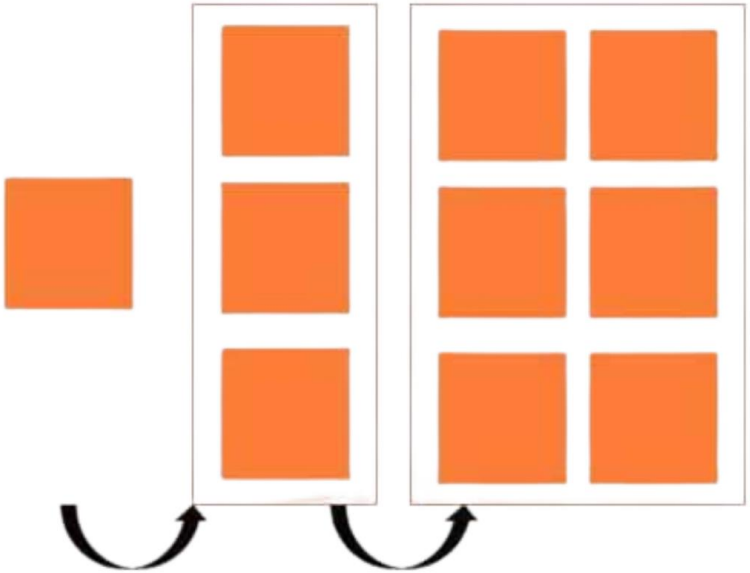
Herausforderungen des Realtime Big Data Stream Processing

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

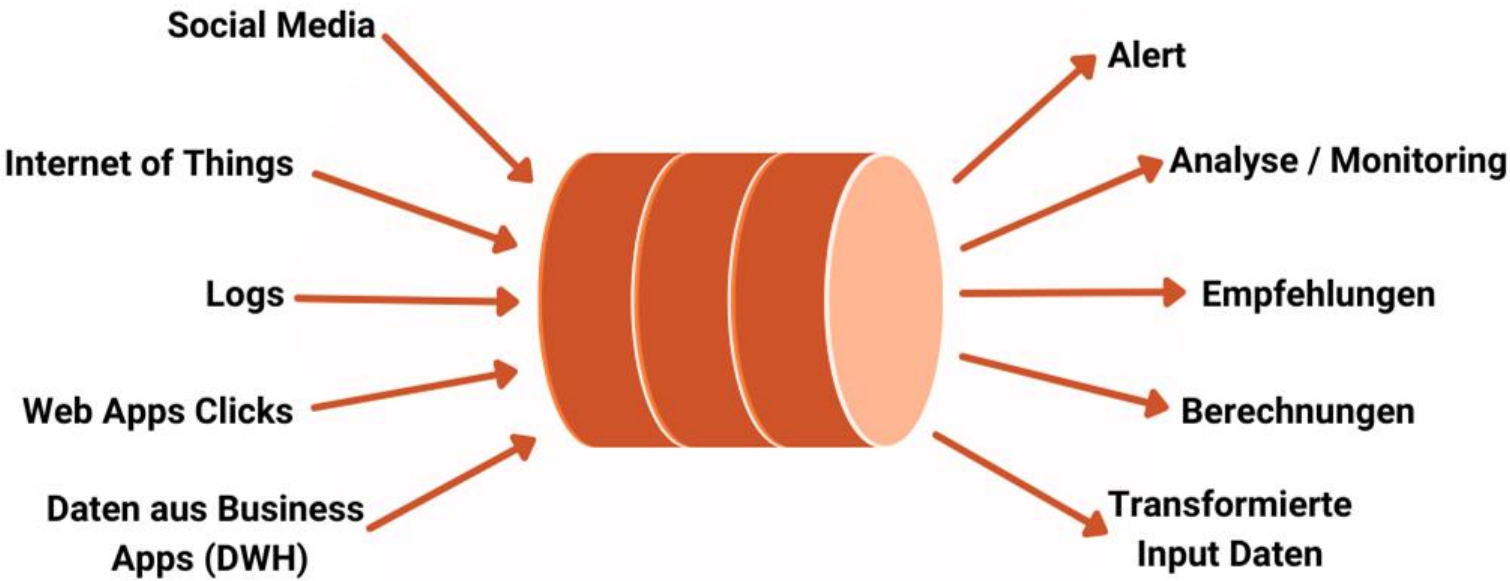


Big Data RZ

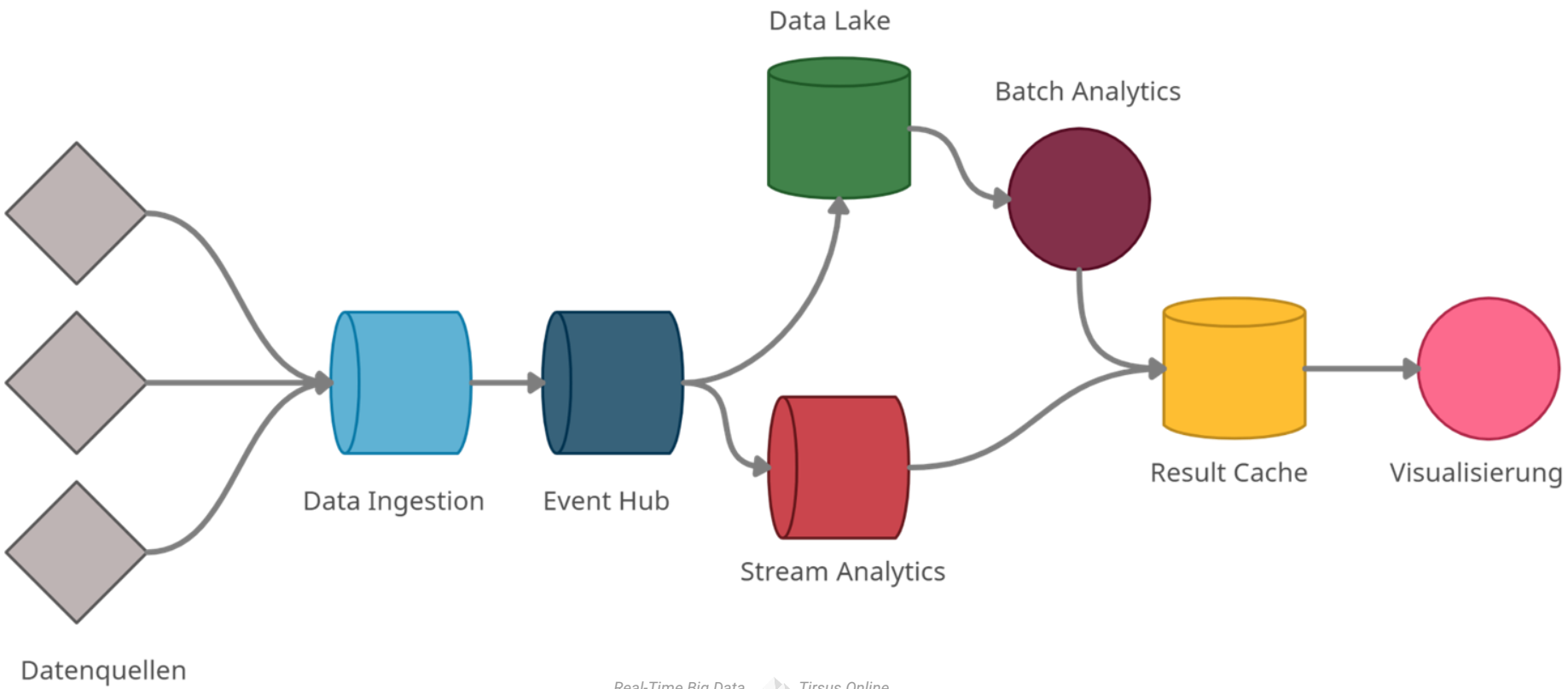
Quelle Bilder: Google

Braucht Big Data auch Big Infrastructure?

Einsatzmöglichkeiten

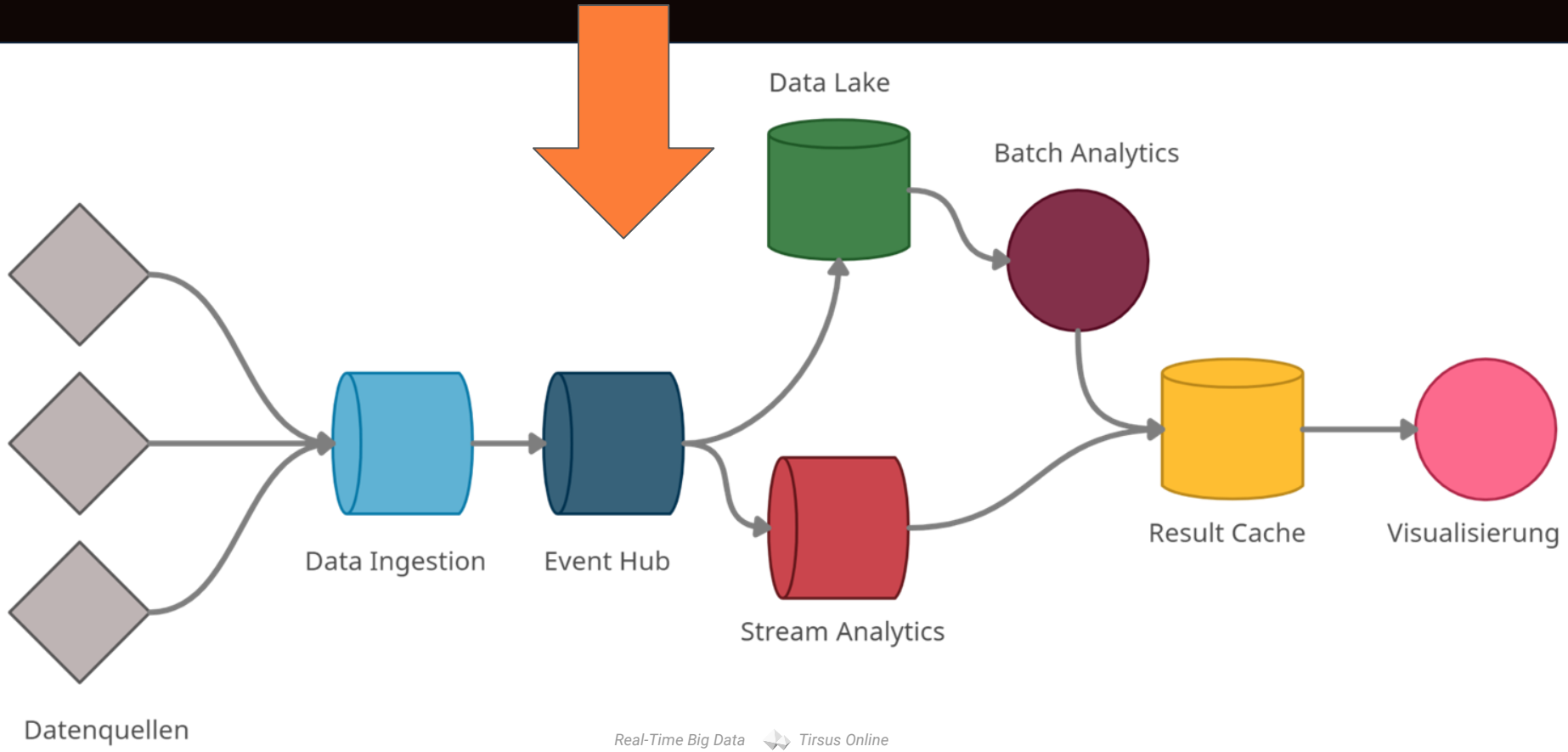


Anatomie einer Big-Data-Streaming-Pipeline

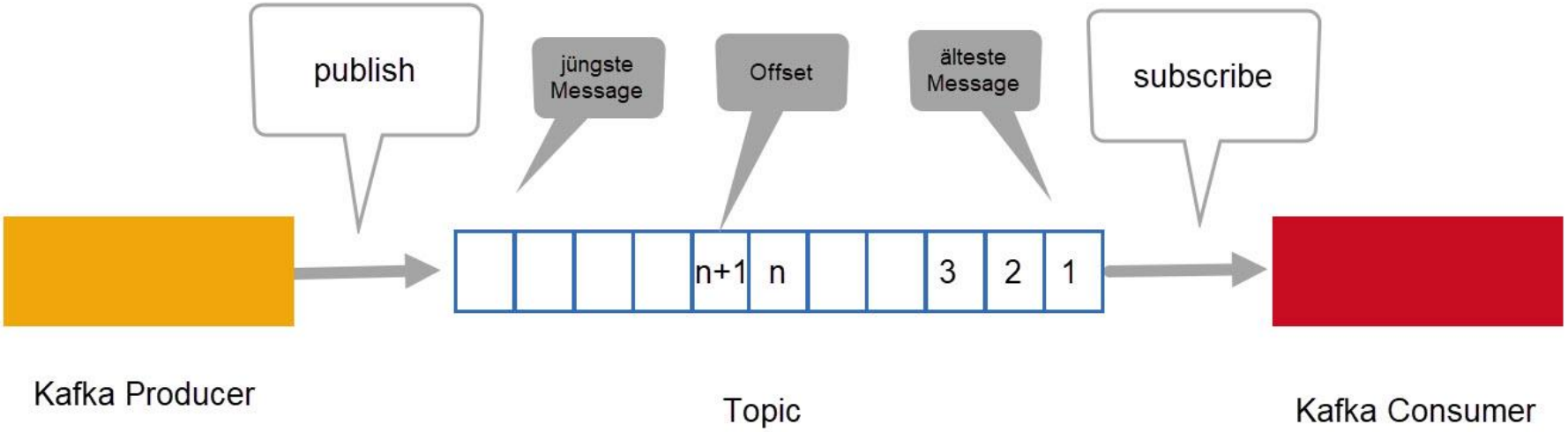


Apache Kafka als Event Hub

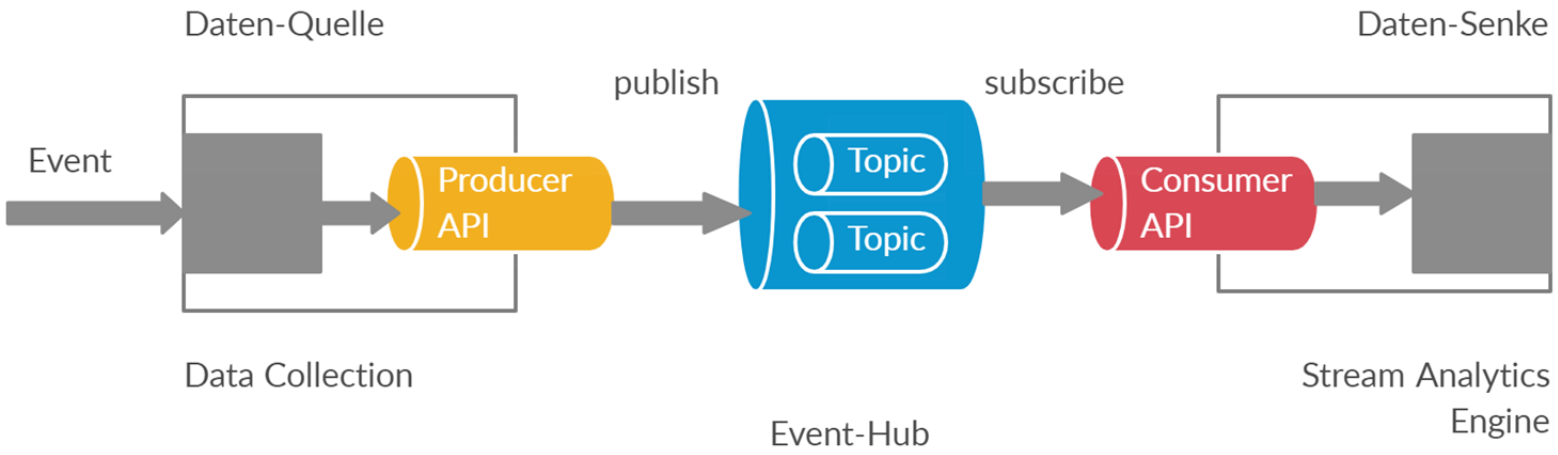
Apache Kafka als Event Hub



Warteschlange

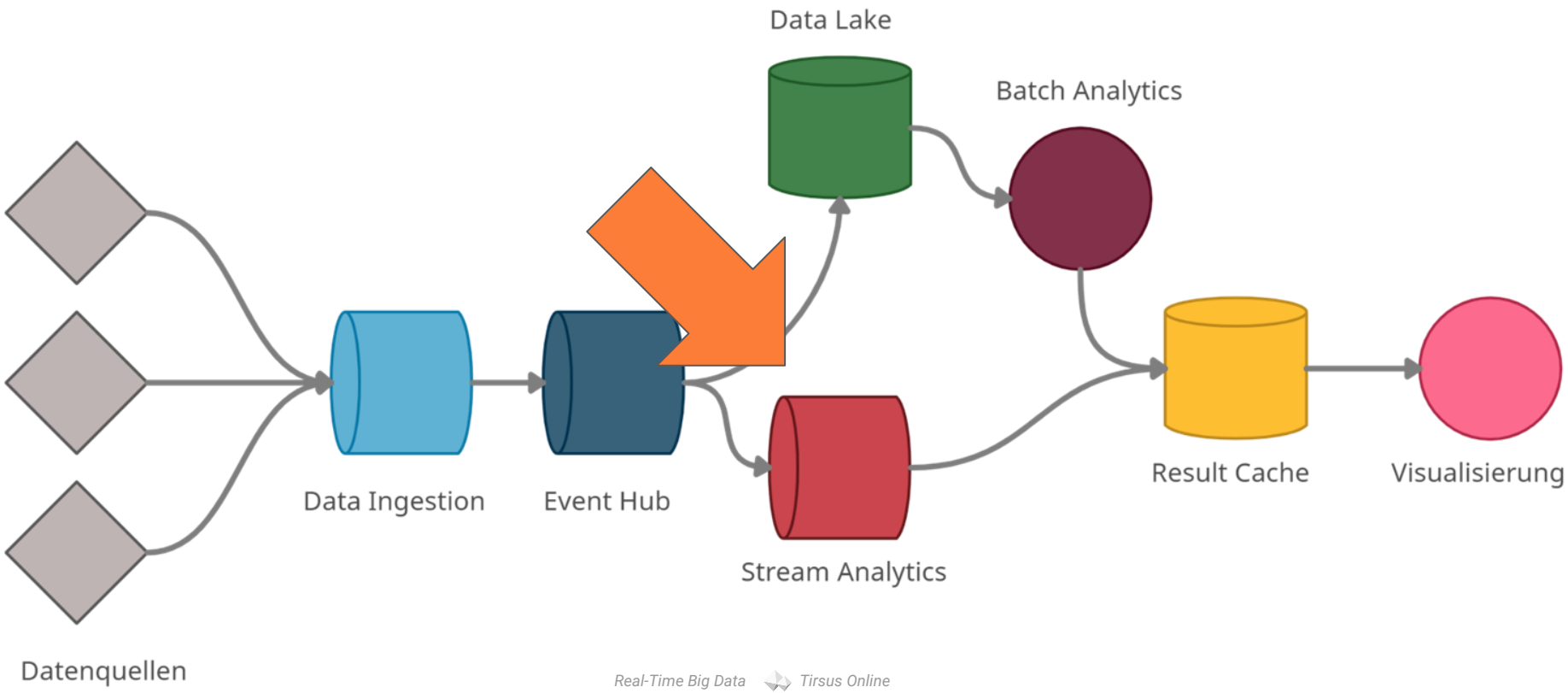


Verteilter Event Hub

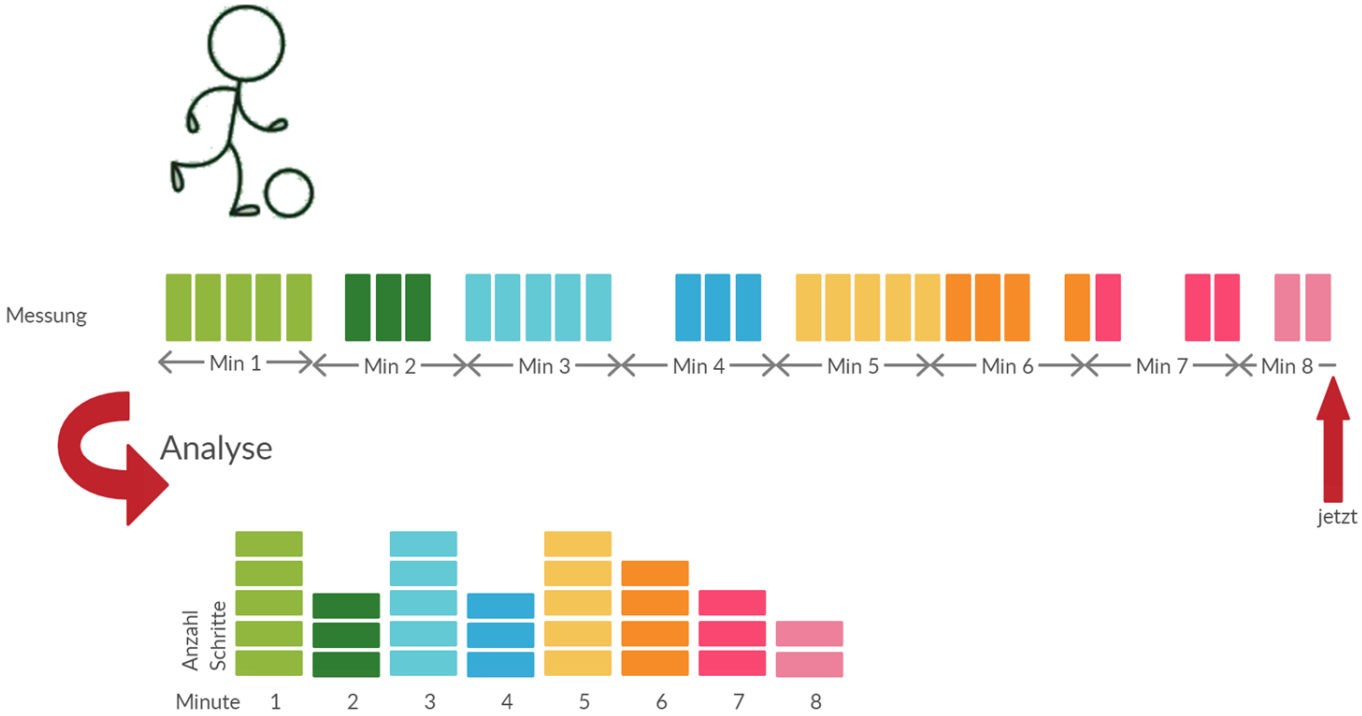


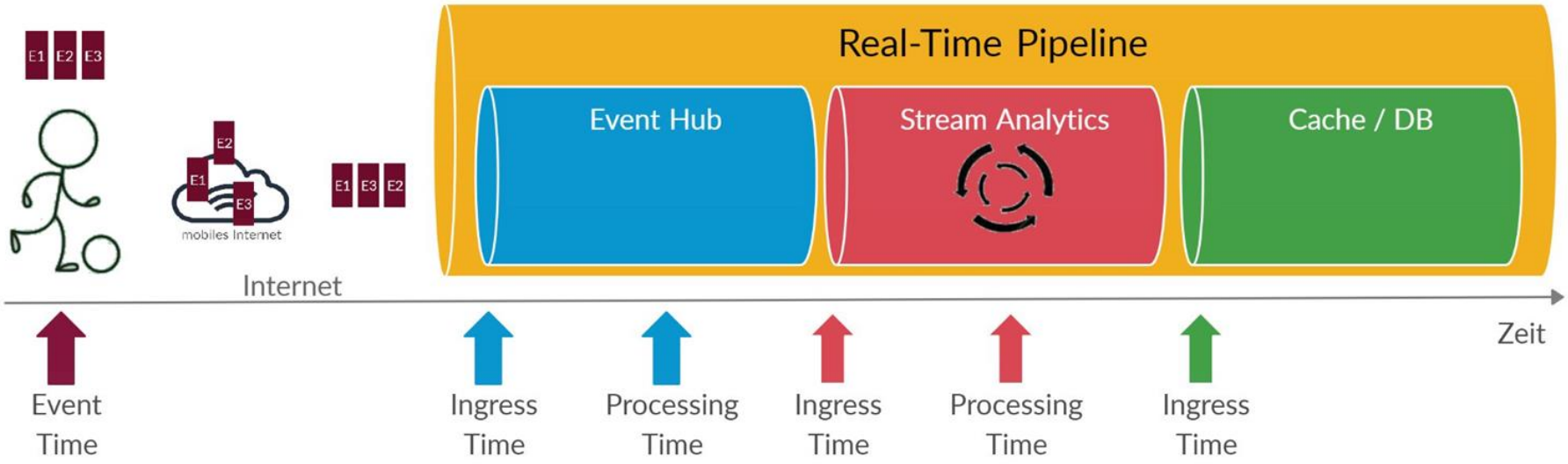
Apache Spark als Stream Analytics Engine

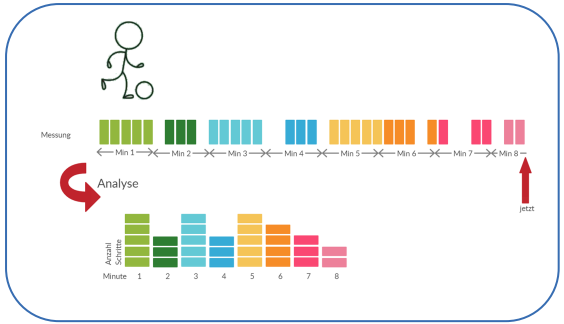
Apache Spark als Stream Analytics Engine



Blick hinter die Kulissen von Apache Spark

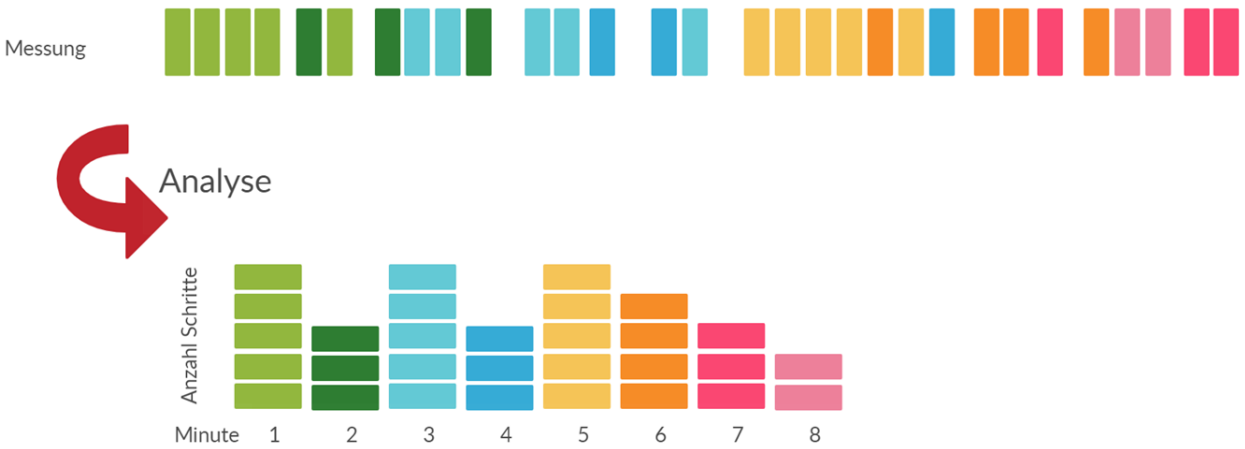

Messung
Min 1
Min 2
Min 3
Min 4
Min 5
Min 6
Min 7
Min 8
jetzt
Analyse
Anzahl Schritte
Minute
1
2
3
4
5
6
7
8


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

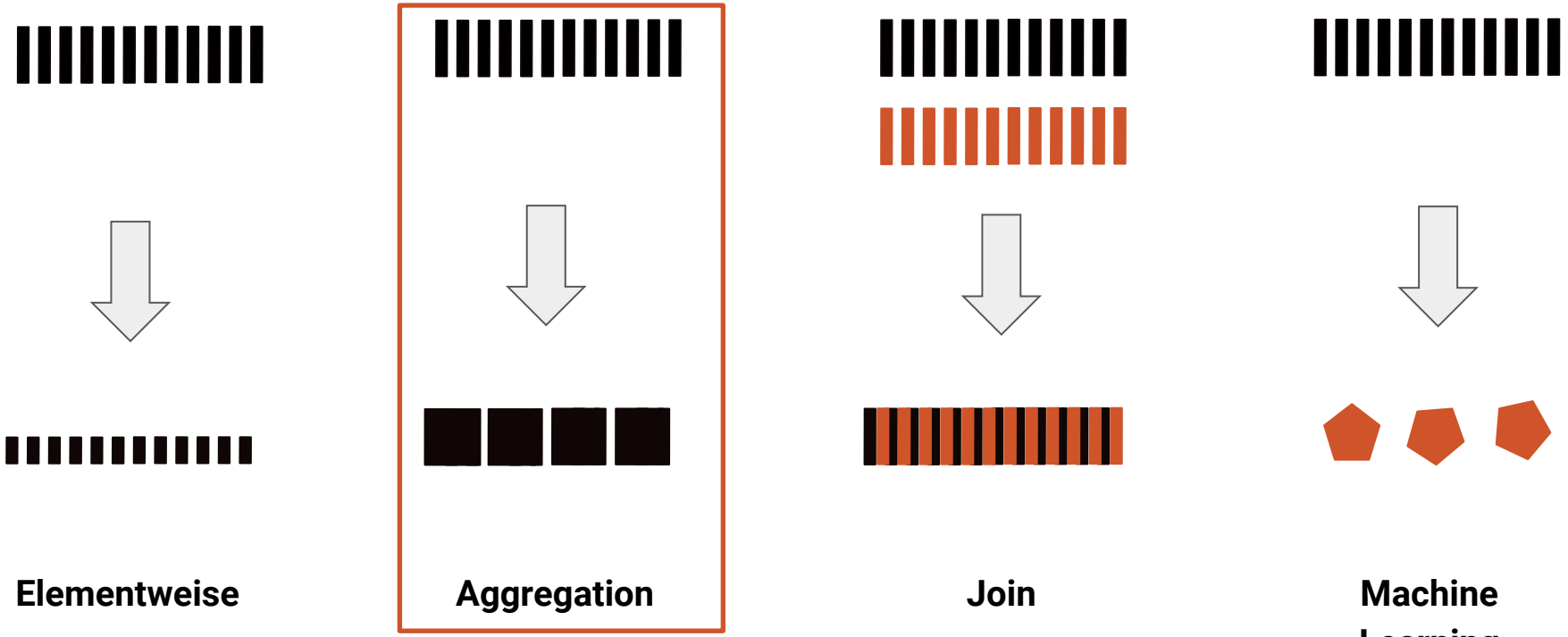


Ziel: Exactly Once

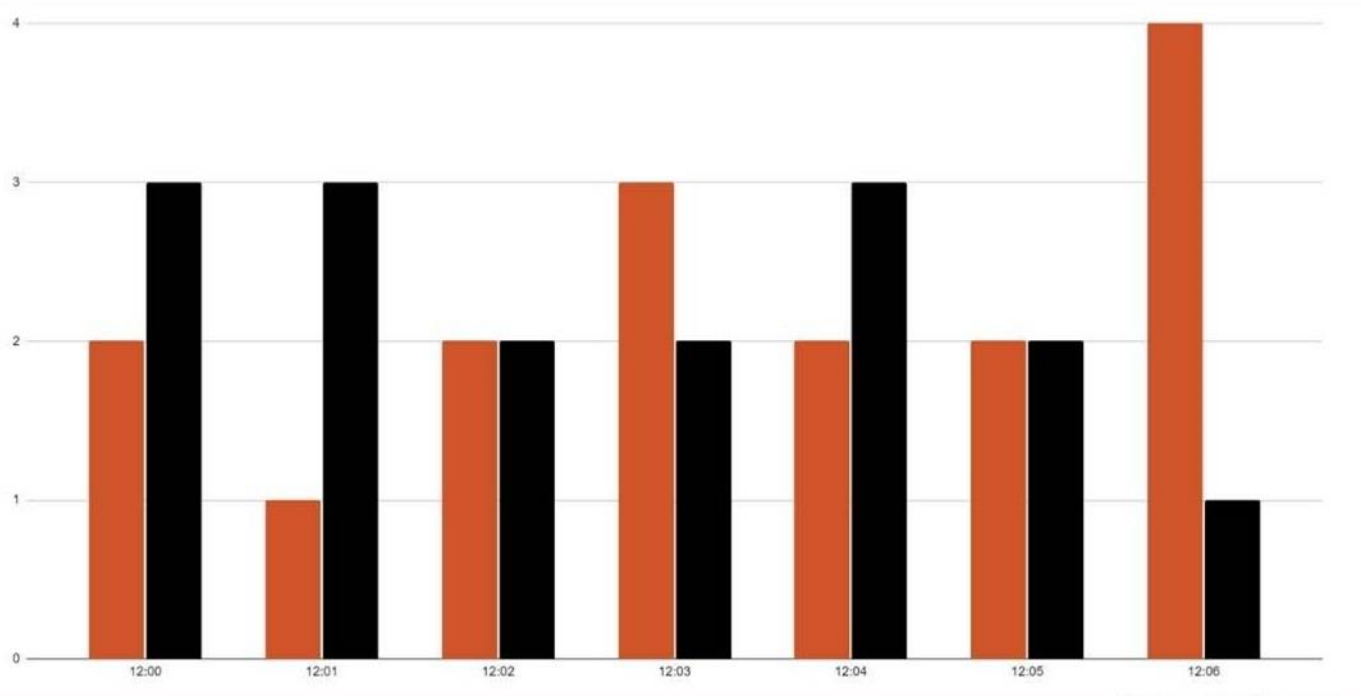
```
schritteProMinute = schrittstream \
.withWatermark("timestamp","2 minutes") \
.groupBy(
  window (schrittstream.timestamp, "1 minute"),
  schrittstream.spieler) \
.count()
```



Data Stream Processing



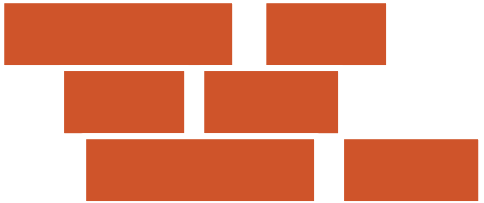
Aggregation



Windows - Zeitfenster

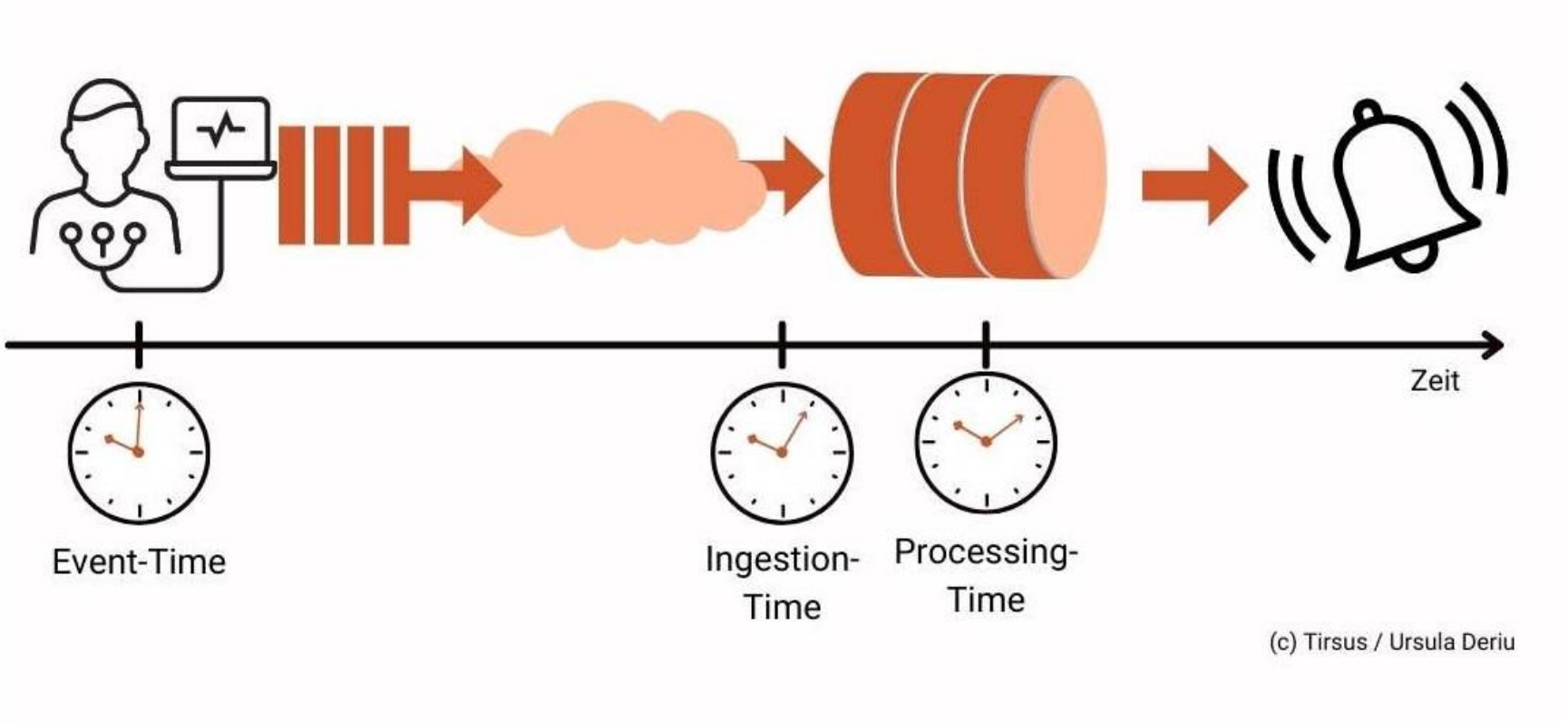
Fixed

Sliding

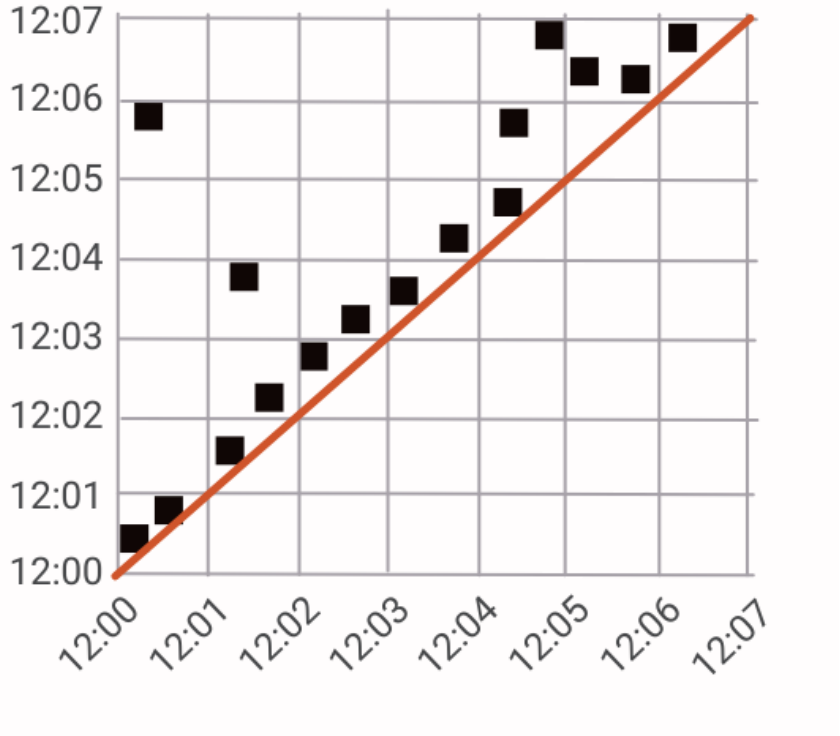


Sessions

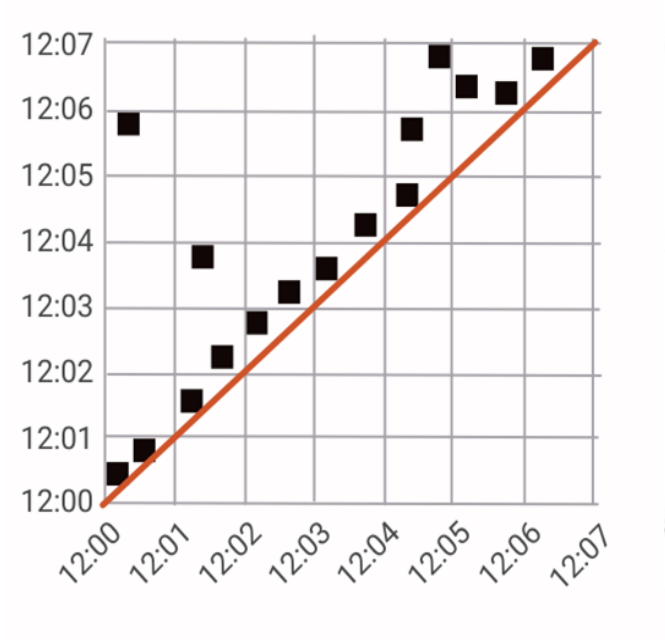
Beispiel: Herzüberwachung

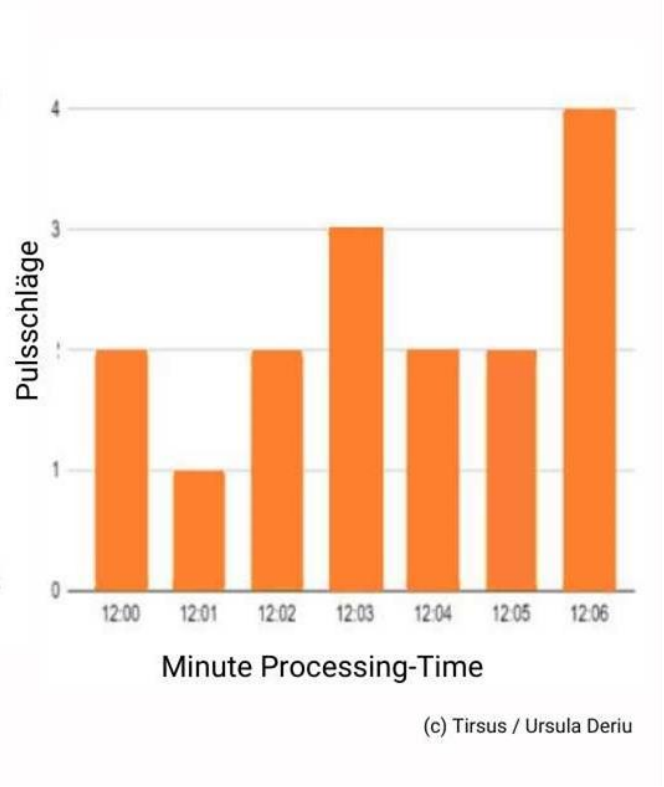


Windows - nach Processing Time

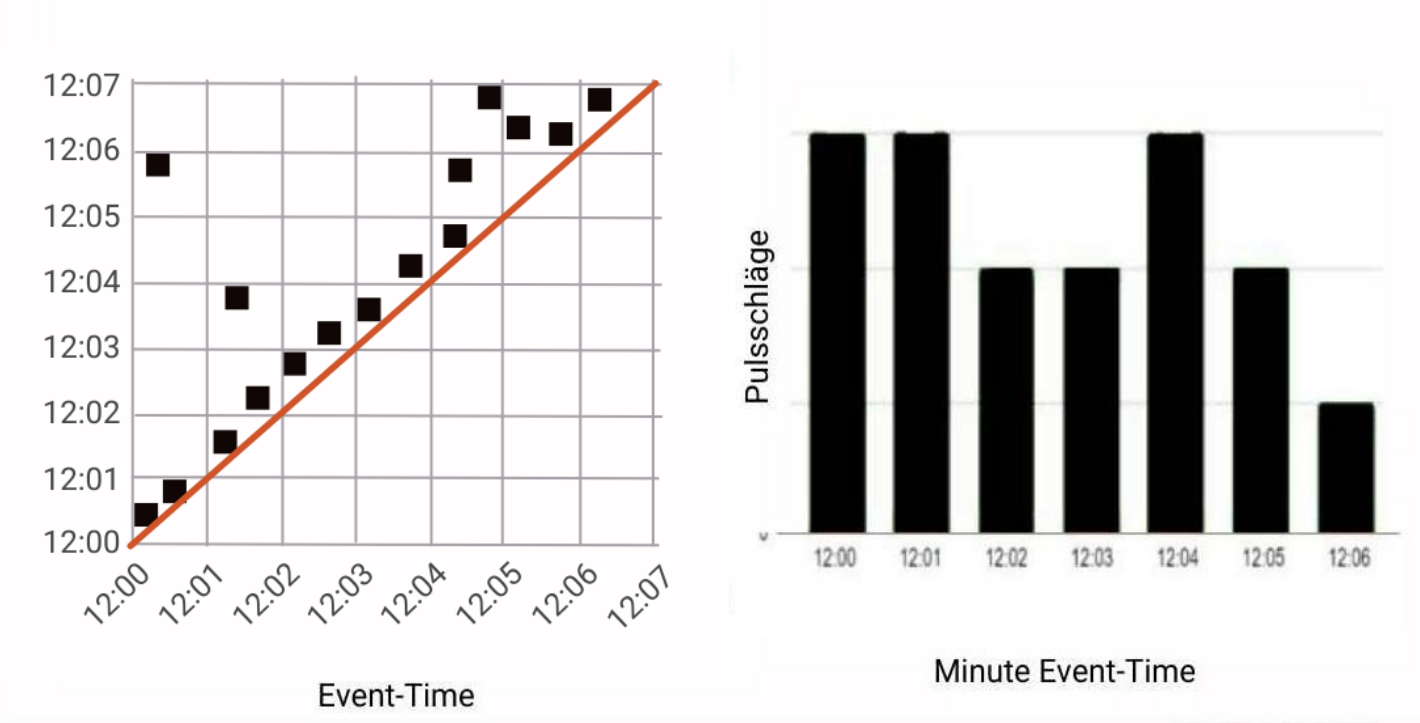


Processing-Time - “einfach erfüllbar”

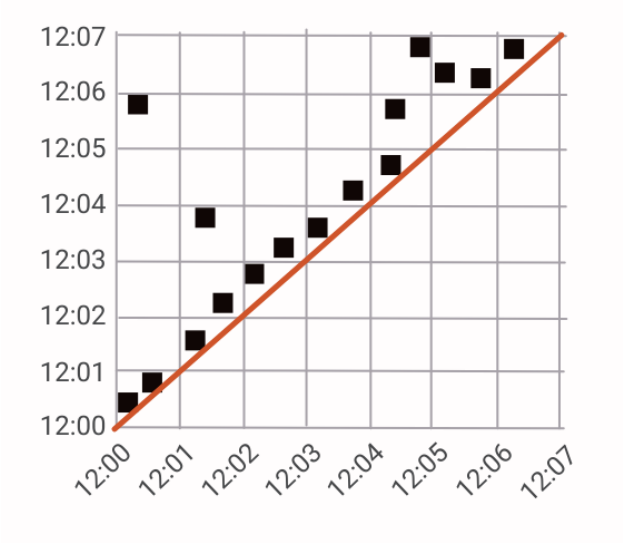


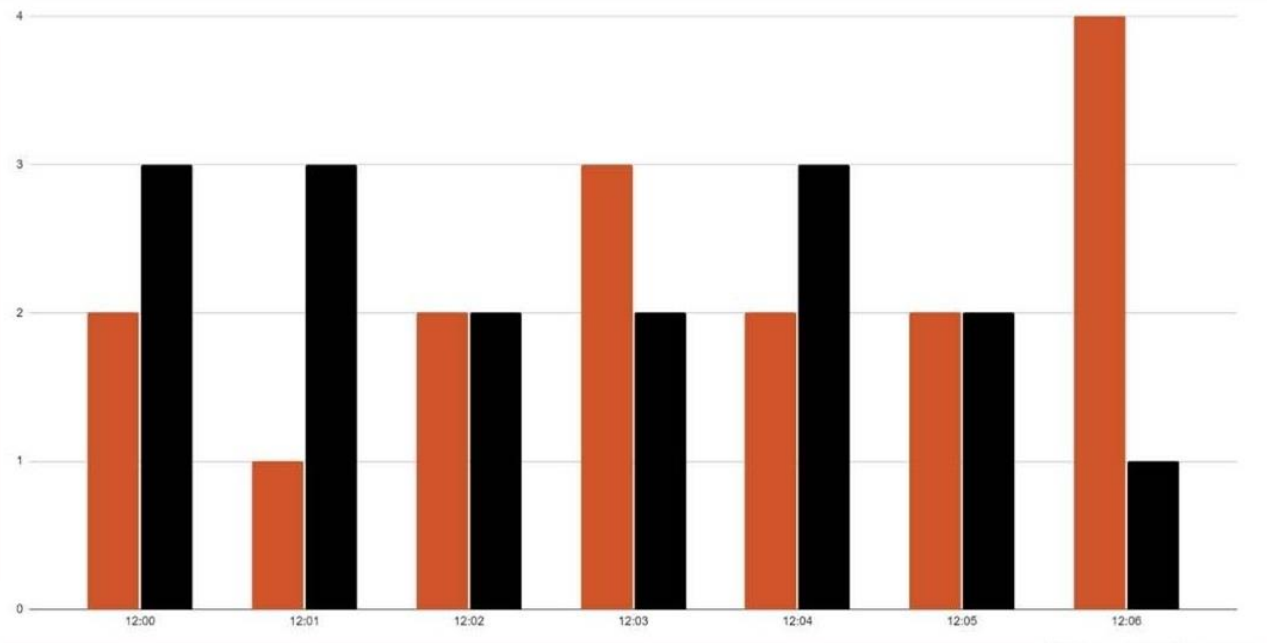


Erwartung: Event-Time

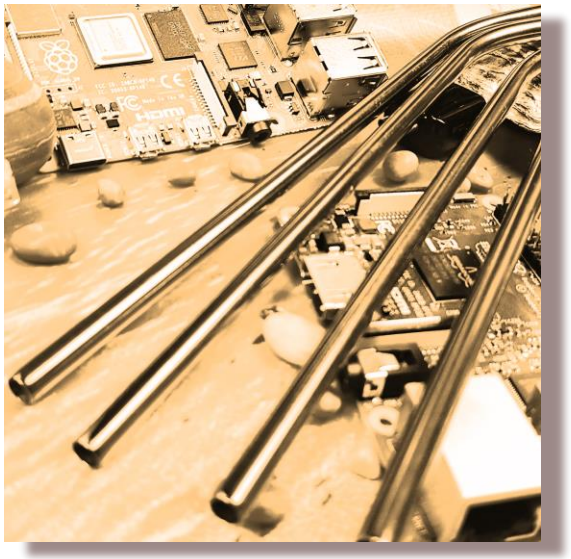


Event-Time vs. Processing-Time





Triggers und Watermarks



Animation hier:

https://tirsus.com/stream-processing-zeit/

Dive Deeper

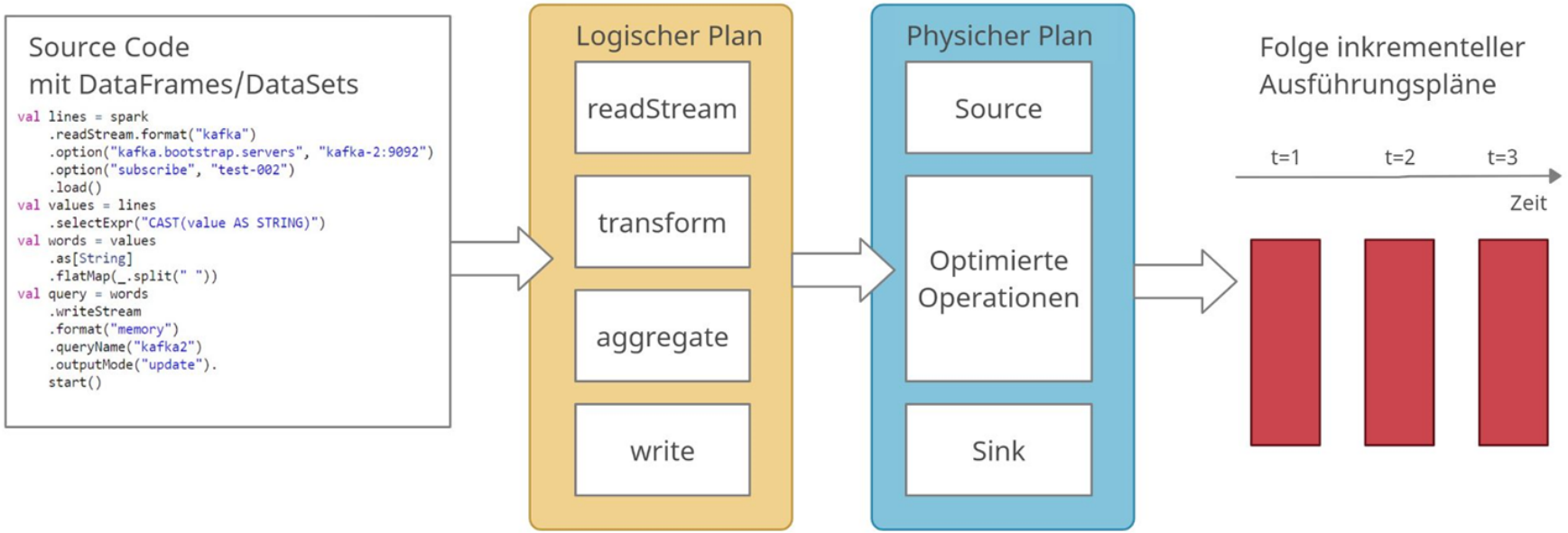
Exactly Once Semantik

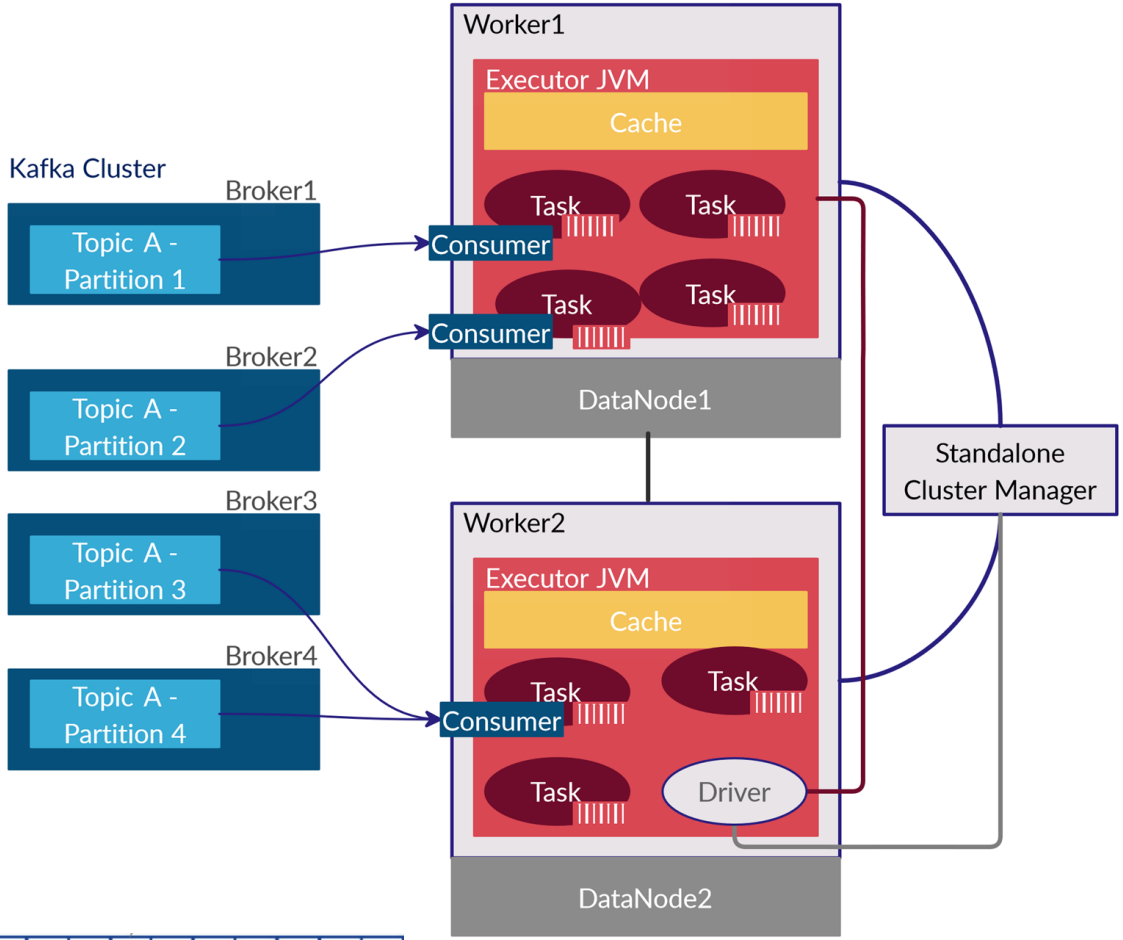
Exactly Once Jedes Event wird genau ein Mal verarbeitet.

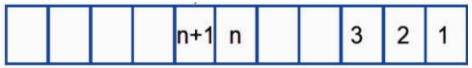
At Most Once Wir nehmen in Kauf, dass im Fehlerfall Events verloren gehen.

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

Exactly Once -> Transaktionen


Source Code
mit DataFrames/DataSets
val lines = spark
.readStream.format("kafka")
.option("kafka.bootstrap.servers", "kafka-2:9092")
.option("subscribe", "test-002")
.load()
val values = lines
.selectExpr("CAST(value AS STRING)")
val words = values
.as[String]
.flatMap(_.split(" "))
val query = words
.writeStream
.format("memory")
.queryName("kafka2")
.outputMode("update").
start()
Logischer Plan
readStream
transform
aggregate
write
Physicher Plan
Source
Optimierte
Operationen
Sink
Folge inkrementeller
Ausführungspläne
t=1
t=2
t=3
Zeit



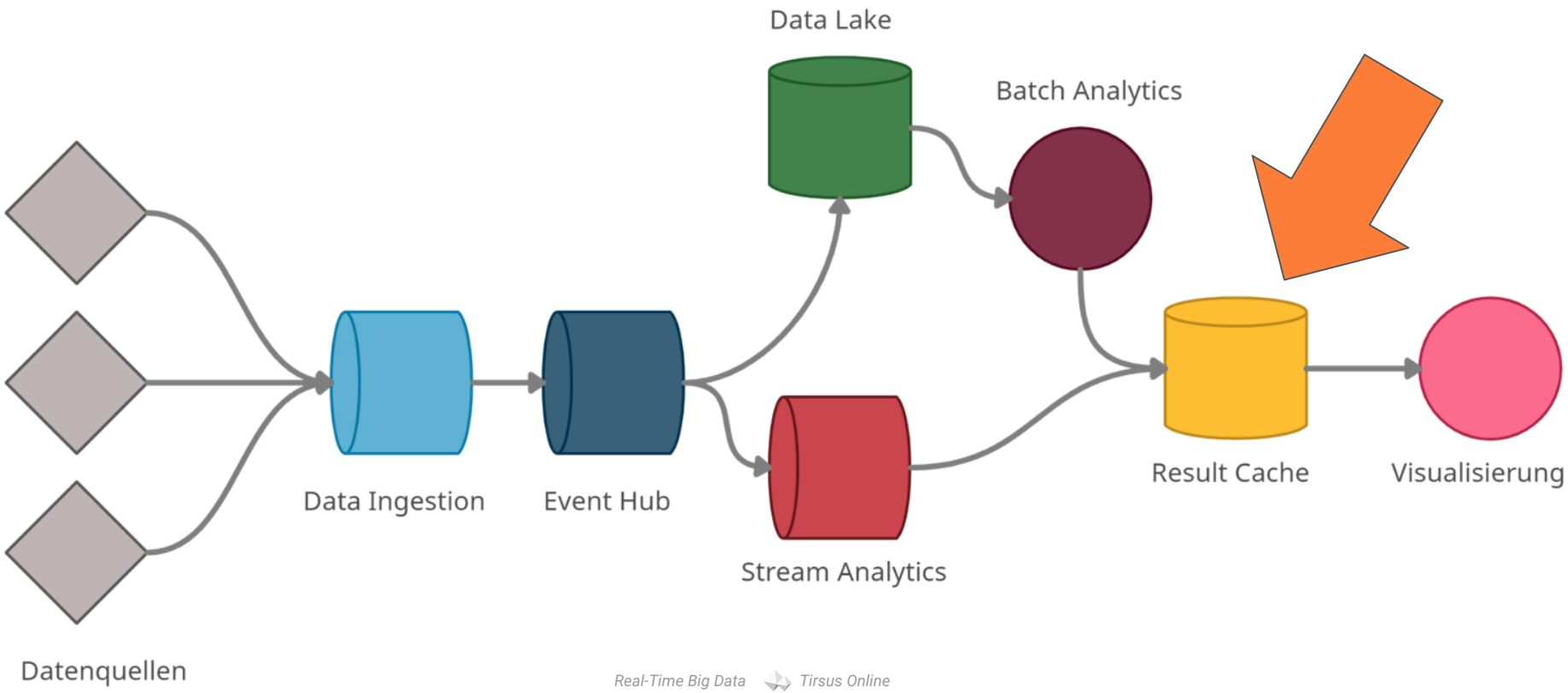


Spark liest von Kafka

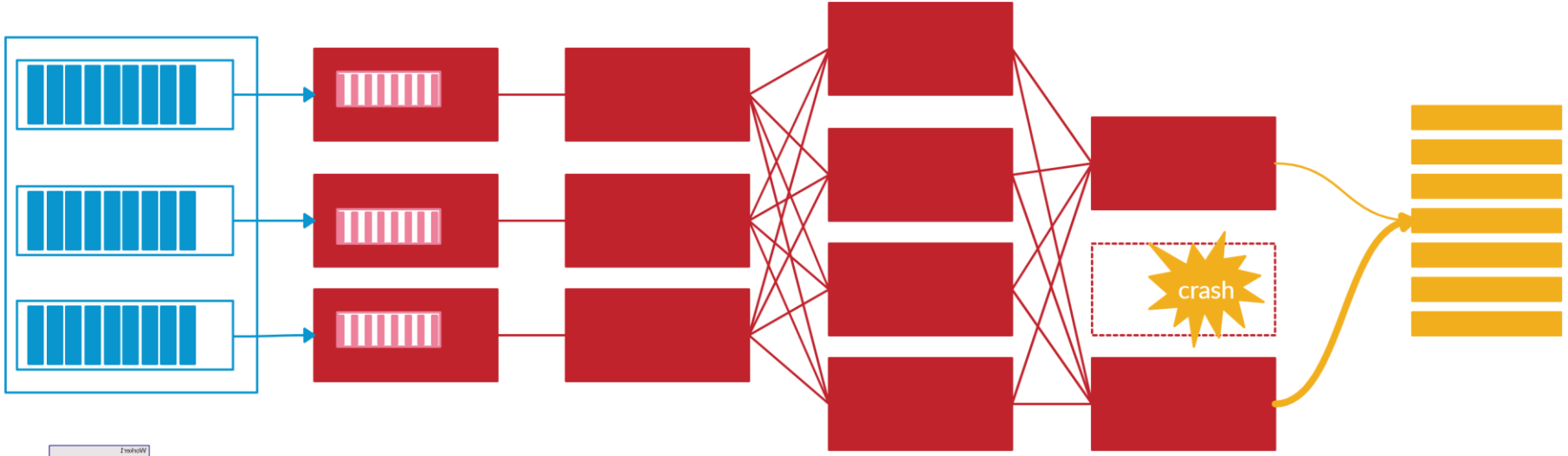
Workerausfall?

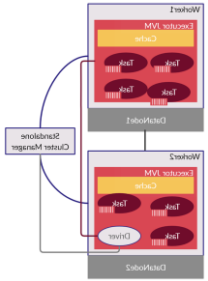
Der Result Cache

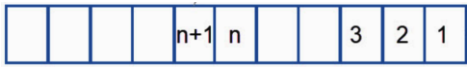
Der Result Cache



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