Copyright Notice

DeepLearning.AI

Data Modeling, Transformation and Serving

Serving Data

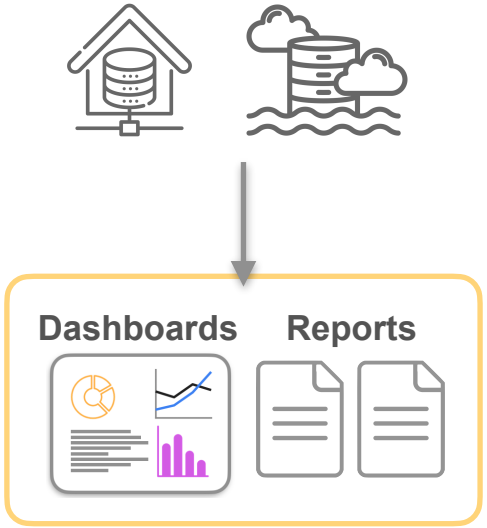
DeepLearning.AI

Serving Data for Analytics and Machine Learning

Week 4 Overview

Serving Data — Analytics Use Cases

Business Intelligence

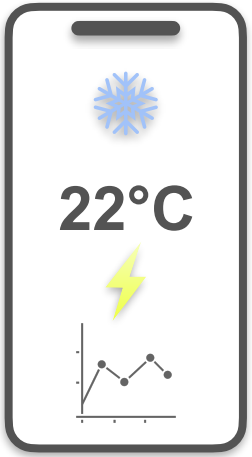


Operational Analytics

Monitor data to inform immediate action

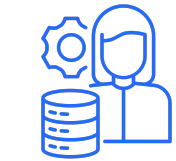
Serve data within the required latency

Embedded Analytics



Serving Data — Machine Learning Use Cases

Becoming widely adopted in many companies



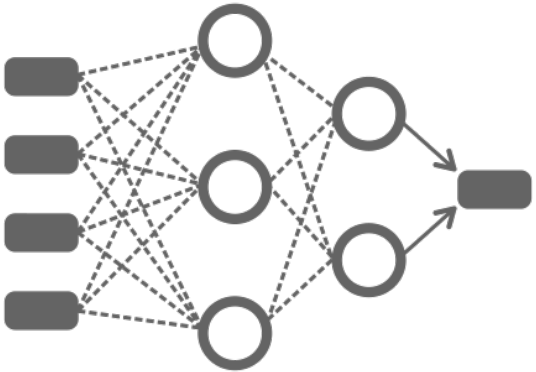
Data Engineer

Data Scientist

Machine Learning Engineer

Model training

- FEATURE 1
- FEATURE 2
- FEATURE 3
- FEATURE 4



Example:

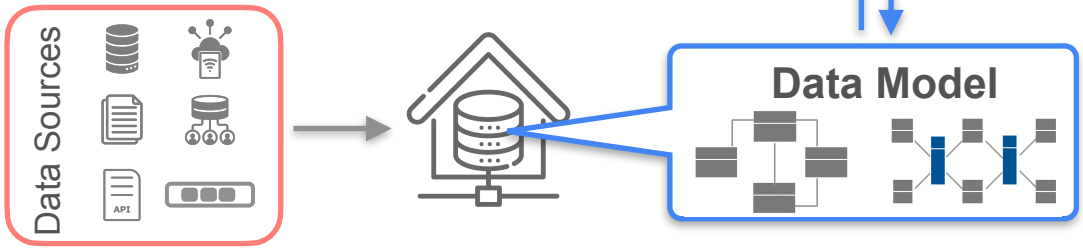
- Customer churn model
- Recommendation system

Serving Data

Stakeholders

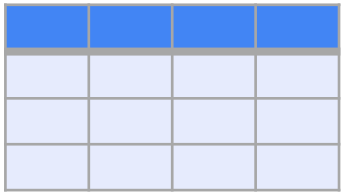
Semantic Layer

- Document definitions (e.g. meaning of an active user)
- Derive business metrics (e.g. revenue)
- Create a common language for data



You can serve data as…

A table



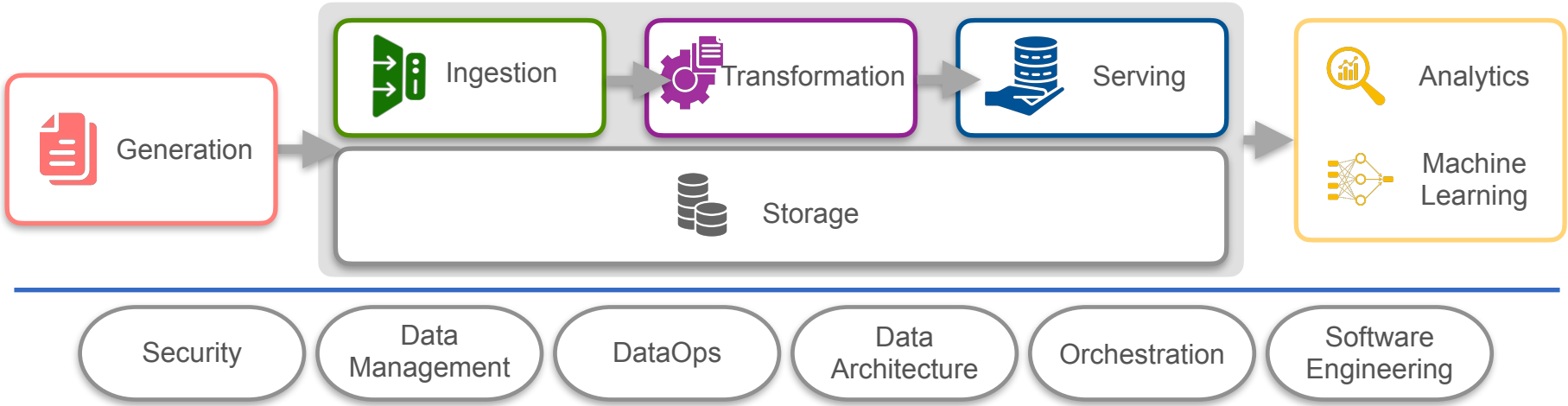
View

Materialized View

Thinking Like a Data Engineer

1. Identify business goals & stakeholder needs
2. Define system requirements
3. Choose tools & technologies
4. Build, evaluate, iterate & evolve

Data Engineering Lifecycle and Undercurrent

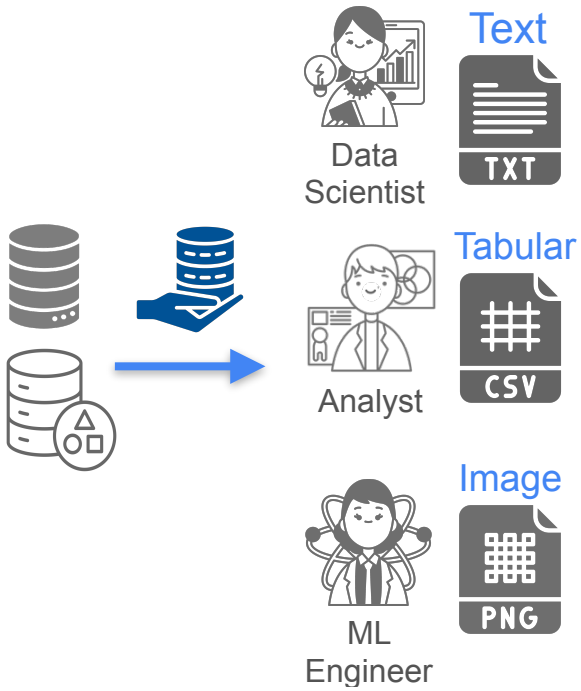


DeepLearning.AI

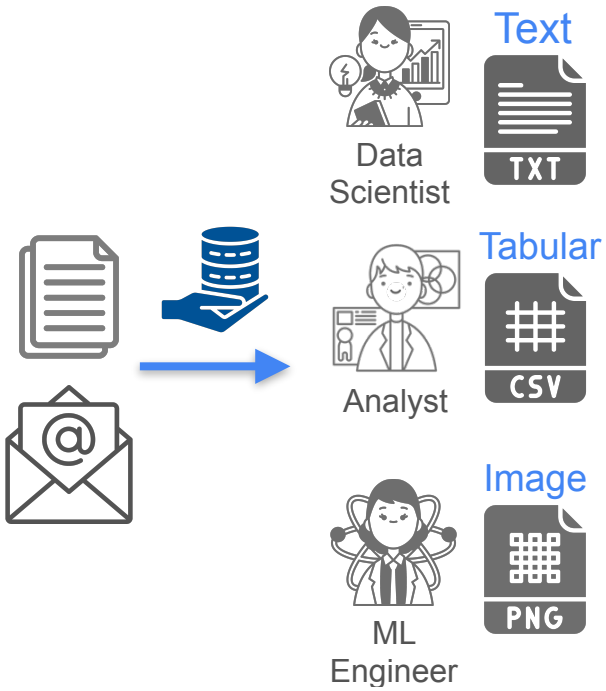
Serving Data for Analytics and Machine Learning

Serving Data

As Files

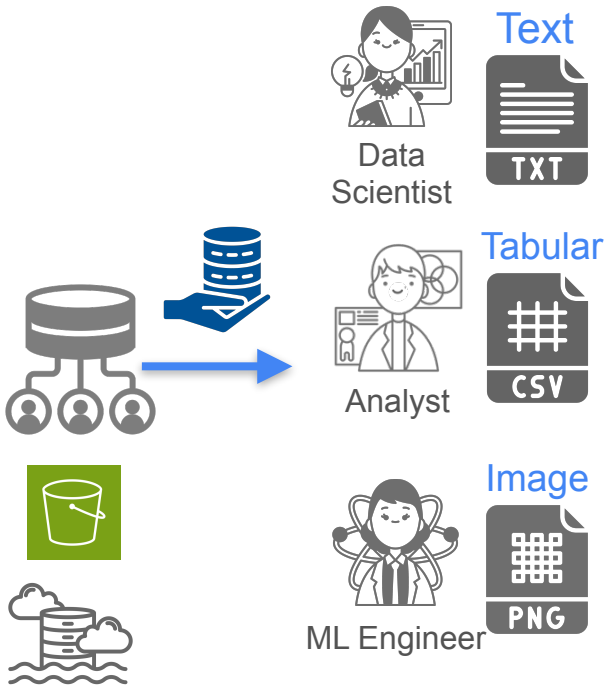

Text
TXT
Data Scientist
Tabular
CSV
Analyst
Image
PNG
ML Engineer

As Files

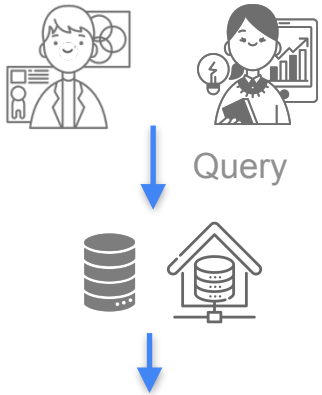

Text
TXT
Data Scientist
Tabular
CSV
Analyst
Image
PNG
ML Engineer

As Files

For ad hoc requests



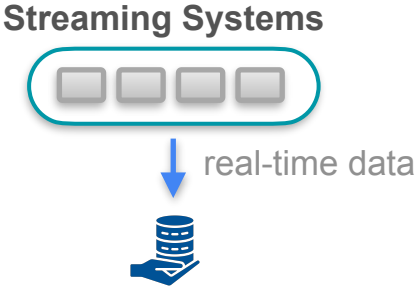
From Databases and Data Warehouses



Benefits:

- Imposes order & structure through schema
- Gives you fine-grained permissions controls
- Offers high performance for queries

From Streaming Systems



Example

Operational analytics database

- Enables low latency analytical queries across historical and current data
- Effectively combining the features of an OLAP database with a stream-processing system

Data Management

Data Engineer

Trust the data

Can interpret the data correctly

Stakeholders

Ensuring data correctness, consistency, trustworthiness

Data Definition

Meaning of data as it is understood throughout the organization
e.g. document and make available the definition of “Customer”

Data Logic

Consists of formulas for deriving metrics from data
e.g. gross sales or customer lifetime value

Customer Churn Metric → Understand the definition of “Customer" → Write SQL query → Define this metric so it can be reused

Semantic Layer

- Translate the underlying data elements and structures into more intuitive business terms
- Ensures consistent definitions for business terms
- Helps end users more easily navigate the data

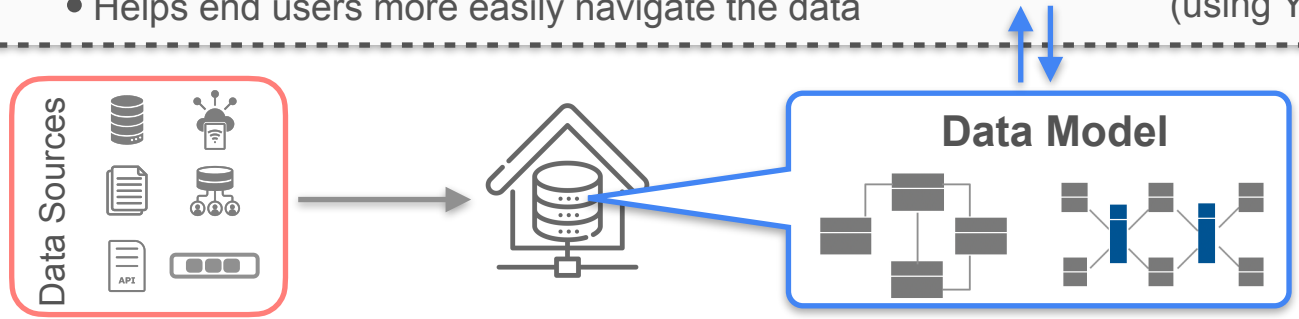
Created in a BI tool

OR

Created using software

dbt

(using YAML files & SQL queries)



Stakeholders

Use BI platform to visualize and analyze data

Amazon Quicksight

Apache Superset

Looker

Use a notebook to explore data, engineer features, or train a model

Amazon SageMaker

Google Cloud Vertex AI

Azure Machine Learning

Semantic Layer

- Translate the underlying data elements and structures into more intuitive business terms
- Ensures consistent definitions for business terms
- Helps end users more easily navigate the data

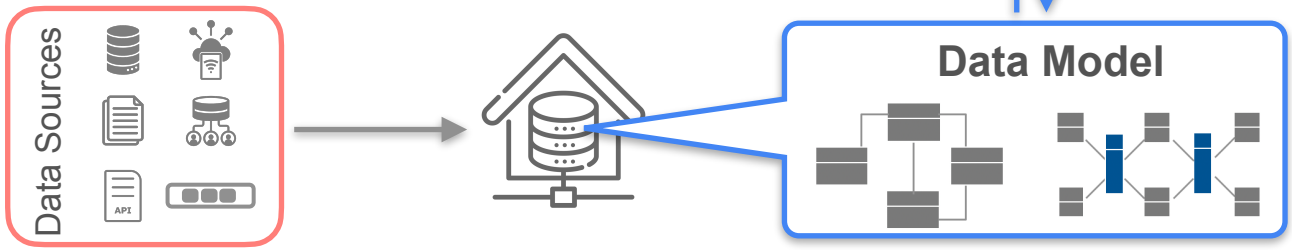
Created in a BI tool

OR

Created using software

dbt

(using YAML files & SQL queries)



DeepLearning.AI

Serving Data for Analytics and Machine Learning

Views and Materialized Views

View

- Stored in the database to provide easier access to common queries
- Represents a virtual table, not a physical one
- When selecting from a view:
  - The database creates a new query
  - The query optimizer optimizes and runs the query

```
CREATE VIEW  customer_info AS
SELECT
     first_name,
     last_name,
     email,
     phone,
     city,
     postal_code,
     country
FROM
     customer
     JOIN address ON address.id = customer.address_id
     JOIN city ON city.id = address.city_id
     JOIN country ON country.id = address.country_id
```

View

- Stored in the database to provide easier access to common queries
- Represents a virtual table, not a physical one
- When selecting from a view:
  - The database creates a new query
  - The query optimizer optimizes and runs the query

```
CREATE VIEW  customer_info AS
SELECT
    first_name,
    last_name,
    email,
    Phone,
    city,
    postal_code,
    country
FROM
    customer
    JOIN address ON address.id = customer.address_id
    JOIN city ON city.id = address.city_id
    JOIN country ON country.id = address.country_id
```

Effectively restricting data access to only the data needed

Marketing Analyst

query

Wide Table:
customer, address, city, country

View

- Stored in the database to provide easier access to common queries
- Represents a virtual table, not a physical one
- When selecting from a view:
  - The database creates a new query
  - The query optimizer optimizes and runs the query
- Database object
- Stored and persists on disk

Common Table Expressions (CTE)

- Represents temporary results that you can reference
- Only exists within scope of the main query where they are referenced

```
WITH name of the CTE AS (
Query),

Subsequent query
```

View

- Stored in the database to provide easier access to common queries
- Represents a virtual table, not a physical one
- When selecting from a view:
  - The database creates a new query
  - The query optimizer optimizes and runs the query
- Database object
- Stored and persists on disk

Can be expensive for frequently-run complex queries

Common Table Expressions (CTE)

- Represents temporary results that you can reference
- Only exists within scope of the main query where they are referenced

Materialized Views

- Does some or all of the view computations in advance
- Caches the query results and allows you to refresh the data
- Some latency between refreshes

```
CREATE MATERIALIZED VIEW  rental_by_category AS
SELECT category.name AS category,
       sum(payment.amount) AS total_sales
FROM payment
   JOIN ... rental, inventory, film, film_category, category ...
GROUP BY category.name
```

DeepLearning.AI

Data Engineering

Summary of the Program Concepts

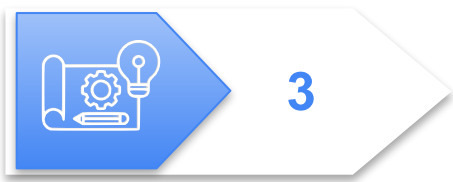
Thinking Like a Data Engineer

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

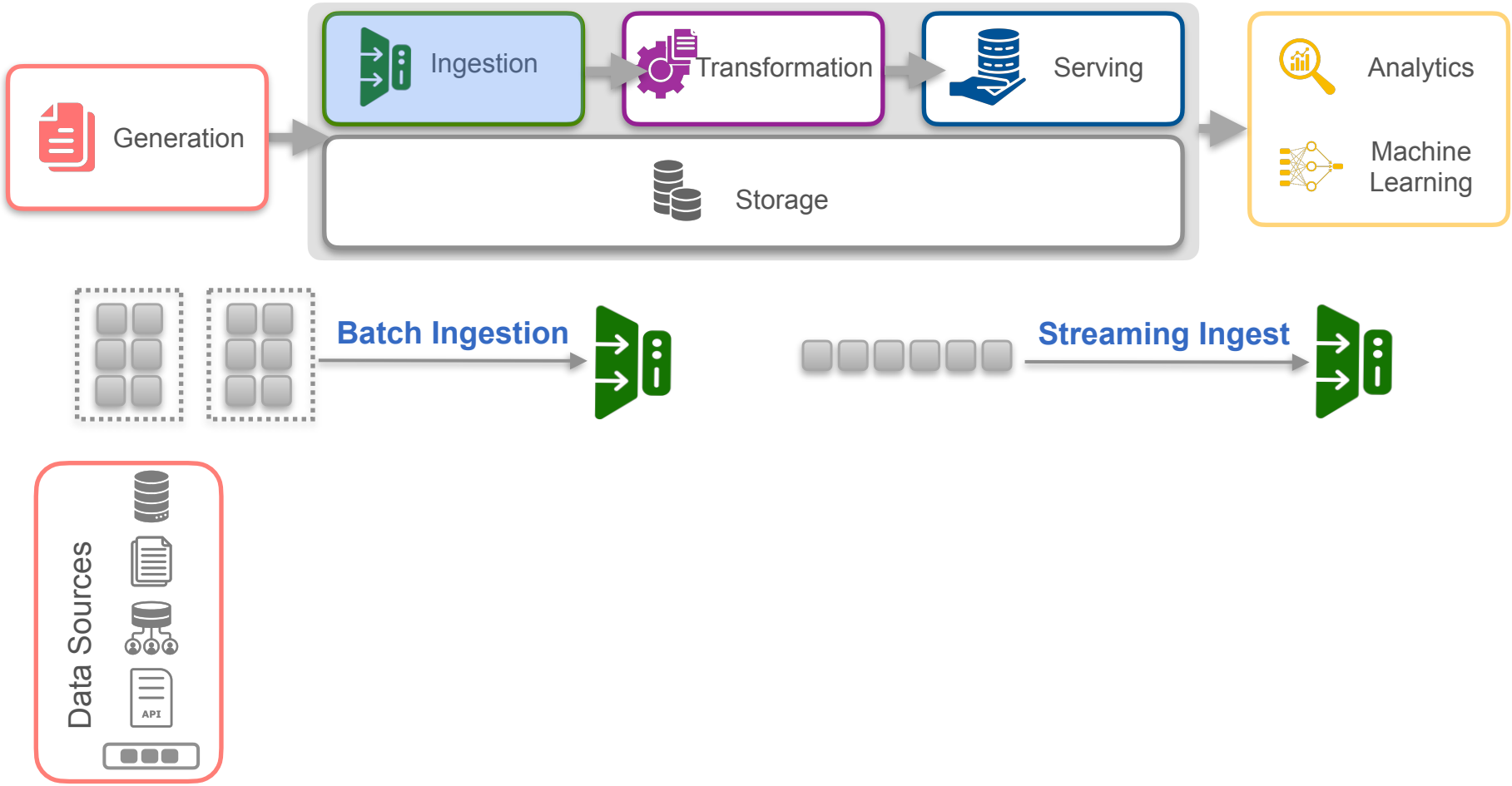


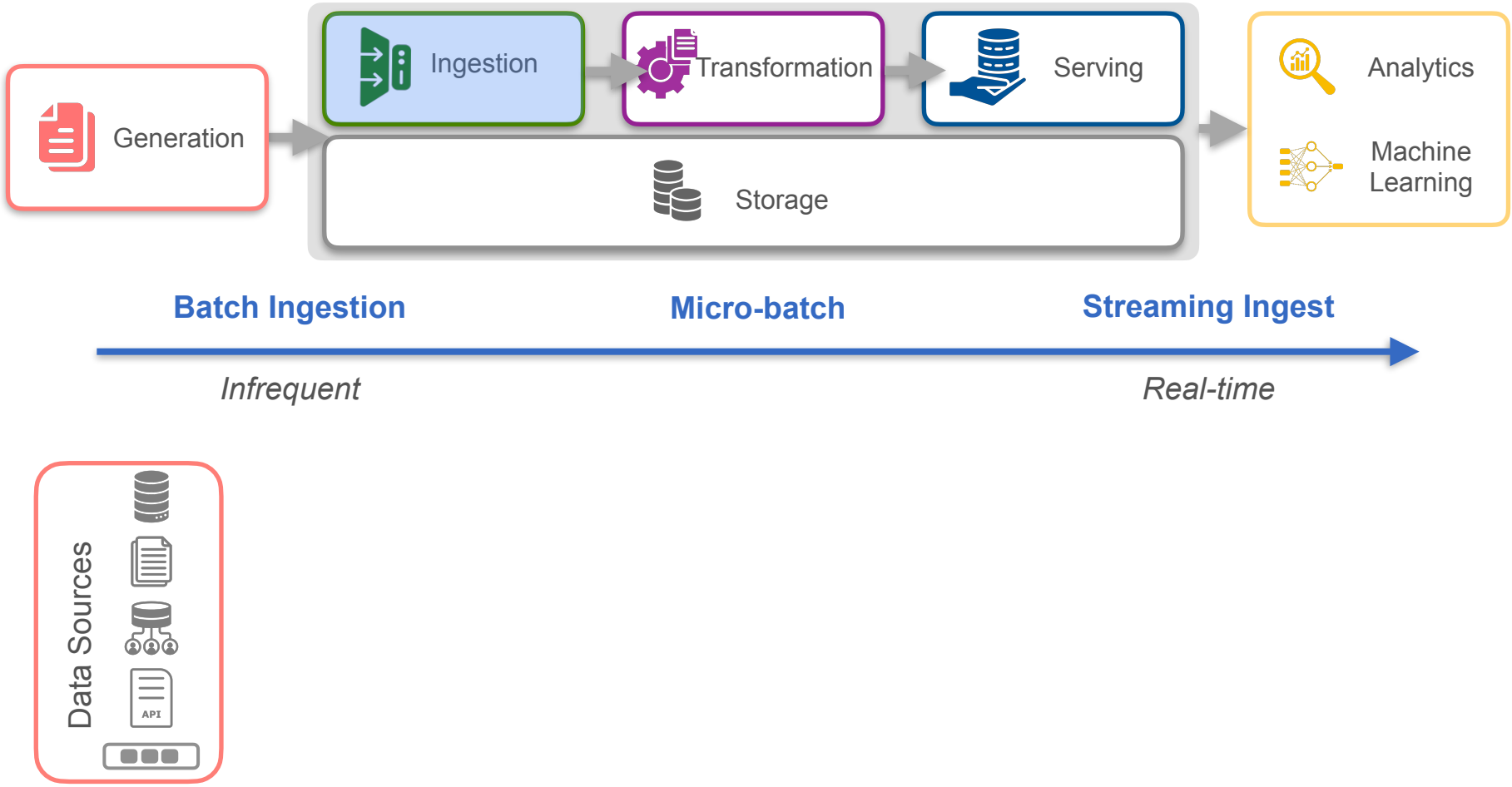
Choose tools & technologies

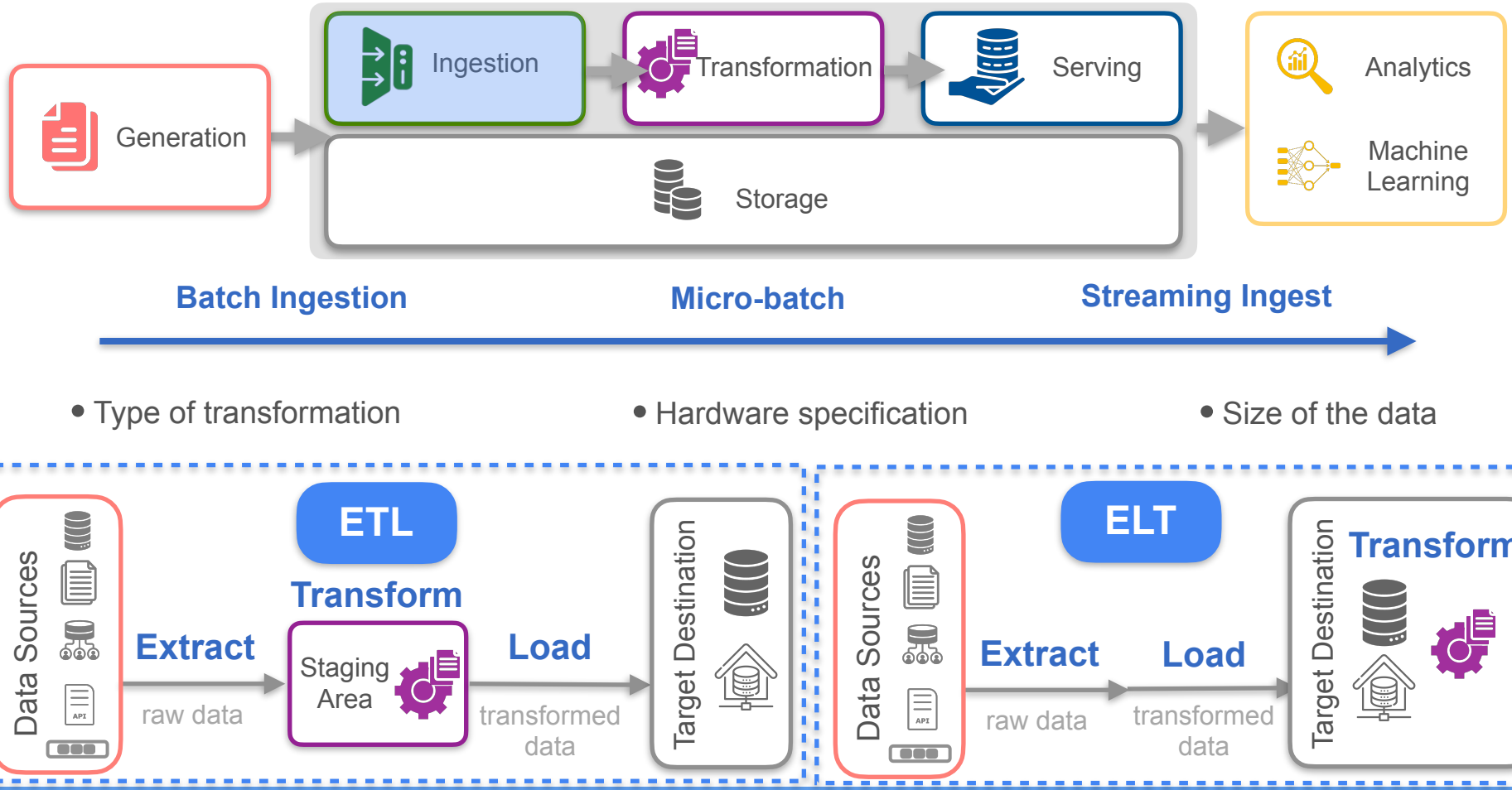
1. Identify tools & tech to meet non-functional requirements
2. Perform cost / benefit analysis and choose between comparable tools & tech
3. Prototype and test your system, align with stakeholder needs

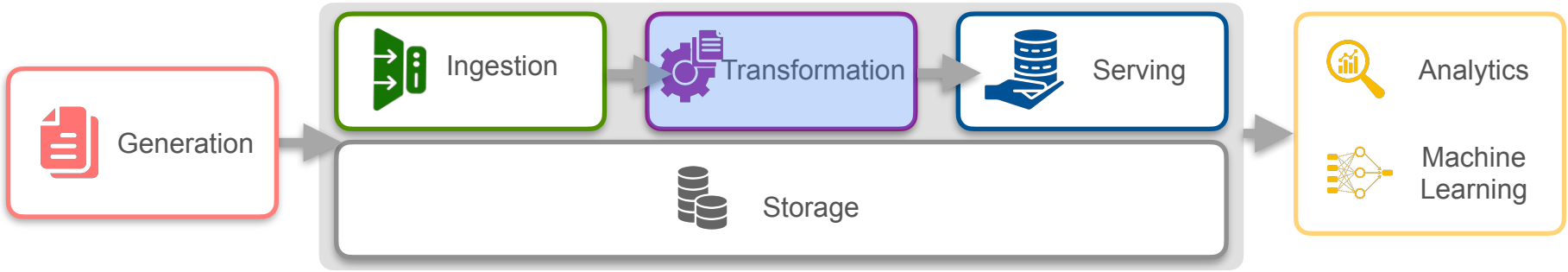
Build, evaluate, iterate & evolve

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


Generation
Ingestion
Transformation
Serving
Storage
Analytics
Machine Learning
Batch Ingestion
Streaming Ingest
Data Sources
API


Generation
Ingestion
Transformation
Serving
Storage
Analytics
Machine Learning
Batch Ingestion
Micro-batch
Streaming Ingest
Infrequent
Real-time
Data Sources
API


Generation
Ingestion
Transformation
Serving
Storage
Analytics
Machine Learning
Batch Ingestion
Micro-batch
Streaming Ingest
• Type of transformation
• Hardware specification
• Size of the data
ETL
Data Sources
Extract
raw data
Transform
Staging Area
Load
transformed data
Target Destination
ELT
Data Sources
Extract
raw data
Load
transformed data
Target Destination
Transform



Transformation

- Cleaning and combining data from multiple sources
- Convert data into a target schema based on a data model

- Apply transformations with SQL queries or Python
- Use a non-distributed or distributed processing tool

pandas Spark

Analytics

Many fields	…	Many fields
…	…	…

…

Process large amounts of data inside a cloud data warehouse to leverage MPP

dbt

Machine Learning

Transformations depend on end use case:

- Data exploration
- Training a machine learning model
- Making predictions

Data Warehouse

- Expects structured datasets with a well-defined schema
- Query optimizer returns results based on best execution plan
- Suitable for analytical workloads because it's based on columnar storage

Data Lake

- Built on top of low-cost object storage
- Supports ML & big data processing
- To prevent creating a data swamp, you can create a data catalog to track and manage data

Data Lakehouse

Low-cost & scalable storage + Structured query & data management

- Serves low-latency analytics and ML

Security

Data Management

DataOps

Data Architecture

Orchestration

Software Engineering

IAM

VPC

Network ACL

172.16.0.0
172.16.1.0
172.16.2.0

Route table

Security group

- Model your data

AWS Glue Data Catalog

- Test & monitor data quality

great expectations

AWS Glue Data Quality

- Infrastructure as code

HashiCorp Terraform

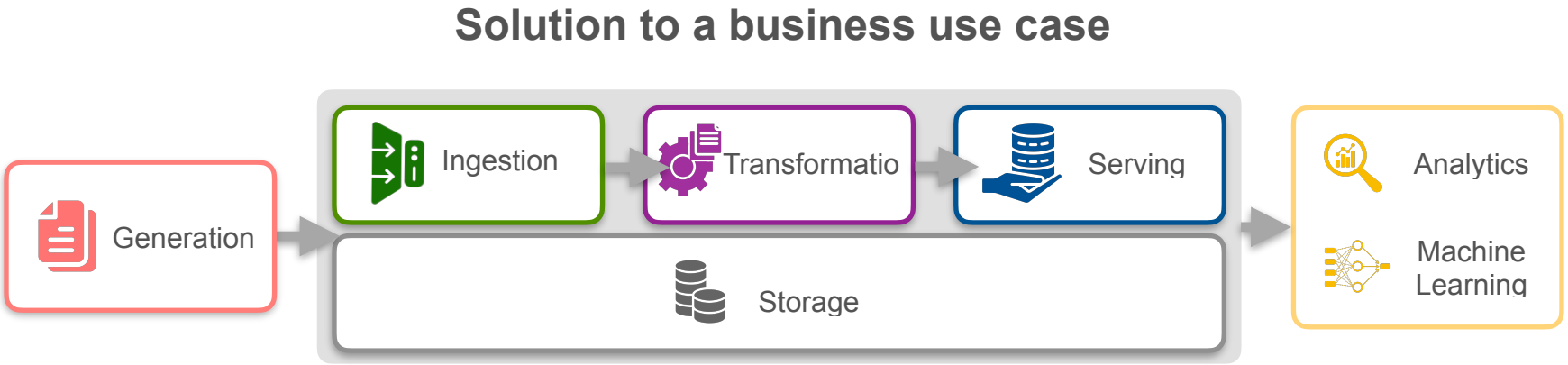
Apache Airflow

DeepLearning.AI

Lab Walkthrough

Capstone Project (Part 1)

Capstone Lab



Part 1 Create and configure the resources for the pipeline

Part 2 Integrate data quality checks and orchestration, & create data visualizations

Capstone Lab

DE Ftunes

- Offers subscription-based music streaming services
- Added a new feature allowing its clients to purchase and download music

GET	/users	Get Users
GET	/sessions	Get Sessions

Data Engineer

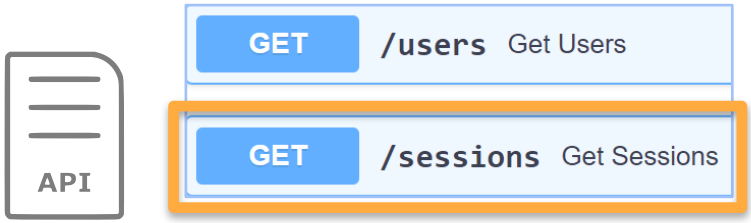
Analysts

Building a data pipeline to serve the purchase data

Analyzing the purchase data

- with respect to the song and artist features
- how the purchases are evolving with time
- how the purchases are affected by the users' personal information

Data Sources



Response of "Get Users" Request


```
{
    "user_id": "a3141825-3a8c-4968-a3af-5362011ef7d5",
    "user_name": "Elizabeth",
    "user_lastname": "Carey",
    "user_location": [
        "46.32313",
        "-0.45877",
        "Niort",
        "FR",
        "Europe/Paris"
    ],
    "user_since": "2020-12-22T14:15:35.936090"
}
```


Response of "Get Sessions" Request


```
{
    "user_id": "6b287203-7cab-4f1a-b1a4-2b5076294682",
    "session_id": "04a5e8ac-1acd-48dc-88b9-651c4ddf489c",
    "session_items": [
        {
            "song_id": "TRXKAGX128F9342DD7",
            "song_name": "3 Cards",
            "artist_id": "AR475MP1187B9A5449",
            "artist_name": "The Balancing Act",
            "price": 1.03,
            "currency": "USD",
            "liked": true,
            "liked_since": "2023-01-27T08:29:54.970697"
        },
        {
            "song_id": "TRUKGBT128F4292C9B",
            "song_name": "Parisian Walls (gband Version_ Barcelona)",
            "artist_id": "ARP9HJX1187FB4E5DA",
            "artist_name": "Apostle Of Hustle",
            "price": 1.31,
            "currency": "USD",
            "liked": true,
            "liked_since": "2023-06-14T00:27:55.876873"
        },
```

Data Sources

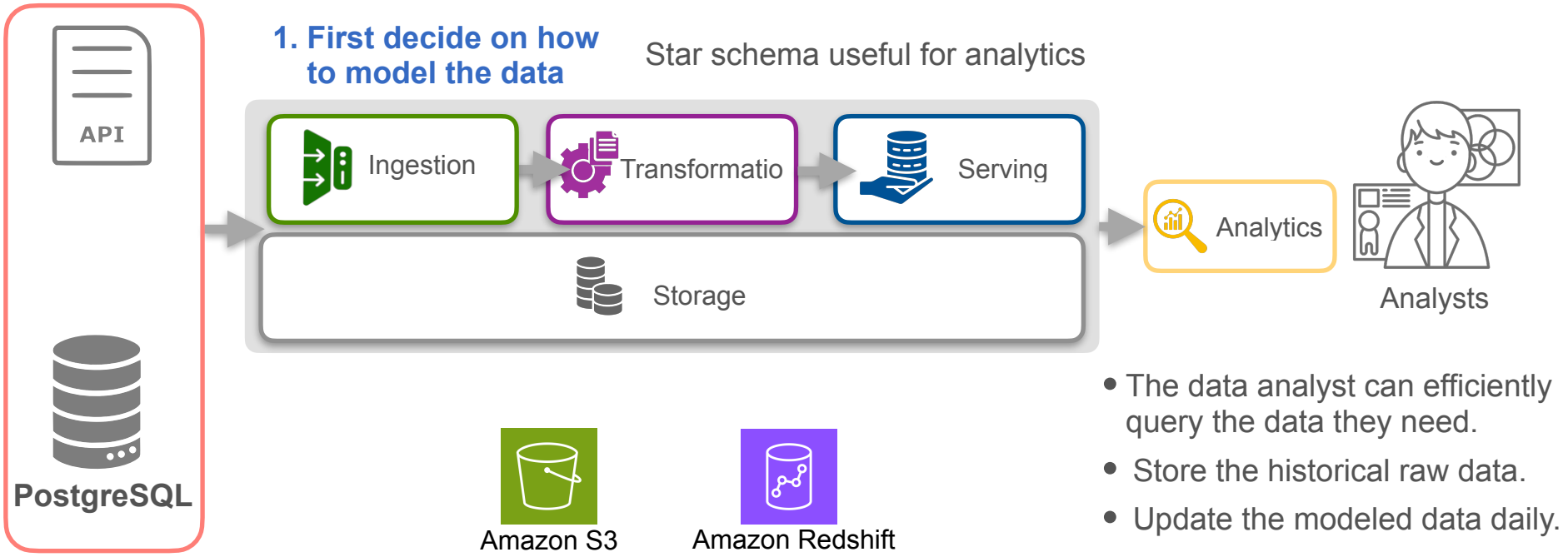
GET	/users	Get Users
GET	/sessions	Get Sessions

PostgreSQL

songs
track_id
title
song_id
release
artist_id
artist_mbid
artist_name
duration
artist_familiarity
artist_hottness
year
track_7digitalid

schema:
deftunes

Requirements

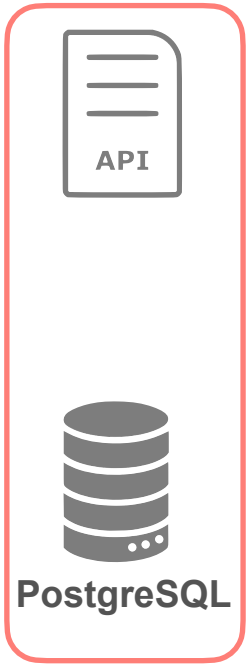


- The data analyst can efficiently query the data they need.
- Store the historical raw data.
- Update the modeled data daily.

Data Model

Business process: Song purchases

Atomic Grain: Individual song item purchased within a session



1. First decide on how to model the data

***Assuming all prices are in USD.*

***If the prices were in different currencies, convert all prices to the same currency.*

dim_users
- user_key (PK)
- user_id
- user_lastname
- user_name
- user_since
- country_code

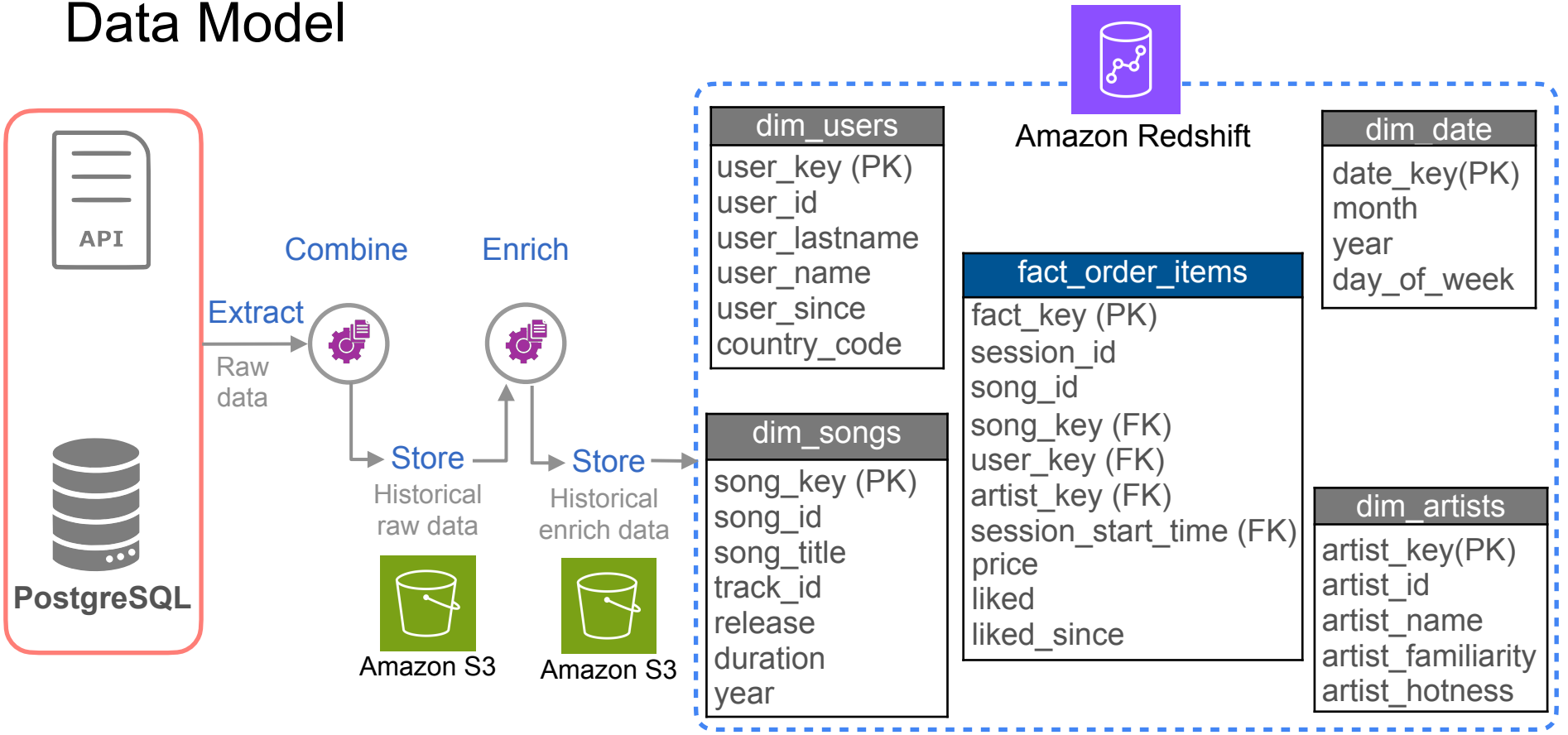
dim_songs
- song_key (PK)
- song_id
- song_title
- track_id
- release
- duration
- year

fact_order_items
- fact_key (PK)
- session_id
- song_id
- song_key (FK)
- user_key (FK)
- artist_key (FK)
- session_start_time (FK)
- price
- liked
- liked_since

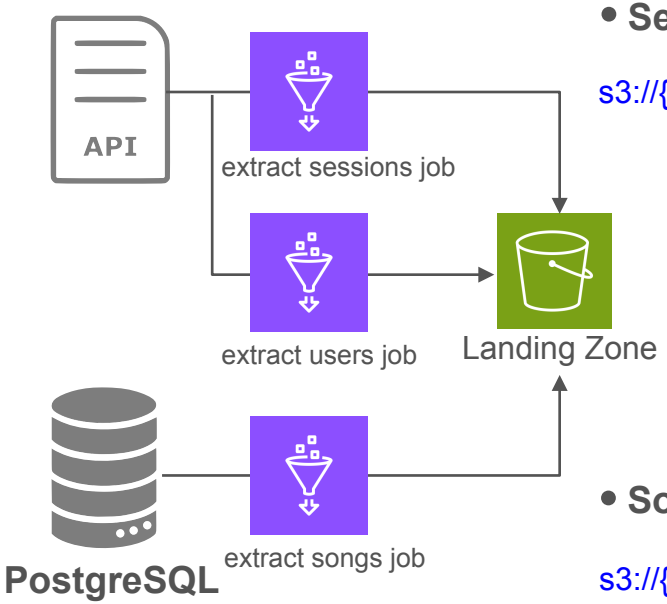
dim_date
- date_key(PK)
- month
- year
- day_of_week

dim_artists
- artist_key(PK)
- artist_id
- artist_name
- artist_familiarity
- artist_hotness

Data Model


API
PostgreSQL
Extract
Raw data
Combine
Store
Historical raw data
Amazon S3
Enrich
Store
Historical enrich data
Amazon S3
Amazon Redshift
dim_users
user_key (PK)
user_id
user_lastname
user_name
user_since
country_code
dim_songs
song_key (PK)
song_id
song_title
track_id
release
duration
year
fact_order_items
fact_key (PK)
session_id
song_id
song_key (FK)
user_key (FK)
artist_key (FK)
session_start_time (FK)
price
liked
liked_since
dim_date
date_key(PK)
month
year
day_of_week
dim_artists
artist_key(PK)
artist_id
artist_name
artist_familiarity
artist_hotness

Solution Architecture



- **Sessions** (json file)

s3://{data_lake_bucket_name}/landing_zone/api/sessions/{current_timestamp}

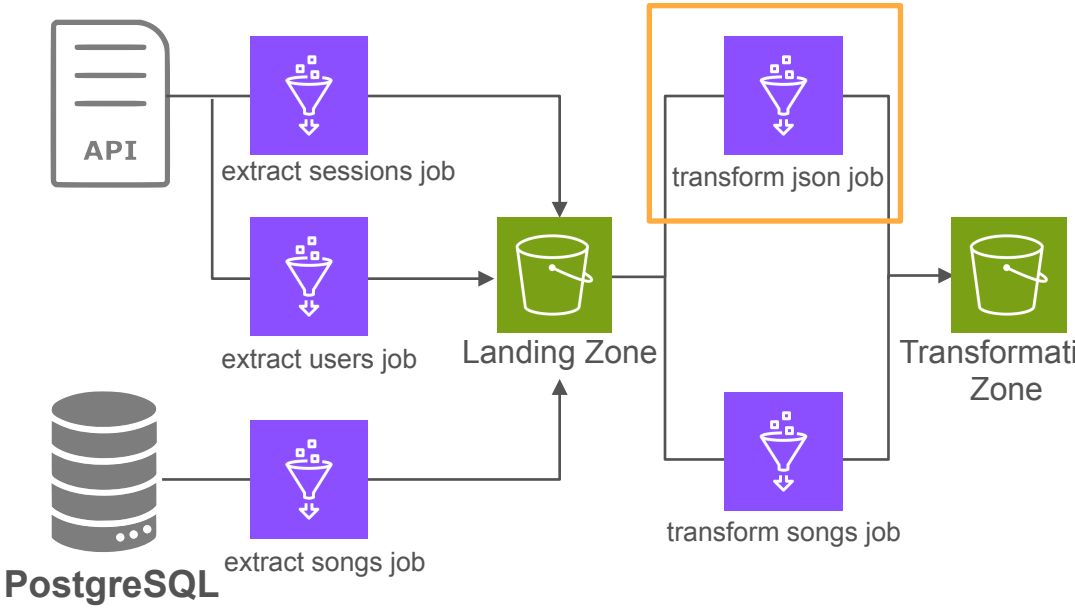
- **Users** (json file)

s3://{data_lake_bucket_name}/landing_zone/api/users/{current_timestamp}

- **Song table** (csv file)

s3://{data_lake_bucket_name}/landing_zone/db_songs/ingest_on={current_timestamp}

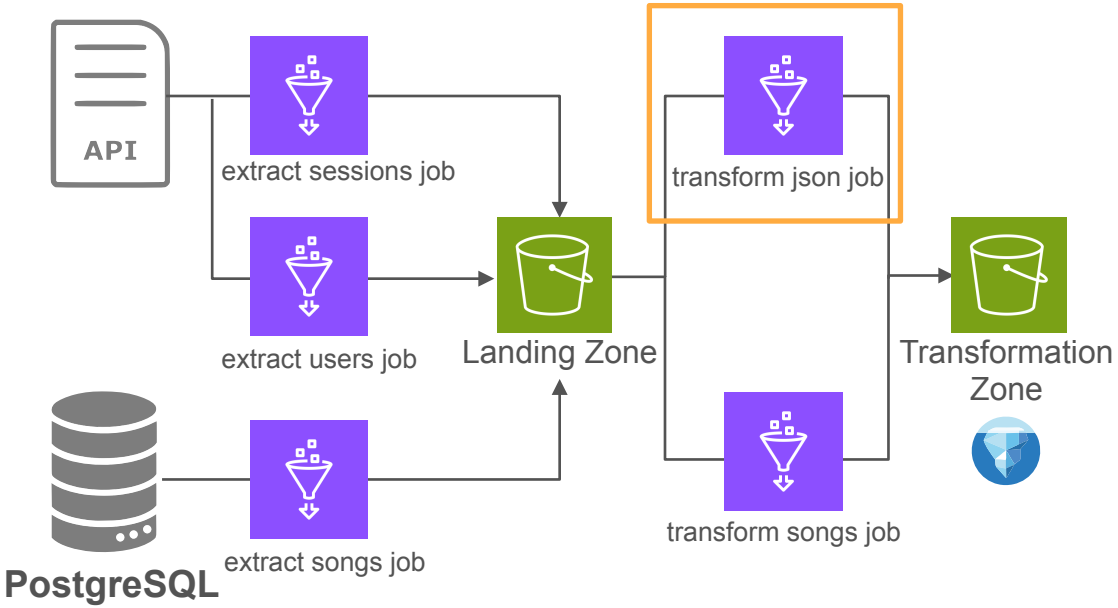
Solution Architecture




```
{
    "user_id": "a3141825-3a8c-4968-a3af-5362011ef7d5",
    "user_name": "Elizabeth",
    "user_lastname": "Carey",
    "user_location": [
        "46.32313",
        "-0.45877",
        "Niort",
        "FR",
        "Europe/Paris"
    ],
    "user_since": "2020-12-22T14:15:35.936090"
}
```


1. Extract the users data into a DataFrame
2. Unnest “user_location” :
    - latitude
    - longitude
    - place name
    - country code
    - timezone
3. Add a column indicating the ingestion time
4. Add a column indicating the processing time

Solution Architecture



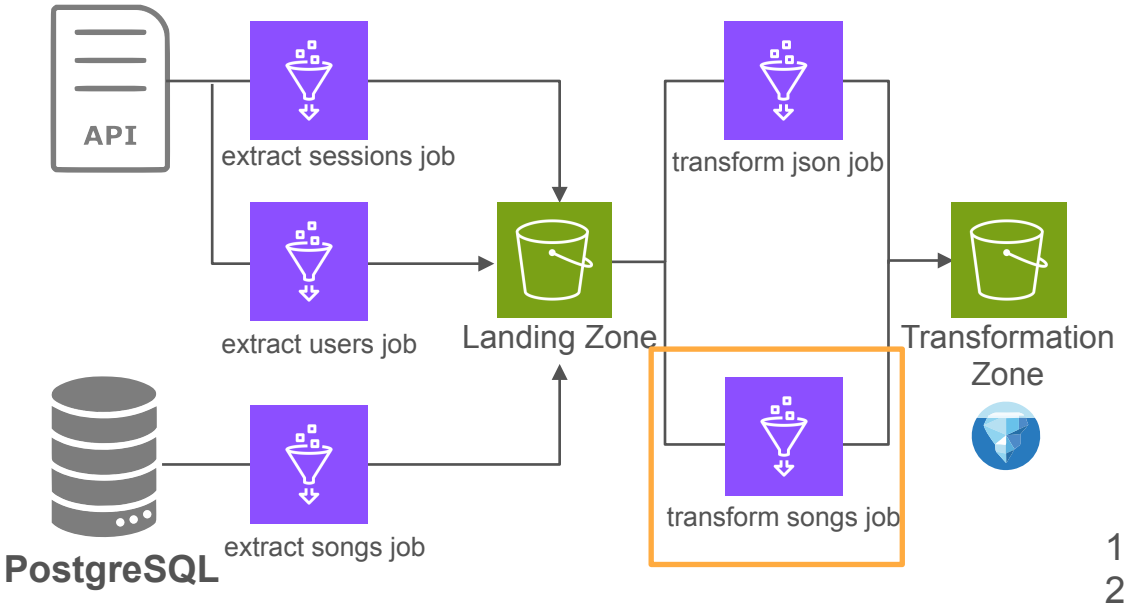

```
{
    "user_id": "6b287203-7cab-4f1a-b1a4-2b5076294682",
    "session_id": "04a5e8ac-1acd-48dc-88b9-651c4ddf489c",
    "session_items": [
        {
            "song_id": "TRXKAGX128F9342DD7",
            "song_name": "3 Cards",
            "artist_id": "AR475MP1187B9A5449",
            "artist_name": "The Balancing Act",
            "price": 1.03,
            "currency": "USD",
            "liked": true,
            "liked_since": "2023-01-27T08:29:54.970697"
        },
        {
            "song_id": "TRUKGBT128F4292C9B",
            "song_name": "Parisian Walls (gband Version_
            "artist_id": "ARP9HJX1187FB4E5DA",
            "artist_name": "Apostle Of Hustle",
            "price": 1.31,
            "currency": "USD",
            "liked": true,
            "liked_since": "2023-06-14T00:27:55.876873"
        },
```


s3://{data_lake_bucket_name}/transform_zone/users

s3://{data_lake_bucket_name}/transform_zone/sessions

1. Extract the sessions data into a DataFrame
2. Unnest “session items”
3. Add a column indicating the ingestion time
4. Add a column indicating the processing time

Solution Architecture

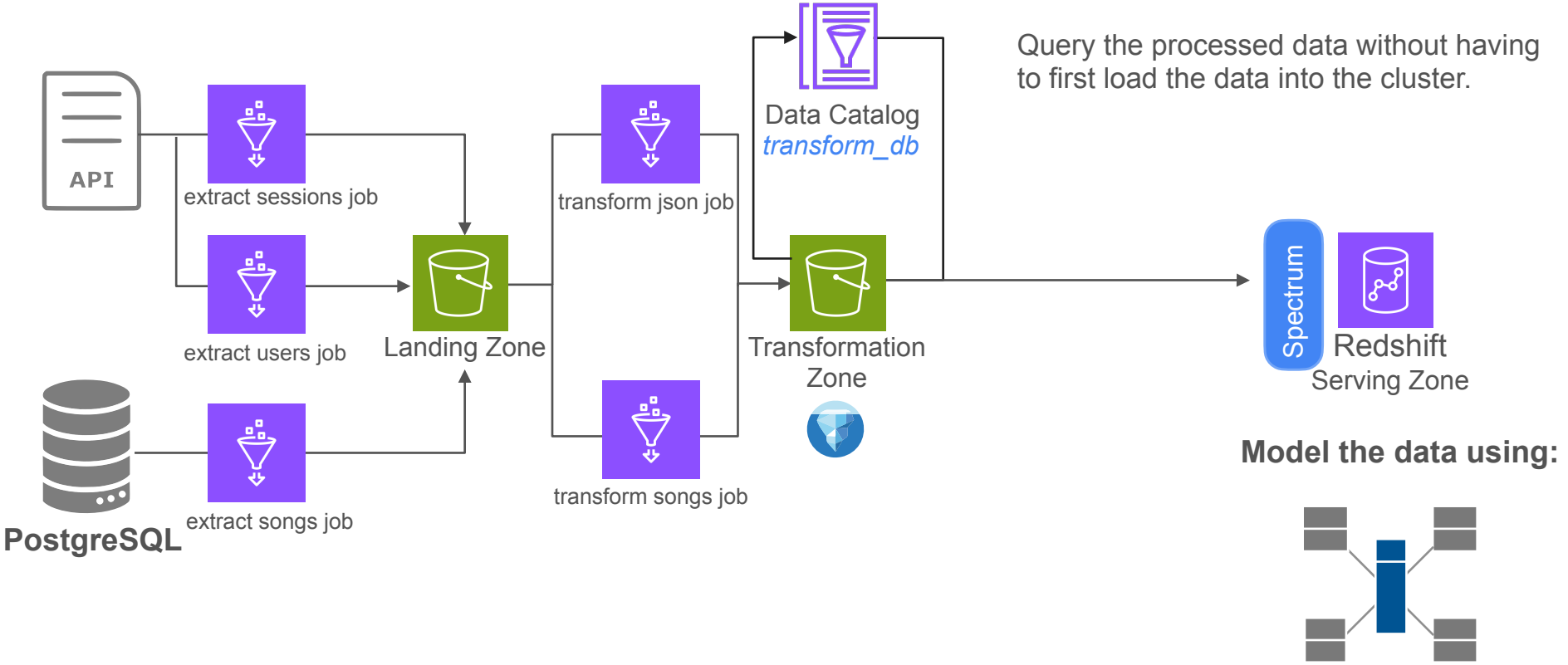


songs
track_id
title
song_id
release
artist_id
artist_mbid
artist_name
duration
artist_familiarity
artist_hottness
year
track_7digitalid

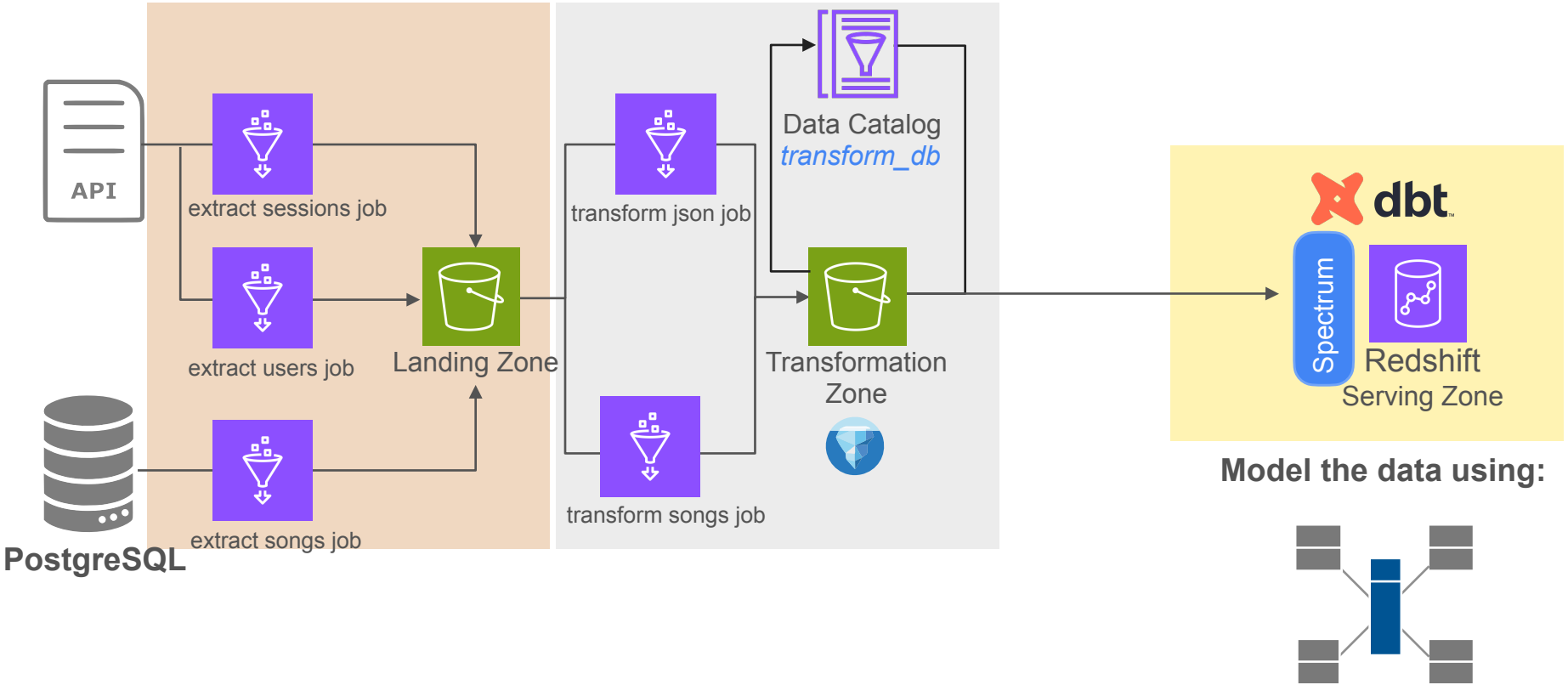
s3://{data_lake_bucket_name}/transform_zone/songs

1. Extract the csv data into a DataFrame
2. Perform schema enforcement
3. Add a column indicating the ingestion time
4. Add a column indicating the name of the database source

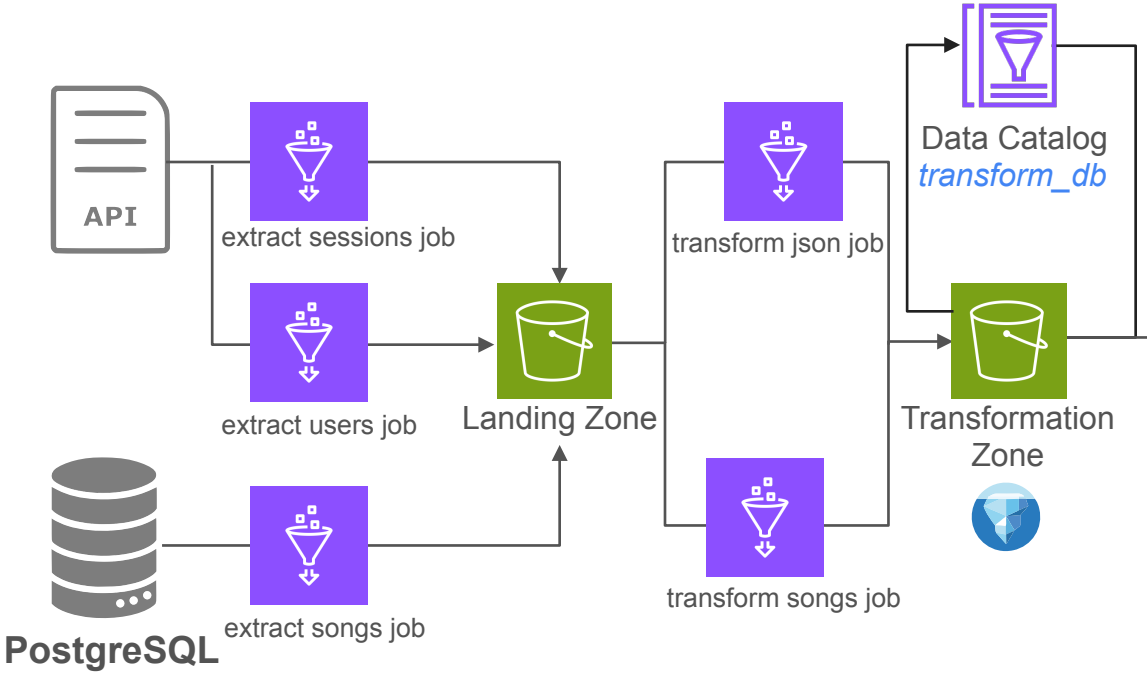
Solution Architecture



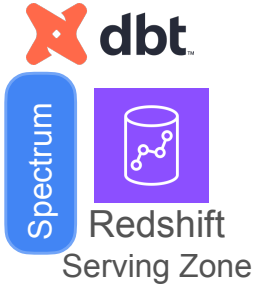
Solution Architecture



Capstone Lab - Part 1



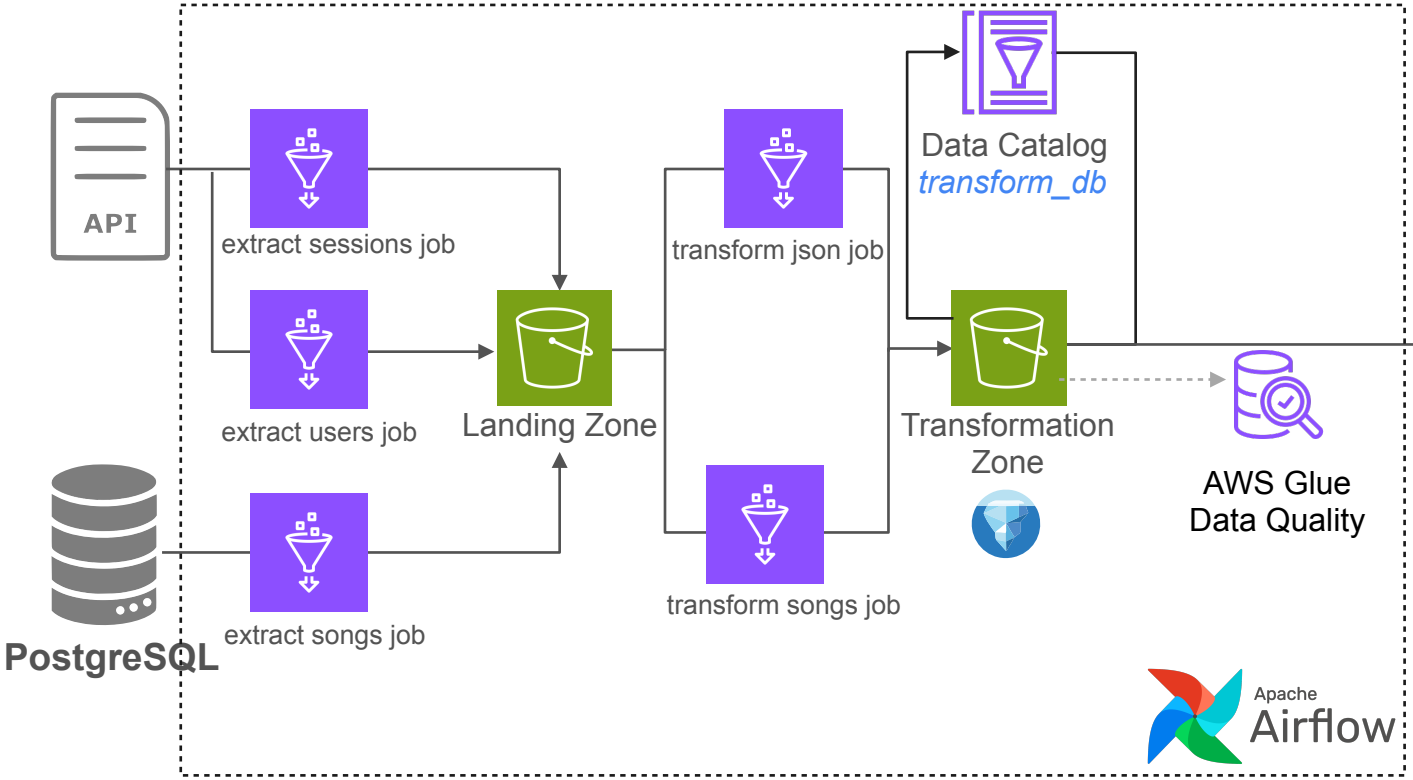
- Set up the dbt project folder
- Optional - Define the tables of the star schema model



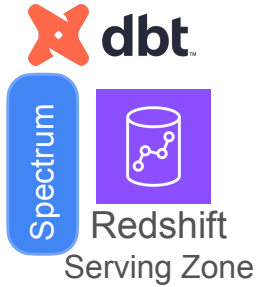
Model the data using:

Use Terraform to create the resources of the pipeline.

Capstone Lab - Part 2



- Build analytical views
- Visualize the views



DeepLearning.AI

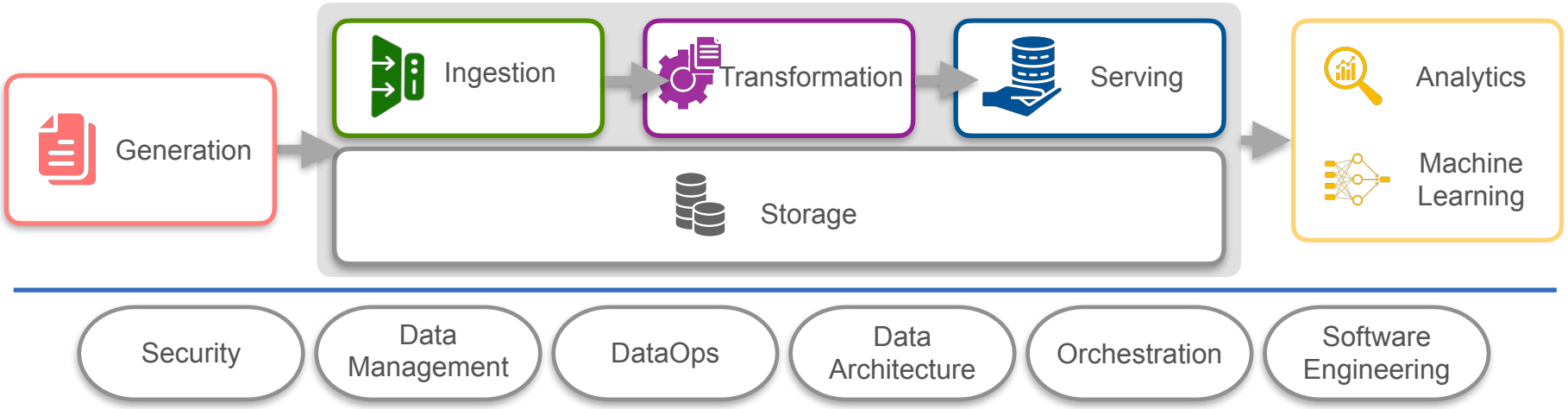
Lab Walkthrough

Capstone Project (Part 2)

DeepLearning.AI

Data Engineering

Conclusion & Thank You


Generation
Ingestion
Transformation
Serving
Storage
Analytics
Machine Learning
Security
Data Management
DataOps
Data Architecture
Orchestration
Software Engineering

The Role of the Data Engineer

Data Team

New role in the realm of data and algorithms

Data Engineer

Software Engineer

Data Scientist

ML Engineer

The Role of the Data Engineer

Data Team

New role in the realm of data and algorithms

Data Engineer

Software Engineer

Data Scientist ML Engineer

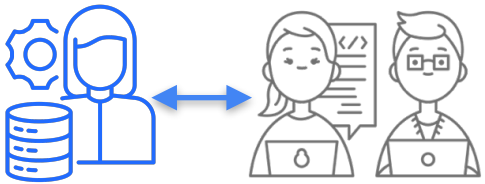
Operational discipline

- Create or use systems that automate the development and operationalization of well-understood ML models
- Monitor data pipelines and quality

The Role of the Data Engineer

Data Team

New role in the realm of data and algorithms



Data Engineer Software Engineer

- Deep understanding in software and DE
- Expertise in streaming, data pipelines, data modeling, and data quality
- Boundaries between application backend systems and DE tools will be lowered

Data Scientist ML Engineer

Operational discipline

- Create or use systems that automate the development and operationalization of well-understood ML models
- Monitor data pipelines and quality