



No, HTAP in the Cloud Is Not Solved – The Industry Traded Freshness for Scalability and Called It Innovation!

Daniel Ritter, SAP
HPTS 2026

Despite Ongoing Controversial **HTAP** Discussions by Industry

NEON

← BLOG / ENGINEERING

HTAP is Dead

After InfoQ Homepage > News > HTAP: T...

AI, ML & DATA ENGINEERING

HTAP: the... base Systems?

NEWS · JUNE 29, 2026 · 4 MINUTES

CedarDB Launches HTAP (10 Years Ago)

Unifying transactions and analytics is once again making news headlines as the holy grail for AI agents. But it is hardly news. Unbra, the research behind CedarDB, started 10 years ago, and even then it was not the first. With CedarDB, what other vendors still call impossible is running in production today.

HTAP: S... Later

Databricks +

HTAP May Converge ≠ Coherence.

Architecture

A decade after HTAP stalled, ... OLAP are converging again — open table formats, mature CDC, engines moving to the middle. The convergence is real. But it answers where your data lives, not whether an automated decision read one coherent version of it at the instant it fired. Those are different problems.

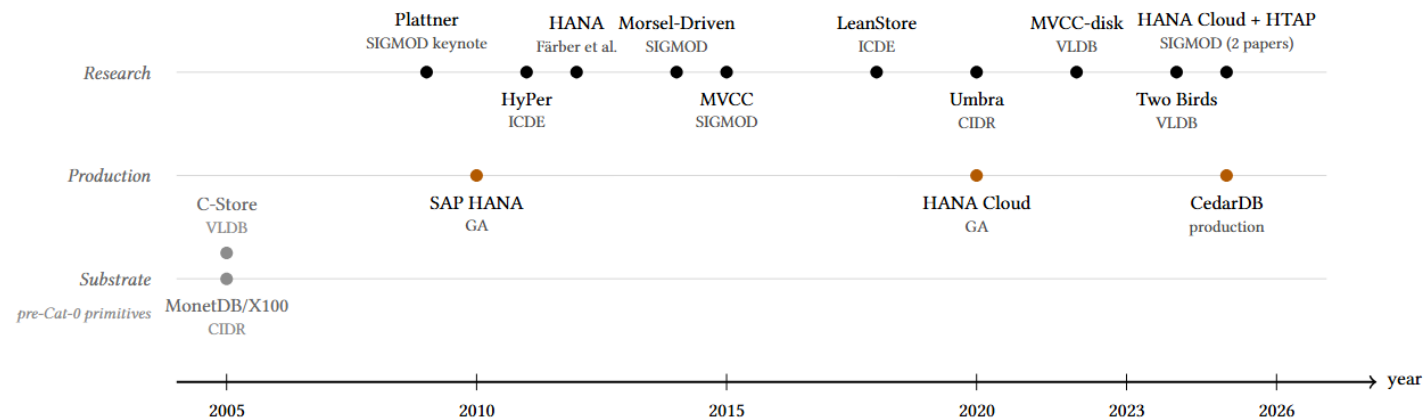
Alex Kimball
Product Marketing

Sep 23, 2026 · 10 min read

actually changed



Position #1.1: Twenty Years of HTAP: Operational Analytics Is Solved (~2005/2008—2026)



What is 2x2?

[system 1 ~ mat. view]

And 17x24?



Trucks queuing at the gate while replicating numbers to [system 2] compute ...

Recall | Original HTAP from 2008

Was born from a **workload observation**: the same business entity (e.g., payment, a stock movement) needs to be both

- **written** (OLTP: short writes, high concurrency, point access) and
- **immediately read analytically** (OLAP: large scans, aggregates, joins) in the **same decision window**. [**CedarDB**: “**same system**”]

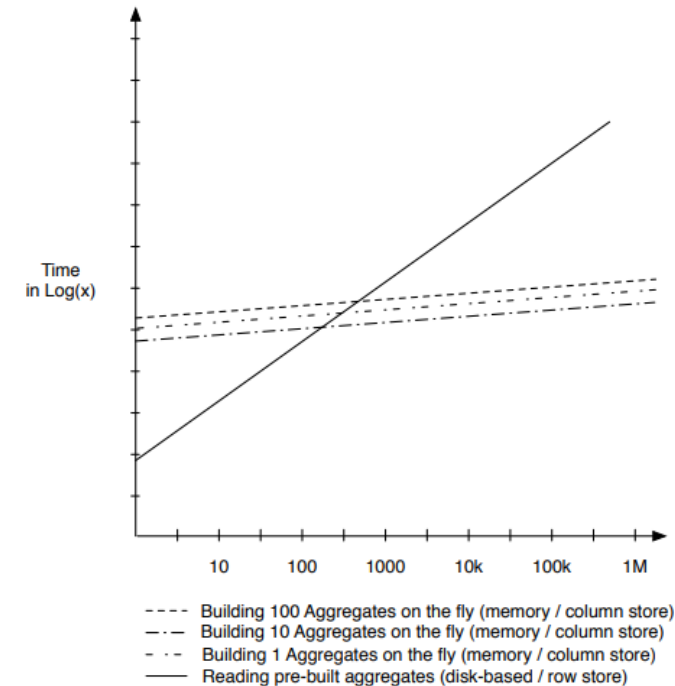
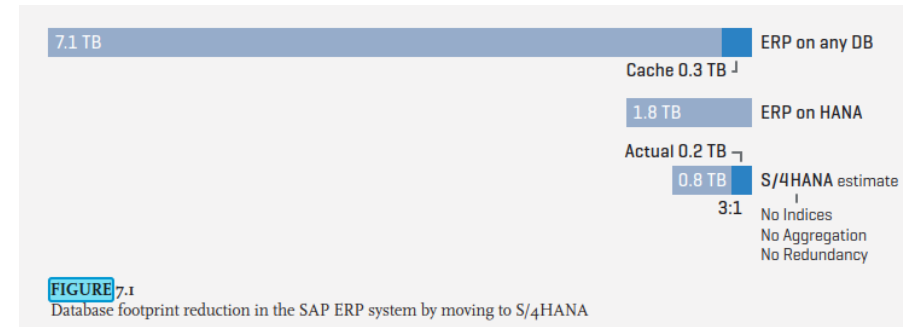
Most of the processing is management of aggregates (~80%) ~ could not focus on enterprise / business logic

and strived to:

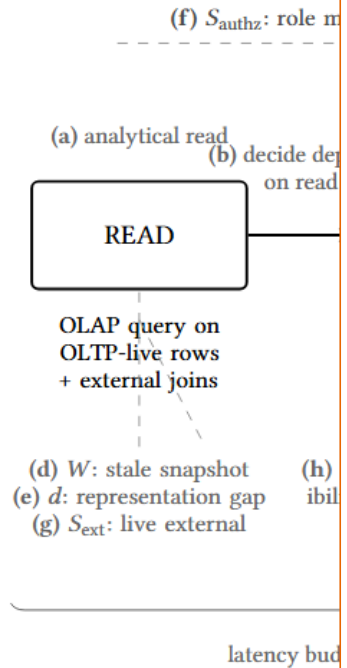
- Eliminate the **replication copy** entirely
- Avoid large amount of **materialized views**
- In-memory, column store, **write-back atomicity in a single semantic domain**

A Common Database Approach for OLTP and OLAP Using an In-Memory Column Database

Hasso Plattner
Hasso Plattner Institute for IT Systems Engineering
University of Potsdam
Prof.-Dr.-Helmert-Str. 2-3
14482 Potsdam, Germany
hasso.plattner@hpi.uni-potsdam.de



HTAP Systems



Use case	Property	Why replication fails
<i>Finance, Accounting & Risk (FA&R)</i>		
Real-time fraud detection	$S_{wb} + F$	decision read against in-flight writes
Intraday liquidity / AML	$S_{wb} + F + S_{authz}$	decision read against in-flight writes; role-propagation lag during regulated write-back
Dunning procedure	S_{wb}	operational write-back on analytical trigger
Financial anomaly screening	$S_{wb} + S_{authz}$	cross-source join without global coordinator; regulated write-back on live payment run
<i>Sales, Marketing, Order Management & Customer (S&C)</i>		
Margin analytics at order entry	$S_{wb} + F$	decision read against in-flight writes
Customer segmentation (hybris)	S_{wb}	operational write-back on analytical trigger
Checkout bonus calculation	$S_{wb} + F$	decision read against in-flight writes
Real-time customer segmenta-	$S_{wb} + F$	sub-second decision window shorter than replication lag
tion		
Signal		
Omni-		
<i>Supply Chain, Logistics, Procurement & Inventory (SC&L)</i>		
Live stock / MRP trigger	S_{wb}	operational write-back on analytical trigger
S&OP planning (IBP)	S_{wb}	operational write-back on analytical trigger
Warehouse labour planning	S_{wb}	operational write-back on analytical trigger
Empty container planning	S_{wb}	operational write-back on analytical trigger
smartPORT Hamburg	$S_{wb} + F$	sub-second decision window shorter than replication lag
<i>Industry 4.0, Operations & Manufacturing (I4&M)</i>		
Hydraulic pump maintenance	S_{wb}	operational write-back on analytical trigger
ConAgra Total Margin Mgmt	S_{wb}	operational write-back on analytical trigger
Smart vending / IoT	$S_{wb} + F$	decision read against in-flight writes
Remote service / predictive maint.	$S_{wb} + F$	sub-second decision window shorter than replication lag
<i>Healthcare & Life Sciences (H&LS)</i>		
Pharma track-and-trace	$S_{wb} + S_{authz}$	role-propagation lag during regulated write-back

Operational Analytics are on the critical path ~ every millisecond counts!

ata, authorization, and
ntation gap.

predicate evaluates
st that snapshot.

with the decision in the
it order.

rite constraint, a mid-TX
write-back.

decides, writes **X**,

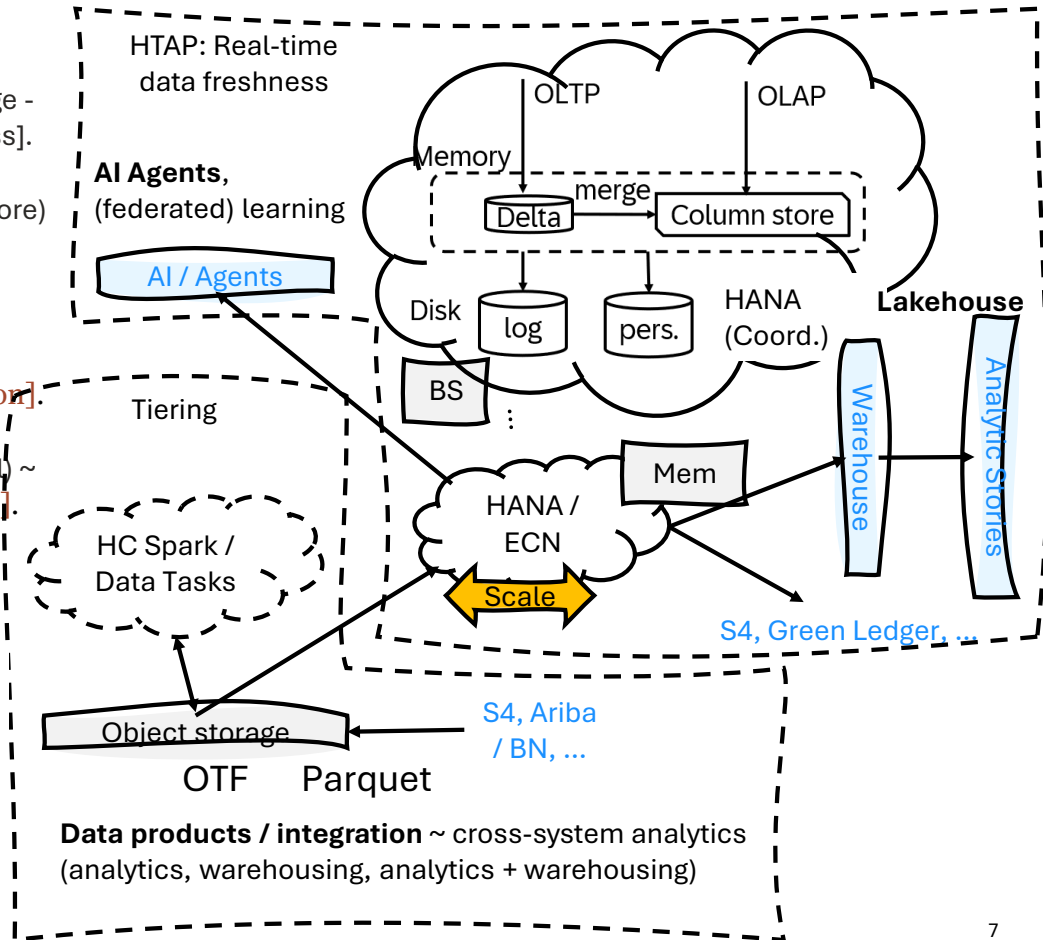
ncluding **X**, decides,

ncluding **Y**, ...

ent.

How does such a [System 1+2] for that Look Like?

- HANA In-memory columnar delta (write optimized) -> delta merge -> main (read-optimized); main property: [real-time data freshness].
- Multi-model (MDS, graph, document, ML, vector, Sparql/triple store)
- Resource disaggregation (compute / storage separation, object storage, PMem / CXL memory enabled) [scale] + [cost]
- Scale-out via table partitioning + distributed execution + [isolation].
- NSE paged column store (disk-resident columns with buffer pool) ~ data tiering: HTAP does not need to fit into RAM (cf. Umbra) [cost].
- S/4HANA Virtual Data Model ~ HTAP as workload pattern; application-database co-design.



The HANA Native Query Engine for Lakehouse Systems

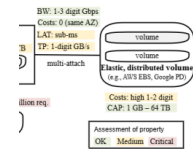
Daniel Ritter, Mihnea Andrei, Sukhyeon Cho, Maik Görgens, Taehyung Lee, Norman May, Amit Pathak, Paul R. Willems
SAP

SAP HANA Cloud: Data Management for Modern Enterprise Applications

Norman May
Alexander Böhm
Daniel Ritter
Frank Renkes
SAP SE
Walldorf, Germany

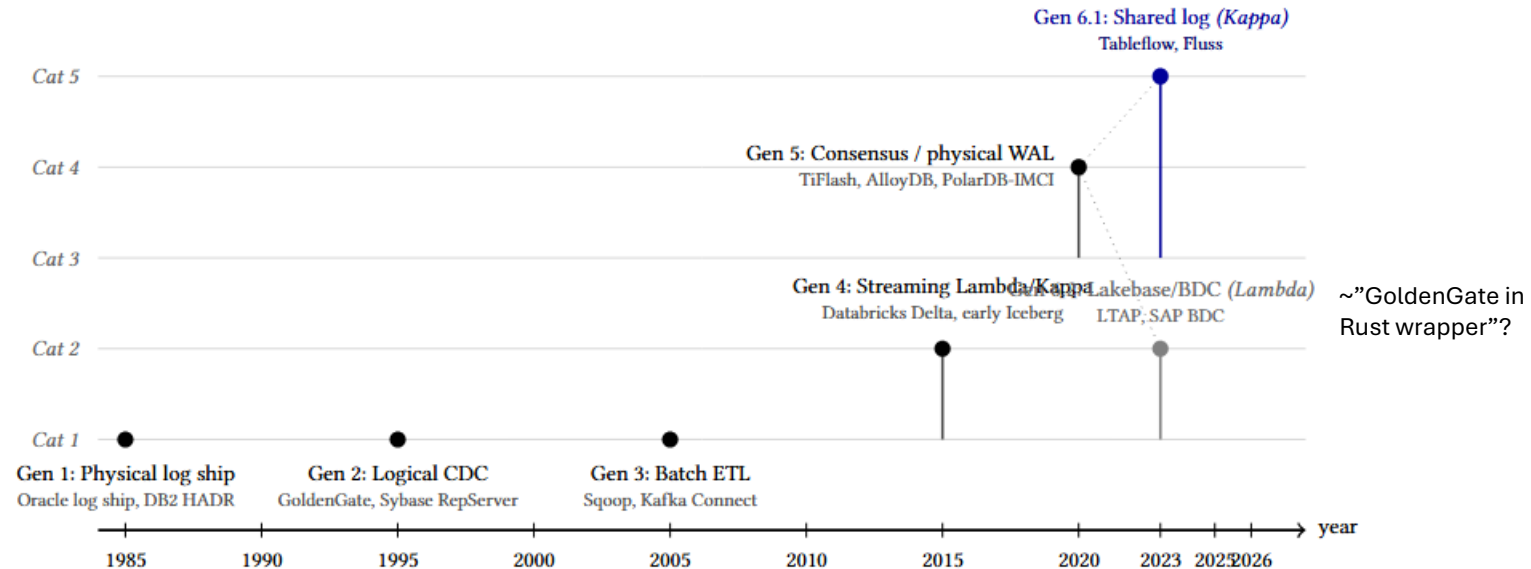
Mihnea Andrei
SAP Labs France
Paris, France
mihnea.andrei@sap.com

Wolfgang Lehner
Dresden University of Technology
Dresden, Germany
wolfgang.lehner@tu-dresden.de



nt cloud infrastructure in terms
about (CPU) and capacity (CAP)

[Gartner] + [Cloud] Position #1.2: 6th Generation of Replication Architectures same as 4th Gen? + still Has a Ceiling



“Gartner HTAP”

HTAP independent of the

- Economic storage: of all operational c
- Resource elasticity idle.
- Separation of stor: the primitive that

Idea: keep OLTP on the combining existing data

Properties:

- The lakehouse as a d (Iceberg, Delta, Hud
- **Eventual consistency** seconds-to-minutes
- **Isolation** / independence

Use case	Deficit	Nature
<i>Finance, Accounting & Risk (FA&R)</i>		
Days Sales Outstanding KPI	—	—
P&L statement calculation	—	—
CFO executive dashboard	S_{authz}	role lag on regulated records
Financial planning in cloud	—	—
Hurricane damage prediction	S_{ext}	external reference data outside OLTP WAL
<i>Sales, Marketing, Order Management & Customer (S&C)</i>		
Streaming sales monitoring	—	—
Multi-channel sales analytics	—	—
POS Explorer / retail KPI	—	—
Consumer Insight 365	S_{ext}	external reference data outside OLTP WAL
BlogIntelligence	S_{ext}	external reference data outside OLTP WAL
<i>Industry 4.0, Operations & Manufacturing (I4&M)</i>		
McLaren F1 telemetry	—	—
Enterprise simulation (DuPont)	—	—
CPG 200B-row ad-hoc analytics	—	—
Celonis process mining	S_{ext}	external reference data outside OLTP WAL
S/4HANA Business Fabric	—	—
<i>Healthcare & Life Sciences (H&LS)</i>		
MRI patient cohort analysis	S_{authz}	role lag on regulated records
Kaplan-Meier survival analysis	S_{authz}	role lag on regulated records
Virtual cancer patient simulation	—	—
Proteome cancer diagnostics	—	—
<i>Facilities, Real Estate & Public Infrastructure (F&PI)</i>		
Real estate building cockpit	S_{ext}	external reference data outside OLTP WAL

Non-operational analytics use cases are not about real-time freshness (few secs++): business reports, dashboards, RCAs, ML training, etc.



Isolation | Two Sides of / Faces on the Same Coin

Claim: Isolation in HTAP ~ two structurally opposite things ~ get one and by construction give up the other.

- Workload **non-interference**, **independent scaling** and **failure domains** (e.g., analytical crash, OOM), **independent storage format** — row-store for TP writes, column-store (Parquet/Iceberg). BUT **representation gap**!

Give up operational analytics properties:

- **Atomic write-back**: a decision made on engine A cannot commit back to engine P in the same TX; the write is a separate TX against P's commit order.
- **Read-your-own-writes** across **replication** boundary: a write to P is visible on A only after the replication window W; within W, the chain cannot see its own prior step.
- Cross-engine transactional composition: **no single BEGIN/COMMIT spans P and A**; 2PC et al. at prohibitive cost C.
- **Single-snapshot authorization**: authz lives on an identity store with its own clock, not on P's WAL; even with $W \rightarrow 0$ for data, the authz snapshot does not align.



Freshness Categorization | Replication Boundary

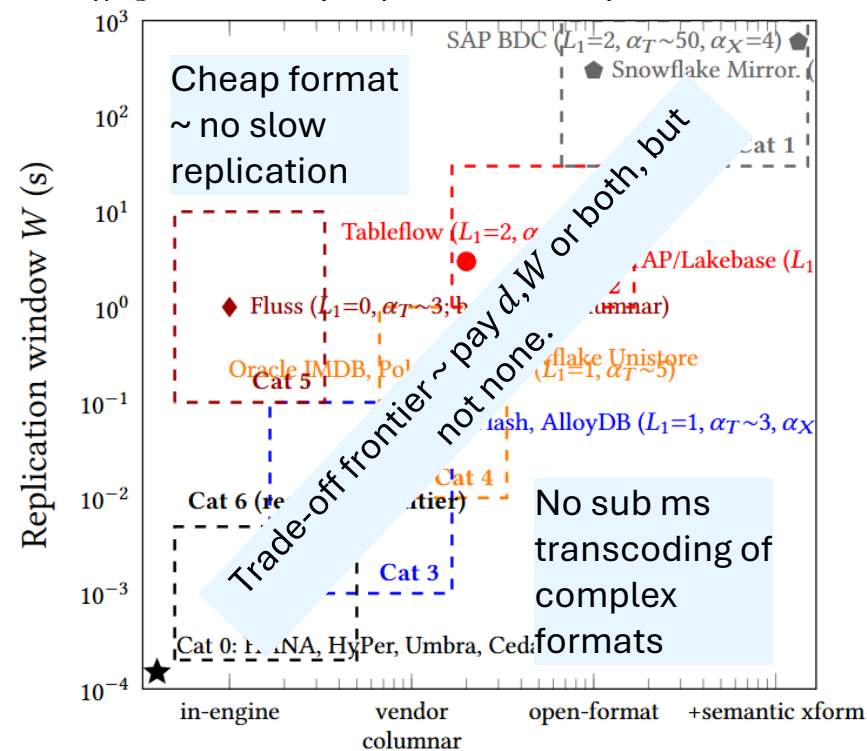
The (d, W) plane — a structural map of replication architectures:

- Every replicated architecture is a point in the plane. W = replication window (how stale). d = RUM-distance (how differently the data is shaped).
- $d \perp W$. Lowering one does not lower the other: two architectures can share the same freshness and still sit far apart in representation (e.g. Kafka log vs. Iceberg table; same W , different d).
- Cats 1–5 are ceilings on freshness F , not preferences. Each cat is a region in (d, W) , not a point.
- Cat 0 vs. the rest is categorical, not quantitative; one engine, no boundary, $(d, W) = (0, 0)$. Everything else has a boundary.

Crossing categories = redesign, not engineering:

- Cat 1 $\rightarrow$ Cat 2 (classic mirror $\rightarrow$ streaming open-format): swap the cost regime — batch ETL out, continuous CDC in.
- Cat 2 $\rightarrow$ Cat 5 (streaming transcoder $\rightarrow$ shared log): replatform storage; the log becomes the table (Kappa), rather than feeding one (Lambda).

Design space map of replication architectures outside cat 0 having a replication boundary: Looking into Freshness ($\sim$ replication window) and with RUM L1-distance

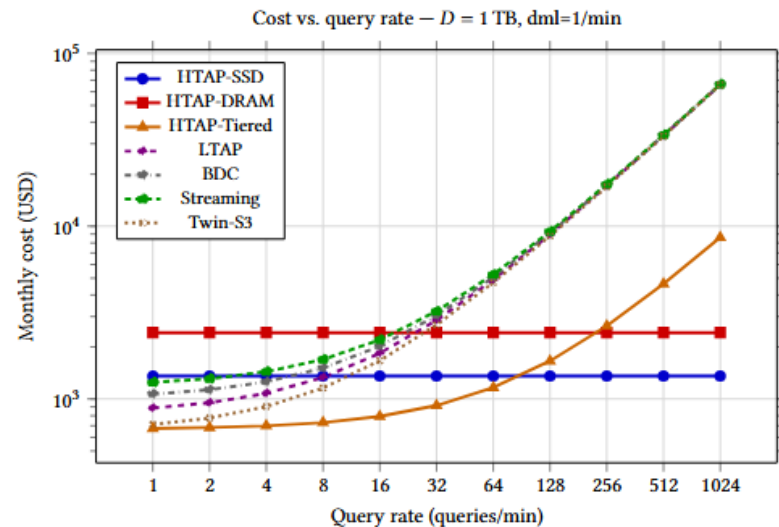


$$\text{RUM-distance } d = \|\rho_P - \rho_A\|$$

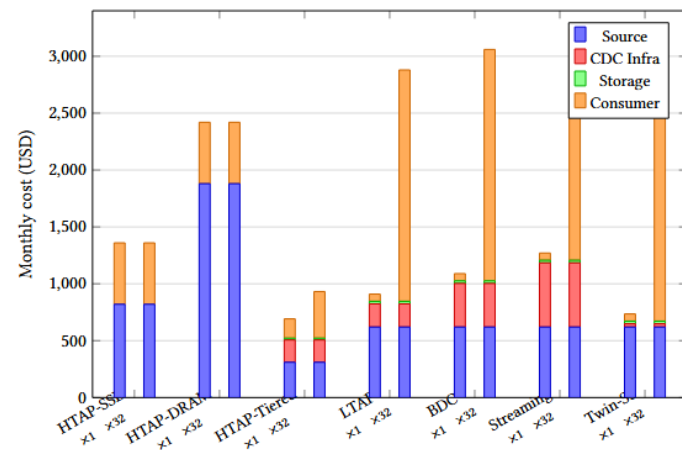
e.g., “Format gap” between primary row-oriented format and analytic format (e.g., Postgres WAL $\rightarrow$ Iceberg, Parquet)

Cost | S3 Strikes back

- HTAP / one system driven by always on compute (+ scaled up for consumption), with different cost floors depending on tiering {mem, ssd, s3}.
- Repl. with “initial load”, CDC overhead, main cost driver are variable costs during consumption.
- Always on components like (smaller) in-memory side-cars add to the CDC overhead, move closer to HTAP.

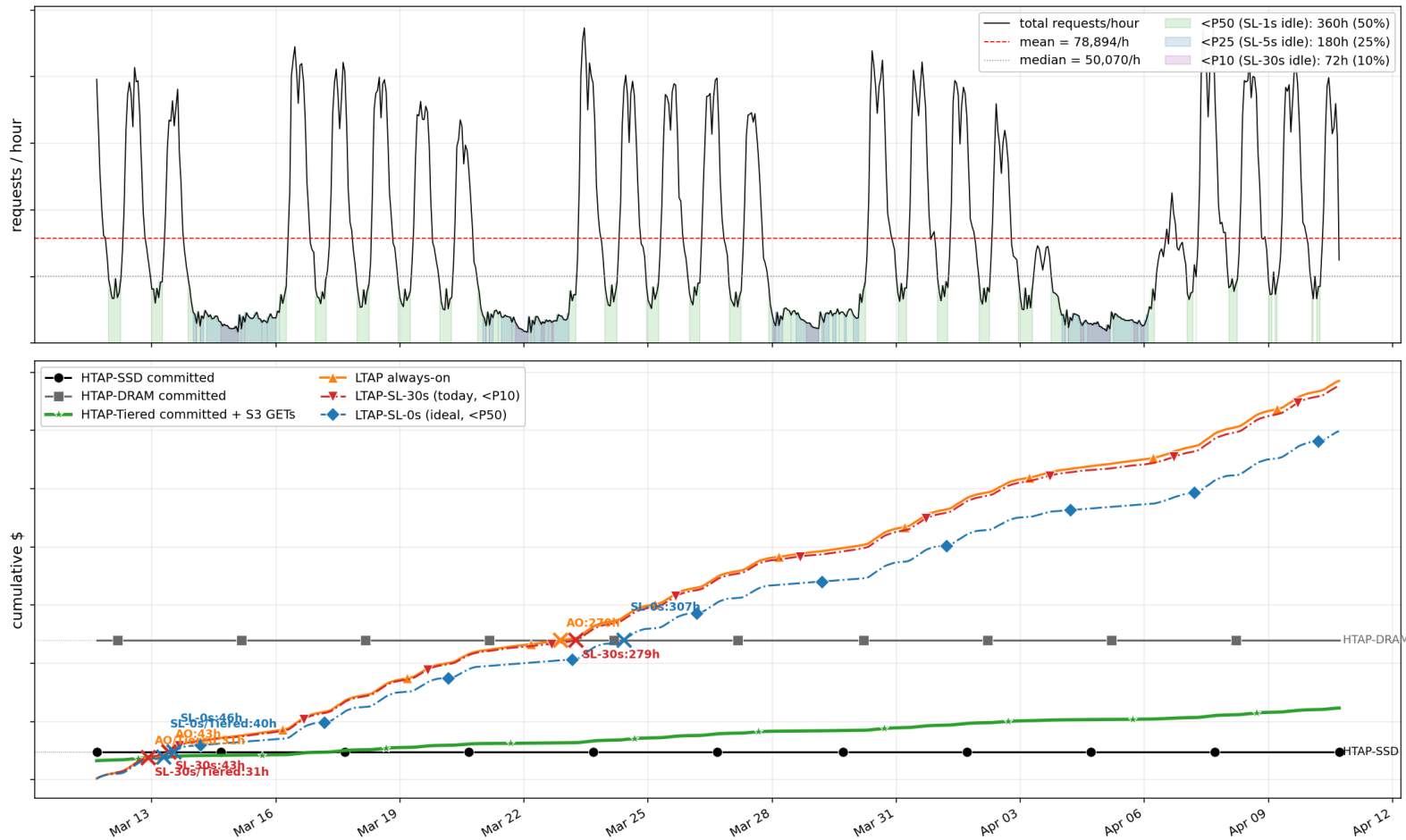


Cost composition at baseline (qps=1/min) and heavy read (qps=32/min) — $D = 1\text{ TB}$, $dml=1/min$

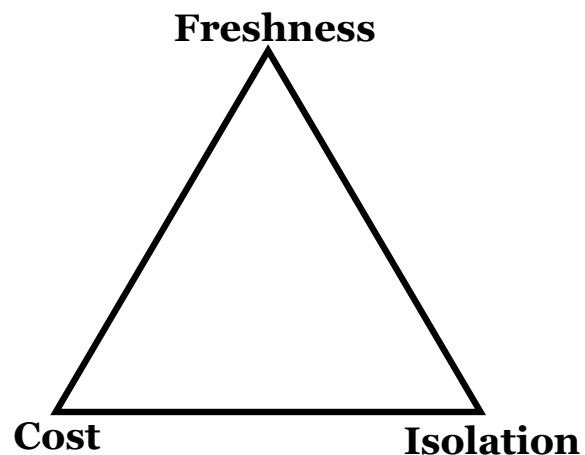


An Enterprise Seldomly Sleeps | SAP Lakehouse Workload 2026

Ina workload (top) and cumulative \$ with fixed-cost HTAP vs. accruing LTAP (bottom)



Can we have it all? | Low Costs, High Freshness, and Full Isolation (CFI)



Intuition:

- HTAP: isolation $\uparrow \sim$ cost $\uparrow$ (bigger instance for workload separation); freshness is high by construction
- Replication: freshness $\uparrow \sim$ cost $\uparrow$ (always-on middleware / sidecar); freshness $\downarrow \sim$ cost $\downarrow$ (batch, scale-to-zero).
- Universal: cost $\downarrow$ costs you either freshness or isolation.

Examples:

- Move data to object storage $\sim$ cost $\downarrow$, isolation $\uparrow$ (physical compute split), freshness $\downarrow$ (S3 latency).
- Collapse into one engine $\sim$ freshness $\uparrow$, wb $\uparrow$, but isolation $\downarrow$, and cost $\uparrow\downarrow$.

Position #1.3: No, “[Gartner] HTAP” in the Cloud Is Not Solved
+ Gaps will become more severe for new Use Cases.

50 Shades of Semantic Drift

Non-trivial but solvable:

- Technical drift (Collation / Unicode normalization, NULL handling, decimal precision & rounding, type-system, timestamps et al.
- What would Codd say about OTFs? multi-table ACID, FK/CHECK, DDL propagation, cross-table referential integrity (Iceberg/Delta carry columns, not constraints).
- ...

Non-monotonic / conceptional challenges:

- Non-monotonicity != latency. Authz grant → revoke → re-grant has no safe default; the replica must know the current value.
- WAL-bound: data values ship, view rules and role assignments don't. Structural ceiling, independent of lag.
- ...
- R/3 déjà vu: aggregates cost 80% of ERP capacity; cloud replication reconstructs the same tax.



Interoperable ACID Transactions for Open Table Formats

Tobias Götz
Technical University of Munich
Munich, Germany
goetz@in.tum.de

Muhammad El-Hindi
Technical University of Munich
Munich, Germany
muhammad.el-hindi@tum.de

Daniel Ritter
SAP
Walldorf, Germany
daniel.ritter@sap.com

Jana Giceva
Technical University of Munich
Munich, Germany
jana.giceva@in.tum.de

ABSTRACT



Entropy Everywhere | The hidden tax of replication

- Cross-system analytics requires **integration of heterogeneous sources** (e.g., microservice / hexagon architectures).
- Pipelines / sidecars explode (e.g., multiple tenants, sources, vendors, consumer-scaling; legacy)
- 1:1 table replication **!=** data product; much harder
- Schema **evolution**, customer **extensions** are important and always fun

Entropy is paid per pipeline x per schema change x per transformation step ~ closing the circle to the origins of HTAP.

[HTAP 2.0 in the Cloud Analogy] ~ [System 3] Enterprises are not solely Read-Only /Analytical

Claim: There won't be read-only agents only ~ prepare for “insights to actions”.

Two parallel directions:

- 1) Make HTAP work for the agentic loop.
- 2) Not another replication layer ~ a compositional write-back contract for semantically correct reverse-ETL-like patterns.

Requires primitives for (1) AI as actor, (2) AI write-back primitives and composition / replication gaps become more severe for AI.

Thank you.

Contact information:

Daniel Ritter
daniel.ritter@sap.com



Special Issue on
Lakehouses
contributions until
Feb'27.

