LET'S GO HYBRID
bigdata
TECHNOLOGY WARSAW SUMMIT

WORKSHOPS
APRIL 26th, 2022 ONSITE ONLY

CONFERENCE
APRIL 27th, 2022 ONLINE OR ONSITE
APRIL 28th, 2022 ONLINE

www.bigdatatechwarsaw.eu

Scaling Your Data Lake w/ Iceberg

- Victoria Bukta (Shopify)

Victoria Bukta

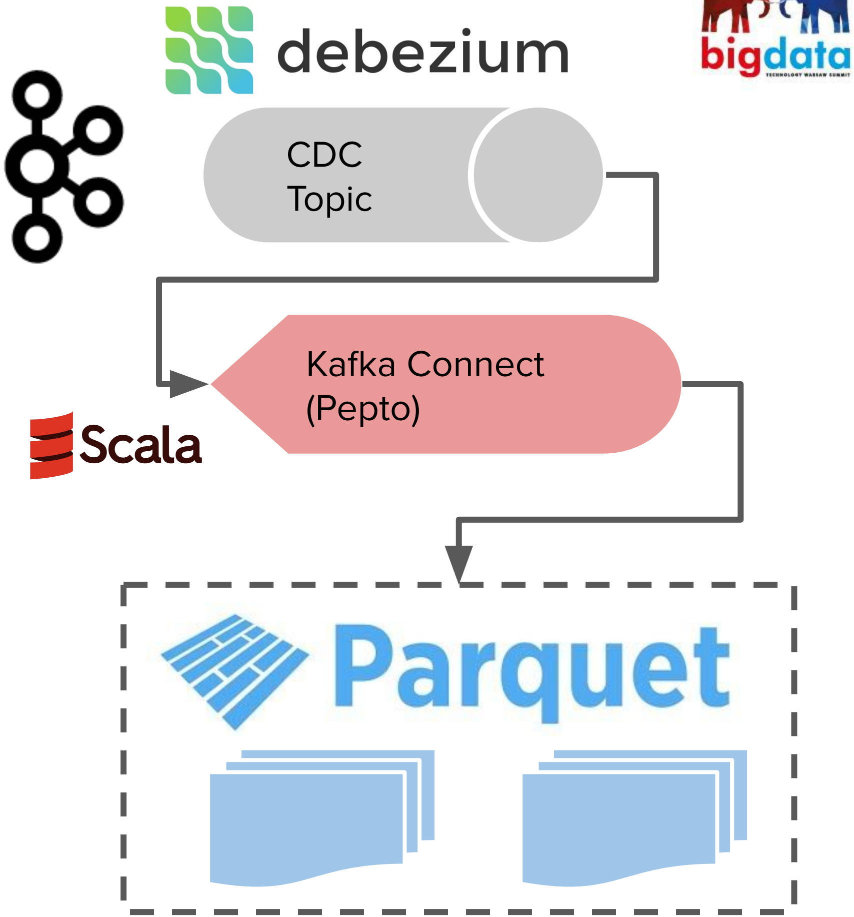
- Based in Toronto
- Senior Data Platform Eng
- At Shopify for 4.5 years
  - Toronto & Berlin offices
- Lakehouse (formally Data Acquisition)
- Hobbies
  - Field Hockey
  - Sailing
  - Backcountry Camping

Agenda

- Context
- Problem
- Solution
  - What is Apache Iceberg?
  - Promise (V2 Spec)
- Result
- Reflection
- Future Challenges

Context

- New Kafka ingestion tooling is being built to support CDC (change data capture) use case (Pepto)
  - **Streaming ingestion**
  - 15 min SLA
  - **Anticipated huge table, trillions of rows**
  - Columnar schematized datasets
  - Time series data
  - Aggressive schema evolutions
  - **Future use case of supporting Type-1 tables**
    - **new data overwrites the existing data**
- **Kafka Connect application** because of internal support / expertise at scale
- Require read support from Spark, Trino, Flink



Problem

1. Transactional Semantics

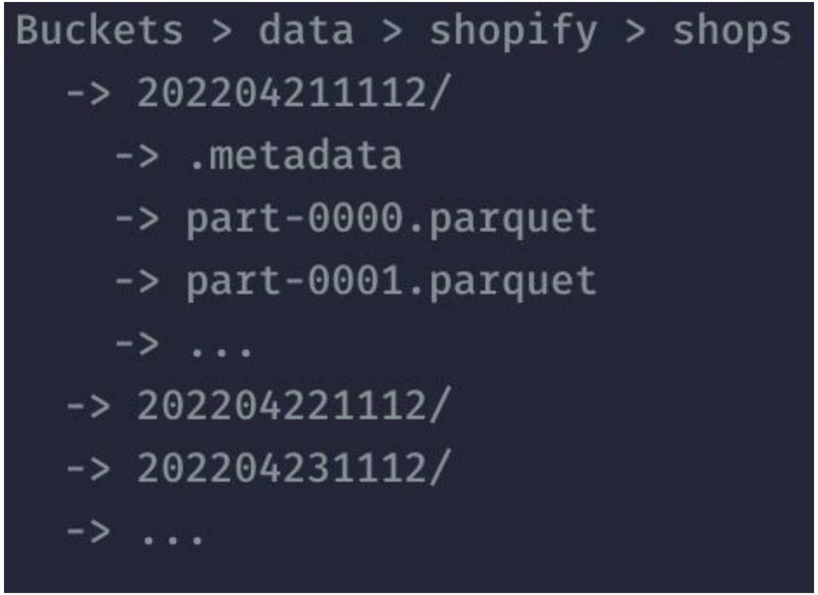
2. Fast upsert to support Type-1 tables

Problem - Transactional Semantics

- **Modeling tools currently tied to our writing implementation**
    - HDFS vs Object Store **(NOT THE SAME)**
        - FS abstraction is missing
            - Atomic move, rename
    - Timestamp folders on GCS
        - Makes it hard to do maintenance tasks without effecting customers
        - Data scientists refer to datasets by folder location
- **Difficult to innovate** when implementation details are exposed

Problem - Transactional Semantics

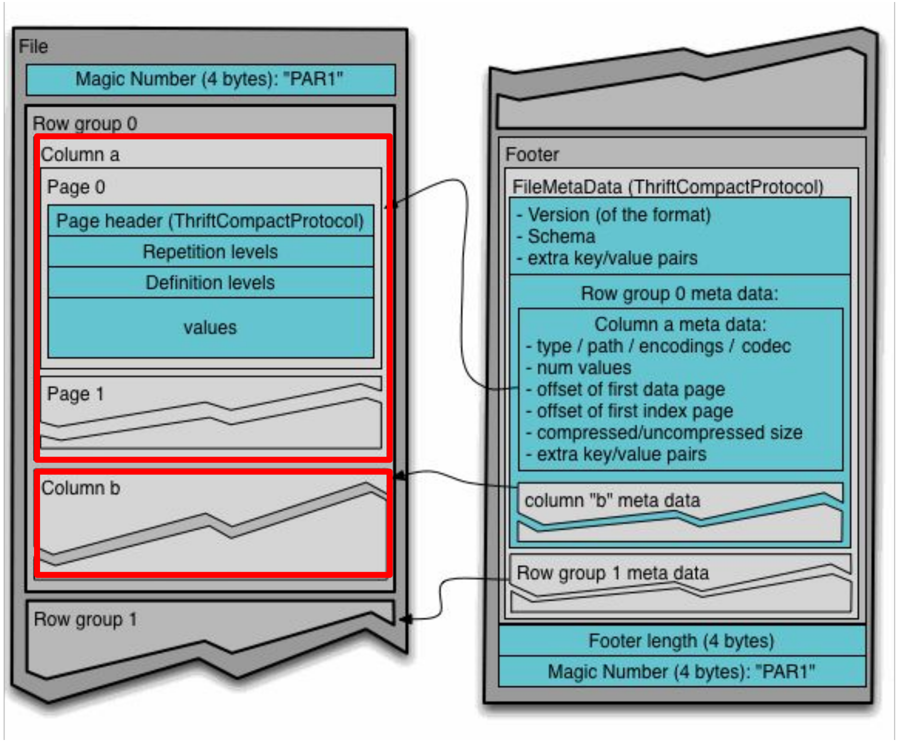
- **Modeling tools currently tied to our writing implementation**
    - HDFS vs Object Store **(NOT THE SAME)**
        - FS abstraction is missing
            - Atomic move, rename
    - Timestamp folders on GCS
        - Makes it hard to do maintenance tasks without effecting customers
        - Data scientists refer to datasets by folder location
- **Difficult to innovate** when implementation details are exposed



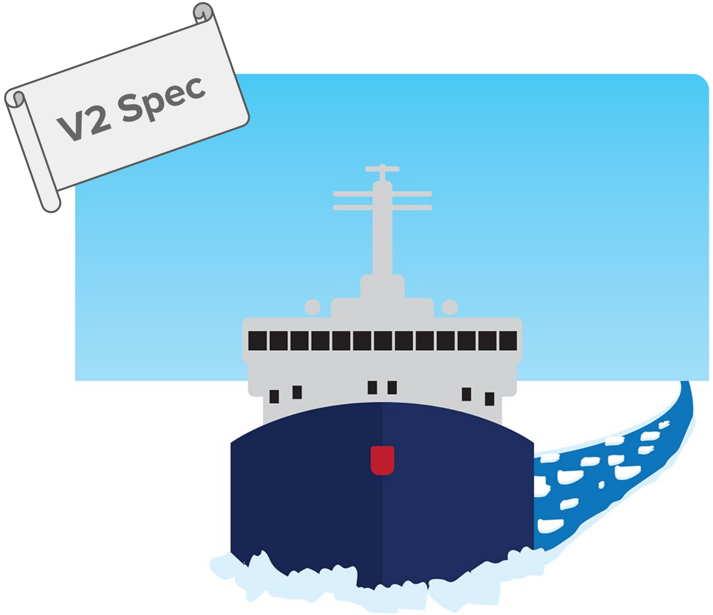
```
Buckets > data > shopify > shops
  -> 202204211112/
    -> .metadata
    -> part-0000.parquet
    -> part-0001.parquet
    -> ...
  -> 202204221112/
  -> 202204231112/
  -> ...
```

Problem - Fast upsert

- **Storing data in columnar format**
  - Efficient compaction of schematized data
  - Optimizing for aggregation analytics over a subset of columns
- **Creating Type-1 dimensions is hard**
  - Columnar files are immutable
  - Rewrite is an expensive operation
  - People want their data **NOW**

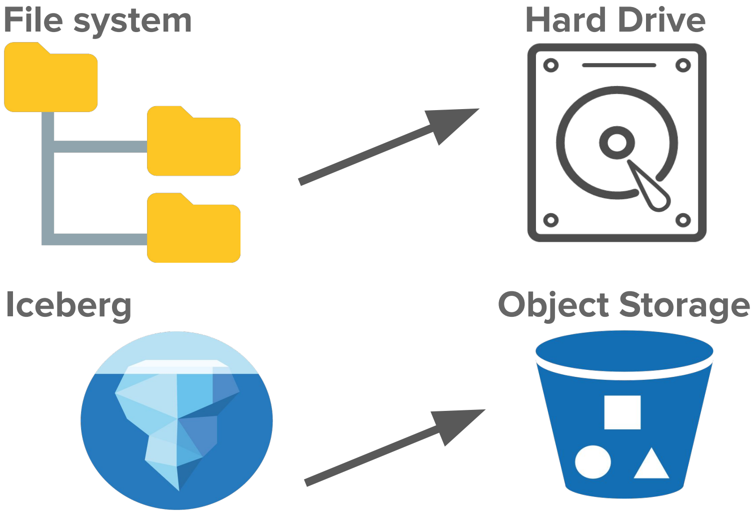

File
Magic Number (4 bytes): "PAR1"
Row group 0
Column a
Page 0
Page header (ThriftCompactProtocol)
Repetition levels
Definition levels
values
Page 1
Column b
Row group 1
Footer
FileMetaData (ThriftCompactProtocol)
- Version (of the format)
- Schema
- extra key/value pairs
Row group 0 meta data:
Column a meta data:
- type / path / encodings / codec
- num values
- offset of first data page
- offset of first index page
- compressed/uncompressed size
- extra key/value pairs
column "b" meta data
Row group 1 meta data
Footer length (4 bytes)
Magic Number (4 bytes): "PAR1"

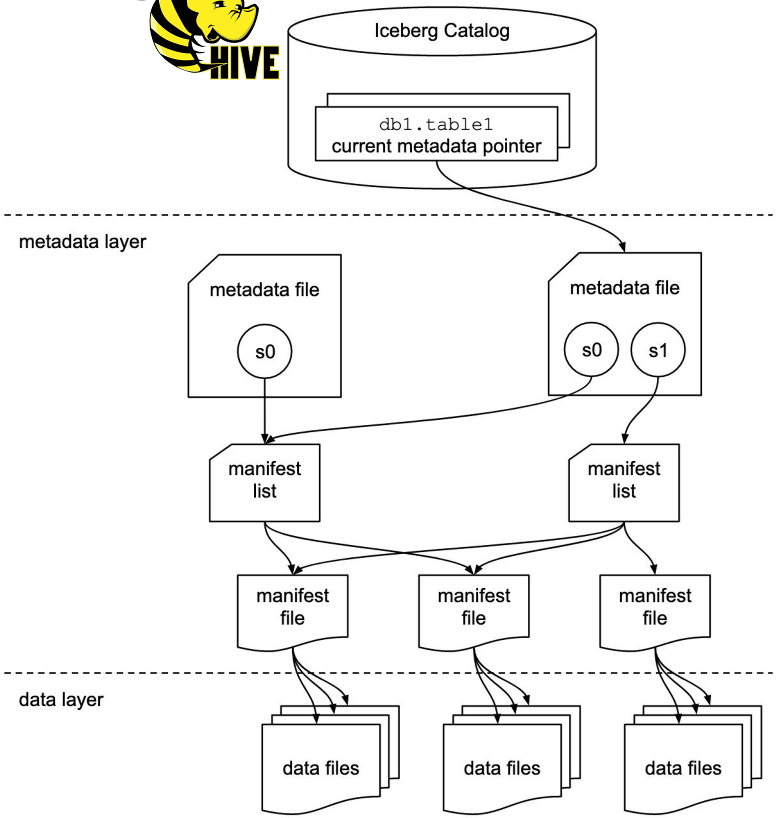
Solution



Solution - What is Apache Iceberg?

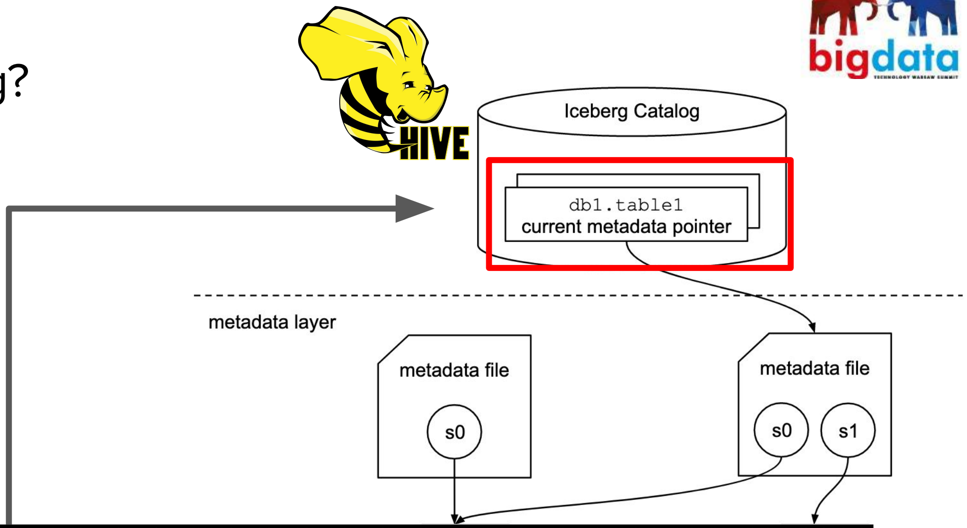
- **Iceberg is a table format**
  - **Just a library**
  - Contents of a table are identified by traversing through metadata files

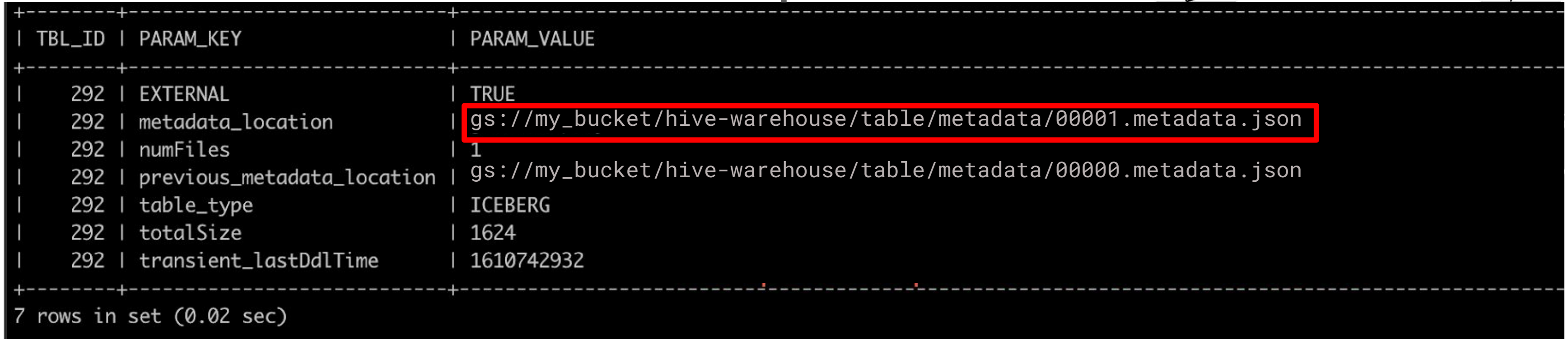




Solution - What is Apache Iceberg?

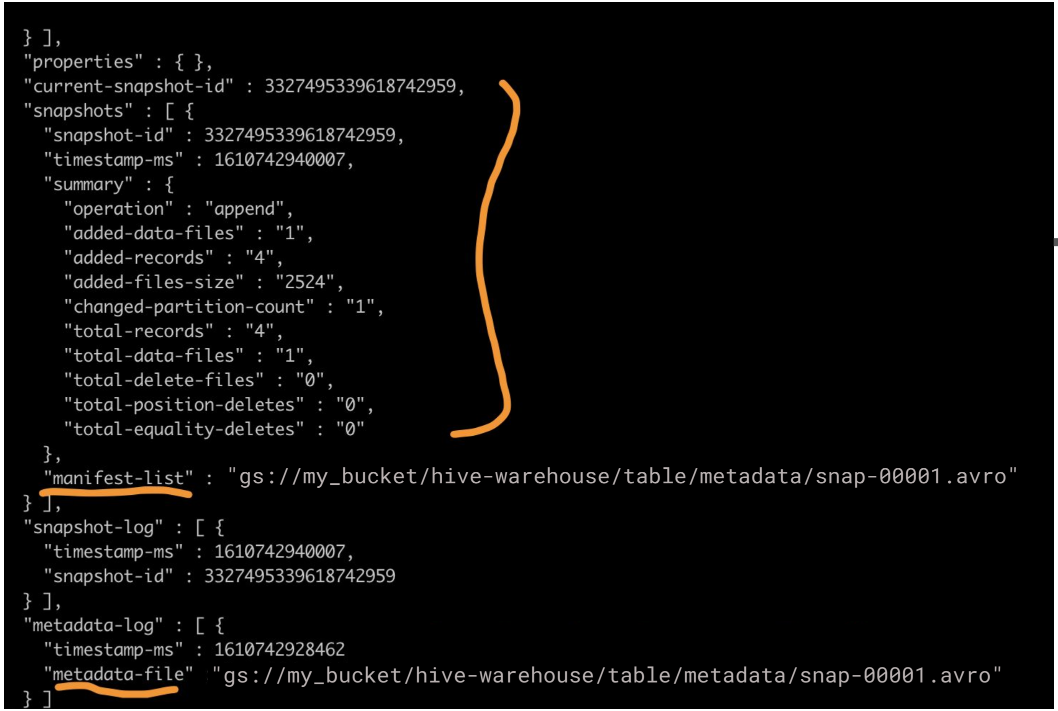
- **Catalog stores a pointer to a metadata file**
  - This files acts as a ledger
  - Has schema information
  - Has partition information
  - Gives us atomic commits



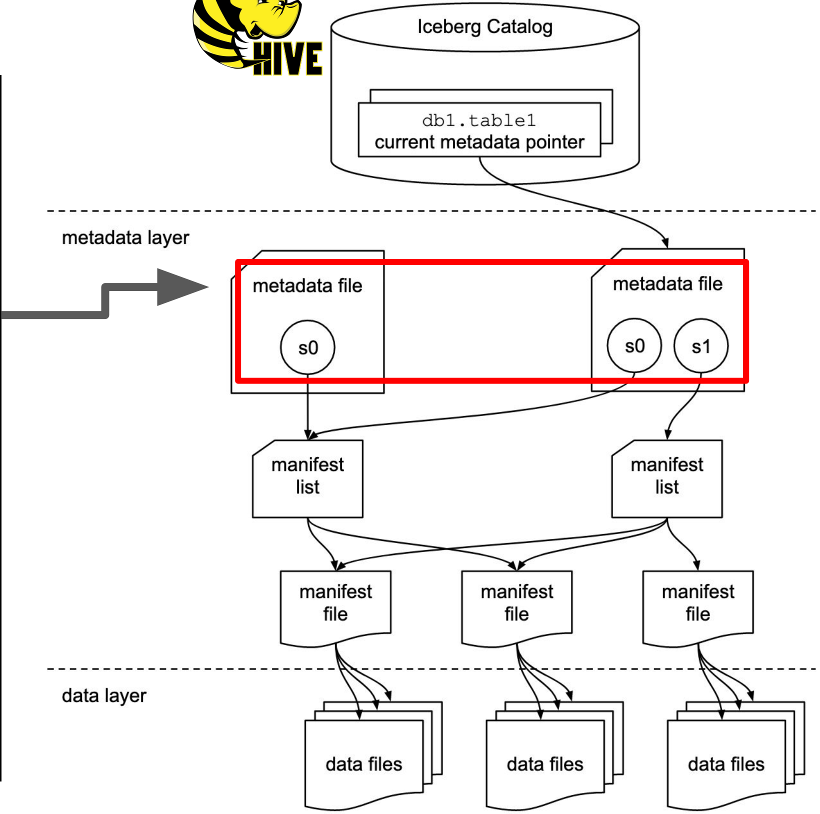


```
+--------+---------------------------+---------------------------------------------------------------
| TBL_ID | PARAM_KEY                 | PARAM_VALUE
+--------+---------------------------+---------------------------------------------------------------
|    292 | EXTERNAL                  | TRUE
|    292 | metadata_location         | gs://my_bucket/hive-warehouse/table/metadata/00001.metadata.json
|    292 | numFiles                  | 1
|    292 | previous_metadata_location | gs://my_bucket/hive-warehouse/table/metadata/00000.metadata.json
|    292 | table_type                | ICEBERG
|    292 | totalSize                 | 1624
|    292 | transient_lastDdlTime     | 1610742932
+--------+---------------------------+---------------------------------------------------------------
7 rows in set (0.02 sec)
```

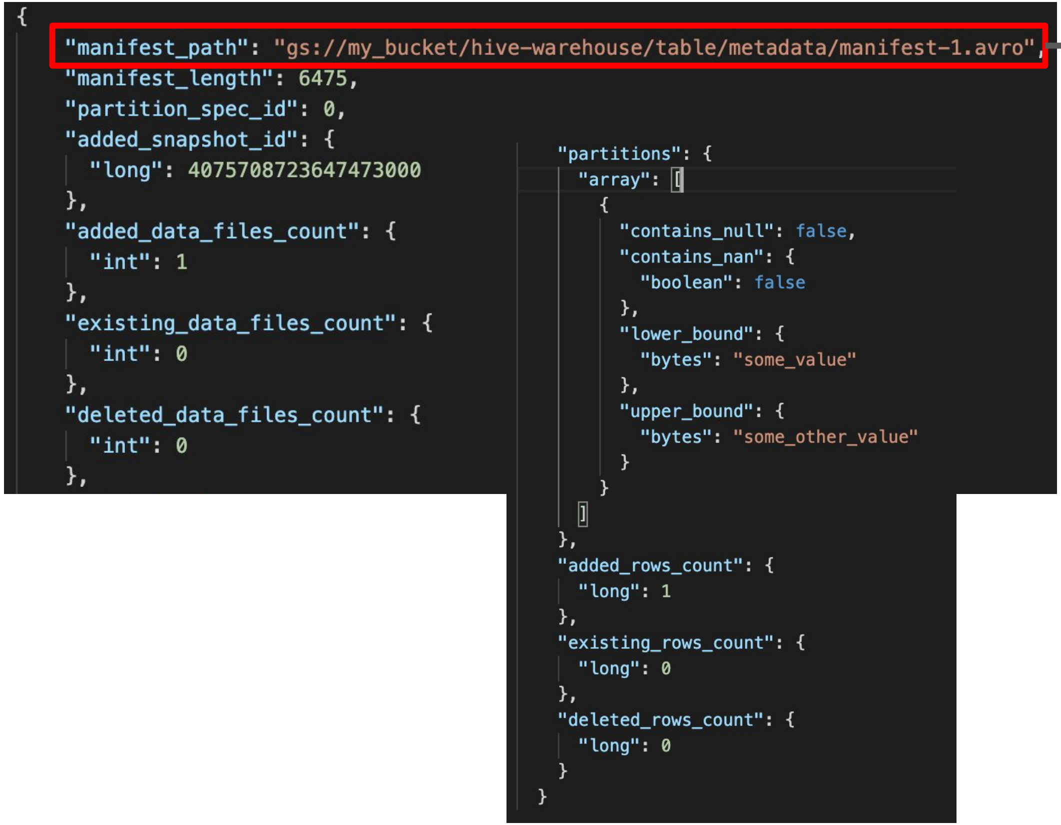
Solution - What is Apache Iceberg?



```
} ],
"properties" : { },
"current-snapshot-id" : 3327495339618742959,
"snapshots" : [ {
  "snapshot-id" : 3327495339618742959,
  "timestamp-ms" : 1610742940007,
  "summary" : {
    "operation" : "append",
    "added-data-files" : "1",
    "added-records" : "4",
    "added-files-size" : "2524",
    "changed-partition-count" : "1",
    "total-records" : "4",
    "total-data-files" : "1",
    "total-delete-files" : "0",
    "total-position-deletes" : "0",
    "total-equality-deletes" : "0"
  },
  "manifest-list" : "gs://my_bucket/hive-warehouse/table/metadata/snap-00001.avro"
} ],
"snapshot-log" : [ {
  "timestamp-ms" : 1610742940007,
  "snapshot-id" : 3327495339618742959
} ],
"metadata-log" : [ {
  "timestamp-ms" : 1610742928462
  "metadata-file" :"gs://my_bucket/hive-warehouse/table/metadata/snap-00001.avro"
} ]
```

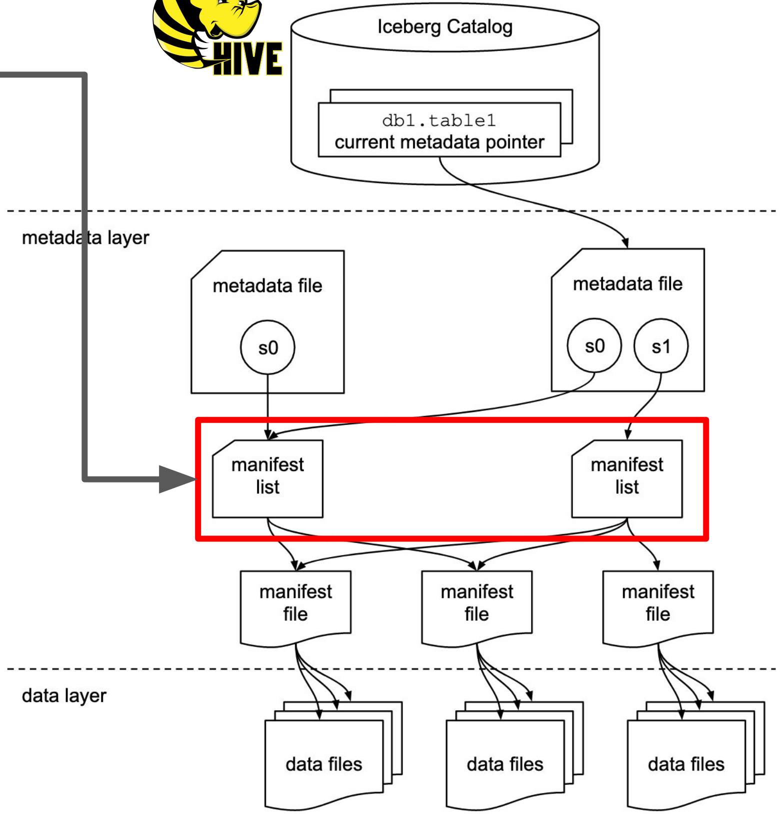


Solution - What is Apache Iceberg?

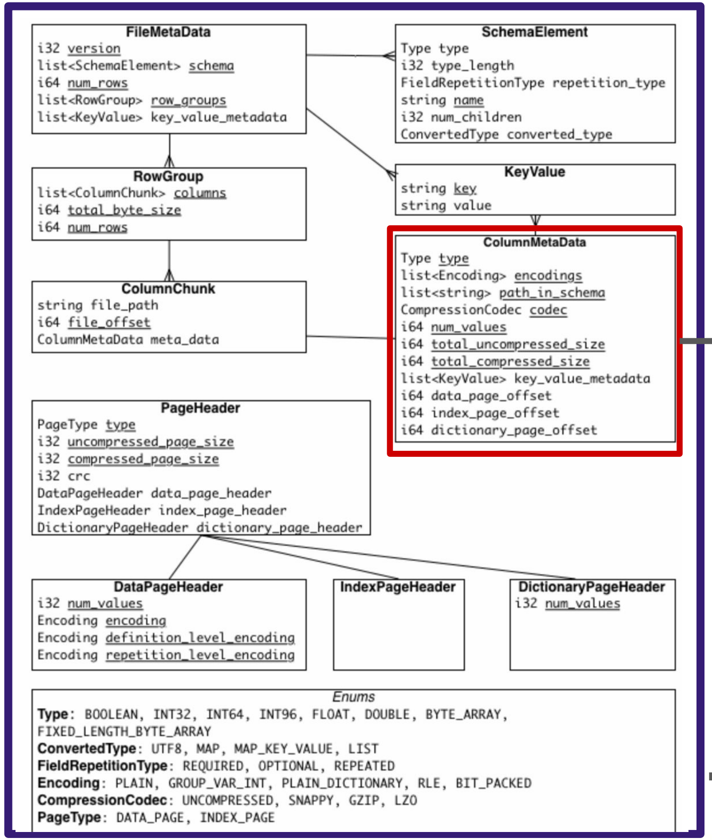


```
{
  "manifest_path": "gs://my_bucket/hive-warehouse/table/metadata/manifest-1.avro",
  "manifest_length": 6475,
  "partition_spec_id": 0,
  "added_snapshot_id": {
    "long": 4075708723647473000
  },
  "added_data_files_count": {
    "int": 1
  },
  "existing_data_files_count": {
    "int": 0
  },
  "deleted_data_files_count": {
    "int": 0
  },
```

```
  "partitions": {
    "array": [
      {
        "contains_null": false,
        "contains_nan": {
          "boolean": false
        },
        "lower_bound": {
          "bytes": "some_value"
        },
        "upper_bound": {
          "bytes": "some_other_value"
        }
      }
    ]
  },
  "added_rows_count": {
    "long": 1
  },
  "existing_rows_count": {
    "long": 0
  },
  "deleted_rows_count": {
    "long": 0
  }
}
```



Solution - What is Apache Iceberg?



FileMetaData
i32 version
list<SchemaElement> schema
i64 num_rows
list<RowGroup> row_groups
list<KeyValue> key_value_metadata

SchemaElement
Type type
i32 type_length
FieldRepetitionType repetition_type
string name
i32 num_children
ConvertedType converted_type

RowGroup
list<ColumnChunk> columns
i64 total_byte_size
i64 num_rows

KeyValue
string key
string value

ColumnChunk
string file_path
i64 file_offset
ColumnMetaData meta_data

ColumnMetaData
Type type
list<Encoding> encodings
list<string> path_in_schema
CompressionCodec codec
i64 num_values
i64 total_uncompressed_size
i64 total_compressed_size
list<KeyValue> key_value_metadata
i64 data_page_offset
i64 index_page_offset
i64 dictionary_page_offset

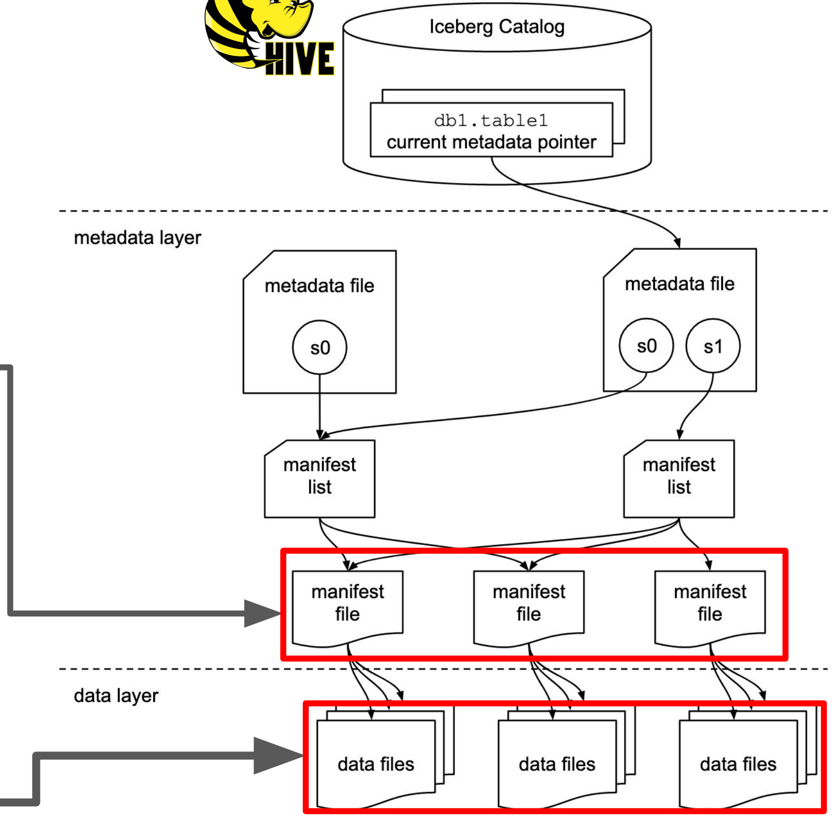
PageHeader
PageType type
i32 uncompressed_page_size
i32 compressed_page_size
i32 crc
DataPageHeader data_page_header
IndexPageHeader index_page_header
DictionaryPageHeader dictionary_page_header

DataPageHeader
i32 num_values
Encoding encoding
Encoding definition_level_encoding
Encoding repetition_level_encoding

IndexPageHeader

DictionaryPageHeader
i32 num_values

Enums
Type: BOOLEAN, INT32, INT64, INT96, FLOAT, DOUBLE, BYTE_ARRAY, FIXED_LENGTH_BYTE_ARRAY
ConvertedType: UTF8, MAP, MAP_KEY_VALUE, LIST
FieldRepetitionType: REQUIRED, OPTIONAL, REPEATED
Encoding: PLAIN, GROUP_VAR_INT, PLAIN_DICTIONARY, RLE, BIT_PACKED
CompressionCodec: UNCOMPRESSED, SNAPPY, GZIP, LZO
PageType: DATA_PAGE, INDEX_PAGE



Iceberg Catalog
db1.table1
current metadata pointer

metadata layer

metadata file
s0

metadata file
s0
s1

manifest list

manifest list

manifest file

manifest file

manifest file

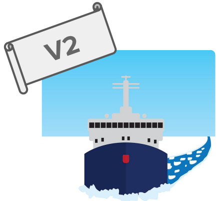
data layer

data files

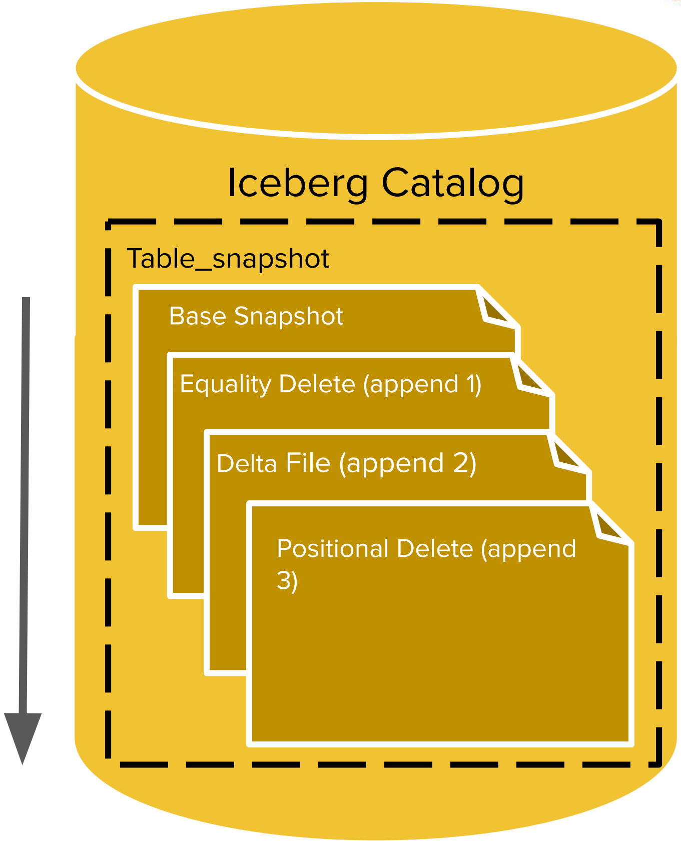
data files

data files

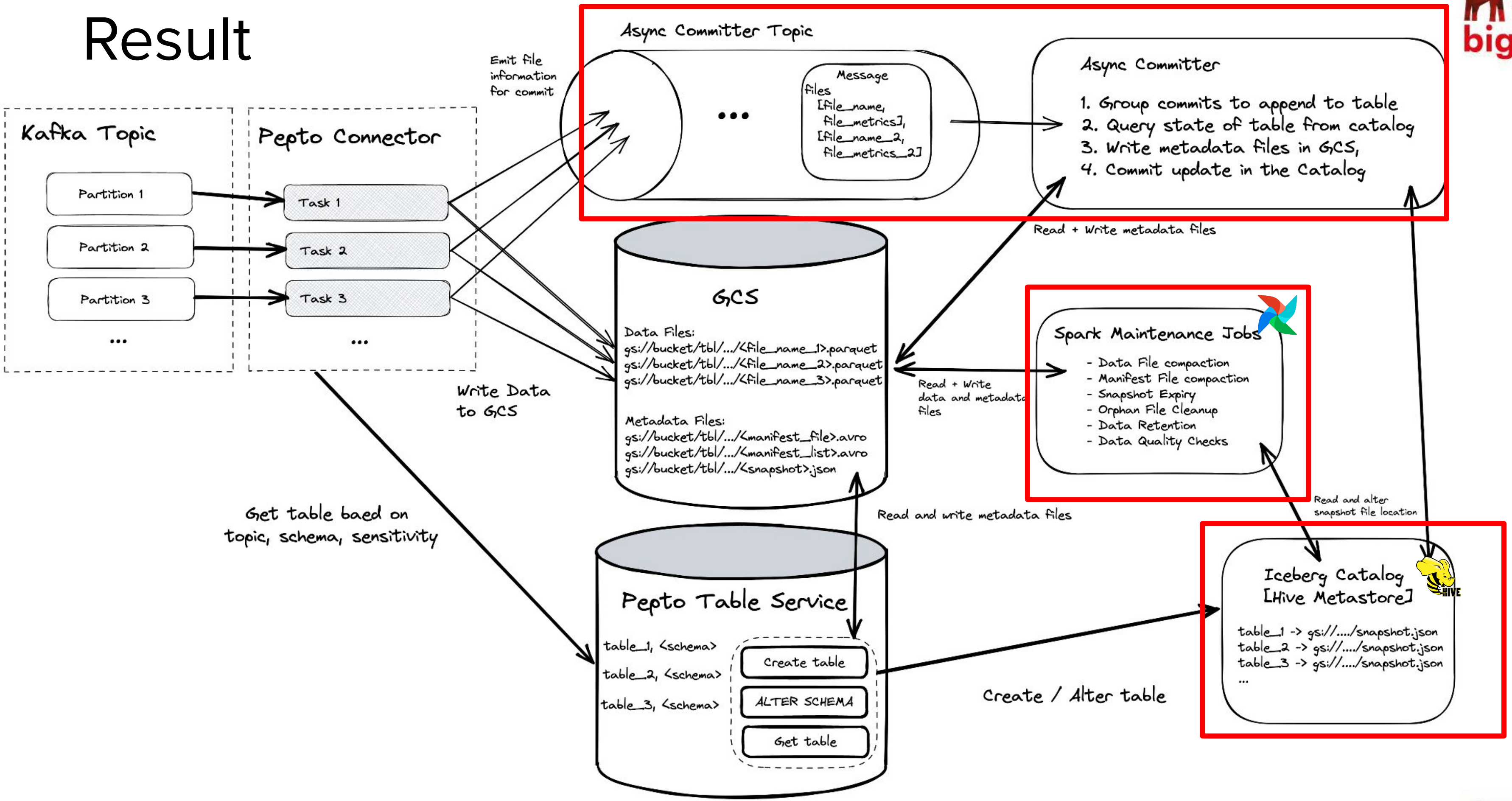
Solution - Promise



- **Merge-on-Read**
  - V2 Spec introduces delete files
    - Positional Delete
    - Equality delete
  - Applied as filters at query time to resolve changes
    - Gets applied to the resultset of your executed query. No full table scan needed!
- **Erik Wright** from Shopify helped write the proposal for Iceberg merge-on-read



Result



Async Committer Topic

Emit file information for commit

Message
files
[file_name,
file_metrics],
[file_name_2,
file_metrics_2]

Async Committer

1. Group commits to append to table
2. Query state of table from catalog
3. Write metadata files in GCS,
4. Commit update in the Catalog

Kafka Topic

Partition 1

Partition 2

Partition 3

...

Pepto Connector

Task 1

Task 2

Task 3

...

Write Data to GCS

Read + Write metadata files

GCS

Data Files:
gs://bucket/tbl/.../<file_name_1>.parquet
gs://bucket/tbl/.../<file_name_2>.parquet
gs://bucket/tbl/.../<file_name_3>.parquet

Metadata Files:
gs://bucket/tbl/.../<manifest_file>.avro
gs://bucket/tbl/.../<manifest_list>.avro
gs://bucket/tbl/.../<snapshot>.json

Read + Write data and metadata files

Spark Maintenance Jobs

- Data File compaction
- Manifest File compaction
- Snapshot Expiry
- Orphan File Cleanup
- Data Retention
- Data Quality Checks

Read and alter snapshot file location

Read and write metadata files

Get table baed on topic, schema, sensitivity

Pepto Table Service

table_1, <schema>
table_2, <schema>
table_3, <schema>

Create table

ALTER SCHEMA

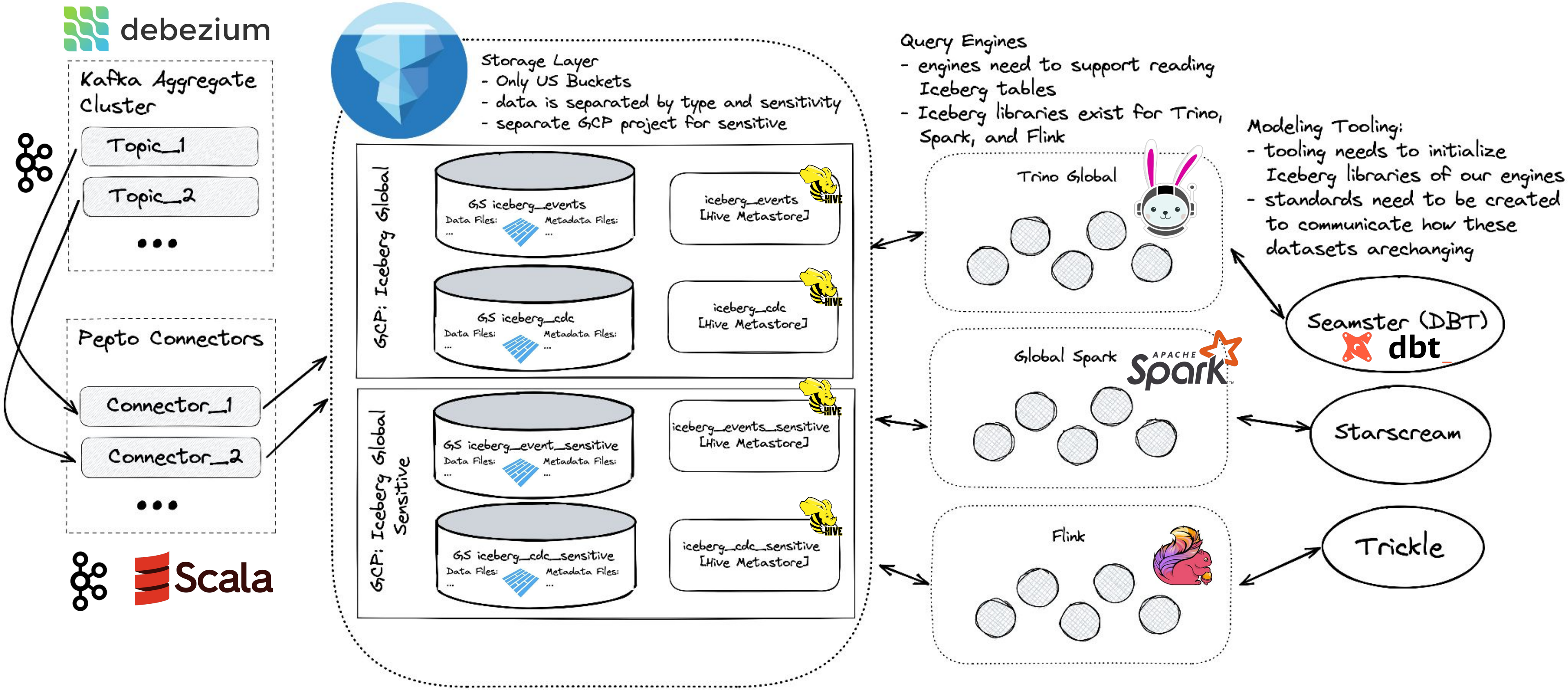
Get table

Create / Alter table

Iceberg Catalog [Hive Metastore]

table_1 -> gs://..../snapshot.json
table_2 -> gs://..../snapshot.json
table_3 -> gs://..../snapshot.json
...

Result - Zoom Out



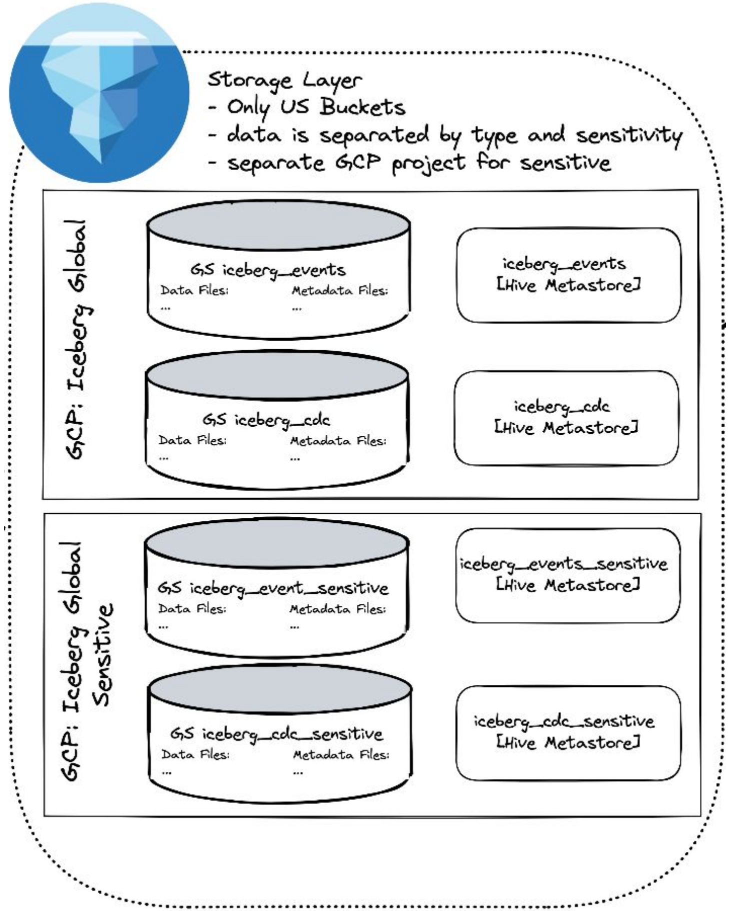
Storage Layer
- Only US Buckets
- data is separated by type and sensitivity
- separate GCP project for sensitive

Query Engines
- engines need to support reading Iceberg tables
- Iceberg libraries exist for Trino, Spark, and Flink

Modeling Tooling:
- tooling needs to initialize Iceberg libraries of our engines
- standards need to be created to communicate how these datasets arechanging

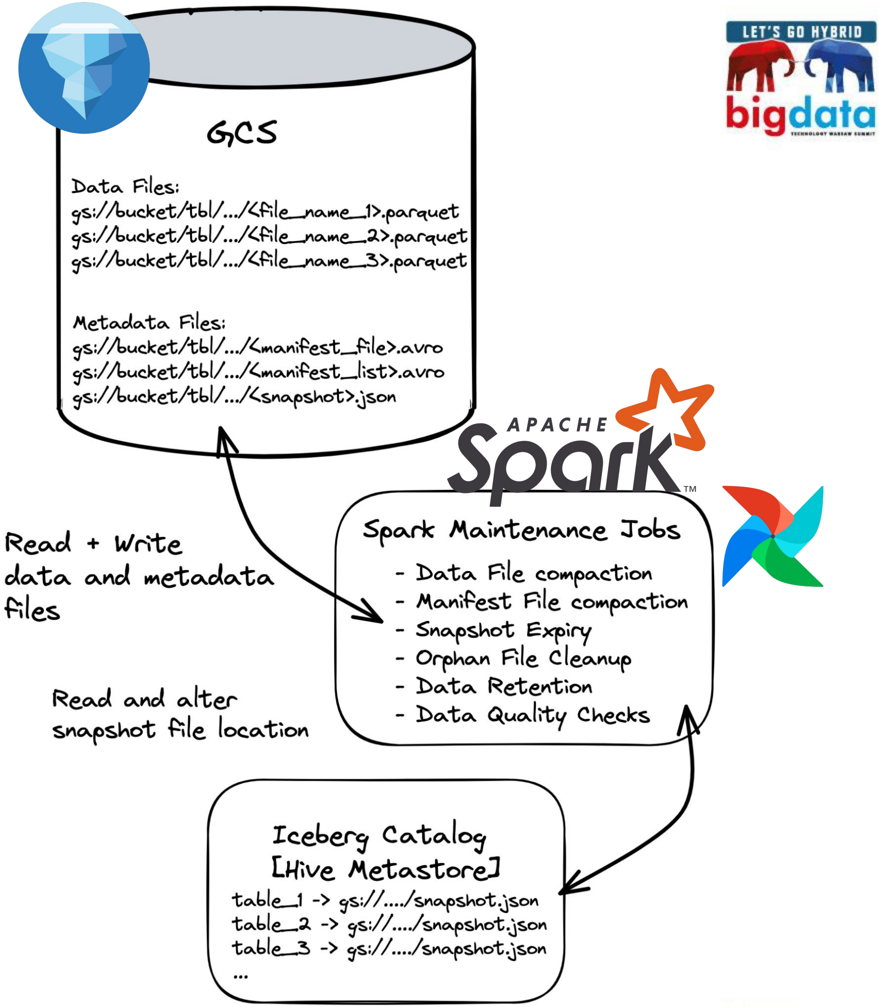
Result - Storage

- **Bucket per Region + Sensitivity**
    - For us this ended up being a bucket per a catalog
    - **Pushback against** additional infra related work for managing + **spinning up + configuring buckets on the fly**
        - Currently buckets are managed by Terraform
    - Future access restricts could be applied via GCS **prefix IAM restrictions**
- **Catalogs group datasets with the same behaviour**
    - CDC vs events vs raw_type_1 etc.



Result - Maintenance

- **Small data + metadata file problem** because of micro-batch processing (streaming)
  - Datafile and manifest compaction solves the small file problem for us
- **Versioning our datasets** by keeping deltas around
- **Privacy**
  - PII is purged after 30 days in GCS
  - Inflight enforcement of data
  - Re-enforcement
- **GCS cleanup**
  - From retention or snapshot expiry, files that are no longer registered to the table are deleted from GCS



GCS

Data Files:
gs://bucket/tbl/.../<file_name_1>.parquet
gs://bucket/tbl/.../<file_name_2>.parquet
gs://bucket/tbl/.../<file_name_3>.parquet

Metadata Files:
gs://bucket/tbl/.../<manifest_file>.avro
gs://bucket/tbl/.../<manifest_list>.avro
gs://bucket/tbl/.../<snapshot>.json

Read + Write data and metadata files

APACHE Spark™

Spark Maintenance Jobs
- Data File compaction
- Manifest File compaction
- Snapshot Expiry
- Orphan File Cleanup
- Data Retention
- Data Quality Checks

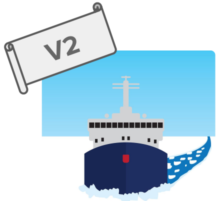
Read and alter snapshot file location

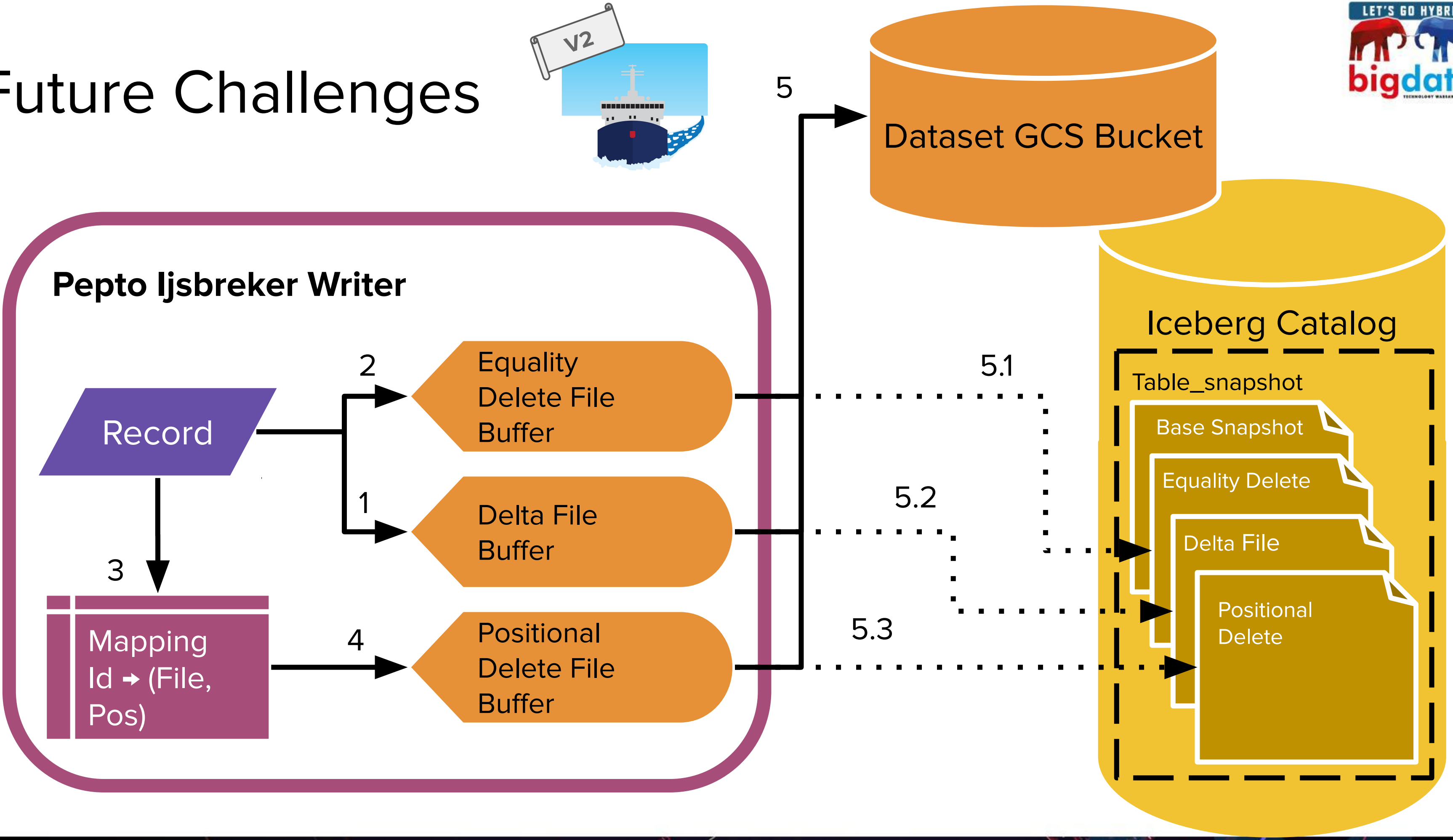
Iceberg Catalog
[Hive Metastore]
table_1 -> gs://..../snapshot.json
table_2 -> gs://..../snapshot.json
table_3 -> gs://..../snapshot.json
...

Reflection

- **I wish we did a bucket per a dataset**
  - A very fine grain of separation
  - Easier to implement specific restrictions / functionality
    - Regulated industry (ex SOX)
  - More upfront work but would have accommodated additional use cases
- **Writing your own engine to utilize Iceberg can be hard** (Kafka connect is not supported)
  - We ended up building our own version of many of the concepts that you see in the Flink Iceberg Connector (async committing to)
  - **Flink connector did not fully exist when we started**
  - Documentation is not great leading to a **divergence between the online docs and their Java API**

Future Challenges





Dataset GCS Bucket

Pepto Ijsbreker Writer

Record

Equality Delete File Buffer

Delta File Buffer

Mapping Id ➔ (File, Pos)

Positional Delete File Buffer

Iceberg Catalog

Table_snapshot

Base Snapshot

Equality Delete

Delta File

Positional Delete

Thank you!