

THE FUTURE OF OBSERVABILITY IS OPEN

**OBSERVABILITY IS BROKEN AND OPEN SOURCE
WILL BE THE SOLUTION**



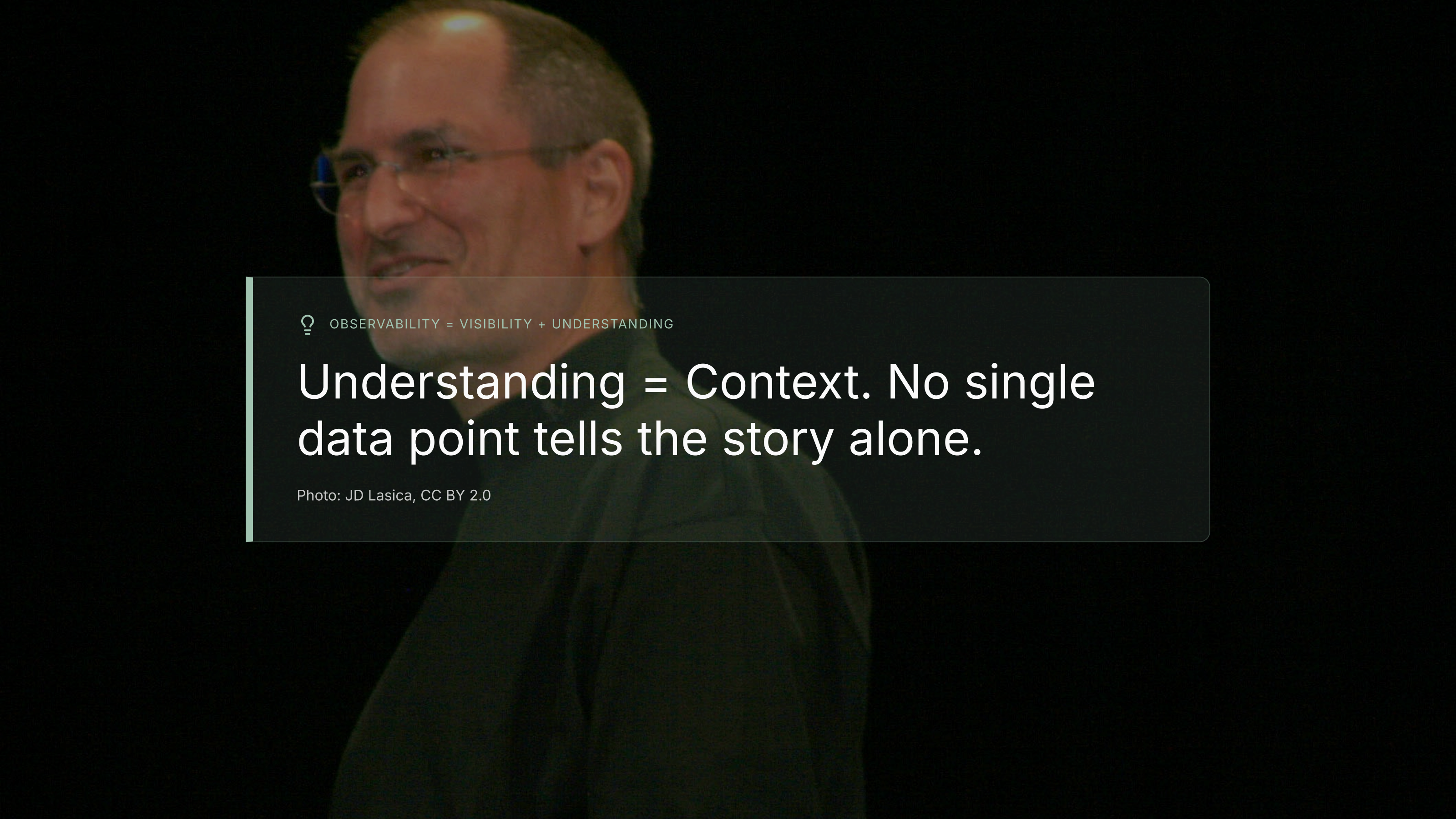
I'M JOSH LEE

- **Open Source Advocate @ Altinity**
- ClickHouse®, OpenTelemetry, & Kubernetes
- Previously Product Manager @ IBM Instana

Opinions are my own

ClickHouse® is a registered trademark of ClickHouse, Inc. Altinity is not affiliated with or associated with ClickHouse, Inc. We are but humble open source contributors

OBSERVABILITY = VISIBILITY + UNDERSTANDING



💡 OBSERVABILITY = VISIBILITY + UNDERSTANDING

Understanding = Context. No single
data point tells the story alone.

Photo: JD Lasica, CC BY 2.0

THAT MEANS LOTS OF DATA...

50_x

more data for observability *

* rule of thumb

OBSERVABILITY IS BROKEN

Why is it broken?

- 1 DON'T HAVE THE DATA**
We need universal collection from every runtime, tool, and environment
- 2 CAN'T GET THE DATA**
We need it in compatible formats that work with the tools we use
- 3 CAN'T AFFORD THE DATA**
We need to spend time fixing issues, not worrying about custom metric costs
- 4 CAN'T FIT THE DATA**
Even with open source, data volumes are too darn high

1. DON'T HAVE THE DATA

Observability requires ubiquitous universal instrumentation

UBIQUITOUS UNIVERSALLY COMPATIBLE INSTRUMENTATION

NOT TOO BAD...

METRICS

Prometheus exposition format is a defacto universal standard

LOGS

It's just text, and Linux/Docker/Kubernetes have standards

MONOLITHS

Individual runtimes may have good tooling

BUT WHAT ABOUT...

POLYGLOT MICROSERVICES

Many runtimes and languages lead to inconsistent metadata at best

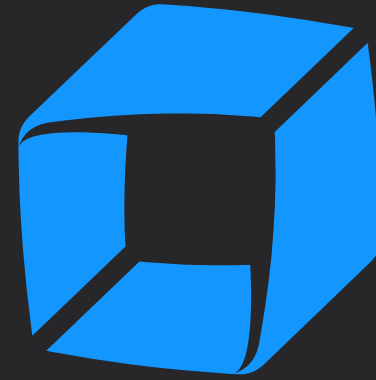
DISTRIBUTED TRACING

Now the runtimes need to exchange context to emit telemetry

LEGACY SOFTWARE

Just another way of saying "software that makes money"... but is difficult to change

Enter Observability Vendors:



2. CAN'T GET THE DATA

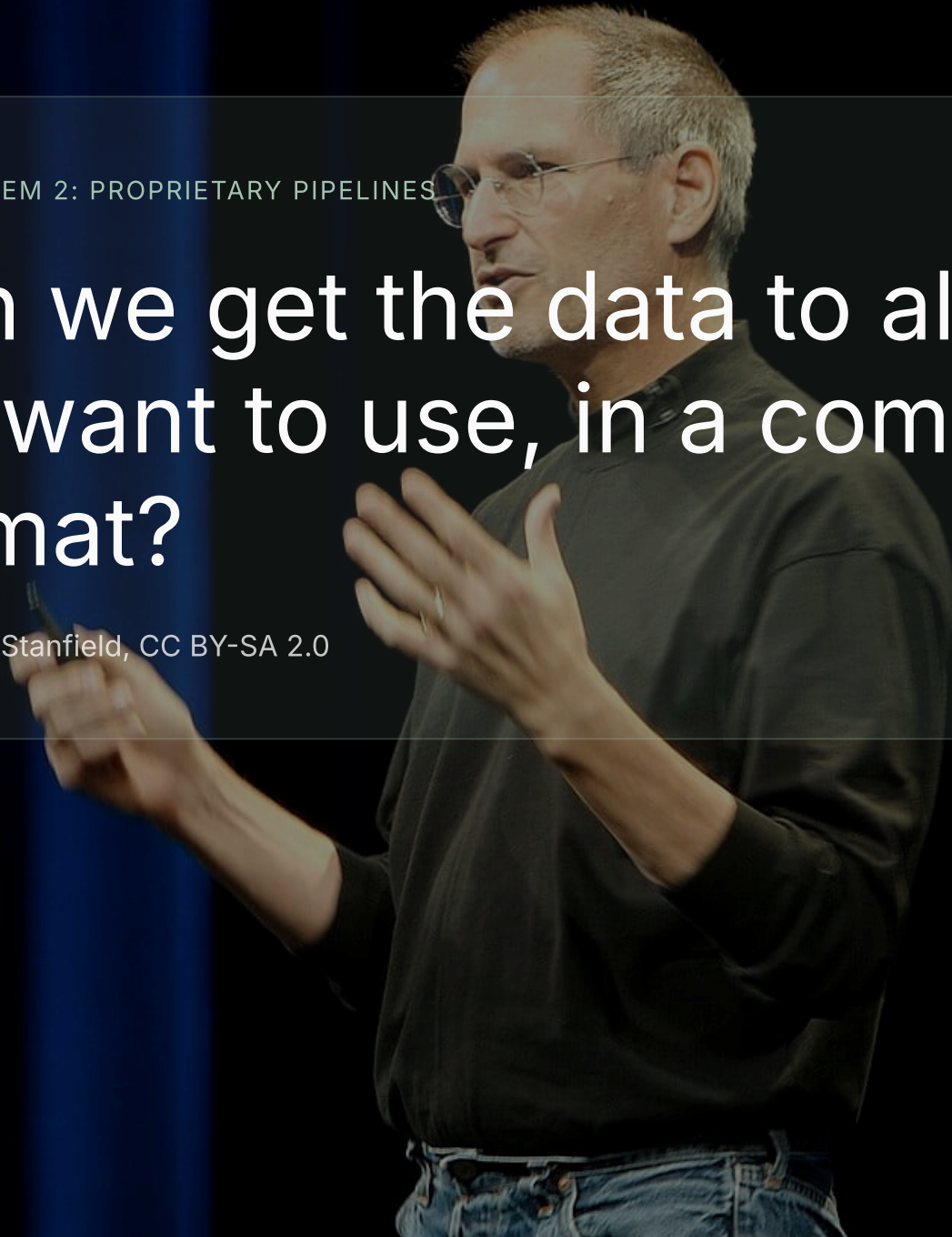
Observability requires interoperable formats and pluggable pipelines



PROBLEM 2: PROPRIETARY PIPELINES

Can we get the data to all of the tools we want to use, in a compatible format?

Photo: Ben Stanfield, CC BY-SA 2.0



3. CAN'T AFFORD THE DATA

The SaaS business model breaks at scale

PROBLEM 3: SAAS IS TOO EXPENSIVE

Vendor markups are unsustainable at scale



Per event
Pay per event ingested



Per GB
Pay per GB ingested



Hidden costs
Egress, storage, retention add up



Tokens?
Will we pay a markup on tokens too?

4. CAN'T FIT THE DATA

Observability data volumes are growing non-linearly

PROBLEM 4: TOO MUCH DATA

1

MORE SOFTWARE

per org

2

MORE COMPONENTS

per app

3

MORE TELEMETRY

per component

4

AI?!!!

...

Photo: Jean-François Gornet, CC BY-SA 2.0

THE FUTURE OF OBSERVABILITY IS OPEN

I ❤️ OPENTELEMETRY

THE REAL BENEFITS OF OPENTELEMETRY'S "VENDOR-NEUTRALITY"

- 1 **UNIVERSAL INSTRUMENTATION**
Instrumentation can be baked in to tools, libraries, and runtimes
-

- 2 **PORTABLE FORMATS**
OTLP allows for flexible pipelines with multiple sinks
-

- 3 **INTEROPERABLE TOOLS**
Use all of the tools together, all the time

UNCOVERED: THE HARD TRUTH ABOUT OTEL'S VENDOR- NEUTRALITY

Scan to watch the talk on YouTube



youtube.com/watch?v=A5L6F8wcVnI

THE PROGRESSION OF INTEROPERABILITY



1

OPEN INSTRUMENTATION

Collaboration on universal instrumentation

2

OPEN PIPELINES

Get the data to any tool (or multiple tools)

3

OPEN STORAGE

Control over infrastructure destiny

4

OPEN FORMATS

Use a single copy of data with multiple tools

OBSERVABILITY IS (PARTLY) BROKEN

OBSERVABILITY IS (PARTLY) BROKEN

- OPENTELEMETRY SOLVES UNIVERSAL COLLECTION

- OPENTELEMETRY SOLVES INTEROPERABLE PIPELINES

- SEMANTIC CONVENTIONS GIVE US A STARTING POINT FOR INTEROPERABLE TOOLS

OBSERVABILITY IS (PARTLY) BROKEN



Vendor lock-in
Data is still locked up with vendors



Cost
Compute, storage, and network transfer are still expensive

DATA IS STILL LOCKED UP WITH VENDORS

DATA IS STILL LOCKED UP WITH VENDORS

CAN'T ACCESS DATA

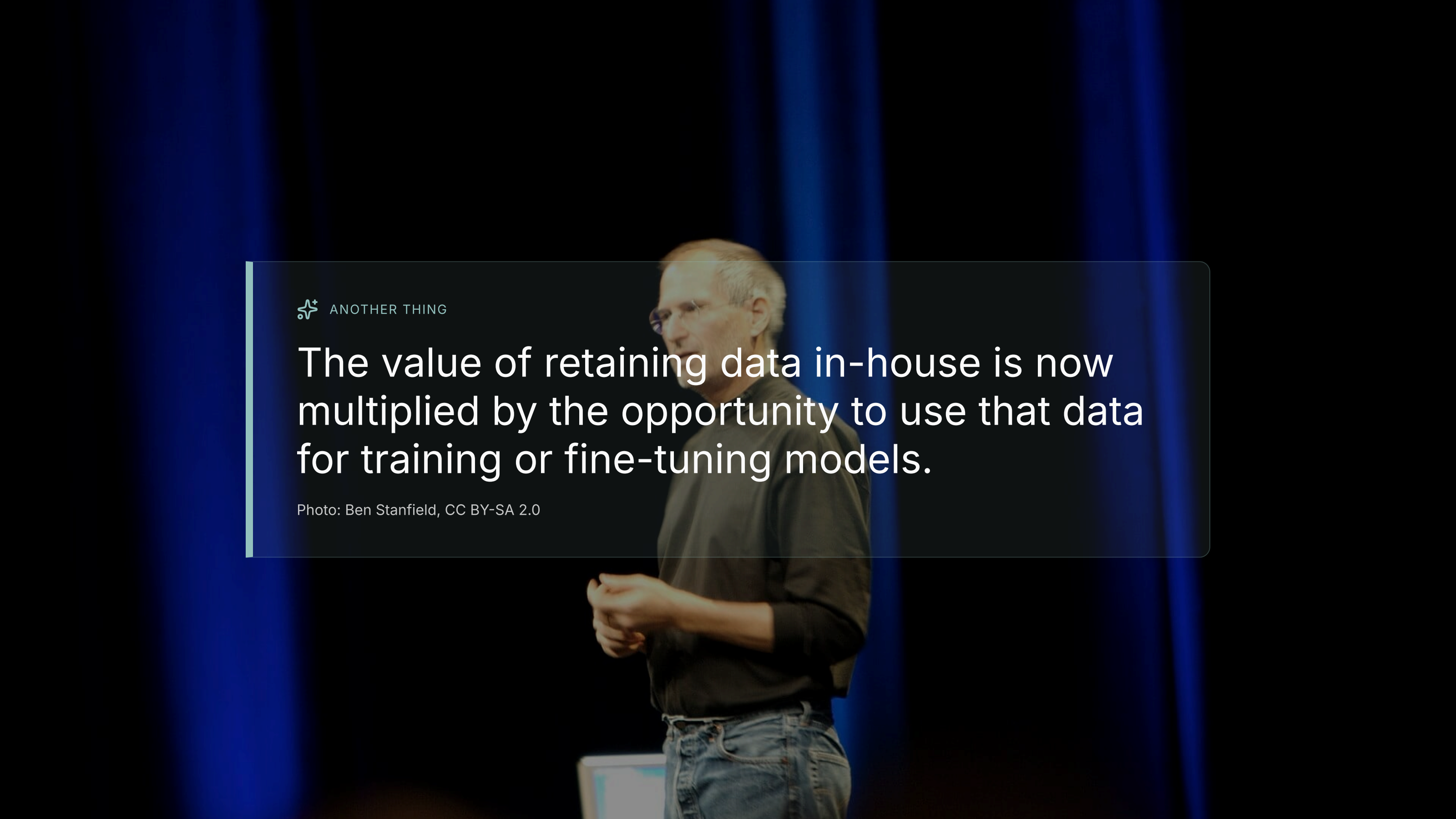
Worst case - no external API

CAN'T EXPORT THE DATA

"It would take us a week to export a week's worth of telemetry..."

DON'T CONTROL THE DATA

Where it rests, who has access, what formats and technologies are used



✦ ANOTHER THING

The value of retaining data in-house is now multiplied by the opportunity to use that data for training or fine-tuning models.

Photo: Ben Stanfield, CC BY-SA 2.0

OPEN SOURCE IS THE SOLUTION (AGAIN)

OPEN SOURCE STORAGE

METRICS

Prometheus, VictoriaMetrics,
Mimir, Thanos, ClickHouse

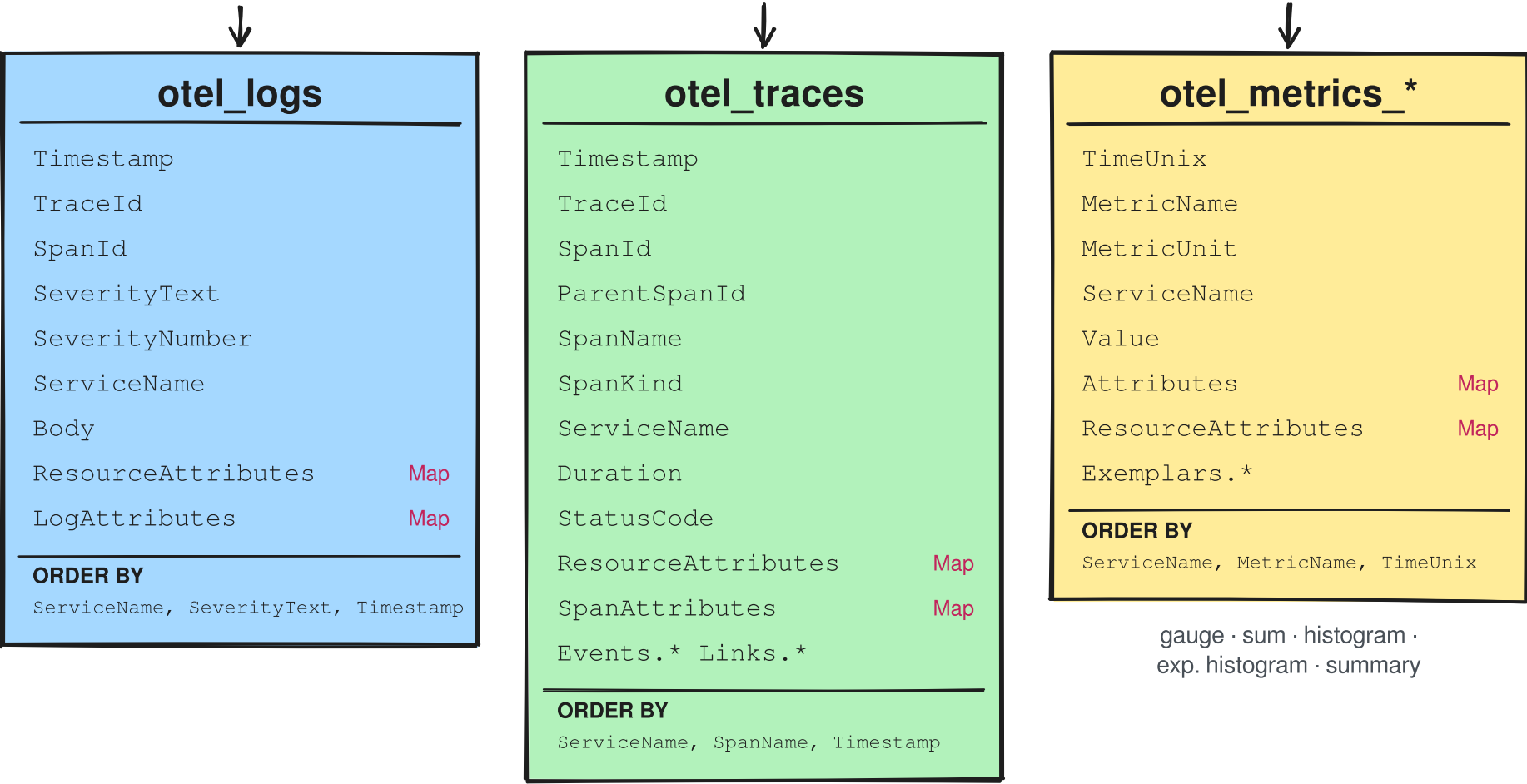
LOGS

Loki, Open/ElasticSearch,
VictoriaLogs, ClickHouse

TRACING

Jaeger, VictoriaTraces,
ClickHouse

OTel Collector → clickhouseexporter → ClickHouse



A SQL SCHEMA FOR OPENTELEMETRY

The ClickHouse exporter's default tables

CLICKHOUSE - ONE DATABASE FOR EVERYTHING

A red Swiss Army knife is shown with several tools extended, including a large blade, a saw, and pliers. The knife is positioned diagonally across the frame, with the handle at the top right and the tools fanning out towards the bottom left. The background is a dark, textured surface.

1

ONE SQL INTERFACE FOR ALL SIGNALS

And it's really really fast

2

BUILT FOR SCALE

Worry about things other than cardinality

3

FULL-TEXT SEARCH

GA since March 2026

O11Y IN ONE: CLICKHOUSE AS A UNIFIED TELEMETRY DATABASE

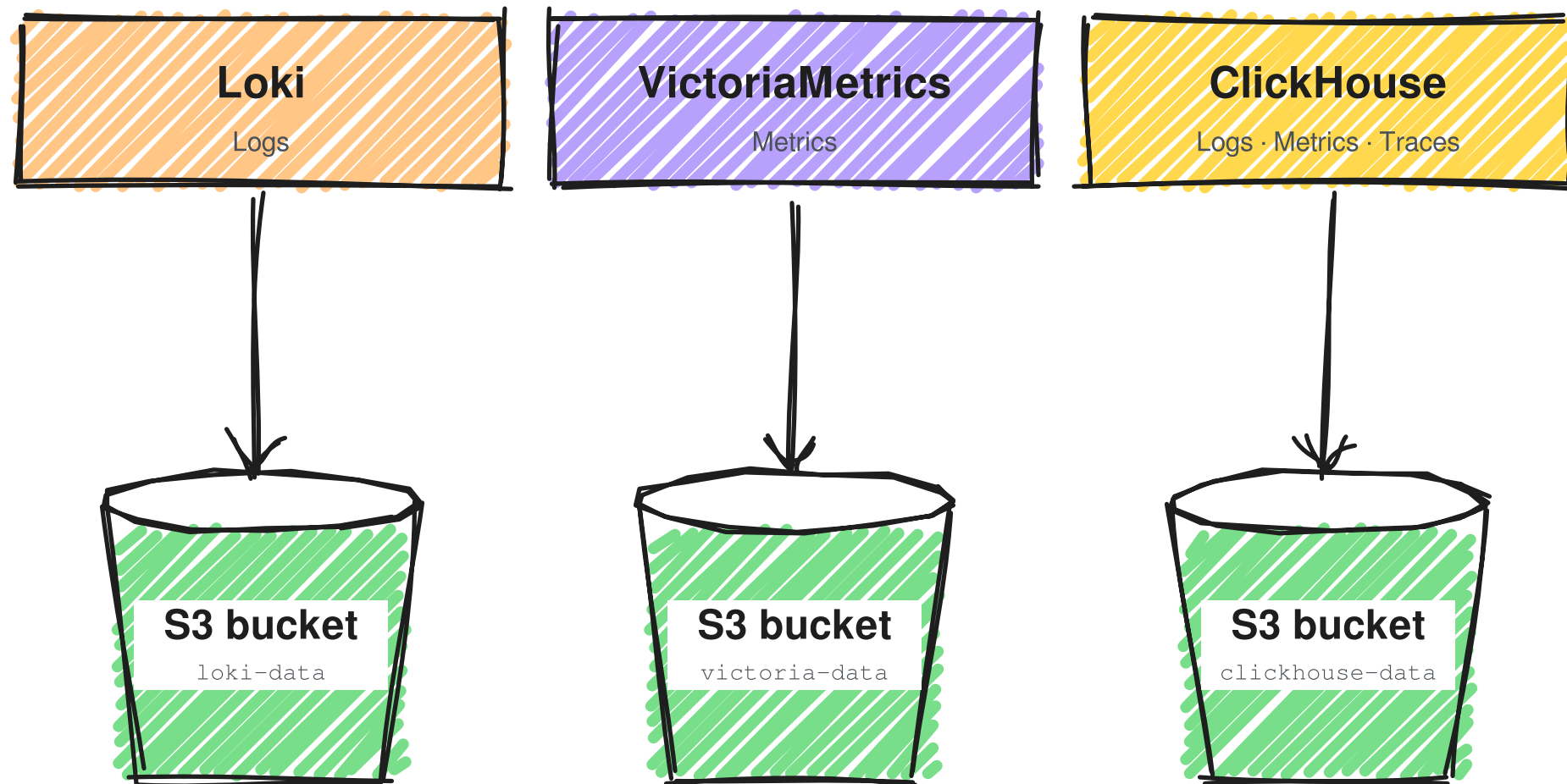
Scan to watch the talk on YouTube (from OSOD
2024!)



youtube.com/watch?v=uDY_EsE47sA

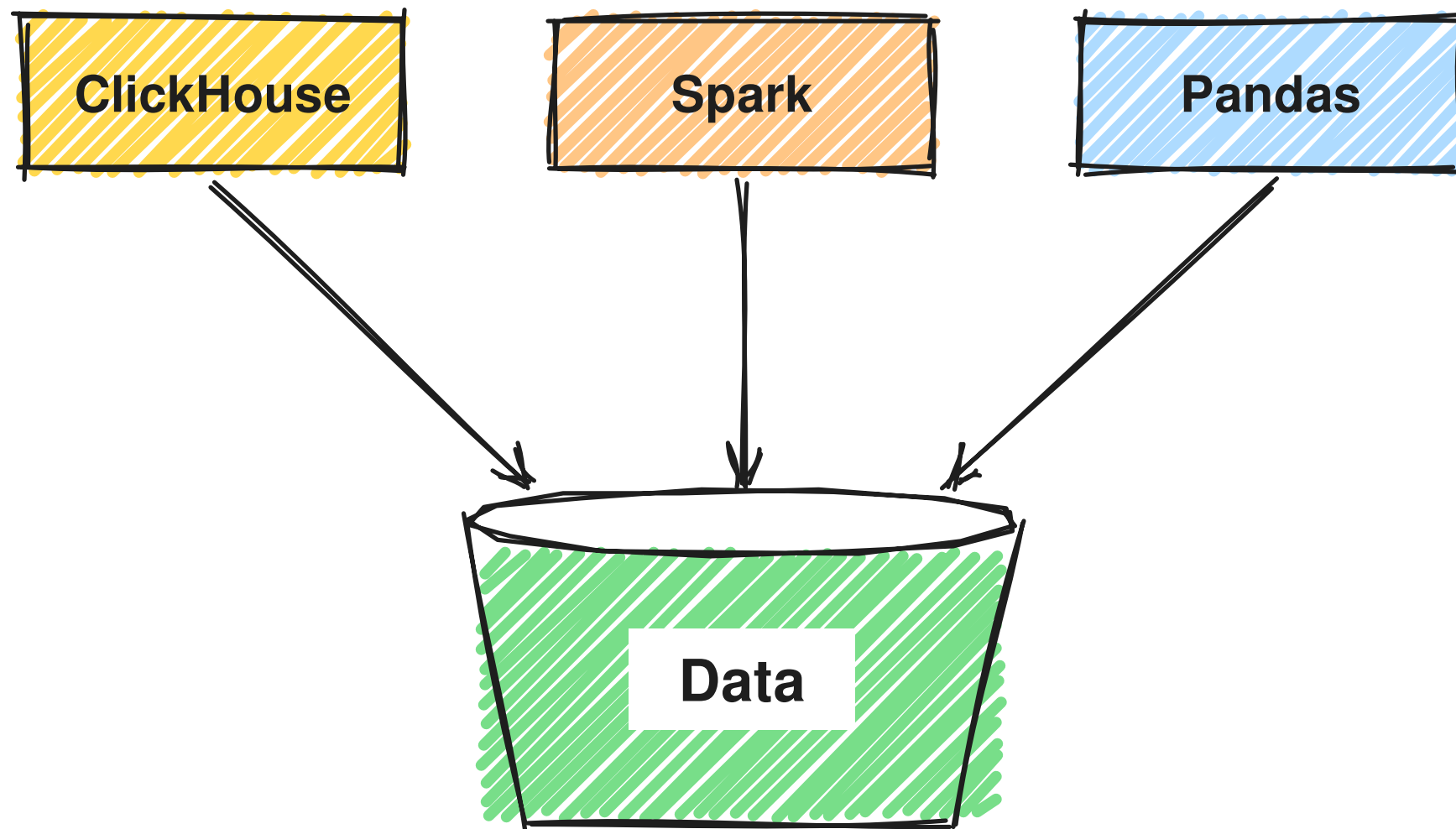
JUST ONE PROBLEM LEFT

How to manage the massive volume of data?



HOW TO MANAGE VOLUME? AND COSTS?

- Object Storage!
- AWS S3
- GCP, Backblaze, Hetzner, Digital Ocean...
- CEPH, RustFS, Garage, SeaweedFS, MinIO...
- **Keep the data where the compute is!**

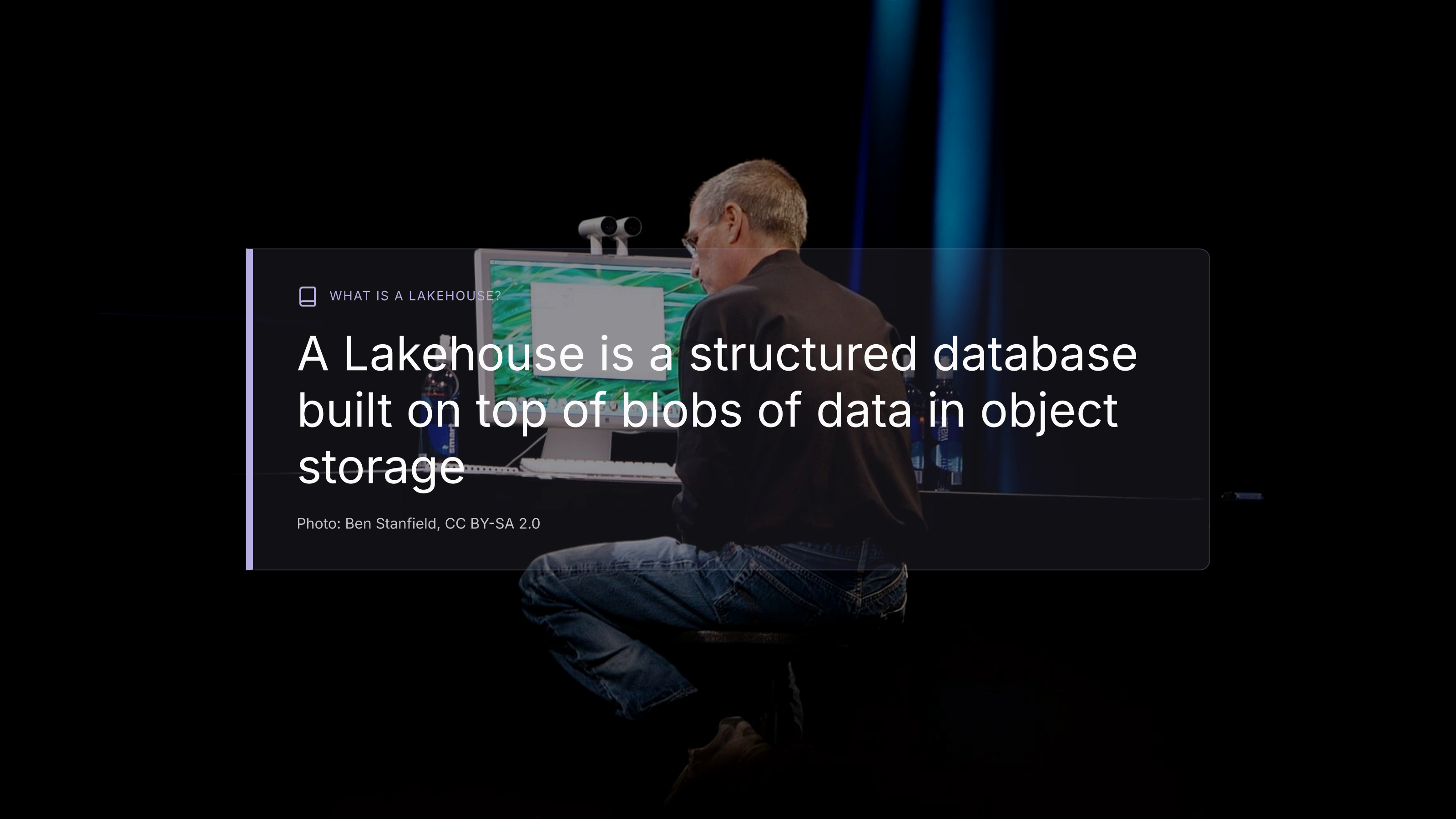


**WHAT IF MULTIPLE
TOOLS COULD SHARE
A SINGLE COPY OF THE
DATA?**

One copy of the data, many tools

LAKEHOUSE 101

WHAT IF MULTIPLE TOOLS COULD SHARE A SINGLE COPY OF
THE DATA?



WHAT IS A LAKEHOUSE?

A Lakehouse is a structured database built on top of blobs of data in object storage

Photo: Ben Stanfield, CC BY-SA 2.0

LAKEHOUSE 101

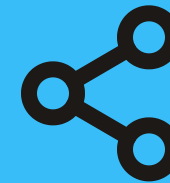
Why build a lakehouse?



Infinitely scalable
Grow without re-architecting



Much cheaper per GB
Object storage economics



Multi-access
Many tools, one copy of the data

LAKEHOUSE 101

What makes a lakehouse?

1

STORAGE

Where does the raw data live?

2

FILE FORMAT

What format is the raw data in?

3

CATALOG

How is the metadata managed?

LAKEHOUSE 101

What makes a lakehouse?

1

STORAGE

S3 (or compatible)

2

FILE FORMAT

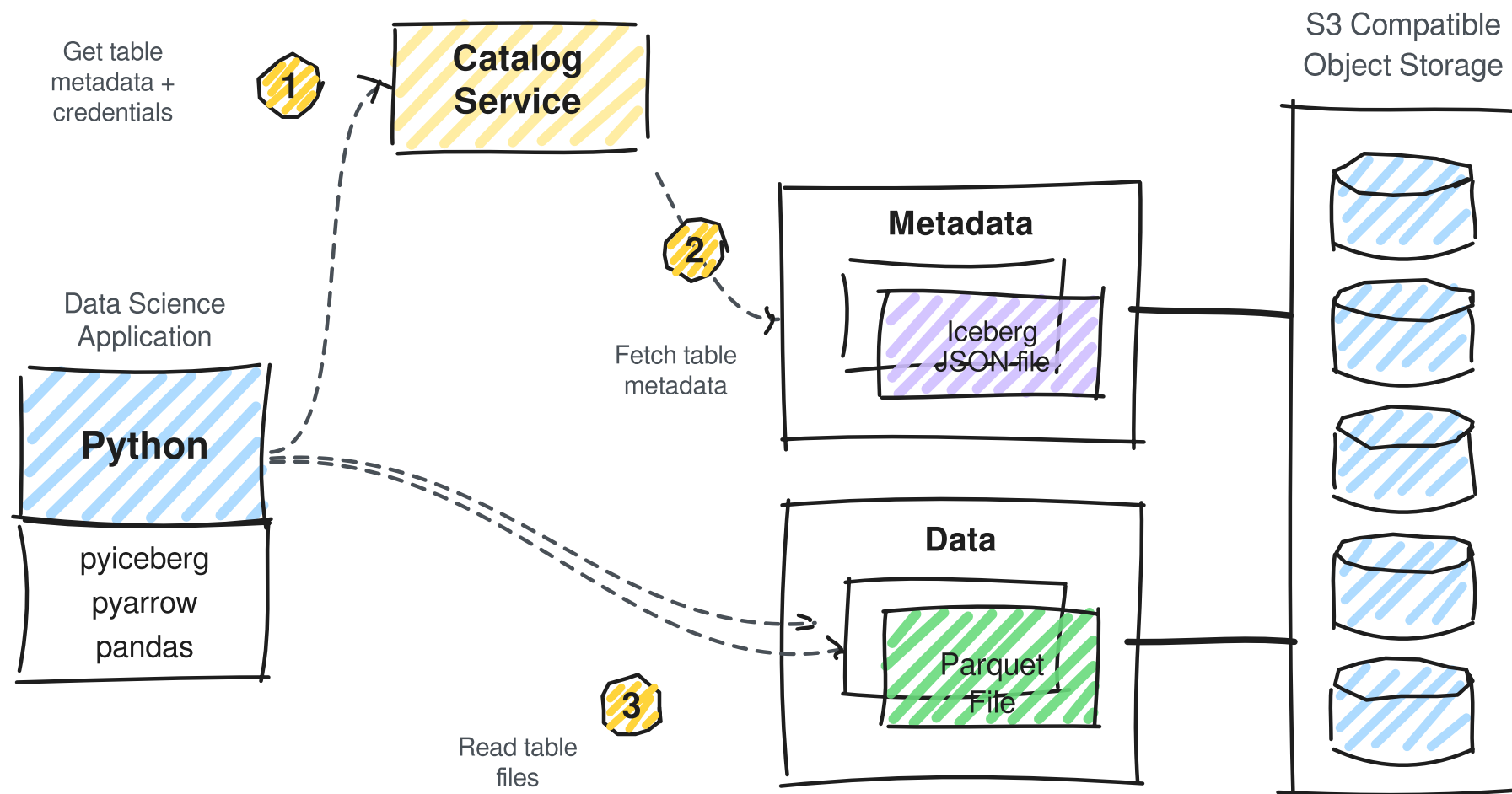
Parquet

3

CATALOG

Iceberg





HOW DO APPLICATIONS ACCESS AN ICEBERG TABLE?

Catalog → metadata → Parquet files

⚠ WARNING

A traditional lakehouse is not real-time

Photo: matt buchanan, CC BY 2.0

THE OBSERVABILITY LAKEHOUSE

OTel in, SQL out, real-time, infinite scale

Heroes of the Observability Lakehouse



A SPEEDBOAT FOR THE LAKEHOUSE



Apache Arrow

Highly-performant transport protocol (and in-memory format)



Altinity Antalya

ClickHouse with Iceberg as primary storage

APACHE ARROW IS AWESOME

APACHE ARROW

Highly performant stateful
transport protocol

Columnar

In-memory format

ADBC

Highly performant Arrow-based
database connectivity (e.g. for
ClickHouse)

OTEL-ARROW

Project to provide end-to-end
OpenTelemetry support using
Rust and Arrow (OTAP)

Speaks Iceberg/Parquet

IT'S COLUMNS ALL THE WAY DOWN

ARROW

Columnar in-memory and wire format

OTEL-ARROW

Emits columnar OTAP, speaks ADBC and Parquet

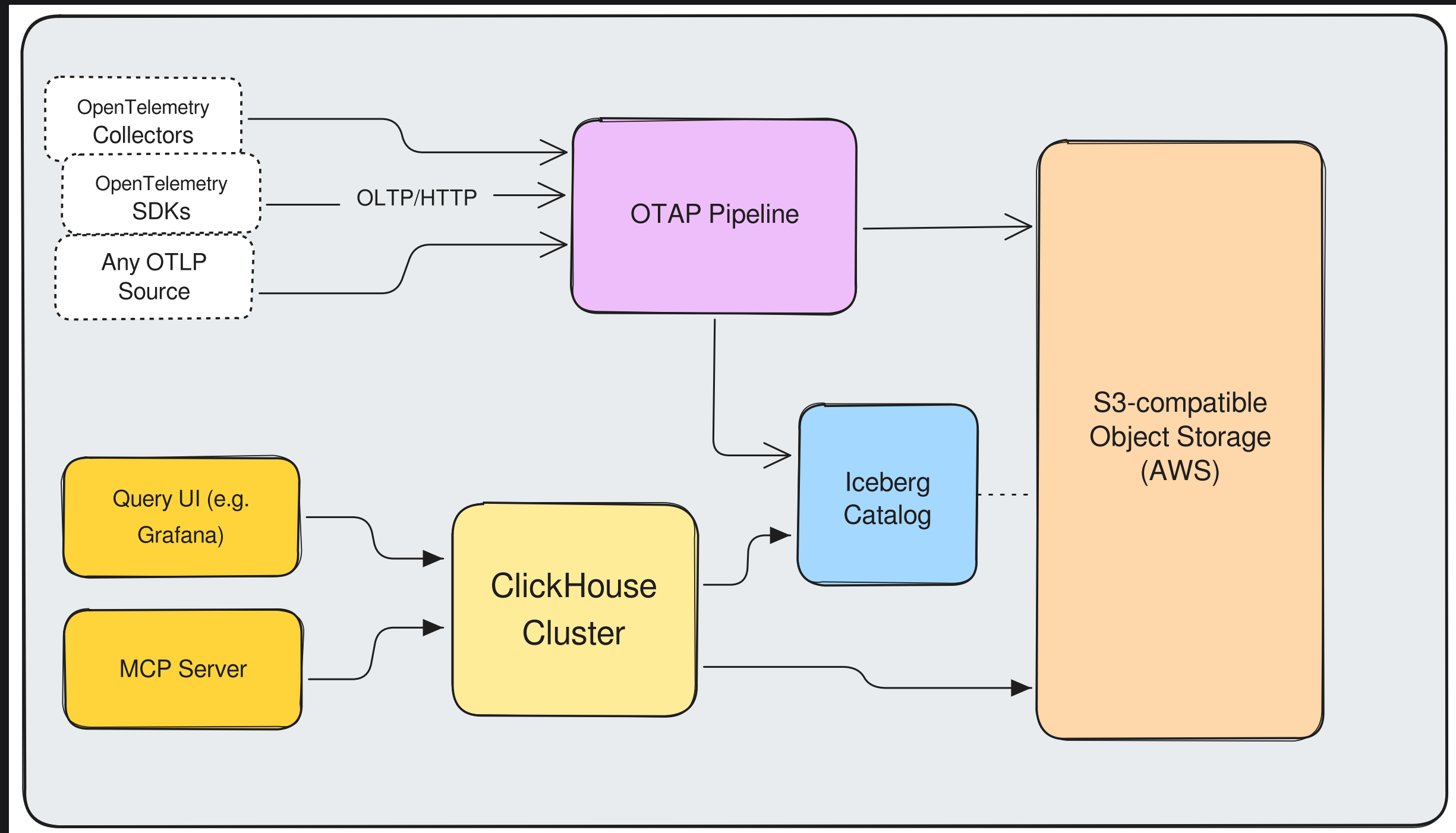
CLICKHOUSE

Columnar query engine, speaks ADBC

PARQUET

Columnar lakehouse storage format

OTAP Architecture



ALTINITY ANTALYA

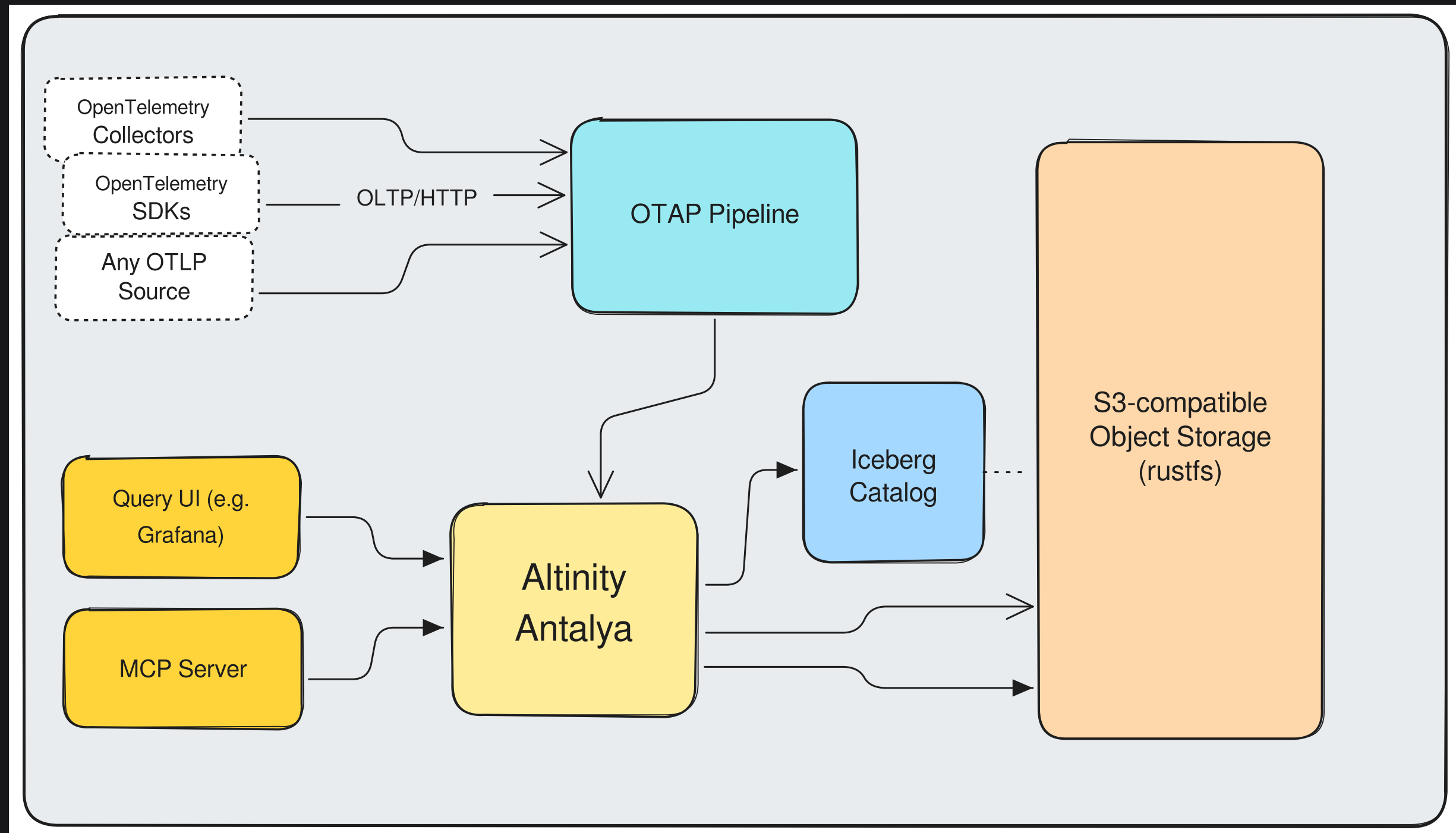
- 1 **ALTINITY'S APACHE 2.0 FORK OF CLICKHOUSE**
100% COMPATIBLE WITH UPSTREAM

- 2 **FIRST-CLASS LAKEHOUSE INTEGRATION**
Iceberg as primary storage

- 3 **HYBRID TABLES**
The best of MergeTree + Iceberg

- 4 **OIDC & OAUTH**
For federated access and delegation to agents

Antalya Architecture



 WARNING

This is just a reference architecture, not a complete solution

Photo: Ben Stanfield, CC BY-SA 2.0

GO BUILD SOMETHING AWESOME

THANK YOU

@joshleecreates.bsky.social

Scan for slides, resources, and to connect with me
on LinkedIn



joshuaamlee.com/events/osod-26