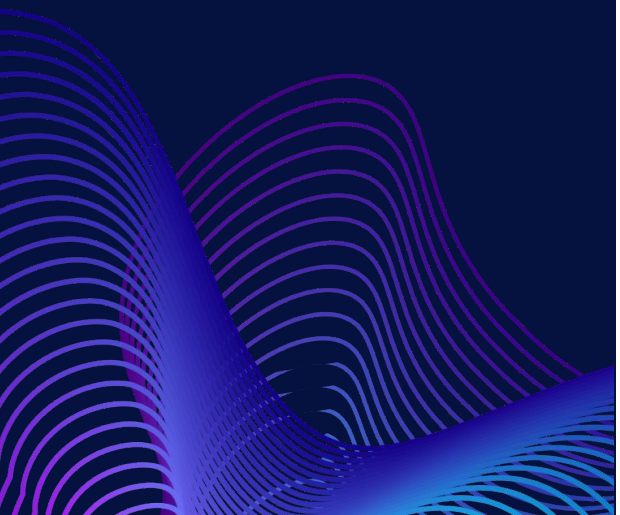
Lessons Learned Making Open Table Formats Enterprise-Ready

James Malone - Senior Manager, Snowflake

Agenda

Here's what I will be covering today.



1. Who I am and why I am here
2. Customer demand for open formats
3. Choosing a format - Apache Iceberg
4. Lessons we've learned along the way

James Malone

Product, Snowflake

I am privileged to lead a team at **Snowflake** focused on all storage, and patterns related to them (lake, etc.)

I've led product teams working with and participating in *open source*, at enterprise scale, in previous lives:

- **PM @ Google** - managed processing and workflows
- **Eng Director @ Disney** - television data platform
- **PM @ Amazon** - building internal tools

Open formats, please!

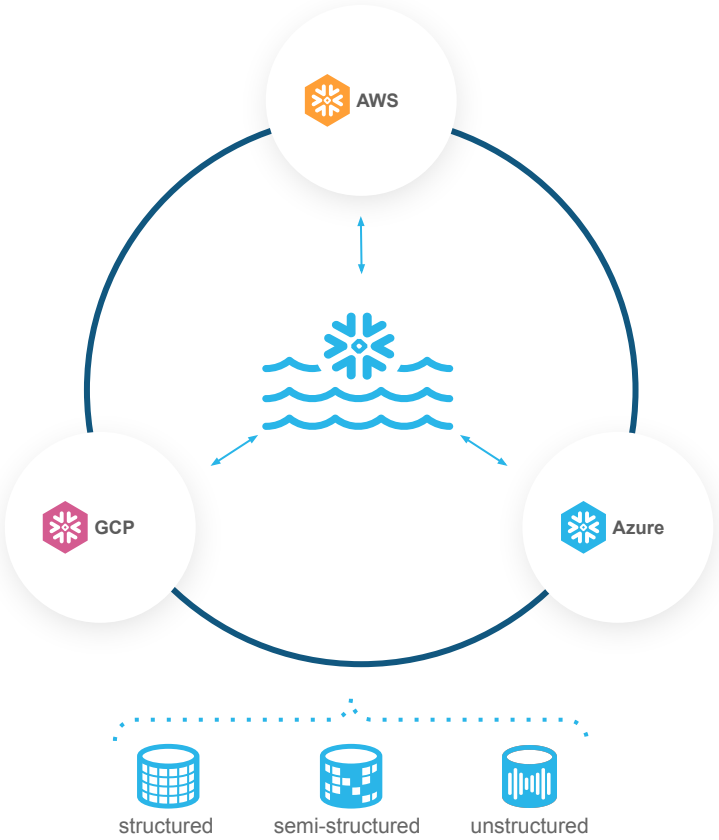
Some customers have told us open formats really matter to them - here is what we have heard from them.

Data lakes, meshes, fabrics, and beyond
Customers choose blob storage because it works

There are several popular storage (architectural) patterns that share a set of storage architecture similarities, including:

- Data Lake
- Data mesh
- Data fabric
- and others

These patterns advocate for storing individual data files in cloud blob stores, e.g. AWS S3, Azure Storage, and Google Cloud Storage.



What we have heard from customers

CONTROL

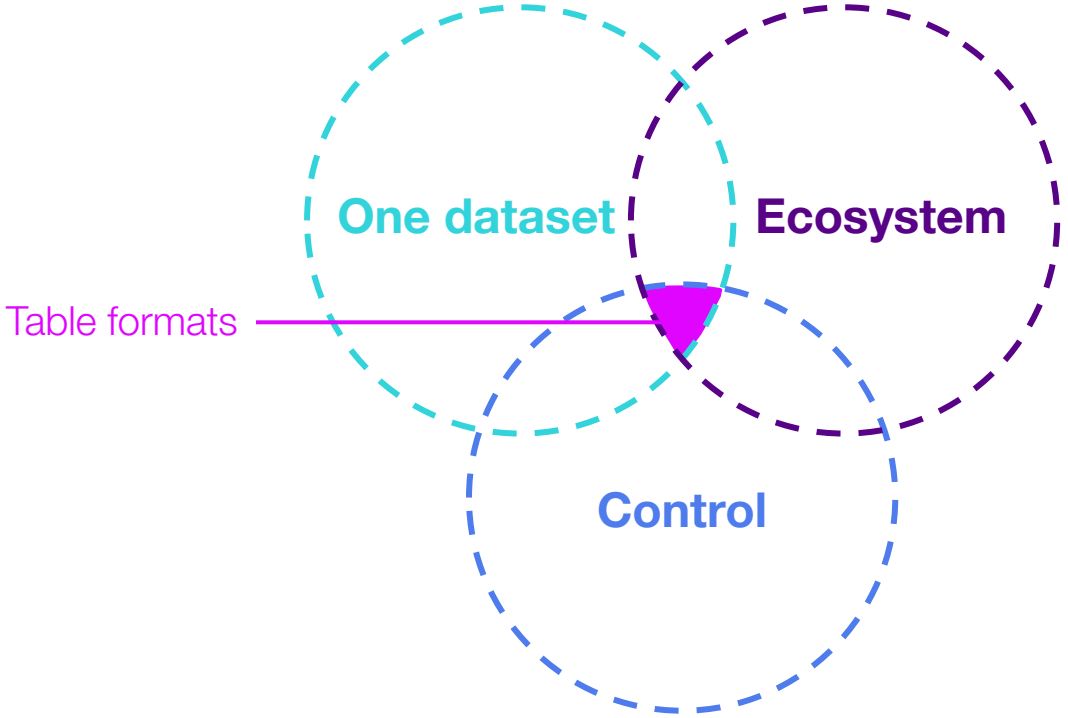
ONE DATASET

ECOSYSTEM

Table formats

Table formats can address these 3 areas:

- **One dataset**: Saved in blob storage
- **Ecosystem**: Tools can work together
- **Control**: Some formats are open



Shared needs, but different priorities (an example)

COMMON NEEDS

- Flexibility
- Security
- Stability
- Latest features
- Governance
- Consistency
- Cost control
- …

DW DEVELOPER

1. Consistency
2. Security
3. Stability
4. Governance
5. Cost control
6. Flexibility
7. Latest features

OPEN FORMAT ENGINEER

1. Flexibility
2. Latest features
3. Cost control
4. Consistency
5. Security
6. Stability
7. Governance

Choosing a table format

Out of the many table formats available, we chose to focus on Apache Iceberg; here's why.

Iceberg is a great project based on 3 key pillars

Vibrant Ecosystem

The number of tools and platforms supporting Iceberg is growing; broad participation from those working with Iceberg.

Velocity

All metrics show a healthy and growing project - number and breadth of comitters, community pull requests, releases, etc.

Interoperability

Iceberg is designed to work across many file formats and engines, increasing customer options and choice.

It’s not just us seeing the growth of Iceberg

Snowflake

Expanding the Data Cloud with Apache Iceberg

TechTarget

Apache Iceberg rising for new cloud data lake platforms

Datanami

Snowflake, AWS Warm Up to Apache Iceberg

Tabular

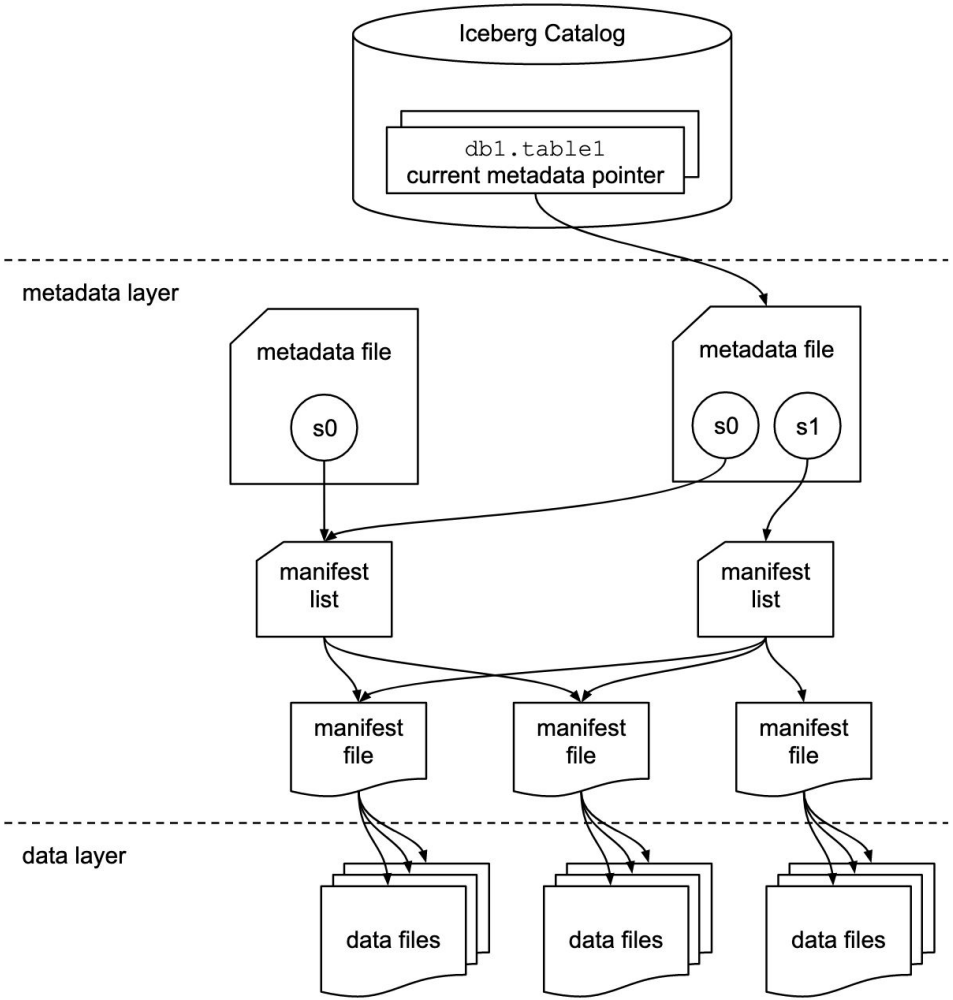
Introducing Tabular

Dremio

Apache Iceberg Becomes Industry Open Standard with Ecosystem Adoption

Great minds think alike

	Iceberg Tables	Snowflake Tables
ACID	✓	✓
Time Travel	✓	✓
Data Mutation	✓	✓
Schema Evolution	✓	✓
Expressive SQL	✓	✓
Compaction	✓	✓
Excellent performance	✓	✓
Multi-cloud	✓	✓
Fast velocity	✓	✓

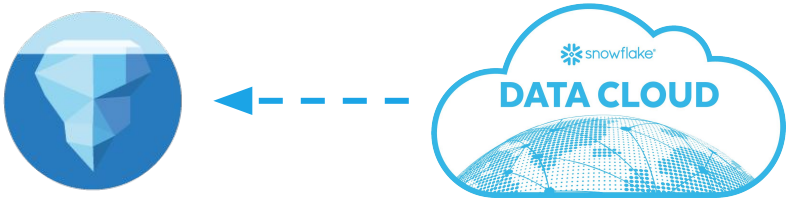


Iceberg spec

The Iceberg spec is interesting because it's:

- Approachable
- Similar to Snowflake
- Versioned with enterprise feature velocity

iceberg.apache.org/spec



External Tables for Iceberg*

We want to make it easy to use Iceberg with Snowflake, regardless of whatever architectural pattern you choose.

Snowflake can query data in place, without moving the table data or existing metadata

Enables easy access to tables with data in customer-supplied storage buckets

Utilizes Iceberg metadata for efficient pruning and performance

*In Development

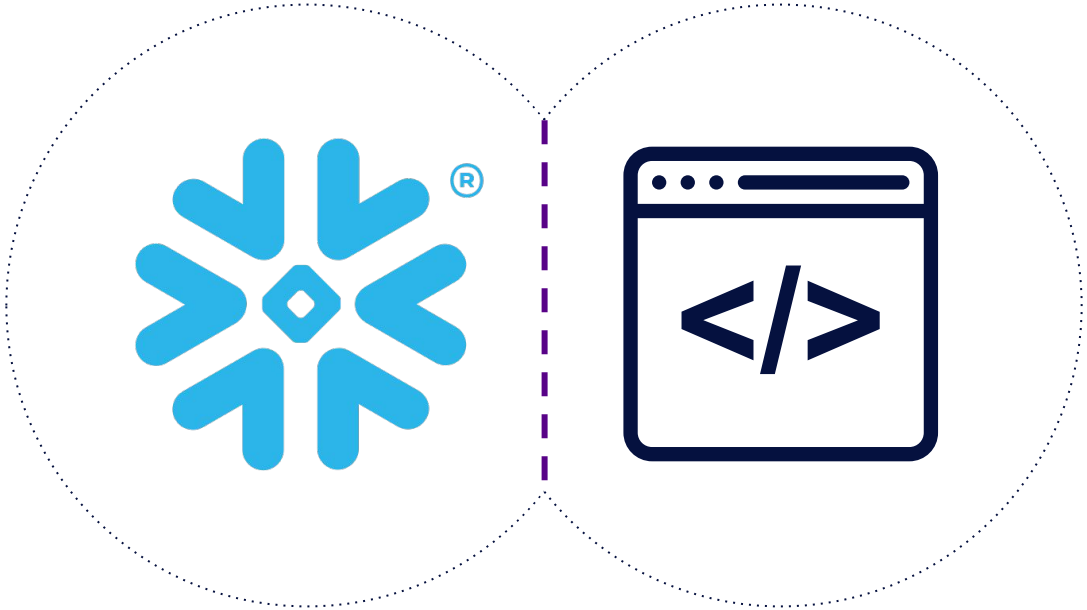
Lessons learned

Making a table format *enterprise ready* can be tricky - here is what we have learned along the way.

Be transparent and add value

- There’s enough confusion out there, don’t add to it
- Customers want us to be clear about what we are doing on top of a format:
  - What benefits are we providing?
  - Are we bringing unique capabilities?
  - Do we offer a different support model?
- Pick a few things and do them well
- Open source is a two-way street

It’s important to be clear about what’s open; what’s not



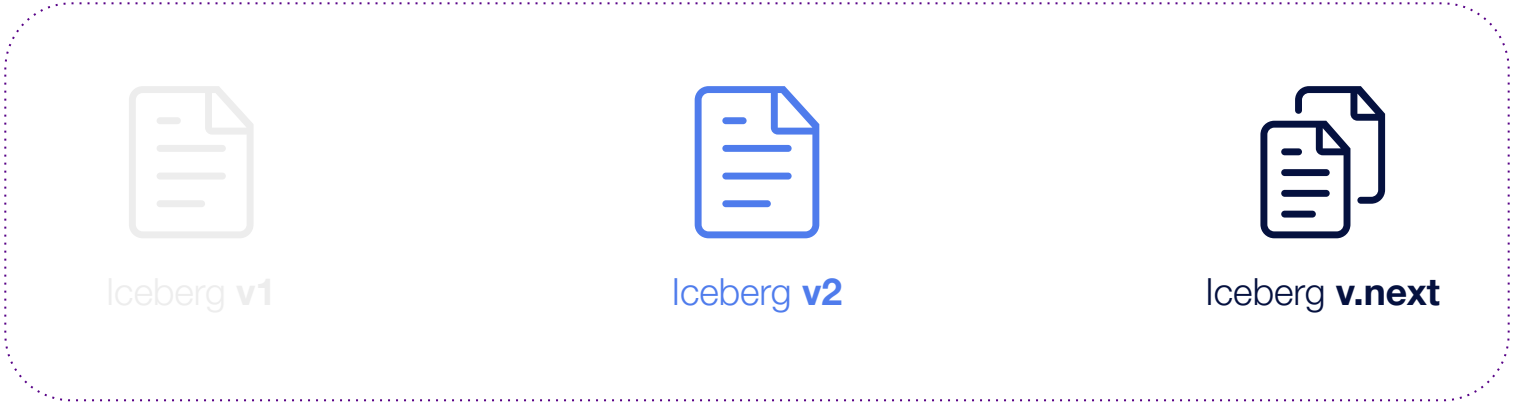
Know when you're adopting something and why

We need to be mindful of what we adopt, and why, because:

- Adding to our stack can increase complexity
- There's never a one-size-fits-all product solution, so we are designing for some set of users - who are they?
- We are extending conceptual and tangible support that our customers will rely on to make decisions
- Product experiences should be deliberate, and match our goals, rather our goals matching our technical choices

"At Snowflake, we think about first principles, about desired outcomes, about intended and unintended consequences and, most importantly, we're always focused on what is best for our customers."

Do not think of OSS with a moment in time mentality



Supporting **all** of these, not just **one**

We need to design thinking ahead, following where the community is going.

Some customers will want stability, while others will want the latest and greatest thing. We need to make sure we are aligned for and help shape what's next.

Get involved, and advocate

Be customer first

Bridge the gaps

Better the community

Subsurface LIVE

The Cloud Data Lake Conference

Thank You