

Building a fast & open-source data stack with ClickHouse and Cube



Adnan Rahic

Senior Developer Advocate at Cube



Robert Hodges

CEO at Altinity



Community Code of Conduct

- We want to foster an open and welcoming environment where everyone feels they belong in the Cube.js community.
- The full text of our Code of Conduct is available at https://github.com/cube-js/cube.js/blob/master/CODE_OF_CONDUCT.md
- Any instances of inappropriate/unacceptable behavior can be reported to conduct@cube.dev.

Some quick notes

- If you have any questions during the workshop, please feel free to type them in the “Q&A” (on Zoom & YouTube LiveStream).
- Recording of today’s workshop will be posted on the Cube Dev YouTube channel.
- All attendees will receive a post-event survey and we’d appreciate your feedback to help us with future events.
- Next workshop: Headless BI with streaming data with Cube’s founders 
 - Date: Oct 13 @ 9 AM PST
 - Topic: Product enhancements to support real-time/streaming data in Cube
 - Details at <https://cube.dev/events/building-real-time-streaming-data-applications-with-cube>

Previous webinars 🧐

- With Firebolt

<https://cube.dev/events/firebolt-webinar-data-apps>

- With Preset

<https://cube.dev/events/replacing-looker-with-cube-and-superset>

What we will discuss today

- Lightning-quick analytics with ClickHouse
- Data replication between MySQL and ClickHouse
- Integrating multiple data sources with Cube
- Exposing multiple APIs with Cube for data visualization
- Q & A

Overview of prerequisites

ClickHouse Setup

- Altinity.Cloud ClickHouse instance
 - <https://altinity.com/cloud-database/>
- Demo instance
 - <https://github.demo.trial.altinity.cloud:8443/play>
 - User: **demo**
 - Password: **demo**
 - Table: **ontime**



Prep Guide and Code

- Prep Guide

<https://descriptive-reply-0b7.notion.site/Workshop-Building-a-fast-open-source-data-stack-with-ClickHouse-and-Cube-Prep-Guide-9410839f6c784ae6a4c9bf16a86b4888>

- Code

<https://descriptive-reply-0b7.notion.site/Workshop-Building-a-fast-open-source-data-stack-with-ClickHouse-and-Cube-Code-2ff2c422c6c14fad8b87ca066b327763>

The goal of this workshop

- Build lightning-quick open-source data apps with ClickHouse and Cube!

Altinity ❤️ **ClickHouse**



Altinity helps users run ClickHouse anywhere

Altinity.Cloud

**Managed ClickHouse in
public and private clouds**

Support

**24x7 support from the best
ClickHouse experts on earth**



Altinity

Builds

**Altinity Stable Builds with 3
years of support**

Training and Content

**Online training,
documentation, and videos**

ClickHouse is a SQL Data Warehouse

Understands SQL

Runs on bare metal to cloud

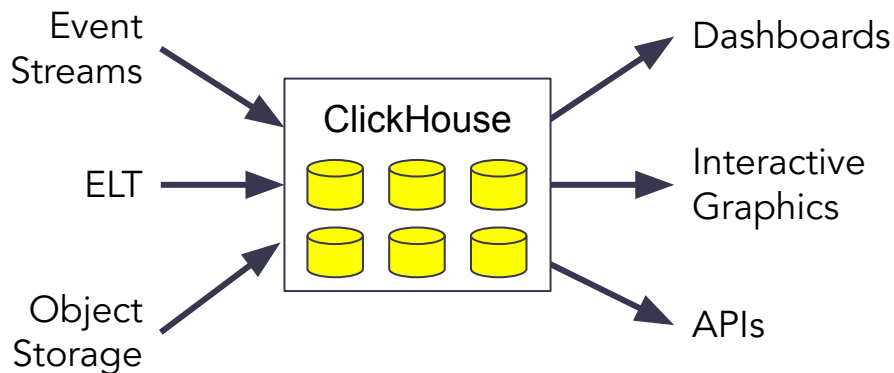
Shared nothing architecture

Stores data in columns

Parallel and vectorized execution

Scales to many petabytes

Is Open source (Apache 2.0)



It's the core engine for
real-time analytics

What ClickHouse Is Not...



ClickHouse does not have full
ACID transactions and
updates are slow

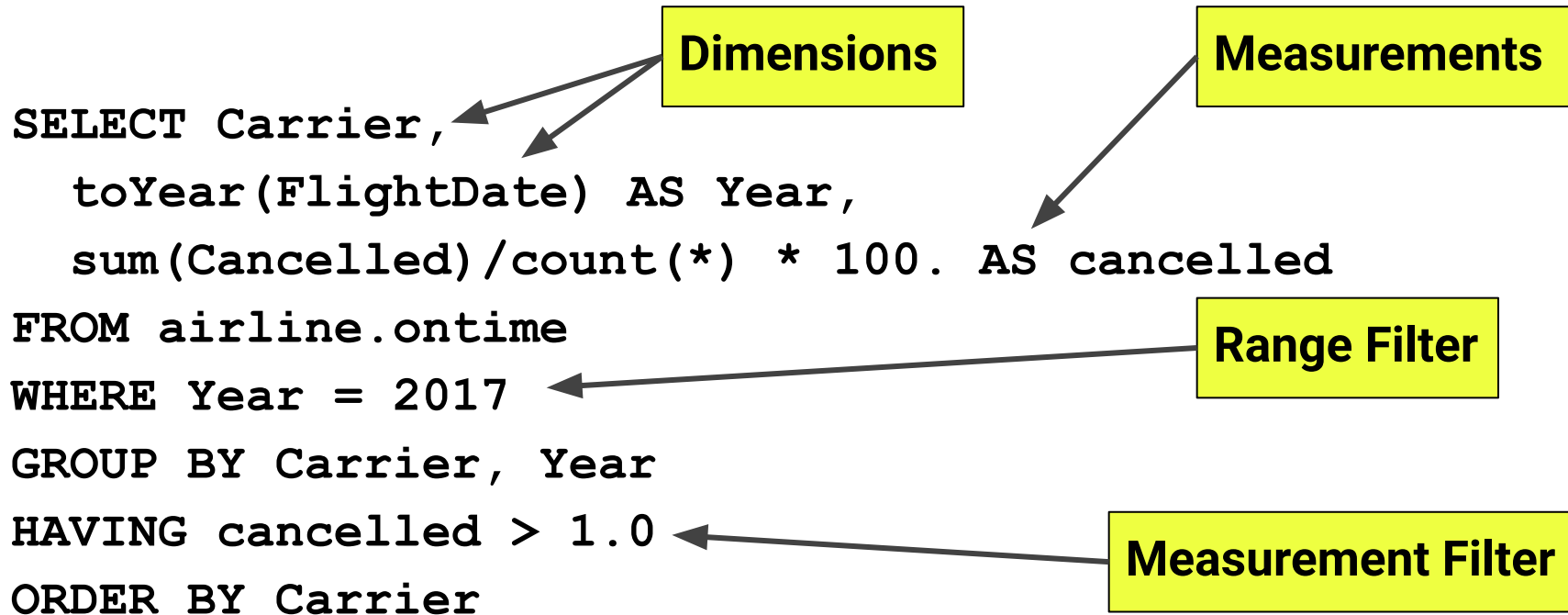


ClickHouse values speed over SQL
standards and runs anywhere

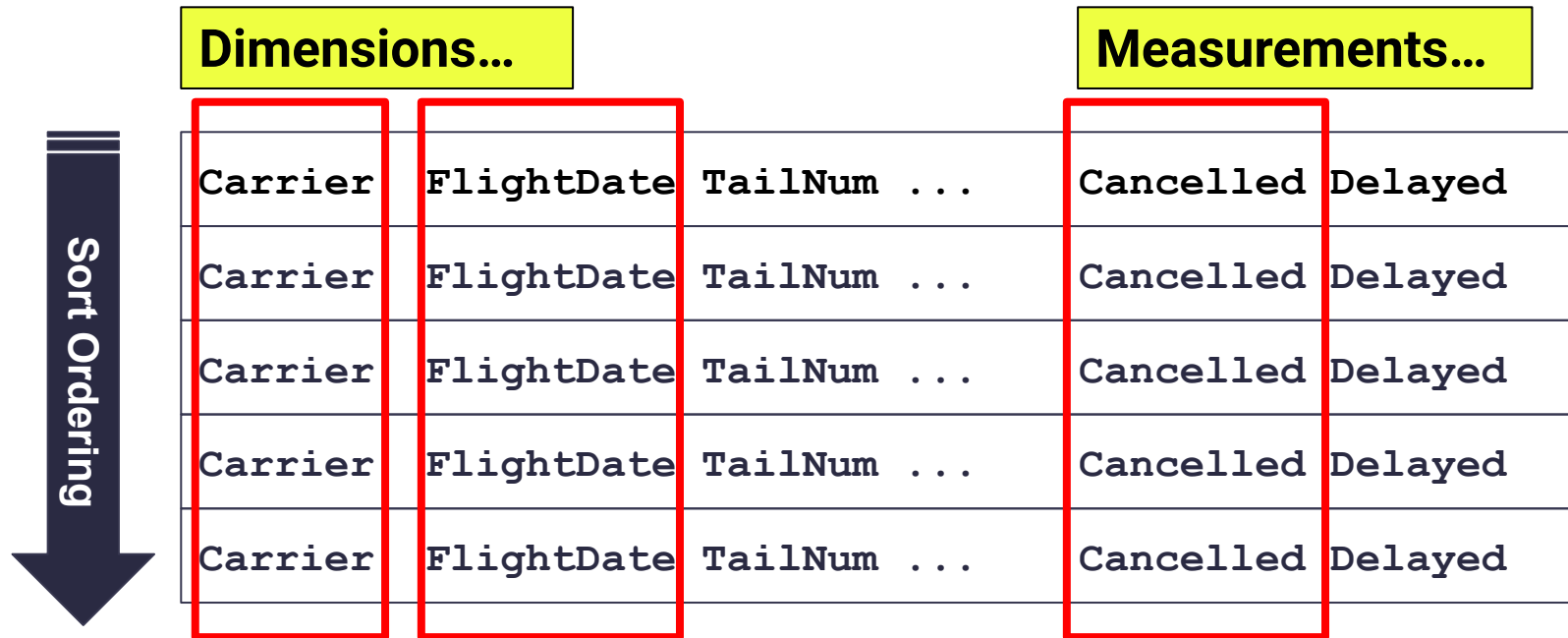


ClickHouse is designed to read large
blocks of data for a few users

Finding airlines with the most cancellations

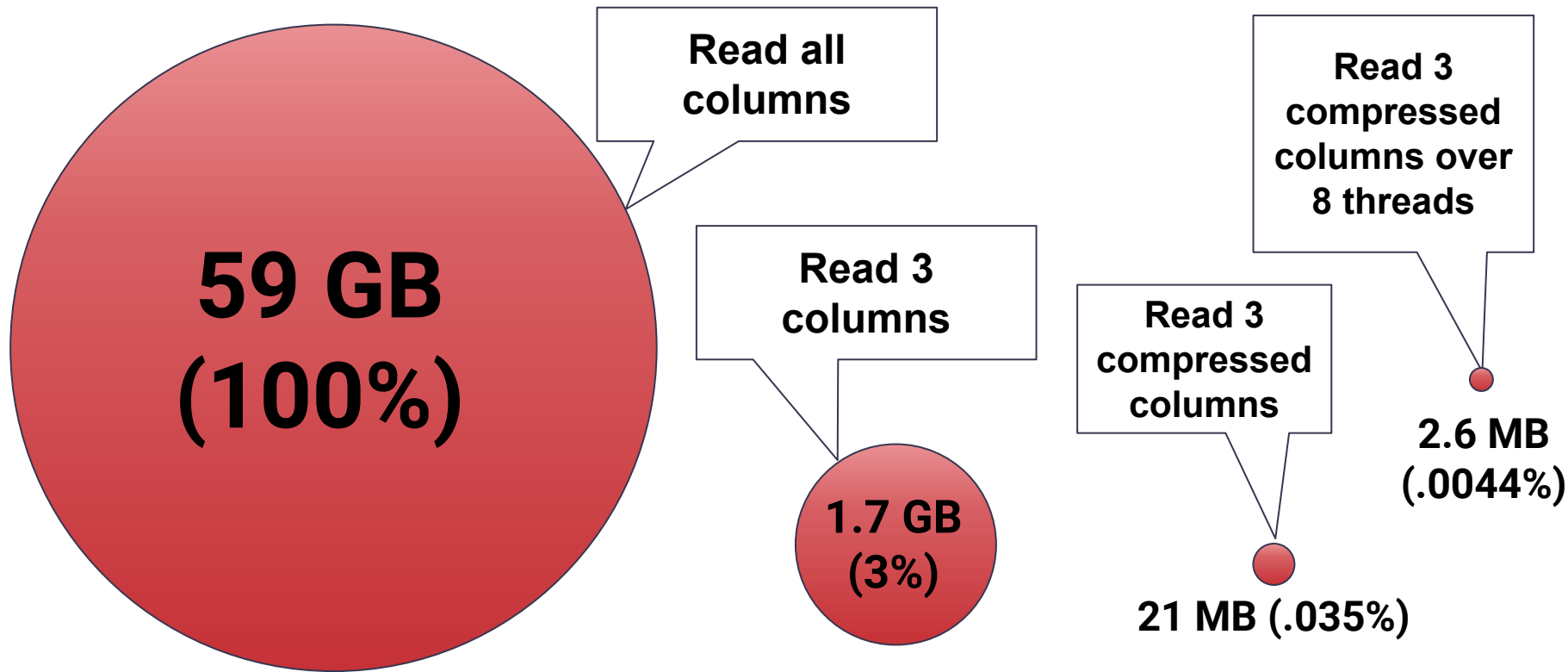


ClickHouse stores data in columns, not rows



Dimensions...			Measurements...	
Carrier	FlightDate	TailNum ...	Cancelled	Delayed
Carrier	FlightDate	TailNum ...	Cancelled	Delayed
Carrier	FlightDate	TailNum ...	Cancelled	Delayed
Carrier	FlightDate	TailNum ...	Cancelled	Delayed
Carrier	FlightDate	TailNum ...	Cancelled	Delayed

ClickHouse reduces I/O and executes in parallel



Example of results

```
SELECT Carrier, toYear(FlightDate) AS Year,  
       (sum(Cancelled) / count(*)) * 100. AS cancelled  
FROM airline.ontime WHERE Year = 2017  
GROUP BY Carrier, Year HAVING cancelled > 1.  
ORDER BY Carrier ASC
```

Carrier	Year	cancelled
AA	2017	1.3541615533252709
B6	2017	2.77511769472366
EV	2017	2.9351389081141894

. . .

8 rows in set. Elapsed: **0.045 sec.** Processed 10.98 million
rows, 38.39 MB (**241.85 million rows/s.**, **856.56 MB/s.**)

Getting started with ClickHouse

Install ClickHouse on your laptop or VM

ClickHouse Community Builds

Monthly builds

LTS builds every 6 months

(1 year of support)

<https://clickhouse.com>

Altinity Stable Builds

Prod-ready LTS builds only

(3 years of support)

<https://docs.altinity.com>

Ubuntu example

```
sudo apt-get install -y clickhouse-server clickhouse-client
```

```
sudo systemctl start clickhouse-server
```

Create a table for your data

```
CREATE TABLE ontime (  
    Year UInt16,  
    Quarter UInt8,  
    Month UInt8,  
    ...  
) ENGINE = MergeTree()  
PARTITION BY toYYYYMM(FlightDate)  
PRIMARY KEY (Carrier, FlightDate)  
ORDER BY (Carrier, FlightDate,  
    DepTime)
```

Table columns

Table engine type

How to break data
into parts

How to index and
sort data in each part

Load data with clickhouse-client

Input file ontime-2022-07.csv

```
"Year", "Quarter", "Month", "DayofMonth", "DayOfWeek",  
"FlightDate", "Reporting_Airline", ...  
2022, 3, 7, 19, 2, 2022-07-19, "OH", ...  
2022, 3, 7, 20, 3, 2022-07-20, "OH", ...  
2022, 3, 7, 21, 4, 2022-07-21, "OH", ...  
2022, 3, 7, 24, 7, 2022-07-24, "OH", ...
```

```
cat ontime-2022-07.csv | clickhouse-client \  
--database=airline \  
--query='INSERT INTO ontime FORMAT CSVWithNames'
```

Make data smaller with compression

Compression reduces a string of bits to a different and hopefully smaller string of bits

LZ4 Compression

Fast to compress and decompress. Default for ClickHouse

TailNum

N3HYAA

N377AA

N377AA

N922AA

N862AA

N3EEAA

N536AA

N665AA

N862AA

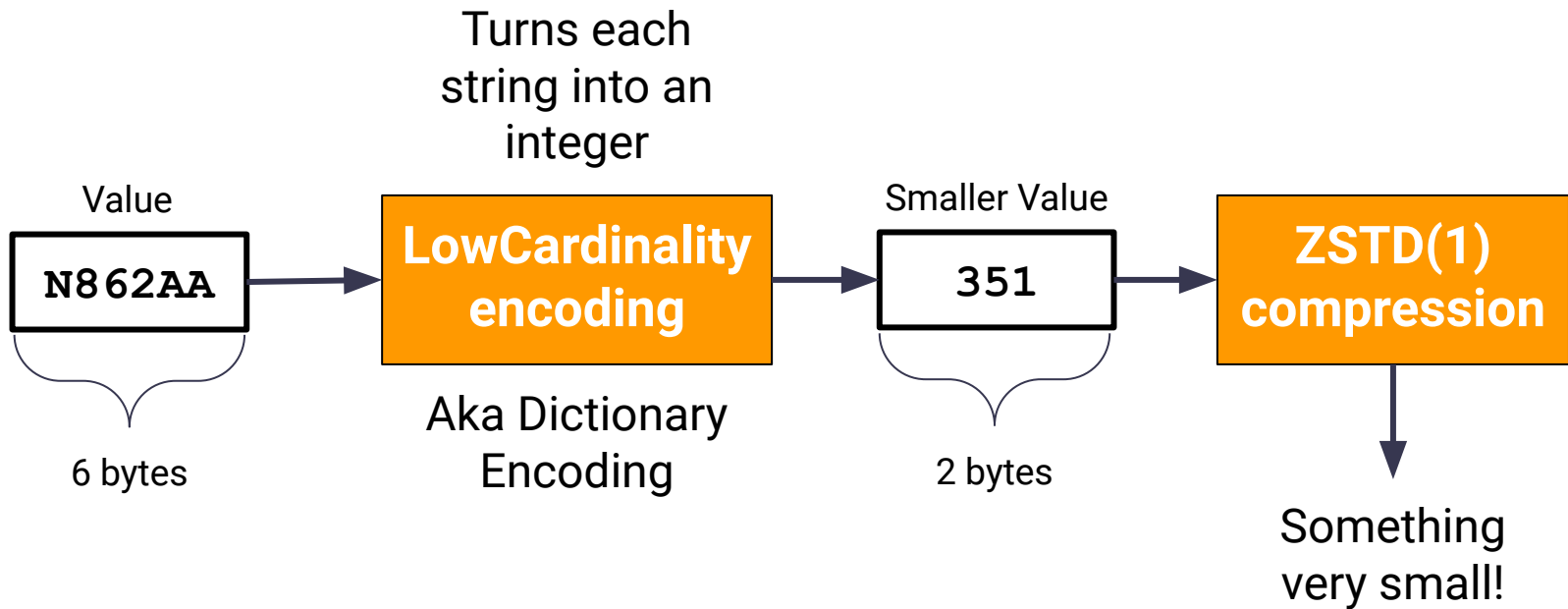
N936AA

N4YBAA

ZSTD Compression

Slower to compress but generally smaller than LZ4. You have to ask for it

Codecs reduce data before compression



How to add codecs and change compression

```
ALTER TABLE ontime  
    MODIFY COLUMN TailNum LowCardinality(String) CODEC(ZSTD(1))
```

```
ALTER TABLE ontime  
    MODIFY COLUMN Year CODEC(DoubleDelta, ZSTD(1))
```

```
-- Force ClickHouse to change data
```

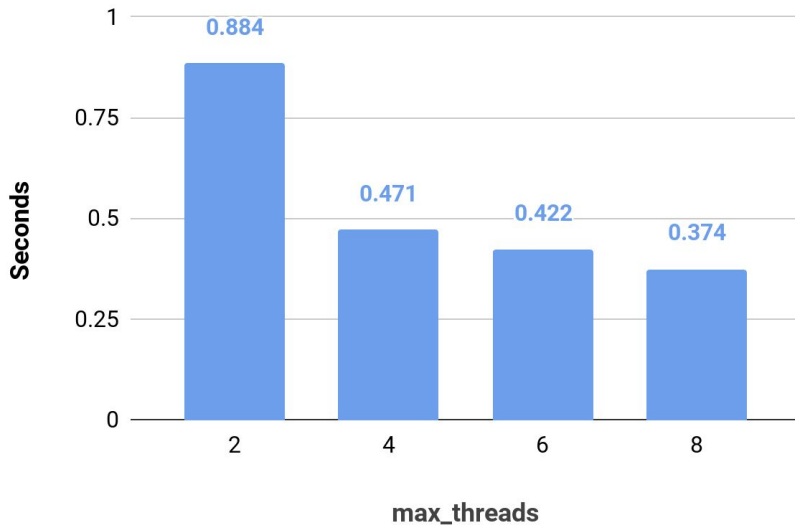
```
ALTER TABLE ontime UPDATE TailNum = TailNum, Year = Year WHERE  
1
```

Add more compute using max_threads property

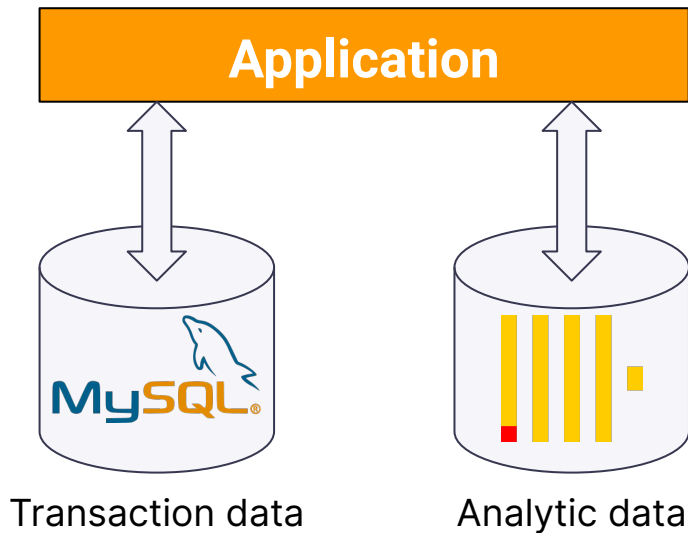
```
SELECT toYear(FlightDate) year,  
       sum(Cancelled)/count(*) cancelled,  
       sum(DepDel15)/count(*) delayed_15  
FROM airline.ontime  
GROUP BY year ORDER BY year LIMIT 10  
  
SET max_threads = 2  
  
SET max_threads = 4  
  
. . .
```

max_threads defaults to half
the number of CPU cores

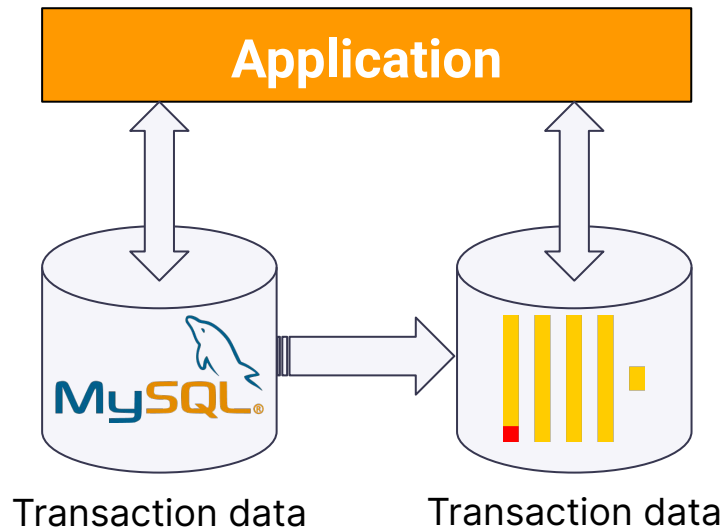
Query Response



Design patterns for versatile OLTP + OLAP



Separate stores for different data

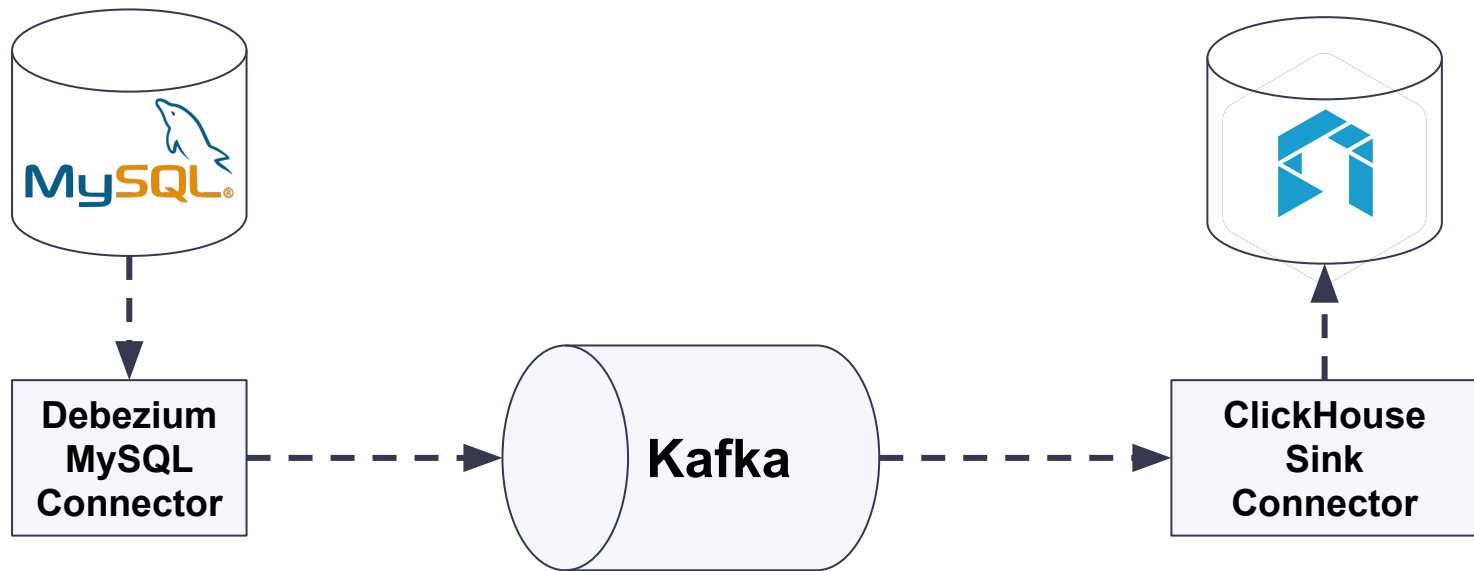


Transaction mirrored to data warehouse for analytics

Implementing MySQL to ClickHouse mirroring

Amazon Aurora

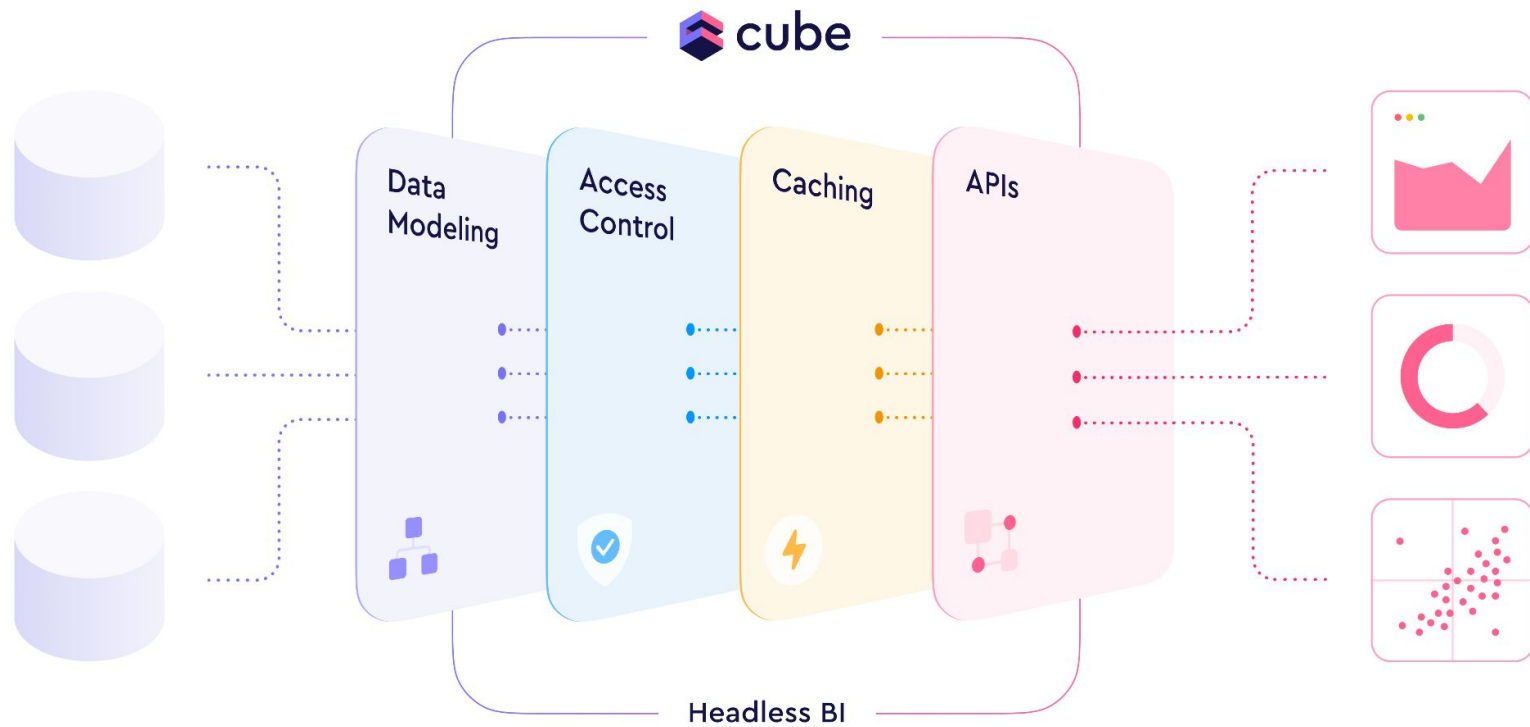
Altinity.Cloud





Cube: API-first Headless BI

- What is Headless BI?
- 4 pillars 





Headless BI

Data Sources

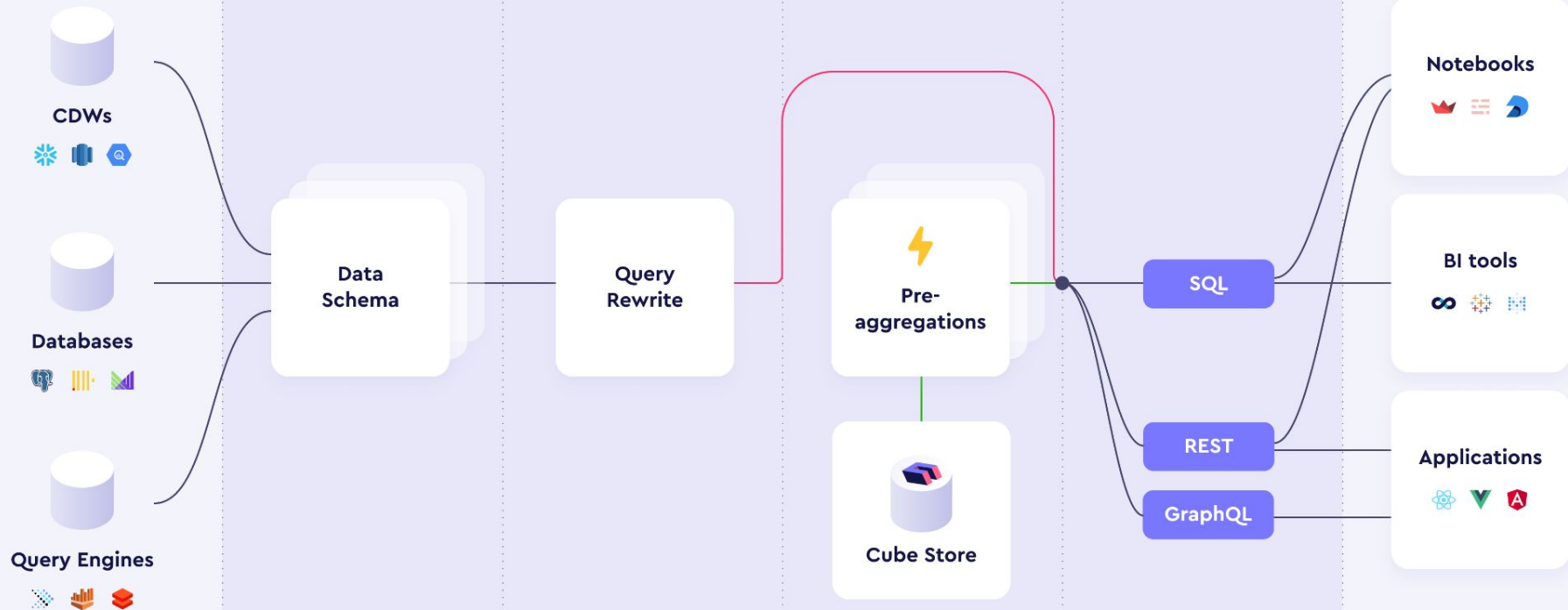
Data Modeling

Access Control

Caching

APIs

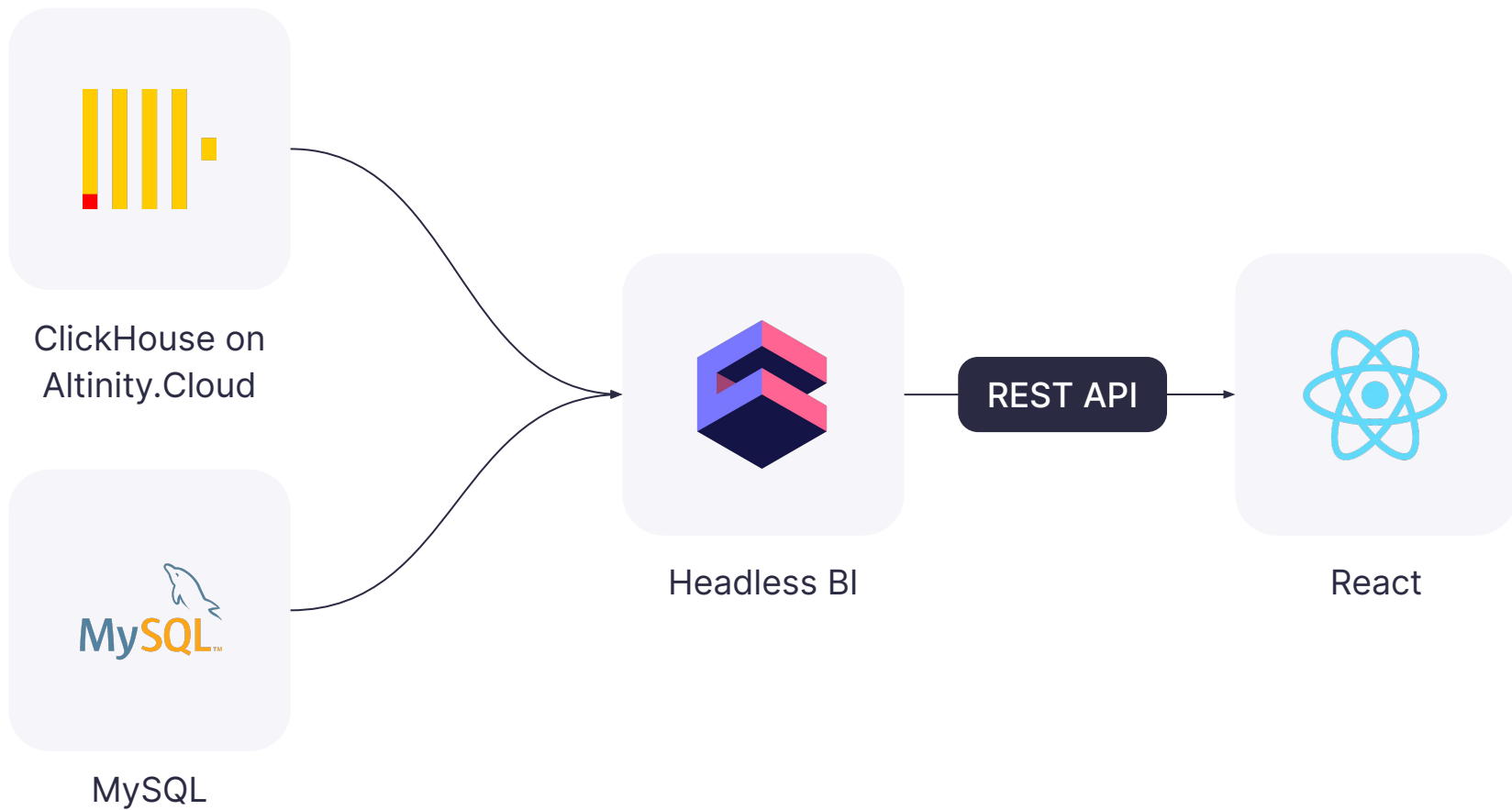
Data Consumers



Multitenancy

The goal

- Use two data sources and seamlessly switch between both
- Use Cube as the intermediary metrics layer
- Seamlessly deliver data from both data sources through REST API



You'll learn...

- **Data modeling** → Multiple data source integrations
- **Access control** → Multi-tenancy
- **APIs** → REST API for data visualization

Part 1:

Seamless integration with multiple data sources

This section will cover...

- Data Modeling
- Multiple Data Sources
- Multi-tenancy

Demo:

Explore the **ontime** dataset

Data Modeling

- Tables mapped to OLAP Cubes
- Table columns mapped to dimensions
- A dimension is an attribute related to a measure
- A measure is an aggregation over a certain dimension

Data Modeling: Custom measures

```
avgDepDelay: {  
  type: `avg`,  
  sql: `${CUBE}.DepDelay`,  
},
```

Data Modeling: Custom measures

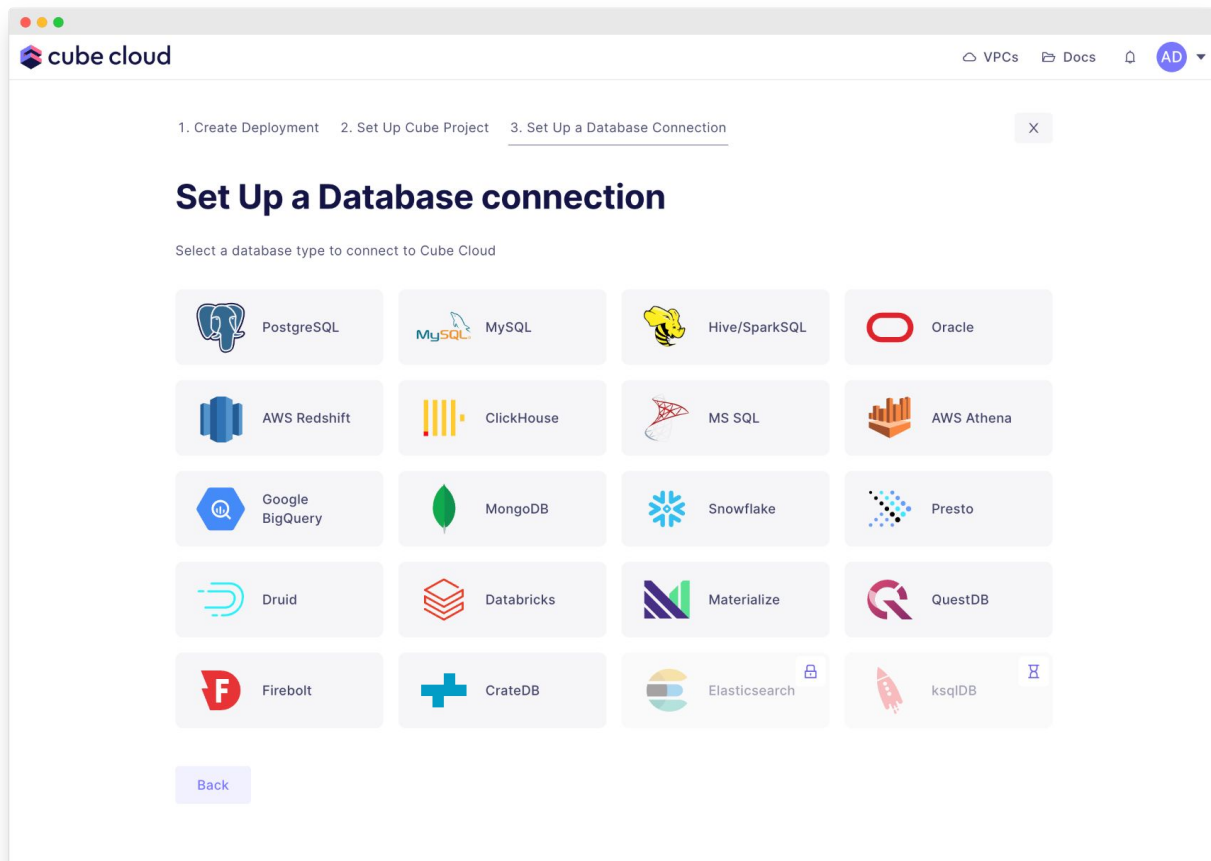
```
avgDepDelayGreaterThanTenMinutesPercentage: {  
  type: `number`,  
  sql: `avg(${CUBE}.DepDelay>10)*100`,  
  format: `percent`  
},
```

Demo:

Multiple data sources:

Custom measures

Cube supports...



Cube supports...

- Data warehouses, transactional databases, query engines, etc...

Today we care about...

- MySQL
- ClickHouse

Multiple data sources with multiple Cube deployments?

- Two physically separate Cube deployments?

cube cloud

1. Create Deployment2. Set Up Cube Project3. Set Up a Database Connection

Set Up a Database connection

←MySQLMySQL

Enter database credentials to connect to your database
To learn more about this configuration and Cube's features specific to MySQL [visit our docs](#).

Hostname

Port

Database

Username

Password

SSL

3306

☐

Allow connections to your database from Cloud IPs

18.206.29.210
3.211.151.66
44.197.48.141
44.196.128.87
18.204.89.172
35.174.97.76

Not ready to connect your data warehouse?
[Schedule a Demo](#)

Need an enterprise plan or don't see your data warehouse here?
[Contact us](#)

Back

Apply

cube cloud

VPCs DocsAD

1. Create Deployment2. Set Up Cube Project3. Set Up a Database Connection

Set Up a Database connection

←ClickHouse

Enter database credentials to connect to your database
To learn more about this configuration and Cube's features specific to ClickHouse [visit our docs](#).

Hostname

Port

Database

Username

Password

SSL

8123

☐

Allow connections to your database from Cube Cloud IPs

18.206.29.210
3.211.151.66
44.197.48.141
44.196.128.87
18.204.89.172
35.174.97.76

Not ready to connect your data warehouse?
[Schedule a Demo](#)

Need an enterprise plan or don't see your data warehouse here?
[Contact us](#)

Back

Apply

Here's a better way...

- Multiple data sources with the `dataSource` attribute
- Multi-tenancy with `securityContext` configuration

Multiple data sources

- When your Cube schema is spread across multiple different databases
 - `dataSource` property

Demo:

Multiple data sources:

Driver configuration

Multi-tenancy

- When you need to serve different datasets for multiple tenants
 - `securityContext` configuration
 - Uses JWT tokens

Demo:

Multi-tenancy

Part 2:

Exposing multiple APIs

This section will cover...

- Cube exposes GraphQL, SQL and REST APIs
- Build two data apps with React with Cube's REST API

Exposed APIs by default!

- No more developing a custom data provider!!
- 3 exposed APIs by default

REST API

[How to connect your application](#)

```
https://blue-stork.aws-us-east-1.cubecloudapp.dev/cubejs-api/v1
```



SQL API

[How to connect your BI tool](#)

With this new API, Cube can now function as a headless BI layer to provide metrics to any SQL-compatible querying and visualization tool.

GraphQL API beta

[How to connect your application](#)

```
https://blue-stork.aws-us-east-1.cubecloudapp.dev/cubejs-api/graphql
```



Exposed APIs by default!

- One central location for accessing all data
- API agnostic
- APIs return identical data for every downstream tool

Querying Cube's API

BI & Data exploration tools



Superset



Tableau



Power BI



Metabase

Querying Cube's API Notebooks



Hex



Jupyter



Streamlit



Deepnote

SQL

Copy to Clipboard

React ▾

Chart.js ▾

Chart

```
SELECT
  "orders".o_orderstatus "orders__o_orderstatus",
  date_trunc(
    'day',
    (
      "orders".o_orderdate :: timestampz AT TIME ZONE 'UTC'
    )
  ) "orders__o_orderdate_day",
  count(
    CAST("orders".o_orderkey as TEXT) || CAST("orders".o_custkey as TEXT)
  ) "orders__count"
FROM
  public.orders AS "orders"
GROUP BY
  1,
  2
ORDER BY
  2 ASC
LIMIT
  10000
```

Querying Cube's API

Low-code tools



Appsmith



Bubble



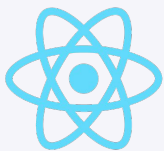
Retool



Budibase

Querying Cube's API

Front-end integrations



React



Vue



Angular

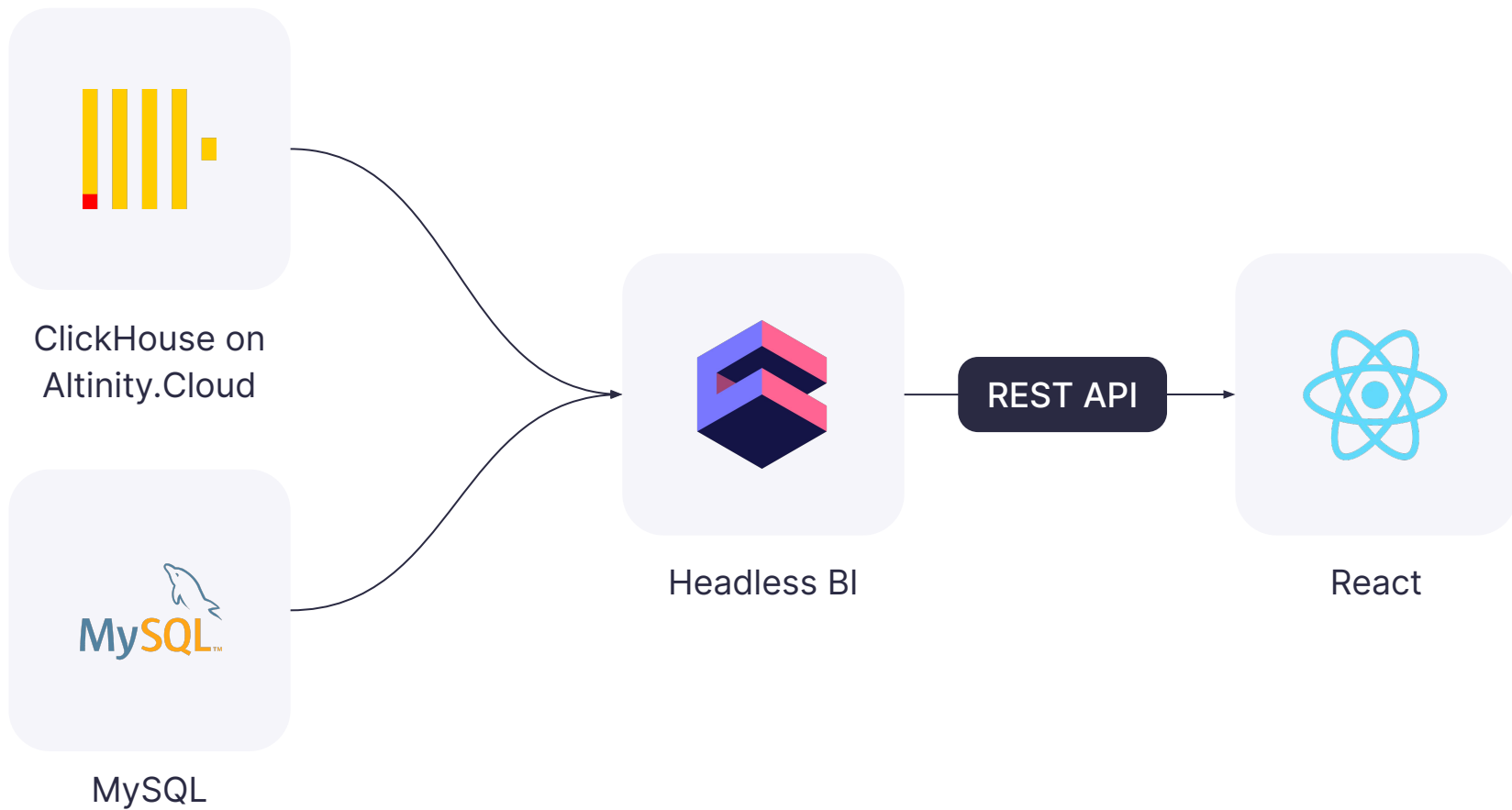


Vanilla JS

Query

 Copy to Clipboard

```
{
  "measures": [
    "Orders.count"
  ],
  "timeDimensions": [
    {
      "dimension": "Orders.createdAt",
      "granularity": "month"
    }
  ],
  "order": {
    "Orders.createdAt": "asc"
  },
  "limit": 5000
}
```



Demo:

Build a data app with React

CUBE\
CU BE\
_ / BE |
BE |
BE /
_ /
BE\
_ | ...

Resources

To learn more about Altinity visit...

- Altinity
 - <https://altinity.com/>
- Altinity.Cloud
 - <https://altinity.com/cloud-database/>

To learn more about Cube visit...

- Documentation:
 - Data sources: <https://cube.dev/docs/config/databases>
 - Multiple data sources and multi-tenancy: <https://cube.dev/docs/multitenancy-setup>
- Blog posts:
 - <https://cube.dev/blog/building-a-clickhouse-visualization-with-altinity-and-cube>
 - <https://cube.dev/blog/clickhouse-scalability-for-data-intensive-apps>
- Community support:
 - Discourse: <https://forum.cube.dev/>
 - Slack: <http://cube-js.slack.com/>

