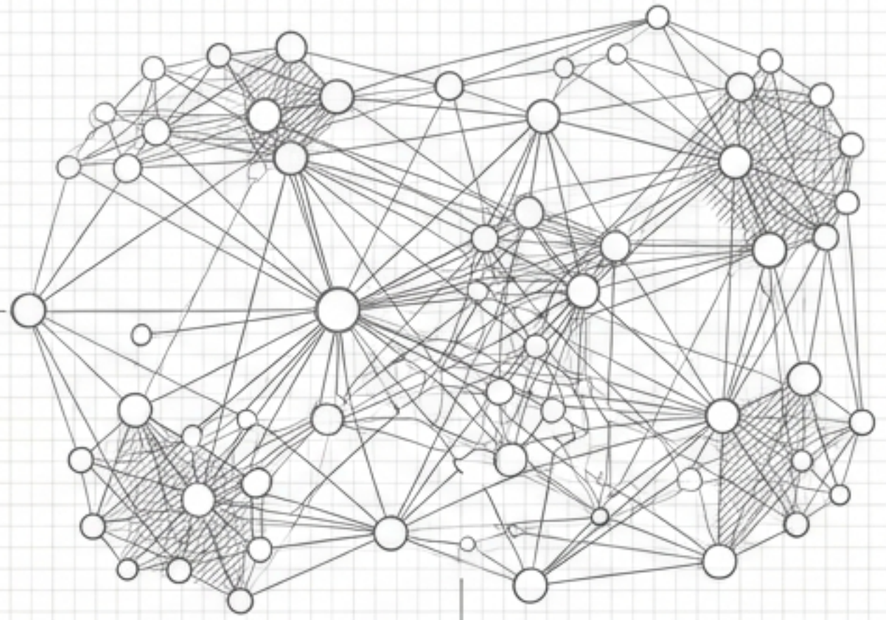
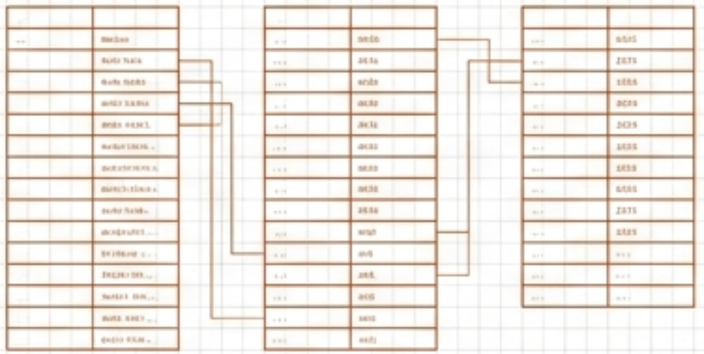
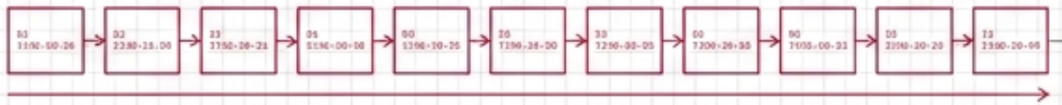
Modern Data Stores

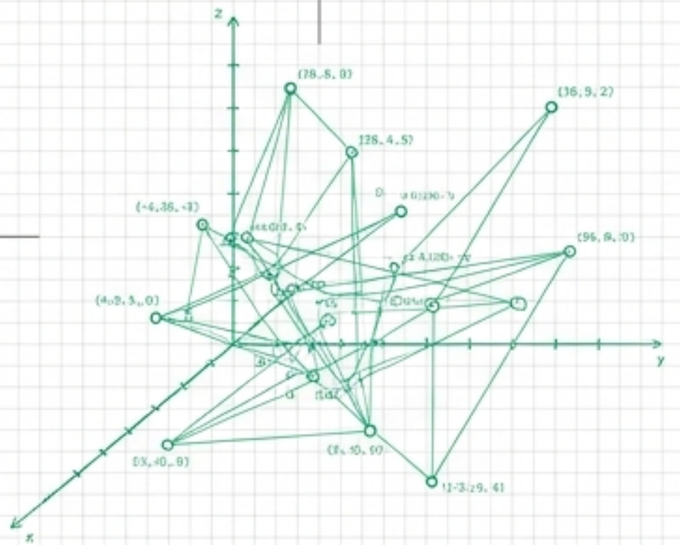
S05E00: Architecture Map of Operational, Analytical, Streaming & Vector Stores

A Master Reference Map for Software Engineers and Data Architects





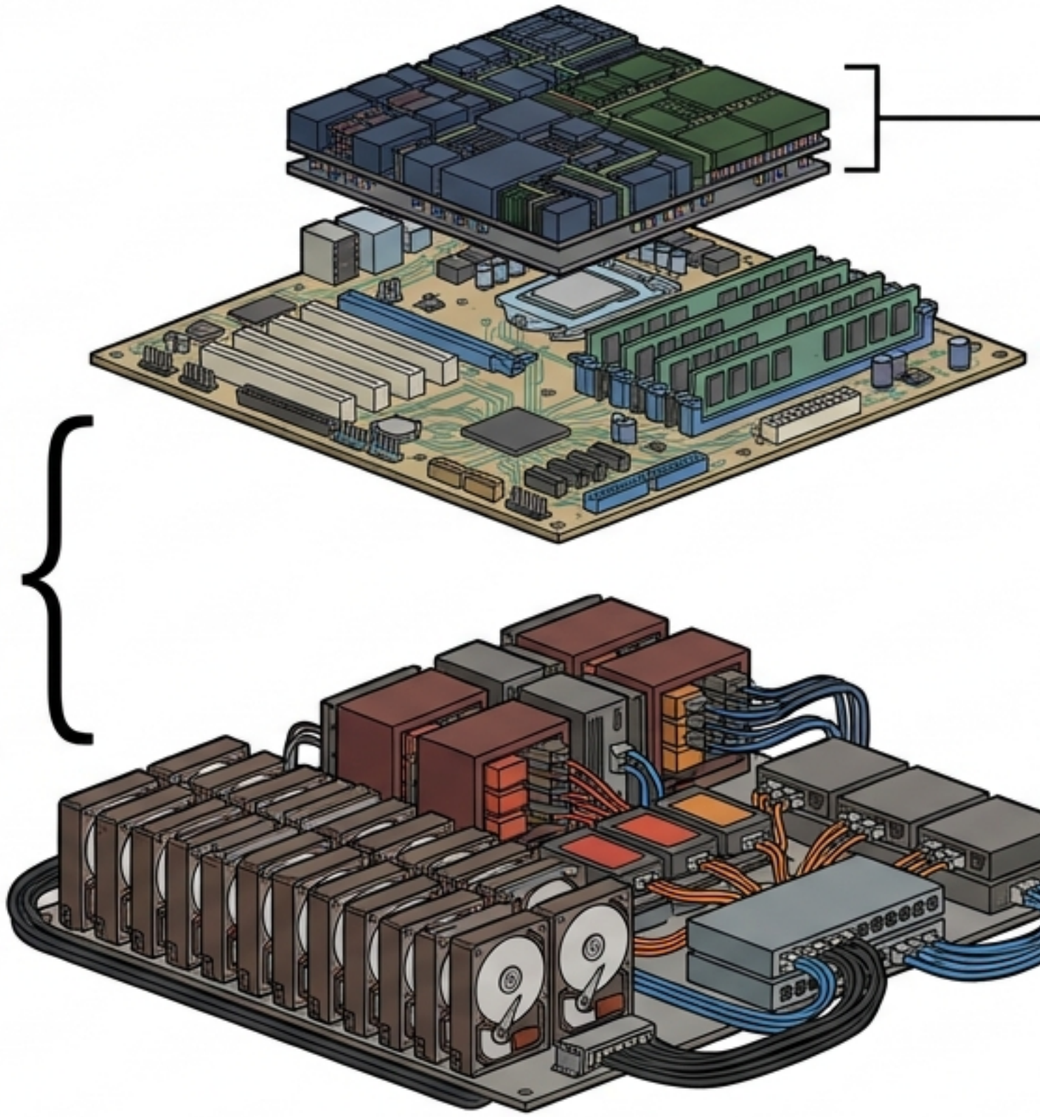




Gemini Notebook

The Iron Law of Storage Engines

Every storage engine is a highly specialized hack to optimize for a specific unit of I/O. There is no silver bullet, only physics.



CPU L1/L2 Cache (Nanoseconds)

Main Memory / RAM (Microseconds)

The Latency Chasm

Disk & Network I/O (Milliseconds)

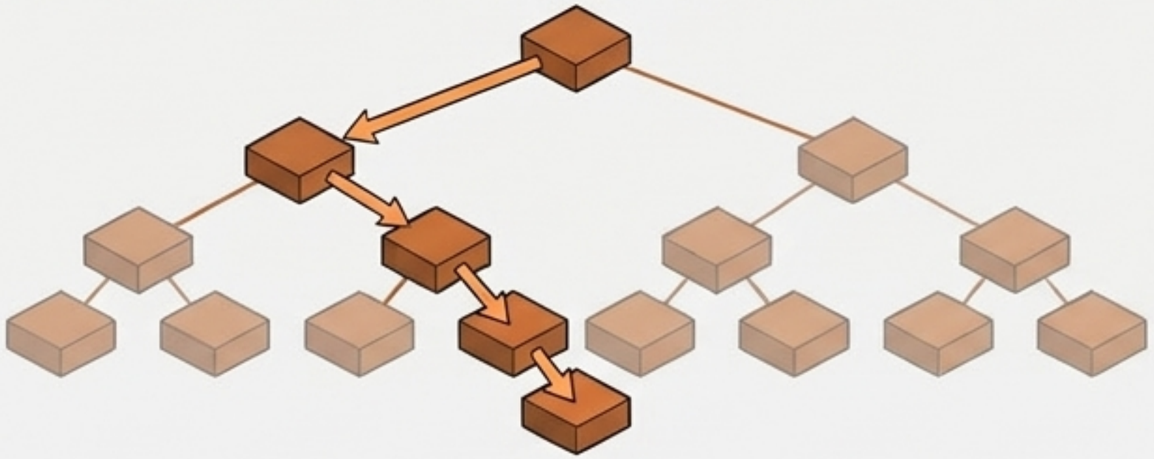
Datastore architecture is purely the science of hiding disk and network latency.

Gemini Notebook

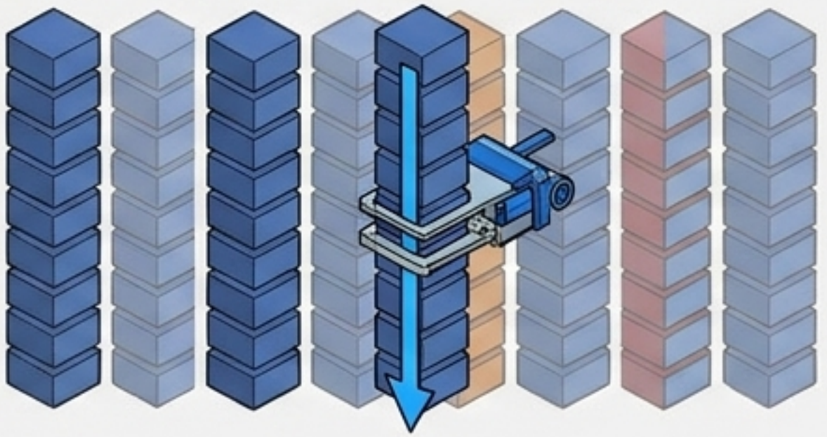
Visualizing the Foundational Access Patterns

These four I/O patterns dictate physical layout on disk and in memory, establishing the roadmap for S05 (E01–E07).

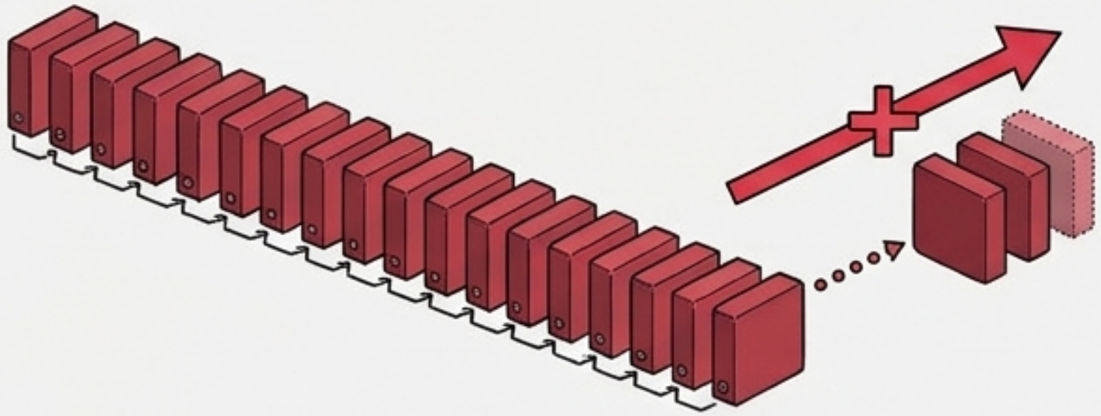
Point Tuple Mutation



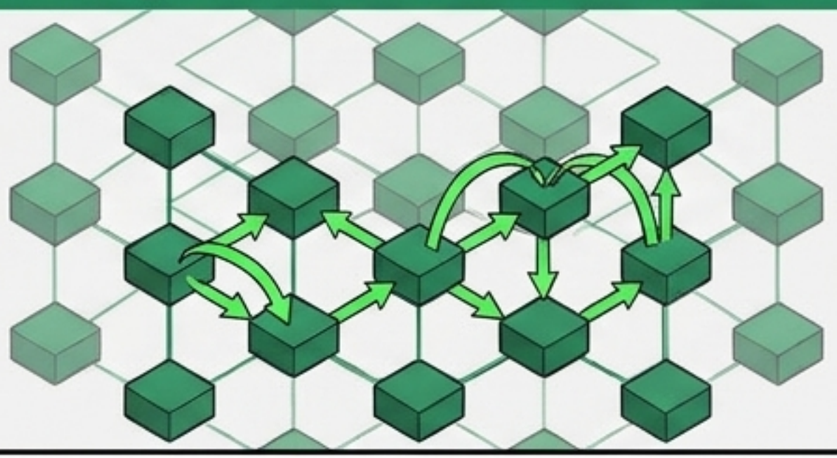
Column-Chunk Scan



Append-Only Offset



High-Dimensional Traversal

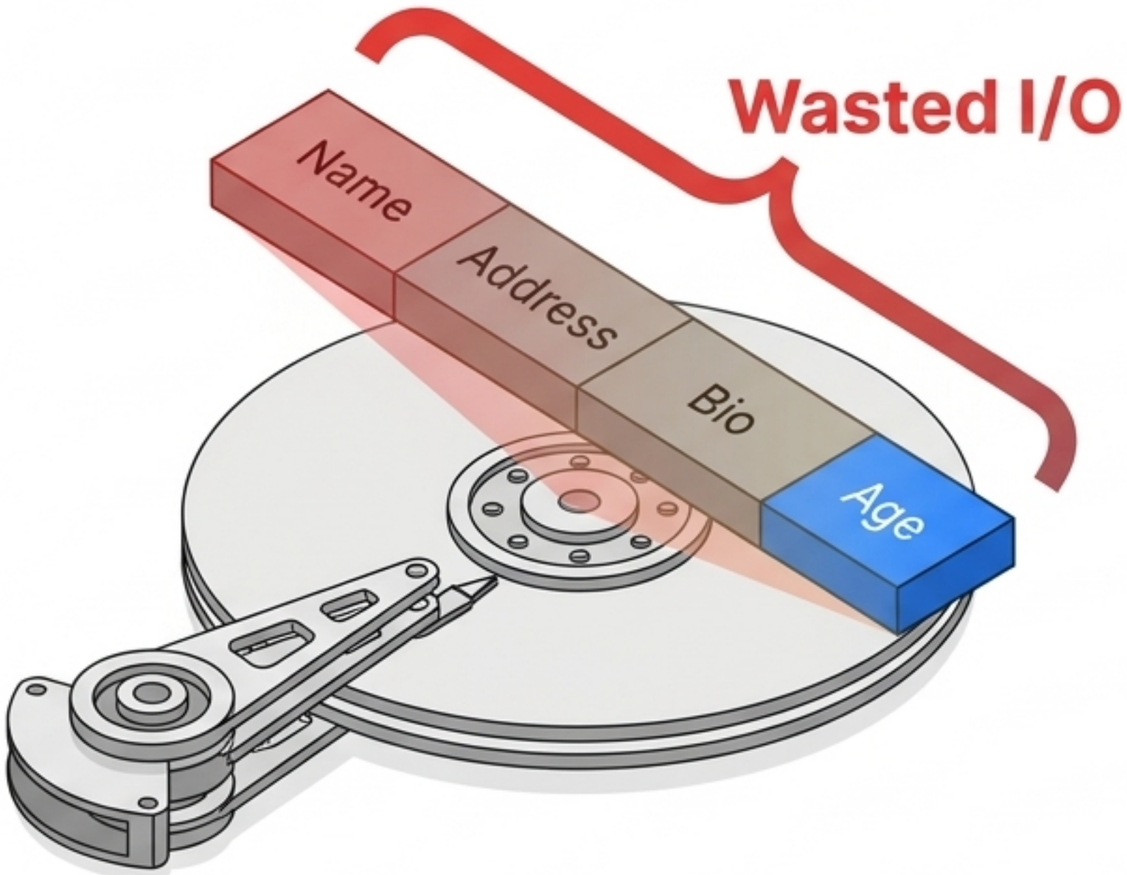


Gemini Notebook

The Great Divide: Row vs. Column

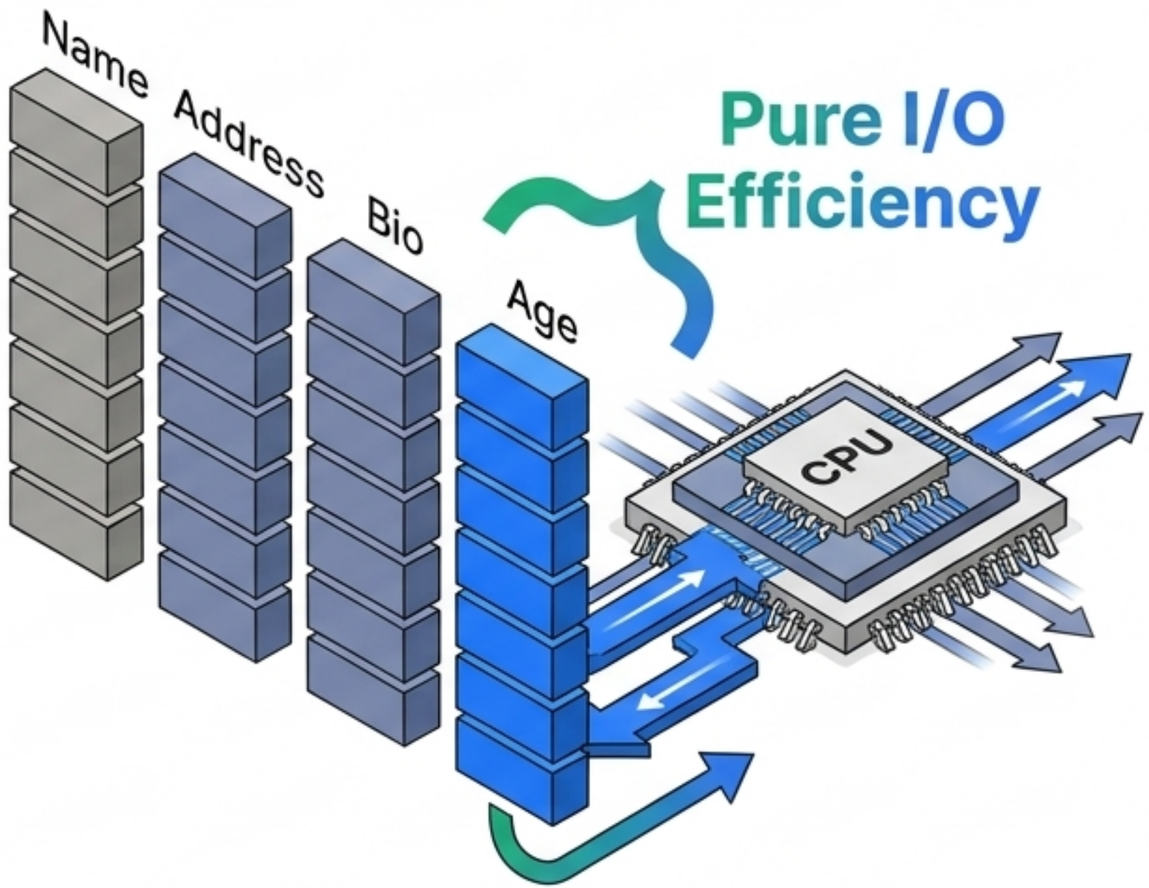
OLTP (Row-Oriented)

Optimized for mutations.
Heap pages keep entity attributes together.



OLAP (Column-Oriented)

Optimized for aggregations. Dense contiguous blocks enable vectorized processing.

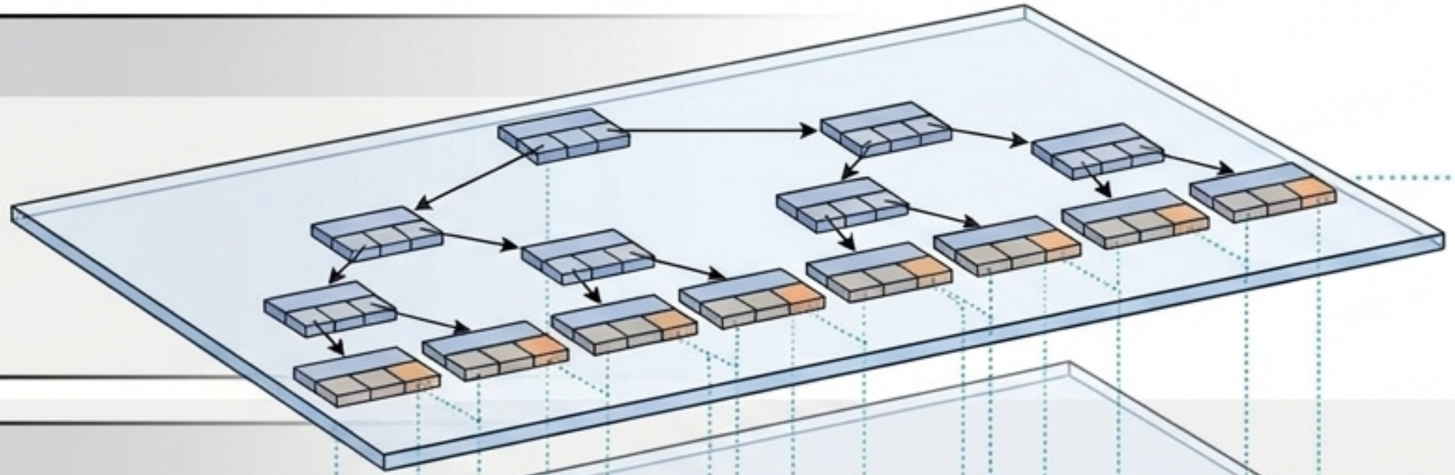


Gemini Notebook

OLTP Anatomy (The Mutators)

Top Layer

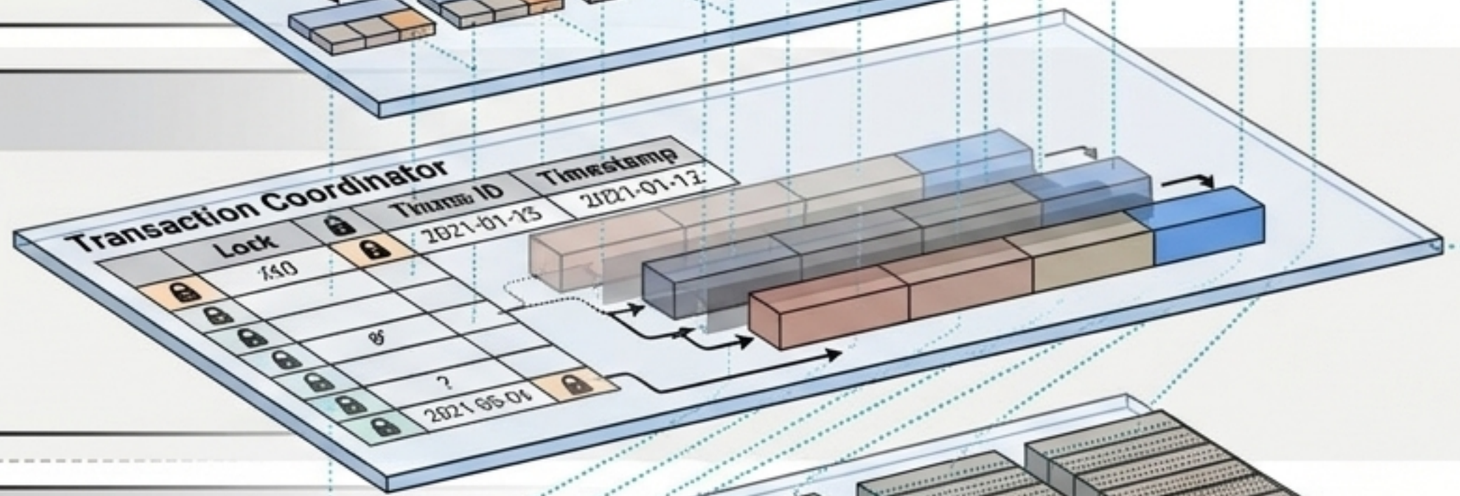
Indexes



Map keys to physical heap page addresses.

Middle Layer

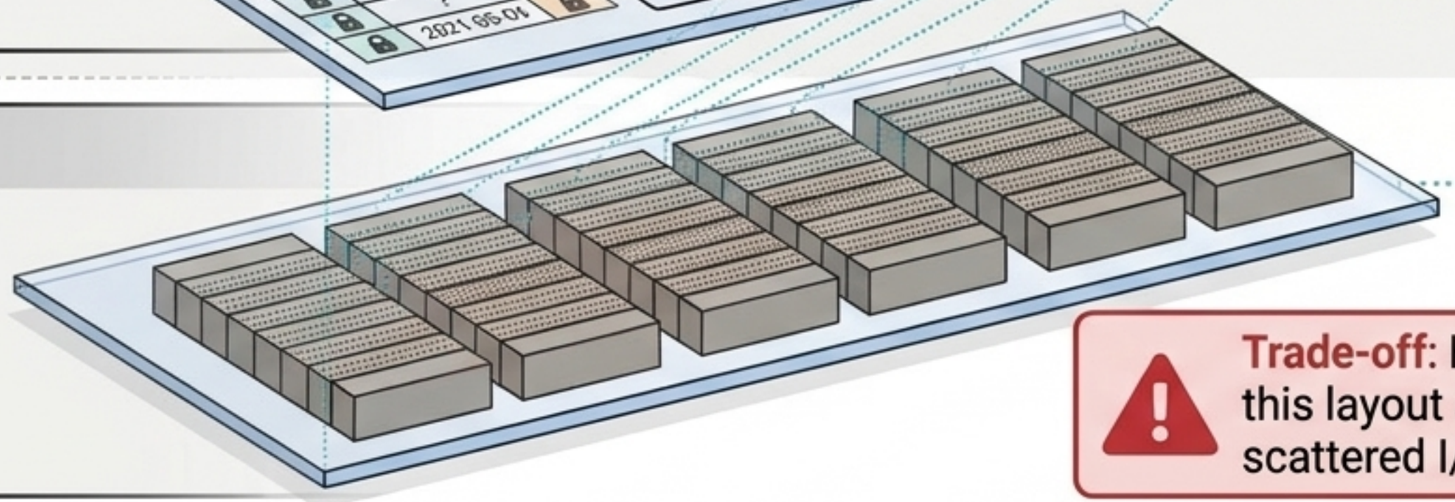
MVCC (Multi-Version Concurrency Control)



Prevents readers from blocking writers by storing `multiple row versions`.

Bottom Layer

Row-Oriented Heap Pages



`Strict ACID guarantees` and `row-level locking`.

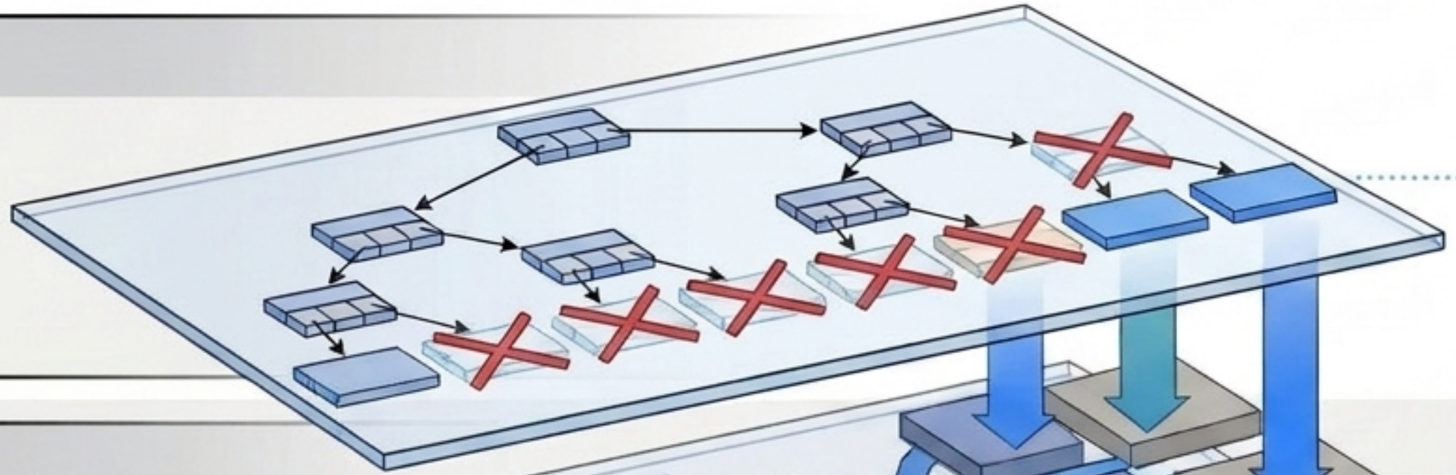
Trade-off: Heavy analytical scans over this layout are painfully slow due to scattered I/O and row-level overhead.

Gemini Notebook

OLAP Anatomy (The Scanners)

Top Layer

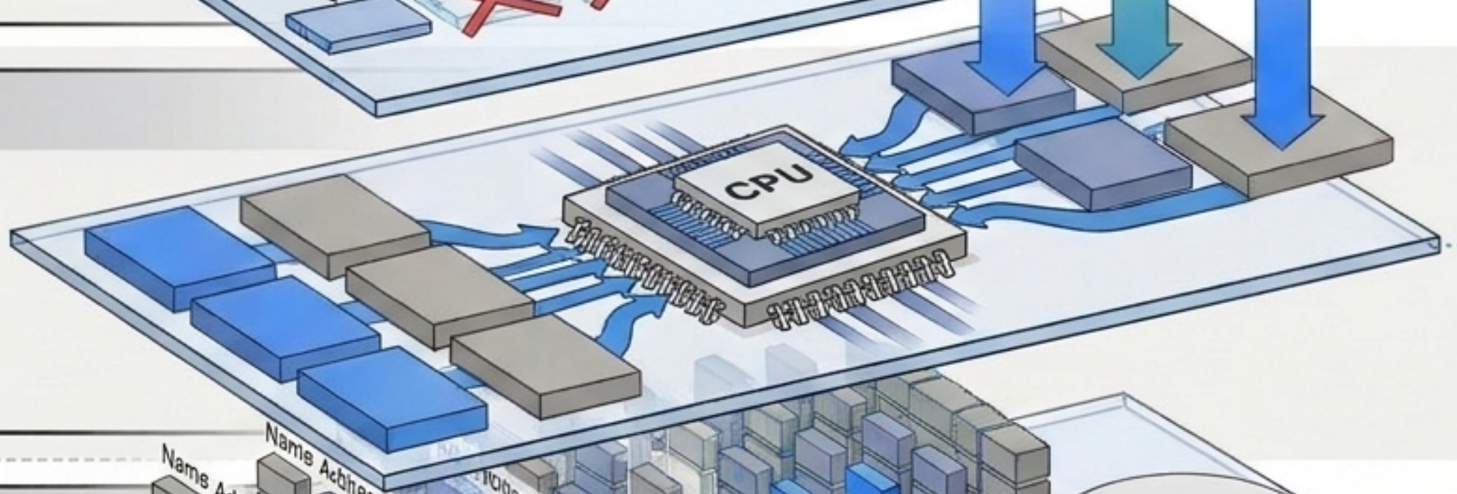
Sparse Indexing (Data Skipping)



Doesn't point to specific rows; skips entire blocks that don't match query filters.

Middle Layer

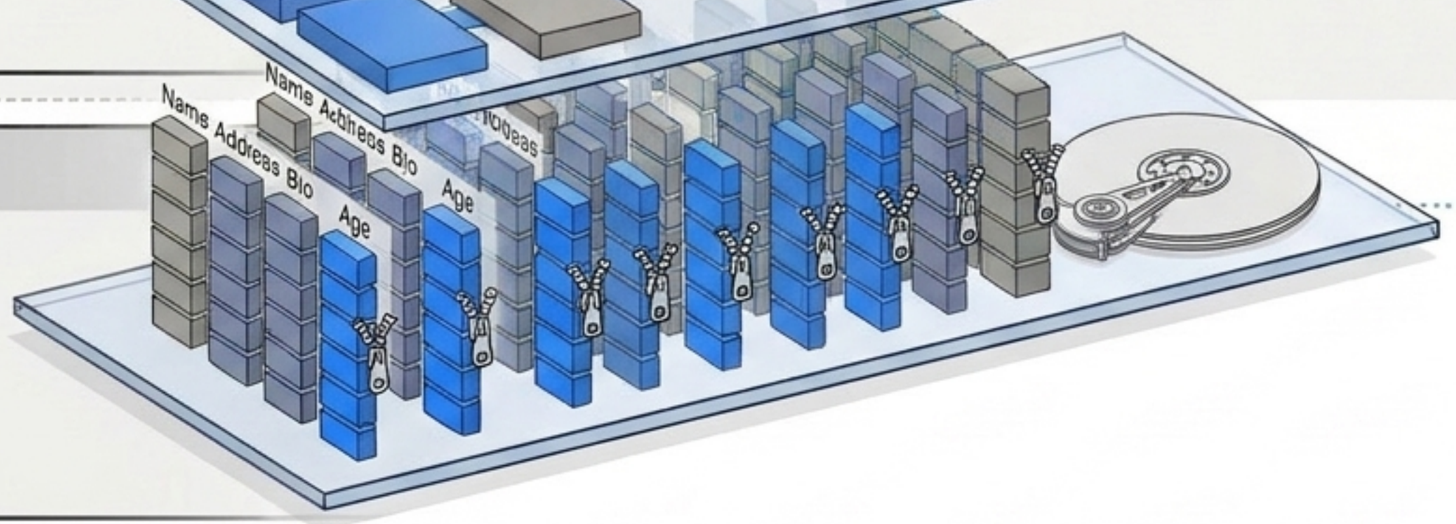
Vectorized Execution & Late Materialization



Processes large batches of values in single CPU operations.

Bottom Layer

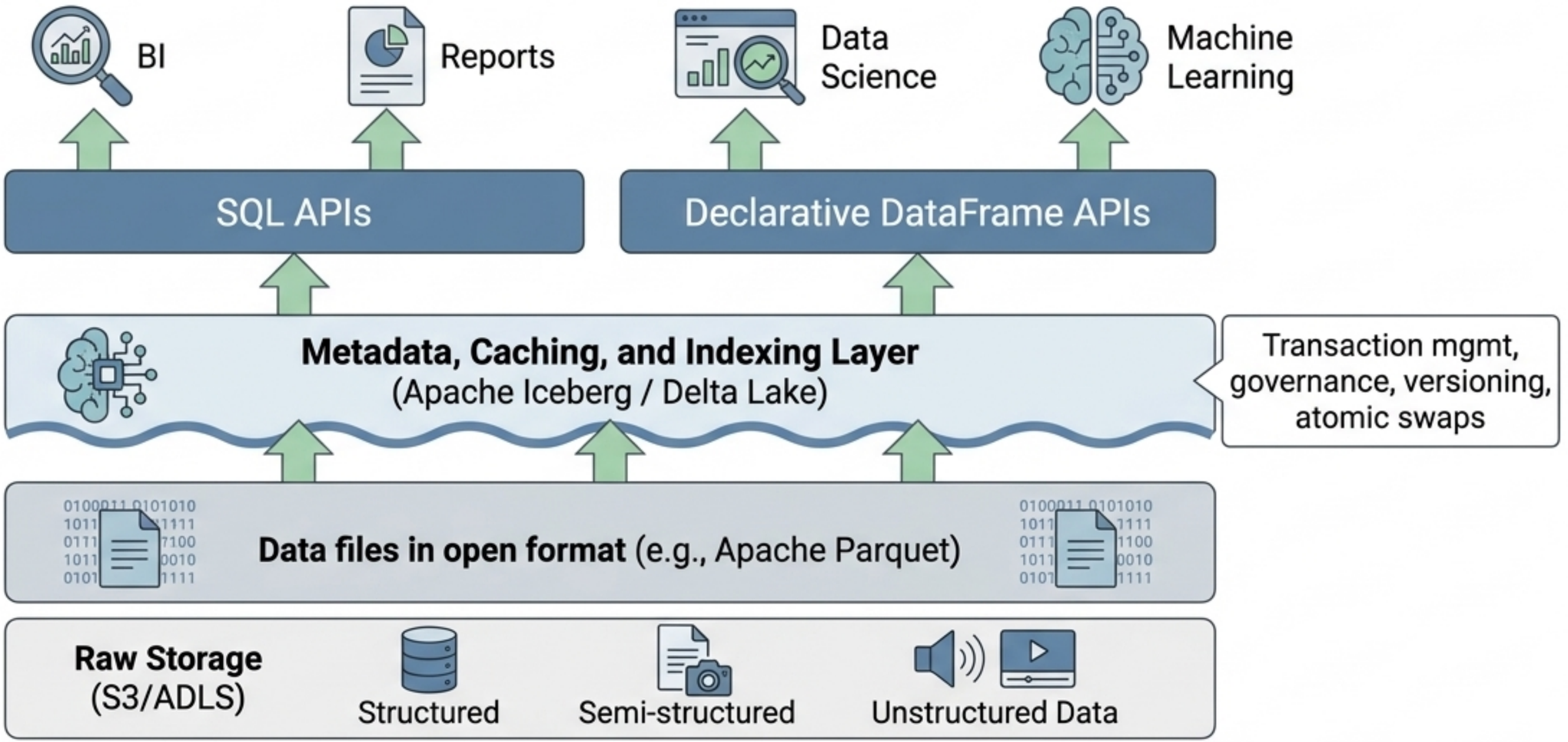
Compressed Columnar Data



`Drastically reduces disk I/O footprint.`

Gemini Notebook

The Lakehouse Evolution: Breaking the Monolith

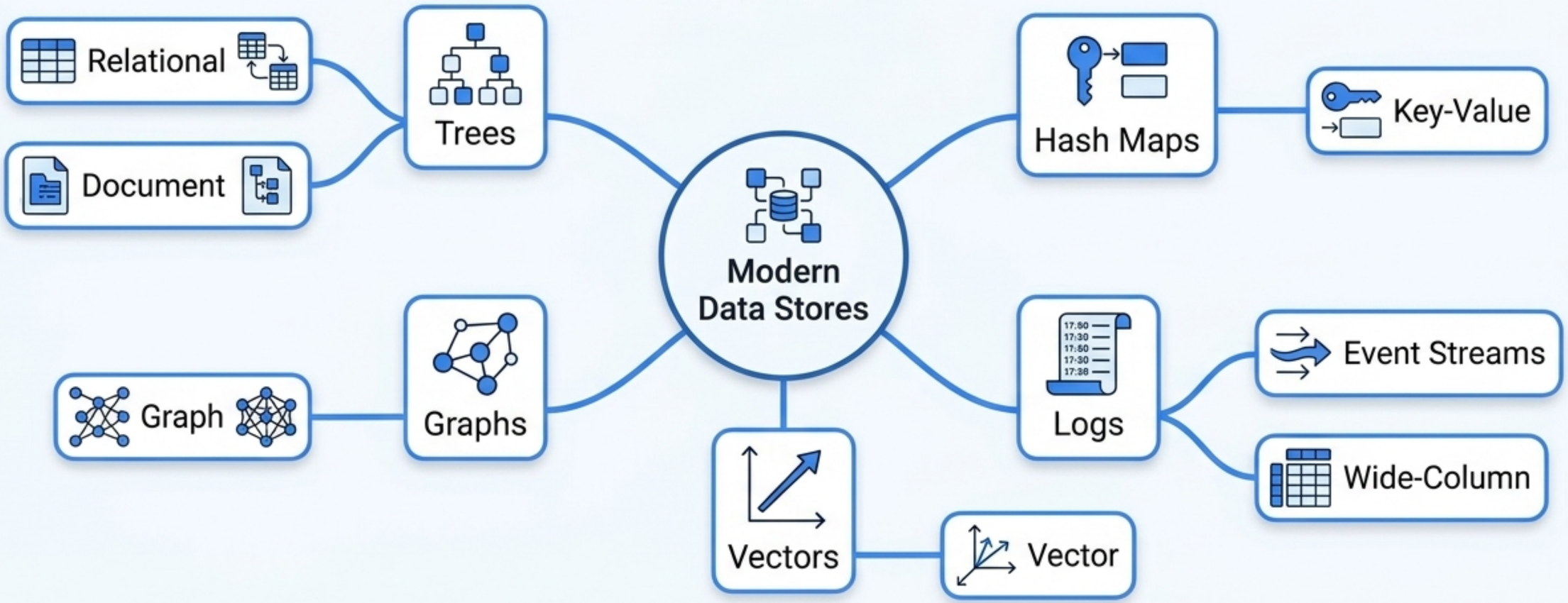


Iceberg/Delta Lake tracks individual data files, providing ACID transactions and time-travel over decentralized open formats. Eliminates proprietary vendor lock-in.

Gemini Notebook

The Data Store Spectrum: Beyond “SQL vs NoSQL”

“The modern landscape is defined by underlying data structures and access paths, not query languages.”



Gemini Notebook

Deep Dive: Relational, Lakehouse & Key-Value

Relational (OLTP)

Tech
PostgreSQL, MySQL

Structure
B-Trees & Heap Pages

Payload
Small / Medium

Use Case
Strict multi-entity ACID transactions. The default source of truth.

Lakehouse (OLAP)

Tech
Iceberg, Snowflake, DuckDB

Structure
Object Store + Metadata Logs

Payload
Petabyte scale / Columnar

Use Case
Massive historical analytical scans, MPP, and business intelligence.

Key-Value

Tech
Redis, Memcached

Structure
In-memory Hash Maps

Payload
Small, opaque values

Use Case
Extreme low-latency (<1ms) lookups and session caching.

Gemini Notebook

Deep Dive: Document, Wide-Column, Graph & Vector

Document

Tech

MongoDB

Structure

B-Trees over BSON

Use Case

Evolving aggregate shapes, polymorphic JSON data. Schema-on-read flexibility.

Wide-Column

Tech

Cassandra, DynamoDB

Structure

LSM Trees / SSTables

Use Case

High-ingest telemetry, partitioned linear scale, and time-series data.

Graph

Tech

Neo4j

Structure

Nodes & Edges (Index-free adjacency)

Use Case

Deep relationship traversals without expensive multi-table JOINs.

Vector

Tech

pgvector, Qdrant

Structure

HNSW / IVFFlat Graphs

Use Case

Semantic similarity and Approximate Nearest Neighbor (ANN) search.

Gemini Notebook

Master Taxonomy & Trade-Off Matrix

The definitive architectural cheat sheet.

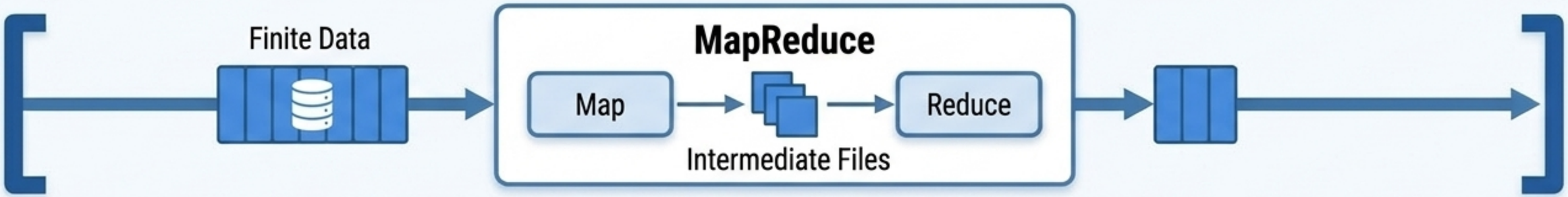
	Query Flexibility	p99 Latency	Horizontal Write Scale	ACID / PACELC
Relational (PostgreSQL)	High	<10ms	Sharded	ACID/CP
Lakehouse (Iceberg/Snowflake)	High	Seconds, Minutes	Near Infinite	Append-Only / ACID
Key-Value (Redis)	Low	<1ms	Sharded	Eventual/AP
Document (MongoDB)	Med	<10ms	Sharded	Eventual/AP
Wide-Column (Cassandra)	Low	<10ms	Near Infinite	Eventual/AP
Graph (Neo4j)	High	<100ms	Hard	ACID/CP
Event Stream (Kafka)	Low	<10ms	Near Infinite	Append-Only
Vector (pgvector)	Med	<100ms	Sharded	ACID/CP

Gemini Notebook

Processing Physics: Bounded Batch vs. Unbounded Streams

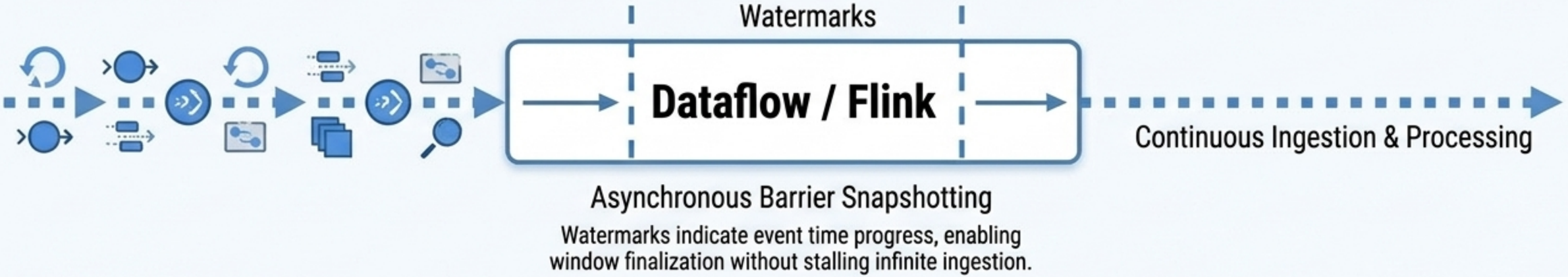
Bounded Batch (Data at Rest)

Finite dataset processed in its entirety. Job stops after completion.



Unbounded Stream (Data in Motion)

Infinite, continuous flow of events. Processing never stops, state is snapshotted.

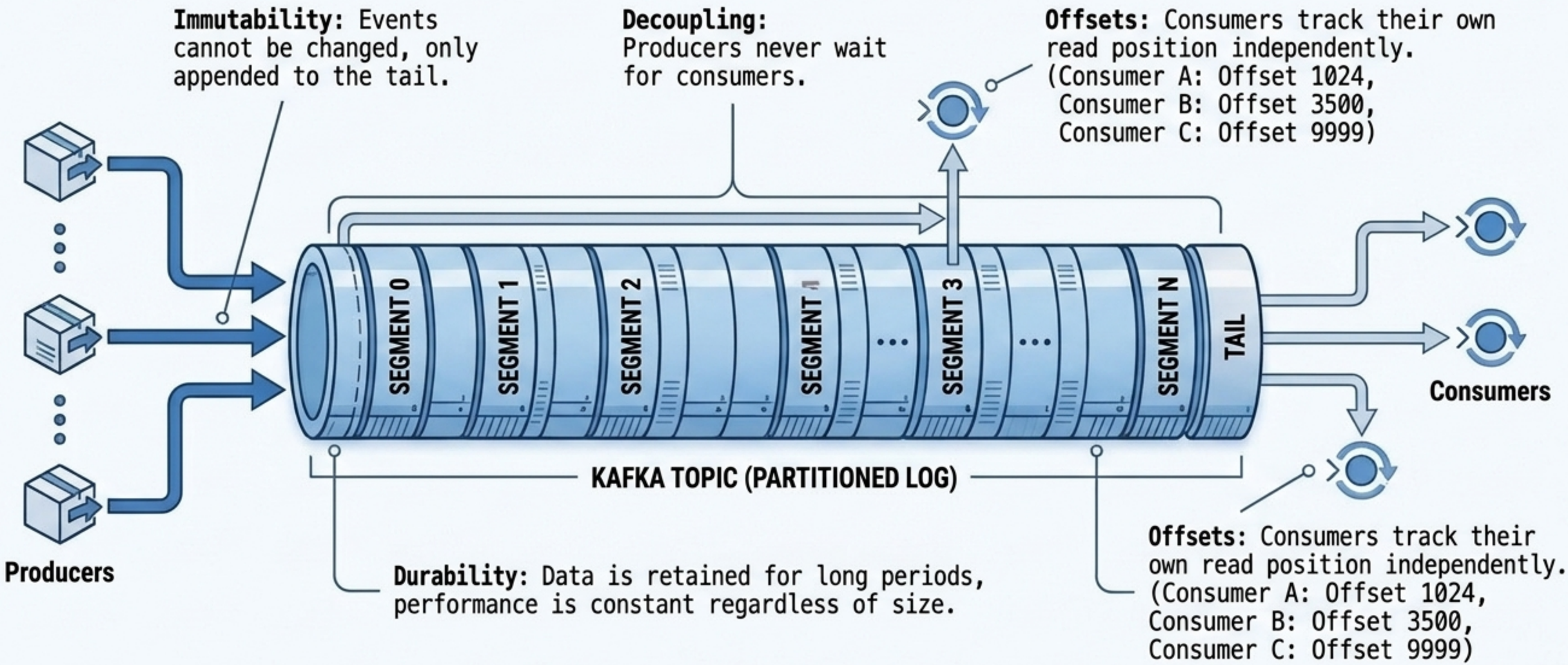


The paradigm shift from querying data at rest to processing data in motion.

Gemini Notebook

The Distributed Commit Log

The primitive that changes everything: the append-only log.



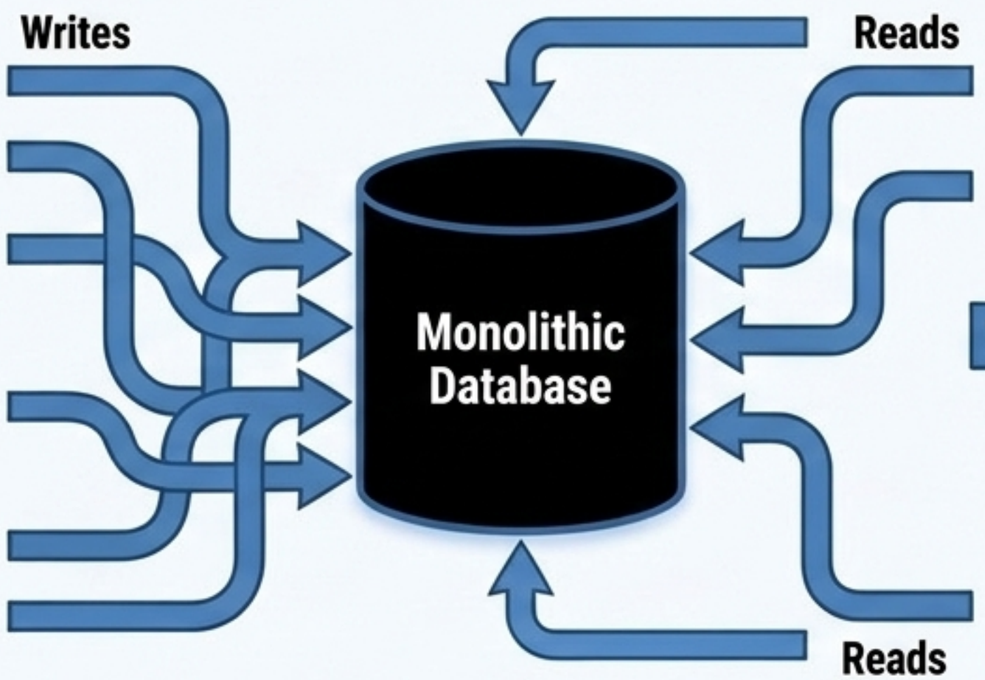
Gemini Notebook

Turning the Database Inside Out

Decouple the **writes** (an immutable stream of facts) from the **reads** (materialized, cached views updated in real-time).

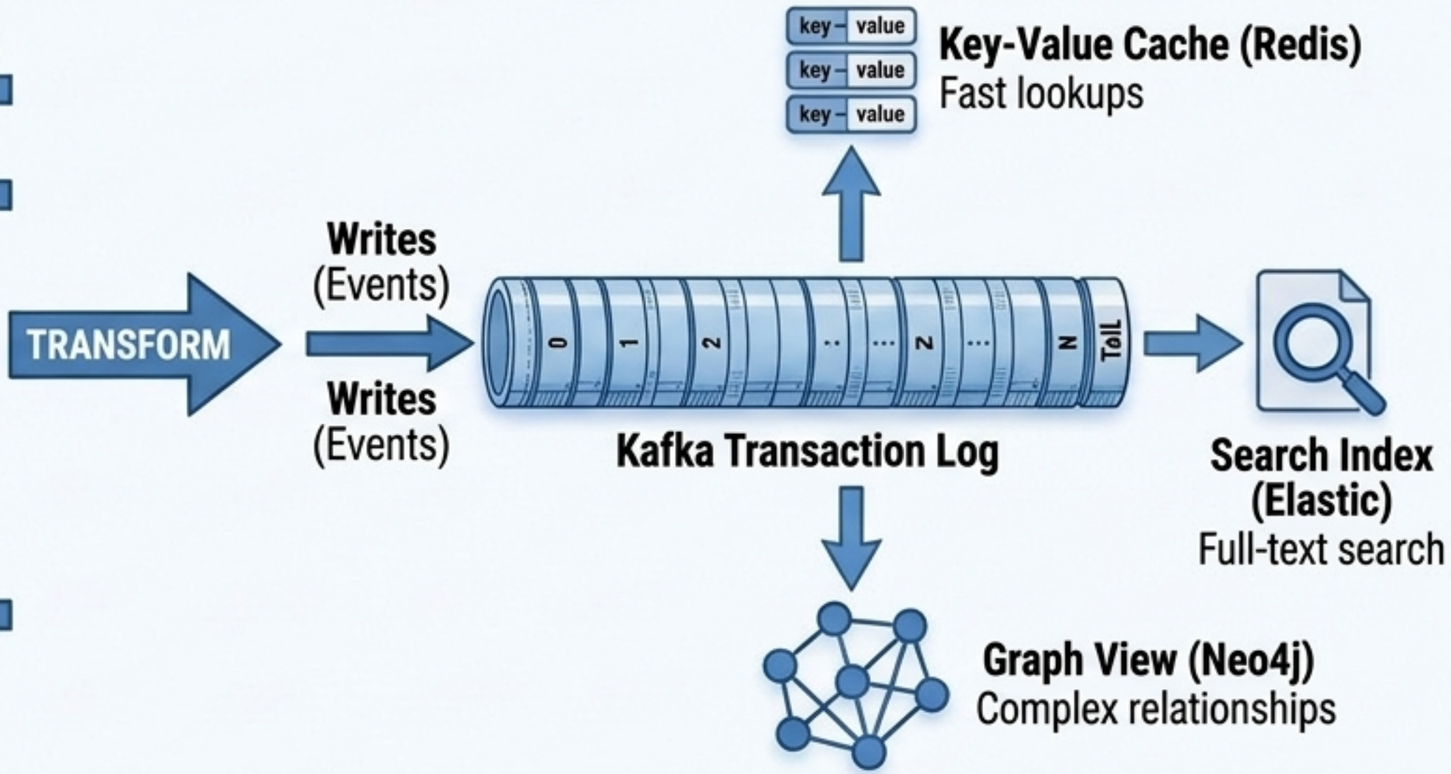
The Monolith

Traditional Approach



Inside Out (Stream-Centric)

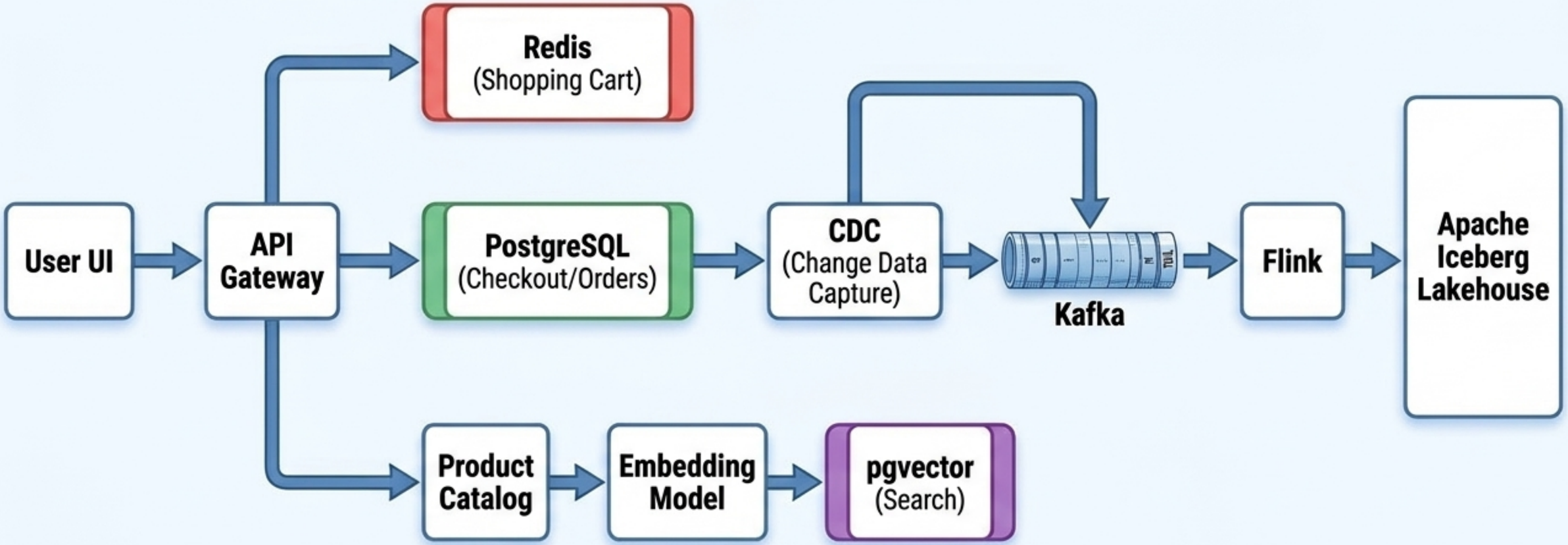
Decoupled Architecture



Gemini Notebook

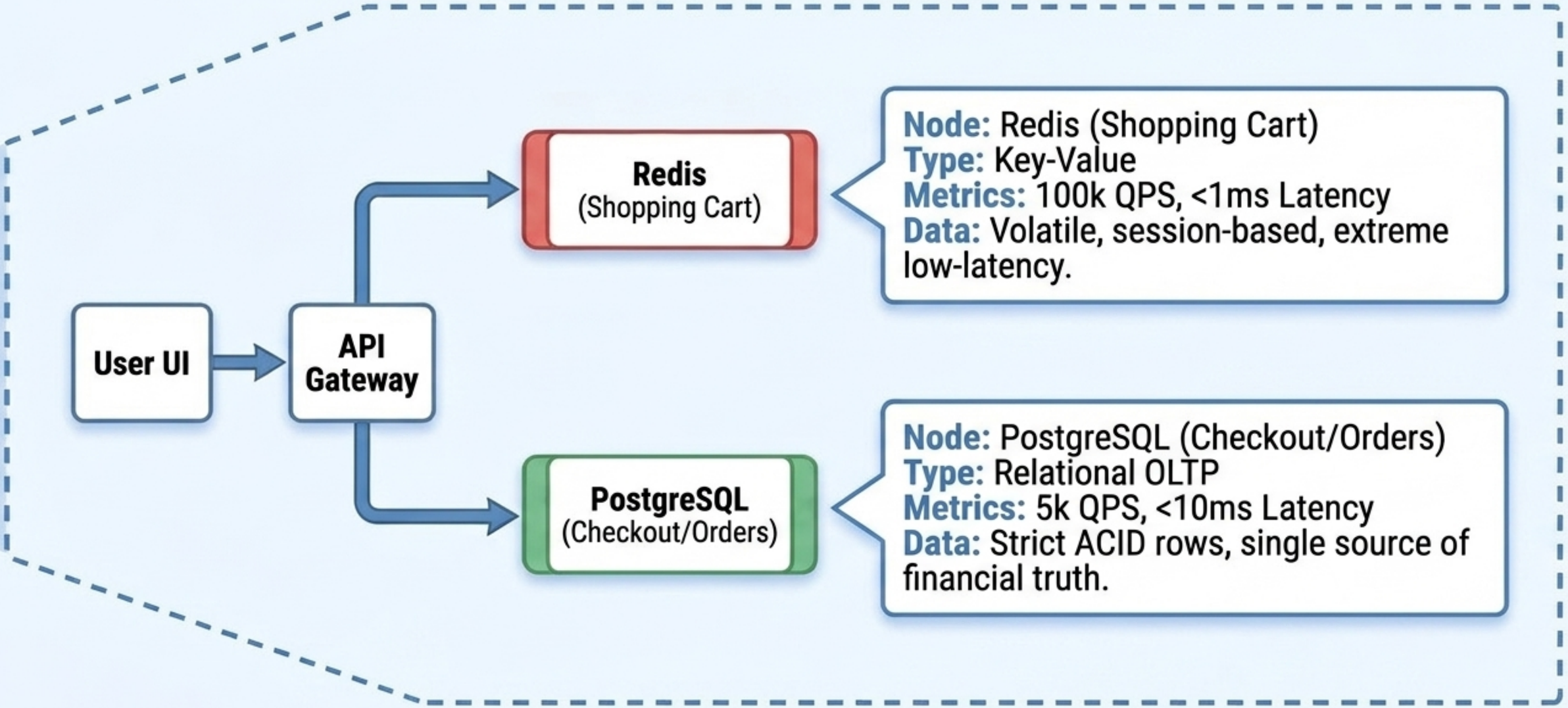
Worked Architecture Trace: E-Commerce Topology

Tracing a user journey across Polyglot Persistence—from sub-millisecond cart caching to semantic vector search.



Gemini Notebook

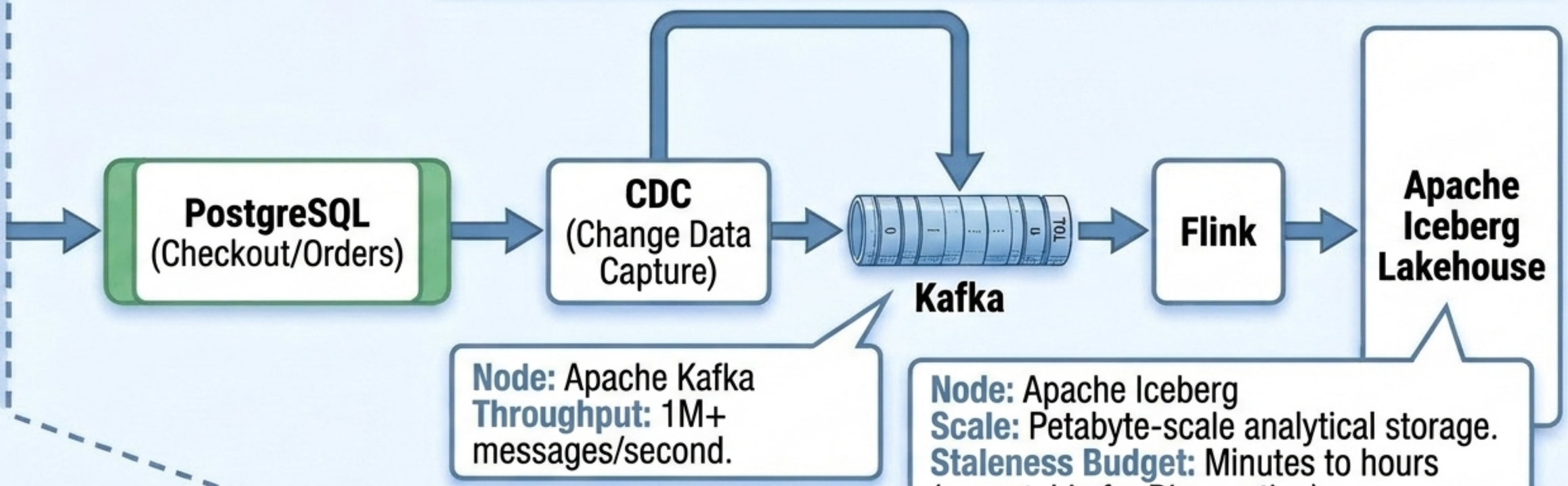
The Fast Path: Cart & Checkout



Gemini Notebook

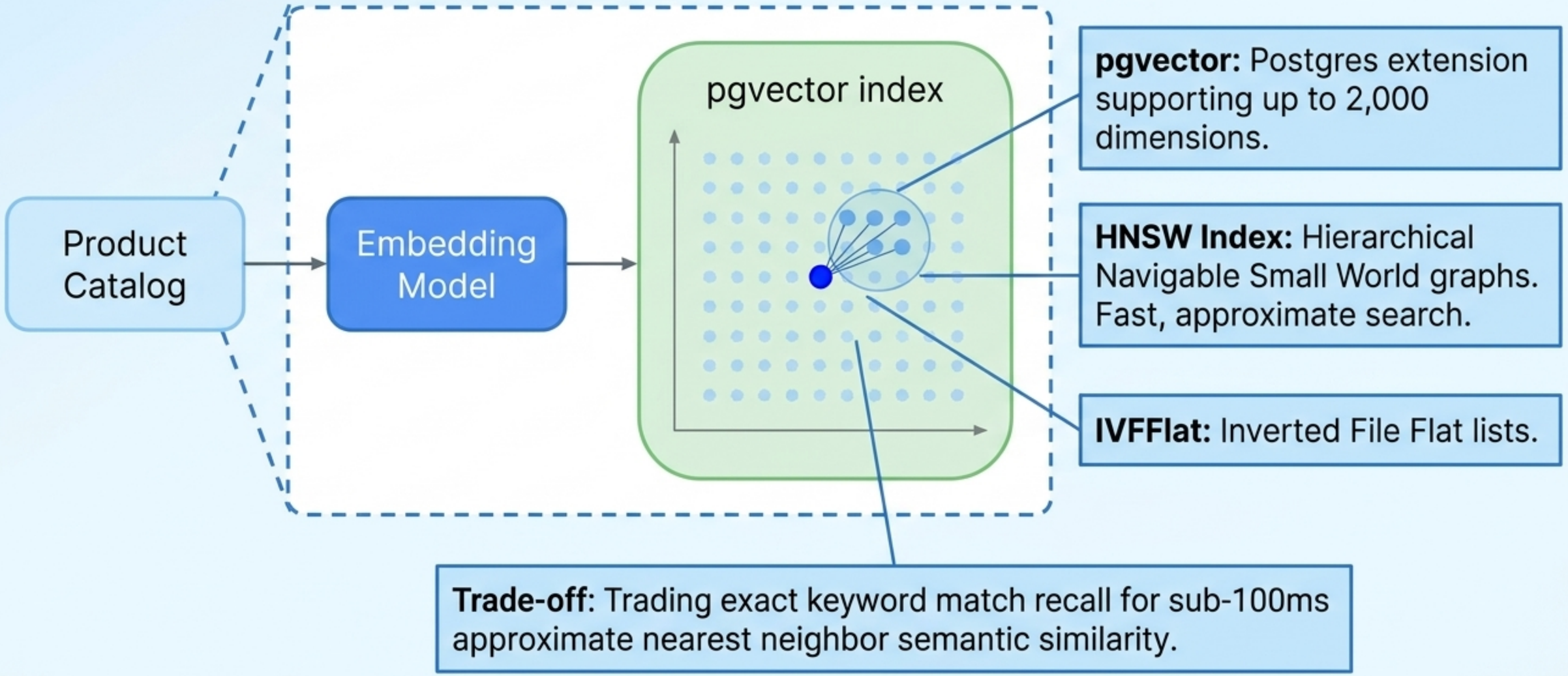
The Analytics Path: CDC to Lakehouse

Path: Streaming mutations out of OLTP via Change Data Capture to avoid bogging down production databases.



Gemini Notebook

The Semantic Path: Product Search



Gemini Notebook

Marketing Myths: ‘Schemaless’ and ‘Infinite Scale’

🚫 Myth

❌ **Myth 1:** Schemaless Document stores mean you don’t need a schema.

❌ **Myth 2:** Our NoSQL database provides infinite horizontal scale.

✅ Reality

✓ **Reality 1:** Schema-on-read means the schema is simply hidden, hardcoded in your application logic. It didn’t disappear.

✓ **Reality 2:** True, but consistent hashing and partitioning mean that any cross-partition JOIN will immediately destroy your p99 latency.

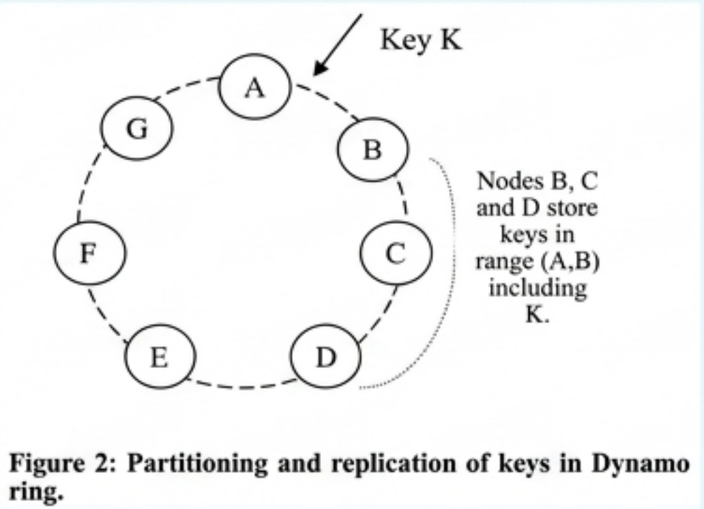
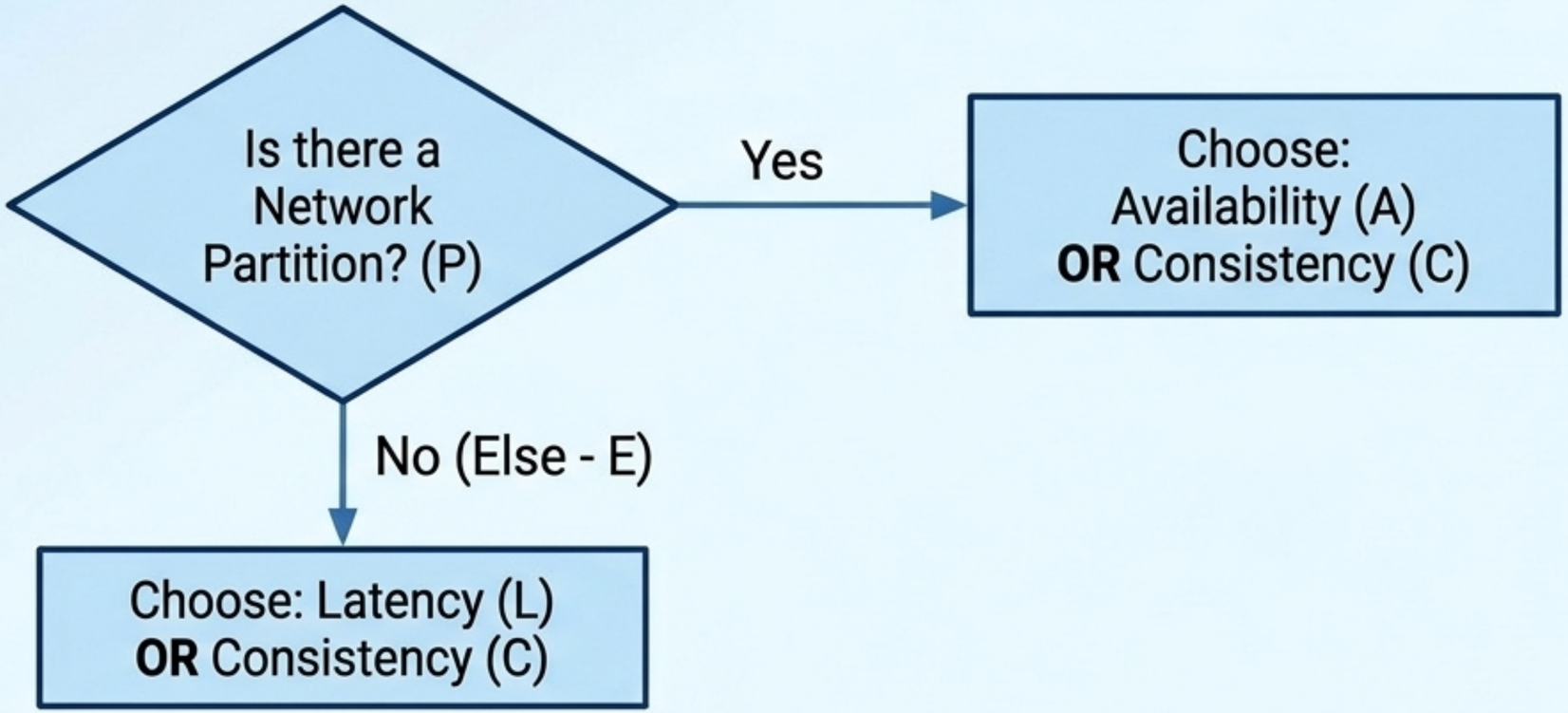


Figure 2: Partitioning and replication of keys in Dynamo ring.

Gemini Notebook

PACELC and the Consistency Reality Check



⚠️ **Jepsen's Law**: Network partitions WILL happen. You must design for degradation.

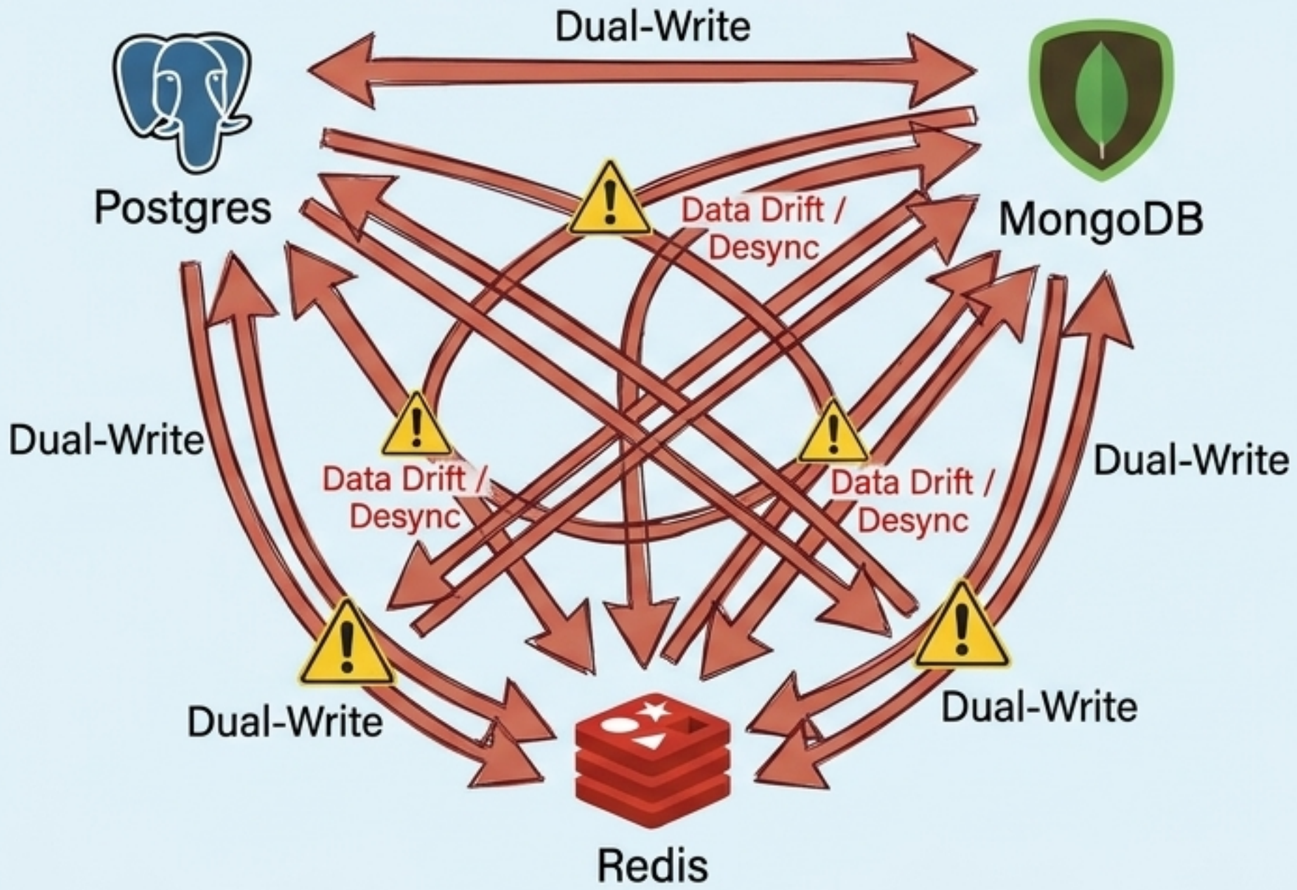
🔒 **Serializability**: Guarantees history is equivalent to a totally ordered execution.

⚠️ **Phenomena**: Beware of G1a (Aborted Reads) and divergent vector clock sub-histories. Naive 'CP/AP' slogans obscure real database behavior.

Gemini Notebook

Polyglot Persistence Traps

Martin Fowler's "Polyglot Persistence" is powerful but expensive.



- Dual-write consistency bugs will silently corrupt your systems.
- Supporting 5 different query languages creates massive cognitive load.
- Only use an alternative persistence model for STRATEGIC domains, not utility apps.

Gemini Notebook

The Workload Intake Checklist

Before provisioning a new database, answer these:

```
[ ] Target QPS (Queries Per Second)?
    -> Dictates compute provisioning and engine type.
[ ] Average Payload Size?
    -> KBs vs GBs dictates storage engine.
[ ] p99 Latency Budget?
    -> Sub-millisecond vs Minutes.
[ ] RPO / RTO Requirements?
    -> Recovery Point / Time Objectives.
[ ] Ops Headcount?
    -> Who is carrying the pager for this cluster?
```

Gemini Notebook

You Now Have the Map.

Series Roadmap & S05E01 Preview



Next week in S05E01: OLTP vs OLAP
A grueling deep dive into physical Page Layouts, B-tree fragmentation, and Vectorized Query Execution engines.

Gemini Notebook