

Optimizing data analytics at Etisalat Egypt

Presto at the edge

Mohamed Taha
Etisalat Egypt

prestocon  DAY



The Speaker



Mohamed Taha

Senior Big Data Engineer
Data Management Team
Etisalat Egypt

✉ mohamed.tahassan@etisalat.com

[in /1212mohamedtaha](https://www.linkedin.com/in/1212mohamedtaha)

Agenda

- 01 About e&
- 02 Legacy Data Landscape at Etisalat
- 03 Key Requirements for Data Transformation
- 04 Journey to Presto
- 05 Experience with Presto at Etisalat
- 06 Learnings & Future Work
- 07 Q&A

About e& (From Telco to Techco)



Founded at **1976**
Operates in **32** countries, across **2** continents
Provide services to more than **170 Million** subscribers.

Brands:

e& international

e& enterprise

e& life

e& capital



Launched in Egypt at **2007**
Provide services to more than **30 Million** subscribers.

Services:

Telecom

ADSL

FinTech

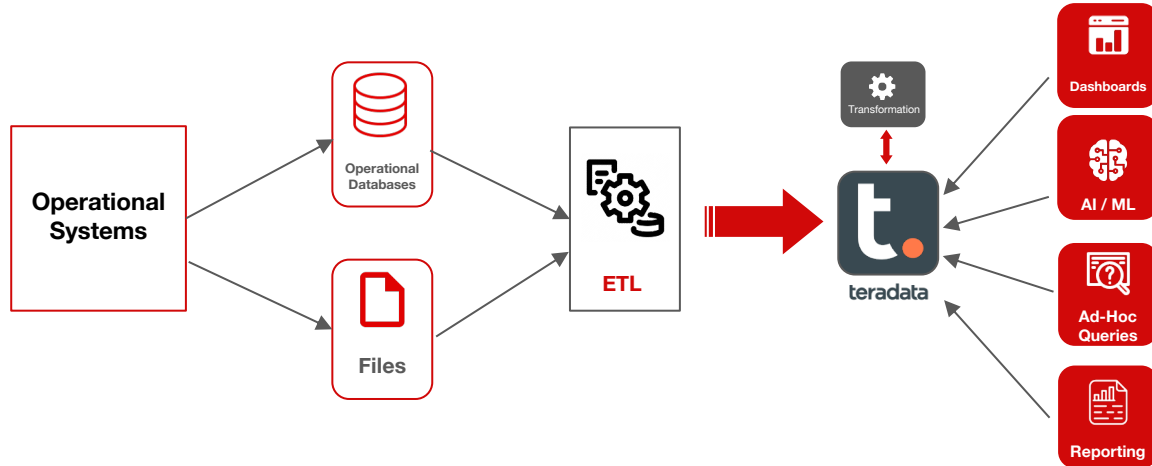
E-Commerce

Entertainment

B2B



Legacy Data Landscape at Etisalat: The Data Warehouse



Data Warehouse

ETL jobs extract data from various sources and load it into a centralized Data Warehouse.

Challenges:

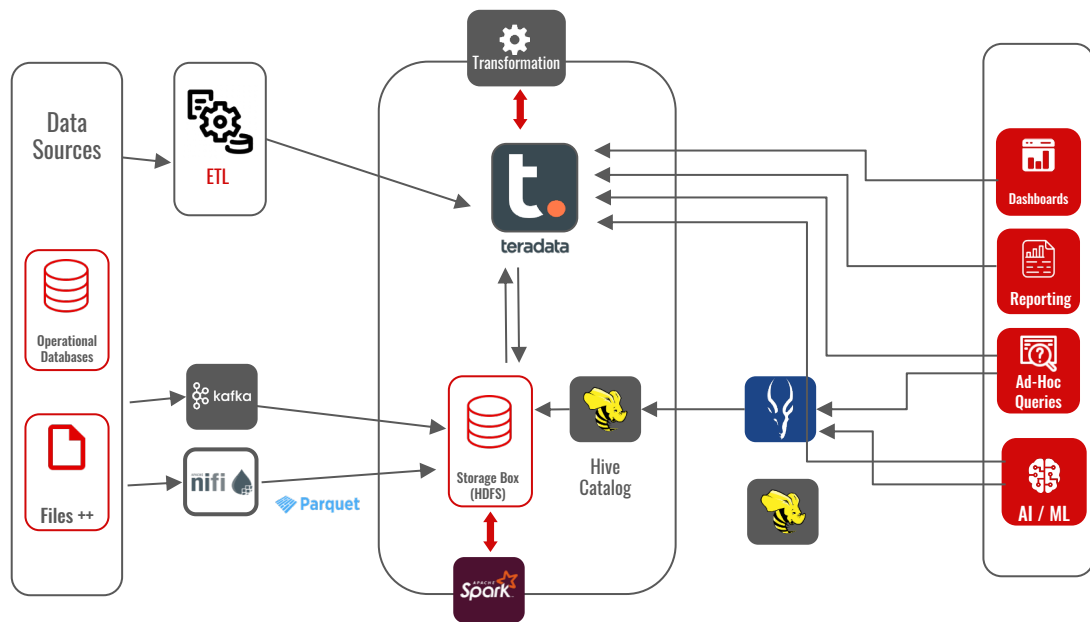
Scalability is expensive.

Slow ETLs

Complex Modeling Challenges.

More ETL Jobs.

Legacy Data Landscape at Etisalat: The Data Lake



A petabyte-scale data lake solution was added.

Streams are splitted over the data lake and the data warehouse

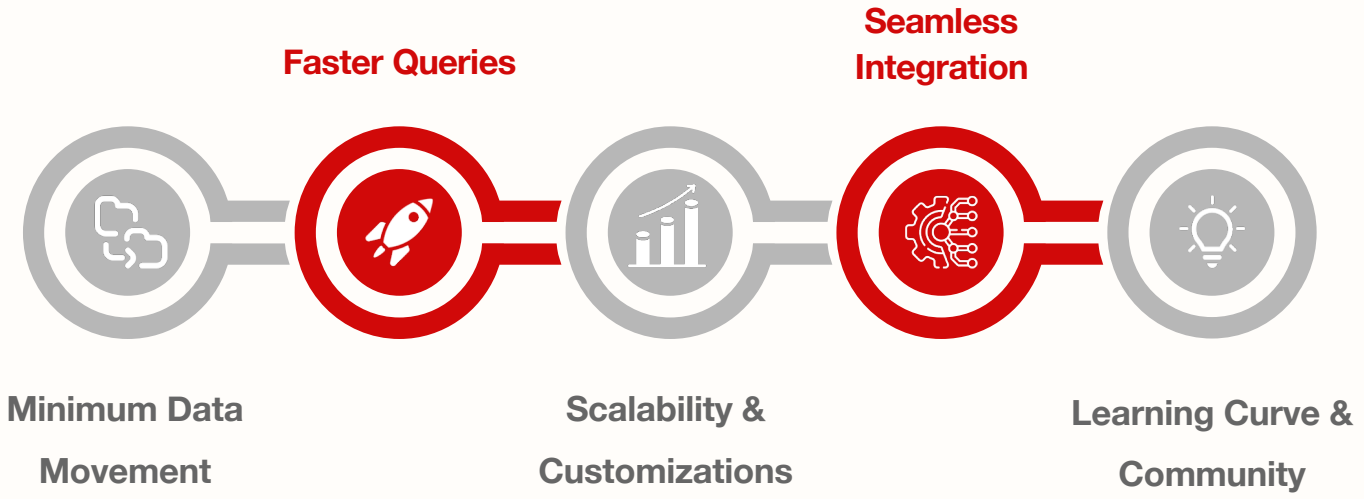
Critical data warehouse streams remained due to migration complexities.

Challenges:

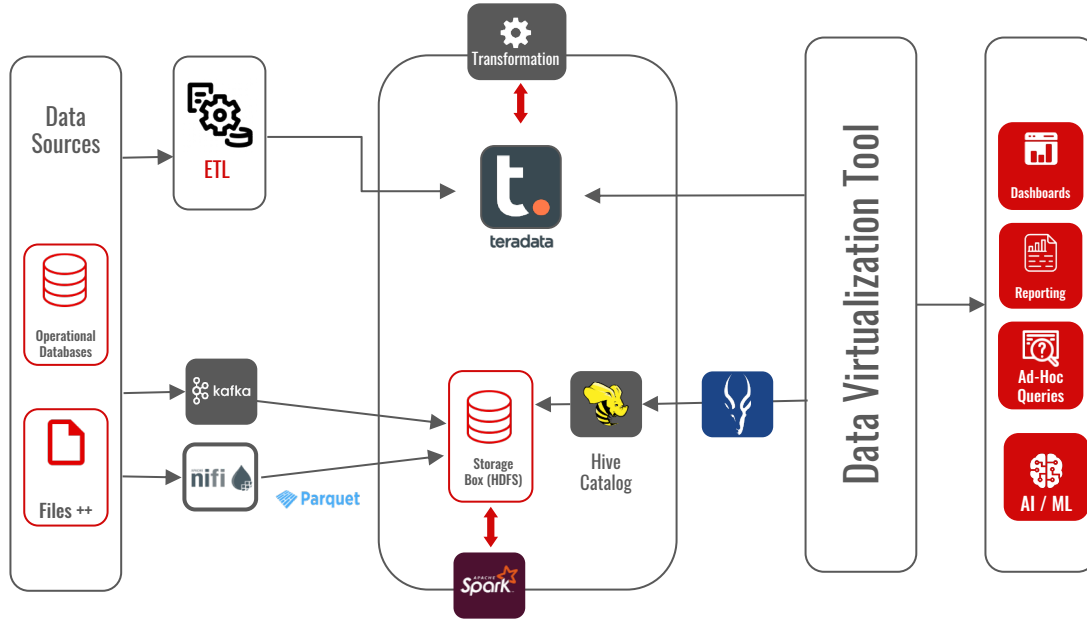
Data Silos: Complexities in managing data across multiple storage systems.

Hive, Impala performance limitations for reporting and dashboards.

Key Requirements for Data Transformation



Journey to Presto: The Data Mesh

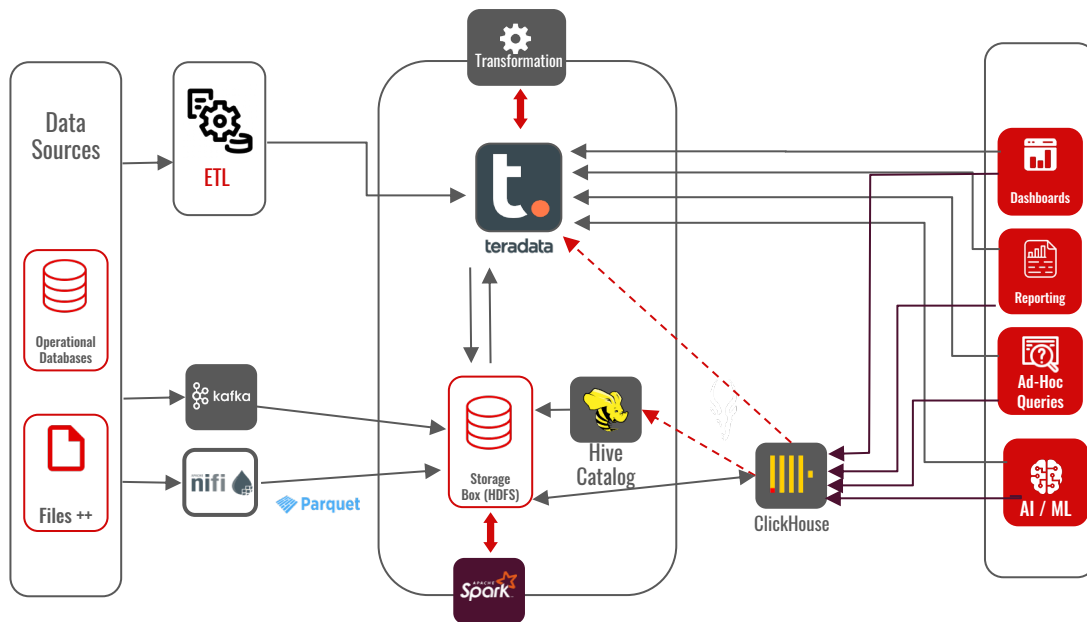


A data mesh promotes a **decentralized** approach where data is organized by specific business domains, empowering data producers to own and manage their data, and offering federated access to data, and reduces data movement.

Challenges:

Relies on Impala as a data source.
Limited connections for data sources.

Journey to Presto: Clickhouse



Clickhouse is an OLAP database and a query engine.

ClickHouse has different tables types:
Internal tables: MergeTree,
AggrigatedMergeTree

External Tables: HDFS, MergeTree,
Hive, JDBC).

Challenges:

Instability

Migration overhead.

Limited integrations.

Shared Nothing Architecture.

Journey to Presto: Clickhouse cont.

```
CREATE TABLE hdfsEngineTable on cluster my_cluster(name String, value UInt32)
ENGINE=HDFS('hdfs://hdfs1:9000/user/hive/warehouse/test.db/test_table/*', 'Parquet')
```

Node 1:

```
CREATE TABLE hdfsEngineTable_local (name String, value UInt32)
ENGINE=HDFS('hdfs://hdfs1:9000/user/hive/warehouse/test.db/test_table/part={1..6}/*', 'Parquet')
```

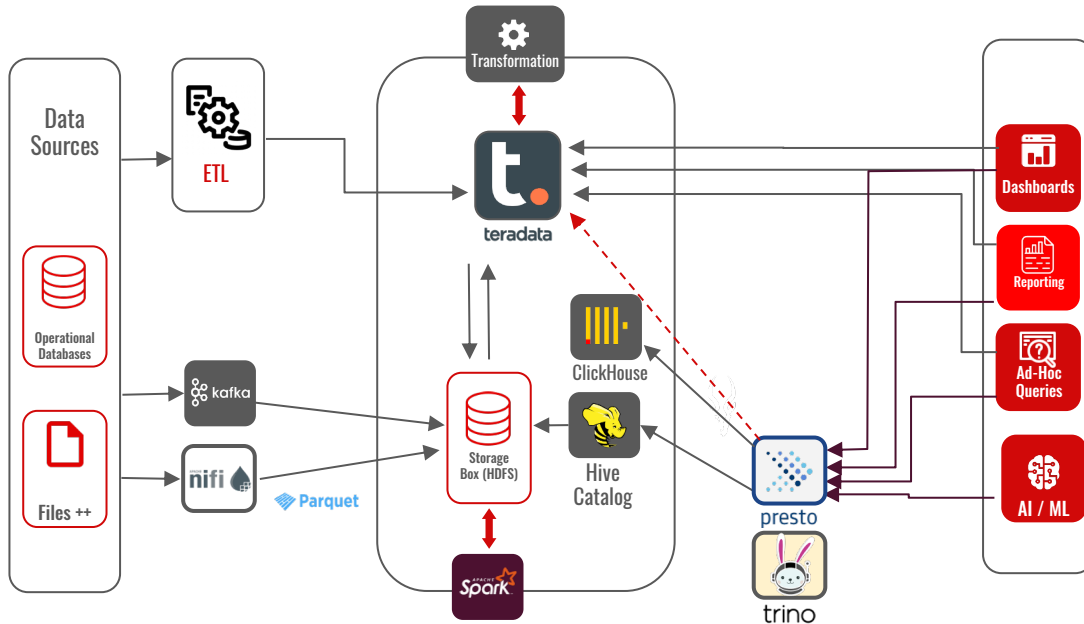
Node 2:

```
CREATE TABLE hdfsEngineTable_local (name String, value UInt32)
ENGINE=HDFS('hdfs://hdfs1:9000/user/hive/warehouse/test.db/test_table/part={7..12}/*', 'Parquet')
```

```
CREATE TABLE hdfsEngineTable on cluster my_cluster (name String, value UInt32)
ENGINE = Distributed(hdfsEngineTable_local, default_schema, hits[, sharding_key[, policy_name]])
```

“Go find another solution” **ClickHouse**

Presto: A Game Changer for Etisalat's Data Analytics



Powerful open-source distributed SQL engine

Easy to set up, and great scalability

Impressive query performance with Hive

SQL similar to Hive and Spark

Can be used in ETL

Challenges:

Lacks a connector to Teradata
Different features for each connector
Push down can't be controlled

Presto and Trino



Slightly faster than presto (Trial on Hive and Parquet).

Built on modern Java 22.

Connectors are more rich in features.

Aims to match the needs of a wide range of companies and businesses.



Prioritizes performance, memory optimization, and caching.

Currently developing C++ based workers for improved efficiency.

This comparison is based on our experience

Presto Future Work

User Interface Enhancements

- Needs improvement in UI for better usability.
- Better administration features:
 - Controlling queues
 - Rushing queries
 - Managing sessions

Technical Improvements

- Full control over pushdown.
- Ability to write source native queries (similar to table functions in Trino).
- The C++ engine.

Community and Contributions

- Creating tutorials on how to contribute or add features.

Our Plans for Presto

- Build Strong knowledge for the team
- Work on a PoC of Iceberg with presto
- Presto configurations and tuning
- Create UDFs that exist in hive and spark
- Use Presto in ETL jobs
- Work on the presto teradata connector

Suggestions?

Thank You