

Zero-Copy Iceberg Migrations with Apache Beam

Ahmed Abualsaud

Migrating to Better Table Semantics

Parquet Lake



- Column pruning
- Row group filtering
- Columnar storage
- Compression efficiency
- Stored schema

Migrating to Better Table Semantics

Parquet Lake



- Column pruning
- Row group filtering
- Columnar storage
- Compression efficiency
- Stored schema



Iceberg Table



- ACID
- Snapshots + time travel
- Schema evolution
- Partition evolution
- Hidden partitioning
- Better query planning



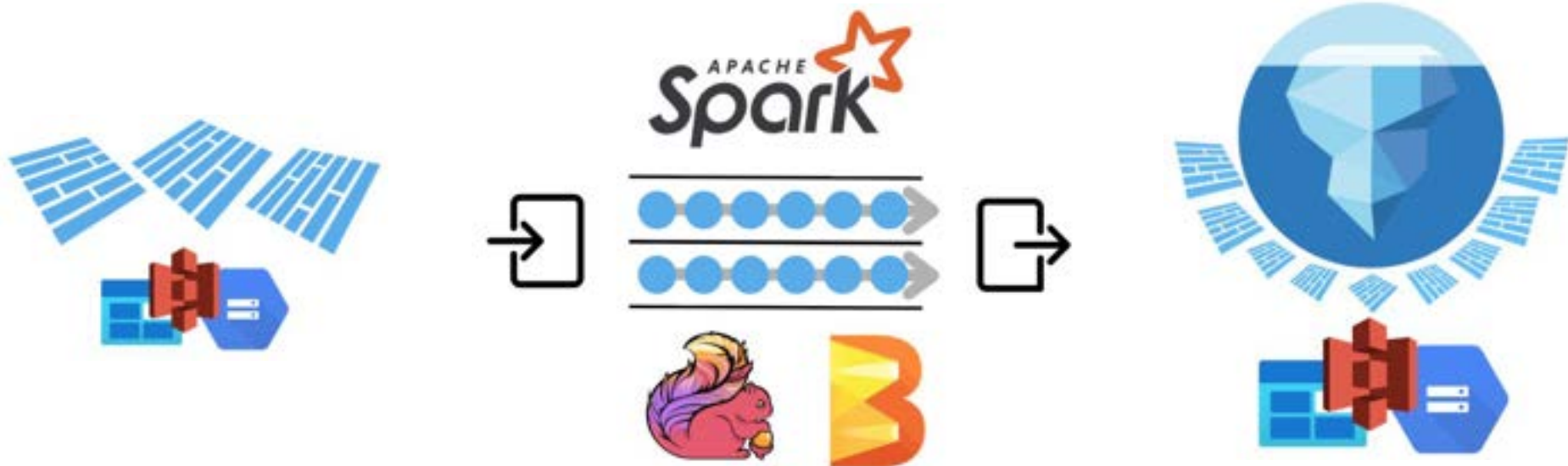
The Hidden Rewrite Tax



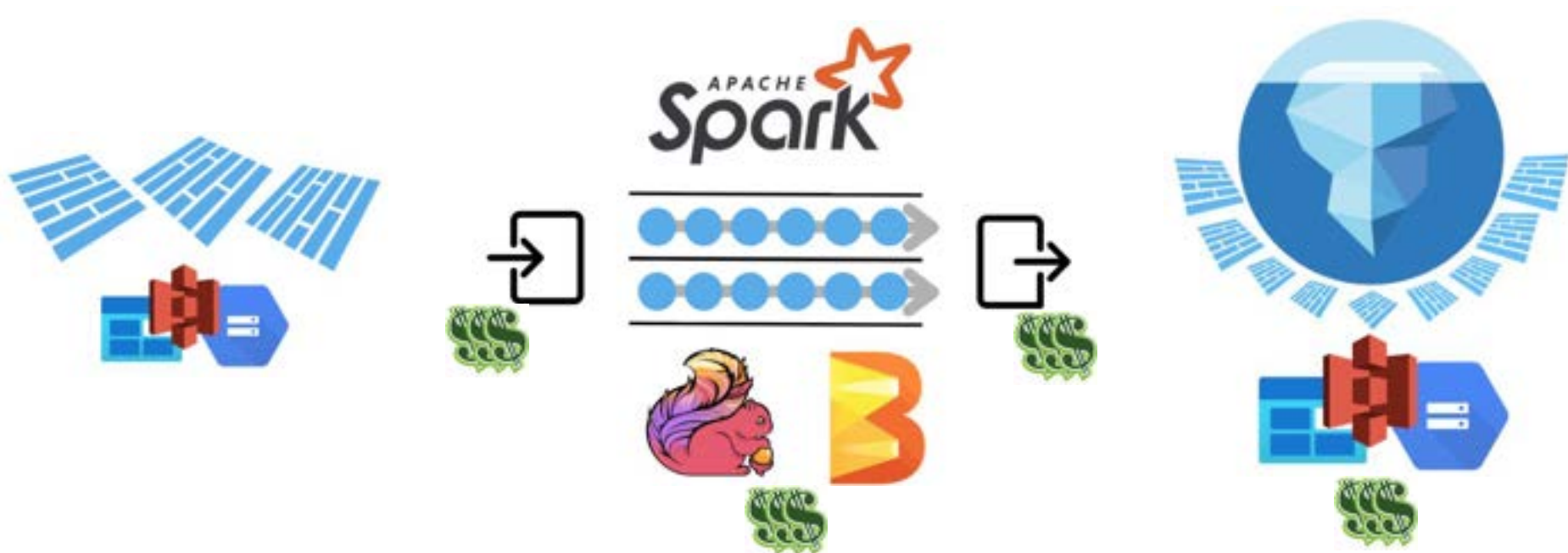
The Hidden Rewrite Tax



The Hidden Rewrite Tax



The Hidden Rewrite Tax



Zero-Copy Migration

keep Parquet files in place



Zero-Copy Migration

create Iceberg metadata on top

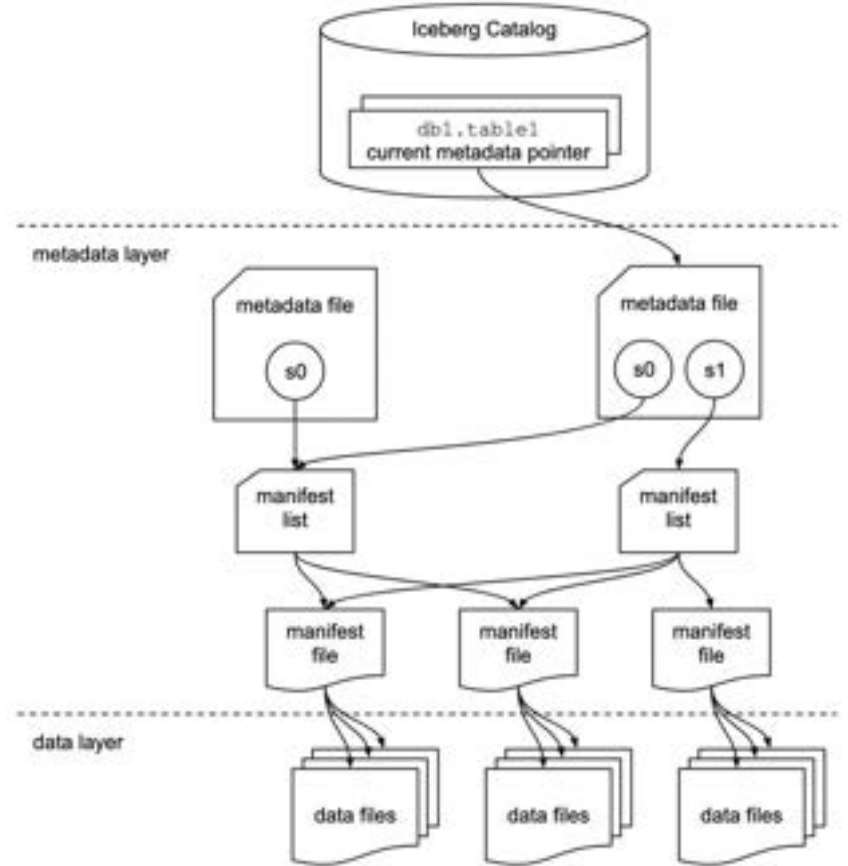
keep Parquet files in place



Zero-Copy Migration

create Iceberg metadata on top

keep Parquet files in place

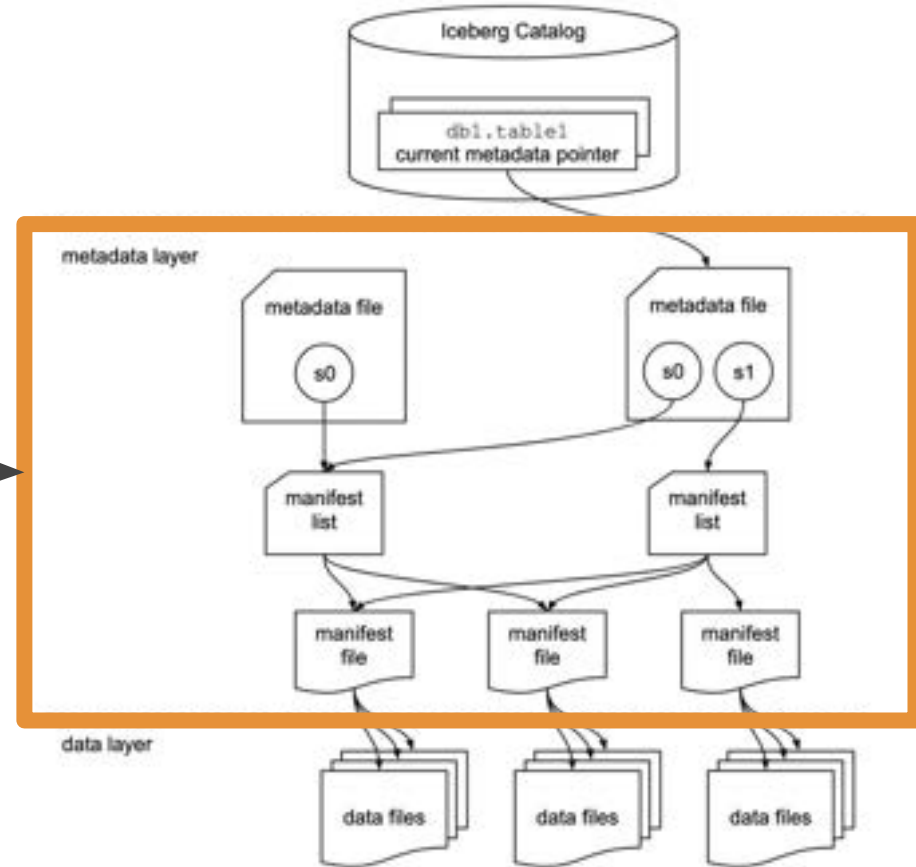


Zero-Copy Migration

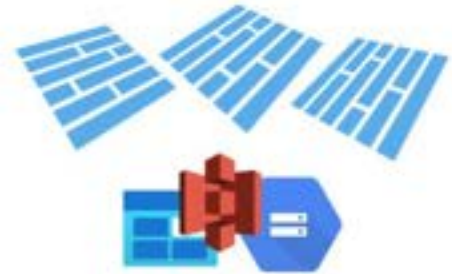
create Iceberg metadata on top



keep Parquet files in place



Migration model



Migration model



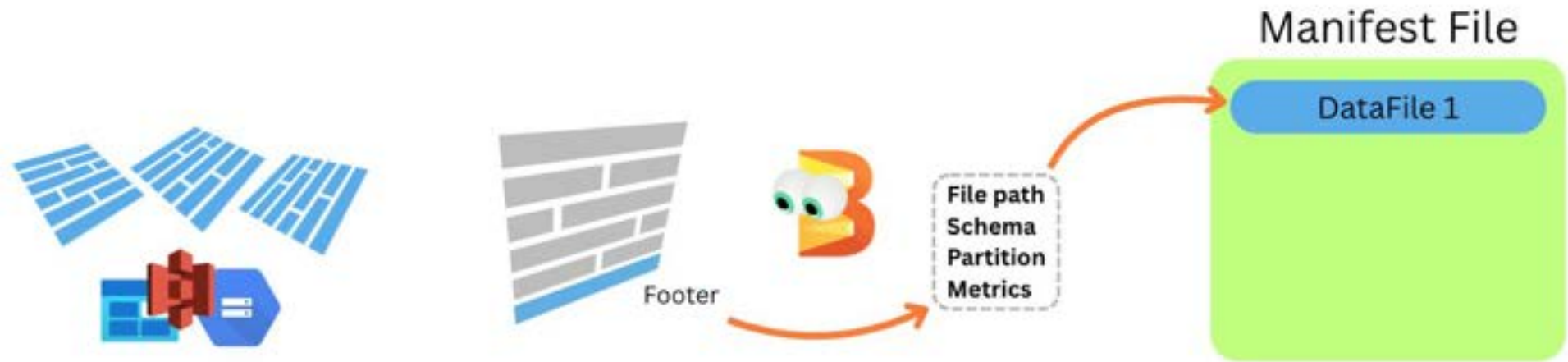
Migration model



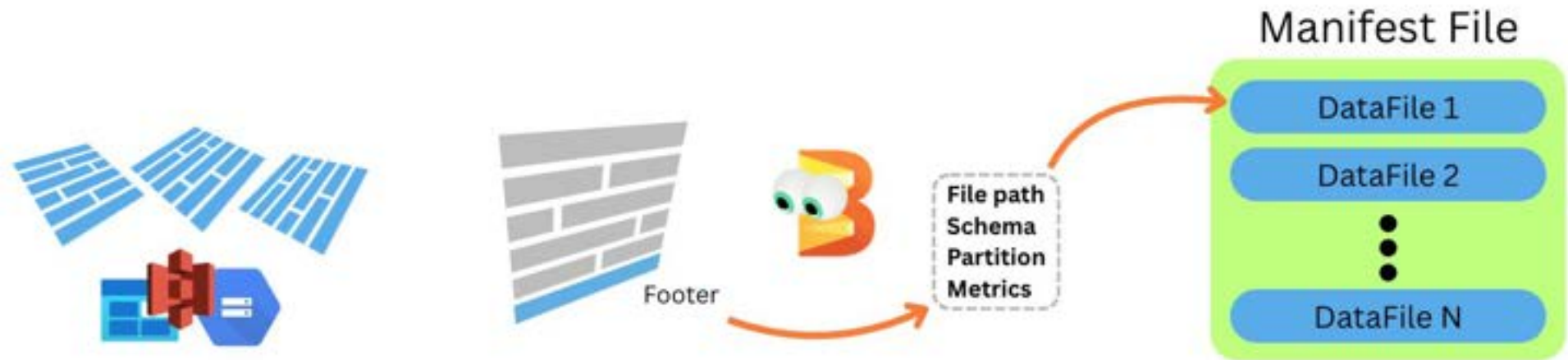
Migration model



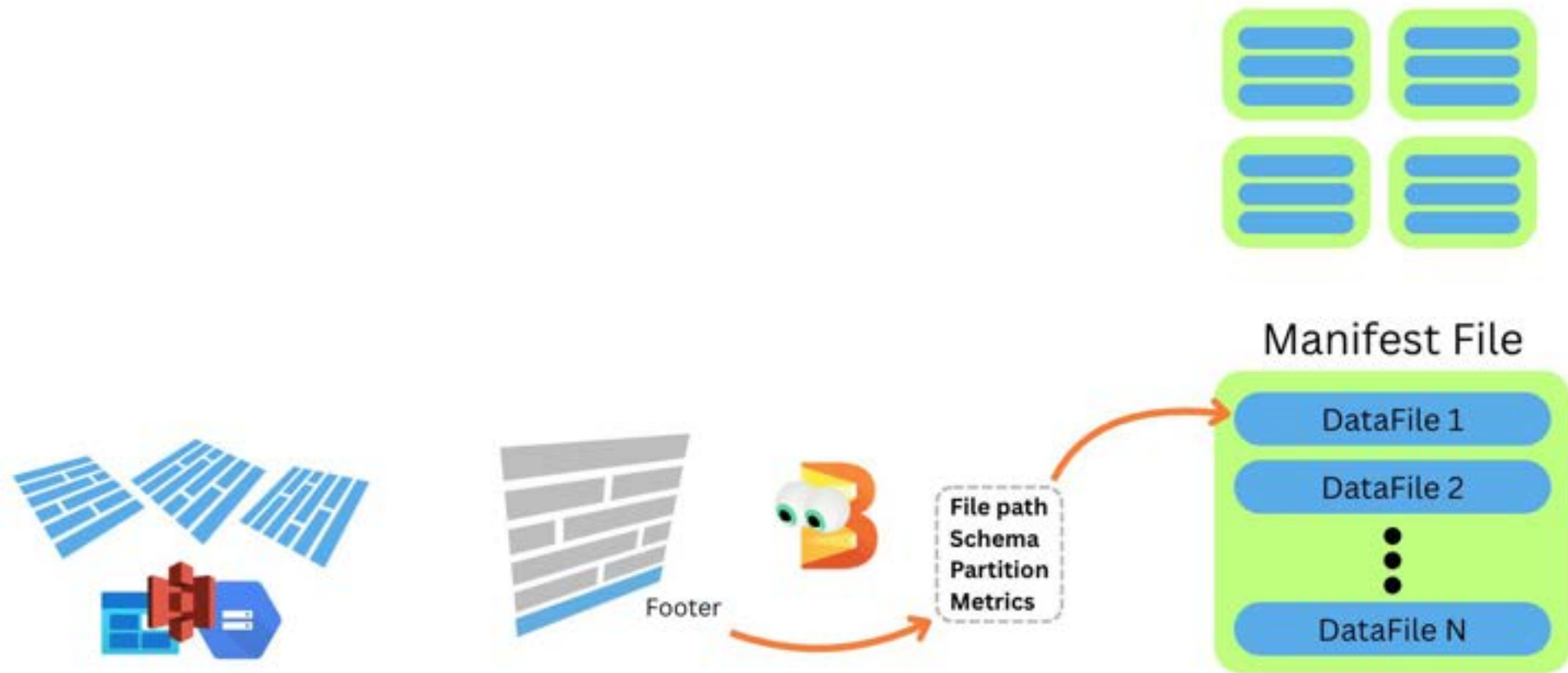
Migration model



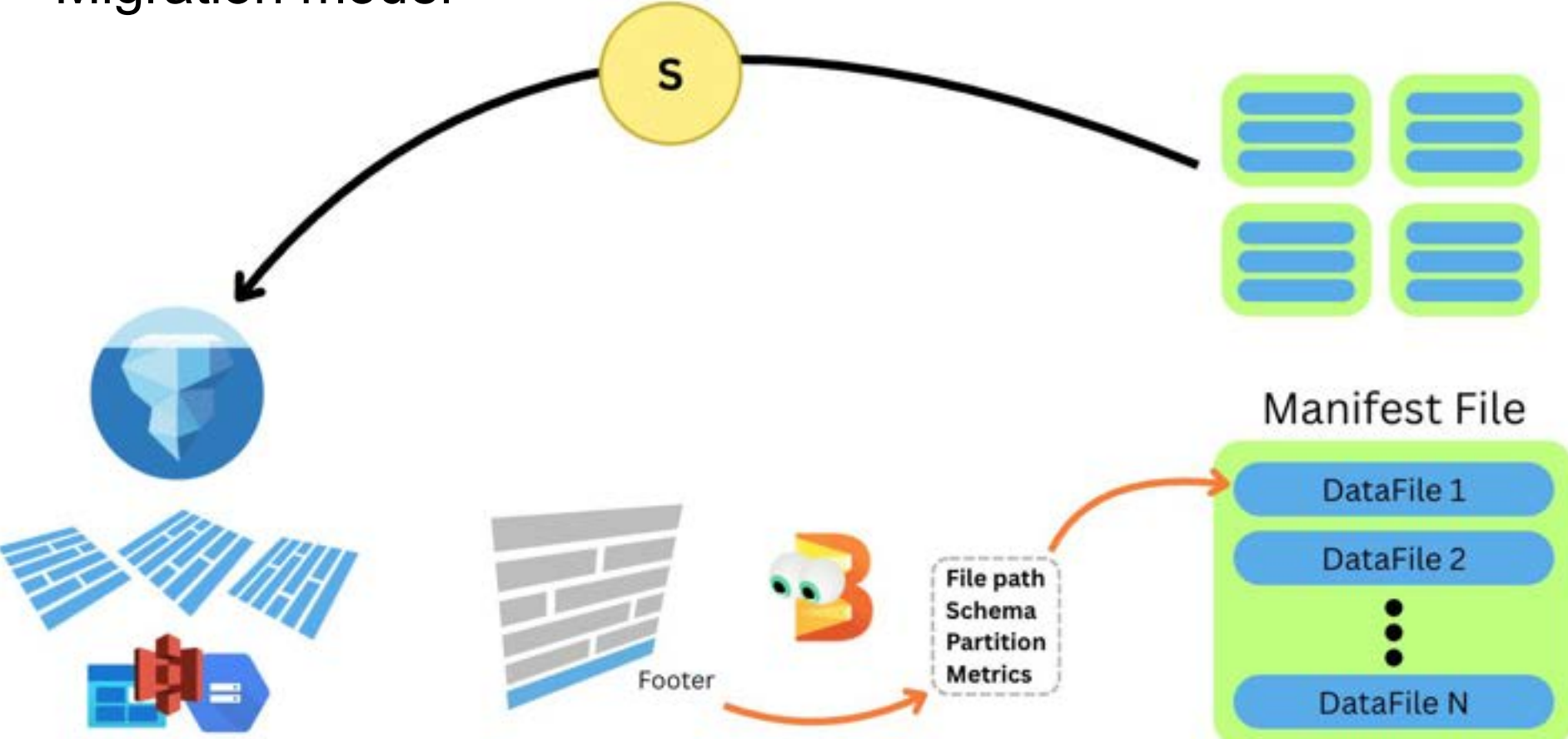
Migration model



Migration model



Migration model



What Beam adds

Batch File Registration

Streaming File Registration



What Beam adds

Batch File Registration

Streaming File Registration



Demo Time

<https://github.com/ahmedabu98/zero-copy-iceberg/>

Inferring Partitions

Two options: path-based or metrics-based

Inferring Partitions

Path-based (we trust the directory)

Partition spec:

- identity(col1)
- year(col2)

Inferring Partitions

Path-based (we trust the directory)

Partition spec:

- identity(col1)
- year(col2)

s3://bucket/data-dir/table/col1=abc/col2=2025/datafile.parquet

Inferring Partitions

Path-based (we trust the directory)

Partition spec:

- identity(col1)
- year(col2)

s3://bucket/data-dir/table/col1=abc/col2=2026/datafile.parquet


location_prefix

Inferring Partitions

Path-based (we trust the directory)

Partition spec:

- identity(col1)
- year(col2)

s3://bucket/data-dir/table/**col1=abc/col2=2026**/datafile.parquet

location_prefix

Partition: [**col1=abc, col2=2026**]

Inferring Partitions

Metrics-based (prove the file belongs to one partition)

Inferring Partitions

Metrics-based (prove the file belongs to one partition)



```
col1 min = "abc"  
     max = "abc"
```

```
col2 min = 2026  
     max = 2026
```

Inferring Partitions

Metrics-based (prove the file belongs to one partition)



```
col1 min = "abc"  
     max = "abc"
```

```
col2 min = 2026  
     max = 2026
```

Partition: [col1=abc, col2=2026]

Inferring Partitions

Metrics-based (prove the file belongs to one partition)



```
col1 min = "abc"  
      max = "abc"
```

```
col2 min = 2019  
      max = 2026
```

Inferring Partitions

Metrics-based (prove the file belongs to one partition)

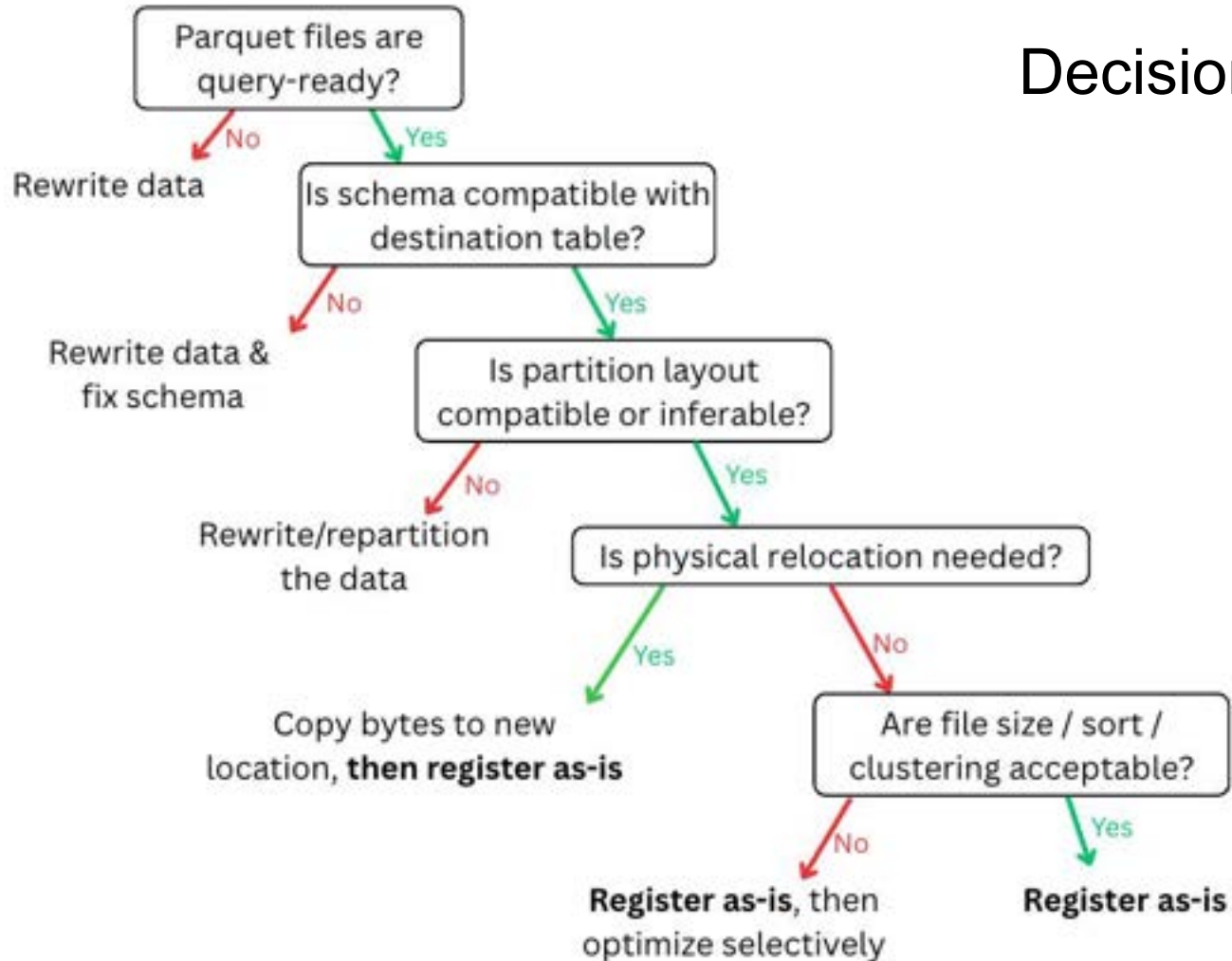


```
col1 min = "abc"  
     max = "abc"
```

```
col2 min = 2019  
     max = 2026
```

DataFile spans multiple partitions. **Sent to dead-letter-queue**

Decision Tree



Final Tips

- **Files are small or unsorted?**

Register first then optimize selectively later.

Final Tips

- **Files are small or unsorted?**

Register first then optimize selectively later.

- **Need physical relocation?**

Do byte-for-byte transfer to new location then register with zero-copy.

Faster and less costly than a full row-level read and write.

Final Tips

- **Files are small or unsorted?**

Register first then optimize selectively later.

- **Need physical relocation?**

Do byte-for-byte transfer to new location then register with zero-copy.
Faster and less costly than a full row-level read and write.

- For production use-cases, make sure the destination table is pre-created with a schema

Ahmed Abualsaud

QUESTIONS?

Contact Info:

[linkedin.com/in/ahmedabu98](https://www.linkedin.com/in/ahmedabu98)

github.com/ahmedabu98

a.abualsaud98@gmail.com