

Wann nutze ich?



kafka

Heutige Fragen:

- 1. Welche IT-Anforderungen kann ich mit Apache Kafka gut lösen?**
- 2. Und welche lieber nicht?**


Apache Kafka Use-Cases

#1: Data Streaming



Was ist Data Streaming?

 Viele Daten (wirklich viele Daten, mehrere 10.000 kleine Datensätze pro Sekunde) die zeitnah transportiert werden müssen.

 **Wichtig:** Data Streaming **nicht** mit Video Streaming verwechseln! Kafka ist **nicht** dafür ausgelegt unstrukturierte Daten, wie Medien-Inhalte, zu transportieren.

Apache Kafka Use-Cases

#1: Data Streaming



Wann brauche ich Data Streaming?



Logistic & Automotive

Tracing der Geometriedaten



Industrie 4.0 & IoT

Sensoren Daten



E-Commerce

Tracing der Benutzeraktivitäten
Marketing, Sales und Business Analyse
Börsenhandel



IT-Monitoring & Reporting

Metriken und Log-Informationen



IT-Sicherheit

Betrugserkennung

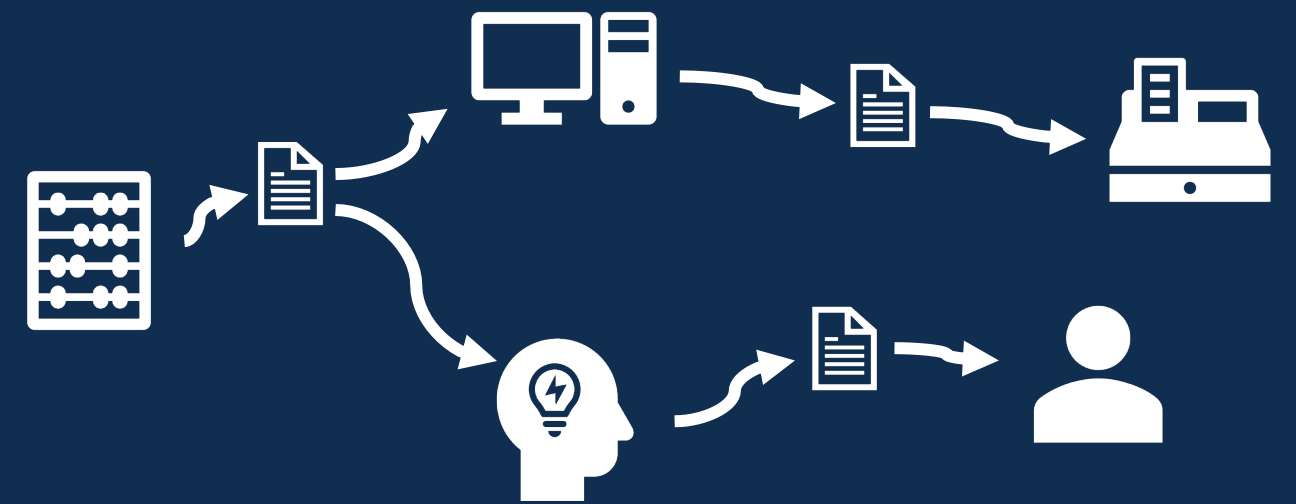


Synchronisation der Daten

DB-Migration, DWH, ...

Apache Kafka Use-Cases

#2: Event-Driven Architecture (EDA)

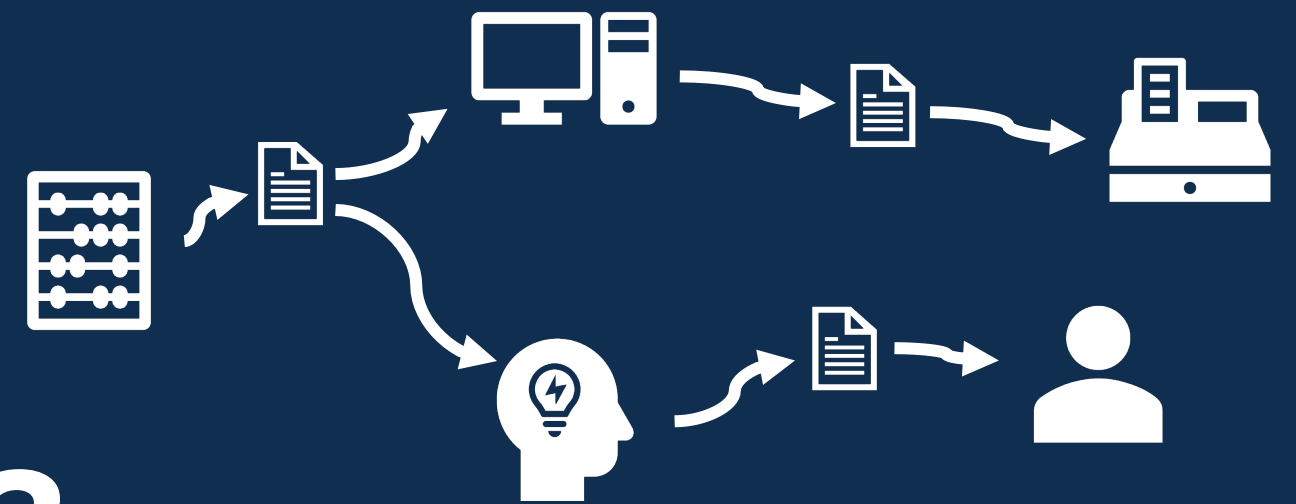


Was ist EDA?

- Bei der Event-Driven Architecture erfolgt die Kommunikation zwischen den IT-Komponenten asynchron und ereignisbasiert nach dem Publish-Subscribe Prinzip.
- Das publizierte Ereignis (Event) ist dabei ein vollständiger Datensatz, angereichert mit den Statusinformationen aus der unmittelbar vor der Publikation erfolgten Aktion.
- **Wichtig:** EDA setzt auf Service-Choreografie anstatt der Service-Orchestrierung.

Apache Kafka Use-Cases

#2: Event-Driven Architecture (EDA)



Wann brauche ich EDA?

- 👥 Ich habe mehrere Entwicklungsteams und möchte, dass sie stark **entkoppelt** interagieren können.
- 🐰 Ich möchte **zügige Integration** der Consumer auf bestehende Pub-Sub-Topics und keine Applikation-CRs für REST-APIs bei jeder Anbindung.
- 🌱 Ich möchte hochverfügbare und skalierbare Komponenten die bedarfsabhängig auch heruntergefahren werden können (scale-down-to-zero).
- 📄 Mehr über die EDA gibt es in einer meiner Folgeposts.

Apache Kafka Use-Cases

#3: Data-Centric Architecture (DCA)



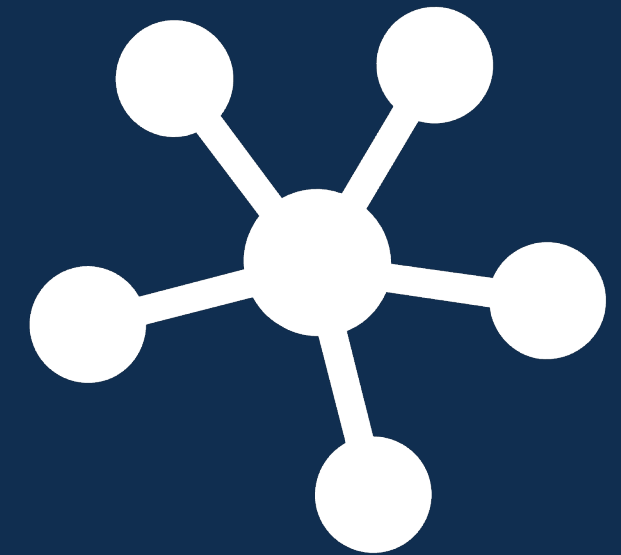
Was ist DCA?

❖ Ziel der datenzentrischen Architektur ist datengetriebene Unternehmen oder Organisationen einfach zu ermöglichen. Dabei sind die Daten nicht hinter den Anwendungen „versteckt“ und schwer erreichbar sondern als Rohstoff der Zukunft zentral zur Verfügung gestellt.

❗ **Wichtig:** Bei der DCA verbleibt die Datenhoheit in den fachlichen Bereichen, es werden lediglich die fachlich relevanten Daten über eine zentrale Plattform bereitgestellt.

Apache Kafka Use-Cases

#3: Data-Centric Architecture (DCA)



Wann brauche ich DCA?

- III Ich möchte **Daten** in meinem Unternehmen ohne Umstände (und Kosten) in neuen Anwendungsfällen direkt nutzen können. Im anwendungszentrischen Ansatz sind Daten nur über die Anwendung erreichbar, was die **Zugänglichkeit** erschwert (das Pain Point #1 in Großunternehmen).
- 🌐 Kafka mag nicht ideal für Kleinunternehmen sein, doch eine datenzentrische Architektur ist essenziell für jedes Unternehmen, das **datengetriebene Geschäftsmodelle** realisieren möchte.
- 📄 Mehr über die DCA gibt es in einer meiner Folgeposts.

Apache Kafka

Was kann Kafka alles?



Event Storage

Events können auf dem Topic über längeren Zeitraum gespeichert bleiben.



Compacted Topic

Stellt sicher das ein Event anhand des fachlichen Schlüssels nur 1x auf dem Topic vorkommt.



At-least-Once & Exactly-Once (> v3.0.0)

Durch den Transaction- & Offset-Commit stellt Kafka sicher, dass ein Event erst nach Commit genau 1x konsumiert wird.



Sehr hoher Datendurchsatz

Events werden on-the-fly in das Topic-Log geschrieben und an Consumern weitergeleitet.



Skalierbarkeit & Verfügbarkeit

In einem Cluster lassen sich Broker hinzufügen. Events sind durch Repliken gesichert.



Technologieagnostik

Anbindung an Kafka ist von vielen Client-Bibliotheken in vielen Sprachen implementiert.

Apache Kafka

Wann nutze ich Kafka nicht und was nutze ich stattdessen?



Einfaches Messaging

RabittMQ, Active MQ, IBM MQ, ...



IoT an der Quelle

MQTT



Video Streaming, Media Dateien, Gaming

??? – gute Frage, schreibts mir in die Kommentare



Klassische Datenbank

MySQL, PostgreSQL, Oracle, MS SQL, ...



NoSQL DB

MongoDB, Graph-DB, Neo4J, Cassandra, ...

Apache Kafka

Fazit



Es gibt noch weitere performante Data Streaming Plattformen, jedoch hat Kafka die führende Rolle auf dem Markt und ist dementsprechend gut in Anwendungen und Frameworks integriert
→ Siehe auch: Amazon Kinesis, Apache Pulsar, Apache Samza, ...



Kafka ist aufgrund der einfachen Integration in den Frameworks in der Softwareentwicklung sehr beliebt.



Kafka lässt sich durch Code-Patterns wie Outboxing auch im transaktionalen Kontext sicher einsetzen.



Mit zusätzlichen Tools (Schema Registry, Observability, ...) und Skalierbarkeit durch Cluster-Fähigkeit, ist Apache Kafka längst in dem unternehmenskritischen Anwendungsfällen angekommen.



Wichtig: Der Betrieb von Apache Kafka ist nicht trivial und benötigt Know-how und Ressourcen.

Falls Du weitere Posts zu den Themen wie:

- Data Streaming
- Event-Driven Architecture
- Data-Centric Architecture
- ...

nicht verpassen willst ... vernetze Dich gerne mit Mir!



Azmir Abdi