Azure Databricks end to end project with Unity Catalog CICD

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Project Architecture

Pedal Cycle

Two Wheeler

LGV

Unity Catalog - Governance

/landing

Bronze Layer

Silver Layer

Gold Layer

Power BI

Container

Bronze Schema

Silver Schema

Gold Schema

Azure Data Lake Storage Gen 2

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

+ Databricks Unity Catalog +

Continuous Integration + Continuous Deployment

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Prerequisites

- No experience needed for Azure Databricks , we will start from Scratch
- An Azure account for hands-on practical
- Basic knowledge on Python and SQL
- Basic knowledge on Azure Cloud Environment

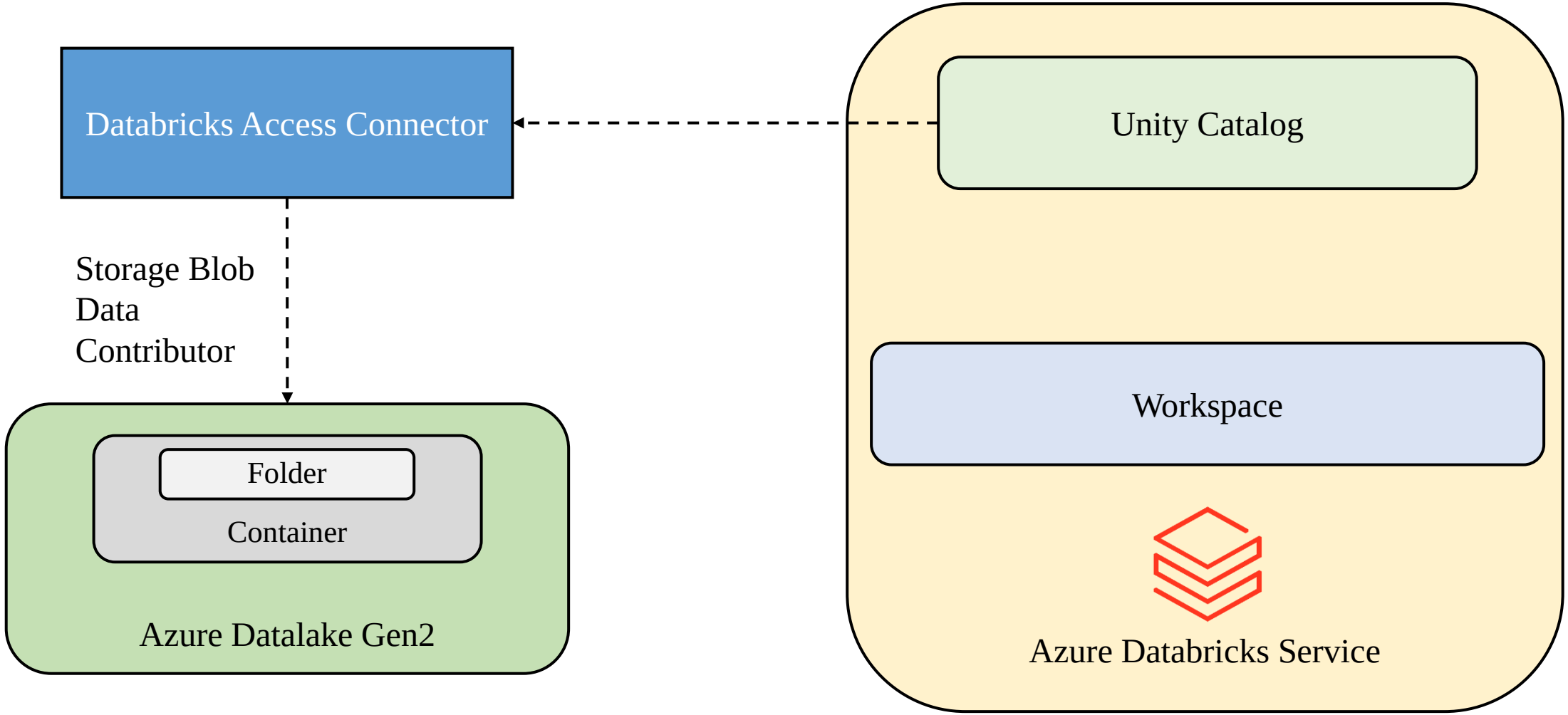
What you'll get from this course?

- Nearly 15+ hours of updated learning content
- Hands-on end to end project
- Practical understanding on Delta lake
- Implementing CICD in Databricks
- Lifetime access to this Course
- Certificate of completion at end of the course

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Environment Setup

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

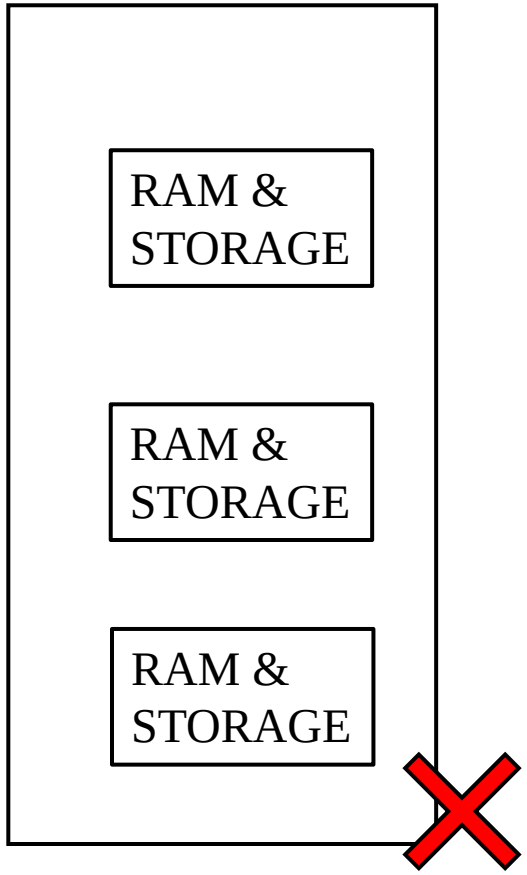


Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

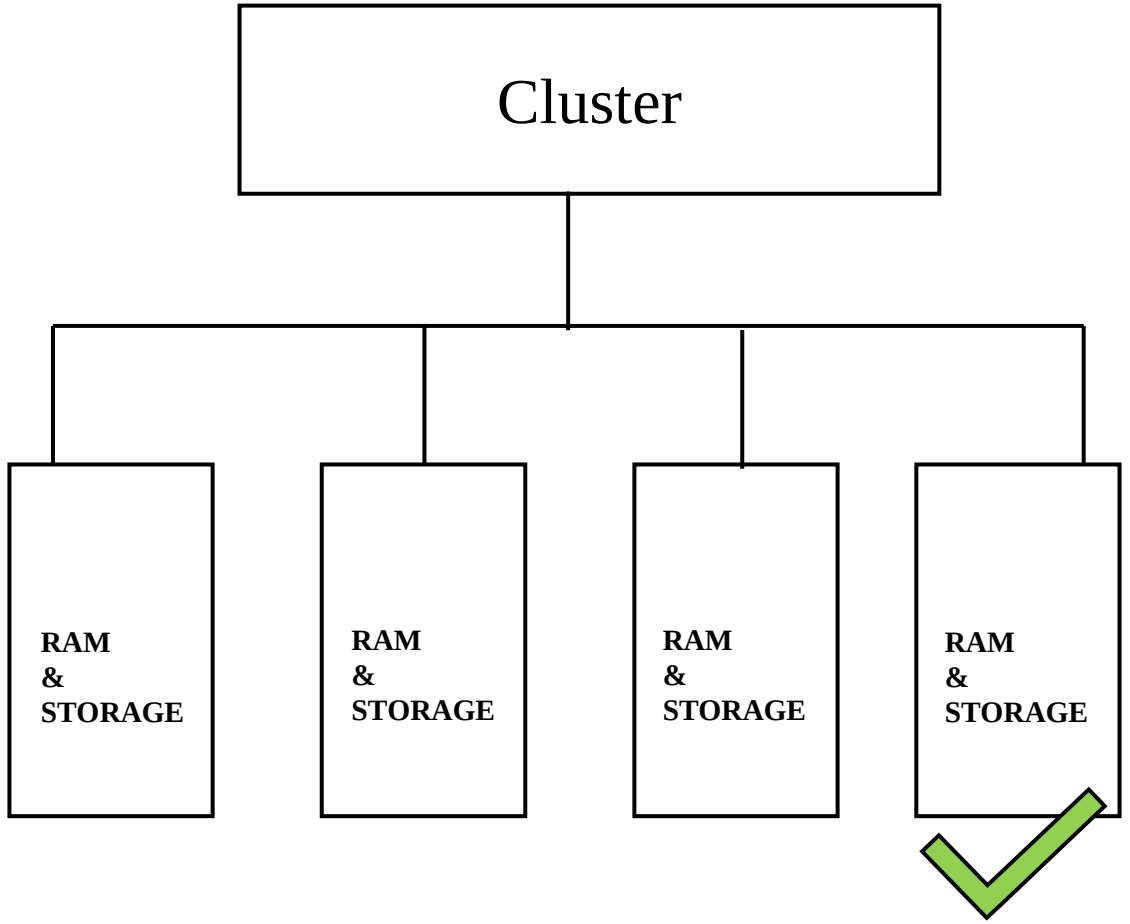
Azure Databricks – An Introduction

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Big data approach



Single Computer for Data Storage and Processing (Monolithic)

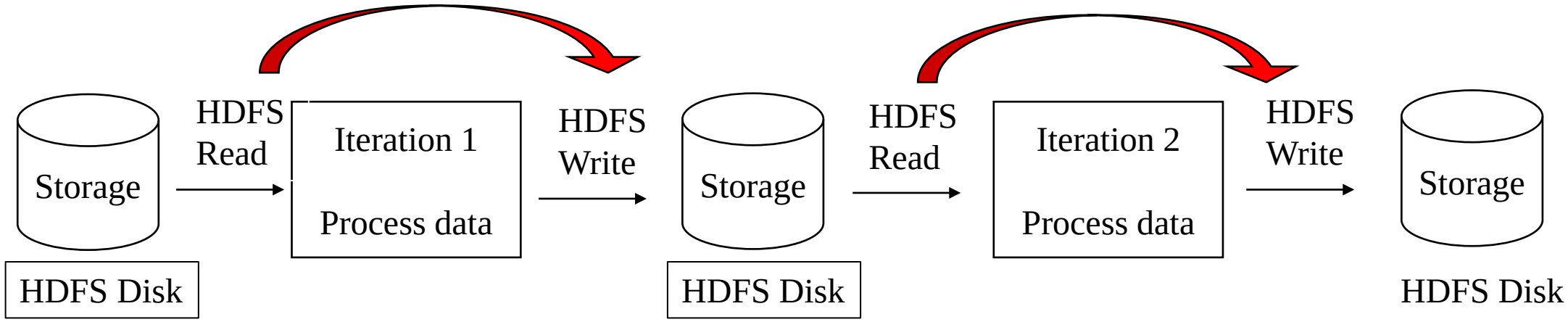


Distributed Approach (adding multiple machines to achieve parallel processing)

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

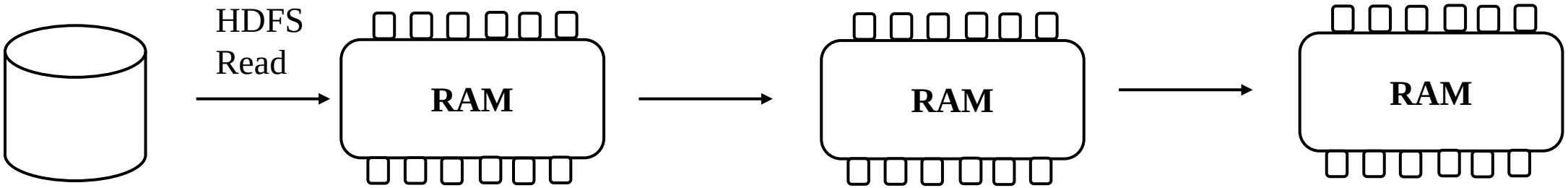
Drawbacks of MapReduce

Traditional Hadoop MapReduce processing



Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Emergence of spark



HDFS Disk
Or
Any cloud
storage

Iteration 1
Analyse Data

Iteration 2
Analyse Data

Iteration 3
Analyse Data

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Apache Spark

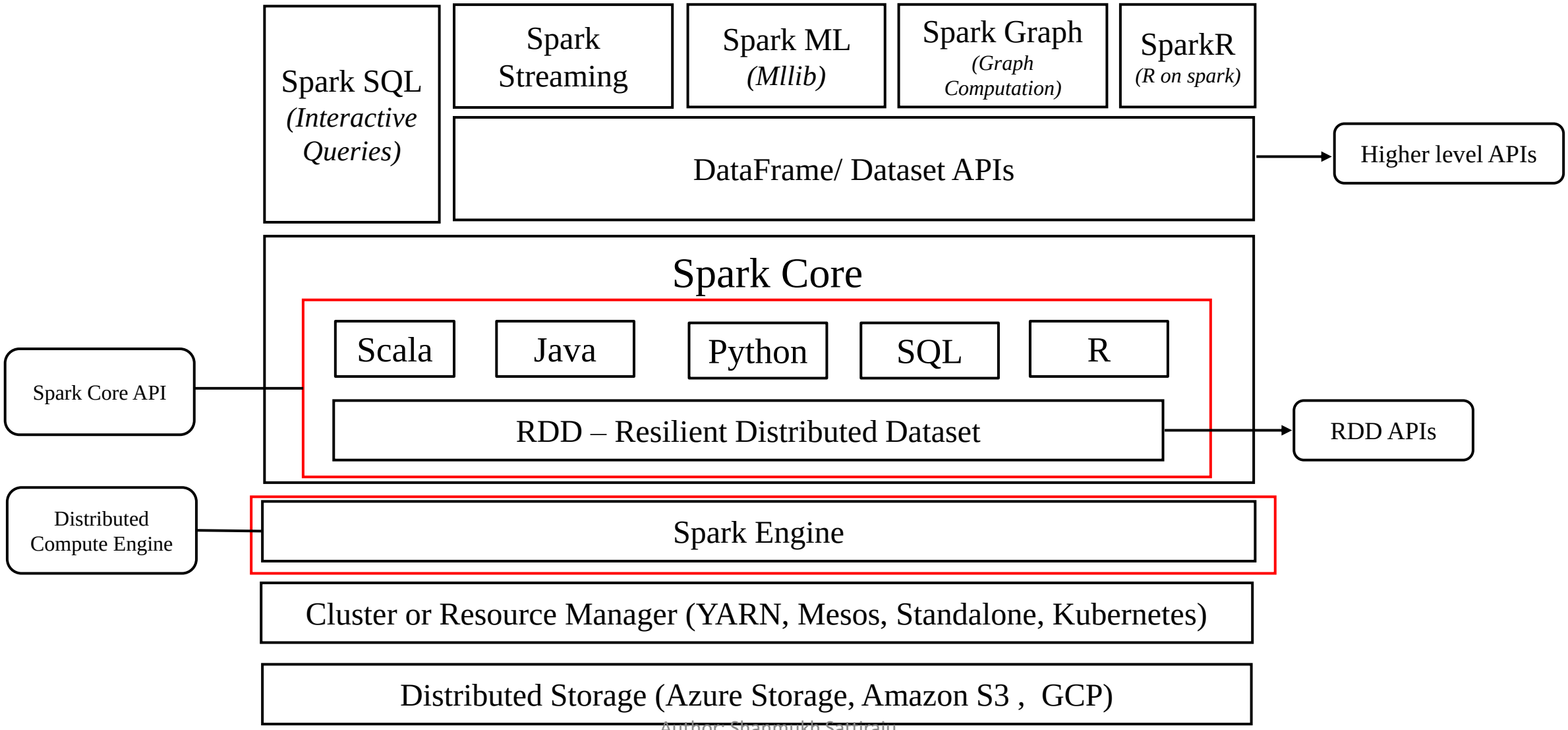
Apache Spark is an **open** source **in-memory** application framework for **distributed data processing** and iterative analysis on massive data volumes

In simple terms, Spark is a

- Compute Engine
- Unified data processing System

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Apache Spark Ecosystem



Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

What is Databricks?

- Unified Interface
- Open analytics platform
- Compute Management
- Notebooks
- Integrates with Cloud Storages
- MLFlow modeling
- Git
- SQL Warehouses

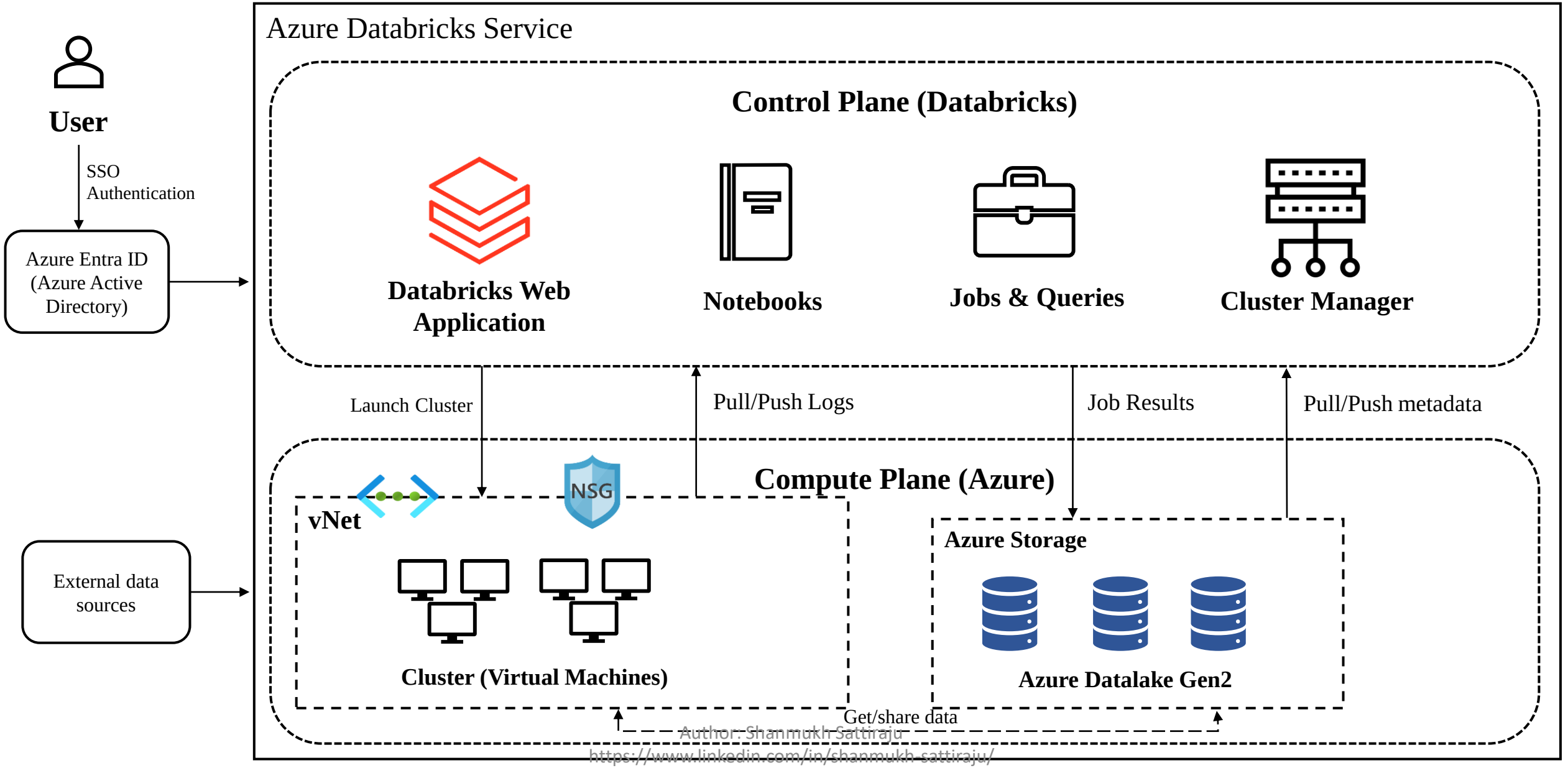
Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

How Databricks Work with Azure?

- Unified billing
- Integration with Data services
- Azure Entra ID (previously Azure Active Directory
- Azure Data Factory
- Power BI
- Azure DevOps

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Azure Databricks Architecture



Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

How Databricks Work with Azure?

- Unified billing
- Integration with Data services
- Azure Entra ID (previously Azure Active Directory)
- Azure Data Factory
- Power BI
- Azure DevOps

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Azure Databricks Compute

- Cluster is a set of computation resources and configurations to run your workloads
- Workloads can be:
  1. Set of commands in a notebook
  2. A job that you run as a automated workflow
- Cluster types:
  1. **All purpose Cluster**
     - To execute set of commands in a notebook
  2. **Job Cluster**
     - To execute a job that you run as a automated workflow

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Cluster Types

1. **All purpose Cluster**
   - To interactively run the commands in your notebook
   - Multiple users can share such clusters to do collaborative interactive analysis.
   - You can terminate, restart, attach , detach these clusters to multiple notebooks
   - You can choose
     - Multi-node cluster = Driver node and executor nodes will be on separate machine
     - Single node cluster = Only there will be a single driver node with single machine
2. **Job Cluster**
   - To run a job that you run as a automated workflow
   - It runs a new job cluster and terminates the cluster automatically when the job is complete.
   - You cannot restart a job cluster.

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

To create a new Cluster

- The policy
- The access mode, which controls the security features used when interacting with data
- The runtime version
- The cluster worker and driver node types

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Cluster Access modes

Access Mode	Visible to user	UC Support	Supported Languages	Notes
Single user	Always	Yes	Python, SQL, Scala, R	Can be assigned to and used by a single user.
Shared	Always (**Premium plan or above required**)	Yes	1. Python (on Databricks Runtime 11.1 and above), 2. SQL, 3. Scala (on Unity Catalog-enabled clusters using Databricks Runtime 13.3 and above)	Can be used by multiple users with data isolation among users.
No Isolation Shared	Yes, Admins can hide this cluster type by enforcing user isolation in the admin settings page.	No	Python, SQL, Scala, R	There is a related account-level setting for No Isolation Shared clusters.

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Cluster Runtime version:

- Databricks Runtime is the set of core components that run on your clusters

So which version to use?

- **For all purpose compute:**
  - Databricks recommends using the latest Databricks Runtime version.
  - Using the most current version will ensure you have the latest optimizations and most up-to-date compatibility between your code and preloaded packages.
- **For Job compute:**
  - As these will be operational workloads, consider using the Long Term Support (LTS) Databricks Runtime version.
  - Using the LTS version will ensure you don’t run into compatibility issues and can thoroughly test your workload before upgrading.
- **For ML Workloads:**
  - For advanced machine learning use cases, consider the specialized ML Runtime version.

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Cluster policies (in Unity Catalog)

- Policies are a set of rules configured by admins
- These are used to limit the configuration options available to users when they create a cluster
- Policies have access control lists that regulate which users and groups have access to the policies.
- Any user with unrestricted policy can create any type of cluster

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Cluster pools (in Unity Catalog)

- Refer documentation
- Also refer videos from Ramesh and Scholarnest
- https://www.databricks.com/blog/2019/11/11/databricks-pools-speed-up-data-pipelines.html

Magic commands

- You can use multiple languages in one notebook
- You need to specify language magic command at the beginning of a cell.
- By default, the entire notebook will work on the language that you choose at the top

Magic command	Language	Description
%python	Python	Execute a Python query against Spark Context.
%scala	Scala	Execute a Scala query against Spark Context.
%sql	Spark SQL	Execute a SparkSQL query against Spark Context.
%r	R	Execute a R query against Spark Context.

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

DBUtils

- Azure Databricks provides set of utilities to efficiently interact with your notebooks
- Most commonly used DBUtils are:
  - File System Utilities
  - Widget Utilities
  - Notebook Utilities

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

File System Utilities

dbutils.fs provides utilities for working with FileSystems

Below are the available utilities

cp : Copies a file or directory, possibly across FileSystems
head : Returns up to the first 20 records
ls : Lists the contents of a directory
mkdirs : Creates the given directory if it does not exist, also creating any necessary parent directories
mv :Moves a file or directory, possibly across FileSystems
put : Writes the given String out to a file
rm : Removes a file or directory

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Widgets Utilities

Dbutils.Widgets Utilities helps to gets the input value using parameters**.**
Widget types are:

- **combobox** : Creates a combobox input widget with a given name, default value and choices
- **dropdown** : Creates a dropdown input widget a with given name, default value and choices
- **get** : Retrieves current value of an input widget
- **multiselect** : Creates a multiselect input widget with a given name, default value and choices
- **remove** : Removes an input widget from the notebook
- **removeAll** : Removes all widgets in the notebook
- **text** : Creates a text input widget with a given name and default value

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Notebook Utilities

- **Exit :** This method lets you exit a notebook with a value
- **Run :** This method runs a notebook and returns its exit value

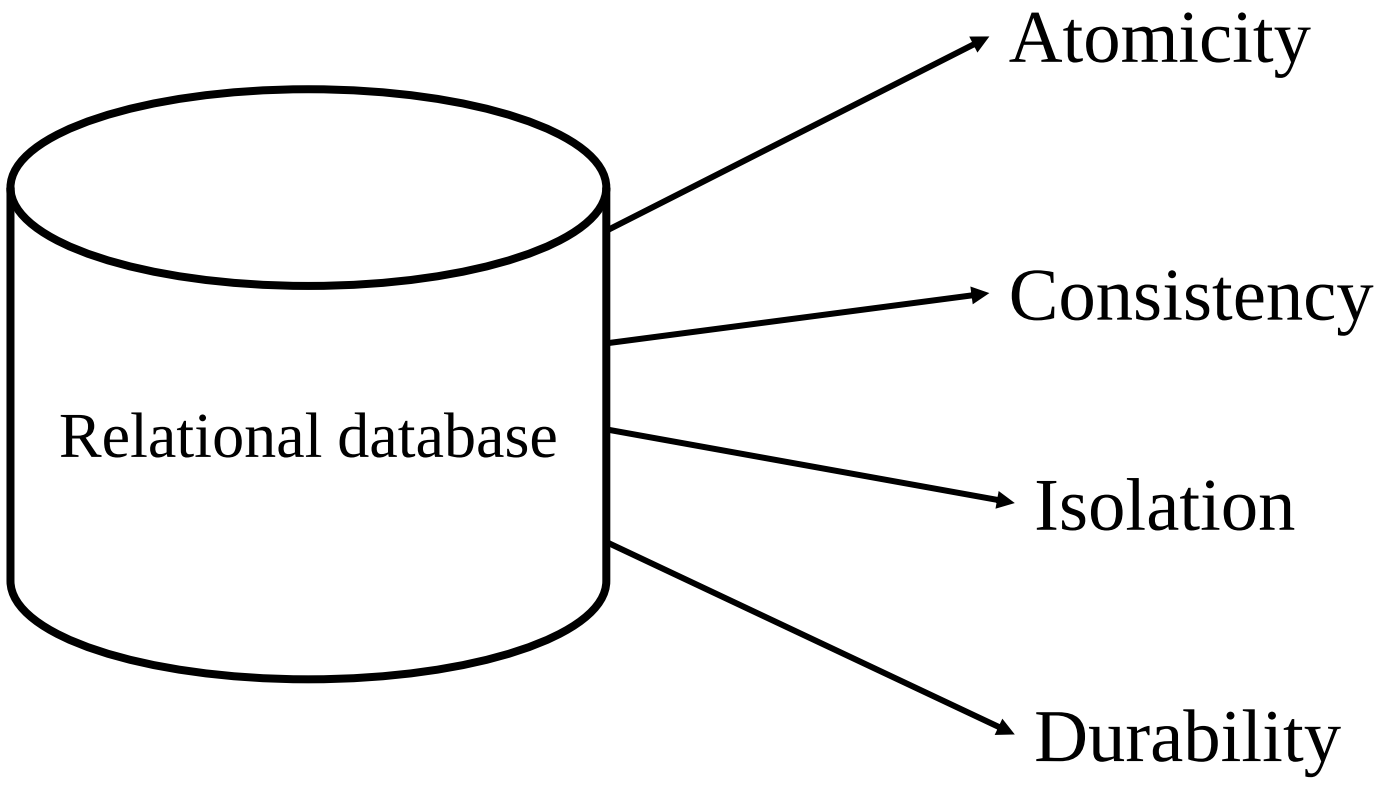
Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Delta Lake

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Drawbacks of ADLS

ADLS != Database



Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Drawbacks of ADLS

- No ACID properties
- Job failures lead to inconsistent data
- Simultaneous writes on same folder brings incorrect results
- No schema enforcement
- No support for updates
- No support for versioning
- Data quality issues

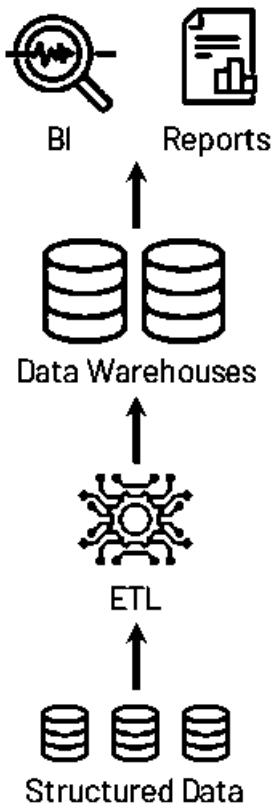
Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

What is delta lake

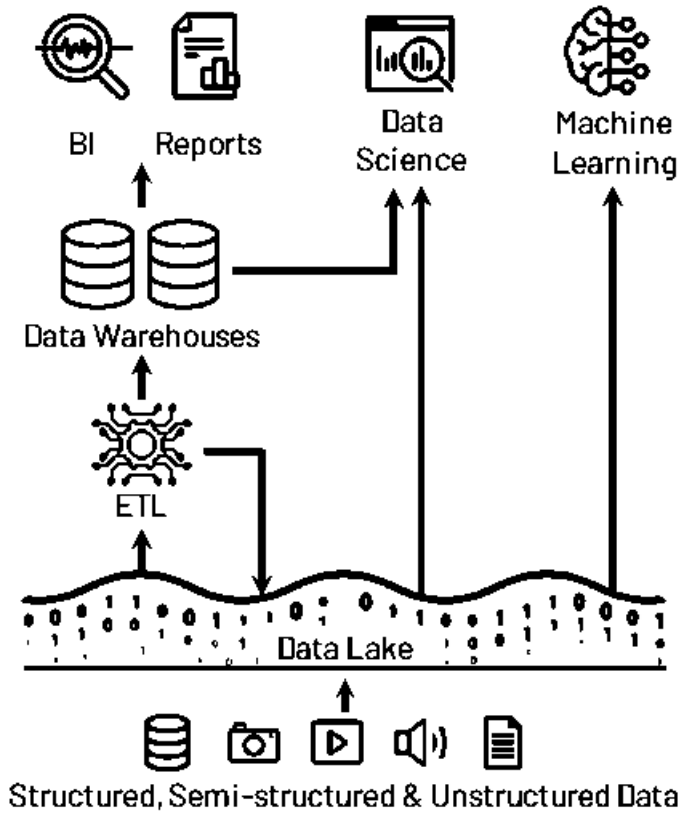
- Open-source storage framework that brings reliability to data lakes
- Brings transaction capabilities to data lakes
- Runs on top of your existing datalake and supports parquet
- Enables Lakehouse architecture

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

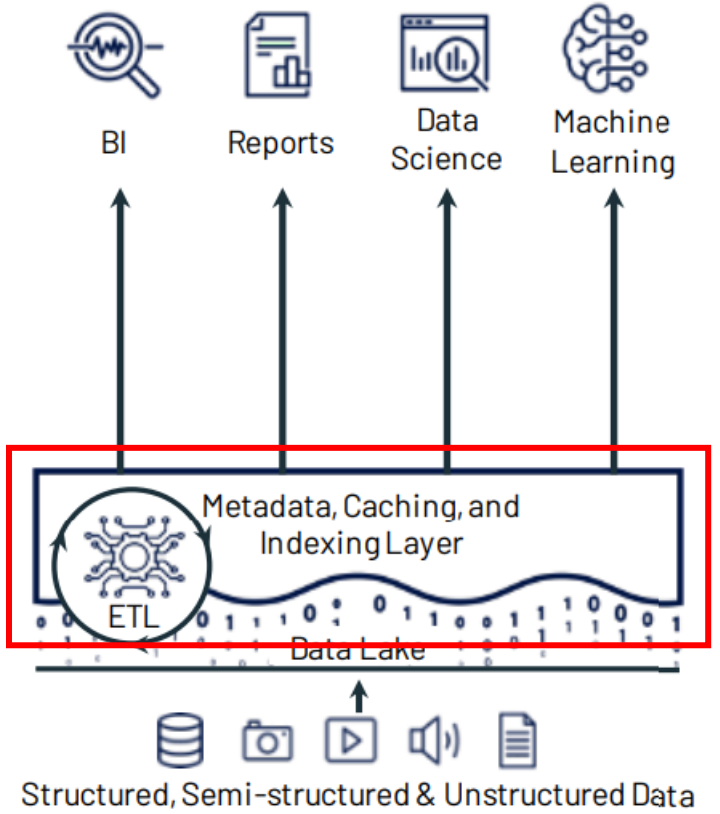
Lakehouse Architecture



Datawarehouse

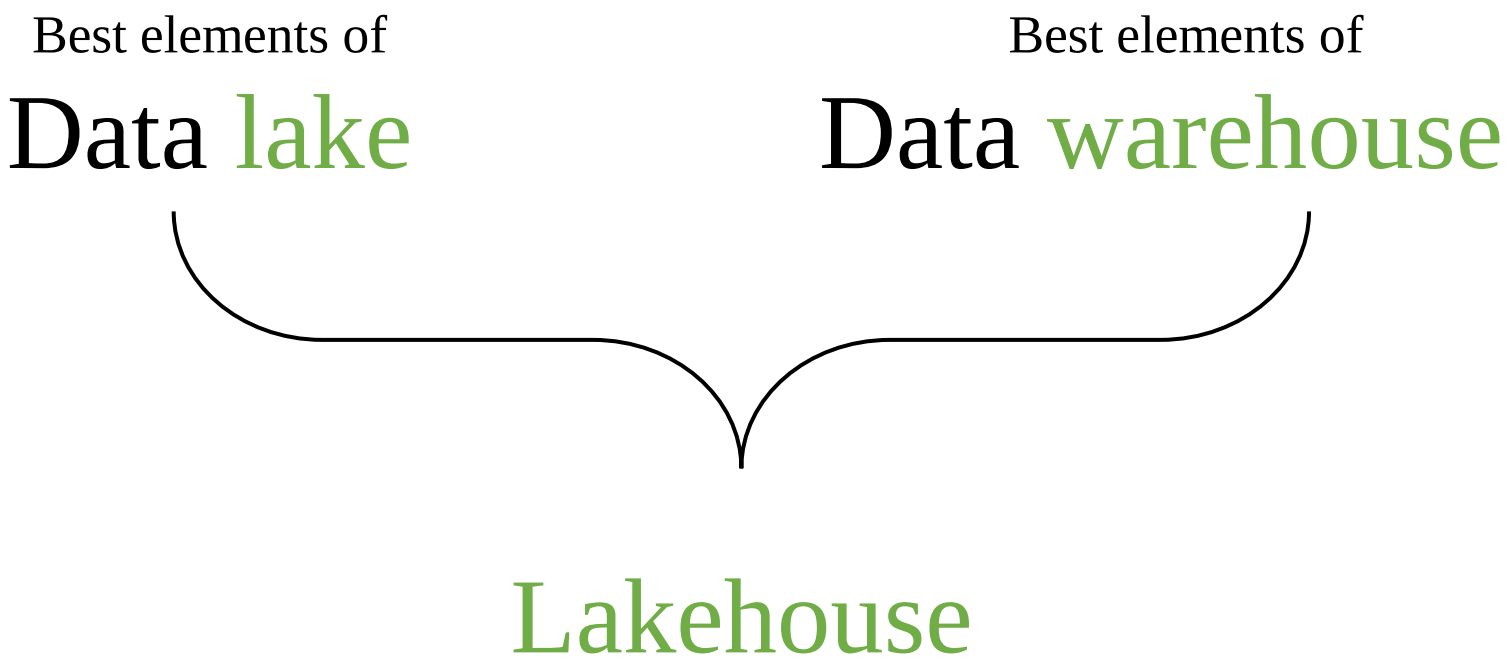


Modern Datawarehouse (uses Datalake)



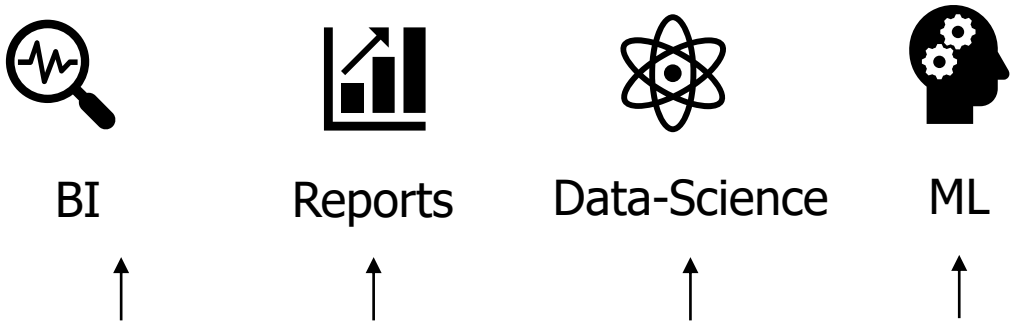
Lakehouse Architecture

Lakehouse Architecture



Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Lakehouse Architecture



Structured, Semi- Structured & Unstructured Data

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

How to create delta lake?

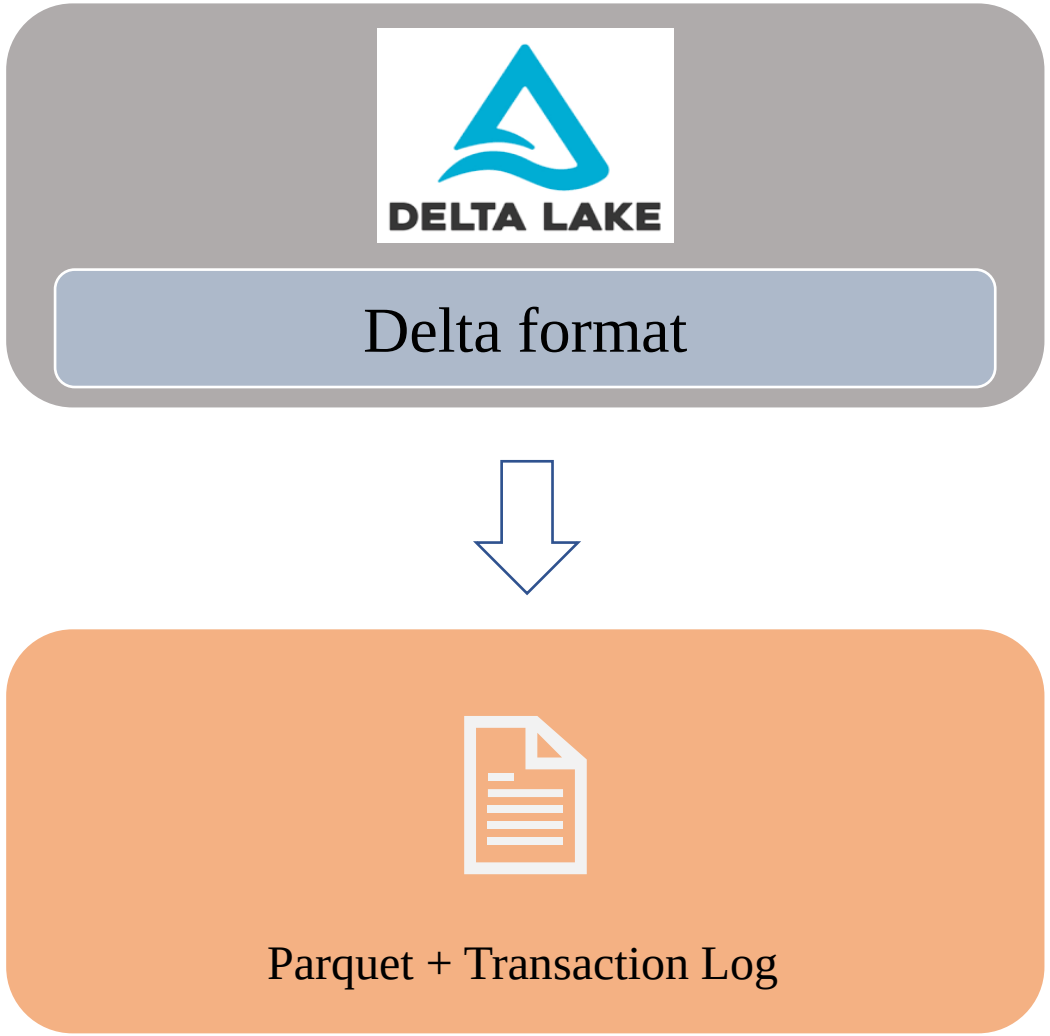
Instead of parquet..

```
dataframe.
        write\
        .format(“parquet”)\
        .save(“/data/”)
```

Replace with delta..

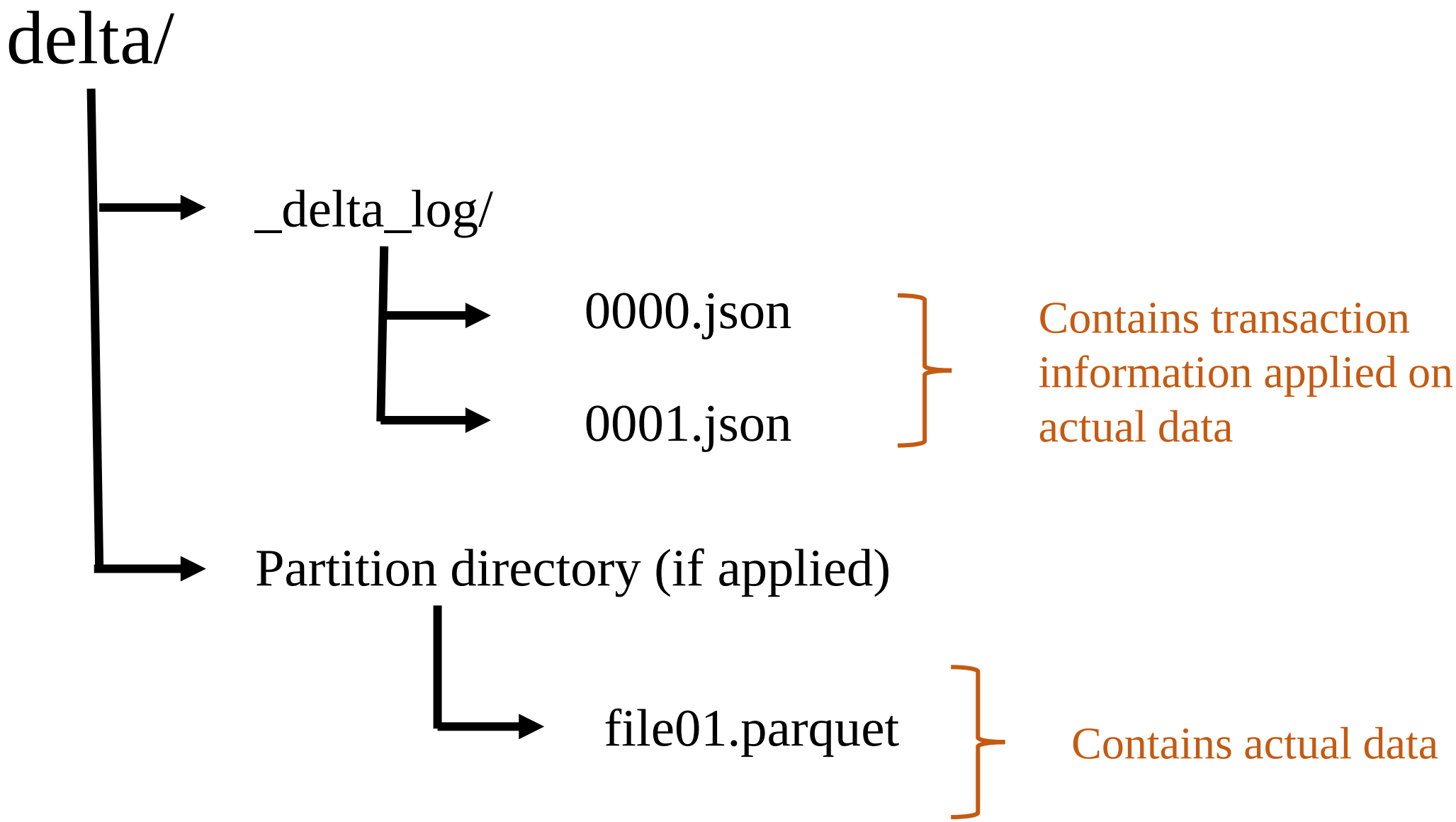
```
dataframe.
        write\
        .format(“delta”)\
        .save(“/data/”)
```

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/



Azure Data Lake Storage

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/



Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Understanding Transaction log file (Delta Log)

- Contains records of every transaction performed on the delta table
- Files under _delta_log will be stored in JSON format
- Single source of truth

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Transaction log contents

JSON File = result of set of actions

- **metadata** – Table's name, schema, partitioning ,etc
- **Add –** info of added file (with optional statistics)
- **Remove –** info of removed file
- **Set Transaction –** contains record of transaction id
- **Change protocol –** Contains the version that is used
- **Commit info –** Contains what operation was performed on this

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Schema Enforcement

Loading new data

Col1	Col2	Col3	Col4	Col5

Delta Table

Col1	Col2	Col3	Col4

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

How does schema enforcement works?

Delta lake uses Schema validation on **“writes” .**

Schema Enforcement Rules:

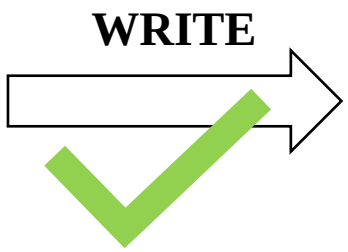
1. Cannot contain any additional columns that are not present in the target table's schema
2. Cannot have column data types that differ from the column data types in the target table.

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Schema Evolution

Loading new data

Col1	Col2	Col3	Col4	Col5



Delta Table

Col1	Col2	Col3	Col4

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Audit Data Changes & Time Travel

- Delta automatically versions every operation that you perform
- You can time travel to historical versions
- This versioning makes it easy to audit data changes, roll back data in case of accidental bad writes or deletes, and reproduce experiments and reports.

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Vacuum in Delta lake

- Vacuum helps to remove parquet files which are not in latest state in transaction log
- It will skip the files that are starting with _ (underscore) that includes _delta_log
- It deletes the files that are older then retention threshold
- Default retention threshold in 7 days
- If you run VACUUM on a Delta table, you lose the ability to time travel back to a version older than the specified data retention period.

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Optimize in Delta lake

Operation	parquet files	_delta_log	Line number Column	State
CREATE TABLE		000.json		
WRITE	aabb.parquet	001.json	100	**Active**
WRITE	ccdd.parquet	002.json	101	Inactive
WRITE	eeff.parquet	003.json	102	Inactive
DELETE 101		004.json		
UPDATE 102	iijj.parquet	005.json	99	**Active**

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

UPSERT (Merge) in delta lake

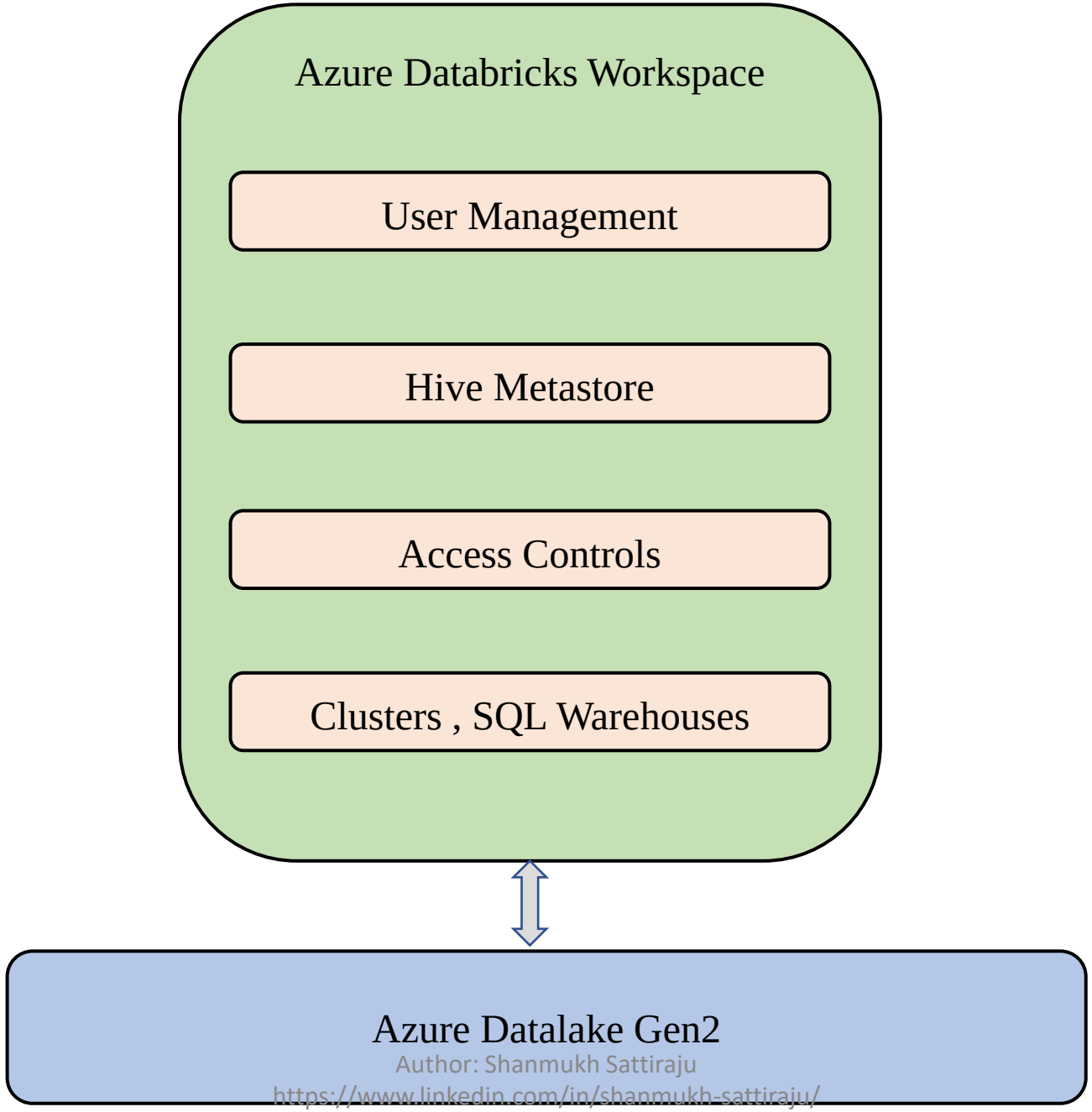
- We can UPSERT (UPDATE + INSERT) data using MERGE command.
- If any matching rows found, it will update them
- If no matching rows found, this will insert that as new row

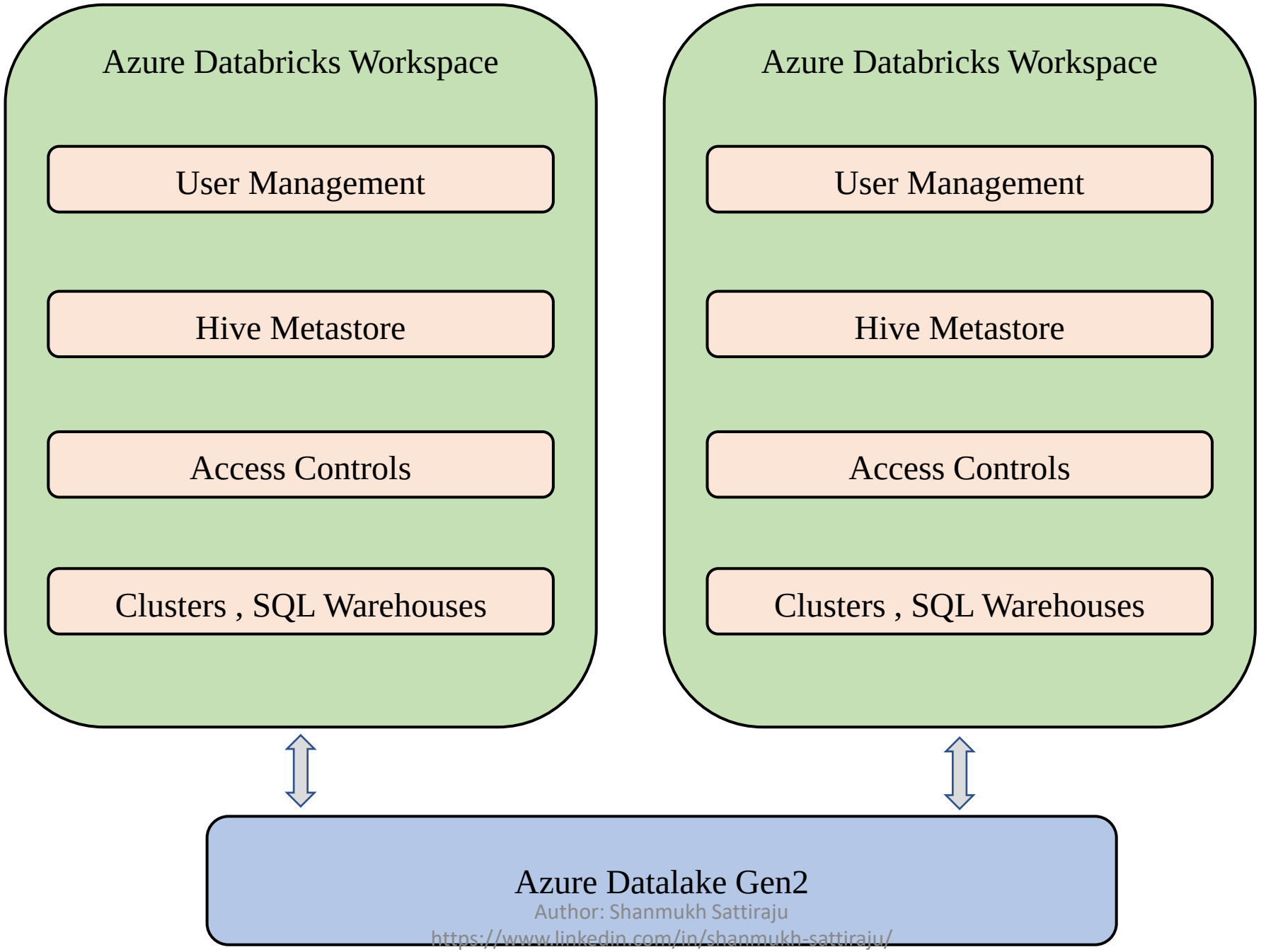
```
MERGE INTO <Destination_Table>
USING <Source_Table>
        ON <Dest>.Col2 = <Source>.Col2
WHEN MATCHED
THEN UPDATE SET
        <Dest>.Col1 = <Source>.Col1,
        <Dest>.Col2 = <Source>.Col2
WHEN NOT MATCHED
THEN INSERT
        VALUES(Source.Col1, Source.Col2)
```

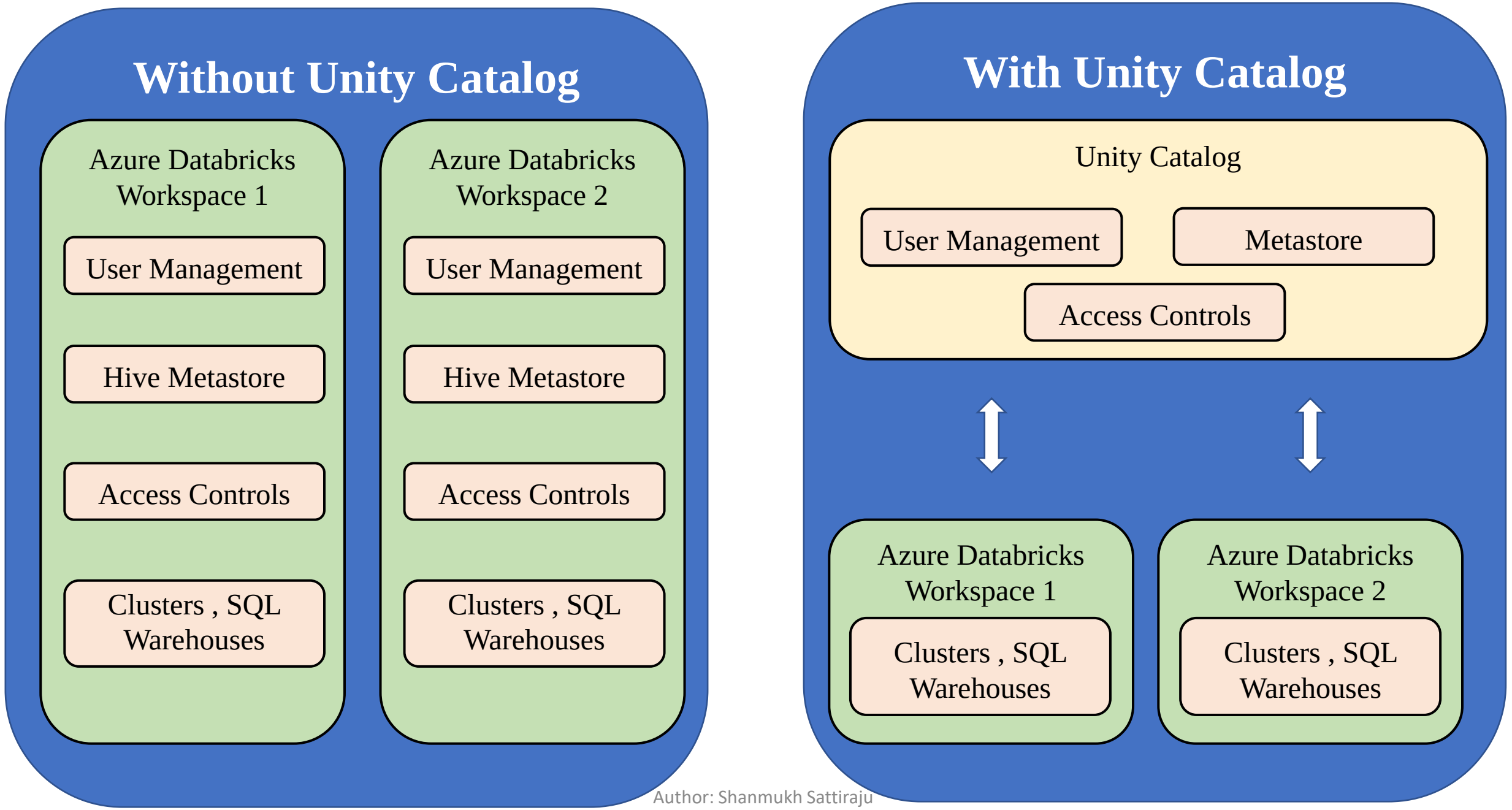
Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Unity Catalog

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

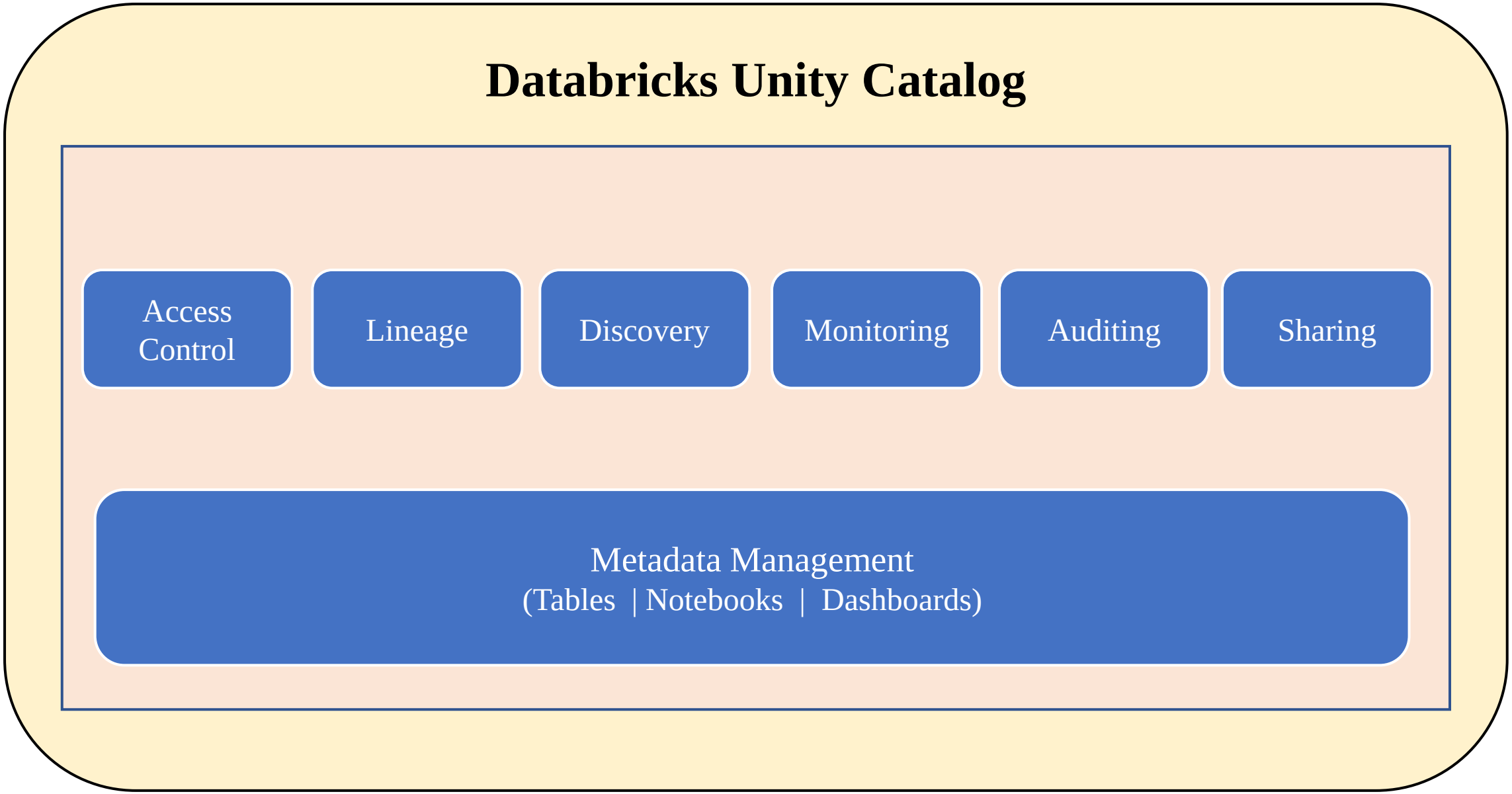

Azure Databricks Workspace
User Management
Hive Metastore
Access Controls
Clusters , SQL Warehouses
Azure Datalake Gen2
Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/


Azure Databricks Workspace
User Management
Hive Metastore
Access Controls
Clusters , SQL Warehouses
Azure Databricks Workspace
User Management
Hive Metastore
Access Controls
Clusters , SQL Warehouses
Azure Datalake Gen2
Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

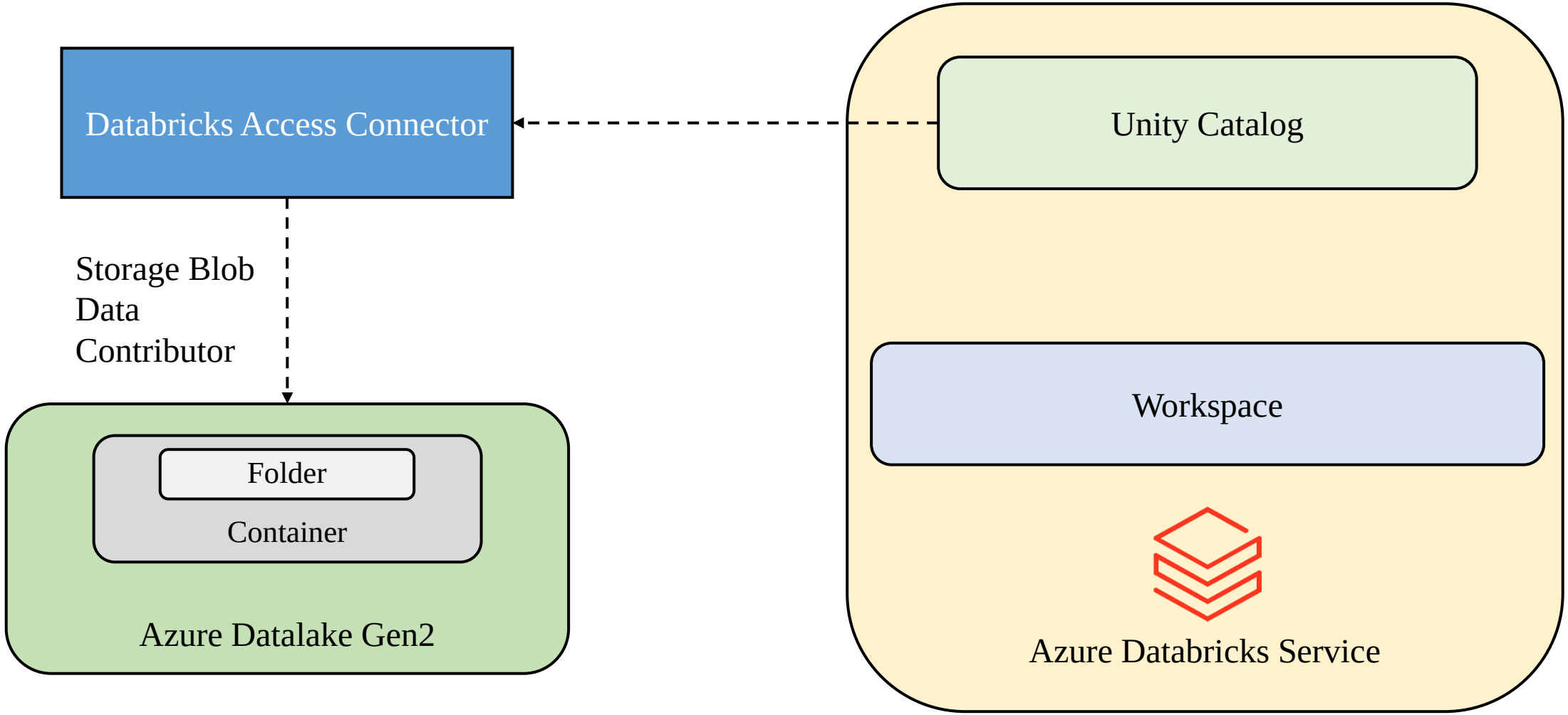


Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Databricks Unity Catalog



Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/



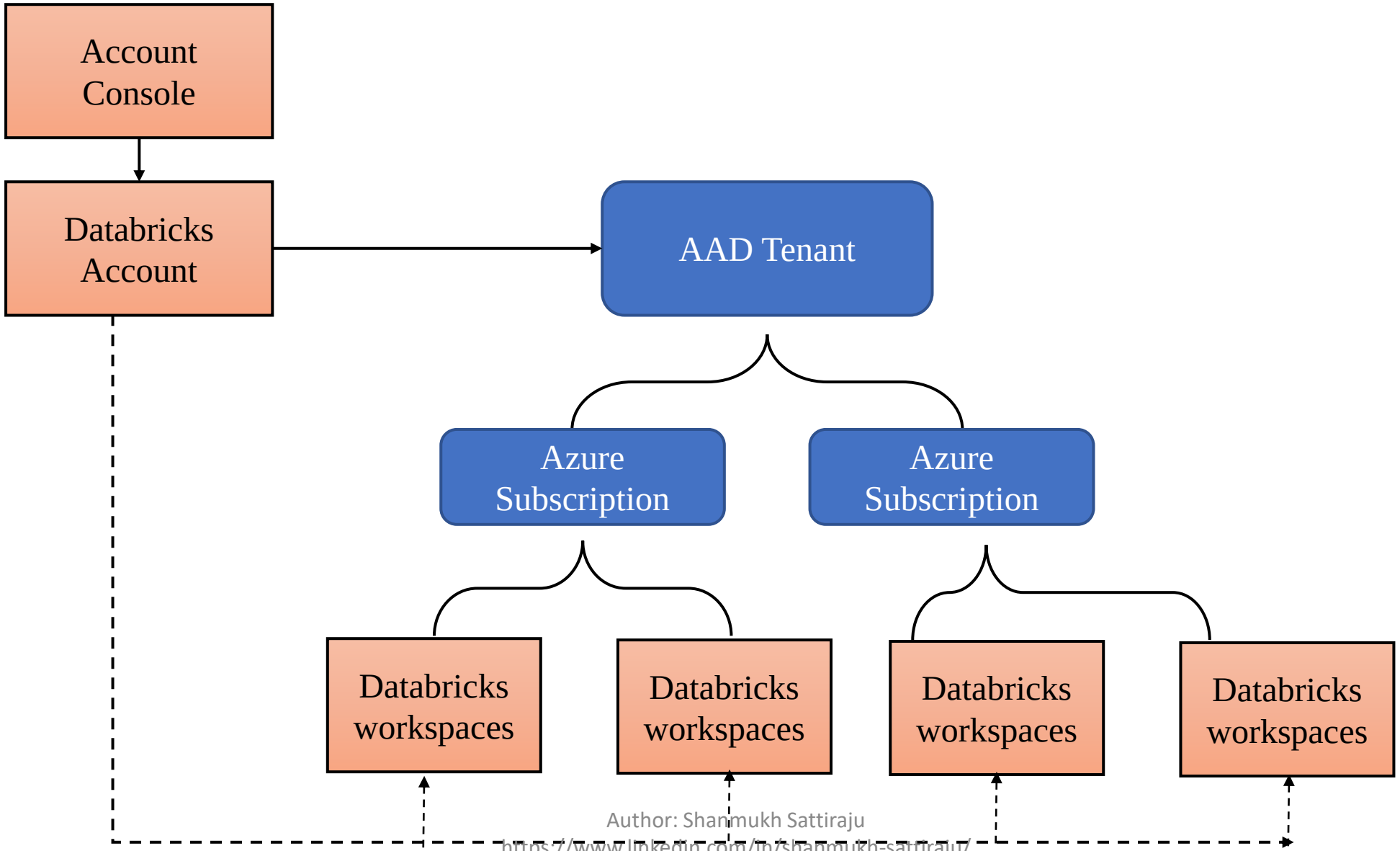
Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

To use Unity Catalog

- Databricks Premium Workspace
- Configure Metastore
- Attach workspace to Metastore

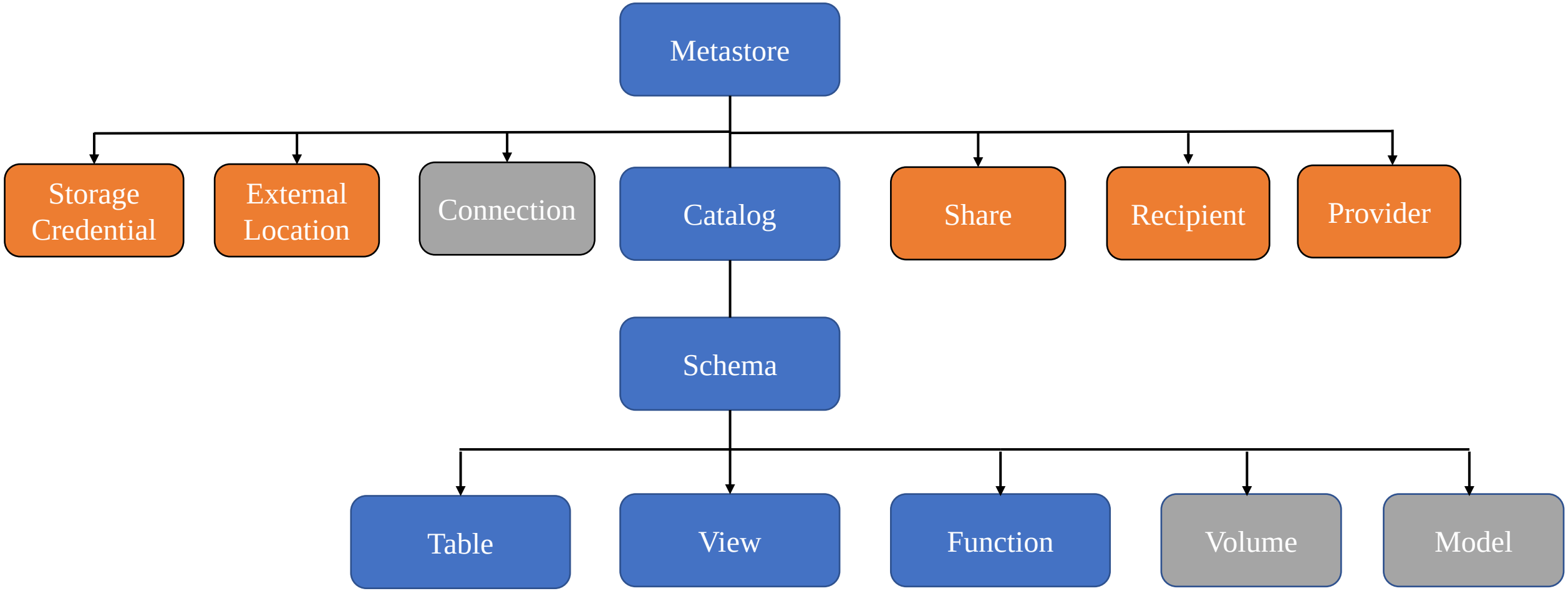
Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Unity Catalog and Azure



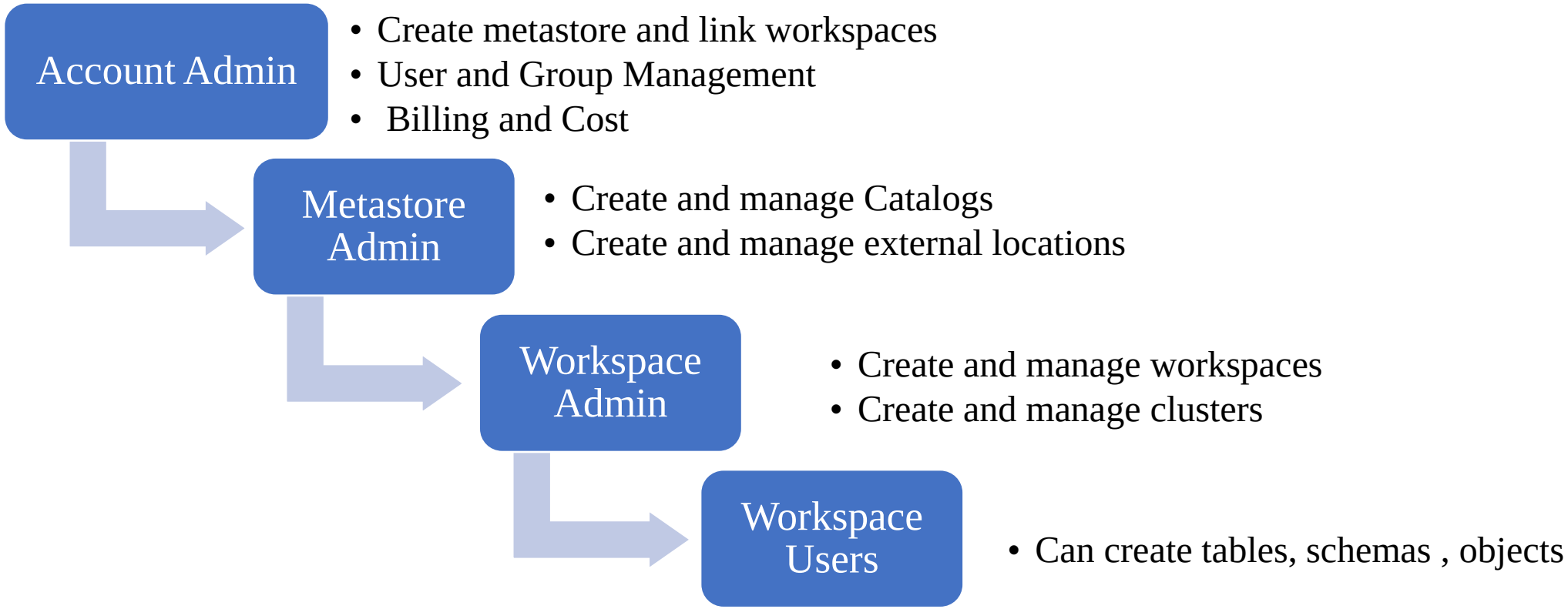
Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Unity Catalog Object Model



Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Roles in Unity Catalog



Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

User and Group Management

- Invite and add users to Unity Catalog
- Create groups
  - Workspace admins
  - Developers
- Assign groups to users
  - Workspace admins – Jarvis
  - Developers - Steve
- Assign roles to groups
  - Workspace Admin – Workspace Admins Group
  - Workspace User – Developers Group

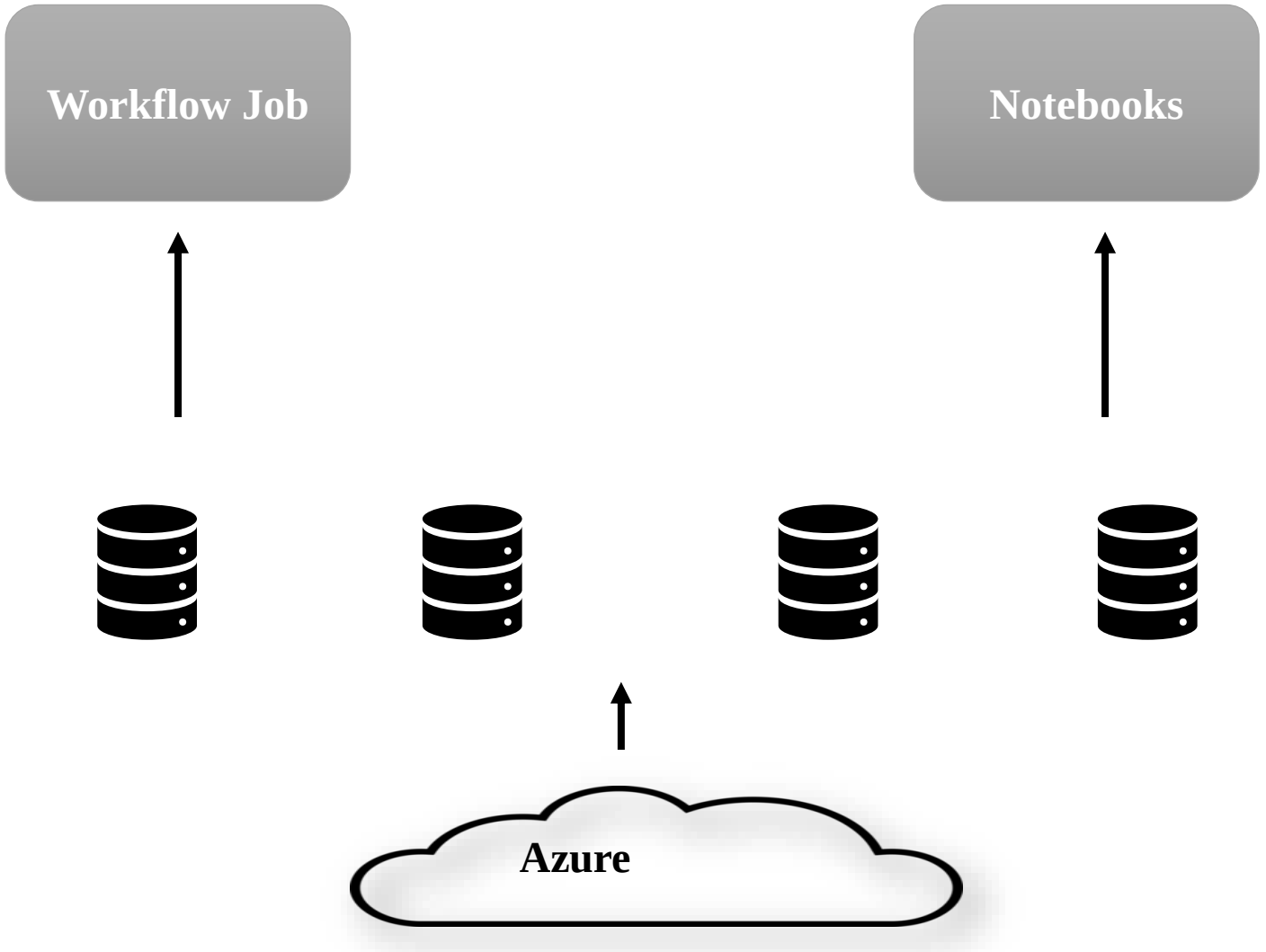
Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Cluster policy

- To control user’s ability to configure clusters based on a set of rules.
- These rules specify which attributes or attribute values can be used during cluster creation.
- Cluster policies have ACLs that limit their use to specific users and groups.
- A user who has unrestricted cluster create permission can select the Unrestricted policy and create fully-configurable clusters.

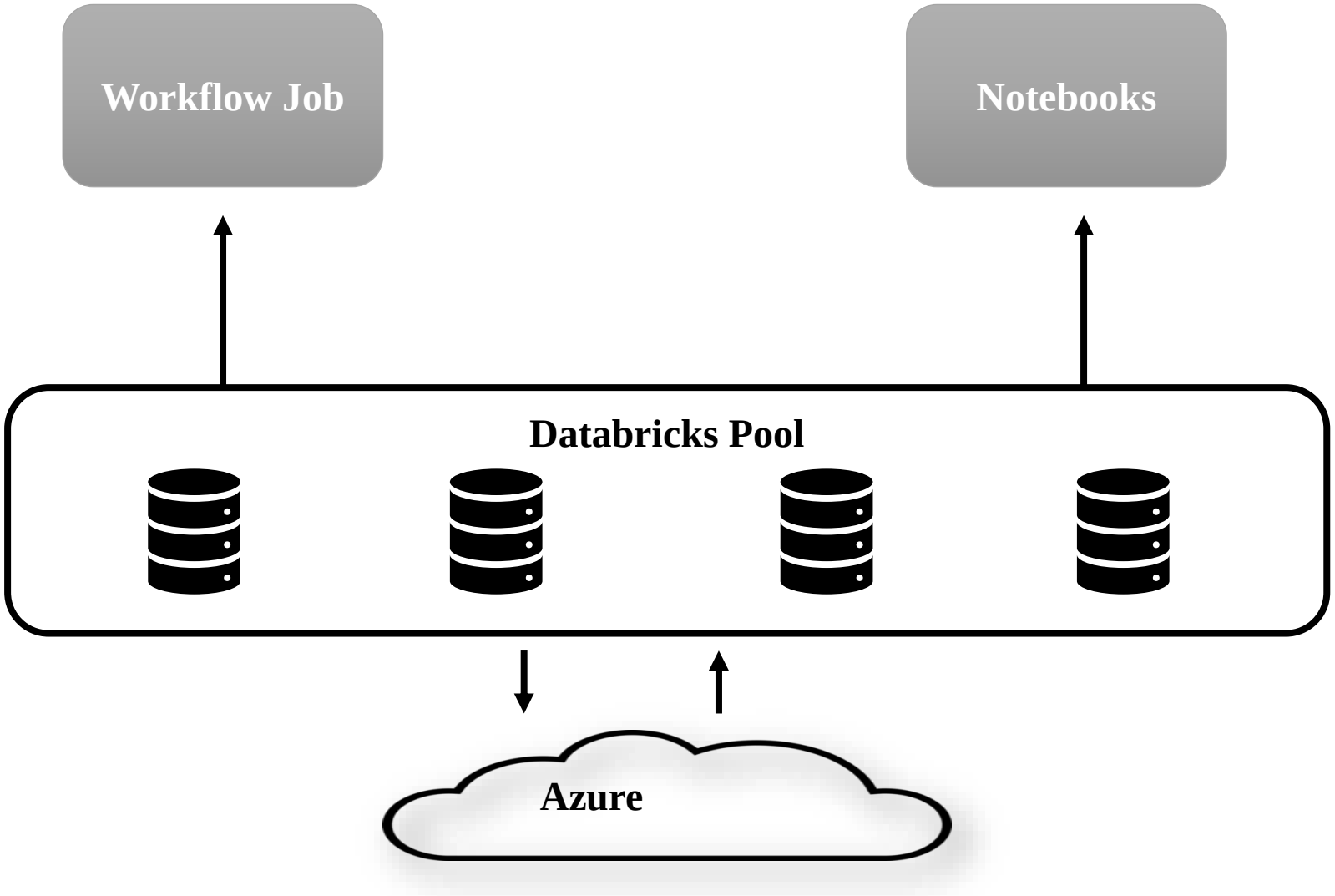
Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Without Cluster pools

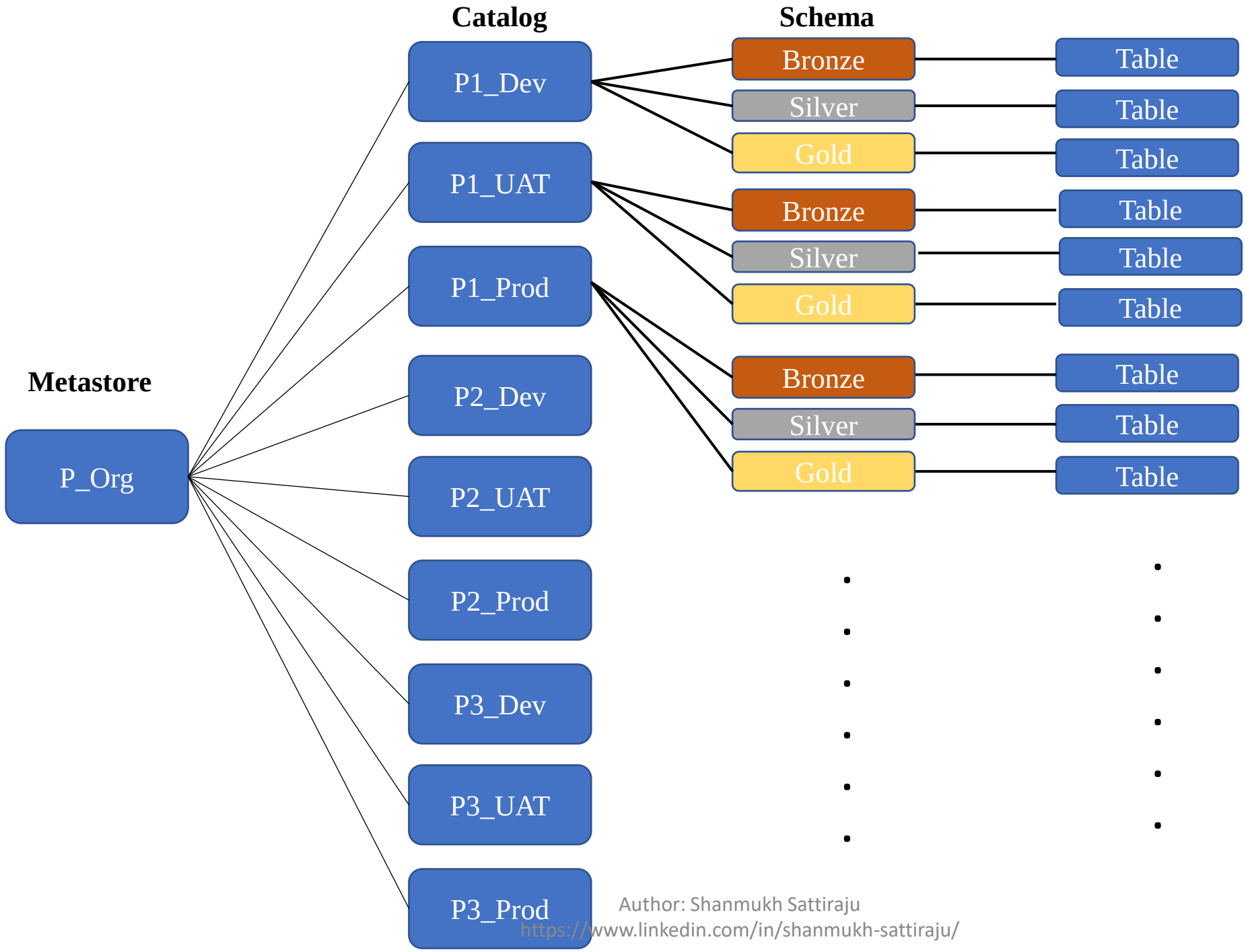


Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

With Cluster pools



Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/



Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Unity Catalog Privileges

- Privileges are permissions that we assign on objects to users
- Can use SQL command or Unity Catalog UI

Eg:

GRANT **privilege_type** ON **securable_object** TO **principal**

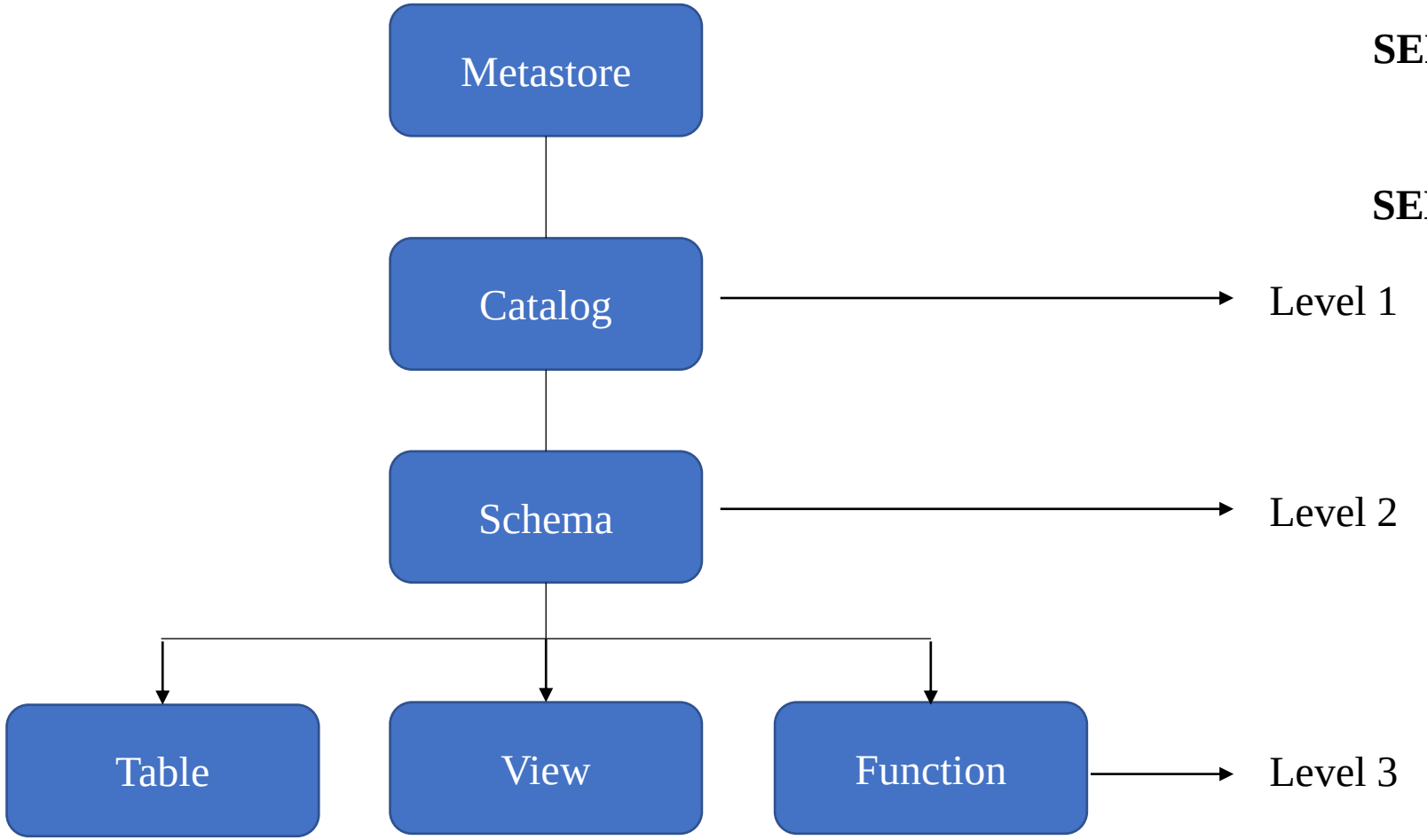
Privilege_Type : Unity Catalog permissions like SELECT, CREATE

Securable_object: Any object like SCHEMA, TABLE , etc

Principal: Can be a user, group, etc.

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

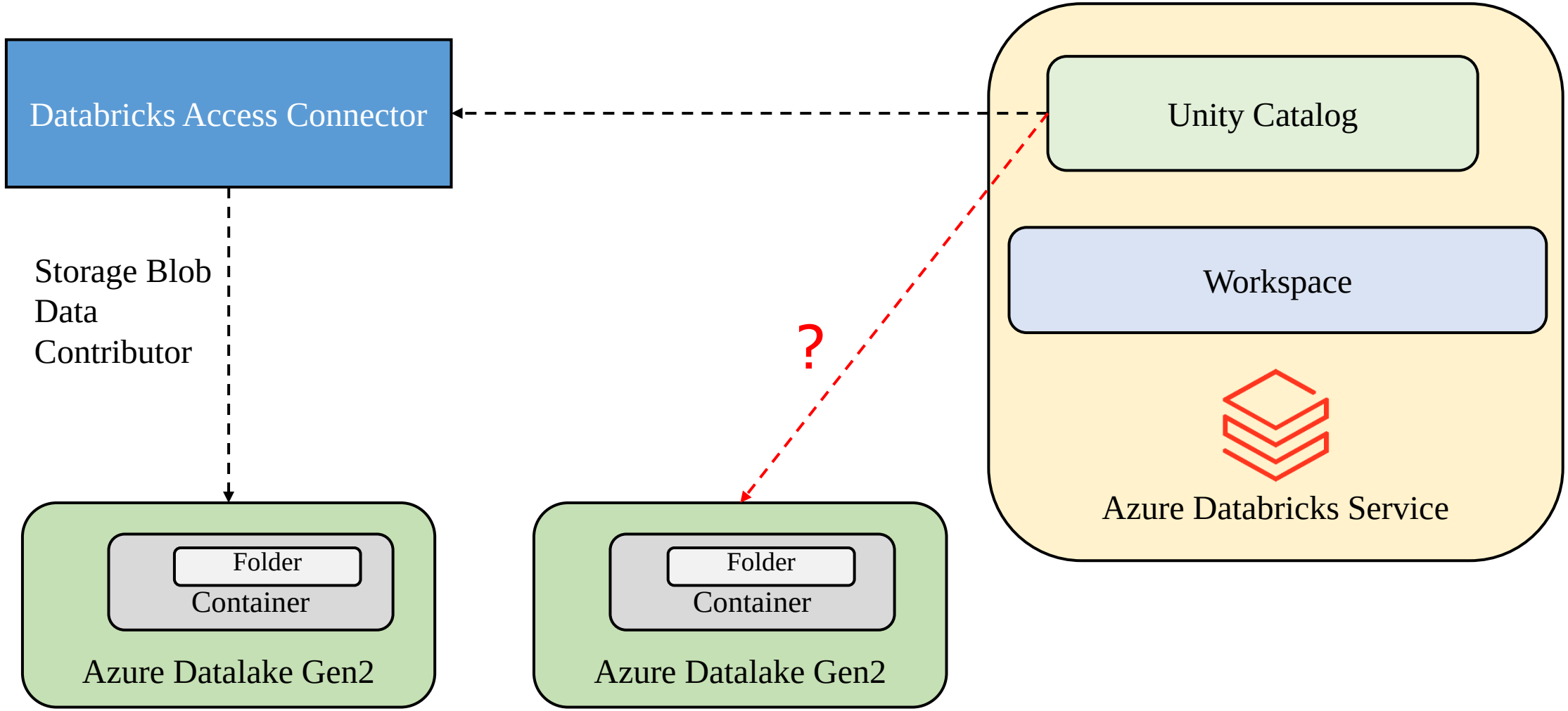
Unity Catalog - Three level Namespace



SELECT * FROM `Catalog`.`schema`.Table

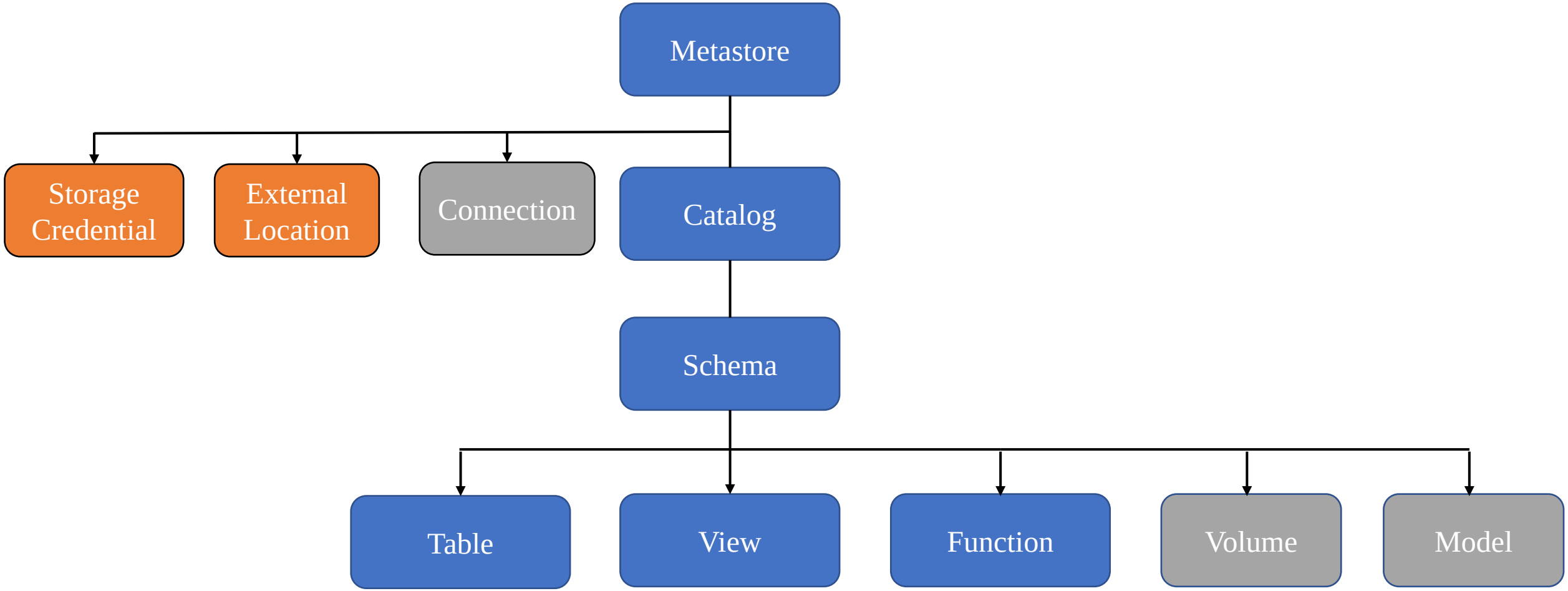
SELECT * FROM `s_dev`.`sales`.products

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

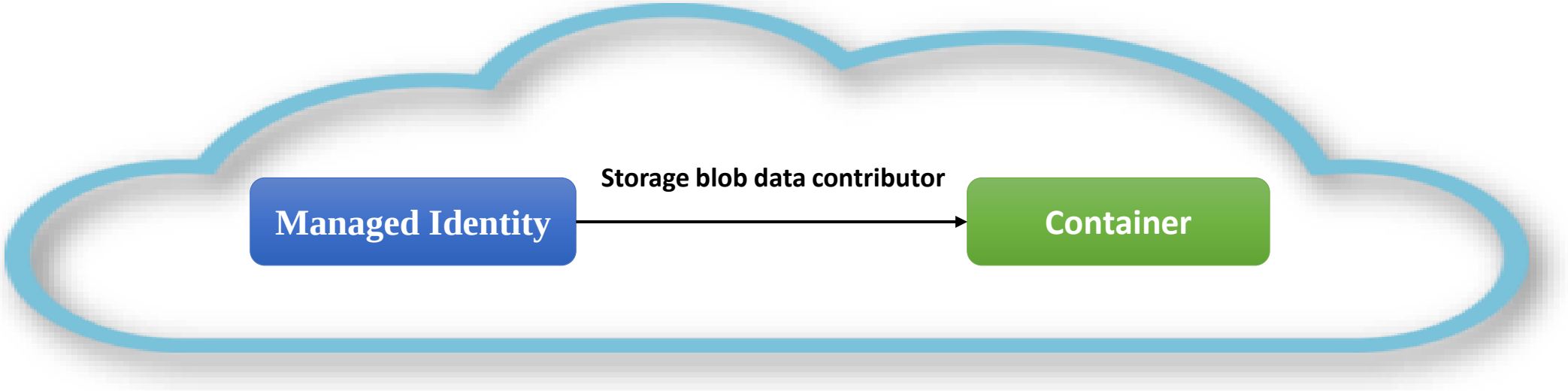


Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Unity Catalog Object Model



Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/



Storage Credential

Managed Identity

External Location

Path of container

Storage Credential

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Storage Credential	External Location
An authentication and authorization mechanism for accessing data stored	Serves as a reference point for External storage
Stores the access Credentials to provide access to External Location	Stores the path of the external storages that you want to access.
Credentials can be Managed Identities / Service principles	Makes use of Storage credential to get access to External Storage

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

- Managed Tables
  - These can be defined without a specified location
  - The data files are stored within managed storage in Delta format
  - Dropping the table not only removes its metadata from the catalog, but also deletes the actual data but in Unity Catalog the underlying data will be present for 30 days.

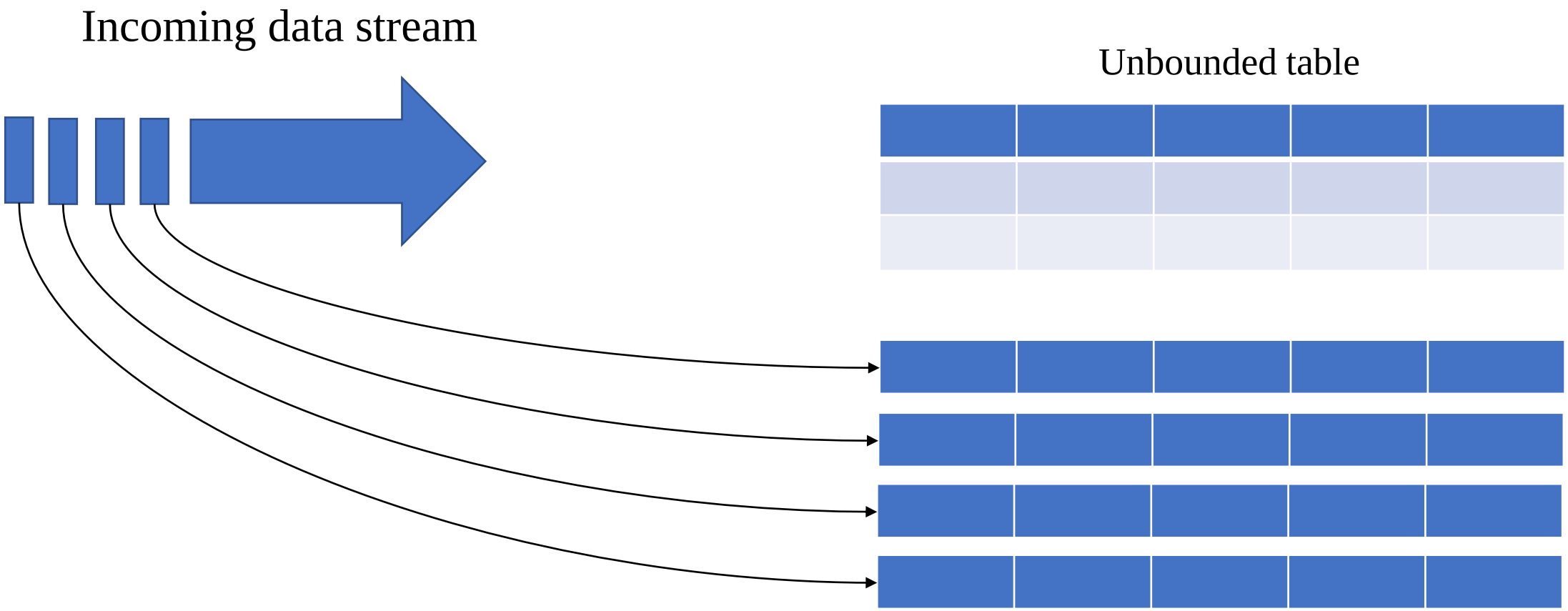
- External Tables
  - You need to have an EXTERNAL LOCATION and STORAGE CREDENTIALS created to access the external storage.
  - These can be defined for a custom file location, other than the managed storage
  - Dropping the table deletes the metadata from the catalog, but doesn't affect the data files.

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Spark Structured Streaming

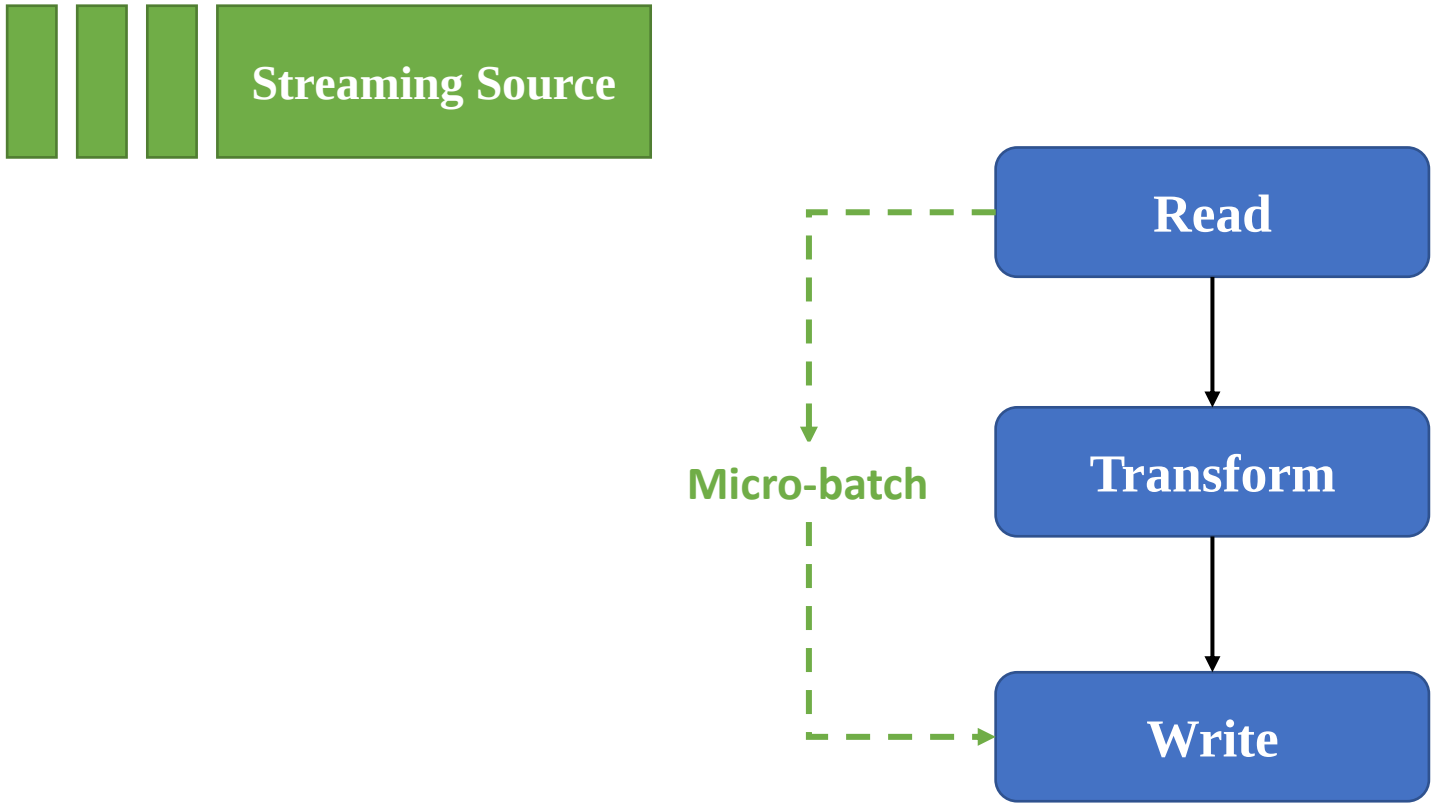
Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Spark Structured Streaming



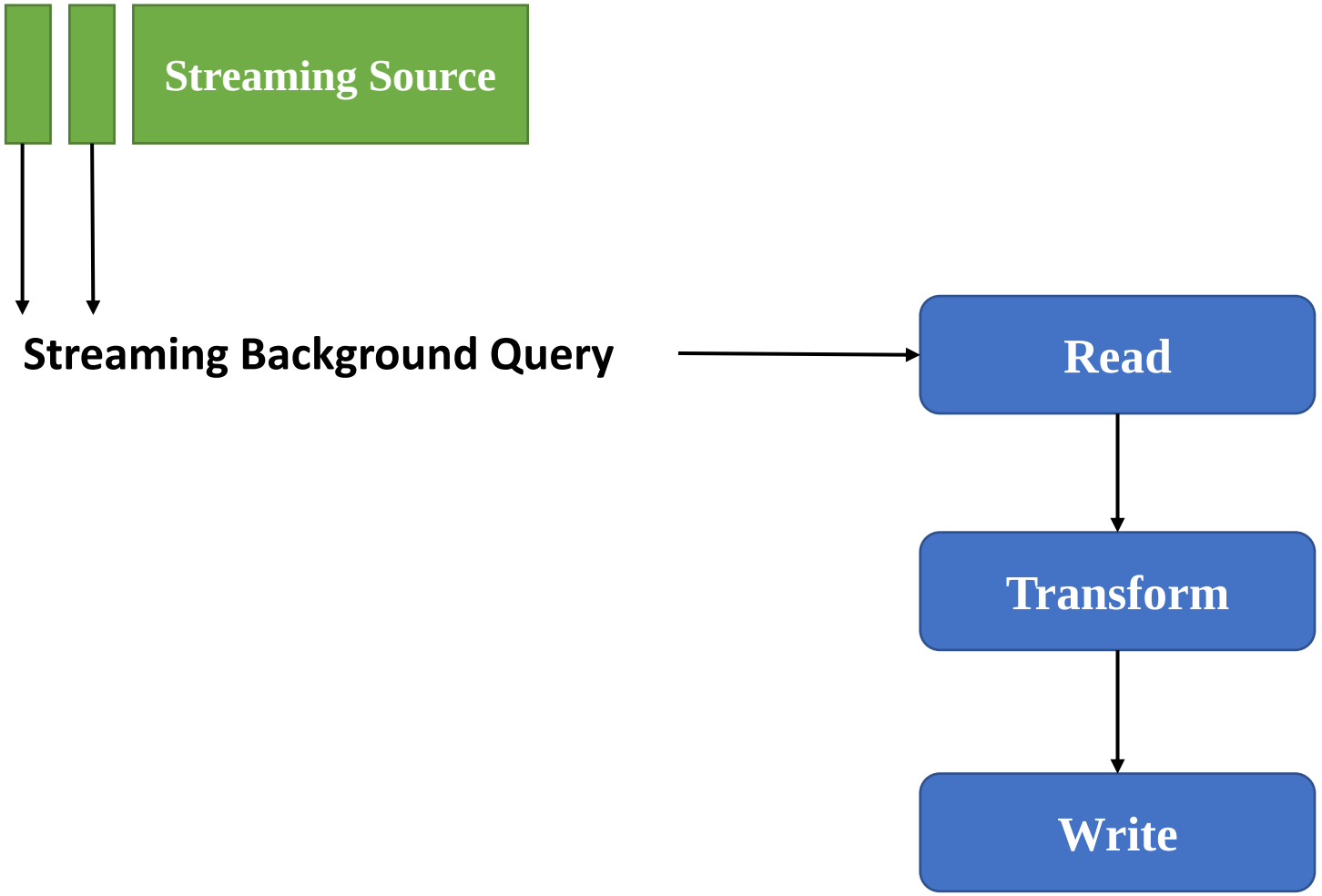
Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Spark Structured Streaming flow



Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Spark Structured Streaming flow



Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Supported Sources and Sinks

Sources

File Source

Kafka Source

Socket Source

Rate Source

Table

Sinks

File Sink

Kafka Sink

Foreach Sink

Console Sink

Table

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

StreamWriter

```
<StreamingDataframe>.writeStream
                        .option('checkpointLocation’,<Location>)
                        .outputMode('append’)
                        .toTable(‘<TableName>’)
```

Checkpoint

- To develop fault-tolerant and resilient Spark applications.
- It maintains intermediate state on fault-tolerant compatible file systems like HDFS, ADLS and S3 storage systems to recover from failures.
- Must be **unique** to each stream

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

outputModes

OutputMode	Usage	Description
Append	outputMode(‘append’)	The records from incoming streams will be appended to destination
Complete	outputMode(‘complete’)	All the processed rows will be displayed
Update	outputMode(‘update’)	Spark will output only updated rows. This is valid only if there are aggregation results; otherwise, this would be similar to Append mode.

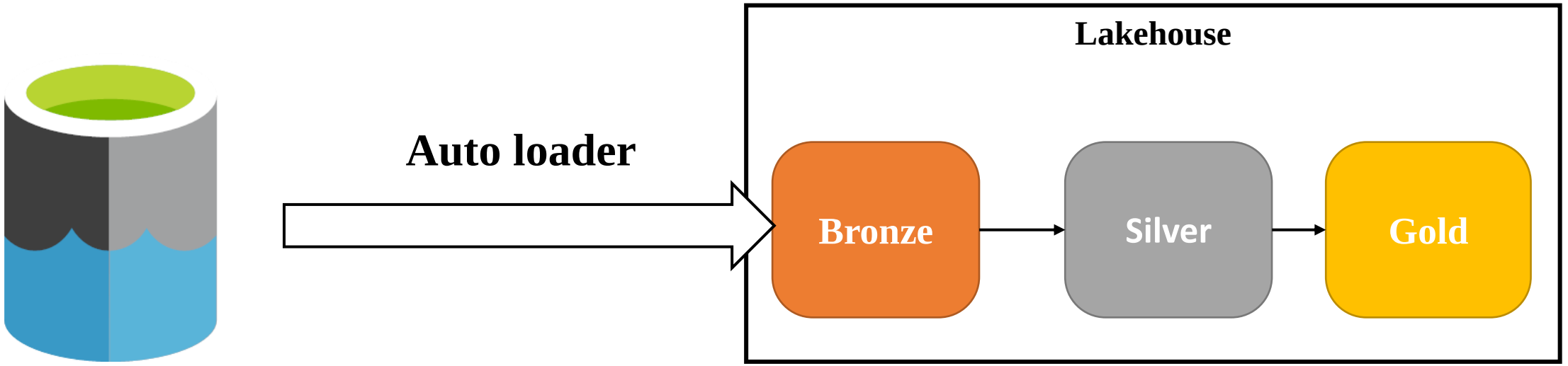
Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Triggers

Triggers	Usage	Description
Unspecified (default)		will trigger the microbatch for every 500 ms or half a second
processingTime (Fixed Interval)	. trigger(processingTime='2 minutes')	You can set processing time or time interval for each execution .
availableNow (OneTime)	.trigger(availableNow = True)	consumes all available records from previous execution as an incremental batch
Continuous (experimental)	.trigger(continuous = ‘1 second’)	For ~1ms latency

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Autoloader



Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Autoloader

- Autoloader is an **optimized data ingestion tool** that incrementally and efficiently processes new data files as they arrive in the cloud storage built into the Databricks Lakehouse.
- Auto Loader incrementally and efficiently processes new data files as they arrive in cloud storage without any additional setup.
- Auto Loader can load data files from Cloud Storages without being vendor specific (AWS S3 , Azure ADLS , Google Cloud Storage, DBFS).
- Auto Loader can ingest JSON, CSV, PARQUET, AVRO, ORC, TEXT, and BINARYFILE file formats
- This Auto loader is beneficial when you are ingesting data into your lakehouse particularly into bronze layer as a streaming query.

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Implementing Autoloader

```
df_str = (spark.readStream
          .format("cloudFiles")     ## This will tell the spark to use AutoLoader.
          .option("cloudFiles.format","csv")  ## Tells Autoloader to expect csv files
         .option('header','true')
         .schema(schema)
         .load(f'{source_dir}')
 )
```

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Schema evolution

- Schema evolution is the process of managing changes in data schema as it evolves over time, often due to updates in software or changing business requirements, which can cause schema drift

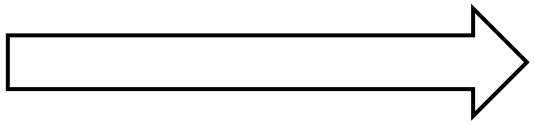
- Ways to handle schema changes
  - Fail the stream
  - Manually change the existing schema
  - Evolve automatically with change in schema

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Schema validation

Col1	Col2	Col3

Autoloader



/schemaLocation

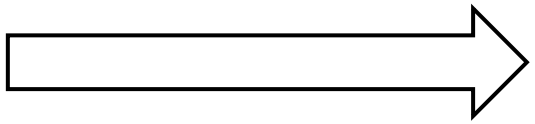
{
Col1 : int
Col2 : String
Col3: int
}

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

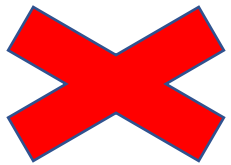
Schema validation

Col1	Col2	Col3	Col4

Autoloader



Validates schema



/schemaLocation

{
Col1 : int
Col2 : String
Col3: int
}

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Schema Evolution

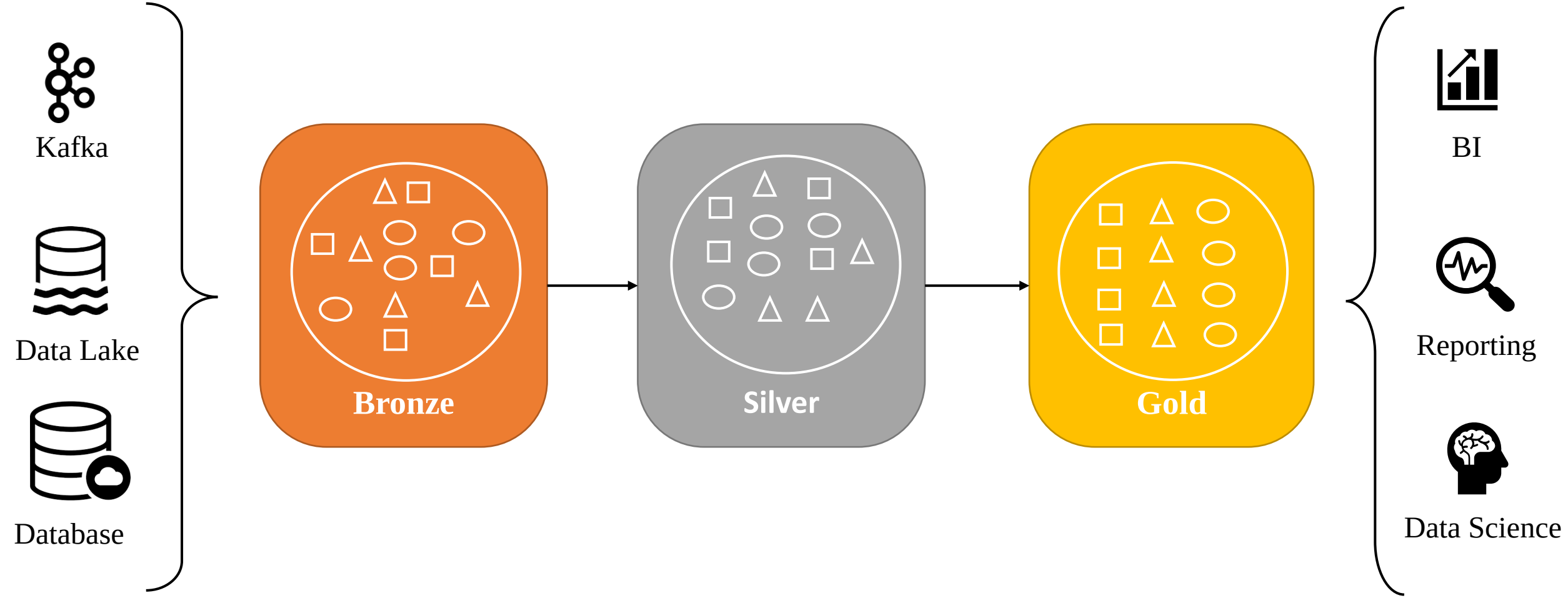
- **addNewColumns** = Stream fails. New columns are added to the schema. Existing columns do not evolve data types.
- **failOnNewColumns** = Stream fails. Stream does not restart unless the provided schema is updated, or the offending data file is removed
- **rescue** = Schema is never evolved and stream does not fail due to schema changes. All new columns are recorded in the rescued data column.
- **none** = ignore any new columns (Does not evolve the schema, new columns are ignored, and data is not rescued unless the rescuedDataColumn option is set. Stream does not fail due to schema changes.)

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Project Overview

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Medallion Architecture



Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Project Architecture

Pedal Cycle

Two Wheeler

LGV

Unity Catalog - Governance

/landing

Bronze Layer

Silver Layer

Gold Layer

Power BI

Container

Bronze Schema

Silver Schema

Gold Schema

Azure Data Lake Storage Gen 2

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Raw Traffic counts dataset

Pedal Cycle

Two Wheeler motor vehicles

Buses and coaches

LGV (Large Goods Vehicle)

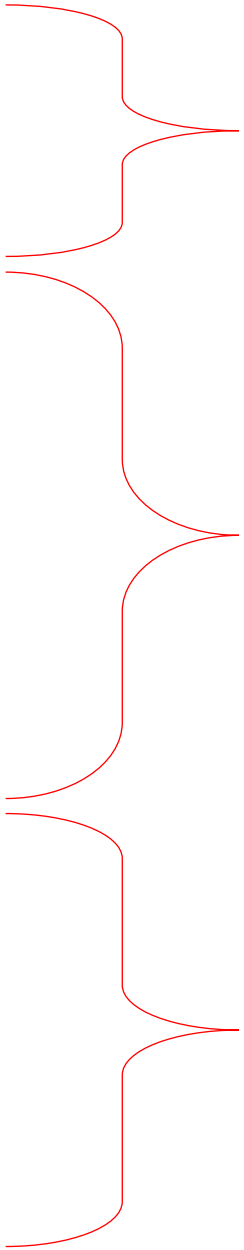
HGV (Heavy Goods Vehicle)

Electric Vehicles

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Data Dictionary

1. Record ID
2. Count point id
3. Direction of travel
4. Year
5. Count date
6. hour
7. Region id
8. Region name
9. Local authority name
10. Road name
11. Road Category ID
12. Start junction road name
13. End junction road name
14. Latitude
15. Longitude
16. Link length km
17. Pedal cycles
18. Two wheeled motor vehicles
19. Cars and taxis
20. Buses and coaches
21. LGV Type
22. HGV Type
23. EV Car
24. EV Bike



Vehicle flow point

Travel info of vehicle

Count of types of vehicle

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Data Dictionary

1. Record ID	=	Uniquely identifies a record
2. Count point id	=	A unique reference for the road link
3. Direction of travel	=	Direction of travel
4. Year	=	Year it happened
5. Count date	=	The date when the actual count took place
6. hour	=	Hour 7 represents from 7am to 8am, and 17 tells from 5pm to 6pm.
7. Region id	=	Website region identifier
8. Region name	=	The name of the Region that travel took place
9. Local authority name	=	Local authority that region
10. Road name	=	This is the road name (for instance M25 or A3).
11. Road Category ID	=	Uniquely identifies road ID
12. Start junction road name	=	The road name of the start junction of the link
13. End junction road name	=	The road name of the end junction of the link
14. Latitude	=	Latitude of the Location
15. Longitude	=	Longitude of the Location
16. Link length km	=	Total length of the network road link
17. Pedal cycles	=	Counts for pedal cycles
18. Two wheeled motor vehicles	=	Counts of Two wheeled motor vehicles
19. Cars and taxis	=	Counts of Cars and taxis
20. Buses and coaches	=	Counts of Buses and coaches
21. LGV Type	=	Counts of LGV Type
22. HGV Type	=	Counts of HGV Type
23. EV Car	=	Counts of EV Car
24. EV Bike	=	Counts of EV Bike

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Raw Roads dataset

Road Category

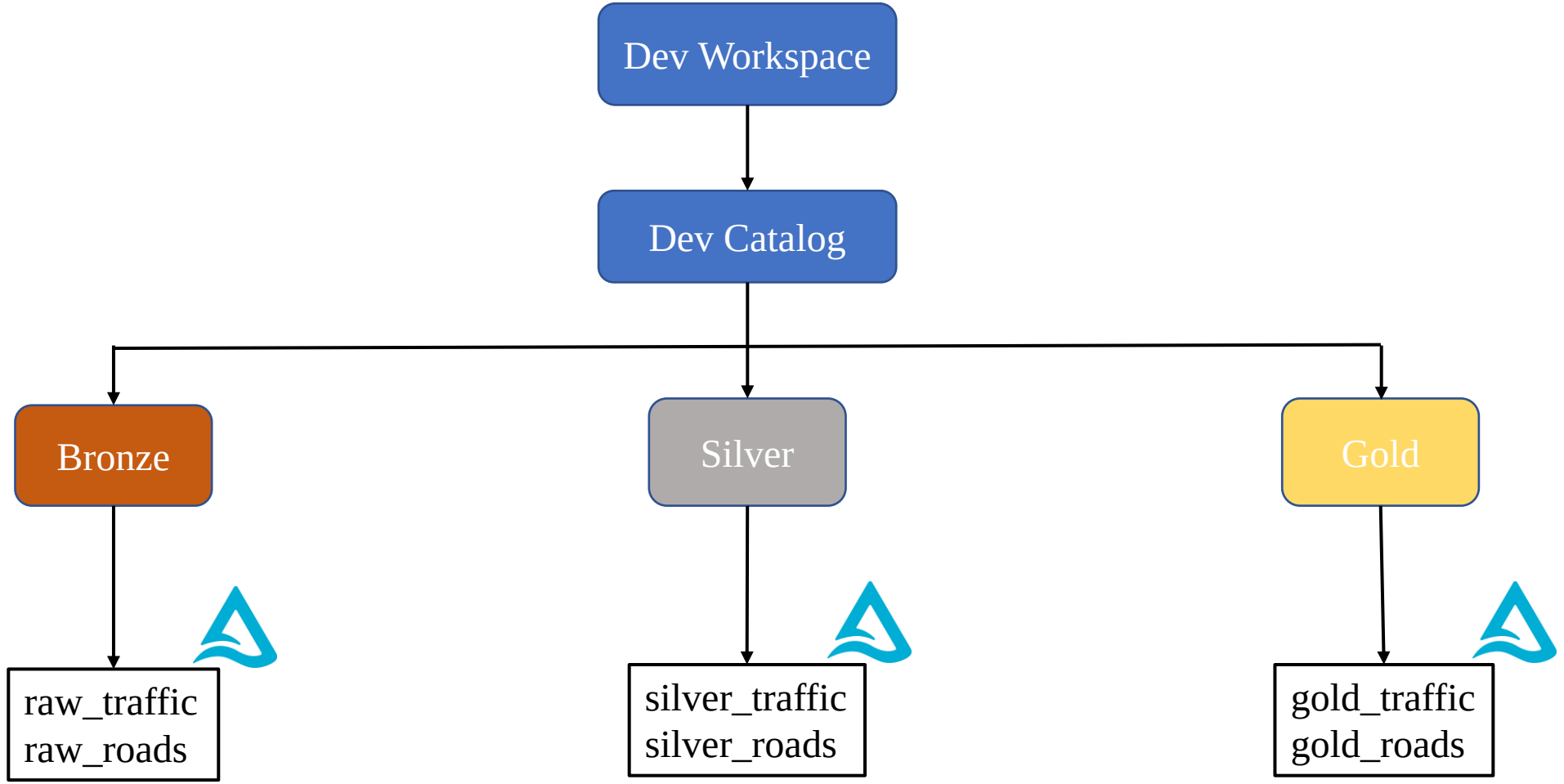
Road Types

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Project Setup

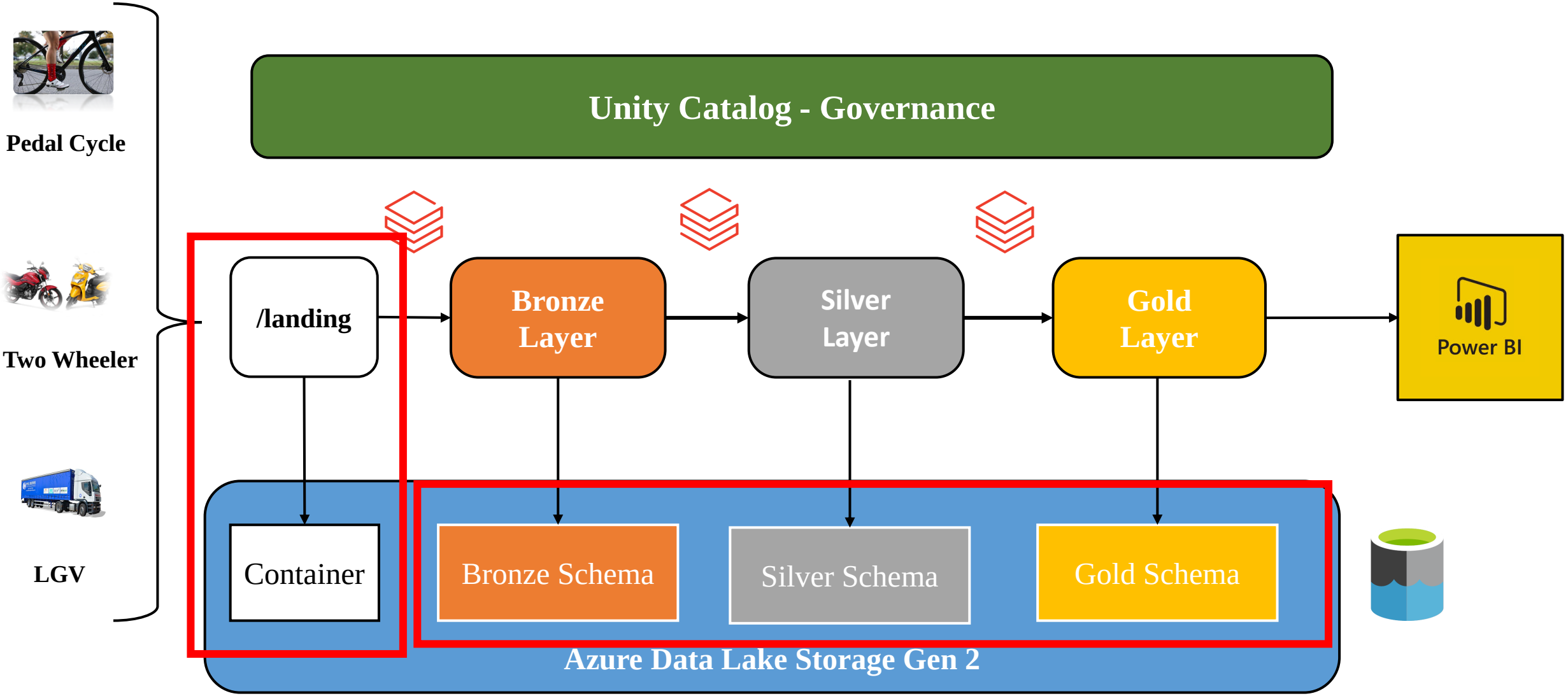
Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Expected Setup

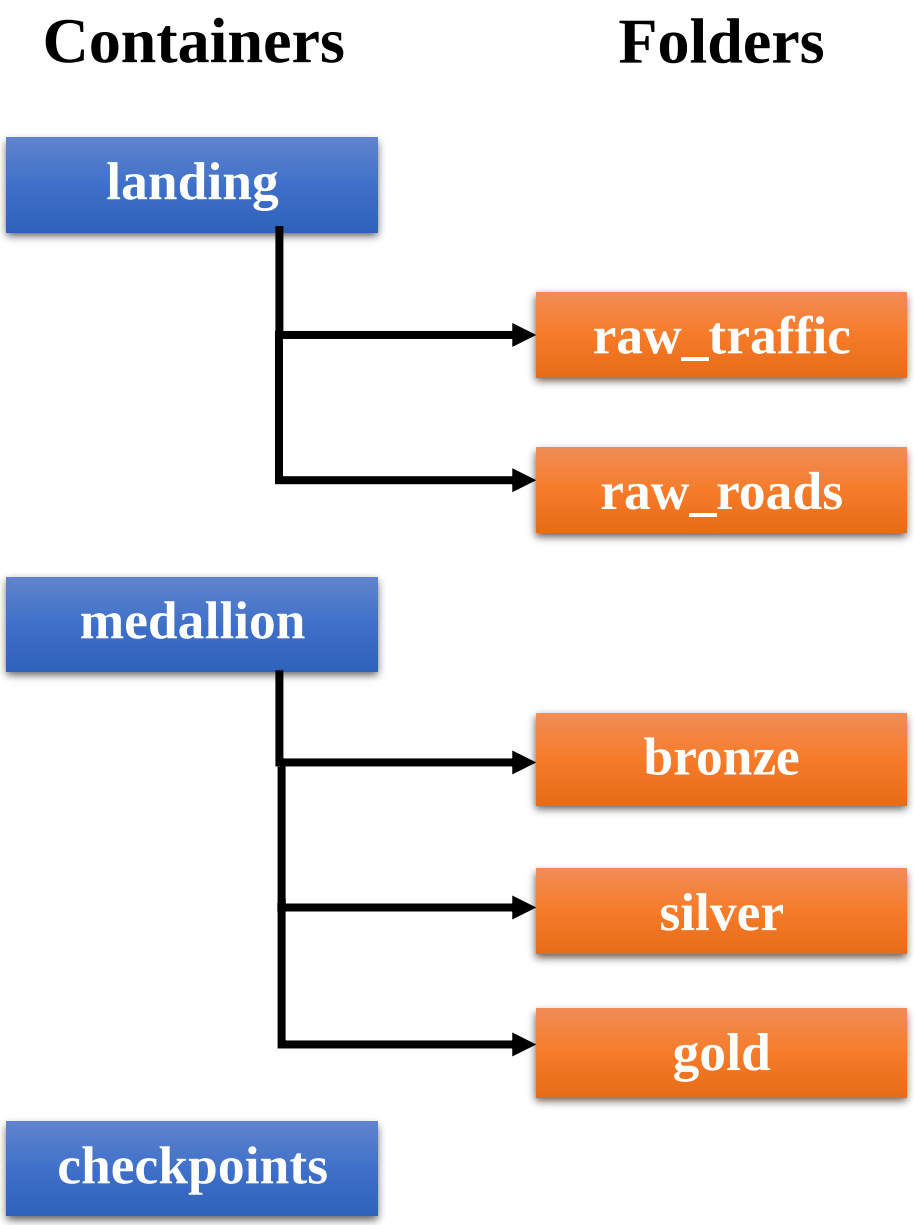


Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Project Architecture



Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/



External Locations

1. Landing
2. Checkpoints
3. Bronze
4. Silver
5. Gold

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Ingesting Raw Traffic dataset

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Ingestion to Bronze

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Project Architecture

Pedal Cycle

Two Wheeler

LGV

Unity Catalog - Governance

/landing

Bronze Layer

Silver Layer

Gold Layer

Power BI

Container

Bronze Schema

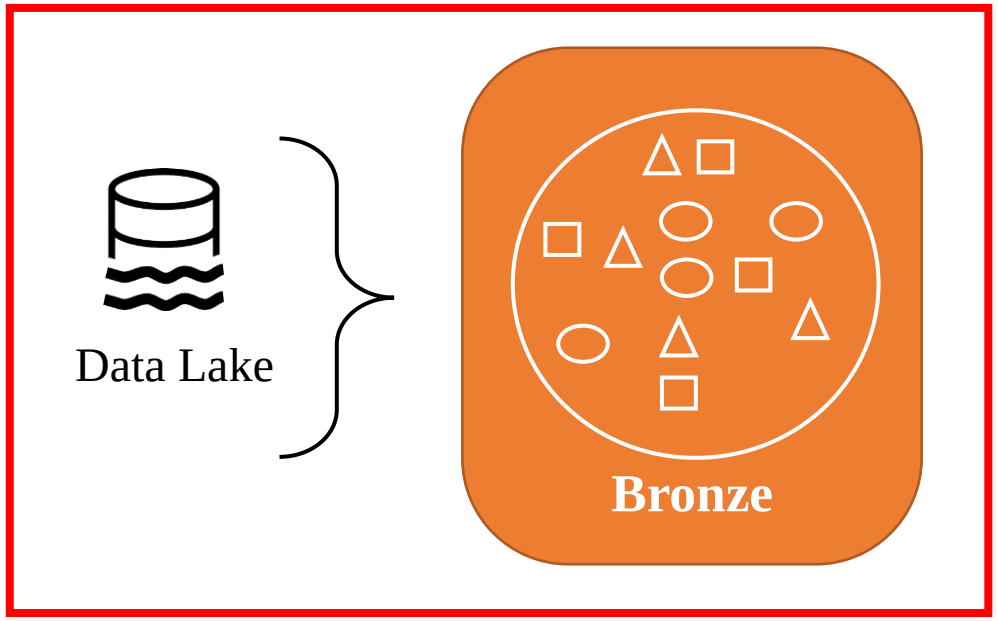
Silver Schema

Gold Schema

Azure Data Lake Storage Gen 2

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Ingesting data to Bronze Layer



Schema: bronze

Tables:

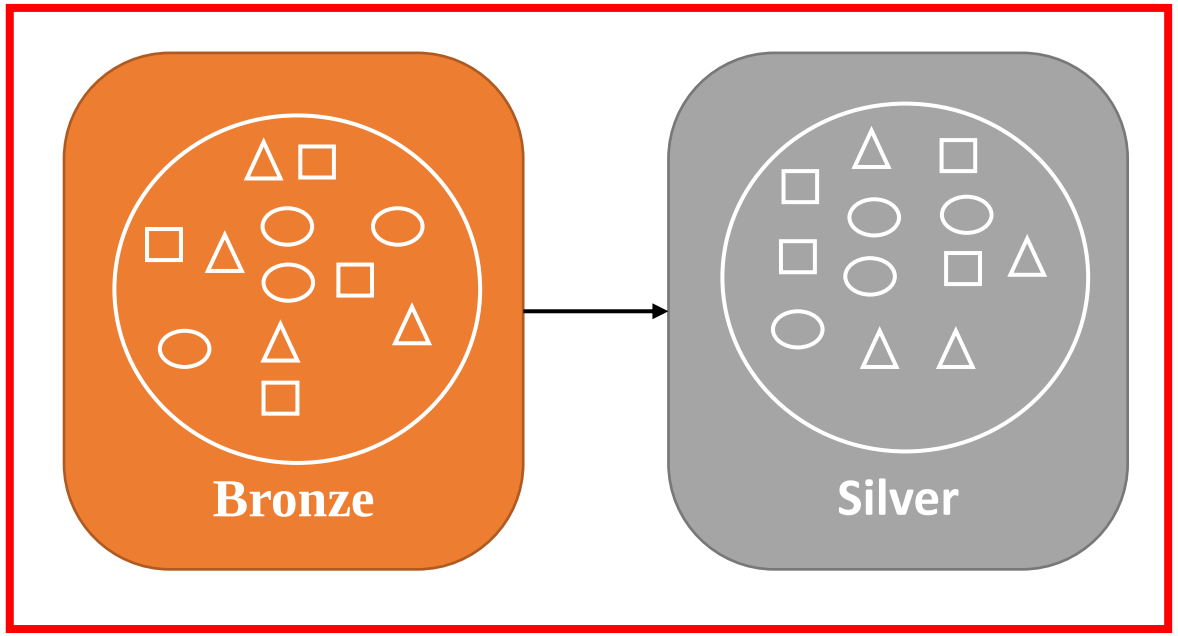
1. raw_traffic
2. raw_roads

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Silver Layer Transformations

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Transforming data in Silver Layer



Schema: Silver

Tables:

1. silver_traffic
2. silver_roads

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Transforming Raw Traffic dataset

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Renaming Columns

1. Record ID	Record_ID
2. Count point id	Count_point_id
3. Direction of travel	Direction_of_travel
4. Year	Year
5. Count date	Count_date
6. hour	hour
7. Region id	Region_id
8. Region name	Region_name
9. Local authority name	Local_authority_name
10. Road name	Road_name
11. Road Category ID	Road_Category_ID
.	.
.	.
.	.

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Creating Electric_Vehicles_Count

1. Record_ID
2. Count_point_id
3. Direction_of_travel
4. Year
5. Count_date
6. hour
7. Region_id
.
.
24. EV_Bike

1. Record_ID
2. Count_point_id
3. Direction_of_travel
4. Year
5. Count_date
6. hour
7. Region_id
.
.
24. EV_Bike
25. Electric_Vehicles_Count

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Creating Motor_Vehicles_Count

1. Record_ID
2. Count_point_id
3. Direction_of_travel
4. Year
5. Count_date
6. hour
7. Region_id
.
.
25. Electric_Vehicles_Count

1. Record_ID
2. Count_point_id
3. Direction_of_travel
4. Year
5. Count_date
6. hour
7. Region_id
.
.
25. Electric_Vehicles_Count
26. Motor_Vehicles_Count

Two_wheeled_motor_vehicle + Cars_and_taxis + Buses_and_coaches + LGV_Type + HGV_Type + Electric_Vehicle_Count

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Transforming Raw Roads dataset

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Raw Roads dataset

Road Category

Road Types

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Renaming Columns

1. Road ID
2. Road category id
3. Road category
4. Region id
5. Region name
6. Total link length km
7. Total link length miles
8. All motor vehicles

1. Record_ID
2. Road_category_id
3. Road_category
4. Region_id
5. Region_name
6. Total_link_length_km
7. Total_link_length_miles
8. All_motor_vehicles

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Creating Road_Category_Name

1. Record_ID
2. Road_category_id
3. Road_category
4. Region_id
5. Region_name
6. Total_link_length_km
7. Total_link_length_miles
8. All_motor_vehicles

1. Record_ID
2. Road_category_id
3. **Road_category**
4. Region_id
5. Region_name
6. Total_link_length_km
7. Total_link_length_miles
8. All_motor_vehicles
9. Road_Category_Name

When **Road_Category** = TA THEN Class A Trunk Road
When **Road_Category** = TM THEN Class A Trunk Motor
When **Road_Category** = PA THEN Class A Principal road
When **Road_Category** = PM THEN Class A Principal Motorway
When **Road_Category** = M THEN Class B road

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Creating Road_Type

1. Record_ID
2. Road_category_id
3. Road_category
4. Region_id
5. Region_name
6. Total_link_length_km
7. Total_link_length_miles
8. All_motor_vehicles
9. Road_Category_Name

1. Record_ID
2. Road_category_id
3. Road_category
4. Region_id
5. Region_name
6. Total_link_length_km
7. Total_link_length_miles
8. All_motor_vehicles
9. **Road_Category_Name**
10. Road_Type

WHEN **Road_Category_Name** Contains Class A THEN Major
WHEN **Road_Category_Name** Contains Class B THEN Minor

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Transforming & Loading Silver datasets

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Creating Vehicle_Intensity

1. Record_ID
2. Count_point_id
3. Direction_of_travel
4. Year
5. Count_date
6. hour
7. Region_id
.
.
26. Motor_Vehicles_Count

1. Record_ID
2. Count_point_id
3. Direction_of_travel
4. Year
5. Count_date
6. hour
7. Region_id
.
.
26. Motor_Vehicles_Count
27. Vehicle_Intensity

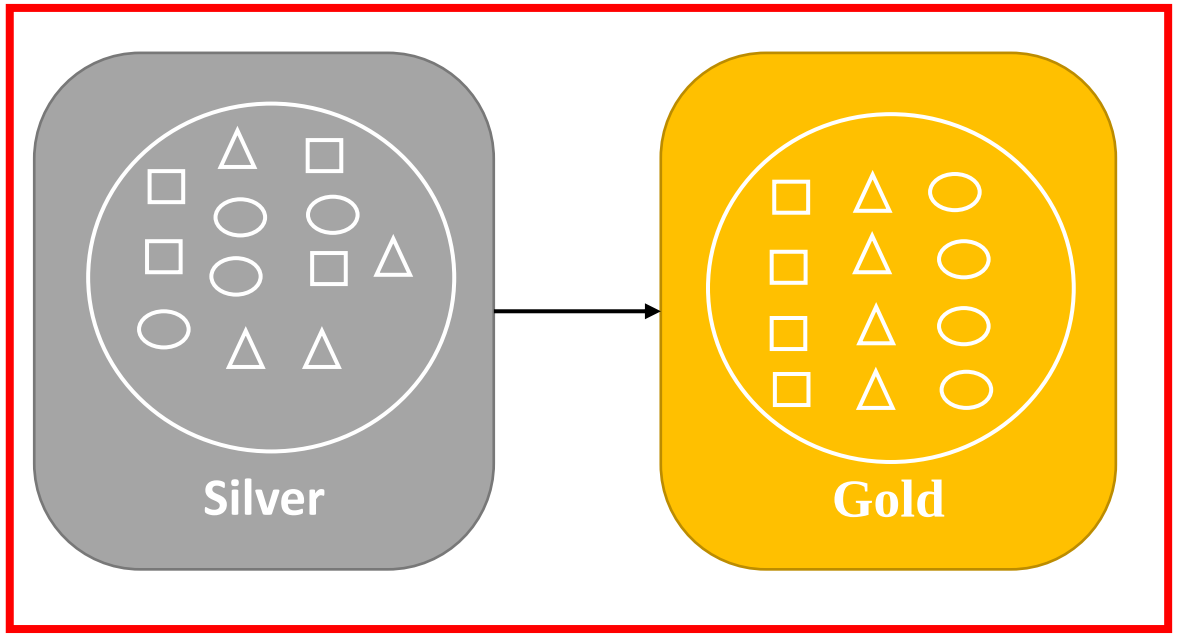
Vehicle Intensity = Motor_Vehicles_Count / Link_length_km

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Loading to Gold Layer

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Loading data to Gold Layer



Schema: Gold

Tables:

1. gold_traffic
2. gold_roads

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Orchestrating with Workflows

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

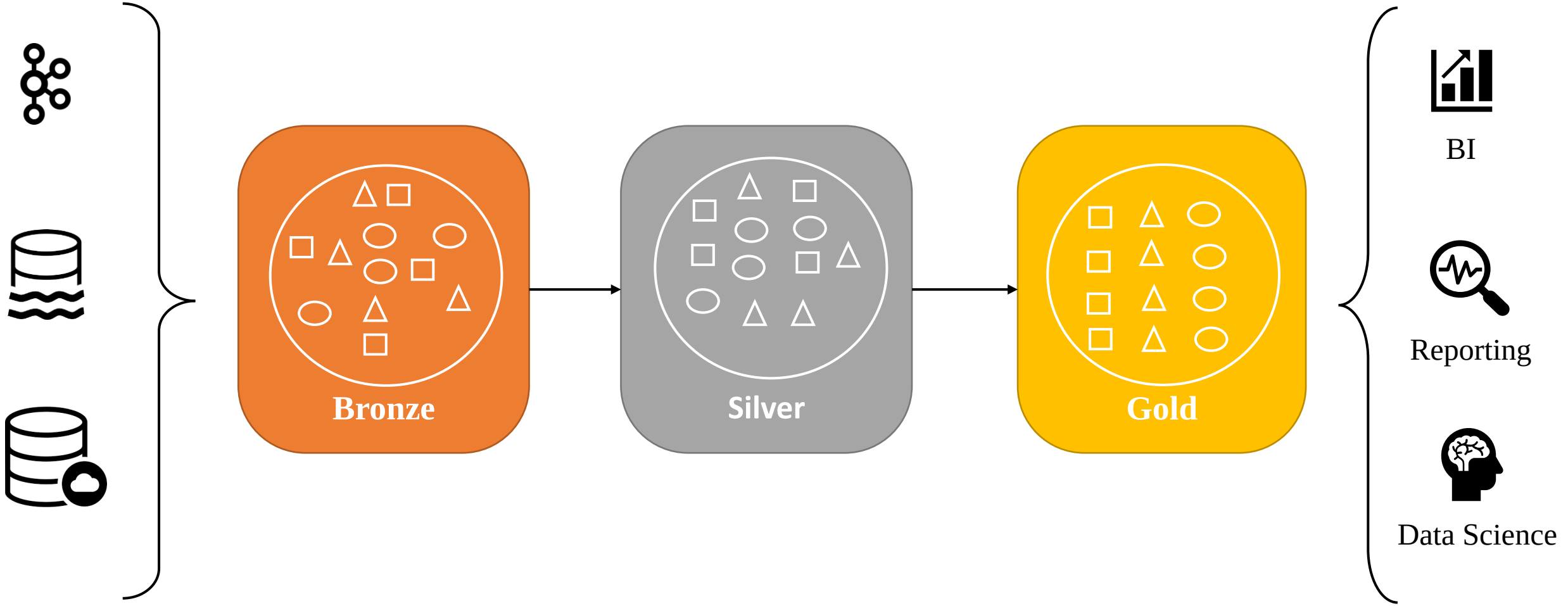
Reporting data to Power BI

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Delta Live Tables (DLT)

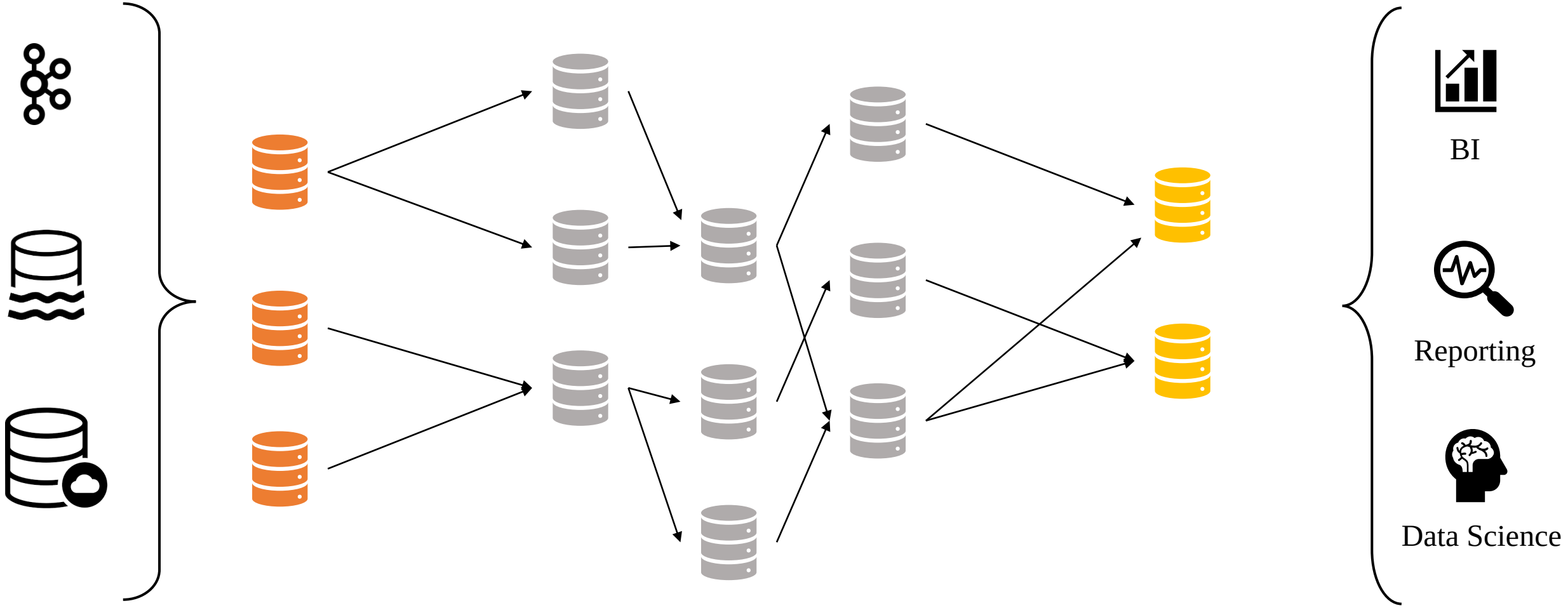
Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Delta Live Tables (DLT) Origin



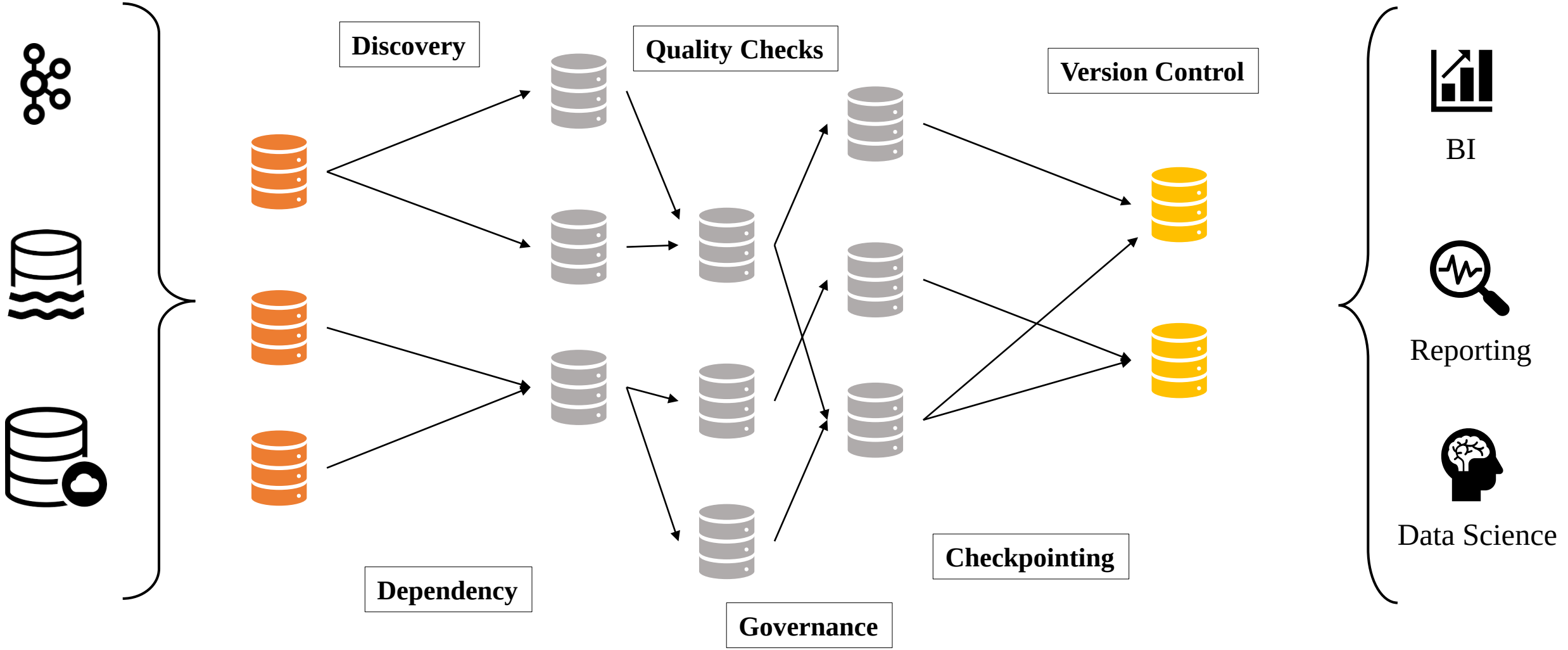
Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Medallion/Lakehouse Architecture Tables



Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Considerations in Lakehouse Architecture



Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Declarative programming

Declarative programming say **what should be done ,** not **how to do it**

Procedural programming

```
Numbers = [..]

Sum = 0

For n in numbers:
        sum = sum + n

Print (n)
```

Declarative programming

```
SELECT SUM(n)
FROM numbers
```

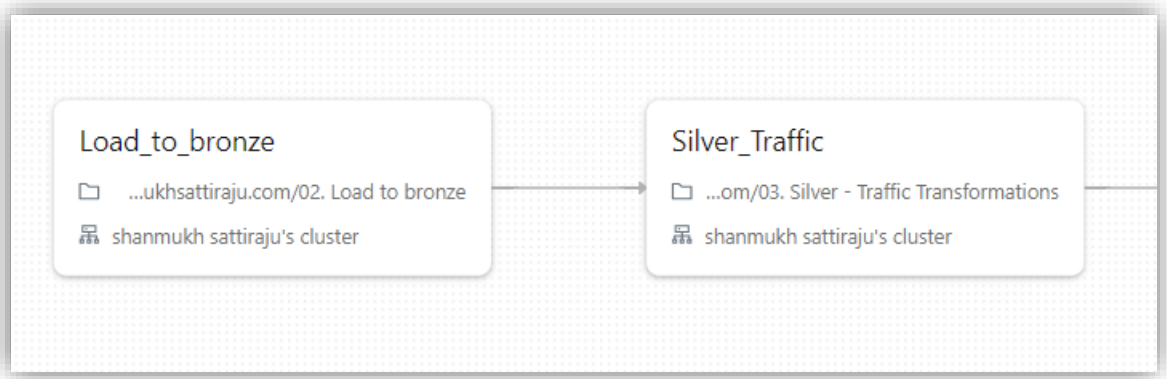
Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Declarative ETL with DLT

Declarative programming say **what should be done ,** not **how to do it**

Procedural ETL

- Apache Airflow
- Azure Data Factory



Declarative ETL

Delta live tables

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Delta Live Tables (DLT)

Delta Live Tables (DLT) is **a declarative ETL framework** for the Databricks Data Intelligence Platform that helps data teams simplify streaming and batch ETL cost-effectively.

Simply define the transformations to perform on your data and let DLT pipelines automatically manage task orchestration, cluster management, monitoring, data quality and error handling.

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Delta Live Table Execution

- Requires premium workspace
- Supports only Python and SQL languages
- Can’t run interactively
- No support for magic commands like %run

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Expectations in DLT pipeline

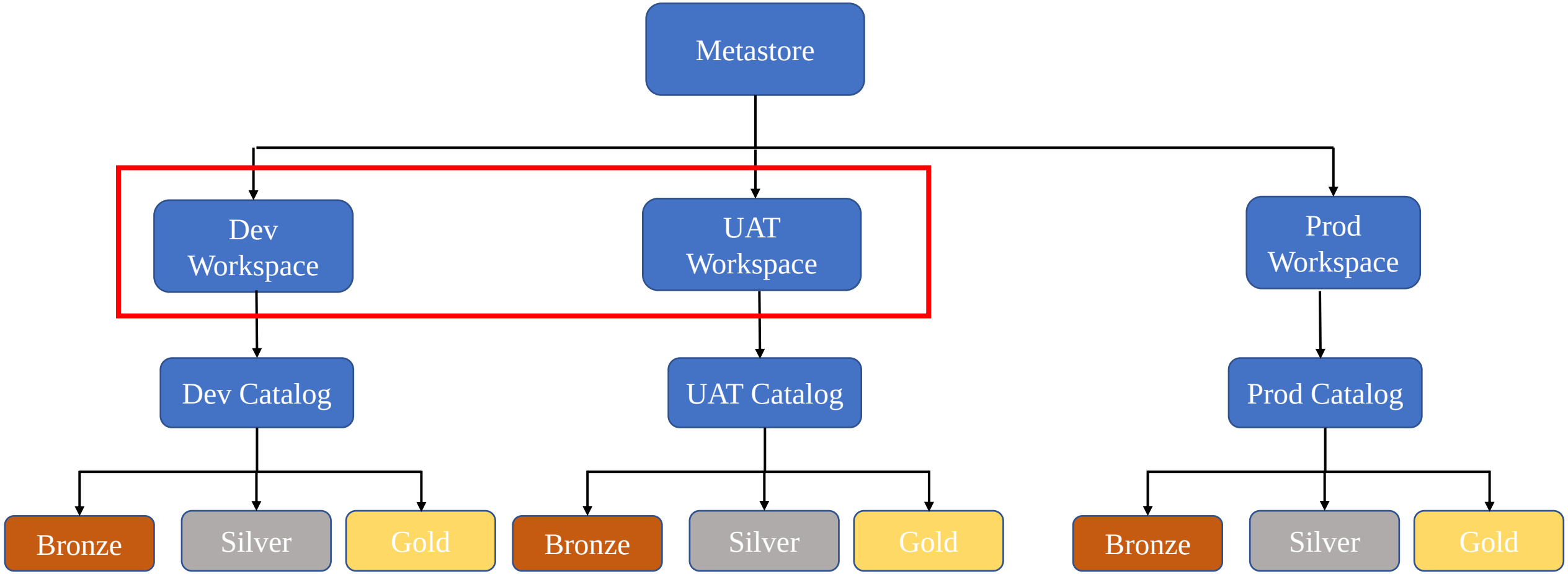
Action	Result	Usage
warn (default)	Invalid records are written to the target; failure is reported as a metric for the dataset.	--
drop	Invalid records are dropped before data is written to the target; failure is reported as a metrics for the dataset.	On Violation Drop Row
fail	Invalid records prevent the update from succeeding. Manual intervention is required before	On Violation Fail Update

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Continuous Integration and Continuous Deployment

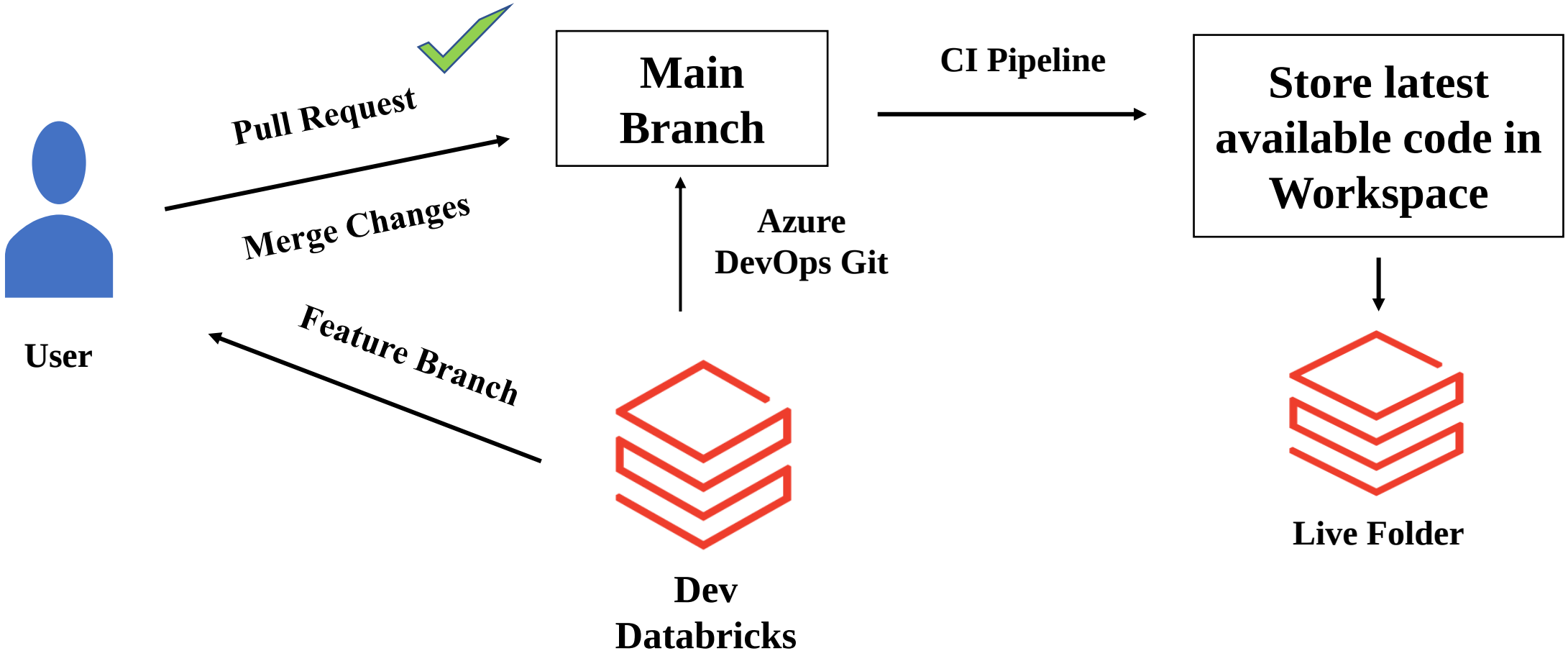
Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Expected Setup



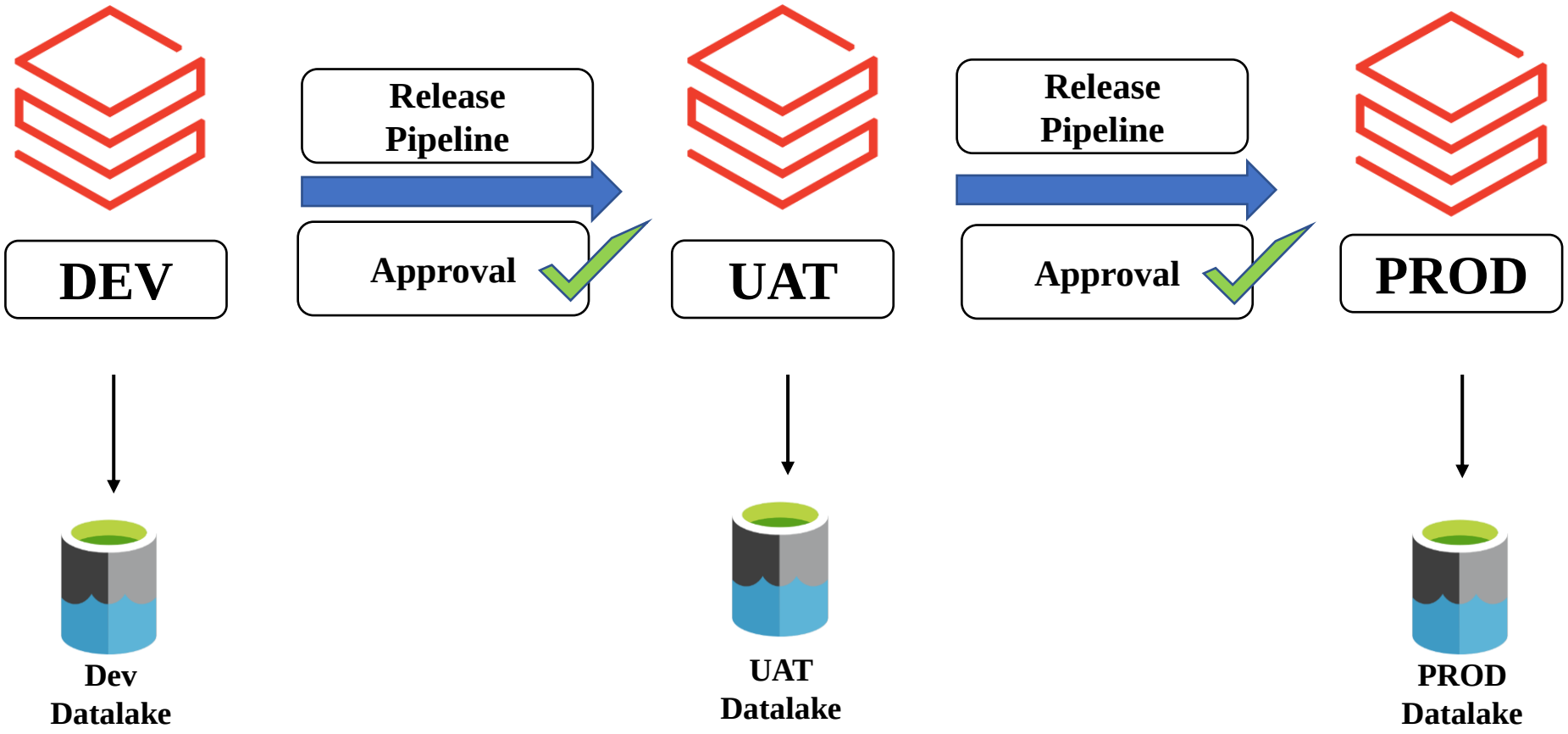
Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Continuous Integration



Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Continuous Deployment



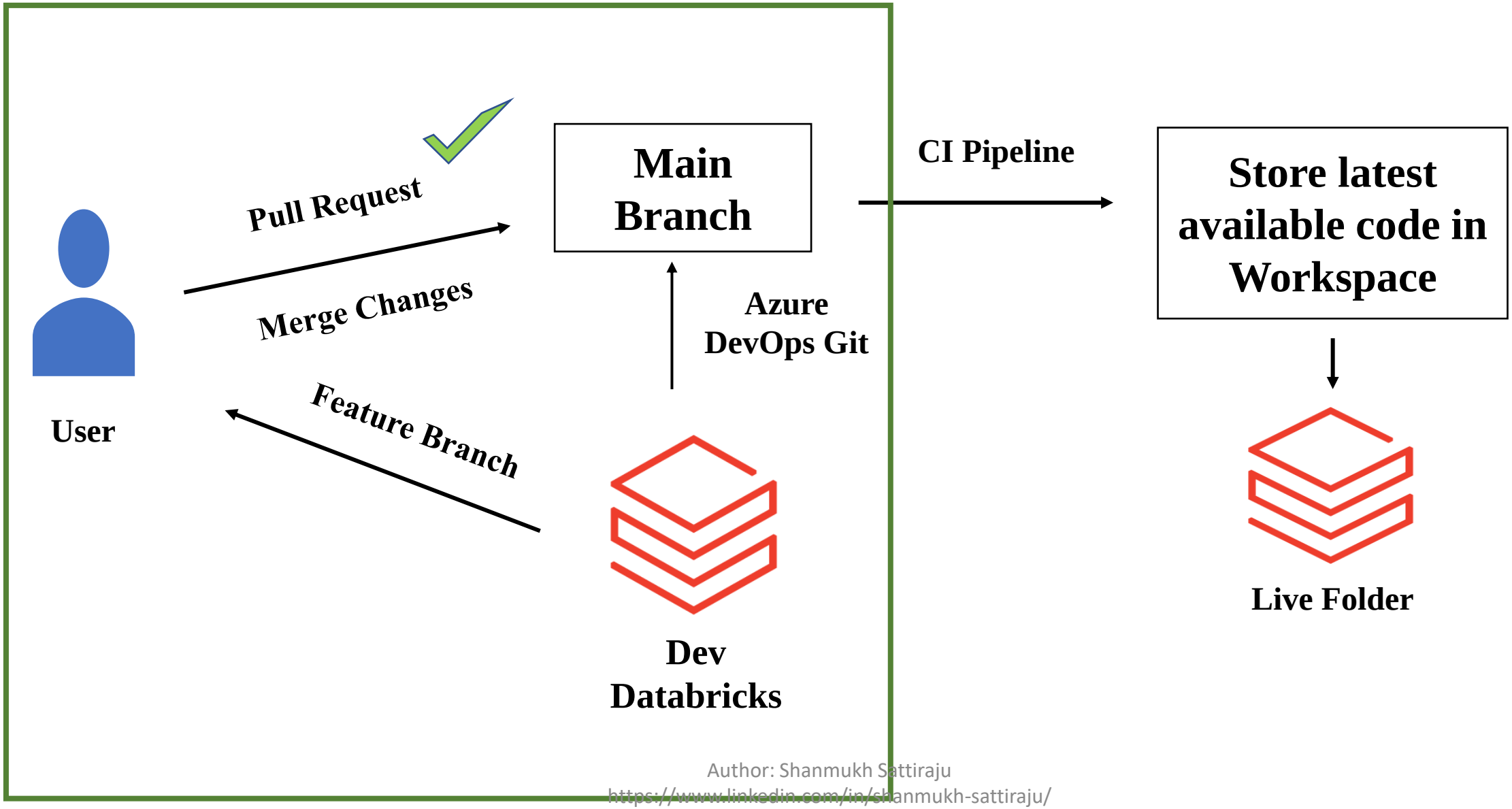
Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Creating UAT resources in Azure

- **Resource Group:** databricks-uat-rg
- **Databricks workspace:** databricks-uat-ws
- **Storage Account:** databricksuatstg

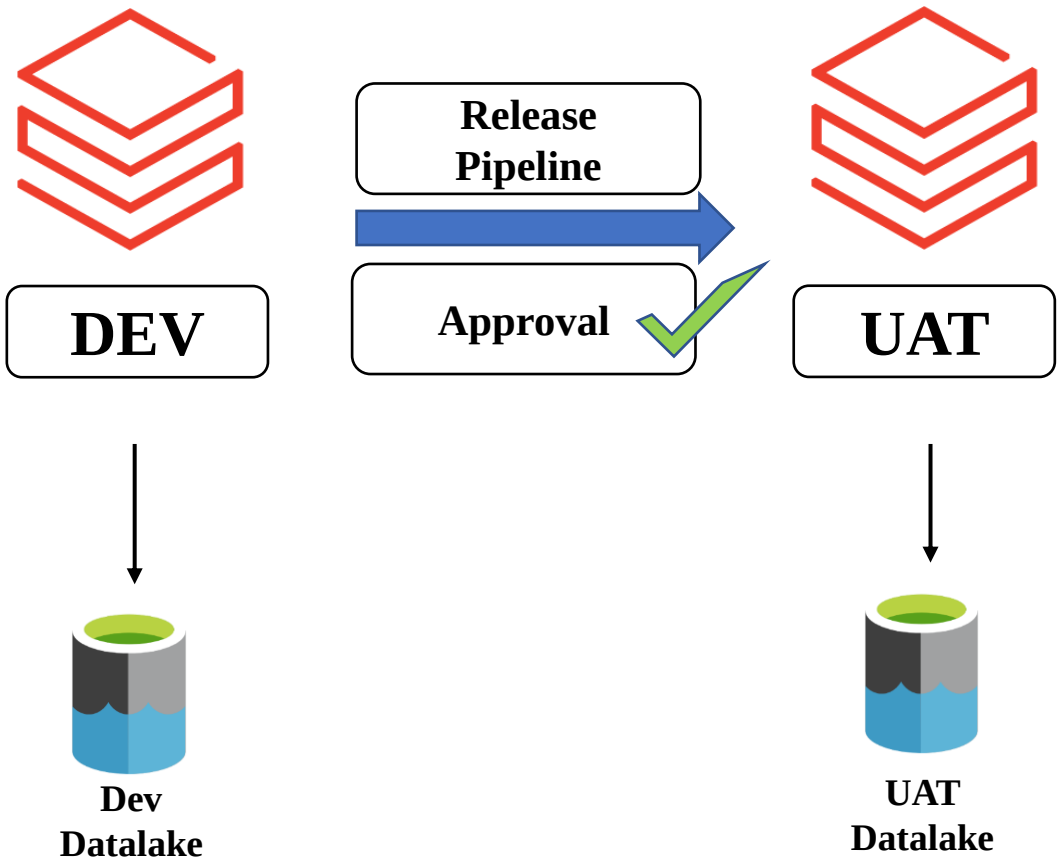
Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Continuous Integration



Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Continuous Deployment



Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/

Congratulations

Author: Shanmukh Sattiraju
https://www.linkedin.com/in/shanmukh-sattiraju/