BIG DATA

Kafka Architecture and Usage

K V Subramaniam
Computer Science and Engineering

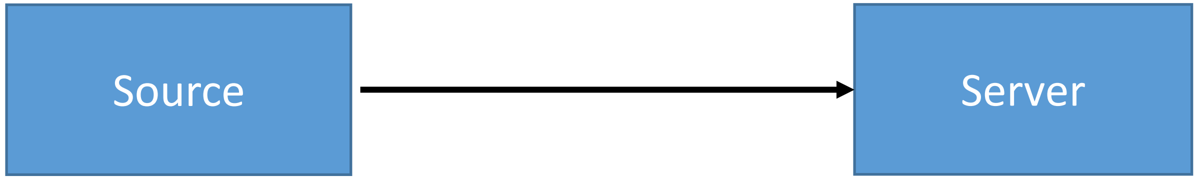
Kafka Overview

- Need for data pipelines
- Kafka architecture
- Kafka components – messaging model
- Communication and Routing
- Messages
- Scaling
- Fault Tolerance

Kafka : Need for data pipelines

The need for processing events

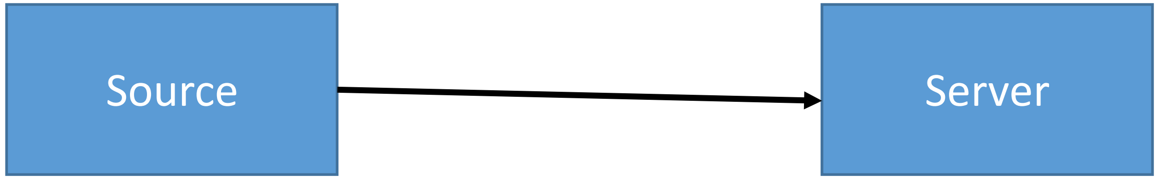
- Stream processing requires requires processing of events
- Think of an event as something that happens at a time
- Events are processed on the server
- For example
  - “Student with ID 23489 entered building”



Data pipelines starts like this.

What are Data Pipelines

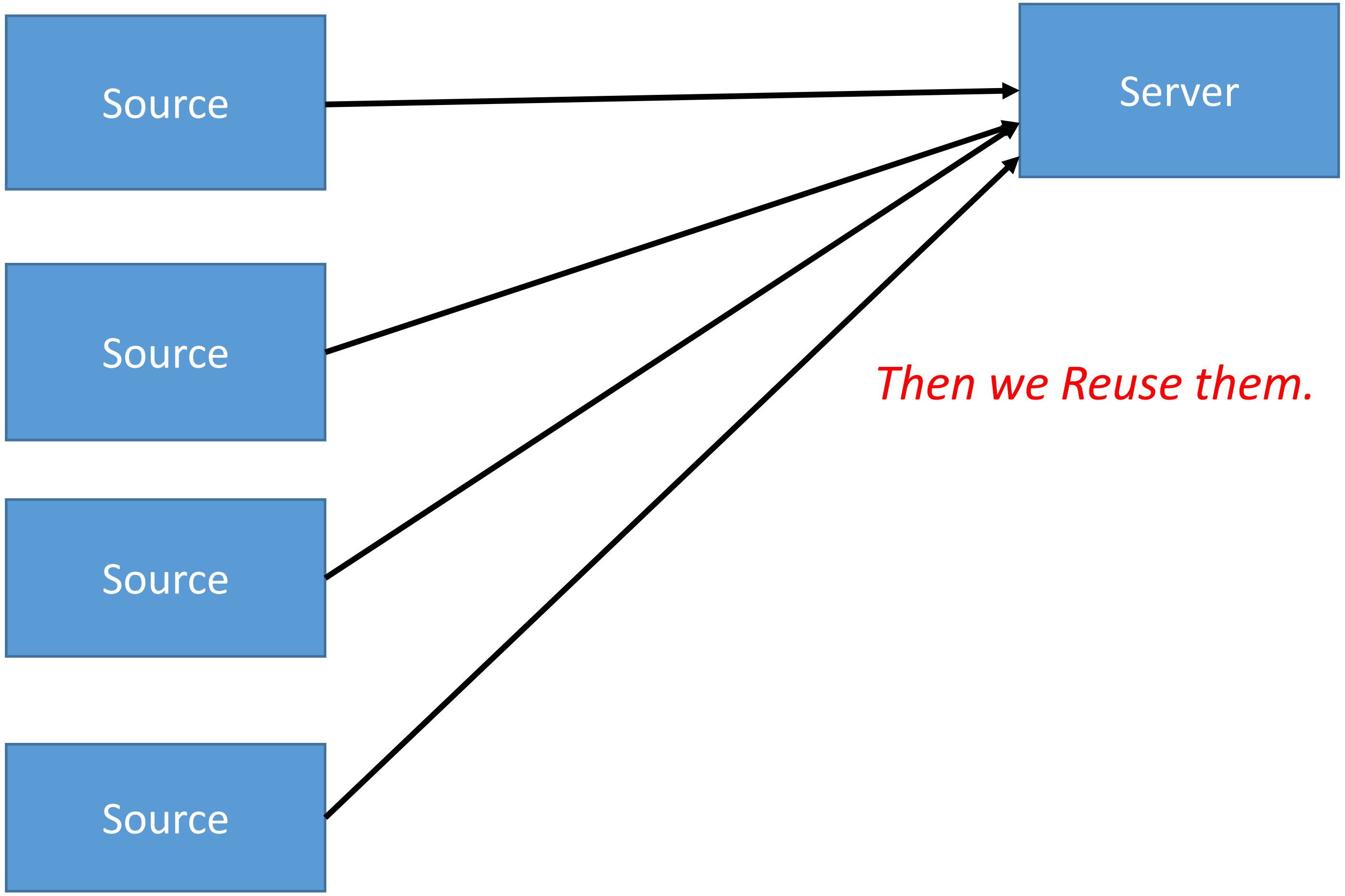
- Data sources are varied and
  - Stream data at varied rates
  - Are very lightweight and not capable of compute
- Analysis is done on servers
- Data has to be transferred from sources to the compute servers
- Point to point connection is a simple way to connect



Data pipelines starts like this.

Handling multiple sources

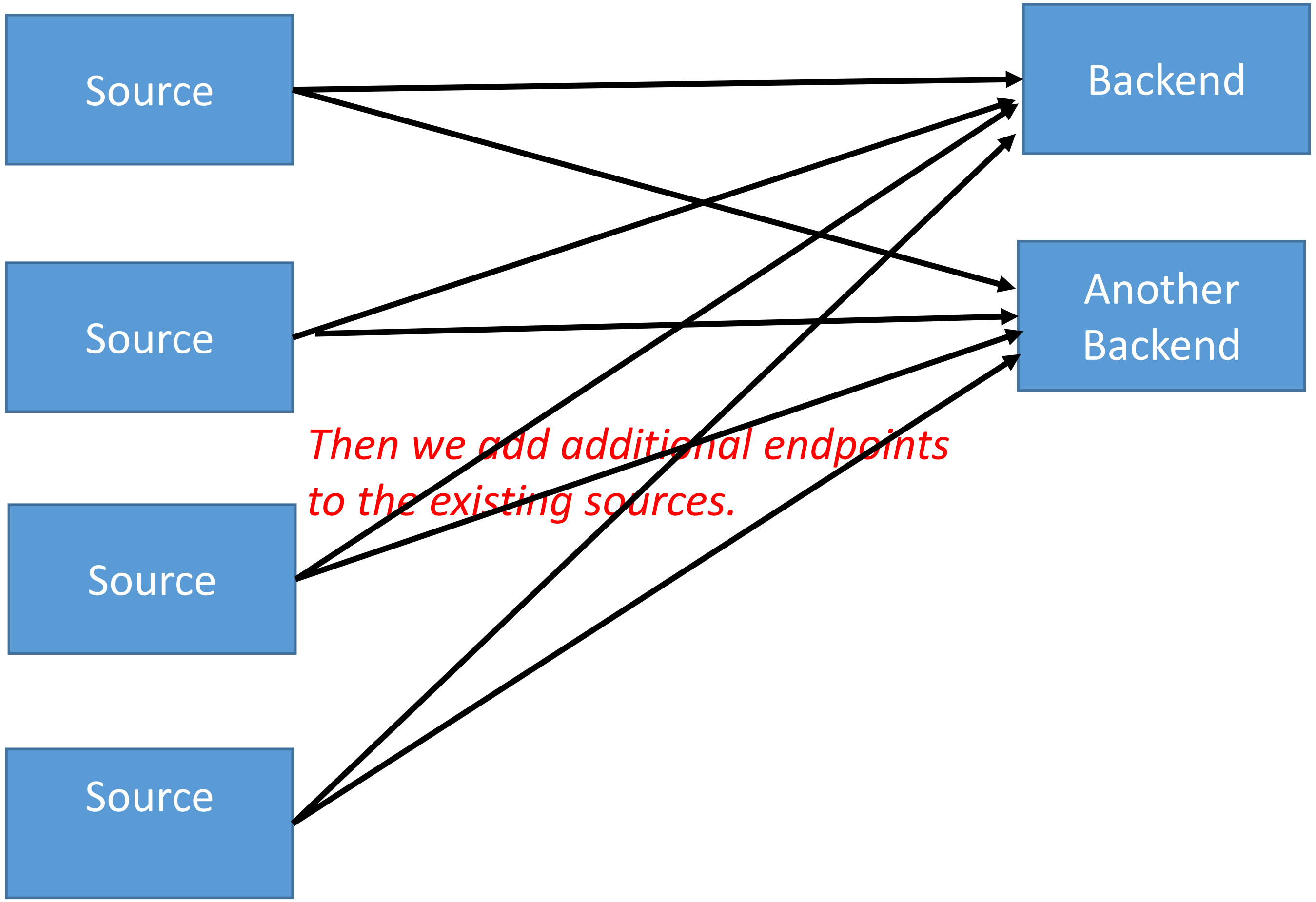
- But sources are many
- We often have to read from multiple data sources
  - E.g: Multiple cameras connected to a central processing system
- We can connect multiple clients each over a pool of connections



Then we Reuse them.

Need for multiple backends

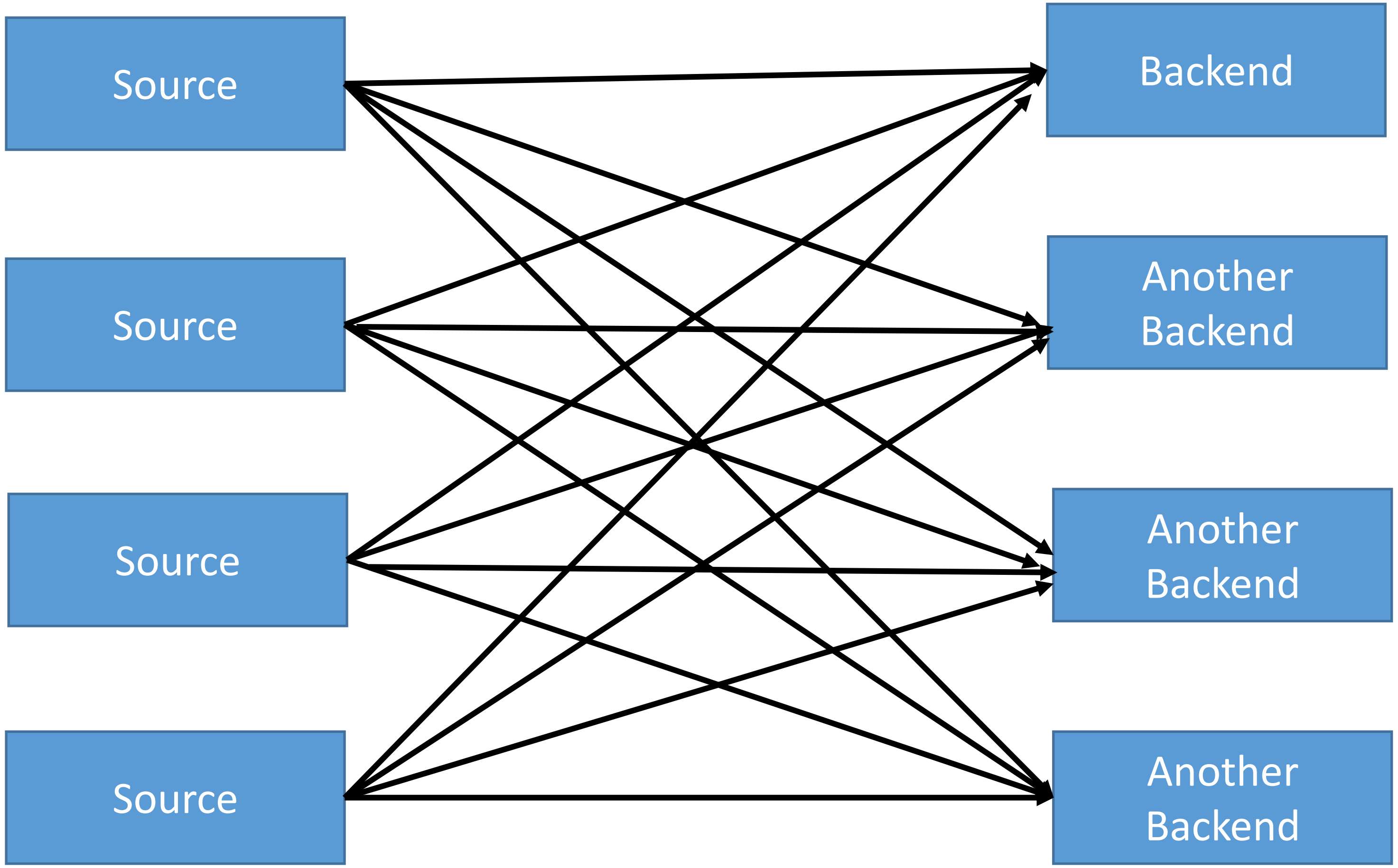
- But we might have multiple servers
- On which to process the same data
- Use case: stock trading – may have different analysts wanting to analyze the behavior



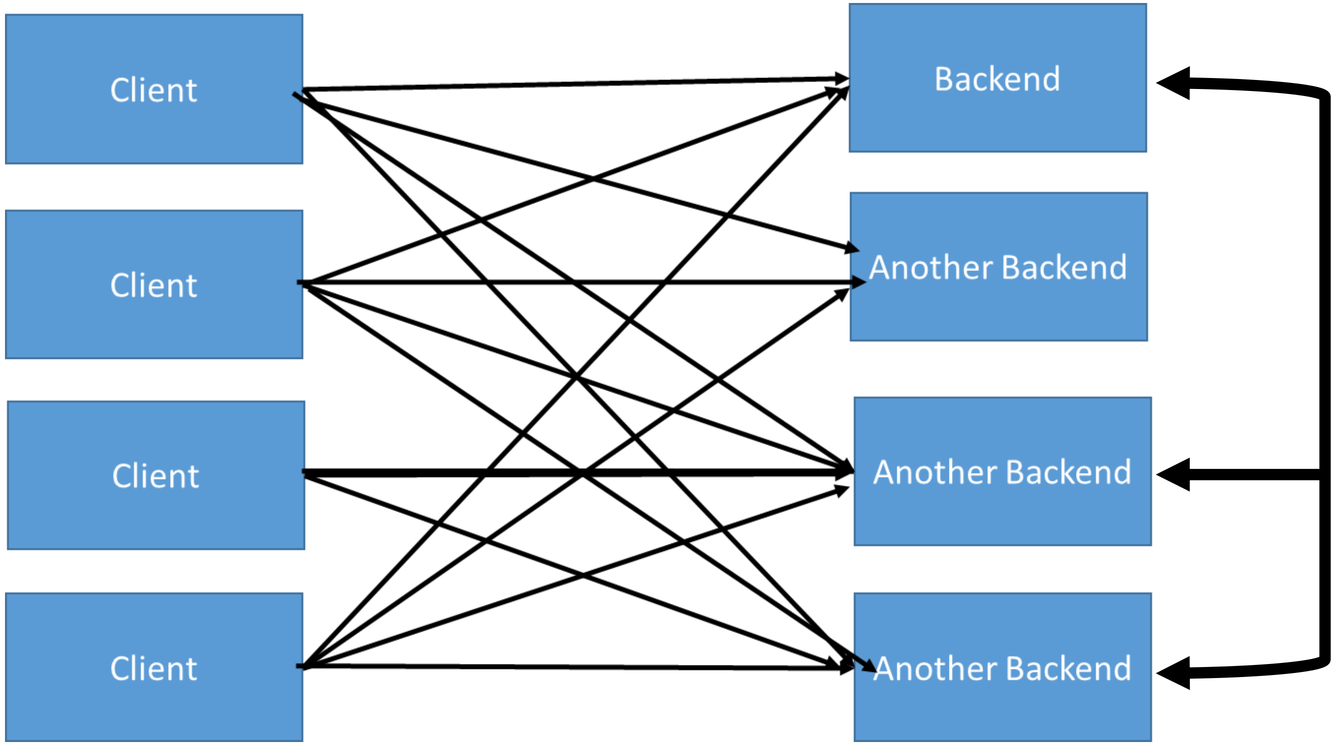
Then we add additional endpoints to the existing sources.

The maze of connections

- Then it starts to look like this.



The maze of connections

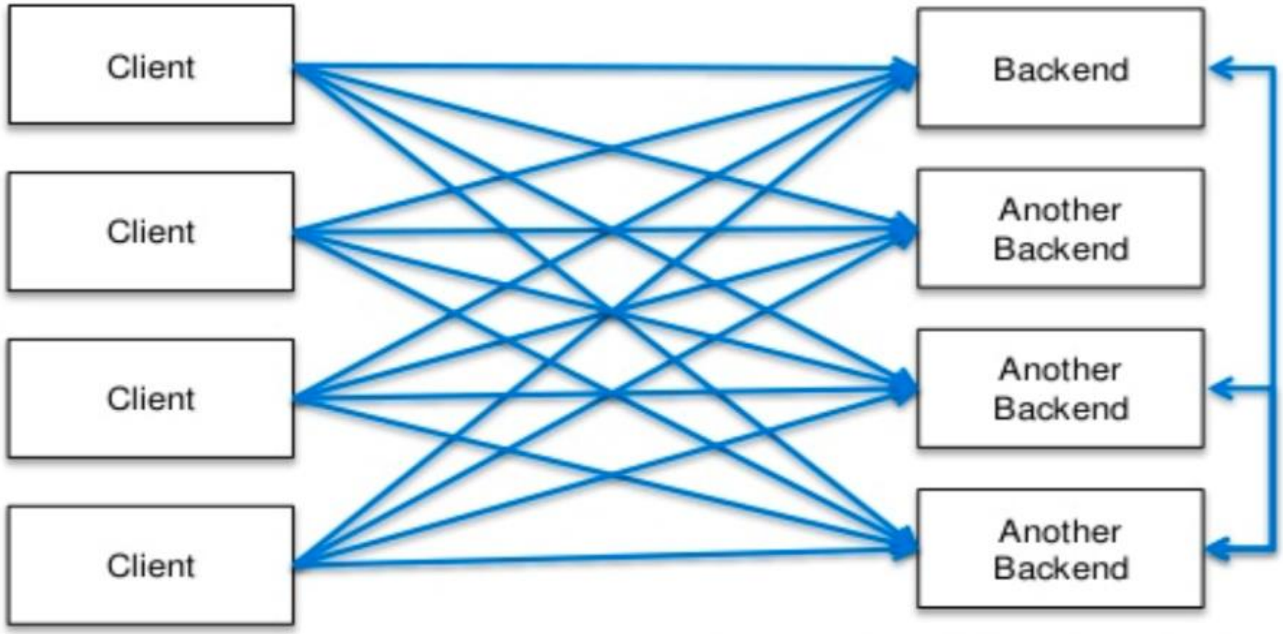


With may be some of these.

- As Distributed systems and services increasingly become part of modern architecture, this makes for a fragile system.

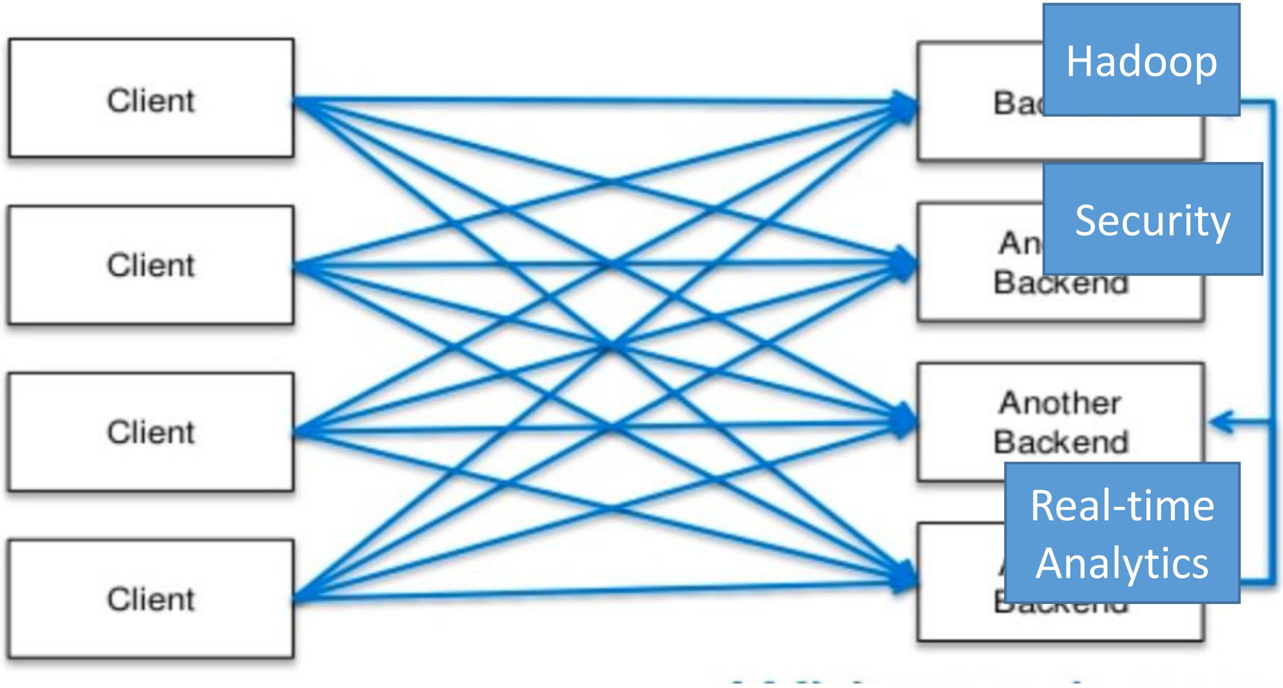
Exercise 1 (5 minutes)

- Give an example of how datapipelines could be used
  - What are some examples of backends?



Exercise 1 (5 minutes)

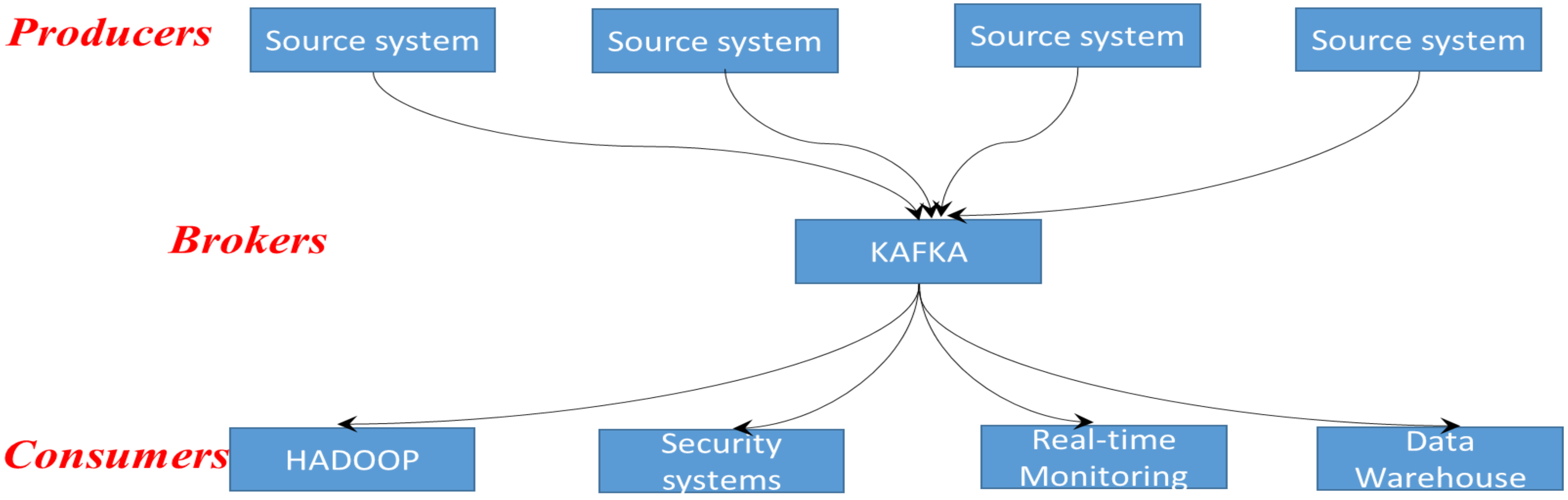
- Give an example of how datapipelines could be used
  - What are some examples of backends?



Kafka architecture

Why Kafka?

- Can we have an intermediary that connects
  - Different sources
  - Multiple backends
- Decouple the pipeline so that producers and consumers do not need to know about each other



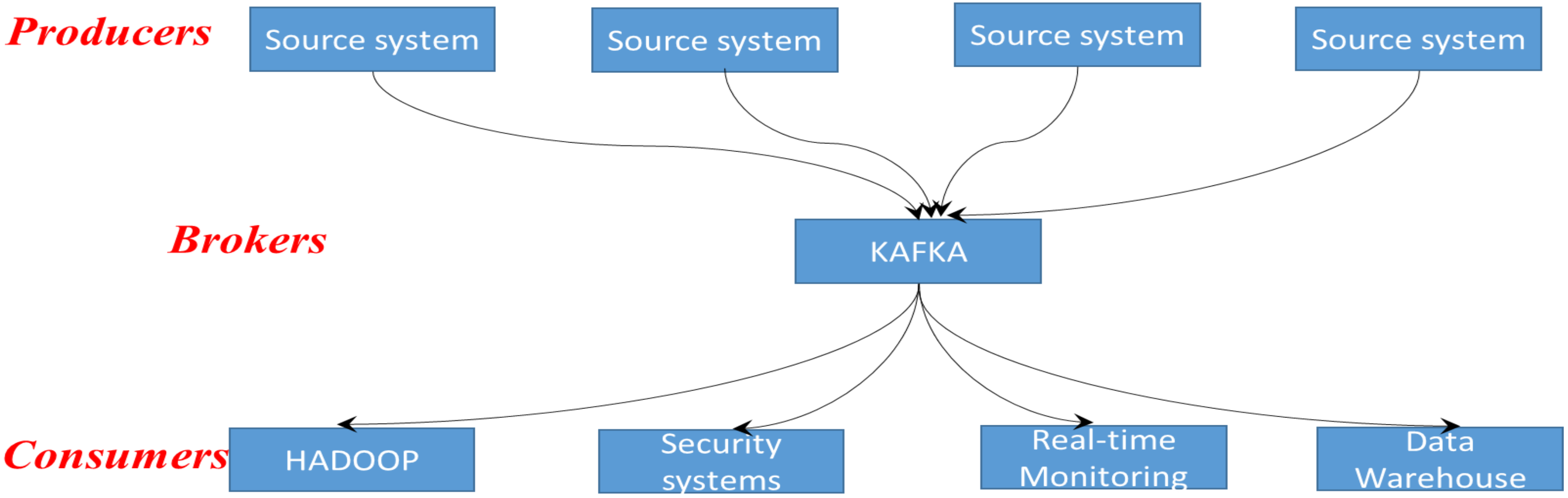
Kafka Decouples Data Pipelines

Kafka Components – Publish Subscribe Model

KAFKA components

The Three essential Elements /Components:

1. Publisher
2. Subscriber
3. Communication Infrastruture



Kafka Decouples Data Pipelines

Publishers and Subscribers

So,

Q) What does a Publisher do ..?

A. It publishes messages to the Communication Infrastructure.

Okay . So

Q) What does a Subscriber do ..?

A. It subscribes to a category of messages.

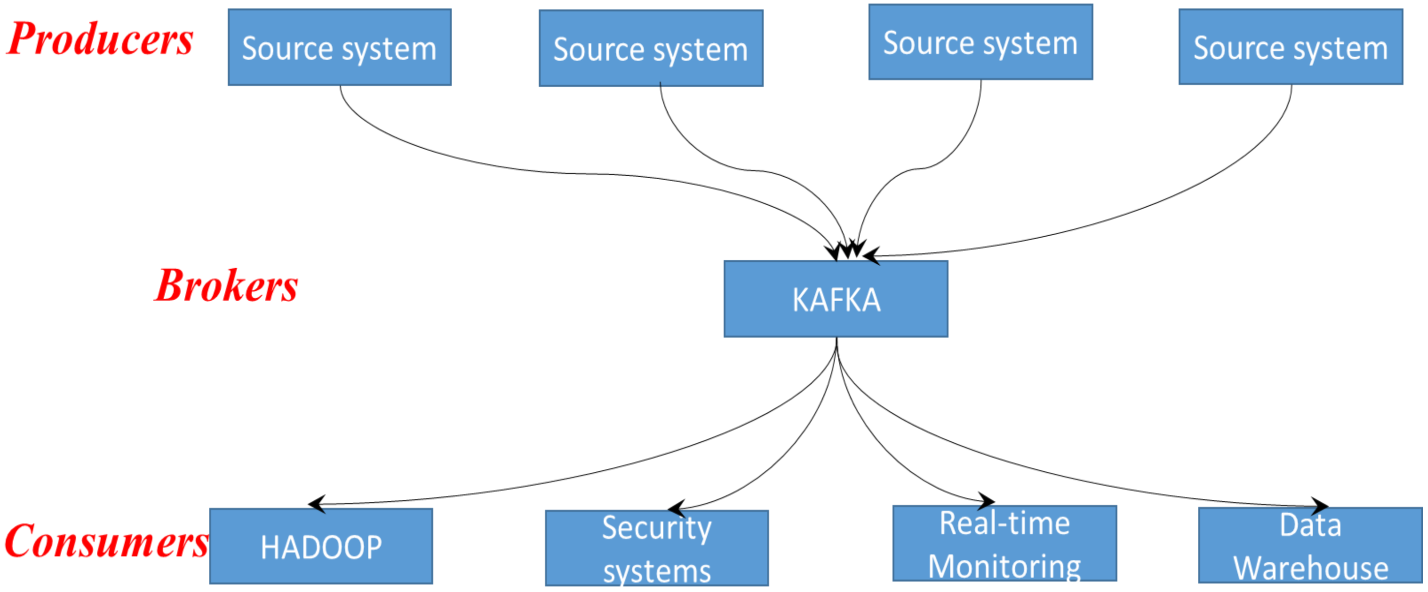
The Routing Mystery

What If I tell you
I am the Publisher
and
I have no idea about
the Subscriber.

What If I tell you
I am the Subscriber
and
I have no idea about
the Publisher.

Role of a producer

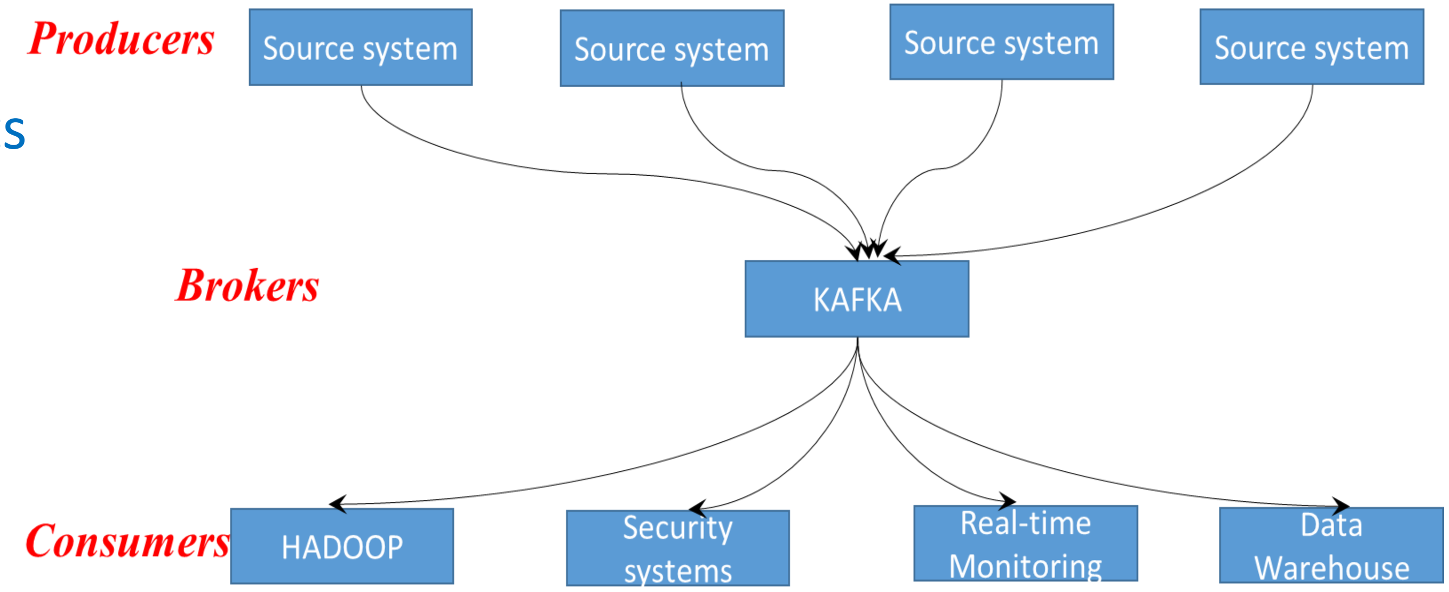
- Defines what data it wants to send
- Packages that go into a message and..
- It publishes messages on the communication infrastructure
- Simplest part → called the *publisher*



Kafka Decouples Data Pipelines

Role of a Consumer

- Tells the communication infrastructure
  - What type of messages it wants to receive
- Does not tell “who” to receive message from.
- Messages are delivered to the consumer by the communication infrastructure.
- Called the *subscriber*



Kafka Decouples Data Pipelines

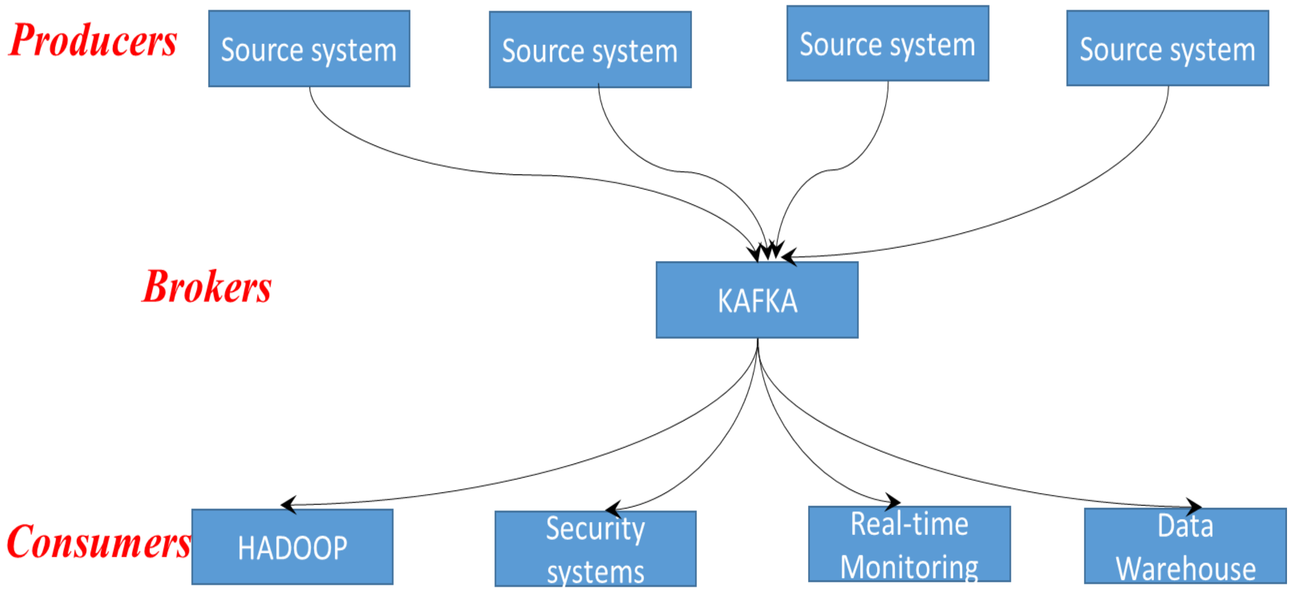
Kafka Communication and Routing

Which Messages get Delivered to Each Subscriber?

- Topic based system
- Content-based system

The Communication infrastructure

- The most complex part
- Which messages are delivered to each subscriber
- Two models of Routing
  - Topic based systems
  - Content based systems



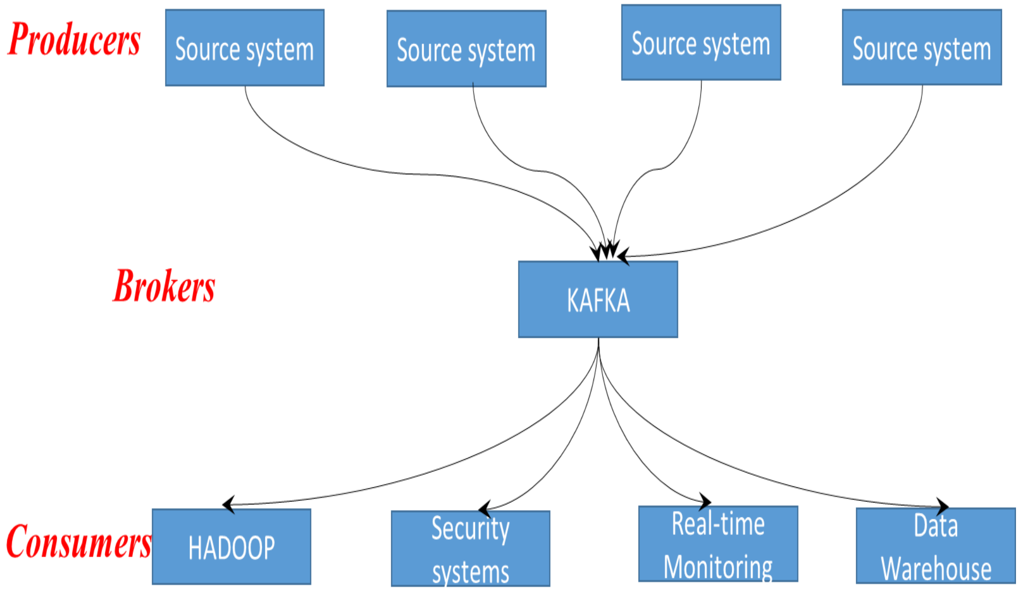
Kafka Decouples Data Pipelines

Topic based routing

- Publishers send messages with topic labels

- Subscribers subscribe to topics
  - And will receive all messages on that topic

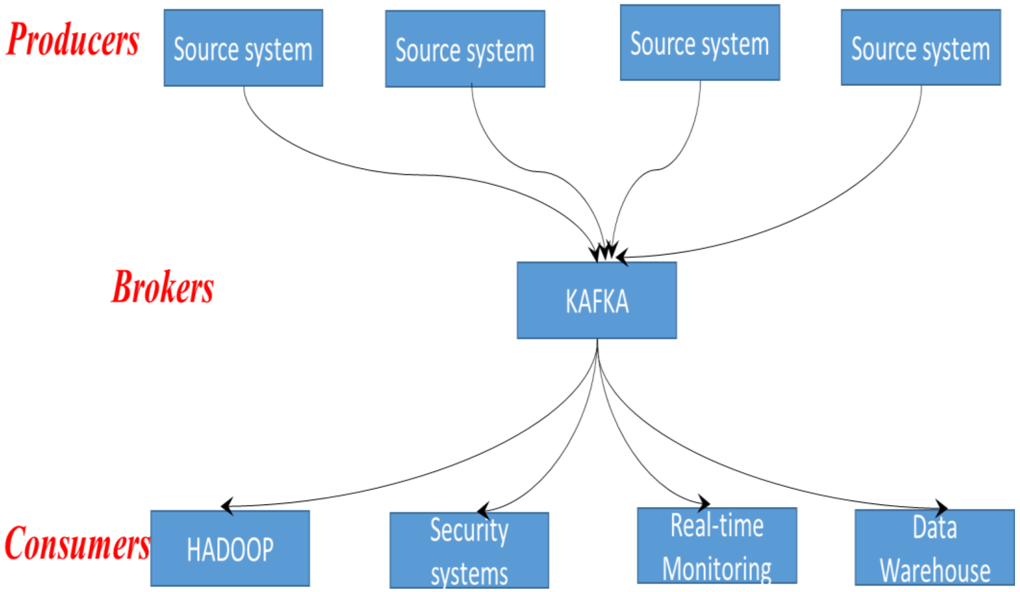
- Example
  - Subscribe to all fire sensors in b block



Kafka Decouples Data Pipelines

Content based routing

- Subscribers define matching criteria
- And will receive all messages that match the criteria
- Example
  - Subscribe to advertisements that feature ViratKohli
  - This is not supported by Kafka
- Pattern based
  - Supports a simpler version called *pattern* based where we can give a wildcard expression for a topic
  - Get all topics that have ipl*



Kafka Decouples Data Pipelines

Communication advantages/disadvantages

Pros

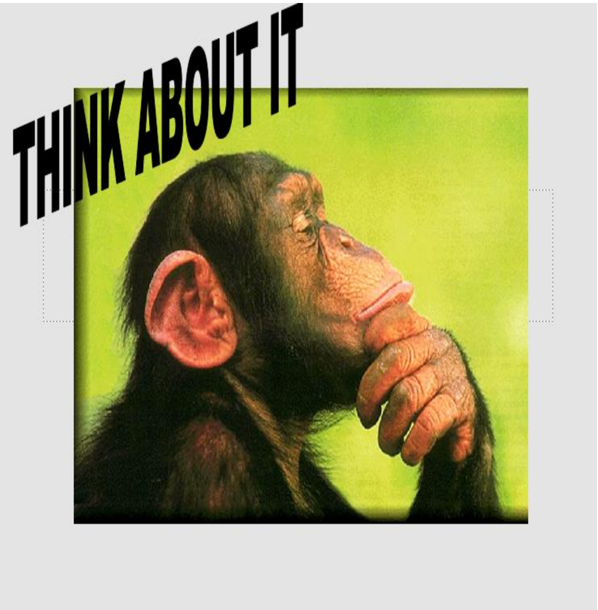
- No hard-wired connections between publishers and subscribers
- Flexible: Easy to add and remove publishers or subscribers

Cons

- Design and maintenance of topics
- Performance overhead due to communication infrastructure

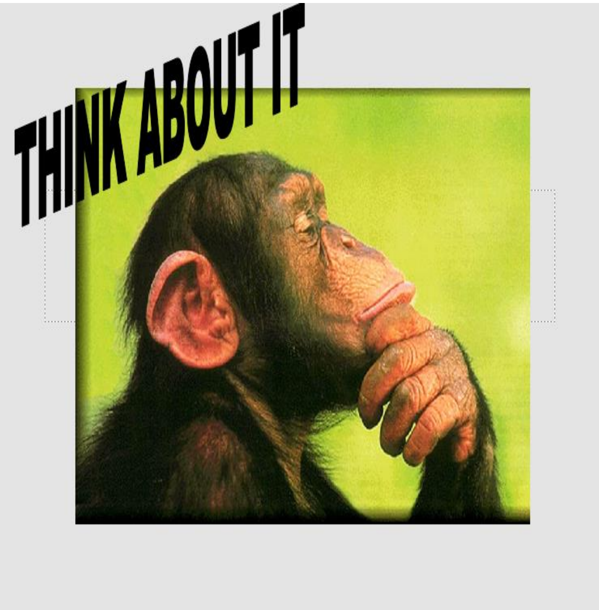
Exercise 2 (5 minutes)

- Consider a bookstore portal with various activities such as
  - Login
  - List books
  - Get book details
  - Buy book
  - Check status of order
  - Return book
  - Logout
- Assume we have 3 backend modules
  - Security
  - Order processing
  - Book information
- **Would you use a topic-based or content-based system? What would be the topics / content..?**



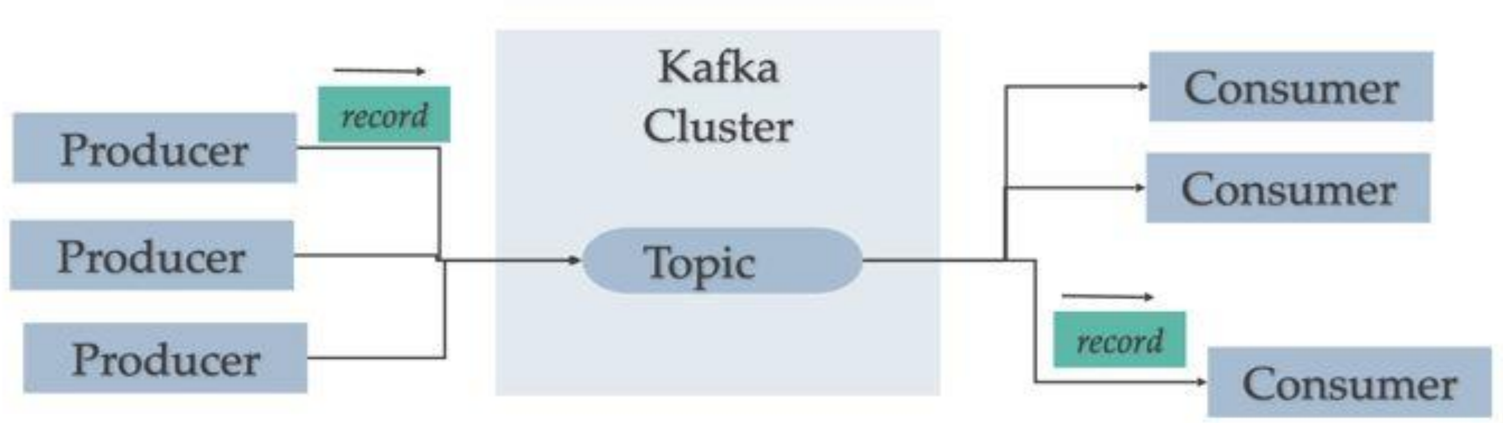
Exercise 2 (Solution)

- Would you use a topic-based or content-based system? What would be the topics / content..?
  - **Probably topic-based, since each message type can be a topic**



Kafka Messages

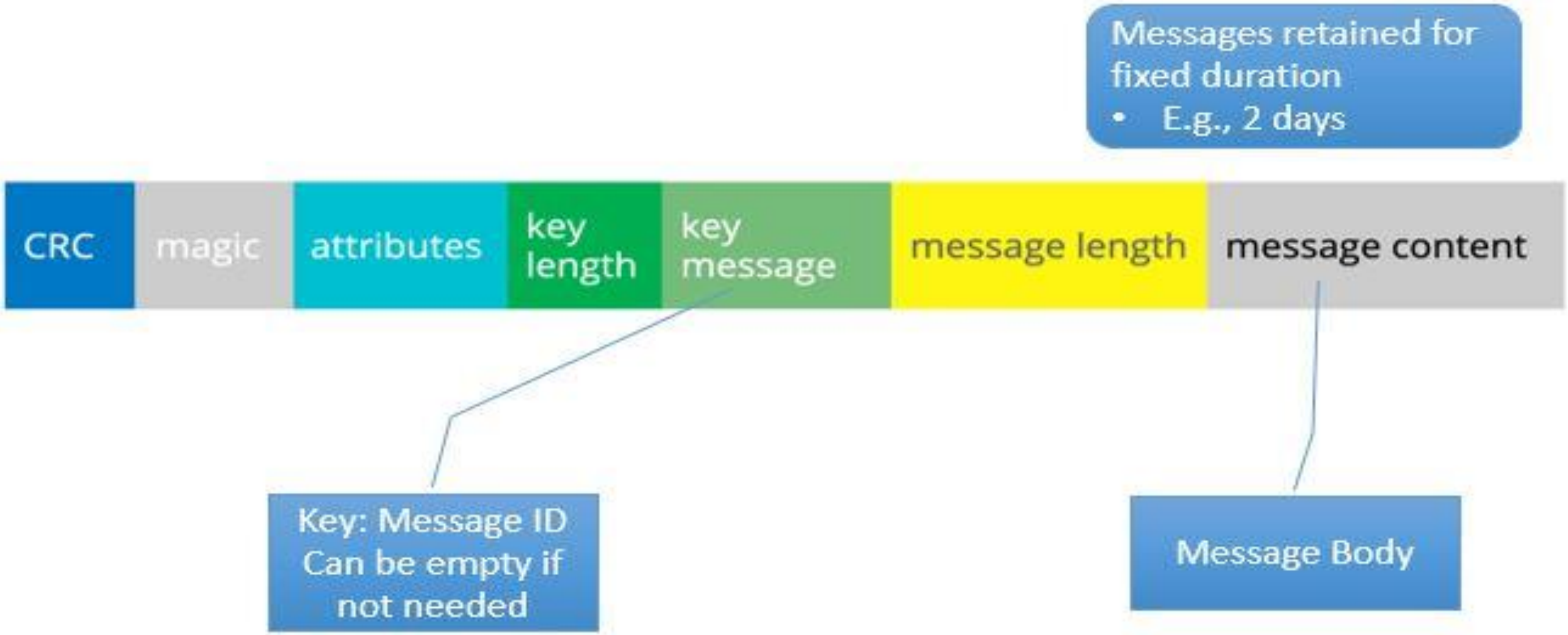
Topics, Producers and Consumers.



Topic name: “/shopping-cart-done“, “/user-signups”

Topic: log data structure on disk

A Kafka Message



Messages retained for fixed duration

- E.g., 2 days

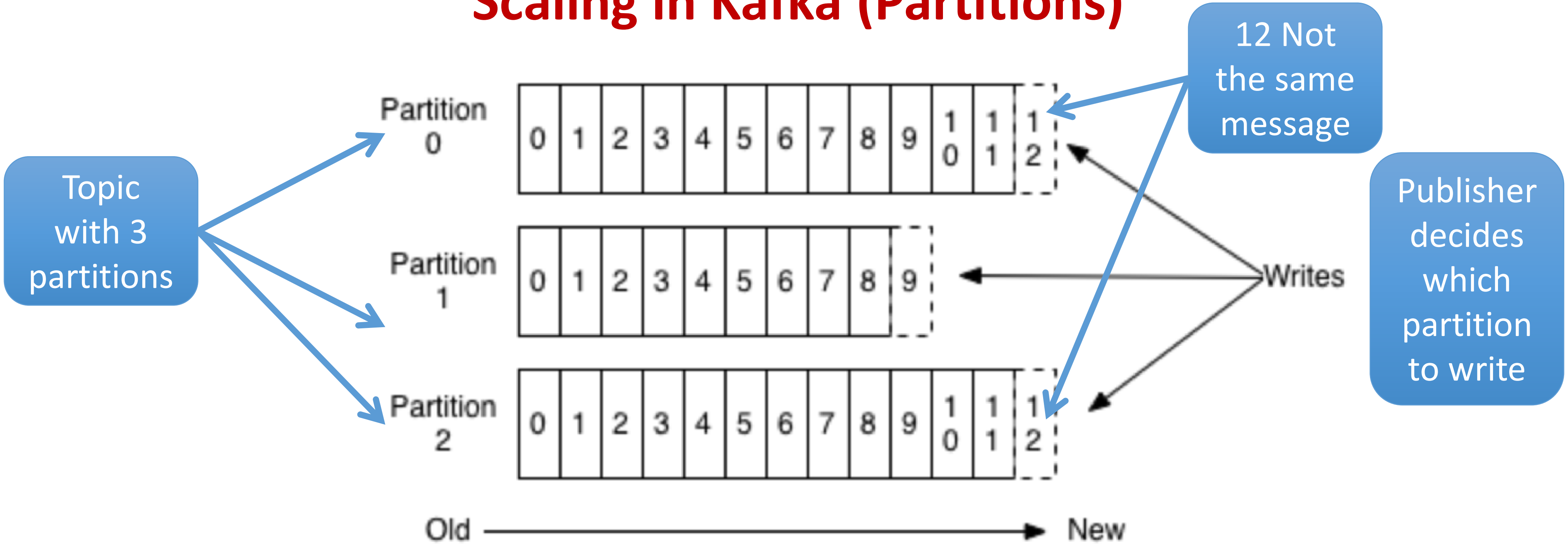
CRC	magic	attributes	key length	key message	message length	message content

Key: Message ID
Can be empty if not needed

Message Body

Kafka Scaling

Scaling in Kafka (Partitions)



Topic with 3 partitions

12 Not the same message

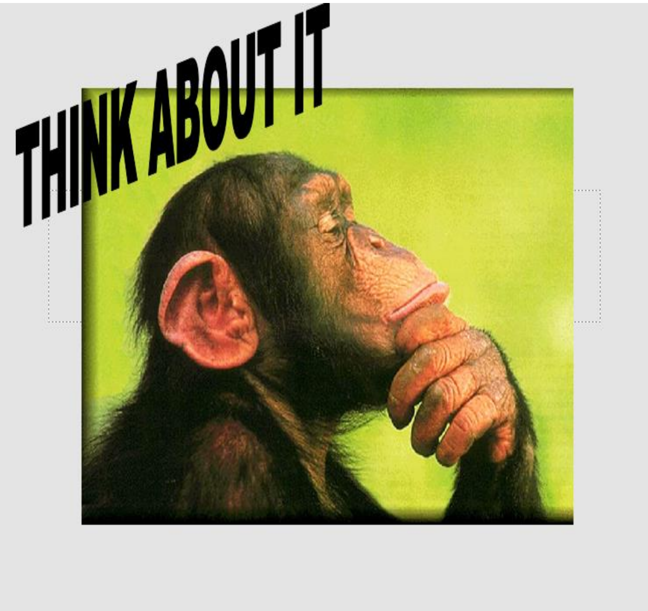
Publisher decides which partition to write

- Partitions allow
  - Log greater than disk size
  - Throughput > single server
- Distributed over servers

- Publisher can load balance
  - Round-robin
  - Based on key
- Messages have an offset

Exercise 3

- How can reliability be guaranteed in Kafka?
  - Hint: How does HDFS guarantee reliability?



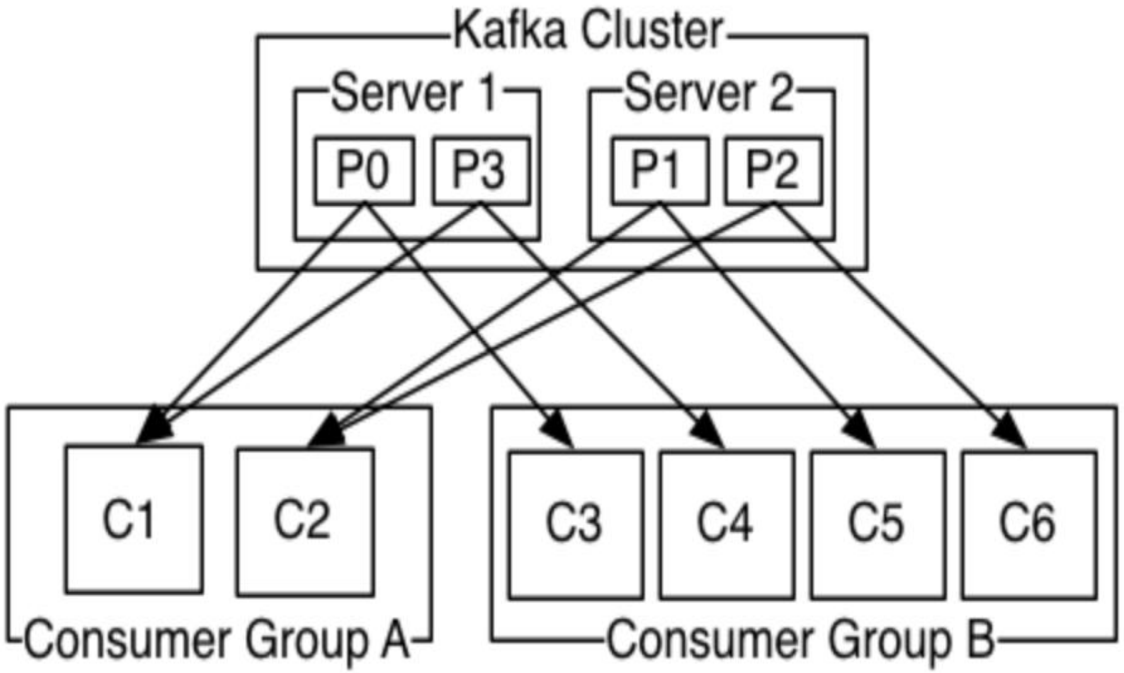
Fault Tolerance in Kafka

Fault-tolerance in Kafka (Replication)

- Partitions can be replicated
  - Leader: all reads, writes
  - Follower replicates
  - New leader if leader fails
- Durability levels
  - Sync: after quorum writes
  - Async
    - 0 = leader only
    - -1 = no write
- Recommended
  - Replicas = 3
  - Quorum = 2

Message delivery to consumers

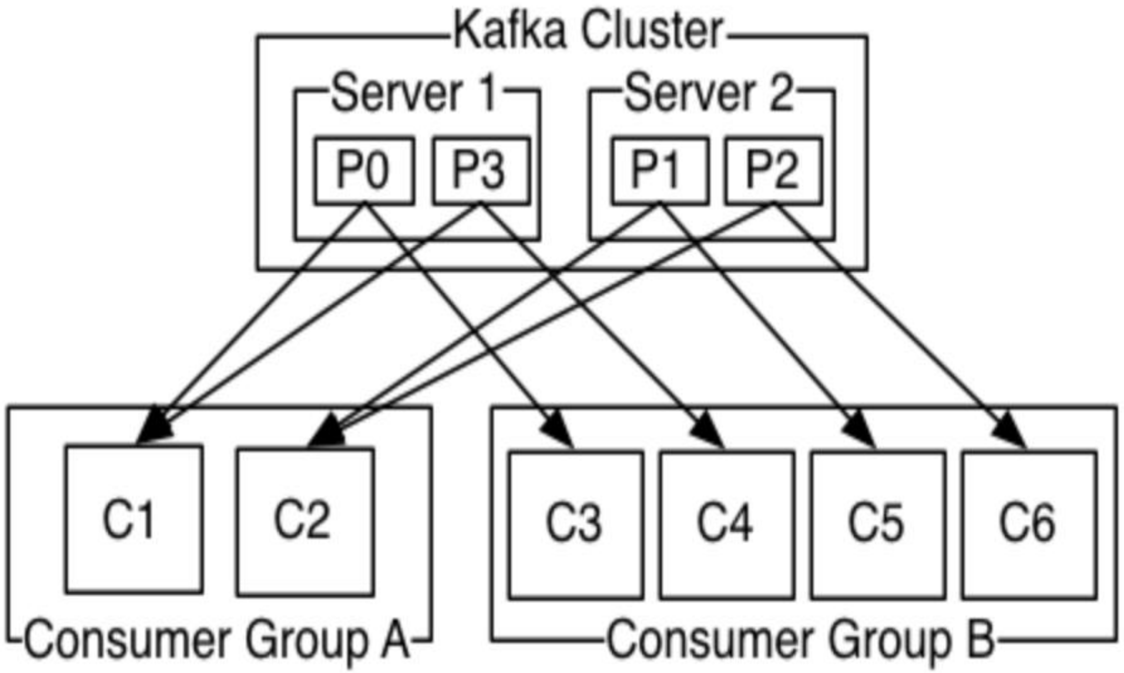
Topic with 4 partitions



- Consumer group
  - Typically multiple instances of an application
- Partition delivers message to ONE of the group members
  - Load balancing

Exercise 4

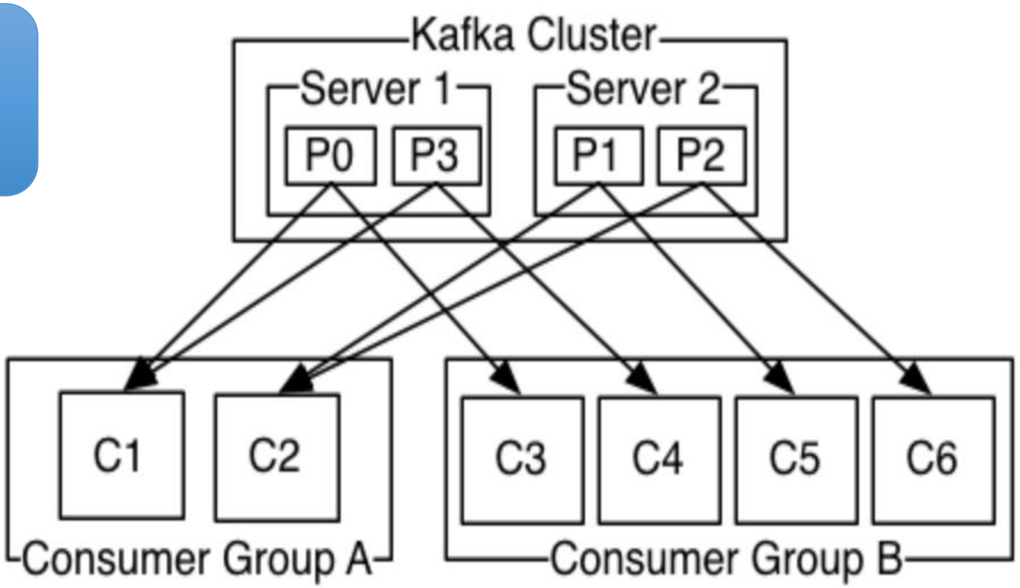
Topic with 4 partitions



- In the above configuration, how is the load balanced over all the instances?

Exercise 4 (Solution)

Topic with 4 partitions



- Group A
  - C1 is assigned P0, P3
  - C2 is assigned P1, P2

- Group B
  - C3 is assigned P0
  - C4 is assigned P3
  - …

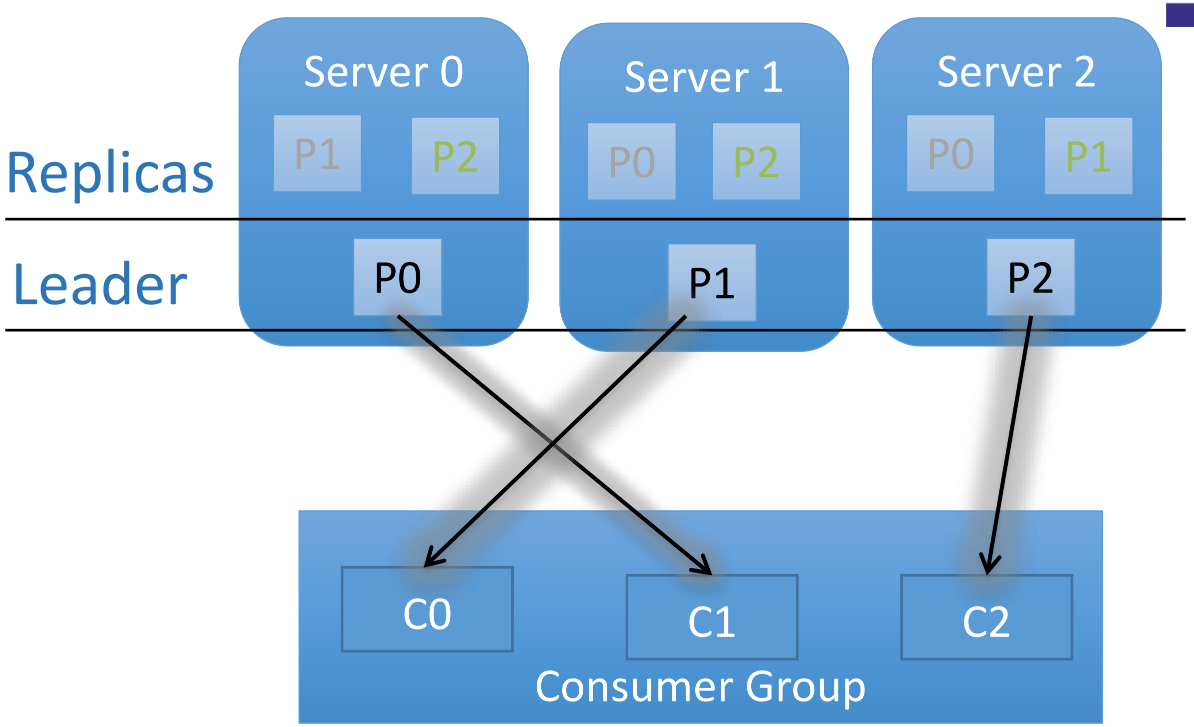
- Each instance has the same number of partitions

Exercise 5

- Suppose we have a Kafka system
  - 1 topic
  - 3 servers
  - 3 partitions
  - 3 replicas per partition
  - Consumer group with 3 instances
- Draw a diagram showing
  - Servers
  - Partitions
  - Consumer instances
  - Partition assignments

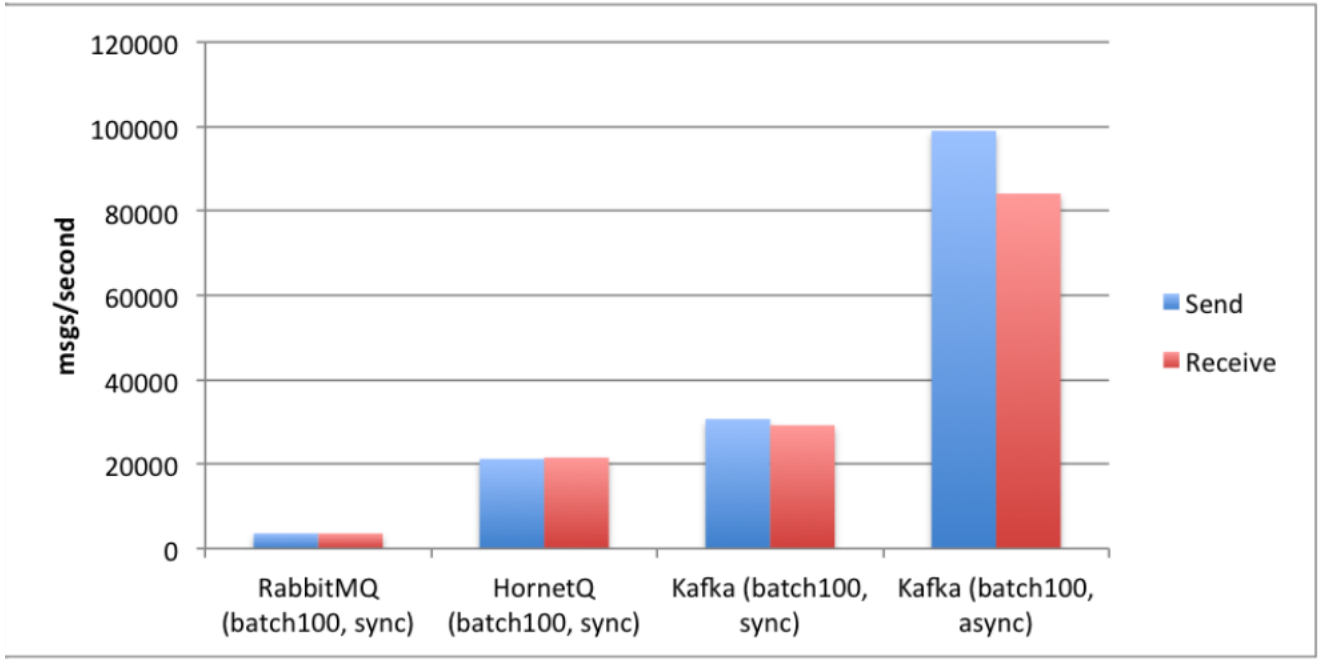
Exercise 5 (Solution)

- Suppose we have a Kafka system
  - 1 topic
  - 3 servers
  - 3 partitions
  - 3 replicas per partition
  - Consumer group with 3 instances
- Draw a diagram showing
  - Servers
  - Partitions
  - Consumer instances
  - Partition assignments



Kafka Performance

Kafka Performance



- I/O
  - Sequential reads by consumers
  - Sequential writes by producers
- Zero Copy I/O
  - DMA
  - No copy from kernel to user
- Write batching

Who uses Kafka ?

- ***LinkedIn***: Activity data and Operational metrics.

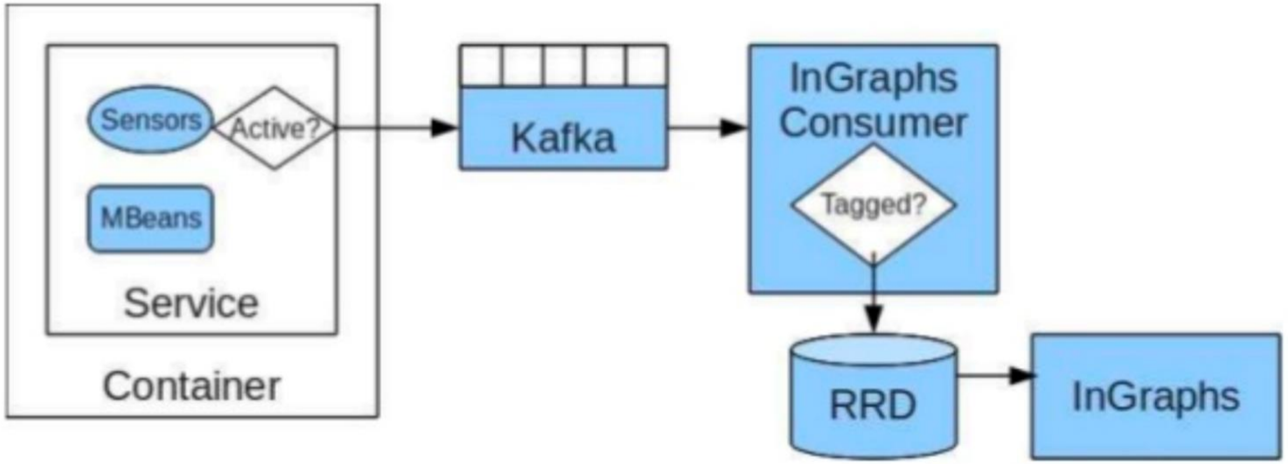
- ***Twitter***: Uses it as part of Storm – stream Processing infrastructure.

- ***Square***: Kafka as bus to move all system events to various Square data centers(logs, custom events, metrics, and so on). Outputs to Splunk, Gtaphite,Esper-like alerting systems.

- ***Spotify, Uber, Tumbler, Goldman Sachs, PayPal, Box, Cisco, Cloud Fatr, DataDog, LucidWorks, MailChimp, Netflix, etc.***

BIG DATA

Kafka at LinkedIn



- Multiple data centers
- System monitoring
  - 320,000,000 metrics/minute
  - 530 TB of disk space
  - > 210,000 metrics / service
- Analysis – Hadoop, …

Additional Reading

- Introduction to Kafka
  - https://kafka.apache.org/intro
  - Watch the video on the introduction.
- Kafka in a nutshell
  - https://sookocheff.com/post/kafka/kafka-in-a-nutshell

THANK YOU

K V Subramaniam, Usha Devi

Dept. of Computer Science and Engineering

subramaniamkv@pes.edu

ushadevibg@pes.edu