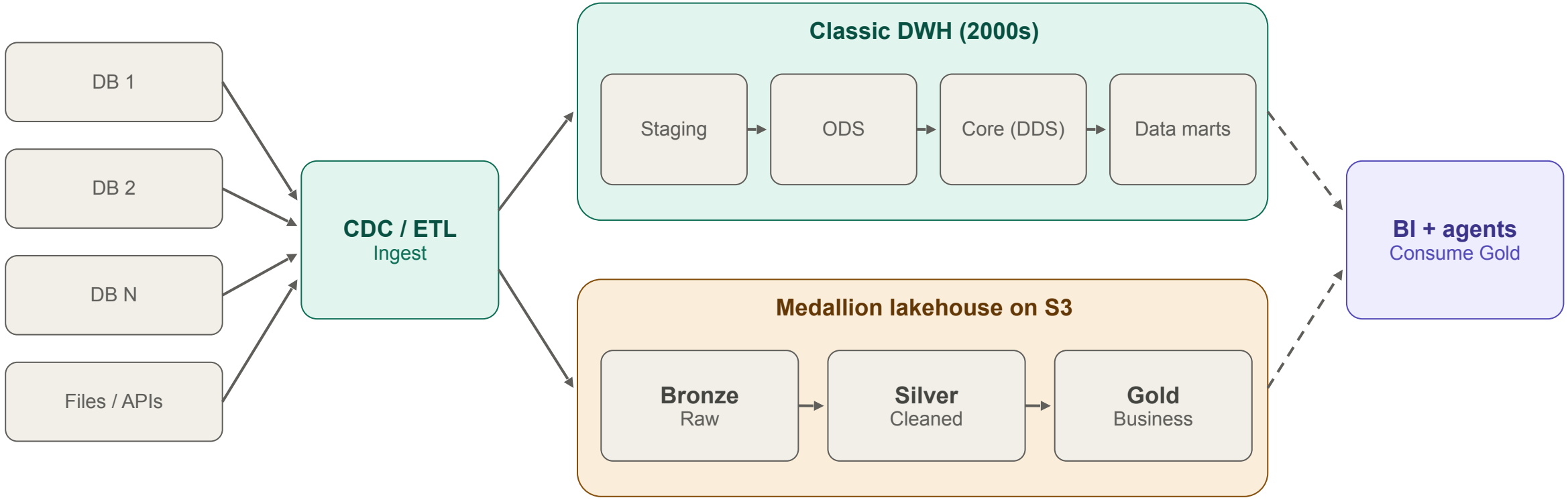
A · Classic layered approach — DWH and Medallion

Sources → CDC / ETL → layered transformation → consumers; two generations of one idea

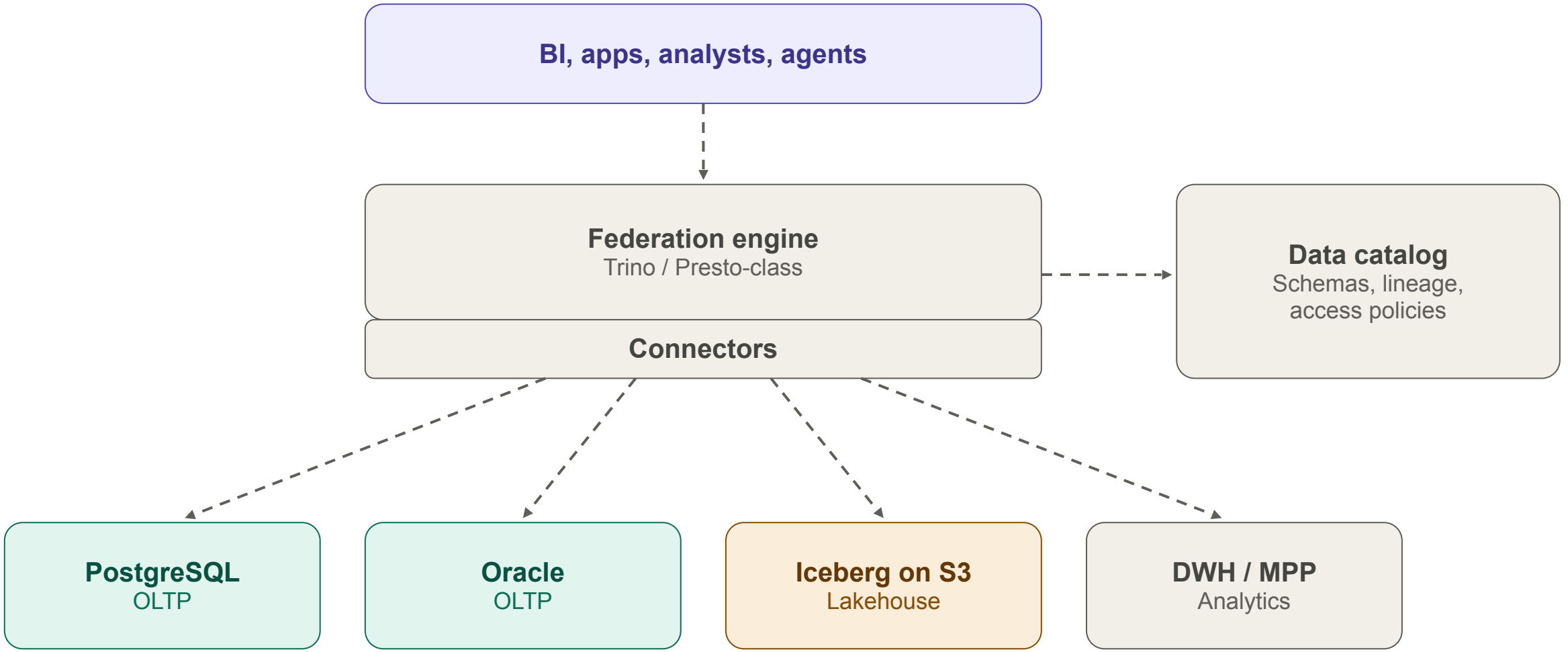


Key idea: Copy everything, then refine it layer by layer before use; Medallion renames the classic DWH layers.

Solid arrow = data written / replicated Dashed arrow = live read / query

B · Federation — data stays in place

Trino / Presto-class engine with connectors and a data catalog

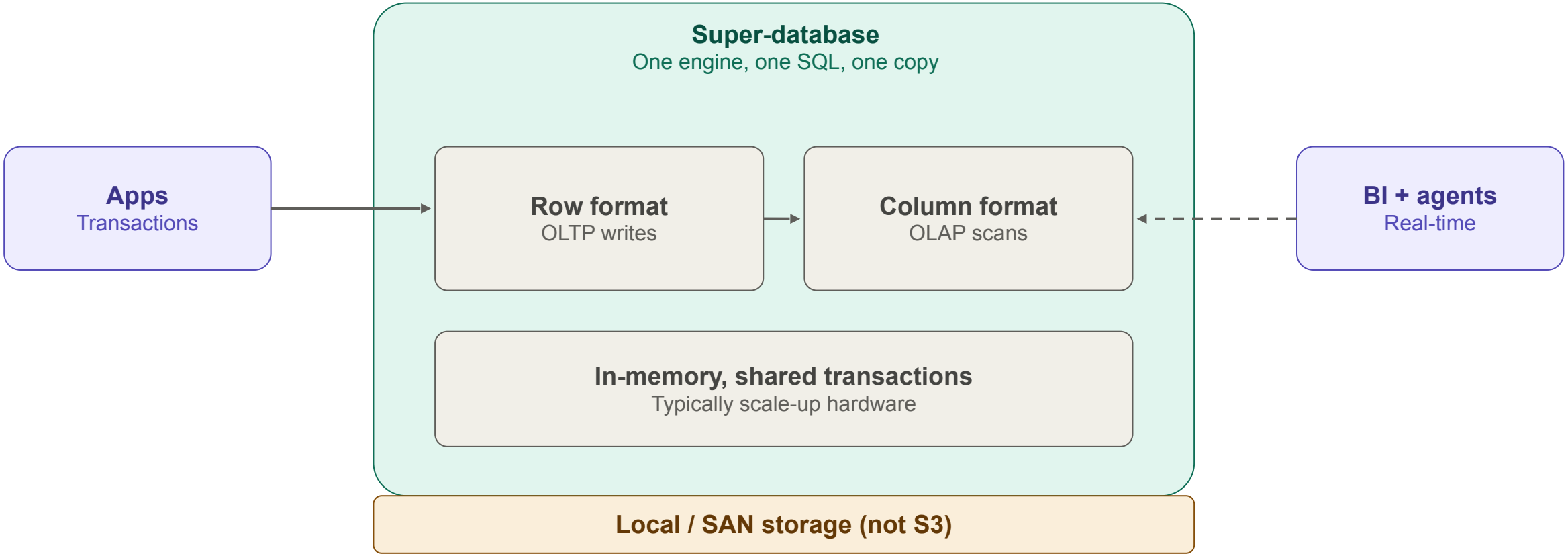


Key idea: No copies: one engine pushes queries down through connectors and joins results at read time.

Solid arrow = data written / replicated Dashed arrow = live read / query

C · Super-database — OLTP + OLAP in one engine

SAP HANA, Oracle Database In-Memory, SingleStore, TiDB-class HTAP

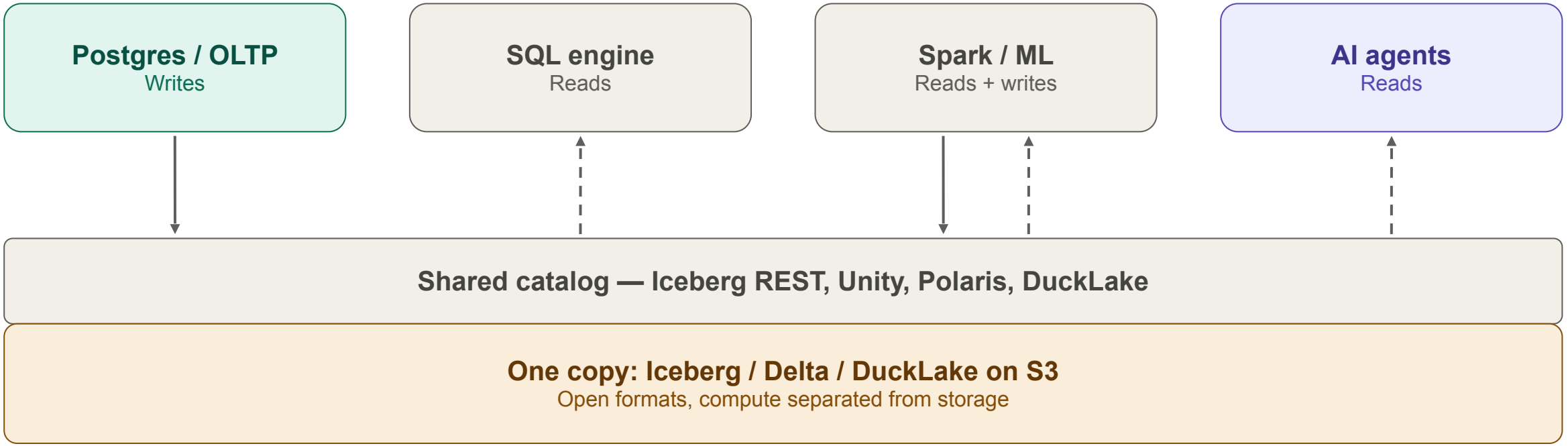


Key idea: No data movement: one engine keeps row and column formats in sync inside one transaction boundary.

Solid arrow = data written / replicated Dashed arrow = live read / query

D · Unified storage — one copy, many engines

Open table formats on object storage, shared catalog, independent engines

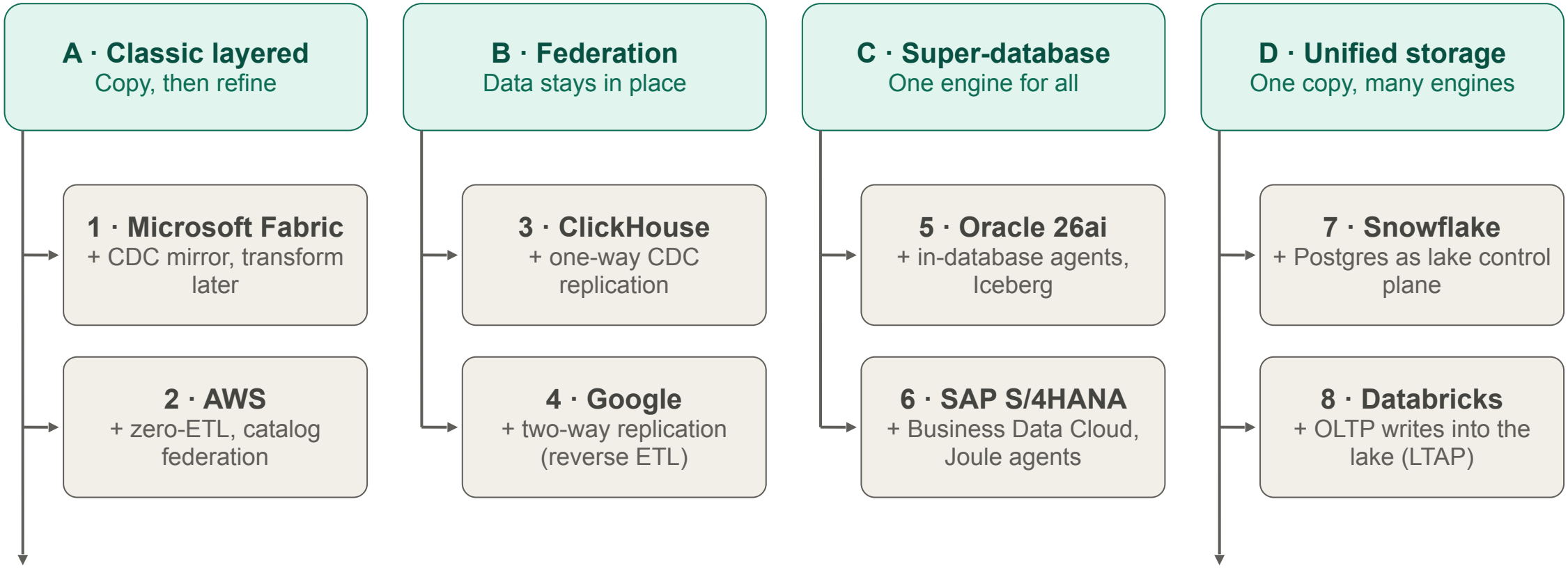


Key idea: Engines are interchangeable; data is one open copy on S3 — control shifts to whoever owns the catalog.

Solid arrow = data written / replicated Dashed arrow = live read / query

Lineage: four archetypes → eight descendant architectures

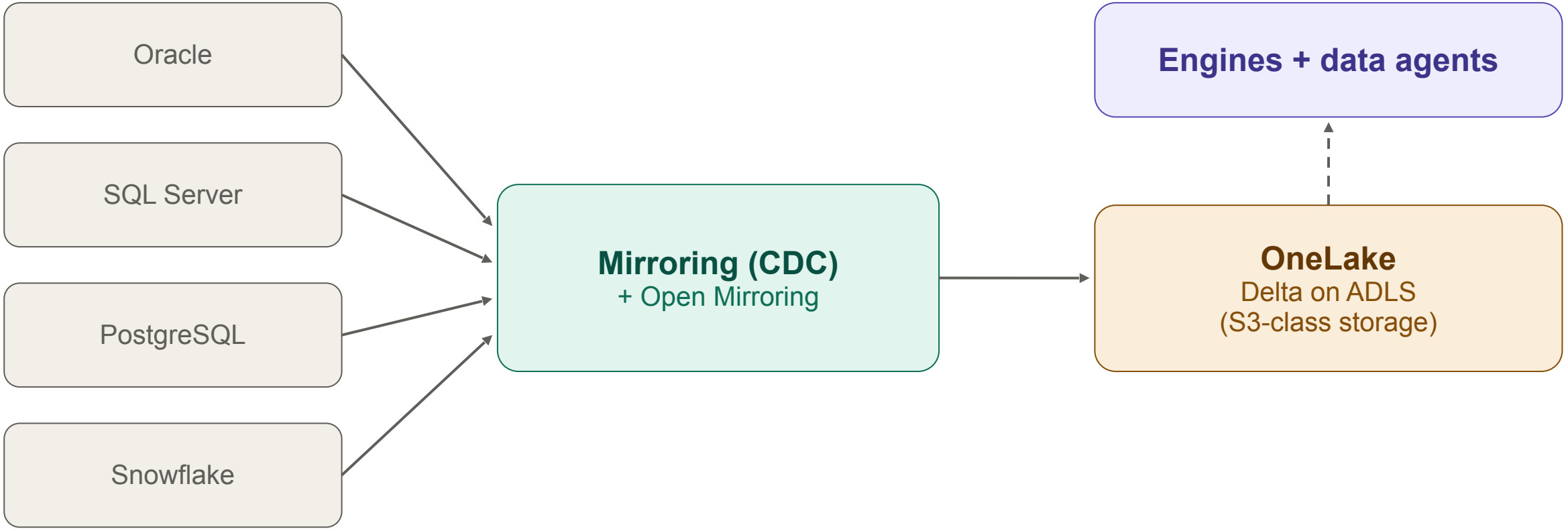
Each descendant = its archetype + one small change



Key idea: Hybrids are the norm, and MCP-connected agents now sit on top of every lineage.

1 · Mirror everything into one lake — Microsoft Fabric

Mirroring + Open Mirroring into OneLake (Delta)

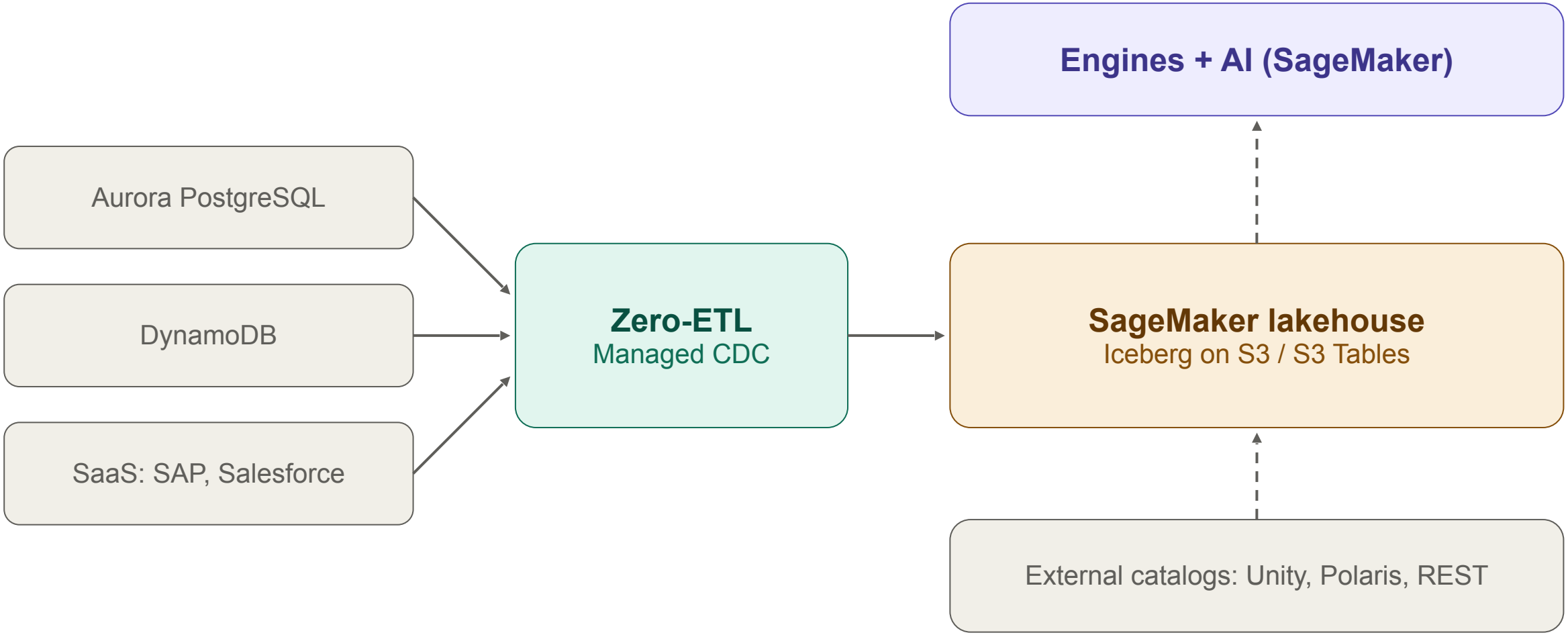


Key idea: Each source's own CDC streams into one lake on object storage.

Solid arrow = data written / replicated Dashed arrow = live read / query

2 · Zero-ETL + catalog federation — AWS

Aurora / DynamoDB / SaaS zero-ETL into the SageMaker Iceberg lakehouse

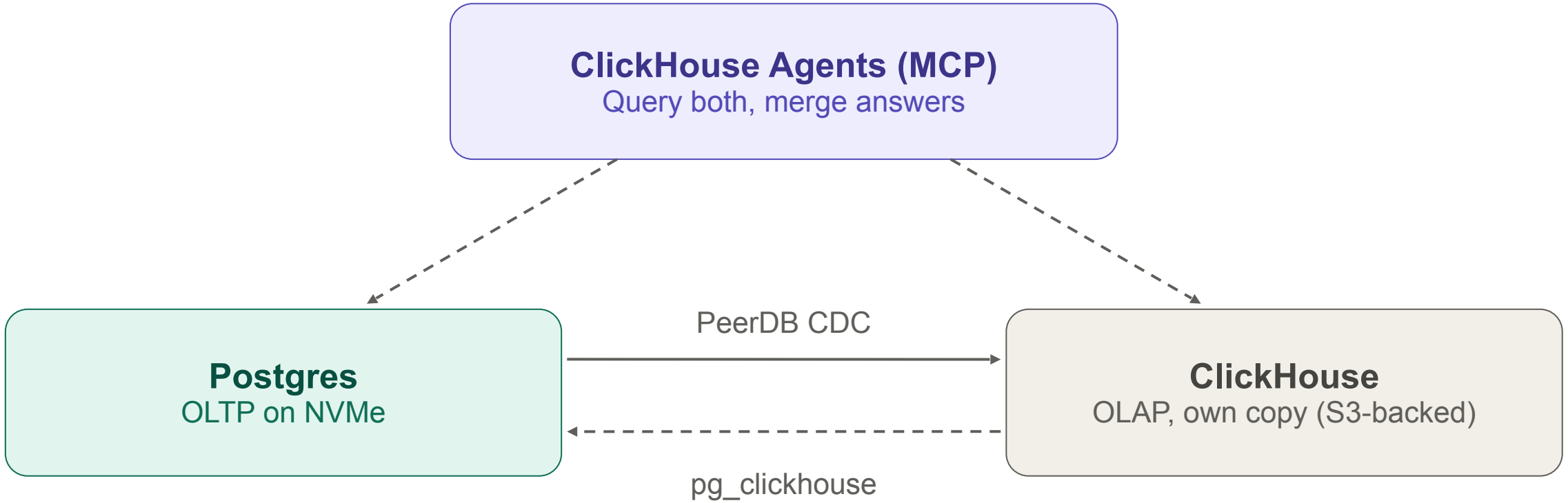


Key idea: Managed CDC lands Iceberg on S3; other vendors' Iceberg catalogs are read without copying.

Solid arrow = data written / replicated Dashed arrow = live read / query

3 · Best-of-breed pair + federation — ClickHouse

Managed Postgres + ClickHouse, PeerDB CDC, pg_clickhouse, ClickHouse Agents

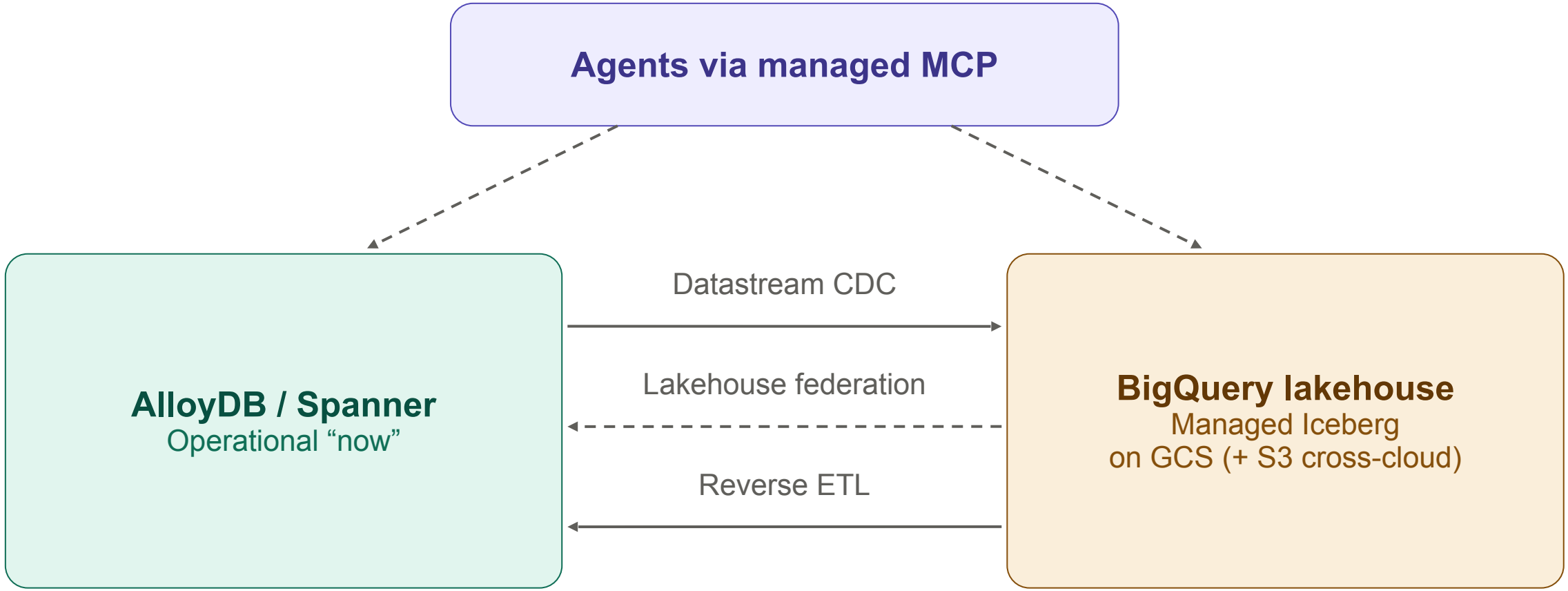


Key idea: No shared storage: two engines, two copies, glued by CDC one way and queries the other.

Solid arrow = data written / replicated Dashed arrow = live read / query

4 · Closed loop — Google Agentic Data Cloud

AlloyDB / Spanner + BigQuery lakehouse, Datastream, Reverse ETL, managed MCP

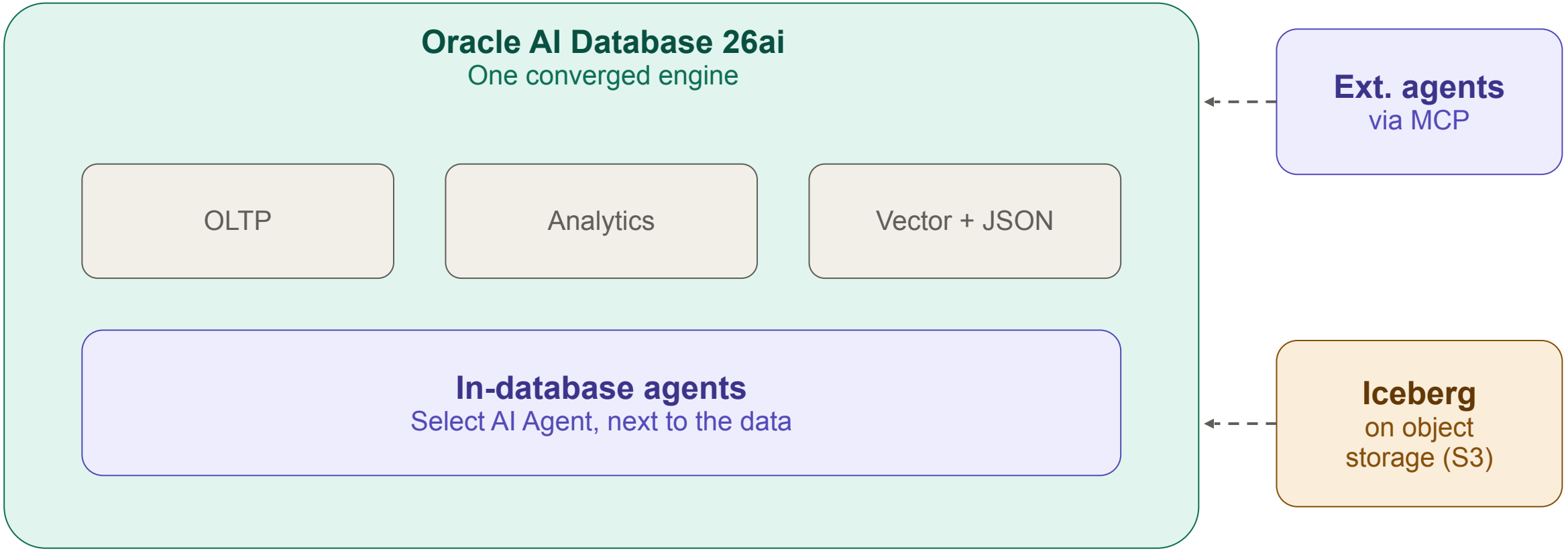


Key idea: The only bidirectional design: CDC in, insights pushed back to OLTP, federation in between.

Solid arrow = data written / replicated Dashed arrow = live read / query

5 · Single converged engine — Oracle

Oracle AI Database 26ai with in-database agents and Autonomous AI Lakehouse

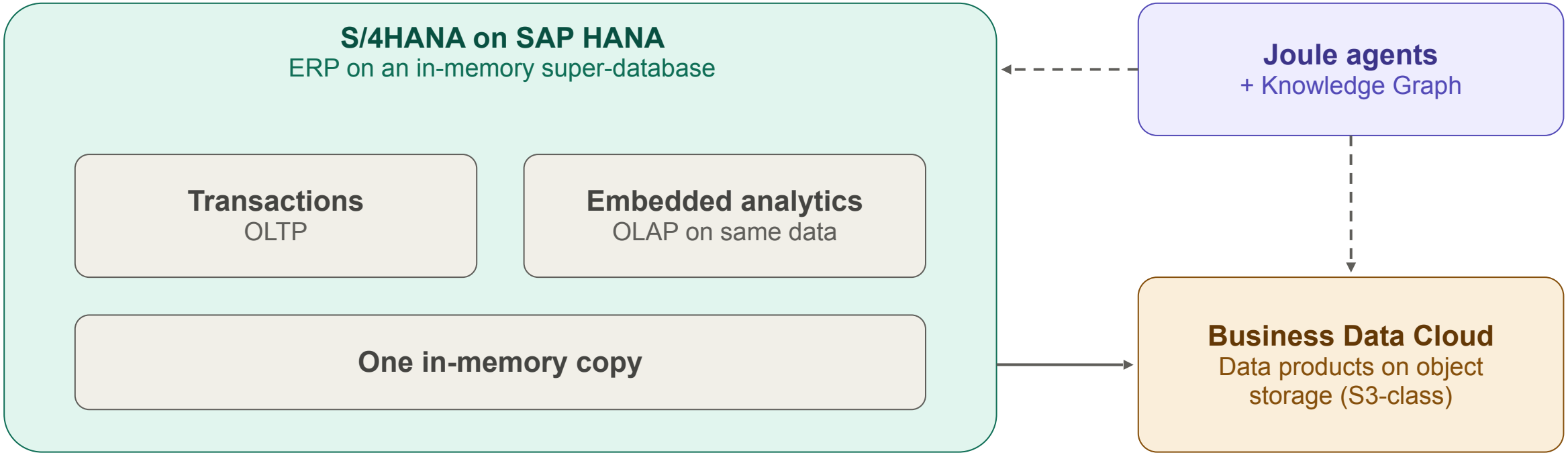


Key idea: Opposite bet to unbundling: one engine does everything; agents live inside the database.

Solid arrow = data written / replicated Dashed arrow = live read / query

6 · ERP on a super-database — SAP S/4HANA

S/4HANA on SAP HANA + Business Data Cloud (Databricks partnership) + Joule agents

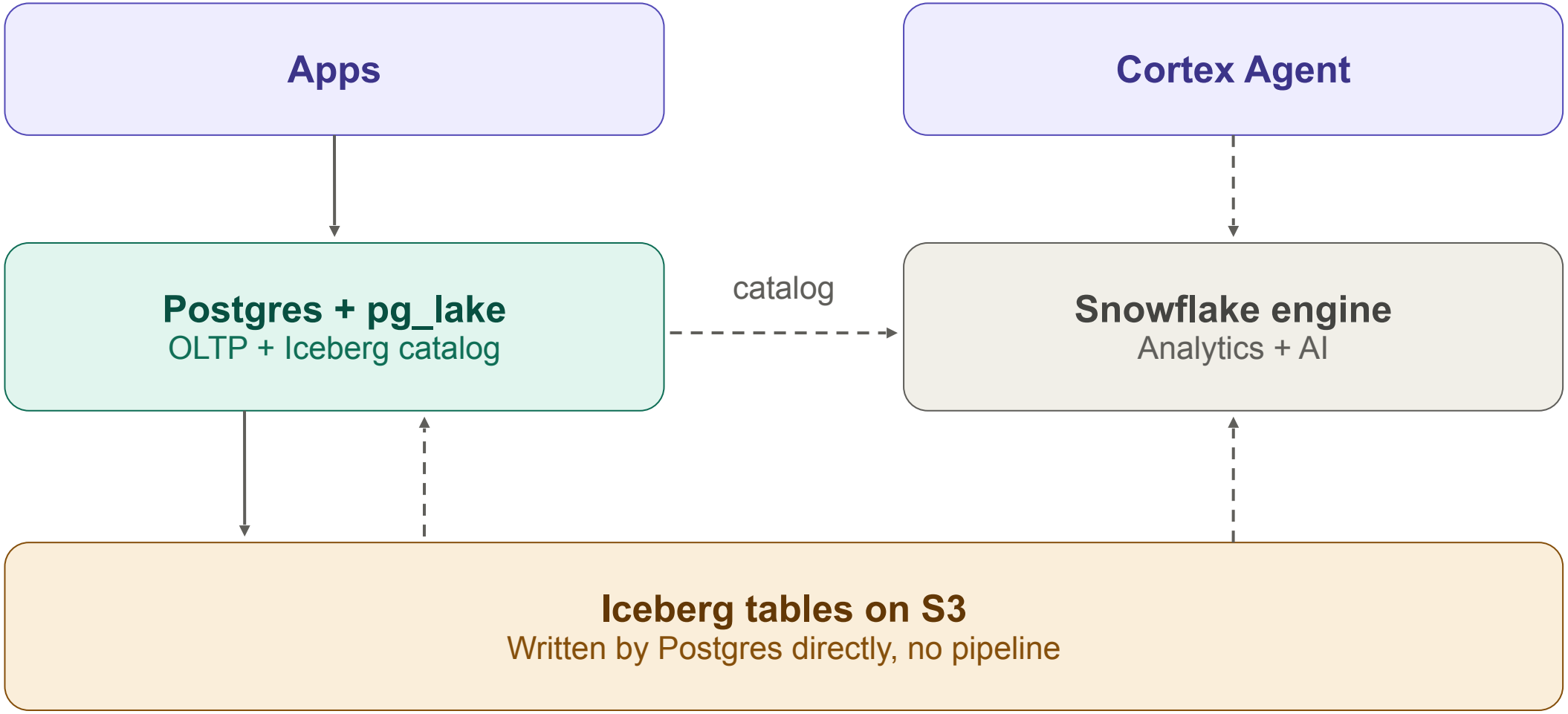


Key idea: The super-database stays the system of record; a data-product cloud and agents are added around it.

Solid arrow = data written / replicated Dashed arrow = live read / query

7 · Postgres as lakehouse front door — Snowflake

Snowflake Postgres + open-source pg_lake (from Crunchy Data)

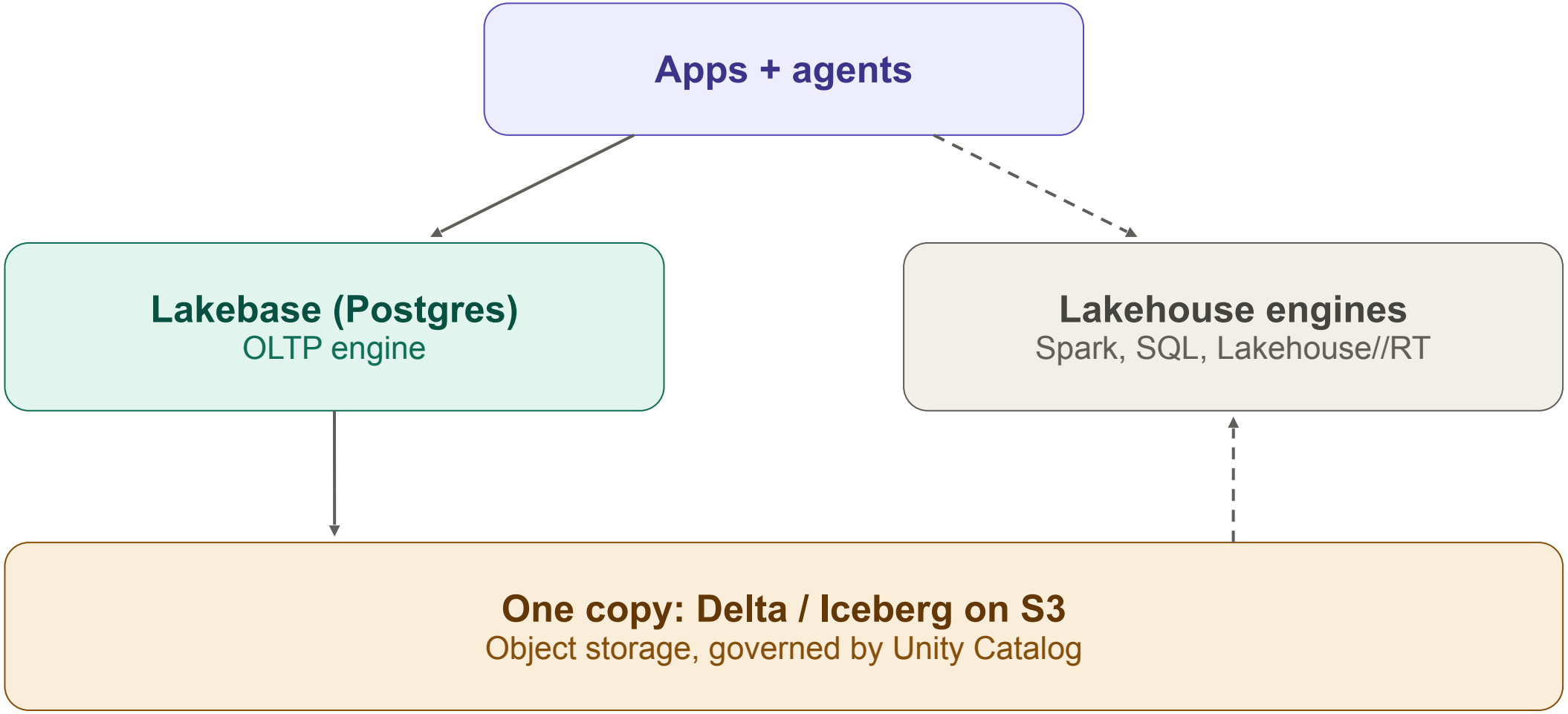


Key idea: Postgres is both the OLTP engine and the control plane that writes and catalogs Iceberg on S3.

Solid arrow = data written / replicated Dashed arrow = live read / query

8 · Storage-layer unification — Databricks LTAP

Lakebase (Postgres) + Lakehouse on one copy of data; Mooncake mirrors Postgres into the lake



Key idea: Not HTAP: two engines, one physical copy on S3; each table has a single writer.

Solid arrow = data written / replicated Dashed arrow = live read / query

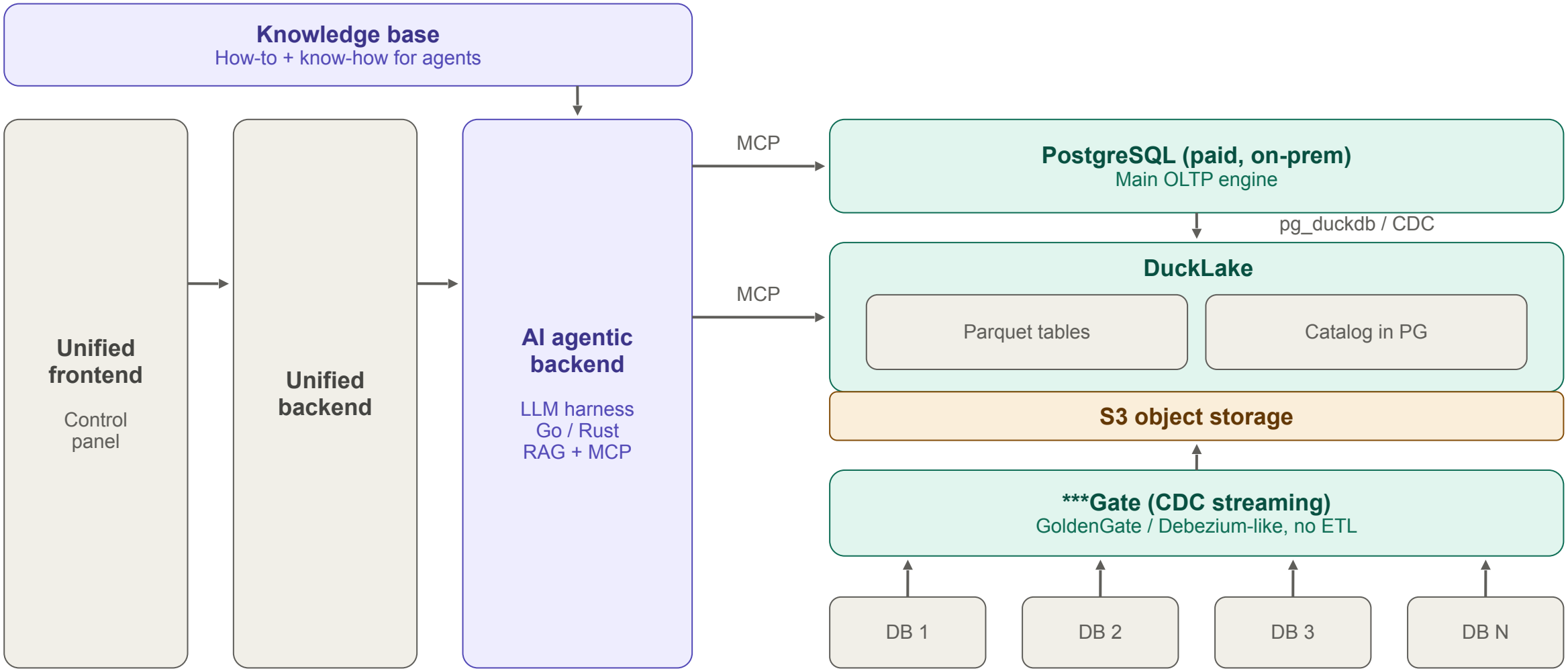
Sheet of options: where does unification happen?

Eight descendant architectures vs. our reference picture

#	Arch.	Vendor / approach	Unification at	Copies	Flow direction	Unified storage
1	A	**Microsoft Fabric**	Replication	Two	One-way, into the lake	OneLake on ADLS (S3-class)
2	A	**AWS zero-ETL + SageMaker**	Replication + catalog fed.	Two	One-way, into the lake	S3 / S3 Tables: Iceberg
3	B	**ClickHouse + Postgres**	Query / federation	Two	CDC one way, queries back	S3-backed ClickHouse
4	B	**Google Agentic Data Cloud**	Closed loop	Two	Both ways: CDC in, reverse ETL out	GCS (+ S3 cross-cloud)
5	C	**Oracle AI Database 26ai**	Single engine	One	None needed (inside engine)	Iceberg on object storage (S3)
6	C	**SAP S/4HANA + BDC**	Single engine + data products	Two	ERP in-engine; data products out	HANA in-memory; BDC object storage
7	D	**Snowflake Postgres + pg_lake**	Storage layer	One	Postgres writes Iceberg directly	S3: Iceberg
8	D	**Databricks LTAP**	Storage layer	One	OLTP writes lake format directly	S3: Delta / Iceberg
9	**A+D**	**Our reference picture**	**Replication (+ storage via pg_ducklake)**	**Two**	**One-way, into DuckLake**	**S3: DuckLake Parquet**

9 · Reference picture: AI / Data platform (appendix)

PostgreSQL OLTP + DuckLake on S3, fed by CDC (***Gate); agentic backend over MCP



Key idea: Throw away ETL — stream every source into bottomless storage on S3 for later agentic use.

Solid arrow = data written / replicated Dashed arrow = live read / query