1. **Modern Streaming Architectures**

- **Modern Streaming Architecture** refers to the design patterns and frameworks that enable **real-time data processing, analytics, and decision-making** across distributed systems.
- Unlike traditional batch systems, these architectures continuously ingest, process, and serve data streams with minimal latency.

Core Principles of Modern Streaming Architecture

- **Real-Time Processing** → Data is processed as it arrives, enabling instant insights.
- **Unified Batch + Stream** → Combines historical and live data in one pipeline.
- **Scalability** → Handles millions of events per second across distributed nodes.
- **Fault Tolerance** → Ensures data consistency and recovery during failures.
- **Low Latency** → Optimized for sub-second response times.

Key Components

Component	Description	Examples
Data Ingestion	Captures continuous event streams	Kafka, Pulsar, Redpanda
Stream Processing	Transforms, aggregates, and enriches data	Spark Streaming, Flink, Beam
Storage Layer	Persists raw and processed data	Delta Lake, Iceberg, Paimon
Serving Layer	Exposes results to dashboards, APIs, ML models	Presto, Druid, Elasticsearch

Popular Modern Architectures

- **Delta Architecture** → ACID transactions and schema enforcement for reliability.
- **Medallion Architecture** → Layered refinement (Bronze–Silver–Gold) for governance.
- **Streamhouse Architecture** → Unified lakehouse + streaming for real-time analytics.

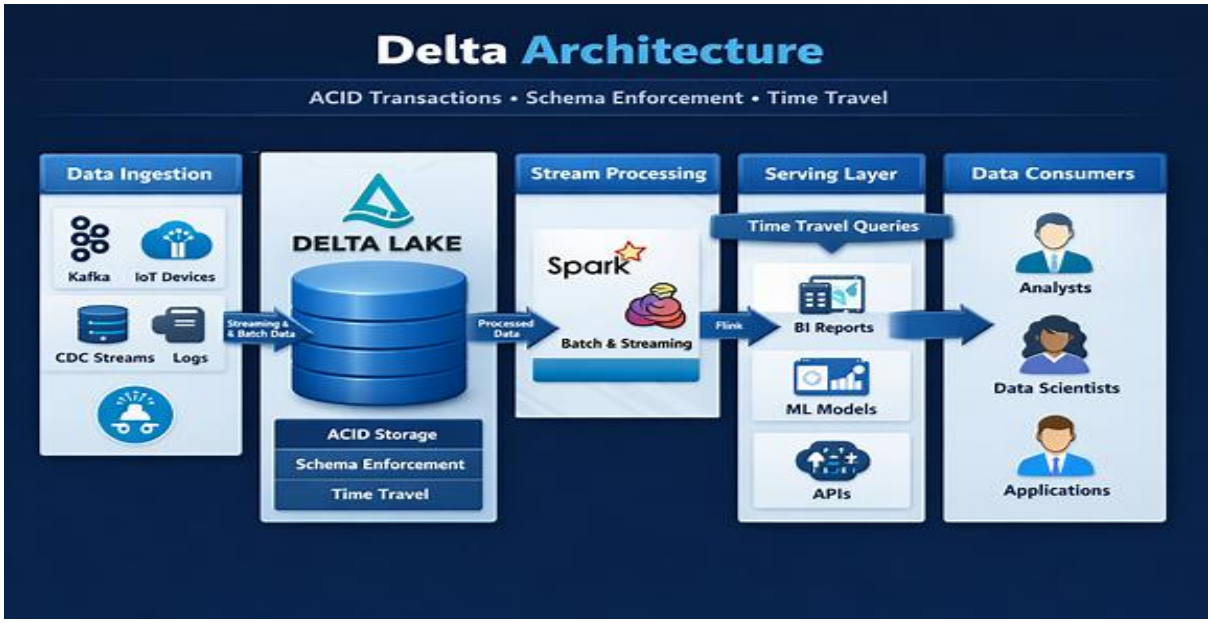
1.Delta Architecture

- Built on **Delta Lake** (Databricks).
- Focuses on **ACID transactions** and schema enforcement for streaming + batch data.
- Provides **time travel** (querying historical snapshots).
- Ideal for **data lakes** needing reliability and consistency.

Use Cases: Financial analytics, compliance reporting, large-scale ETL

- Data is processed in three layers: batch, speed, and serving
- Uses a combination of batch-processing frameworks, stream-processing engines, and real-time databases
- Storage is typically in a **data lake** or a **cloud-based storage system**
- Offers the ability to update and delete data in real-time, which is not possible in the other two architectures(Lambda,Kappa)
- Designed to address the limitations of both Kappa and Lambda architectures

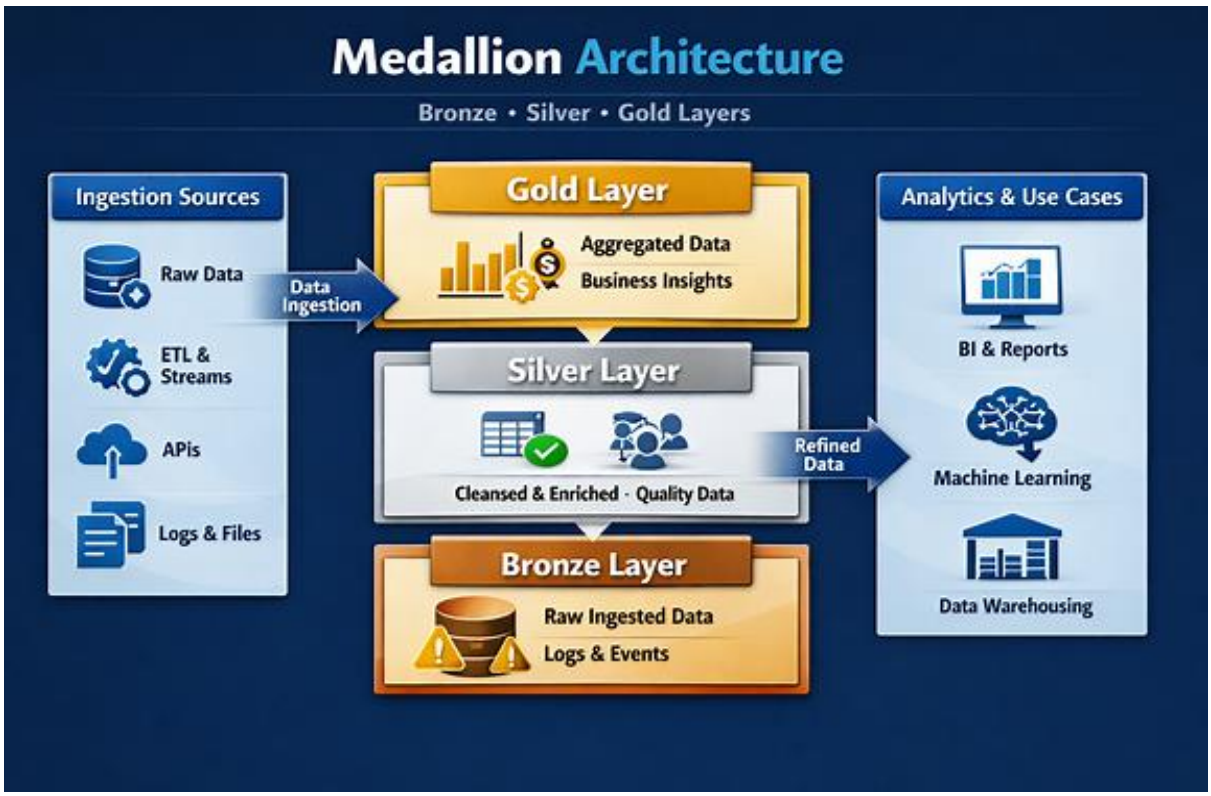
Delta Architecture



- **Data Ingestion** → Kafka, IoT devices, CDC streams, logs.
- **Delta Lake** → ACID storage, schema enforcement, time travel.
- **Stream Processing** → Spark, Flink for batch + streaming.
- **Serving Layer** → BI dashboards, ML models, APIs.
- **Consumers** → Analysts, data scientists, applications.

2.Medallion Architecture

- Organizes data into **Bronze, Silver, and Gold layers**.
    - **Bronze**: Raw ingested data.
    - **Silver**: Cleaned, enriched, and structured data.
    - **Gold**: Aggregated, business-ready datasets.
- Simplifies **data lineage** and governance.
- **Use Cases**: Enterprise data warehouses, BI dashboards, machine learning pipelines



Layer Overview

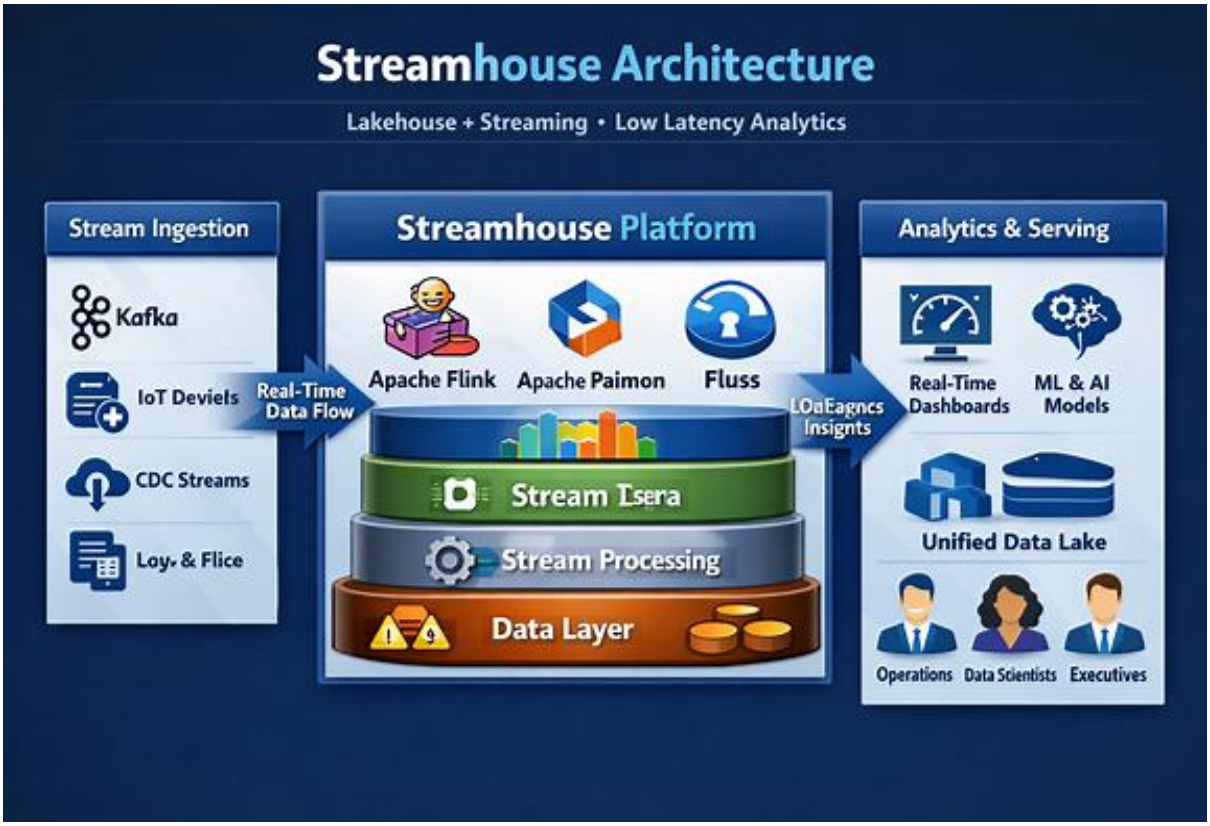
- **Bronze Layer** → Raw ingested data from APIs, logs, and streams.
- **Silver Layer** → Cleansed, deduplicated, and enriched datasets.
- **Gold Layer** → Aggregated, business-ready data for analytics and ML.

Work Flow

- **Ingestion Sources** → Databases, IoT, ETL pipelines feed into Bronze.
- **Transformation** → ETL jobs refine Bronze into Silver.
- **Aggregation** → Silver data is summarized into Gold for BI and ML.
- **Consumption** → Dashboards, reports, and predictive models use Gold data.

1. Streamhouse Architecture

- A hybrid of **Lakehouse + Streaming systems**.
- Combines **real-time low-latency streaming** with **robust analytics** of Lakehouse.
- Powered by tools like **Apache Flink, Apache Paimon, Fluss**.
- Enables **unified batch + stream processing** in one system.
- **Use Cases**: IoT analytics, fraud detection, operational monitoring, unified data lakes



Streamhouse Architecture Overview

- **Core Idea:** Combines **streaming ingestion** with **lakehouse storage** for real-time and historical analytics in one platform.
- **Key Technologies:** Apache Flink, Apache Paimon, Fluss — enabling unified batch and stream processing.
- **Goal:** Deliver **low-latency insights** without maintaining separate pipelines for streaming and batch.

Architecture Flow

- Stream Ingestion
  - Sources: Kafka, IoT sensors, CDC streams.
  - Data flows continuously into the Streamhouse platform.

- **Streamhouse Platform**
    - **Unified Storage:** Combines lakehouse reliability with streaming speed
    - **Stream Processing:** Real-time transformations via Flink and Paimon.
    - **Batch Analytics:** Historical queries and aggregations
- **Analytics & Serving**
    - Real-time dashboards, ML models, unified data lakes
    - Consumers: Operations teams, data scientists, executives.

Comparison Table

Architecture	Key Feature	Strengths	Best Use Cases
Delta	ACID transactions in data lakes	Reliability, schema enforcement	Finance, compliance, ETL
Medallion	Layered (Bronze–Silver–Gold)	Clear lineage, governance	BI dashboards, ML pipelines
Streamhouse	Lakehouse + Streaming	Unified batch + stream, low latency	IoT, fraud detection, real-time analytics

Trade-offs

- **Delta**: Strong consistency but higher overhead.
- **Medallion**: Requires careful pipeline design; more storage layers.
- **Streamhouse**: Emerging model; tooling still maturing.

Real-World Adoption

- **Databricks** → Delta & Medallion widely used in enterprise analytics.
- **Ververica (Apache Flink)** → Streamhouse for real-time + historical unification.
- **Global Enterprises** → Finance, healthcare, and e-commerce industries are shifting toward these models for **scalable real-time insights**.

Comparison Table

Architecture	Core Principle	Processing Model	Storage System	Strengths	Limitations	Best Use Cases
Lambda	Dual pipeline (batch + stream)	Batch + Stream	HDFS, S3, NoSQL	Accuracy, fault tolerance	Complex maintenance, duplicate logic	Historical + real-time analytics
Kappa	Stream-only pipeline	Stream	Kafka, Pulsar, Redpanda	Low latency, simplicity	Harder debugging, limited batch support	IoT, fraud detection, real-time dashboards
Delta	Unified batch + stream with ACID	Batch + Stream	Delta Lake	Reliability, schema enforcement, time travel	Slight overhead for ACID	Finance, compliance, ETL
Medallion	Layered data refinement (Bronze–Silver–Gold)	Batch + Stream	Delta Lake / Lakehouse	Clear lineage, governance	More storage layers, design complexity	BI dashboards, ML pipelines
Streamhouse	Unified Lakehouse + Streaming	Stream + Batch	Flink, Paimon, Fluss	Real-time + historical unification	Emerging tooling	IoT analytics, operational monitoring