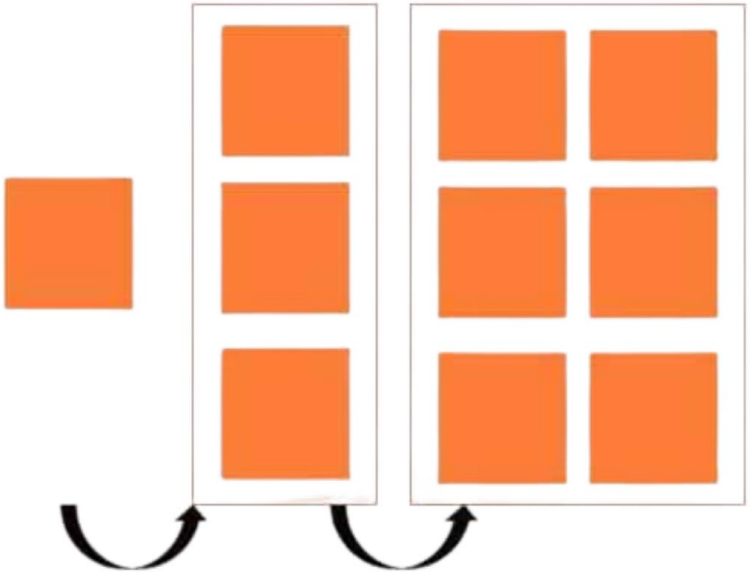
Realtime Big Data ist doch keine Kunst - wirklich?

Ursula Deriu / jugs.ch

Definition Big Data

Der Begriff ‘Big Data’ bezeichnet Methoden zur Speicherung, Abfrage, Verarbeitung und Analyse von Daten mit Hilfe von verteilten und horizontal skalierbaren Systemen.

Horizontal skalierbar

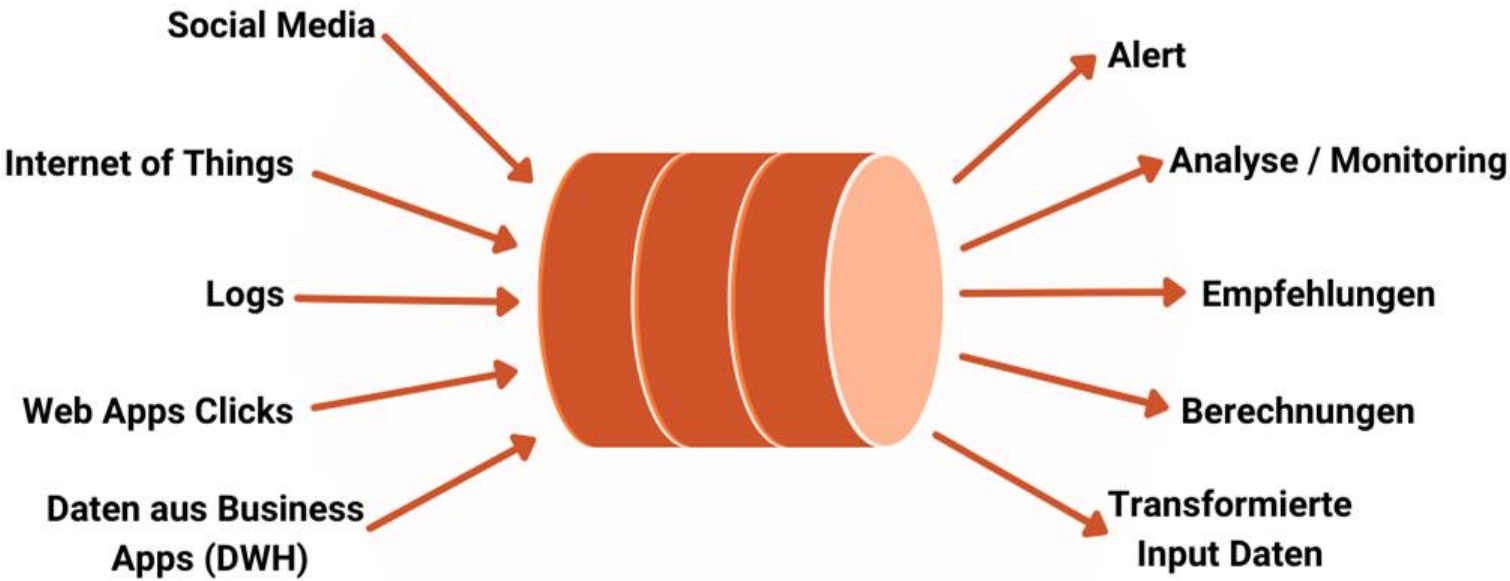


Big Data RZ

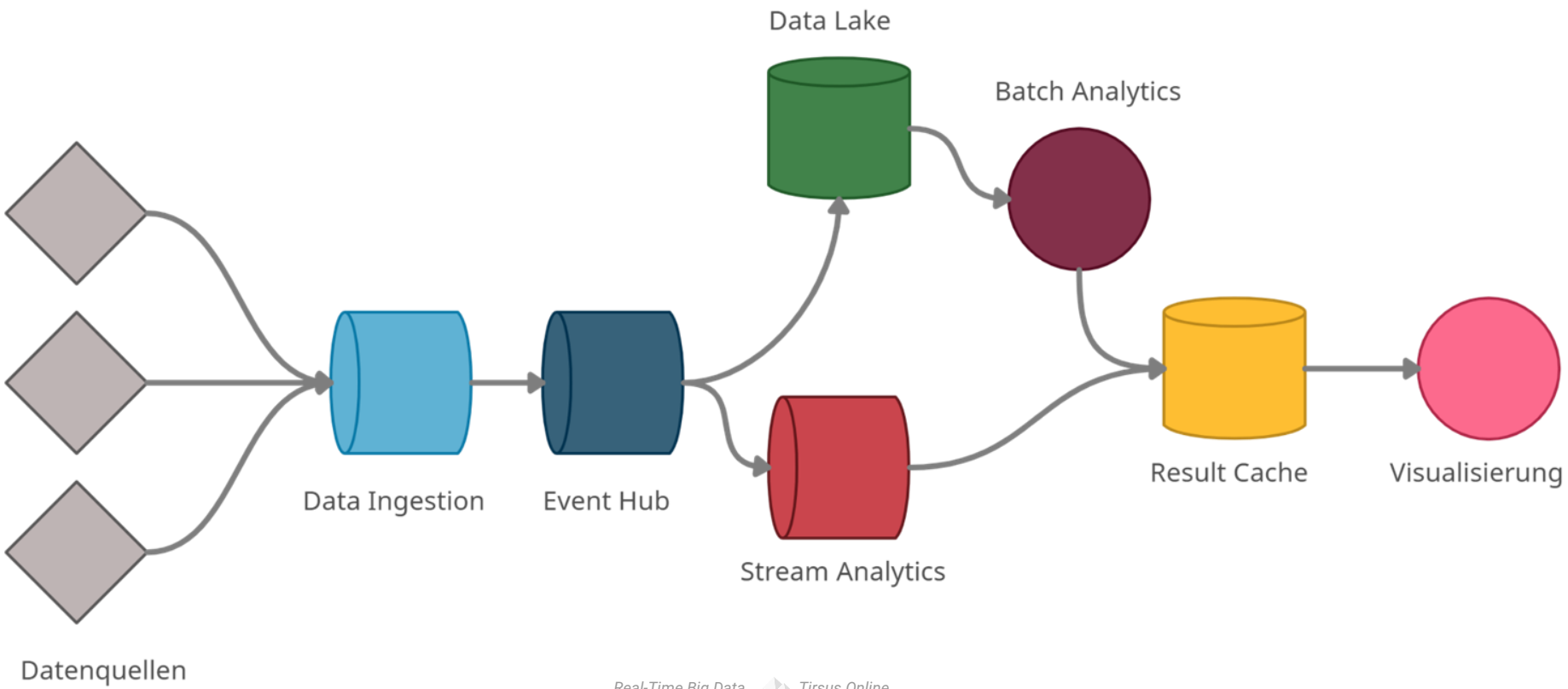
Quelle Bilder: Google

Braucht Big Data auch Big Infrastructure?

Einsatzmöglichkeiten

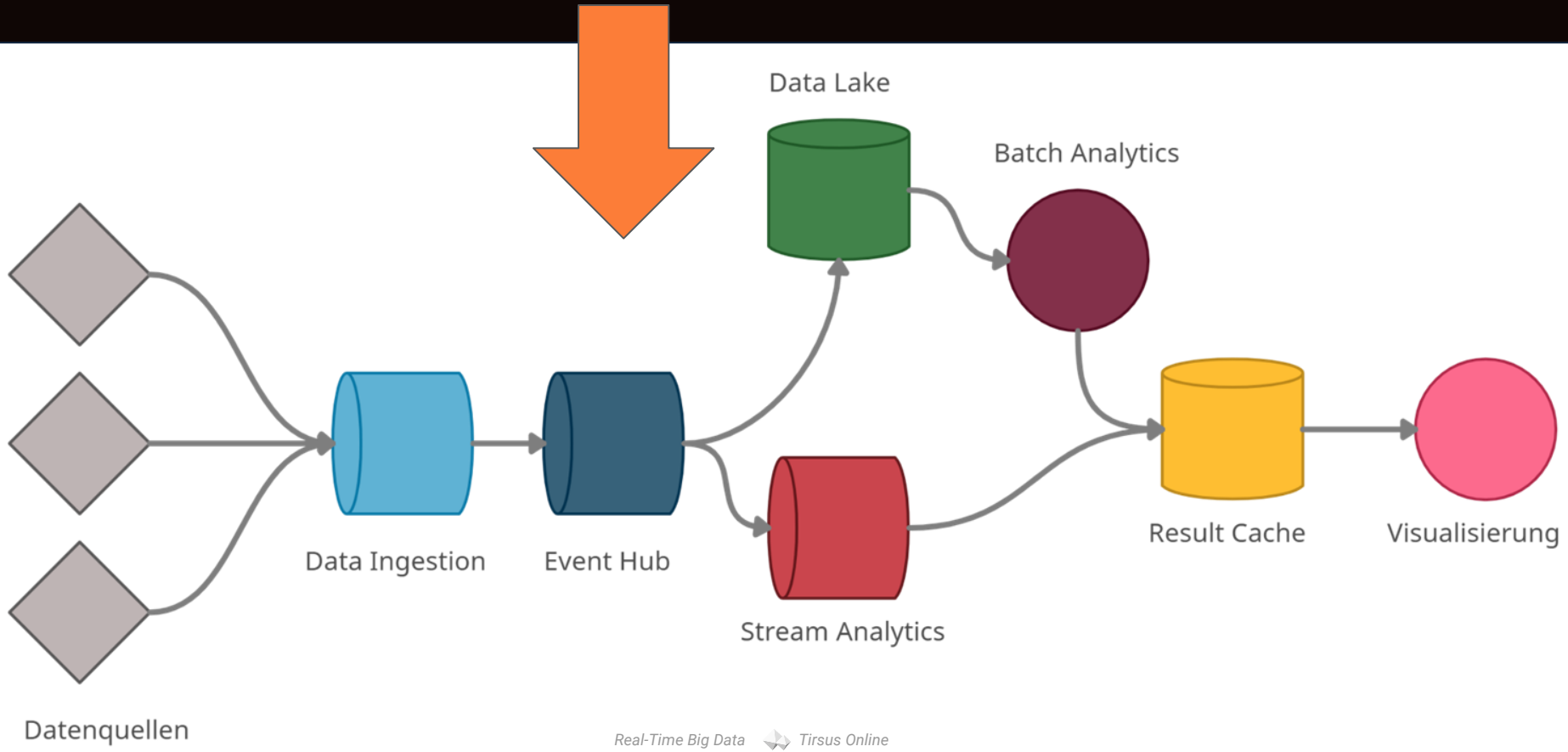


Anatomie einer Big-Data-Streaming-Pipeline

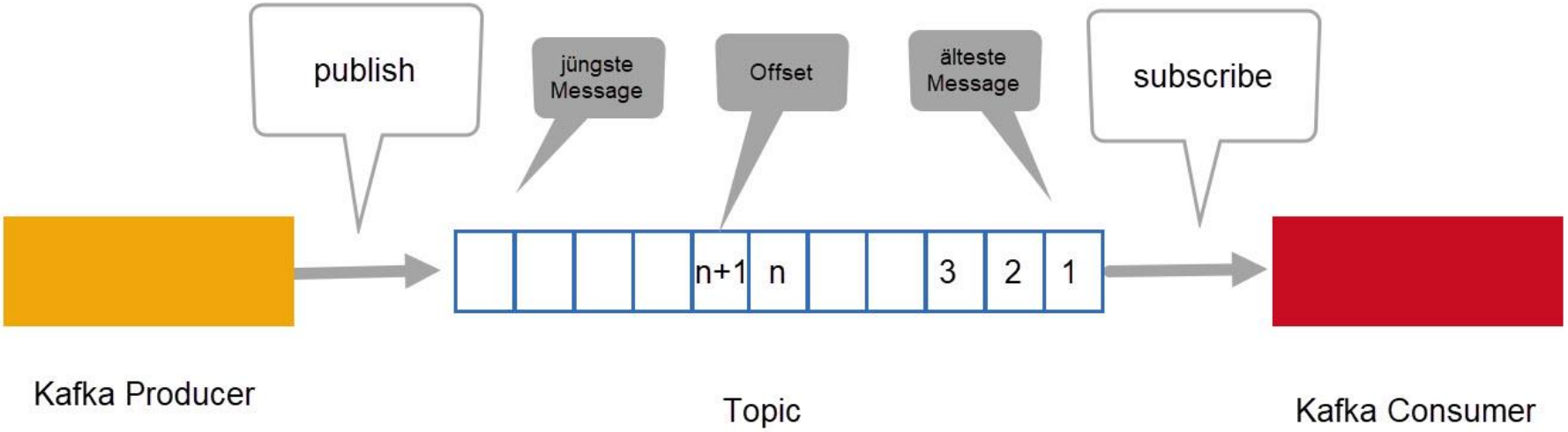


Apache Kafka als Event Hub

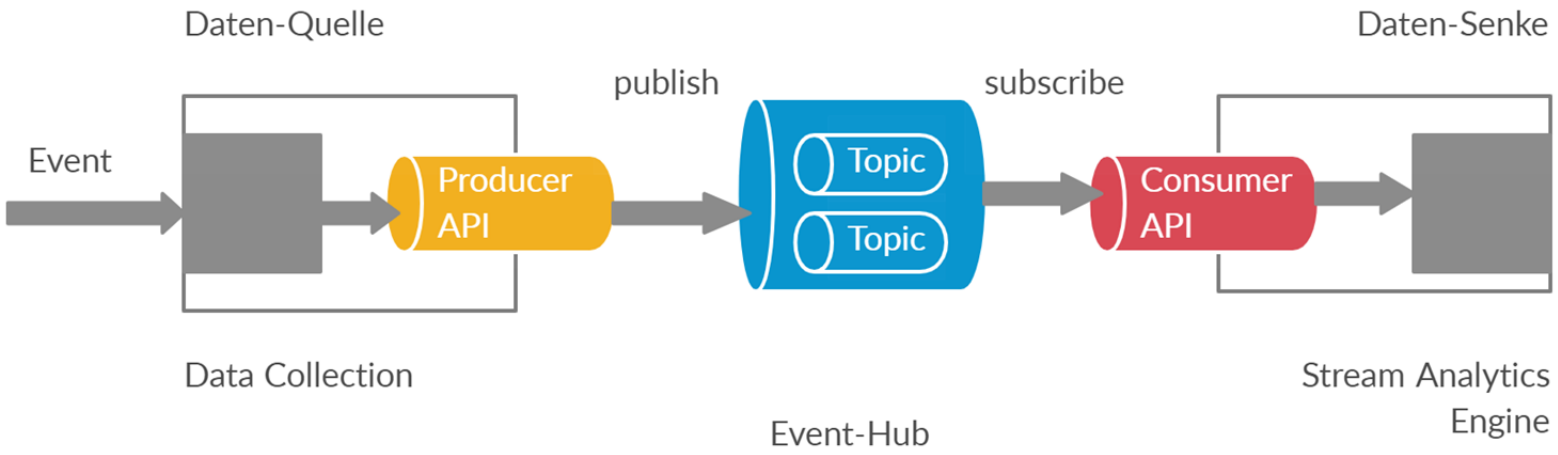
Apache Kafka als Event Hub



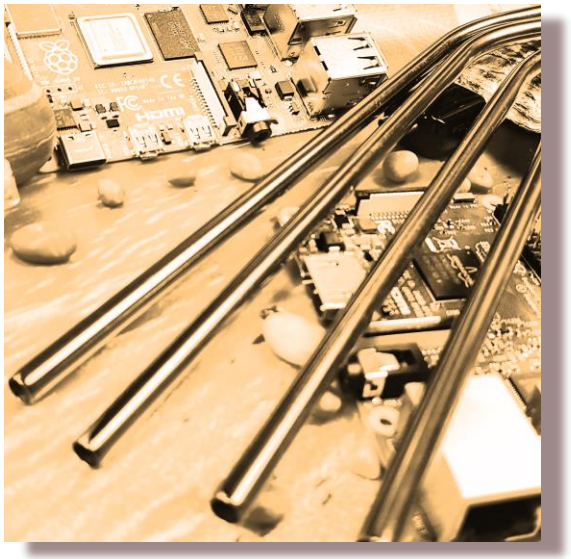
Warteschlange



Verteilter Event Hub



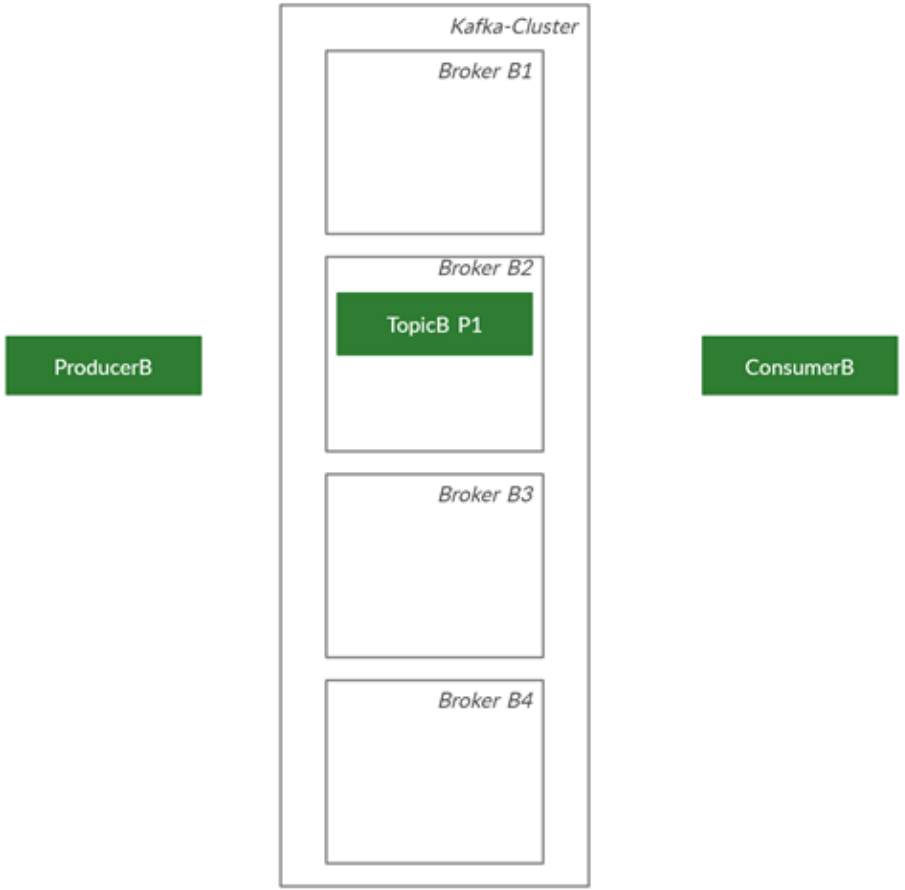
Ordering Guarantee in Apache Kafka

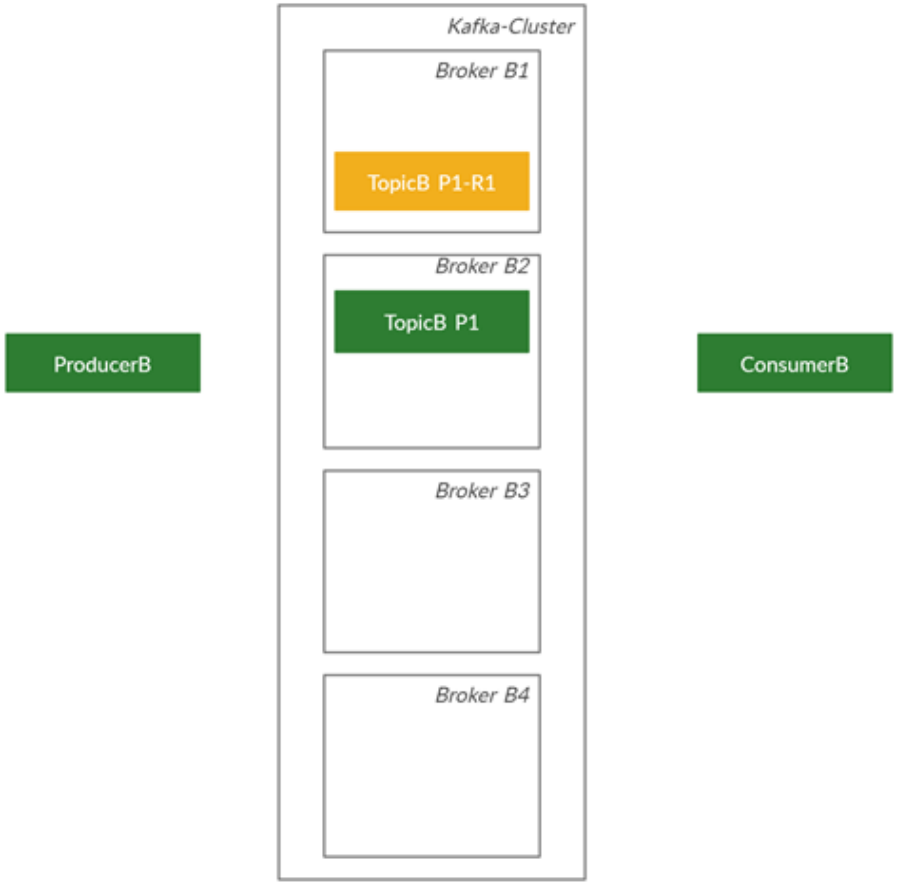


Video hier: https://tirsus.com/big-data/realtime-streaming/ordering-guarantee-in-apache-kafka/

Blick hinter die Kulissen von Apache Kafka

TopicB mir einer Partition P1

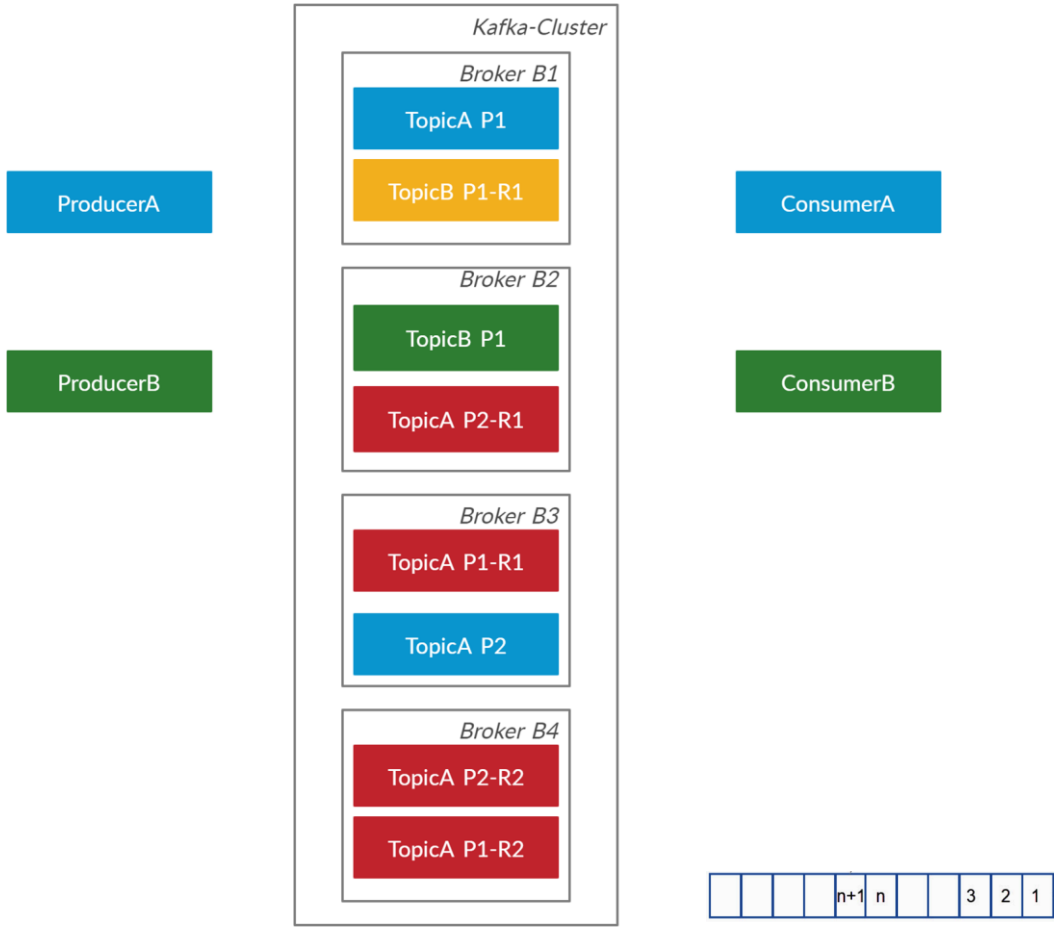




TopicB mit einer Partition und zwei Replikaten P1, P1-R1

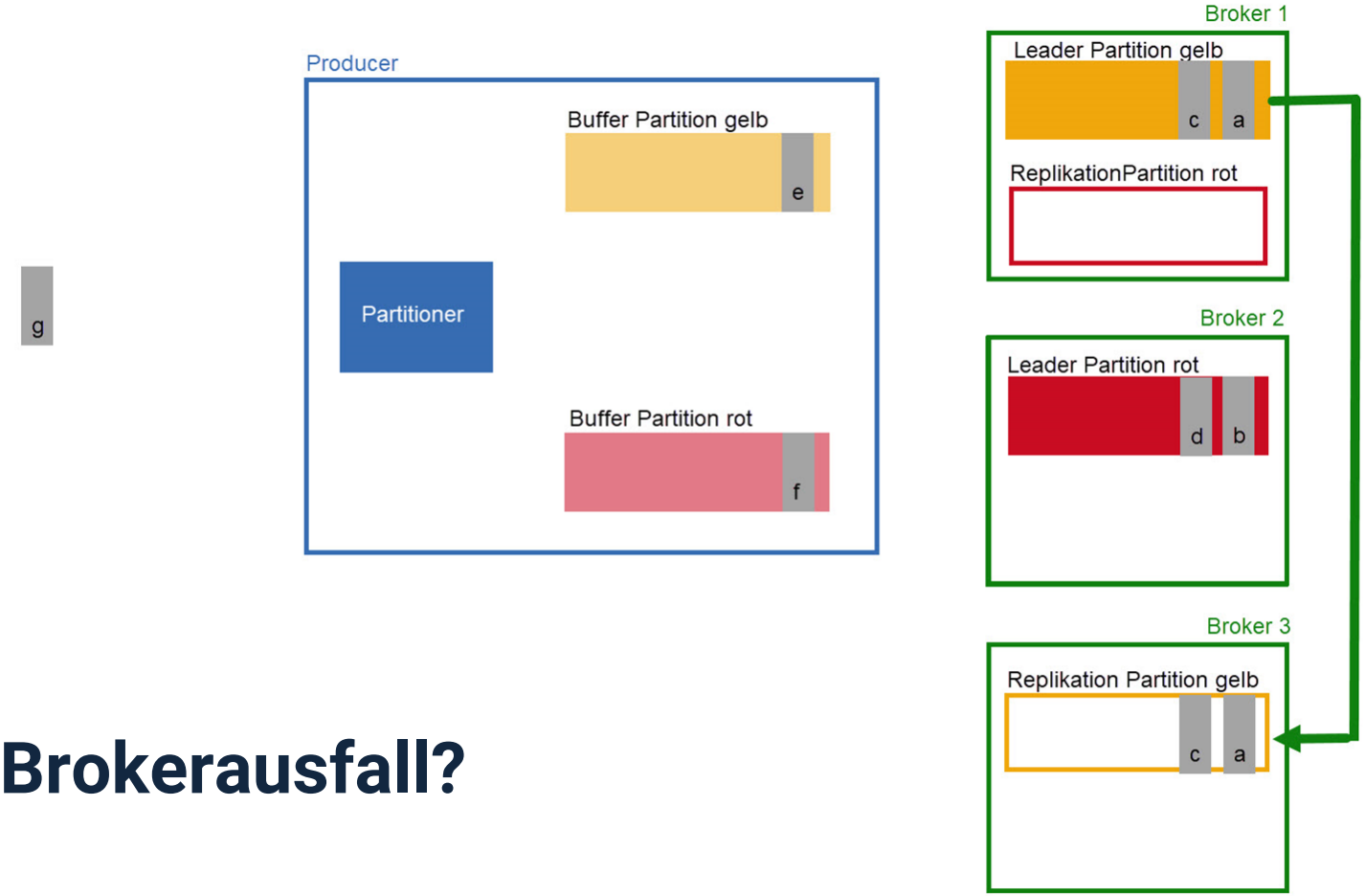
TopicA mit 2 Partitionen und Replikations-faktor 3

TopicB mit einer Partition und zwei Replikaten P1, P1-R1



Auf ein Topic schreiben

Brokerausfall?



Wegweisend für Big Data: Vor 20 Jahren bewiesen

Das CAP Theorem

Ein verteiltes Datensystem kann nur zwei der drei folgenden Garantien geben:

Consistency

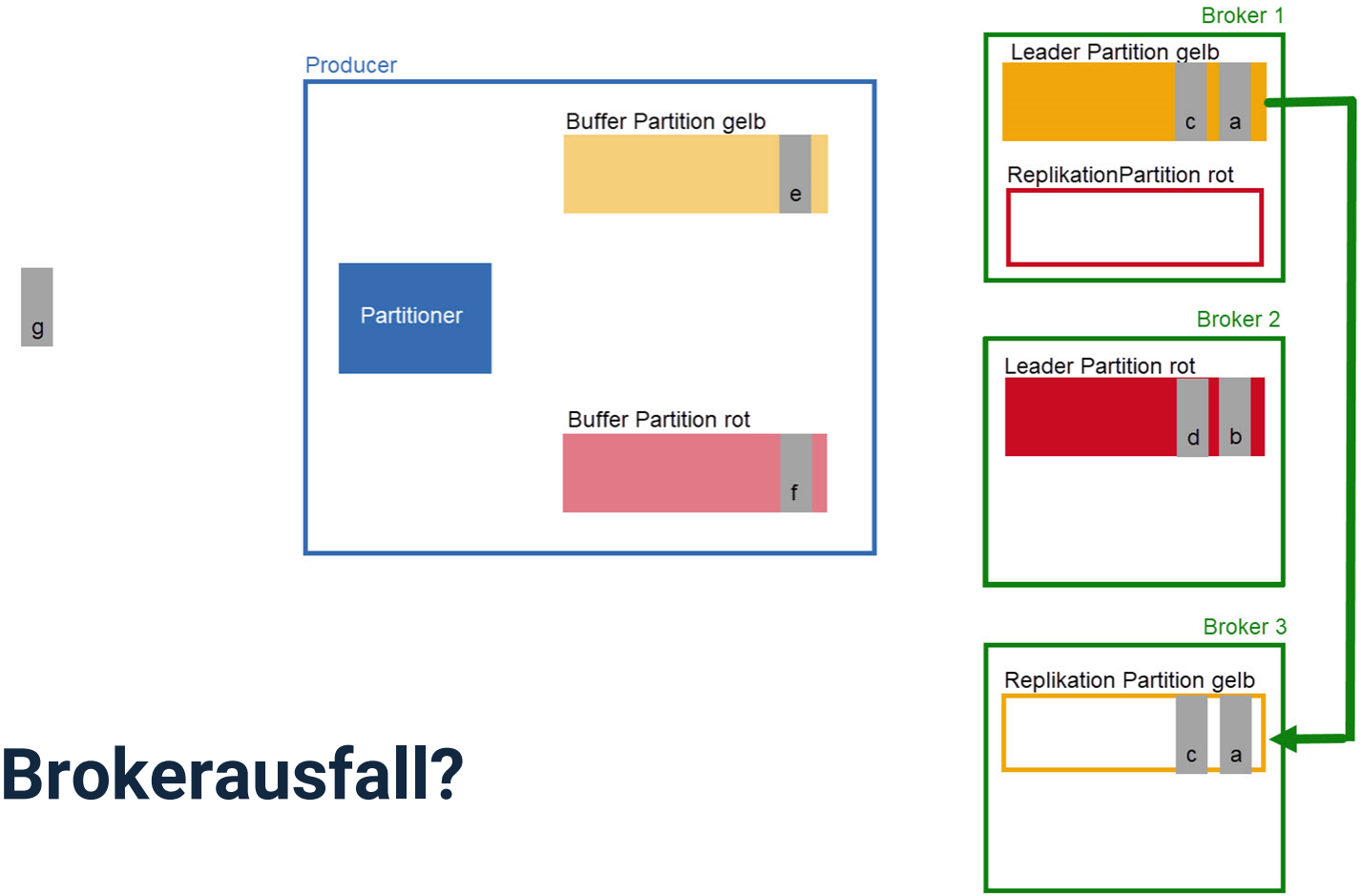
Jeder Lesebefehl erhält das Ergebnis des jüngsten Schreibbefehls oder eine Fehlermeldung.

Availability

Jede Anfrage erhält eine fehlerfreie Antwort, jedoch ohne Garantie, dass die aktuellsten Daten berücksichtigt werden.

Partition Tolerance

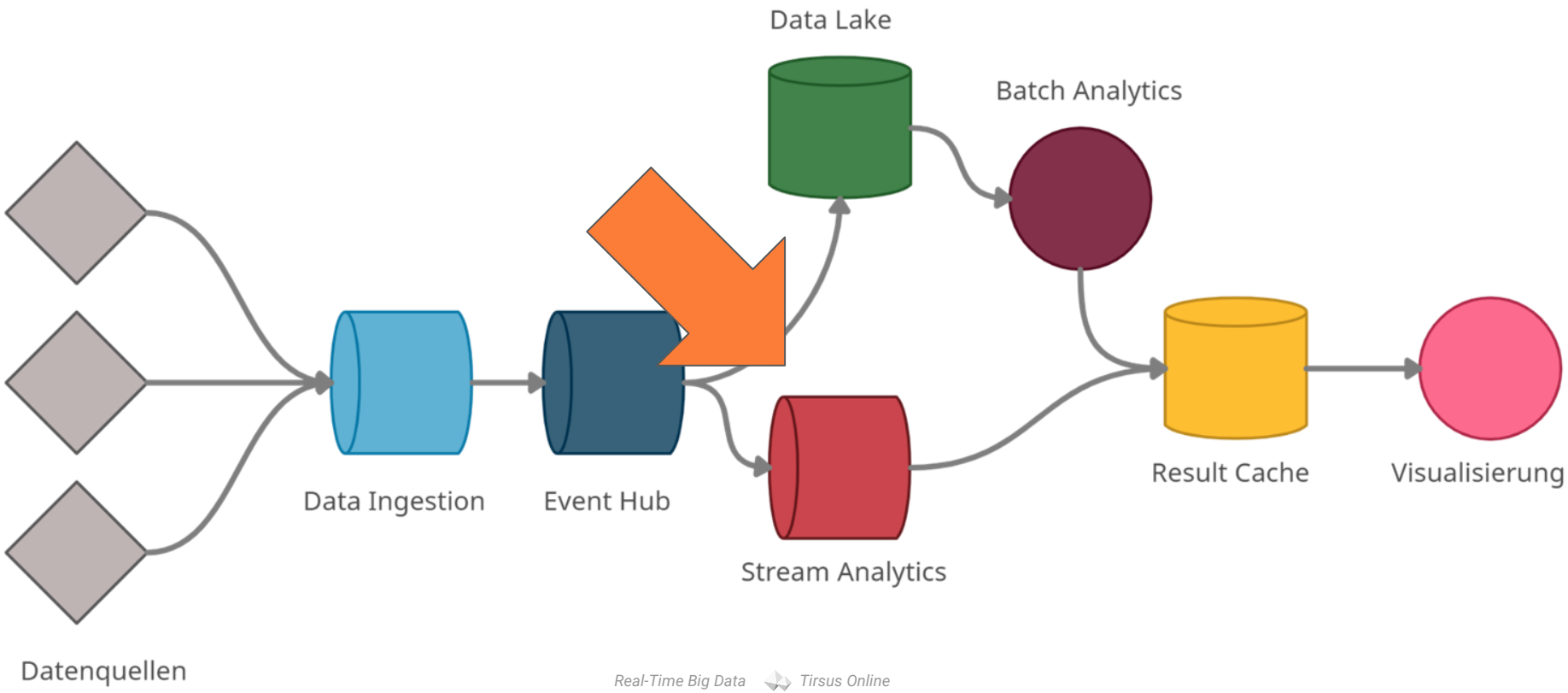
Das Gesamtsystem funktioniert weiter auch, wenn nicht alle Nachrichten, die zwischen den Systemkomponeten getauscht werden, eine zeitnahe Antwort erhalten, oder auch wenn Nachrichten verloren gehen.



Brokerausfall?

Apache Spark als Stream Analytics Engine

Apache Spark als Stream Analytics Engine



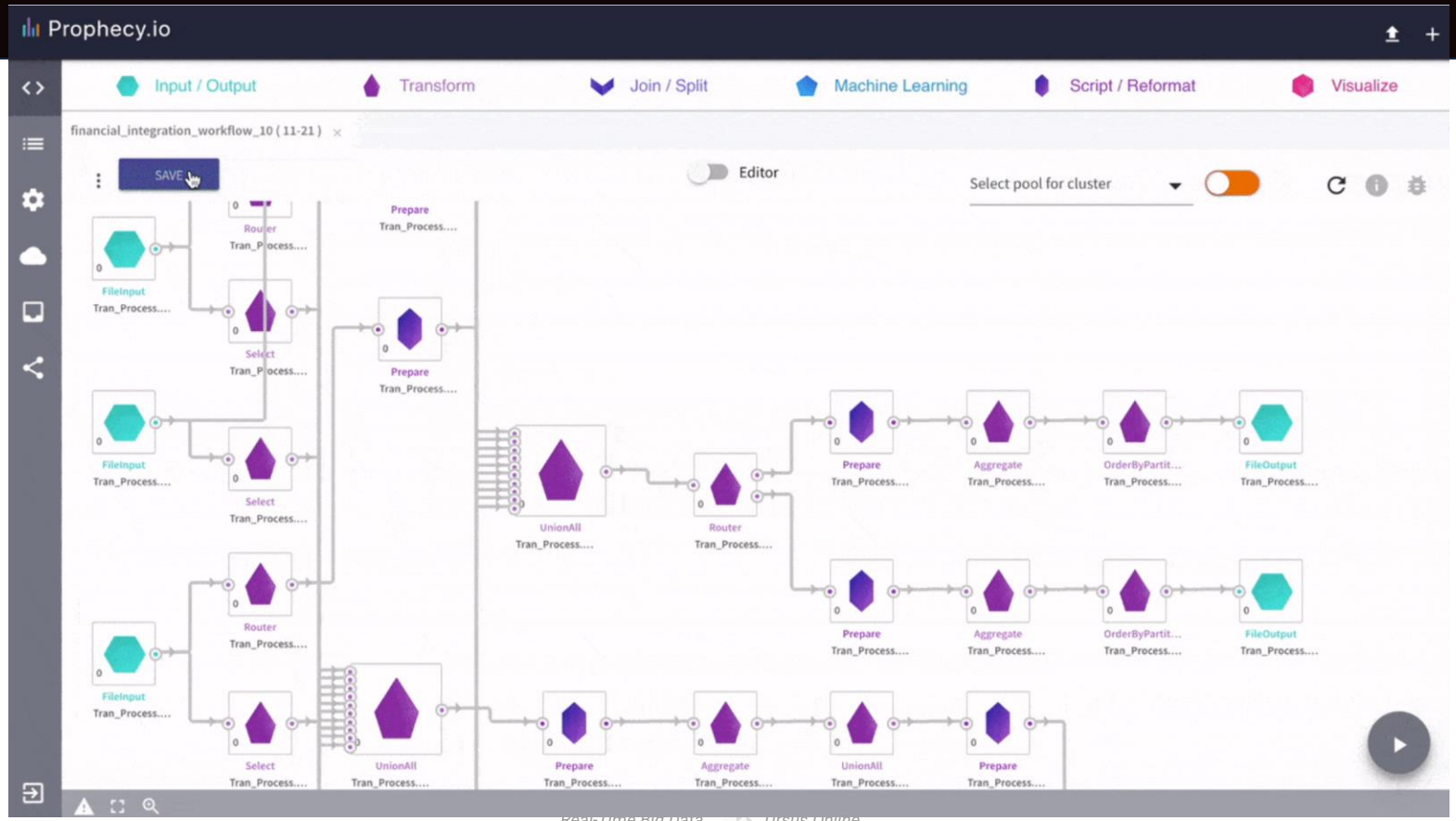
Daten von Apache Kafka lesen

```
messungen = spark\
  .readStream\
  .format('kafka')\
  .option('kafka.bootstrap.servers', 'kafka-1:9092')\
  .option('subscribe', 'sunus')\
  .option('startingOffsets','earliest')\
  .option('includeTimestamp', 'true')\
  .load()\
  .selectExpr("cast(value as string) as value","timestamp")\
  .selectExpr("cast(split(value,',')[0] as double) as x",
    "cast(split(value,',')[1] as double) as y",
    "cast(split(value,',')[2] as timestamp) as event_time",
    "timestamp",
    "substring(split(value,',')[2], 1, 16) as minute"
  )
```

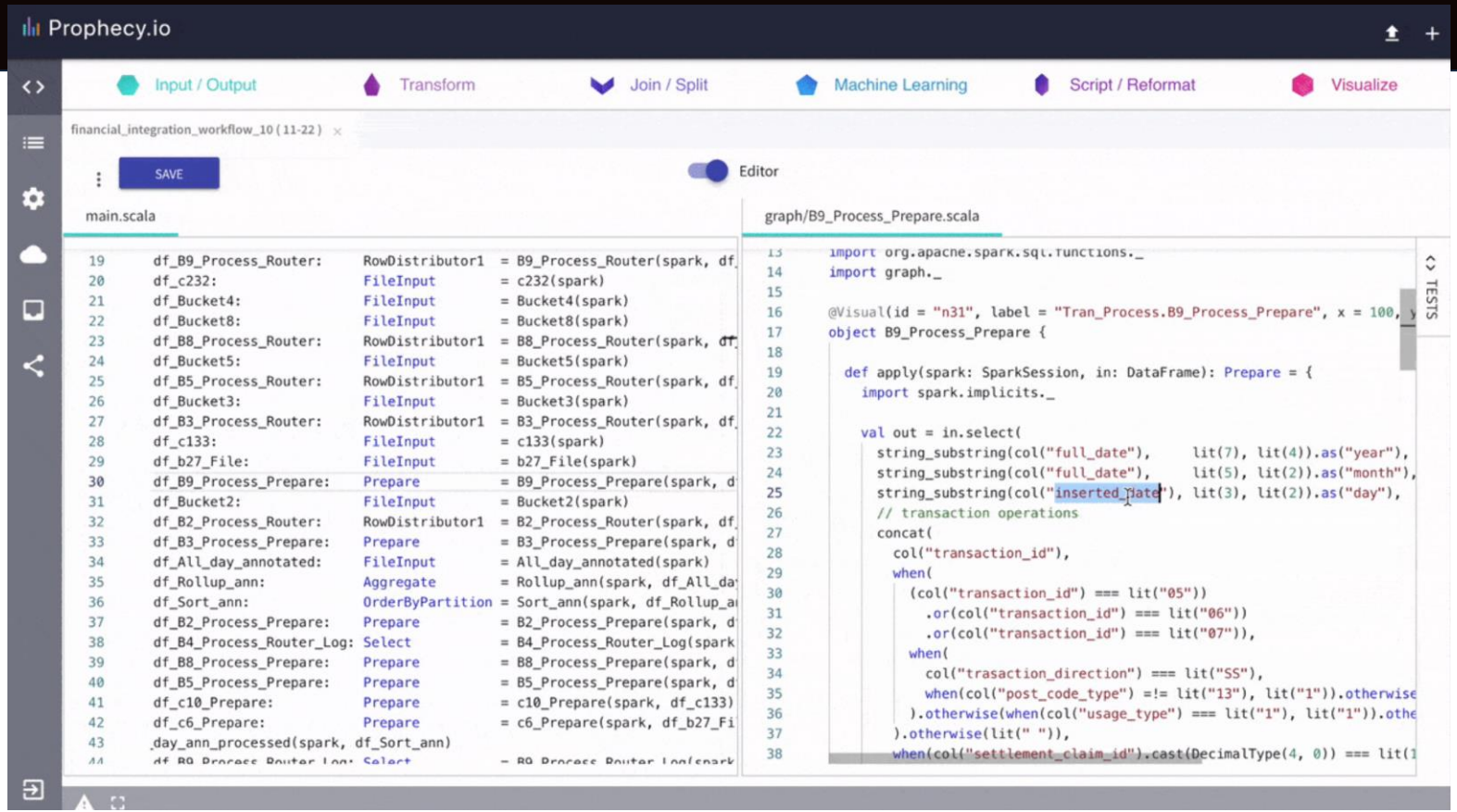
Daten analysieren und auf Result Cache schreiben

```
windowedCounts = messungen.groupBy(
 window(messungen.timestamp, '3 seconds'),
 messungen.minute
 )\
 .avg()
data = windowedCounts.select('window.start', 'window.end', 'avg(y)', 'minute')
query = data\
 .writeStream\
 .outputMode("update")\
 .foreachBatch(writeToResultCache)\
 .start()\
 .awaitTermination()
```

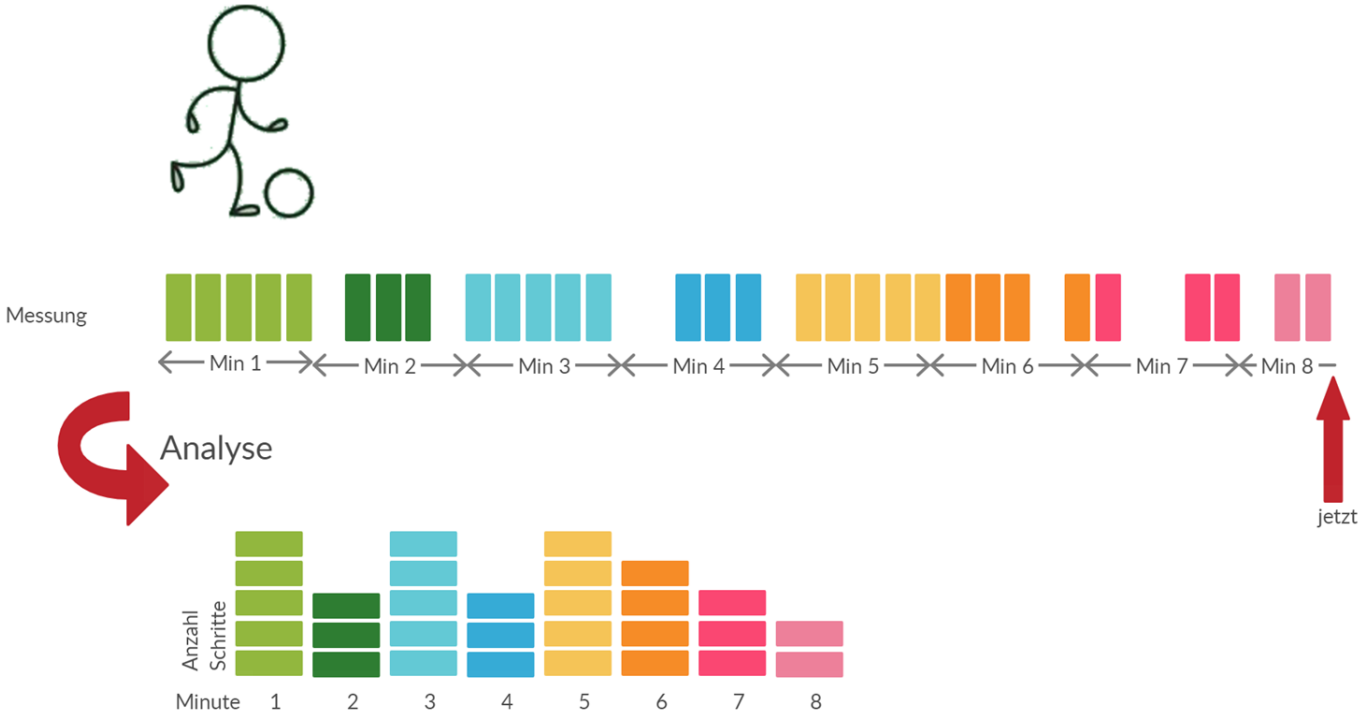
Spark IDE – ein neues Beispiel

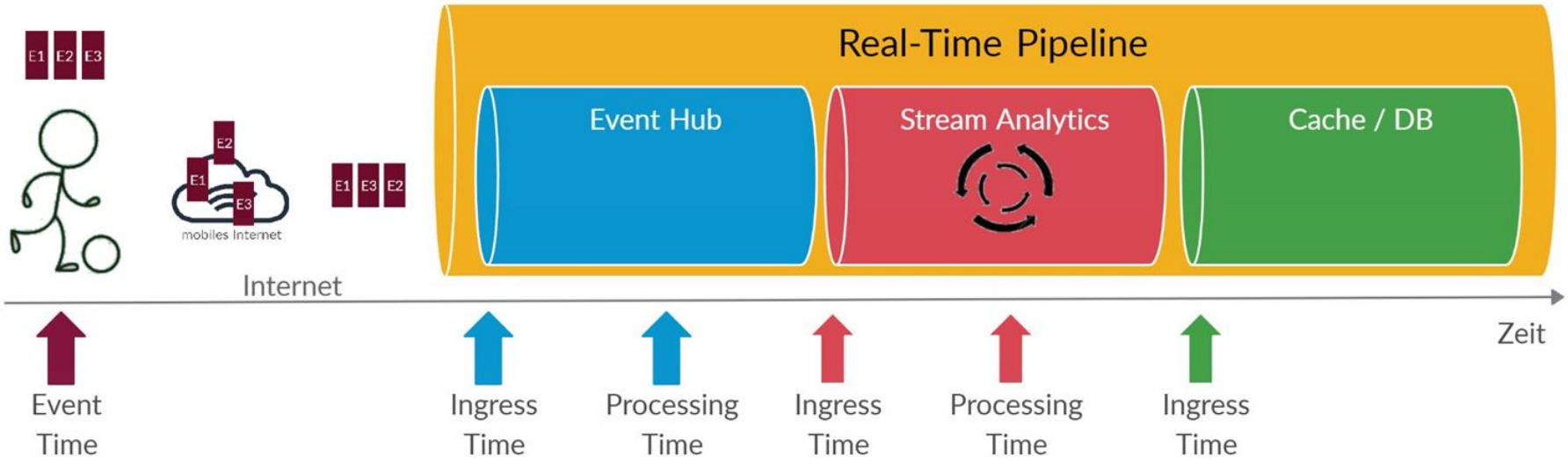


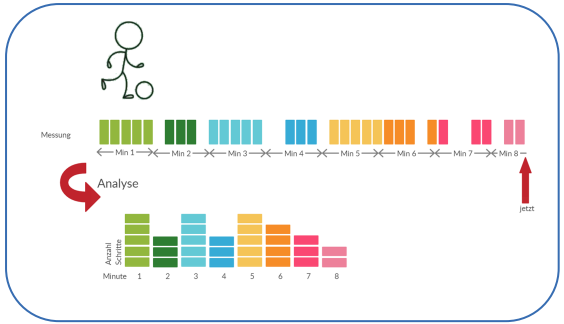
Spark IDE – ein neues Beispiel



Blick hinter die Kulissen von Apache Spark

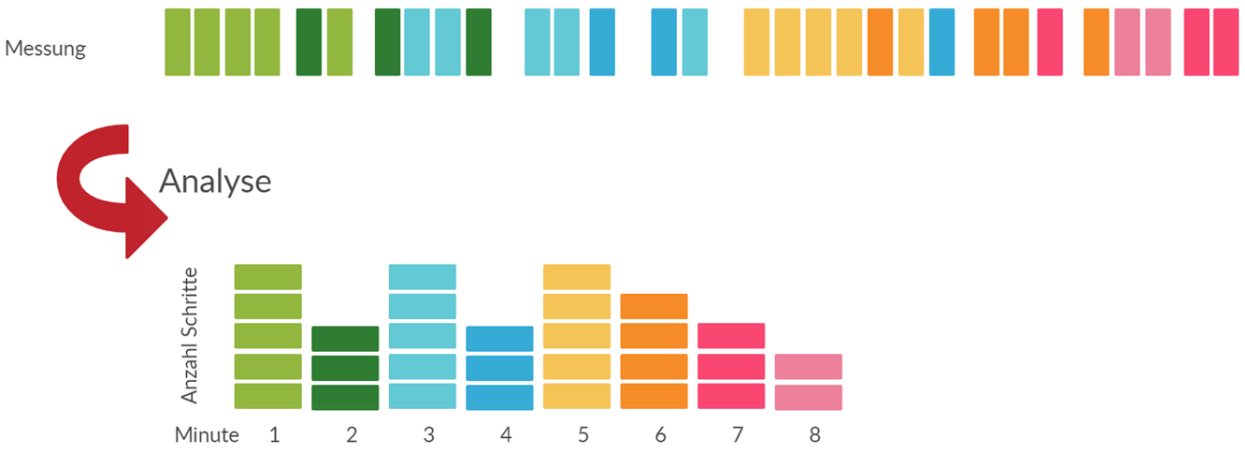

Messung
Min 1
Min 2
Min 3
Min 4
Min 5
Min 6
Min 7
Min 8
jetzt
Analyse
Anzahl Schritte
Minute
1
2
3
4
5
6
7
8


E1 E2 E3
E2
E1
E3
mobiles Internet
E1 E3 E2
Real-Time Pipeline
Event Hub
Stream Analytics
Cache / DB
Internet
Zeit
Event Time
Ingress Time
Processing Time
Ingress Time
Processing Time
Ingress Time



Ziel: Exactly Once

```
schritteProMinute = schrittstream \
.withWatermark("timestamp","2 minutes") \
.groupBy(
  window (schrittstream.timestamp, "1 minute"),
  schrittstream.spieler) \
.count()
```



Dive Deeper

Exactly Once Semantik

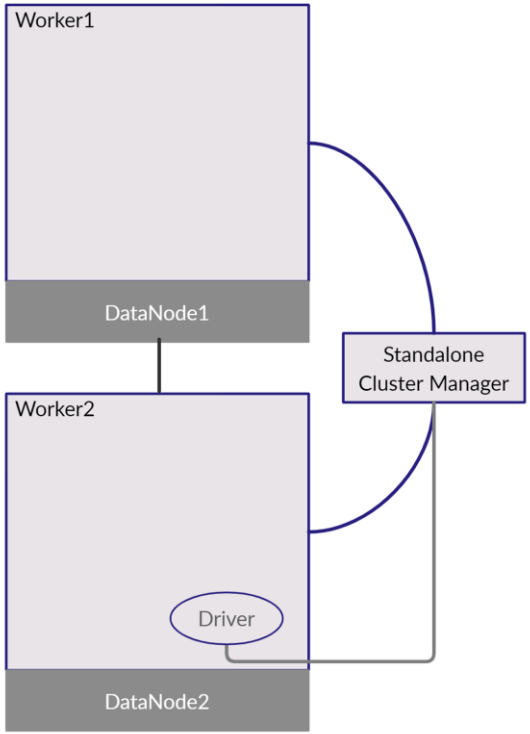
Exactly Once Jedes Event wird genau ein Mal verarbeitet.

At Most Once Wir nehmen in Kauf, dass im Fehlerfall Events verloren gehen.

At Least Once Wir nehmen in Kauf, dass im Fehlerfall Events mehrfach verarbeitet werden.

Exactly Once -> Transaktionen

Cluster Manager
Workers
HDFS



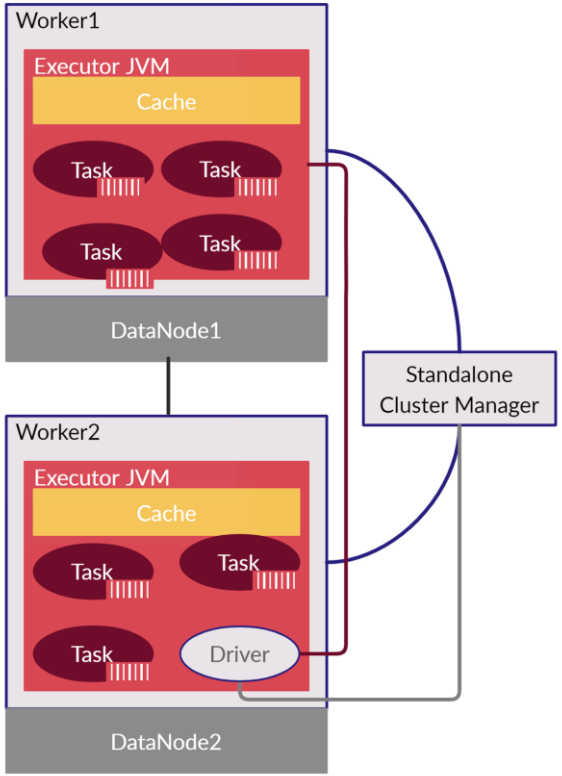
Standalone Cluster Manager

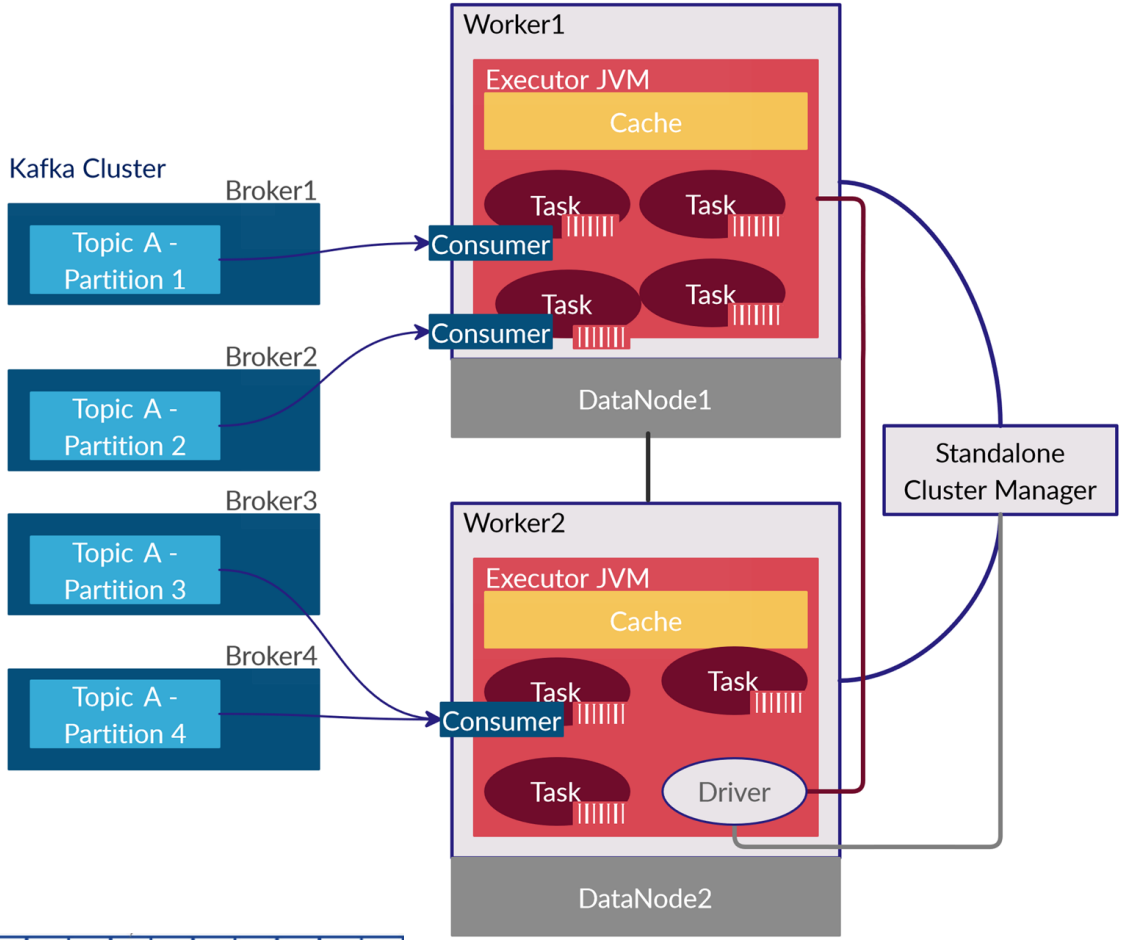
Apache YARN

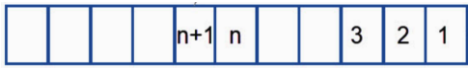
Apache Mesos

Kubernetes

Driver - Executor





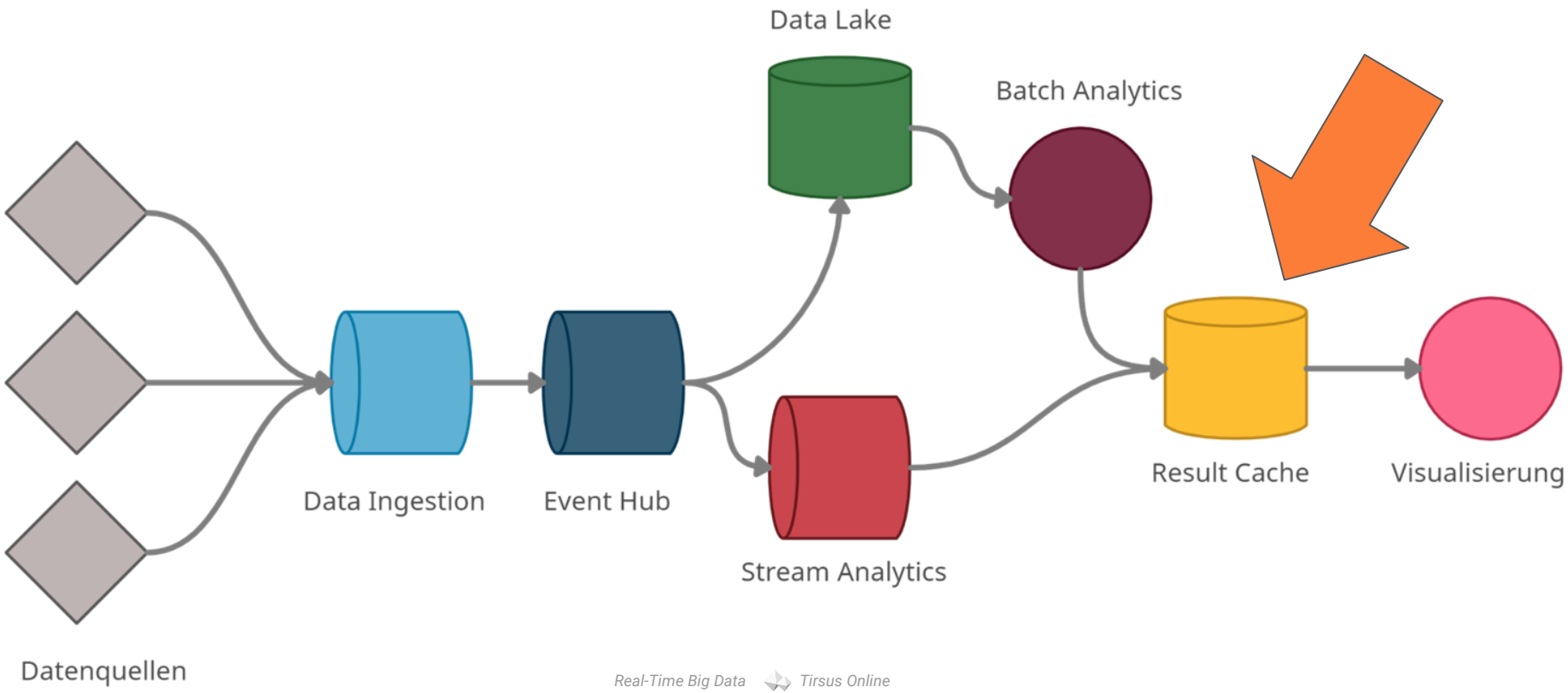


Spark liest von Kafka

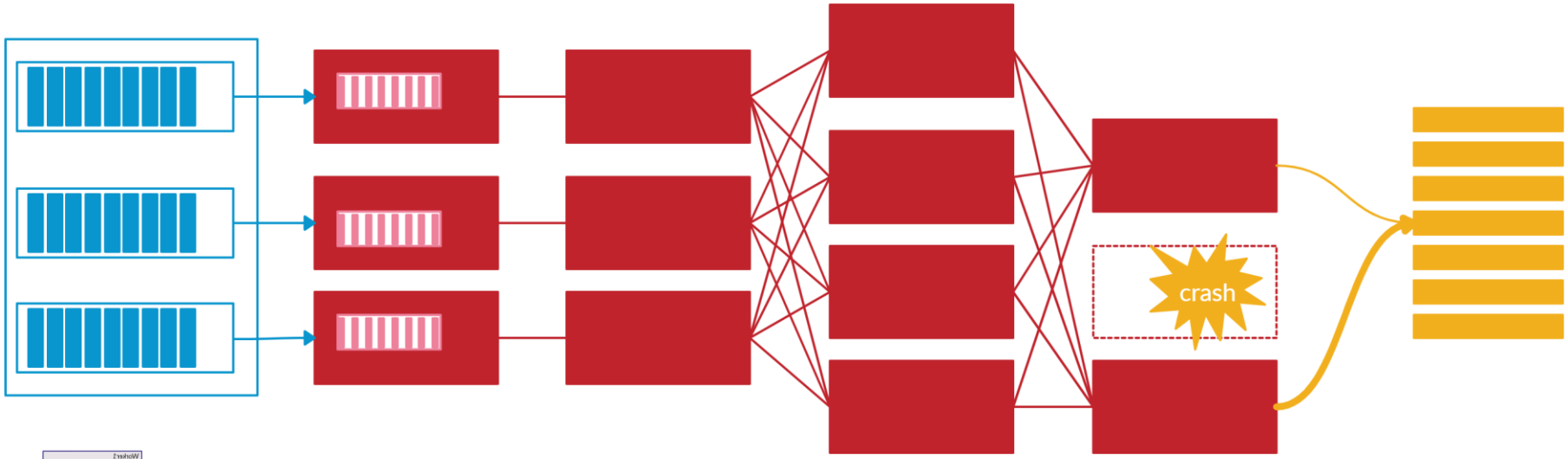
Workerausfall?

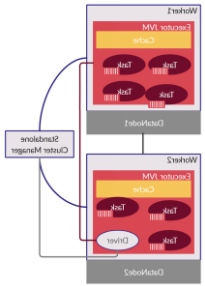
Der Result Cache

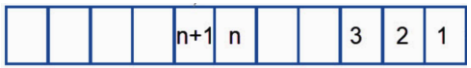
Der Result Cache



Idempotente Schreiboperationen







Trade-Offs der Real-Time Analyse großer Datenströme

Free E-Mail-Training

https://tirsus.com/pipeline/


Ursula Deriu
ursula.deriu@tirsus.com
https://www.linkedin.com/in/ursuladeriu
tirsus.com/big-data
Big Data Training

Q & A