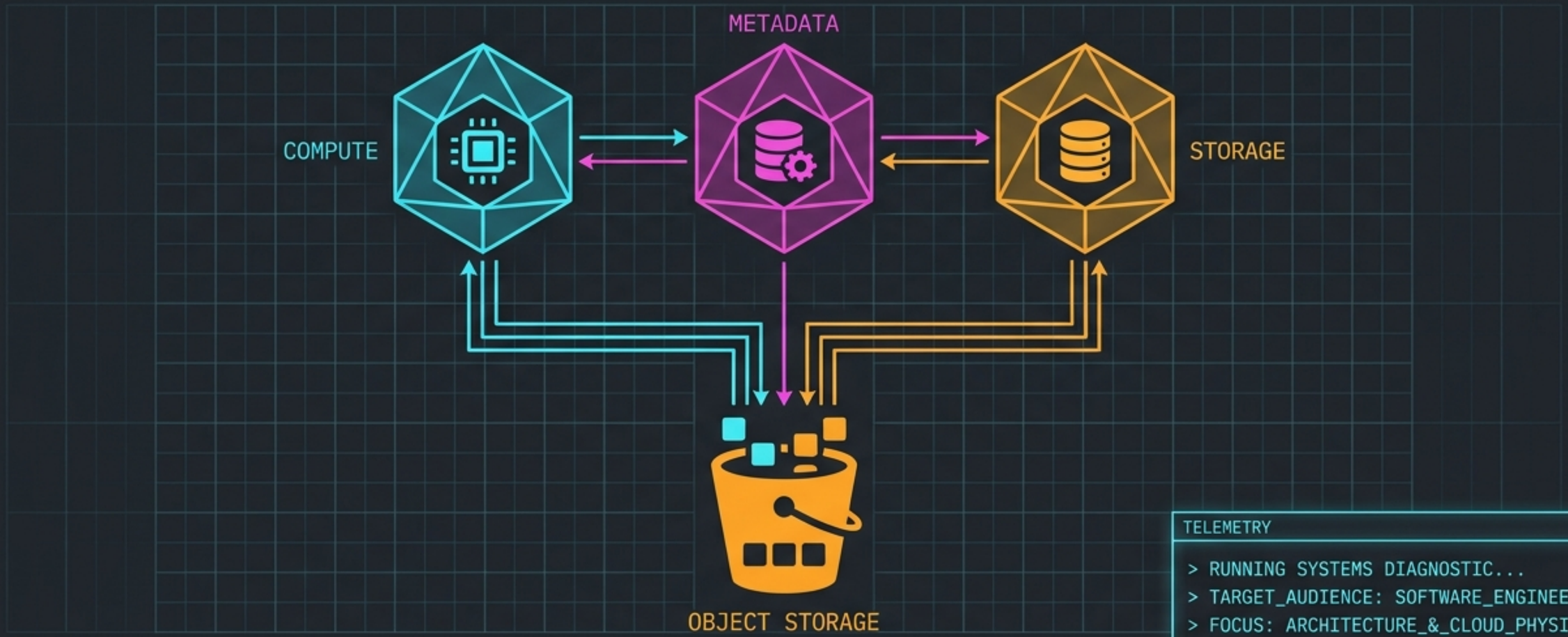
Modern Data Stores – S05E02

Cloud Warehouses, Data Lakes & Open Lakehouse Internals

Compute Separation, Open Formats (Iceberg, Delta, Hudi), and the Physics of Analytical Scale.

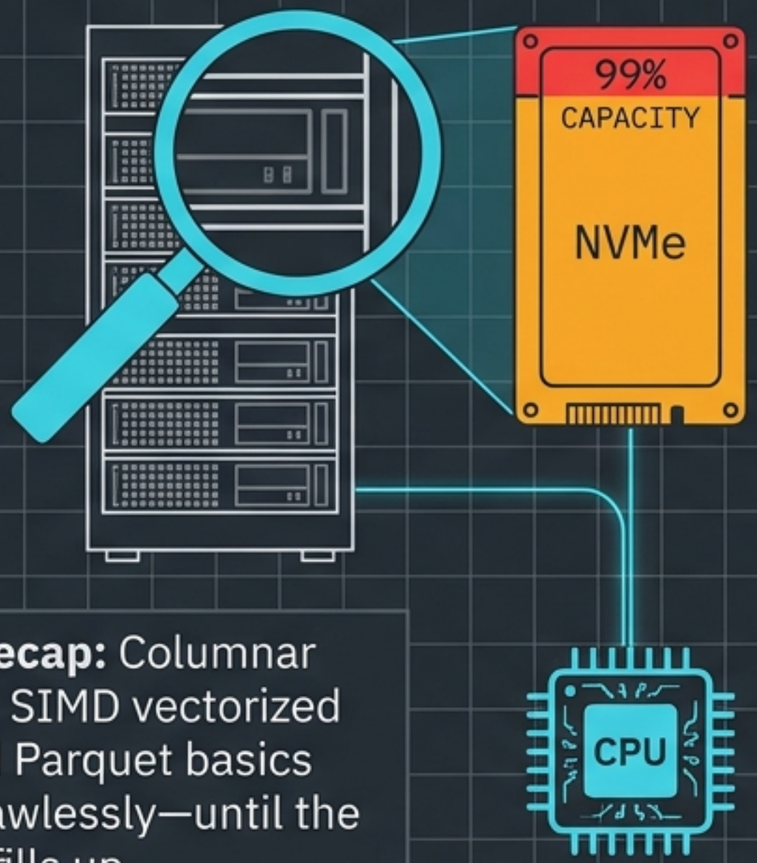


TELEMETRY

> RUNNING SYSTEMS DIAGNOSTIC...
> TARGET_AUDIENCE: SOFTWARE_ENGINEERS
> FOCUS: ARCHITECTURE_&_CLOUD_PHYSICS

Gemini Notebook

The Single-Node Ceiling



S05E01 Recap: Columnar execution, SIMD vectorized scans, and Parquet basics operate flawlessly—until the local disk fills up.

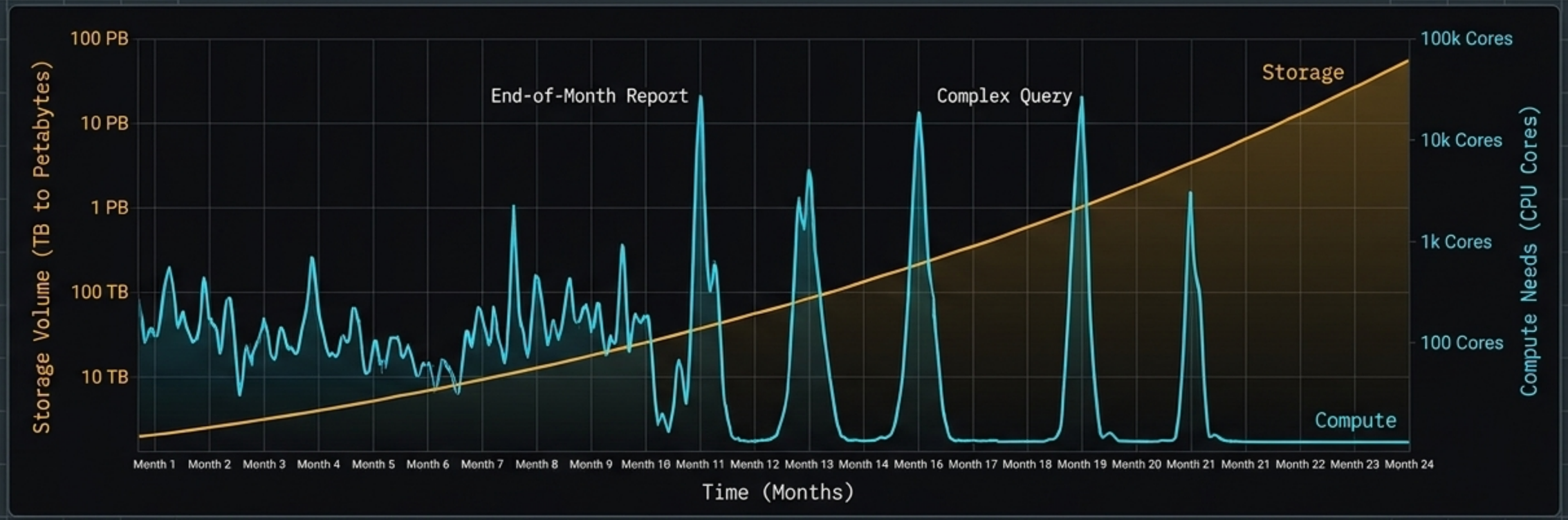
Telemetry Alert

```
MAX_LOCAL_NVME: ~30 TB
CURRENT_DATASET: 45 TB
STATUS: OUT_OF_SPACE_ERROR
```

When analytical data retention surpasses **tens** of **terabytes**, coupling compute CPU directly to local storage fundamentally breaks down. You cannot buy a single server big enough for petabyte-scale history.

Gemini Notebook

The Breaking Point: Why Coupling Fails at Scale



Storage grows monotonically. Compute demand is highly elastic and bursty.

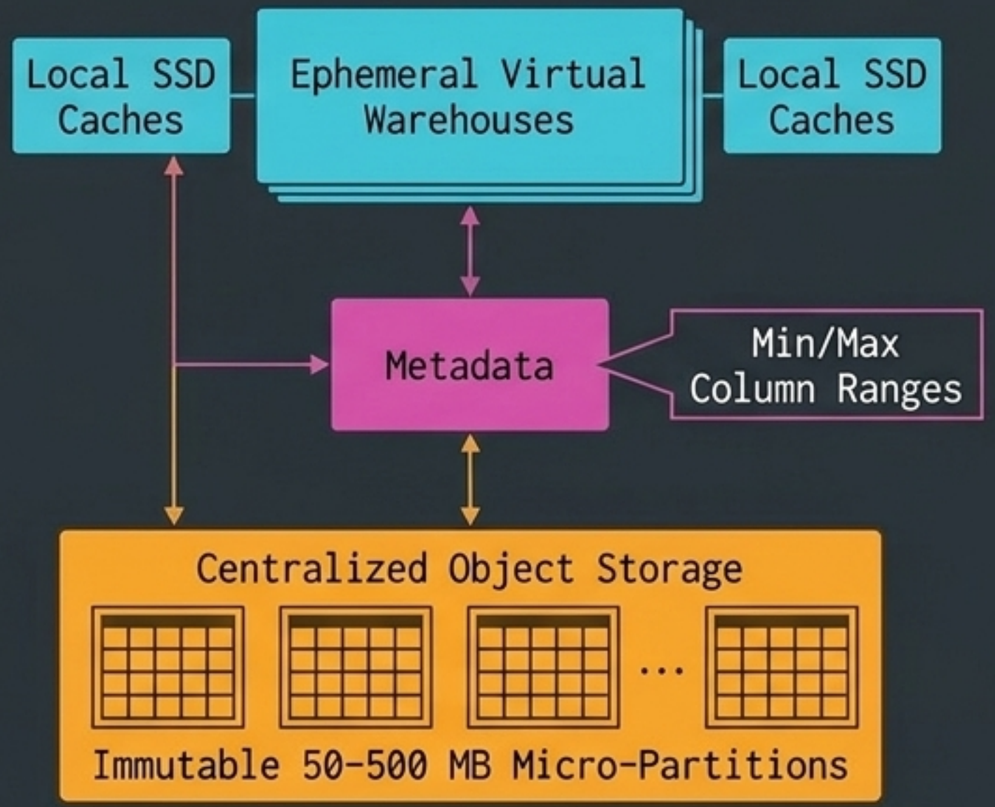
If coupled: Scaling storage to petabytes forces you to pay for thousands of idle CPUs. Scaling compute for a complex query forces you to needlessly migrate terabytes of data across nodes.

Physically sever the link. Decouple stateless compute from persistent storage.

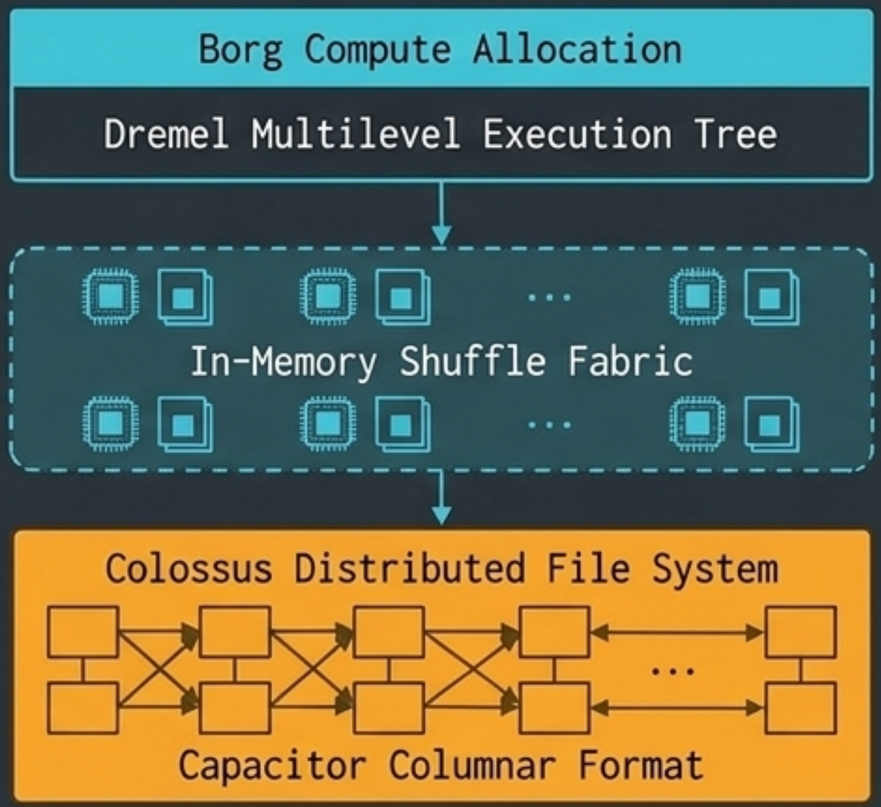
Gemini Notebook

The Decoupled Cloud Warehouse

Snowflake Architecture



BigQuery Architecture



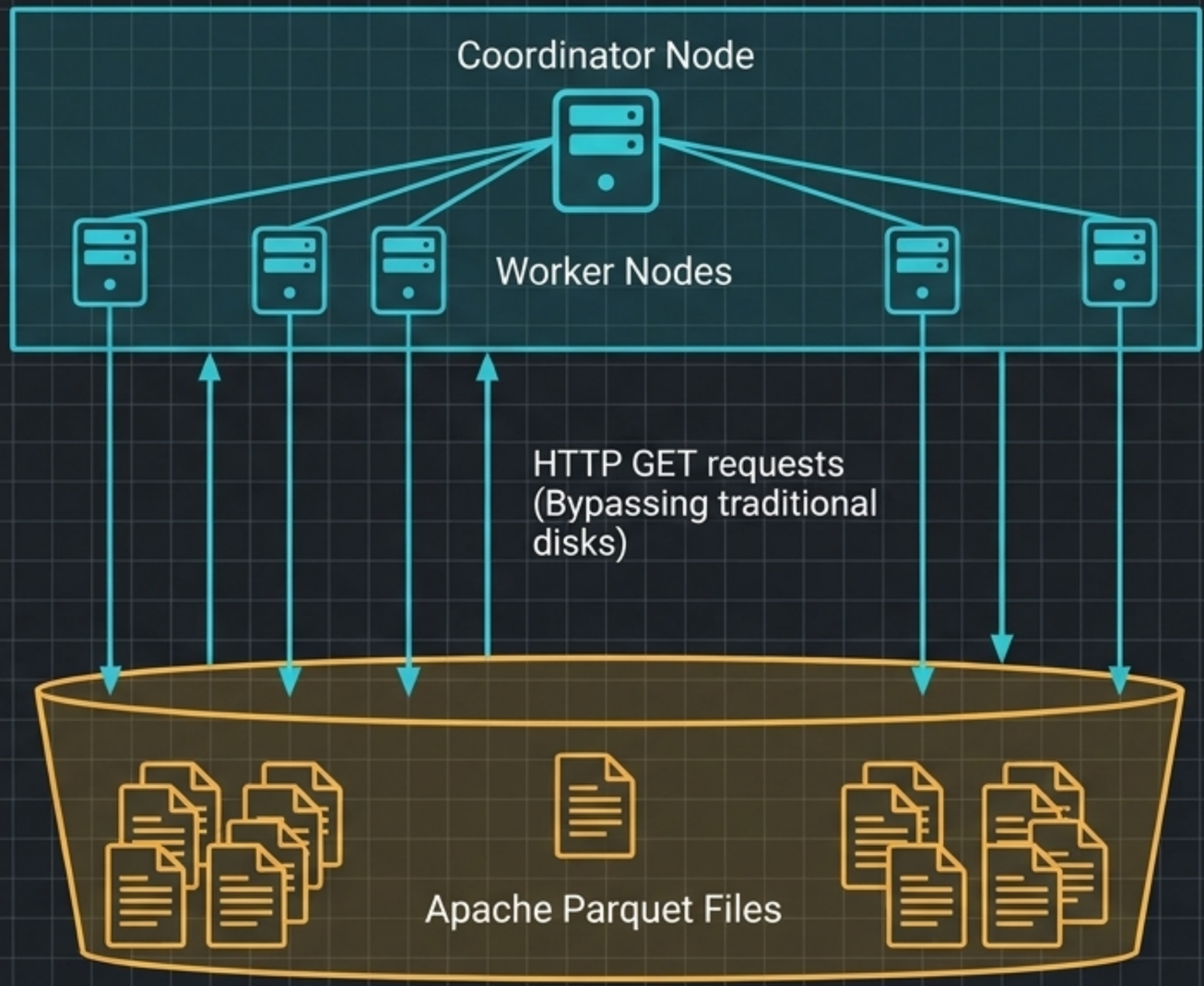
Both systems solve the scaling problem by decoupling, but lock your data inside proprietary columnar formats and closed compute ecosystems.

Gemini Notebook

The Raw Data Lake: Open Storage, Distributed Compute

Trino/Presto
MPP Engines

Network



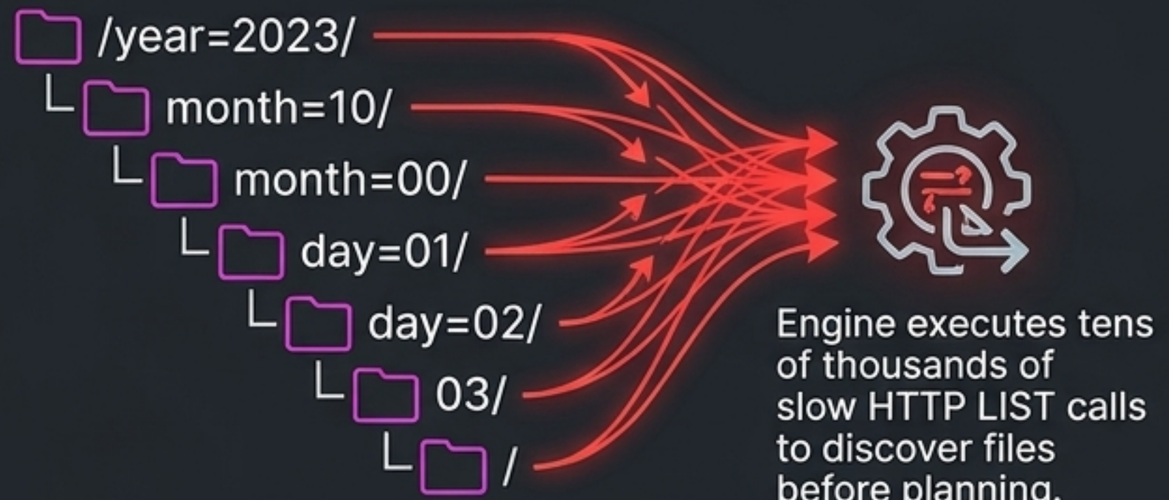
The Promise

Schema-on-read. Store petabytes of raw Parquet in cheap S3 buckets. Point an open-source MPP engine like Trino at the bucket and query immediately. No proprietary ingestion ETL, no vendor storage markup.

Gemini Notebook

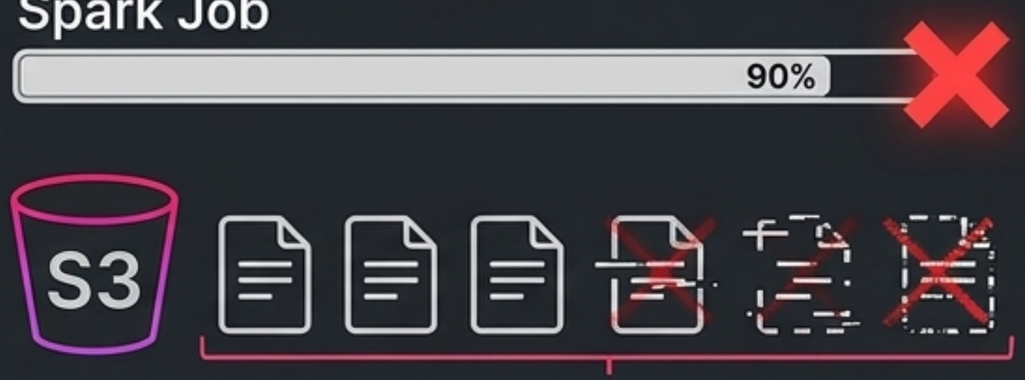
The ‘Data Swamp’ Failure Mode

Failure 1: The O(N) S3 LIST Horror



Engine executes tens of thousands of slow HTTP LIST calls to discover files before planning.

Failure 2: The Partial Write



No ACID guarantees. Concurrent readers immediately read corrupted, duplicated, or partial data.

Failure 3: Rigid Directory Schemas

```
WHERE event_time = ...
```

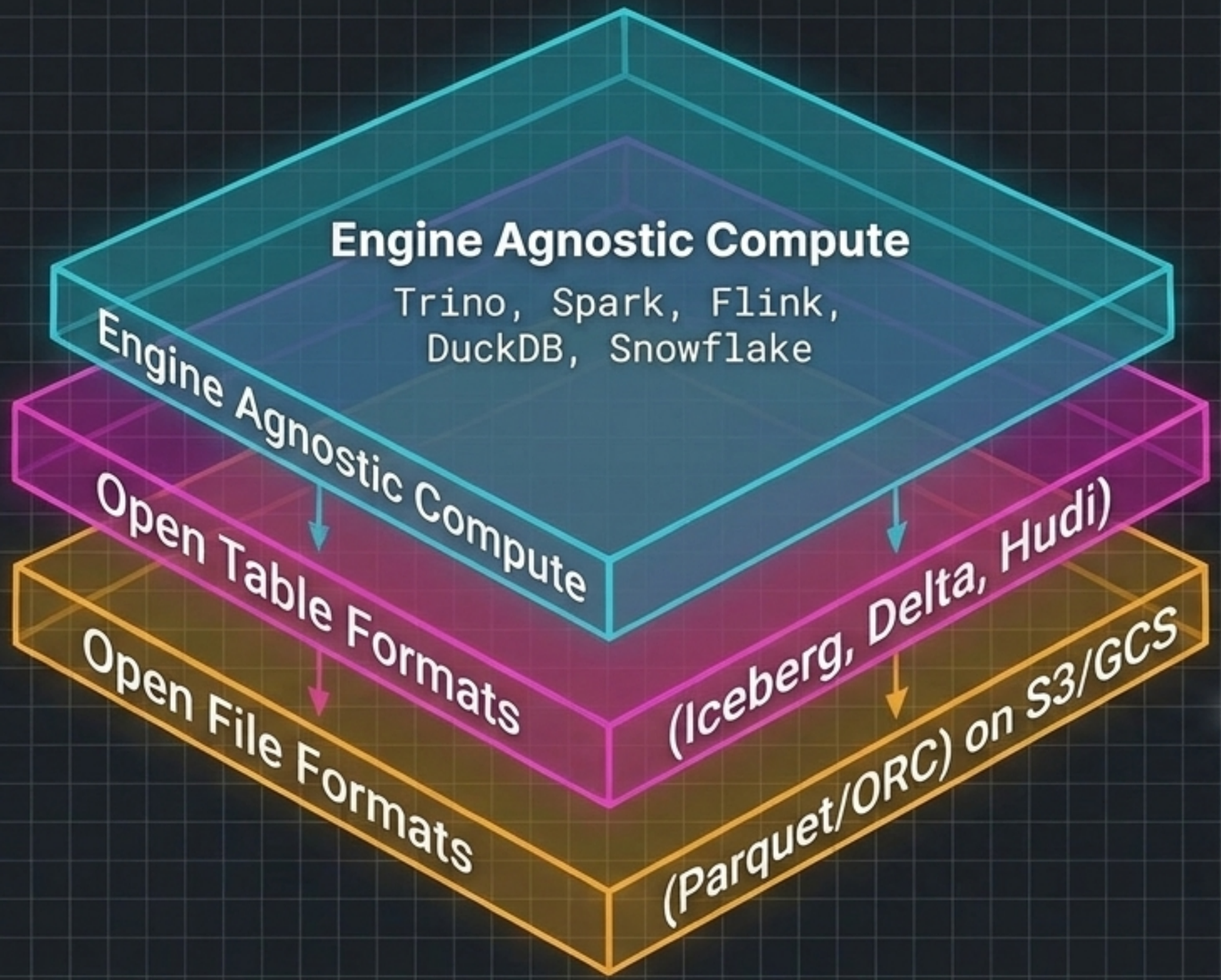


Query fails to prune files because data was manually hive-partitioned by “`event_date`” strings.

Raw Parquet in S3 is just a bunch of files. It is NOT a database.

Gemini Notebook

The Lakehouse Synthesis



The Missing Link: Transactional metadata layers placed directly atop object storage. Defines exact table snapshots and provides ACID guarantees.

The Lakehouse architecture provides the ACID transactions, versioning, and indexing of a warehouse, while maintaining direct access to open formats from advanced ML/DS workloads.

— CIDR 2021 Lakehouse Paper

Gemini Notebook

Comparison Matrix A: Architectural Trade-Offs

	Cloud Warehouse (Snowflake, BigQuery)	Raw Data Lake	Open Lakehouse
Storage Cost	$40-$50/TB-mo (Markup)	$23/TB-mo (Direct S3)	$23/TB-mo (Direct S3)
Vendor Lock-In	High (Proprietary formats)	None (Open Parquet)	None (Open Table Formats)
ACID Governance	Strict ACID	None (Eventual consistency chaos)	Strict Serializable Isolation
Ops Burden	Zero ops (Fully managed)	High (Manual partitioning, broken jobs)	Medium (Mandatory compaction & vacuuming)

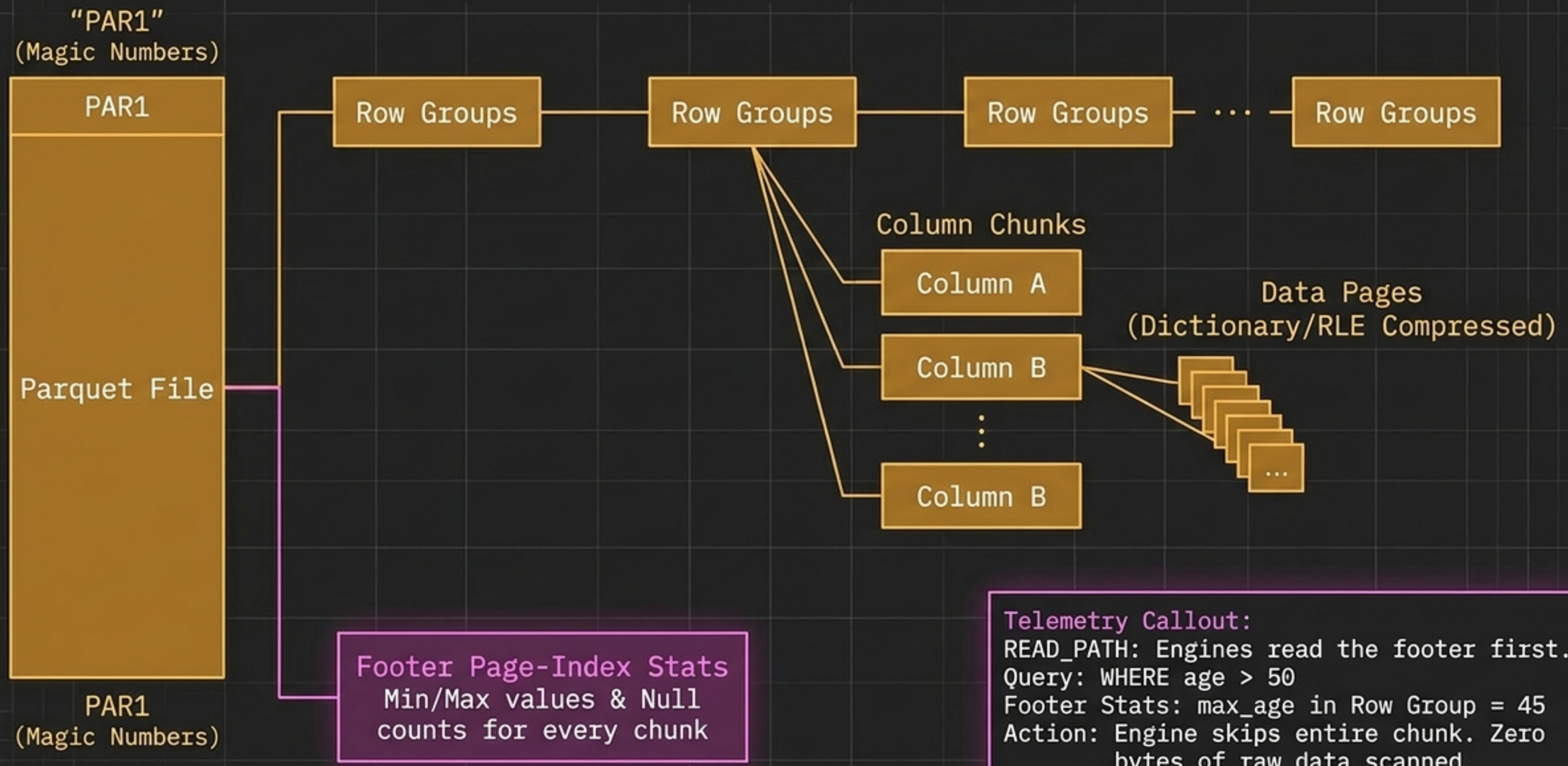
Gemini Notebook

Comparison Matrix B: Open Table Formats

	Apache Iceberg	Delta Lake	Apache Hudi
Log Structure	Hierarchical metadata trees: Catalog -> Manifest List -> Manifest	_delta_log directory with JSON commits & Parquet checkpoints	Timeline directories tracking instants
Partitioning	Hidden partition evolution (zero-rewrite)	Generated columns, Liquid Clustering	Partition path extraction
Upserts (CDC)	Merge-on-Read with Deletion Vectors (v4)	Deletion vectors	Native Copy-on-Write OR Merge-on-Read with bloom/record indexes
Concurrency	Optimistic Concurrency Control (OCC)	OCC (with ConcurrentAppendException)	OCC / MVCC

Gemini Notebook

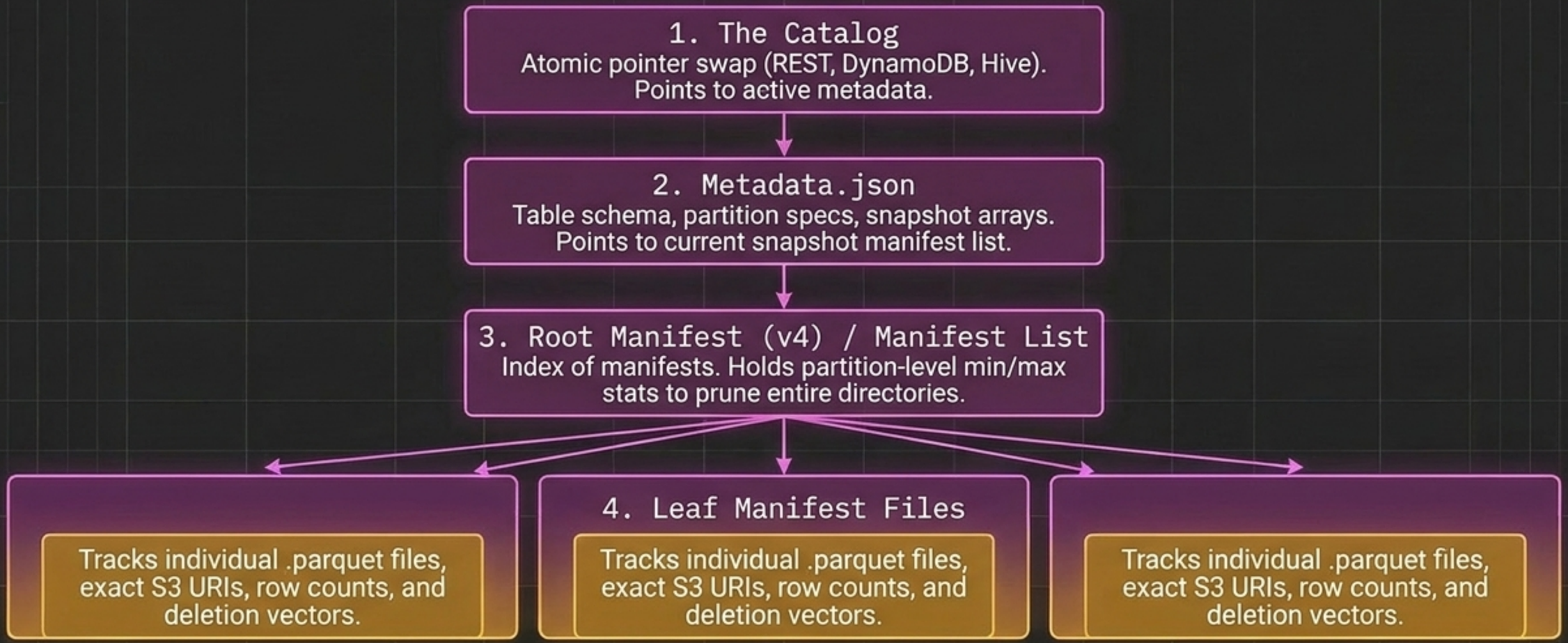
Parquet File Anatomy



Telemetry Callout:
READ_PATH: Engines read the footer first.
Query: WHERE age > 50
Footer Stats: max_age in Row Group = 45
Action: Engine skips entire chunk. Zero bytes of raw data scanned.

Gemini Notebook

Apache Iceberg's 4-Tier Metadata Tree



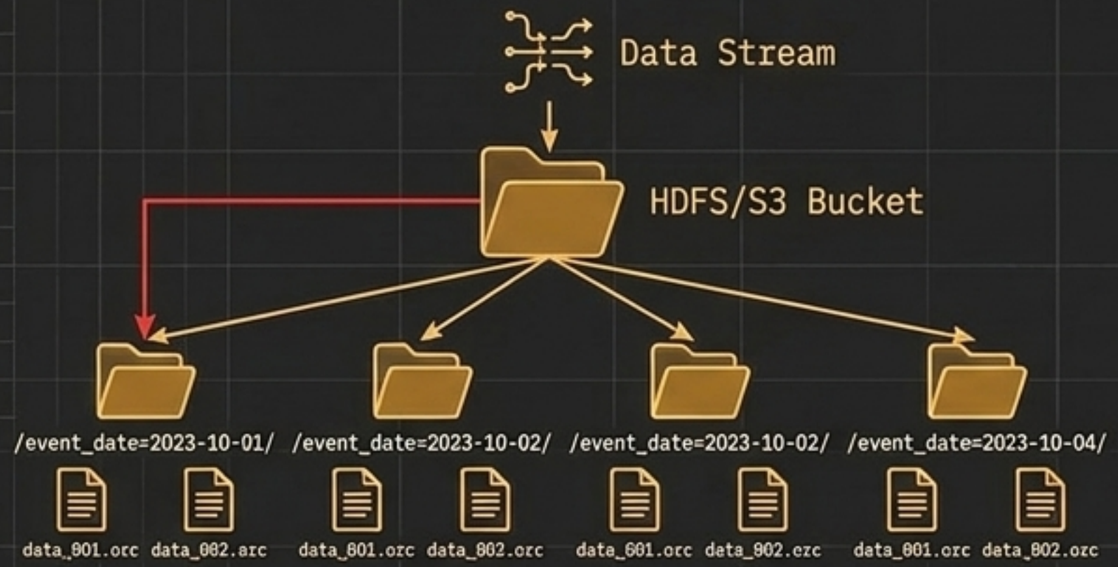
This O(1) RPC lookup architecture entirely bypasses O(N) object storage directory listings, scaling to petabytes instantly.

Gemini Notebook

Hidden Partitioning & Zero-Rewrite Evolution

The Hive Way

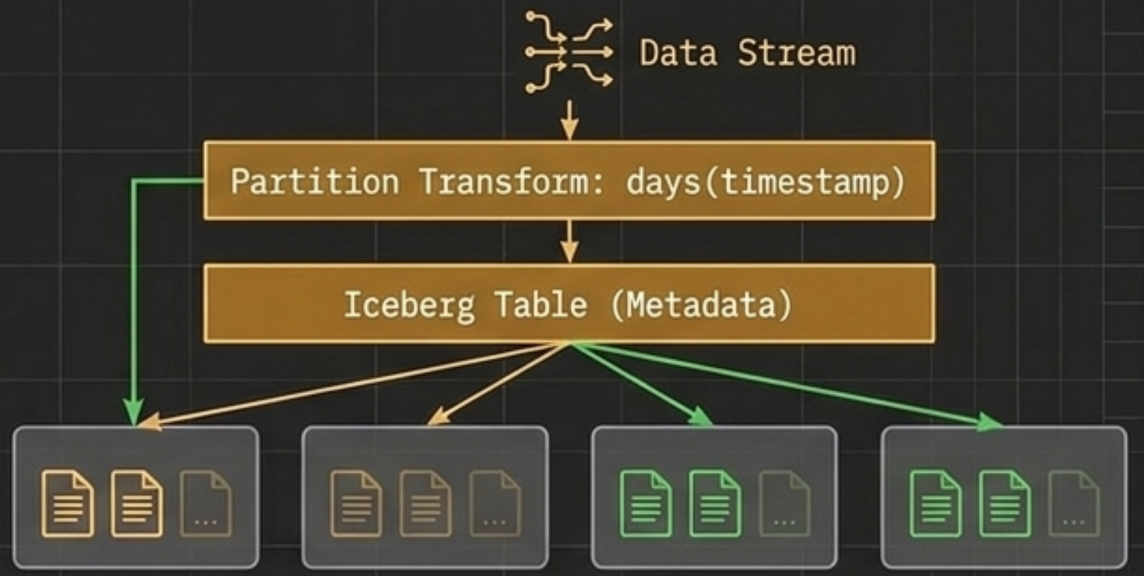
❌ `WHERE event_date = '2023-10-01'`



Users manually extract date from timestamps. If querying 'timestamp > ...', the engine does a full table scan because it doesn't know the relationship.

The Iceberg Way

✅ `WHERE timestamp > '2023-10-01 00:00:00'`

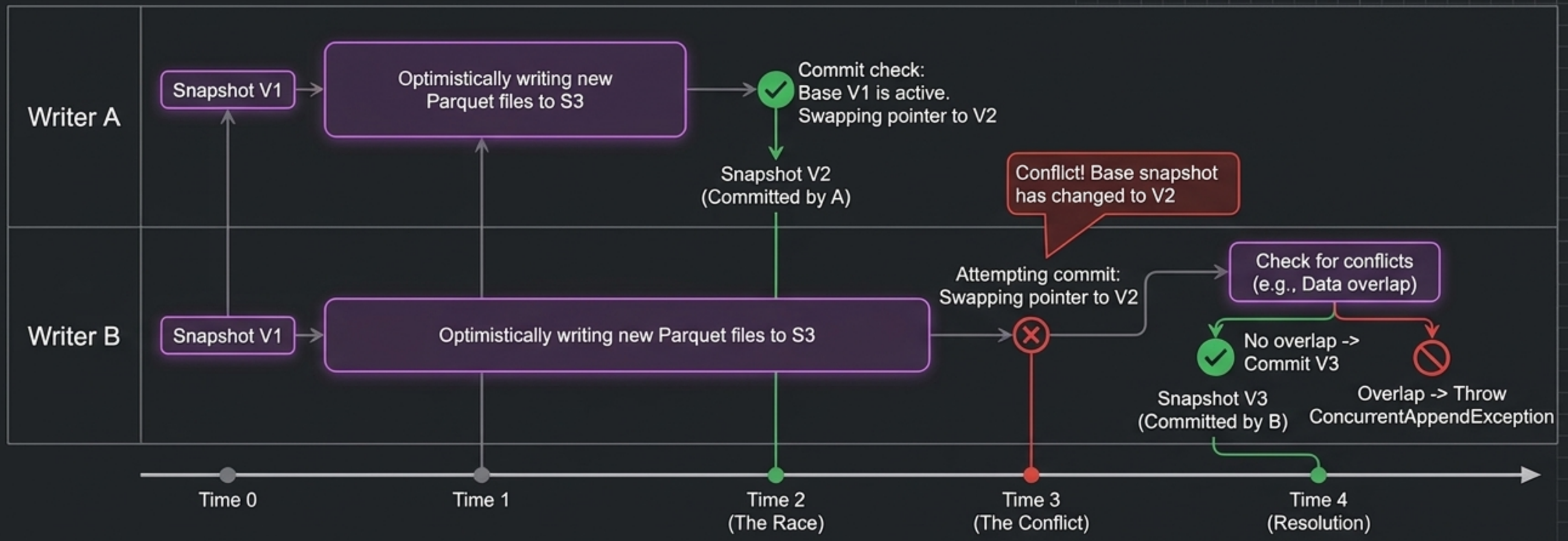


Engine automatically translates the filter and prunes exact underlying files.

Schema Evolution: Partition specs can update on the fly (e.g., daily to hourly). Old data remains in daily buckets, new data in hourly buckets. A single query seamlessly reads across both without rewriting a single byte.

Gemini Notebook

Optimistic Concurrency Control (OCC) & Snapshot Branching



WAP Pattern: Write-Audit-Publish.
Because writers build separate metadata trees before committing, you can branch a table, write data, run data quality tests, and only publish the pointer swap if it passes.

Gemini Notebook

The Physics of Cloud Billing: 50 TB Event Table



50 TB Event Data

```
TABLE_SIZE: 50 Terabytes
FORMAT:     Parquet on AWS S3
STORAGE_COST: ~$23 / TB-month
-> Monthly baseline storage: ~$1,150
```

The Compute Variable



BigQuery On-Demand

Pay per byte scanned (~$6.25 per TB).



Snowflake/Databricks

Pay for virtual warehouse uptime/credits while scanning data.

The Problem: Storage is cheap, but compute I/O is ruthlessly expensive. The financial physics of the cloud dictate that minimizing scanned bytes is the only way to survive at scale.

Gemini Notebook

The Pruning Payoff

The Bill Shock

Query: SELECT * FROM events

Setup: Unpartitioned Data Lake / Brute Force

Action: Engine scans all 50 TB

Math: 50 TB x $6.25

TOTAL: $312.50 (Single Query)

Diagnostic: I/O saturated. High latency. Developer gets yelled at on Monday morning.

The Architect's Query

Query: SELECT user_id, sum(revenue) FROM events WHERE date = 'today'

Setup: Iceberg metadata prunes 99.9% of files. Columnar layout drops unused columns.

Data Scanned: 40 GB

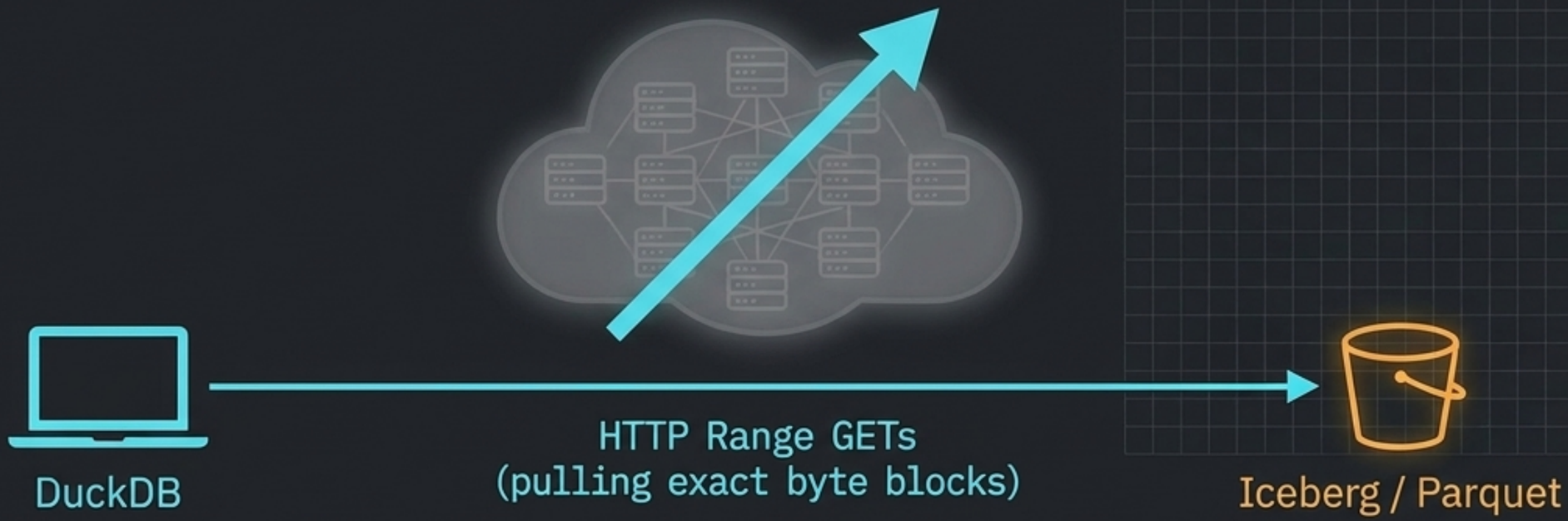
Math: 0.04 TB x $6.25

TOTAL: $0.25

Diagnostic: Instant results. Fraction of a penny.

Gemini Notebook

The Single-Node Lakehouse (DuckDB)



The Concept

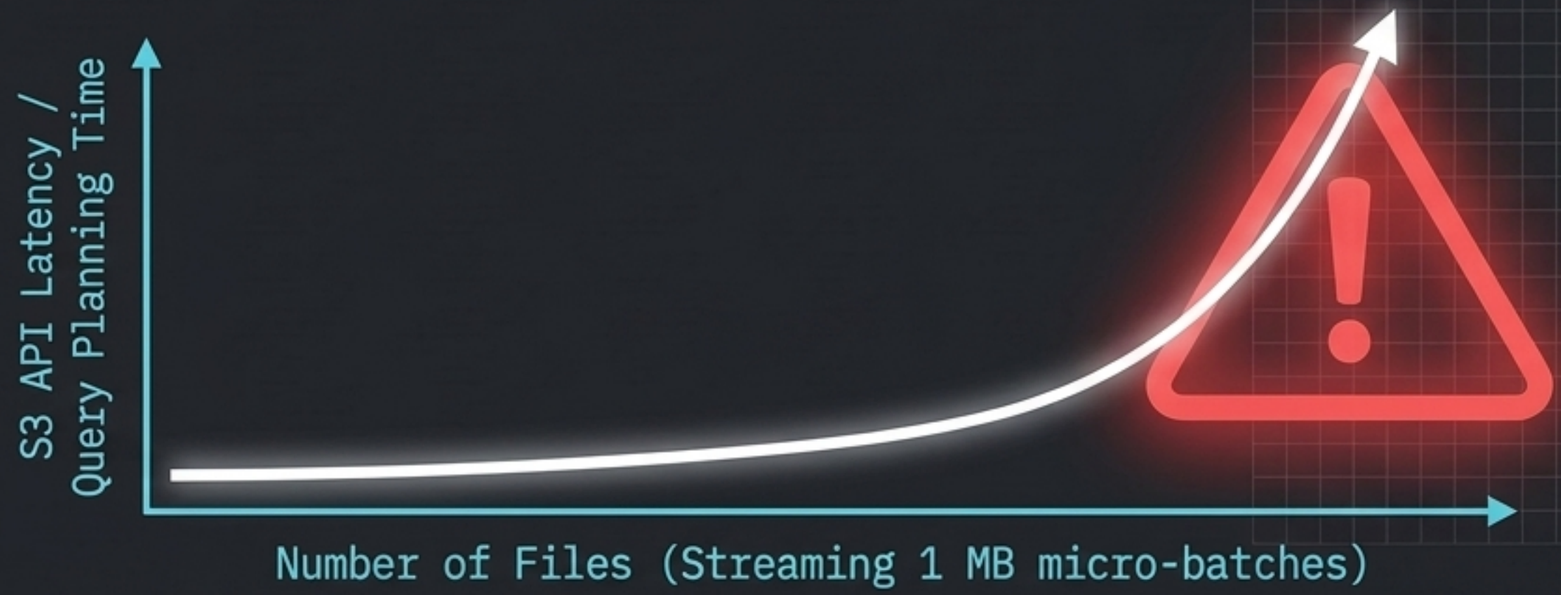
Because Iceberg and Parquet are open standards, you don't need a distributed MPP cluster to read them. Single-node engines can read metadata directly over the network and pull only required bytes into local RAM.

Threshold Callout

For datasets < 500 GB (or highly pruned queries), a single 128 GB RAM node outperforms a distributed cluster by eliminating network shuffle and startup times. Cost to execute compute: $0.00.

Gemini Notebook

Lakehouse Anti-Patterns: The ‘Small-File’ Trap



The Physics

1-minute streaming micro-batches generate millions of tiny 1 MB Parquet files.

The Collapse

- S3 API metadata requests (GET/HEAD) become a massive bottleneck.
- Parquet compression ratios degrade (dictionaries fail on 1MB chunks).
- Iceberg manifest files bloat, destroying query planning speed.

The Fix

Actively monitor file sizes and asynchronously bin-pack small files into 128MB-512MB chunks.

Open storage is NOT zero ops.

Gemini Notebook

OCC Contention & Snapshot Bloat

Write Contention



Heavy concurrent writers targeting the same partition cause OCC commit conficts. System enters an endless commit retry loop, tanking throughput.

Fix: Route/shuffle writers by partition key before writing.

Storage Bloat

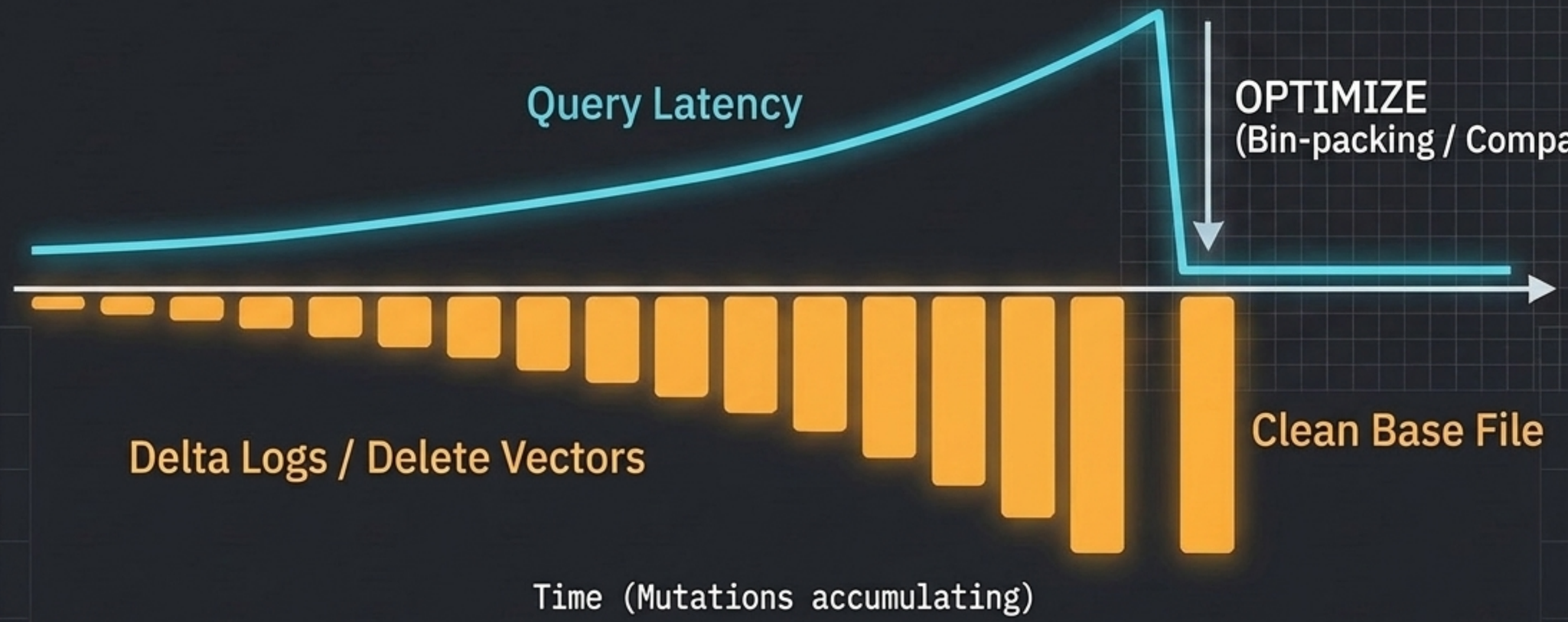


Every commit creates a new snapshot. Deleted/updated data is never physically removed—just logically omitted.

The Trap: Without running Snapshot Expiration and VACUUM, storage costs silently double and triple as you pay for unreferenced orphan files.

Gemini Notebook

The Compaction Cliff: Merge-on-Read Amplification



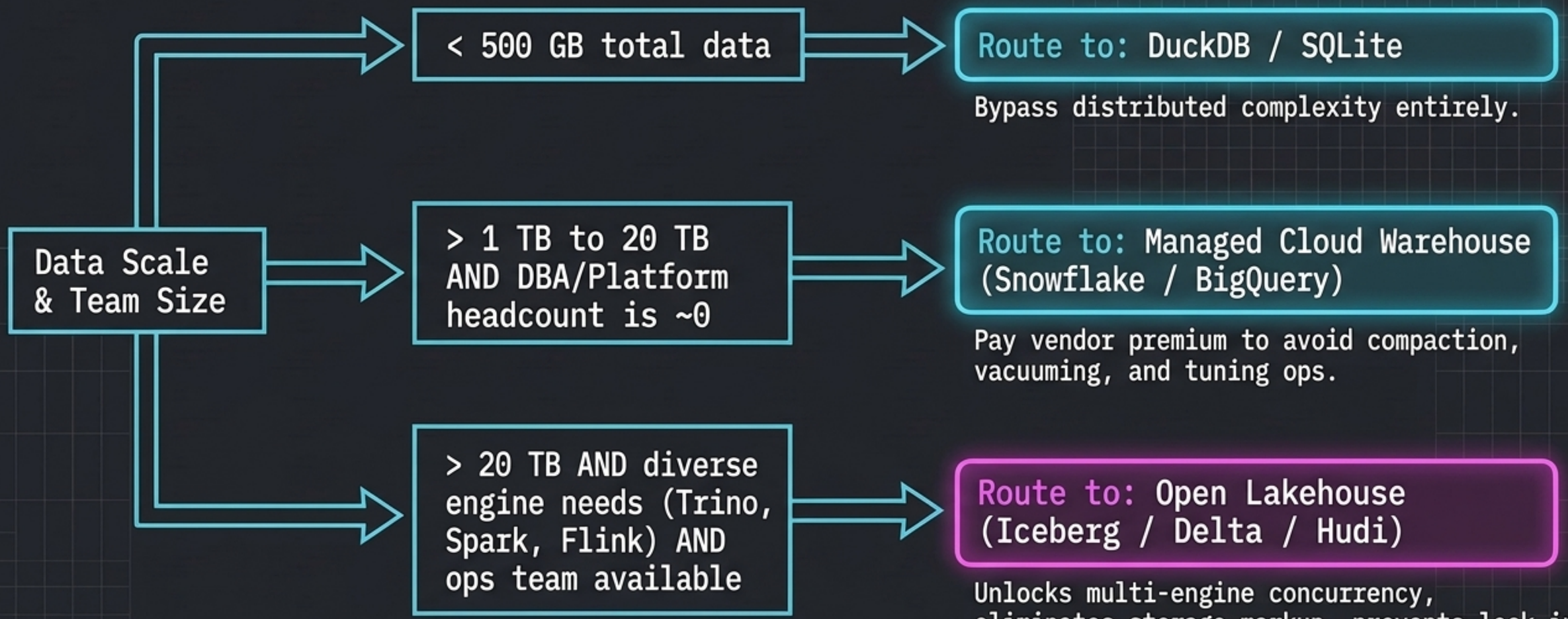
The Write Amplification (CoW): Requires rewriting an entire 500MB Parquet base file to update 1 record. Very high write cost.

The Read Amplification (MoR): Engine must scan base file AND on-the-fly merge thousands of delta logs Query latency degrades severely.

The Cliff: Latency resets only when a background OPTIMIZE job merges logs into a fresh Parquet file.

Gemini Notebook

The Platform Architect’s Decision Tree



Gemini Notebook

Epilogue & Next Up

Episode Summary

- [❌] Scale dictates decoupling compute from storage.
- [✅] Open table formats (Iceberg/Delta/Hudi) restore ACID guarantees to raw S3 data lakes.
- [✅] Query costs are governed by metadata pruning and columnar chunking.
- [✅] Open storage requires active ops (VACUUM, OPTIMIZE) to prevent compaction cliffs.

Next Episode Teaser

> LOADING S05E03...



NoSQL Families - Operational Non-Relational Stores

Next time, we leave the analytical data warehouse behind and dive into high-throughput OLTP: Key-Value, Document, Wide-Column, and Graph architectures.

Gemini Notebook