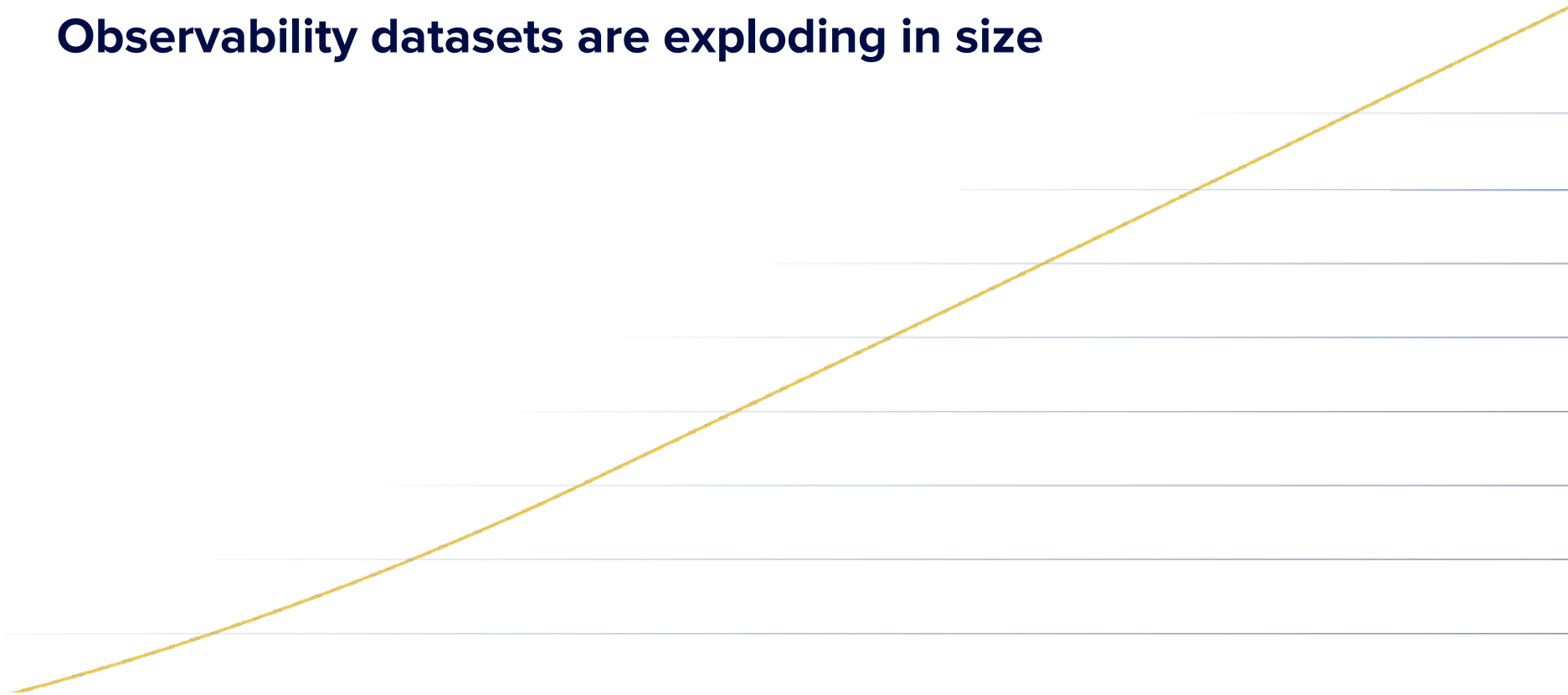


Observability on ClickHouse®

What Breaks at Scale and How to Model Around It

27 May 2026

Observability datasets are exploding in size



Observability in 2026

The observability tradeoff...



ALTINITY®

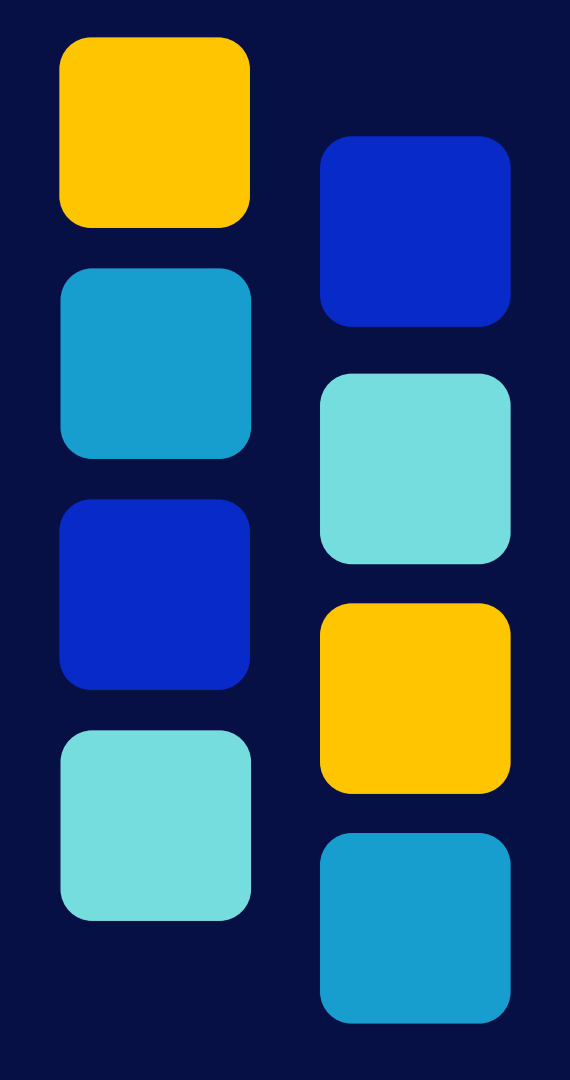
Run Open Source ClickHouse® Better

Altinity.Cloud Enterprise Support

Altinity® is a Registered Trademark of Altinity, Inc.
ClickHouse® is a registered trademark of ClickHouse, Inc.;
Altinity is not affiliated with or associated with ClickHouse, Inc.

Last9

AI-native observability for the whole team
— developers, SREs, and AI agents.



Agenda

- Architecting Observability (in 2026)
- ClickHouse is awesome
- How Last9 does ClickHouse at scale
- Why BYOC?
- Demos 🙄
- Tales from production

Architecting an Observability Platform

The Glass Triangle of Observability

- Cost
- Performance
- User-Experience

What is Observability

- ➔ Instrumentation
- ➔ Ingestion
- ➔ Storage
- ➔ Query
- ➔ Alerting
- ➔ Agents

OpenTelemetry Changed the Game

The New Model

*The responsibility for telemetry quality and volume is now shared.
Volume continues to grow non-linearly.*

ClickHouse Solves the Data Problem

How ClickHouse supports observability

- Control over data storage
- Single unified source
- Extreme volume and cardinality

Clickhouse + OpenTelemetry

- Out of the box exporter
- Default schema for logs metrics traces
- Engines for logs metrics traces
- Grafana data source for Clickhouse
- Read/Write isolation

```
clickhouse/logs:  
  endpoint: ${logs_endpoint}  
  ttl_days: 14  
  logs_table_name: ${logs_table_name}  
  timeout: 50s  
  database: ${logs_database_name}  
  username: ${logs_writer_username}  
  password: "${logs_writer_password}"  
  retry_on_failure:  
    enabled: true  
    initial_interval: 5s  
    max_interval: 30s  
    max_elapsed_time: 300s
```

How Last9 Balances the Needs

On the ground challenges

- Lack of Structured Data
- Data Silos
- Service specific needs
- Alert thresholds
- Too much data, far less insights
- Legacy code

What do customers need?

What we have today?

- Multiple dashboards
- Scattered telemetry data
- Configuration sprawl
- Alert fatigue
- Constant Instrumentation changes

What we need?

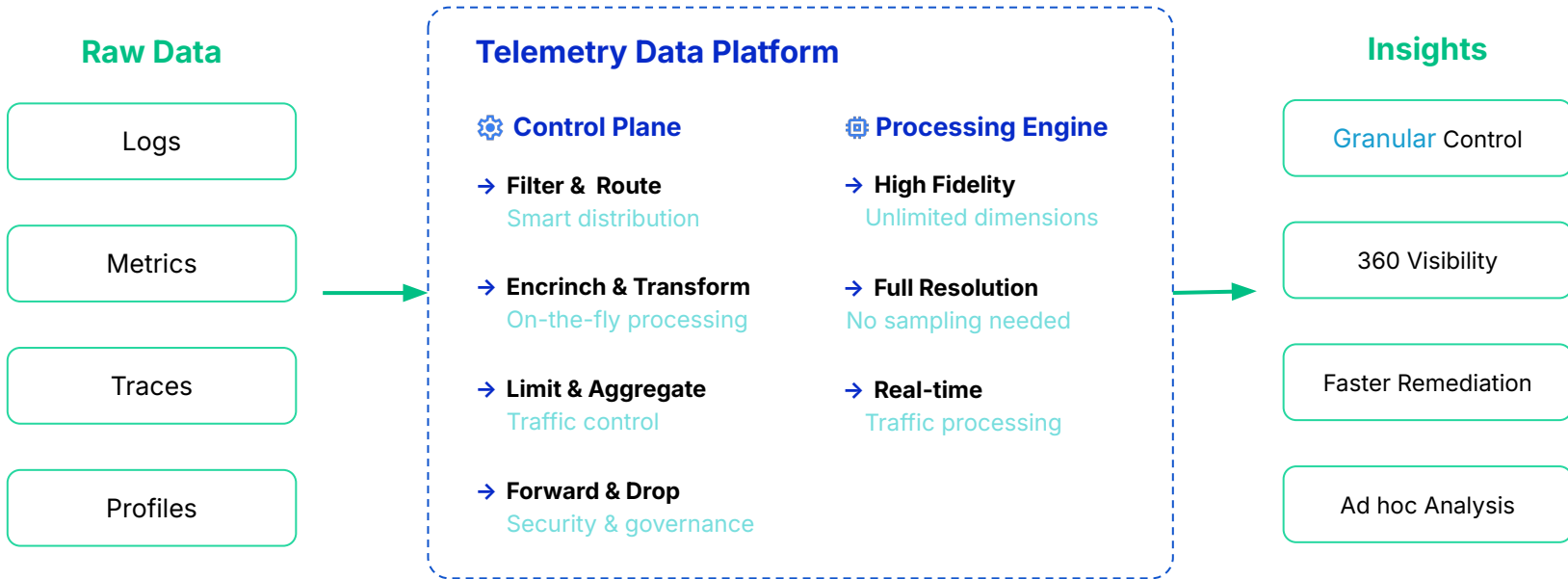
- Single source of truth
- Unified telemetry
- Centralized control
- Signal vs Noise

Multitenancy tradeoffs

Managing Multitenancy

- Cost vs QoS
- Proxy API layer
- More control over SQL
- Source multiple backends

Last9: Telemetry Data Platform



1M+
samples/second

100ms
Query Latency

20M+
Dimensions

360°
Visibility

Flexibility of Access

- ➔ Native UI
- ➔ LogQL
- ➔ PromQL
- ➔ TraceQL
- ➔ SQL
- ➔ Agents / MCP

Clickhouse At Scale

- Merges increase with massive spikes
- Materialized views resulting into ingestion becomes bottleneck
- Too batch size -> clickhouse ingestion problem
- Adding Buffer -> you have 2 problems now
- Vertical scaling -> slow, rightly so. Need more predictability
- Traffic variations from 100k to 300Mn per minute

Why Managed BYOC?

Why Managed BYOC?

- No vendor lock-in
- Data never leaves your environment
- Reliability of a managed service
- Low, predictable costs

**ClickHouse can be your entire
lakehouse, not just your observability
database**

Demo of Last9

Demo of Altinity BYOC

[slide intentionally left blank]

Demo of Altinity BYOC

Your ClickHouse just works

Tales from production

The 5 mb span attribute

Logs from 2096

OTel Payloads > 100mb

PII data everywhere

O'rely? Do i really have this much data?

10 MB of logs crashing browser

Book a demo of Last9



Thank you!
Questions?

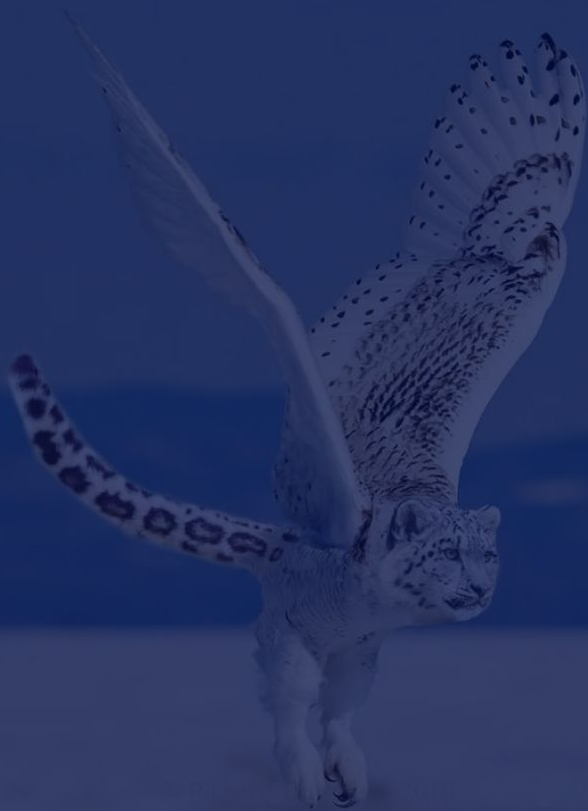
<https://last9.io>

<https://altinity.com/slack>



Resources

- OTel / Parquet / Iceberg POC Demo:
<https://github.com/Altinity/demo-otel-parquet-antalya>
- O11y in One talk:
https://www.youtube.com/watch?v=uDY_EsE47sA
- OTel Vendor-Neutrality talk:
<https://www.youtube.com/watch?v=A5L6F8wcVnI>
- Project Antalya Roadmap:
<https://github.com/Altinity/ClickHouse/issues/1359>
- All Altinity Demos:
<https://altinity.com/demos/>



Introducing Hybrid Tables: Transparent Query on ClickHouse® MergeTree and Iceberg Data



An LLM is only as good as the context it can reach

Three ways we see these tools (actually) helping

1. Natural-language querying*
2. Code-generation for analytics applications
3. Diagnostic and troubleshooting assistance

The new model

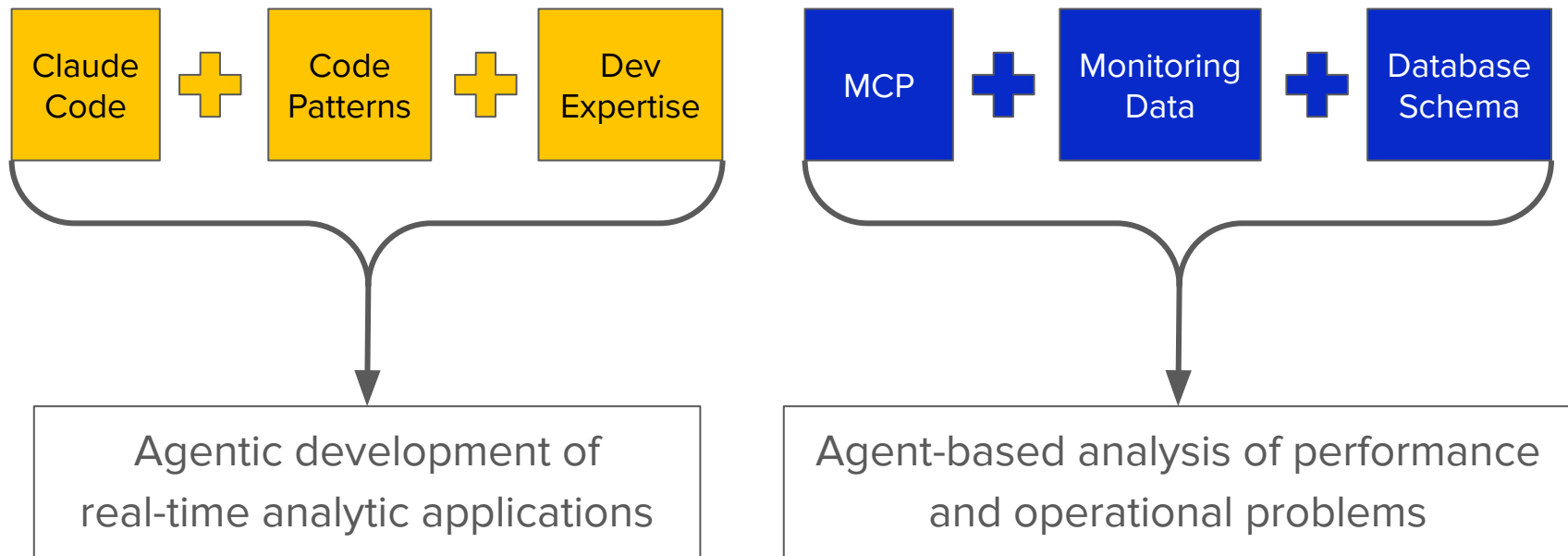
Agents ask many unknown questions.

**They generate queries, call tools, inspect results, and
*repeat quickly***



How do we get meaningful answers out of data-intensive applications and systems?

Where can we see AI actually working?



Requirement 1:

Fresh data has to be queryable now

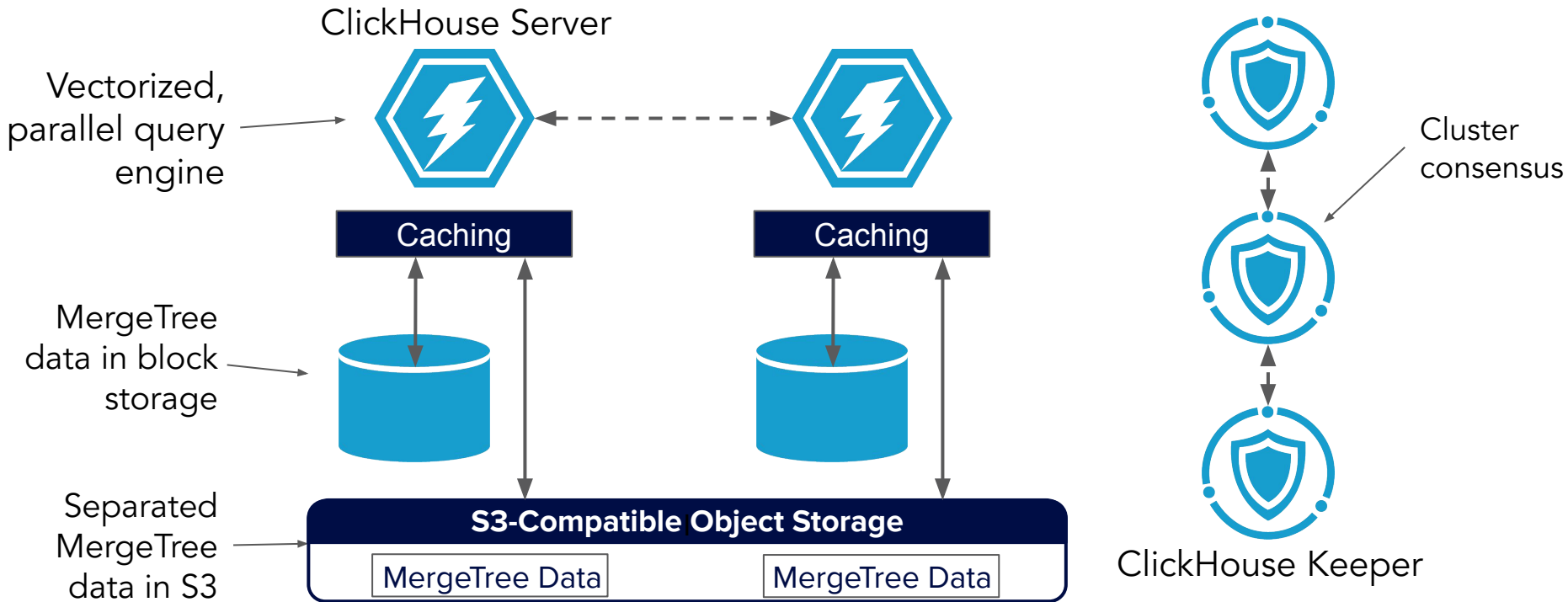
A black stealth bomber, likely a B-2 Spirit, is shown in flight from a high-angle perspective. The aircraft is dark and sleek, with its wings spread. It is flying over a rugged, mountainous landscape with snow-capped peaks and deep valleys. The sky is a pale blue. A semi-transparent white banner is overlaid across the middle of the image, containing the text 'Introducing ClickHouse®'.

Introducing ClickHouse®

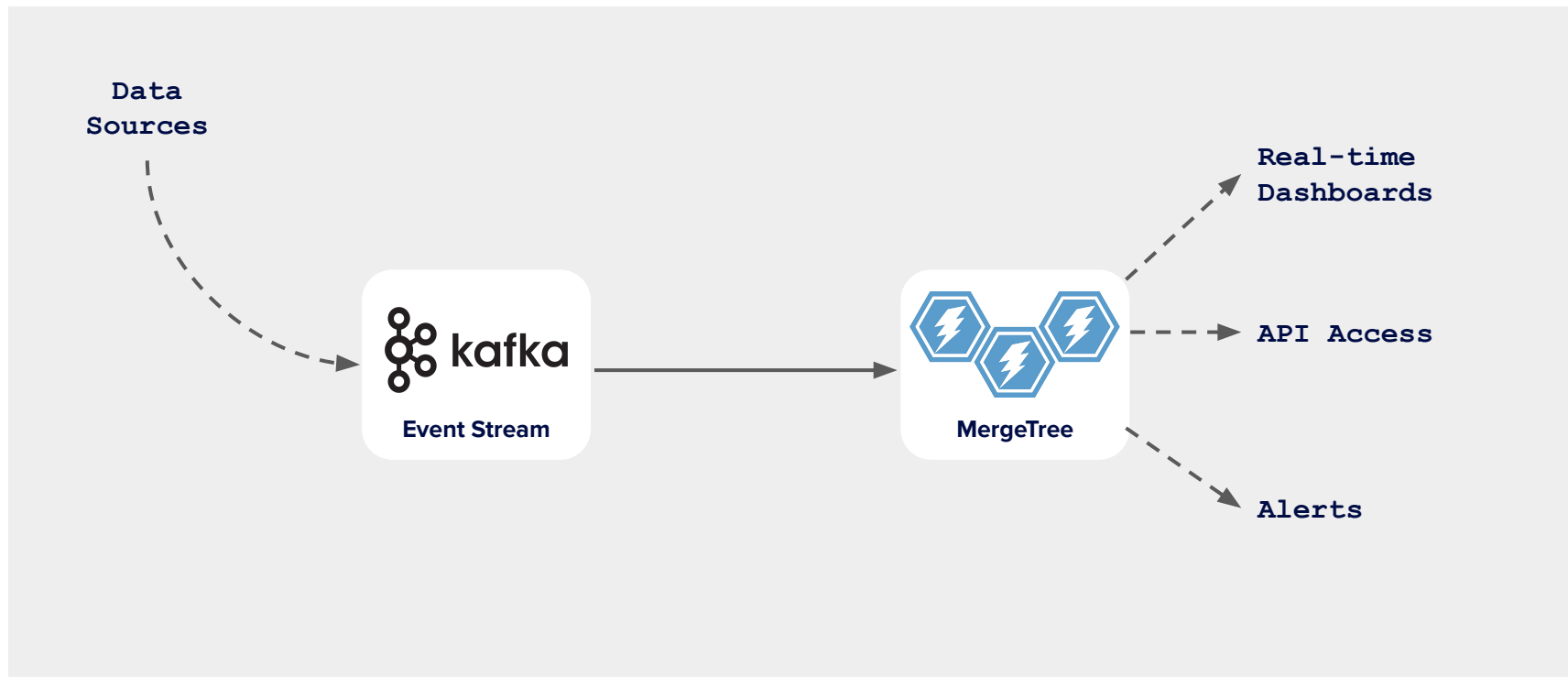
What is ClickHouse?

- Apache 2.0
- Single binary
- SQL Compatible
- Columnar
- *Really really fast*
- Laptop to exabyte scale
- Eats cardinality for breakfast

ClickHouse shared nothing architecture



Traditional real-time event pipeline with ClickHouse



ClickHouse works great for almost any real-time analytics

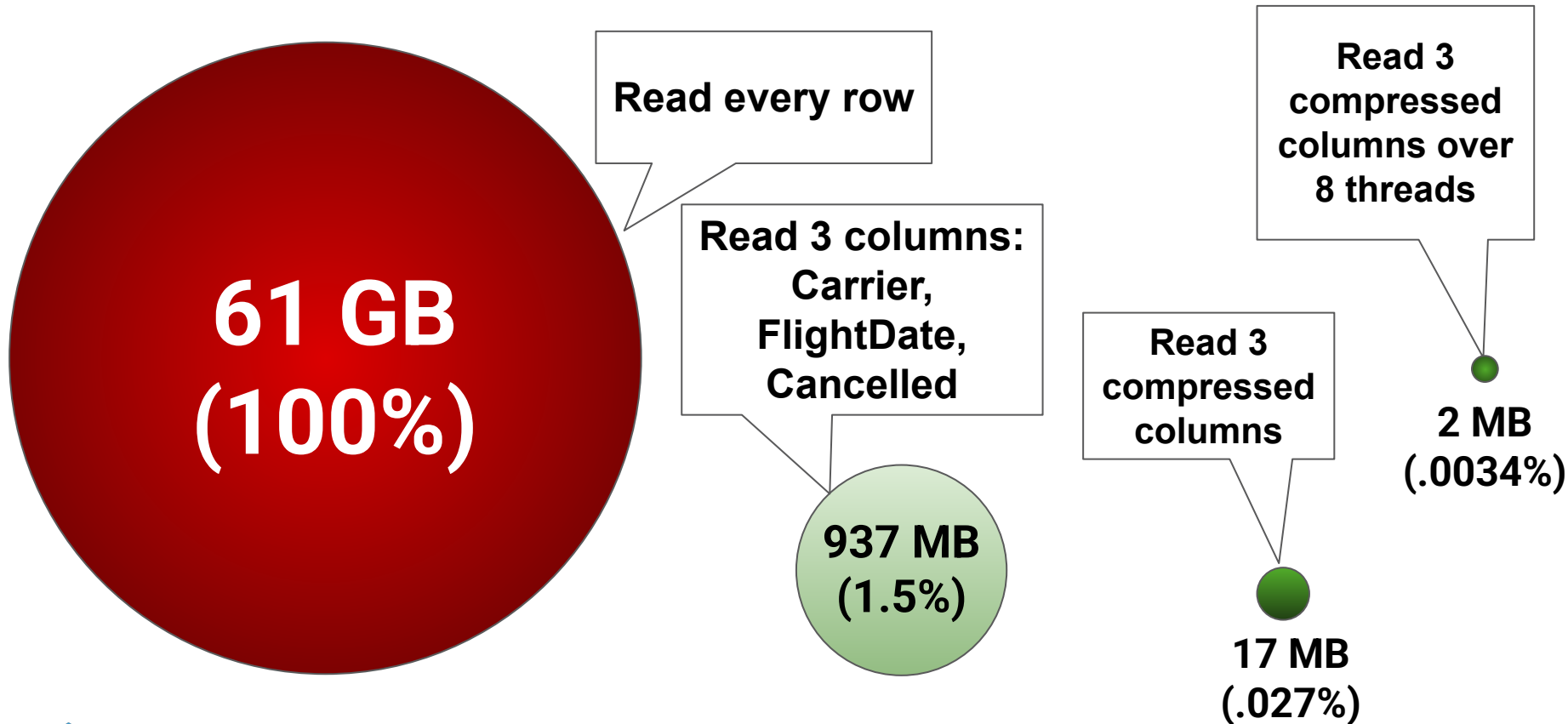
```
SELECT Carrier, toYear(FlightDate) AS Year,  
       (sum(Cancelled) / count(*)) * 100. AS cancelled_pct  
FROM default.ontime_ref  
GROUP BY Carrier, Year HAVING cancelled_pct > 1.  
ORDER BY cancelled_pct DESC LIMIT 10
```

	Carrier	Year	cancelled_pct
1.	G4	2020	16.733186040434276
2.	EA	1989	10.321500966388536
3.	WN	2020	9.284307653599388

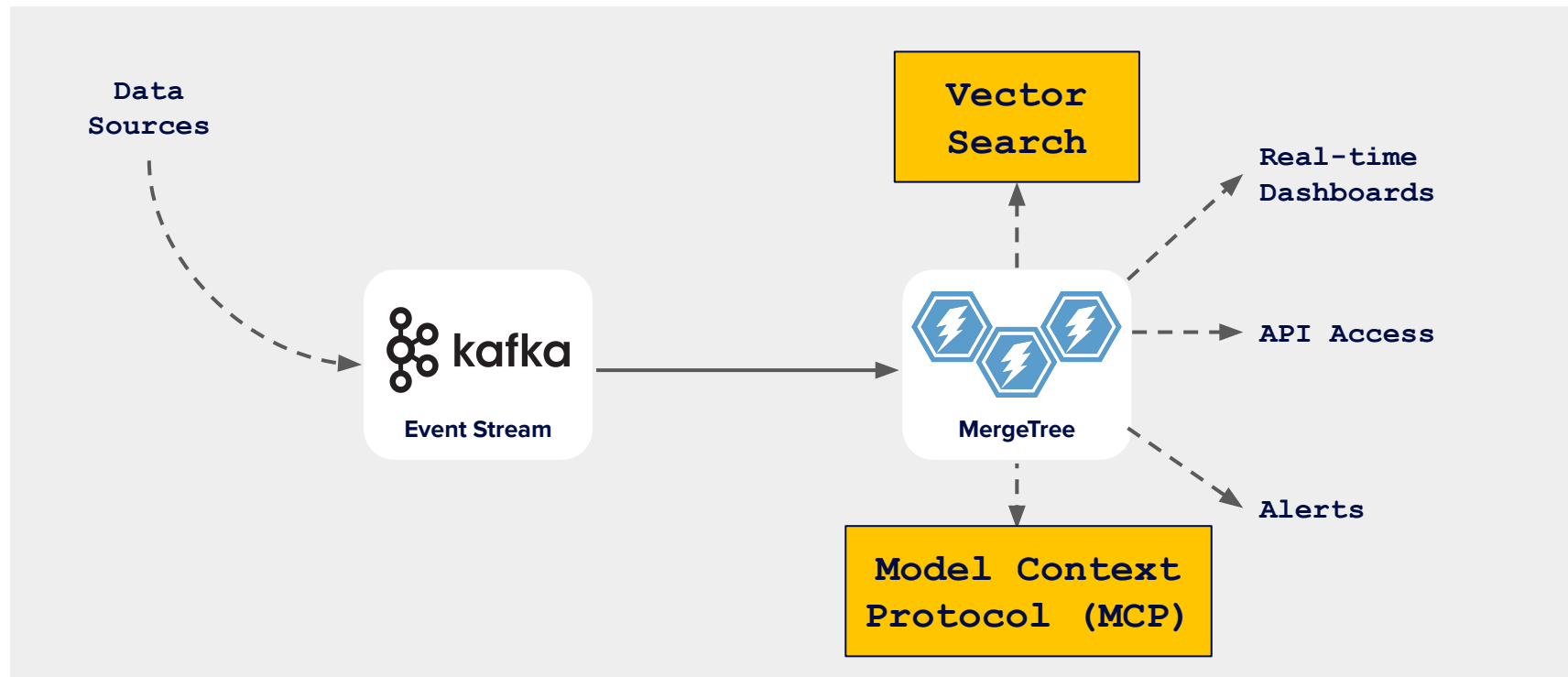
. . .

10 rows in set. Elapsed: **0.674 sec.** Processed **196.51 million rows**, 982.57 MB (**291.68 million rows/s.**, 1.46 GB/s.)10 rows

Visualizing effect of columns and compression on I/O

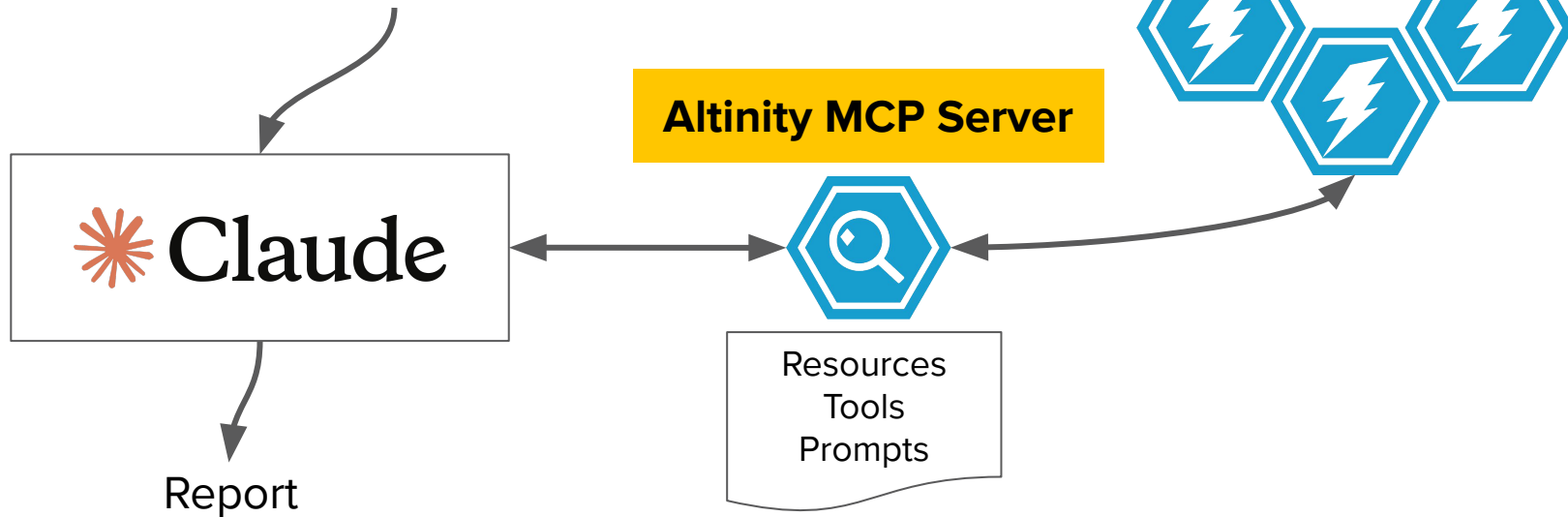


ClickHouse is adding capabilities for AI



New workloads, new costs of operation

Show me the minute-by-minute
value of exchange- traded
derivatives on the VIX “fear index”



Project Antalya builds a foundation for AI on ClickHouse

Cheap Storage

Hybrid tables on Iceberg
lower cost without
sacrificing latency

Scalable Compute

Stateless swarms
located anywhere
boost speed

Single Sign-on

OAuth/OIDC allow
safe access from
AI tools

Open Source – Runs in Any Environment



ALTINITY®

Run Open Source ClickHouse® Better

Altinity.Cloud Enterprise Support

Altinity® is a Registered Trademark of Altinity, Inc.
ClickHouse® is a registered trademark of ClickHouse, Inc.;
Altinity is not affiliated with or associated with ClickHouse, Inc.

Try Altinity Antalya Now

Project Antalya Demos - <https://altinity.com/demos><https://docs.altinity.com>

Altinity ClickHouse Repo - <https://github.com/Altinity/ClickHouse>

Project Antalya Builds - <https://builds.altinity.cloud>

Altinity Slack - <https://altinity.com/slack>

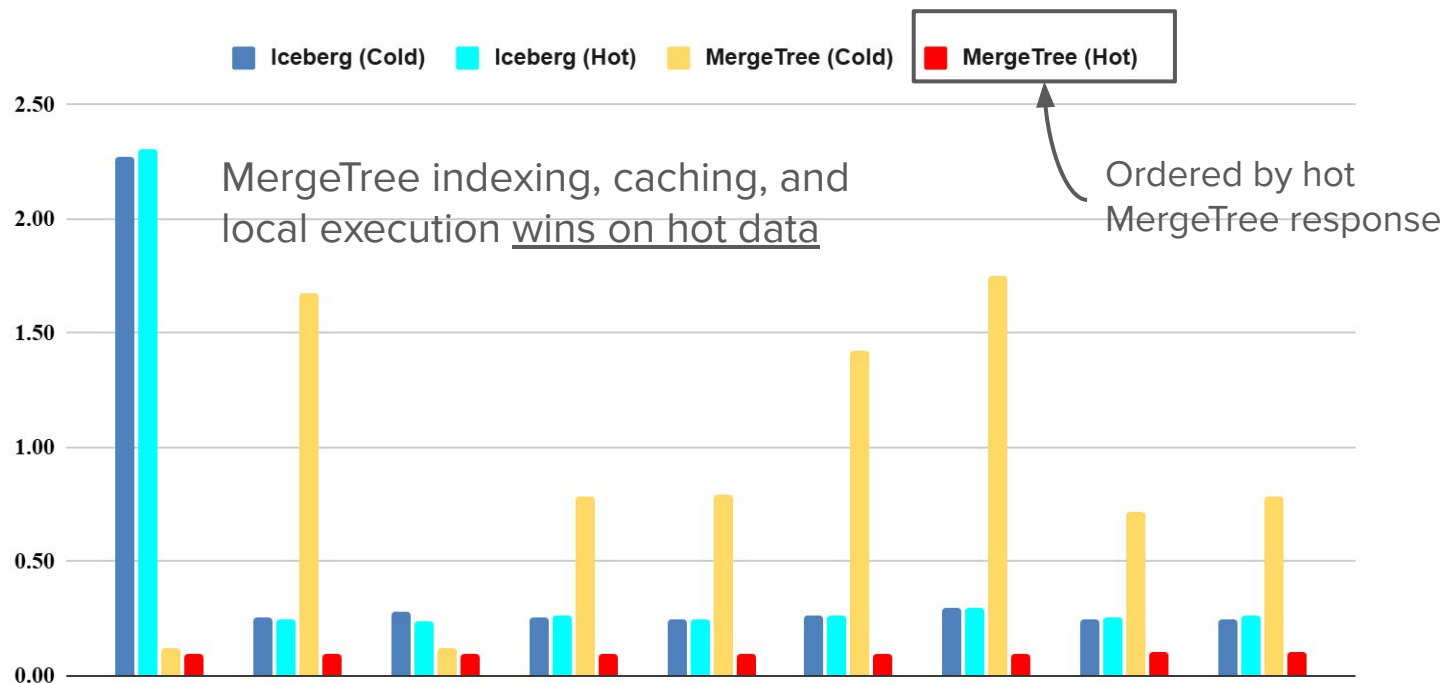
We're learning what are the foundations of AI

- ClickHouse and Iceberg have complementary strengths
- Making the combination transparent is the hard problem
 - Data types, timestamps, partitioning, ordering
- Movable compute leads to magical effects on performance
- Put every capability in SQL
- Automation is crucial
- OAuth is the way. We are putting it into open source ClickHouse.

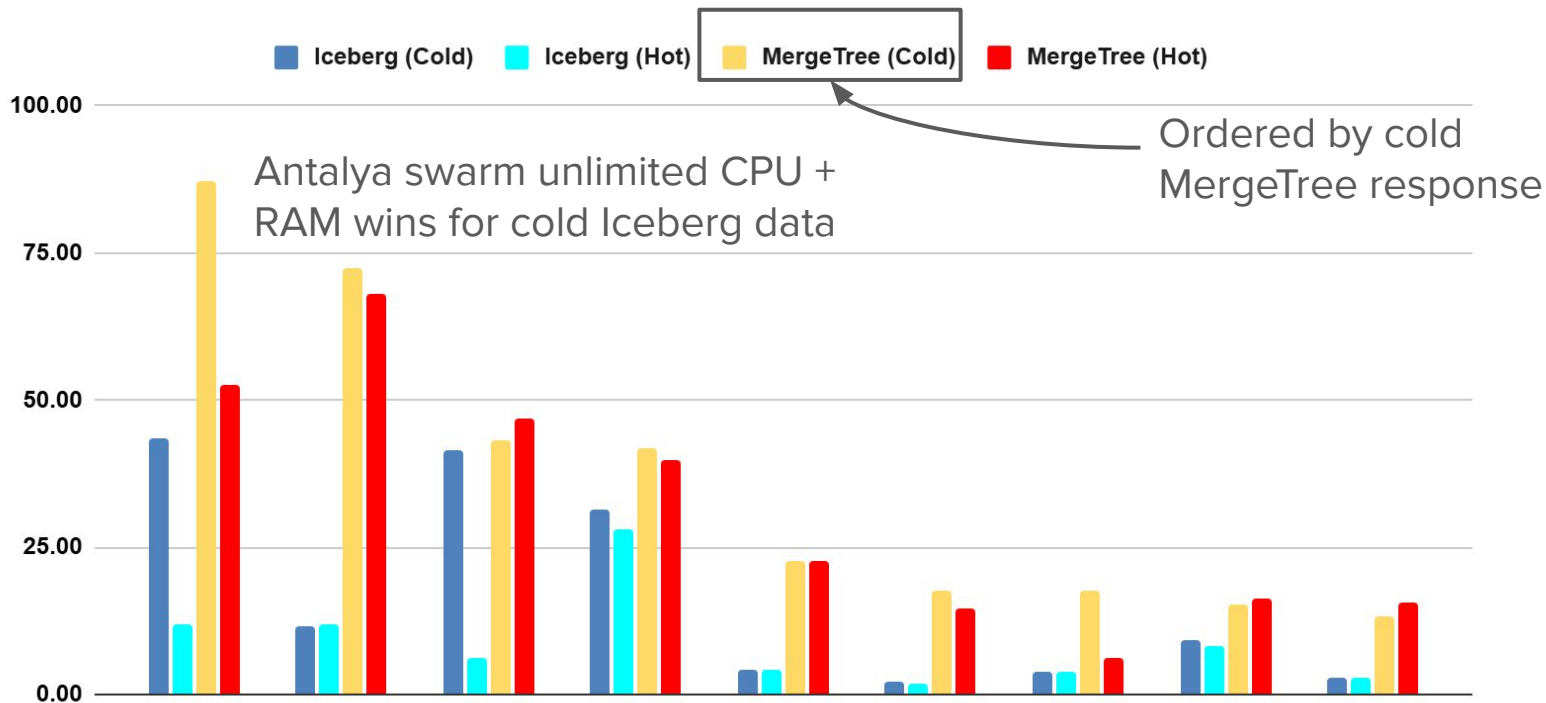
Let's compare Iceberg vs native MergeTree performance

- Simulated customer data
 - 40 most common queries
 - Compare MergeTree vs. Iceberg
 - Measure best response hot & cold
- 1 Initiator with MergeTree data
 - c8g.4xlarge (16 vCPU, 32 GiB RAM),
 - Encrypted EBS GP3 500Mb/s
 - 4 swarm nodes
 - c8g.4xlarge (16 vCPU, 32 GiB RAM)
 - Encrypted EBS GP3 500MiB/s
 - Spot instances enabled

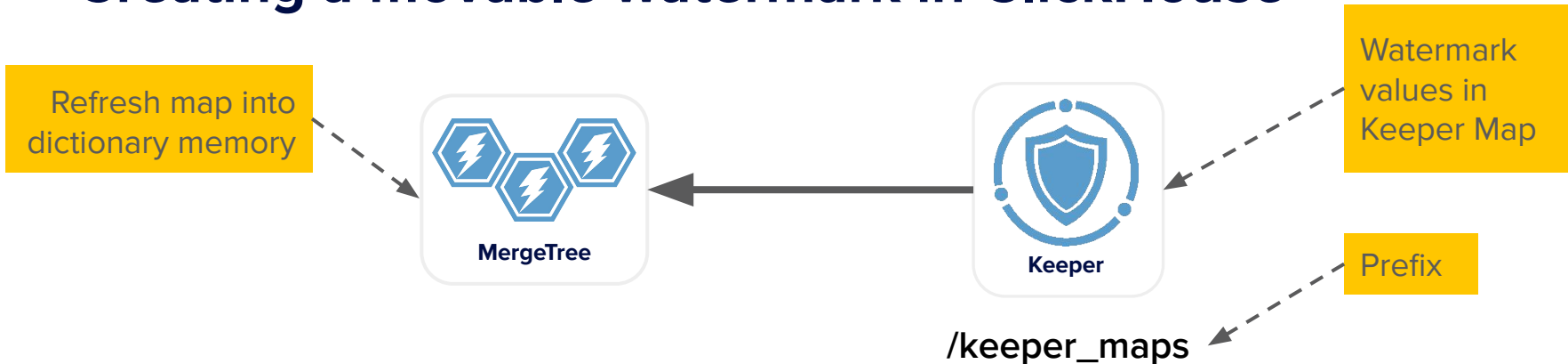
Performance: Iceberg vs MergeTree (HOT DATA)



Performance: Iceberg vs MergeTree (COLD DATA)



Creating a movable watermark in ClickHouse



```
CREATE DICTIONARY
system.hybrid_watermarks_dict (
    table String,
    watermark Date
)
PRIMARY KEY (table)
...
```

```
CREATE TABLE
system.hybrid_watermarks (
    table String,
    watermark Date
) Engine =
KeeperMap('/hybrid_watermarks')
PRIMARY KEY (table)
```

Say hello to the Hybrid table engine

```
CREATE TABLE btc.transactions AS mt.transactions
ENGINE = Hybrid(
  cluster('{cluster}', mt.transactions),
  date >= today() - 7,
  ice.`mt.transactions`,
  date < today() - 7
)
```

Use schema
from MergeTree

Segment

Condition

We added commands to dump parts easily to Parquet / S3

```
ALTER TABLE mt.transactions_table  
EXPORT PART '20210722 1 1 0'  
TO TABLE s3.staging_mt_transactions  
SETTINGS allow_experimental_export_merge_tree_part = 1;
```

```
ALTER TABLE mt.transactions_table  
EXPORT PARTITION ID '2021'  
TO TABLE s3.staging_mt_transactions  
SETTINGS allow_experimental_export_merge_tree_part = 1;
```

<https://altinity.com/blog/project-antalya-progress-alter-table-export-and-other-cool-features>

And we added Altinity Ice to load Parquet files into Iceberg

```
ice insert mt.transactions -p --thread-count=12 \  
  --s3-copy-object \  
  --partition=' [{"column": "pickup_date",  
"transform": "month"}] ' \  
  "s3://mybucket/LOAD/transactions_staging/*.parquet"
```

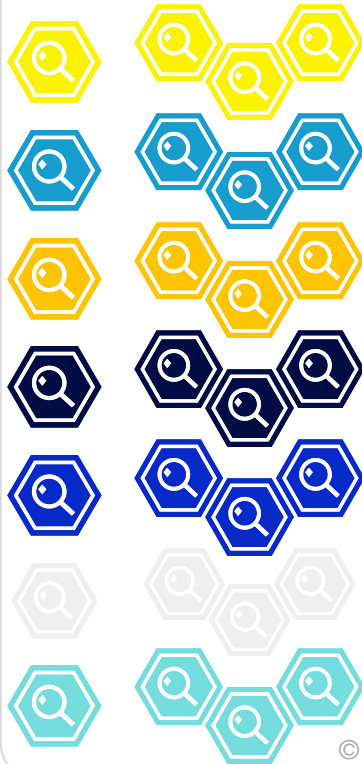
<https://altinity.com/blog/moving-data-in-bulk-from-clickhouse-to-iceberg-using-altinity-ice>

Icons- Transparent

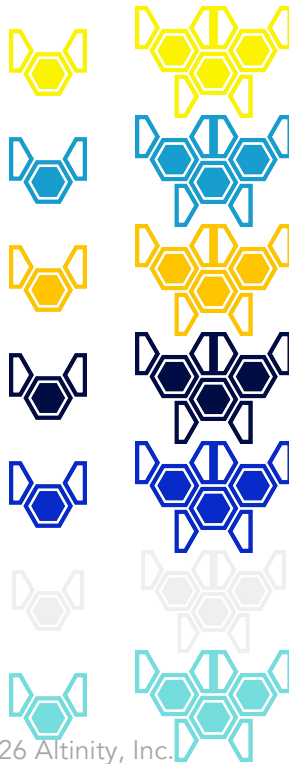
Clickhouse (Native) Cluster



Director Cluster



Swarm Cluster



Keeper



Other

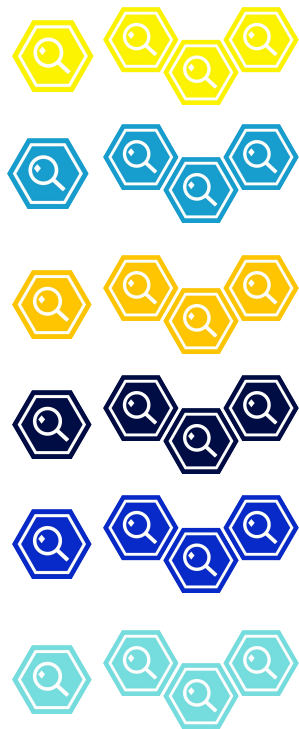


Icons-Stackable on White Backgrounds

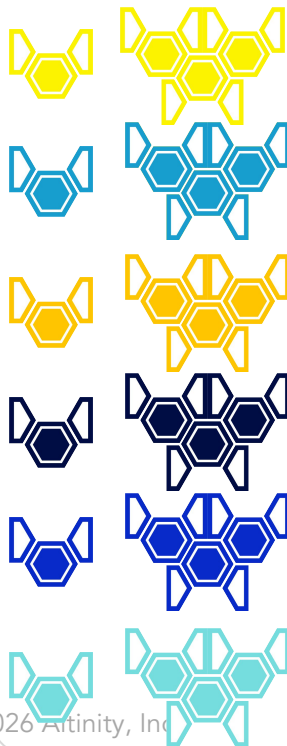
Clickhouse (Native)
Cluster



Director Cluster



Swarm Cluster



Keeper



Other



Icons-Stackable on Dark Backgrounds

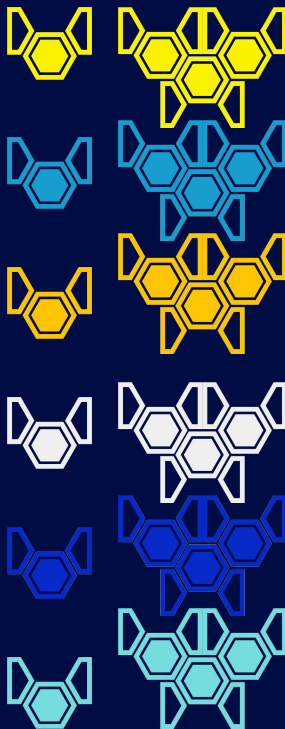
Clickhouse (Native) Cluster



Director Cluster



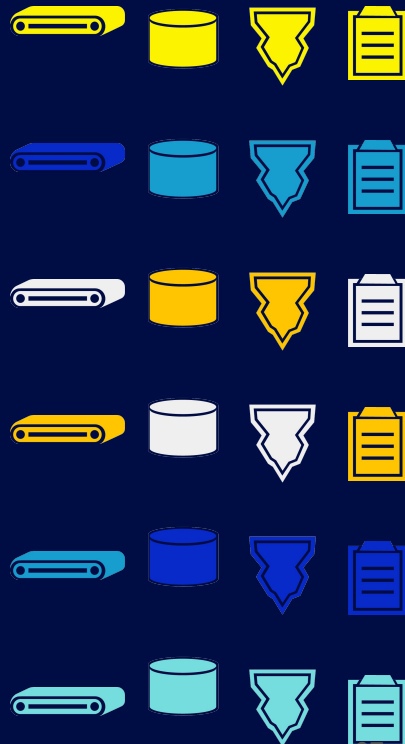
Swarm Cluster



Keeper



Other





Goals for Project Antalya

Build the next generation architecture for real-time analytics and AI

- Reduce storage costs
- Deliver separation of compute and storage
- Single copy of data, universally accessible
- Preserve ClickHouse compatibility
- Run anywhere
- Keep system building blocks 100% open source