
BEAM
SUMMIT

The ASF Data Ecosystem Bridging the data stream with Apache Beam

JB Onofré
<jbonofre@apache.org>

BEAM SUMMIT NYC 2025

JB Onofré

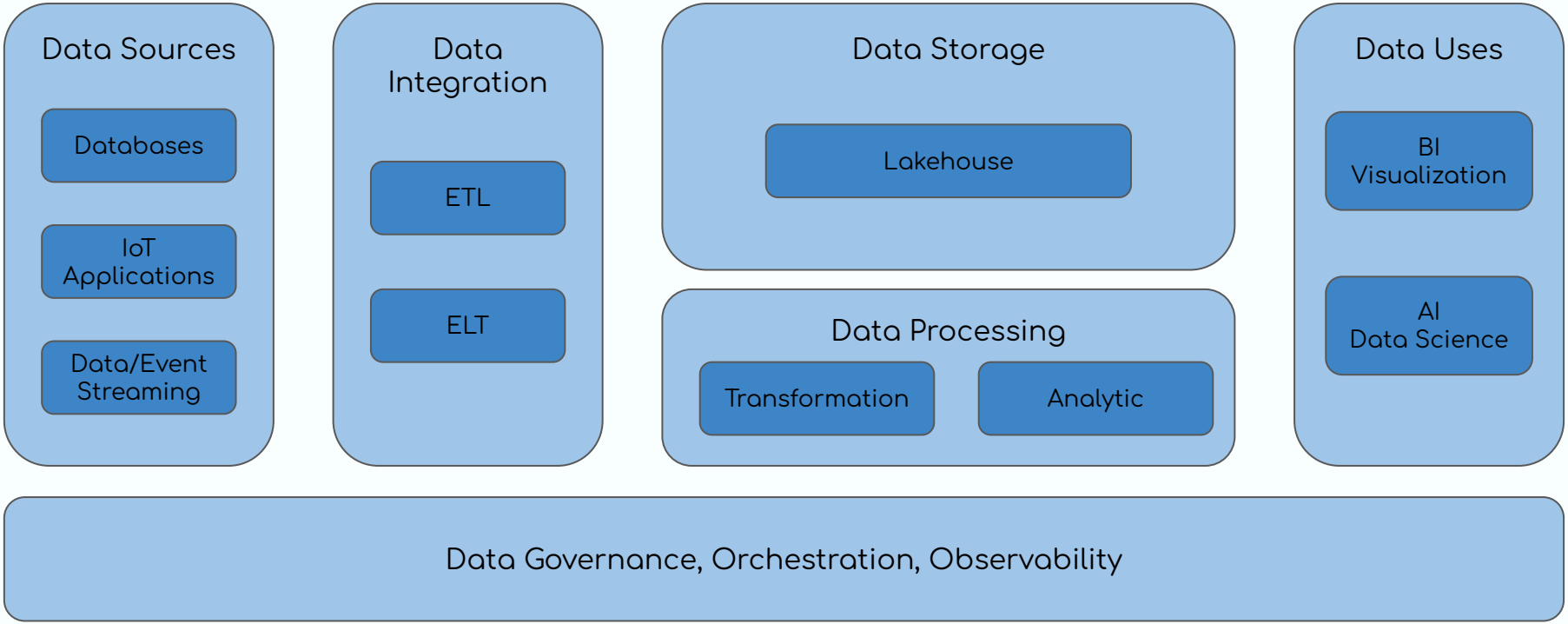
- Director of the Apache Software Foundation, member for a long long time :)
- Committer/PMC member on 20+ Apache projects (Apache ActiveMQ, Apache Beam, …)
- Mentor/champion on Apache Incubator projects (podlings)
- Principal engineer at Dremio, leading the OSPO team

Today: the modern data stack

- New use cases (AI) need tools to value and refine the data
  - Modular & Flexible
  - Performance (cloud/hybrid ready, automatic workload management)
  - Cost
  - Easy deployment (by area
  - Scalable

- A lot of stacks in the modern data stack :)
  - Data streaming
  - Transformation
  - Lakehouse
  - …

Modern Data Stack Map



Modern Data Stack with ASF ecosystem

- The ASF is the natural home for data open source projects (since Apache Hadoop)
- With ~ 300 projects, the ASF provides all components of the Modern Data Stack
  - A real Open Source and Open Govern Modern Data Stack
  - No vendor lock-in
  - A large choice for almost all use cases
  - Evolutivity and flexibility

Modern Data Stack with ASF ecosystem

Data Sources

Data Integration

Data Storage


Data Processing

Data Uses

Data Governance, Orchestration, Observability

Example of lakehouse

Query Engine

Request the catalog to get entities and use table format metadata for planning and query data

Catalog

Index entities from table format for query engine

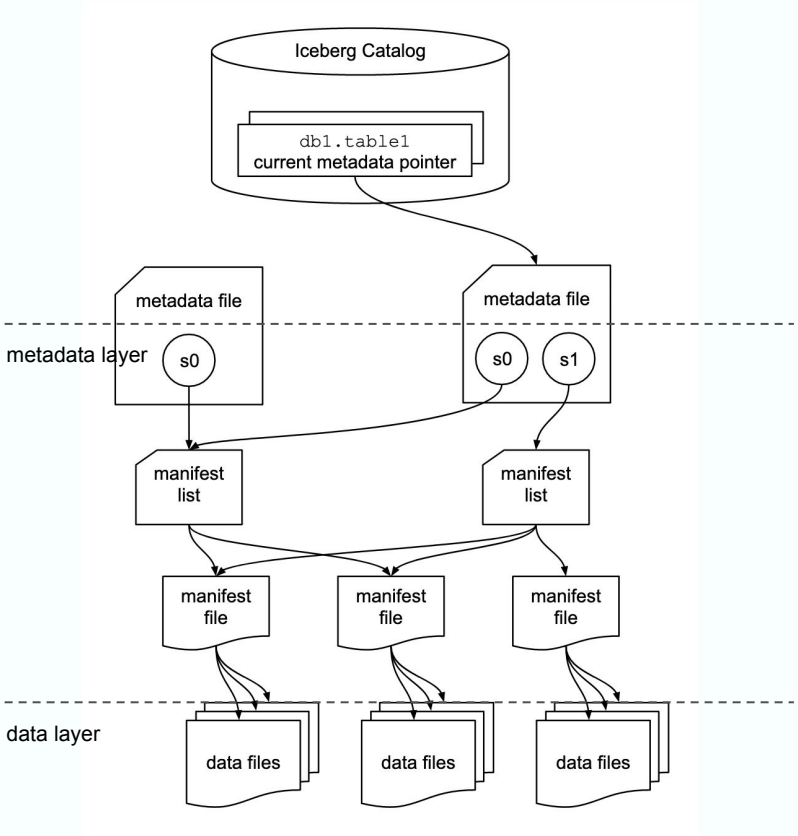
Table Format

Add metadata on top of data files

Data Files Format / Storage

Where the data is actually stored (data files, object store, ...)

Apache Iceberg



- 4 specifications (Table, View, Puffin, REST Catalog)
- Libraries to facilitate use in query engines (and tools), in different languages
- Apache Iceberg is not a storage engine, an execution engine, a service

Apache Iceberg - Key features

Consistency guarantees/ACID

- All engines use catalog reference to table data as source of truth ensuring consistent view of data across tools
- Catalog table/view reference only updated when a transaction is successfully completed

Query engine performance

- Using Iceberg's metadata, engines can avoid file-listing operations and more efficiently skip data files in the table that don't need to be scanned (especially with partition and delete files)
- Less files scanned == faster and cheaper query

Schema evolution

- Table schema is tracked in the metadata, allowing the schema to evolve without the need to rewrite all data

Partition evolution

- Partition is part of the metadata, so you can change partitioning without having to rewrite the table
- Hidden partitioning is also possible based on the transformed value of a column instead of its raw value

Where is Apache Beam?

Where is Apache Beam in the Modern Data Stack ?

Actually, Apache Beam can be used in multiple layers !

Beam in the Modern Data Stack

Data Sources

Data Integration

beam

Data Storage

Data Processing

beam

Data Uses

Data Governance, Orchestration, Observability

beam

Where is Apache Beam?

- Beam as Data Preprocessing framework:
  - Data integration
  - Data cleaning
  - Data transformation
- Data Processing (batch and streaming), Beam as unified framework
- Data Orchestration, Beam can “orchestrate” data processes

Apache Beam quick intro

- Unified programming model for batch and streaming
- Java, Go, and Python implementations of the model
- Parallel data processing
  - Smaller chunks of data processed in parallel and independently
  - Can transform a dataset of any size coming from a batch data source (bounded data) or a streaming data source (unbounded data)
- **Pipeline** is the processing logic from start to end. Each step read the input data, transform that data, write output. Pipelines run with execution option defining where and how to run.
- **PCollection** is the distributed data processed inside a pipeline. The data can be bounded or unbounded.
- **PTransform** is an operation in the pipeline. The input for every PTransform is a PCollection object, performs the processing logic, and gives zero or more PCollection objects as an output.

Beam Pipeline

- Create a pipeline object
- Read data into the pipeline - Read/Create PCollection
- Apply transforms to process pipeline data
- Output the final transformed PCollections
- Run the pipeline
- Testing a pipeline
- Testing DoFn and Composite Transforms
- Testing a pipeline end to end (using TestPipeline)

Beam Transforms

- Element transforms
  - Filter
  - FlatMapElements
  - ParDo/DoFn
  - Keys
  - Values
  - …
- Aggregate PCollections
  - CoGroupByKey
  - Combine
  - GroupByKey
  - Count
  - Max
  - …

Bridging the data stream

- The ASF is the natural home for data framework
- Those projects are great choice to implement Modern Data Stack
  - Vibrant and active community - CommunityOverCode
  - Cover a large part of the Modern Data Stack
  - Flexible and extensible
  - No vendor lock-in
- Apache Beam can be swiss knife of the Modern Data Stack as it can be used in different layers thanks
  - Unique model/language for batch and streaming
  - Multiple runners to match the expectations of each layers
  - Multiple connectors to read/write data

Apache Beam bridges the data stream in the Modern Data Stack !

QUESTIONS?

jbonofre@apache.org
jb.onofre@dremio.com