

# **Apache Kafka and the Rise of Event-Driven Microservices**

Jun Rao

Co-founder of Confluent

# LinkedIn at 2010 : World's Largest Professional Network

Connecting Talent ⇔ Opportunity. At scale...



200M+

Members Worldwide

2 new

Members Per Second

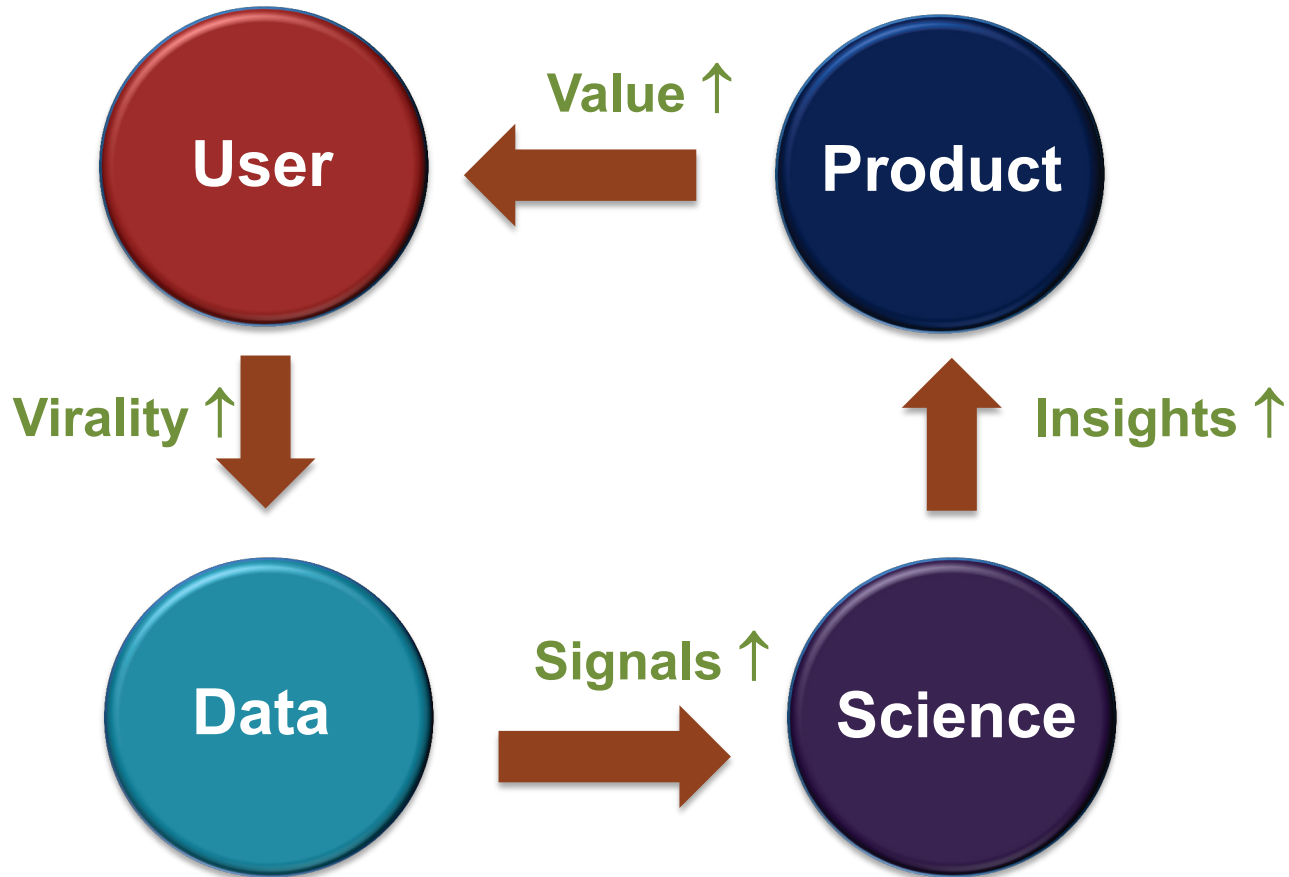
100M+

Monthly Unique Visitors

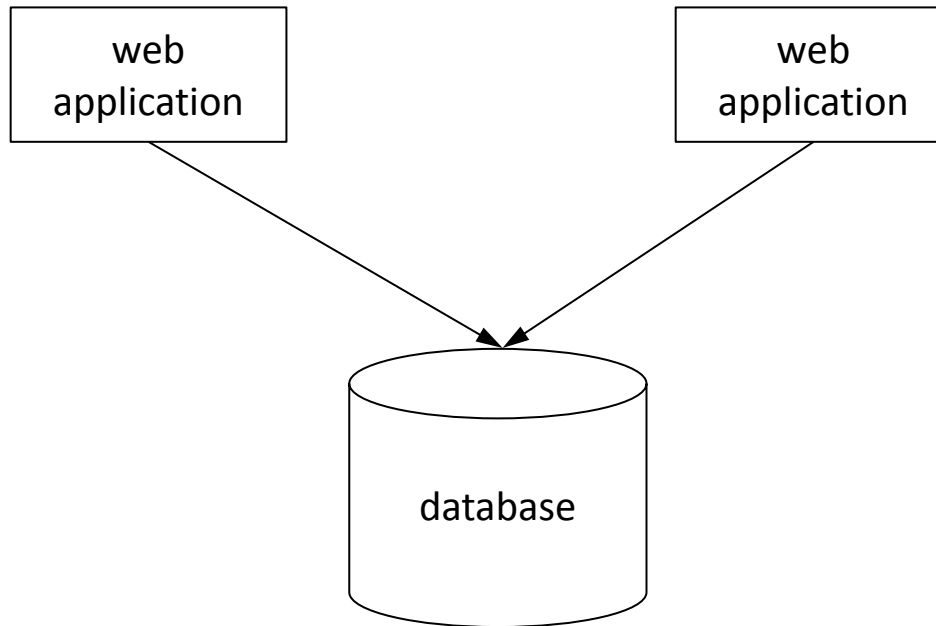
2M+

Company Pages

# It's all about data!



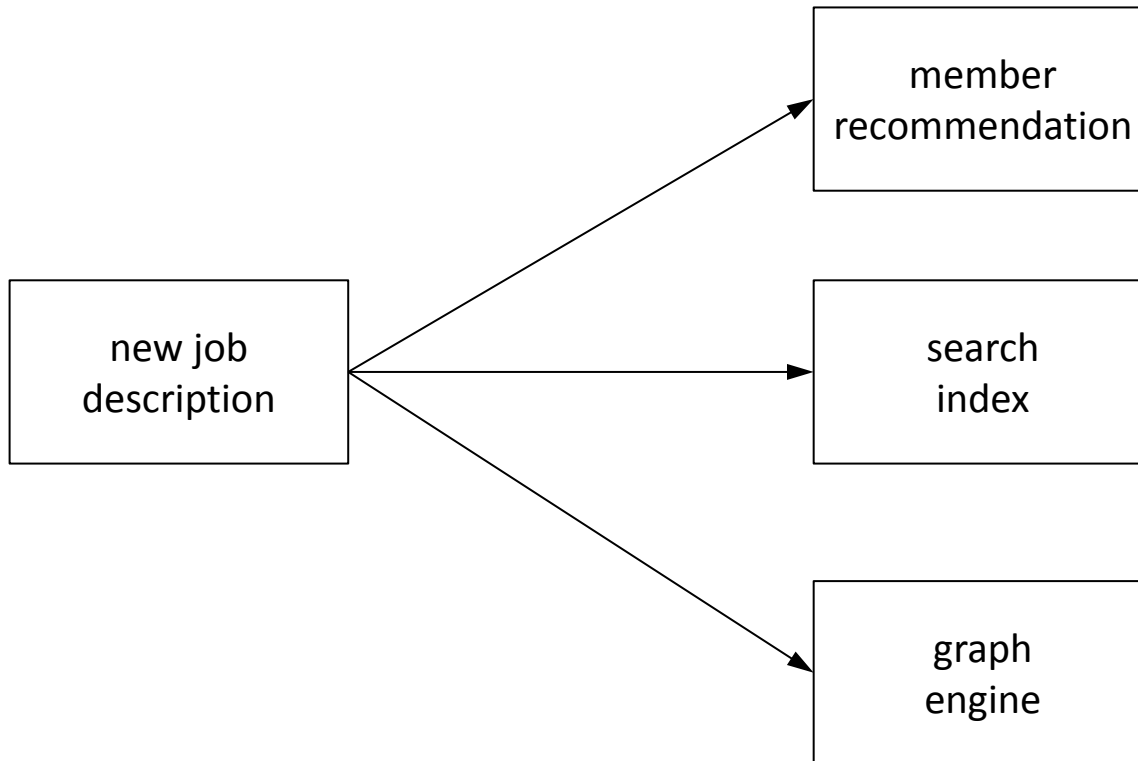
# Initial database driven architecture



# Realization #1: Event > State

- State: I work at Confluent
- Event: I changed job to work at Confluent

# Event driven microservices



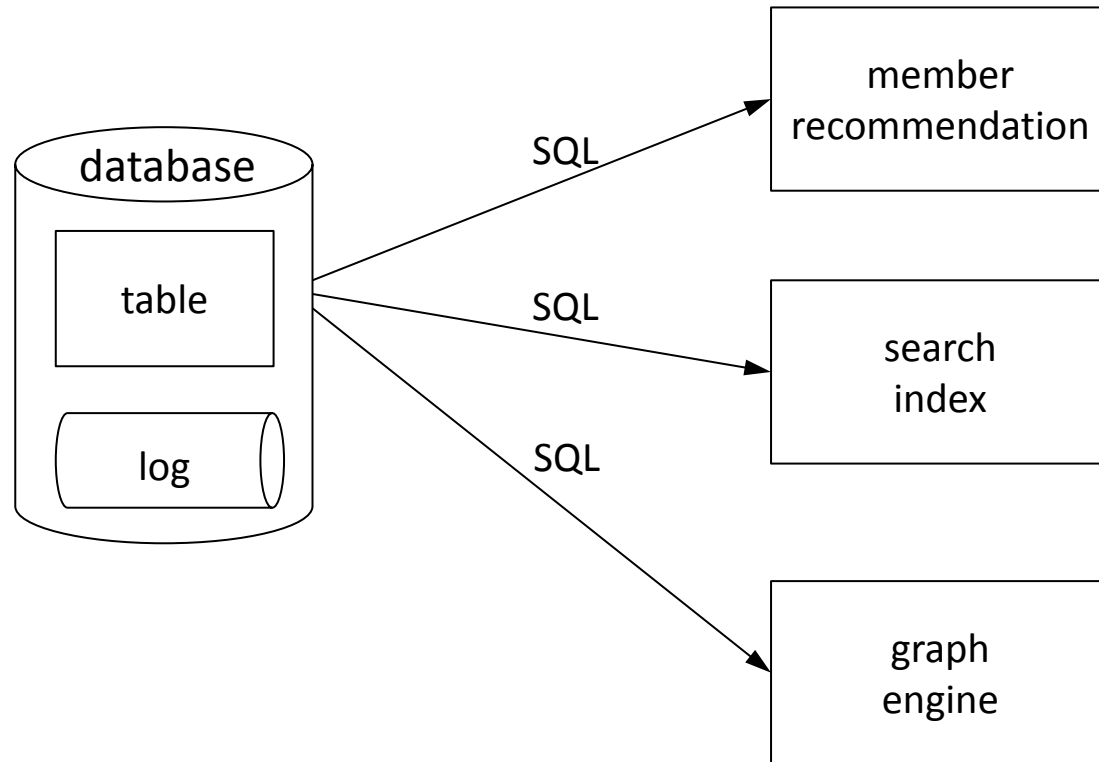
# Realization #2: leverage non-transactional data

- Business metrics
  - clicks, search keywords, pageviews
- Operational metrics
  - requests/sec, request types/sec
- Application logs
  - service calls, errors
- IOT
- ...

Database a mismatch for both!



# Mismatch #1: no first class API for events

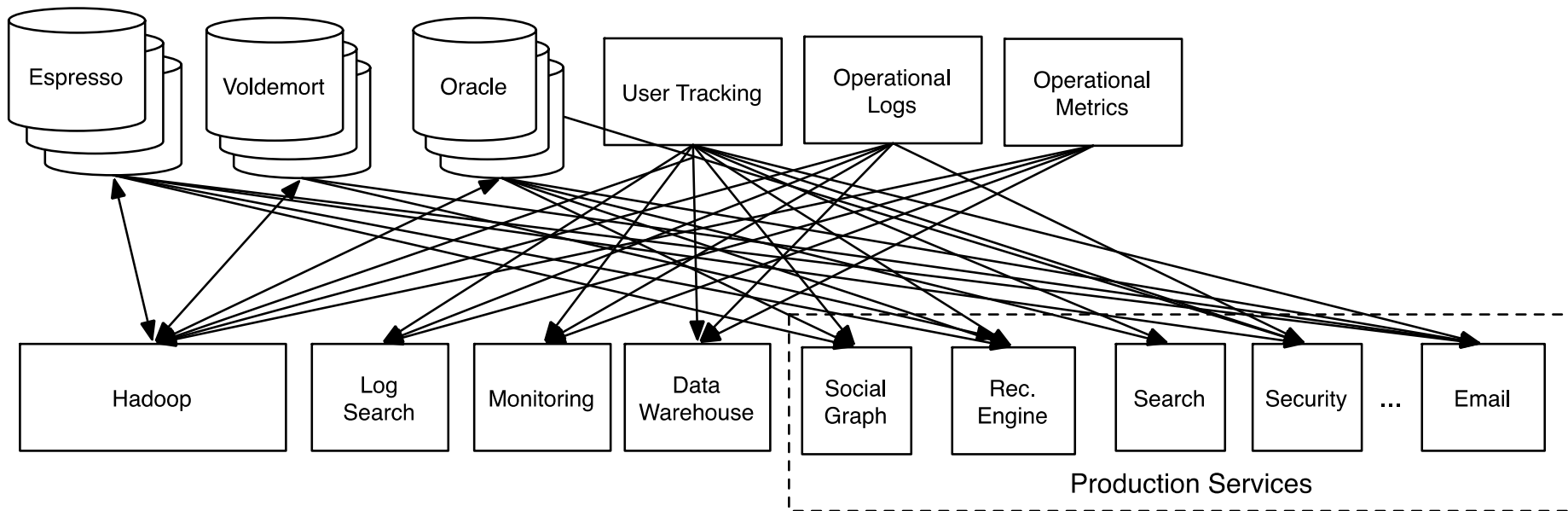


Tremendous load pressure on database!

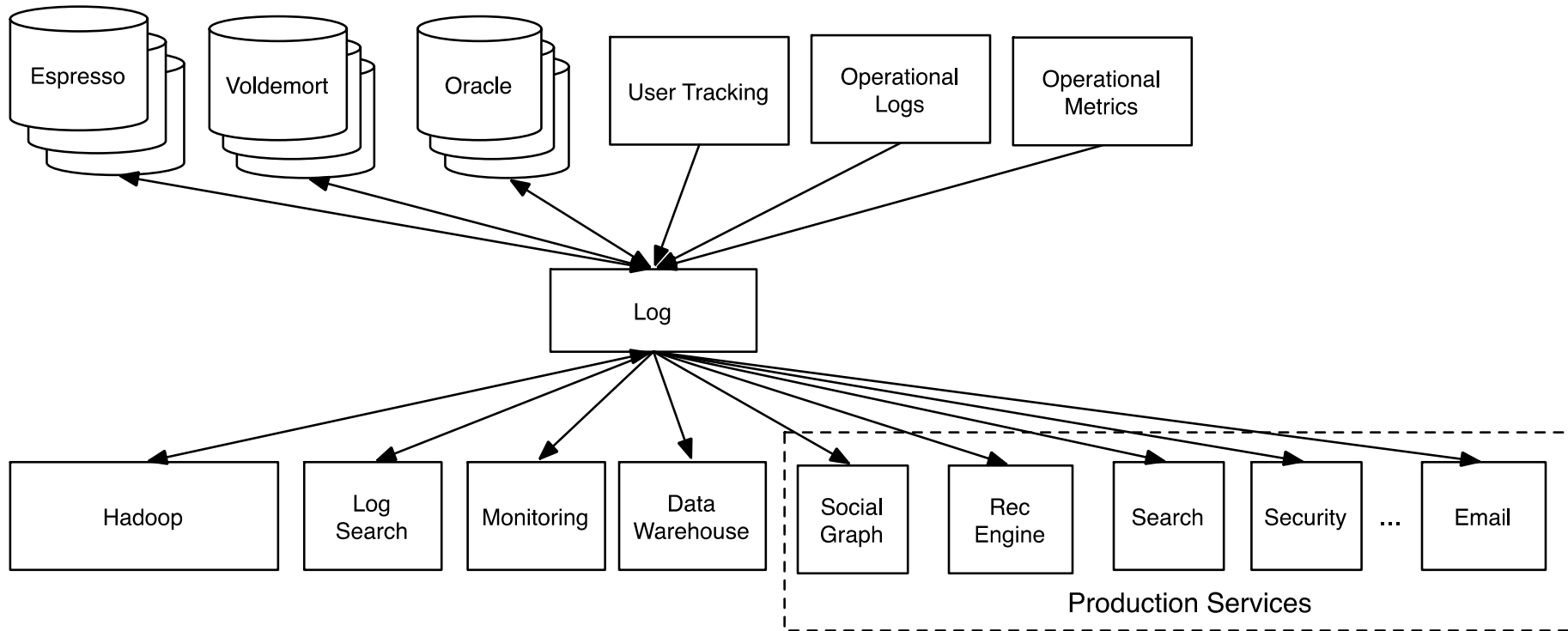
# Mismatch #2: not suitable for non-transactional data

- 1000X more volume
- Different transactional needs
- Not always needing a relation view

# Danger of Point-to-point Pipelines

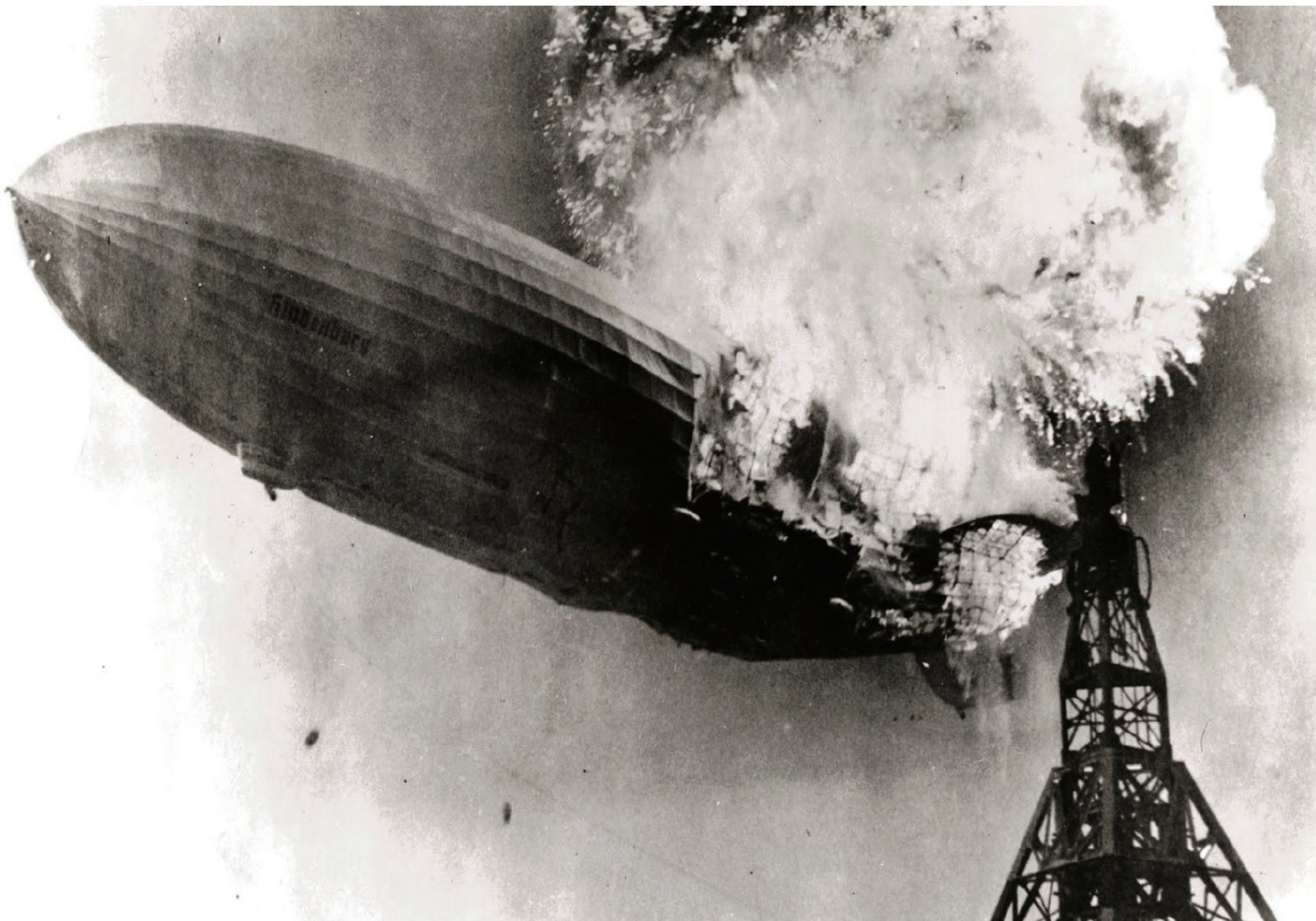


# Ideal Architecture



# 1st Attempt: Don't Reinvent the Wheels

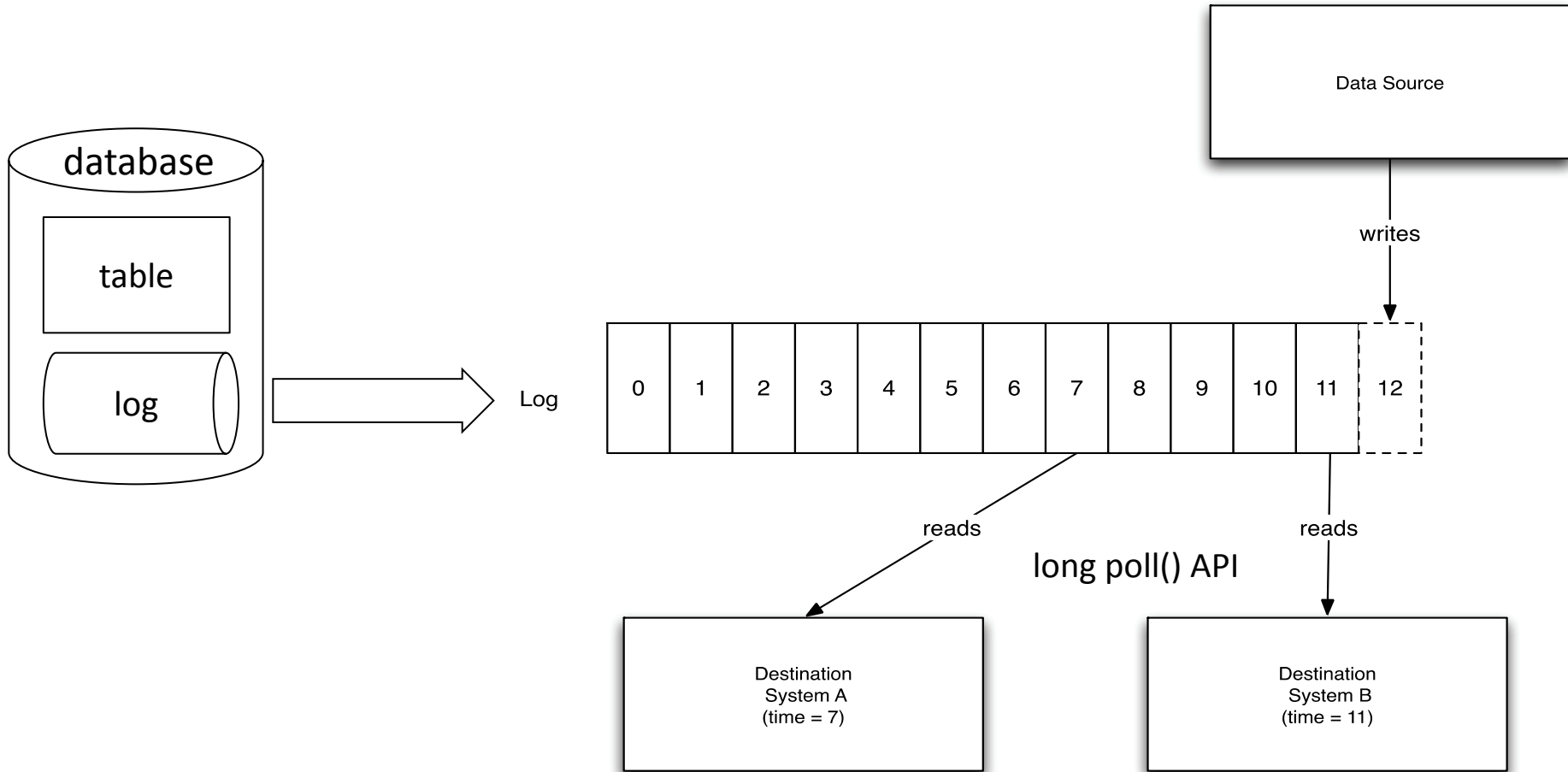
- Why not messaging systems?



# Version 1 of Kafka

- High throughput pub/sub
  - Design 1: make log first class citizen
  - Design 2: distributed architecture

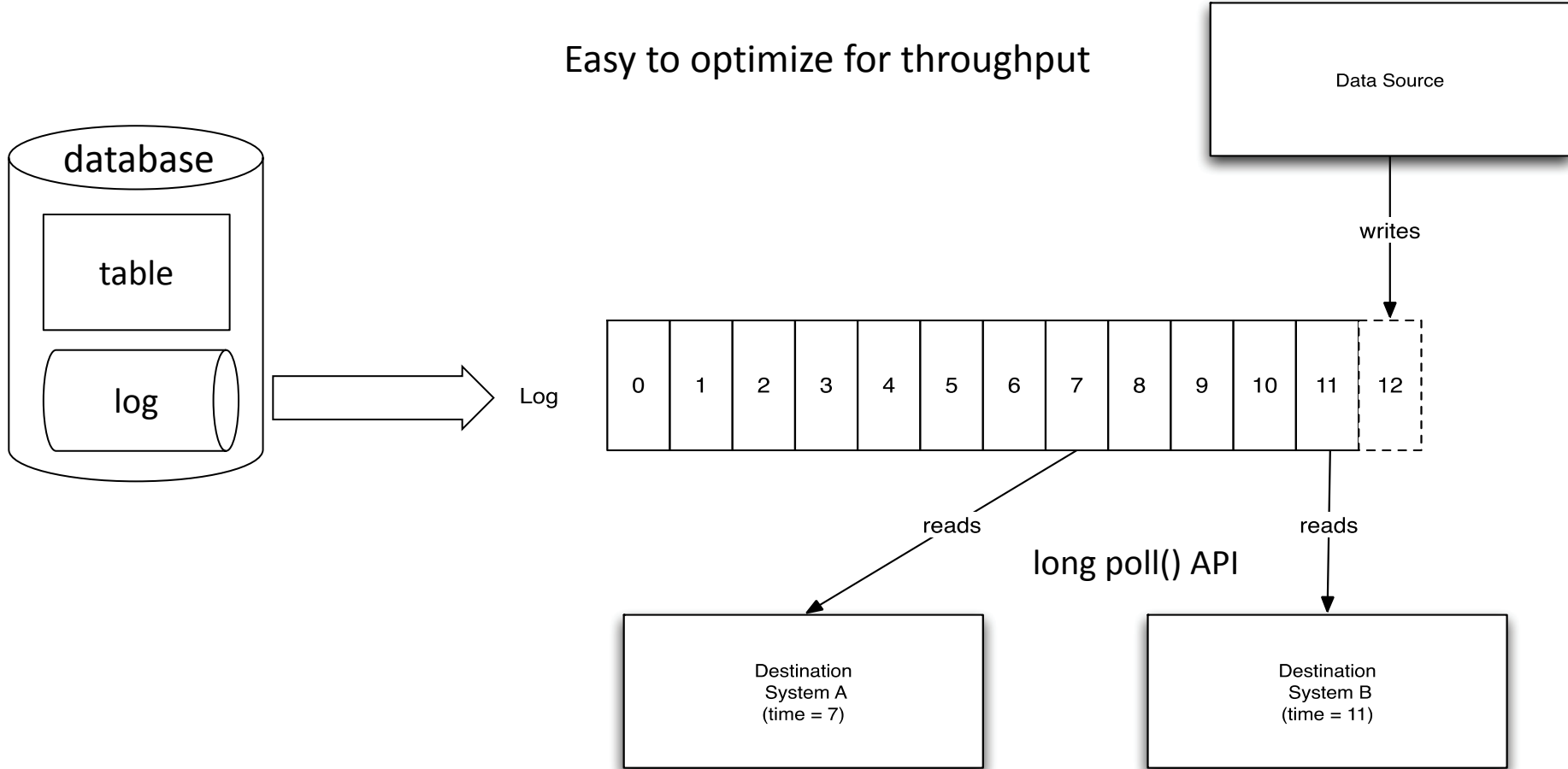
# Design #1: log as first a class citizen



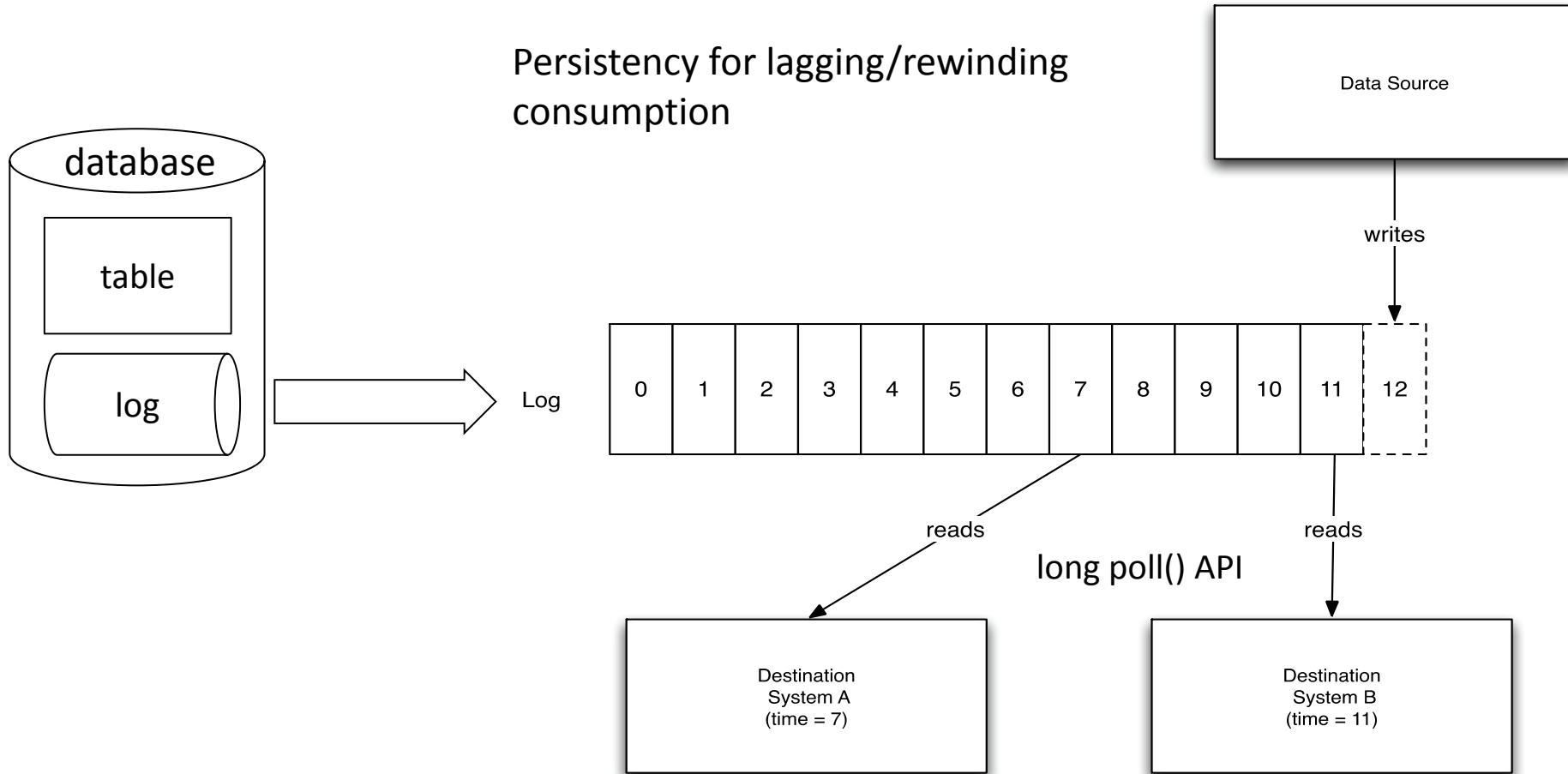


# Design #1: log as first a class citizen

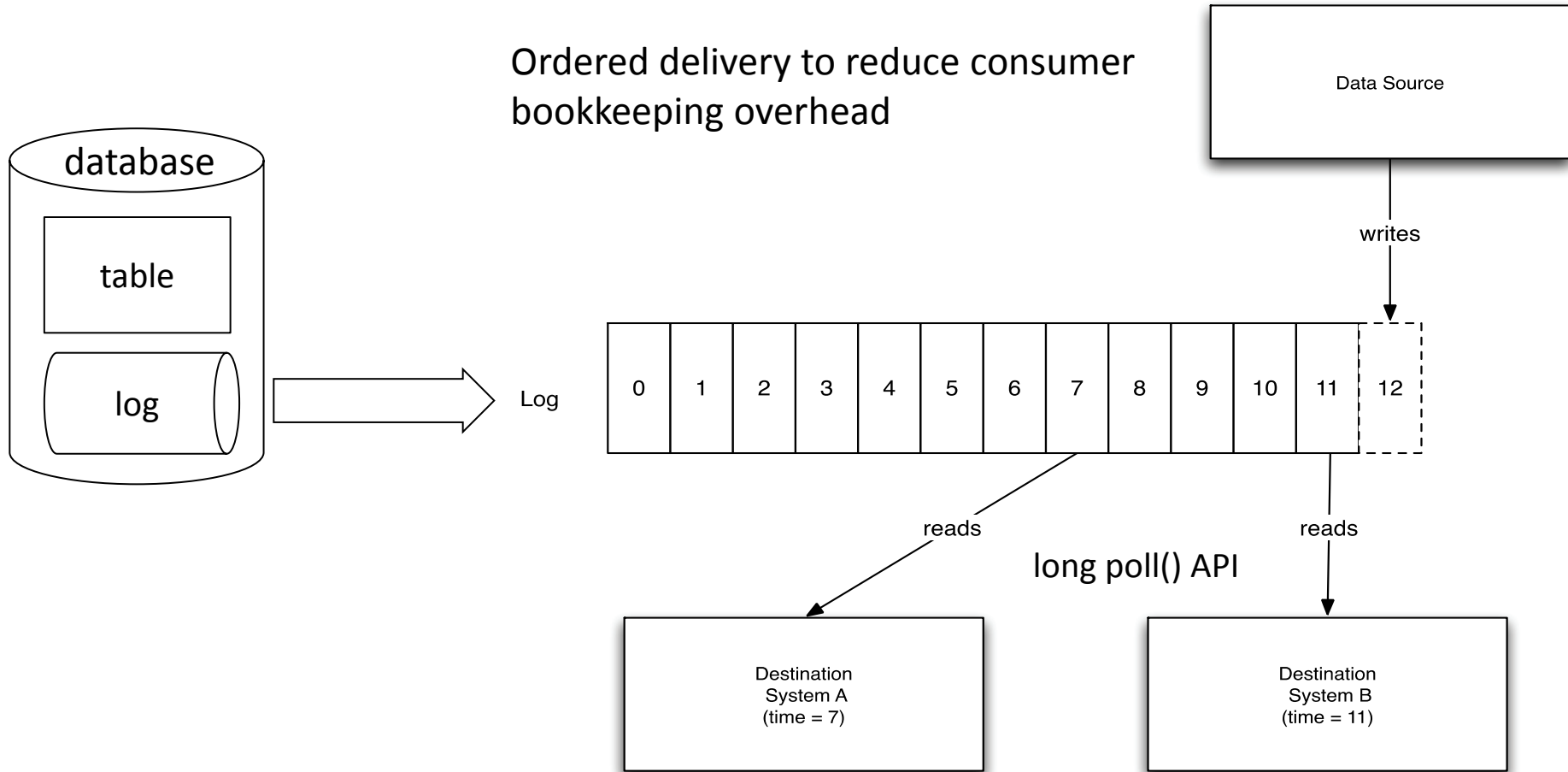
Easy to optimize for throughput



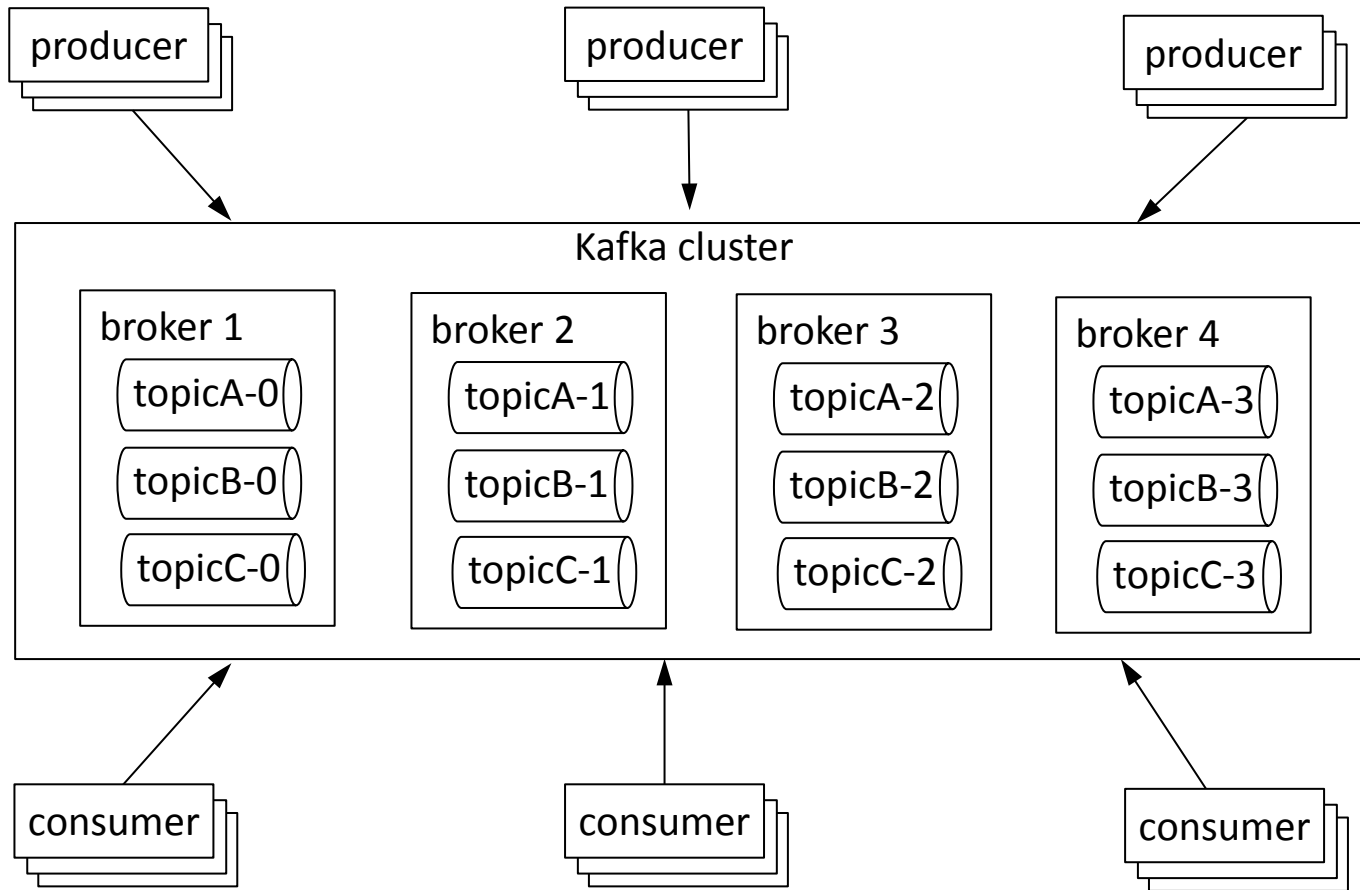
# Design #1: log as first a class citizen



# Design #1: log as first a class citizen



# Design #2: distributed architecture



# Kafka at LinkedIn in 2011

- 28 billion messages/day
- 460 thousand messages written/sec
- 2.3 million messages read/sec
- Tens of thousands of producers
  - Every production service is a producer
- Data democracy!

# Kafka => Apache in 2011



**6 of the top 10  
travel companies**



**7 of the top 10  
global banks**

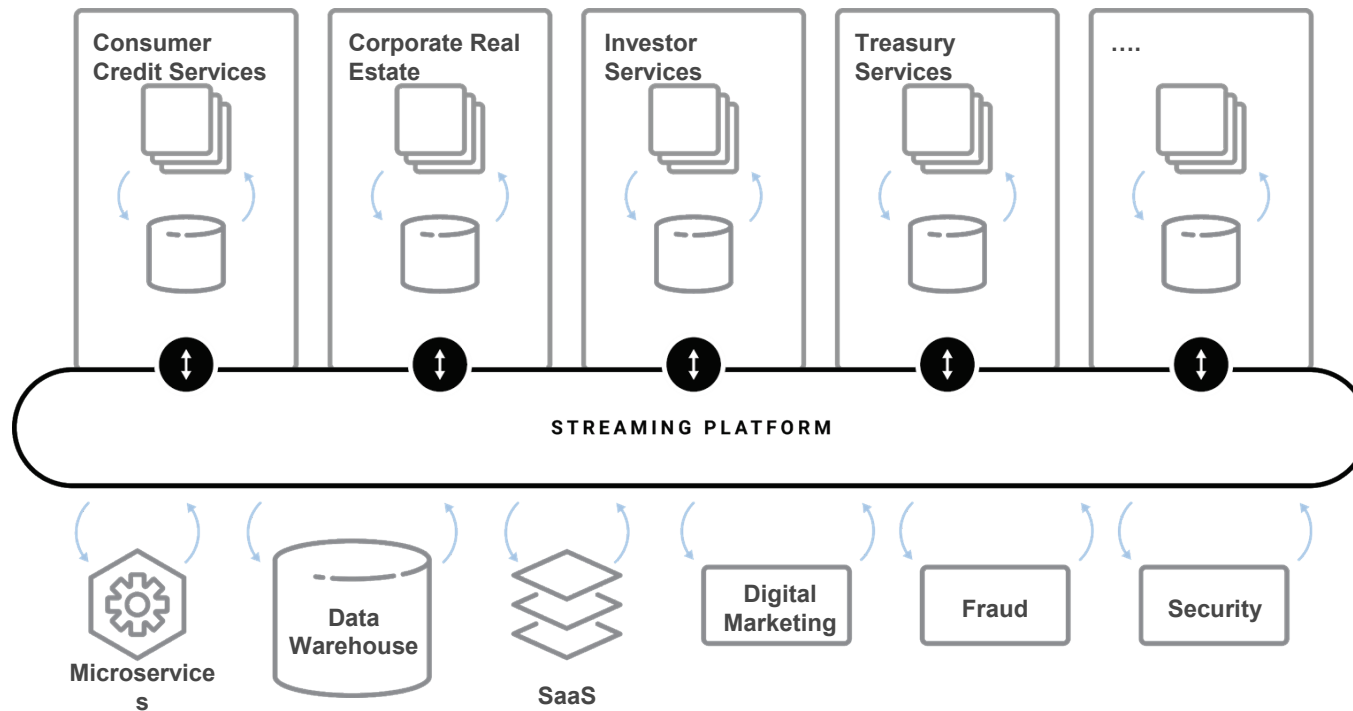


**8 of the top 10  
insurance companies**



**9 of the top 10  
telecom companies**

# Royal Bank of Canada Event-Driven Banking



30+ Use-cases

50+ apps

10+ different lines  
of businesses

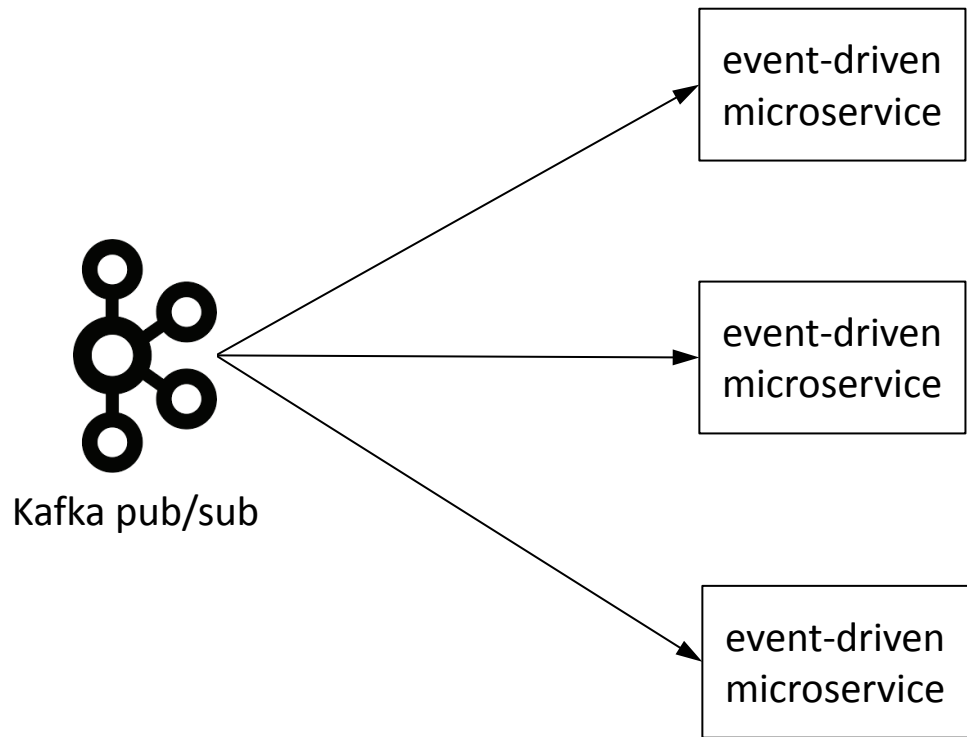
Lowering anomaly  
detection from weeks  
to real-time

# Carnival cruise line





# Building the processing layer



- Transformation
- Enrichment
- Aggregation

# Kafka Streams

```
KStream<Integer, Integer> input = builder.stream("numbers-topic");

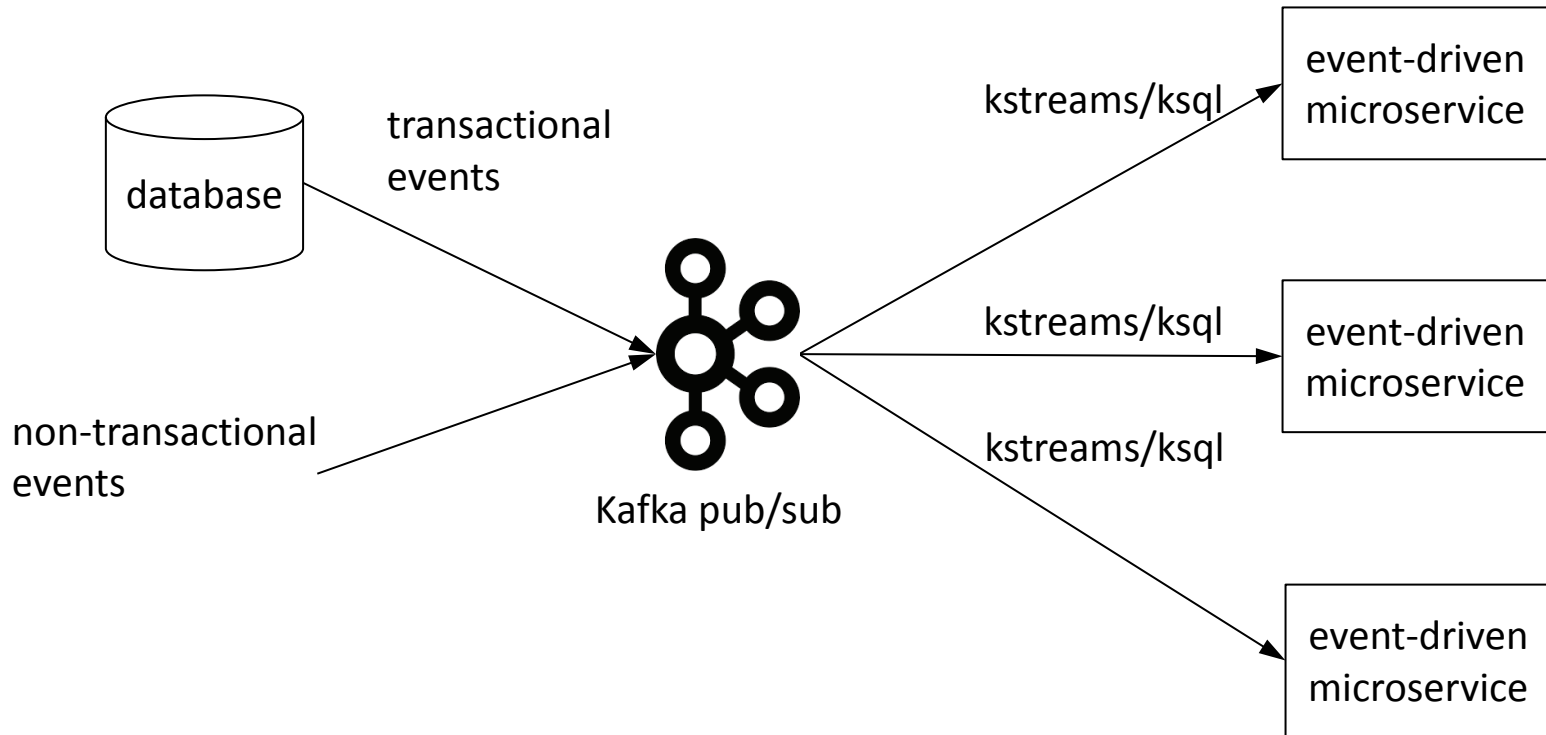
// Stateless computation
KStream<Integer, Integer> doubled = input.mapValues(v -> v * 2);

// Stateful computation
KStream<Integer, Integer> sumOfOdds = input
    .filter((k,v) -> v % 2 != 0)
    .selectKey((k, v) -> 1)
    .reduceByKey((v1, v2) -> v1 + v2, "sum-of-odds")
    .toStream();
```

# KSQL (from Confluent)

```
CREATE STREAM vip_actions AS  
  SELECT userid, page, action  
  FROM clickstream c  
  LEFT JOIN users u  
    ON c.userid = u.user_id  
  WHERE u.level = 'Platinum';
```

# Event driven platform



# Still interesting work ahead

- Scalability in metadata
- Streaming database
- Cloud integration

# Conclusion

- The success for business not only depends on software, but how they build software
- Apache Kafka offers a new platform than traditional database
- This is an exciting time to work on streams