databricks

The Well-Architected Lakehouse

Leo Mao

Sr. Specialist Solutions Architect

Housekeeping

- Your connection will be muted
- We will share recording with all attendees after the session
- Submit questions in the Q&A panel
- If we do not answer your question during the event, we will follow up with you to get you the information you need!

You will learn

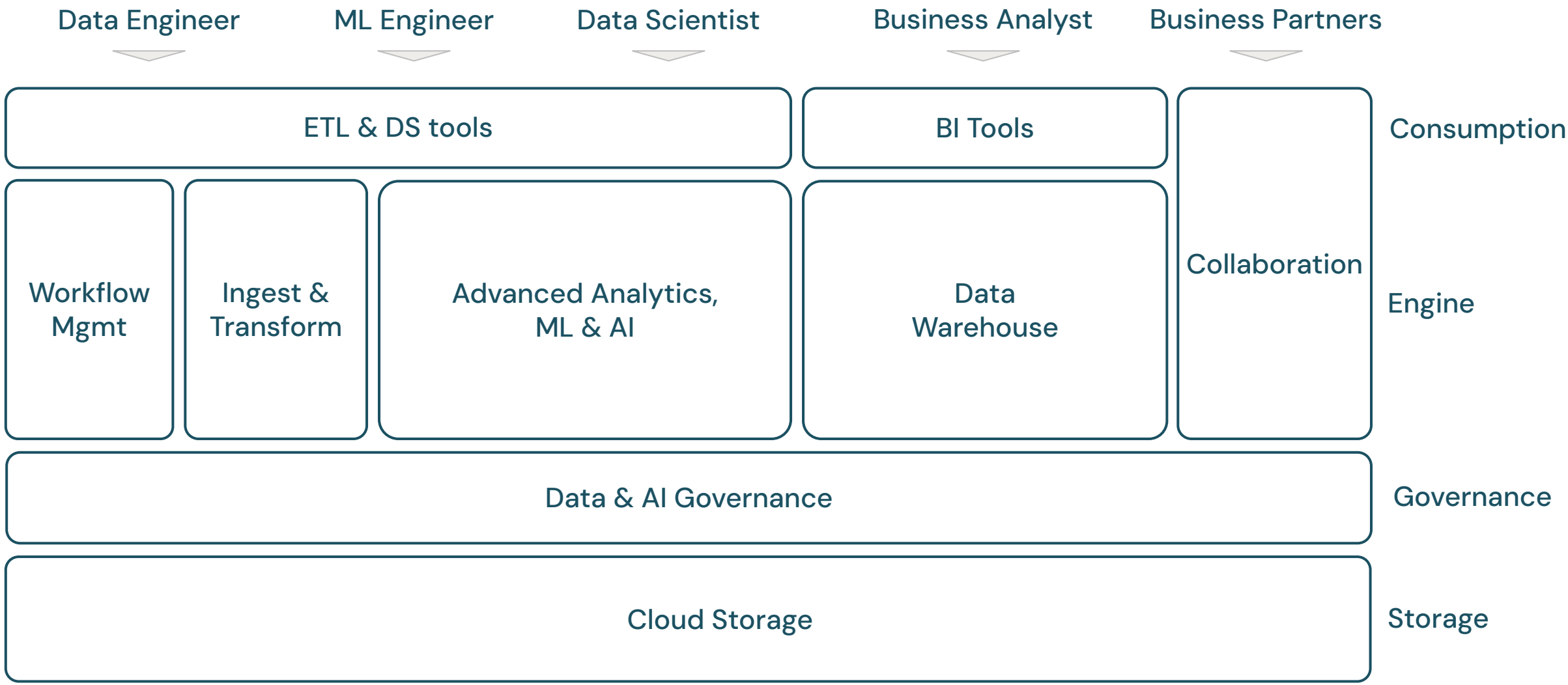
What is the Well Architected Framework and Why

Well Architected Lakehouse and Guiding Principles

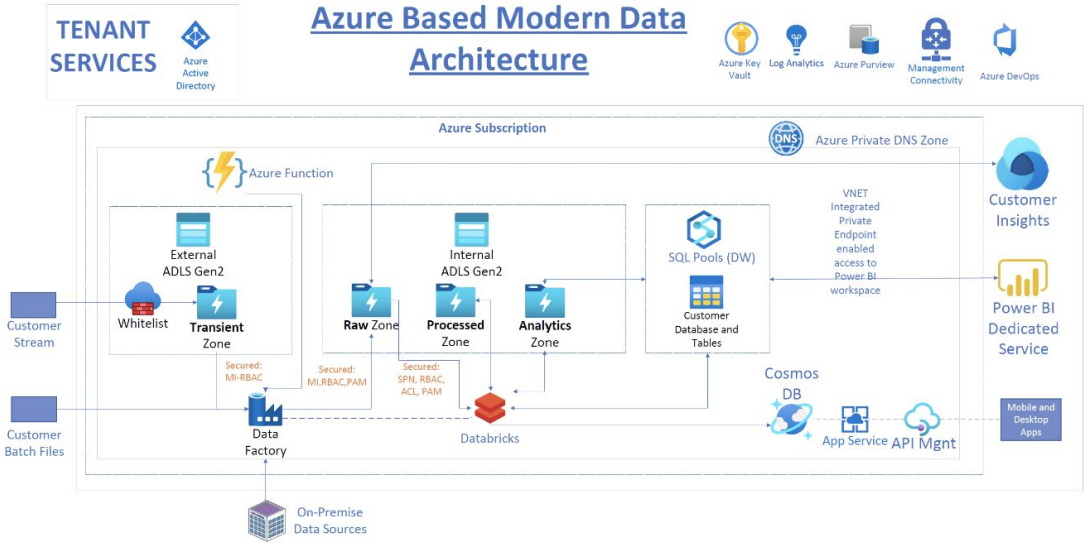
How Databicks help customers to implement a “good” Lakehouse?

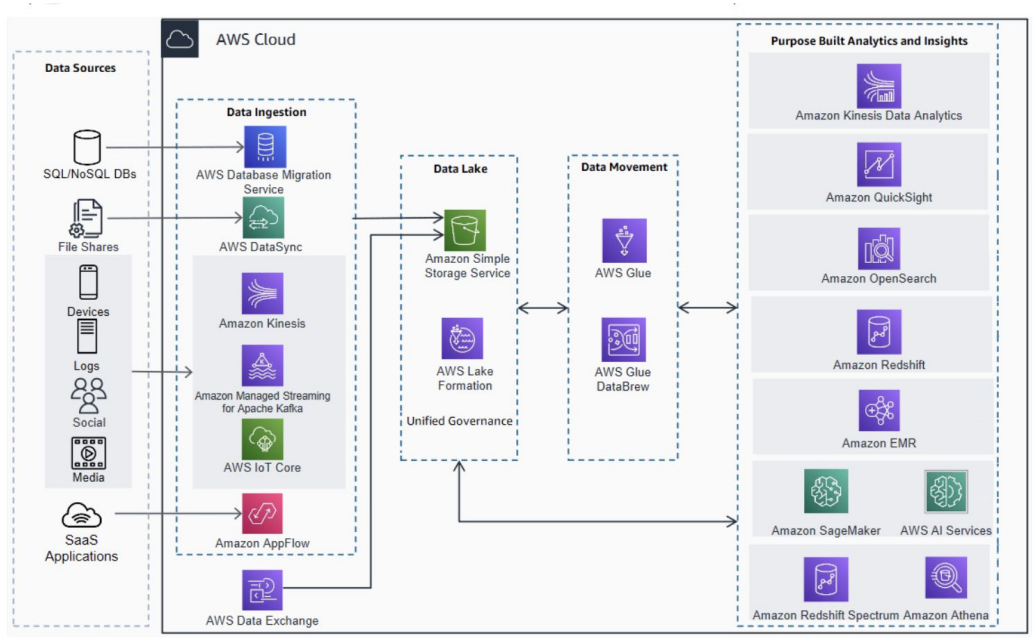
What is the Well Architected Framework & Why

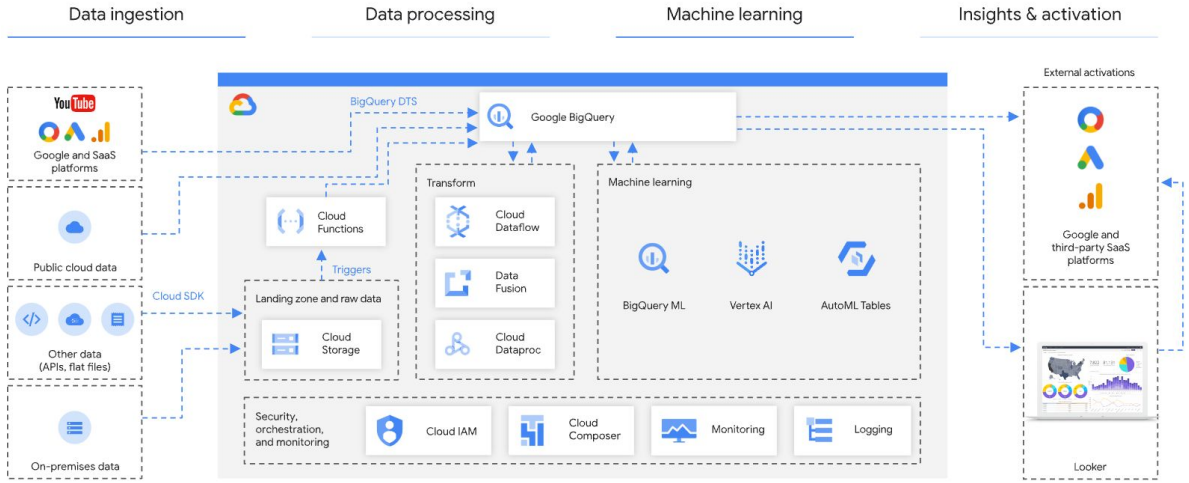
Modern Data Stack - Capability



Cloud Data Architecture – Examples

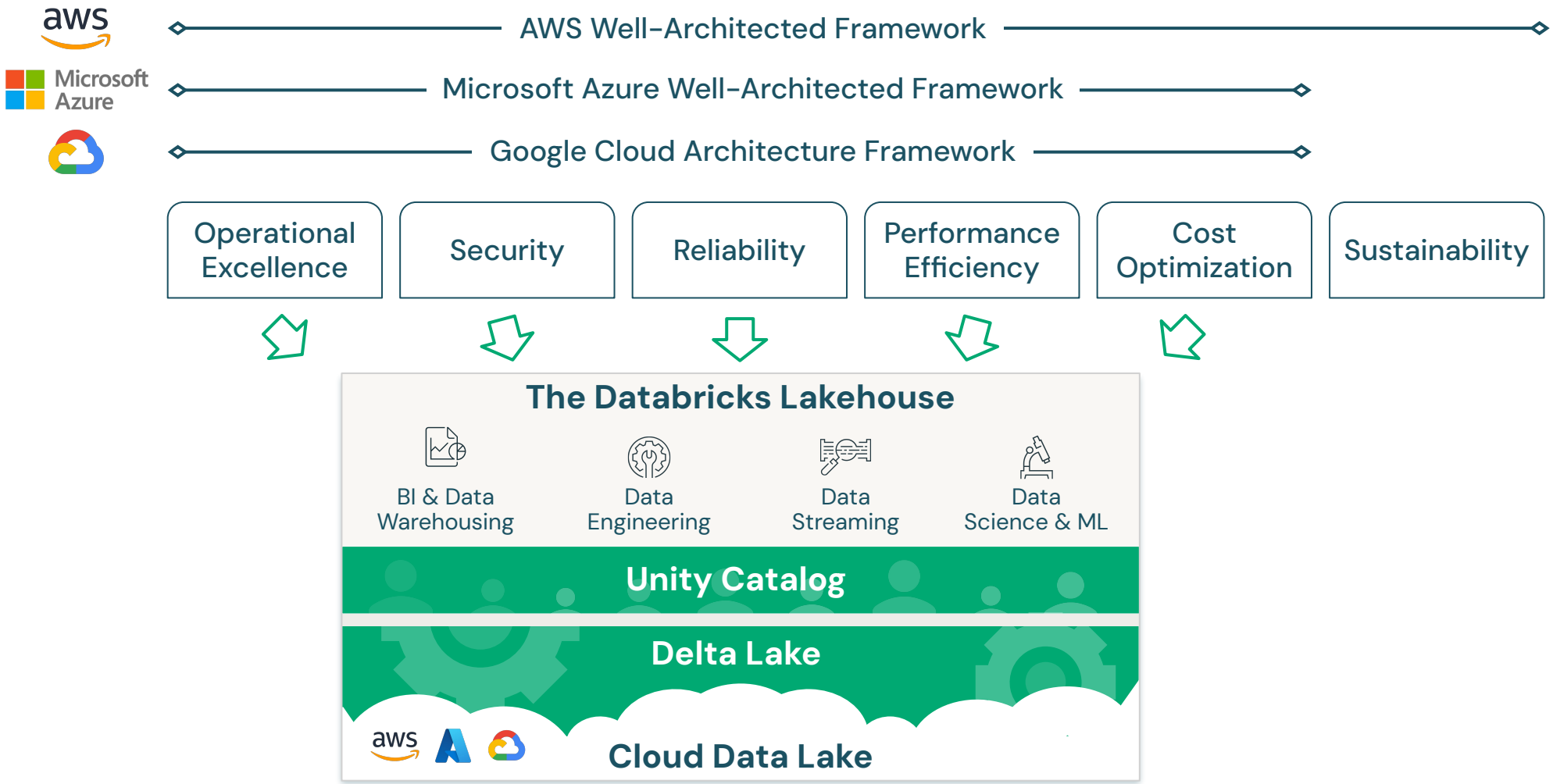




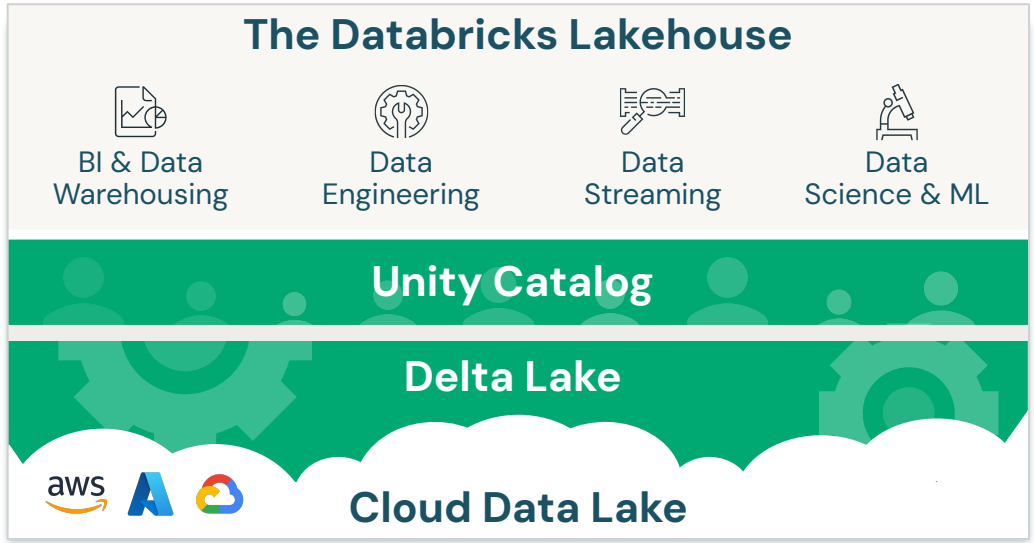


The Well-Architected Lakehouse

Extends the Cloud Well-Architected Frameworks to the Lakehouse



The Well-Architected Lakehouse



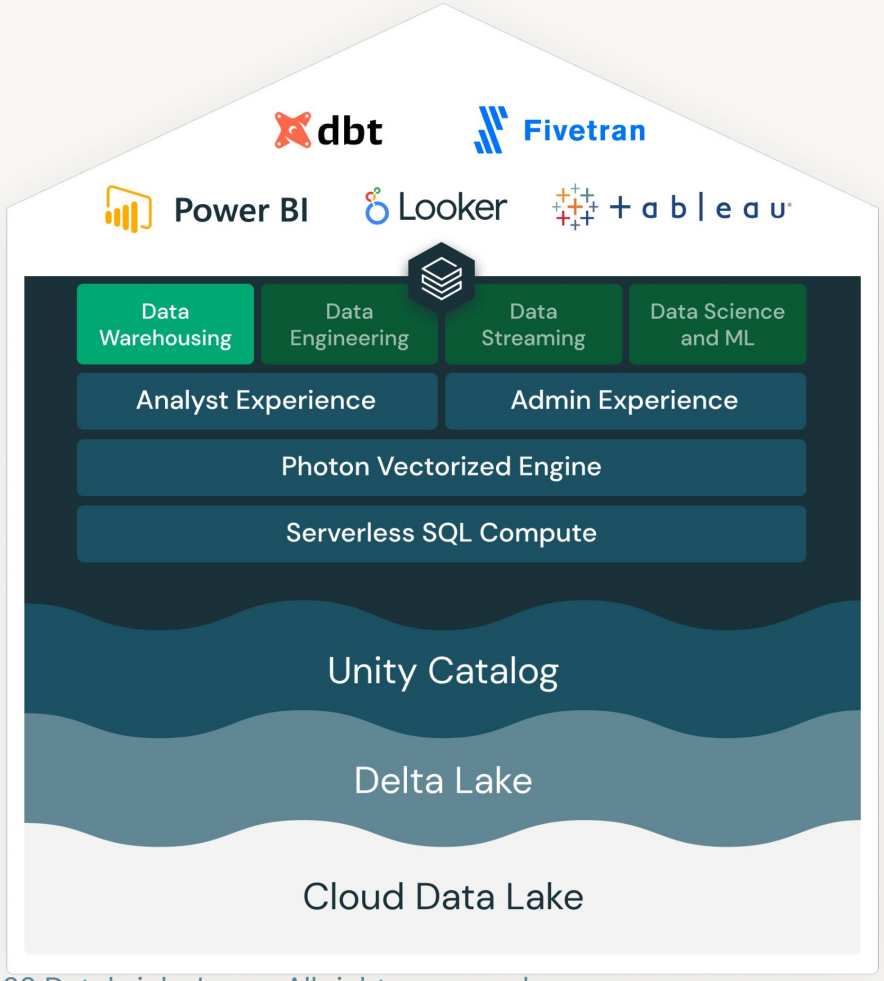
Lakehouse specific

Pillars

Data Governance	Interoperability and Usability	Operational Excellence	Security, Privacy, Compliance	Reliability	Performance Efficiency	Cost Optimization
Scope						
The oversight to ensure that data brings value and supports your business strategy.	The ability of the Lakehouse to interact with users and other systems	All operations processes that keep the Lakehouse running in production	Protect Databricks application, customer workloads and customer data from threats	The ability of a system to recover from failures and continue to function	The ability of a system to adapt to changes in load	Managing costs to maximize the value delivered

Lakehouse for your data platform

Powered by Databricks



Seamless Integration with the Ecosystem

Ease of Use

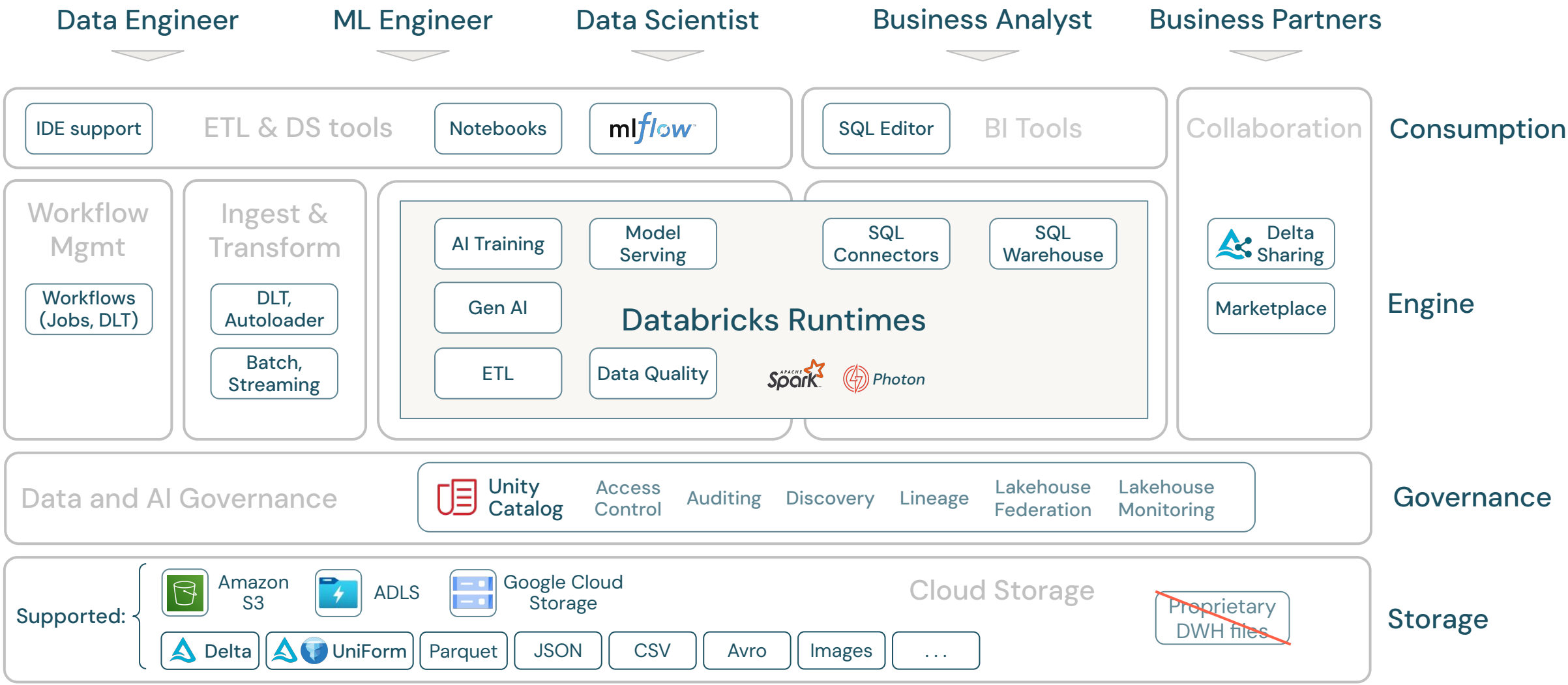
Real-world Performance

Centralized Governance

Open and Reliable Data Lake as the Foundation

Well Architected Lakehouse & Guiding Principles

Modern Data Stack – Lakehouse



Guiding Principles for the Lakehouse

Curate Data and Offer Trusted Data-as-Products

Adopt an Organization-wide Data Governance Strategy

Remove Data Silos and Minimize Data Movement

Encourage the Use of Open Interfaces and Open Formats

Democratize Value Creation through Self-Service Experience

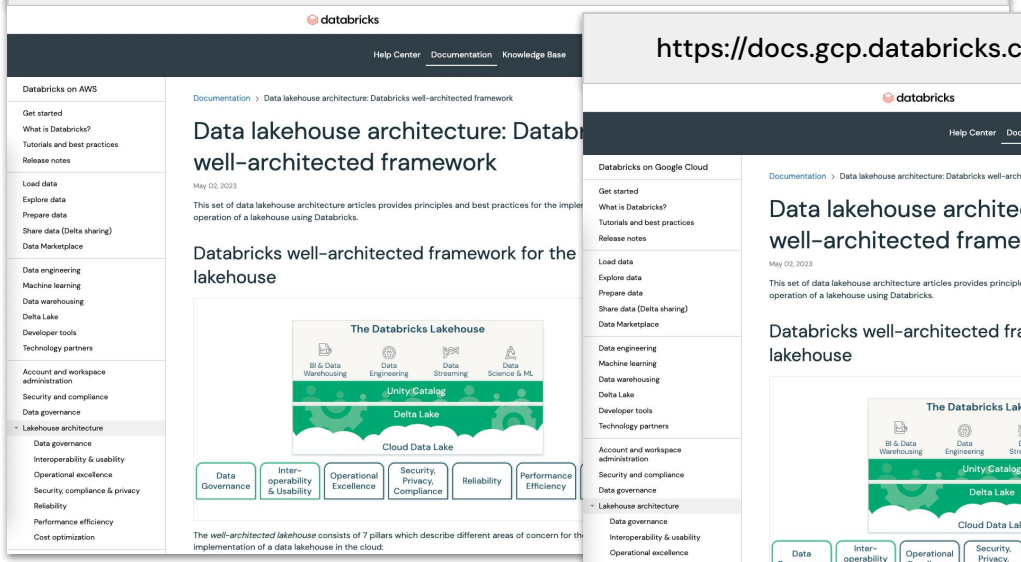
Build to Scale and Optimize for Performance & Cost

Well documented Principles and Best Practices

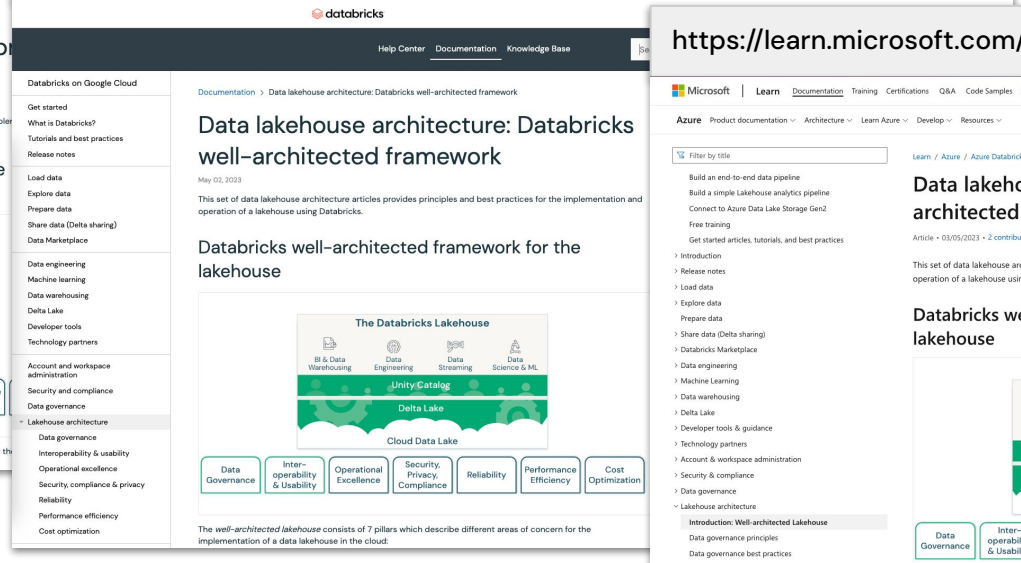
Data Governance | Interoperability and Usability | Operational Excellence | Security, Privacy, Compliance | Reliability | Performance Efficiency | Cost Optimization

Principles and Best Practices

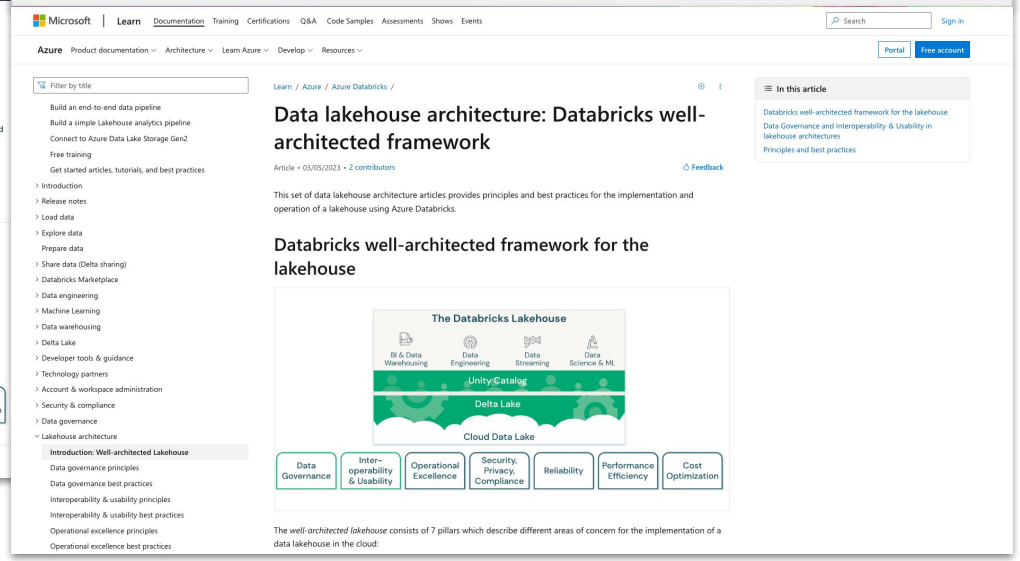
https://docs.databricks.com/lakehouse-architecture/index.html



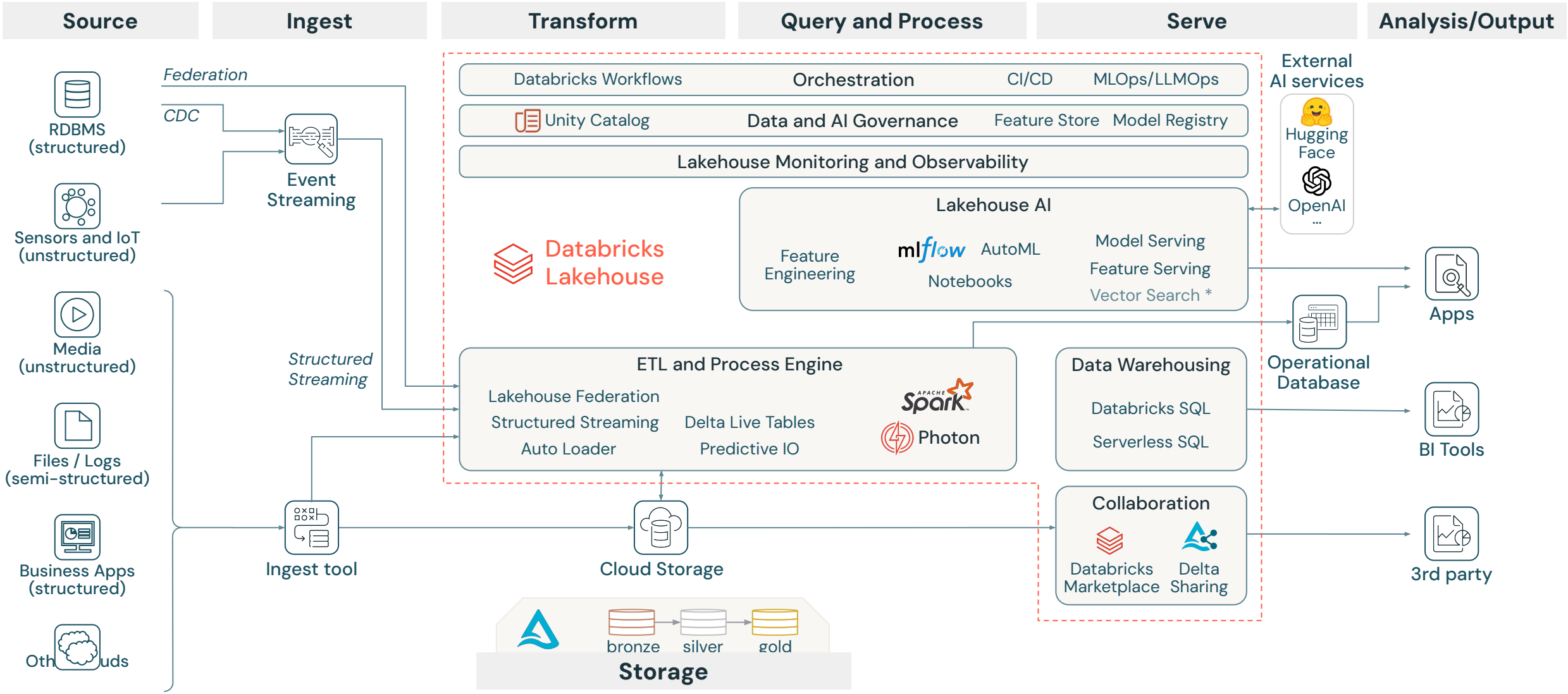
https://docs.gcp.databricks.com/lakehouse-architecture/index.html



https://learn.microsoft.com/en-gb/azure/databricks/lakehouse-architecture



Lakehouse Reference Architecture – Cloud Agnostic



* Private Preview

How Databricks help customers to implement a “good” Lakehouse

Grouped best practices

Data Governance

- Unify Data Management
- Unify Data Security
- Manage Data Quality
- Securely Share Data

Interoperability & Usability

- Integration standards
- Open formats & interfaces
- Democratize Data
- Consistency & usability

Operational Excellence

- Processes
- Automation
- Monitoring

Security, Compliance and Privacy

- Identity & Privileges
- Data security
- Network security
- Compliance and Privacy
- Security Monitoring

Performance efficiency

- Design for performance
- Performance testing
- Serverless services
- Performance monitoring

Cost Optimization

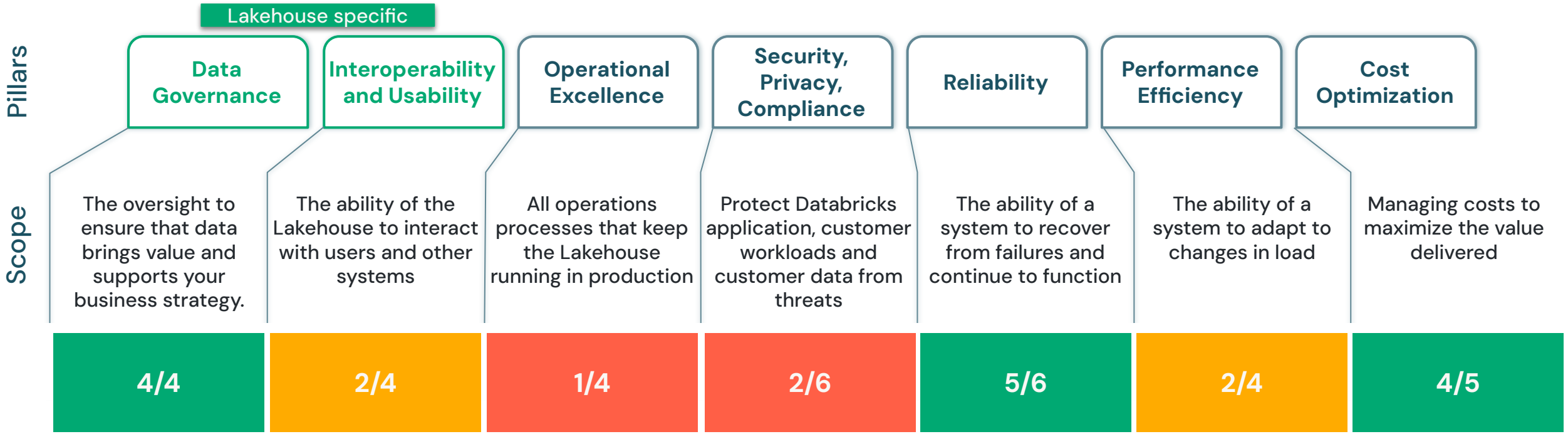
- Resource allocation
- Workload optimization
- Cost attribution
- Cost monitoring

Reliability

- Design for failure
- Data quality
- Scaling
- Automation
- Monitoring

Assessment Summary

High level summary

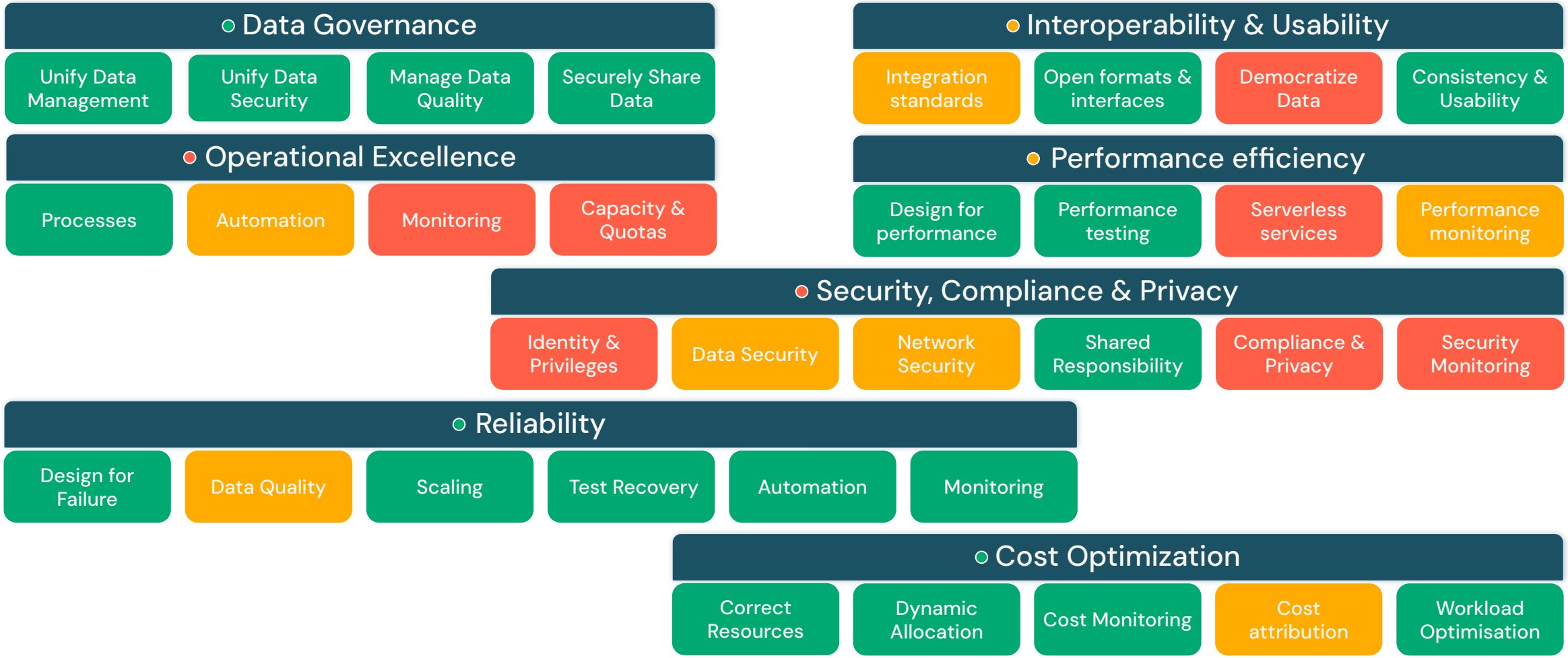


Colour Legend

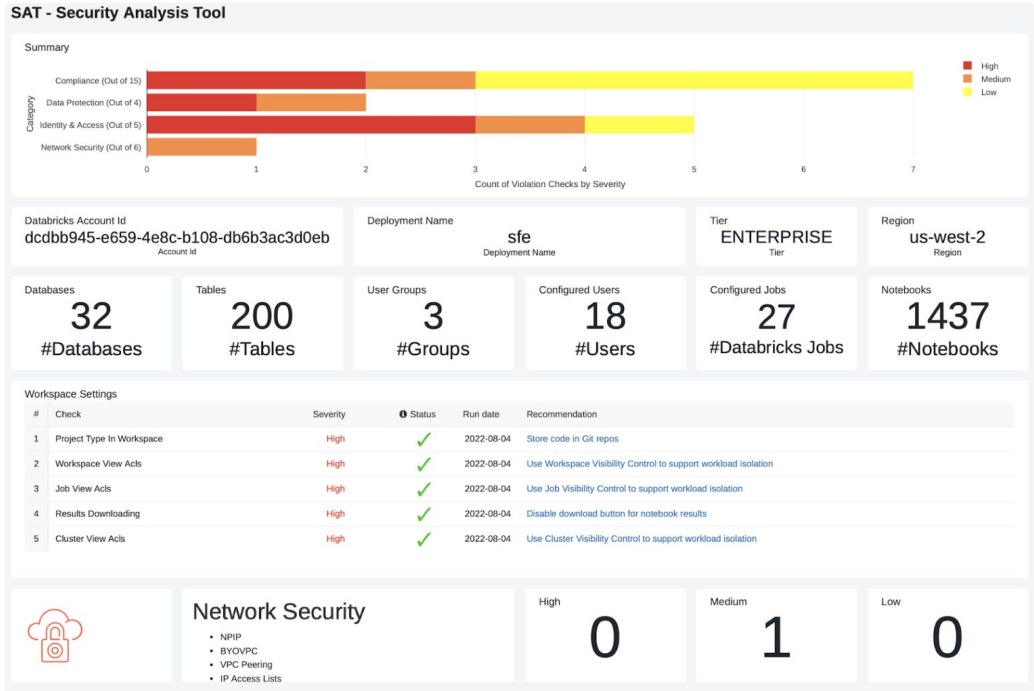
- = Well Architected, following most or all Best Practices (> 80%)
- = Some Best Practices followed(> 40%)
- = Large Gaps in Best Practices (< 40%)

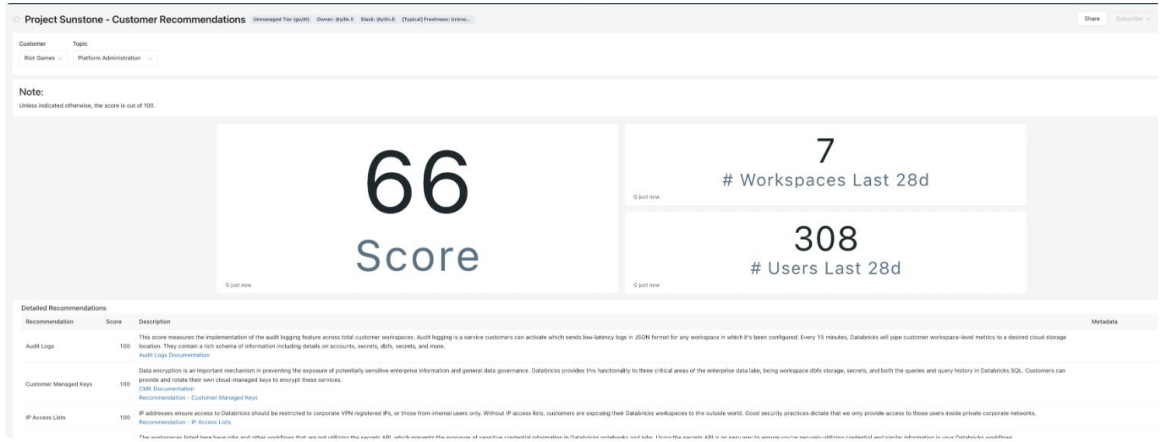
Assessment Summary

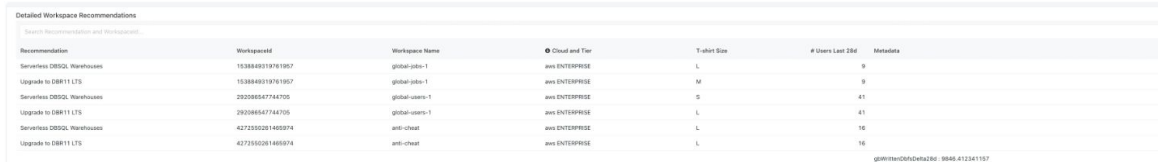
Best practices summary



Tools and Process







Data Governance

Summary: Data Governance

Dimension	Principle		Health and Key Findings	Key Recommendations
Unify Data Governance	1	Unify data management	Unity Catalog is enabled, however only two roles are defined. Security Model has three dimensions, however Information Security classification is flattened at the table level, while Security Classification is tagged at table level rather than column level	• Define Databricks UC Roles per data governance policies • Use Catalogs to provide segregation across EQL organisation's information architecture
	2	Unify data security	Account-level, workspace-level and Unity Catalog audit logging are enabled	None
Data Quality	3	Manage data quality	Current implementation of metadata-driven ingestion into Bronze layer has scope to improve capability to scale and evolve to business needs	• Define Schema / Metadata data quality rules and implementation • Build data ingestion auditing frameworks
Data Sharing	4	Share data securely and in real-time	Currently not applicable, however Delta Sharing can be enabled with Unity Catalog	• Consider using Delta Sharing to manage, govern, audit and track usage of shared data

Well Architected Lakehouse Assessment -Gold

Review the current data architecture with Lakehouse Well Architected Framework & Best Practices

10 days	DSA/RSA
1 day	Project Manager
3 weeks	

Well Architected Lakehouse

- Security and Data Governance reviews, unify data management, security and best practices
- Operational Excellence: Optimize build and release processes, including CI/CD, pipeline design, testing and orchestration, etc.
- Delta Lake design and performance efficiency, including data layout, partitioning strategy, table design and Z-ordering
- Reliability: design for autoscaling and DR best practices, monitoring and logging, etc.
- Usability: standards, open interfaces, data consumption patterns and best practices

Curious to learn more?

- Refer to the docs: Databricks well-architected framework
- Talk to your Databricks Representative or
- Contact us @ https://www.databricks.com/company/contact

Thank you

Q & A

Appendix

1 Data Governance

Dimension	Principle	Features and Best Practices
Unify Data Governance	Unify data management	• Manage metadata for all data assets in one place • Track data lineage to drive visibility of the data • Discover data and related information via the Data Explorer
	Unify data security	• Centralize access control • Configure audit logging • Audit Unity Catalog events • Audit data sharing events
Data Quality	Manage data quality	• Use a layered storage architecture • Improve data integrity by reducing data redundancy • Actively manage schemas • Use constraints and data expectations • Take a data-centric approach to machine learning
Data Sharing	Share data securely and in real-time	• Use the open Delta Sharing protocol for sharing data with partners • Use Databricks-to-Databricks Delta sharing between Databricks users

2 Interoperability and Usability

Dimension	Principle	Features and Best Practices
Interoperability	Settle on standards for integration	• Use the Databricks REST API for external integration • Use optimized connectors to access data sources from the Lakehouse • Leverage partners available in Partner Connect • Use Delta Live Tables and Auto Loader
	Prefer open interfaces and open data formats	• Use the Delta data format • Use Delta Sharing to exchange data with partners • Use MLflow to manage Machine Learning workflows
Usability	Lower the barriers to access the Lakehouse	• Provide a self-service experience across the platform • Use the serverless services of the platform • Offer predefined clusters and SQL Warehouses for different use cases
	Ensure data consistency and usability	• Offer reusable data-as-products that the business can trust • Publish data products semantically consistent across the enterprise • Use Unity Catalog for data discovery and lineage exploration

3 Operational Excellence (1/2)

Dimension	Principle	Features and Best Practices
Release Processes	Optimize build and release processes	• Use Databricks Repos to store code in Git • Standardize on DevOps processes (CI/CD) • Standardize in MLOps processes
Automation and Reproducibility	Automate deployments and workloads	• Use Infrastructure as Code for deployments and maintenance • Use cluster policies • Use automated workflows for jobs • Use Auto Loader • Use Delta Live Tables • Follow the “deploy code” approach for ML workloads • Use a model registry to decouple code and model lifecycle • Use MLflow Autologging • Reuse the same infrastructure to manage ML pipelines

3 Operational Excellence (2/2)

Dimension	Principle	Features and Best Practices
Monitoring	Set up system / workload monitoring, alerting, and logging	• Platform Monitoring via cloud monitoring solutions • Cluster Monitoring via Ganglia • SQL Warehouse monitoring • Auto Loader Monitoring • Delta Live Tables Monitoring • Streaming Monitoring • ML Models Monitoring • Security monitoring • Cost Monitoring
Capacity Management	Manage capacity and quota	• Manage capacity and quota • Invest in capacity planning

4 Security, Privacy, Compliance (1/2)

Dimension	Principle	Features and Best Practices
Identity and Access	Manage identity and access using least privilege	• Authenticate via single sign-on • Leverage multi-factor authentication • Disable local passwords • Set complex local passwords • Separate admin accounts from normal user accounts • Token management • SCIM synchronization of users and groups • Configure access control • Limiting cluster creation rights • Store and use secrets securely • Cross-account IAM role configuration • Customer-approved workspace login • Use clusters that support user isolation • Use Service Principals to run jobs and all automated tasks
Data security	Protect data in transit and at rest	• Avoid storing production data in DBFS • Secure access to cloud storage • Leverage data exfiltration settings within the admin console • Use bucket versioning • Encrypt storage and restrict access • Add a customer-managed key for managed services • Add a customer-managed key for workspace storage

4 Security, Privacy, Compliance (2/2)

Dimension	Principle	Features and Best Practices
Network security	Secure your network and identify and protect endpoints	• Deploy with a customer-managed VPC or VNet • Use Secure cluster connectivity • Use IP access lists • Implement network exfiltration protections • Apply VPC service controls • Use VPC endpoint policies • Configure PrivateLink • Configure domain name firewall rules
Compliance	Review shared responsibility	• Review the Shared Responsibility Model
	Meet the compliance and data privacy requirements	• GDPR and CCPA compliance using Delta Lake • HIPAA compliance features • PCI-DSS compliance controls • FedRAMP compliance
Monitoring	Monitor system security	• Use Databricks audit log delivery • Configure AWS, Azure, GCP tagging to monitor usage and enable charge-back • Monitor provisioning activities • Enhanced Security Monitoring
Generic Controls	n/a	• Service quotas • GCP org policies • Controlling libraries • Isolate sensitive workloads into different workspaces • Leverage CI/CD processes to scan code for hard coded secrets • Use AWS Nitro instances

5 Reliability

Dimension	Principle	Features and Best Practices
Resilience	Design for failure	• Use Delta Lake • Use Apache Spark or Photon for your distributed workloads • Automatically rescue invalid or non-conforming data • Configure jobs for automatic retries and termination • Use a scalable and production grade model serving infrastructure
Data Quality	Manage data quality	• Use a layered storage architecture • Improve data integrity by reducing data redundancy • Actively manage schemas • Use constraints and data expectations • Take a data-centric approach to machine learning
Autoscaling	Design for autoscaling	• Enable autoscaling for batch workloads • Enable autoscaling for SQL Warehouse • Leverage DLT's enhanced autoscaling
Recovery	Test recovery procedures	• Backup via the Databricks migration tool • Recover from Structured Streaming query failures • Recover ETL jobs based on Delta time travel • Leverage Databricks Workflows and its built in recovery • Configure a disaster recovery
Automation	Automate deployments and workloads	See Operational Excellence - Automation
Monitoring	Set up system / workload monitoring, alerting, and logging	See Operational Excellence - Monitoring

6 Performance Efficiency

Dimension	Principle	Features and Best Practices
Design	Use serverless architectures	• Use serverless compute
	Design for performance	• Use parallel computing where beneficial • Be aware of data access patterns • Use parallel computation where it is beneficial • Analyze the whole chain of execution • Prefer larger clusters • Use native operations • Use Photon • Use caching • Optimize file management • Join optimizations
Testing	Run performance testing in the scope of development	• Test on data representative for production data • Take prewarming of resources into account • Control caching • Identify bottlenecks
Monitoring	Monitor performance	See Operational Excellence

7 Cost Optimization

Dimension	Principle	Features and Best Practices
Resource Selection	Choose the correct resources	• Use Delta • Use automated Job clusters for jobs • Use SQL Warehouse for SQL workloads • Use the right runtime for your workloads • Ensure to use CPU and GPU for the right workloads • Consider using spot instances
Right Sizing	Dynamically allocate and deallocate resources	• Use auto termination
	Optimize workloads, aim for scalable costs	• Only use online streaming when the latency requirements demand for it • Choose the most efficient cluster size • Leverage auto-scaling compute • Use auto termination
Cost/performance profiling	Monitor and control cost	• Monitor costs • Evaluate Photon for your workloads • Evaluate Serverless for your workloads
Cost Control and attribution	Analyze and attribute expenditure	• Tag clusters for cost attribution