vijfhart
IT-OPLEIDINGEN

Dat klopt voor jou!

Event Streaming with Kafka

Overview

- 9.00-9.15: Introduction round
- 9. 15-10.00 Kafka theory
- 10.00-10.10 Coffee refill
- 10.10-10.30 Exercise
- 10.30-11.00 Interactive demo 1
- 11.00-11.10 Coffee refill
- 11.10-11.40 Interactive demo 2
- 11.40-12.00 Quiz + Wrap up

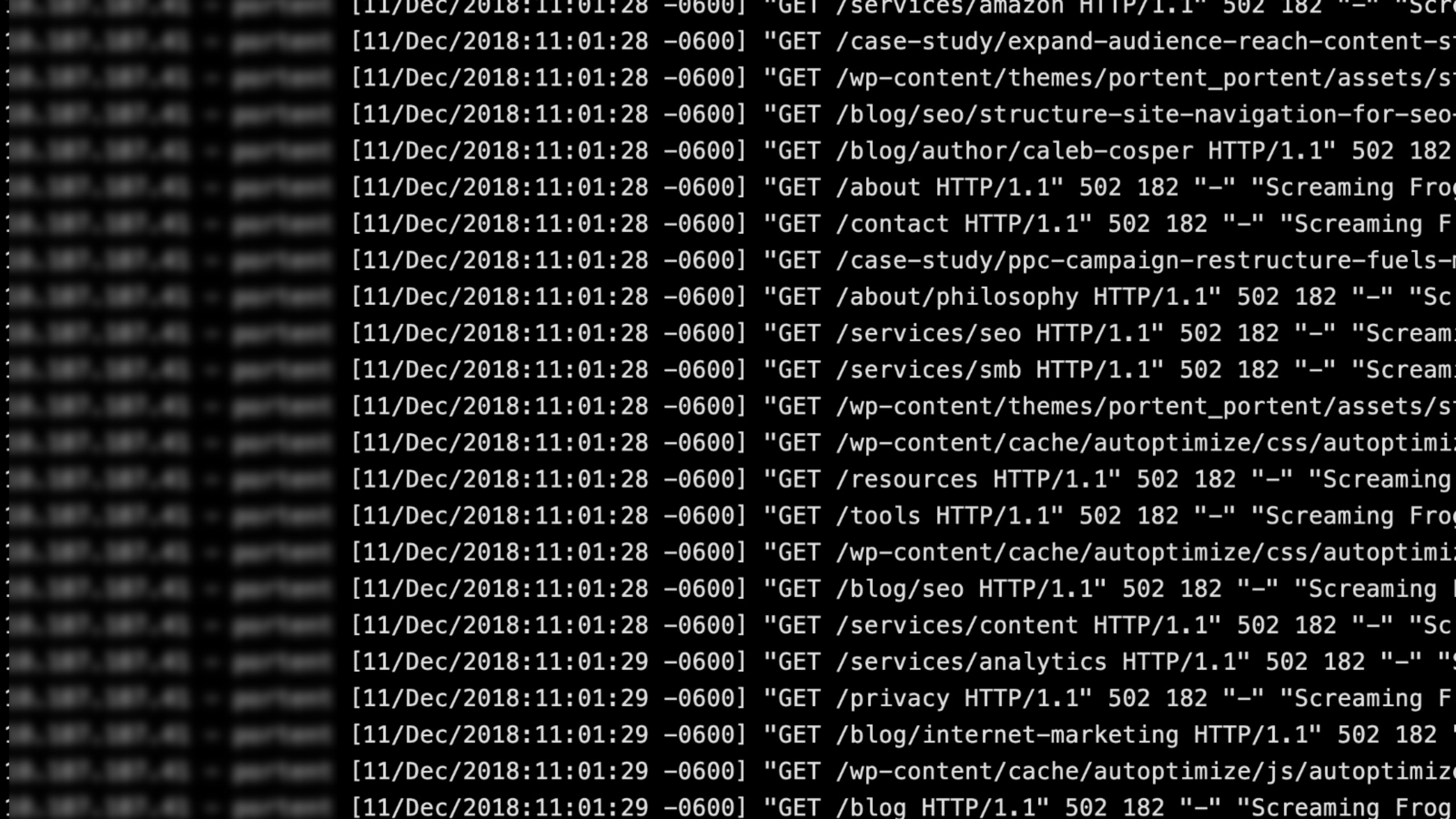
Introduction round

- Name, job, background
- Hobbies, interests, family, pets, where do you live, etc
- What do you hope to learn during this workshop?

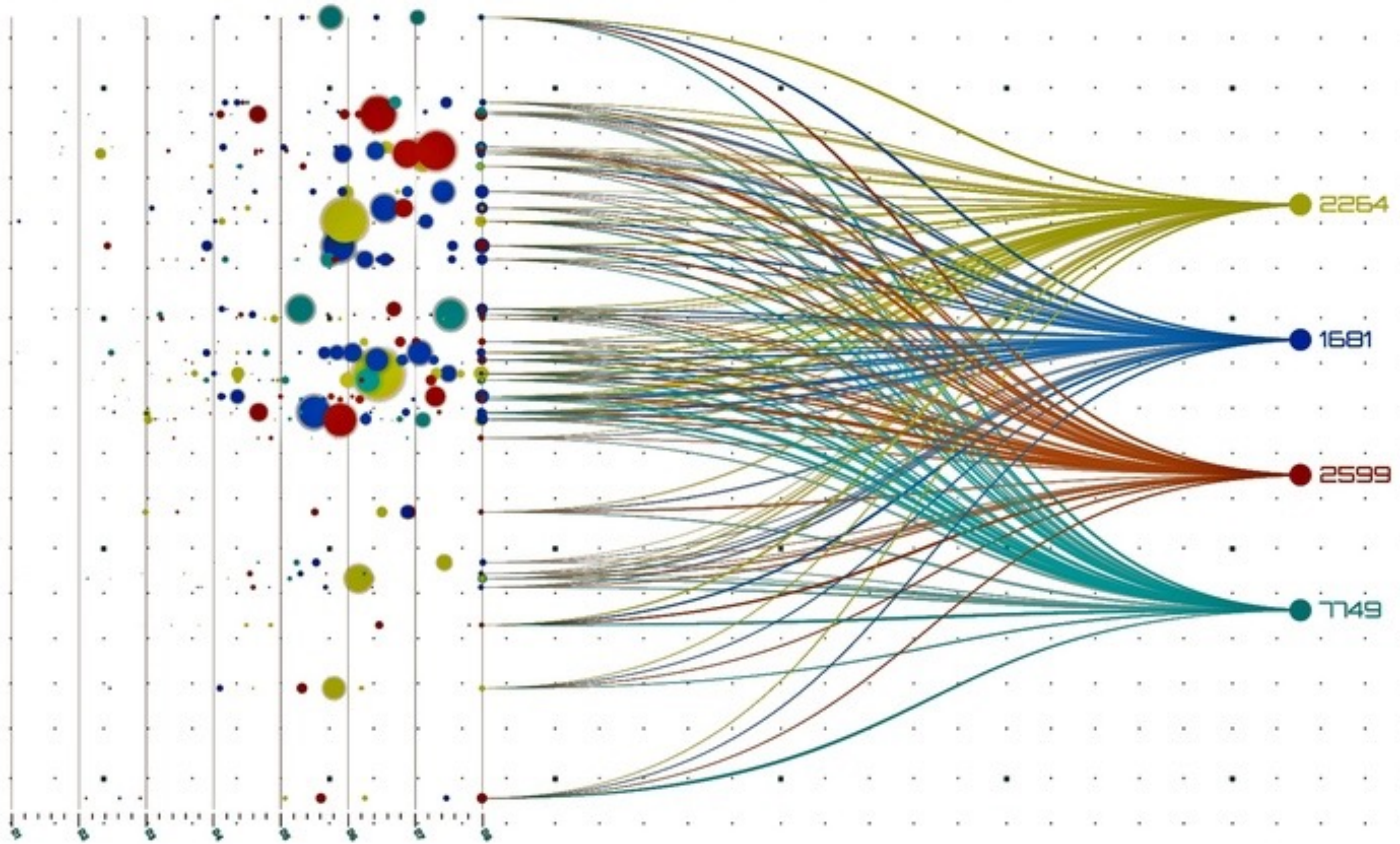
What is data streaming?

- Continuous transfer and processing of data
- Real-time or near-real-time handling
- Low-latency insights and responses
- Applications: real-time analytics, monitoring, decision-making

14
#
1
8



```
[redacted] [11/Dec/2018:11:01:28 -0600] "GET /services/amazon HTTP/1.1" 502 182 "-" "Scr
[redacted] [11/Dec/2018:11:01:28 -0600] "GET /case-study/expand-audience-reach-content-s
[redacted] [11/Dec/2018:11:01:28 -0600] "GET /wp-content/themes/portent_portent/assets/s
[redacted] [11/Dec/2018:11:01:28 -0600] "GET /blog/seo/structure-site-navigation-for-seo
[redacted] [11/Dec/2018:11:01:28 -0600] "GET /blog/author/caleb-cosper HTTP/1.1" 502 182
[redacted] [11/Dec/2018:11:01:28 -0600] "GET /about HTTP/1.1" 502 182 "-" "Screaming Fro
[redacted] [11/Dec/2018:11:01:28 -0600] "GET /contact HTTP/1.1" 502 182 "-" "Screaming F
[redacted] [11/Dec/2018:11:01:28 -0600] "GET /case-study/ppc-campaign-restructure-fuels-
[redacted] [11/Dec/2018:11:01:28 -0600] "GET /about/philosophy HTTP/1.1" 502 182 "-" "Sc
[redacted] [11/Dec/2018:11:01:28 -0600] "GET /services/seo HTTP/1.1" 502 182 "-" "Screami
[redacted] [11/Dec/2018:11:01:28 -0600] "GET /services/smb HTTP/1.1" 502 182 "-" "Screami
[redacted] [11/Dec/2018:11:01:28 -0600] "GET /wp-content/themes/portent_portent/assets/s
[redacted] [11/Dec/2018:11:01:28 -0600] "GET /wp-content/cache/autoptimize/css/autoptimi
[redacted] [11/Dec/2018:11:01:28 -0600] "GET /resources HTTP/1.1" 502 182 "-" "Screaming
[redacted] [11/Dec/2018:11:01:28 -0600] "GET /tools HTTP/1.1" 502 182 "-" "Screaming Fro
[redacted] [11/Dec/2018:11:01:28 -0600] "GET /wp-content/cache/autoptimize/css/autoptimi
[redacted] [11/Dec/2018:11:01:28 -0600] "GET /blog/seo HTTP/1.1" 502 182 "-" "Screaming
[redacted] [11/Dec/2018:11:01:28 -0600] "GET /services/content HTTP/1.1" 502 182 "-" "Sc
[redacted] [11/Dec/2018:11:01:29 -0600] "GET /services/analytics HTTP/1.1" 502 182 "-" "
[redacted] [11/Dec/2018:11:01:29 -0600] "GET /privacy HTTP/1.1" 502 182 "-" "Screaming F
[redacted] [11/Dec/2018:11:01:29 -0600] "GET /blog/internet-marketing HTTP/1.1" 502 182
[redacted] [11/Dec/2018:11:01:29 -0600] "GET /wp-content/cache/autoptimize/js/autoptimiz
[redacted] [11/Dec/2018:11:01:29 -0600] "GET /blog HTTP/1.1" 502 182 "-" "Screaming Frog
```


2264
1681
2599
7749

What is event streaming?

- Subset of data streaming focused on individual events
- Events represent changes or actions within a system
- Timestamp, metadata, and payload in each event
- Event-driven architectures
- Real-time processing, analysis, and reaction to events

What is Kafka?

Apache Kafka is an open-source, distributed streaming platform designed to handle high-throughput, fault-tolerant, and scalable real-time data streaming.

Key features

1 Fault-tolerance

2 Scalability

3 Durability

4 Low-latency

5 Stream processing

vijfhart IT-OPLEIDINGEN

Dat klopt voor jou!

Kafka use cases

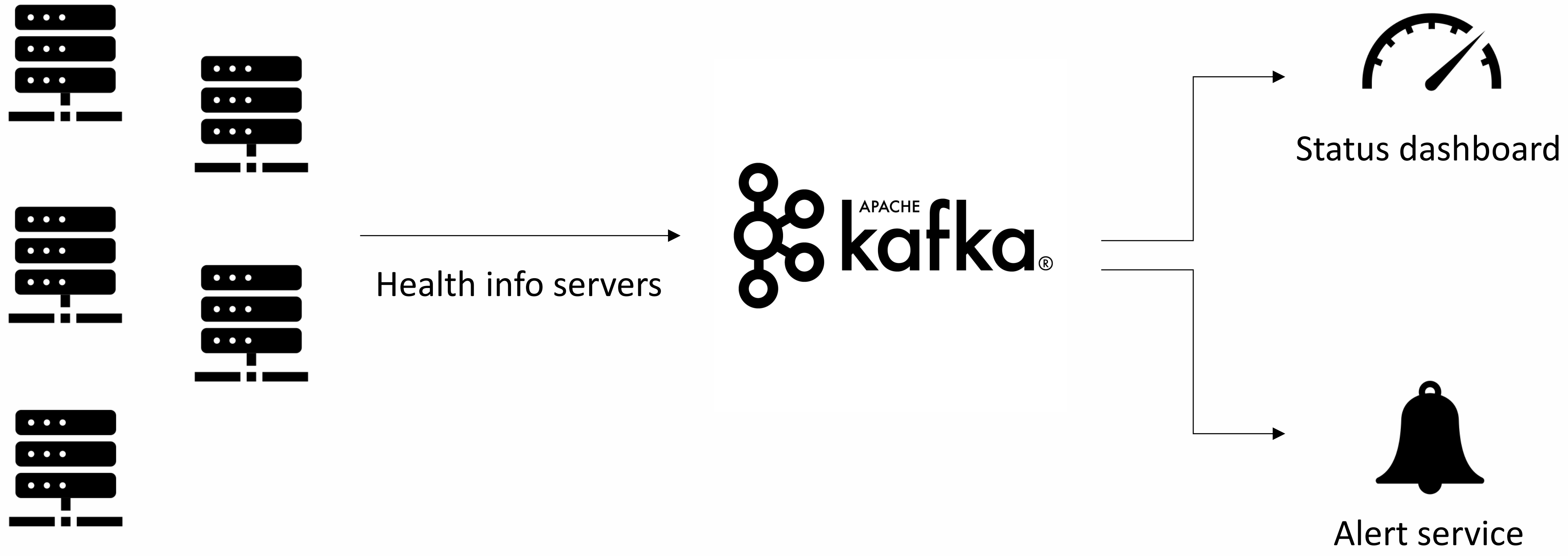
1 Real-time data processing

2 Event-driven architectures

3 Log aggregation

4 Messaging system for microservices

Kafka use case example



Kafka and KPN

1 Increasing demand for realtime data

2 Therefore, Kafka use is encouraged

3 Kafka is used quite a bit already: DSP, Service guard, data service hub (DSH), Udex department

vijfhart IT-OPLEIDINGEN

Dat klopt voor jou!

Overview Kafka concepts

1 Topics

2 Partitions

3 Broker

4 Cluster

5 Replication factor

6 Producer

7 Message

8 Consumer

9 Consumer group

10 Offsets

11 Zookeeper

Topics

1 Topic: stream of data, comparable to a table in a database

2 Topic has a name

3 As many topics as needed can be made

Partitions

Partitions: topics are divided into partitions

1. Partition is ordered, only within same partition
2. Partition contains messages
3. Each message has an auto incremented id, this is called offset
4. Partitions don’t need to have the same length
5. Operate independently of each other, same offset id in two partitions are not related
6. Data in a partition is immutable
7. Data in partition will be removed after certain time (default: week)

Kafka broker

1 Broker: a server with an ID that contains some topic partitions

2 Every broker is a bootstrap server. Each broker knows about all brokers, topics and partitions. This data is given to the caller. So it knows how to find the partitions on other brokers in same cluster.

Kafka cluster

1 Cluster: consist of multiple brokers

2 When you connect to the Kafka broker, you connect to the entire cluster

Topic replication factor

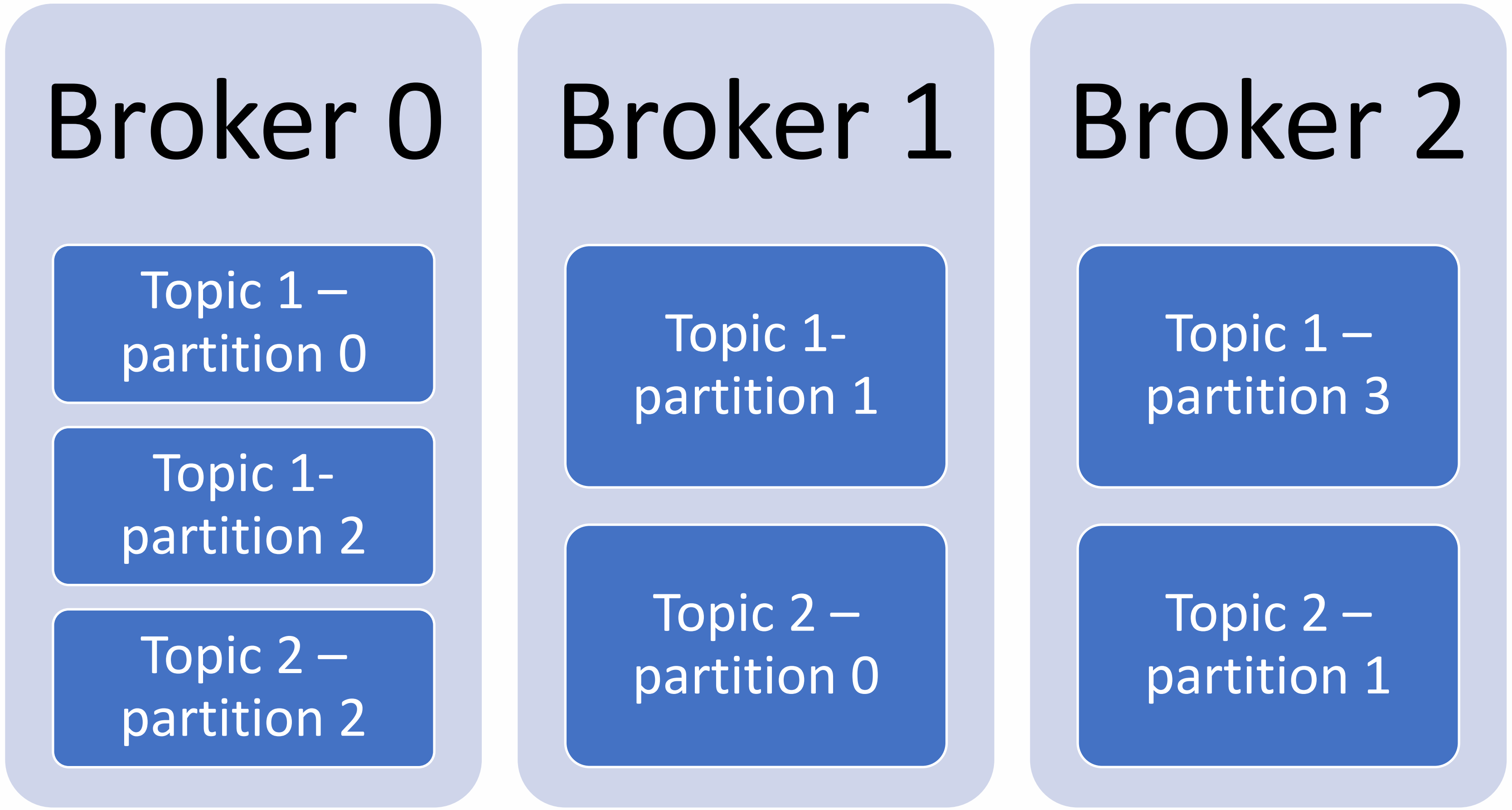
1 in order to avoid data loss when one broker goes down, topics can be backed up on other brokers.

2 Every partition has one leader. This one receives and serves messages for that partition. The backup broker syncs with the leader. When broker with leader goes down, the other broker becomes the leader of that partition

3 The image on the right has replication factor 1, this means, no replicates. Replication factor X means that X-1 brokers can be down without data loss.

4 When it would have been replication factor 2, the partitions would all be double (never same partition replicated on same broker).

Topic replication factor



Producer

1 Producer: write/commit messages to topic/partition

2 Producer knows to which broker / partition to write this message and load balances automatically round-robin

Messages

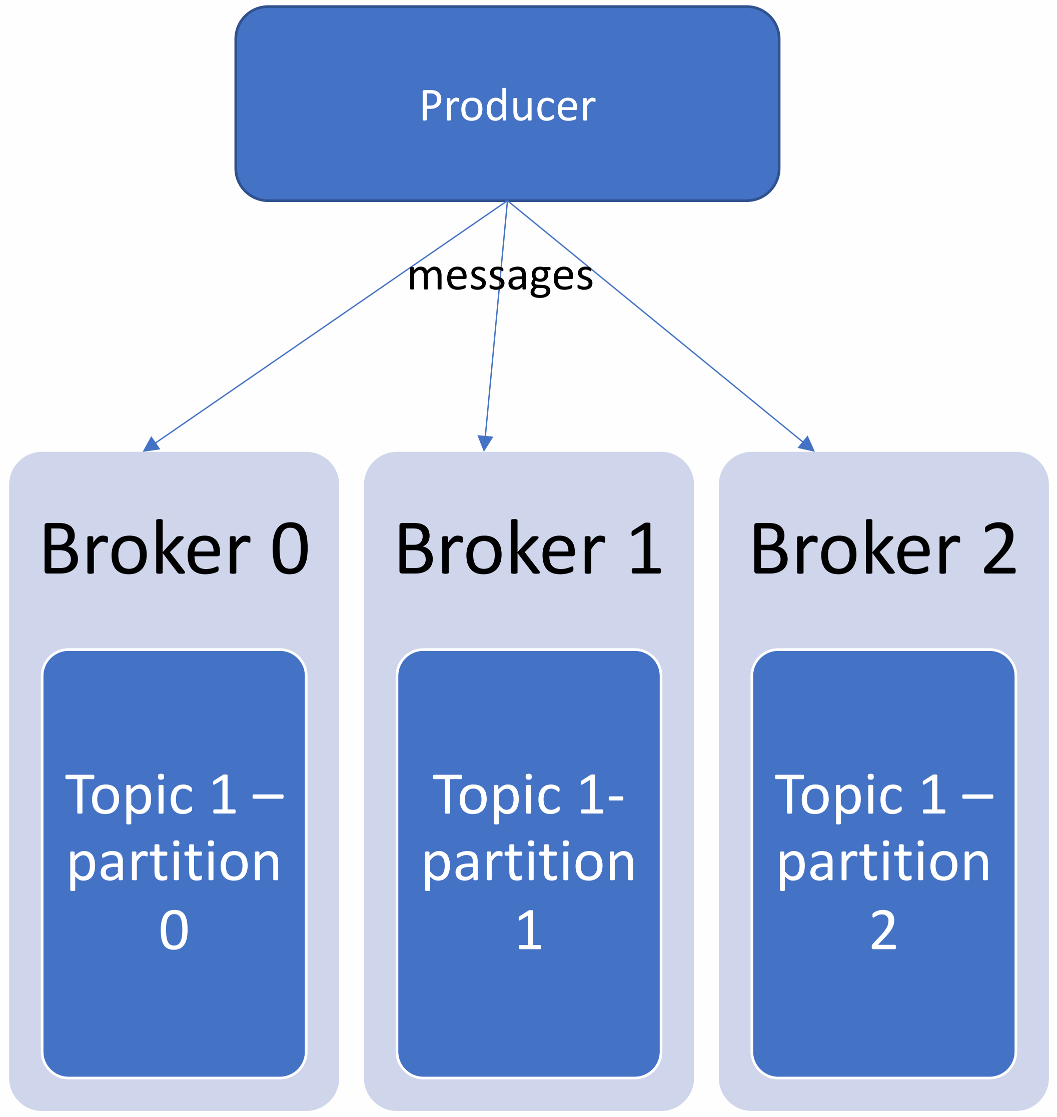
Three levels of confirmation for sending messages:

1. Acks=0: no confirmation, risky for data loss
2. Acks=1: wait for leader to confirm receival, less risky (only when server goes down before syncing)
3. Acks=all: all leaders and all replications need to confirm data receival, least risky

Partition that sync are called: in-sync-replicas (ISR)

Messages can have keys, messages with the same key will always go to the same partition. Useful when ordering for a certain key is desired.

Message without key will be distributed over partitions round-robin


Producer
messages
Broker 0
Topic 1 – partition 0
Broker 1
Topic 1- partition 1
Broker 2
Topic 1 – partition 2

Consumer

1 Consumers: read data from topic

2 Consumer knows which broker to read. The consumer reads the data in the order of the offsets (so within each partition it is reading it in order)

3 Consumer could be a Java application

Consumer group

Consumers are grouped in consumer groups. A consumer in a group reads from a partition exclusively within that group. So when there are more consumers than partitions, some consumers will be inactive. So usually you have as many consumers as partitions at most.

Consumer offsets

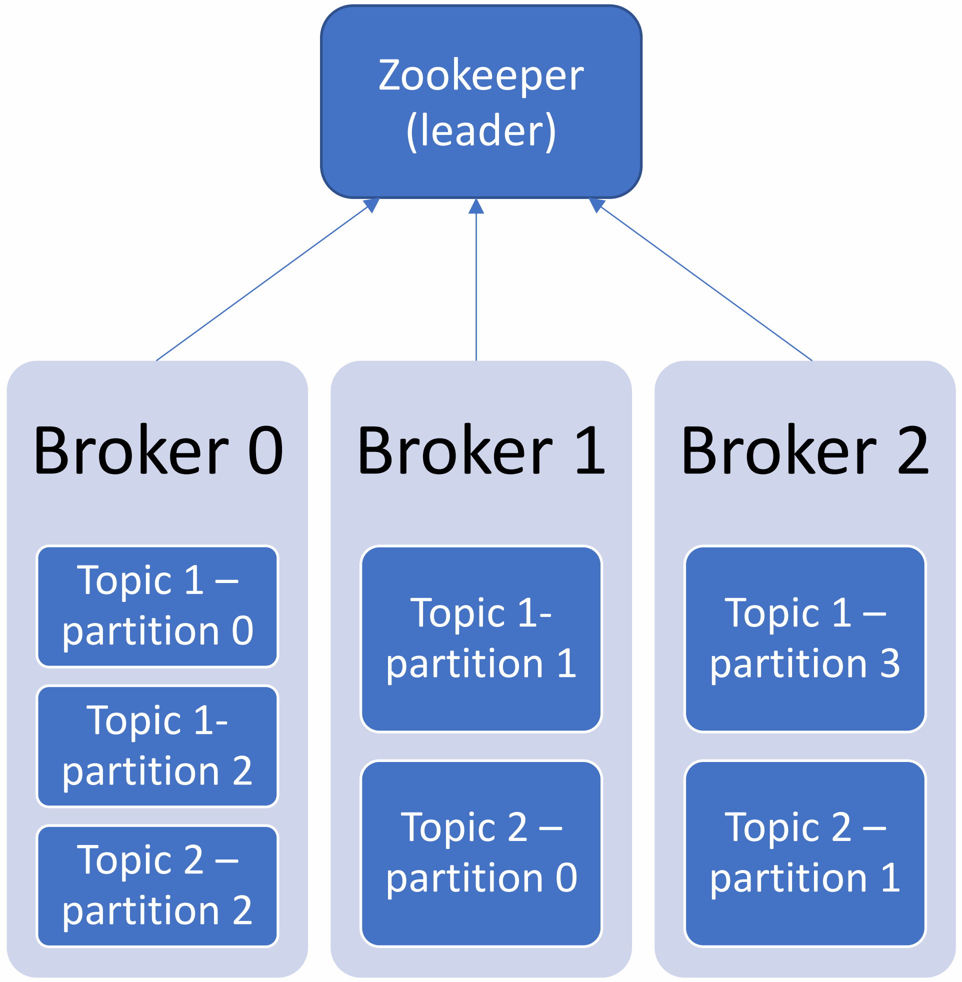
- Kafka remebers the offset the consumer group has been reading
- The consumer writes these offsets to a special kafka topic that stores the offsets. Consumer commits this offset at a set time. There three delivery sematics:
  - At most once: commited upon receival
  - At least once: commited after reading messages
  - Exactly once: only from kafka to kafka communication
- This Kafka topic is called "__consumer_offsets"
- This is useful, because if a consumer goes down, another consumer can continue the work, because it can see where the previous consumer left off

Zookeeper

1 Manages brokers, elect leaders for partitions, zookeeper takes care of new topic, removed topic, broken brokers, new brokers and everything related to brokers

2 Zookeeper is part of a cluster of zookeepers. One zookeeper is the leader, other are followers. Leader can write, followers can read. It is always an odd number of zookeepers.

3 Kafka heavily relies on zookeeper



Exercise: Design the Kafka Architecture

vijfhart IT-OPLEIDINGEN

Dat klopt voor jou!

Demo: Using Kafka with the CLI

Demo: Using Kafka in Spring Boot

Quiz

→ Kahoot

Wrap up

- Kafka theory
- Design a Kafka architecture
- Demos

vijfhart IT-OPLEIDINGEN

Dat klopt voor jou!

Vijfhart

Rokus Janssen, adviseur en accountmanager bij Vijfhart voor KPN

Heb je vragen of opmerkingen, neem dan vooral contact op met mij voor passend advies. Je kan mij bereiken via:

E: r.janssen@5hart.nl

T: 088 542 78 88

Graag tot snel!