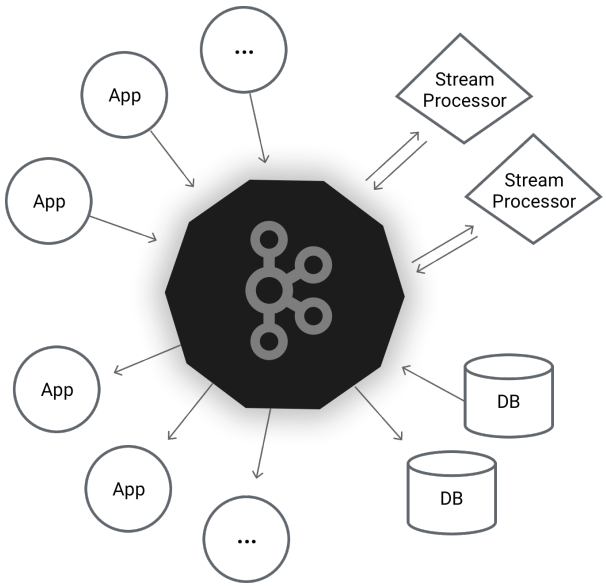
Apache Kafka

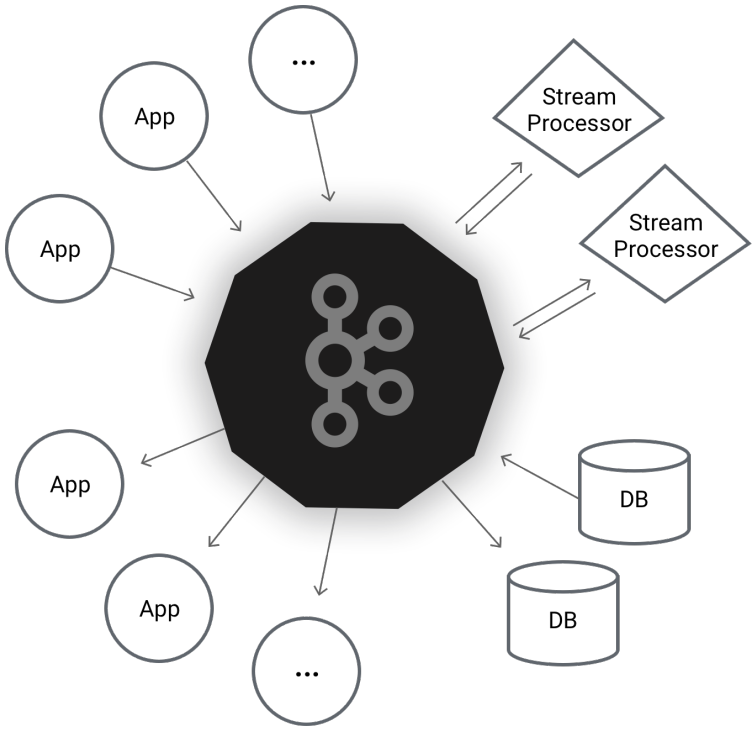
Daniel, Fabian, Hauke und Tom

Modellierung von Informationssystemen
Department Informatik
HAW Hamburg



01. Dezember 2017

Was ist Apache Kafka?



Apache Kafka ist eine verteilte skalierbare Streaming Plattform.

Eigenschaften

Kafka ...

- ist ein Message Queuing System
- kann Nachrichten speichern
- kann Nachrichten verarbeiten
- kann all das in Echtzeit

Unternehmen und Use Cases

Operational Metrics

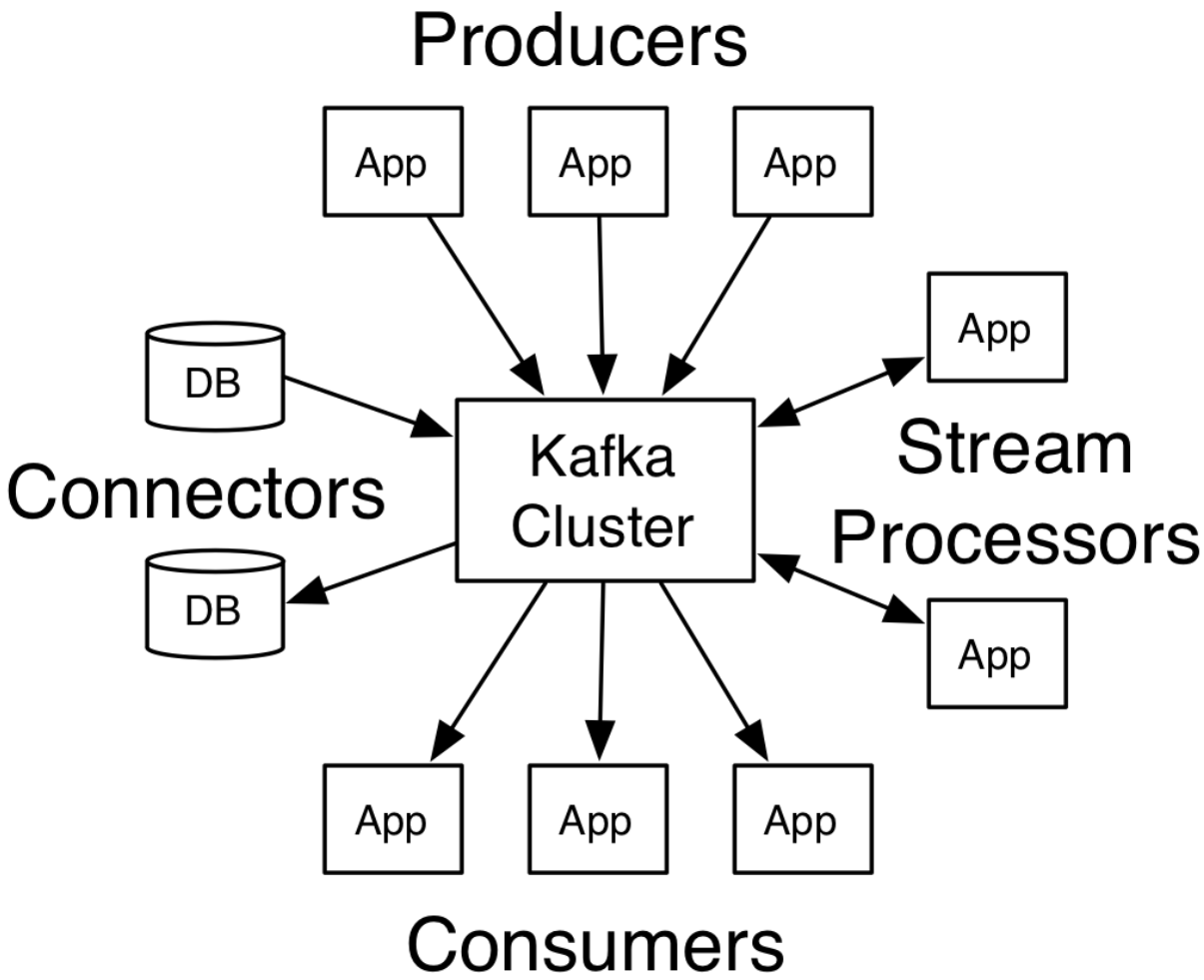
OpenSOC (Security Operations Center)

Real-time monitoring and event-processing pipeline

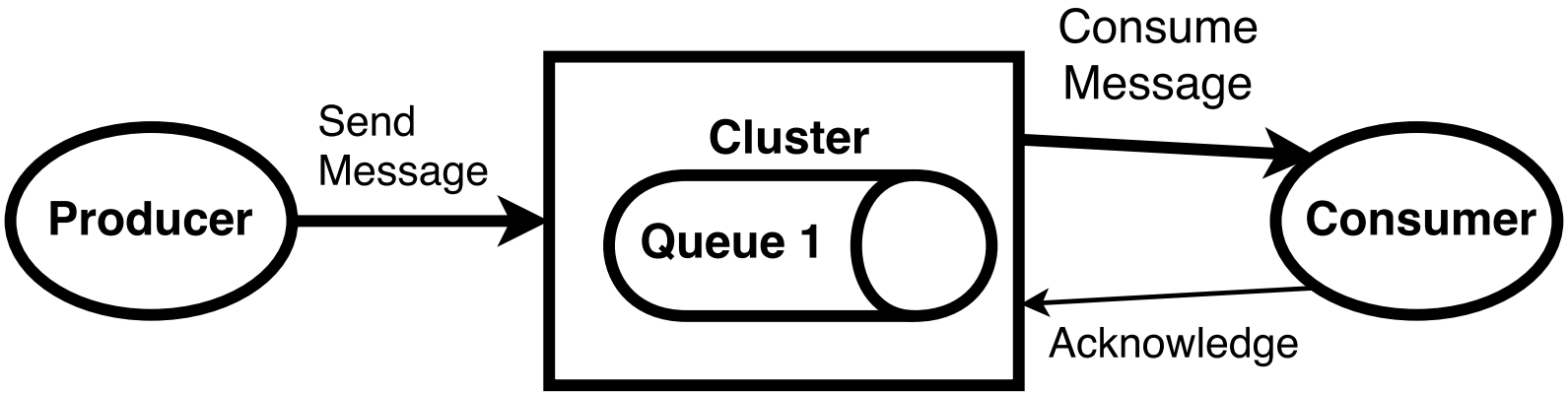
Log Delivery System

Part of Storm stream processing infrastructure

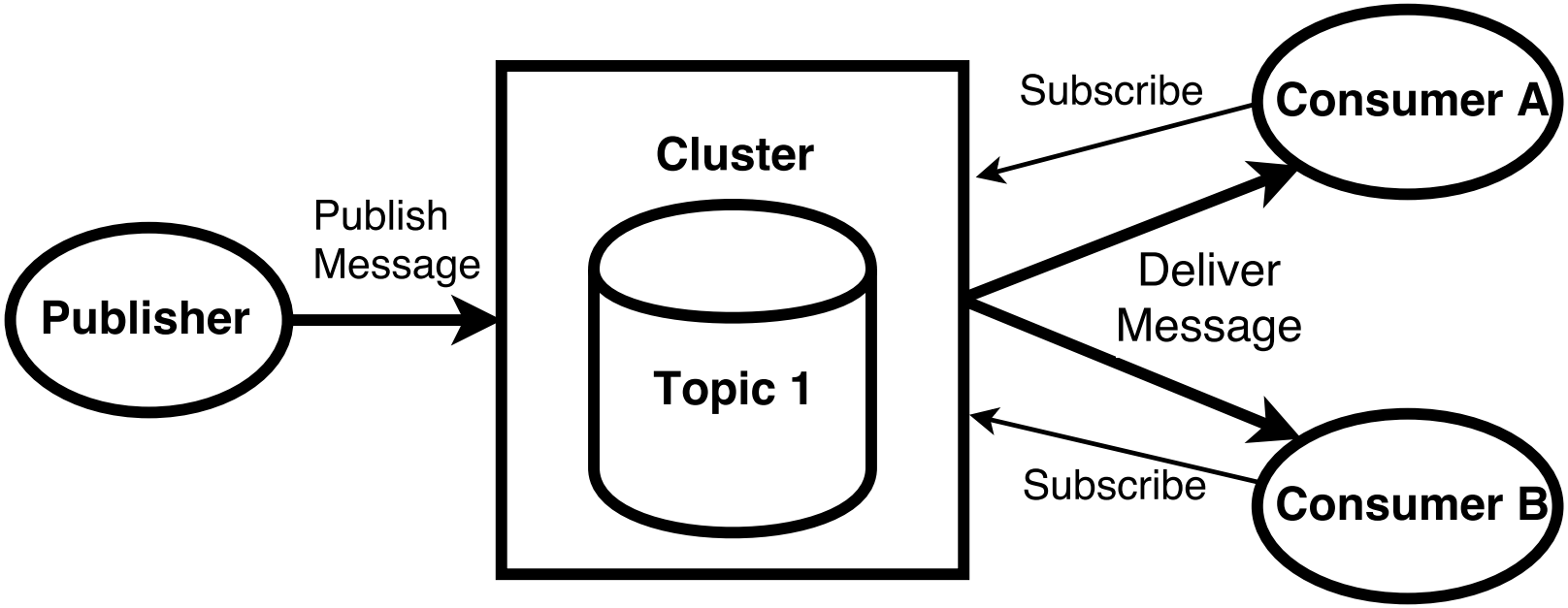
Komponenten



Queue


Producer
Send
Message
Cluster
Queue 1
Consume
Message
Consumer
Acknowledge

Topic

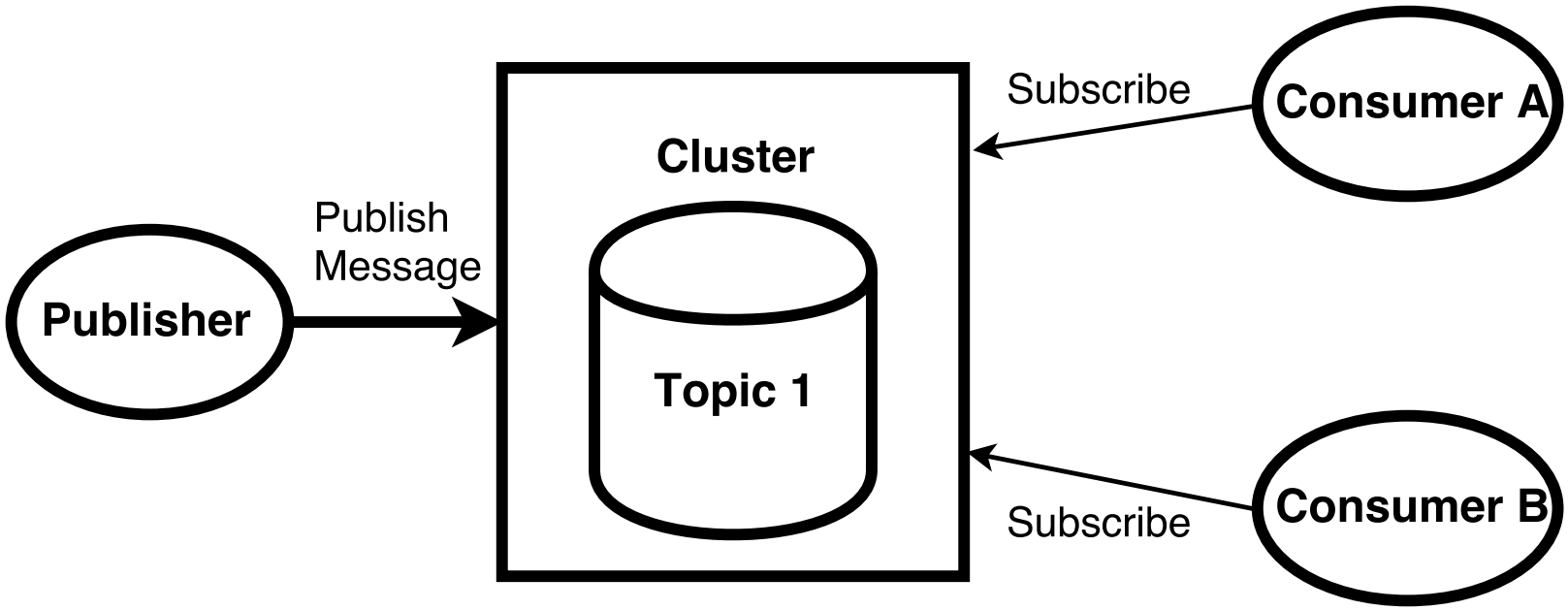

Publisher
Publish
Message
Cluster
Topic 1
Subscribe
Consumer A
Deliver
Message
Subscribe
Consumer B

Kafka als Nachrichtensystem

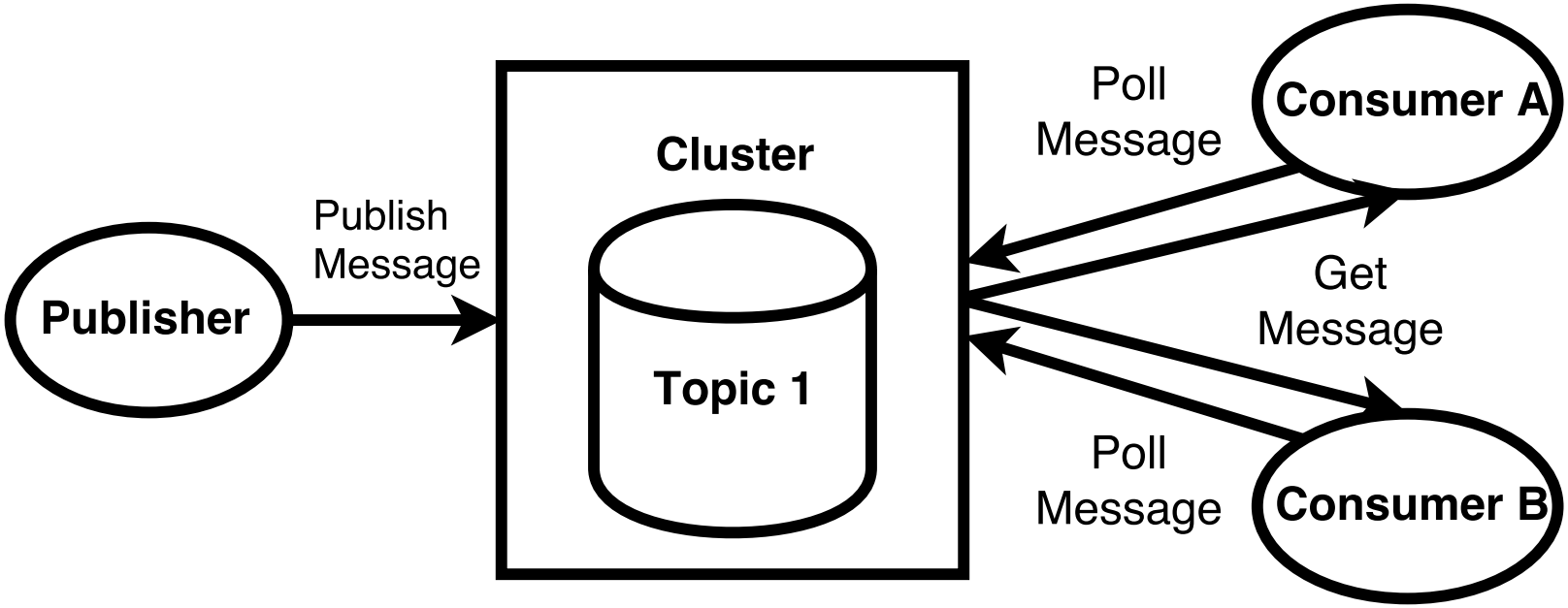
Bisher:

- Queueing
  - Nachrichten an einen
  - Nachrichtenverarbeitung skaliert
  - Nachricht abgerufen = Nachricht weg
- Publish-Subscribe
  - Nachrichten an alle
  - Skaliert nicht

Kafka Topic


Publisher
Publish
Message
Cluster
Topic 1
Subscribe
Consumer A
Subscribe
Consumer B

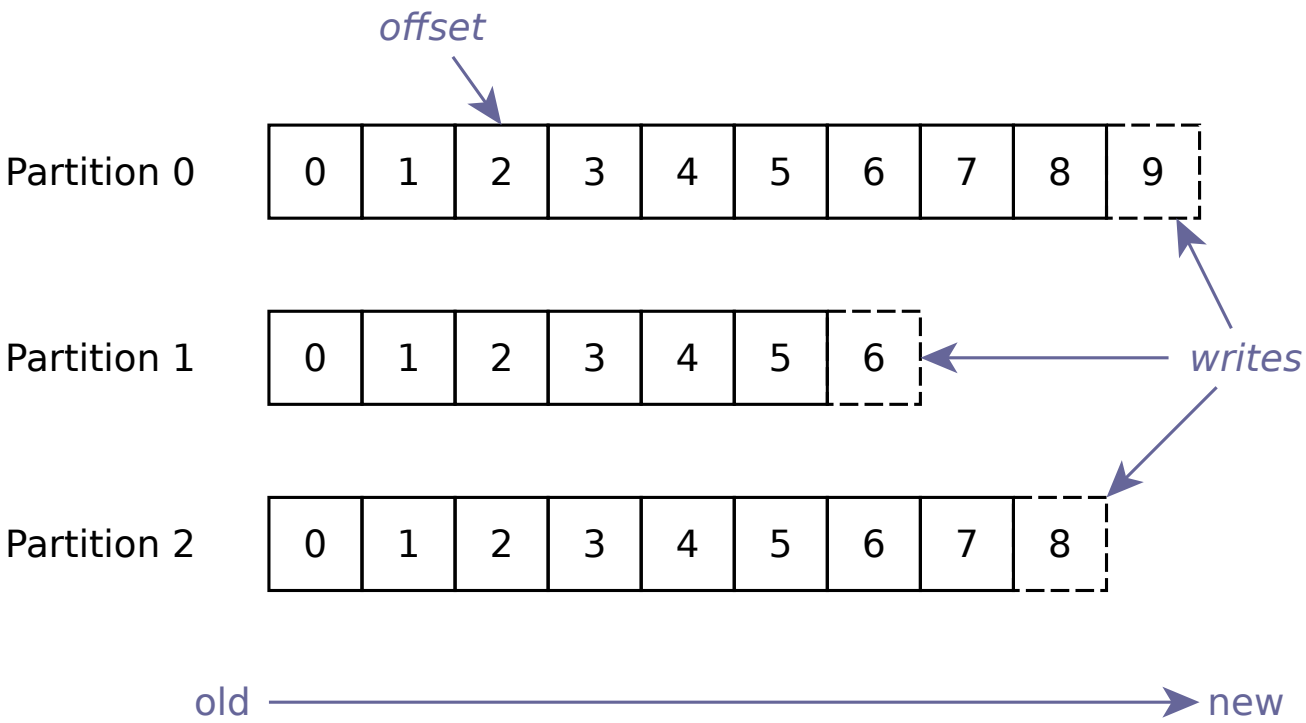
Kafka Topic


Publisher
Publish
Message
Cluster
Topic 1
Poll
Message
Consumer A
Get
Message
Poll
Message
Consumer B

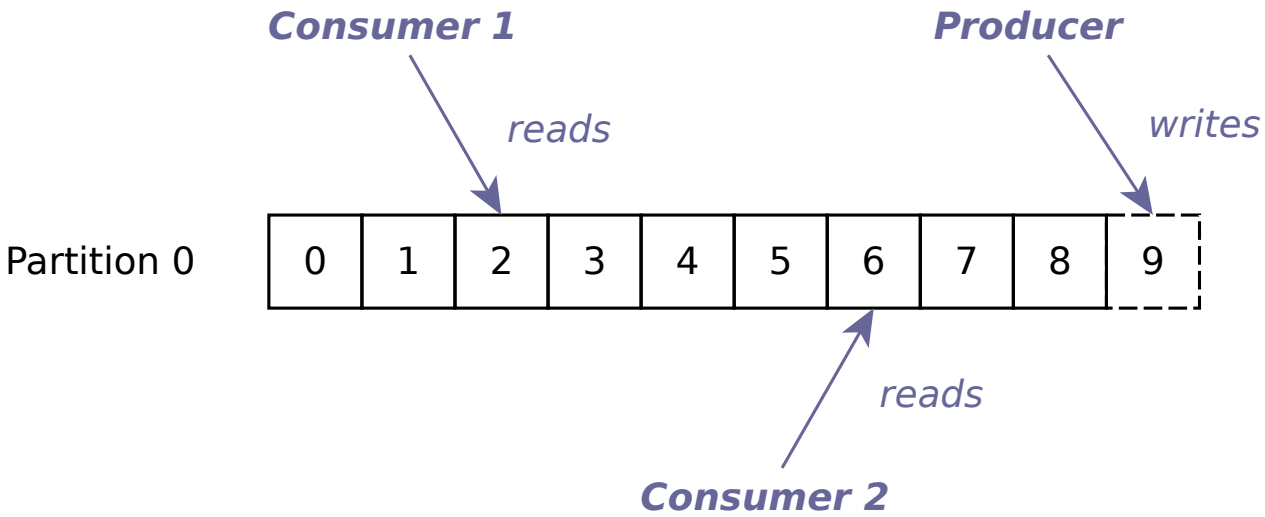
Kafka Topics

- Multi-Subscribe (0 bis n Consumer)
- Kein Push-System
- Records in Topics werden persistent gehalten
- Topics benötigen eine Cleanup-Policy
  - Retention-Time
  - Retention-Size
  - Log-Compaction
- Topics besitzen Partitionen (partition log)
- Guarantees (dazu später mehr)

Partitionen



Partitionen



Partitionen

- Eine Partition für jedes Topic
- Eine Partition ist
  - Geordnet
  - Nicht-Veränderbare Sequenz von Records
  - Records können angehängt werden
- Records sind nummeriert
- Records nach Cleanup-Policy entfernt
- Sequentielle Abarbeitung ist Standart
- Sprung im Record-Log möglich

Partitionen

- Skalierende Loggrößen ermöglicht
- Parallelität wird ermöglicht
- Können verteilt werden

Kafka als Nachrichtensystem

Bisher:

- Queueing
  - Nachrichten an einen
  - Nachrichtenverarbeitung skaliert
  - Nachricht abgerufen = Nachricht weg
- Publish-Subscribe
  - Nachrichten an alle
  - Skaliert nicht

Kafka als Nachrichtensystem

- Consumer Groups
  - Kombiniert Queueing und Publish-Subscribe
  - Nachrichtenverarbeitung in Gruppen
  - Mehrere Consumer in einer Gruppe
- Vorteile
  - Nachrichtenverarbeitung skaliert
- Reihenfolge wird eingehalten

Parallelität

- Ordnung
  - Gesichert für alle Consumer Groups
- Lastverteilung
  - Nachricht 1x pro Consumer Group verarbeitet

Kafka als Datenbank

- Durch Funktionalität bedingt
  - Entkopplung sorgt für Speicherbedarf
- Daten werden repliziert
  - Bestätigungsmechanismen sind vorhanden
  - Wird erst bestätigt, wenn Replication abgeschlossen ist

Kafka als Datenbank - 2

- Performanz bei steigender Datenmenge gleich
- Eigenschaften:
  - Hohe Performanz
  - Geringe Latenz
  - Replikation
  - Weiterleitung

Kafka für Streams

- Echtzeit Stream-Verarbeitung
- Ein Stream Processor:
  - Nimmt kontinuierlich Daten aus einem Topic
  - Bearbeitet die Daten
  - Schreibt kontinuierlich Daten in ein Topic

Kafka für Streams - 2

- Extra Stream-API wird angeboten
  - Ermöglicht komplexere Operationen
  - Bearbeitet die Daten
  - Schreibt kontinuierlich Daten in ein Topic
- Kann auch umgehen mit:
  - Daten die nicht in Reihenfolge sind
  - Daten neu verarbeiten wenn sich die Operation ändert
  - Status behaftete Operationen sind möglich

Zusammenfassung

- geeignet für:
  -
  - Bearbeitet die Daten
  - Schreibt kontinuierlich Daten in ein Topic

Tutorial