Stream Processing Basics

Ignacio (Nacho) Solis
Messaging in Distributed Systems
Messages

- Finite
- Structured / Unstructured
- Immutable
Use case 1

- Sender - Send data to receiver
- Receiver - Receive data
- Sync / Blocking
Use case 2

• Sender - Send data to receiver
• Receiver - Receive data
• Async
Use case 3

- Publishers - Publish data to topic
- Subscribers - Read data from topic
- Async
Messaging

- Multiple senders / receivers
- Scalable
- Guarantees
Guarantees

- Fault Tolerant
- Ordered
- Reliable
- Exactly-once
Kafka
Kafka

- Pub/Sub
- Data organized in topics
- Topics divided into partitions
Kafka Guarantees

- Fault tolerant
- Order by sender/partition
- Reliable
- At-least/exactly once
Kafka messages

- Key (bytes)
- Value (bytes)
- Schema*
- Registry*
Kafka clusters

- Leader + Replicas
- Offset / partition
- Zero downtime
- Zookeeper coordination
Kafka clients

- Batching
- Compression
- Async
- Group coordination
Kafka protocol

- Not!
- Request + Response
- Evolution (?)
- Headers!
Kafka cache

- Replay
- Impedance matching
- Retention
- Compaction
Apache Kafka

- Client/Broker
- Connect
- Streams
Apache Samza

- Stream Processing
- Stateless / Statefull
- Consistent
- At-least once