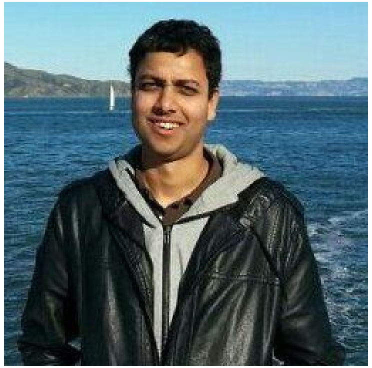
# Near Real-Time Analytics

## Challenges & Lessons at Uber Engineering

UBER

# Quick Introduction



Chinmay Soman

@ChinmaySoman

- Staff Software Engineer @ Uber
- Tech Lead on Streaming Platform
- Background in distributed storage and filesystems
- Apache Samza Committer, PMC

# Apache Kafka at Uber

**Billion to Trillions**
Messages/day

**~ PB**
bytes/day

# Near Real-Time Analytics at Uber

**Billions**
Messages Processed / day

**100s of TB - PB**
Bytes Processed / day

# What is near real-time ?

# Agenda

- Evolution of Business Needs
- The case for SQL as building block
- New ecosystem using Flink
- The road ahead

# Evolution of Business Needs

# Case I - Growth Metrics

“How many cars are active right now ?”

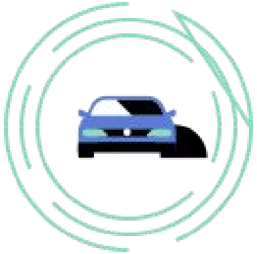
“What % of trips have been delayed in the last 5 mins ?”

“What is the % of Uber X trips taken by Android users ?”

UBER

# Events logged to Kafka

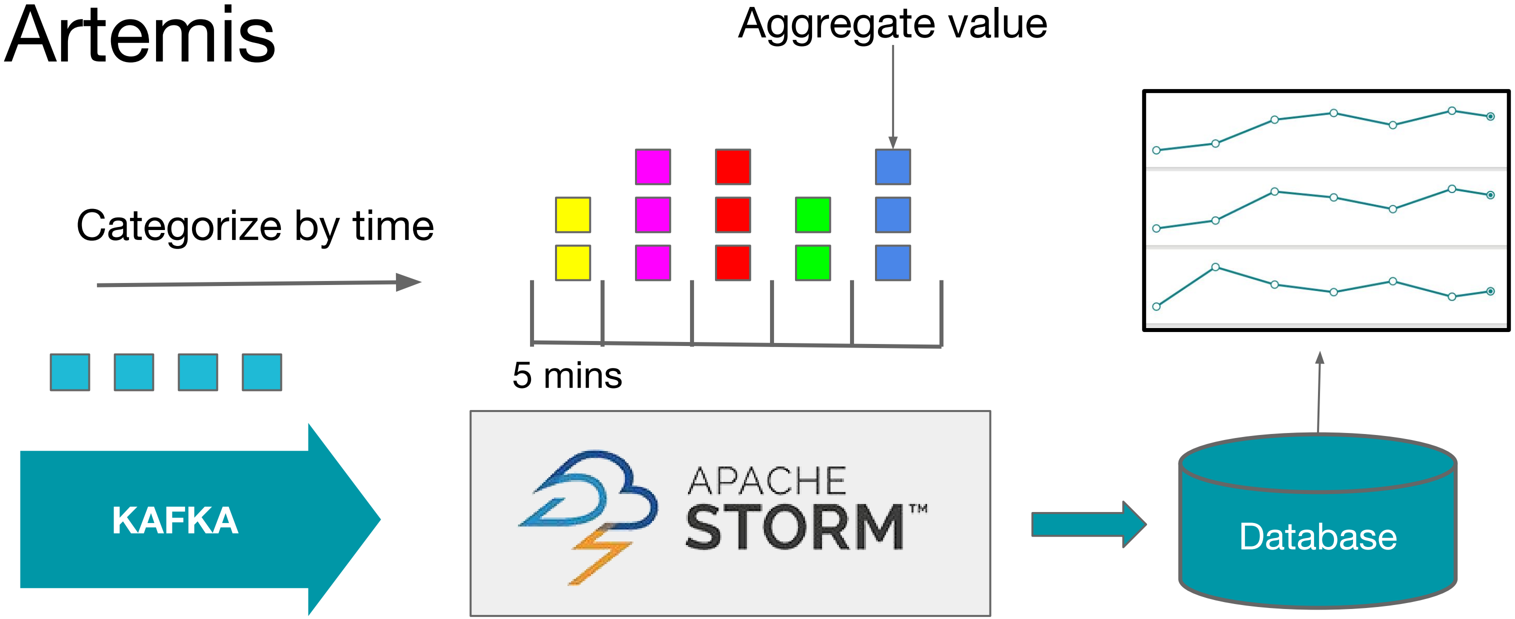
Rider eyeballs



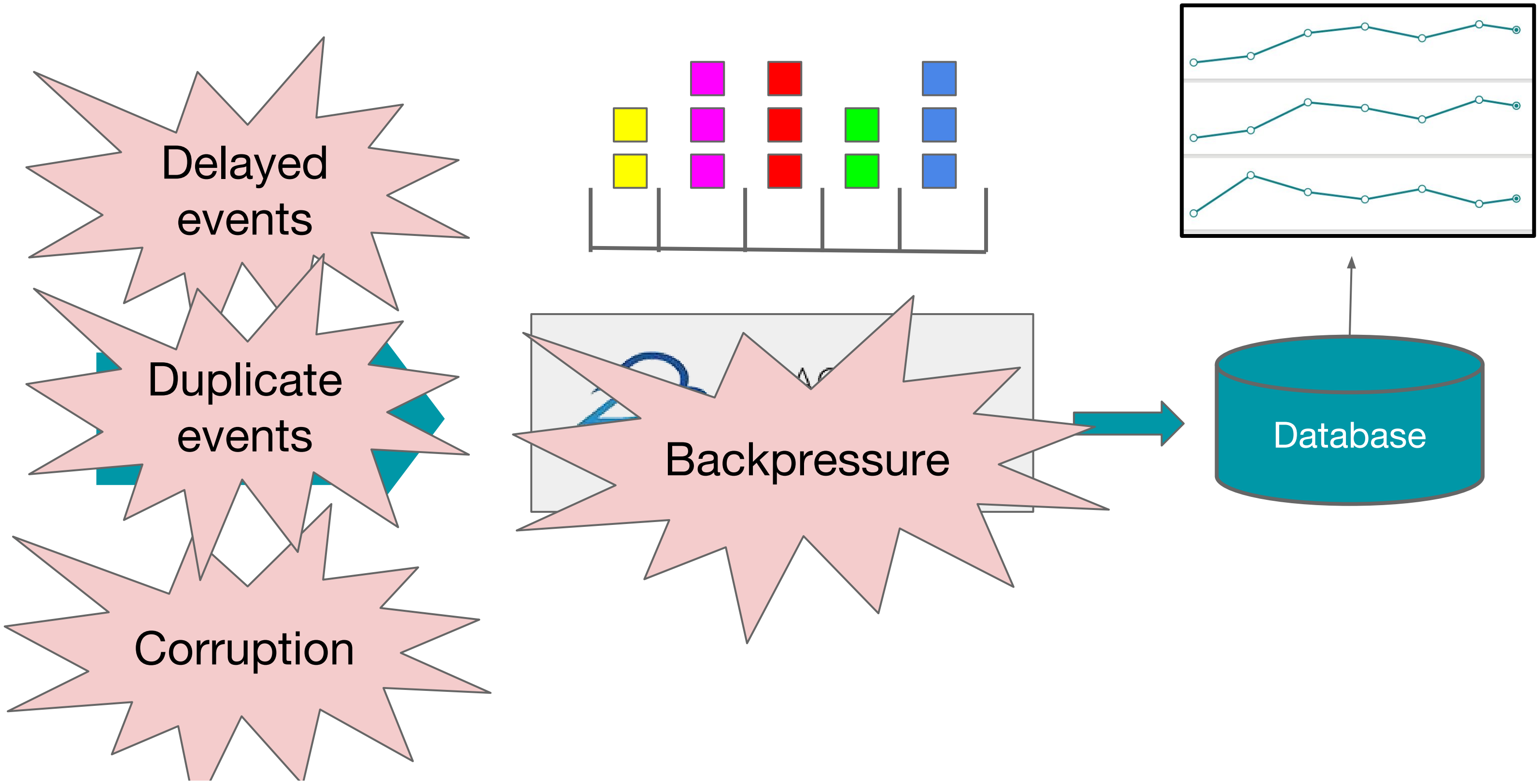
Trip updates

kafka

# Artemis

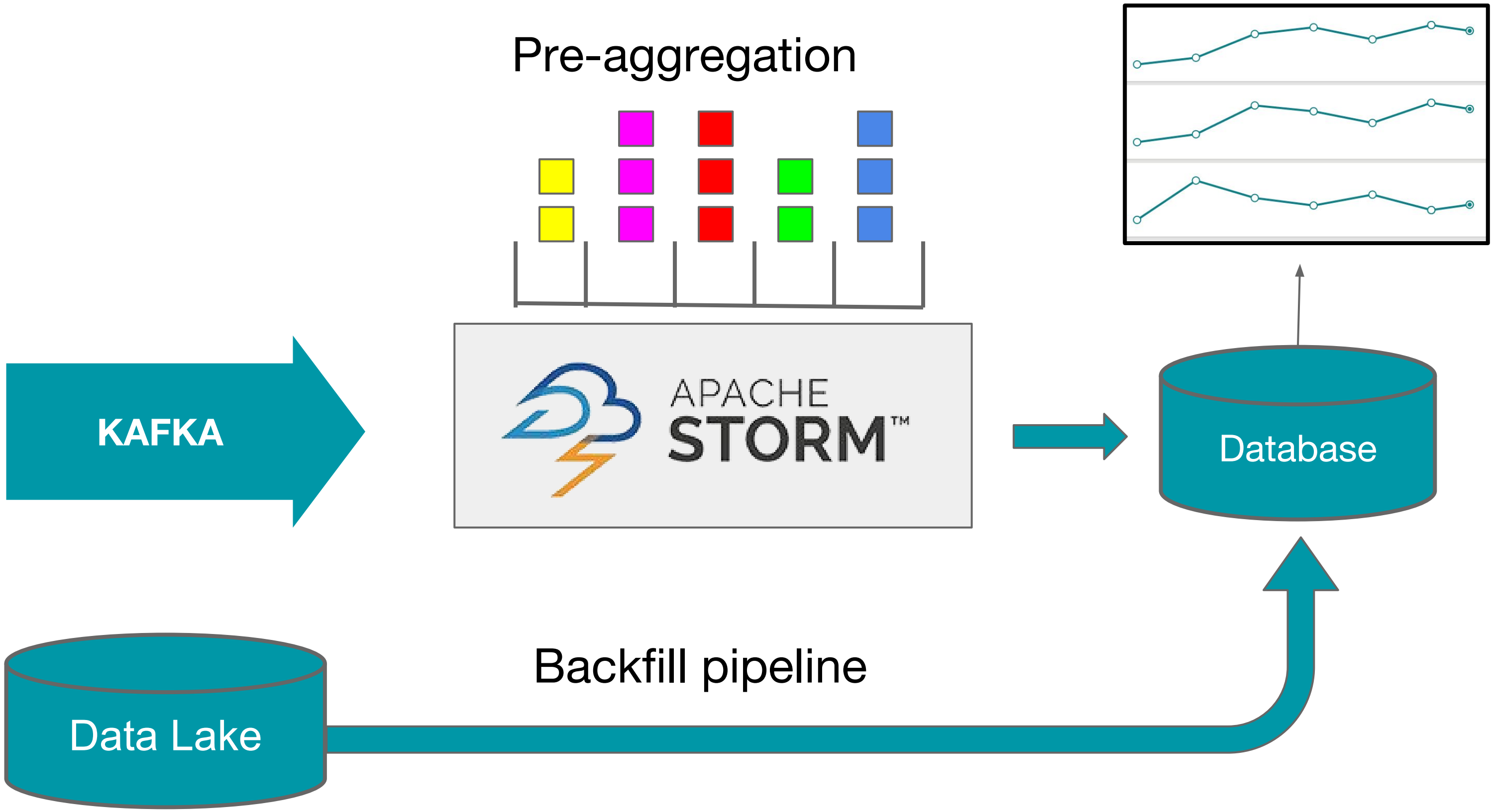

Aggregate value
Categorize by time
5 mins
KAFKA
APACHE
STORM™
Database

# Artemis



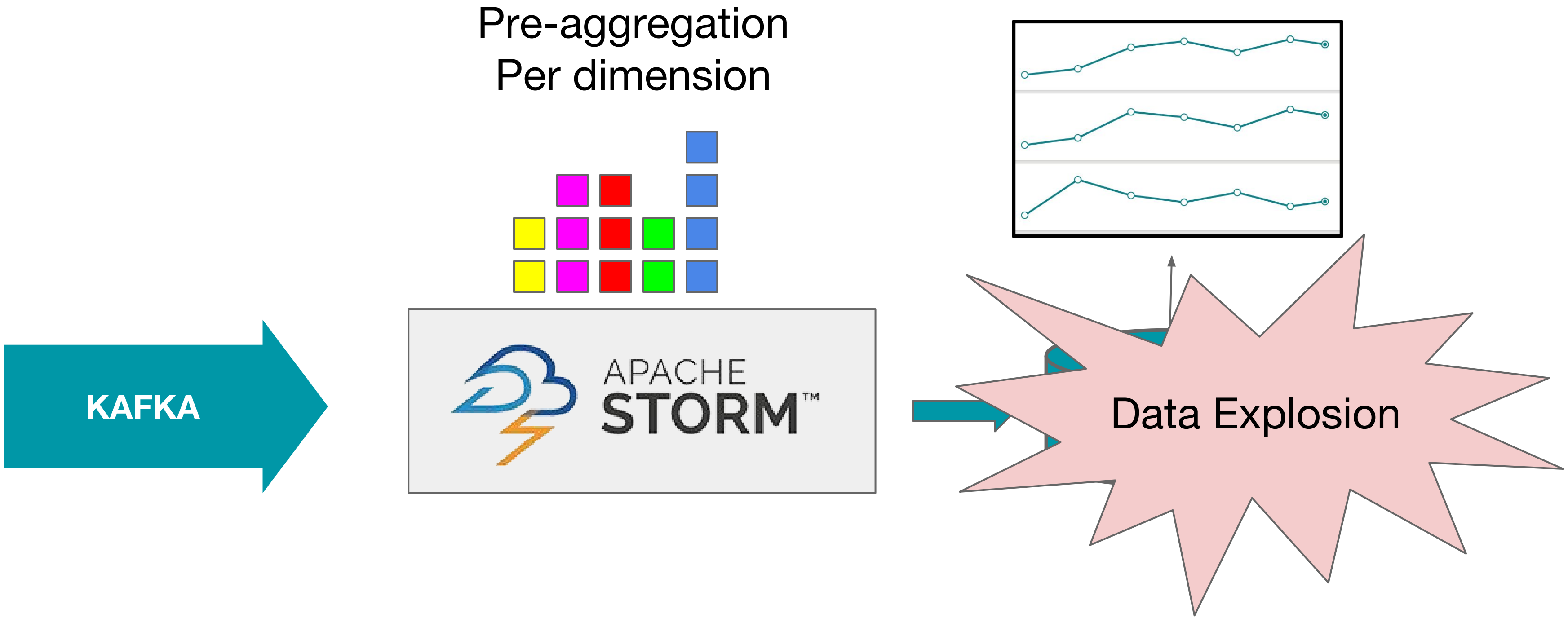
Delayed events
Duplicate events
Corruption
Backpressure
Database

# Artemis



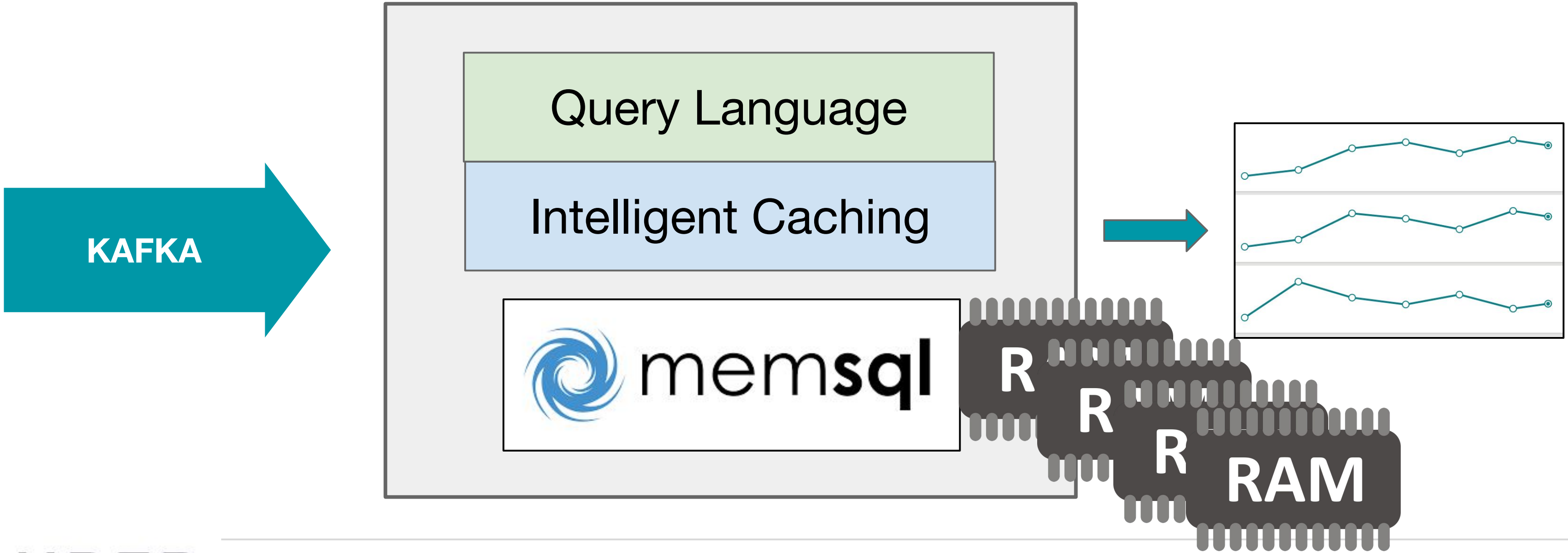
Pre-aggregation
KAFKA
APACHE
STORM™
Database
Data Lake
Backfill pipeline

# Artemis

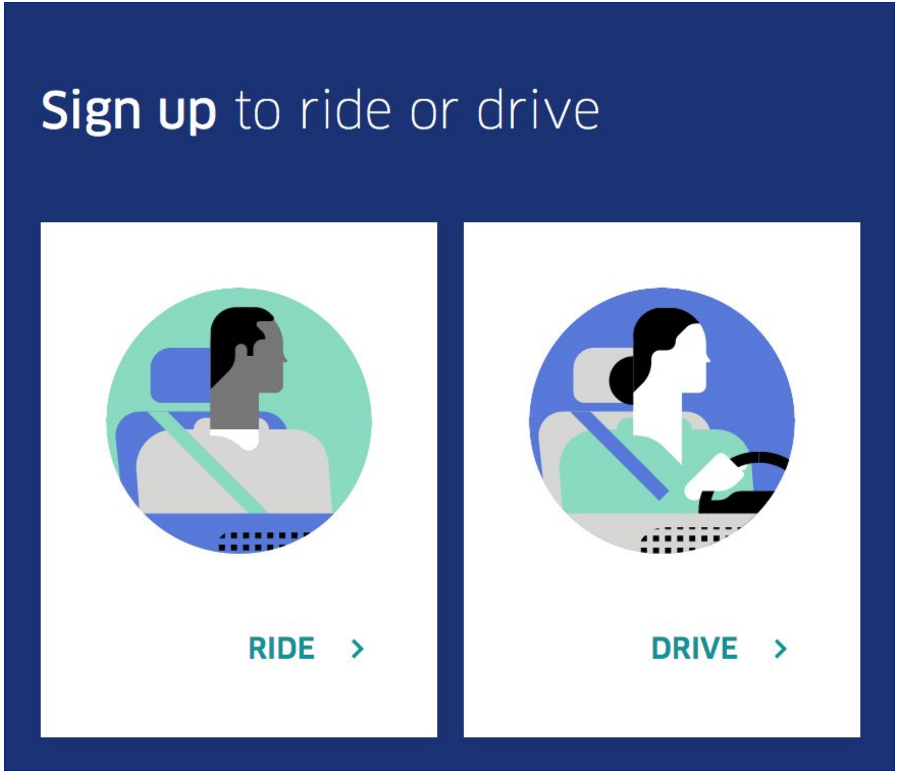

Pre-aggregation
Per dimension
KAFKA
APACHE
STORM™
Data Explosion

# Apollo

✓ Fast ✓ Accurate ✓ Scalable



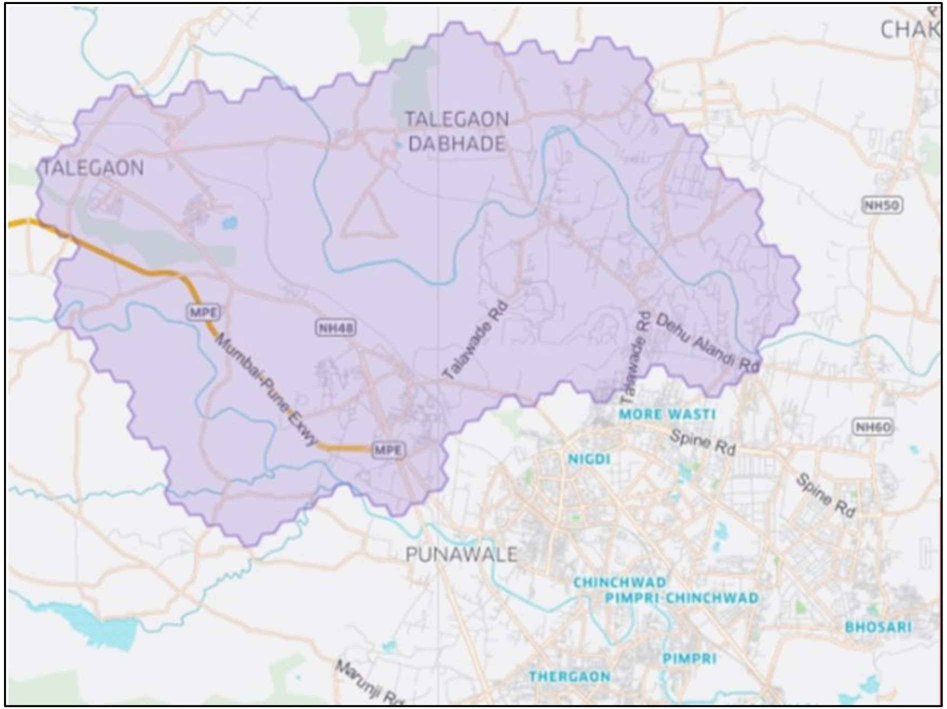
# Case II - Event processing




FRAUD

“If # Signups per device look suspicious -> Ban the driver/rider”

# Case II - Event processing




## INTELLIGENT ALERTS

"Send me an alert if a leased vehicle leaves a geo-fence"

# Athena platform using Apache Samza

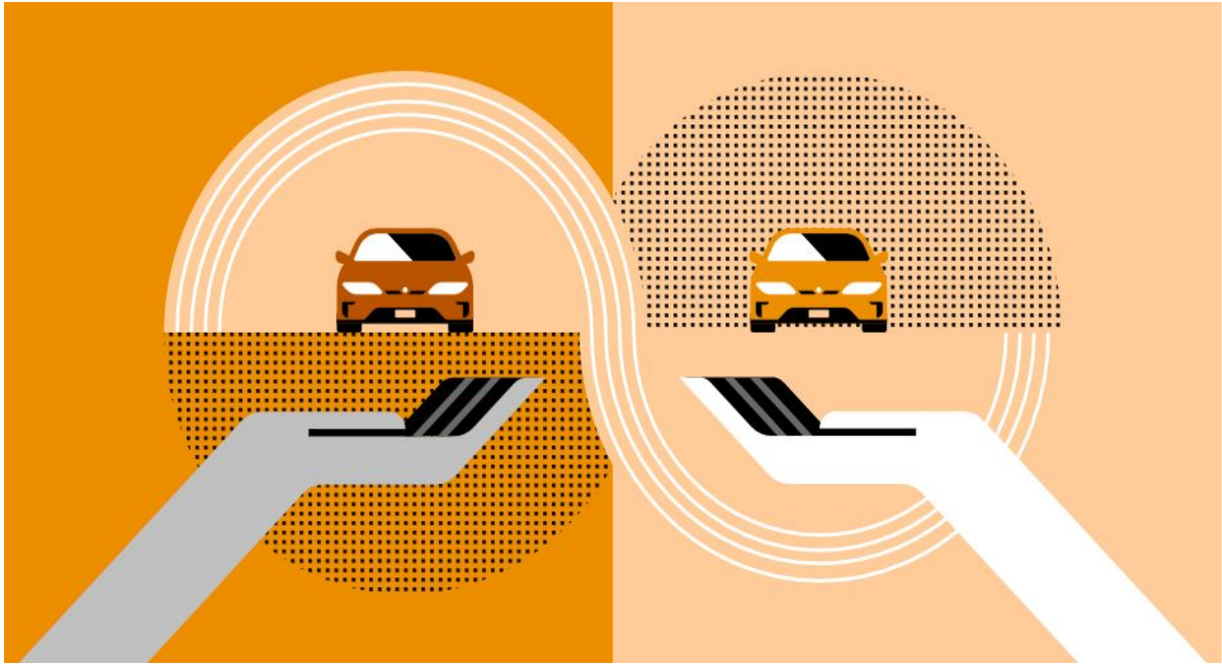
- Robust
- Ease of operation
- No backpressure issues
- Built in state management

# Event processing - Apache Samza



Track count # of sign_ups
categorized by device_imei

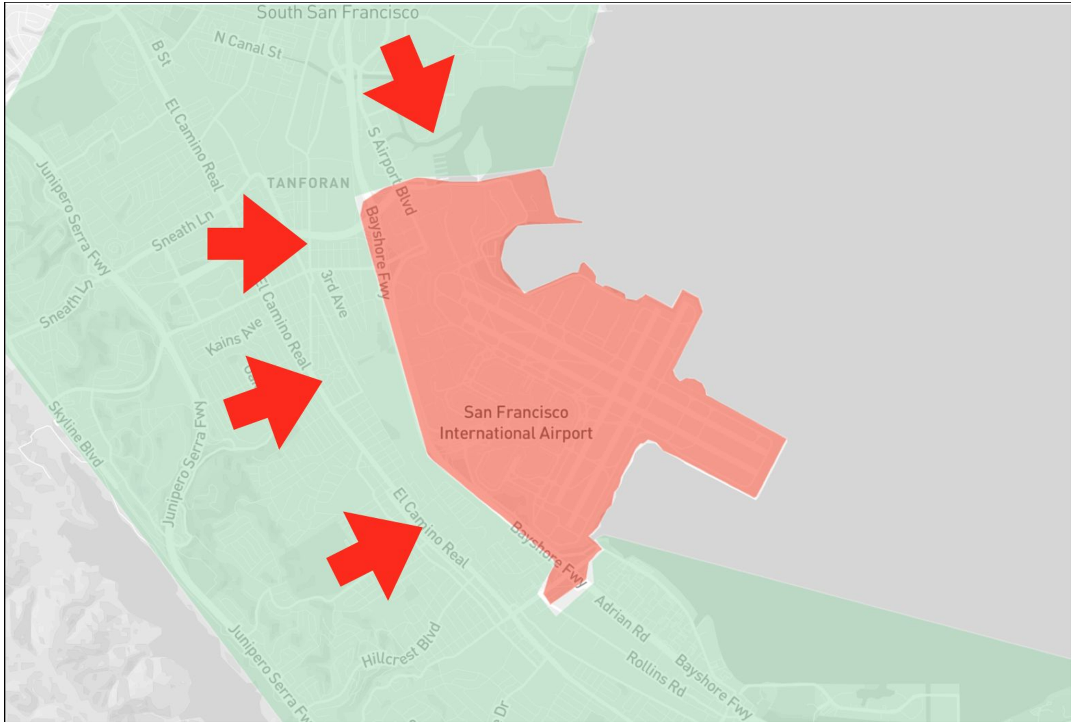
# Case III - OLAP (OnLine Analytical Processing)



A / B Tests

See progress of tests in real-time

# Case III - OLAP use case




FORECASTING

"How many first time riders will be dropped off in a given geofence ?"

UBER

# Our integrated platform

samza

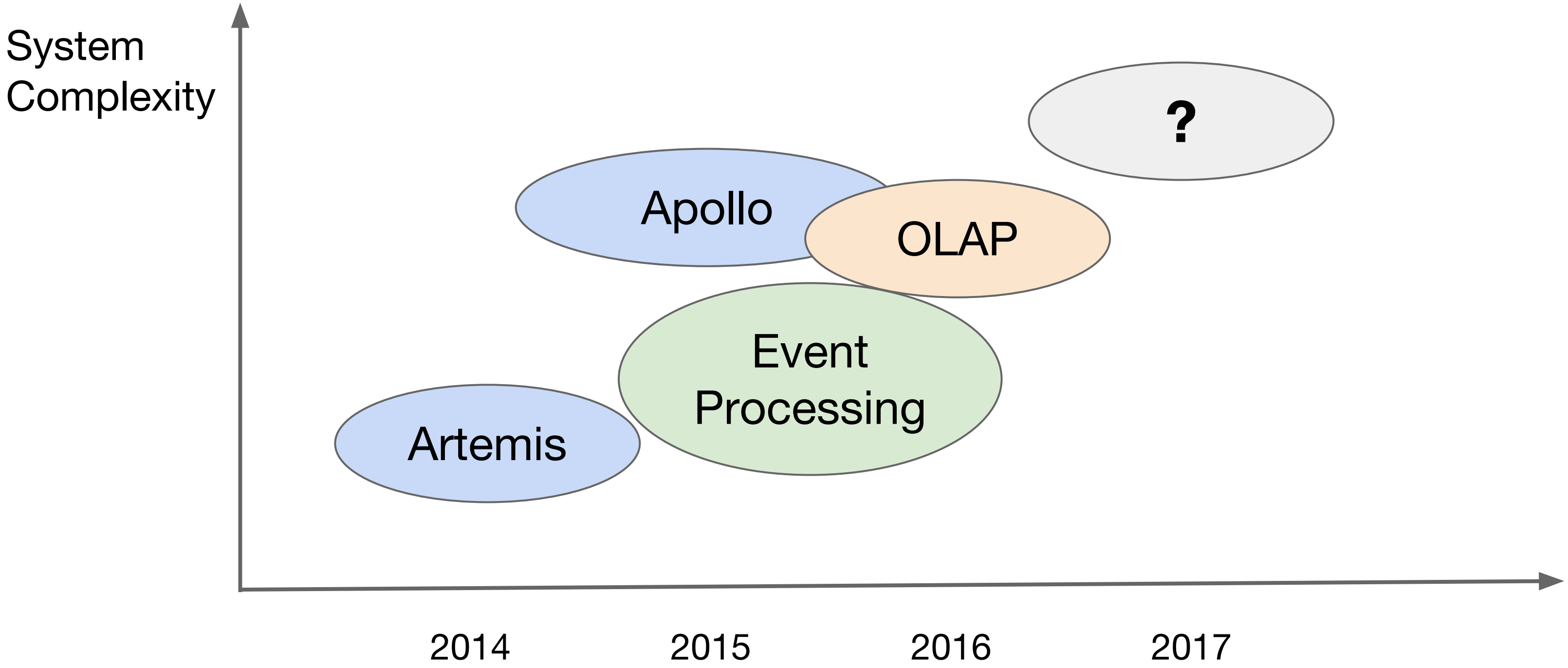
- Filter events
- Merge streams
- Decorate with external data

pinot

elastic

memsql

# Are we there yet ?



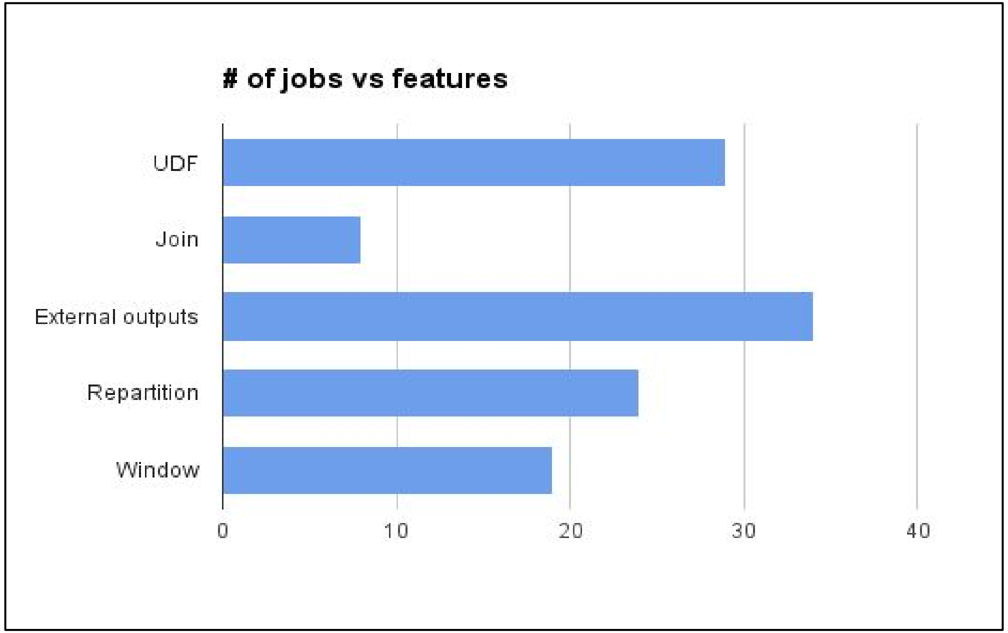

UBER

# What's missing ?

- Cumbersome for data scientists / Ops people
- Redundant code
- Custom backfill pipelines

# SQL as the building block

# SQL + Stream Processing




70-80% of jobs can be implemented via SQL

# SQL + Stream Processing: Powerful abstraction

## Intelligent Promotions

| Rule | Threshold | Action |
| --- | --- | --- |
| "All trips worth > 10$ in San Francisco between Friday 5 pm and Sunday 9 pm | > 100 | "Give bonus of $500" |

UBER

# SQL + Stream Processing: Powerful abstraction

**Complicated rules**

- "If number of hours online > 10 …"
- "If amount earned > 700 in a given week, then …"
- "If # uberPOOL rides >10, then …"
- "If trip happens over some geo-fence 10 times in a given weekend, then …"

# SQL + Stream Processing: Powerful abstraction

**Intelligent Promotions**

| Rule | Threshold | Action |
| --- | --- | --- |
| select count(*) from hp_api_created_trips WHERE city_id = 1 AND fare > 10 AND request_at > 1491105600 AND request_at <= 1491177600 | > 100 | trigger_payment() |

# SQL + Stream Processing: Powerful abstraction

**Complicated rules**

- "If number of hours online > 10 …"
- "If amount earned > 700 in a given week, then …"
- "If # Uber Pool rides >10, then …"
- "If trip happens over some geo-fence 10 times in a given weekend, then …"

What if we created specific rules for specific driver partners ?

# SQL + Stream Processing: Powerful abstraction

**Can be used for alerts as well:**

"If a driver X is outside a geofence, then …"

# New eco-system: Athena X

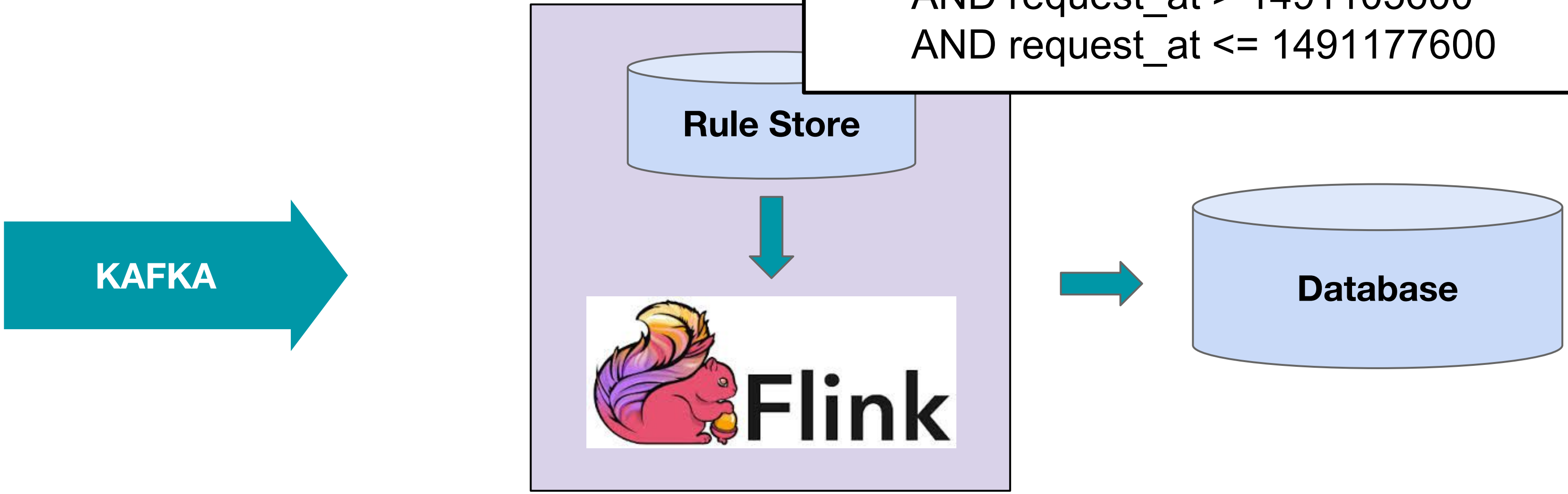
# Enter Flink

Flink

- ★ Apache Calcite (SQL) Integration
- ★ Easy to manage and scale
- ★ No backpressure problem
- ★ Built in state management support
- ★ HDFS integration
- ★ Not dependent on Kafka

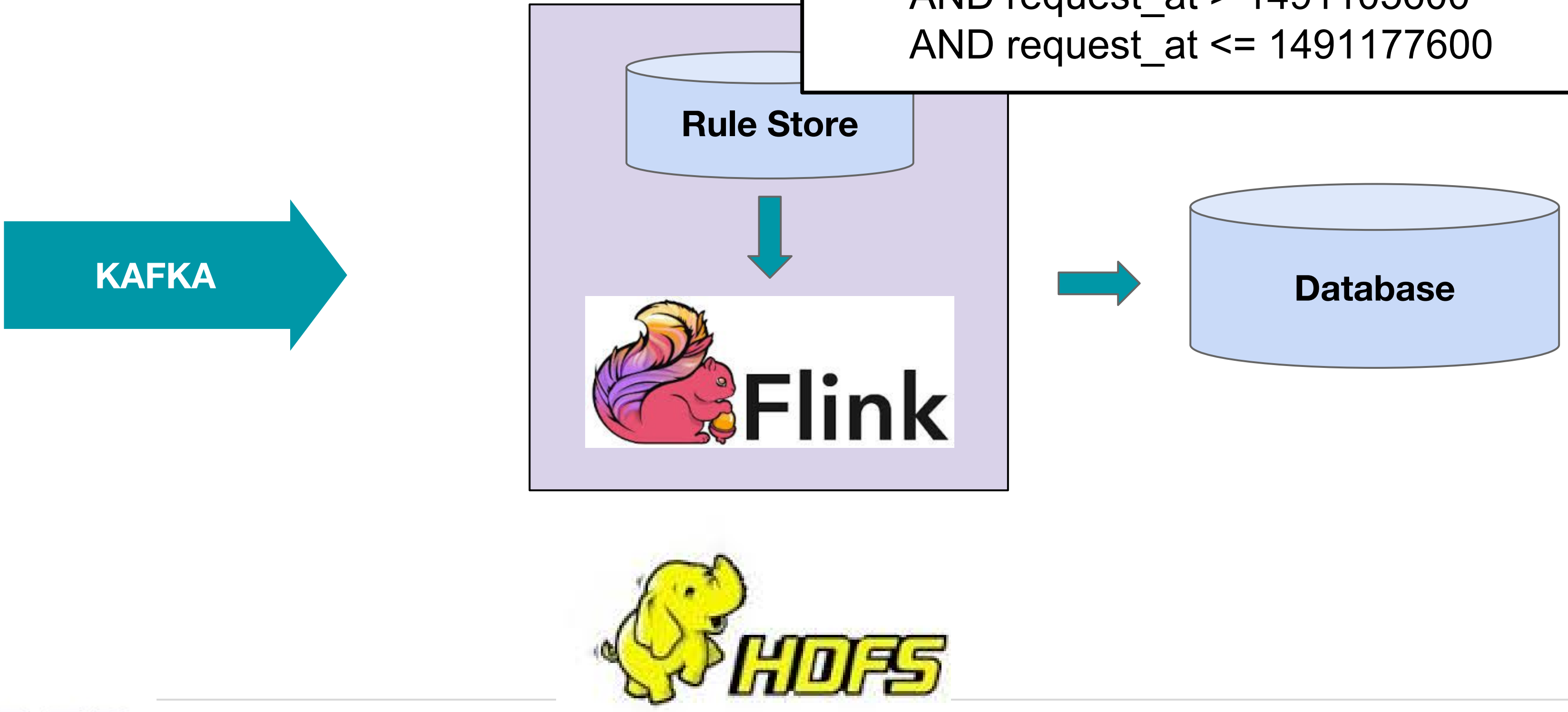
# Promotions using Flink

```
select count(*) from hp_api_created_trips
    WHERE city_id = 1
    AND fare > 10
    AND request_at > 1491105600
    AND request_at <= 1491177600
```

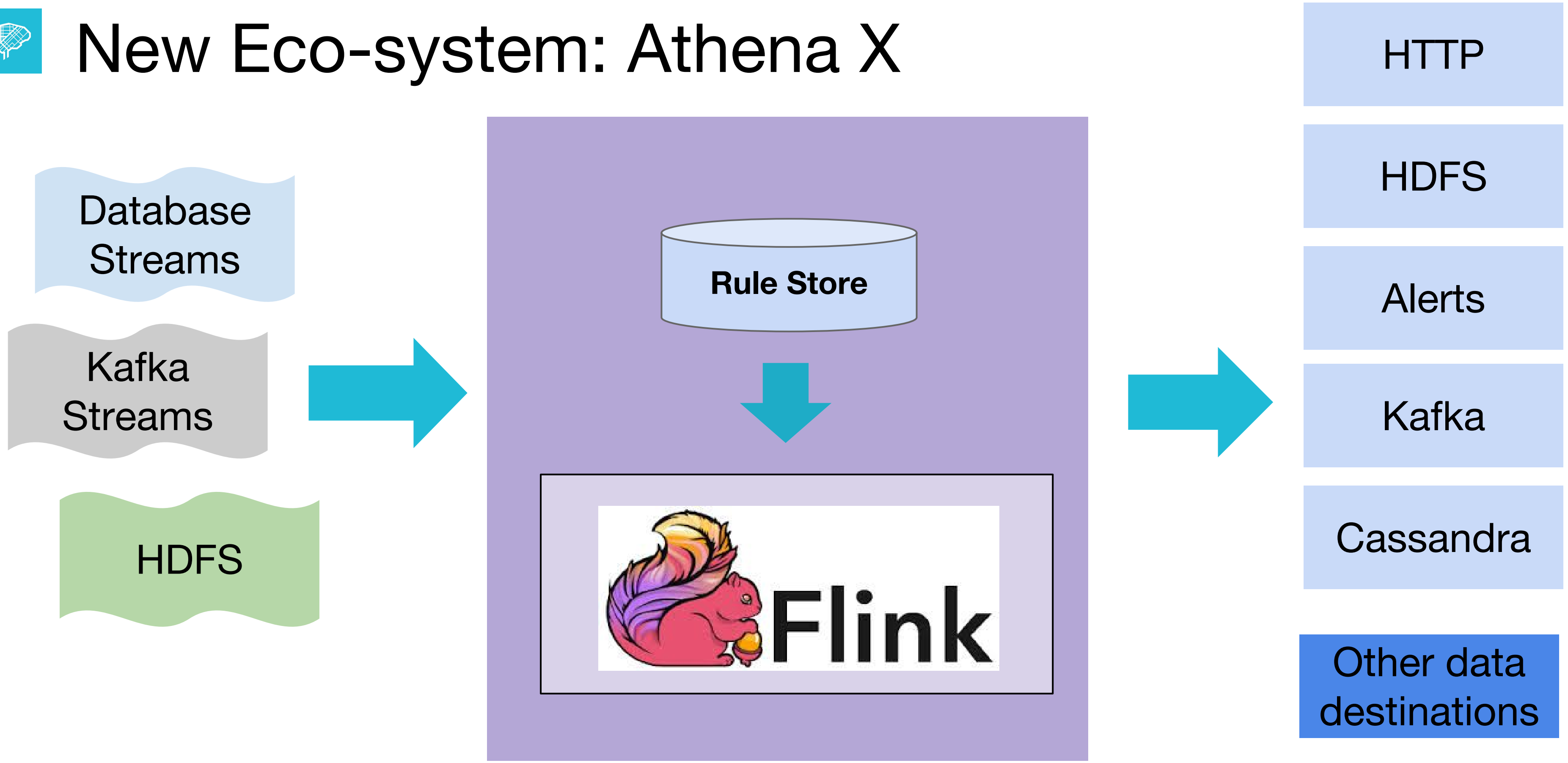


# Promotions using Flink

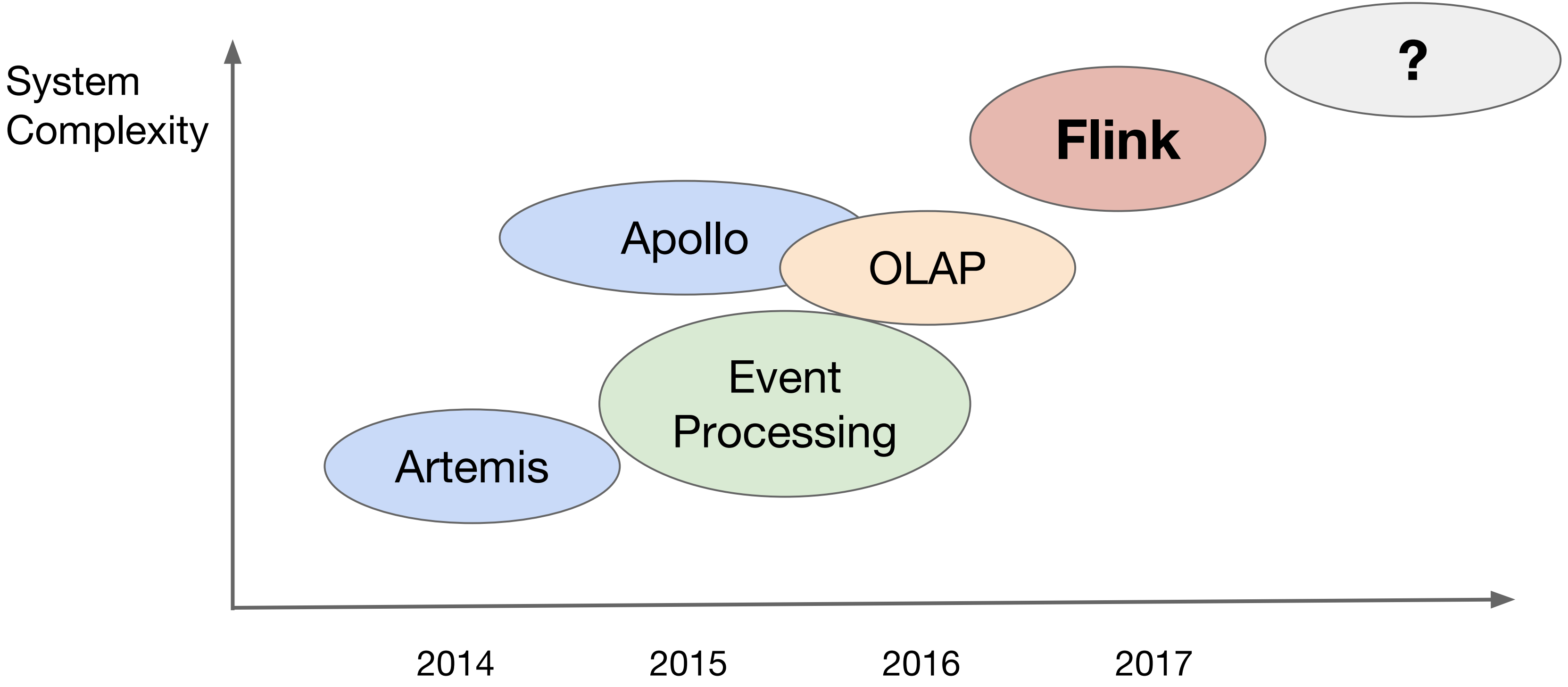
```
select count(*) from hp_api_created_trips
    WHERE city_id = 1
    AND fare > 10
    AND request_at > 1491105600
    AND request_at <= 1491177600
```



# New Eco-system: Athena X



# Are we there yet ?



# The road ahead ...

# Future Discussions

- To (Apache) Beam or not to Beam?
- Real-time Machine Learning
- Auto scaling

# AthenaX - Flink deep dive

Haohui Mai
Bill Liu

(11:45 am)

Thank you

For more: eng.uber.com
Twitter: @UberEng