STREAM PROCESSING AT LINKEDIN: APACHE KAFKA & APACHE SAMZA

Processing billions of events every day

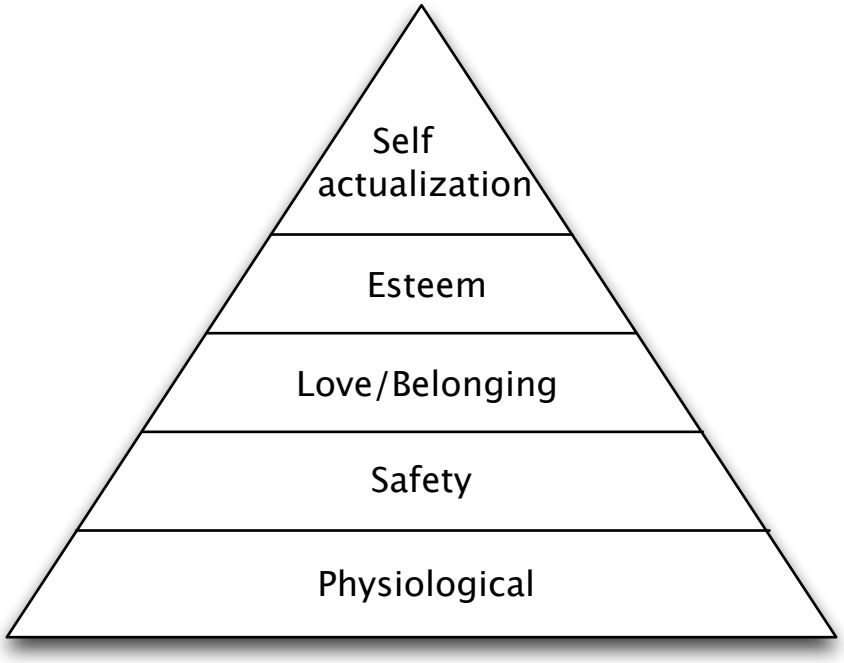
Neha Narkhede

- Co-founder and Head of Engineering @ Stealth Startup
- Prior to this…
  - Lead, Streams Infrastructure @ LinkedIn (Kafka & Samza)
  - One of the initial authors of Apache Kafka, committer and PMC member
- Reach out at @nehanarkhede

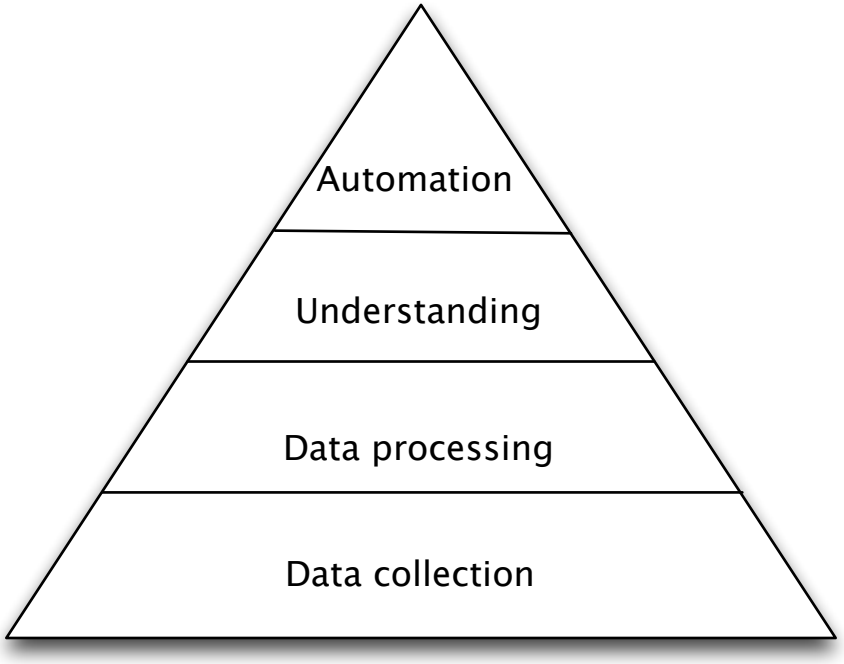
Agenda

- Real-time Data Integration
- Introduction to Logs & Apache Kafka
- Logs & Stream processing
- Apache Samza
- Stateful stream processing

The Data Needs Pyramid



Maslow's hierarchy of needs

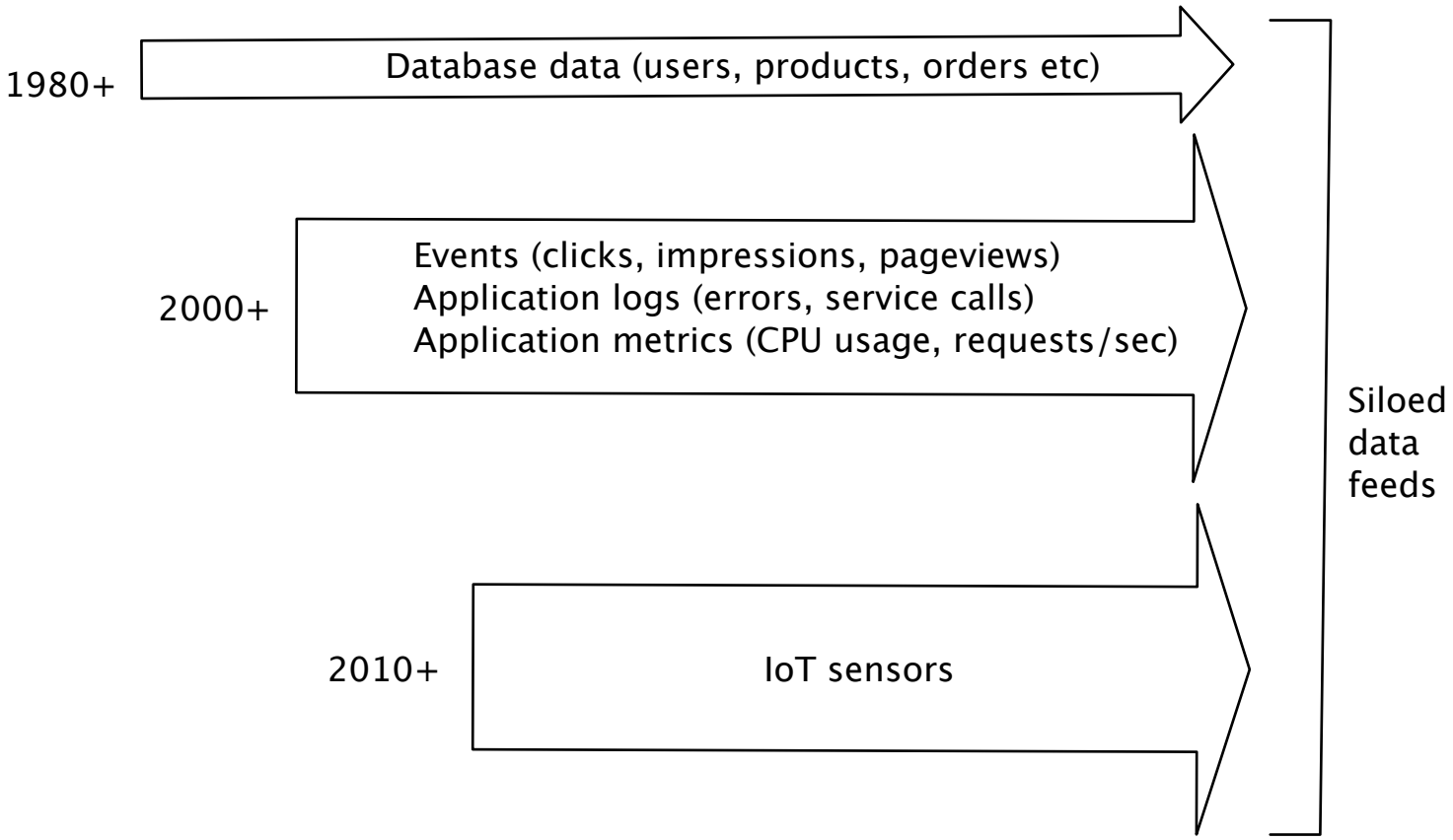


Data needs

Agenda

- Real-time Data Integration
  - Introduction to Logs & Apache Kafka
  - Logs & Stream processing
  - Apache Samza
  - Stateful stream processing

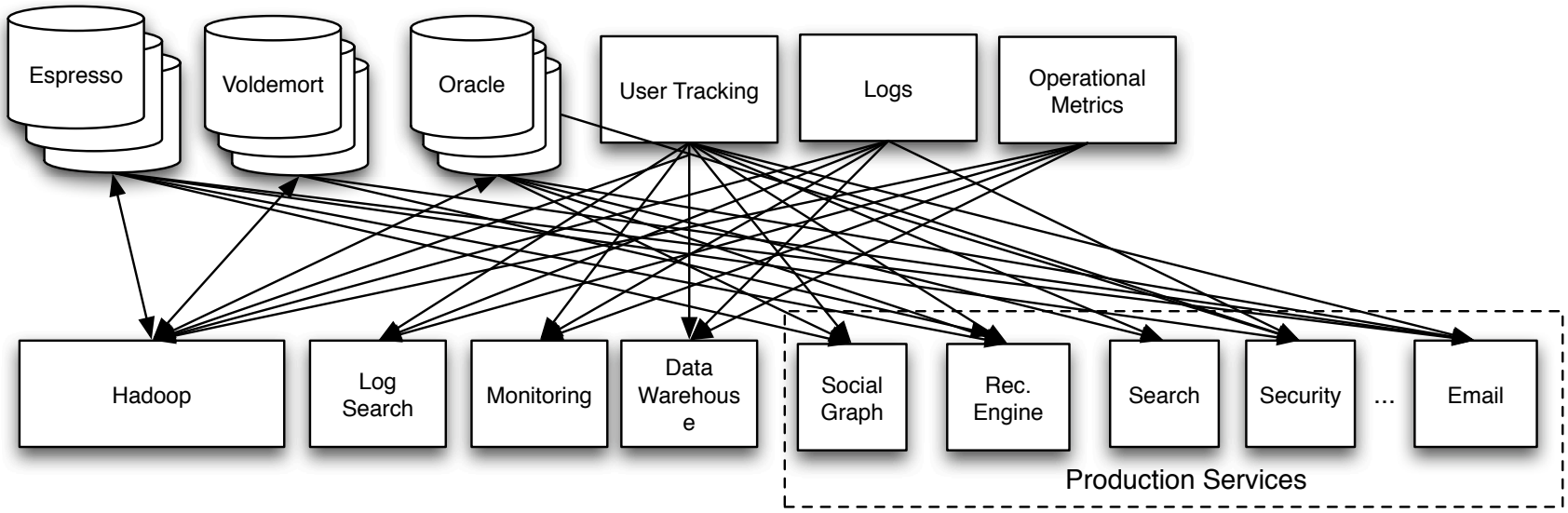
Increase in diversity of data



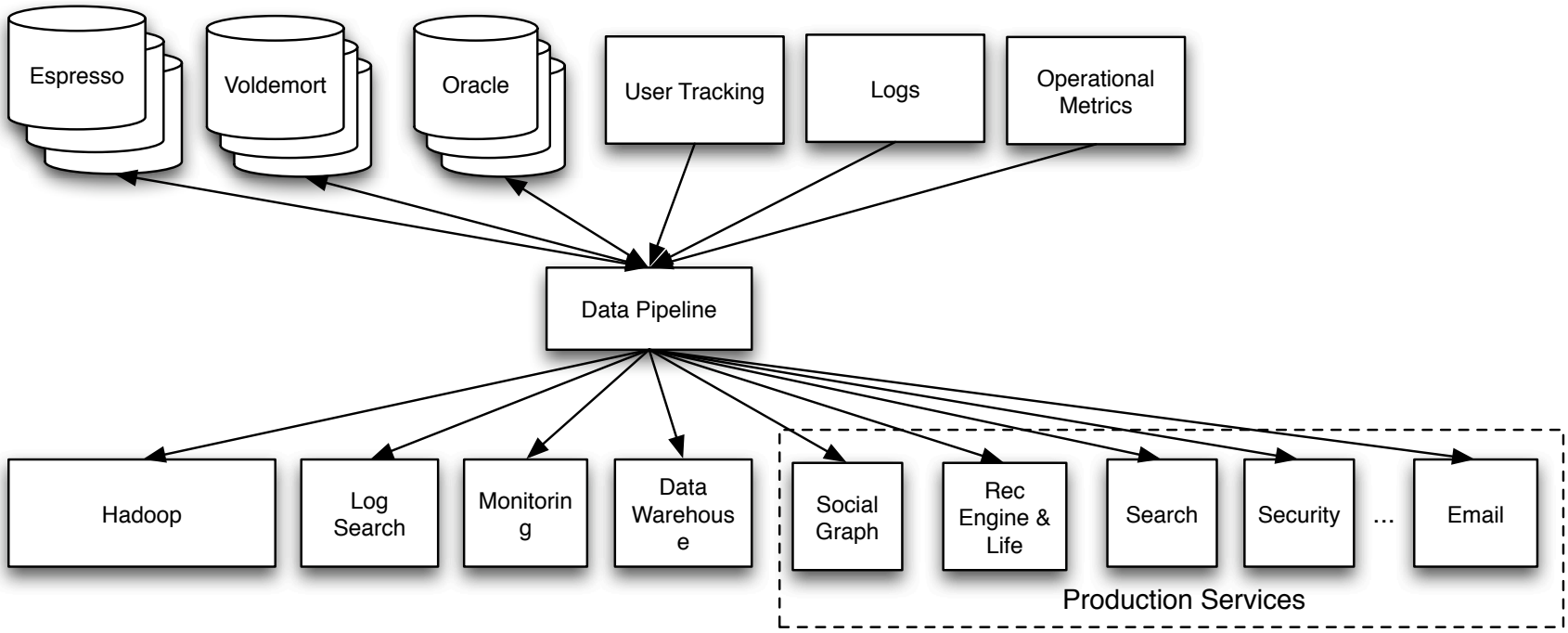
Explosion in diversity of systems

- Live Systems
  - Voldemort
  - Espresso
  - GraphDB
  - Search
  - Samza
- Batch
  - Hadoop
  - Teradata

Data integration disaster



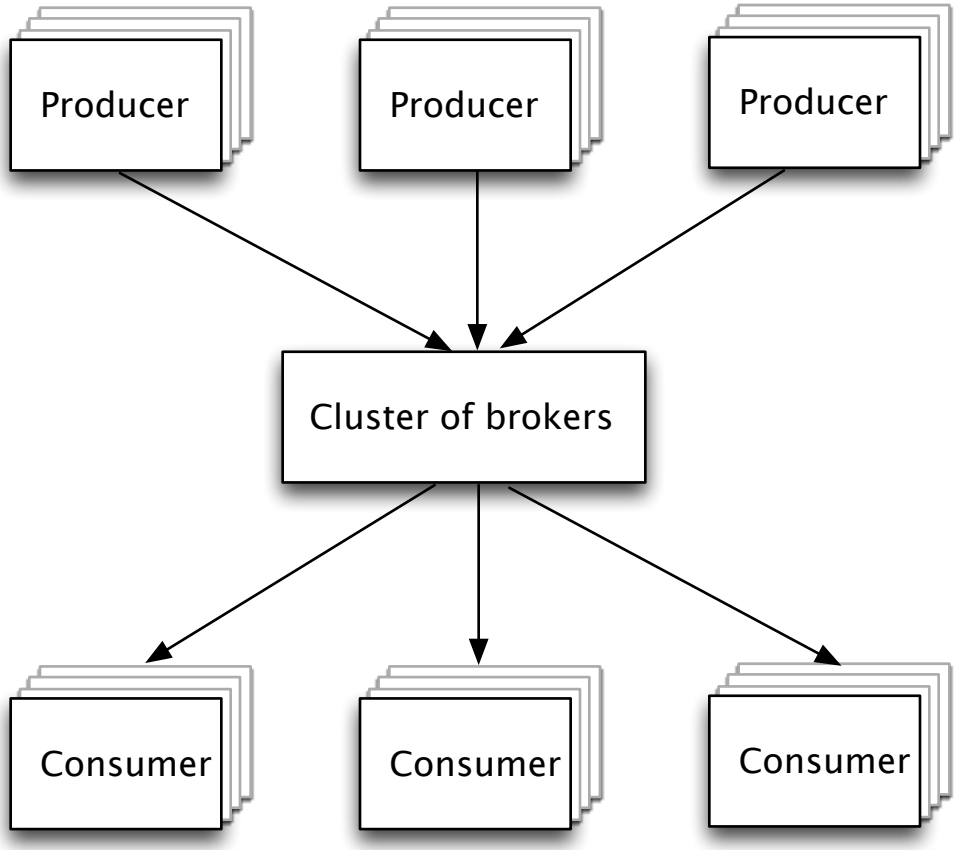
Centralized service



Agenda

- Real-time Data Integration
- **Introduction to Logs & Apache Kafka**
- Logs & Stream processing
- Apache Samza
- Stateful stream processing

Kafka at 10,000 ft



- Distributed from ground up
- Persistent
- Multi-subscriber

Key design principles

- Scalability of a file system
  - Hundreds of MB/sec/server throughput
  - Many TBs per server
- Guarantees of a database
  - Messages strictly ordered
  - All data persistent
- Distributed by default
  - Replication model
  - Partitioning model

Kafka adoption

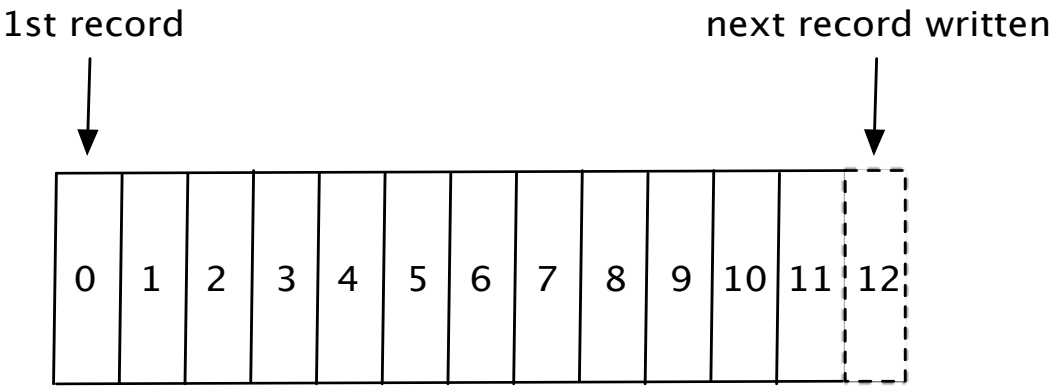
Apache Kafka @ LinkedIn

- 175 TB of in-flight log data per colo
- Low-latency: ~1.5ms
- Replicated to each datacenter
- Tens of thousands of data producers
- Thousands of consumers
- 7 million messages written/sec
- 35 million messages read/sec
- Hadoop integration

Logs

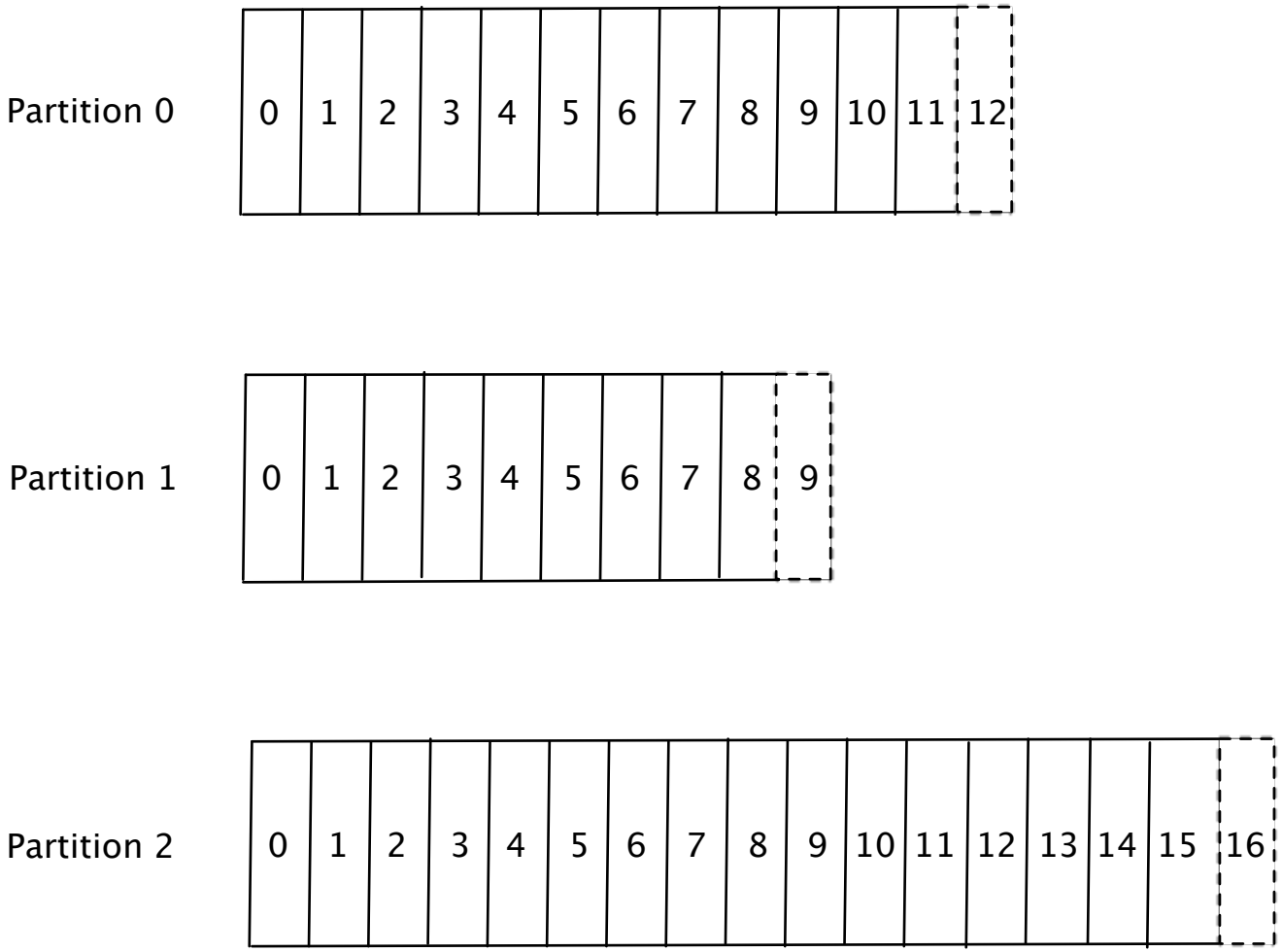
The data structure every systems engineer should know

The Log

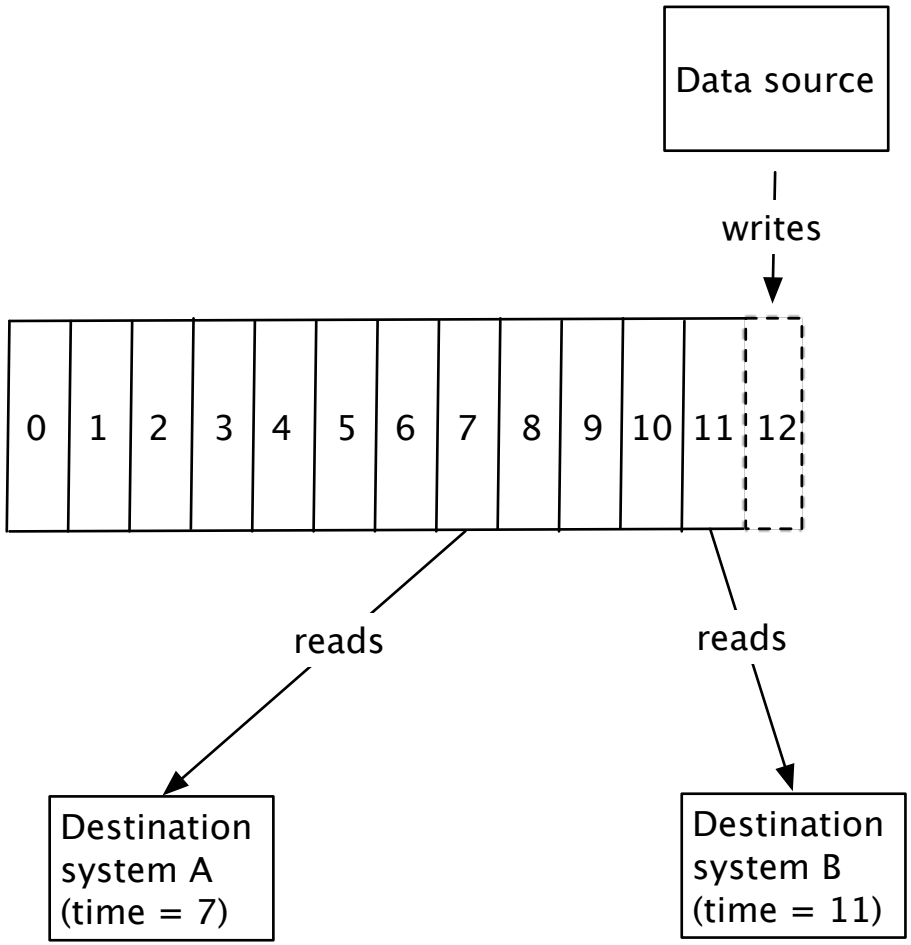


- Ordered
- Append only
- Immutable

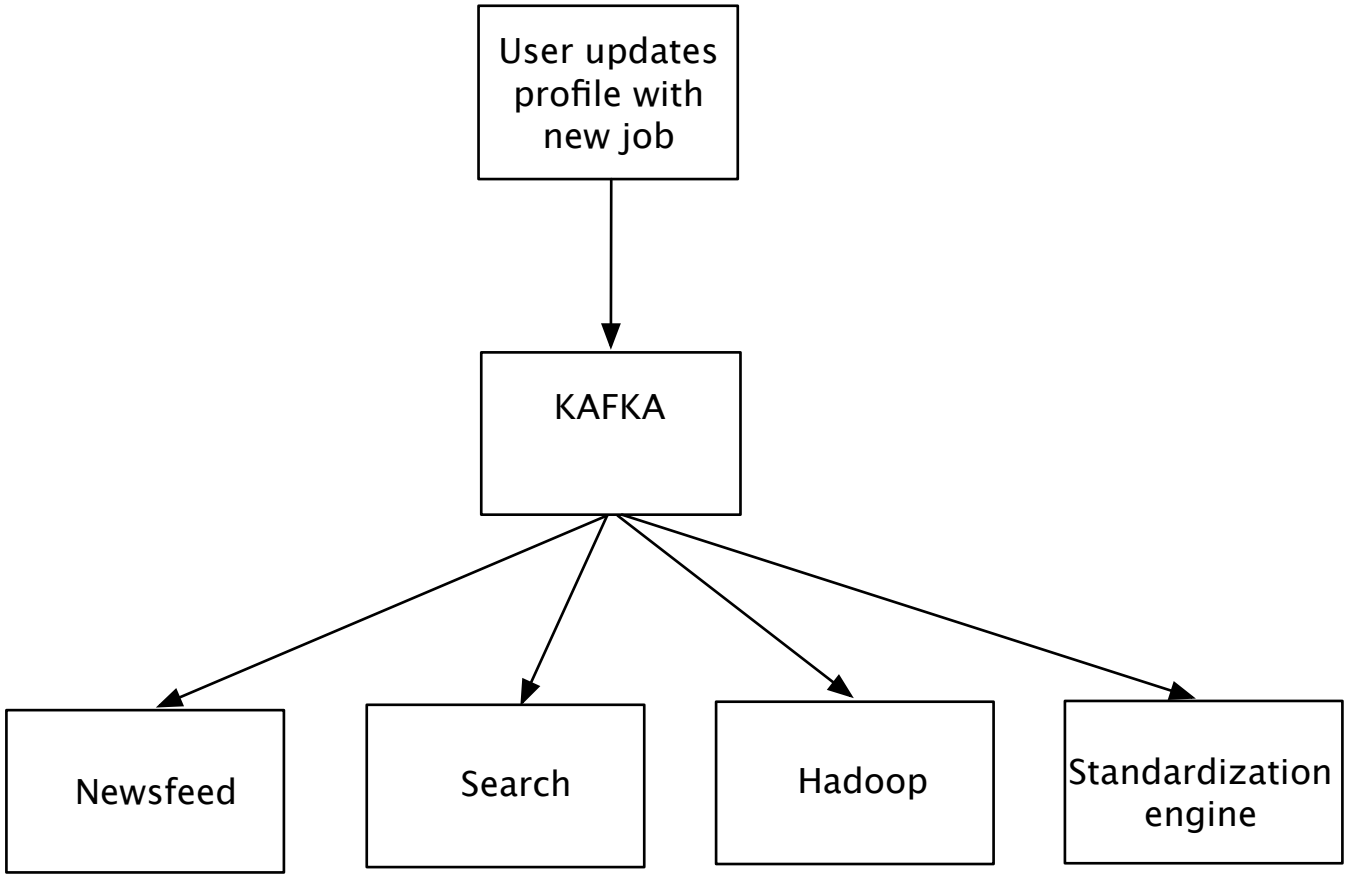
The Log: Partitioning



Logs: pub/sub done right



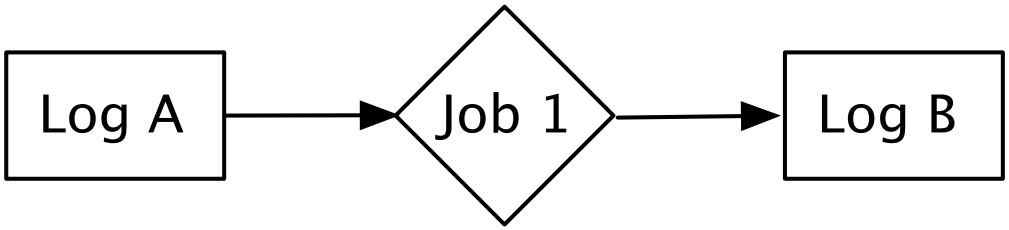
Logs for data integration



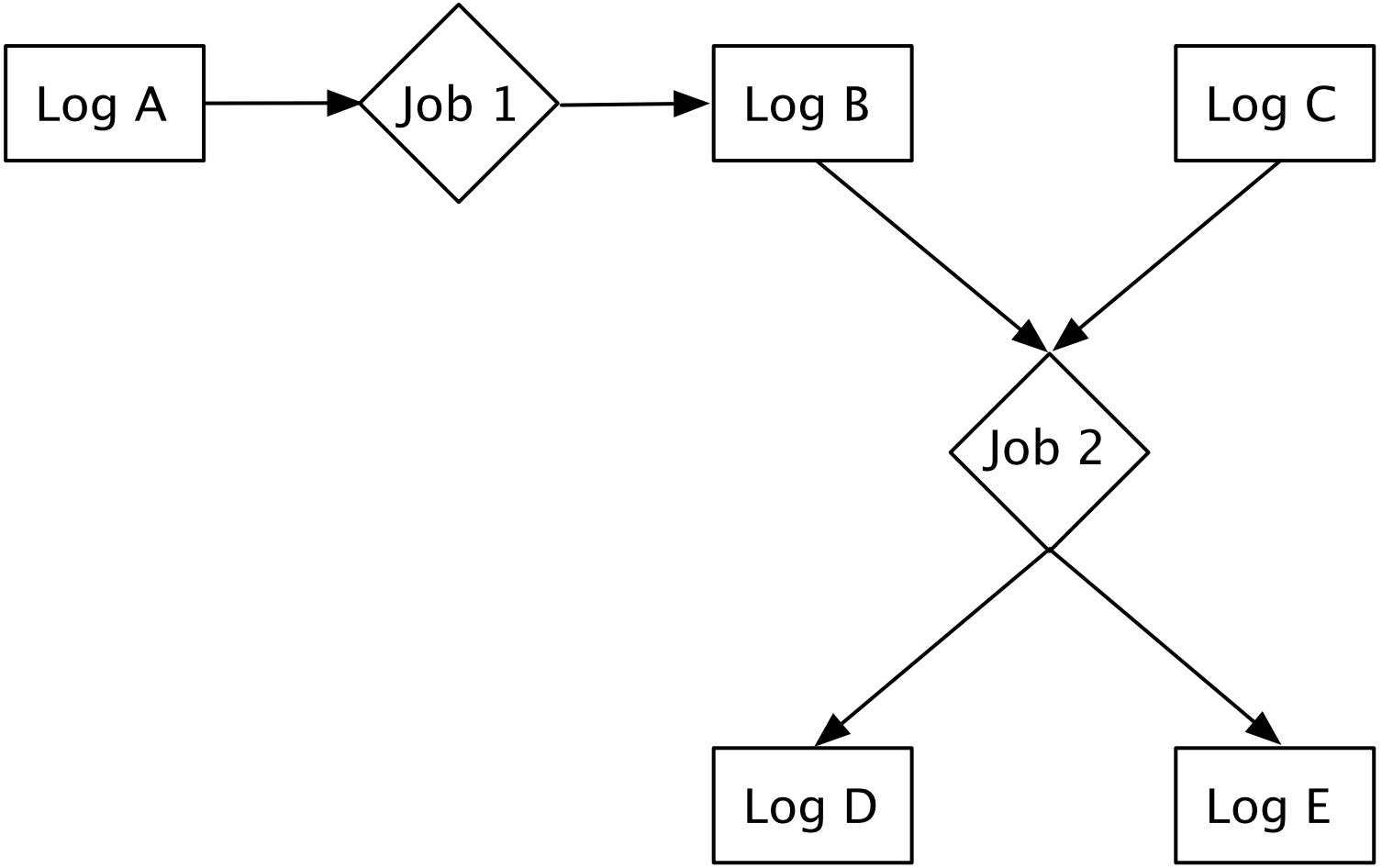
Agenda

- Real-time Data Integration
- Introduction to Logs & Apache Kafka
- **Logs & Stream processing**
- Apache Samza
- Stateful stream processing

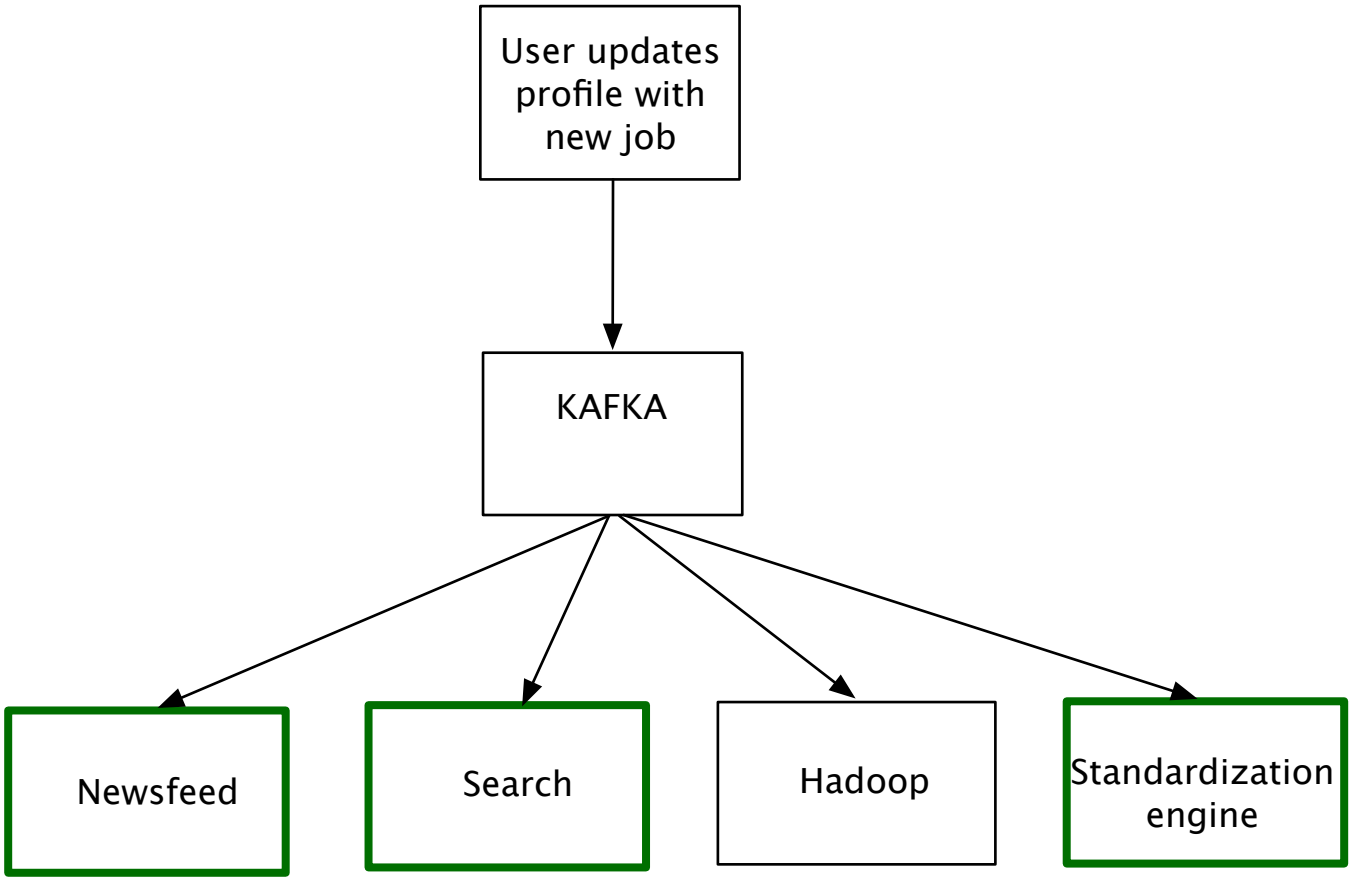
Stream processing = f(log)



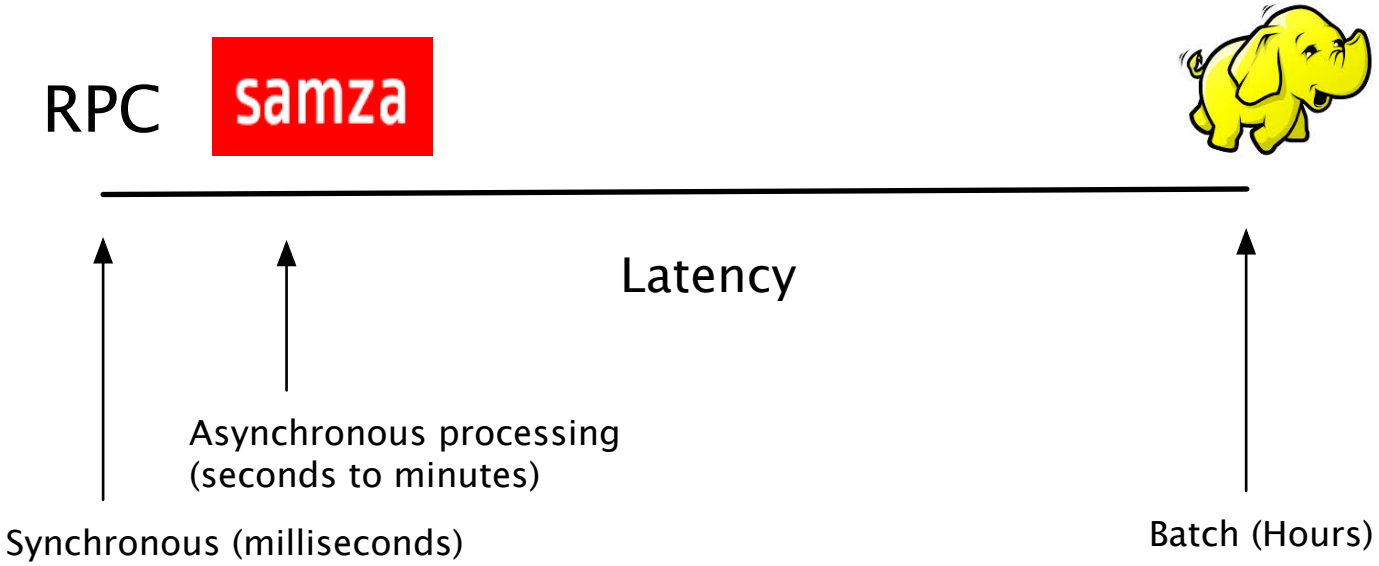
Stream processing = f(log)



Apache Samza at LinkedIn



Latency spectrum of data systems



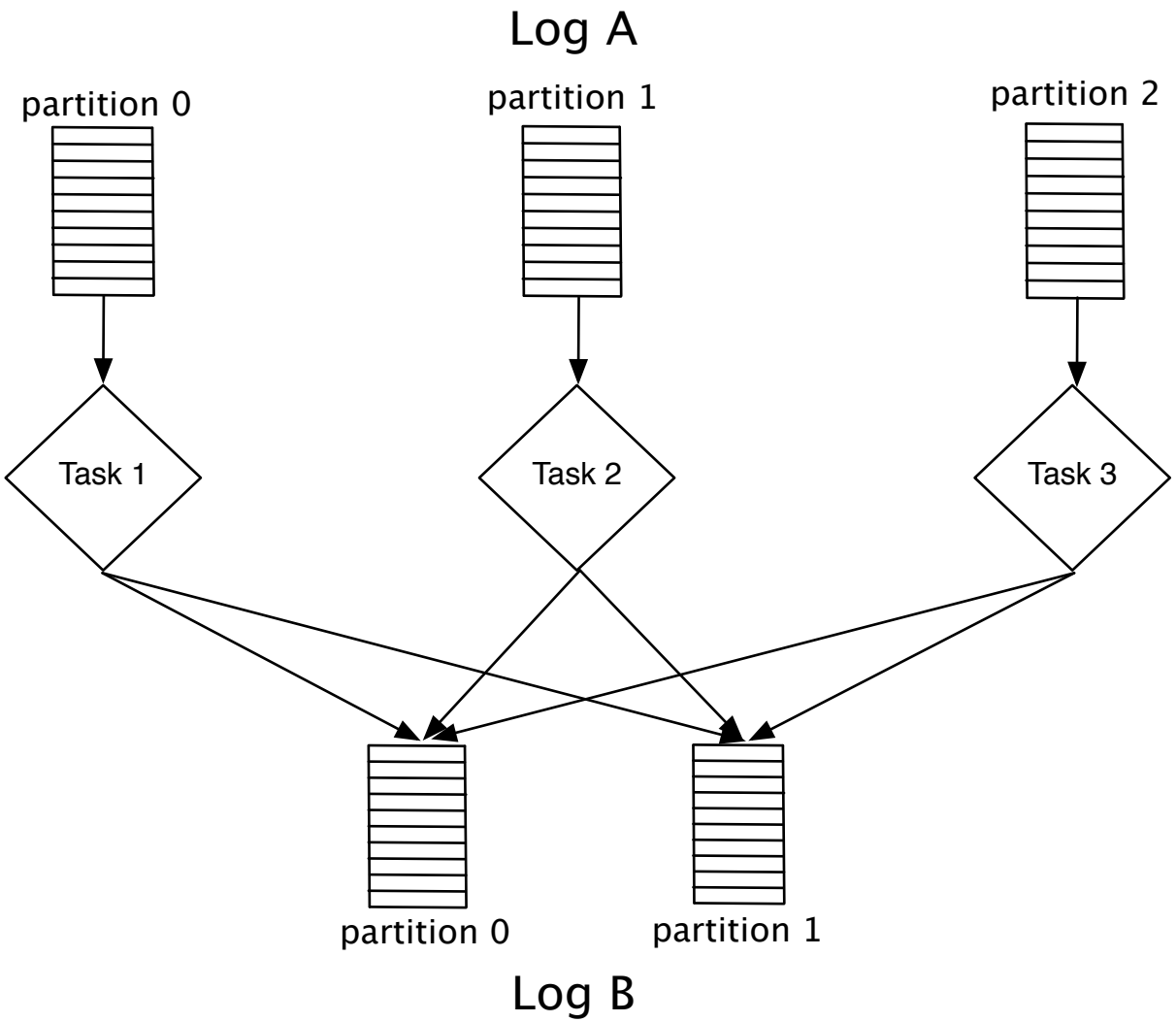
Agenda

- Real-time Data Integration
- Introduction to Logs & Apache Kafka
- Logs & Stream processing
- **Apache Samza**
- Stateful stream processing

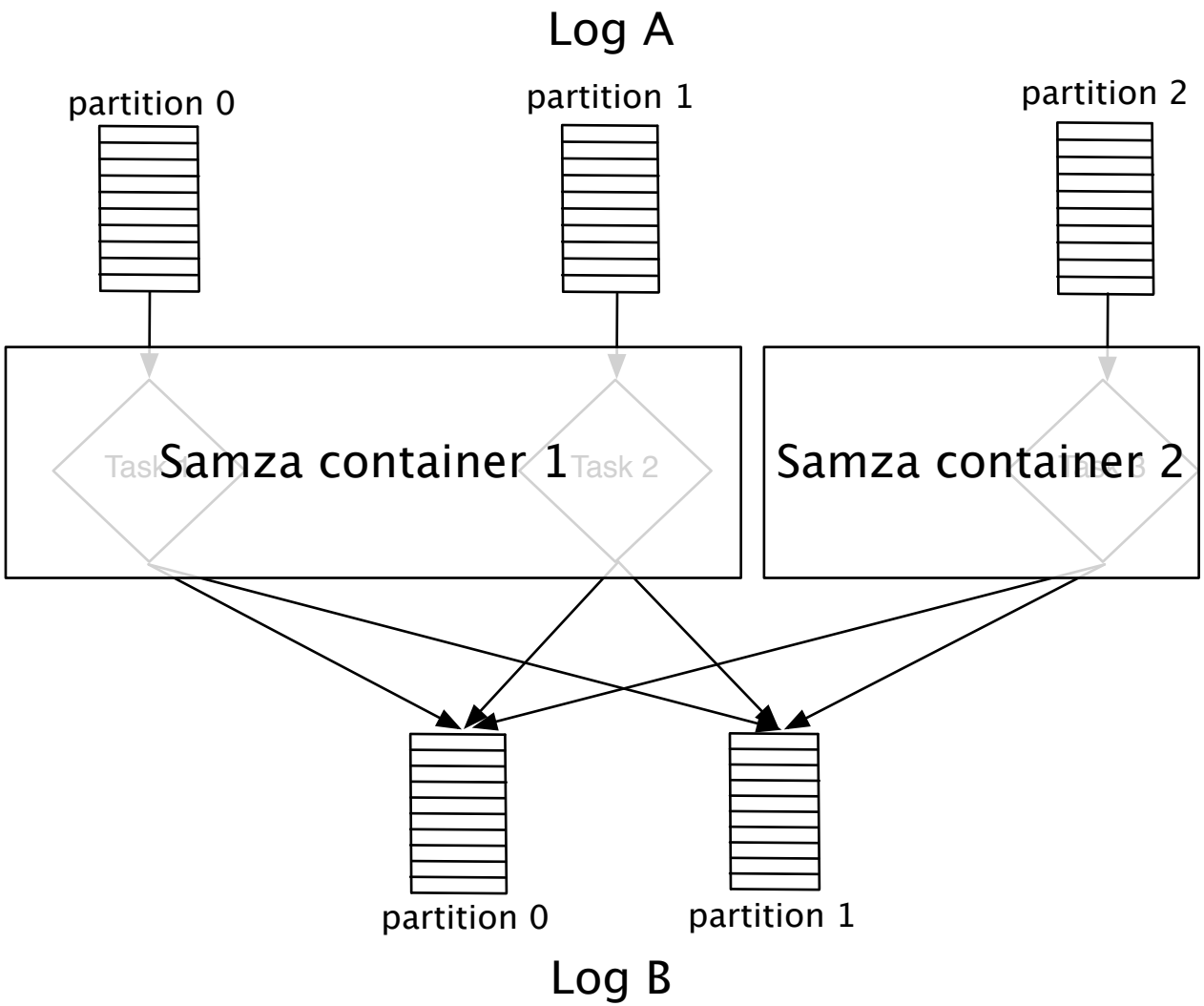
Samza API


getKey(), getMsg()
public interface StreamTask {
void process (IncomingMessageEnvelope envelope,
MessageCollector collector,
TaskCoordinator coordinator);
}
sendMsg(topic, key, value)
commit(), shutdown()

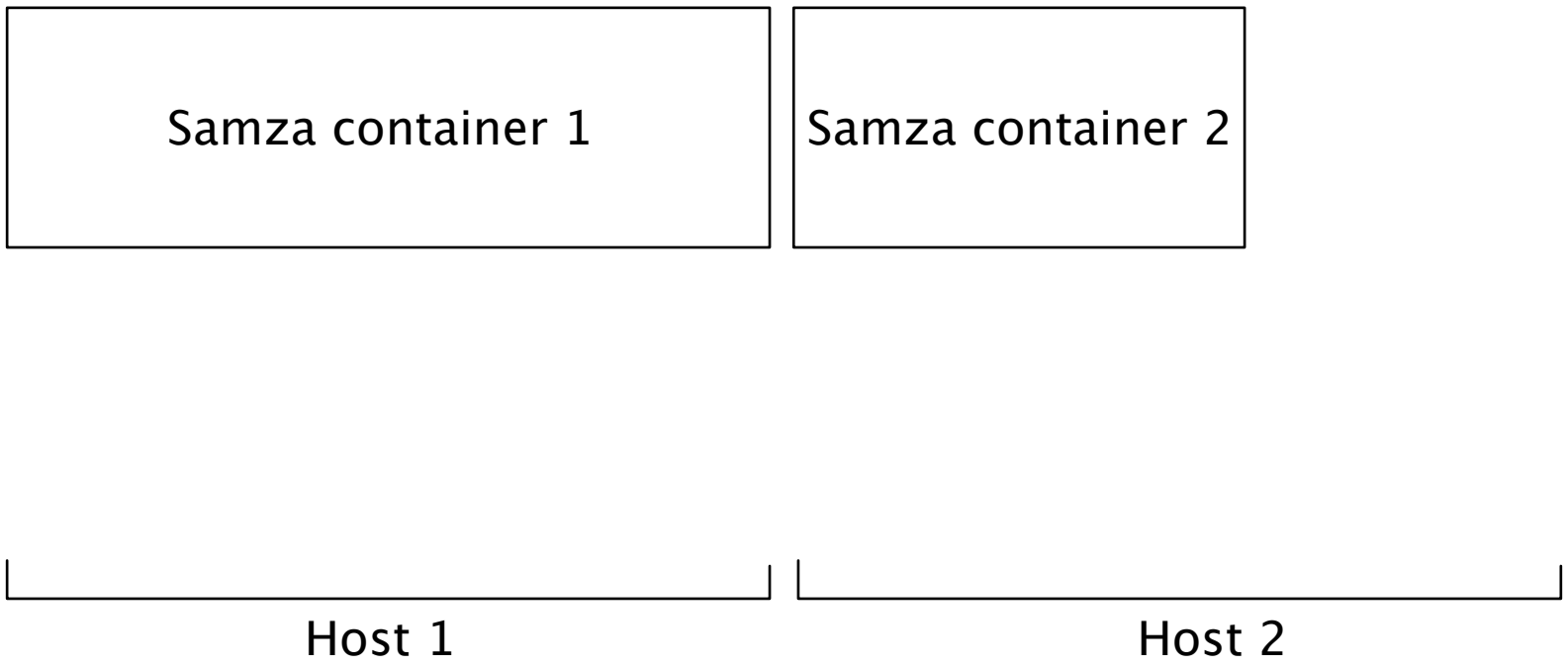
Samza Architecture (Logical view)



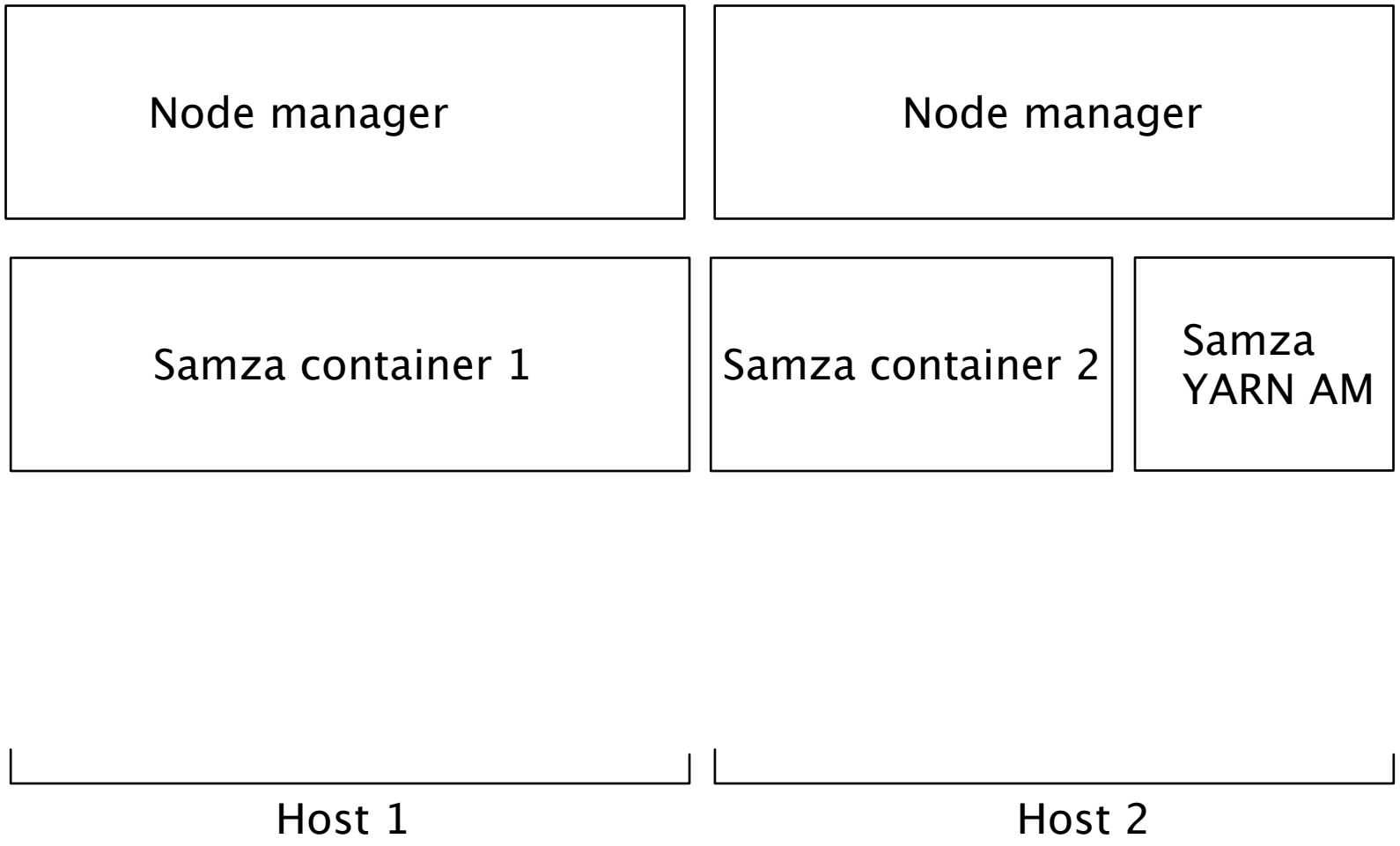
Samza Architecture (Logical view)



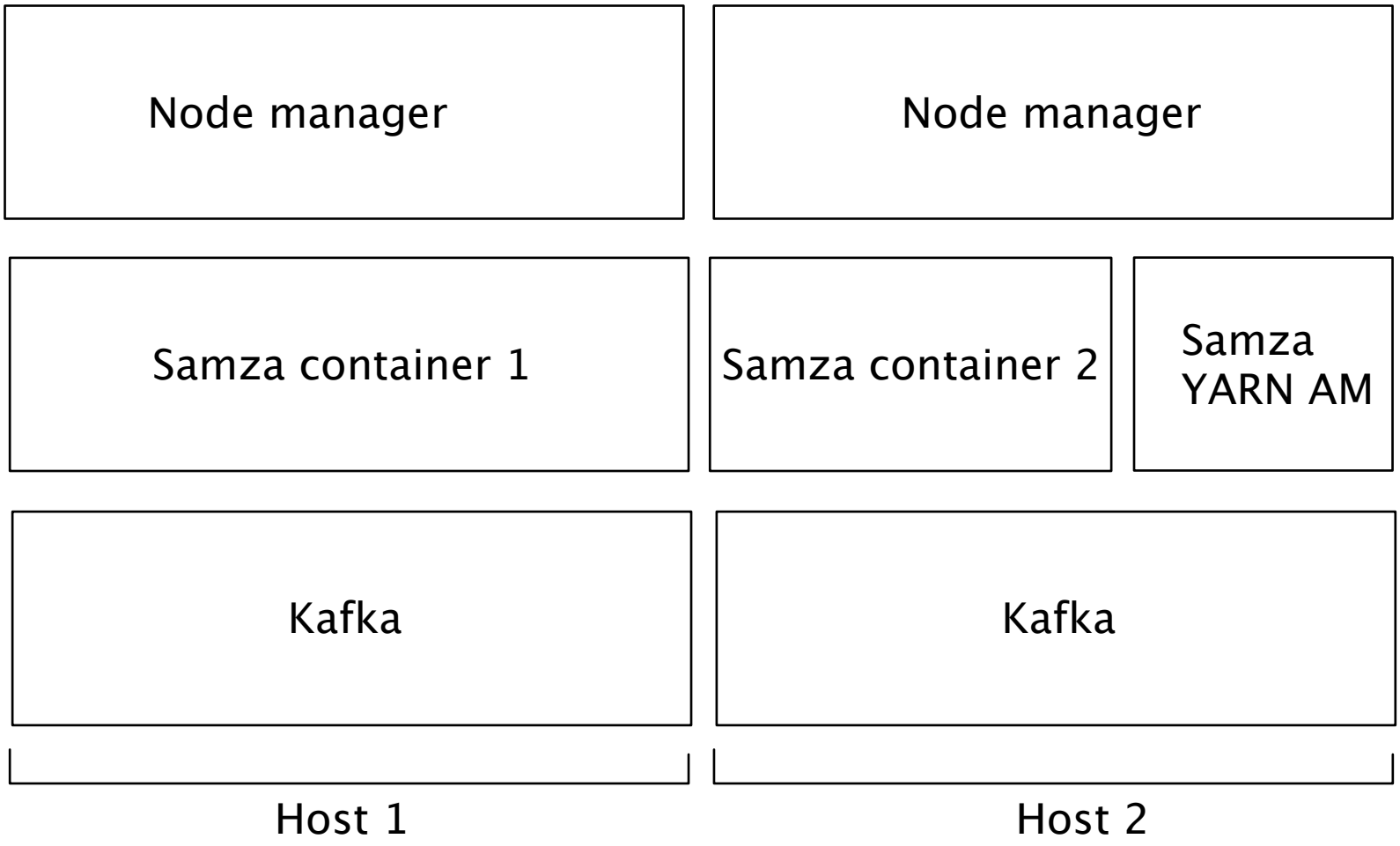
Samza Architecture (Physical view)



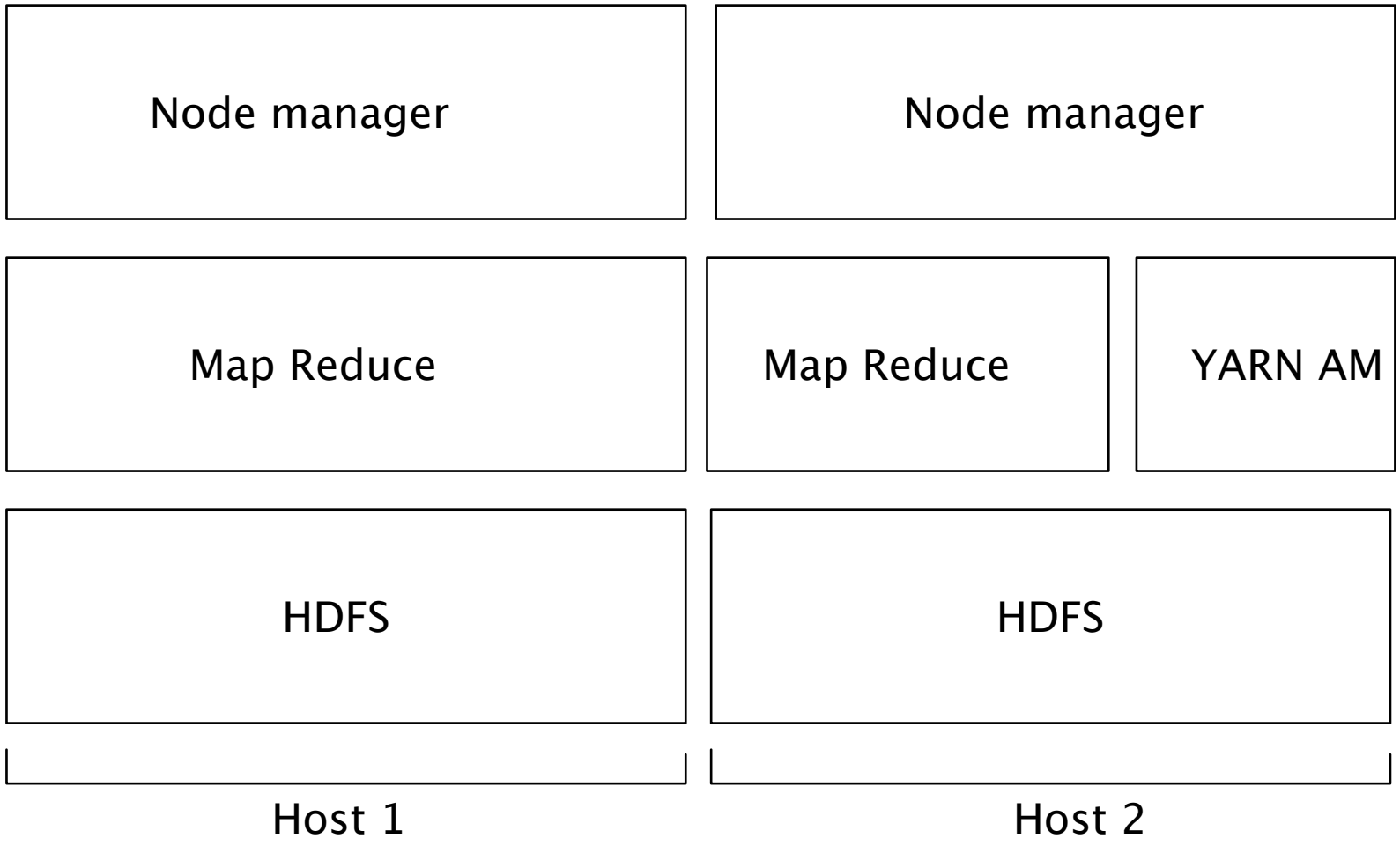
Samza Architecture (Physical view)



Samza Architecture (Physical view)



Samza Architecture: Equivalence to Map Reduce



M/R Operation Primitives

- Filter — records matching some condition
- Map — record = f(record)
- Join — Two/more datasets by key
- Group — records with same key
- Aggregate — f(records within the same group)
- Pipe — job 1's output => job 2's input

M/R Operation Primitives on streams

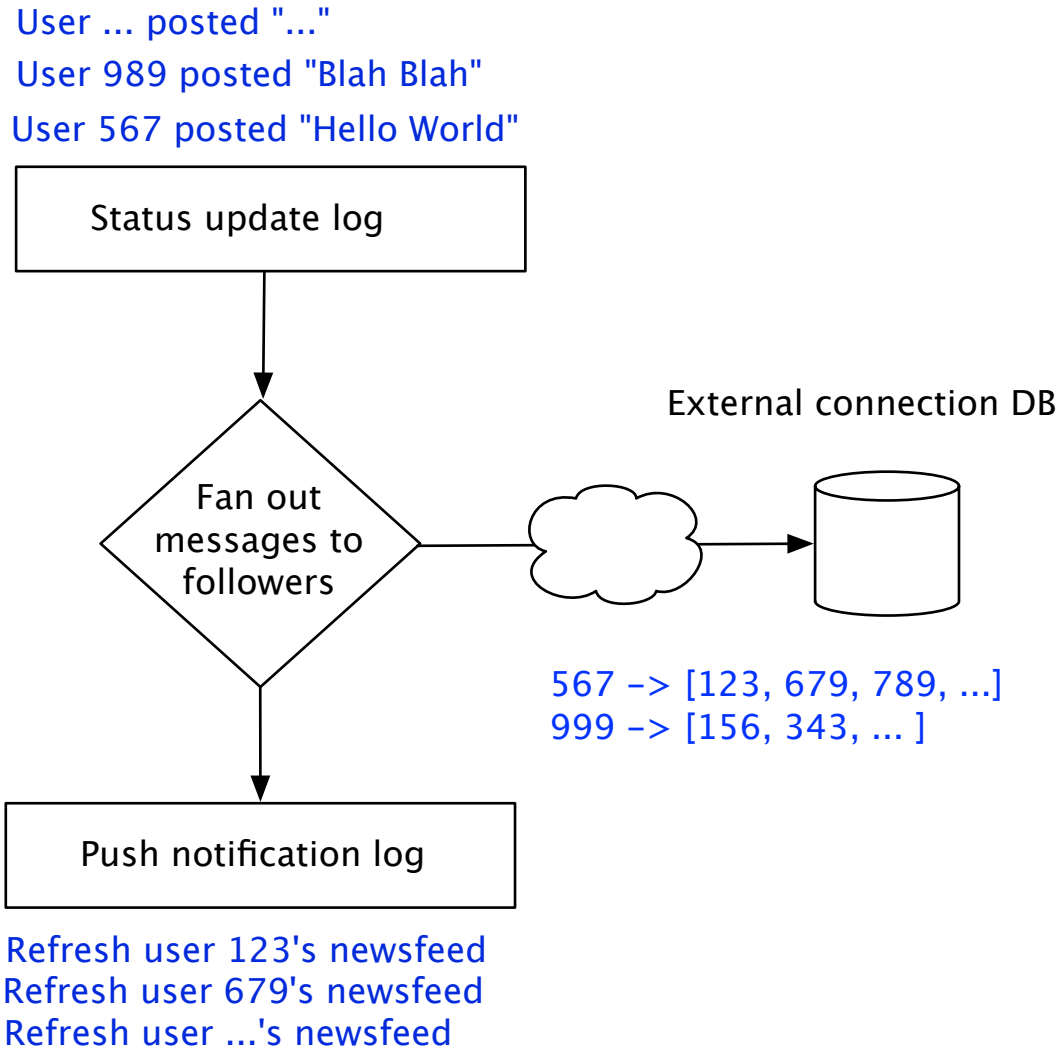
- Filter — records matching some condition
- Map — record = f(record)
- Join — Two/more datasets by key
- Group — records with same key
- Aggregate — f(records within the same group)
- Pipe — job 1's output => job 2's input

Requires state maintenance

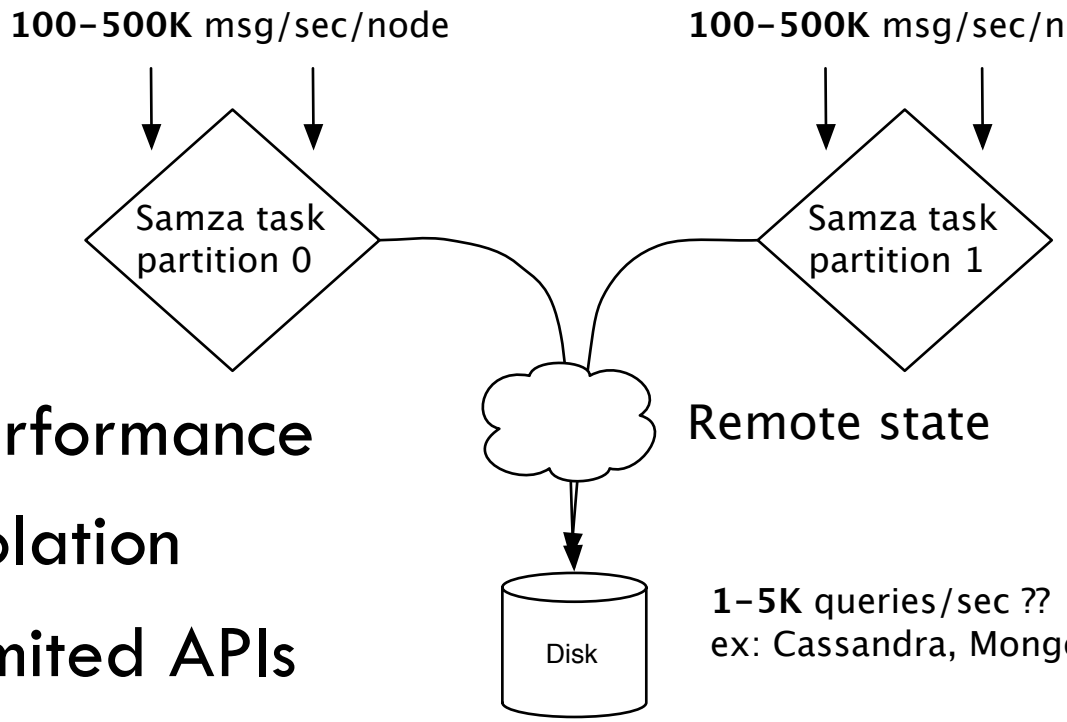
Agenda

- Real-time Data Integration
- Introduction to Logs & Apache Kafka
- Logs & Stream processing
- Apache Samza
- **Stateful stream processing**

Example: Newsfeed

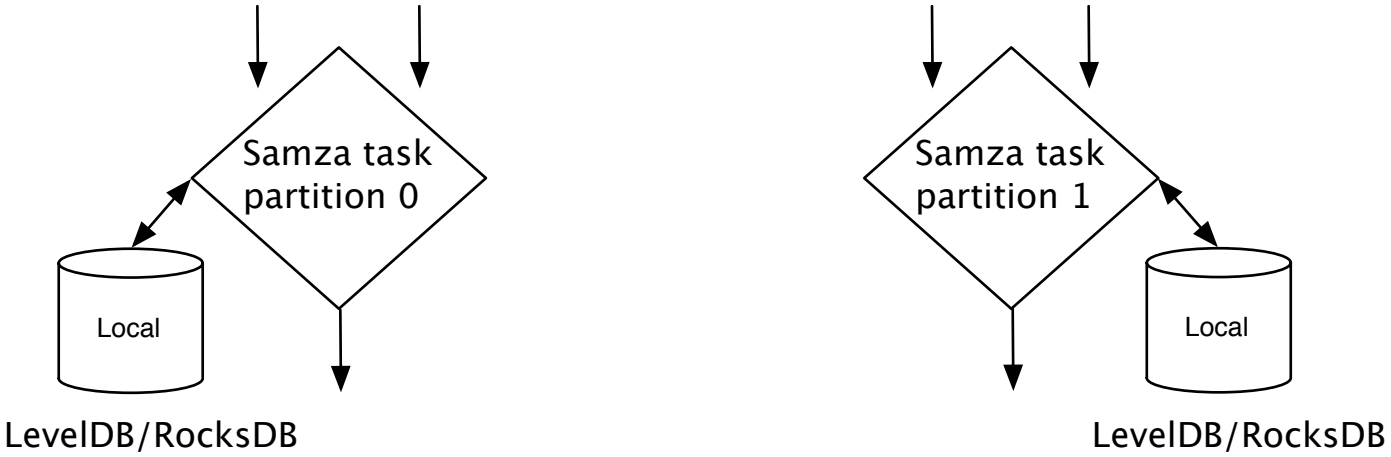


Local state vs Remote state: Remote

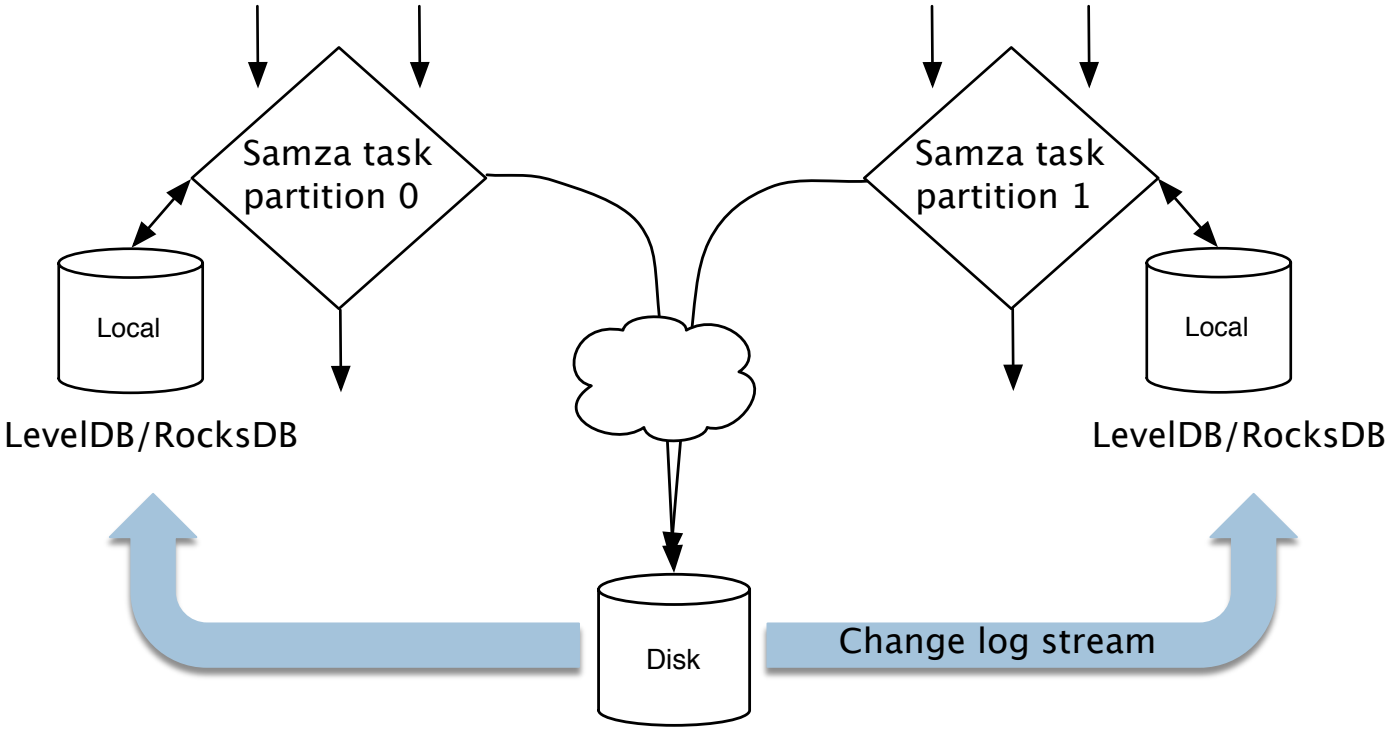


- ✗ Performance
- ✗ Isolation
- ✗ Limited APIs

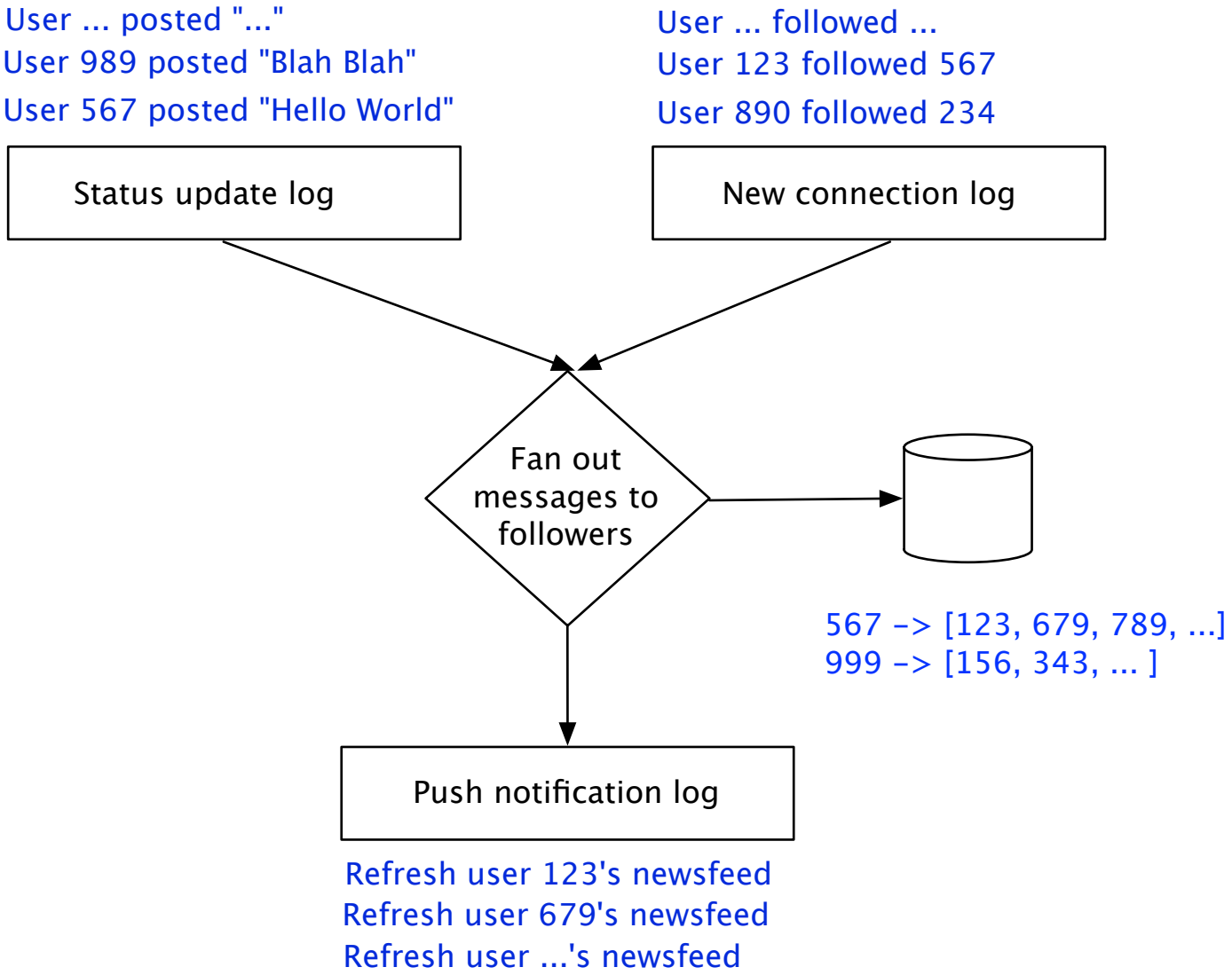
Local state: Bring data closer to computation



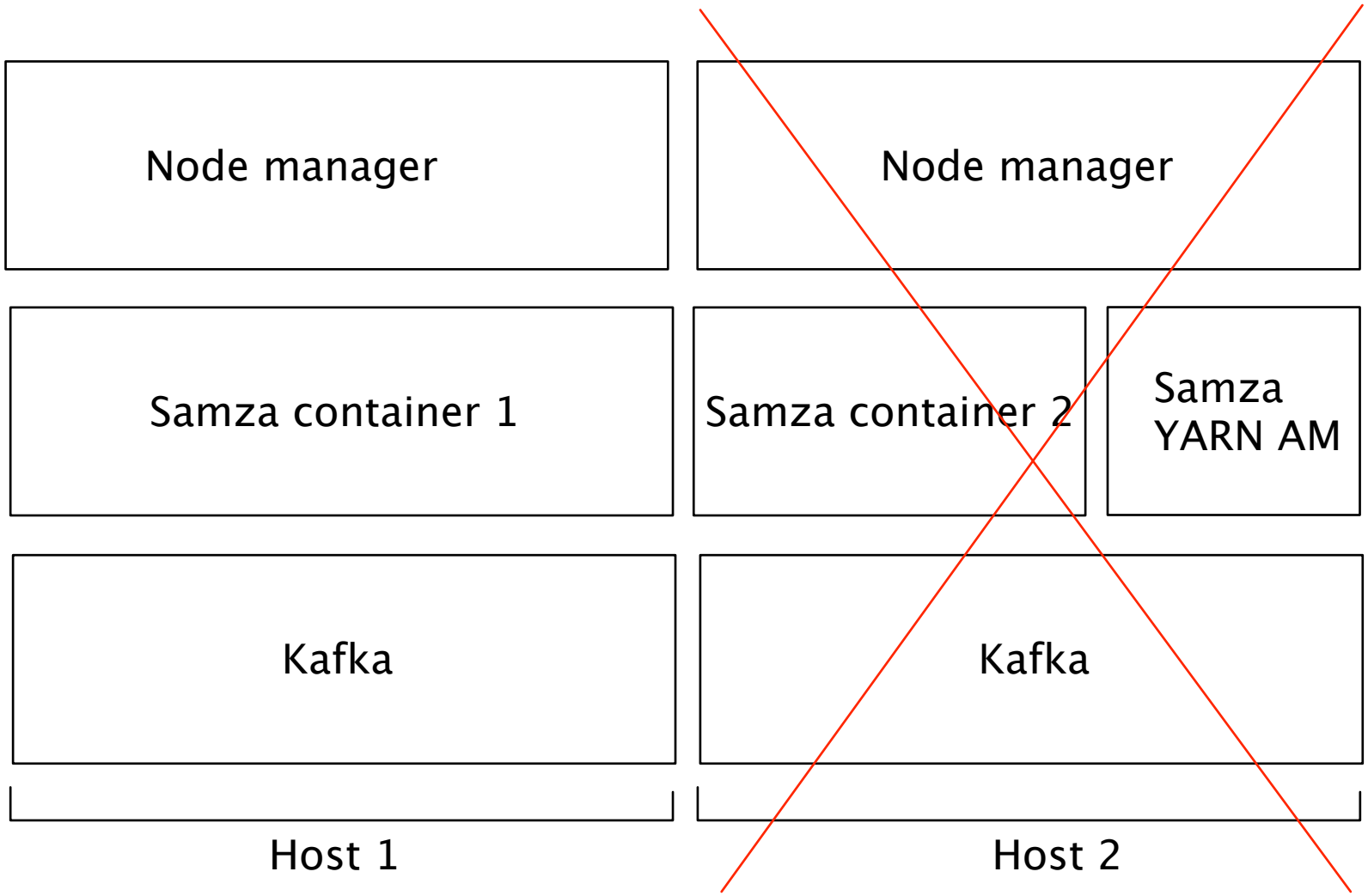
Local state: Bring data closer to computation



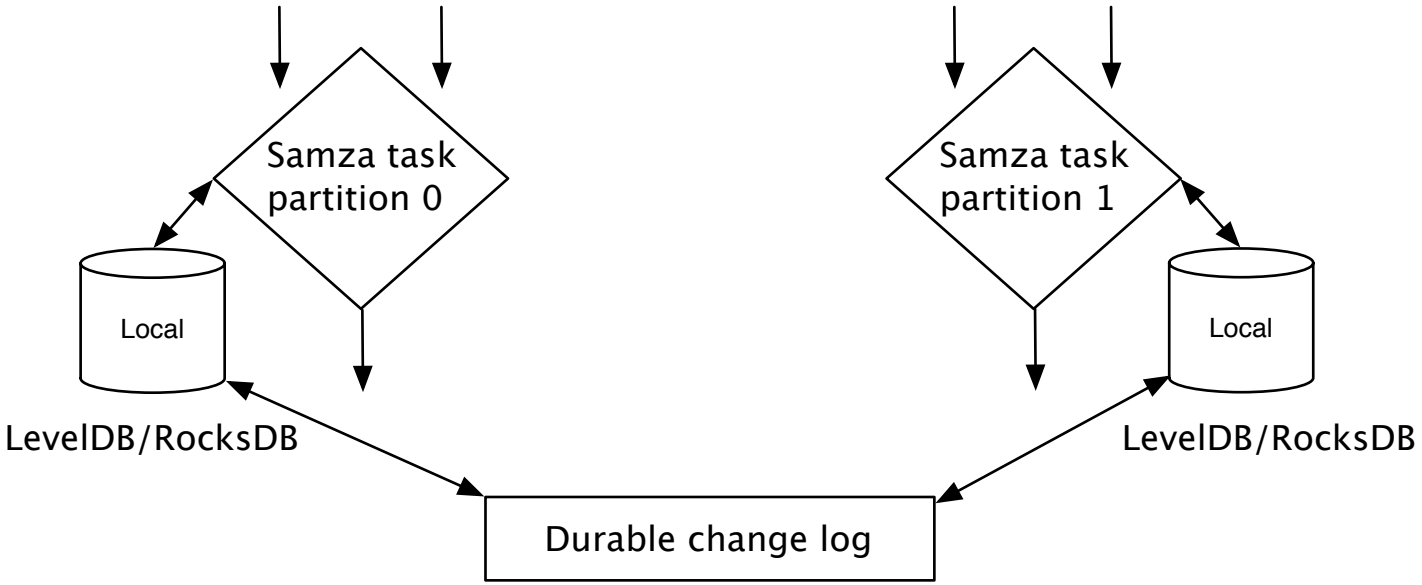
Example Revisited: Newsfeed



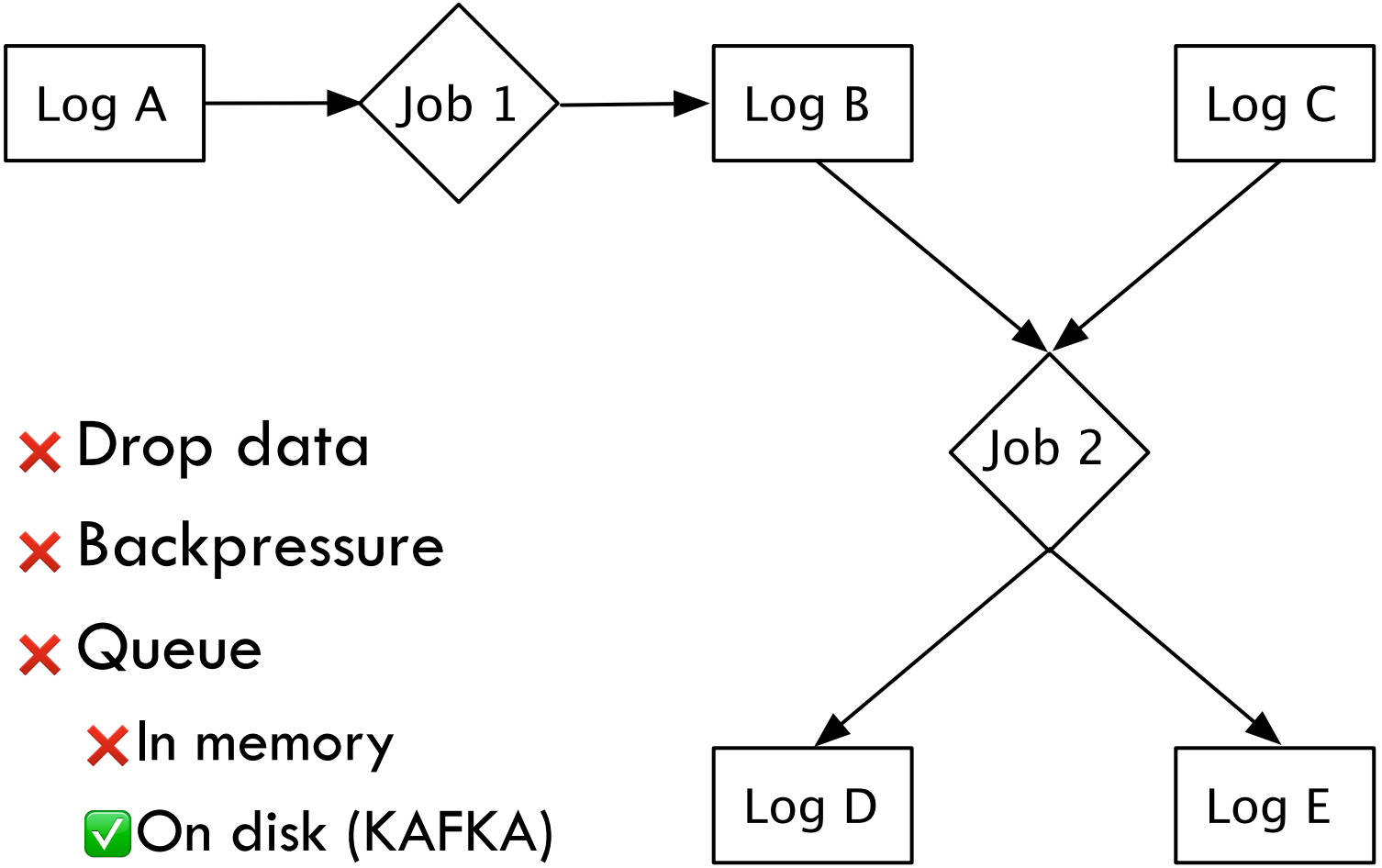
Fault tolerance?



Fault tolerance in Samza



Slow jobs



- ❌ Drop data
- ❌ Backpressure
- ❌ Queue
  - ❌ In memory
  - ✅ On disk (KAFKA)

Summary

- Real time data integration is crucial for the success and adoption of stream processing
- Logs form the basis for real time data integration
- Stream processing = f(logs)
- Samza is designed from ground-up for scalability and provides fault-tolerant, persistent state

Thank you!

- The Log
  - http://bit.ly/the_log
- Apache Kafka
  - http://kafka.apache.org
- Apache Samza
  - http://samza.incubator.apache.org
- Me
  - @nehanarkhede
  - http://www.linkedin.com/in/nehanarkhede