Databricks Architecture – Block Flow Diagram

┌──────────────────────────────────────────────────────────────┐  
│ **EXTERNAL DATA SOURCES** │  
│ - RDBMS (MySQL, Oracle) - REST APIs │  
│ - Event Streams (Kafka) - IoT/Logs/Flat Files (CSV, etc) │  
│ - ERP/CRM Systems - SaaS Platforms │  
└──────────────────────────────────────────────────────────────┘  
 │  
 ▼  
┌──────────────────────────────────────────────────────────────┐  
│ **INGESTION LAYER** │  
│ - Auto Loader (Streaming/Batch) │  
│ - Spark Structured Streaming │  
│ - COPY INTO / Partner tools (Fivetran, Informatica) │  
│ - Kafka / Event Hubs / PubSub │  
└──────────────────────────────────────────────────────────────┘  
 │  
 ▼  
┌──────────────────────────────────────────────────────────────┐  
│ **STORAGE LAYER + MEDALLION ARCHITECTURE** │  
│ (Implemented via Delta Lake) │  
│ │  
│ ┌────────────┐ ┌────────────┐ ┌────────────┐ │  
│ │ BRONZE │ → │ SILVER │ → │ GOLD │ │  
│ │ (Raw Data) │ │ (Cleaned) │ │ (Aggregated│ │  
│ │ │ │ (Joined) │ │ Curated) │ │  
│ └────────────┘ └────────────┘ └────────────┘ │  
│ - Stored in cloud: S3 / ADLS / GCS using Delta format │  
└──────────────────────────────────────────────────────────────┘  
 │  
 ▼  
┌──────────────────────────────────────────────────────────────┐  
│ **DATA PROCESSING & ENGINEERING LAYER** │  
│ - Apache Spark (PySpark, Scala, SQL) │  
│ - Delta Live Tables (DLT Pipelines) │  
│ - MLlib / XGBoost / Custom ML pipelines │  
│ - Databricks Workflows (job orchestration) │  
└──────────────────────────────────────────────────────────────┘  
 │  
 ▼  
┌──────────────────────────────────────────────────────────────┐  
│ **COMPUTE / RUNTIME ENVIRONMENT** │  
│ - Databricks Clusters (Interactive / Job Clusters) │  
│ - Photon Runtime (optimized engine for SQL) │  
│ - Serverless SQL Warehouses │  
└──────────────────────────────────────────────────────────────┘  
 │  
 ▼  
┌──────────────────────────────────────────────────────────────┐  
│ **GOVERNANCE & CATALOG LAYER** │  
│ - Unity Catalog (RBAC, Data Lineage, Tags, Audits) │  
│ - Table & Column Level Access Control │  
│ - Integration with external IAM (e.g., Azure AD) │  
└──────────────────────────────────────────────────────────────┘  
 │  
 ▼  
┌──────────────────────────────────────────────────────────────┐  
│ **CONSUMPTION & INTERFACE LAYER** │  
│ - Notebooks (Python, SQL, Scala, R) │  
│ - SQL Editor / BI Dashboards (Power BI, Tableau) │  
│ - Databricks Jobs (ETL Scheduling) │  
│ - REST APIs / SDKs / dbx CLI │  
└──────────────────────────────────────────────────────────────┘  
 │  
 ▼  
┌──────────────────────────────────────────────────────────────┐  
│ **CLOUD INFRASTRUCTURE PLATFORM** │  
│ - AWS / Azure / GCP │  
│ - Cloud Object Storage (S3, ADLS, GCS) │  
│ - Managed Kubernetes / VMs for Cluster Provisioning │  
└──────────────────────────────────────────────────────────────┘

**Explanation:**

1. **EXTERNAL DATA SOURCES**  
   This layer represents all possible sources of data that feed into Databricks.

* RDBMS (Relational Database Management System): Databases like MySQL, Oracle, PostgreSQL.
* REST APIs: Interfaces that allow applications to communicate over HTTP.
* Event Streams (Kafka): Real-time data pipelines using tools like Apache Kafka.
* IoT/Logs/Flat Files: Machine-generated data, device data, CSVs, JSON, etc.
* ERP/CRM Systems: Enterprise systems like SAP (ERP) or Salesforce (CRM).
* SaaS Platforms: Cloud-based services such as Google Analytics, Salesforce, Workday.

1. **INGESTION LAYER**  
   This is where raw data enters Databricks from the external sources.

* Auto Loader: A Databricks utility for ingesting data in batch or streaming mode automatically.
* Spark Structured Streaming: Framework for processing streaming data.
* COPY INTO: SQL command for loading data into Delta Lake.
* Partner Tools: Tools like Fivetran, Informatica that offer prebuilt connectors.
* Kafka / Event Hubs / PubSub:
  + Kafka: Distributed event streaming platform.
  + Event Hubs: Azure’s event ingestion service.
  + Pub/Sub: Google Cloud’s messaging system.

1. **STORAGE LAYER + MEDALLION ARCHITECTURE (Delta Lake)**  
   This is the structured data lake using Delta Lake and implements the Medallion Architecture.

* Delta Lake: An open-source storage layer that brings ACID transactions to data lakes.
* Medallion Architecture: A 3-tier logical design to organize data:  
  • BRONZE: Raw ingestion data (unfiltered, unvalidated).  
  • SILVER: Cleaned and joined data (validated, deduplicated).  
  • GOLD: Business-level curated data (aggregated, enriched).
* Cloud Storage:  
  • S3 – Amazon Simple Storage Service  
  • ADLS – Azure Data Lake Storage  
  • GCS – Google Cloud Storage

1. **DATA PROCESSING & ENGINEERING LAYER**  
   This layer transforms, enriches, and prepares the data for analytics or ML.

* Apache Spark: Distributed data processing engine (Databricks is built on this).
* PySpark: Python API for Apache Spark.
* Delta Live Tables (DLT): Declarative pipelines that automate ETL workflows.
* MLlib / XGBoost: Built-in libraries for machine learning.
* Databricks Workflows: Orchestrates jobs, schedules, and dependencies.

1. **COMPUTE / RUNTIME ENVIRONMENT**  
   Resources used to execute queries, notebooks, pipelines, and ML models.

* Databricks Clusters:  
  • Interactive Cluster – For development & notebooks.  
  • Job Cluster – For production jobs.
* Photon Runtime: Highly performant SQL engine built by Databricks.
* Serverless SQL Warehouses: Auto-managed compute for SQL users.

1. **GOVERNANCE & CATALOG LAYER**  
   Handles access, security, metadata, and lineage.

* Unity Catalog:  
  • RBAC – Role-Based Access Control  
  • Lineage – Track data origin and transformation  
  • Tags – Metadata labels  
  • Audits – Logs for compliance and monitoring
* IAM (Identity and Access Management): Integrated with providers like Azure Active Directory for user identity and permission control.

1. **CONSUMPTION & INTERFACE LAYER**  
   Interfaces and tools used to consume and interact with the data.

* Notebooks: Interactive environment for coding in Python, SQL, Scala, R.
* SQL Editor: GUI-based SQL editor in Databricks.
* BI Dashboards:  
  • Power BI – Microsoft’s analytics dashboard tool.  
  • Tableau – Popular data visualization platform.
* Databricks Jobs: Scheduler for notebooks, scripts, pipelines.
* REST APIs / SDKs / dbx CLI:  
  • REST API – Allows programmatic access.  
  • SDK – Software Development Kit for integration.  
  • dbx CLI – CLI tool to manage jobs, repos, deployments.

1. **CLOUD INFRASTRUCTURE PLATFORM**  
   Underlying cloud services where Databricks is deployed.

* AWS / Azure / GCP – Cloud platforms supported by Databricks.
* Cloud Object Storage: Persistent storage used by Delta Lake.
* Managed Kubernetes / VMs:  
  • Kubernetes – Container orchestration engine.  
  • VMs – Virtual Machines provisioned for running Spark clusters.

**Summary: How It All Connects**

* Data flows from external sources to the Ingestion Layer.
* The data is then stored in Delta Lake following the Medallion Architecture (Bronze → Silver → Gold).
* It’s processed using Spark and ML pipelines in the Data Engineering Layer.
* These transformations run on compute clusters managed by Databricks.
* Governance ensures secure and compliant data access.
* Finally, data is consumed via notebooks, dashboards, or APIs.
* All of this is built on cloud infrastructure like AWS, Azure, or GCP.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*