CLOUD ENGINEERING

Data Engineering

Ashish Pujari

Lecture Outline

- Application Architecture
- Data Architecture
- Data Lakes

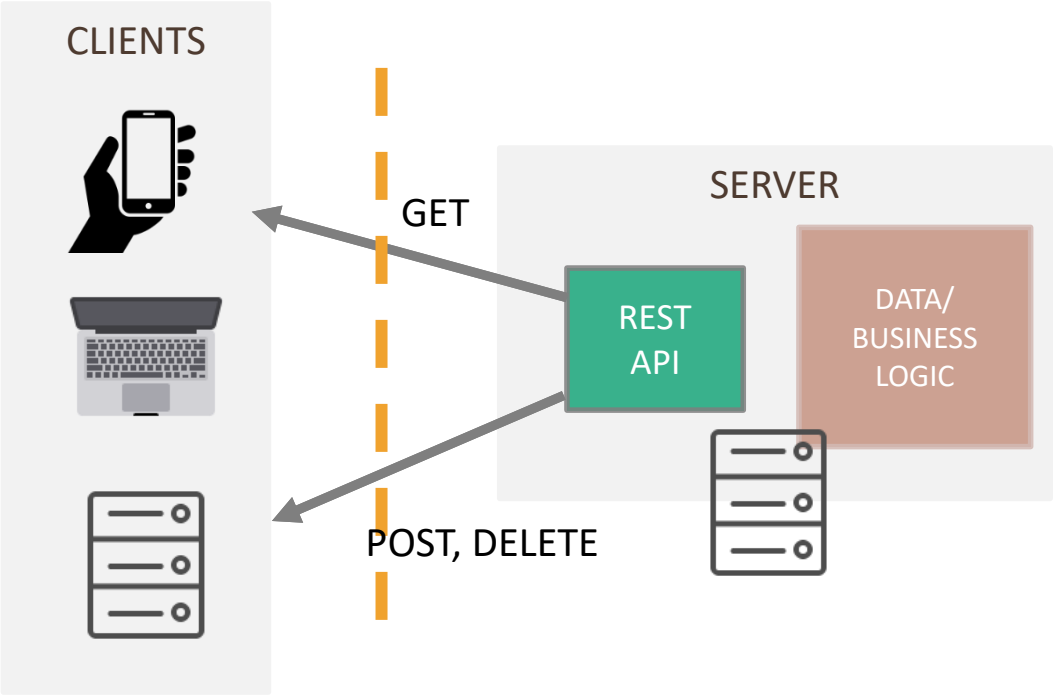
APPLICATION ARCHITECTURE

Application Programming Interfaces (APIs)

- Software components that allows two programs to communicate with each other
- E.g., Weather Service, Stock Quote Service, etc.
- A curated list of public APIs

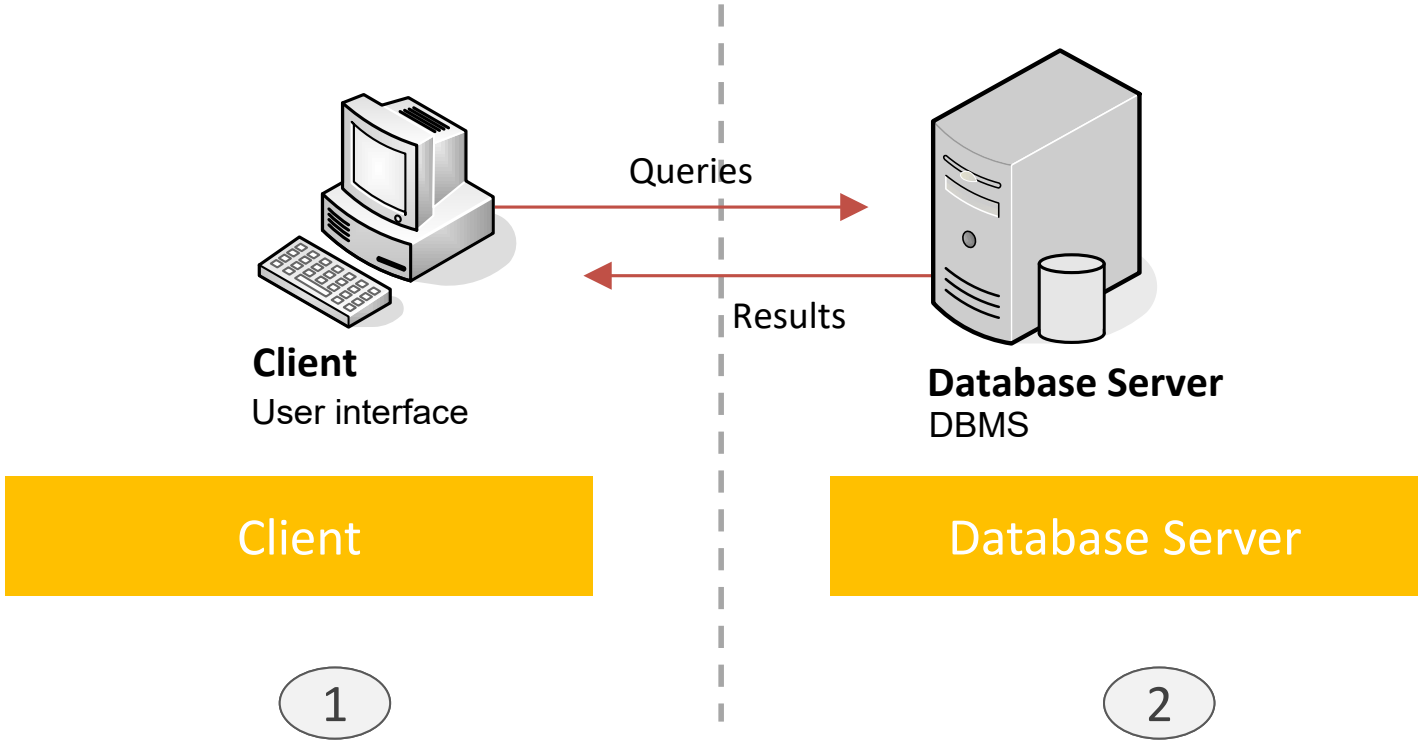
Web Services

- Web Services are APIs that use open protocols and standards
- SOAP, REST, GRPC are widely used web services protocols
- Data exchanged between client and the server is typically in the JSON or XML format

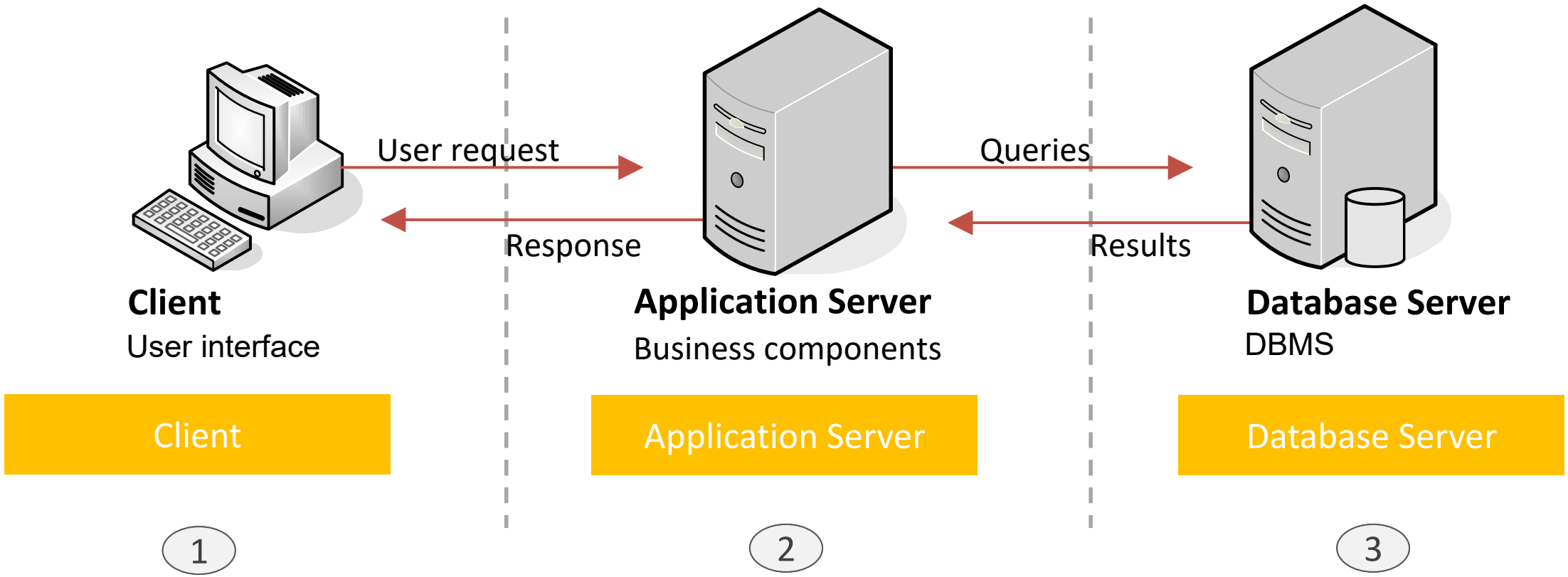


RESTful API

2-tier Application Architecture (Physical)



3-tier Application Architecture (Physical)



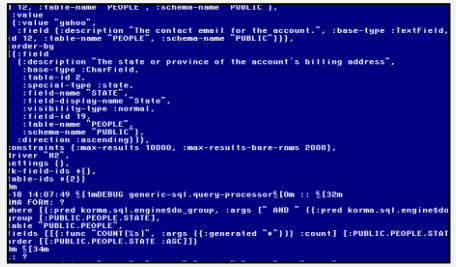
3-tier Application Architecture (Logical)

Presentation Tier

GUI/Frontend

1

Application Tier



Business Logic

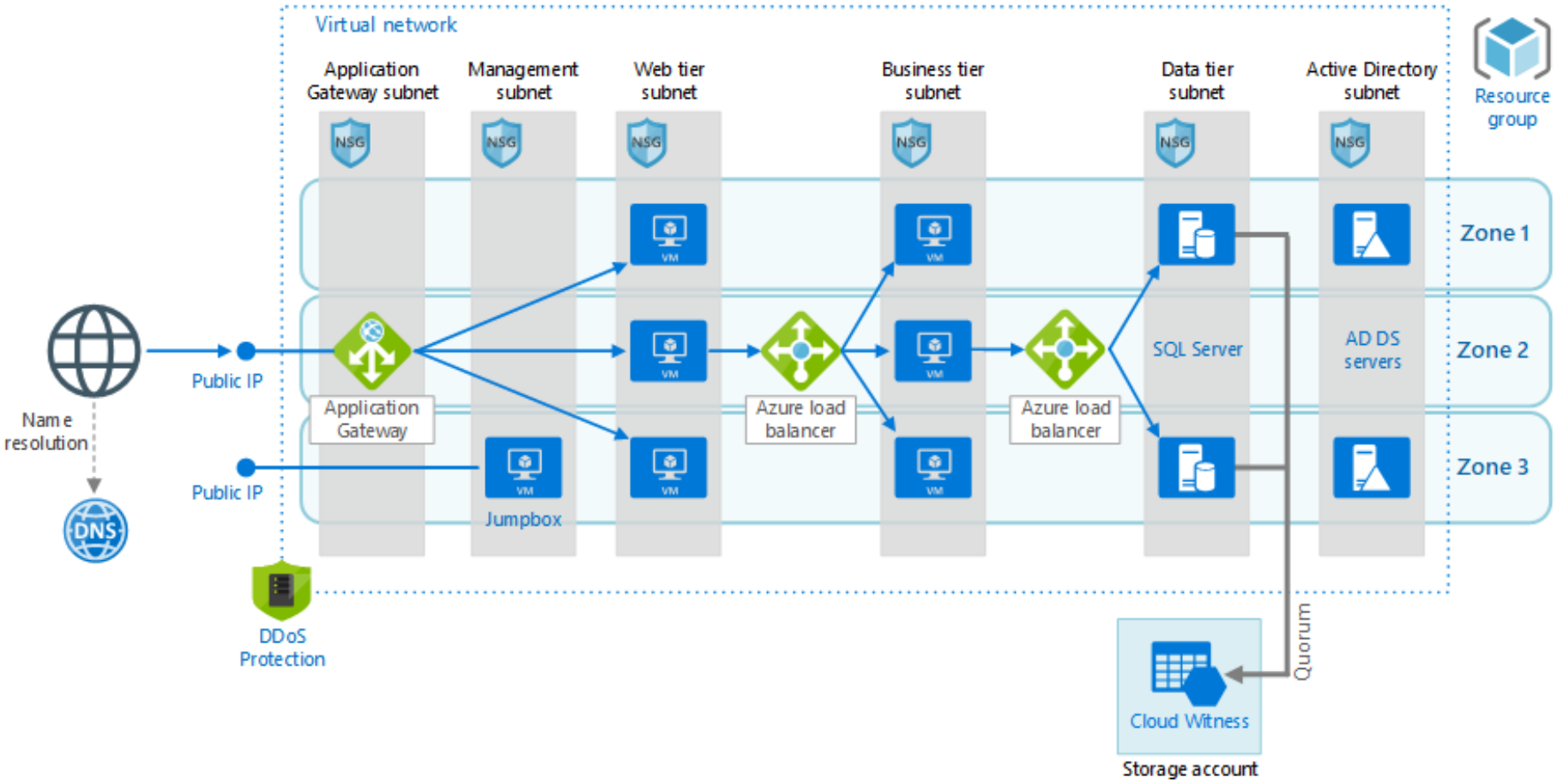
2

Data Tier

Databases/Backend

3

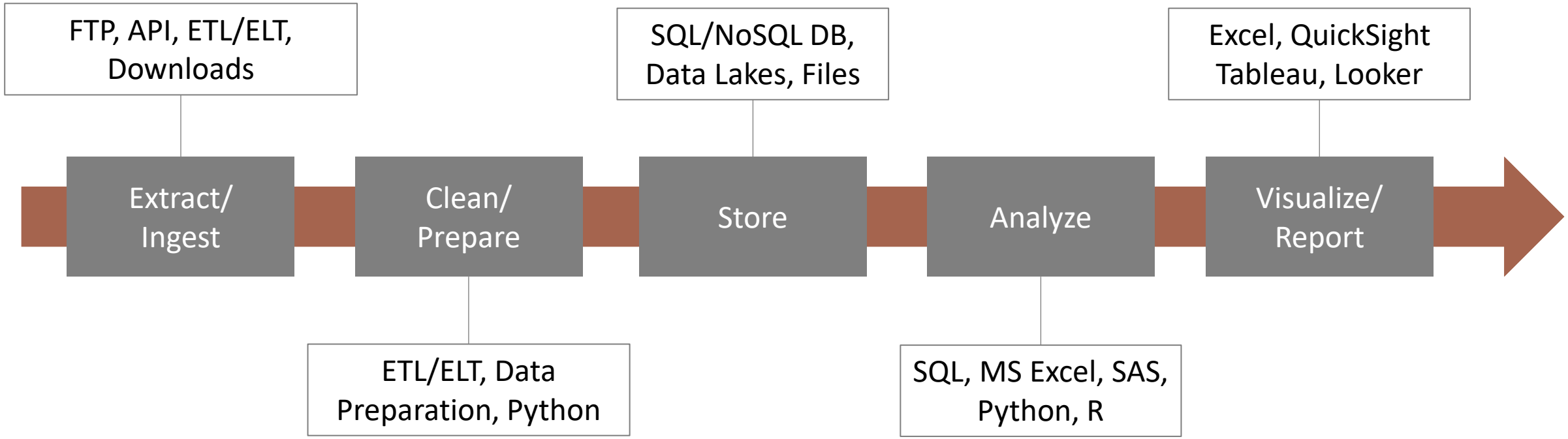
N-Tier Application Architecture



Source: Microsoft

DATA ARCHITECTURE

Data Pipelines



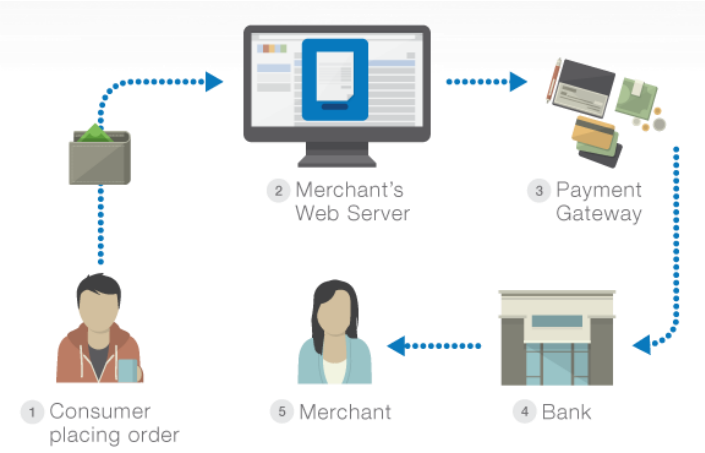
Modern Data Architecture: Characteristics

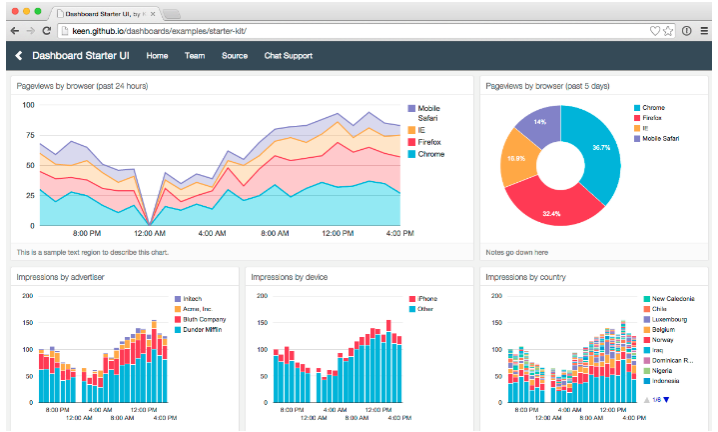
- Built for end-users and self-service analytics
- Automated with data pipelines and data flows
- Resilient and fault tolerant
- Scalable to meet unpredictable demands
- Enables collaboration and trust
- Provides data governance and security by design

Operational and Analytical Databases

Databases

- Operational (OLTP)
- Analytical (OLAP)



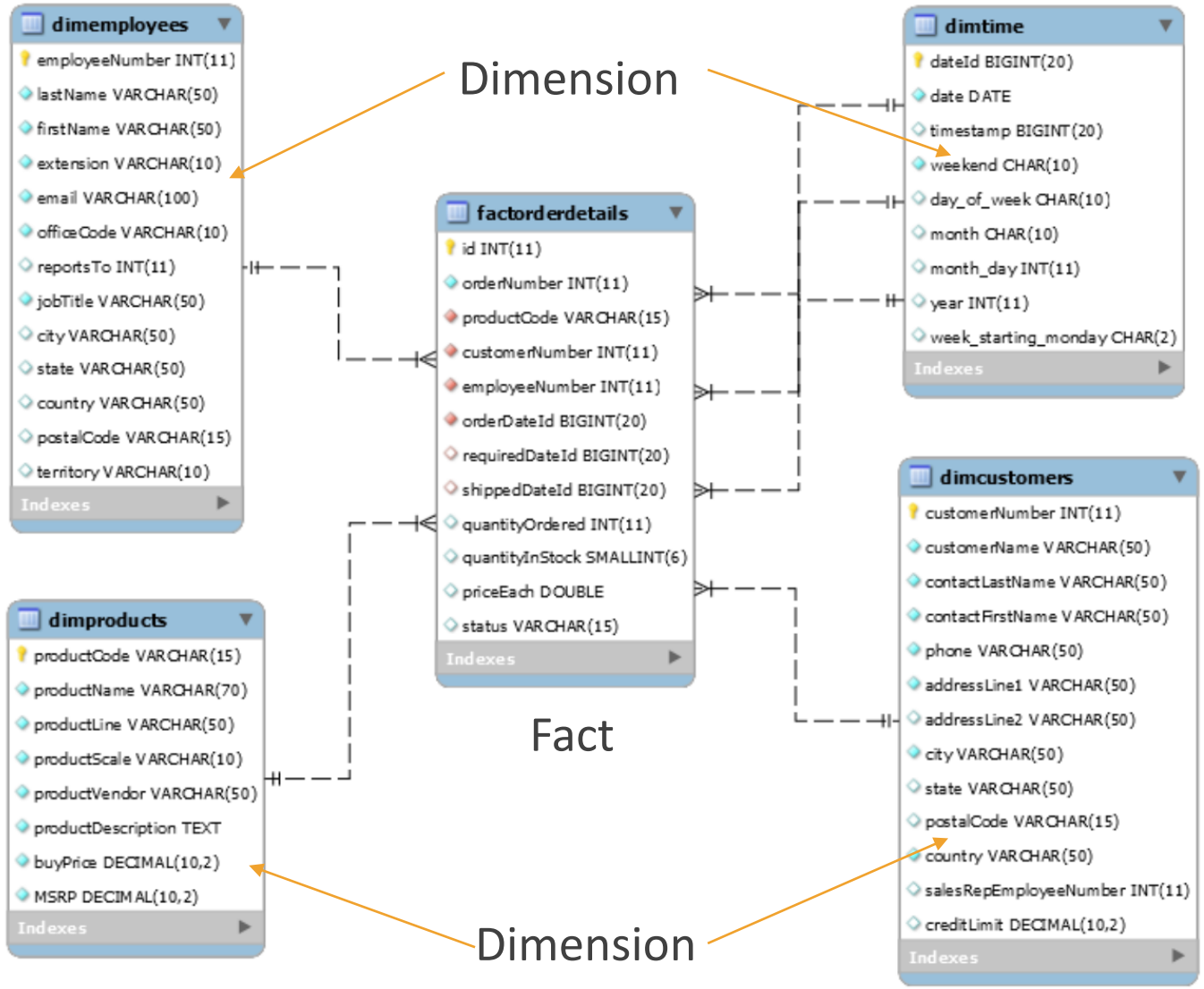


OLTP vs OLAP

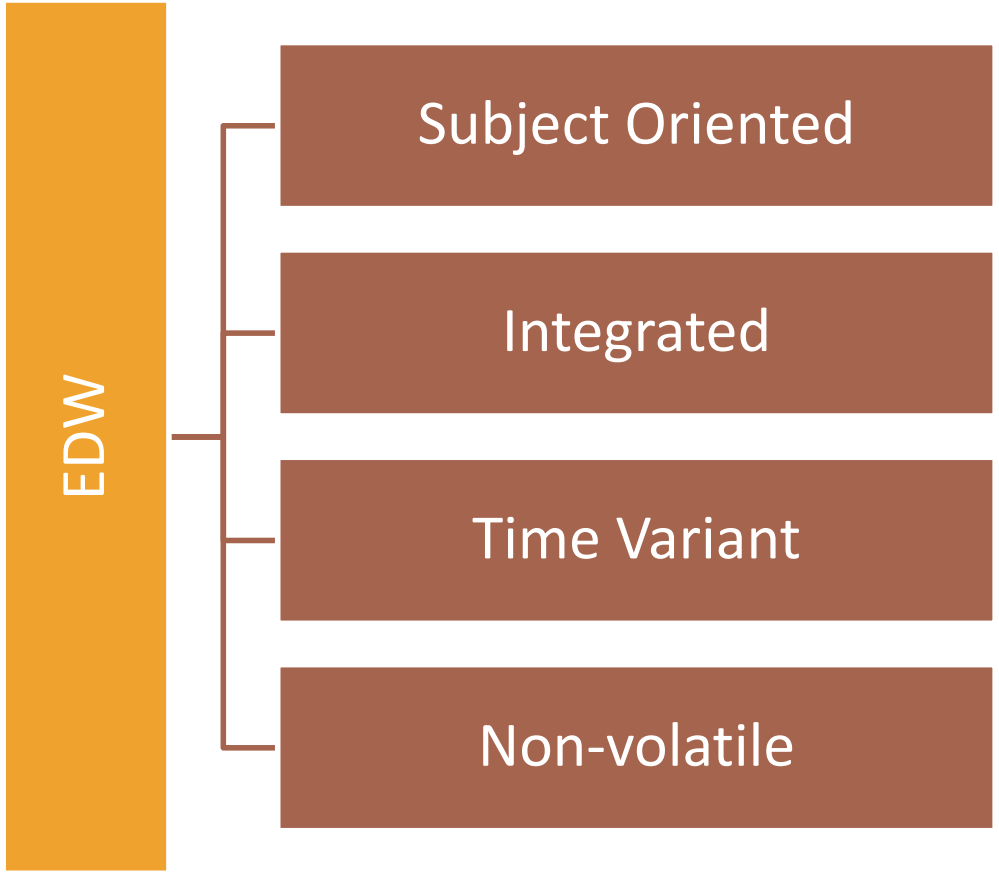
	Transactional (OLTP)	Analytical (OLAP)
Application	Online, Transactional	Reporting, Business Intelligence
Operations	Update	Retrieval
Information	Transactional	Actionable
Data Age	Current	Recent and Historical
Data Size	< 100 GB	> 100 GB
Data model	Entity-relationship	Multi-dimensional
Normalization	Normalized	De-normalized
Function	Application Oriented	Subject Oriented

Data Model: Entity Relationship

Data Model: Star Schema



Enterprise Data Warehouse (EDW)



Developed to analyze specific business subject areas

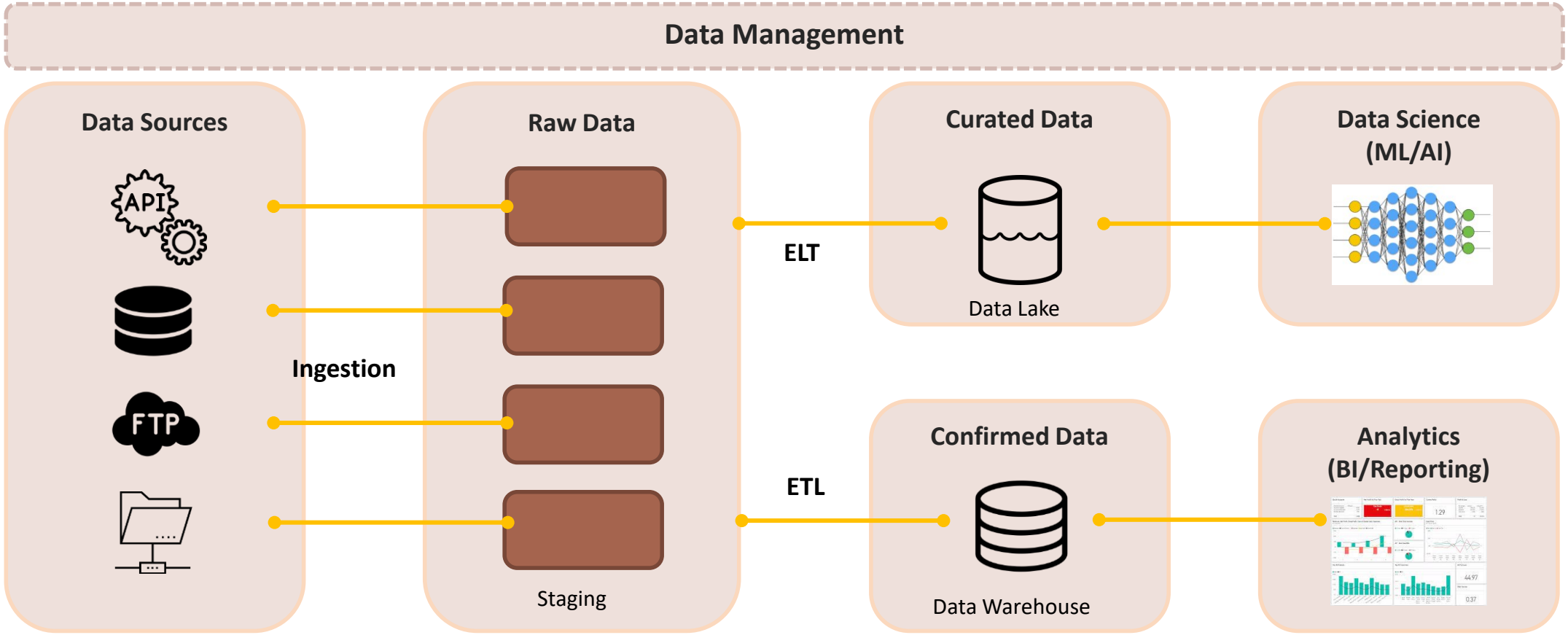
Process of bringing the data from multiple operational data sources into a singular data warehouse

Contains slices or snapshots of data from different periods of time

Once entered into the warehouse, data should not be updated or erased

Extract Transform Load (ETL)

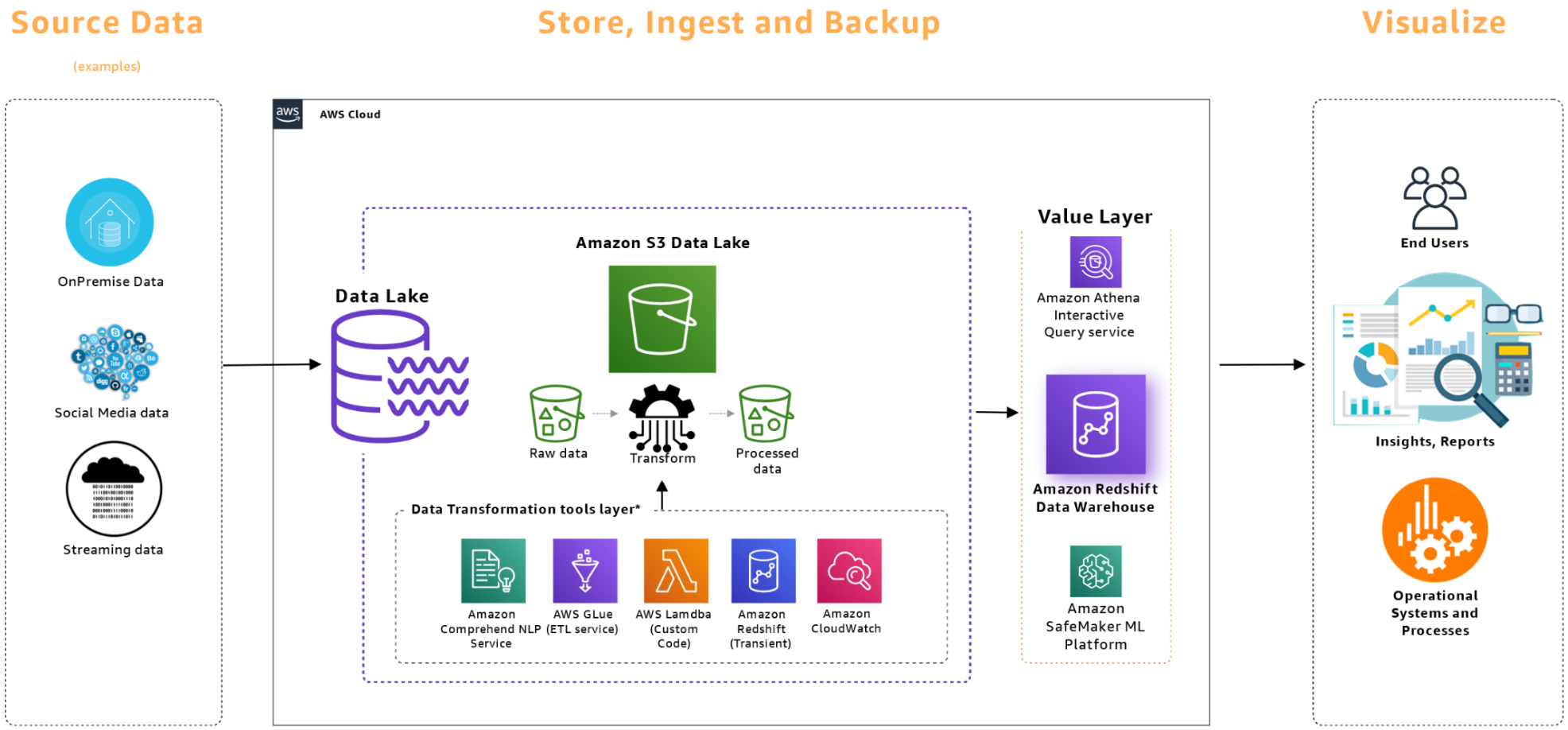
Modern Data Architecture



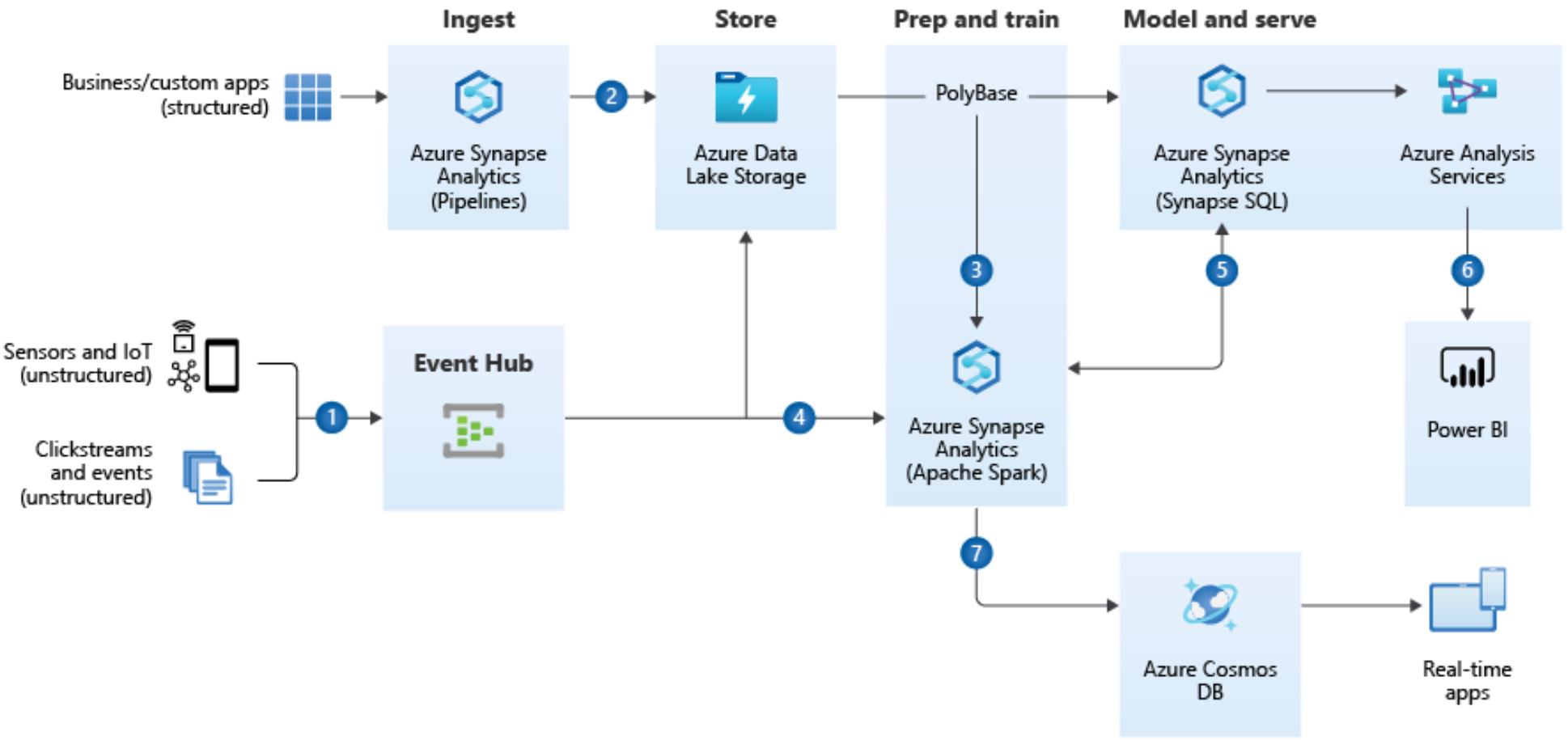
Data Warehouse vs Data Lake

Attribute	Data Warehouse	Data Lake
Schema	Schema on write (predefined schemas)	Schema on read (no predefined schemas)
Scale	Scales to large volumes at moderate cost - limited number of server nodes	Scales to huge volumes at low cost - tens of thousands of storage and compute nodes
Access methods	Accessed through standardized SQL and BI tools	Accessed through SQL-like systems, programs, and other methods
Workloads	Batch processing, concurrent users performing interactive Analytics	Batch processing, stream processing, predictive analytics, improved capability over EDWs for interactive queries
New data	Time consuming to introduce new content	Fast ingestion of new data/content
Cost/efficiency	Efficiently uses CPU/IO.	Efficiently uses storage and processing capabilities at very low cost.
Data Retention	Limited - driven by retention policies	Potential to retain all data (subject to retention policies)
Users	Reporting, Business Intelligence users	Analytics, Data Scientists, Data Engineers
Key Benefits	Provides a single enterprise-wide view of data from multiple sources	Allows usage of raw structured and unstructured data from a centralized low-cost store

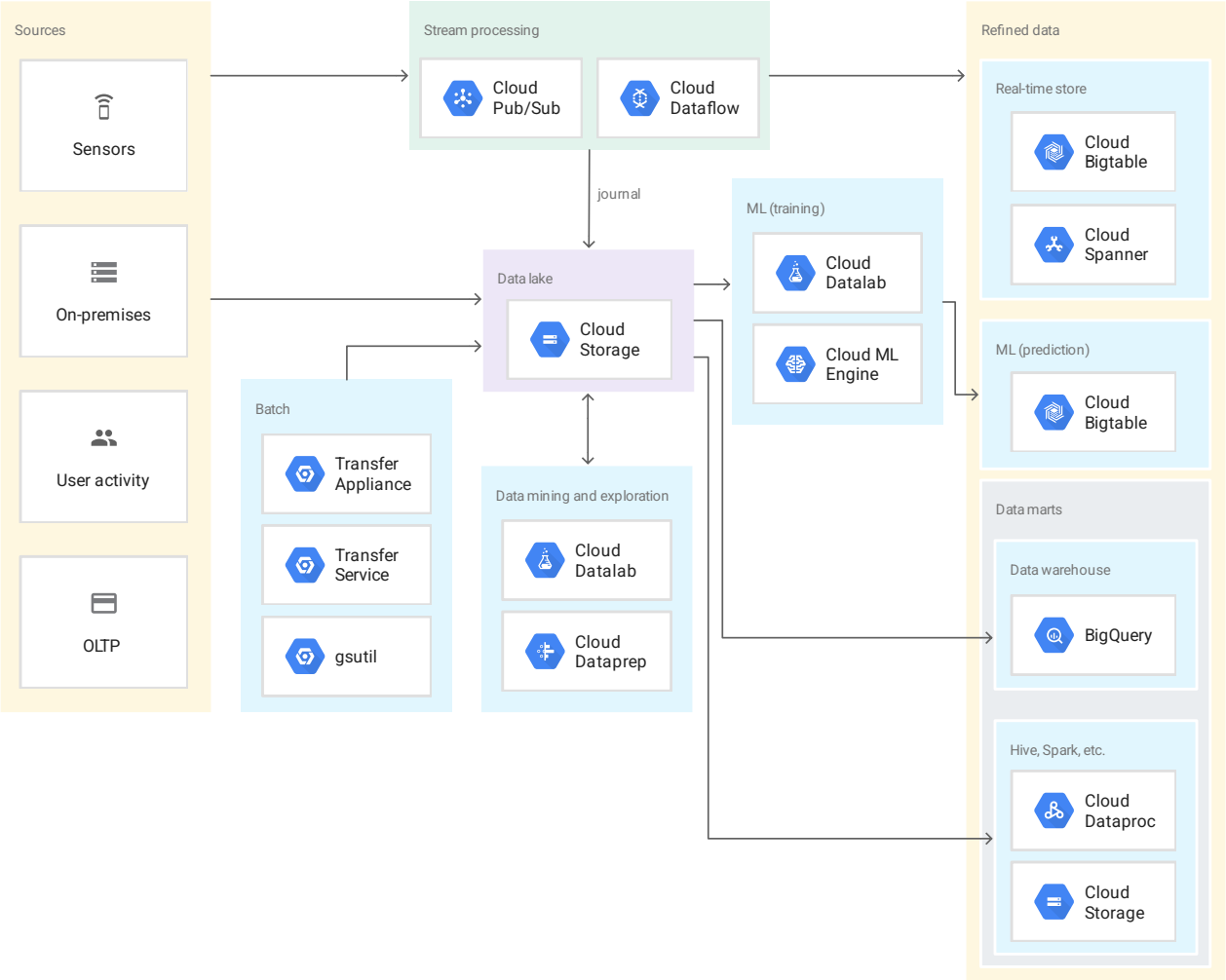
AWS Data Lake Pipeline



Azure Data Lake Pipeline



GCP Data Lake Pipeline



DATA LAKES

Amazon S3

Azure Data lake

Google Cloud Storage

Exercise - Google Cloud Storage (GCS)

- Create storage bucket on GCS
- Download and configure client-side security certificate
- Programmatically connect to GCS using GCS Python API
- Add new storage bucket and files to GCS

- Reference
  - https://googleapis.github.io/google-cloud-python/latest/storage/client.html

Create Cloud Storage Bucket

Cloud Storage

Buckets

Cloud Storage lets you store unstructured objects in containers called buckets. You can serve static data directly from Cloud Storage, or you can use it to store data for other Google Cloud Platform services.

Create bucket or Take the quickstart

← Create a bucket

Name
Must be unique across Cloud Storage. If you're serving website content, enter the website domain as the name.

depa1

Default storage class
Objects added to this bucket are assigned the selected storage class by default. An object's storage class and bucket location affect its geo-redundancy, availability, and costs. You can set storage classes for individual objects in gsutil. Learn more

Nearline and Coldline data in multi-regional locations is now stored geo-redundantly. New locations nam4 and eur4 (available in beta) enable co-location of compute and storage for high performance with geo-redundancy. Learn more

Dismiss

- Multi-Regional
- Regional
- Nearline
- Coldline

Nearline or Coldline are the cheaper options

Location

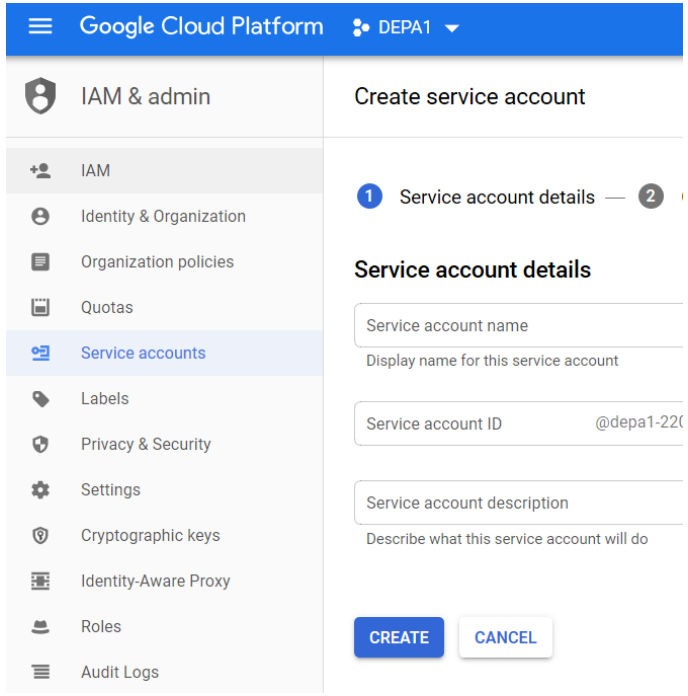
us-central1

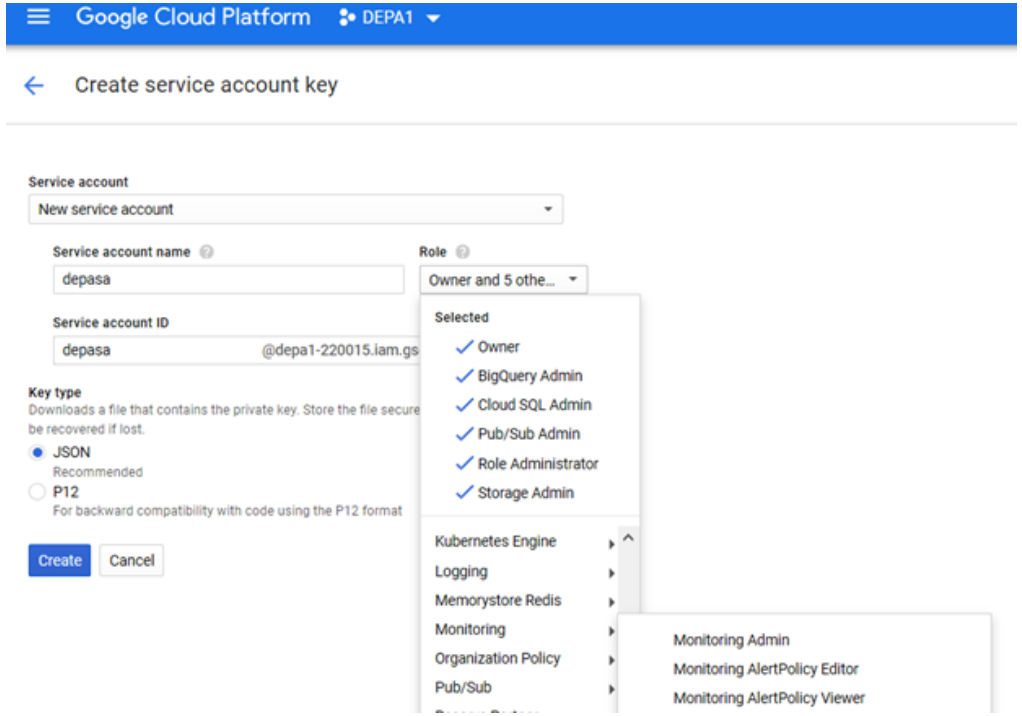
Compare storage classes

Storage cost	Retrieval cost	Class A operations	Class B operations
$0.007 per GB-month	$0.05 per GB	$0.01 per 1,000 ops	$0.005 per 1,000 ops

Show advanced settings

Create Service Account





The service account can be used to access a Cloud API by configuring your code to send credentials for the service account to the service

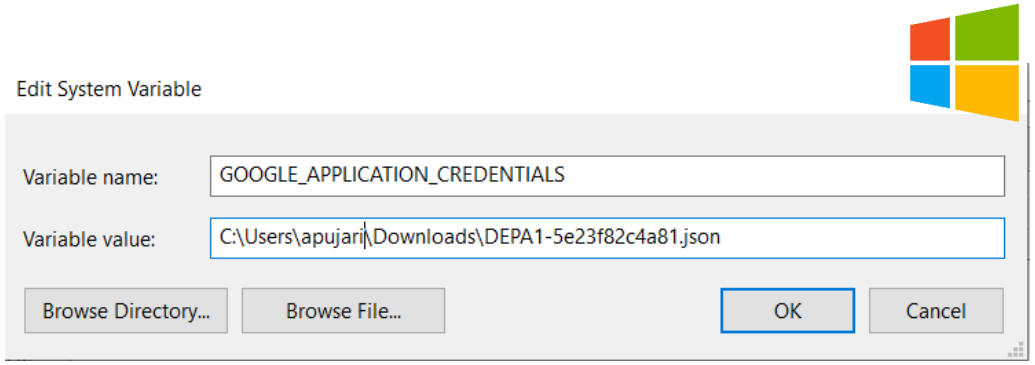
Authenticate using ADC (App Default Credentials)

1. Download Service Account JSON File

```
{
  "type": "service_account",
  "project_id": "project-id",
  "private_key_id": "some_number",
  "private_key": "-----BEGIN PRIVATE KEY-----\n....
  =\n-----END PRIVATE KEY-----\n",
  "client_email": "<api-name>api@project-id.iam.gserviceaccount.com",
  "client_id": "...",
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  "token_uri": "https://accounts.google.com/o/oauth2/token",
  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
  "client_x509_cert_url": "https://www.googleapis.com/...<api-name>api%40project-id.iam.gservic
}
```

2. Set System Environment Variable

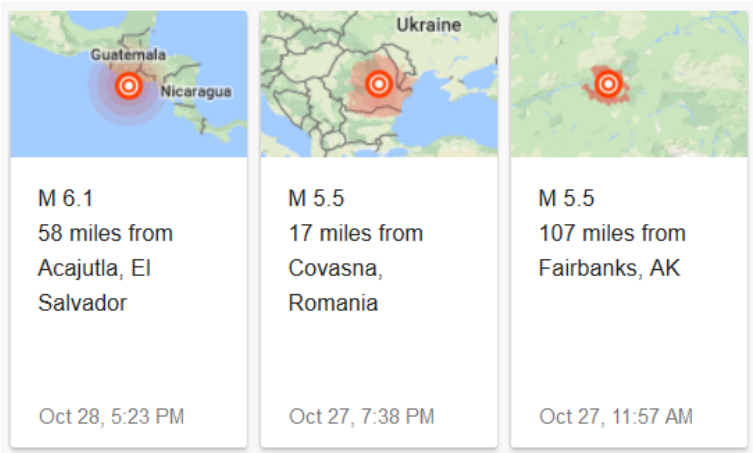
```
Make below changes in ~/.bash_profile
export
GOOGLE_APPLICATION_CREDENTIALS=<path_to_service_acc
ount_file>
```



GCS Python API

- Copy USGS Earthquake data into GCS

```
pip install --upgrade gcloud
pip install --upgrade google-cloud-storage
```



https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/all_week.csv

Exercise – Amazon S3

Amazon S3 > Buckets > Create bucket

Create bucket Info

Buckets are containers for data stored in S3. Learn more

General configuration

Bucket name

bigdata-ashish

Bucket name must be globally unique and must not contain spaces or uppercase letters. See rules for bucket naming

AWS Region

US East (Ohio) us-east-2

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Choose bucket

amazon S3

Buckets (3) Info

Buckets are containers for data stored in S3. Learn more

Copy ARN | Empty | Delete | Create bucket

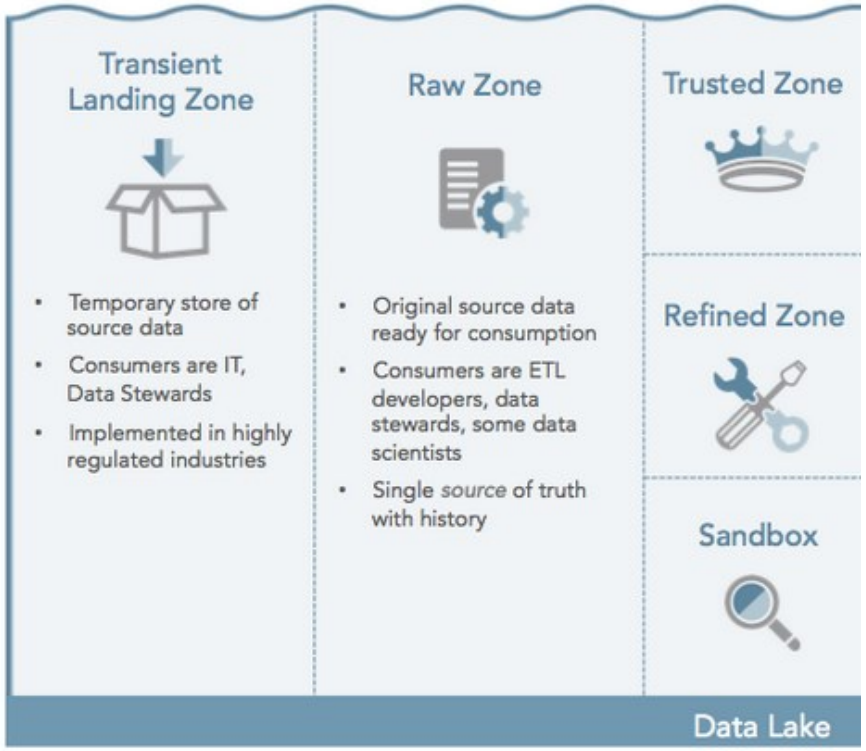
Find buckets by name

1

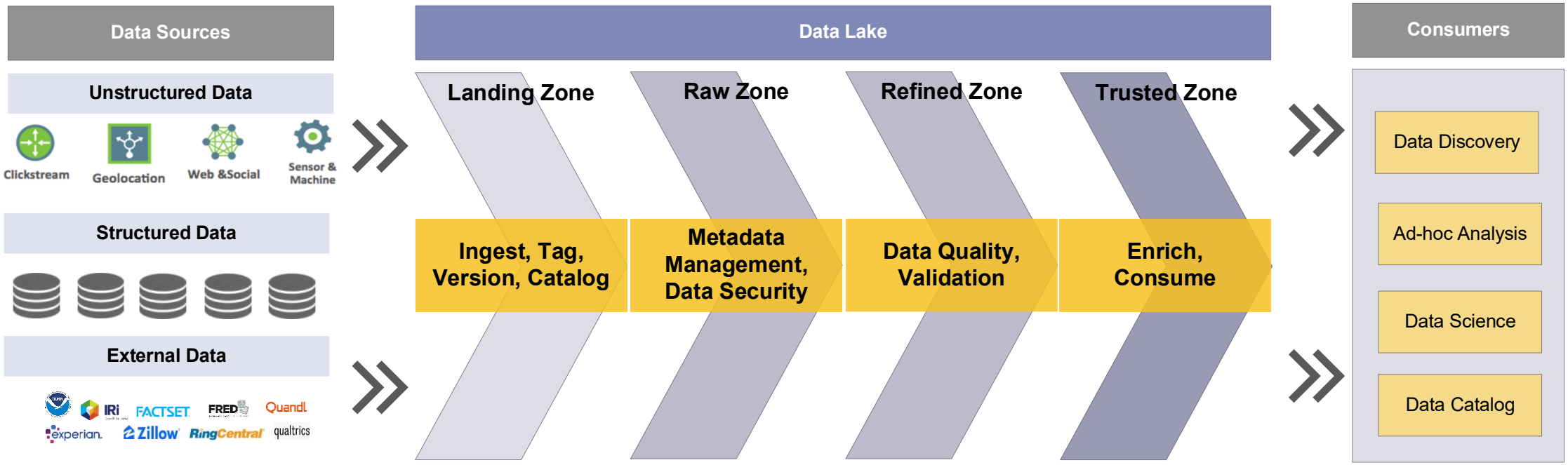
Name	AWS Region	Access	Creation date
bigdata-ashish	US East (Ohio) us-east-2	Bucket and objects not public	December 10, 2022, 21:20:41 (UTC-06:00)

Data Lake Management

- Questions about data
  - Where does your data live?
  - What types of data do you have?
  - What's happening to your data?
  - Is your data accurate and secure?
  - How can you avoid technology or vendor lock-in?
  - How will you be able to leverage future industry innovations?
- Solution
  - Enterprise-ready data lake management for self-service data ingestion and data preparation, integrated metadata management, governance, security



Data Lake Governance



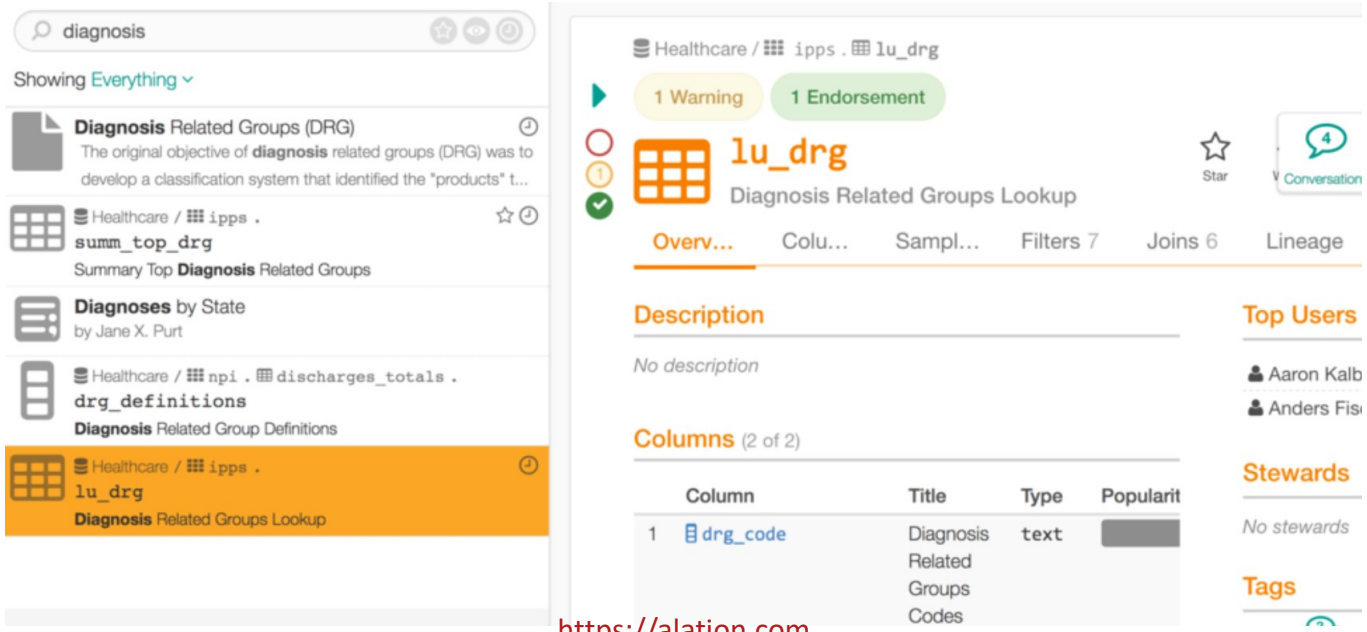
Capability to store, process and archive both raw structured and unstructured data

Governance process ensures that data flows seamlessly and securely from the landing zone to the refined zone before it is ready for consumption. Data retention and security best practices are applied in each zone.

Business users can securely access and share data

Data Catalog

- A data catalog serves as a searchable business glossary of data sources and common data definitions gathered from automated data discovery, classification, and cross-data source entity mapping



https://alation.com

- Automated data population
- Crowdsourced ratings, reviews and tagging
- Enterprise scalability,
- Open APIs for integration
- Search, Data lineage

Data Catalog Reference Architecture