# Best Practices for Building a Modern Data Ecosystem

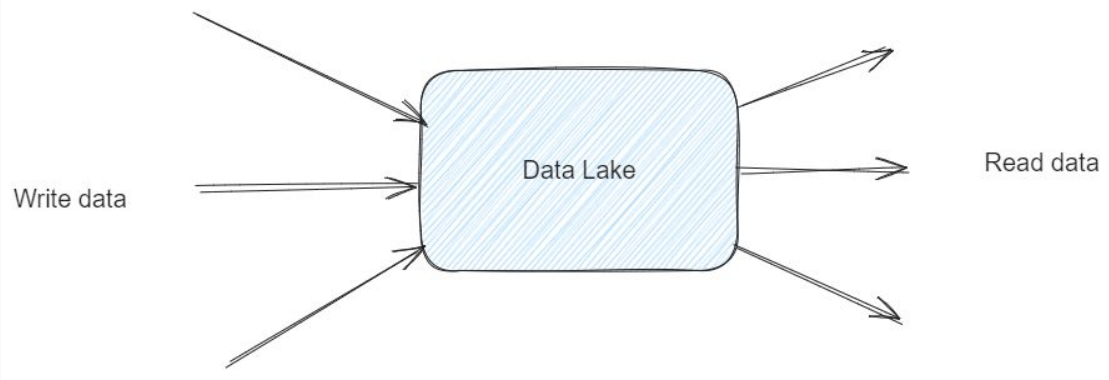
## Challenges and Opportunities

*Raja Chattopadhyay*

*Conf42 Cloud Native 2024*

# An organization needs a place to store data

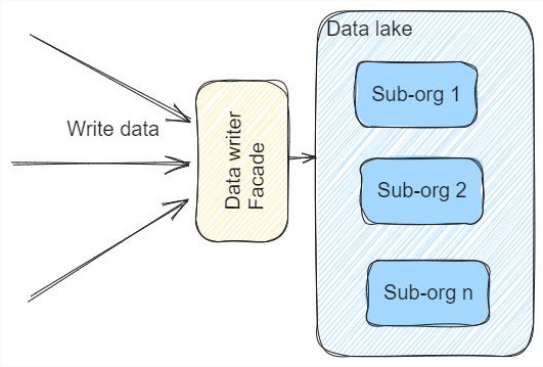
In its simple form, a Data lake is a simple storage where structured and unstructured data can be stored and consumed from.



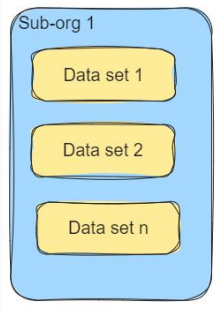

Challenges:

1. Disparate format (parquet, CSV, etc) makes it hard for consumers
2. Organizing the data to be written so that the consumers know where to read from
3. Data is added at some interval. This creates a problem to present a comprehensive view

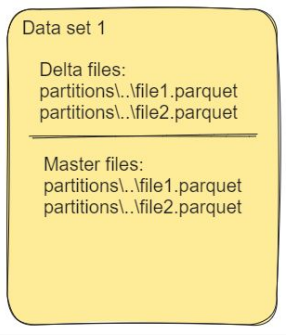
# Organizing Data Storage




- Data writer writes the data under an area designated for the sub-org. For this it needs to know the identity of the publisher and the dataset it is writing.
- Data reader let's the consumer of data read it based on its identity and the dataset.



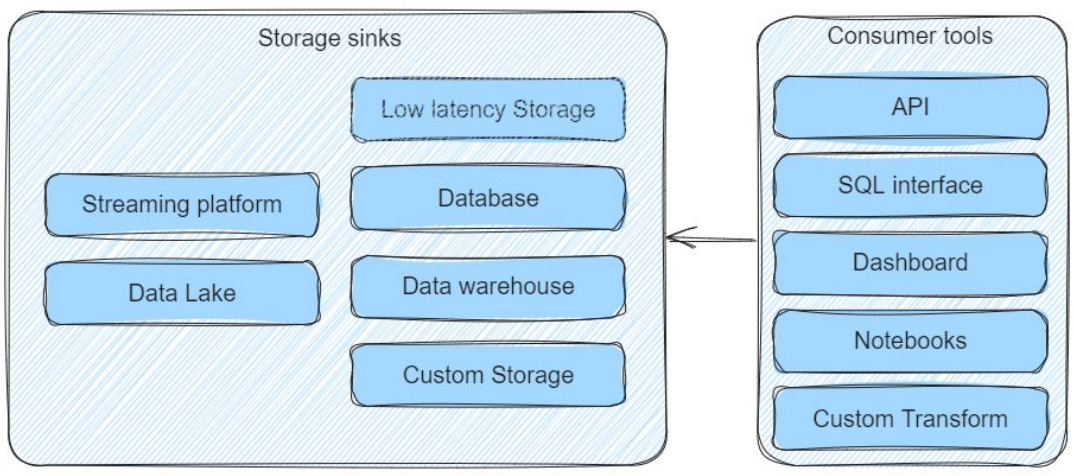

- Data within the sub-org is structured as datasets




- Data within the area for a dataset will be organized as delta files and aggregates.
- Data will be partitioned
- Data will be stored in a standard format like parquet.
- Data can be stored in formats like Iceberg or Hudi to enable transaction.

# Transitioning from Storage to Utilization: Empowering Data Consumption

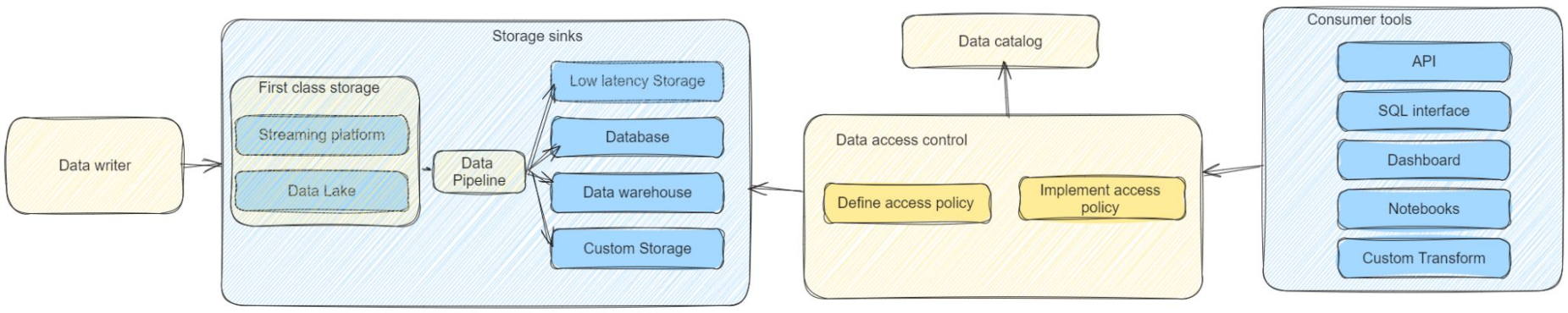
| Who will consume? | Consume from where? | How will they consume? |
| --- | --- | --- |
| • Data analysts<br>• Data scientists<br>• Real time applications<br>• Batch applications | • Data warehouse<br>• Low latency store<br>• Database<br>• Custom storages | • API<br>• Dashboard<br>• SQL interface<br>• Custom transformation |




Challenges:

1. Consistency of data in various sinks
2. Deduplication of effort to write data
3. Control to prevent unauthorized access

# Data Pipeline, Sinks, Catalog and Access control




**Data pipeline**:

- Knows how to and where to write data
- It gets this information from the data catalog
- Responsible for data consistency between sinks

**Sinks**:

- Batch data first goes to the data lake
- Real time data first goes to streaming platform
- Lake and streaming platform then sends data to other platforms and to each other
- Alerting service for others to consume

**Catalog**:

- Data in each platform is registered in the catalog[1]
- This stores the routing rules and other information

**Access control**:

- This has policies that control access to the underlying data
- Data owners are responsible for granting access
- Consumers are granted access according to their credentials[2]

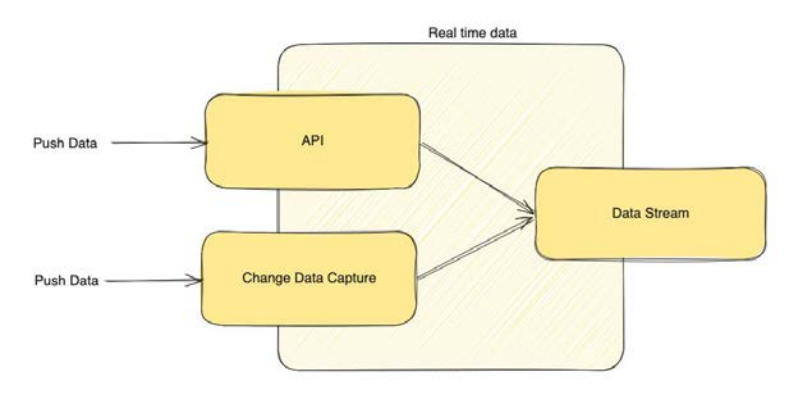
---

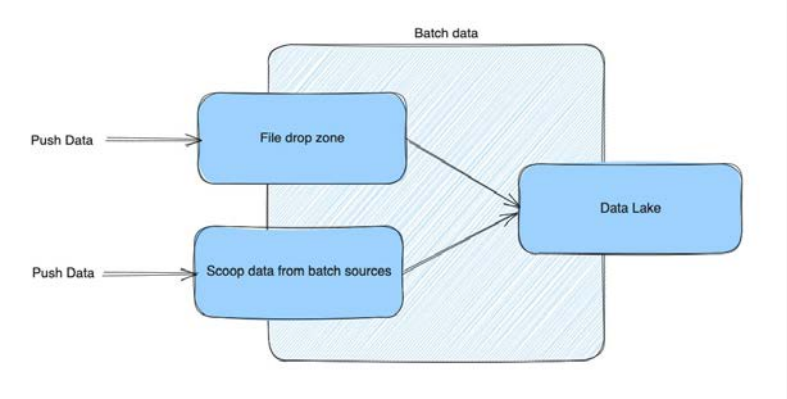
**Note**:
[1] Data catalog can be created by following DCAT
[2] Access to the consumers can be granted using “Tag based access control” (TBAC)

# Data Writer

The data writer is an important component. Some of its characteristics should be:

1. Be able to read and write data from various sources and various formats
2. Be able to scale up and down according to load
3. Cater to both batch as well as real time data



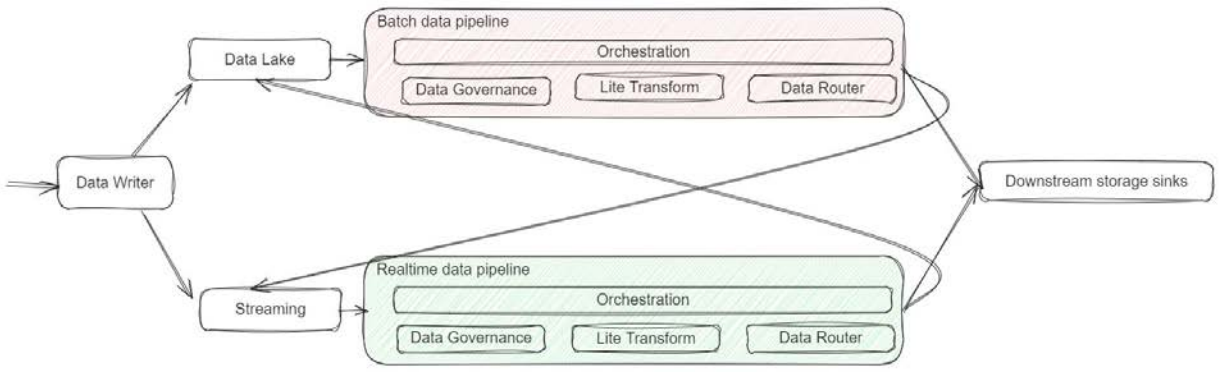

**Pull real time data from**:

- Other streams
- Databases

**Pull batch data from**:

- File based stores
- Logs
- Databases

# Data Pipeline




What should happens after data is written to the first class storages (Lake and Stream):

- Data governance
    - Schema check
    - Quality checks
    - Scan for sensitive data
- Routing data to various other destinations.
    - Ensure that data is eventually written
- Lite transformations like date type changes and converting to a unique format.

**Data pipeline**:

- Should implement a plug and play approach so that other governance could be added in time.
- If the data is batch data then route it to streaming platform as well as other batch platforms.
- If the data is real-time then route it to the lake after batching. It will also send the data to other real-time platforms.

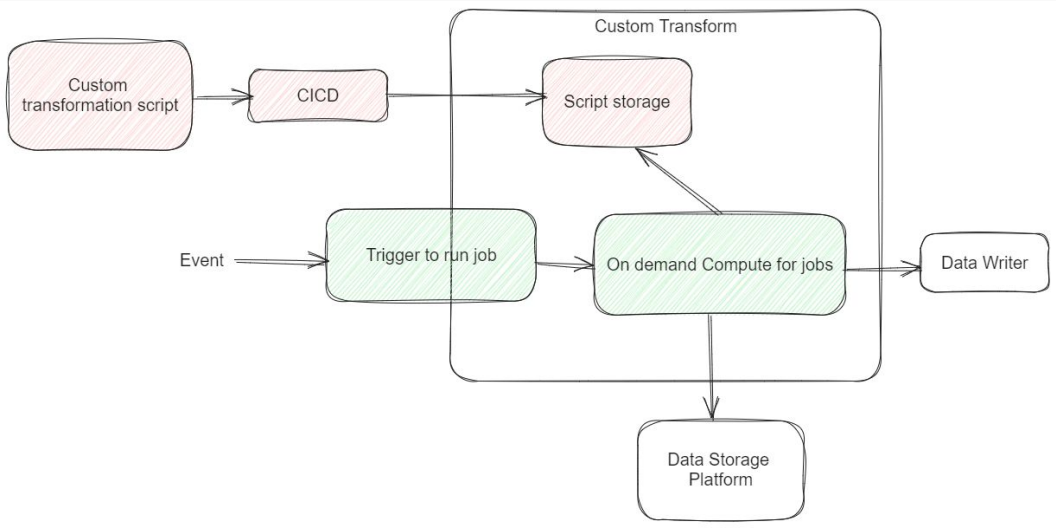
# Data transformation platform

**Challenges:**

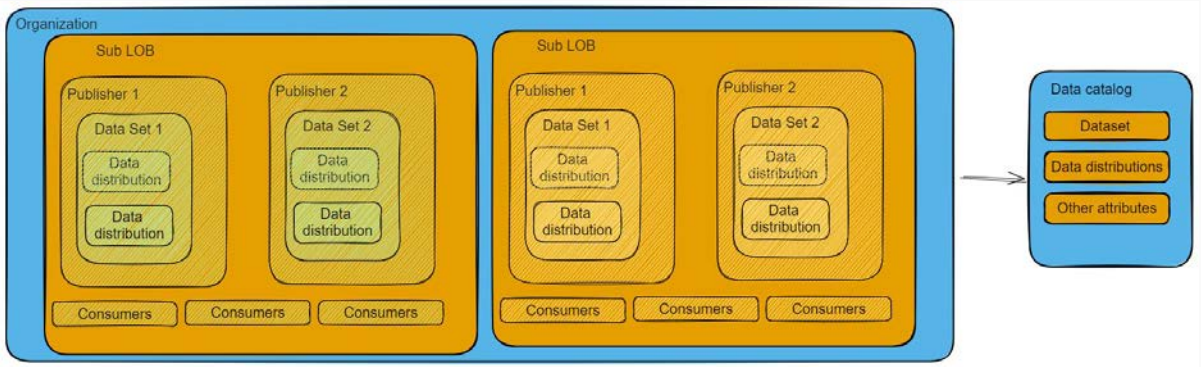
- If every team has to create their own compute platform then
    - Capacity will not be optimally used
    - Each team has to spend effort in maintaining the infrastructure
- No unique way to implement dataops

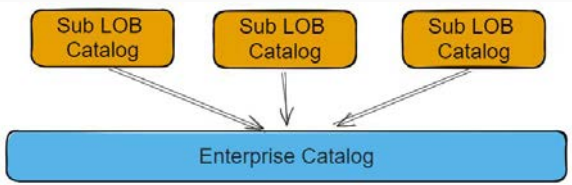
**Data Transformation Platform**:

- The platform should be able to intake transformation scripts in a variety of flavors through a cicd pipeline
- The jobs that need to be run will be triggered and would run on an on-demand compute
- Based on the context of the job, it will be able to access certain data
- The new data will be sent to the data writer for writing to the data platforms



# Data Organization




**Data organization**:

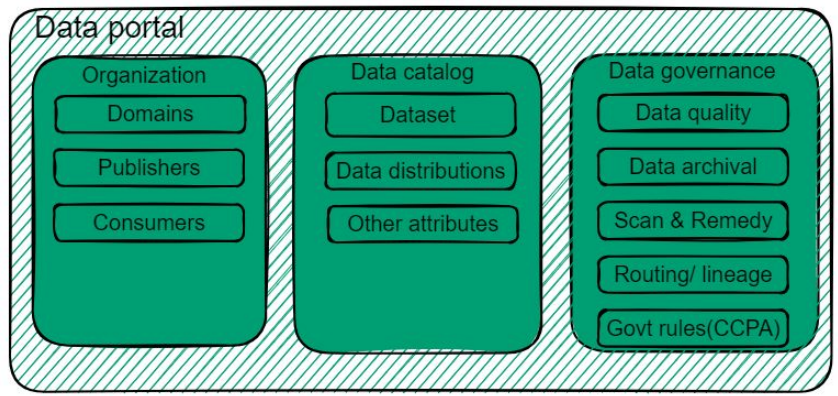
- Data would be organized according to the Sub-LOBs.
- Each dataset would be the responsibility of a publisher.
- Each LOB will have a number of publisher and consumer.
- Information of a dataset should be registered in a catalog before publishing.

**Data catalog**:

- Each Sub-Lob may have their own catalog.
- We can organize these together into an enterprise catalog

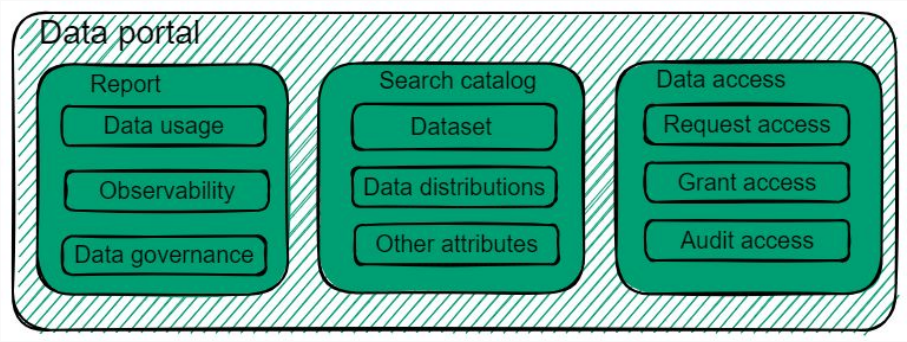
*Note: Diagram for Illustration Purposes Only – Not Actual Implementation*

# Data portal




**Data portal**:

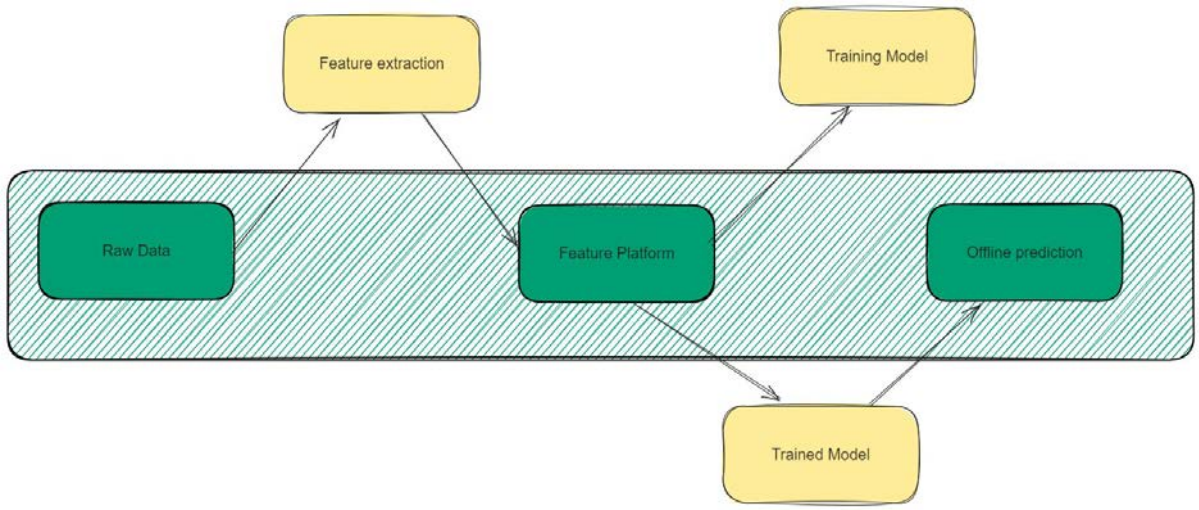
- Enables arranging and cataloging of data.
- Onboarding of datasets should be made frictionless so that more users onboard data.
  - Tier 1 - Just publish but no consumption
  - Tier 2 - Publish data with basic governance information like schema
  - Tier 3 - Additional information like data quality
- Data governance would be defined here.
- Users should be able to search and request access to the data




**Design time use of Data portal**:

- Reports should be run from here
- Catalog search and request for access
- Requesting and granting of access

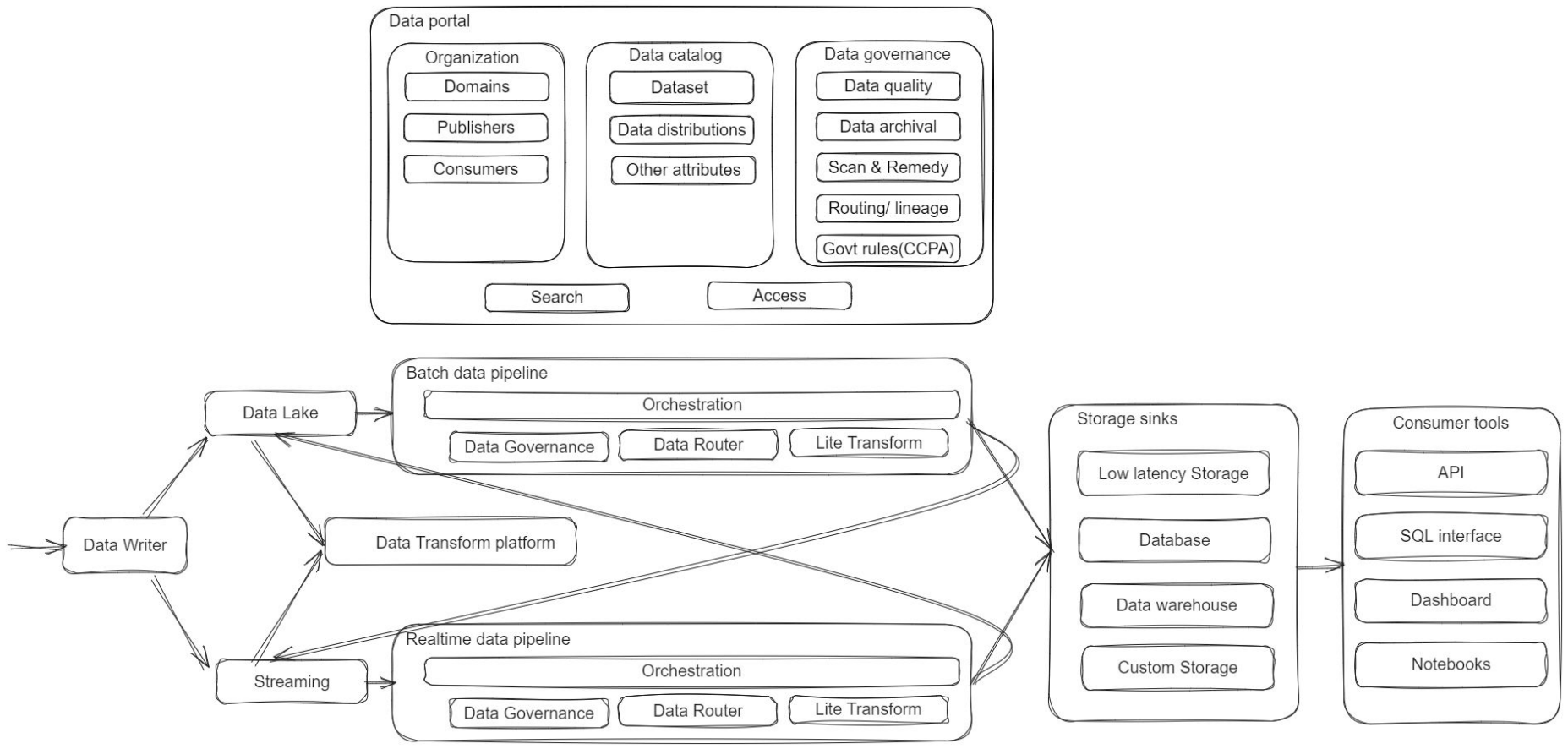
# Machine Learning




**Components of ML**:

- Raw Data - This is the data from where features will be extracted. This would ideally be in the data lake.
- Feature extraction - This is a transformation that will extract the features from the data lake and put them in a certain place within the data lake.
- Feature platform - This consists of a storage in the data lake and a feature serving platform.
- Training model - This will be a compute platform that will train the model
- Offline prediction - This is the prediction that the trained model creates. This is stored in the datalake.

# Putting everything together



# Unique challenges for cloud

# Cloud challenges - AWS

| Challenge | Solution options |
|---|---|
| How each sub-lob will create or share their aws accounts. | 1. Create one central account to store data<br>2. Each sub-lob creates their own account |
| Data security - There is an added concern of data breach | Sensitive data should be scanned and appropriate remediation performed such that even if data is compromised, it will be of very little use. Some options are: Encryption, Masking or Deletion |
| Data integration - Integrating on premise data with new data | 1. Historical data remain on premise<br>2. Migrating historical data to cloud and disconnect with on-prem data |
| Workflow migration | 1. Migrate on premise jobs to the cloud so that new data can be created on cloud<br>2. Let the job remain on premise and create new data on premise. Create another job to move data from premise to cloud. |
| Finops | Each service and storages are tagged with the sub-lob's identity so that this information can be used for billing. |
| Data governance | 1. Use CDMC as a standard<br>2. Use custom data governance components. |
| Cloud capacity | Cloud is someone else's datacenter. Although it is a good abstraction, still we need to plan for the capacity as each account has a soft and a hard limit on the capacity available. |

# Thanks!

Email: raja.chattopadhyay@gmail.com

# An example: Reader and Writer Interaction