INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X

Impact Factor: 6.078

(Volume 10, Issue 5 - V10I5-1324)

Available online at: https://www.ijariit.com

Building a Comprehensive Enterprise Data Lake Architecture

Ramla Suhra
ramlais@gmail.com
H-E-B Digital, Texas

ABSTRACT

Organizations need to be driven by "data" more than ever to stay ahead of the curve and be competitive. With the tremendous data growth of data both by volume as well as variety, it is no longer sustainable to store the data in traditional data warehouses as they are not designed to be scalable. Data lake architecture, which is typically built on top of cheap hardware is the most economically viable solution for this problem as they are elastic and can scale up based on the increasing data needs of an organization. While the solution might seemingly look straightforward there are many nuances associated with this shift in paradigm and a very careful and thoroughly thought through design is necessary when building an enterprise data lake architecture.

This white paper explores various aspects related to setting up a comprehensive enterprise data lake which can steer towards the success of the organization. It also touches up on the pit falls and opportunities based on the research and case studies relevant in this area. Finally, a summary and outlook on data lake management is presented to the readers.

Keywords: *Artificial Intelligence, Big Data, Data Lake, Data Lake House, Heterogeneous data, Data Science, Data warehouse, Machine Learning, Heterogeneous computing.*

INTRODUCTION

Data is proliferating in an unprecedented rate in this modern era. Data sources for organization range from existing databases and backends to click streams, social media, smart devices, connected applications and internet of things (IoT) devices. Over the years companies might also expand their operations to different business zones, which potentially multiplies their data needs.
Traditionally companies used to extract subset of data from transactional/operational data sources like RBMS systems, transform them and loaded them into data warehouses. Data warehouses were used for reporting and data analytics.

Gradually the organization started realizing that the traditional data warehouse could not scale up with the data explosion rate they hit. While switching to a cheaper storage could have solved this problem intermittently, the core problem was beyond the storage needs. It was about finding a solution which can store the data, process it, analyze it in a secure and controlled manner. With the evolution of machine learning (ML) and Artificial Intelligence (AI), companies saw a new potential on intelligent decision making driven by data.
Thus, the shift to data lake architecture which encompasses all the above features became a crucial need.

DEFINITION OF A DATA LAKE

Data lake in its simplest definition is a central repository of data in various formats originating from disparate systems. It can store both structured and unstructured data.

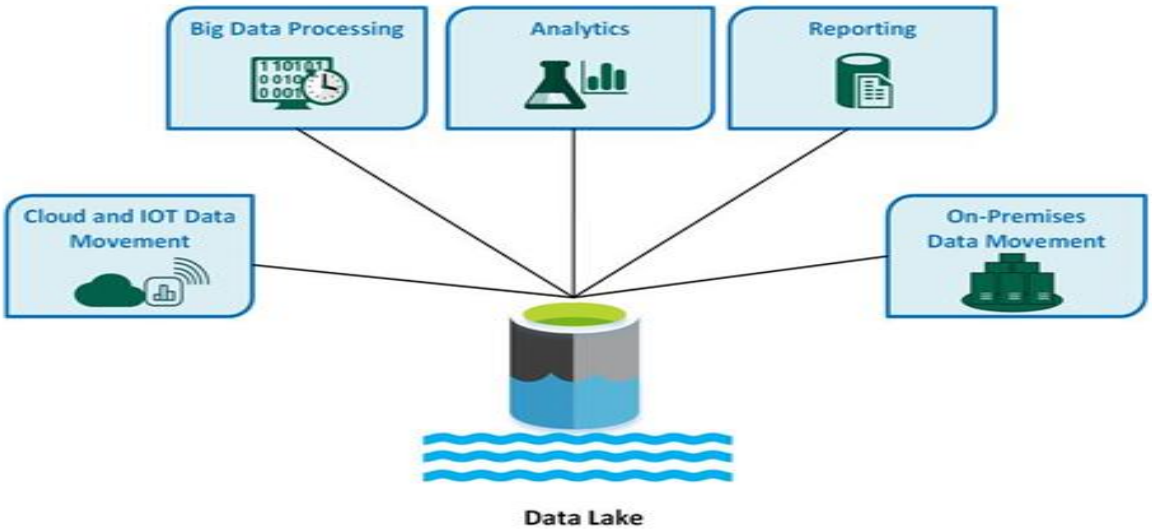

Fig. 1: Data Lakes [18]

In contrast to data warehouses, data lake uses a bottom-up approach where it ingests the data in raw or native format, processes them and analyzes them later.
Since the data is typically read in native format, it stores data with no data model associated with it, making it "schema on read".
Data lake due to the nature of its data storage is useful for users who do deep analysis like data scientists and analysts whereas data transformed from them can be served to warehouses for use by business analysts.

USE CASES

- Acts as a single source of truth by storing data in a central repository, ingested from multiple sources in the enterprise.
- Easy integration with unstructured or semi structure datasets.
- Ability to handle streaming data, for example, IoT, social media data.
- Helps with optimizing cost incurred by data warehouses for storage of large volume data and transforming them. Warehouses can focus on storing transformed data for querying.
- Storage location for archiving data from warehouses and other applications
- Greater performance with distributed computing framework and can be processed quickly.
- Helps with leveraging power of data for ML and AI use cases.
- Can act as source of truth for data warehouses.

Through this approach data can be served with low latency in a secured manner to stakeholders.

CHALLENGES

Data lake lacks ACID compliance as it is not usually a crucial requirement in Analytical systems. However, this can cause impacts like data corruption or loss, performance degradation eventually leading to data swamps if the data lake is not properly managed.
Solution:

Data Lake House

Lakehouse storage systems implement ACID transactions and other management features over data lake storage. Because of their ability to mix data lake and warehouse functionality, lake house systems are used for a wide range of workloads, often larger than those of lakes or warehouses alone. Examples are Delta Lake, Apache Hudi and Apache Iceberg. [8]. Refer Fig. 2.

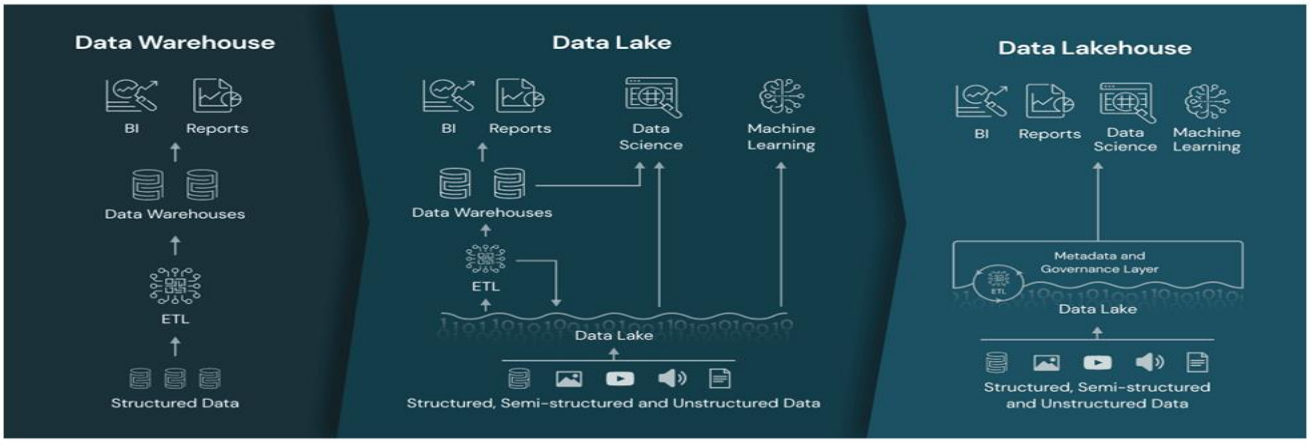

Fig. 2: Data Lakehouse [19]

DESIGNING THE DATA LAKE

A thorough analysis and design are required while building a comprehensive data lake for your enterprise. The major areas to focus on are platform, data ingestion, storage, processing, management, advanced analytics and data science.

Platform

Generally, for enterprises dealing with large data volume which requires elastic storage with highly scalable processing, cloud storage is preferred. Enterprises that are strict about security and privacy controls over the data tend to store them on-premises by setting up a private cloud like architecture.

Data Ingestion

Data ingestion is a process of moving and transferring different types of data (structured, un-structured and semi-structured) from their sources to other system for processing, this process starts with prioritizing data sources then validating information and routing data to the correct destination"[1].
This phase involves detailed research and analysis on below areas

Identify Data Sources

Identify all the data sources from which data would need to flow into the data lake. We need to make sure the data lake can support those data sources and establish connections with them with connectors or integrations.
Typical data sources range from common sources like RDBMS systems, APIs, SaaS Applications, file storages, to complex sources like social media and IoT devices.

Identify Data Ingestion Tools

Data Ingestion methods can be broadly classified into batch and real-time. Batch ingestion typically extracts data in chunks in a periodical manner. This is useful for offline analytics which are not time sensitive.
Cloud providers generally offer ingestion services as part of their data lake offerings.

Table 1 lists offerings from the major cloud providers as well as some trending tools, details extracted from [9].

Table-1: Ingestion Services

Ingestion Type	Cloud Specific			Cloud agnostic
Batch	Amazon Web Services (AWS)	Microsoft Azure	Google Cloud Platform (GCP)	
	Amazon AppFlow, Amazon Data Pipeline, AWS Glue	Azure Data Factory	Cloud Data Fusion	Five Tran, Databricks
Real-time	Amazon Kinesis Data Streams	Azure Event Hubs	Cloud Pub/Sub	Kafka

Although cloud providers have offerings for data transfer into data lake, enterprises can also review tools in the market built for this cause. This would allow them to be not locked into a vendor and are more flexible to port.

Data Storage

As the literature "Operationalizing the Data Lake By Holden Ackerman, Jon King"[10] suggests, one of your first considerations in building your data lake will be storage. A suggested approach from the literature is as below:
There are three basic types of data storage: immutable raw storage, optimized storage, and scratch databases.

Immutable Raw Storage Bucket

Data kept in immutable storage cannot, and should not, be changed after it has been written. In an immutable raw storage area in your data lake, you store data that hasn't been scrubbed. You might never have even looked at it. But it should have sufficient self-descriptive language, or metadata, around it—such as table names and column names—so that you can determine where the data came from.

Optimized Storage Bucket

As your raw data grows, your queries into it become slower. No one likes waiting hours to see whether their query succeeds, only to find that it failed. Data scientists and analysts need their questions answered to turn data into insights faster than that. To gain this speed, transforming your data by storing it using one of the many optimized formats available. Widely used open-source choices are Delta, Parquet, optimized row column (ORC), and Avro.

Scratch Database

Finally, you will usually create what are called user scratch databases. These are necessary because data scientists and analysts will want to take data out of the optimized schema and build test tables for their own purposes. But because you don't want anyone to inadvertently mess up your data lake—or turn it into a data swamp—you need a place where users can have their own little sandboxes to play in that won't mess up the clean, well-defined, and well-structured data in the optimal data space".
An alternate approach widely used is the "Medallion Architecture".[3]

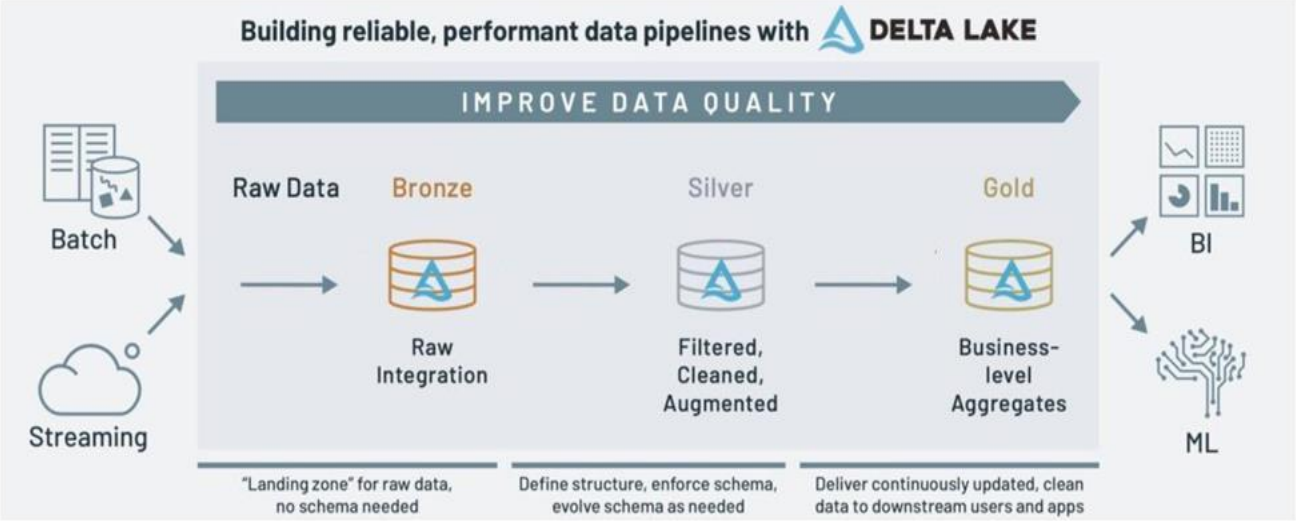

Fig. 3: Delta Lake Medallion Architecture, Databricks [3]

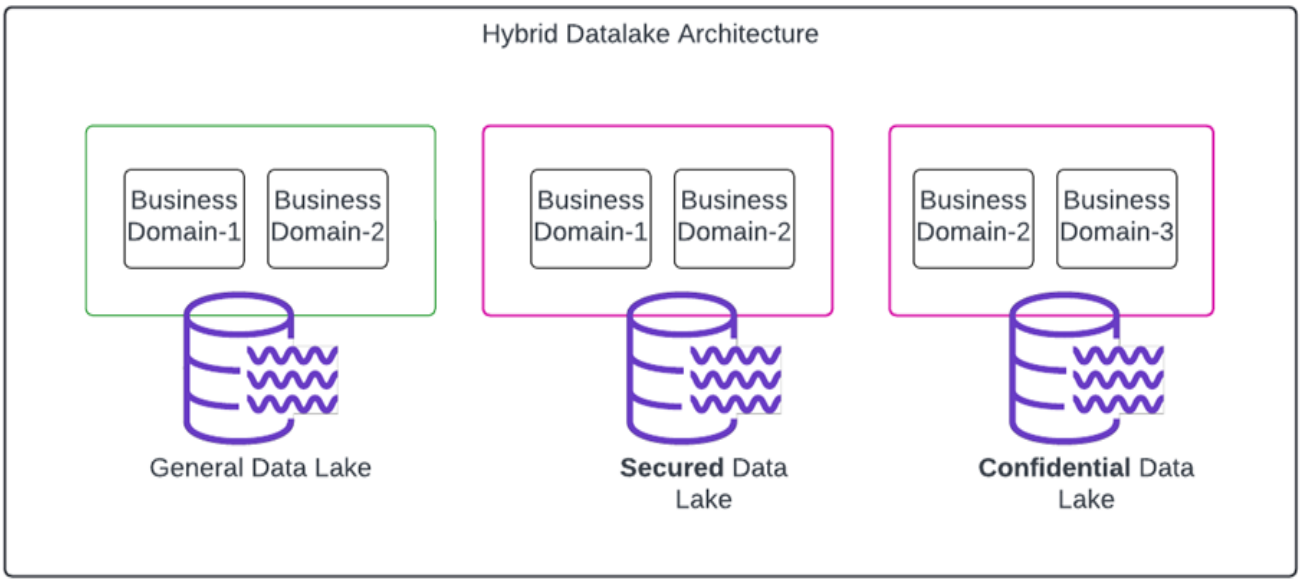

Fig. 4: Hybrid Data Lake Architecture

A medallion architecture is a data design pattern, coined by Databricks, used to logically organize data in a lakehouse, with the goal of incrementally improving the quality of data as it flows through various layers. This architecture consists of three distinct layers – bronze (raw), silver (validated) and gold (enriched) – each representing progressively higher levels of quality. Medallion architecture is sometimes referred to as "multi-hop" architectures.
While these are some prescriptive formats, enterprises often tailor them to suit their organization structure:

Sample Case Study:
A leading retail company designed their domain driven data lake structure by following a hybrid storage distribution structure. Based on sensitivity of the data elements, data lake was split into multiple accounts. Each domain data is stored in separate object storages so that they can be controlled and governed separately. Each domain owner gets full access to their domain object storage. They can also request access to other domain datasets.

Data Processing
According to [2] there are four critical requirements for big data processing.

The **first** requirement is fast data loading. Since the disk and network traffic interferes with the query executions during data loading, it is necessary to reduce the data loading time.
The **second** requirement is fast query processing. To satisfy the requirements of heavy workloads and real-time requests, many queries are response-time critical. Thus, the data placement structure must be capable of retaining high query processing speeds as the amounts of queries rapidly increase.

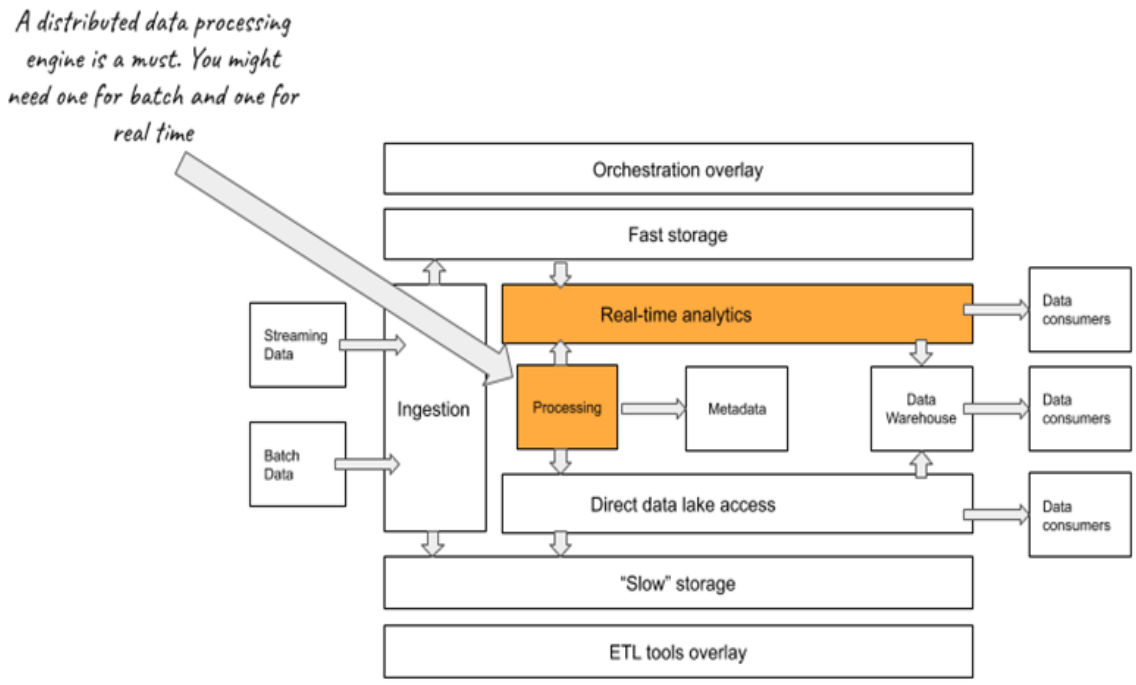

Fig. 5: Layers of Cloud Data Platform [16]

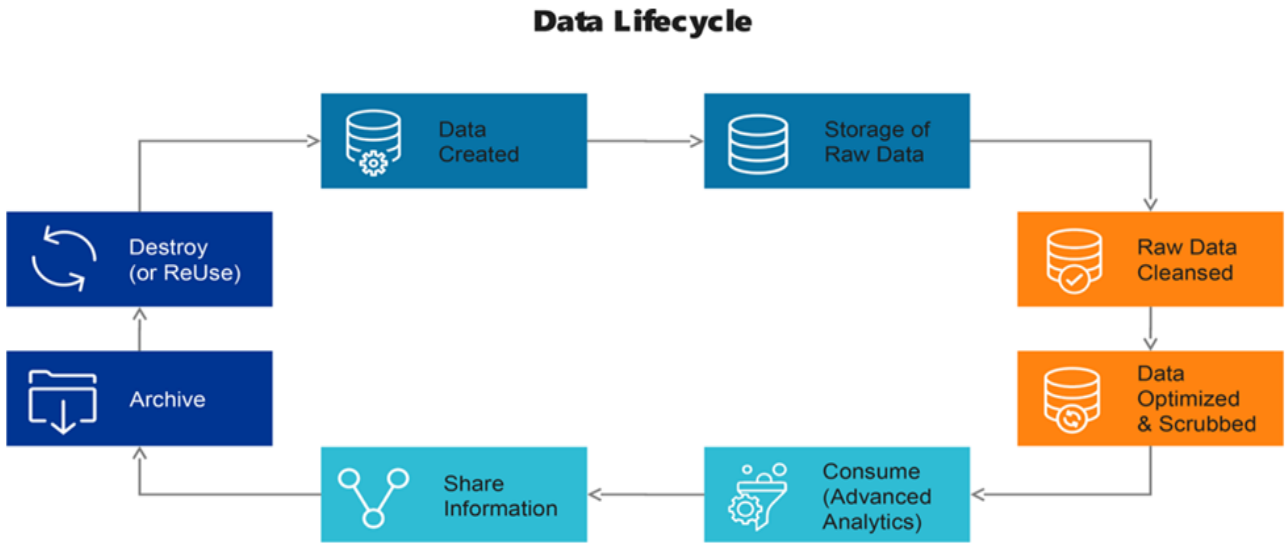

Fig. 6: Data Lifecycle [12]

Additionally, the third requirement for big data processing is the highly efficient utilization of storage space. Since the rapid growth in user activities can demand scalable storage capacity and computing power, limited disk space necessitates that data storage be well managed during processing, and issues on how to store the data so that space utilization is maximized be addressed.
Finally, the fourth requirement is the strong adaptivity to highly dynamic workload patterns. As big data sets are analyzed by different applications and users, for different purposes, and in various ways, the underlying system should be highly adaptive to unexpected dynamics in data processing, and not specific to certain workload patterns.
Distributed computational frameworks are the solution for these above-mentioned requirements. The pioneer frameworks like MapReduce[4] supported batch processing.

Later Apache Spark was introduced which drastically improved the processing speed. This model is based on a data-sharing abstraction called Resilient Distributed Datasets and is used to perform in-memory computation for workloads in SQL, streaming, machine learning and graph processing. [4][5][6]

Apache Spark is now the most popular engine for distributed data processing at scale used by large number of organizations to support their big data processing and analytics use cases. As organizations invest more and more on newer like AI and ML technologies, we anticipate that Spark will continue to play a big role in the modern data analytics stack [11].

Today, there are open-source frameworks and cloud services which allow you to process data both in a batch mode as well as real-time. Enterprises can choose to build vs buy such platforms based on their operational flexibility and investment.

Data Management and Governance
An especially important aspect of your architectural plans is a good data-management strategy that includes data governance and metadata, and how you will capture that. This is critical if you want to build a managed and governed data lake instead of the much-maligned "data swamp. [13]

Life cycle management
An important management task is to move data between storage types over the course of its life cycle. Cloud providers offer life cycle management options which are rules based on which data can be moved from frequently accessed to less frequently accessed stages [12].

Disaster Recovery
Another important area to analyze is the disaster recovery strategy. You need to set your organization's Recovery Point Objective (RPO) as well Recovery Time Objective (RTO) during the system design phase so that the system build is aligned with those objectives.

Infrastructure Management
While data management is critical, infrastructure management is also equally critical. In modern times Infrastructure as code is used for any infrastructure deployment and proper infrastructure as code management will help with faster recovery during any disasters. It also ensures that the infrastructure being rolled out is peer reviewed and proper data lake governance principles are applied while rolling out the data lake.

Operations Plan
For every stakeholder in the system, there will be different levels of service agreements that they are concerned about. While data engineers will be concerned about the timely processing and availability, data analysts will be concerned about the reliability of the data. The data platform team on the other hand will be focused on data accessibility and management and cost control. A proper operation plan is hence crucial while building a data lake. \

Data Security Plan
In its annual Cloud Threat Report, Unit 42™ [14] analyzed workloads in 210,000 cloud accounts across 1,300 different organizations and found:
Threat actors are getting smarter and more powerful every day.They're learning from new security strategies and finding creative ways to work around them, exploiting hidden weak spots and using vulnerabilities to their advantage.
MFA is not enforced for cloud users.76% of organizations don't enforce MFA for console users, and 58% of organizations don't enforce MFA for root/admin users.

Attacks on software supply chains are on the rise.The prevalence of open-source usage and the complexity of software dependency make securing the software supply chain difficult.
Unpatched vulnerabilities continue to be low-hanging fruit for attacks.63% of the codebases in production have unpatched vulnerabilities rated high or critical (CVSS >= 7.0), and 11% of the hosts exposed in public clouds have high or critical vulnerabilities.[14]

Thus, below key areas should be mandated on the data lake.

1. Secure login and access to data lake
2. Encryption of data at rest and in transit
3. Data access controls in place based on roles (RBAC)
4. Data vulnerability scans run regularly.
5. Thorough review of architecture with digital security teams within the organization

Data Governance
A data governance team needs to be formed, and processes need to be put in place even before opening data lake to users. While Data governance has a direct impact on security and compliance, it also affects user productivity and overall operational cost.
There are various subcategories under governance which need deeper review.

Quality Control
Nearly 60% of organizations don't measure the annual financial cost of poor-quality data, according to the Gartner survey. "Failing to measure this impact results in reactive responses to data quality issues, missed business growth opportunities, increased risks and lower ROI," Selvage says.[15]
Data Profiling tools can be rolled out on the data lake to help with quality checks. Quality deviations need to be reported and corrected with utmost priority.

Data Security and compliance
Ensure that data policies are formed and documented. Access requests to any data set is always reviewed and approved. Data lineage is tracked for transparency. Regulatory compliance is always monitored via automation and alerted to stakeholders in the event of anomaly detection.

Data security in data lake should have reviewed and planned for addressing below areas.

- Customer data
- Data encryption
- User management
- Infrastructure identity and access management
- Definitions of users' roles and responsibilities (perhaps using personas)

Financial Governance
Frequently monitor the spend within data lake and look out for opportunities for optimization. Upgrade the tech stack as regularly as possible to take advantage of the latest advancements in the industry.

Here are a few questions to ask yourself regularly when managing analytics in a cloud data lake:

- Are your projects delivering ROIs that push key performance indicators in a positive direction?
- Are you spending well? Are you getting your resources for the best price possible?
- Are you keeping in mind that sometimes it's not about spending less, it's about spending smarter?

Some examples of the optimization techniques are

- Efficient storage strategy
- Monitoring auto scaling and resource utilization on cloud
- Setting up compute policies to limit usage of unwanted resources.
- Anomaly detection and alerting

Advanced Analytics, Machine Learning and AI
Leverage intelligent data platforms like Databricks for this use case as they bring in a suite of required tools. While the tool suites can be availed from SaaS offerings, the key responsibility of maintaining data quality remains with the data teams and governance. This has a great impact on model bias and efficiency [7].
As Fig. 7 shows, data plays the key role in this implementation and setting up data quality and access is critical for the success of ML/AI initiatives.

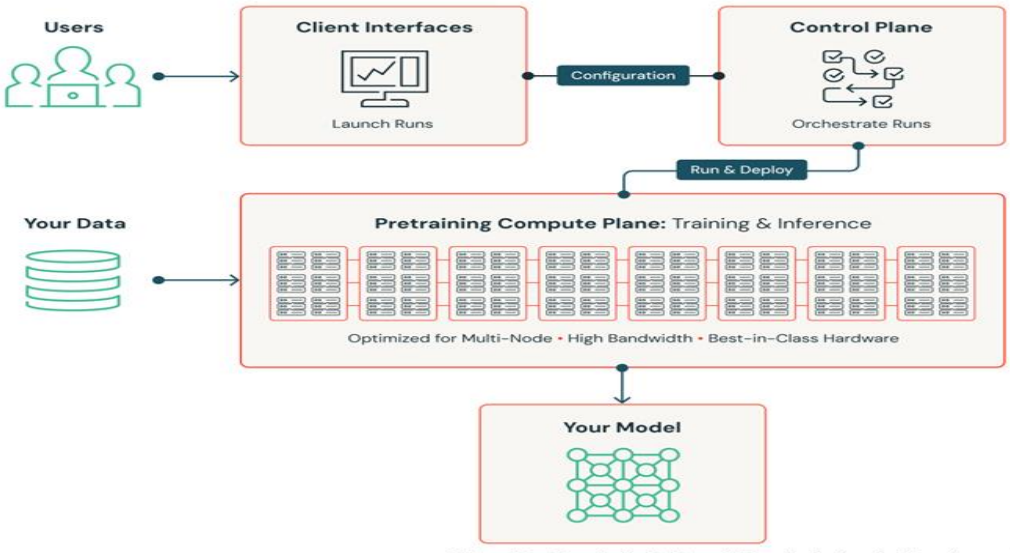

Fig. 7: Databricks Model Architecture [17]

CONCLUSION

We have explored the design of a data lake architecture and factors to consider during its design and implementation. Although there are different ways in which data lake can be set up, we have restricted ourselves to follow data lake architecture pattern. Other patterns like data mesh for instance will be another leap forward and can be reviewed in future after the organization has reached a certain level of maturity.
Thus, we conclude that building a cloud native data lake by following the above aspects can help the organizations succeed in solving the toughest problems faced by the business.

REFERENCES

[1] MATACUTA, A., & POPA, C. (2018). Big Data Analytics: Analysis of Features and Performance of Big Data Ingestion Tools. Informatica Economica, 22(2/2018), 25–34.
[2] He, Y., Lee, R., Huai, Y., Shao, Z., Jain, N., Zhang, X., Xu, Z.: RCFile: A Fast and Space-
efficient Data Placement Structure in MapReduce-based Warehouse Systems. In: IEEE International Conference on Data Engineering (ICDE), pp. 1199–1208 (2011)
[3] "Medallion Architecture - Data Engineering Wiki," Dataengineering.wiki,2024.https://dataenginee ring.wiki/Concepts/Medallion+Architecture (accessed Oct. 16, 2024).
[4] Dean, J. and Ghemawat, S. MapReduce: Simplified data processing on large clusters. In Proceedings of the Sixth OSDI Symposium on Operating Systems Design and Implementation (San Francisco, CA, Dec. 6–8). USENIX Association, Berkeley, CA, 2004.
[5] M. Zaharia et al., "Apache Spark," Communications of the ACM, vol. 59, no. 11, pp. 56–65, Oct. 2016, doi: 10.1145/2934664.
[6] Zaharia, M. et al. Discretized streams: Fault-tolerant streaming computation at scale. In Proceedings of the 24th ACM SOSP Symposium on Operating Systems Principles (Farmington, PA, Nov. 3–6). ACM Press, New York, 2013.
[7] M. Zaharia et al., "Accelerating the Machine Learning Lifecycle with MLflow," 2018. https://www.semanticscholar.org/paper/Accelerating-the-Machine-Learning-Lifecycle-with-Zaharia-
Chen/b2e0b79e6f180af2e0e559f2b1faba66b2bd578a
[8] P. Jain, P. Kraft, C. Power, T. Das, I. Stoica, and M. Zaharia, "Analyzing and comparing lakehouse storage systems,"2023. https://www.semanticscholar.org/paper/
Analyzing-and-Comparing-Lakehouse-Storage-Systems
-Jain-Kraft/52a02a99c5bb79a7e4ebddd8fa9867e22
aff6c97
[9] "Compare AWS and Azure services to Google Cloud," Google Cloud, 2024. https://cloud.google.com/docs/get-started/aws-azure-gcp-service-comparison (accessed Oct. 16, 2024).
[10] H. Ackerman and J. King, "Operationalizing the data lake," O'Reilly Online Learning. https://www.oreilly.com/library/view/operationalizing-the-data/9781492049517/titlepage01.html
[11] D. Armlin, "Inside the Modern Data Analytics Stack," Chaossearch.io, Apr. 25, 2024. https://www.chaossearch.io/blog/modern-data-analytics-stack (accessed Oct. 16, 2024).
[12] H. Ackerman and J. King, "Operationalizing the data lake," O'Reilly Online Learning. https://www.oreilly.com/library/view/operationalizing
-the-data/9781492049517/ch05.html#data_governance_
chap_five
[13] B. Sharma, "Architecting Data Lakes, 2nd Edition," O'Reilly Online Learning. https://www.oreilly.com/library/view/architecting-data-lakes/9781492033004/ch05.html#security_strategy
[14] "Unit 42 Cloud Threat Report, Volume 7: Navigating the Expanding Attack Surface," Paloaltonetworks.com, 2021. https://start.paloaltonetworks.com/unit-42-cloud-threat-report-volume-7 (accessed Oct. 16, 2024).
[15] S. Moore, "4 Steps to Overcome Data Quality Challenges," Gartner, Jan. 18, 2018. https://www.gartner.com/smarterwithgartner/how-to-stop-data-quality-undermining-your-business (accessed Oct. 16, 2024).
[16] "The Layers of a Cloud Data Platform | Manning," Manning.com, 2020. https://freecontent.manning.com/the-layers-of-a-cloud-data-platform/ (accessed Oct. 16, 2024).
[17] "Mosaic AI | Databricks," Databricks, 2022. https://www.databricks.com/product/machine-learning (accessed Oct. 16, 2024).
[18] PRASADA1207, "What is a data lake? - Azure Architecture Center," Microsoft.com, Aug. 26, 2024. https://learn.microsoft.com/en-us/azure/architecture/
data-guide/scenarios/data-lake (accessed Oct. 16, 2024).
[19] B. Lorica , M. Armbrust, R. Xin, M. Zaharia, and A. Ghodsi, "What Is a Lakehouse?," Databricks, Jan. 30, 2020. https://www.databricks.com/blog/2020/
01/30/what-is-a-data-lakehouse.html (accessed Oct. 16, 2024).

