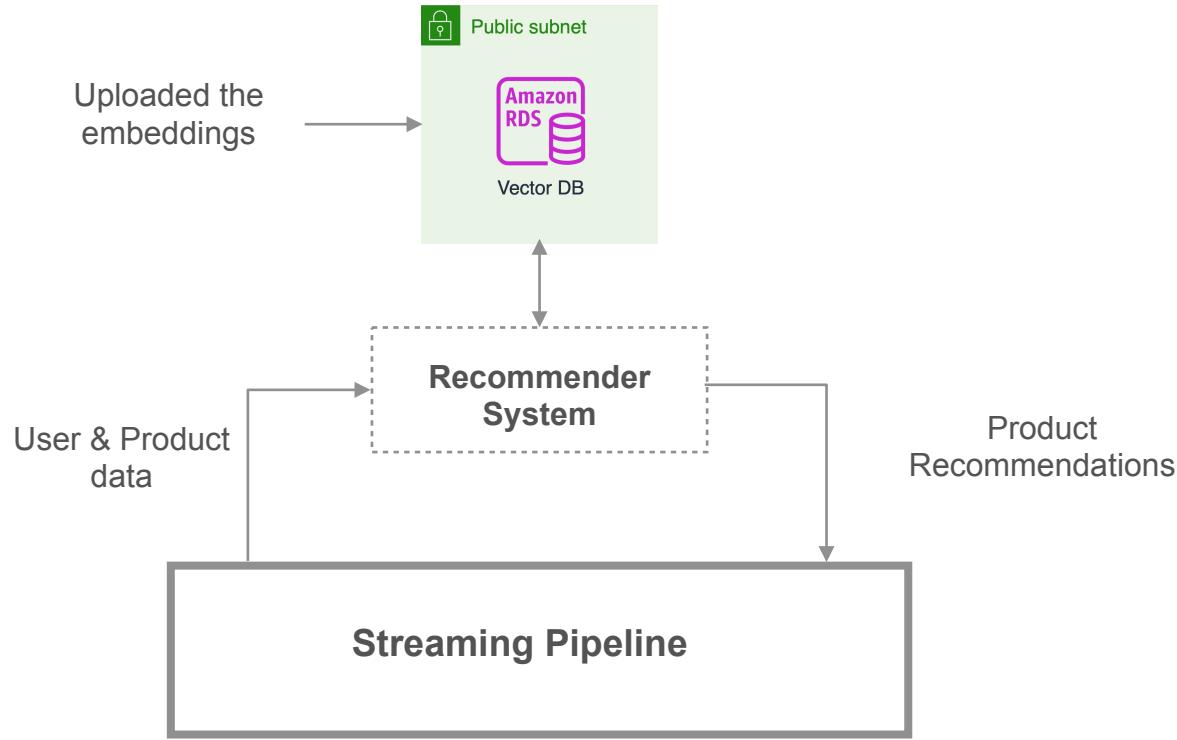
The item embeddings will be used to retrieve similar products.



Vector Database



Vector Database



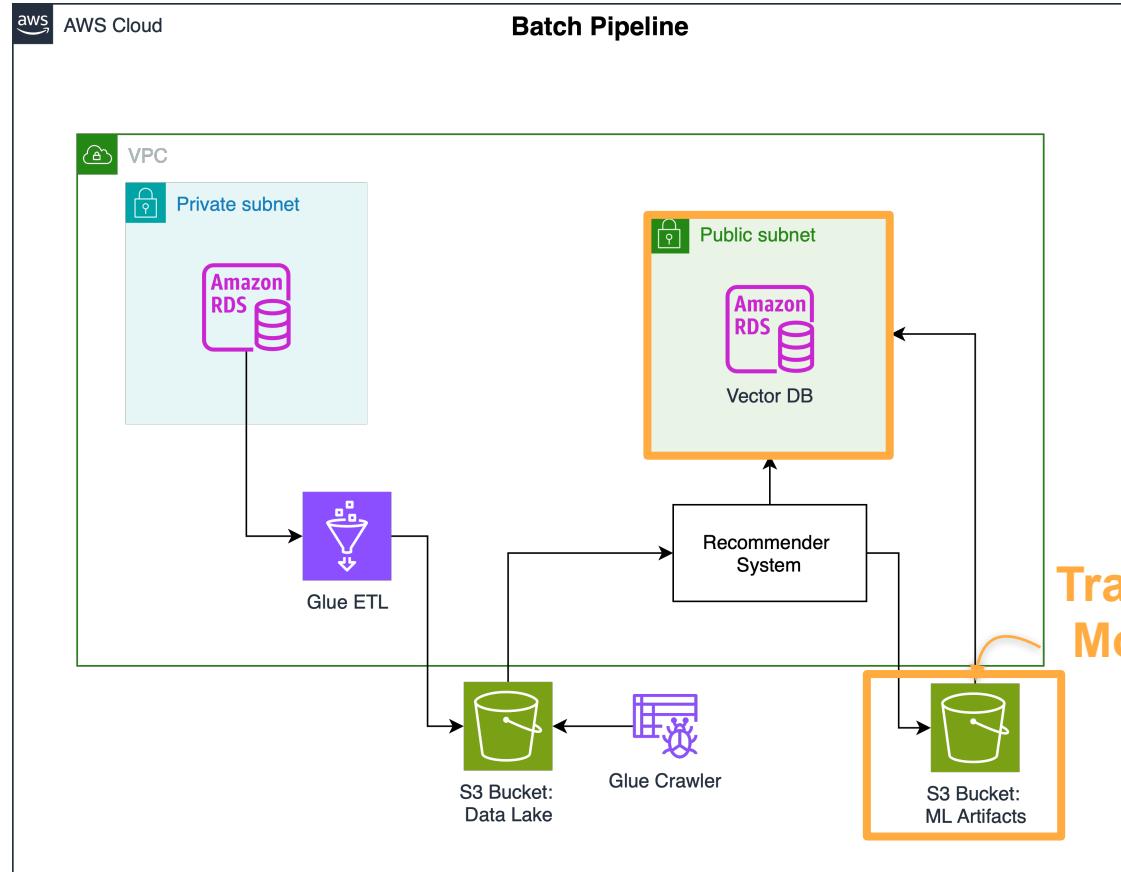


DeepLearning.AI

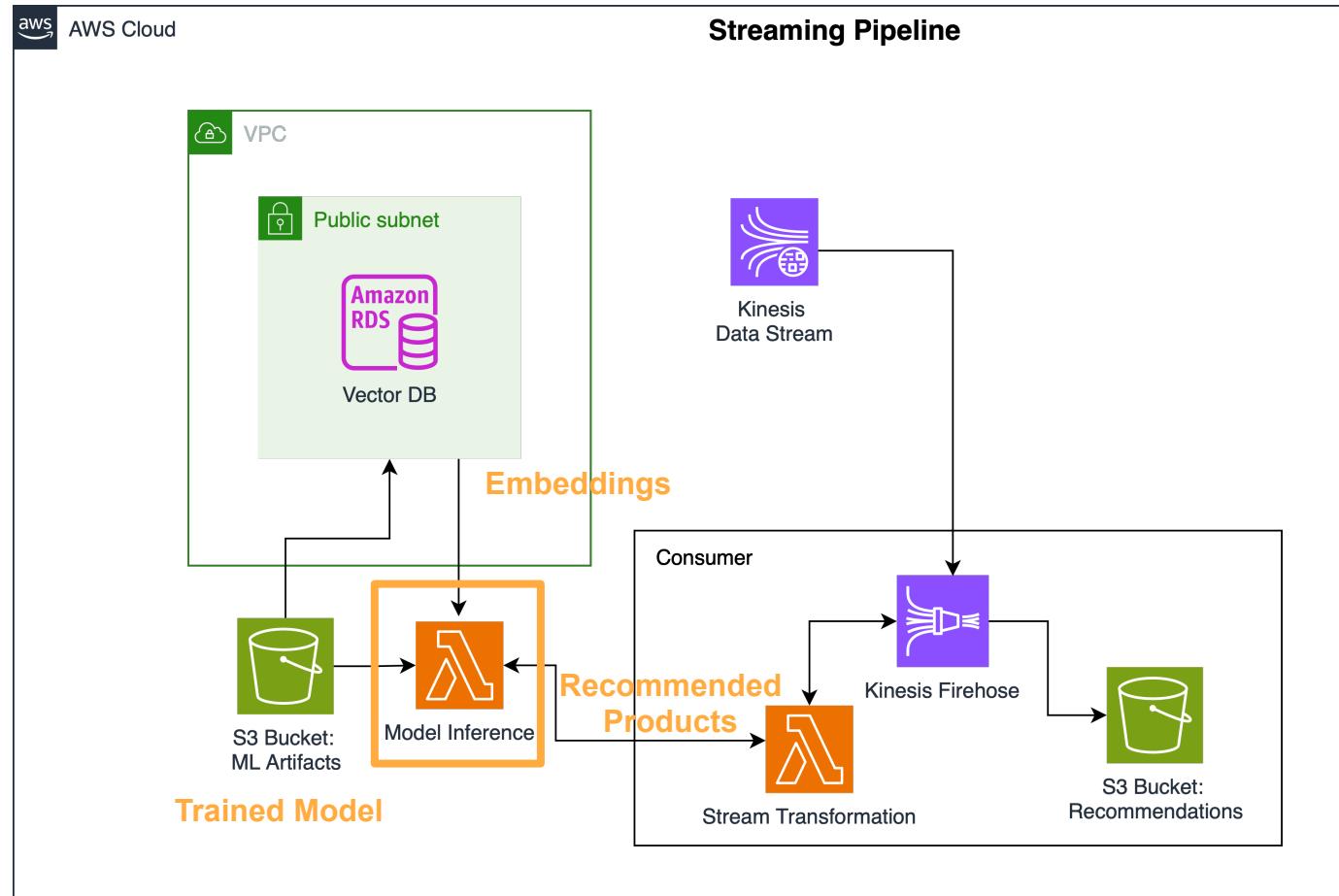
Lab Walkthrough

**Implementing the streaming
pipeline**

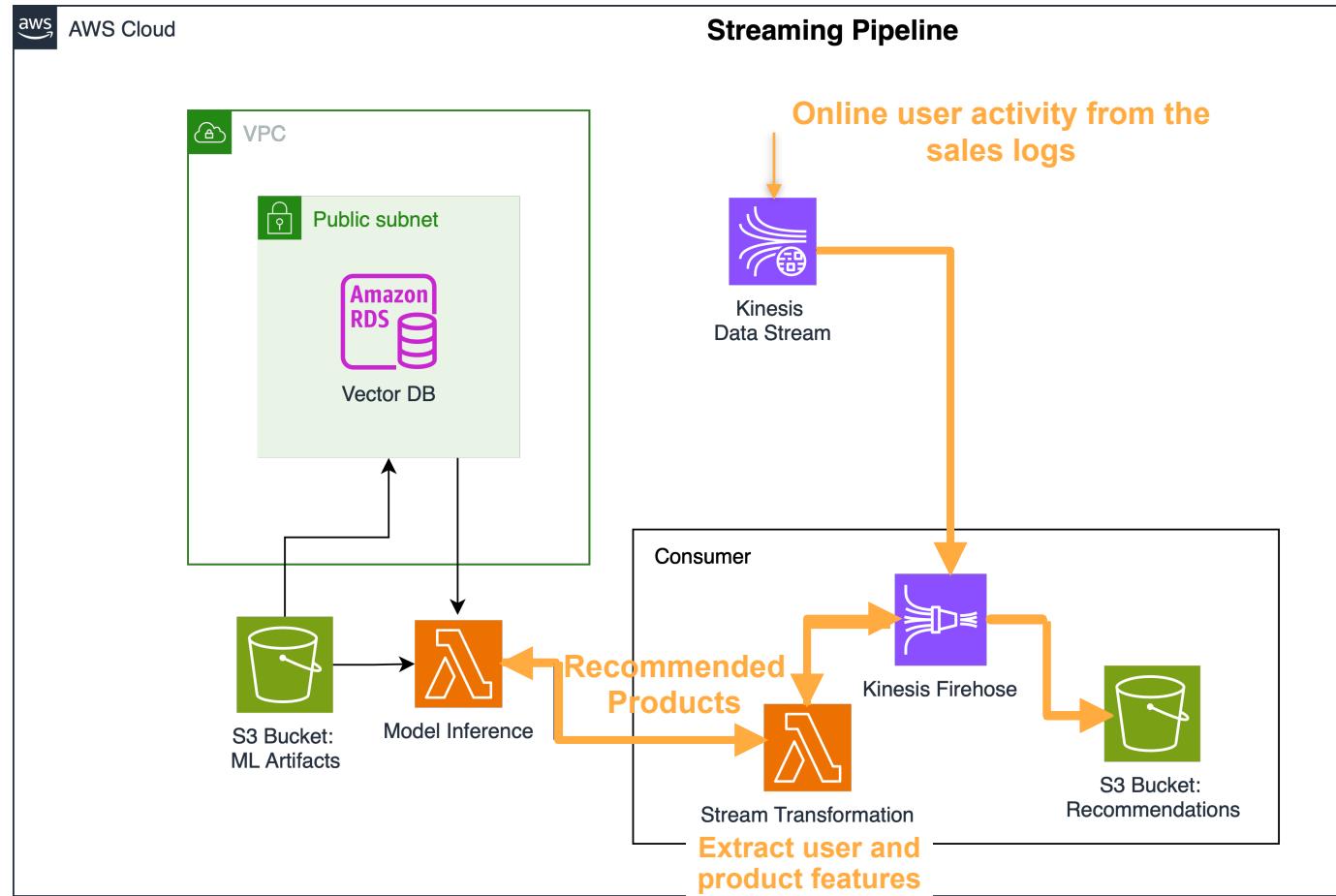
Last Video



Streaming Pipeline Overview



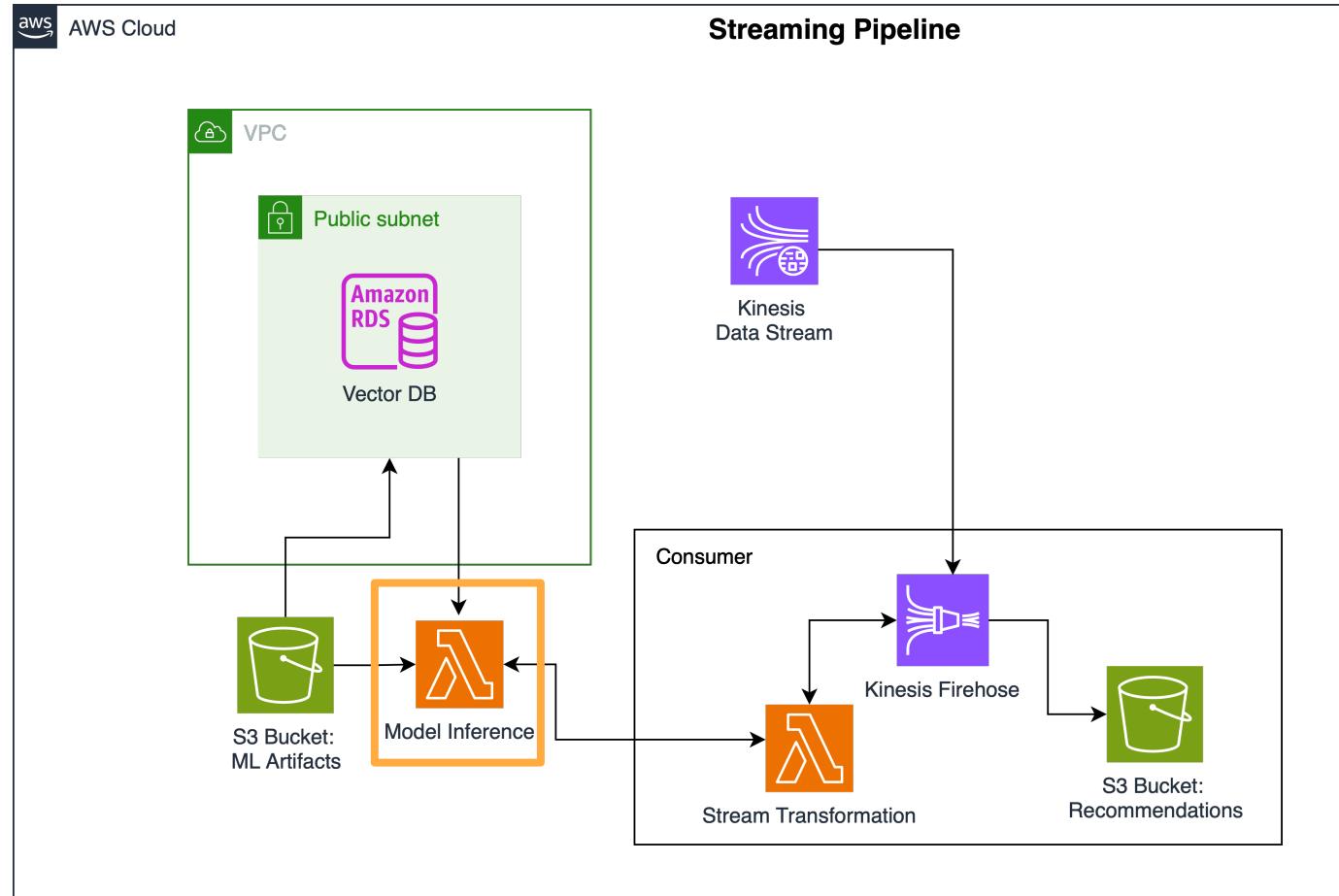
Streaming Pipeline Overview



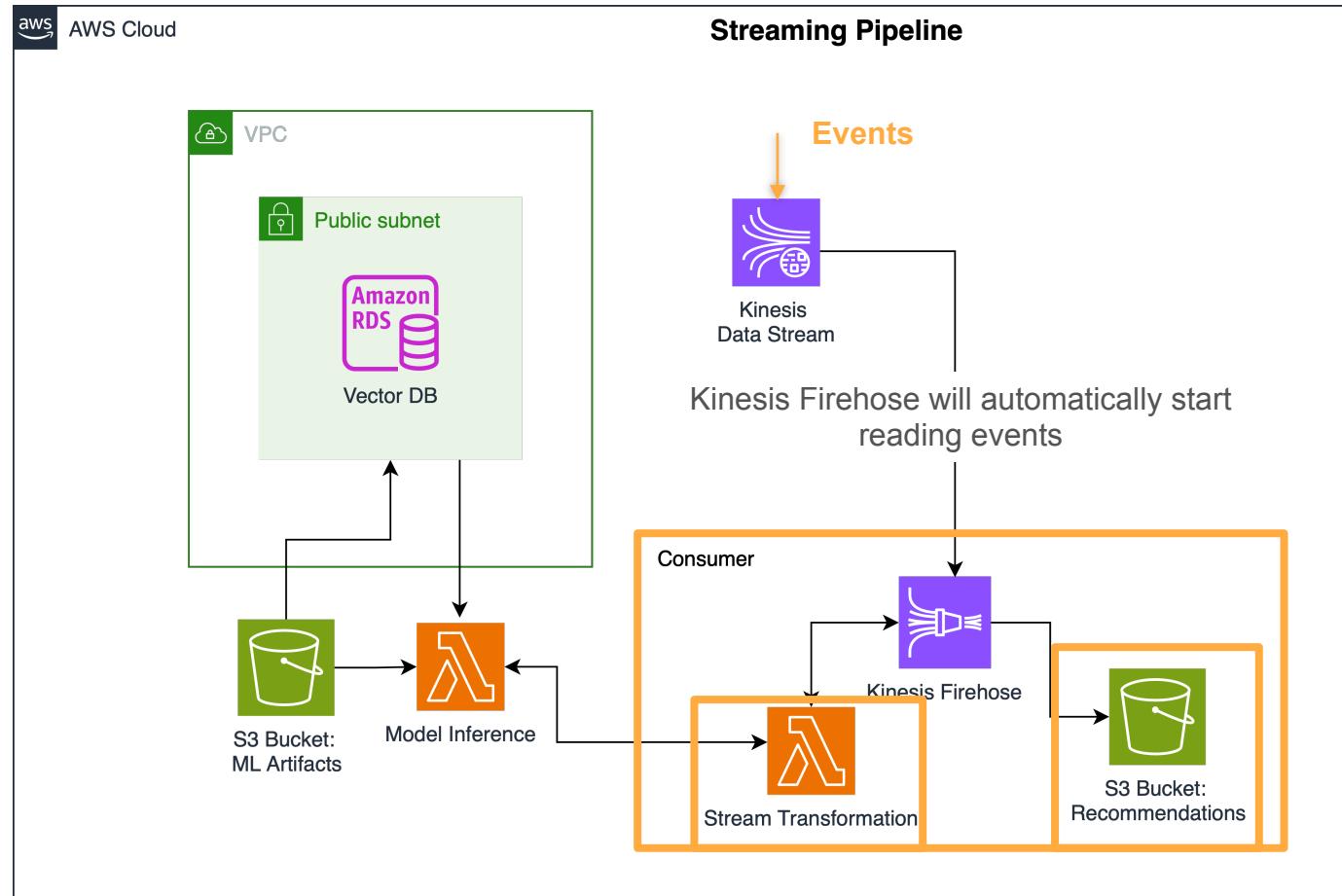
Streaming Pipeline Overview

Section 3:

Configure the lambda function so that it can connect to the vector database.



Streaming Pipeline Overview





DeepLearning.AI

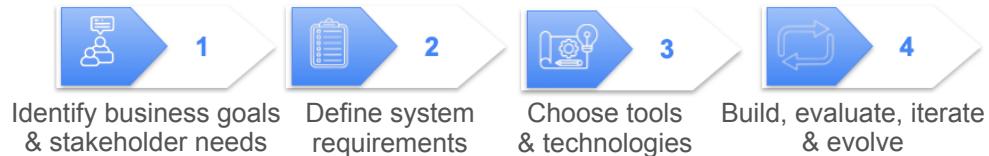
Introduction to Data Engineering

Course Summary

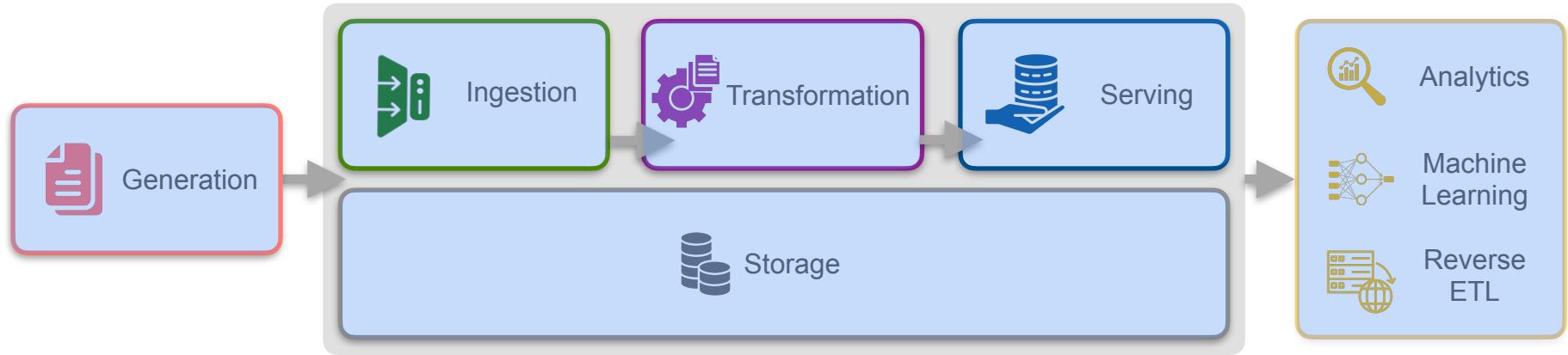
Week 1: Introduction to Data Engineering

- History of data engineering
- Data engineering defined
- Data engineering lifecycle
- Framework for thinking like a data engineer

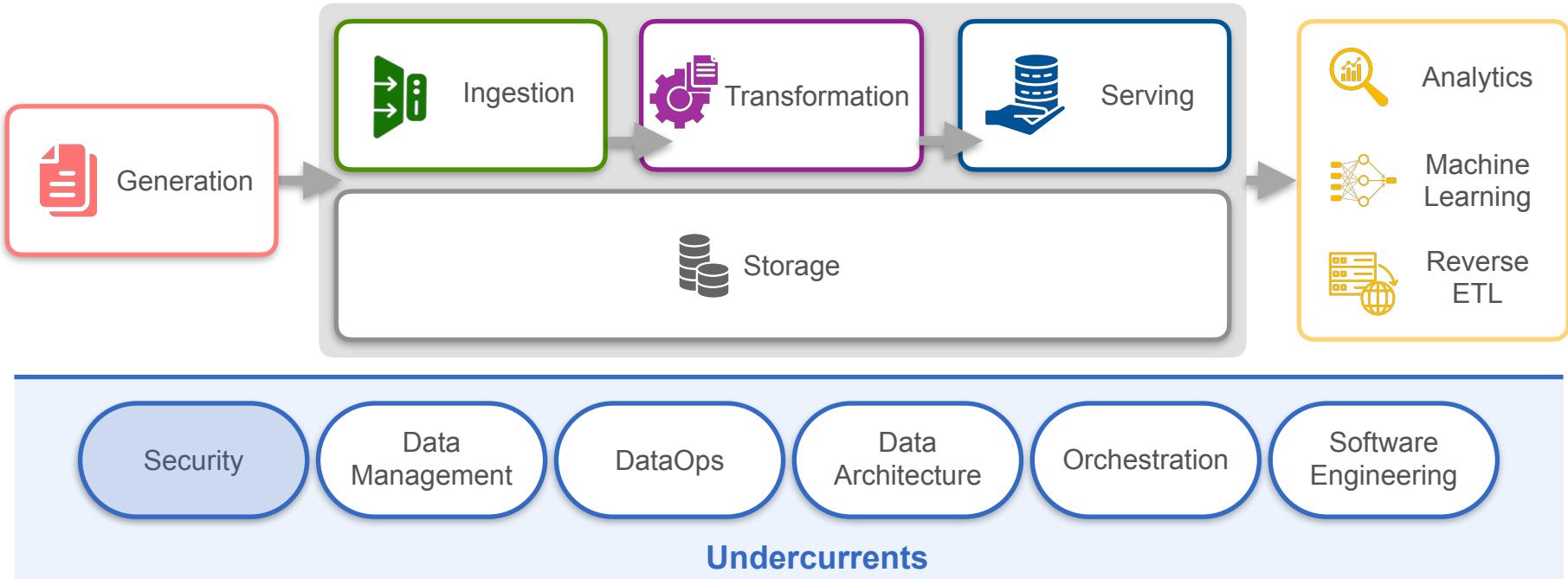
Data engineering: “Take in raw data and produce high-quality, consistent information that supports downstream use cases”



Week 2: Data Engineering Lifecycle & Undercurrents



Week 2: Data Engineering Lifecycle & Undercurrents

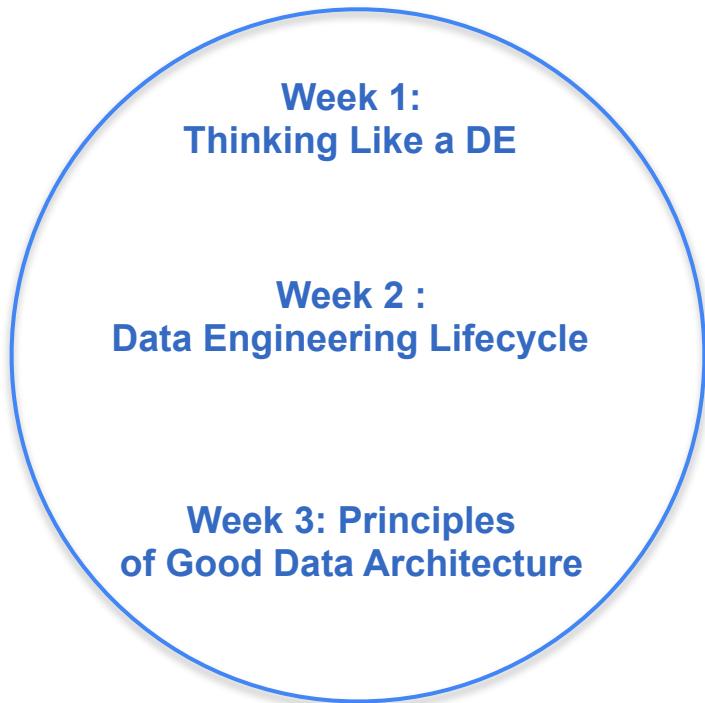


Week 3: Data Architecture

Principles of Good Data Architecture

1. Choose common components wisely
2. Plan for failure!
3. Architect for Scalability
4. Architecture is leadership
5. Always be Architecting
6. Build loosely coupled systems
7. Make reversible decisions
8. Prioritize Security
9. Embrace FinOps

Week 4: Bringing It All Together



Data Engineer

