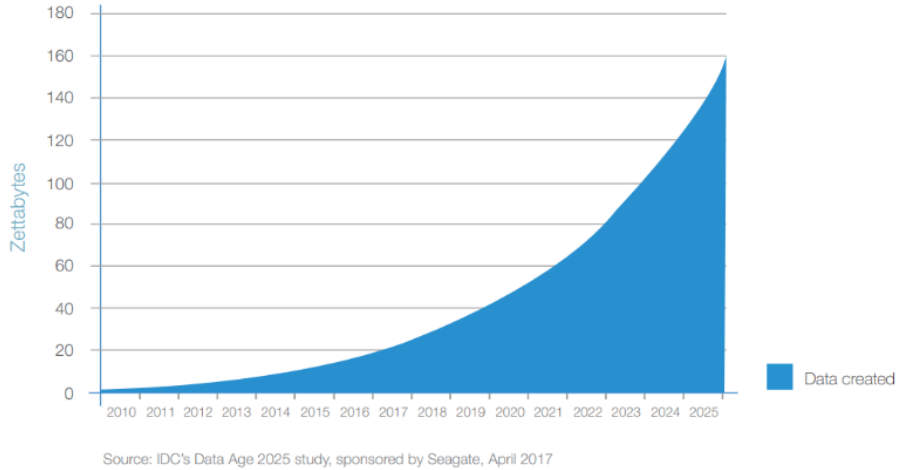
Streaming Architectures

am Beispiel von Apache Kafka

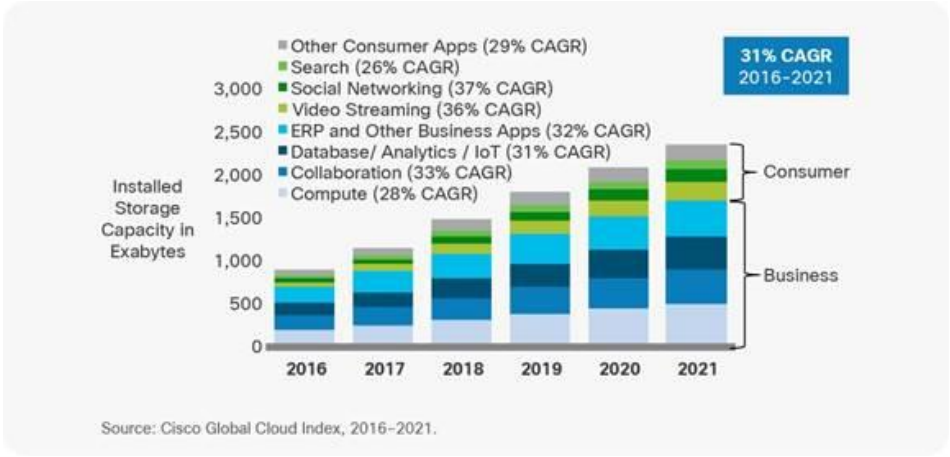
Mario Cichonczyk

02.07.2018

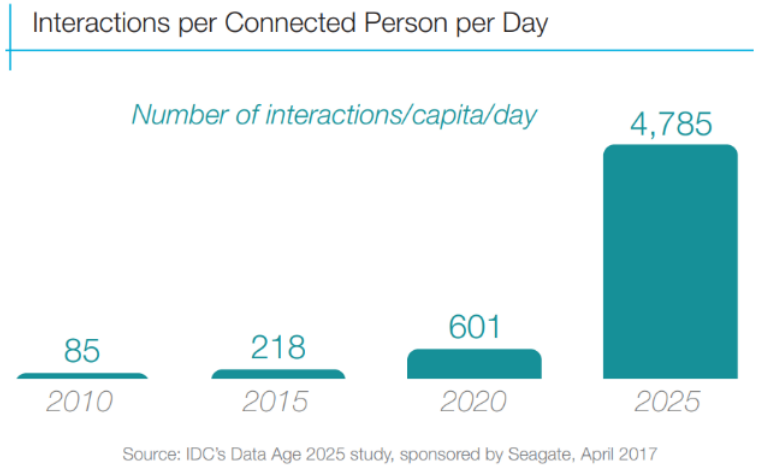
Motivation



Quelle: Reinsel, D., J. Gantz, and J. Rydning. "Data Age 2025: The Evolution of Data to Life-Critical. Don't Focus on Big Data; Focus on the Data That's Big." IDC, Seagate, April (2017).

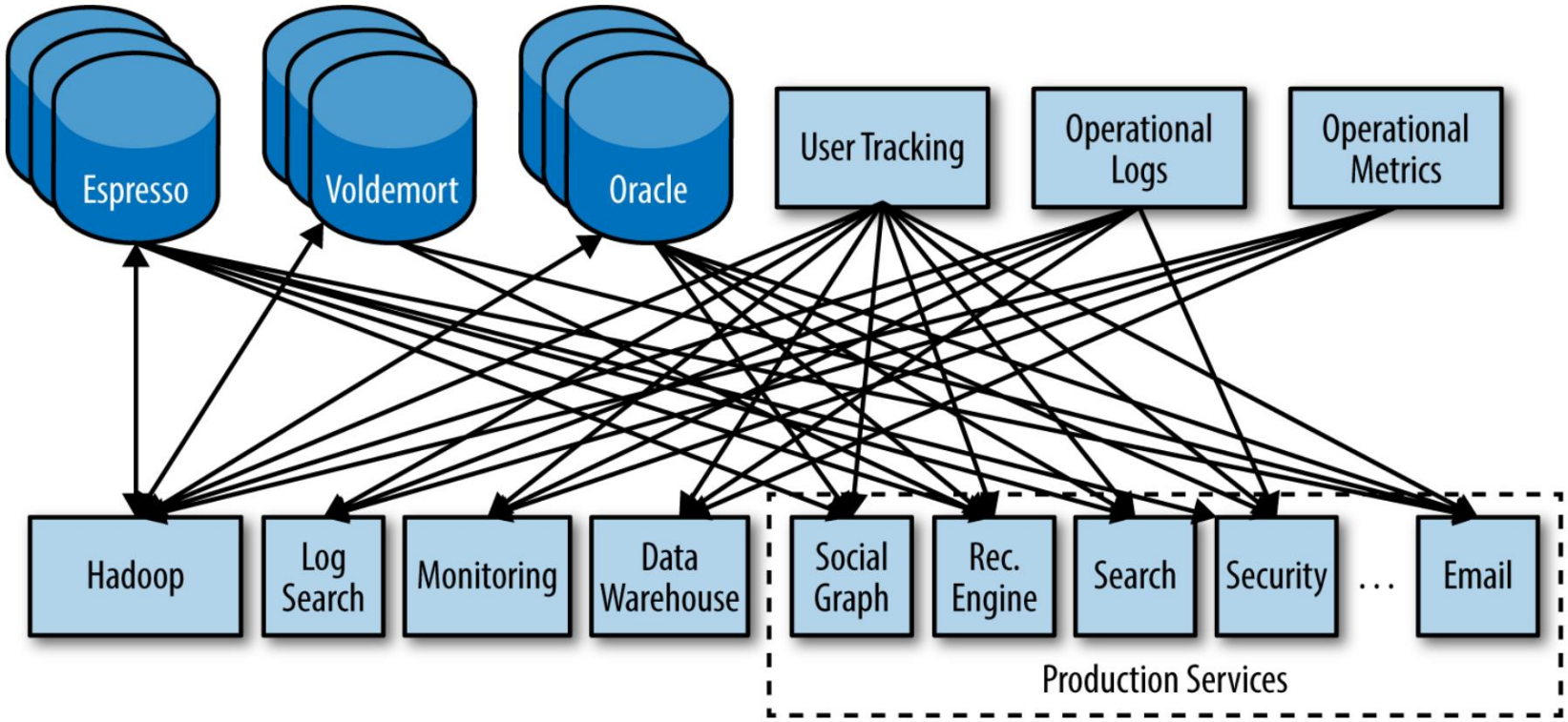


Quelle: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/white-paper-c11-738085.html#_Toc503317524



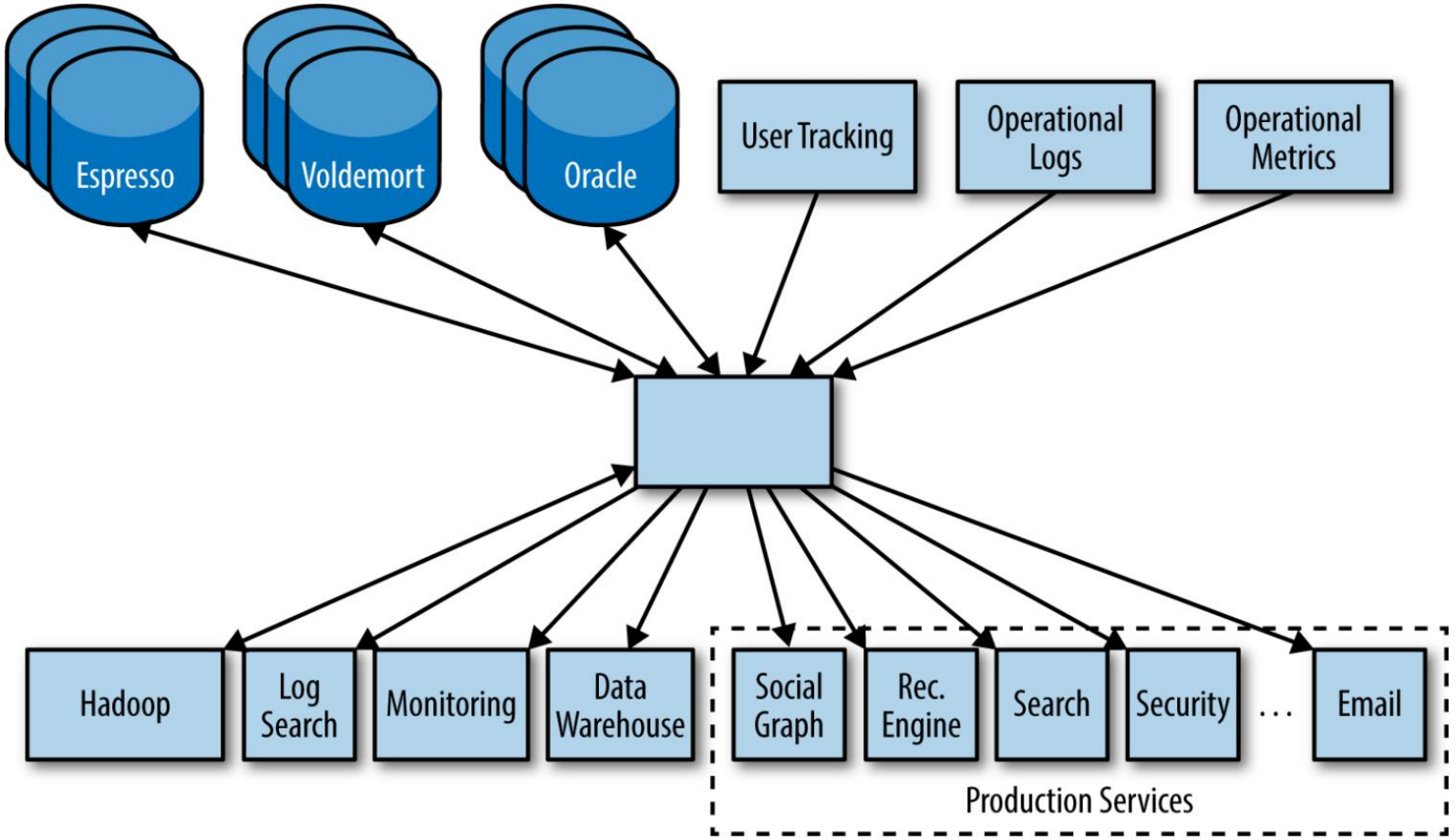
Quelle: Reinsel, D., J. Gantz, and J. Rydning. "Data Age 2025: The Evolution of Data to Life-Critical. Don't Focus on Big Data; Focus on the Data That's Big." IDC, Seagate, April (2017).

Motivation



Quelle: Jay Kreps. 2014. I Heart Logs. Event Data, Stream Processing, and Data Integration. O'Reilly Media, Inc..

Motivation



Quelle: Jay Kreps. 2014. I Heart Logs. Event Data, Stream Processing, and Data Integration. O'Reilly Media, Inc..

Motivation

Quelle: https://upload.wikimedia.org/wikipedia/commons/thumb/6/67/Dove-Lake_from_South-2008.jpg/300px-Dove-Lake_from_South-2008.jpg

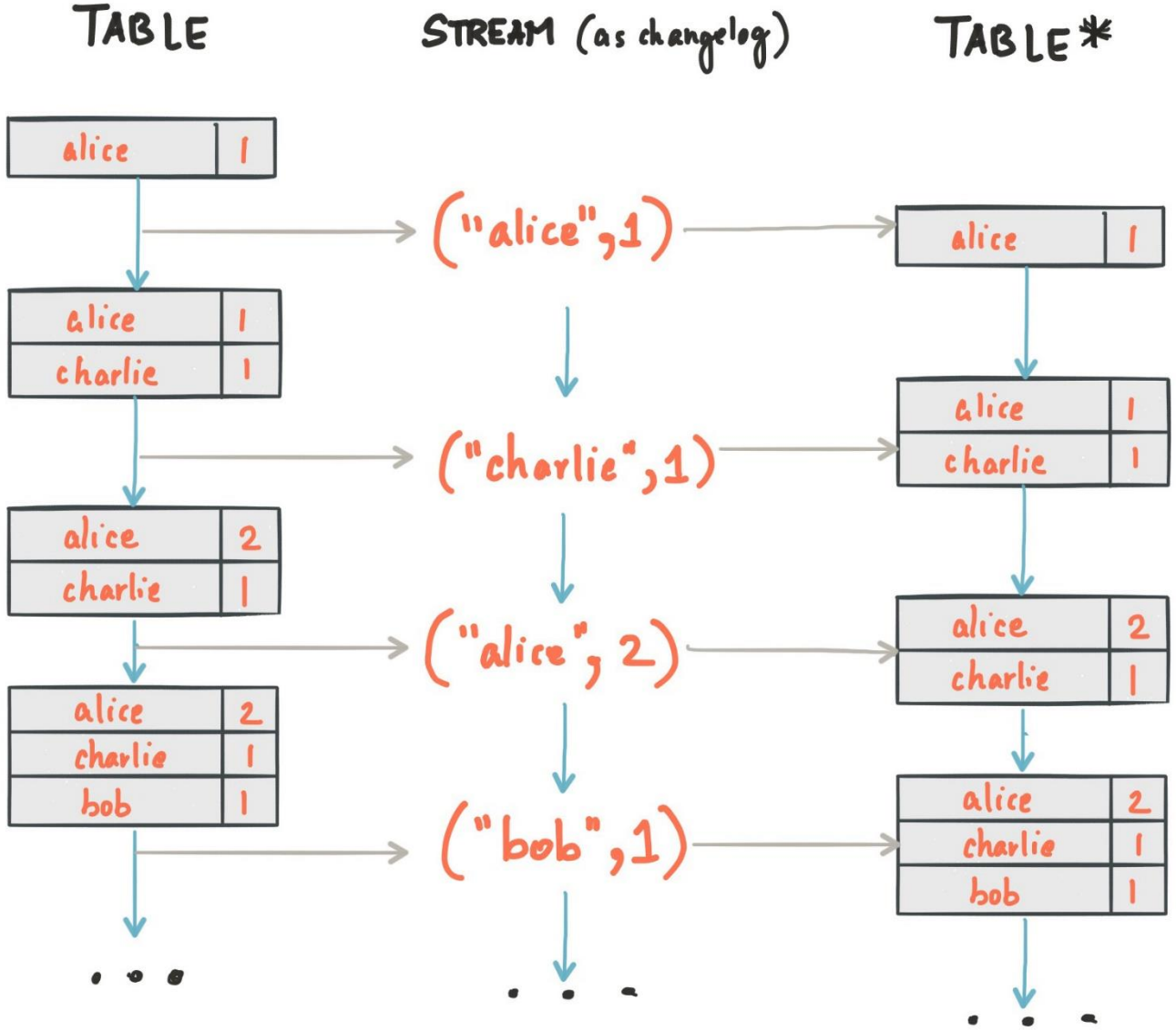
Motivation

Quelle: http://faunatec.com/wp-content/uploads/2015/02/night-diver_nachttauchen.jpg

Architektur

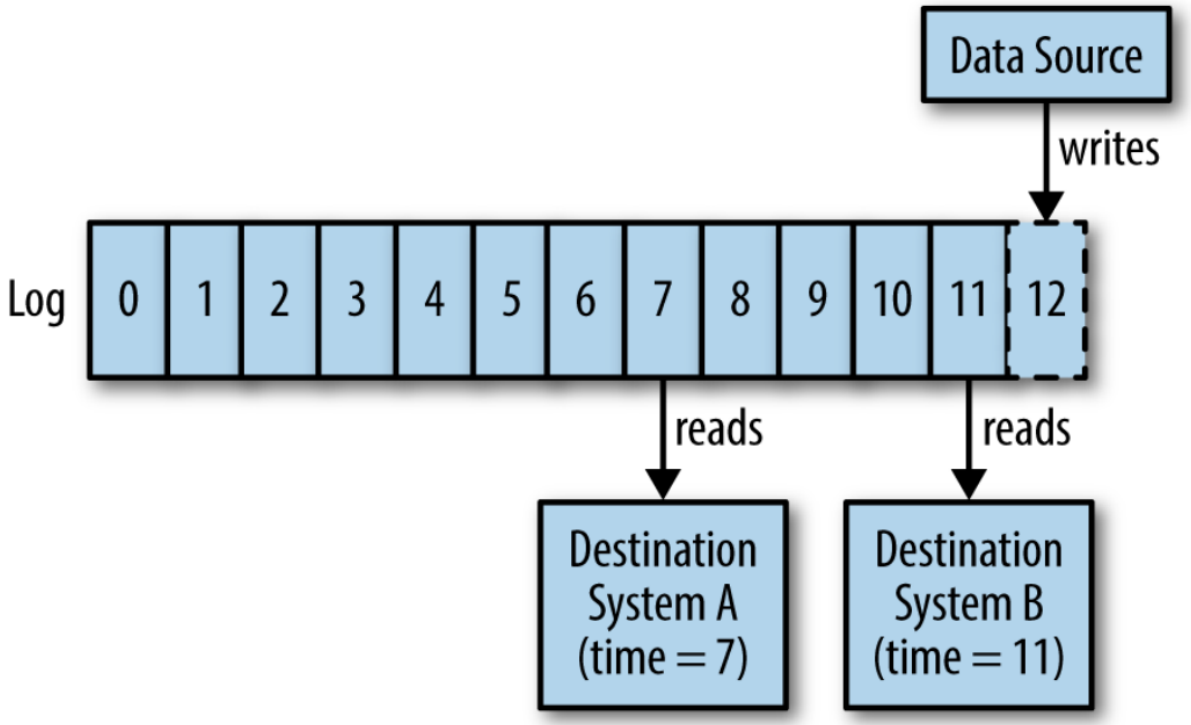
Quelle: https://www.blm.gov/or/districts/roseburg/recreation/wild_and_scenic_river/photo_gallery.html

Architektur



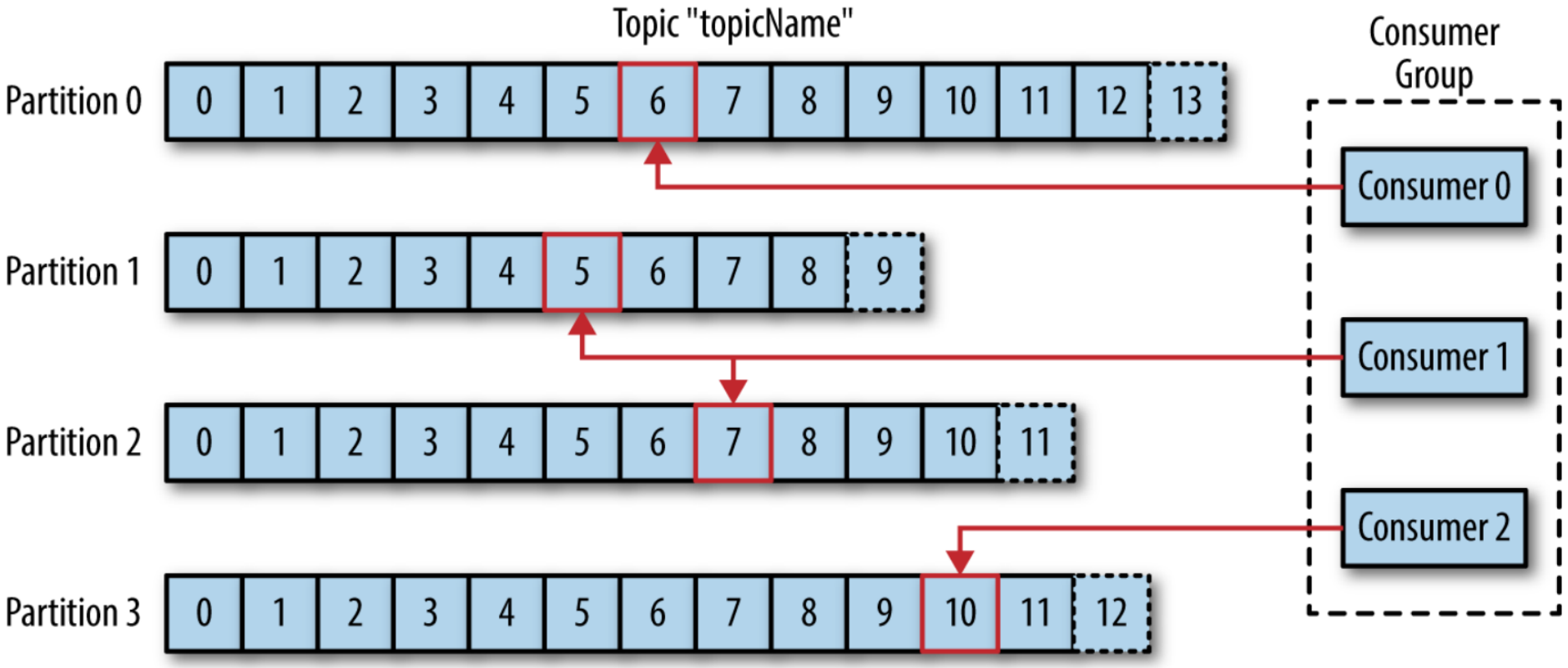
Quelle: https://docs.confluent.io/current/streams/concepts.html#streams-concepts-kstream

Datenstruktur



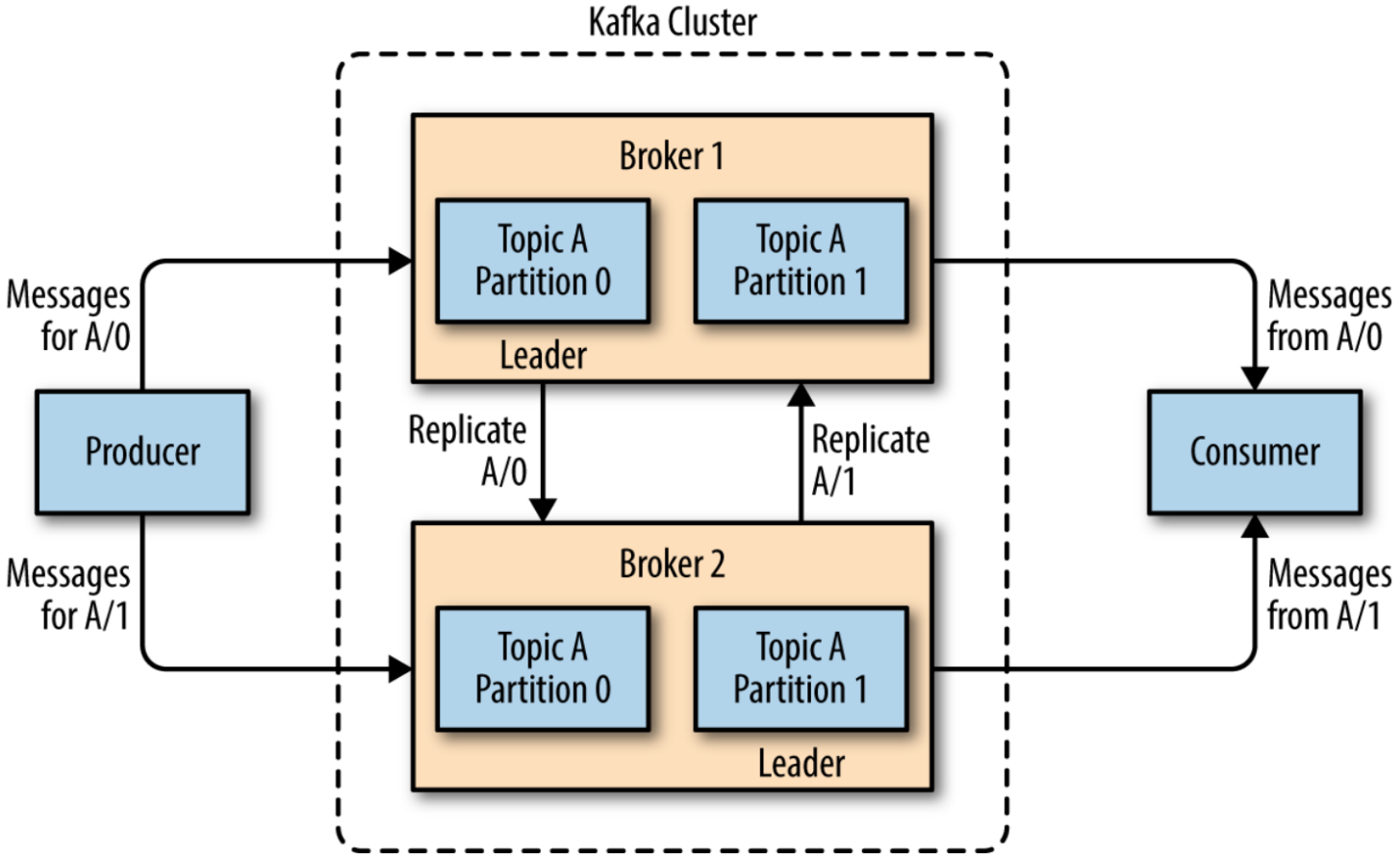
Quelle: Jay Kreps. 2014. I Heart Logs. Event Data, Stream Processing, and Data Integration. O'Reilly Media, Inc..

Kafka



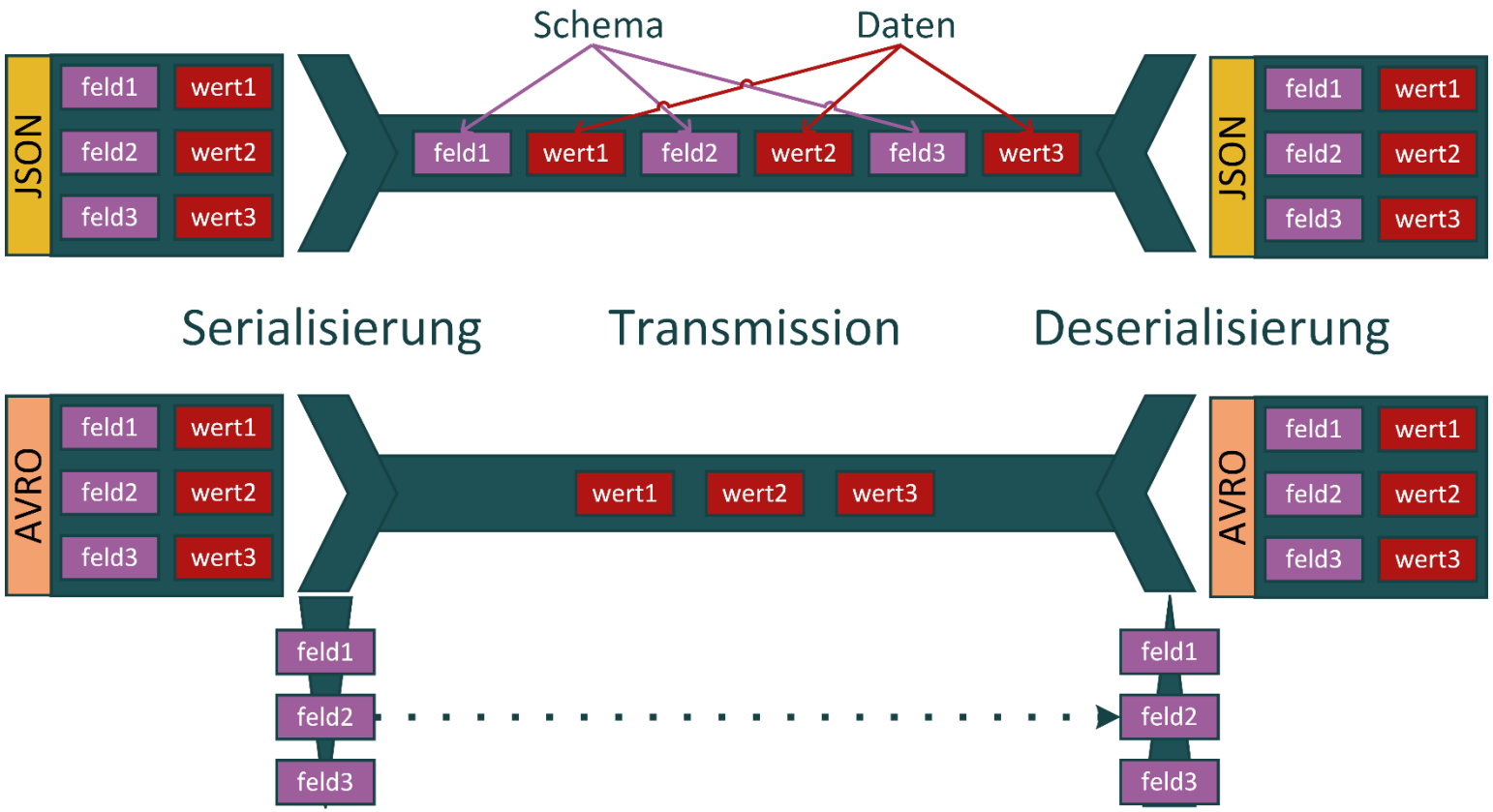
Quelle: Gwen Shapira, Neha Narkhede & Todd Palino. 2017. Kafka: The Definitive Guide. Real-Time Data and Stream Processing at Scale. O'Reilly Media, Inc..

Kafka

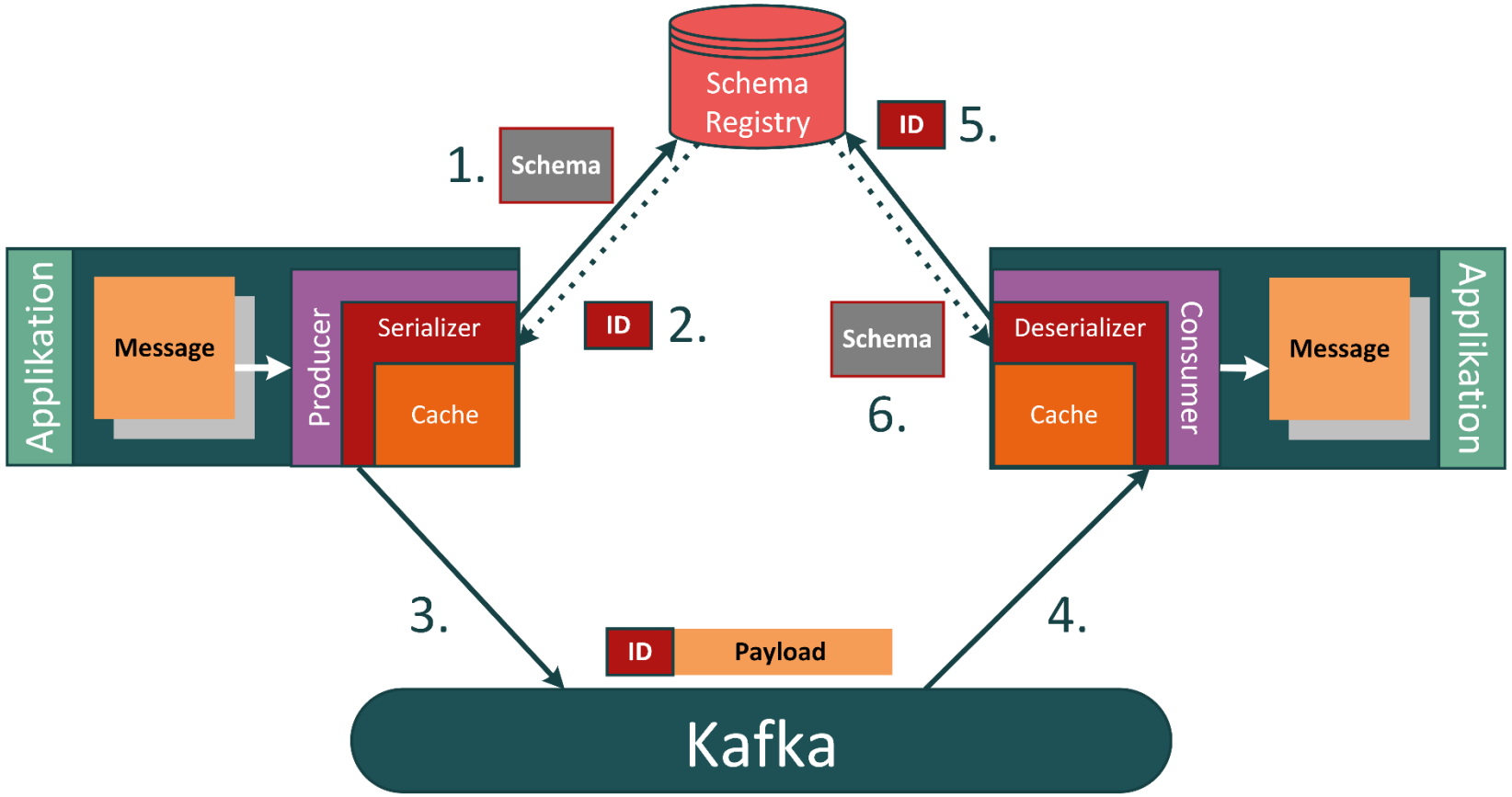


Quelle: Gwen Shapira, Neha Narkhede & Todd Palino. 2017. Kafka: The Definitive Guide. Real-Time Data and Stream Processing at Scale. O'Reilly Media, Inc..

(De)Deserialisierung



(De)Deserialisierung



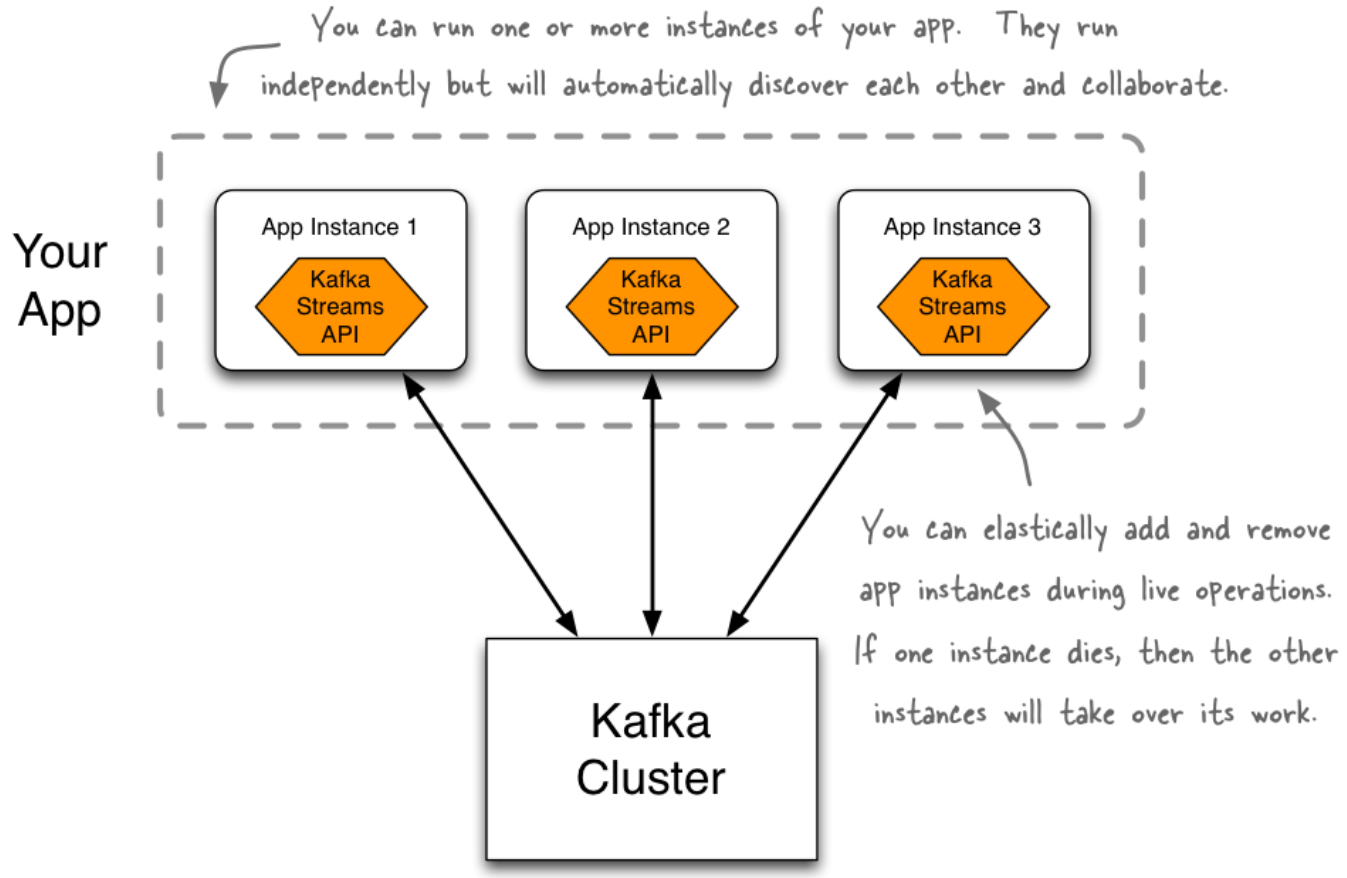
Stream Processing

Streams API

```
// Daten deserialisieren und aus Kafka lesen
KStream<byte[], String> text = builder.stream("topic", Consumed.with(Serdes.ByteArray(), Serdes.String()));
// Daten transformieren
KStream<byte[], String> textGross= text.mapValues(String::toUpperCase));
// Daten in anderes Topic zurueckschreiben
textGross.to("topicGross", Produced.with(Serdes.ByteArray(), Serdes.String()));
```

Stream Processing

Streams API



Quelle: https://docs.confluent.io/current/streams/introduction.html

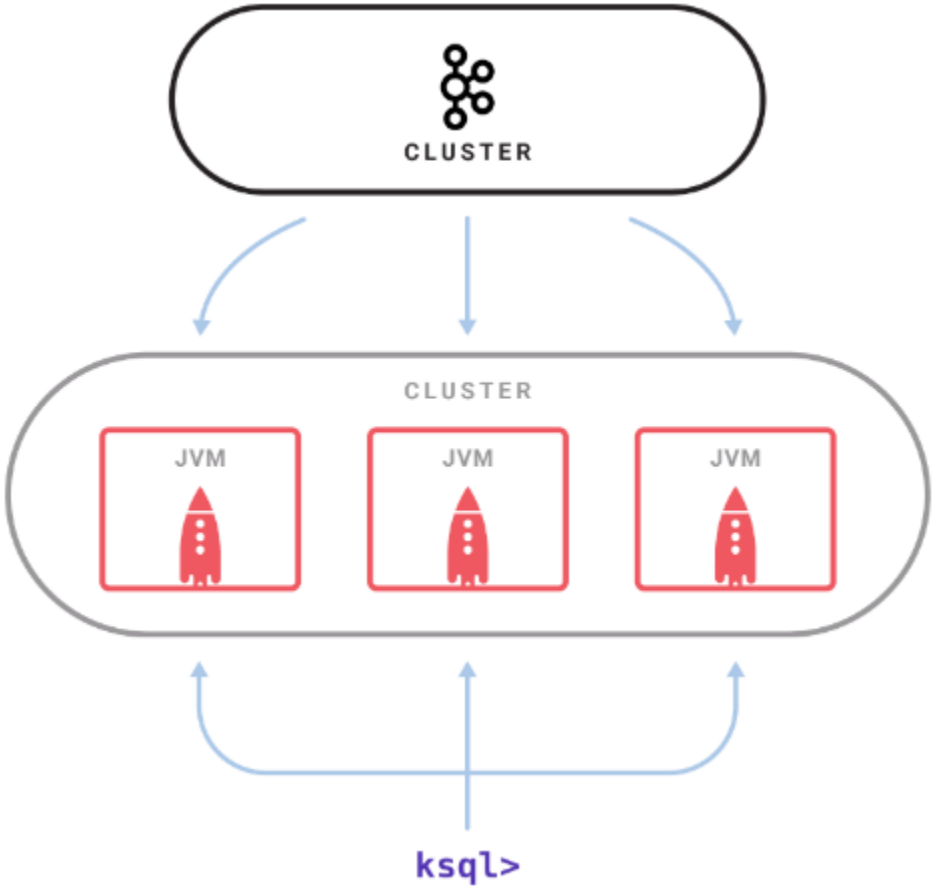
Stream Processing

KSQL

```
SELECT * FROM kafka-topic WHERE memberfield > 0.8;
```

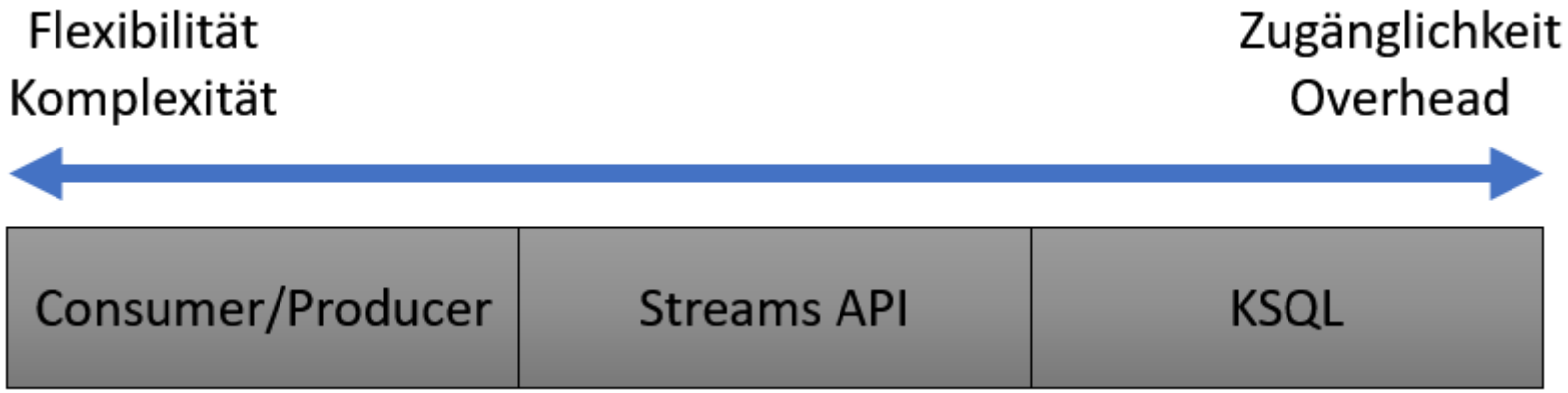
Stream Processing

KSQL

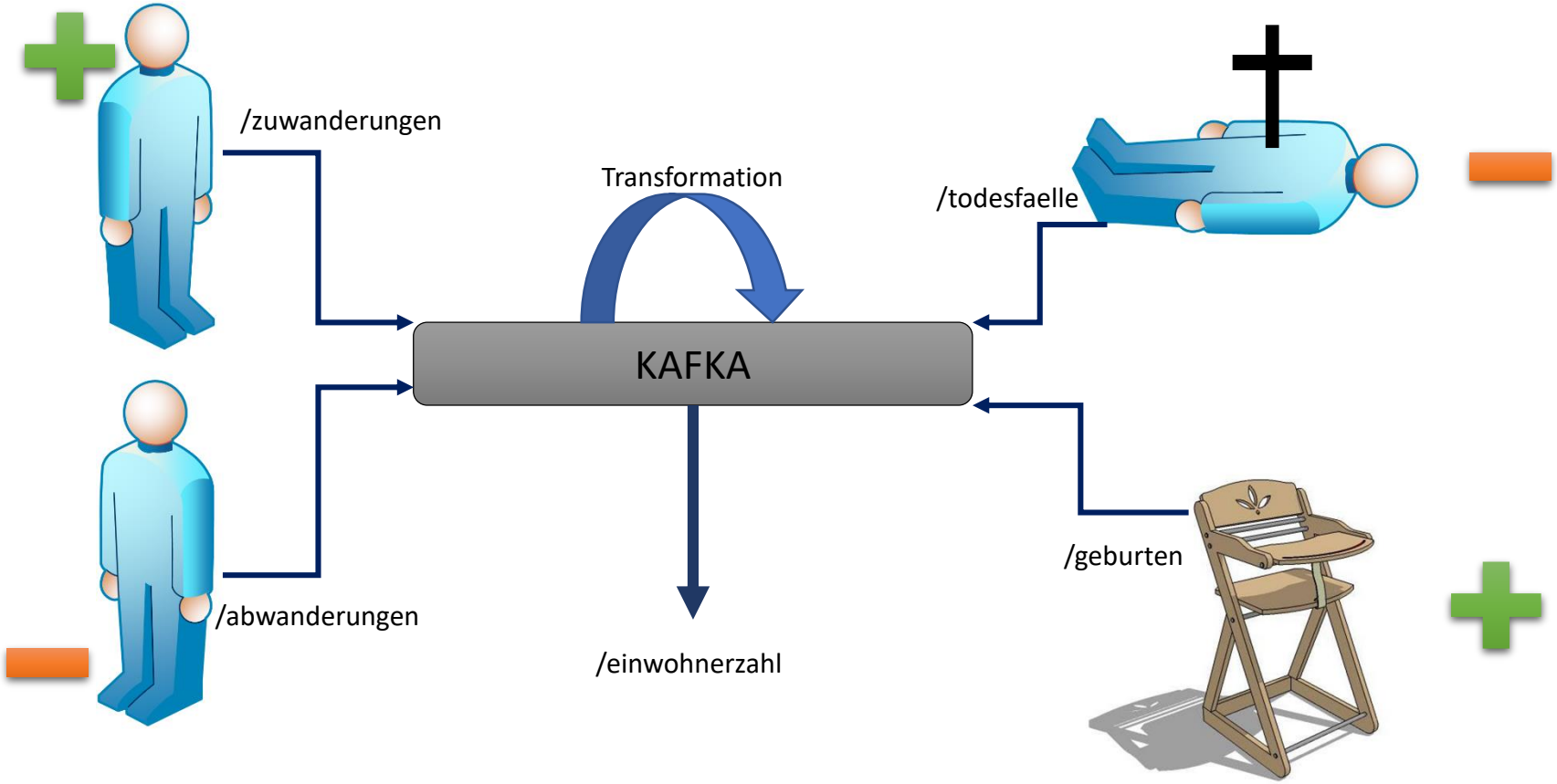


Quelle: https://www.confluent.io/wp-content/uploads/ksql_cluster-min.png

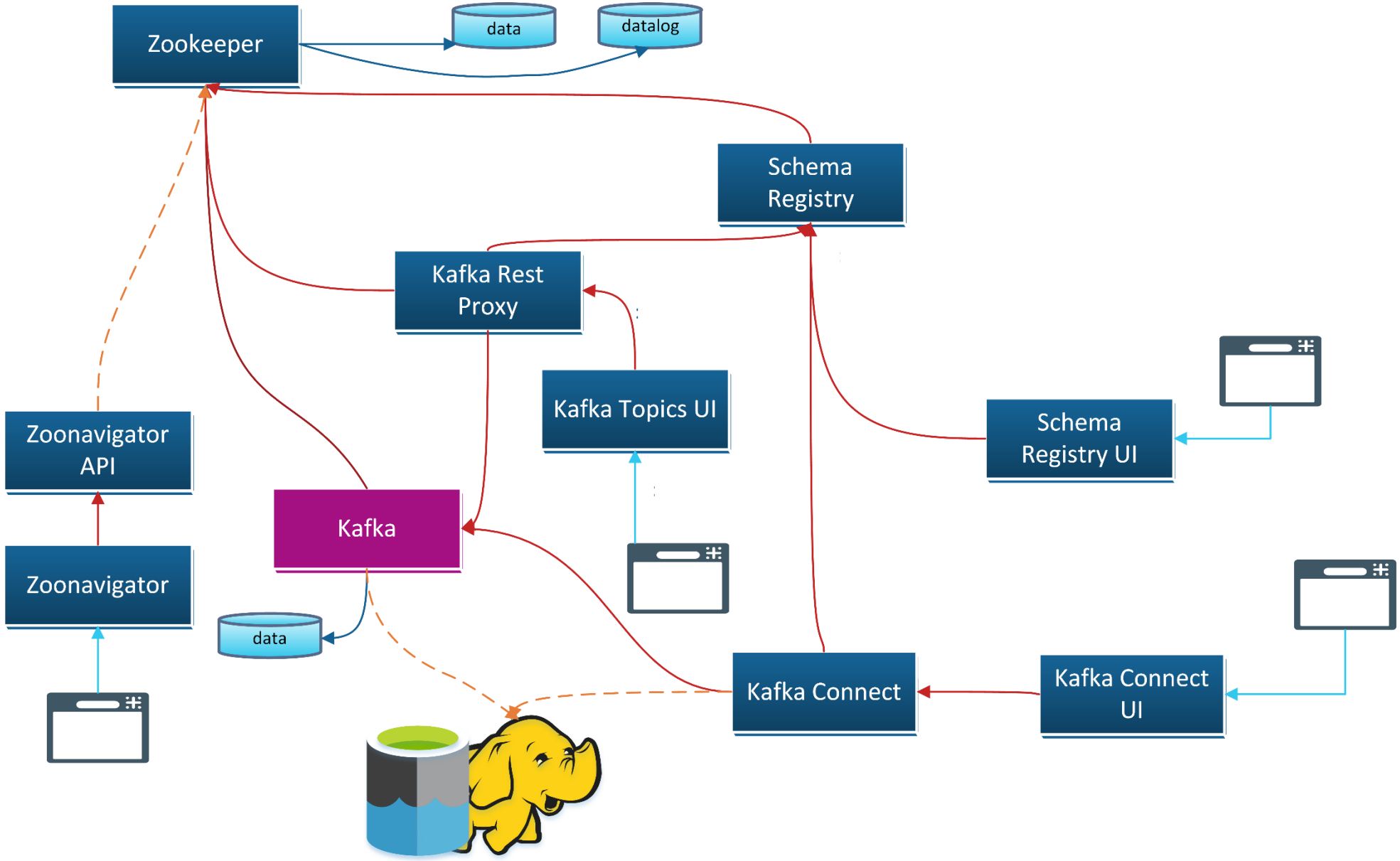
Stream Processing



Demo


/zuwanderungen
Transformation
/todesfaelle
KAFKA
/abwanderungen
/einwohnerzahl
/geburten

Docker-Plattform der Demo



Vielen Dank! Fragen?

Streaming Architectures am Beispiel von Apache Kafka

Mario Cichonczyk

02.07.2018