DOZIERENDER: MAX MUSTERMANN

DATA ENGINEERING

THEMENLANDKARTE

Datensystem-Grundlagen	1
Datenverarbeitung „at Scale“	2
Microservices	3
Governance und Sicherheit	4
Verbreitete Cloud-Plattformen und -Dienste	5
Data Ops	6

LEKTION 2

DATENVERARBEITUNG „AT SCALE“

LERNZIELE

- die Unterschiede zwischen Echtzeit-, Batch- und Stream-Prozessierung erklären
- Anwendungsbereiche und fundamentale Konzepte der Batch-Prozessierung erläutern
- MapReduce für Batch-Prozessierungen einsetzen
- Anwendungsbereiche und fundamentale Konzepte der Stream-Prozessierung erklären
- Apache Kafka und Spark Streaming für Stream-Prozessierungen einsetzen

EINFACH ERKLÄREN

1. Beschreiben Sie die **Hauptunterschiede** zwischen einer **Batch-** und einer **Stream-Prozessierung**.
2. Beschreiben Sie in einfachen Worten, wie das **Hadoop Distributed File System (HDFS)** und **MapReduce** im Prinzip funktionieren. Was erlaubt es diesen Systemen, riesige Datenmengen zu verarbeiten?
3. Erklären Sie in einfachen Worten, was die **Hauptunterschiede** zwischen den Streaming-Plattformen **Apache Kafka** und **Spark Streaming** sind.

Batch-Prozessierung

- Schritte eines Batch-Jobs
- Zustände und Entscheidungen eines Batch-Jobs
- HDFS
- MapReduce

Stream-Prozessierungssysteme

- Technologien zur Verarbeitung von Informationsflüssen
- Moderne Stream-Prozessierungssysteme
- Apache Kafka
- Spark Streaming

WIEDERHOLUNG DER KERNPUNKTE DER LEKTION

Batch-Prozessierung

- Schritte eines Batch-Jobs
- Zustände und Entscheidungen eines Batch-Jobs
- HDFS
- MapReduce

Stream-Prozessierungssysteme

- Technologien zur Verarbeitung von Informationsflüssen
- Moderne Stream-Prozessierungssysteme
- Apache Kafka
- Spark Streaming

- **Chunk-orientierte Schritte**
  - zerlegen die Daten in **kleinere Datenpakete (Chunks)**
  - können **parallelisiert** werden, indem **unterschiedliche Chunks** auf verschiedenen Rechnern verarbeitet werden

- **Task-orientierte Schritte**
  - zerlegen die Verarbeitung in mehrere Teilaufgaben
  - können parallelisiert werden, indem unterschiedliche Teilaufgaben auf verschiedenen Rechnern ausgeführt werden

Quelle Text: Oracle, 2017.

TASK-ORIENTIERTE PARALLELISIERUNG

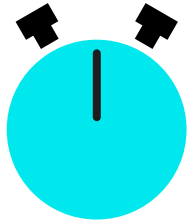
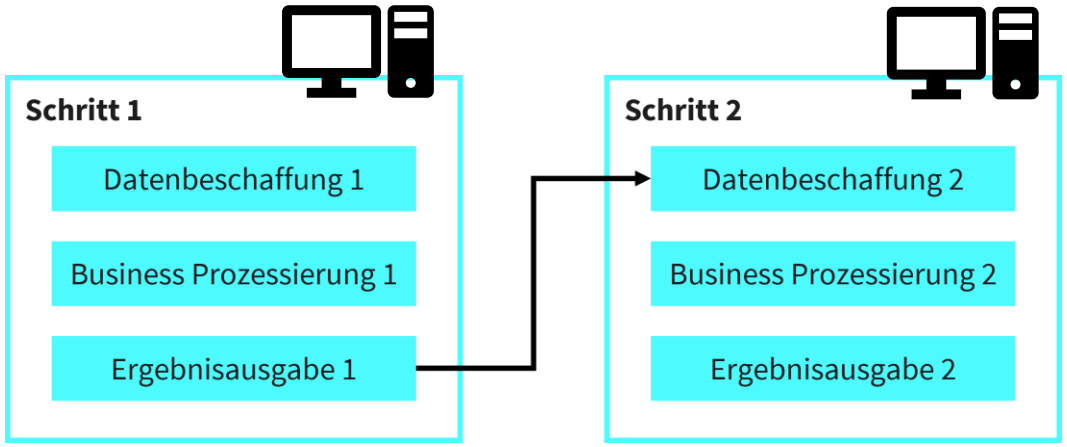


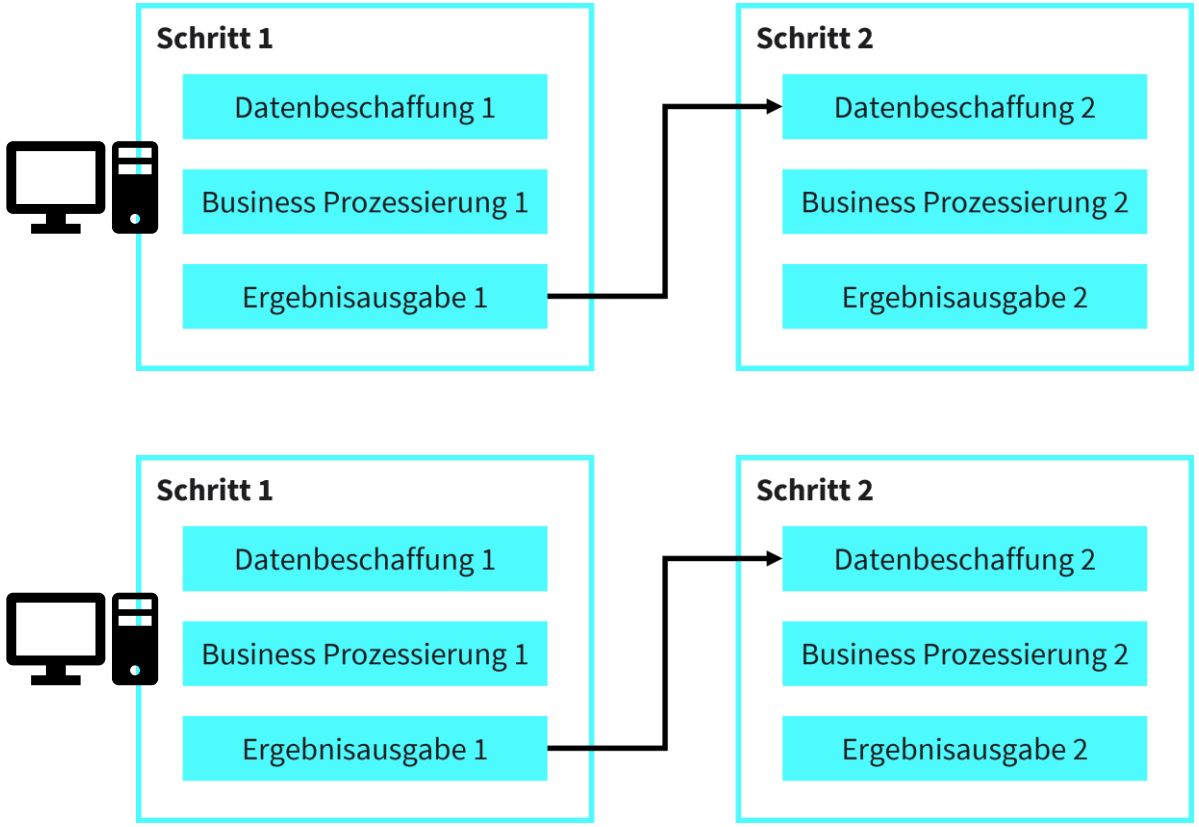
Abb. 3: Batch-Prozessierung auf zwei Maschinen - Sequentiell

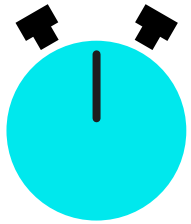


Quelle Text: Abb. 3: Christian Müller-Kett, 2022, in Anlehnung an Oracle, 2017.

CHUNK-ORIENTIERTE PARALLELISIERUNG

Abb. 4: Batch-Prozessierung auf zwei Maschinen – Parallel im engeren Sinne

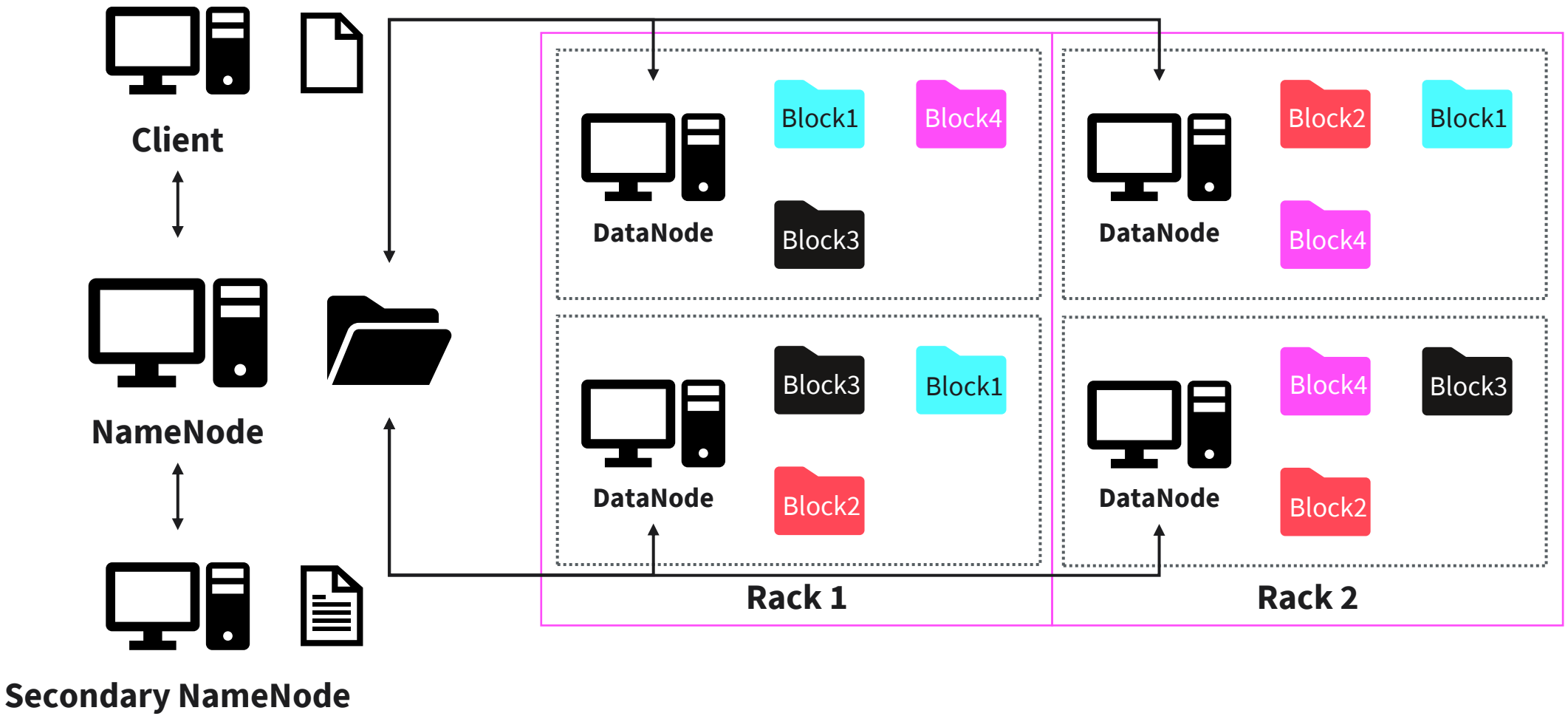




Quelle Text: Abb. 4: *Anmerkung:* Christian Müller-Kett, 2022, in Anlehnung an Oracle, 2017.

HDFS-ARCHITEKTUR

Abb. 5: HDFS-Daemons

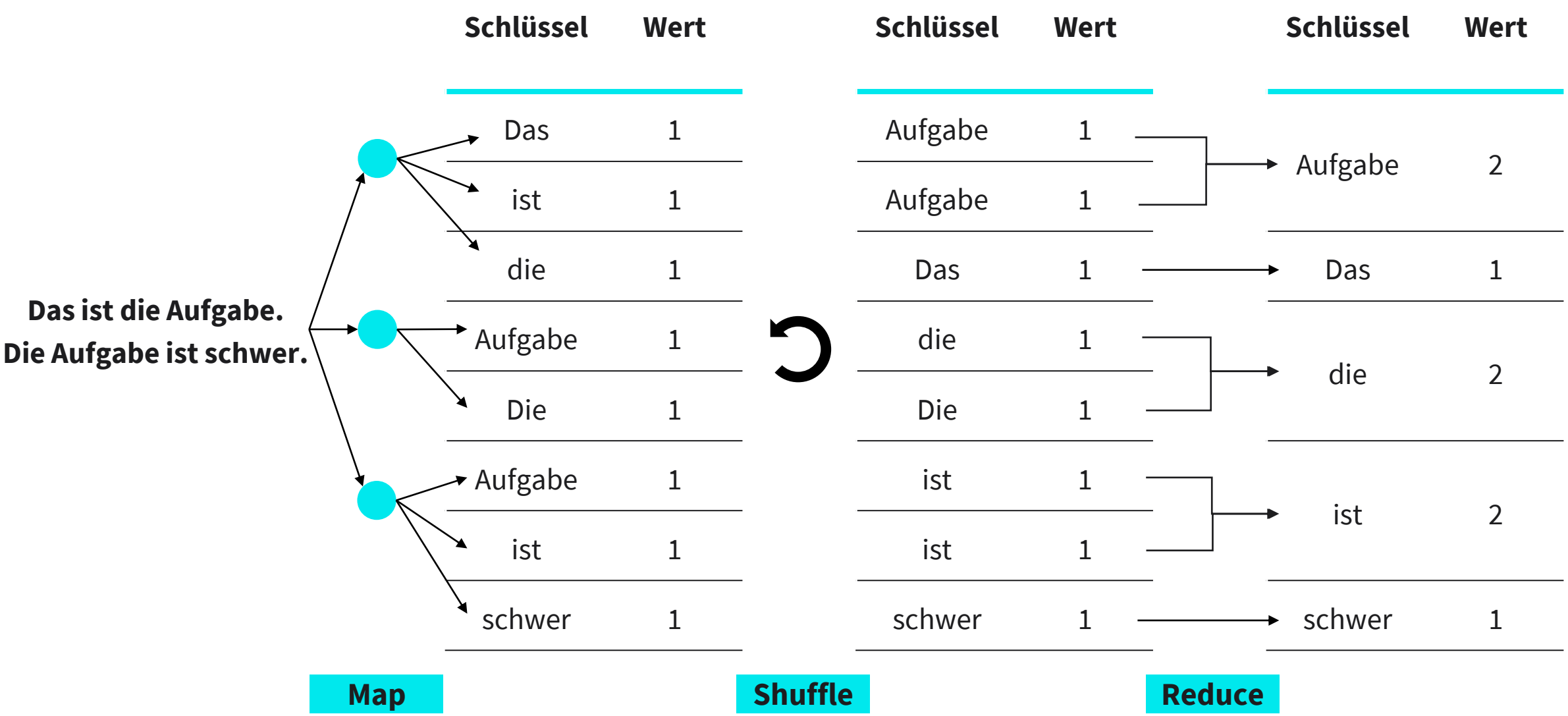


Quelle Text: Abb. 5: *Anmerkung:* Christian Müller-Kett, 2022, in Anlehnung an Oracle, 2017.

- von **Google** zur **Verarbeitung großer Datenmengen** im **HDFS** entwickelt (Dean & Ghemawat, 2008)
- der Code wird auf die **DataNodes**/**Partitionen** gebracht und dort ausgeführt
- im **Mapping-Teil** werden **key-value-Paare** (Schlüssel-Wert-Paare) erzeugt
- diese werden anschließend sortiert (**shuffle**)
- im **Reduce-Teil**, werden **Werte** (values) mit gleichem Schlüssel (key) **aggregiert**

MAPREDUCE

Abb. 6: MapReduce



Quelle Abbildung: Christian Müller-Kett, 2022.

WIEDERHOLUNG DER KERNPUNKTE DER LEKTION

Batch-Prozessierung

- Schritte eines Batch-Jobs
- Zustände und Entscheidungen eines Batch-Jobs
- HDFS
- MapReduce

Stream-Prozessierungssysteme

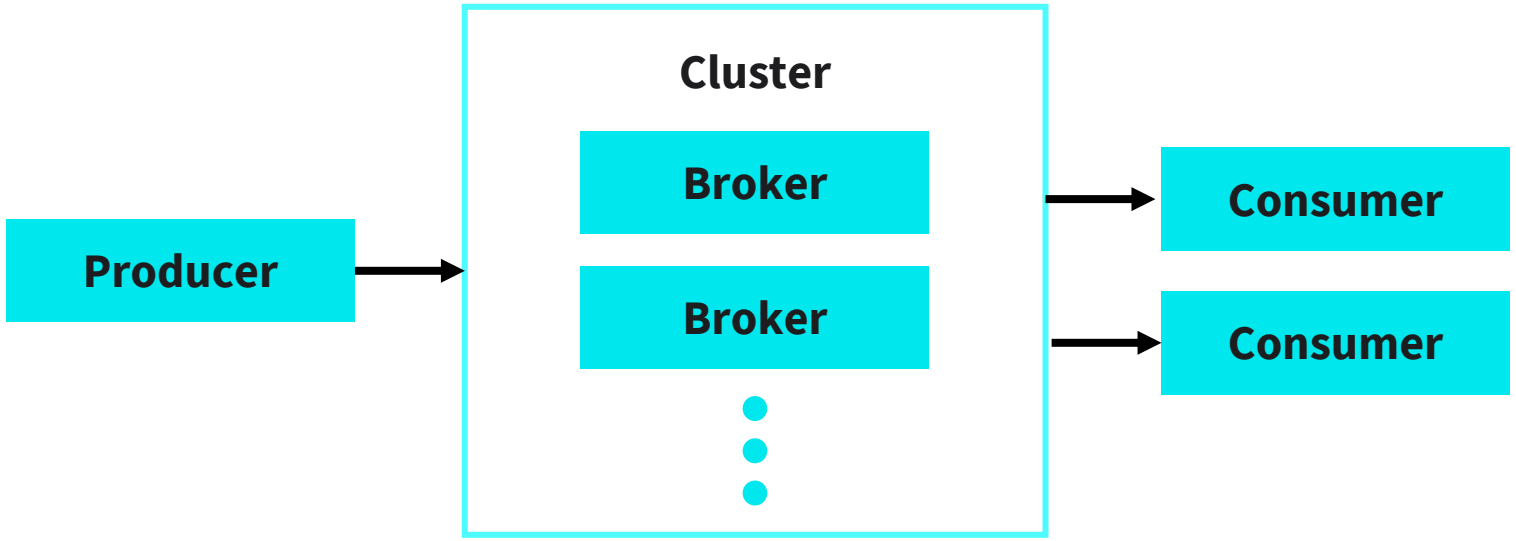
- Technologien zur Verarbeitung von Informationsflüssen
- Moderne Stream-Prozessierungssysteme
- Apache Kafka
- Spark Streaming

- **Aktive Datenbanken**
  - Event-Kondition-Aktion
  - geschlossene/offene aktive Datenbanken
- **Kontinuierliche Abfragesysteme**
  - Continuous Query (CQ)
  - Abfrage-Trigger-Stop
- **Publish-Subscribe-Systeme**
  - Publisher-Broker-Consumer
  - topic-based (themen-basiert) / content-based (inhalts-basiert)
- **Komplexe Event-Verarbeitungssysteme**
  - Kombinationen einfacher Events

- **Vorgänger** moderner Systeme sind im **Umfang der Datenverarbeitung limitiert**
- **Stream-Prozessierungs-Paradigma**
  - Streaming-Daten-Tupel
  - Operatoren
  - Stream-Verbindungen
  - Stream-Prozessierungs-Anwendungen (SPA)
- **Stream-Prozessierungs-System** (SPS)
  - Entwicklungsumgebung
  - Runtime-Umgebung

- ursprünglich als **Message-Queue-System** bei LinkedIn entwickelt
- **unveränderliches** und **verteiltes Logging**

Abb. 7: Apache Kafka High-Level-Konzept

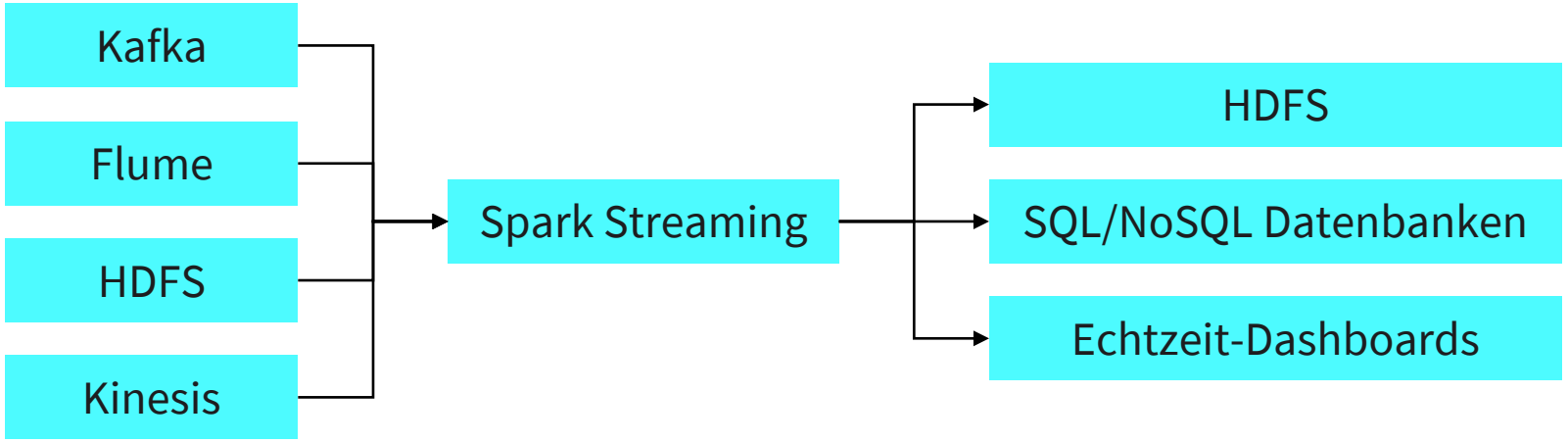


Quelle Abbildung: Christian Müller-Kett, 2022.

Erweiterungsbibliothek für **Apache Spark**

- **fehlertolerante** Stream-Prozessierung
- **skalierbar**, hoher Durchsatz (**Throughput**)
- nahtlos mit Apache **Spark** und dazugehörigen Bibliotheken **integrierbar**
- API **unterstützt** eine Vielzahl von **Datenquellen** und **Datensenken**

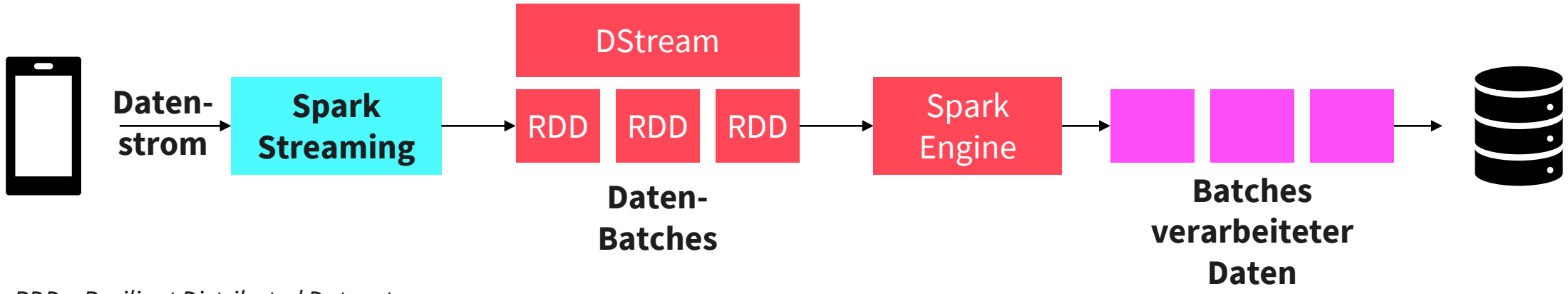
Abb. 9: Spark Streaming – Datenquellen und Datensenken



Quelle Abbildung: Christian Müller-Kett, in Anlehnung an IU – Internationale Hochschule.

SPARK-STREAMING-WORKFLOW

Abb. 10: Spark Streaming Workflow



RDD = Resilient Distributed Dataset
DStream = Discretized Stream

Quelle Abbildung: Christian Müller-Kett, in Anlehnung an IU – Internationale Hochschule.

- **Spark Streaming** ~ Millisekunden bis ein paar Sekunden
- **Kafka** ~ weniger als Millisekunden

Gründe, **Spark Streaming** Kafka **vorzuziehen:**

- **nahtlose Integration** in Spark-Umgebungen
- **reichhaltige APIs**, die es erlauben, eine große Bandbreite unterschiedlicher **Datenquellen** und **Datensenken einzubinden**
- ...

REVIEW LERNZIELE

- die Unterschiede zwischen Echtzeit-, Batch- und Stream-Prozessierung erklären
- Anwendungsbereiche und fundamentale Konzepte der Batch-Prozessierung erläutern
- MapReduce für Batch-Prozessierungen einsetzen
- Anwendungsbereiche und fundamentale Konzepte der Stream-Prozessierung erklären
- Apache Kafka und Spark Streaming für Stream-Prozessierungen einsetzen

EINHEIT 1

TRANSFERAUFGABE

TRANSFERAUFGABE

Ein Start-Up das **nachhaltige Produkte in kleineren Geschäften** vertreibt war in den letzten Jahren sehr erfolgreich. In Folge sollen **weltweit weitere Filialen** eröffnet werden. Als Data Engineer:in sind Sie damit beauftragt, das **Datensystem zu entwerfen**, welches Daten über die **angebotenen Produkte** und **deren Zulieferer** speichert und verarbeitet.

Den Filialverantwortlichen soll **jeden Morgen ein Bericht** darüber zur Verfügung stehen, aus dem ersichtlich wird, **welche Waren** in welcher **Menge** vorliegen und welche Waren bei welchem **Zulieferer** bestellt werden müssen. Außerdem soll es, um **Trends frühzeitig** zu **erkennen**, ein Warnsystem geben welches **im Minutentakt** überwacht, ob es weltweit für einen Artikel eine besonders hohe Nachfrage gibt. Dieses Warnsystem soll mit **möglichst vielen externen Diensten kompatibel** sein, da es in eine größere Systemlandschaft eingebunden werden soll.

Erstellen Sie im Team einen **konzeptionellen Plan** für das **Datensystem**, welches diesen Voraussetzungen gerecht wird. Gehen Sie dabei auf die **Speicherung** und die **Verarbeitung** der Daten ein und machen Sie **konkrete Vorschläge für die einzusetzende Technologie**. Beschreiben Sie **kurz** und in **einfachen Worten**, wie diese am Beispiel dieses konkreten Anwendungsfalls **funktionieren**.

TRANSFERAUFGABE
PRÄSENTATION DER ERGEBNISSE

Bitte stelle deine Ergebnisse vor.
Im Plenum werden die Ergebnisse diskutiert.

1. Welcher Daemon in HDFS ist der Master-Node?
   a) DataNode
   b) JobTracker
   c) NameNode
   d) NodeTracker

2. Wie heißt eine Kategorie oder ein Feed-Name in Kafka, in dem die Nachrichten veröffentlicht werden?
   a) Topic
   b) Tupel
   c) Consumer
   d) Producer

3. Was definiert die Struktur von Daten-Tupeln?
   a) Schema
   b) Tabelle
   c) Daten-Tupel
   d) Datenstrom

QUELLENVERZEICHNIS

Dean, J. & Ghemawat, S. (2008). MapReduce. *Communications of the ACM, 51*(1), 107–113. https://doi.org/10.1145/1327452.1327492

Oracle. (2017). *Introduction to Batch Processing*. Java Platform, Enterprise Edition (Java EE) 8 - The Java EE Tutorial. https://javaee.github.io/tutorial/batch-processing001.html#BABFJBAH