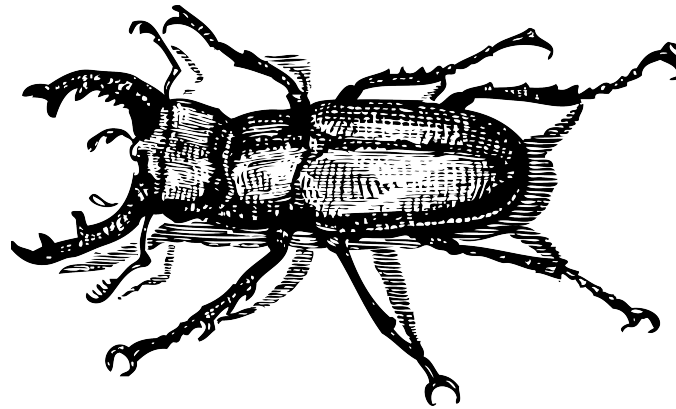


# I ♥ Logs

Apache Kafka, Stream Processing, and  
Real-time Data

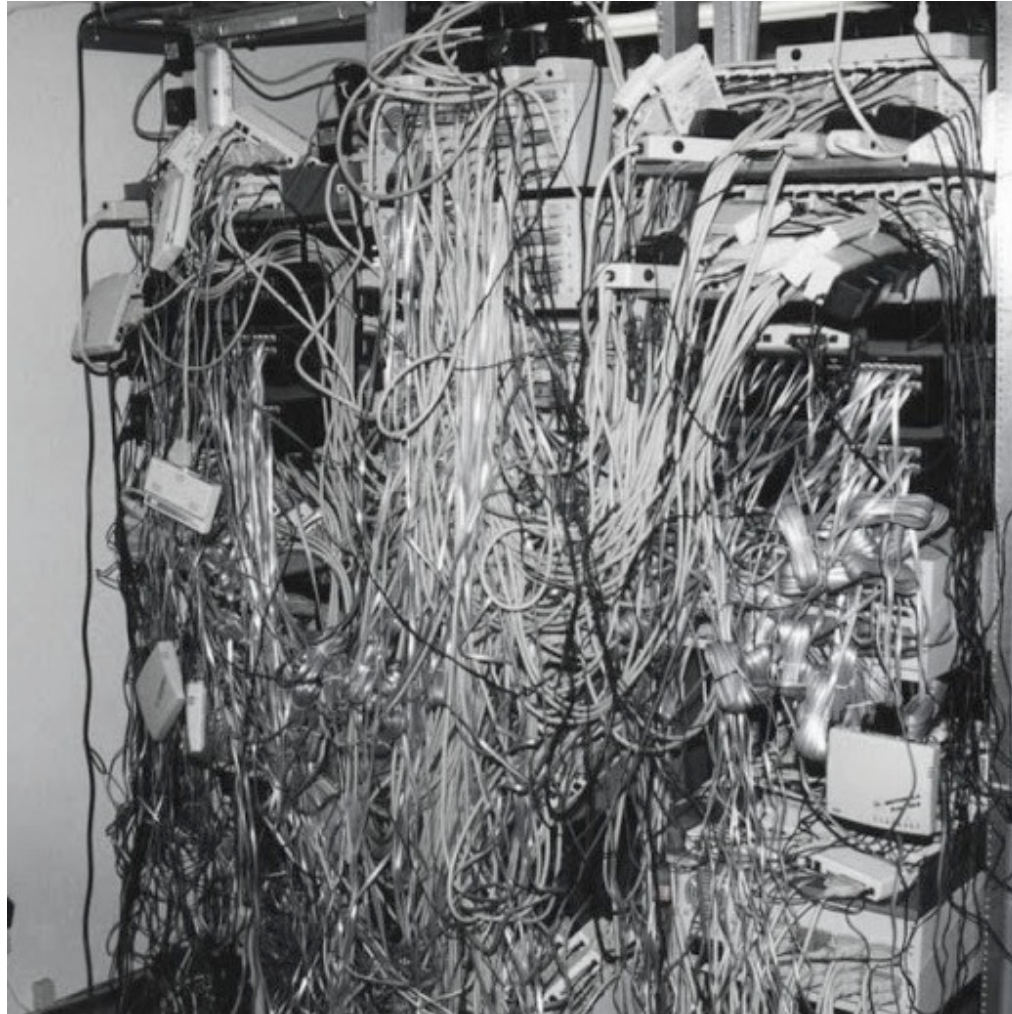
Jay Kreps



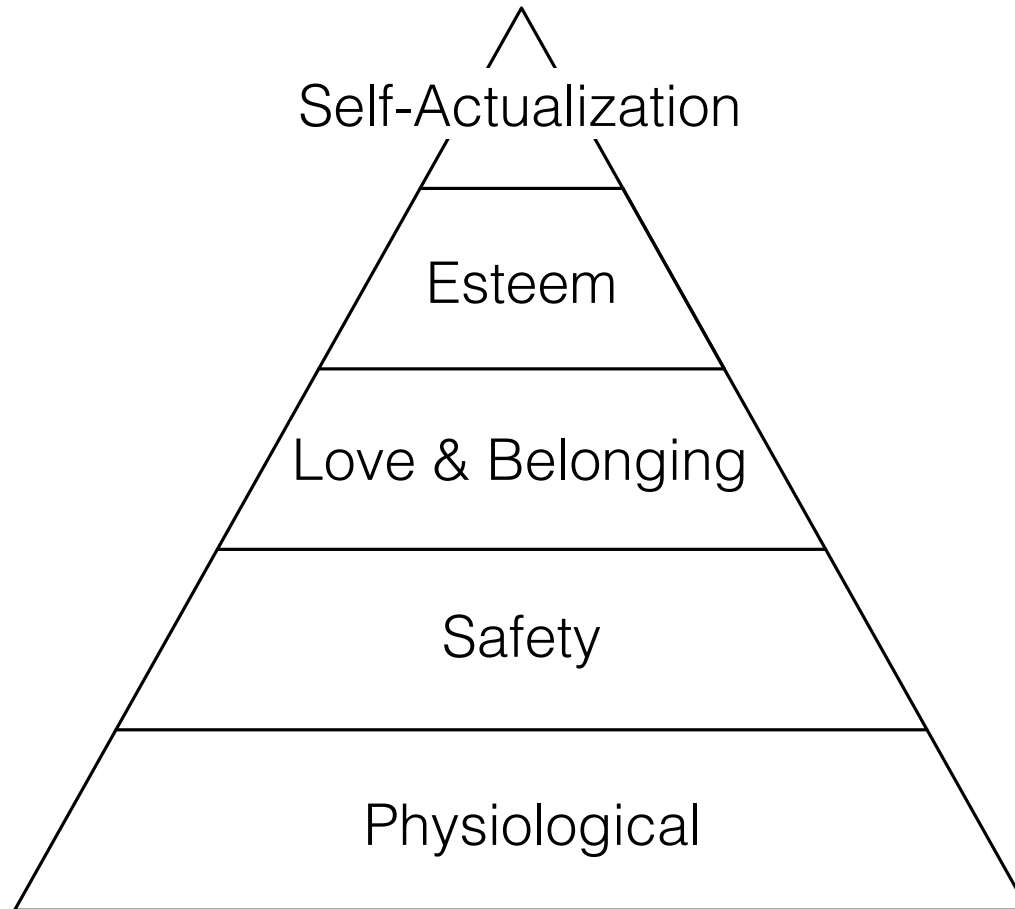
# The Plan

1. What is Data Integration?
2. What is Apache Kafka?
3. Logs and Distributed Systems
4. Logs and Data Integration
5. Logs and Stream Processing

# Data Integration

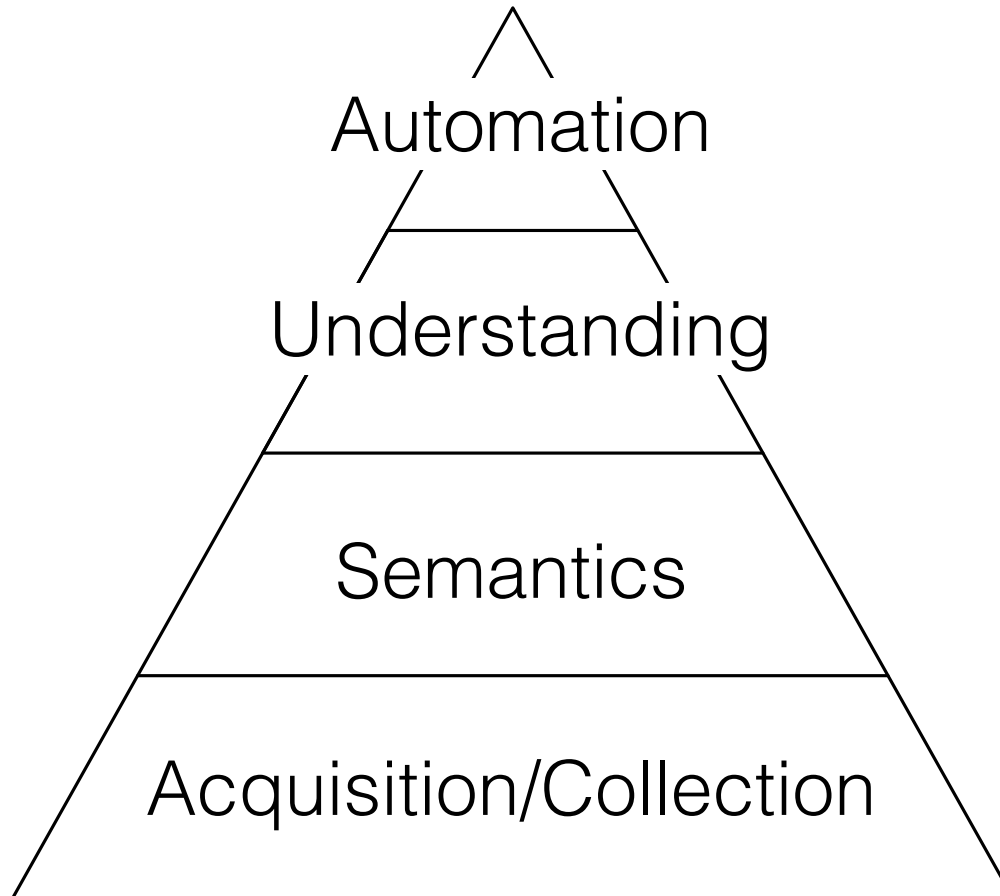


# Maslow's Hierarchy





# For Data



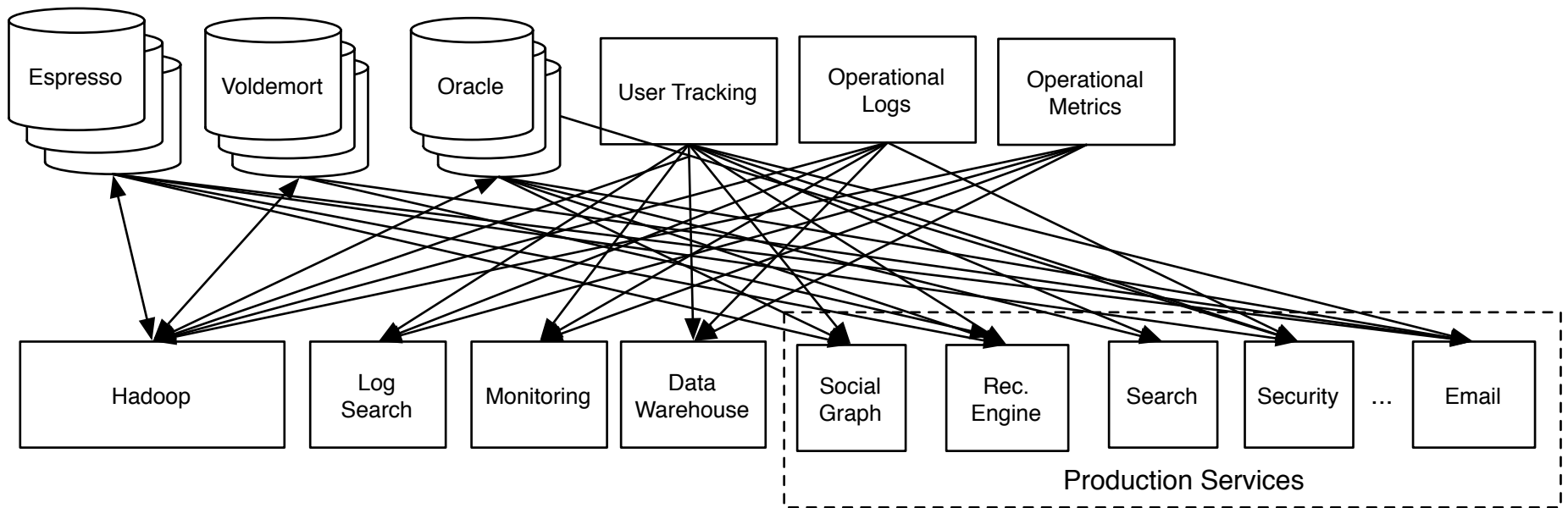
# New Types of Data

- Database data
  - Users, products, orders, etc
- Events
  - Clicks, Impressions, Pageviews, etc
- Application metrics
  - CPU usage, requests/sec
- Application logs
  - Service calls, errors

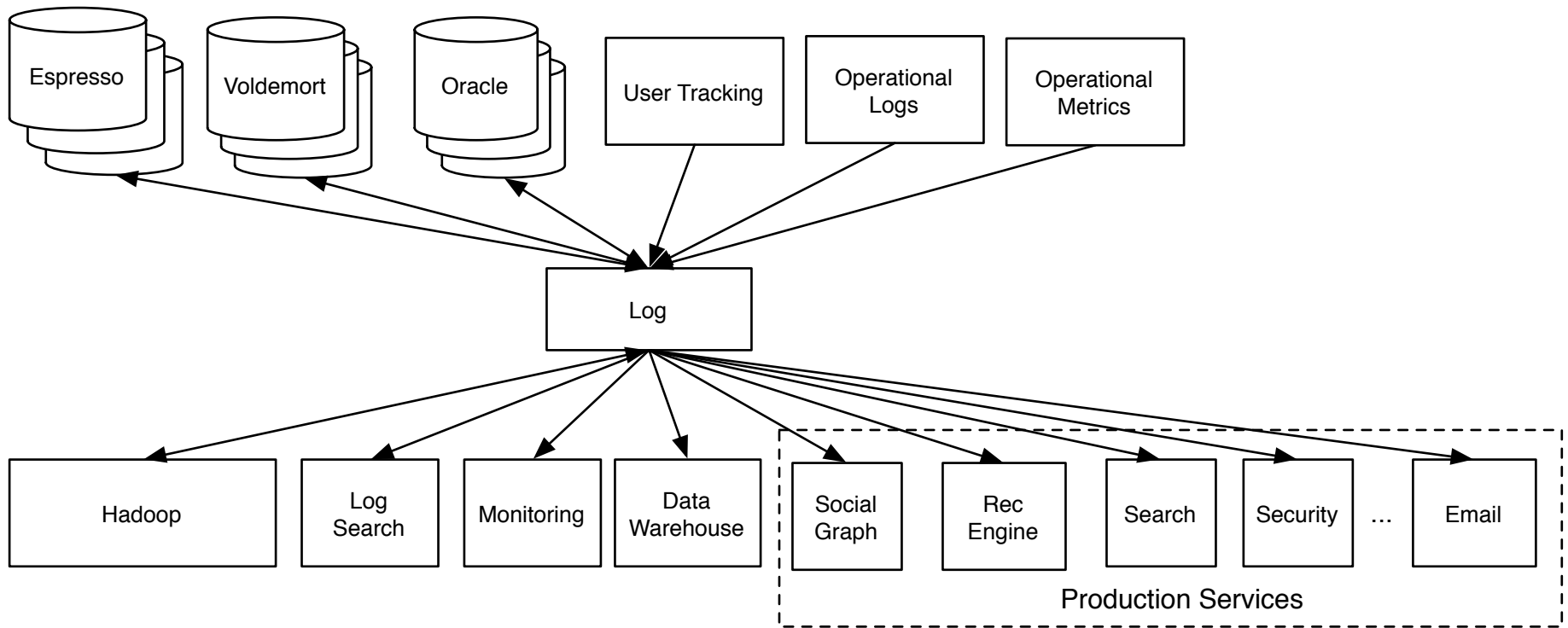
# New Types of Systems

- Live Stores
  - Voldemort
  - Espresso
  - Graph
  - OLAP
  - Search
  - InGraphs
- Offline
  - Hadoop
  - Teradata

# Bad



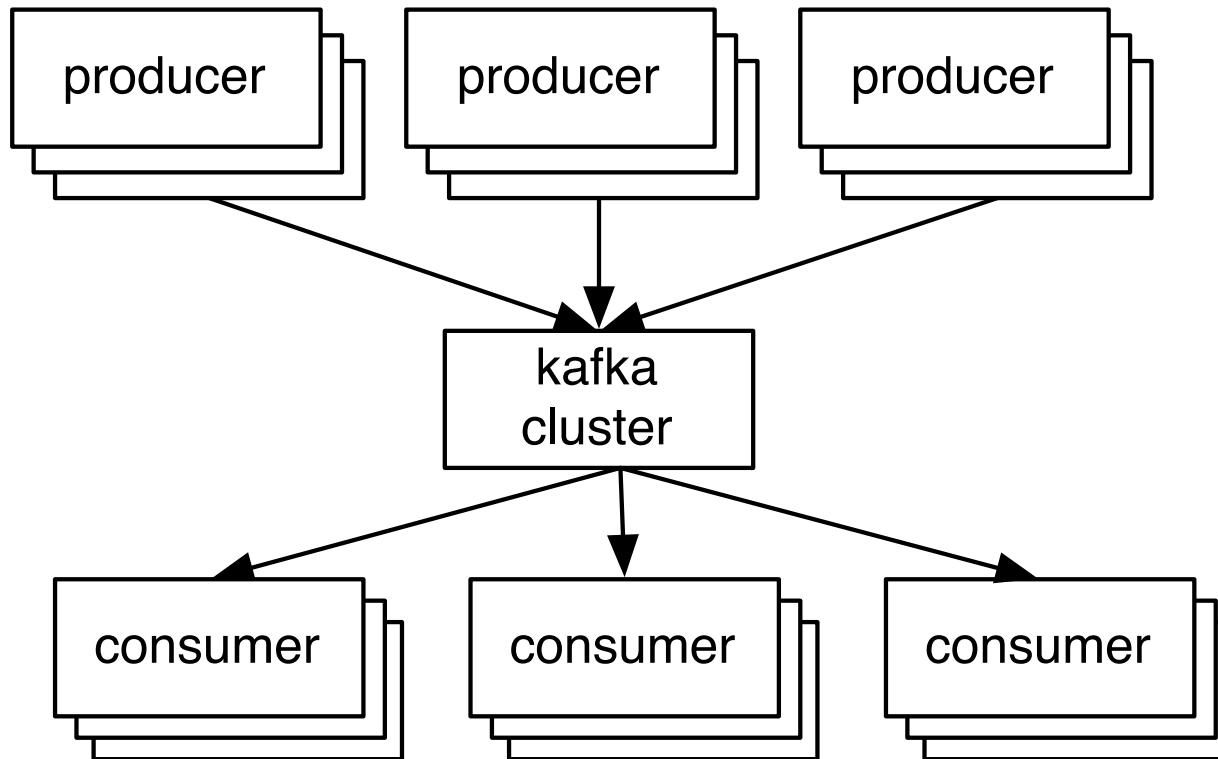
# Good



# The Plan

1. What is Data Integration?
2. What is Apache Kafka?
3. Logs and Distributed Systems
4. Logs and Data Integration
5. Logs and Stream Processing

# Apache Kafka



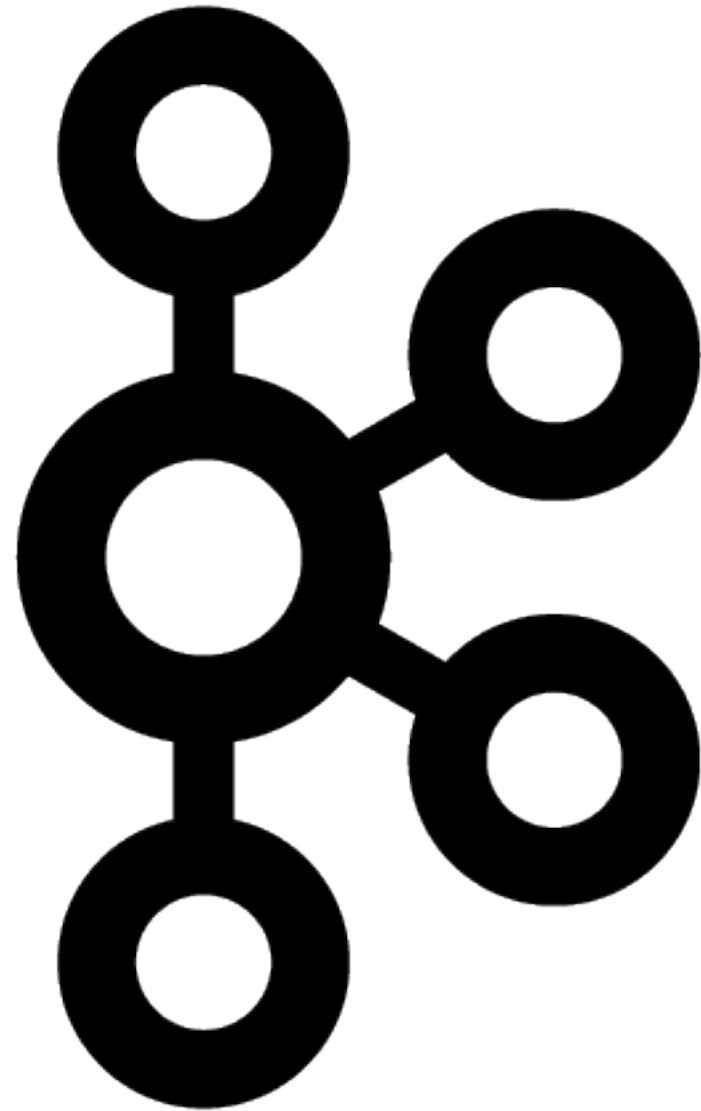
# Amazon Kinesis



**amazon**  
web services



A  
brief  
history  
of  
Kafka



# Three design principles

1. One pipeline to rule them all
2. Stream processing >> messaging
3. Clusters not servers

# Characteristics

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

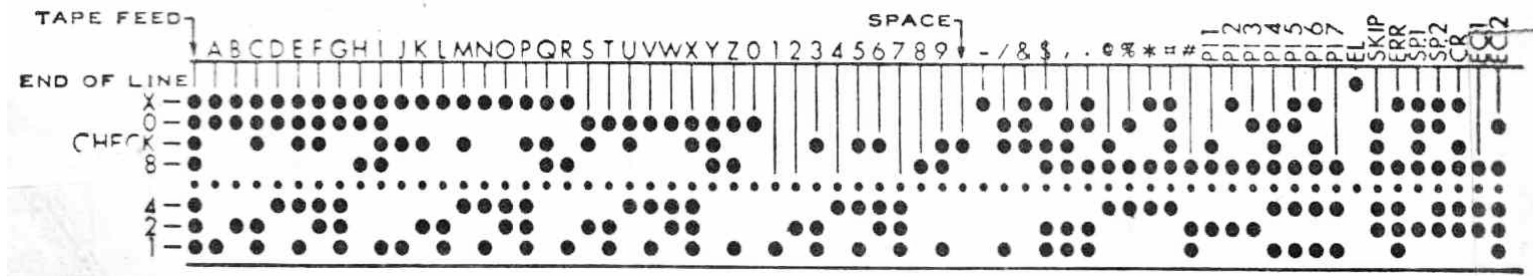
# Kafka At LinkedIn

- 175 TB of in-flight log data per colo
- Low-latency: ~1.5 ms
- 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

# The Plan

1. What is Data Integration?
2. What is Apache Kafka?
3. Logs and Distributed Systems
4. Logs and Data Integration
5. Logs and Stream Processing

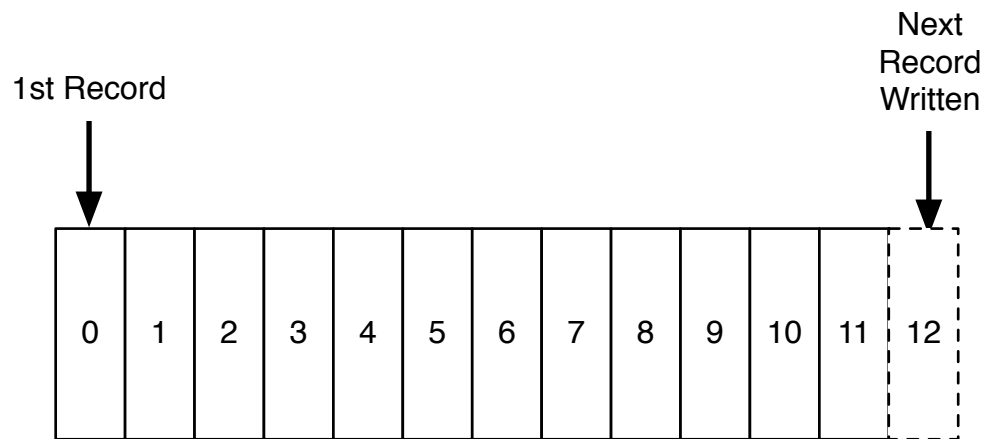
# Kafka is about logs



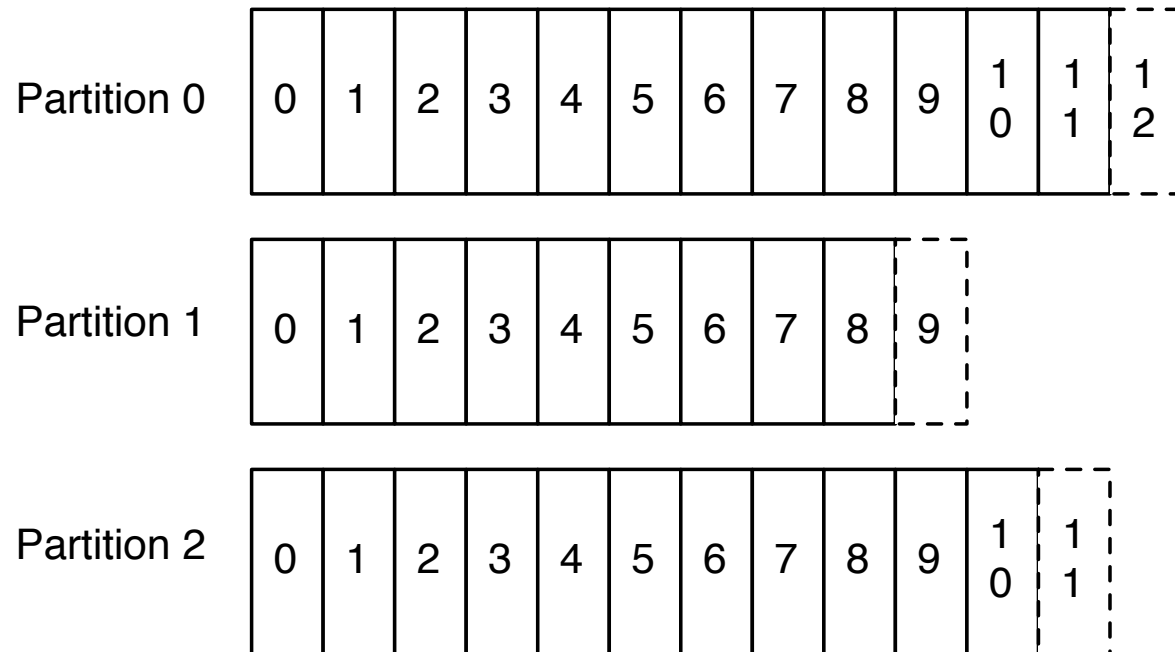
What is a log?

```
jkreps-mn:~ jkreps$ tail -f -n 20 /var/log/apache2/access_log
::1 - - [23/Mar/2014:15:07:00 -0700] "GET /images/apache_feather.gif HTTP/1.1" 200 4128
::1 - - [23/Mar/2014:15:07:04 -0700] "GET /images/producer_consumer.png HTTP/1.1" 200 86
::1 - - [23/Mar/2014:15:07:04 -0700] "GET /images/log_anatomy.png HTTP/1.1" 200 19579
::1 - - [23/Mar/2014:15:07:04 -0700] "GET /images/consumer-groups.png HTTP/1.1" 200 268
::1 - - [23/Mar/2014:15:07:04 -0700] "GET /images/log_compaction.png HTTP/1.1" 200 4141
::1 - - [23/Mar/2014:15:07:04 -0700] "GET /documentation.html HTTP/1.1" 200 189893
::1 - - [23/Mar/2014:15:07:04 -0700] "GET /images/log_cleaner_anatomy.png HTTP/1.1" 200
::1 - - [23/Mar/2014:15:07:04 -0700] "GET /images/kafka_log.png HTTP/1.1" 200 134321
::1 - - [23/Mar/2014:15:07:04 -0700] "GET /images/mirror-maker.png HTTP/1.1" 200 17054
::1 - - [23/Mar/2014:15:08:07 -0700] "GET /documentation.html HTTP/1.1" 200 189937
::1 - - [23/Mar/2014:15:08:07 -0700] "GET /styles.css HTTP/1.1" 304 -
::1 - - [23/Mar/2014:15:08:07 -0700] "GET /images/kafka_logo.png HTTP/1.1" 304 -
::1 - - [23/Mar/2014:15:08:07 -0700] "GET /images/producer_consumer.png HTTP/1.1" 304 -
::1 - - [23/Mar/2014:15:08:07 -0700] "GET /images/log_anatomy.png HTTP/1.1" 304 -
::1 - - [23/Mar/2014:15:08:07 -0700] "GET /images/consumer-groups.png HTTP/1.1" 304 -
::1 - - [23/Mar/2014:15:08:07 -0700] "GET /images/log_cleaner_anatomy.png HTTP/1.1" 304
::1 - - [23/Mar/2014:15:08:07 -0700] "GET /images/log_compaction.png HTTP/1.1" 304 -
::1 - - [23/Mar/2014:15:08:07 -0700] "GET /images/kafka_log.png HTTP/1.1" 304 -
::1 - - [23/Mar/2014:15:08:07 -0700] "GET /images/mirror-maker.png HTTP/1.1" 304 -
::1 - - [23/Mar/2014:15:09:55 -0700] "GET /documentation.html HTTP/1.1" 200 195264
```

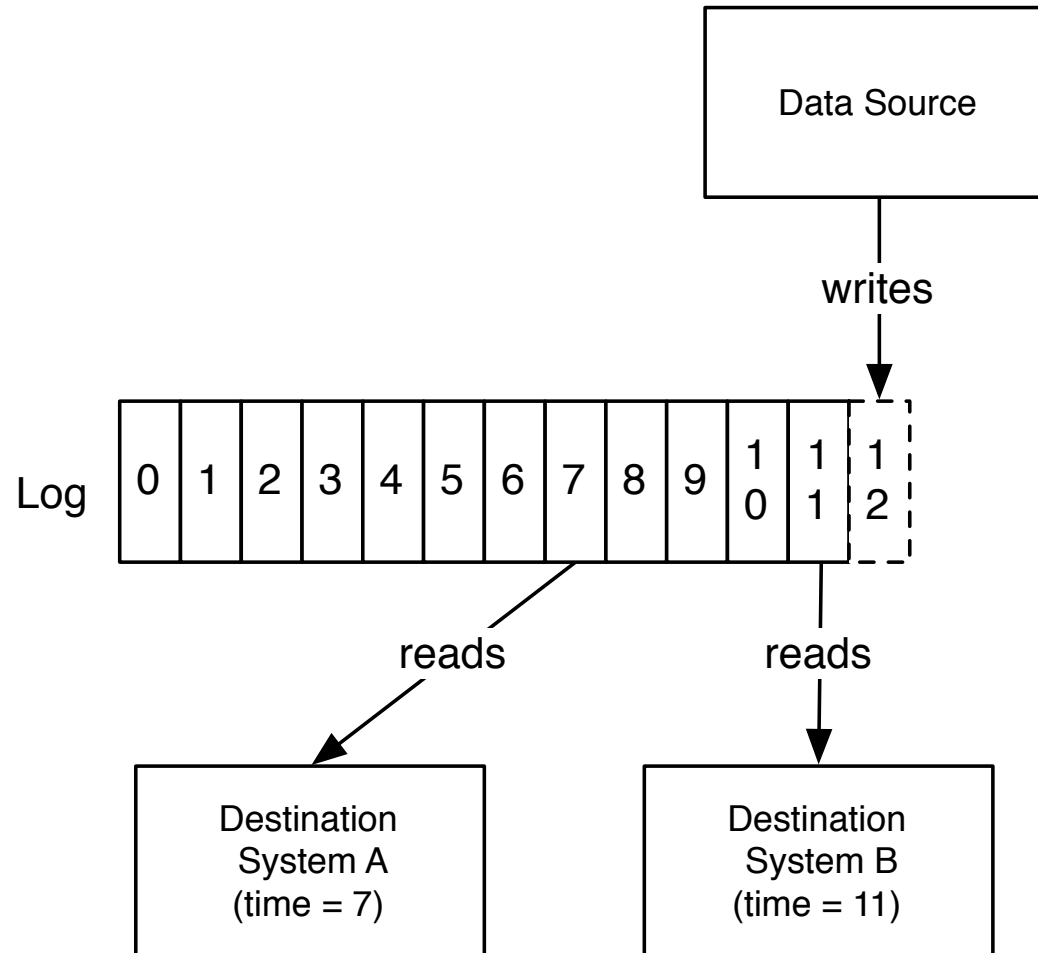




# Partitioning



# Logs: pub/sub done right



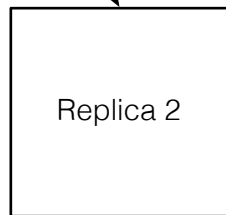
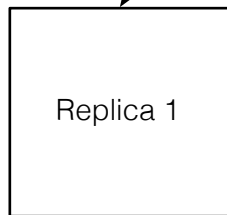
# Logs And Distributed Systems



Example:  
A Fault-tolerant CEO Hash Table

# Operations

```
PUT('microsoft', 'bill gates')  
PUT('apple', 'steve jobs')  
PUT('microsoft', 'steve ballmer')  
PUT('google', 'larry page')  
PUT('yahoo', 'terry semel')  
PUT('google', 'eric schmidt')  
PUT('yahoo', 'jerry yang')  
PUT('yahoo', 'carol bartz')  
PUT('apple', 'tim cook')  
PUT('google', 'larry page')  
PUT('yahoo', 'scott thompson')  
PUT('yahoo', 'marissa mayer')  
PUT('microsoft', 'satya nadella')
```



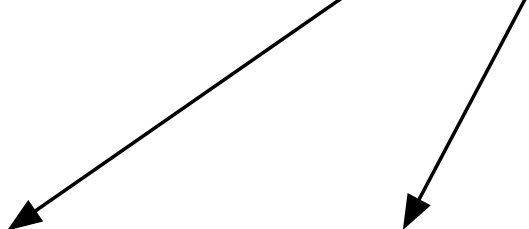
# Final State

```
{  
  'microsoft': 'satya nadella',  
  'apple': 'tim cook',  
  'google': 'larry page',  
  'yahoo': 'marissa mayer'  
}
```

0	PUT(microsoft, bill gates)
1	PUT(apple, steve jobs)
2	PUT(microsoft, steve ballmer)
3	PUT(google, larry page)
4	PUT(yahoo, terry semel)
5	PUT(google, eric schmidt)
6	PUT(yahoo, jerry yang)
7	PUT(yahoo, carol bartz)
8	PUT(apple, tim cook)
9	PUT(google, larry page)
10	PUT(yahoo, scott thompson)
11	PUT(yahoo, marissa mayer)
12	PUT(microsoft, satya nadella)

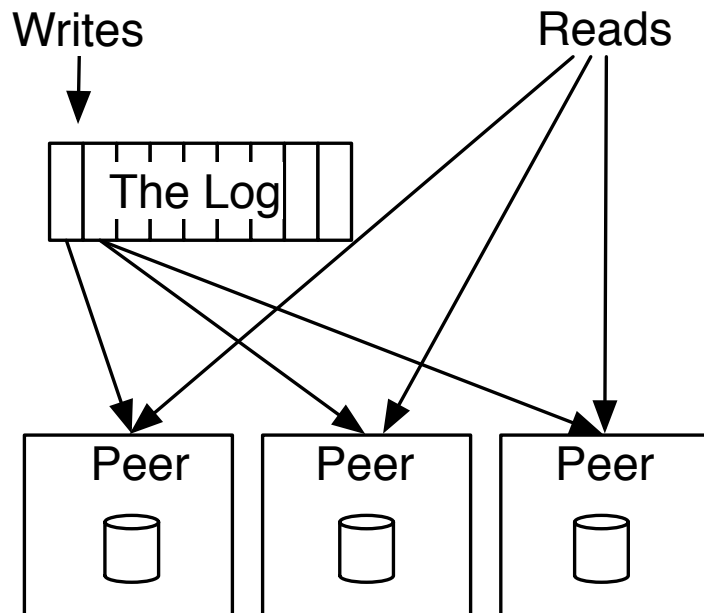
Replica 1  
(offset=10)

Replica 2  
(offset=12)

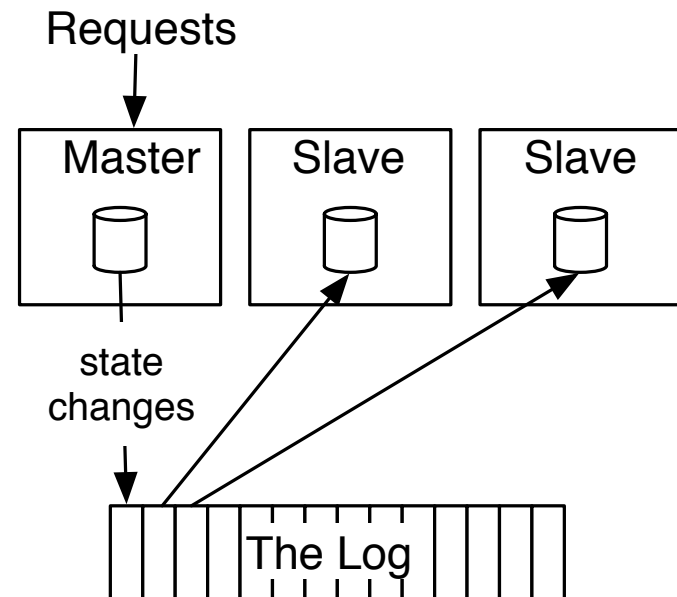


# Two System Design Styles

## State-machine Replication



## Primary-backup

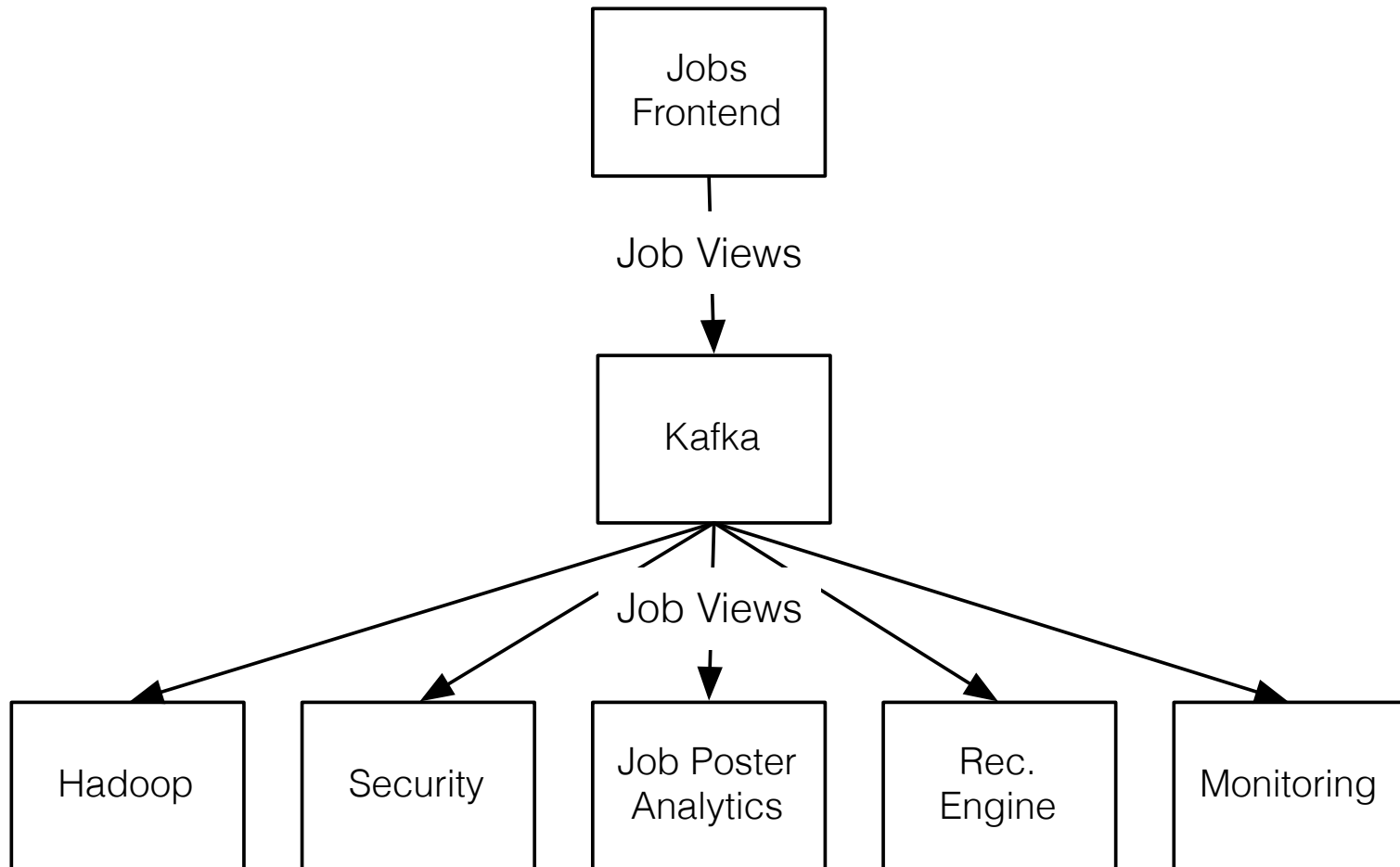




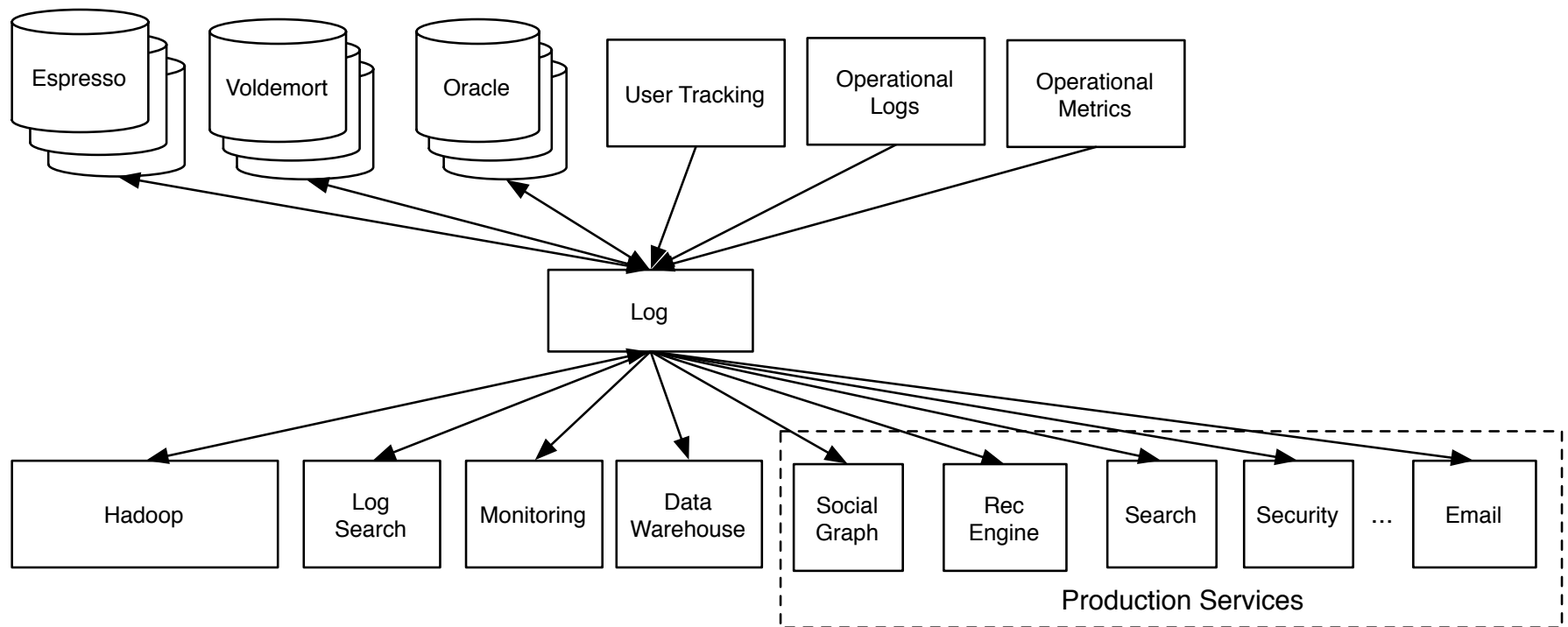
# The Plan

1. What is Data Integration?
2. What is Apache Kafka?
3. Logs and Distributed Systems
4. Logs and Data Integration
5. Logs and Stream Processing

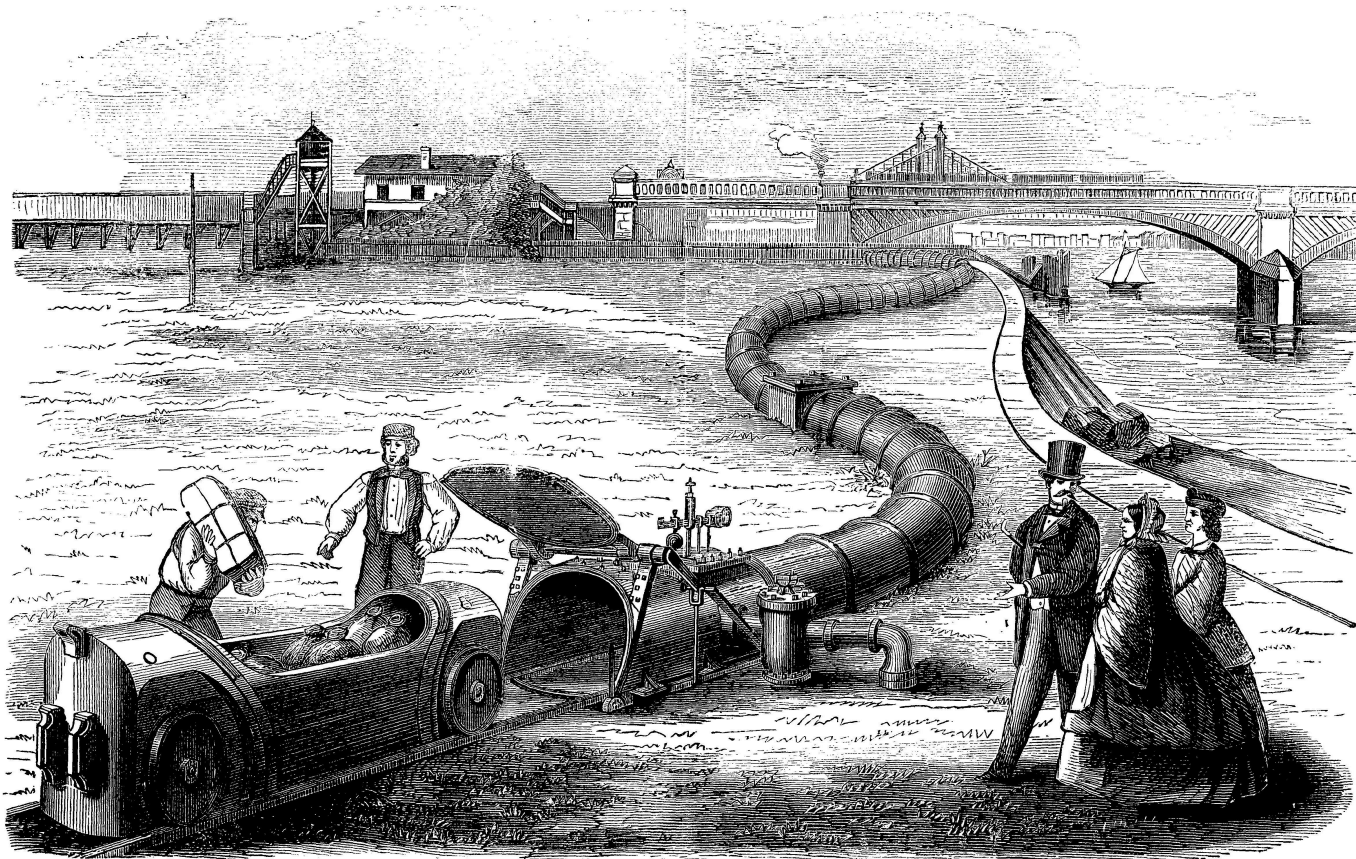
# Example: User views job



# It's all one big distributed system



# Comparing Data Transfer Mechanisms

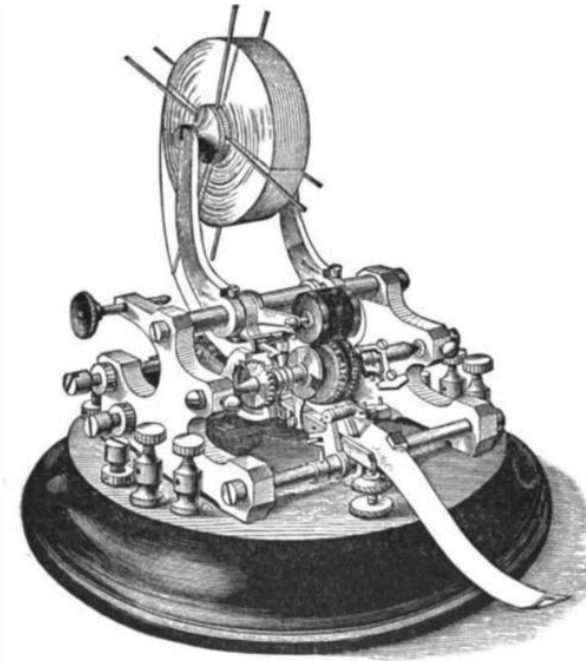


*Fig. 1.—The Pneumatic Dispatch—as first operated at Battersea Fields, London, in 1861.*

# The Plan

1. What is Data Integration?
2. What is Apache Kafka?
3. Logs and Distributed Systems
4. Logs and Data Integration
5. Logs and Stream Processing

# Stream Processing



R E T U R N

OF THE WHOLE

NUMBER OF PERSONS

WITHIN THE

SEVERAL DISTRICTS

OF THE

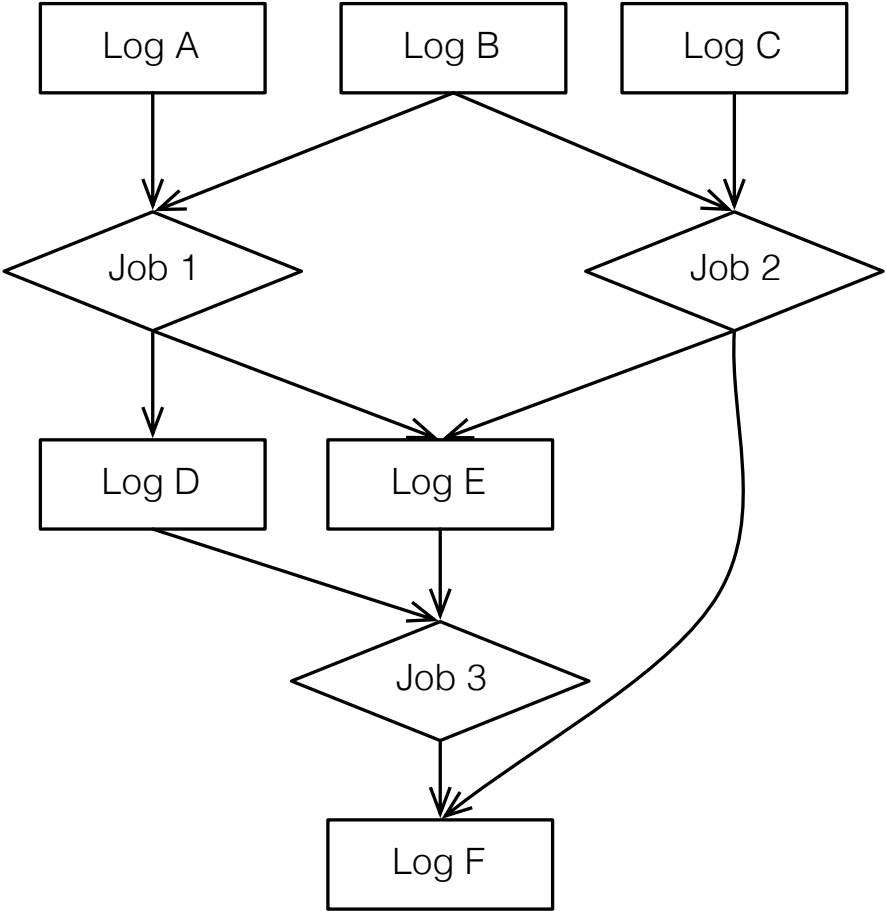
UNITED STATES,

ACCORDING TO

*“AN ACT PROVIDING FOR THE ENUMERATION OF  
THE INHABITANTS OF THE UNITED STATES;”*

PASSED MARCH THE FIRST, ONE THOUSAND SEVEN  
HUNDRED AND NINETY-ONE.

# Stream Processing = Logs + Jobs





Stream processing is a  
*generalization*  
of batch processing

# Examples

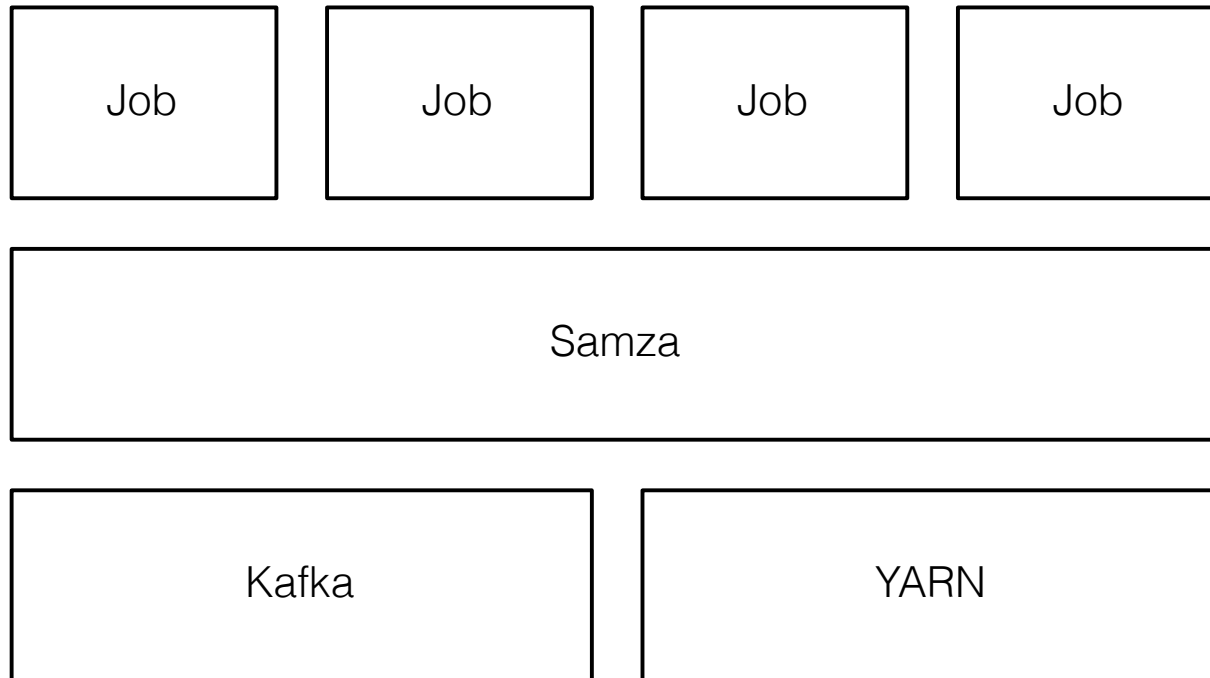
- Monitoring
- Security
- Content processing
- Recommendations
- Newsfeed
- ETL

# Systems Can Help

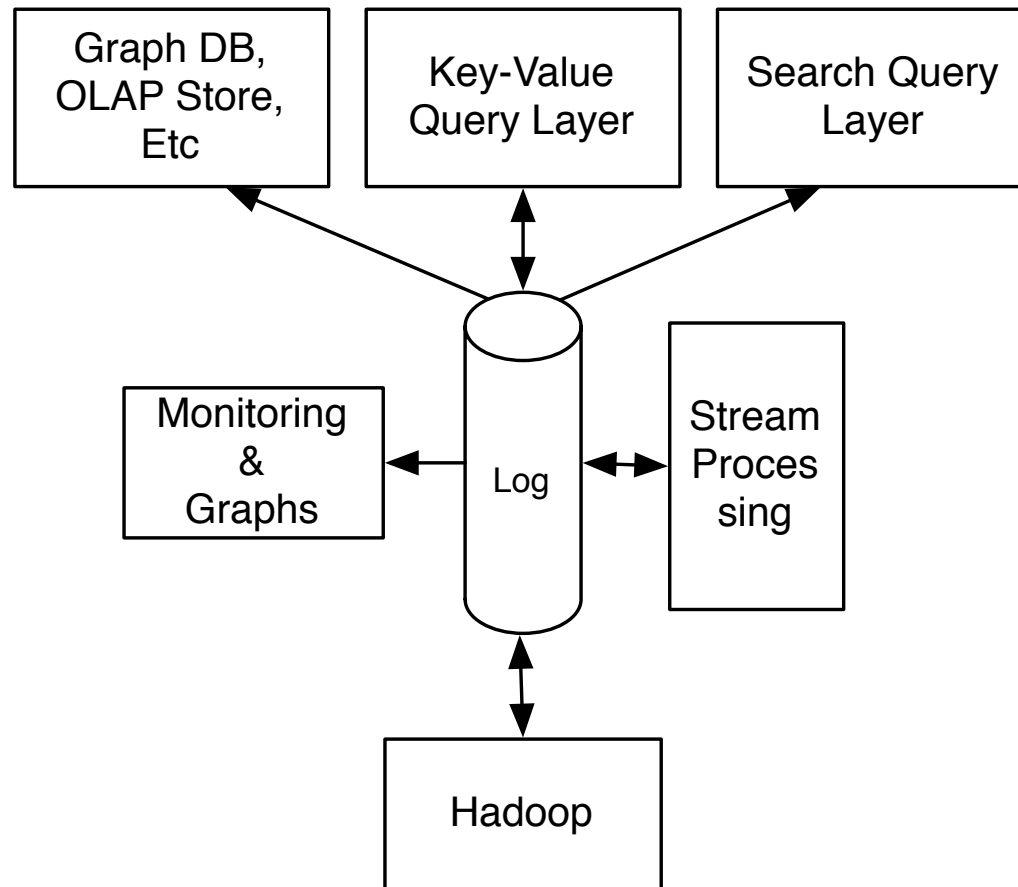
samza



# Samza Architecture



# Log-centric Architecture



Kafka

<http://kafka.apache.org>

Samza

<http://samza.incubator.apache.org>

Log Blog

<http://linkd.in/199iMwY>

Me

<http://www.linkedin.com/in/jaykreps>  
[@jaykreps](#)