The Ins & Outs of Data Lakehouse Versioning at the File, Table, and Catalog Level

Presented by Alex Merced

Alex Merced

Developer Advocate, Dremio

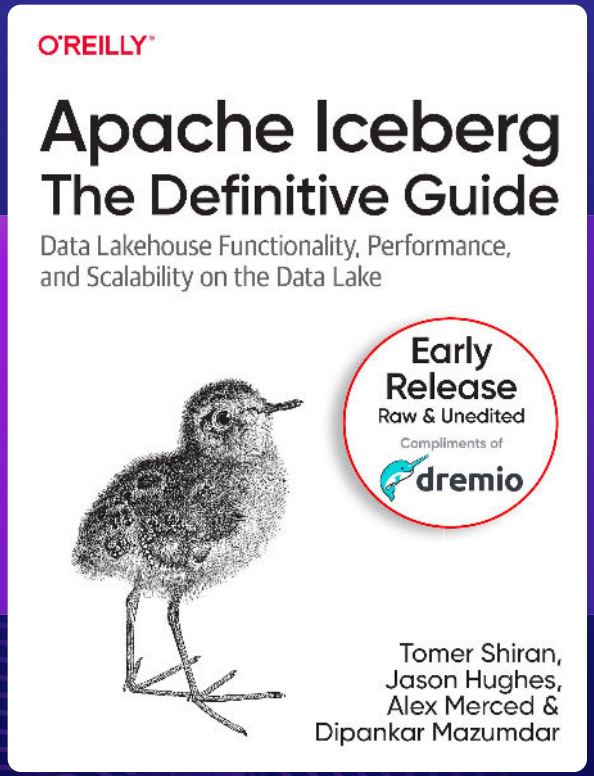
Alex Merced is a developer advocate for Dremio, a developer, and a seasoned instructor with a rich professional background. Having worked with companies like GenEd Systems, Crossfield Digital, CampusGuard, and General Assembly.

Alex is a co-author of the O'Reilly Book "Apache Iceberg: The Definitive Guide." With a deep understanding of the subject matter, Alex has shared his insights as a speaker at events including Data Day Texas, OSA Con, P99Conf and Data Council.

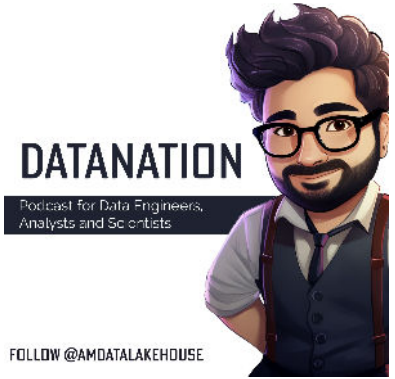
Driven by a profound passion for technology, Alex has been instrumental in disseminating his knowledge through various platforms. His tech content can be found in blogs, videos, and his podcasts, Datanation and Web Dev 101.

Moreover, Alex Merced has made contributions to the JavaScript and Python communities by developing a range of libraries. Notable examples include SencilloDB, CoquitoJS, and dremio-simple-query, among others.

Apache Iceberg: The Definitive Guide

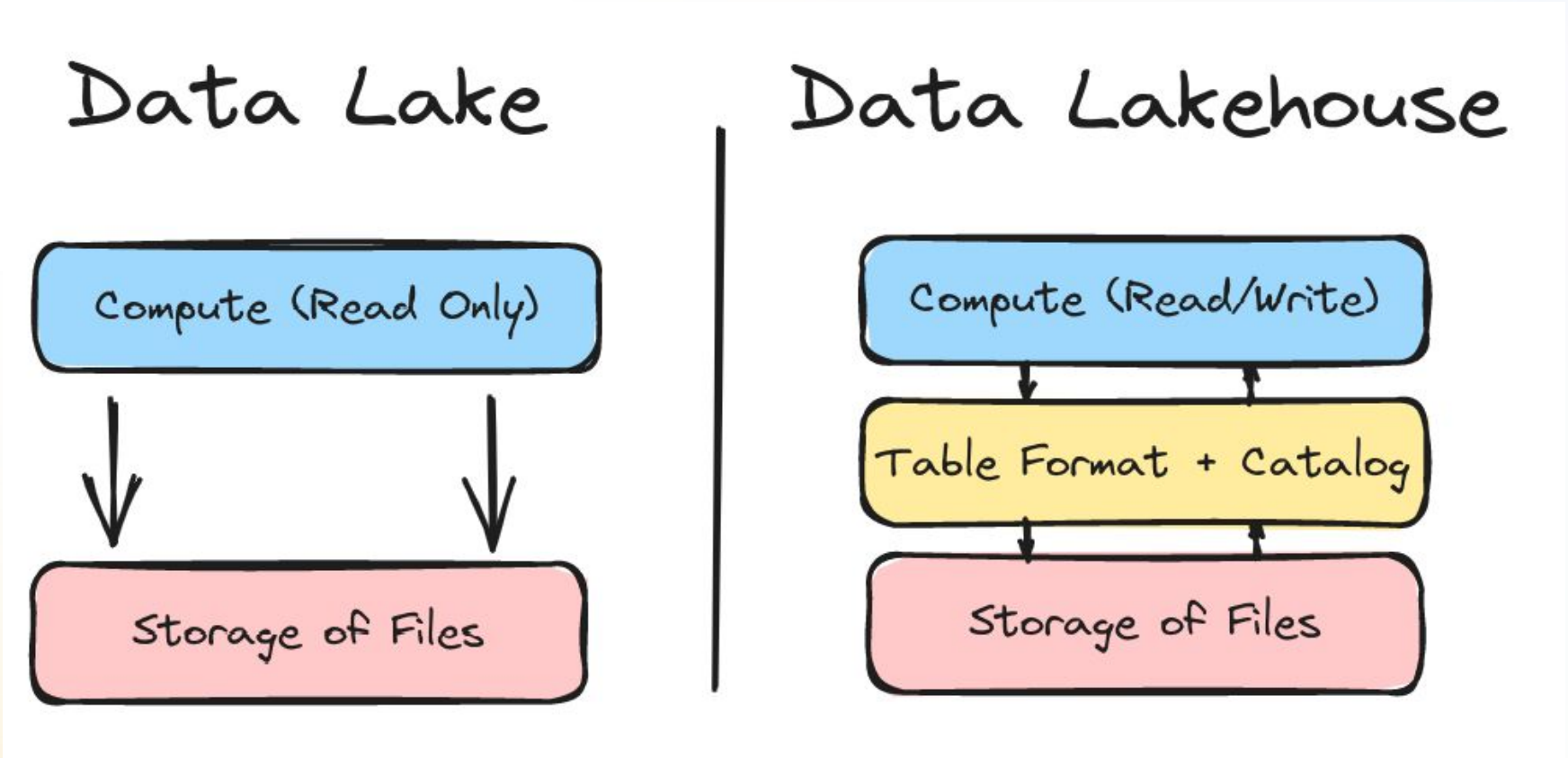


Podcasts



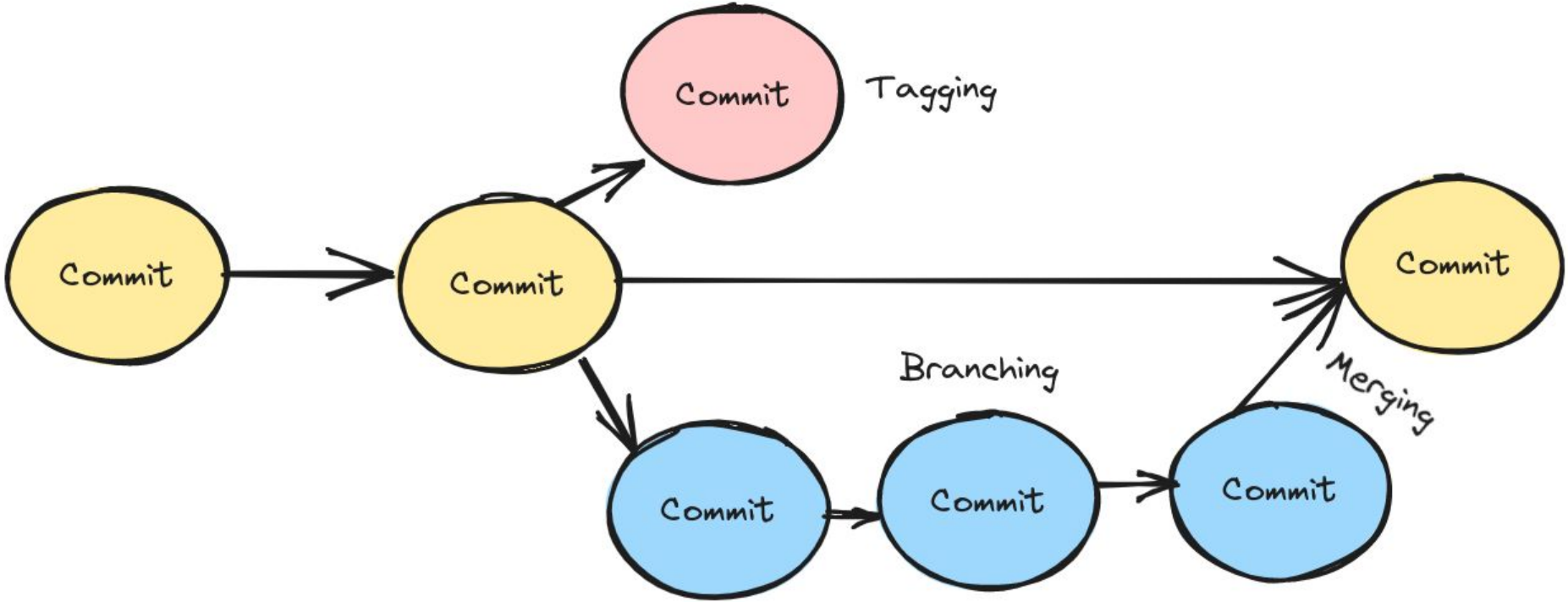
Subscribe on Spotify/iTunes

What is a Data Lakehouse



What is Data as Code?

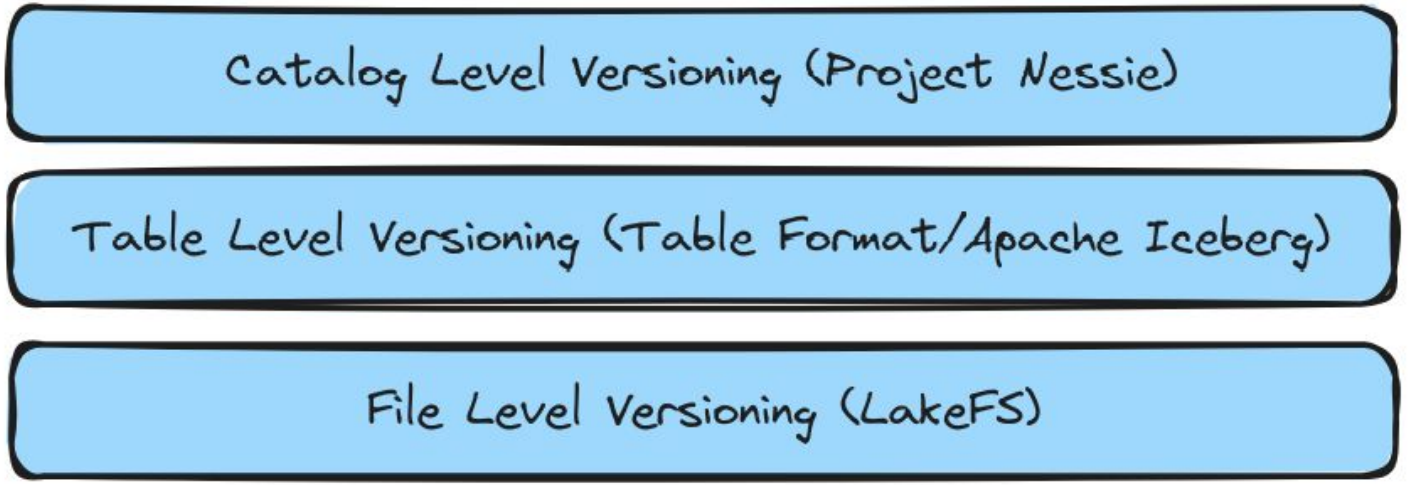
Bringing Code Like Practices like CI/CD & Versioning to Data



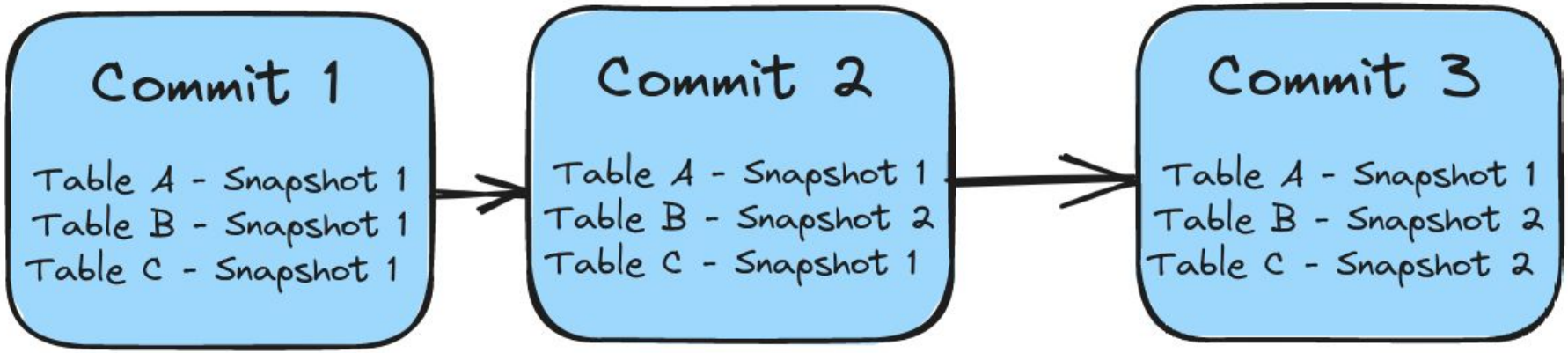
Possible Benefits of Versioning

- Isolation
- Multi-Table Transactions
- Rollbacks
- Reproducability
- Consistency, Quality and Data Validation

Levels of Lakehouse Versioning



Catalog Level Versioning with Nessie



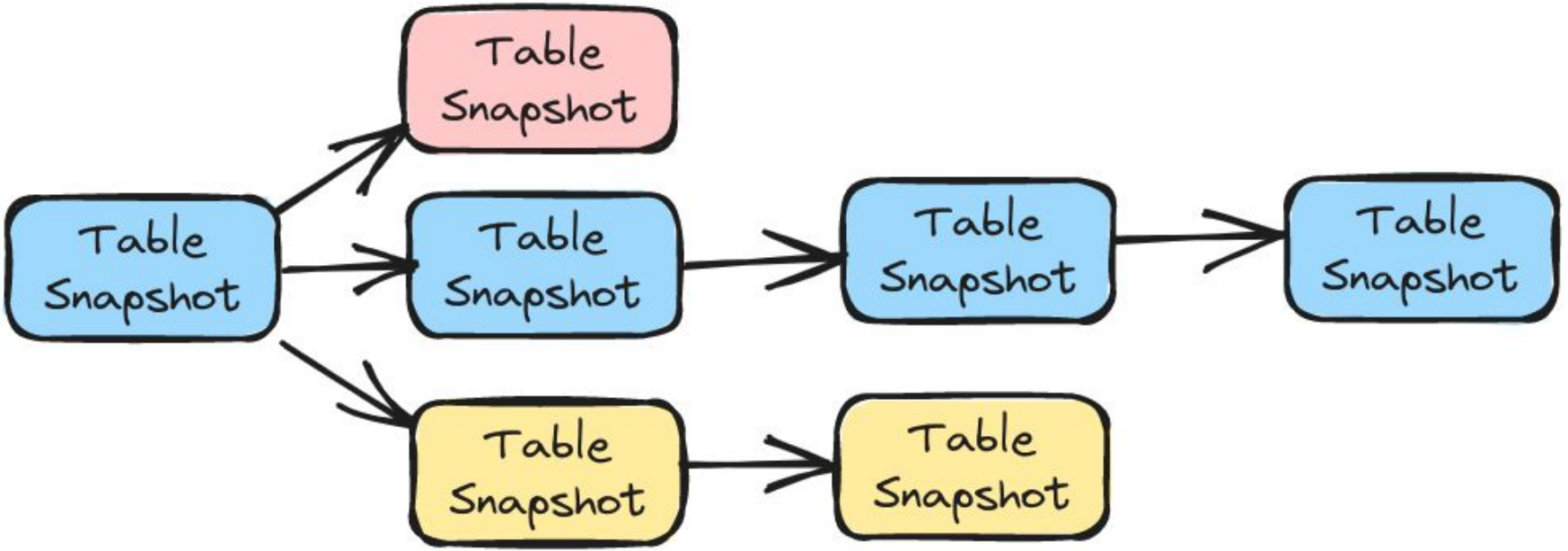
Pros

Multi-Table Transactions
Multi-Table Tagging
Multi-Table Rollbacks
Branching & Merging
All Operations Via SQL, REST API, Python
Cloud Managed Service (Dremio Arctic)
Lightweight Architecture
Storage & Cloud Agnostic

Cons

Currently Iceberg Tables & views Only
(Maybe Delta Lake in Future)
Precludes other Iceberg Catalogs
Requires a Running a Service whether
self-deployed or managed

Table Level Commits



Branch with 7 Day Life and Max 2 Commits

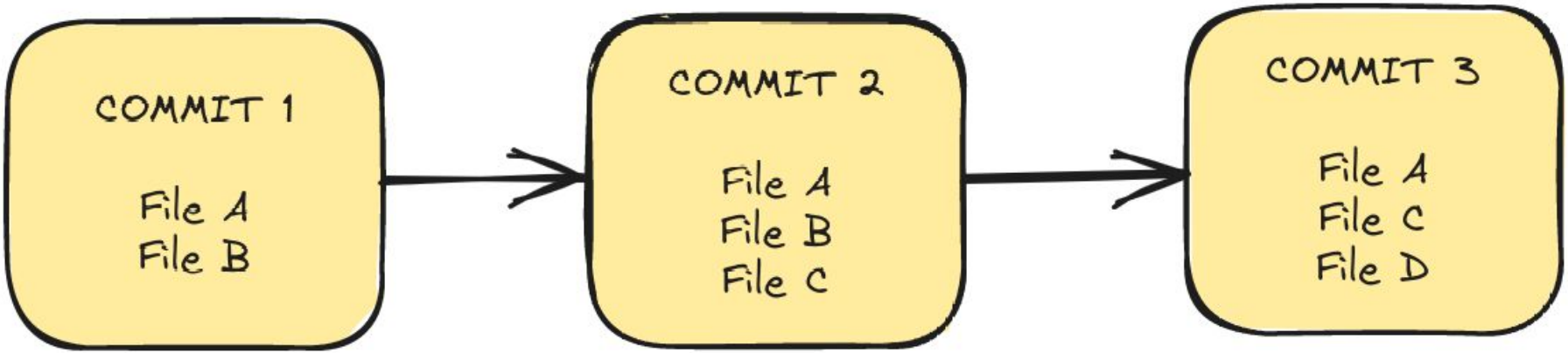
Pros

Single Table Transactions
Single Table Tagging
Single Table Rollbacks
Single Table Branching & Merging
Some Operations via SQL
Works with all catalogs
Requires No other Service
Storage & Cloud Agnostic

Cons

Apache Iceberg Only
No Multi-Table Transactions (Yet)
No Multi-Table Rollbacks
No Multi-Table Tagging

File Level Versioning



Pros

Multi-File Transactions
Multi-File Tagging
Multi-File Rollbacks
Works with Delta, Hudi and No Format

Cons

Must use LakeFS Catalog for Iceberg

Can't use SQL to
create or merge branches

Requires S3 Compatible
Object Storage (No Hadoop)

Summary

	Catalog	Table	File
Islolation/Branching	✅	✅	✅
Tagging/Reproducability	✅	✅	✅
Multi-Table Operations	✅	❌	✅
Rollback	✅	✅	✅
Cloud Agnostic	✅	✅	✅
Storage Agnostic	✅	✅	❌
Table Format Agnostic	❌	❌	✅
SQL Operations	✅	✅	❌