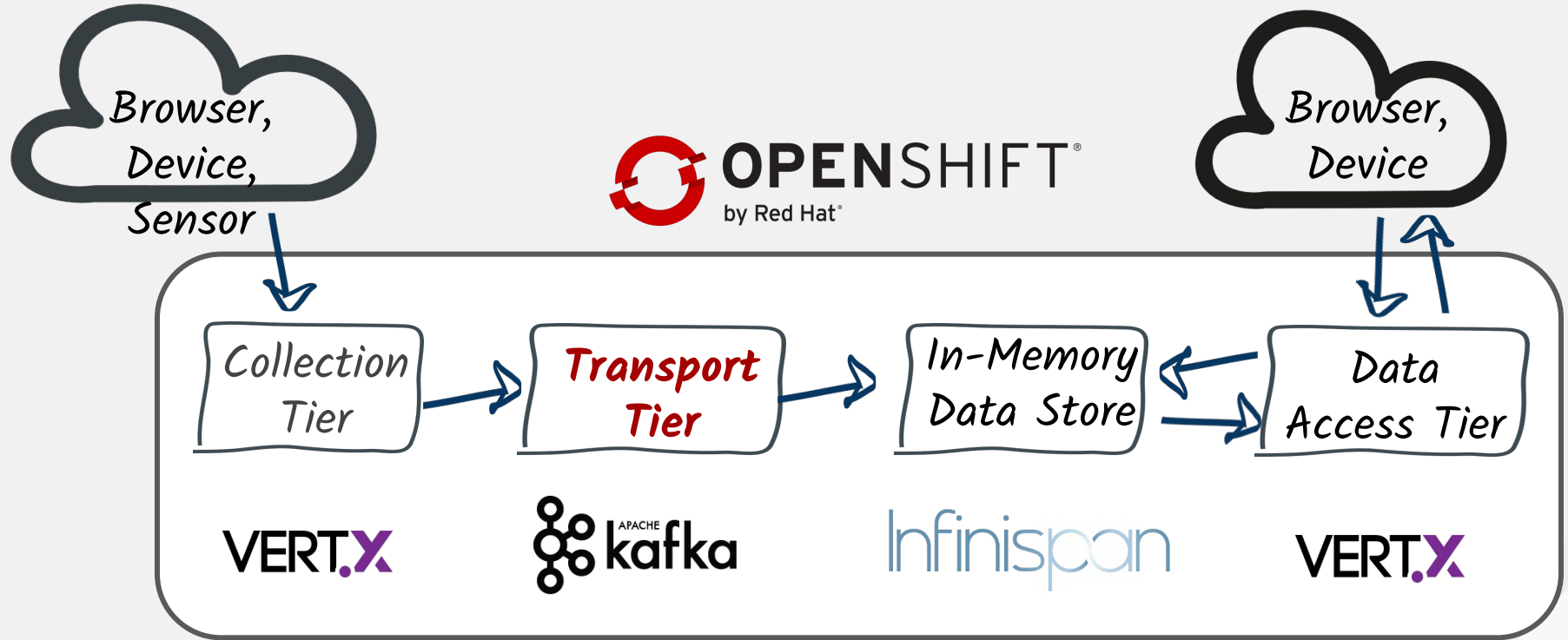


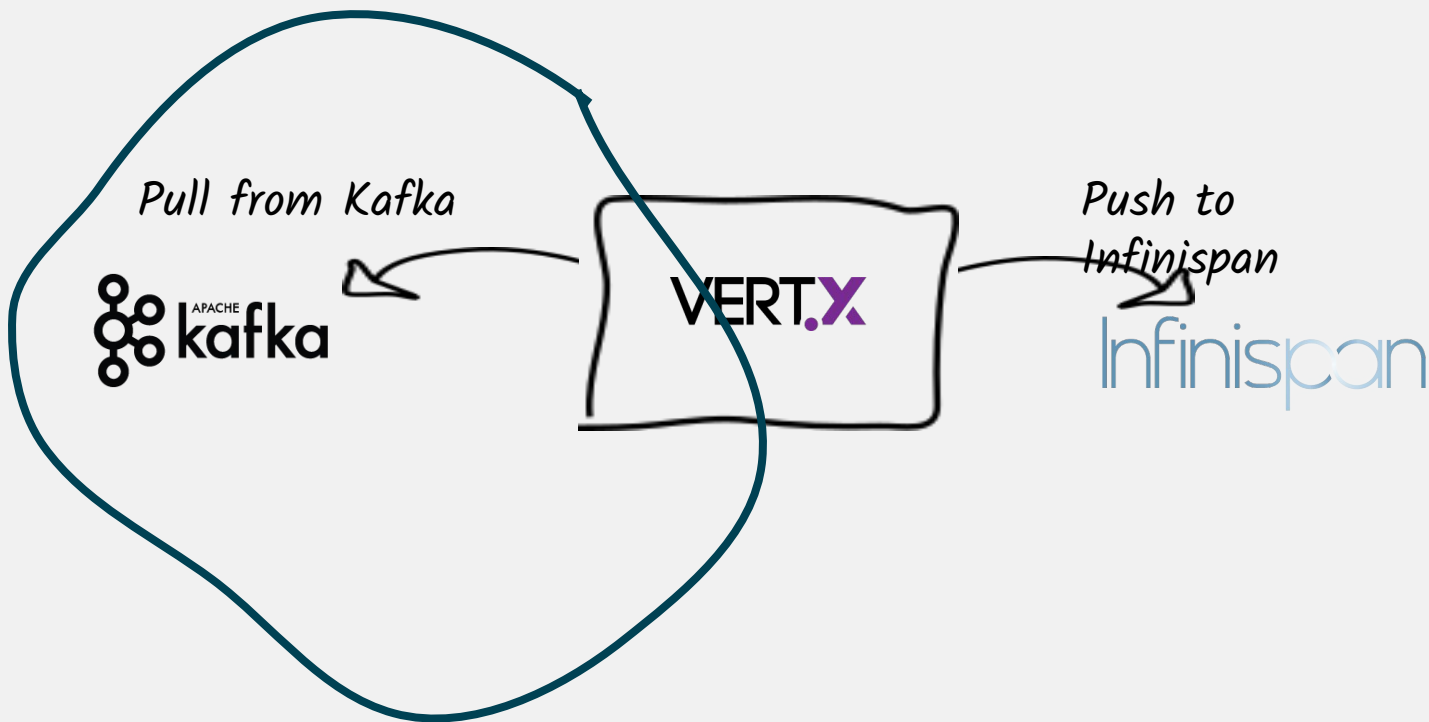


Transport - From Kafka to Infinispan

Streaming Data - Deep Dive



Message Queuing Tier

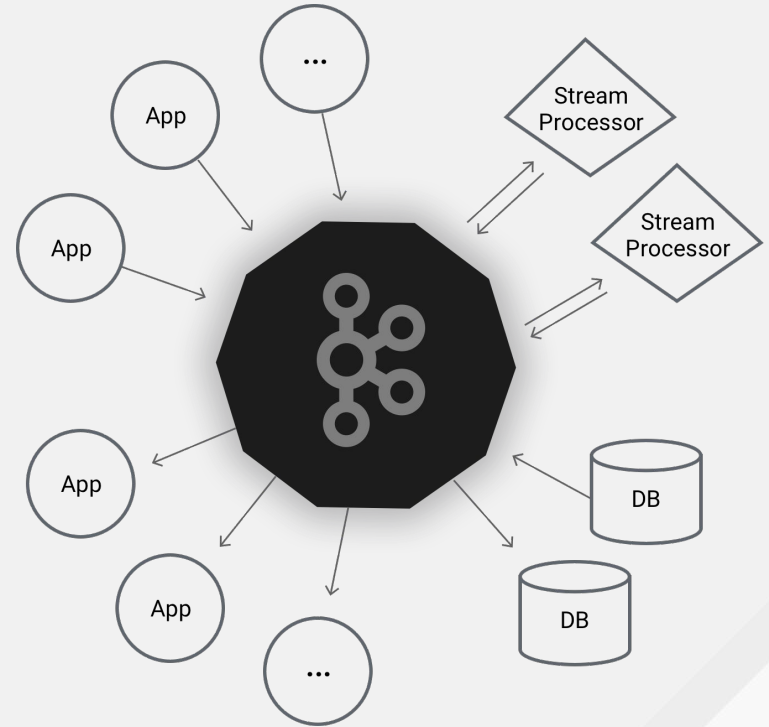


Kafka in 5 minutes

Kafka - Streaming framework

Publish / Subscribe

- streams of **records** on **topics**
- fault-tolerant
- clustered



Stream of records => topic

Topic

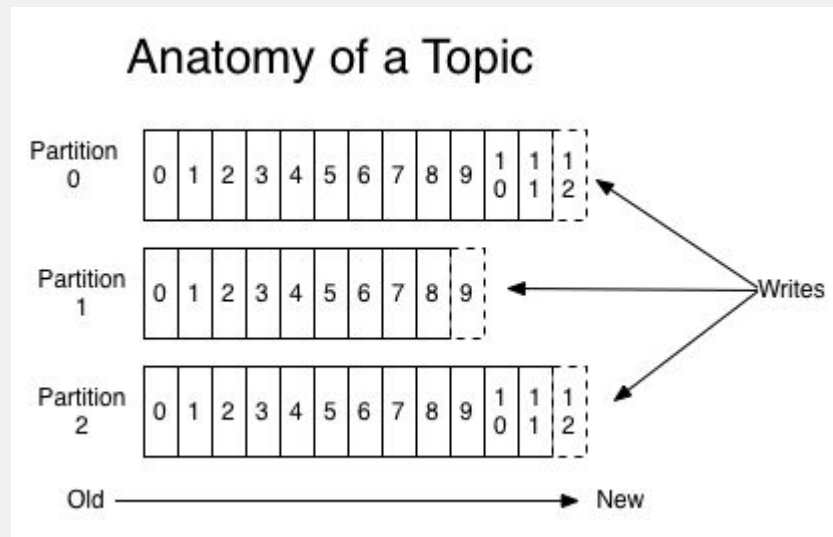
- Multi-subscribers
- Topic => partitioned log

Partition

- an ordered, immutable sequence of records
- An *offset* identifies each record within the partition.

Record

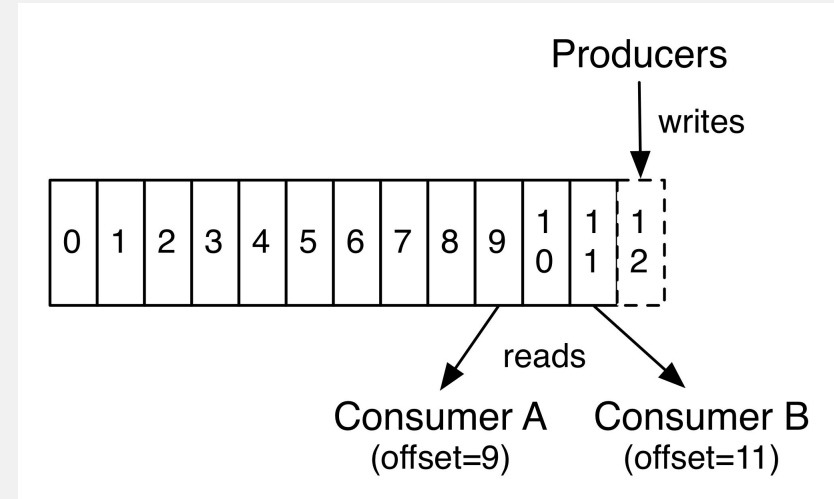
- Each record consists of a key, a value, and a timestamp.
- **Retention period**



Consumers

The consumer reads the records and manage the *offset*

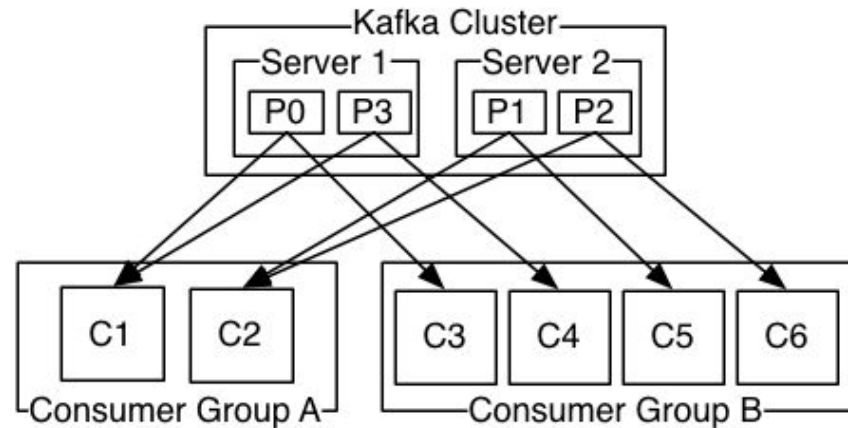
Each consumer can read different records



Consumer Group & Broadcasting

Consumer belongs to a **consumer group**

Each record published to a topic is delivered to **one** consumer from each group



Reading Kafka records using Vert.x

Consuming records using the Vert.x Kafka Client

Kafka Read Stream generates a stream from the records of a specific topic

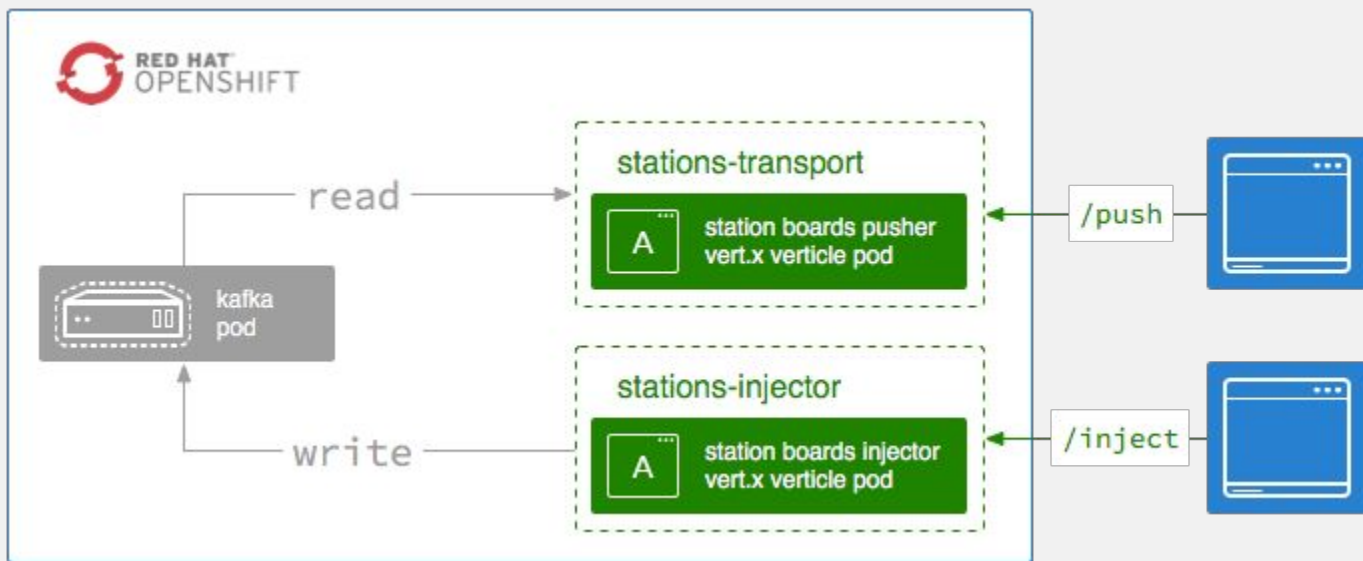
```
FlowableHelper.toFlowable(consumer)
```

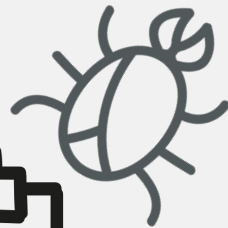
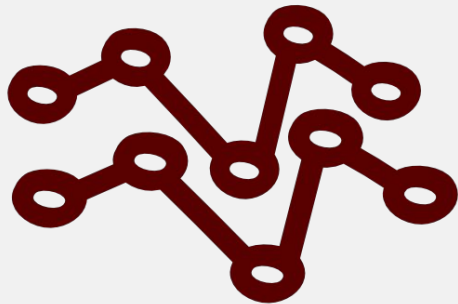
```
// here you get your records...
```

```
// Instruct the consumer to start reading the records
```

```
consumer.subscribe(Collections.singleton("topic"), ar -> {  
  if (ar.succeeded()) {  
    Log.info("YEAH!");  
  } else {  
    Log.error("D'oh!", ar.cause());  
  }  
});
```

Updated Architecture





DEMO TIME!

