ClickHouse at MessageBird

Analysing billions of events in real-time*

Aleksandar Aleksandrov & Félix Mattrat

NOVEMBER 2018

About us

Data engineers & Team leads

Aleksandar
Aleksandrov

Félix
Mattrat

ABOUT

Introducing MessageBird

MessageBird is a cloud communications platform that empowers consumers to communicate with your business in the same way they communicate with their friends - seamlessly, on their own timeline and with the context of previous conversations.

For additional information visit: www.messagebird.com

225+ Agreements

We have 225+ direct-to-carrier agreements with operators worldwide.

15,000+ Customers

Customers in over 60+ countries, across a great variety of industries.

180+ Employees

More than 180 employees speaking over 20 languages based in the Americas, Europe & Asia.

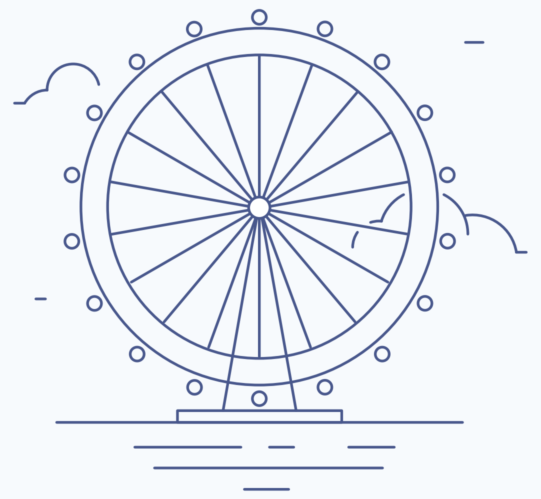
Our offices

Our bird's nests around the world provide customers in every time zone with 24/7 support.

AMSTERDAM

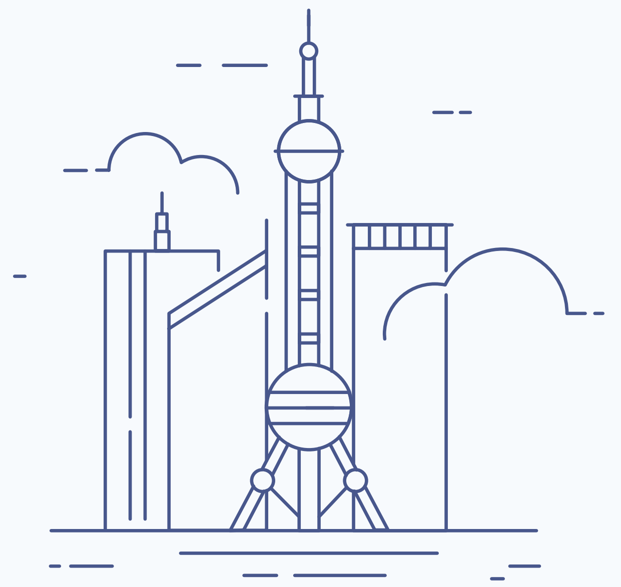
SINGAPORE

HAMBURG



LONDON

SAN FRANCISCO



SHANGHAI

SYDNEY

What’s on the menu?

Needs

Mostly about statistics and reporting

Internal needs

- State of the system
- Routing SMS
- Training algorithms
- ML Models

External needs

- Customer dashboard
- Reporting API

The landscape

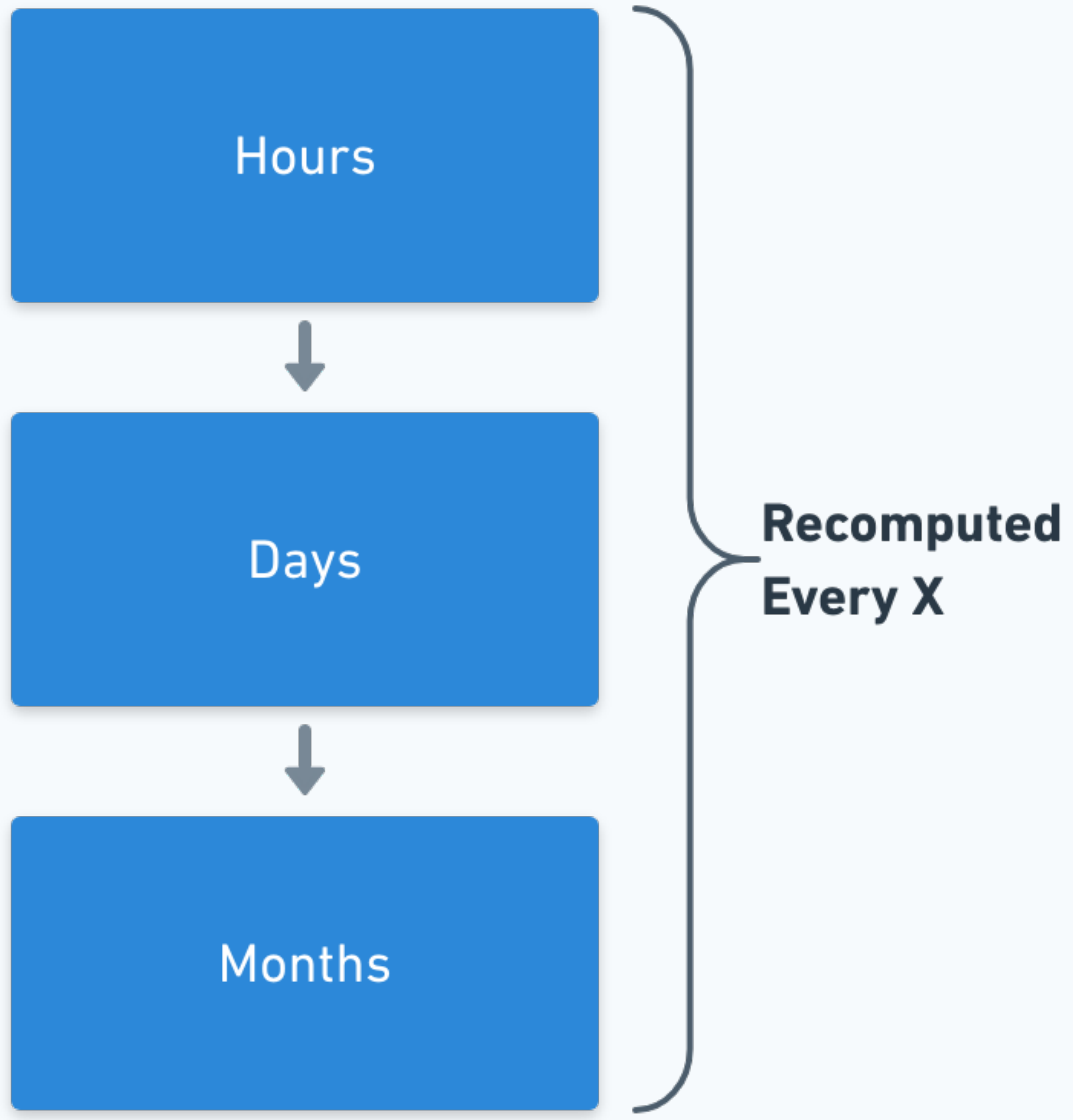
- Multiple carriers is messy - no uniformity of the data
- SMS messages go through many state changes up to months into the past
- Pricing (both carrier and customer) changes retro-actively

Age of Darkness

AGE OF DARKNESS

Hello CRON my old friend

- MySQL based
- Aggregates re-computed every X period of time
- Served us well for +5 years



AGE OF DARKNESS

Scaling problems

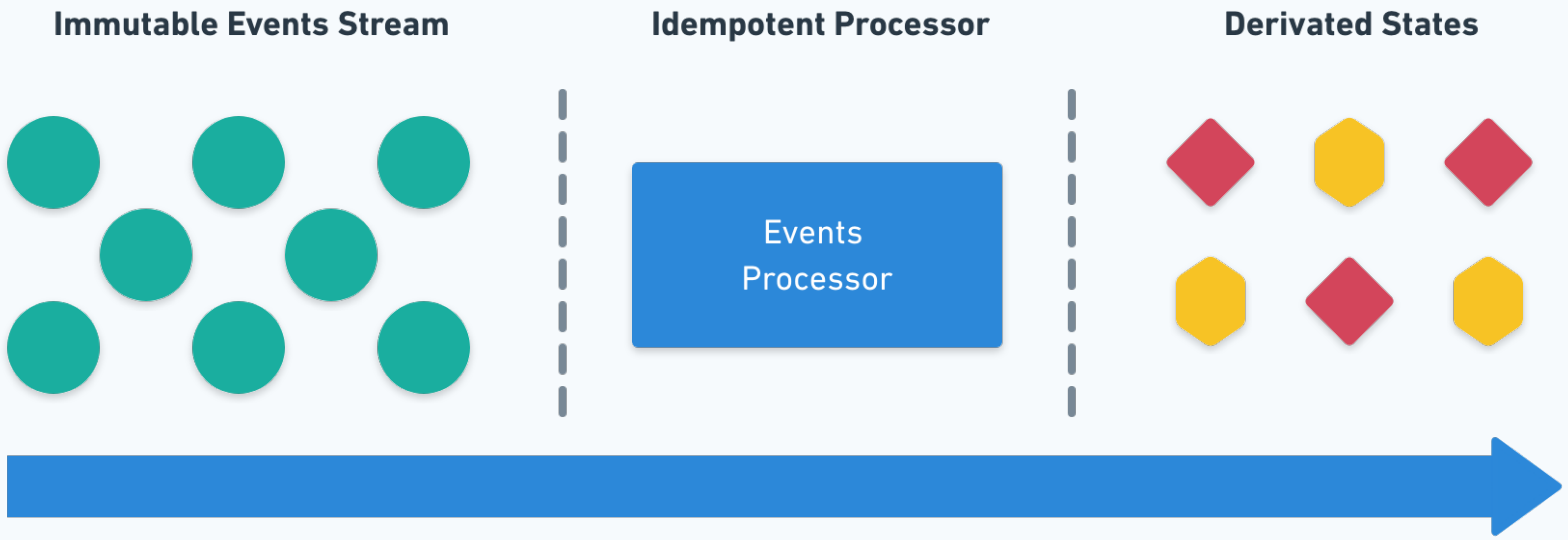
- The system had difficulty scaling and was often lagging
- Loss of granularity with pre-aggregation
- Performed poorly while doing analytical queries
- Inaccuracies

Re-thinking data collection

- Able to keep up with continuously changing SMS message states
- In real time*
- Scalable to handle MessageBird's global growth
- More flexible to accommodate wider use of data

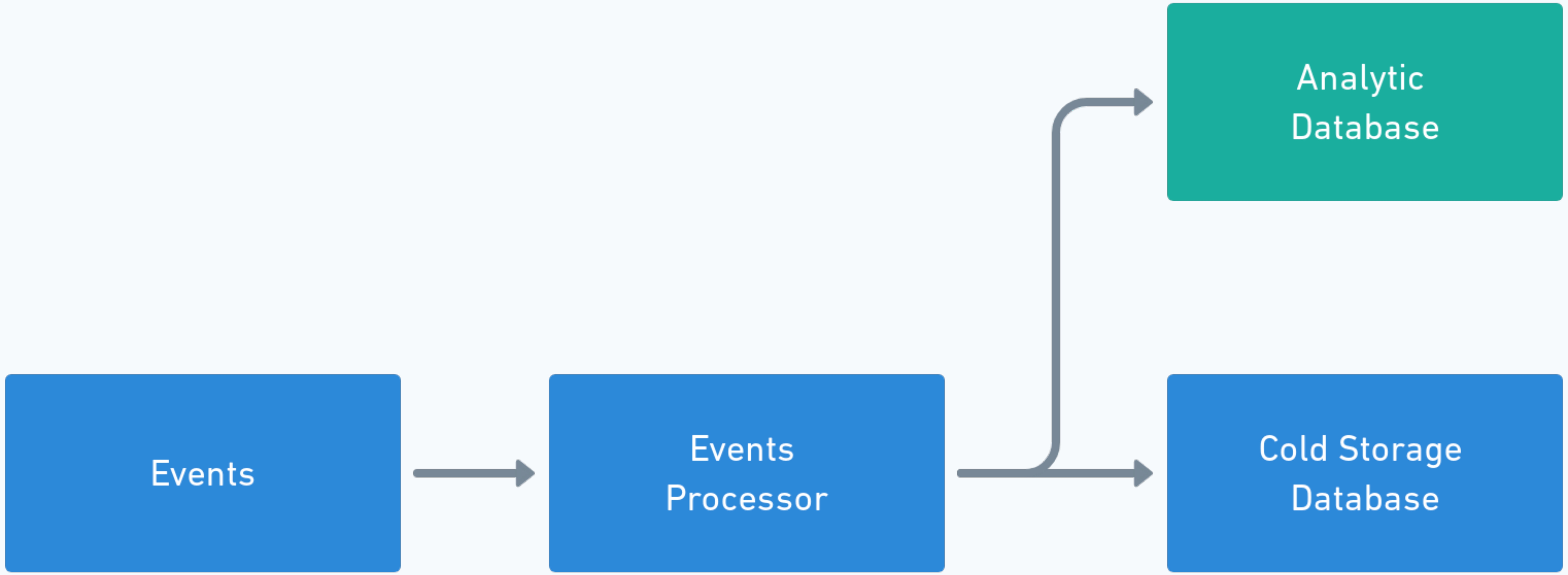
Introducing event sourcing

- Event sourcing, fairly common technique
- An immutable stream of events from which all states can be derivate


Immutable Events Stream
Idempotent Processor
Derivated States
Events Processor

Introducing event sourcing

- Problem: now we have increased our data by an order of magnitude.
- How can we query this efficiently?


Events
Events Processor
Analytic Database
Cold Storage Database

ENLIGHTENMENT - QUEST FOR AN ALTERNATIVE

What is our unicorn database?

- Able to ingest large amount of data
- Data available immediately after ingestion
- No loss of granularity
- Flexible querying capabilities
- Sub-second response time
- Horizontally scalable

ENLIGHTENMENT - QUEST FOR AN ALTERNATIVE

Vitess

- Let’s shard the data
- Now we have N shards of problems
- Still has the limitations of MySQL
- Poor analytical support (at the time)

ENLIGHTENMENT - QUEST FOR AN ALTERNATIVE

Kudu/Impala

- Promising, very clean and well defined SQL interface
- Compatible with HDFS & Parquets
- Column oriented
- But unable to reach sub-second querying time over billions of rows

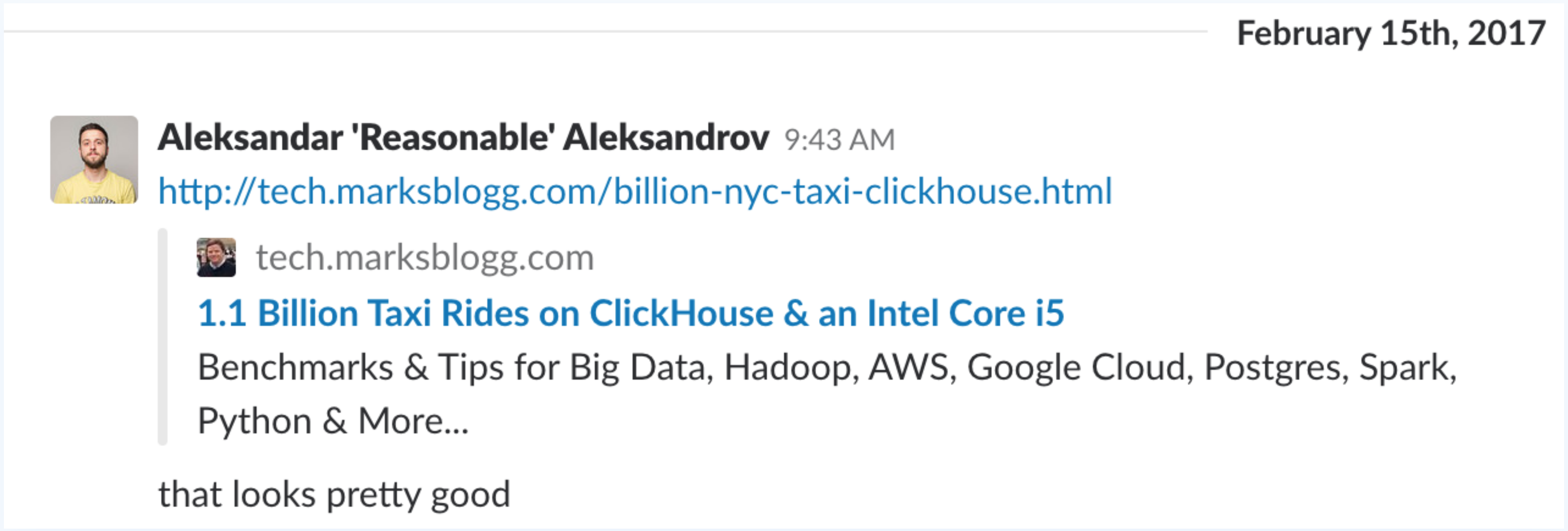
ENLIGHTENMENT - QUEST FOR AN ALTERNATIVE

Google BigQuery

- Scale well, millions or billions doesn’t matter
- Fully managed: it’s someone else problem
- Standard SQL support
- Not open source
- Not made for sub-second querying

ENLIGHTENMENT - CLICKHOUSE USE CASE

ClickHouse



February 15th, 2017

Aleksandar 'Reasonable' Aleksandrov 9:43 AM

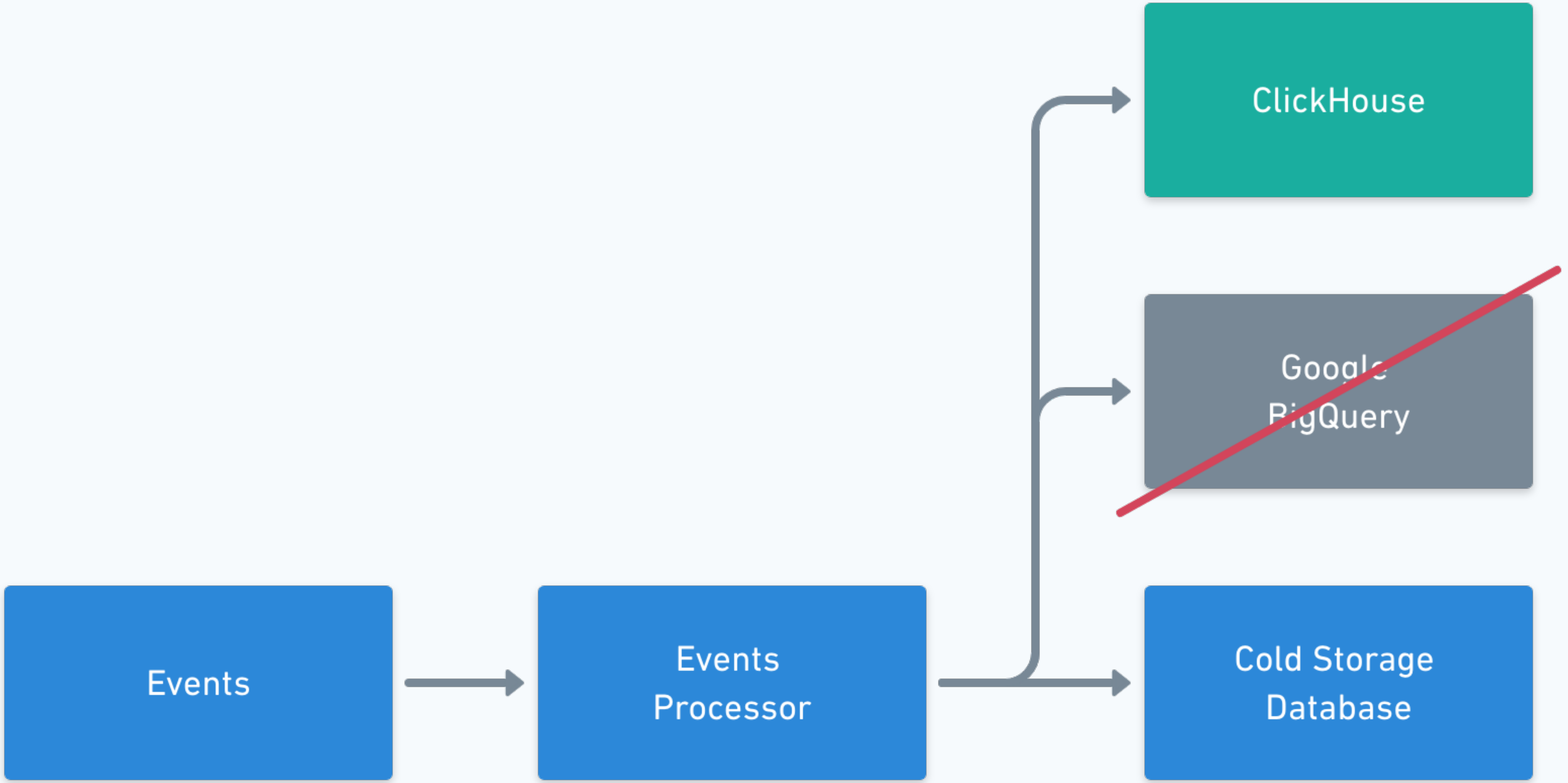
http://tech.marksblogg.com/billion-nyc-taxi-clickhouse.html

tech.marksblogg.com

1.1 Billion Taxi Rides on ClickHouse & an Intel Core i5

Benchmarks & Tips for Big Data, Hadoop, AWS, Google Cloud, Postgres, Spark, Python & More...

that looks pretty good


ClickHouse
~~Google BigQuery~~
Events
Events Processor
Cold Storage Database

ClickHouse

- Able to ingest a huge amount of data
- Sub-second on large dataset of non-aggregated data
- Flexible query capabilities: SQLish dialect
- Column oriented
- Scales **very** well vertically
- Horizontally scalable
- Open source

ClickHouse

```
SELECT
    toStartOfQuarter(created_at) AS Quarter,
    mcc                          AS Country,
    floor(sum(sign * rate))      AS Total,
    sum(sign)                    AS MessageCount
FROM messages
WHERE created_at >= '2018-01-01' AND customer = 666
GROUP BY Quarter, Country
```

30 rows in set.

Elapsed: **0.33sec.**

Processed 497.91 million rows,

4.95 GB

(1.42 billions rows/s., 14.39 GB/s.)

ClickHouse what’s the trick?

- Column oriented, you only pay for what you select
- Each column can potentially be processed in parallel
- Carefully crafted code makes use of vectorisation instructions
- Different table engines fit for different needs
- Horizontally scalable

So, how to ingest ever changing data into ClickHouse

CollapsingMergeTree

- You write twice the amount of data, but eventually end up with a single row per PK
- Based on the idea of log compaction
- Excels at analytical queries on a large amount of data

Primary key style

ENLIGHTENMENT - CLICKHOUSE USE CASE

Collapsing what?

sign	date	id	status	price
1	2018-10-08	666	ACCEPTED	0.01

Collapsing what?

sign	date	id	status	price
1	2018-10-08	666	ACCEPTED	0.01
-1	2018-10-08	666	ACCEPTED	0.01

Collapsing what?

sign	date	id	status	price
1	2018-10-08	666	ACCEPTED	0.01
-1	2018-10-08	666	ACCEPTED	0.01
1	2018-10-08	666	DELIVERED	0.05

Collapsing what?

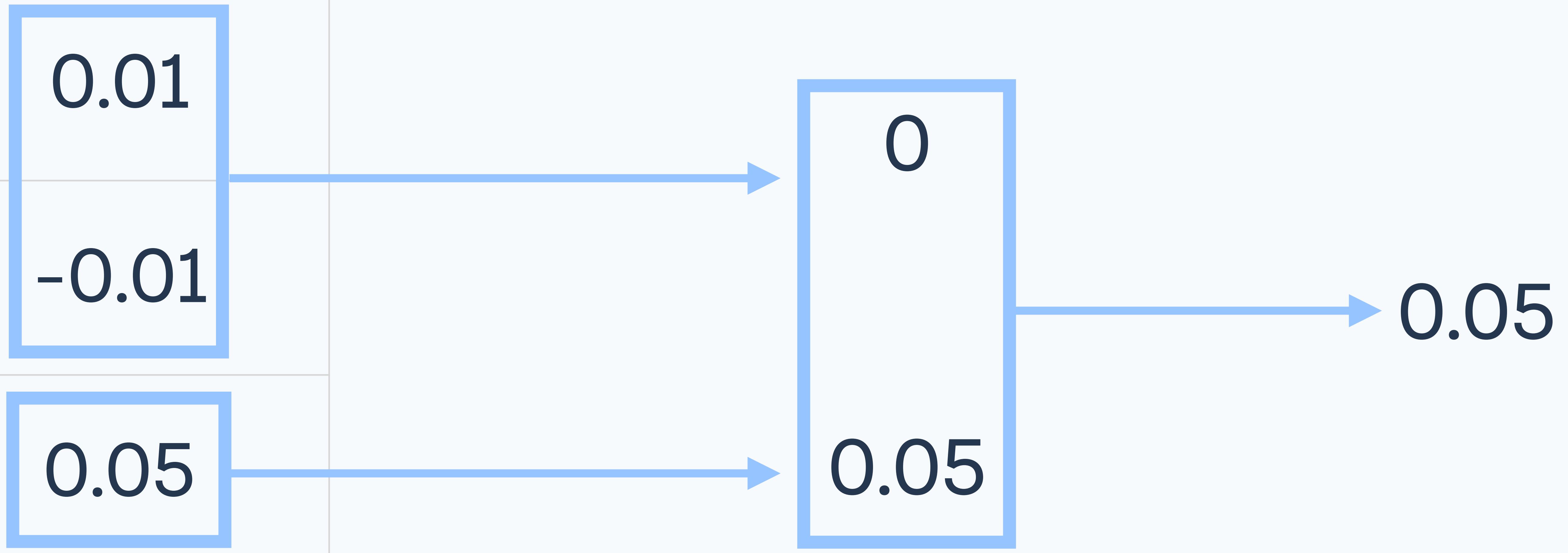
```
SELECT sum(sign * price) AS total FROM dataset
```

sign	date	id	status	price
1	2018-10-08	666	ACCEPTED	0.01
-1	2018-10-08	666	ACCEPTED	0.01
1	2018-10-08	666	DELIVERED	0.05

Collapsing what?

```
SELECT sum(sign * price) AS total FROM dataset
```

sign	price	sign * price
1	0.01	0.01
-1	0.01	-0.01
1	0.05	0.05



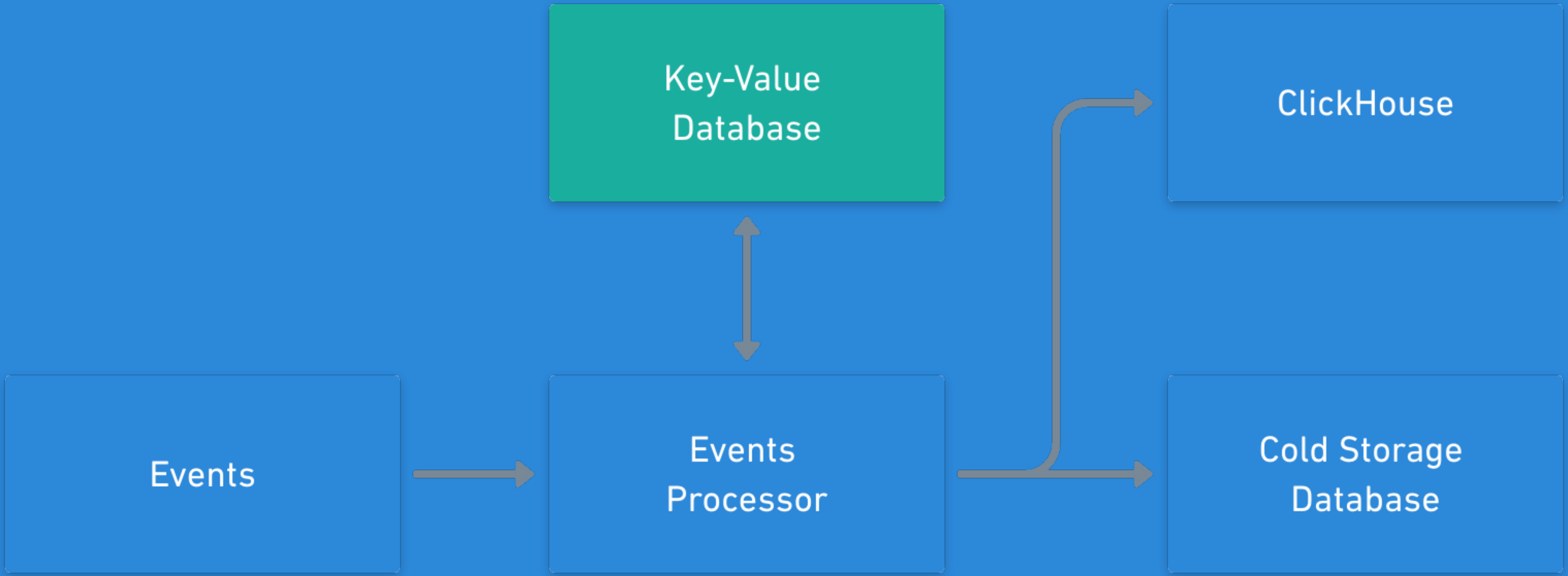
Collapsing what?

sign	date	id	status	price
~~1~~	~~2018-10-08~~	~~666~~	~~ACCEPTED~~	~~0.01~~
~~-1~~	~~2018-10-08~~	~~666~~	~~ACCEPTED~~	~~0.01~~
1	2018-10-08	666	DELIVERED	0.05

How to insert the proper “negative” row?

CollapsingMergeTree, keeping track of states

- Need to be aware of the previous row to properly negate it
- ClickHouse is not made for random access of single rows


Key-Value Database
ClickHouse
Events
Events Processor
Cold Storage Database

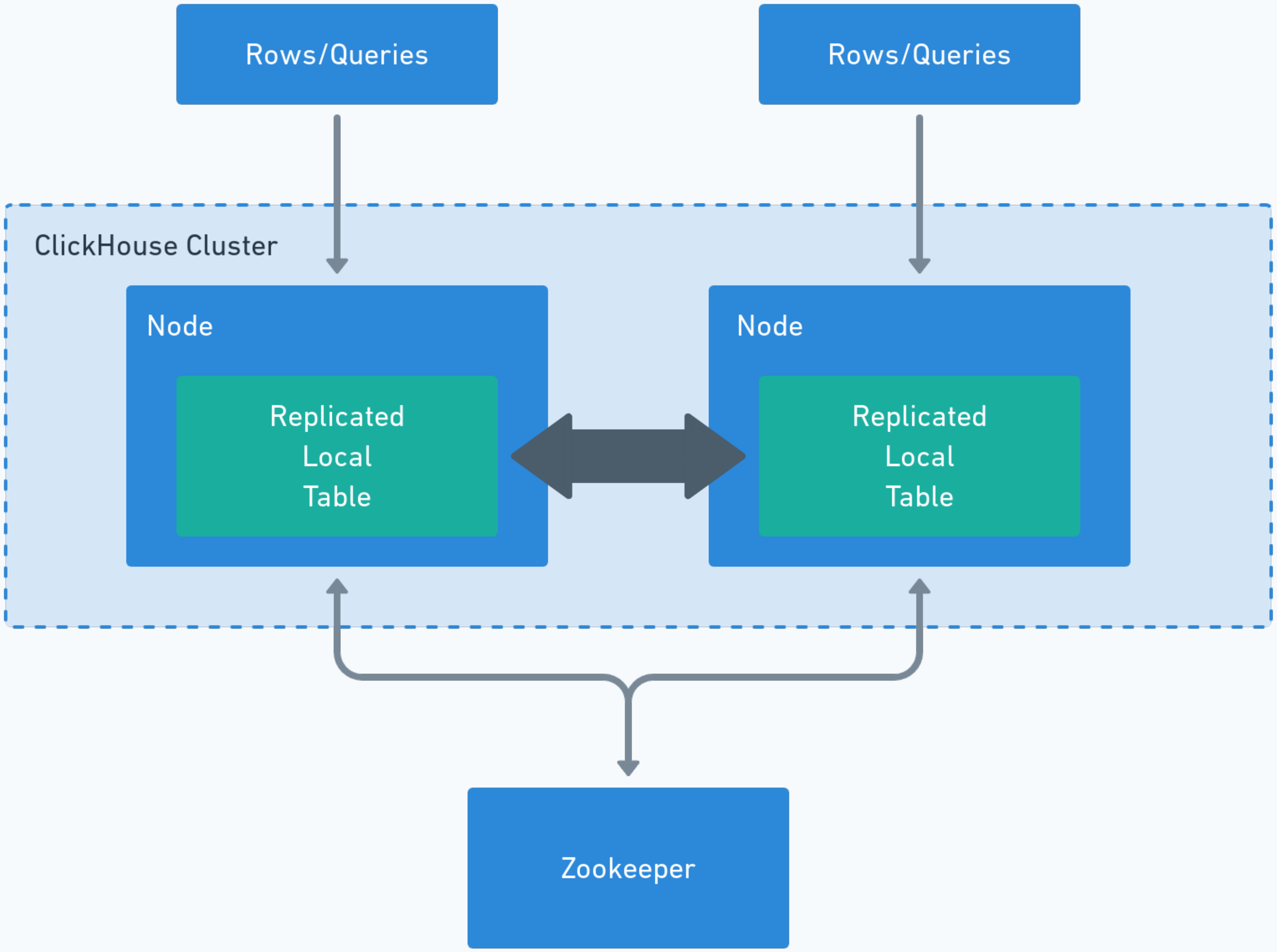
What about availability?

Replication

- High availability and reliability
- To bring data closer to consumer
- More than one way to do it with ClickHouse

ReplicatedMergeTree*

- Is supported by the MergeTree table family
  - ReplicatedCollapsingMergeTree
  - ReplicatedAggregatingMergeTree
- Uses Zookeeper to coordinate the replication between nodes


Rows/Queries
Rows/Queries
ClickHouse Cluster
Node
Replicated Local Table
Node
Replicated Local Table
Zookeeper

ClickHouse scalability?

Horizontal scalability

- Distributed engine
  - Dispatch read queries to all the nodes
  - Shard the data and dispatch it to the right node
- Flexible sharding capabilities
  - Let ClickHouse do the work
  - Shard manually: inserting directly into the wanted node and only use the distributed engine to dispatch read queries

Vertical scalability

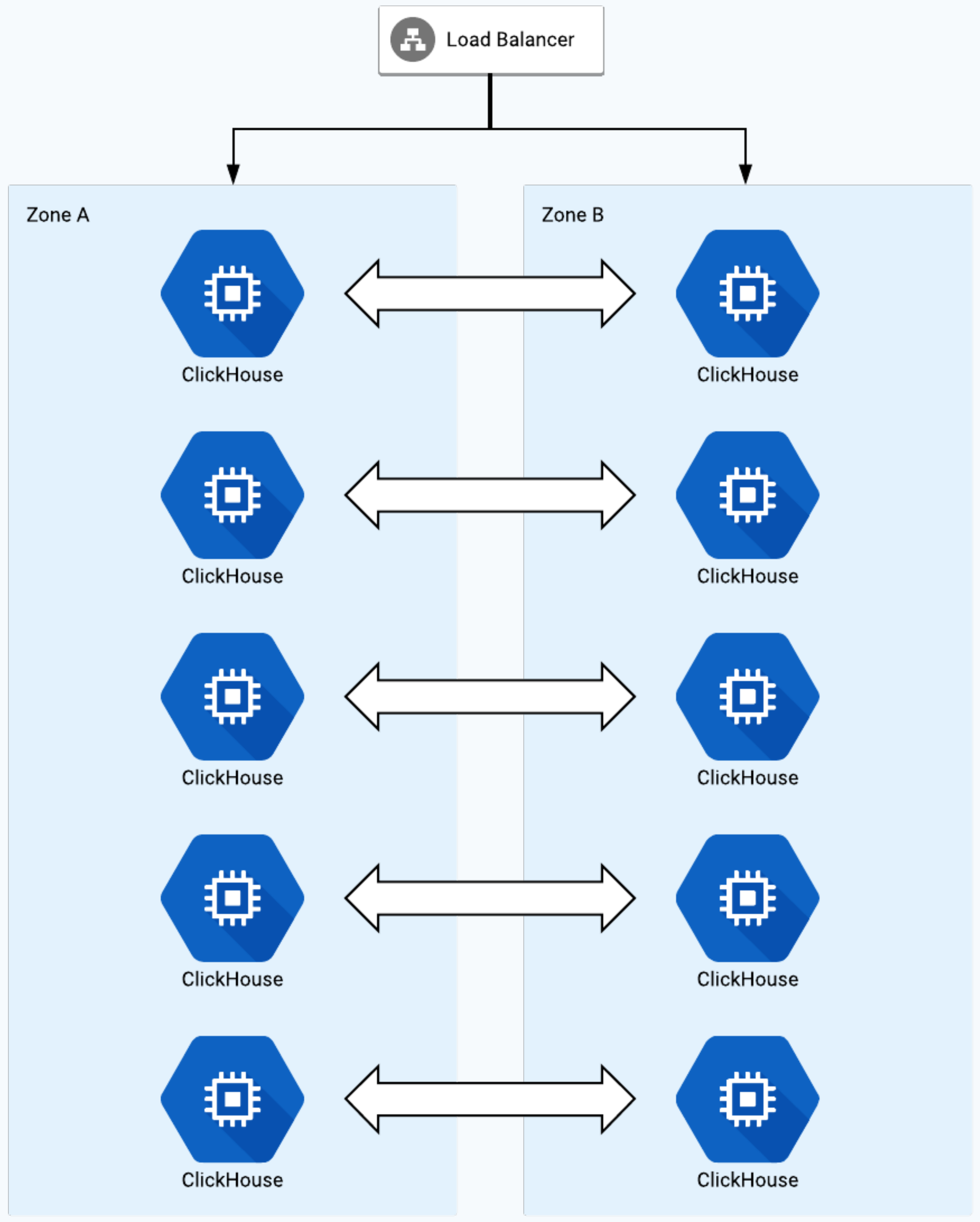
- Very efficient use of available CPU
- Data is on one machine (or even in memory) makes queries even faster
- You don’t care about sharding of the data, operations can be done on local table
- Generally accepted to have more CPU, rather than more servers

ClickHouse in production?

ENLIGHTENMENT - CLICKHOUSE USE CASE

Our setup

- Single region
- Two availability zones
- 8 CPU/30 GB RAM
- 2TB+ compressed
- 10 nodes
- Replica factor 2



How far ClickHouse took us

- Between the moment we designed and implemented our fist data pipeline with ClickHouse from an average of 1000 events/s to 10000+ event/s without having to scale the cluster.
- Most of MessageBird products' data is in ClickHouse

Rough edges

Don’t forget it’s not a RDBMS

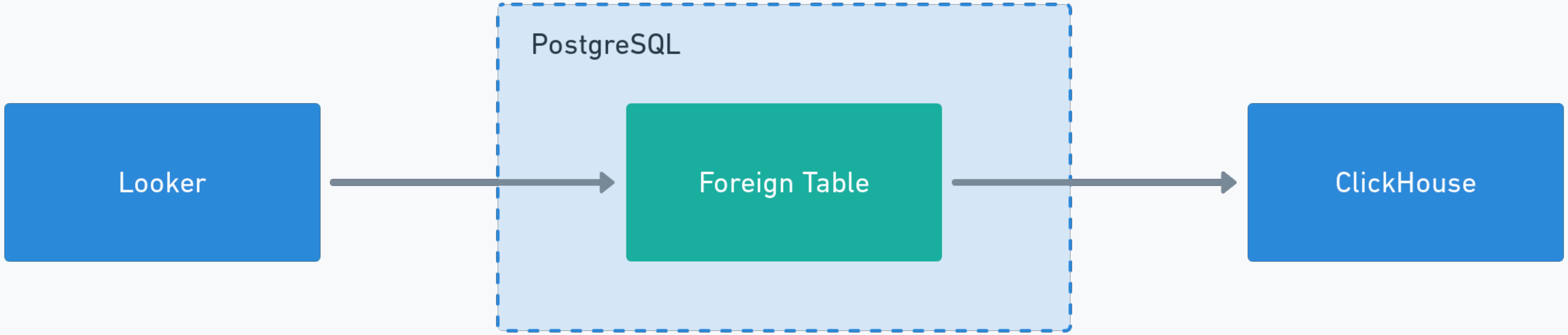
- Eventual consistency *
- No transactions
- A single non unique primary index
- Limited support of JOIN
- Experimental features are experimental FOR REAL that stuff will break
- Resharding isn’t out-of-the-box
- Not made for deleting/updating random rows

ClickHouse among many

- ClickHouse is still one among many
- Dictionaries: periodically refreshed view of external databases
- JDBC/OBDC drivers, remote/local file, custom executable
- Non standard SQL can make third party like business intelligence tools integration can be challenging

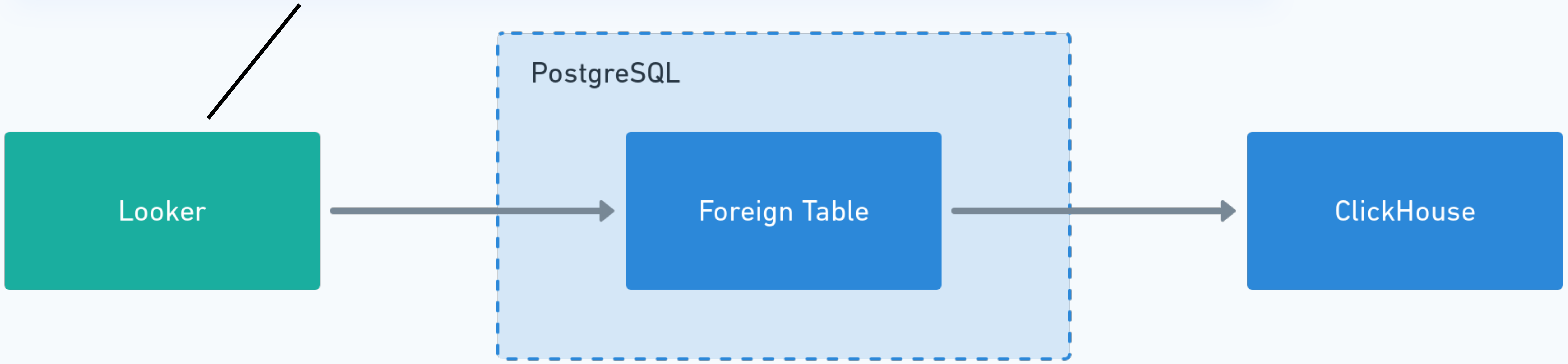
ENLIGHTENMENT - CLICKHOUSE USE CASE

PostgreSQL + ClickHouse



Query forwarding

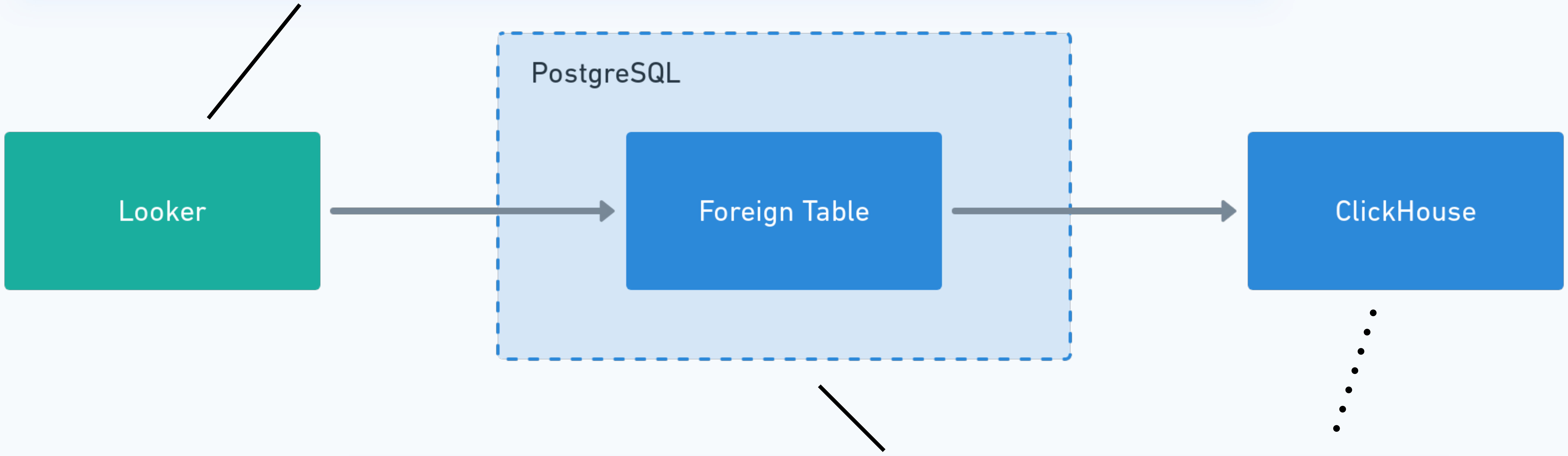
```sql
SELECT sum(sign * price) AS total FROM dataset WHERE dataset = 666
```



```sql
SELECT sign, price AS total FROM dataset
```

Query forwarding and push down

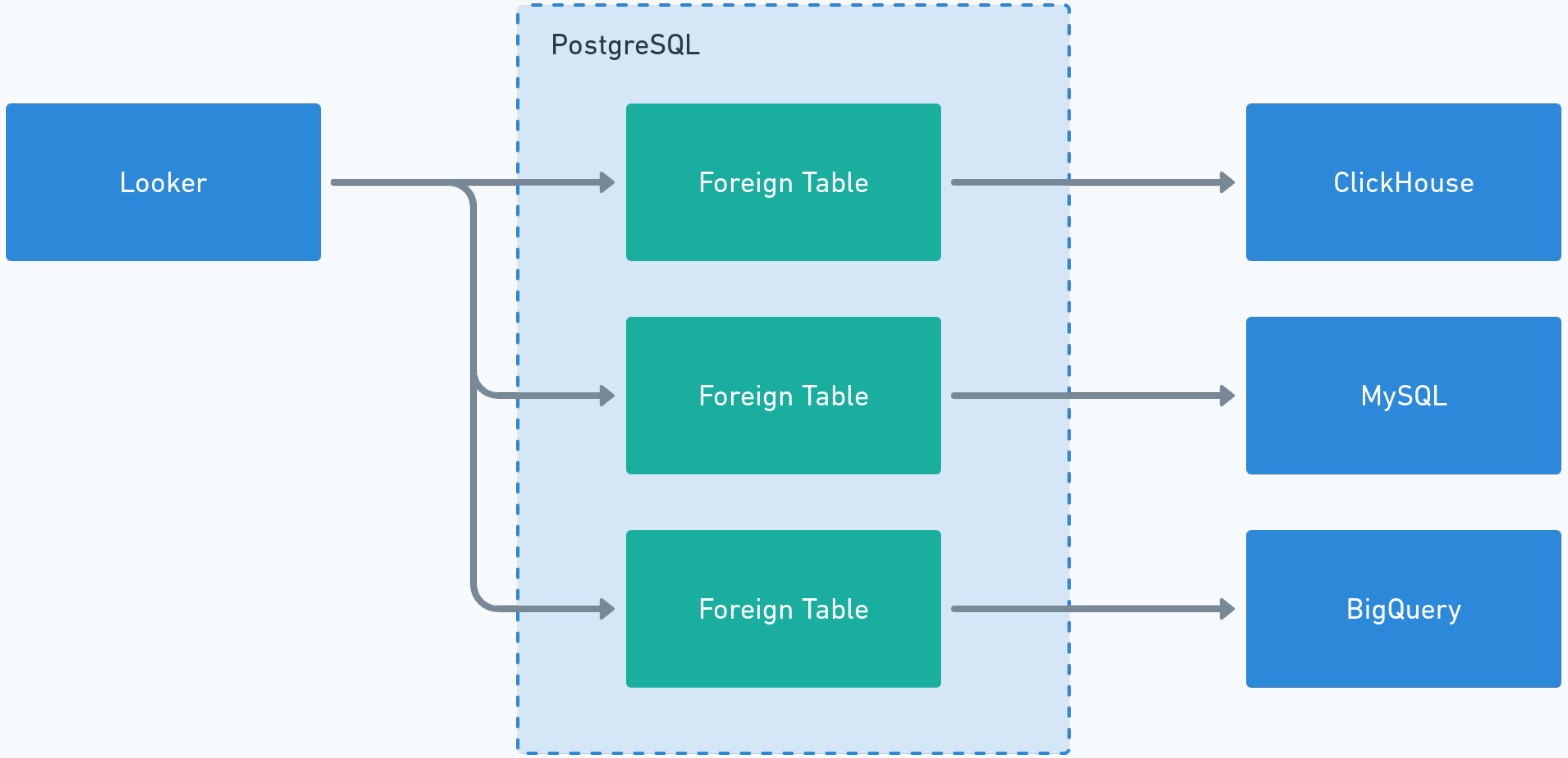
```sql
SELECT sum(sign * price) AS total FROM dataset WHERE customer = 666
```



```sql
SELECT sign, price AS total FROM dataset WHERE customer = 666
```

PostgreSQL + ClickHouse, looping the loop

- Instantly gain to one of the most standard SQL interface
- Still leverage the most important feature of ClickHouse by pushing down the filters and aggregations
- Bastion like approach to share data with third-party BI tools


PostgreSQL
Looker
Foreign Table
Foreign Table
Foreign Table
ClickHouse
MySQL
BigQuery

PostgreSQL + ClickHouse, looping the loop

- Almost out-of-the-box data federation
- But only a PoC, we are still dreaming of production

Did we say we are hiring?

FUTURE - WRAPPING WORDS

Even more possibilities

- ML features with catboost
- Kafka base table engine
- Upcoming better JOIN supports
- Cap’n Proto and upcoming Protobuf / Parquet support

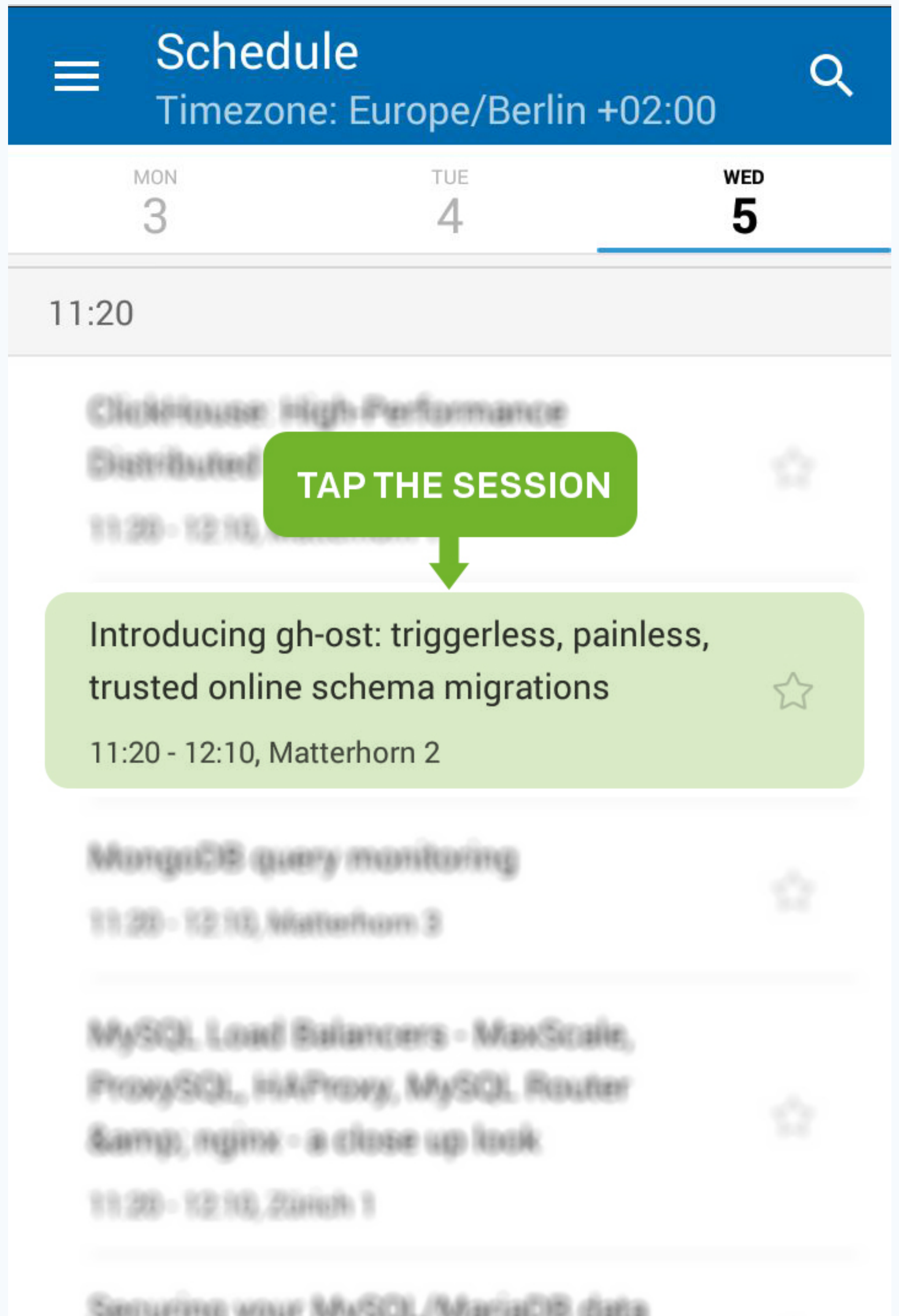
Questions

Late questions? Come say hello or drop us an email.

aleksandar@messagebird.com & felix@messagebird.com

www.messagebird.com/careers

Rate the session

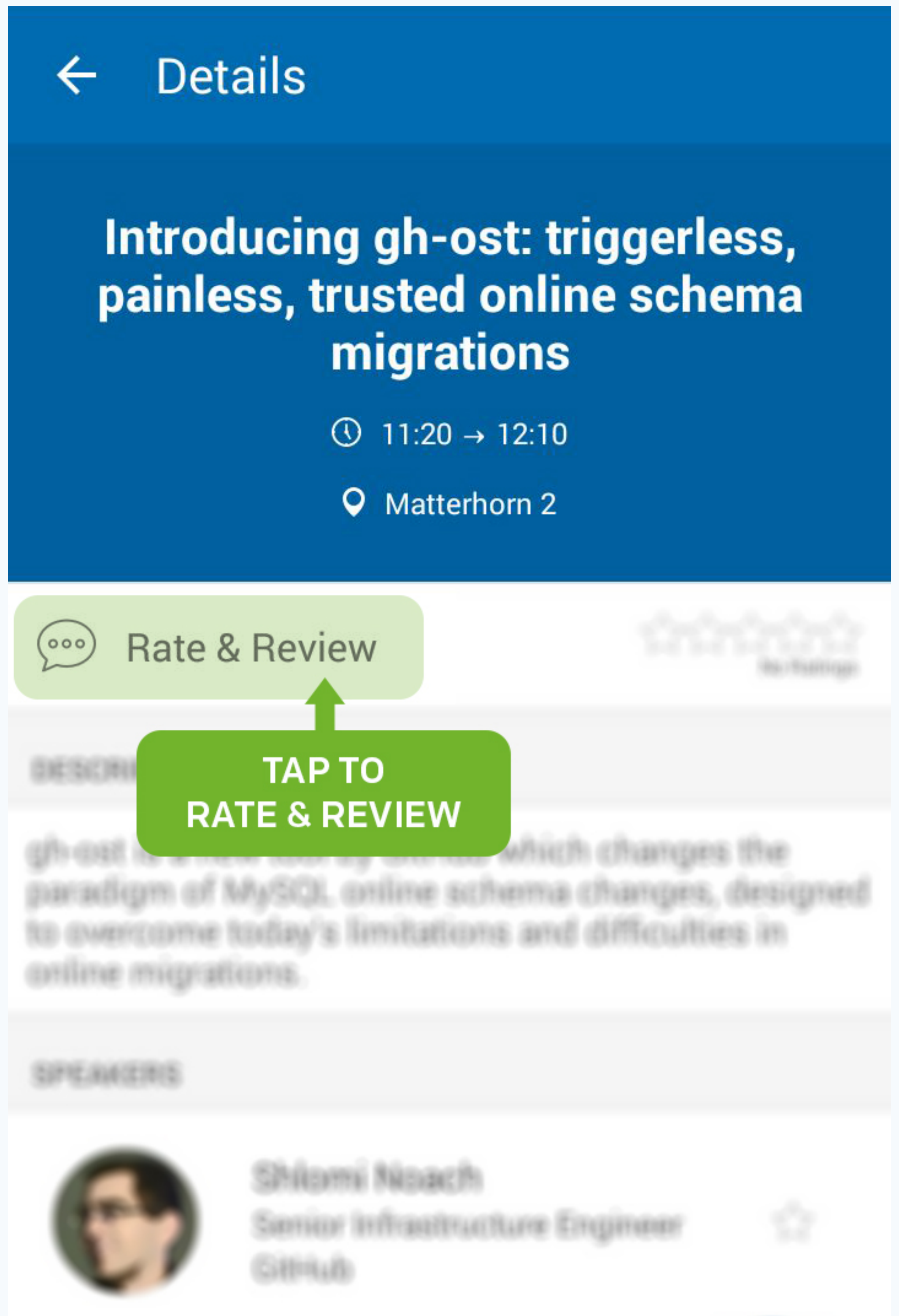


Schedule
Timezone: Europe/Berlin +02:00

MON 3 · TUE 4 · WED 5

11:20

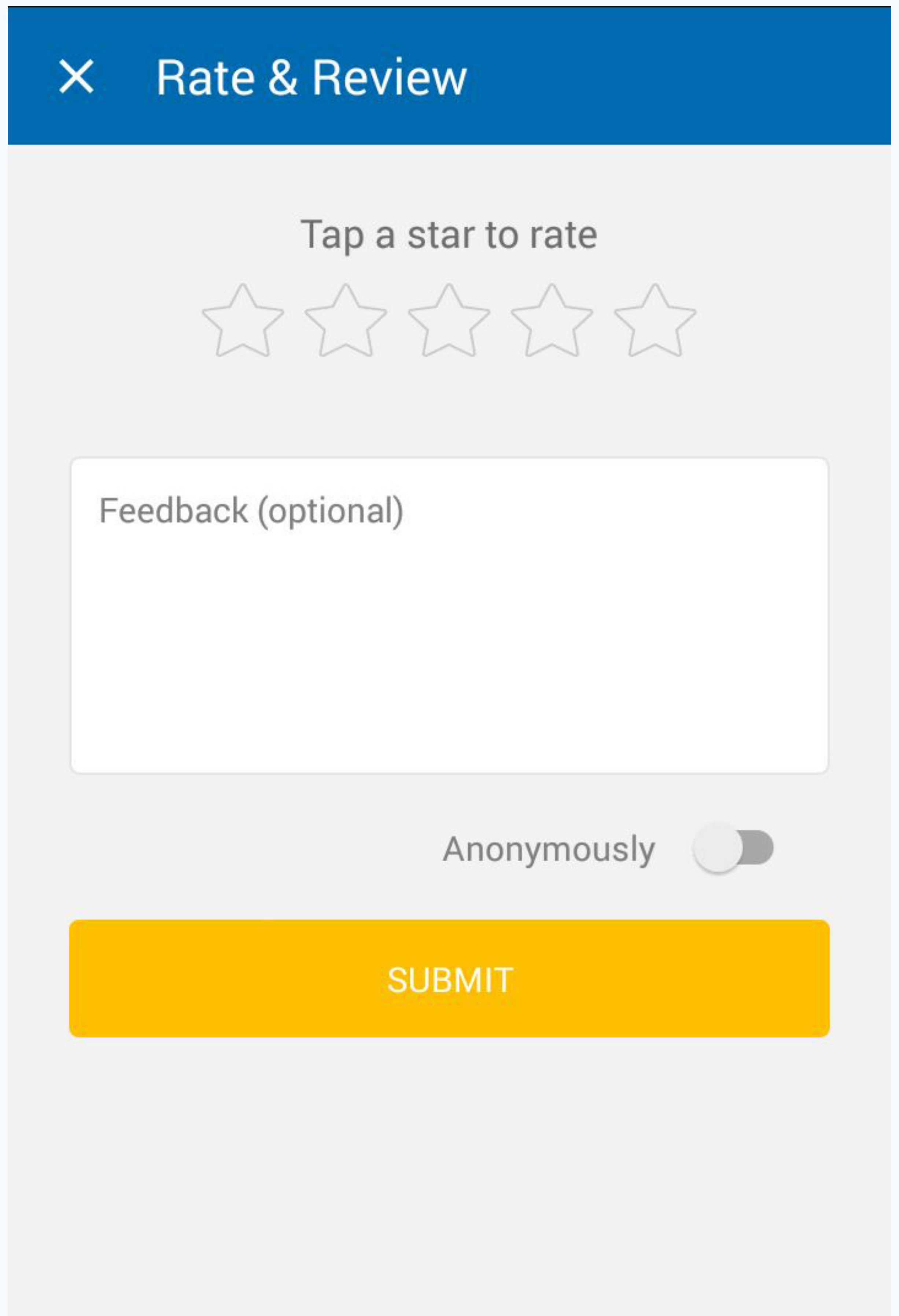
Introducing gh-ost: triggerless, painless, trusted online schema migrations
11:20 - 12:10, Matterhorn 2



Details

Introducing gh-ost: triggerless, painless, trusted online schema migrations
11:20 → 12:10
Matterhorn 2

Rate & Review



Rate & Review

Tap a star to rate

Feedback (optional)

Anonymously

SUBMIT