Shaun Thomas

mini sLab | Kafka in fintech: An introduction to Apache Kafka

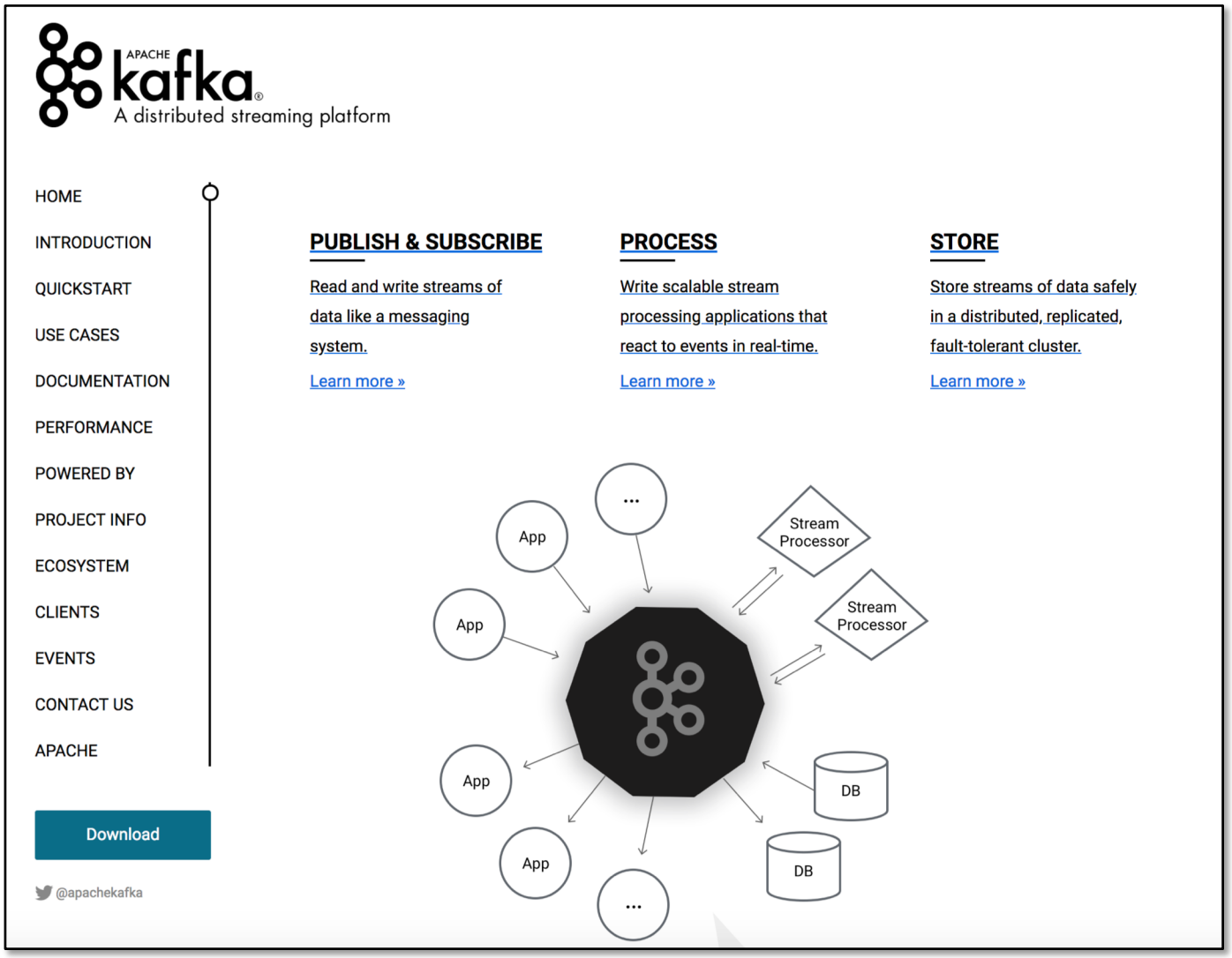
https://www.meetup.com/Sytac-sLab/events/256434066/

Introduction

Kafka

Kafka® is used for building real-time data pipelines and streaming apps. It is horizontally scalable, fault-tolerant, wicked fast, and runs in production in thousands of companies.

https://kafka.apache.org/



https://kafka.apache.org/

History/Facts/Figures

- Originally developed at LinkedIn in 2010
- Open sourced in 2011
- Graduation from the Apache Incubator in 2012
- Jun Rao, Jay Kreps, and Neha Narkhede (worked on Kafka at LinkedIn)
  - created a new company named **Confluent**
- Written in Java and Scala
- Kafka adoption exploded in 2017
  - Used by 1/3 of Fortune 500 companies
  - 6 of top ten travel companies, 7 of top ten banks, 8 of top ten insurance companies, 9 of top ten telecom companies uses Kafka in production
  - Version – 0.10.x
- Current version – 2.1.x
- Spring-kafka – 2.2.x (based on kafka-clients jar – Official client)
- Client are available for many programming languages

Basic concepts

3 main components of Kafka

Component	API	Description
Publish/Subscribe (Pub/Sub)	Producer API Consumer API	Sending and receiving messages efficiently and reliably at scale.
Kafka Connect	Connect API	Integrating Kafka with external data sources
Kafka Streams	Streams API	Processing messages in real time

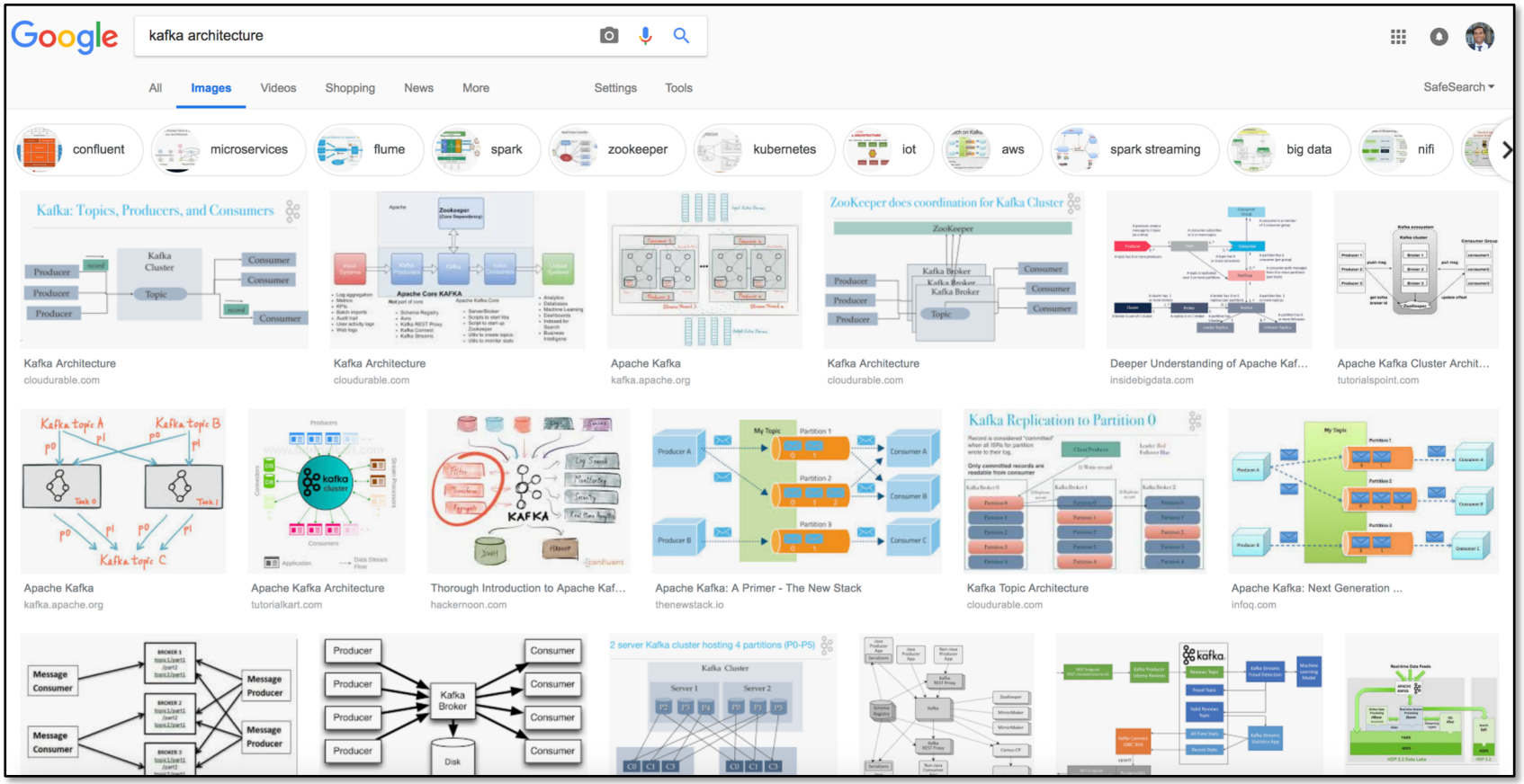
Broker, record and topic

Broker / Server / Bootstrap server	Servers where data are stored. Multiple brokers can be used to form a cluster.
Message / **Record**	Data stored in Kafka. A record contains a key, a value, a timestamp and a list of headers
Topic / Event	Logical name for producing, consuming or separating records

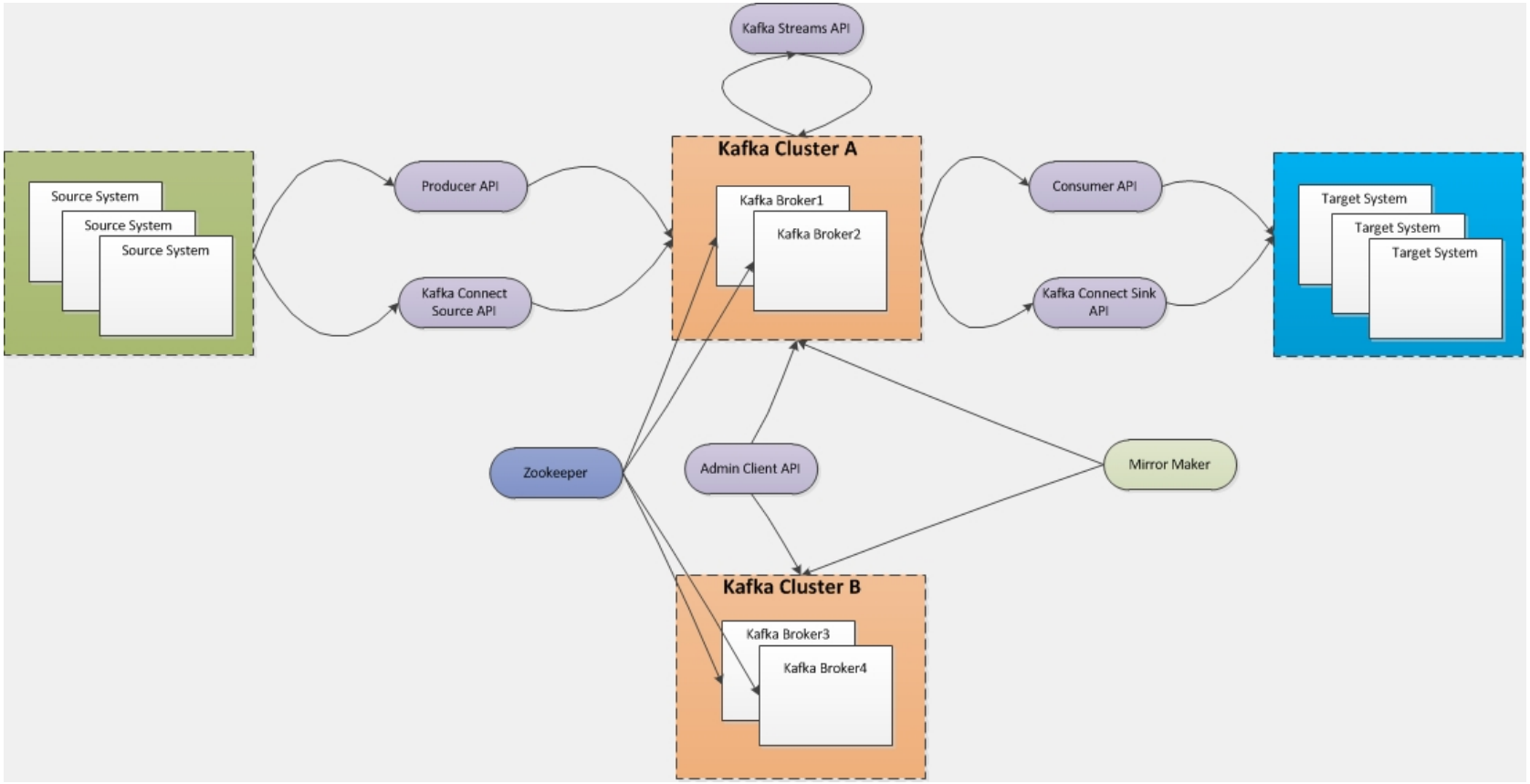
Zookeeper

- Zookeeper is an open source server that enable highly reliable distributed co-ordination
- We just have to start Zookeeper and don't worry about it at all
- It helps in maintaining the cluster membership. For example, when a new broker is added, a broker is removed, a new topic is added or a topic is deleted, when a broker goes down or comes up etc, Zookeeper manages such situations, informs Kafka.

Kafka Architecture – Too many varieties…

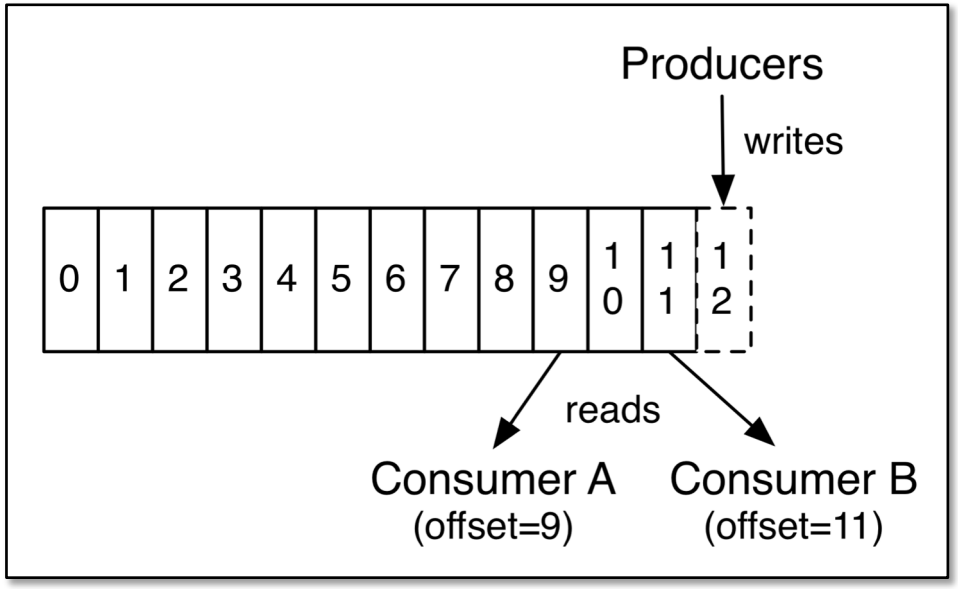


Kafka Architecture / Ecosystem



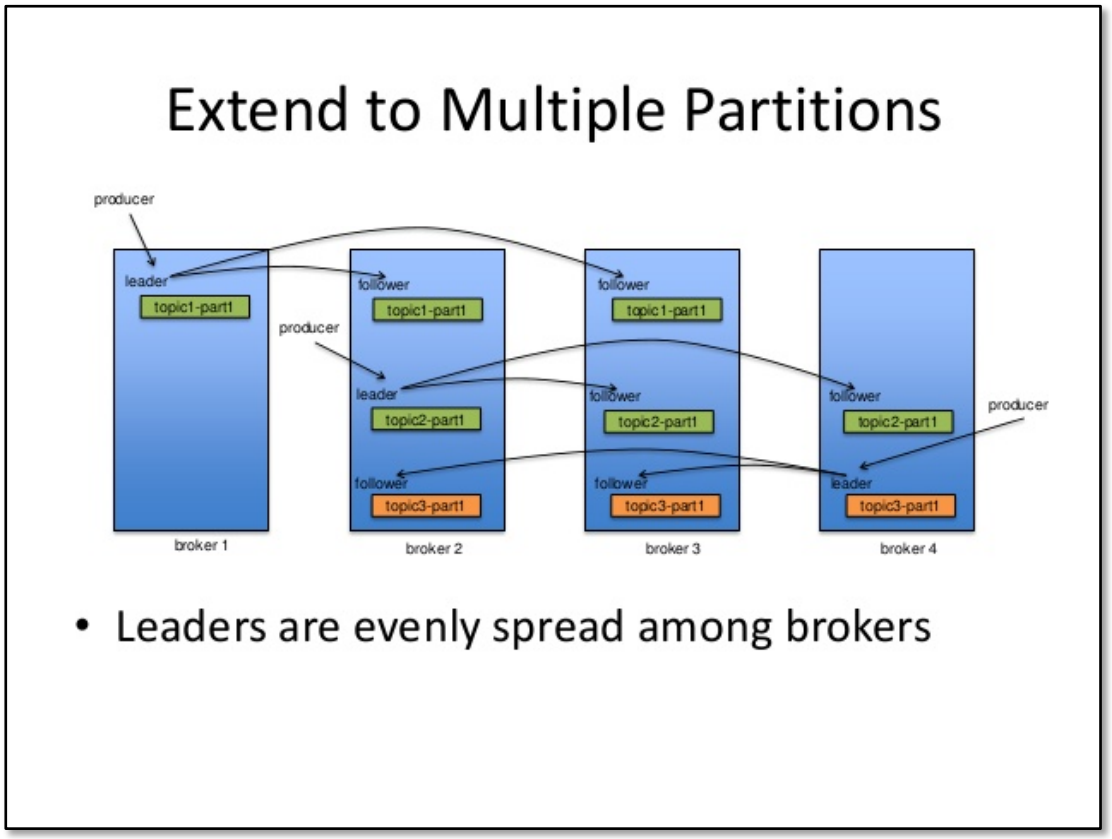
https://iteritory.com/beginners-guide-apache-kafka-basic-architecture-components-concepts/

Commit log, retention policy and offset



- Commit log
  - ordered, immutable sequence of records
  - continually appended
  - storage mechanism in Kafka
- Retention
  - Common policies
    - 'log.retention.hours' - time
    - 'log.retention.bytes' - size
- Offset
  - sequence id assigned to a record in a commit log
  - Uniquely identifies a record
  - Maintained as metadata per consumer

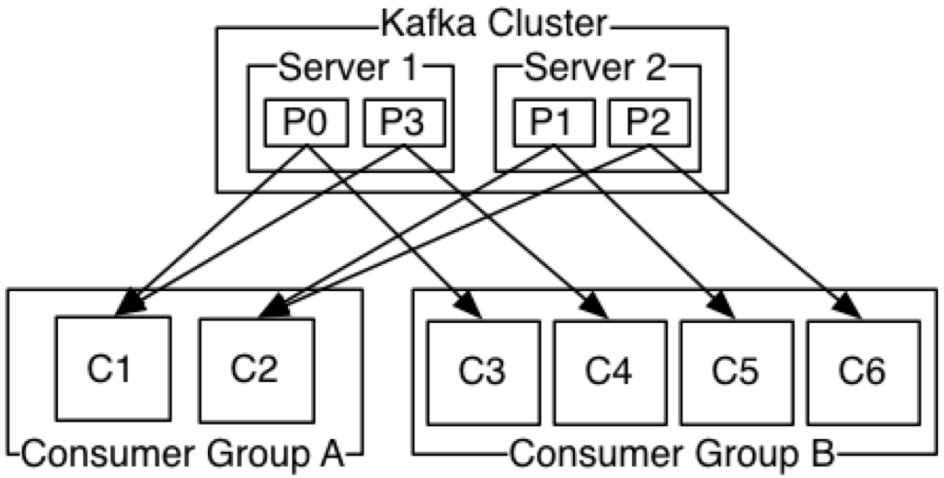
Partitions, replicas, ISRs (In-Sync Replicas), leader and followers



https://www.slideshare.net/junrao/kafka-replication-apachecon2013

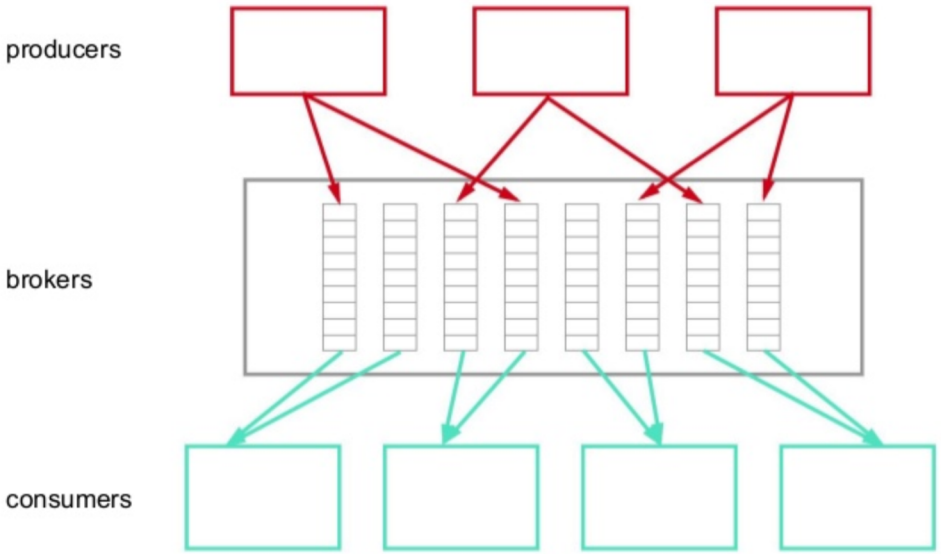
- Each topic should have atleast 1 partition and 1 replica
- Commit log = Partition
- 1 Partition = 1 Replica
- Record is said to be committed when all replicas per partition writes to their log

Consumer group



- Consumer(s) of a topic should belong to a consumer group
- Consumer instances can be in separate processes or on separate machines
- Guaranteed processing of record by atleast one consumer instance
  - Acknowledgement
- No guarantee of order across multiple consumer instances

Core Kafka Pt. 5: How Scalable is Kafka?



- **No bottleneck!**
  - Many brokers
  - Many producers
  - Many consumers
- **Limits?**
  - Internet giants are driving the limits higher - you won't need to worry.
  - e.g. LinkedIn > 1 trillion messages / day through Kafka clusters.
  - 100 brokers / 2 billion messages a day is "straightforward" to operate
  - Don't over partition ~< 100k partitions

https://www.slideshare.net/ConfluentInc/a-tour-of-apache-kafka

Quick start -
Kafka command line tool

Hand-on – Command line

- Download and install Kafka from - https://www.apache.org/dyn/closer.cgi?path=/kafka/2.0.1/kafka_2.11-2.0.1.tgz
- Extract the contents of zip file to a directory in your filesystem
- Follow instructions in http://kafka.apache.org/quickstart
  - Start Zookeeper
  - Start Kafka server/broker

Topic commands

- Create topic

*bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 2 *

--partitions 4 --topic my-replicated-topic

- Describe topic

bin/kafka-topics.sh --describe --zookeeper localhost:2181 --topic my-replicated-topic

- Delete topic

bin/kafka-topics.sh --delete --zookeeper localhost:2181 --topic my-replicated-topic

Pub/Sub –
Kafka Producer/Consumer

Hands-on – Pub/Sub

Base project - https://github.com/shaunthomas999/kafka-workshop-base
Complete project - https://github.com/shaunthomas999/kafka-workshop

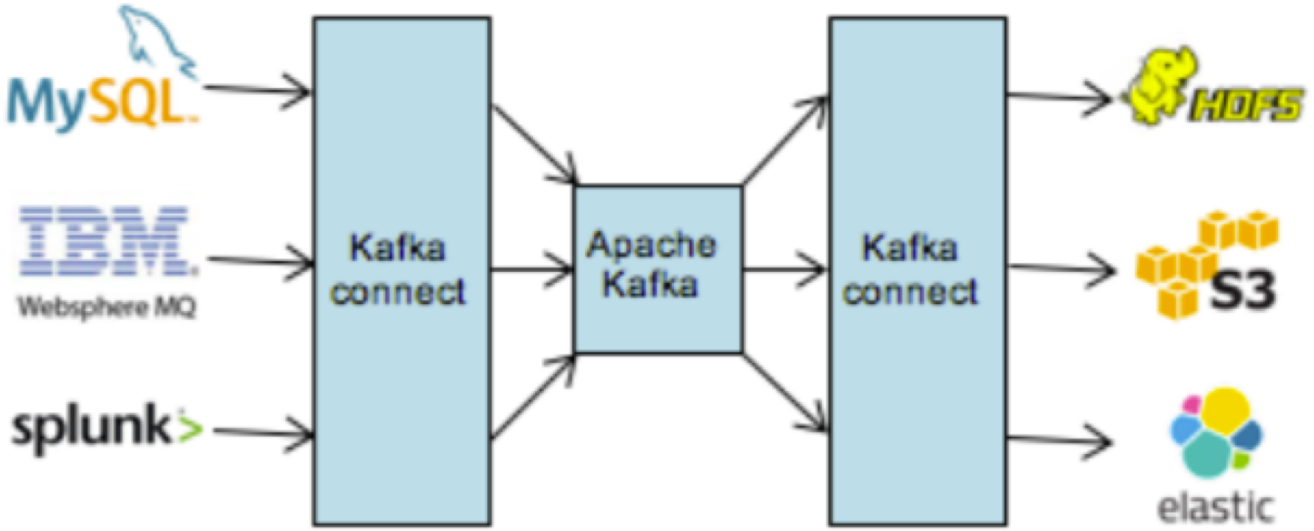
```
StockPrice {
  String name;
  String symbol;
  String type;
  String datetime;
  BigDecimal price;
  String currency;
}
```

Pub/Sub - Advanced

- Messages types
  - Json
  - Protobuf
  - Avro
- Schema Management
  - https://docs.confluent.io/current/schema-management.html

Kafka Connect

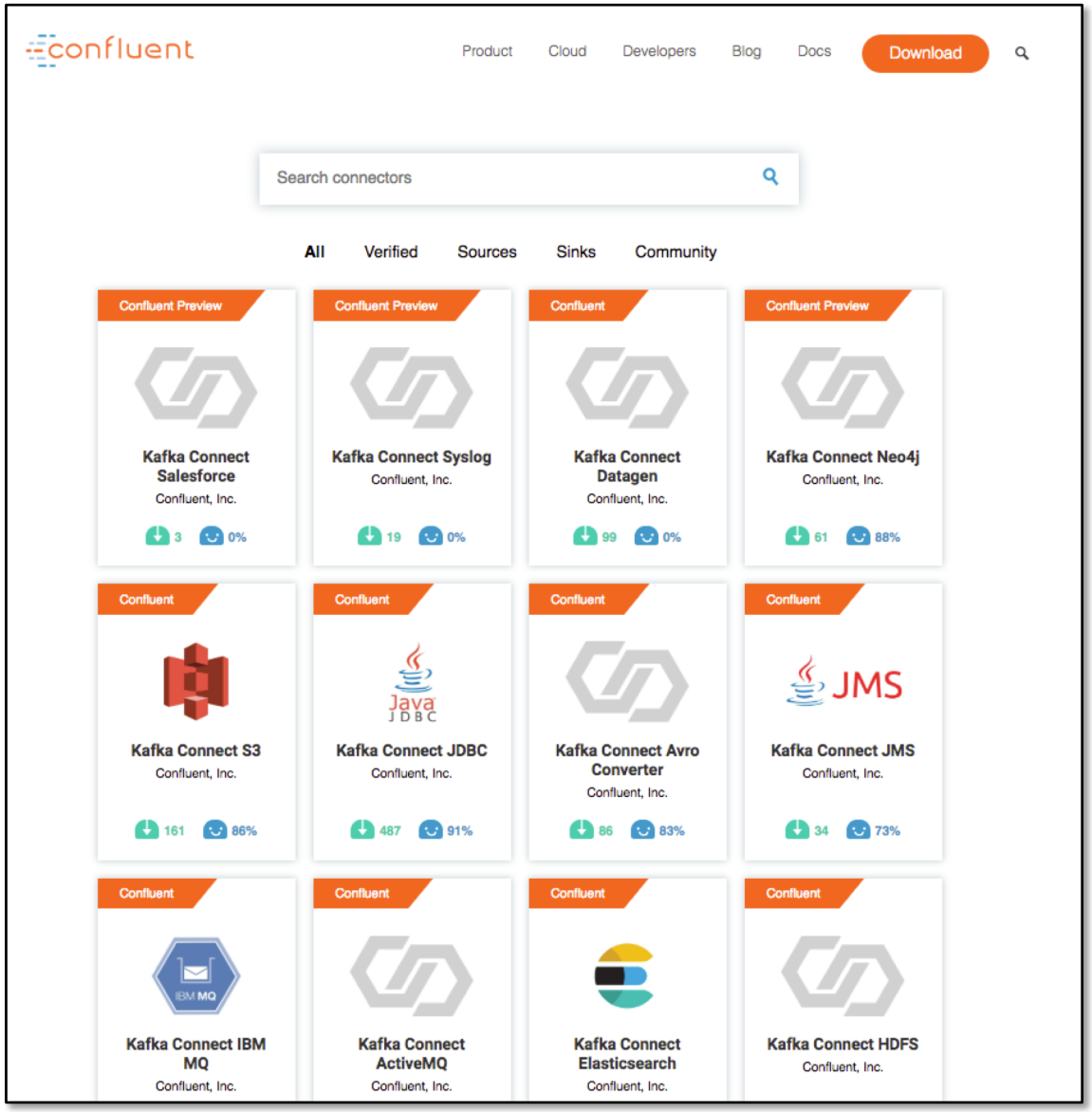
Kafka Connect Overview



Discover Connectors

https://www.confluent.io/hub/

Mind: Sources/Sink



Connect REST API

CONNECT REST API	MEANING
`GET /connectors`	Return a list of active connectors
`POST /connectors`	Create a new connector
`GET /connectors/{name}`	Get the information of a specific connector
`GET /connectors/{name}/config`	Get configuration parameters for a specific connector
`PUT /connectors/{name}/config`	Update configuration parameters for a specific connector
`GET /connectors/{name}/status`	Get the current status of the connector

- Typically runs in distributed mode
- Connect can be managed through REST API

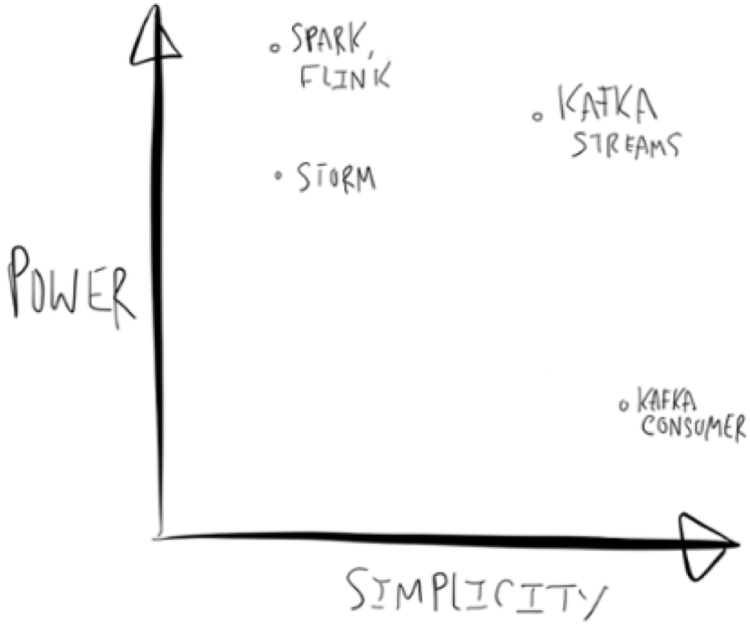
Kafka Streams

Stream Processing Introduction

- Stream Processing is a Big data technology
- Query continuous data stream and detect conditions fast within a small time period from the time of receiving the data
- It is also called by many names: real-time analytics, streaming analytics, complex event processing, real-time streaming analytics, and event processing

https://medium.com/stream-processing/what-is-stream-processing-1eadfca11b97

Streaming platforms comparison



https://www.confluent.io/blog/introducing-kafka-streams-stream-processing-made-simple/

Kafka Streams Types

- KStreams and KTables (high level)
  - Stream processing DSL
  - Offers high level operators like filter, map, grouping, aggregation, joins, windowing etc.
- Stream Processors (low level)
  - Can create custom operators
  - Low level development approach
- KSQL
  - Streaming SQL for Apache Kafka
  - https://www.confluent.io/product/ksql/

KStreams vs KTables

	KSTREAM	KTABLE
CONCEPT	Each record is treated as an append to the stream.	Each record is treated as an update to an existing key
USAGE	Model append-only data such as click streams.	Model updatable reference data such as user profiles.

(key , value) records	Sum of values As KStream	Sum of values As KTable
(“k1”, 2) (“k1”, 5)	7	5

Hands-on – KStream

Method Summary

All Methods | **Instance Methods** | **Abstract Methods**

Modifier and Type	Method	Description
`KStream<K,V>[]`	`branch(Predicate<? super K,? super V>... predicates)`	Creates an array of `KStream` from this stream by branching the records in the original stream based on the supplied predicates.
`KStream<K,V>`	`filter(Predicate<? super K,? super V> predicate)`	Create a new `KStream` that consists of all records of this stream which satisfy the given predicate.
`KStream<K,V>`	`filterNot(Predicate<? super K,? super V> predicate)`	Create a new `KStream` that consists all records of this stream which do *not* satisfy the given predicate.
`<KR,VR> KStream<KR,VR>`	`flatMap(KeyValueMapper<? super K,? super V,? extends java.lang.Iterable<? extends KeyValue<? extends KR,? extends VR>>> mapper)`	Transform each record of the input stream into zero or more records in the output stream (both key and value type can be altered arbitrarily).
`<VR> KStream<K,VR>`	`flatMapValues(ValueMapper<? super V,? extends java.lang.Iterable<? extends VR>> mapper)`	Create a new `KStream` by transforming the value of each record in this stream into zero or more values with the same key in the new stream.
`<VR> KStream<K,VR>`	`flatMapValues(ValueMapperWithKey<? super K,? super V,? extends java.lang.Iterable<? extends VR>> mapper)`	Create a new `KStream` by transfor into zero or more values with the s
`void`	`foreach(ForeachAction<? super K,? super V> action)`	Perform an action on each record
`<KR> KGroupedStream<KR,V>`	`groupBy(KeyValueMapper<? super K,? super V,KR> selector)`	Group the records of this `KStrea` provided `KeyValueMapper` and
`<KR> KGroupedStream<KR,V>`	`groupBy(KeyValueMapper<? super K,? super V,KR> selector, Serialized<KR,V> serialized)`	Group the records of this `KStrea` provided `KeyValueMapper` and s
`KGroupedStream<K,V>`	`groupByKey()`	Group the records by their current the original values and default seri
`KGroupedStream<K,V>`	`groupByKey(Serialized<K,V> serialized)`	Group the records by their current the original values and using the s

https://kafka.apache.org/20/javadoc/org/apache/kafka/streams/kstream/KStream.html

https://github.com/shaunthomas999/kafka-workshop

Architectures

confluent

Product Cloud Developers Blog Docs Download

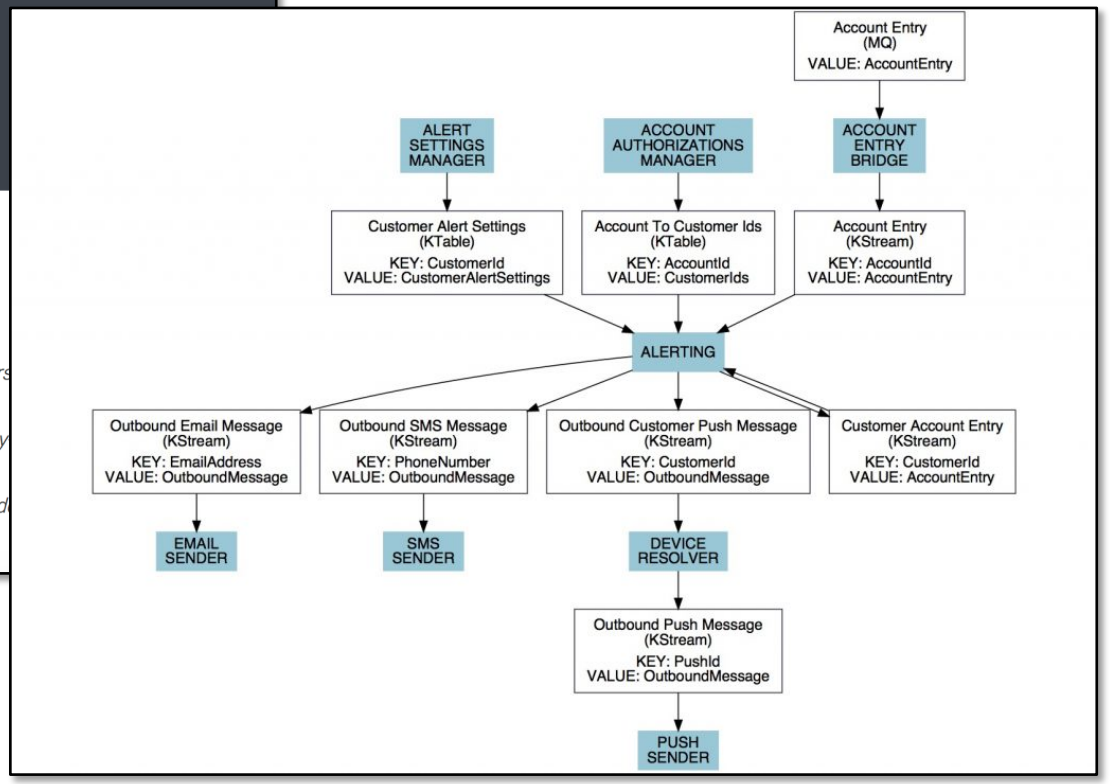
STREAM PROCESSING

Real-time Financial Alerts at Rabobank with Apache Kafka's Streams API

Jeroen van Disseldorp

July 20, 2017

This article discusses the use of Apache Kafka's Streams API for sending out alerts to customers
Rabobank. Rabobank is based in the Netherlands with over 900 locations worldwide, 48,000
employees, and €681B in assets. It is a bank by and for customers, a cooperative bank, a socially
responsible bank. It aims to be market leader across all financial markets in the Netherlands.
Rabobank is also committed to being a leading bank in the field of food and agriculture worldwid
Rabobank provides financial products and services to millions of customers around the world.



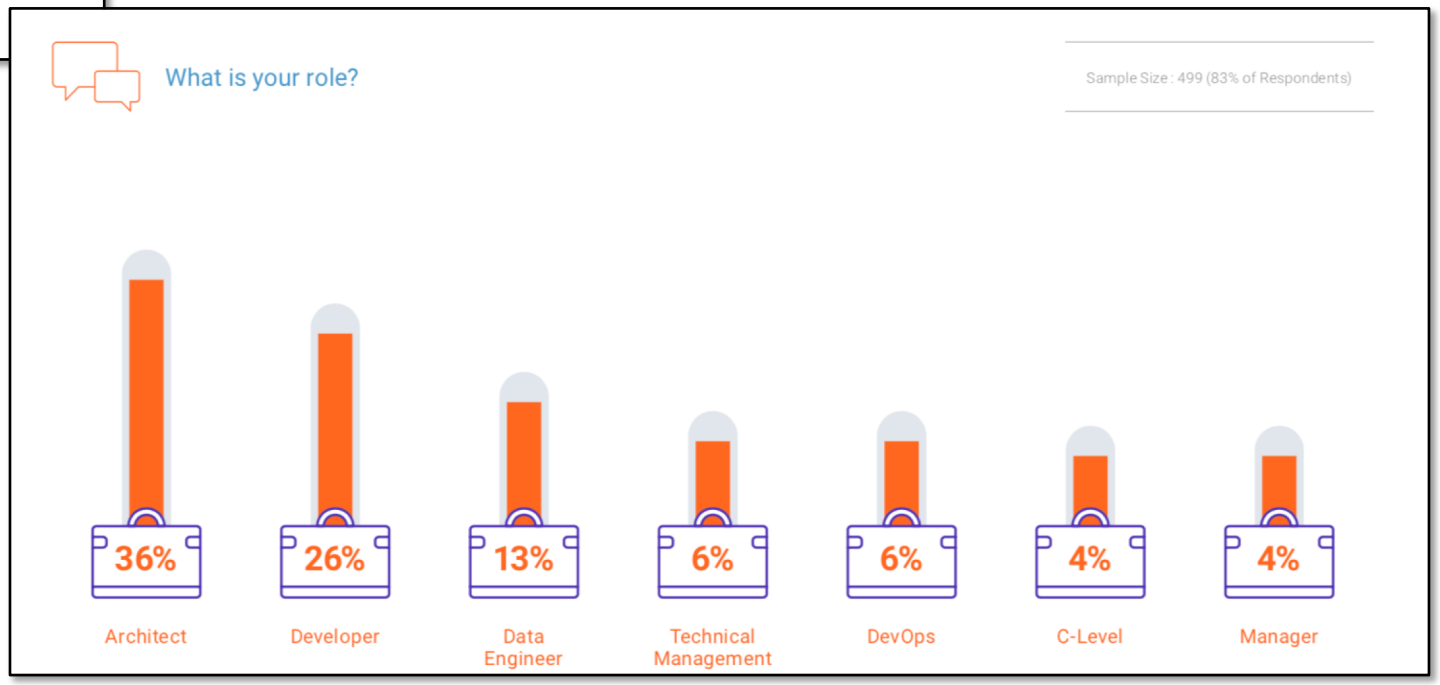
https://www.confluent.io/blog/real-time-financial-alerts-rabobank-apache-kafkas-streams-api/

Survey results

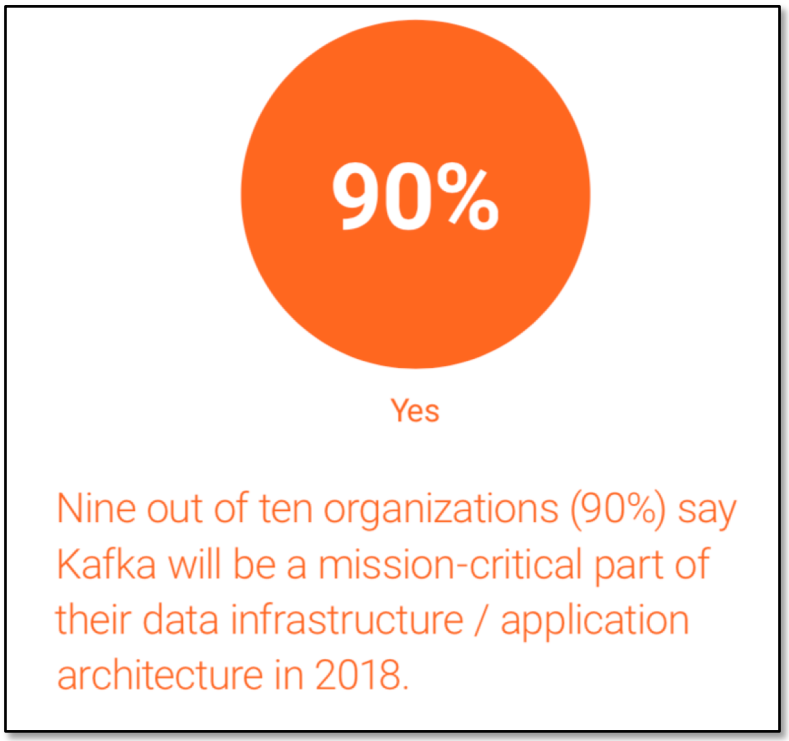
2018 Apache Kafka Report

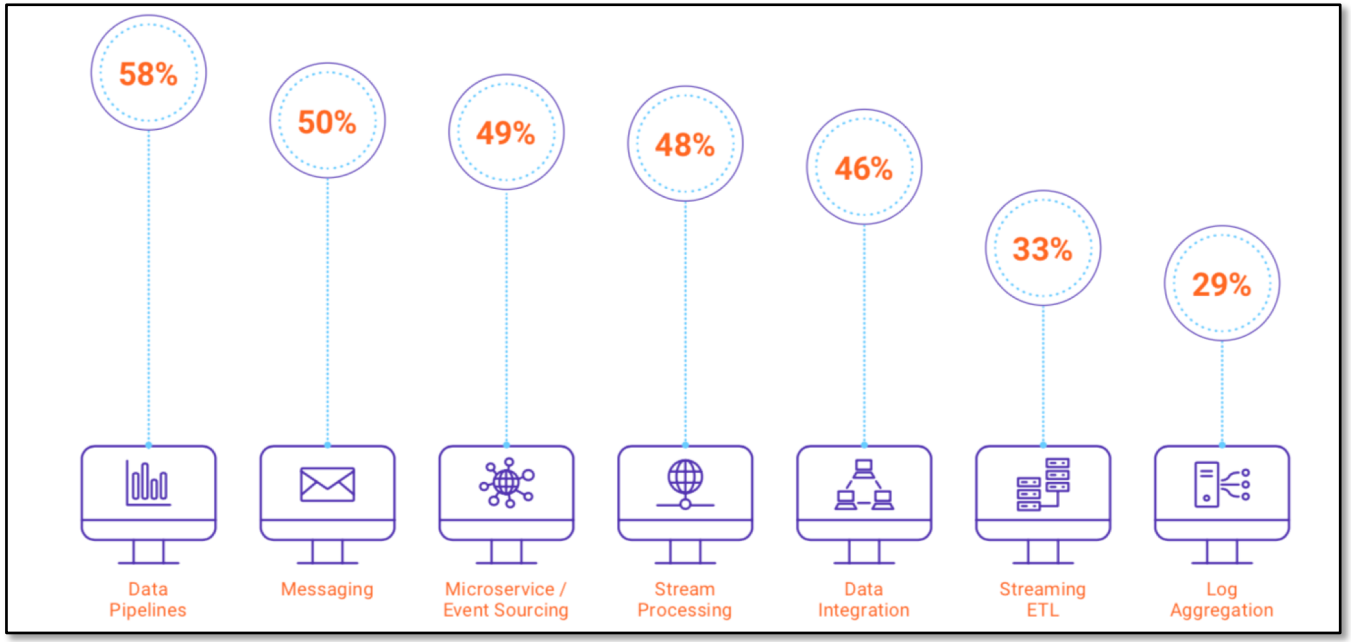
Confluent surveyed over 600 Apache Kafka users from 59 countries in order to better understand the behavior, usage and attitudes of Kafka's growing user base.

https://www.confluent.io/apache-kafka-report/



Important results/trends





Contact

shaunthomas999@gmail.com

www.shaunthomas999.com

Twitter / LinkedIn / Facebook: shaunthomas999