BTS – MBDS
Big Data Infrastructure

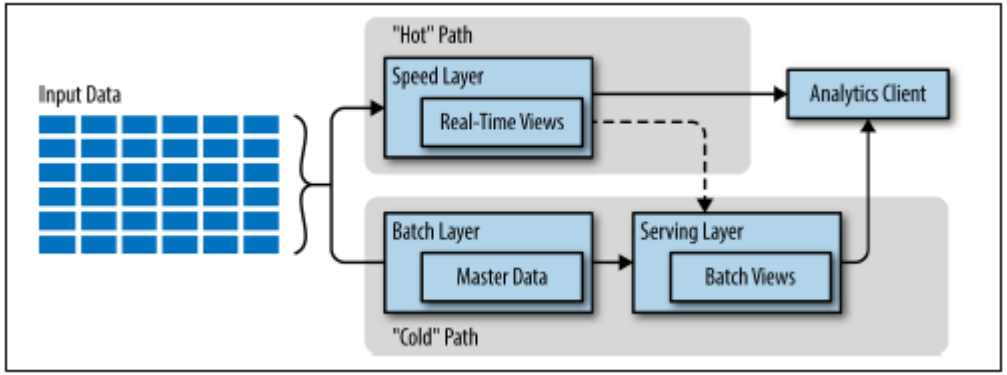
A1: Lambda vs Delta Architecture

Ennio Maldonado

18 February, 2021

Lambda Architecture

Lambda architecture is a data-processing architecture designed to handle massive quantities of data (i.e. “Big Data”) by using both batch-processing and stream-processing methods. This idea is to balance latency, throughput, scaling, and fault-tolerance by using batch processing to provide comprehensive and accurate views of batch data, while simultaneously using real-time stream processing to provide views of online data. The two view outputs may be joined before presentation.



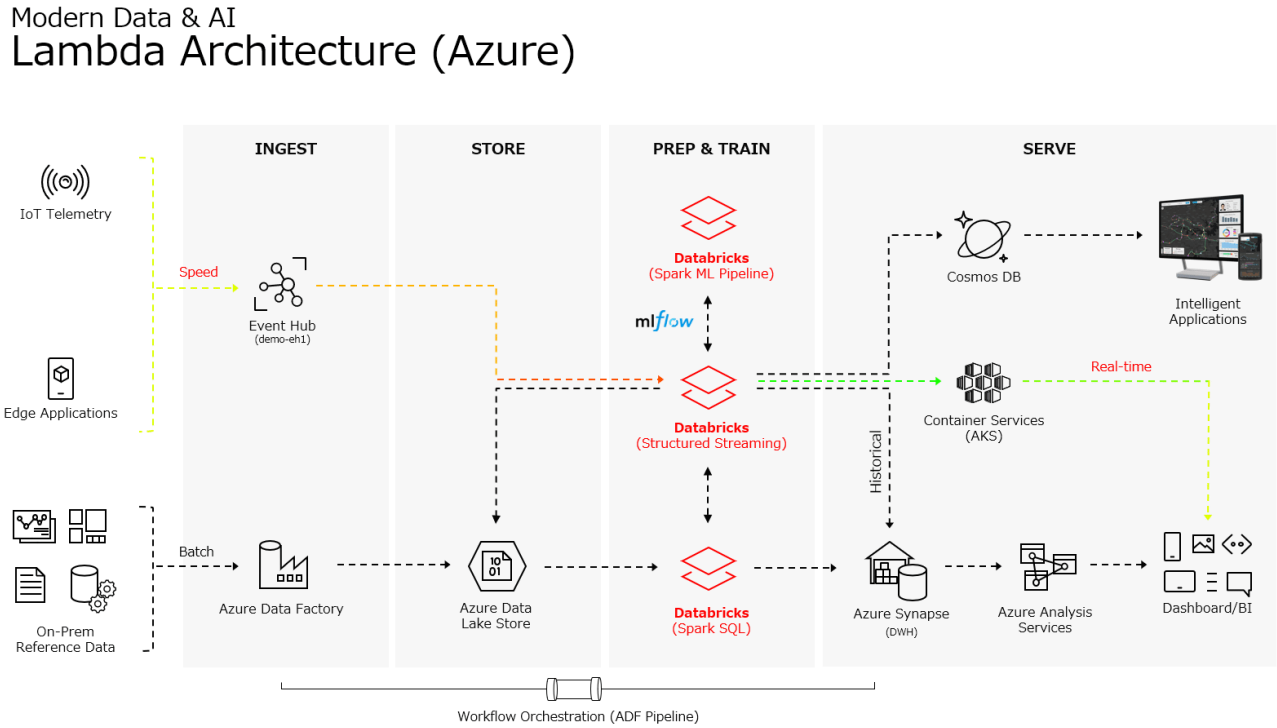
Lambda - Layers

Data Consumption: Import the data.

Stream Layer: Incremental updating.

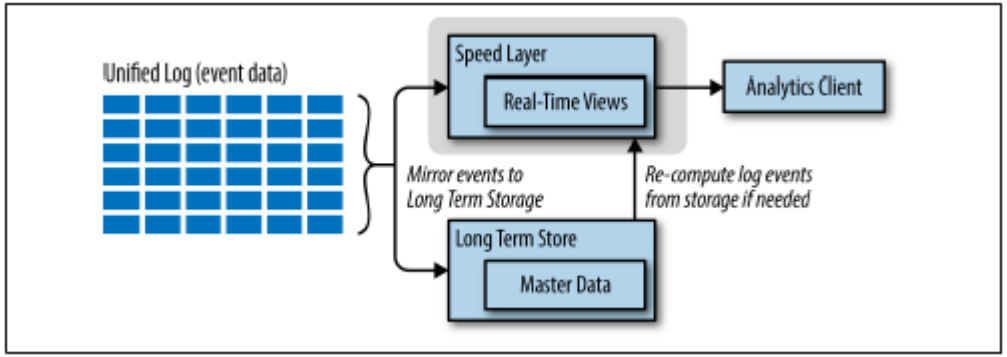
Batch Layer: All the data at once.

Presentation Layer: mediator, it accepts queries and decides when to use the batch layer and when to use the speed layer.



Kappa Architecture

The evolution of Kappa Architecture can be thought of as a response to simplifying the Lambda Architecture by eliminating the cold path and to ensure that all processing happens in a near real-time streaming node and that recomputation on the data happens when needed being streamed through the Kappa pipeline again.



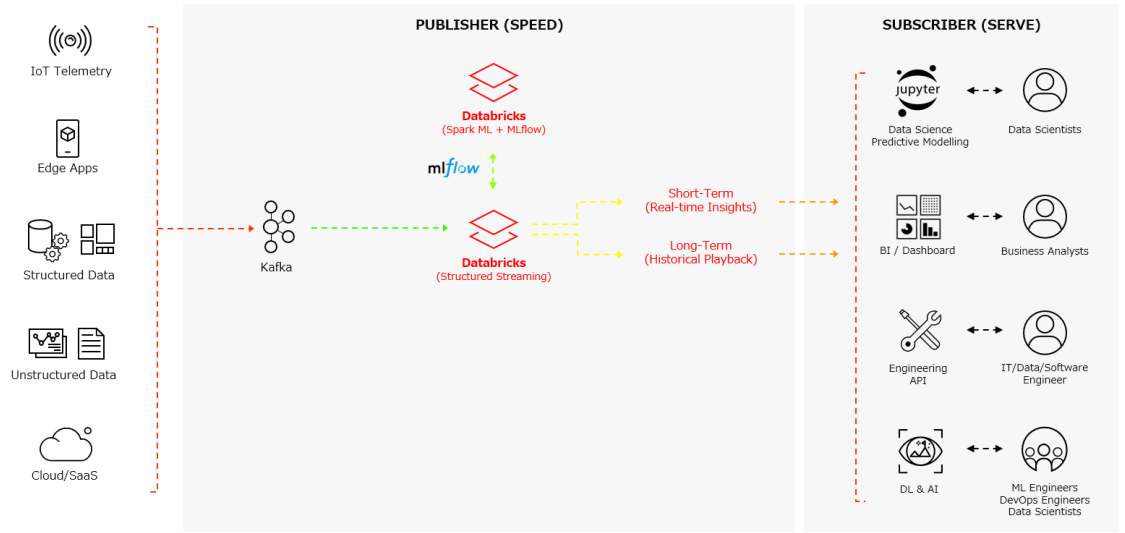
Kappa - Layers

Data Consumption: Import the data.

Speed Layer ONLY

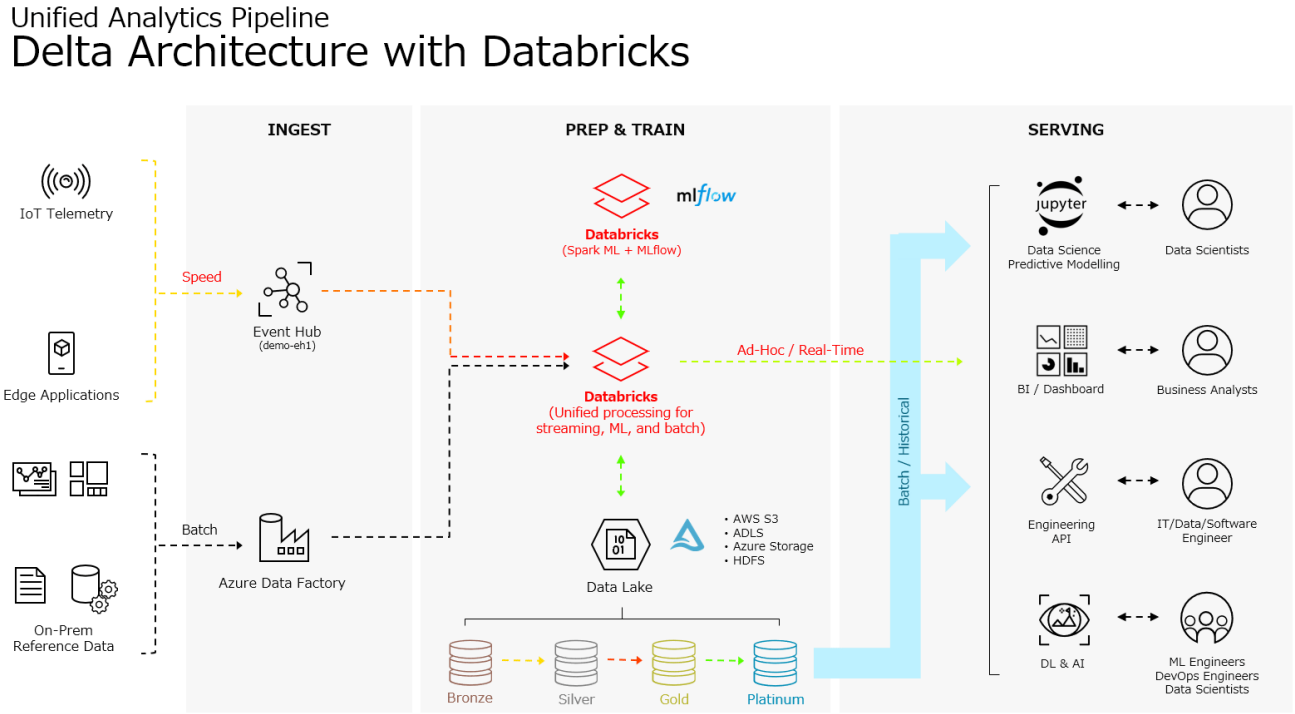
Presentation Layer: mediator, it accepts queries and decides when to use the batch layer and when to use the speed layer.

Kappa Architecture



Delta Architecture

Delta architecture no longer considers data lake as immutable. In contrary, incoming data is processed as “delta” records (i.e. the differentials as per the Greek letter “Δ“) rather than the append-only new records. Batch transformations can perform DML such as CRUD operations on existing data structures in the data lake by using the technology known as Delta Lake. In fact, the Delta Lake brings Datawarehouse-like capabilities (ACID transactions, DML, a specialized indexing technology for distributed datsets etc.) to legacy Data Lake, making it more perform ant and reliable in Big Data processing pipelines. As a result, it effectively unifies the two layers for a seamless processing (i.e. same engine, same API and same code for batch and streaming) with lesser overheads (i.e. performance optimization). The benefit is huge as this capability means organizations no longer have to treat data differently based on the speed of ingestion and processing method.



Delta – Possible architecture

Sources: Salesforce and SAP

Ingest: Kafka

Delta Lake: Databricks

Query: MongoDB

Machine Learning: Sagemaker

Analytics: Tableau