

Einführung in Data Science – Block1

Big Data und NoSQL für die Datenanalyse



Programm

Thema	Form	Zeit
Organisatorisches		9:45 – 10:00
Besprechung der Vorbereitung	Diskussion	10:00 – 10:30
Big Data und NoSQL für die Datenanalyse	Vorlesung	10:30 – 11:00
Pause		11:00 – 11:15
Data Science und Ethik	Diskussion	11:15 – 11:30
Analyse von Social Media Daten	Vorlesung	11:30 – 11:45
Workshop	Workshop	11:45 – 12:45
Wissenschaftliche Arbeit		12:45 – 13:00

Organisatorisches

- Installation der Arbeitsumgebung
- Bitte Semesterarbeit pünktlich abgeben (eine Woche vor der nächsten Präsenz)!

Organisatorisches

- Semesterarbeit 50%
- Nach den Blöcken 1, 2, 3 und 4
 - Implementierung von Datenanalysen gemäss Moodle
 - Ausführliche Interpretation der Ergebnisse
 - Zusätzliche Recherche von wissenschaftlichem Material
 - Darstellung in Form von wissenschaftlichen Essays in deutscher Sprache mit korrekten wissenschaftlichen Literaturnachweisen
- Modulprüfung 50%
- Umfasst den Stoff des gesamten Moduls
 - welche in den PVAs behandelt werden
 - zu welchen es Textverständnis-Fragen gibt
 - zu welchen Übungen gelöst wurden
 - Zugelassene Hilfsmittel: 10 Seiten Notizen

Besprechung der Vorbereitung

Verständnisfragen 1

- Welche inwiefern kann Twitter als Netzwerk (Graph) betrachtet werden? Was sind Knoten, was sind Kanten.
- Welche Eigenschaften hat der Twitter Graph?
- Inwiefern ist Twitter ein Graph? Welche Eigenschaften hat der Graph?
- Wie sind die folgenden Begriffe definiert: Taxonomie und Folksonomie und Ontologie.
- Wie ist Lexical Diversity definiert? Was sagt diese Zahl aus?

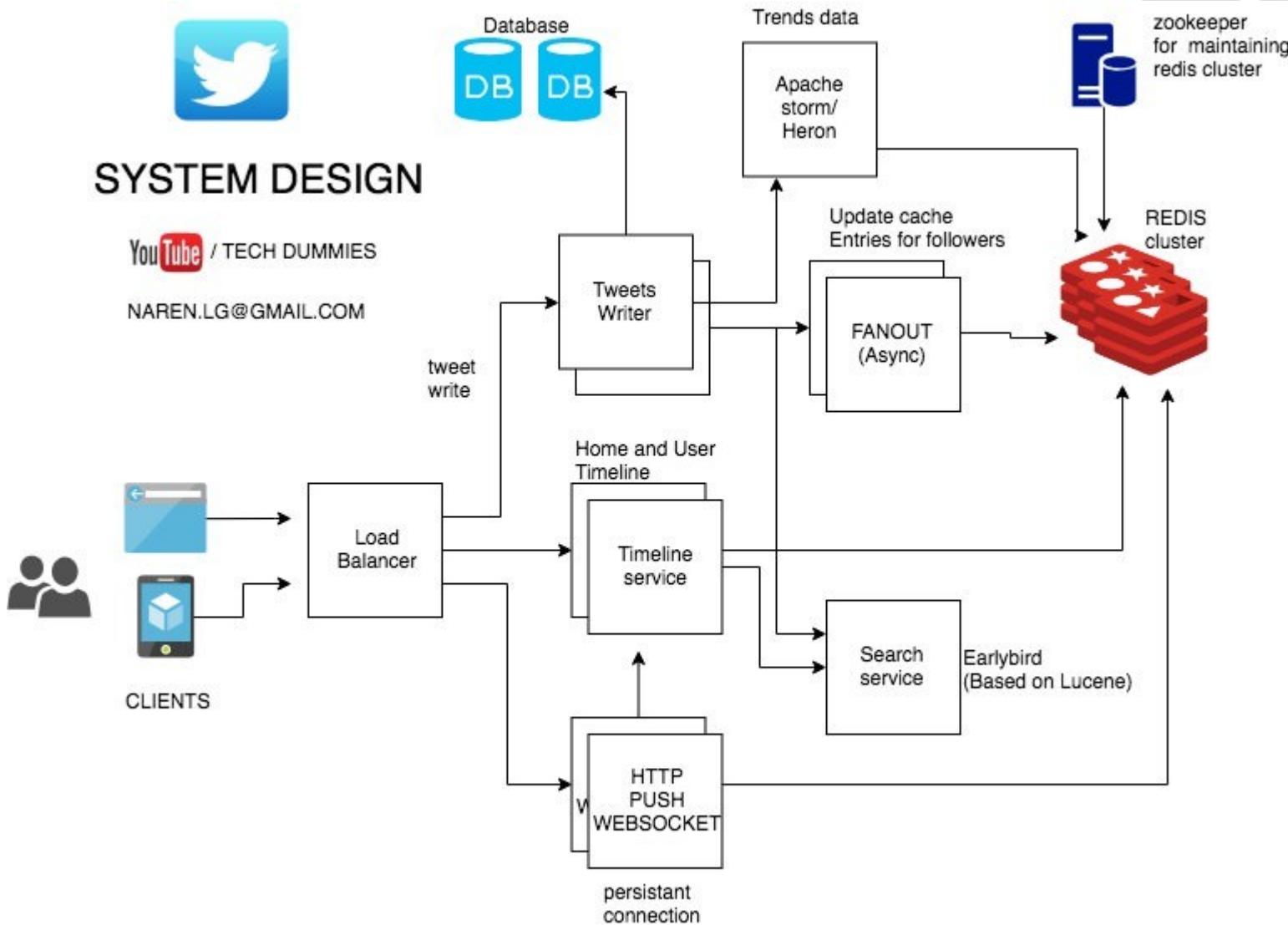
Verständnisfragen 2

- Welche Datenmengen verarbeitet Twitter? (Recherchieren Sie aktuelle zahlen)
Anzahl Tweets pro Minute (rsp. Sekunde)
Vorhandenes Datenvolumen
Anzahl Nodes im Big Data Cluster
- In der vorgestellten Systemarchitektur werden die folgenden Systemkomponenten erwähnt:
Redis
Storm / Heron
Kafka
Zookeeper
Search Service (Lucene)
- Recherchieren Sie zu diesen Systemen die folgenden Angaben:
Wer stellt das System her?
Wie beschreibt der Hersteller das System
Wer (ausser Twitter) setzt das System ein
Welche Datenmengen kann das System verarbeiten?
- Was ist "Eventual Consistency"? In welchem Zusammenhang steht der Begriff mit den Systemen aus der Liste oben?

SYSTEM DESIGN

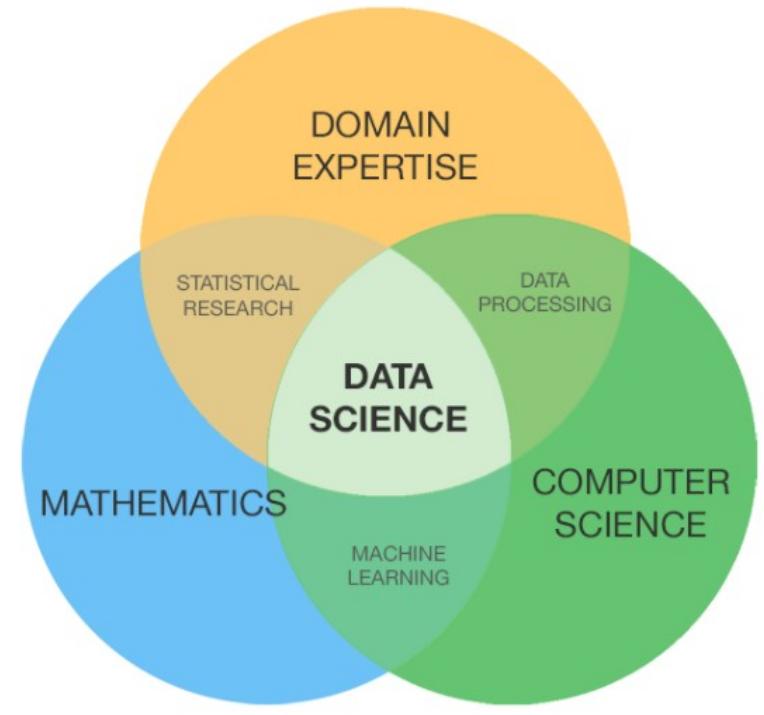
YouTube / TECH DUMMIES

NAREN.LG@GMAIL.COM



Data Science

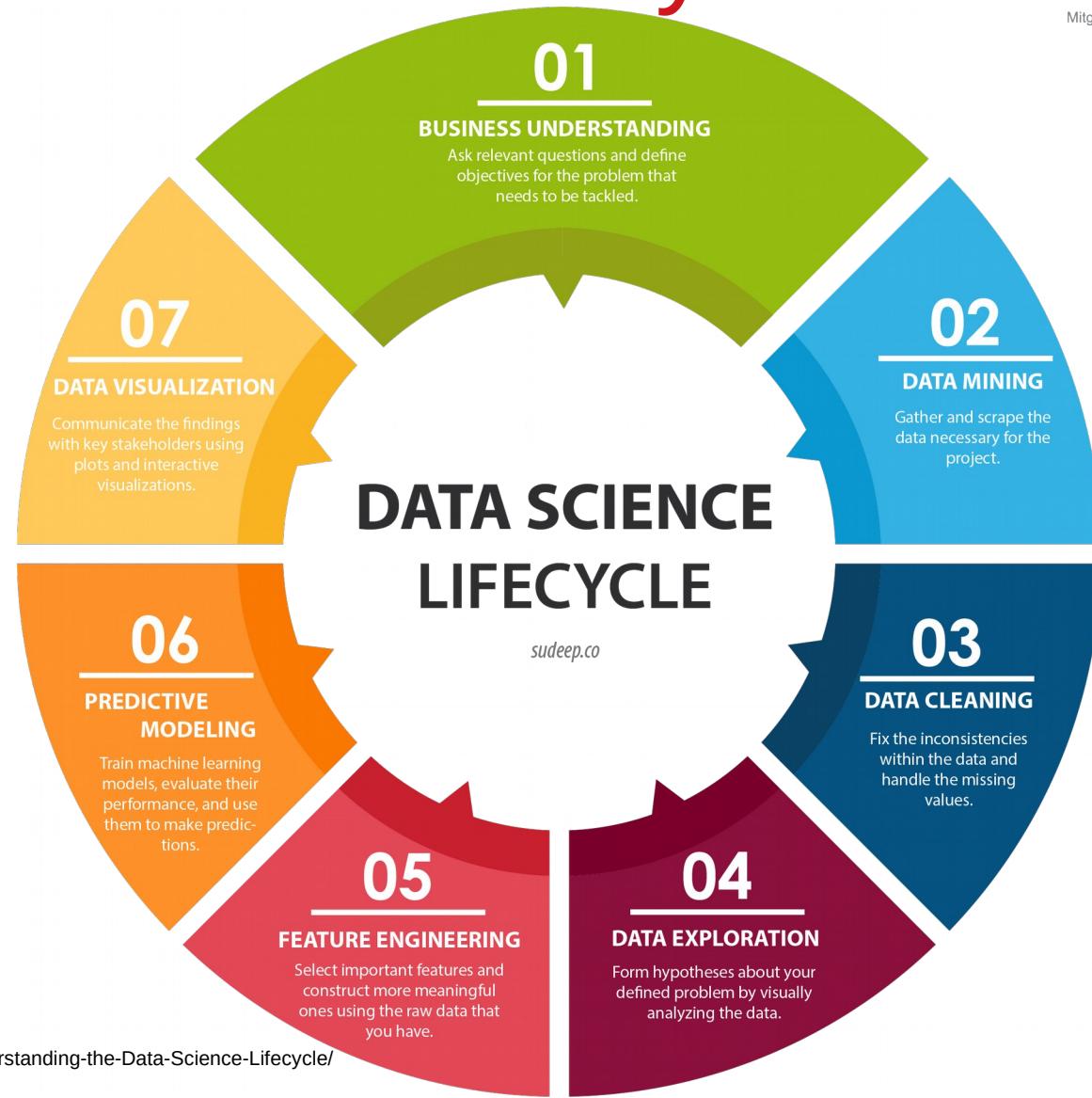
- Data Science beschäftigt sich mit einer zweckorientierten Datenanalyse und der systematischen Generierung von Entscheidungshilfen und -grundlagen.



<https://towardsdatascience.com/data-science-interview-guide-4ee9f5dc778>

VIDEO

Data Science Lifecycle



Data Science Technologien

- Block 1 → Big Data und NoSQL
- Block 2 → Netzwerk Analyse
- Block 3 → Supervised Learning
- Block 4 → Unsupervised Learning
- Block 5 → Information Retrieval

Big Data - Definition

Big Data bezeichnet Datenmengen, die

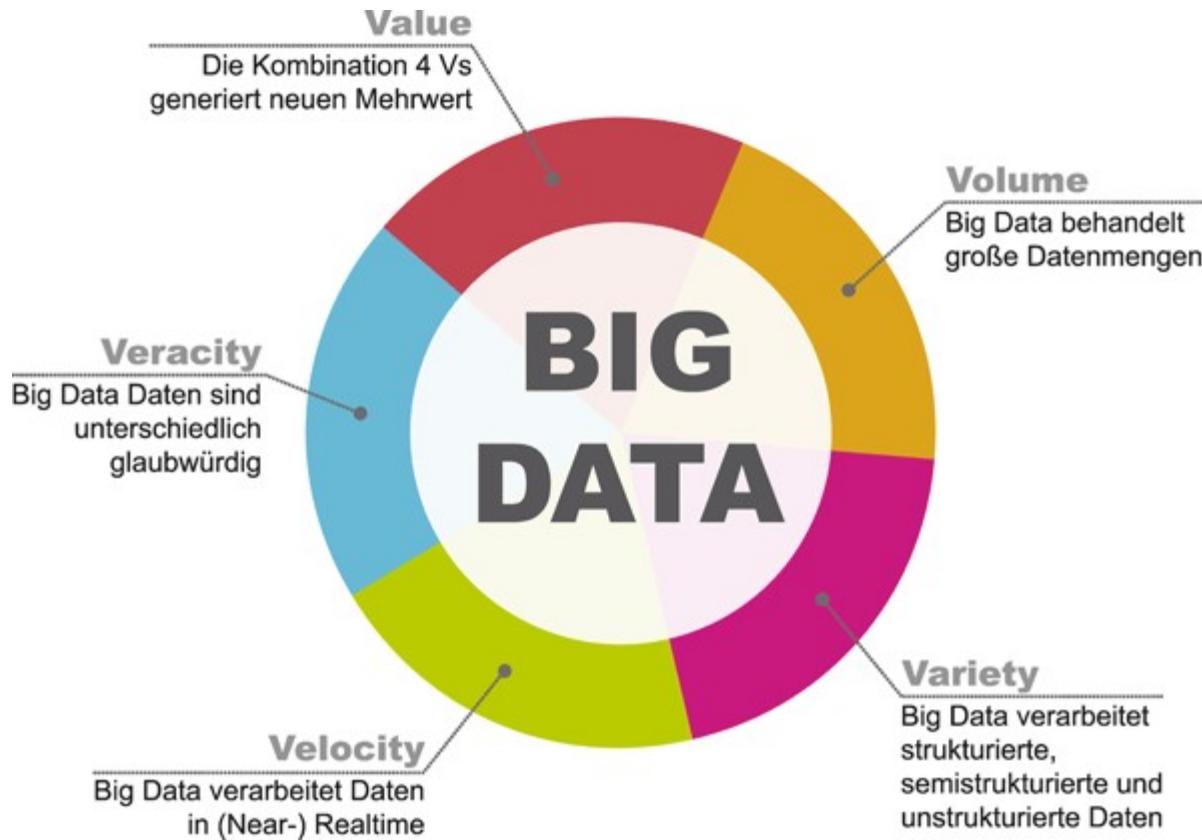
- zu gross sind (Masse oder **Volume**), oder/und
- zu komplex sind (Vielfalt oder **Variety**), oder/und
- sich zu schnell ändern (Geschwindigkeit oder **Velocity**)

in Anlehnung an http://de.wikipedia.org/wiki/Big_Data (2014)



Quelle: <https://blog.unbelievable-machine.com/was-ist-big-data-definition-f%C3%BCnf-v>

Big Data Dimensionen



Quelle: https://link.springer.com/chapter/10.1007/978-3-658-02541-0_9

Big Data Ecosystem



Big Data Ecosystem

- Eine Big Data-Environment muss ermöglichen, Daten zu speichern, zu verarbeiten, zu analysieren und zu visualisieren.



Verarbeiten, speichern und analysieren.
Vorher: SQL
Datenbanken
Jetzt:
Hadoop
NoSQL



Analytics Platforms

- Visualization Platforms
- Business Intelligence Platforms
- Machine Learning



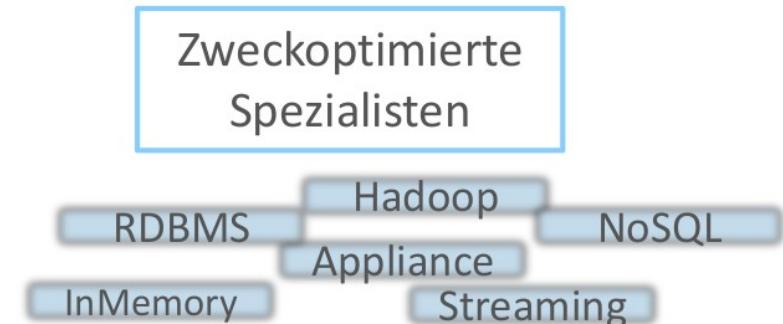
Von Unternehmen und Startups entwickelt
Beispiel: Mintlabs: 3D Gehirnscans für Diagnose

Systemarchitektur im Wandel

Gestern und heute

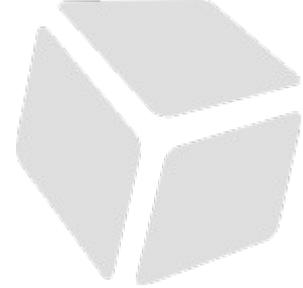


Heute und morgen





Datenbanken



- SQL kann: Abfragen ausführen, Daten aus einer Datenbank abrufen, Datensätze in einer Datenbank einfügen, Datensätze insert, update und delete, eine neue Datenbank erstellen, Tabellen in einer Datenbank erstellen, kann Speicherprozeduren in einer Datenbank erstellen, kann views erstellen, Permission einstellen
- SQL benutzt ein RDBMS Datenbankprogramm (MS Access, SQL Server, MySQL), eine serverseitige Skriptsprache (PHP) und HTML/CSS
- Effizient für kleine Transaktionen, oder grosse Batch-Transaktionen mit seltenen Schreibzugriffen

RDBMS (Relational Database Management System)



SQl Server Home Create View Favorites Grid Table data

AdventureWorks2014: localhost\SQLEXPRESS - SQL Database Studio 2.2.0 PRO

Refresh data Switch connection Search Design query Natural Sort

Columns Default Quick export Save to local storage Other

New Row * Save changes * Delete Row * Revert row changes Change set

Column manager SQL WHERE Filter Offline rows 1000 Use offline

Connections Map6.map *Query1.sql dbo.outcodepostcodes Person.Address Person.AddressType dbo.AWBuildVersion

Tables (74) - AdventureWorks2014: localhost\SQLEXPRESS

Search

Schema Name

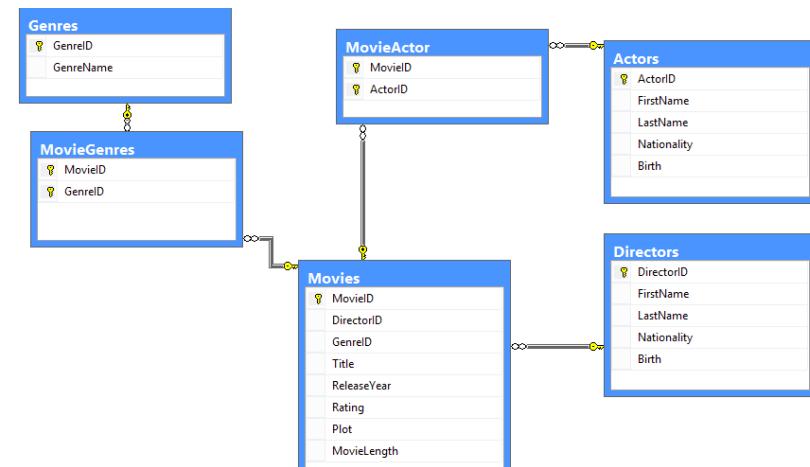
- Production
- WorkOrderRouting
- Production
- WorkOrder
- Purchasing
- Vendor
- Production
- UnitMeasure
- Production
- TransactionHistoryArch
- Production
- TransactionHistory
- Sales
- Store
- Person
- Sales
- SpecialOfferProduct
- Sales
- SpecialOffer
- Sales
- ShoppingCartItem
- Purchasing
- ShipMethod
- HumanResources
- Shift
- Production
- ScrapReason
- Sales
- SalesTerritoryHistory
- Sales
- SalesTerritory
- Sales
- SalesReason
- Sales
- SalesRate
- Sales
- SalesPerson
- Sales
- SalesPersonQuotaHistory
- Sales
- SalesTerritory
- Sales
- SalesOrderHeaderSalesR
- Sales
- SalesOrderHeader
- Sales
- SalesOrderDetail
- dbo
- RouteCopter
- Purchasing
- PurchaseOrderHeader
- Purchasing
- PurchaseOrderDetail

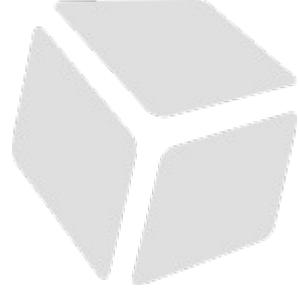
AddressID AddressLine1 AddressLine2 City StateProvinceID PostalCode SpatialLocation

AddressID	AddressLine1	AddressLine2	City	StateProvinceID	PostalCode	SpatialLocation
1	1970 Napa Ct.		Thehell	79 WA	98011	POINT (-122.64644615406 4)
2	9833 Mt. Dias Bv.		Thehell	79 WA	98011	POINT (-122.25018528911 4)
3	7484 Roundtree Drive		Thehell	79 WA	98011	POINT (-122.274625789912 4)
4	9539 Glenside Dr.		Thehell	79 WA	98011	POINT (-122.335726442416 4)
5	1226 Shoe St.		Thehell	79 WA	98011	POINT (-122.09123832402 4)
6	1399 Firestone Drive		Thehell	79 WA	98011	POINT (-122.360166703417 4)
7	5672 Hale Dr.		Thehell	79 WA	98011	POINT (-122.335726442416 4)
8	6387 Scenic Avenue		Thehell	79 WA	98011	POINT (-122.37236633918 4)
9	8713 Yosemite Ct.		Thehell	79 WA	98011	POINT (-122.189084676407 4)
10	250 Race Court		Bc	Has Not Empty Value	98011	POINT (-122.042443310399 4)
11	1318 Lasalle Street		Bc	Contains..	98011	POINT (-122.225745267909 4)
12	5415 San Gabriel Dr.		Bc	Does Not Contain..	98011	POINT (-122.335726442416 4)
13	9265 La Pal		Bc	Begins With..	98011	POINT (-122.054634049 47.8)
14	8157 W. Book		Bc	Does Not Begin With..	98011	POINT (-122.16444615406 4)
15	4912 La Vuelta		Bc	Ends With..	98011	POINT (-122.323506311915 4)
16	40 Ellis St.		Bc	Does Not End With..	98011	POINT (-122.16444615406 4)
17	6696 Anchor Drive		Bc	Custom Filter..	98011	POINT (-122.25018528911 4)
18	1873 Lion Circle		Bc		98011	POINT (-122.0668385714 47.7)
19	3148 Rose Street		Bothell	79 WA	98011	POINT (-122.347946572916 4)
20	6672 Thornwood Dr.		Bothell	79 WA	98011	POINT (-122.14020455405 4)
21	5747 Shirley Drive		Bothell	79 WA	98011	POINT (-122.0302317989 4)
22	636 Vine Hill Way		Portland	58 OR	97205	POINT (-122.6223011785 45.5)
23	6657 Sand Point Lane	(NULL)	Seattle	79 WA	98104	POINT (-122.373607213 47.72)

Tables (7) Views (2) Functions 19614 rows Ready

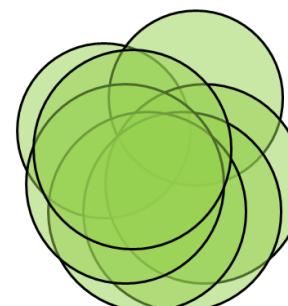
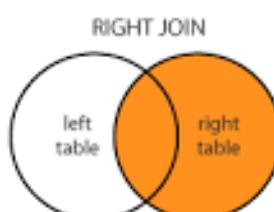
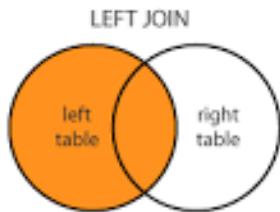
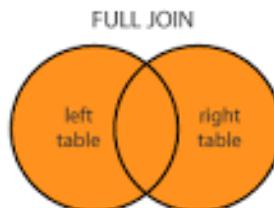
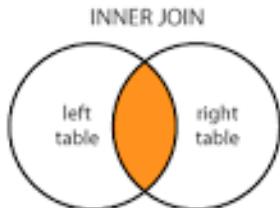
Connected localhost\SQLEXPRESS AdventureWorks2014 (Windows) Classical layout 74 20 10 10 SQL 0 0 0 0





Warum ist SQL wichtig?

- Es ist reif und gut verstanden
- Es kann mit relationalen, hierarchischen und strukturierten Datenmodellen umgehen
- Es kann komplexe Transaktionen behandeln
- Populäre SQL Datenbanken: SQL Server, Oracle, MySQL, SQLite, PostgreSQL
- Grenzen: Skalierbarkeit, Distribution, Umgang mit semi-strukturierten Daten

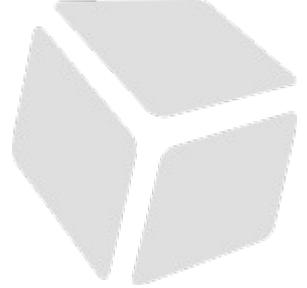


```
SELECT <select list>
FROM TableAA
LEFT OUTER JOIN TableBB ON A.pid = B.pid
JOIN TableCC on C.p_id = A.p_id
AND C.aid = A.aid
AND A.a_date = C.a_date
AND C.d_id = 'NULL'
JOIN TableDD on D.p_id = A.p_id
AND D.eid = A.eid
JOIN TableEE on E.p_id = A.p_id
AND E.rid = C.rid
LEFT OUTER JOIN TableFF on E.p_id = F.p_id
LEFT OUTER JOIN TableGG on F.ps2id = G1.mlid
AND G1.ml_type = 'some_string'
LEFT OUTER JOIN TableGG on F.ps2id = G2.mlid
AND G2.ml_type = 'some_string'
LEFT OUTER JOIN TableHH on A.c_by = H.uid
AND H.lid = 'some_other_string'
```



NoSQL

- Non SQL, Not Only SQL, Non Relational
- Sie werden manchmal als Cloud-Datenbanken, non-relational Datenbanken oder Big Data Datenbanken bezeichnet
- Eine NoSQL Datenbank ist ein non-relational und weitgehend verteiltes Datenbanksystem, das eine schnelle, ad-hoc Organisation und Analyse von extrem hochvolumigen, verschiedenen Datentypen ermöglicht
- Es braucht keine festgelegten Tabellenschema und versucht “joins” zu vermeiden
- Entwickelt, weil relationale Datenbanken unter Leistungsprobleme bei dataintensiven Applikationen leiden (z.B. Indexierung grosser Dokumentmengen, Webseiten mit hoher Lastaufkommen, Streaming-Media Applikationen)



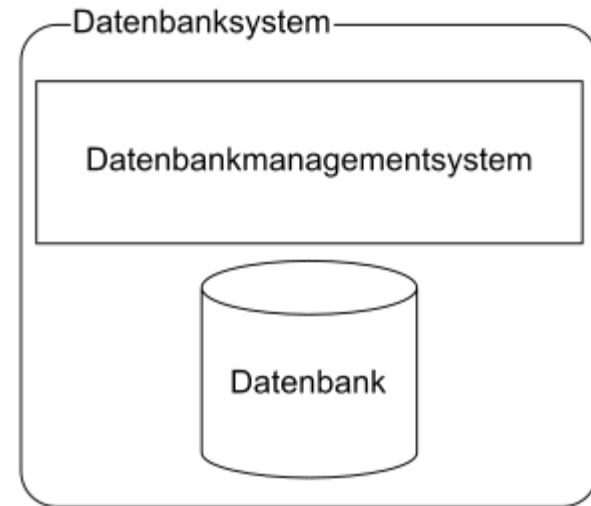
Warum NoSQL?

- Advent von Big Data
- Kontinuierliche Datenverfügbarkeit
- Moderne Transaktionsmöglichkeiten
- Flexible Datamodelle
- Bessere Architektur
- Analytics und Business Intelligence
- Populäre NoSQL Datenbanken: MongoDB, Redis, Cassandra, Couch DB, HBase

Datenbanksysteme - DBS



- **DBS: System zur dauerhaften Speicherung und Verwaltung grosser strukturierter Datenmengen**
- **Datenbank (DB) + Datenbankverwaltungssystem (DBMS)**
- **Datenbankmodell: relational oder nicht relational**
- Motivation für Einsatz eines DBS, typische Probleme bei Informationverarbeitung ohne DBMS
 - Redundanz und Inconsistenz
 - Beschränkte Zugriffsmöglichkeiten
 - Verlust von Daten
 - Sicherheitsprobleme
 - Hohe Entwicklungskosten
 - Änderungen im Informationsbedarf
 - Leistungsanforderungen
- Beispiel: Banken und Versicherungen arbeiten mit Datenbanksystemen. Im Datenbanksystem sind alle Konteninformationen und Buchungen strukturiert abgelegt. In diesem Einsatzumfeld haben Datenschutz und Datensicherheit höchste Priorität. Diese Datenbanksysteme werden zum Tagesgeschäft (OLTP) und periodisch für Massendrucksachen, Analysen und ähnliches verwendet (OLAP).



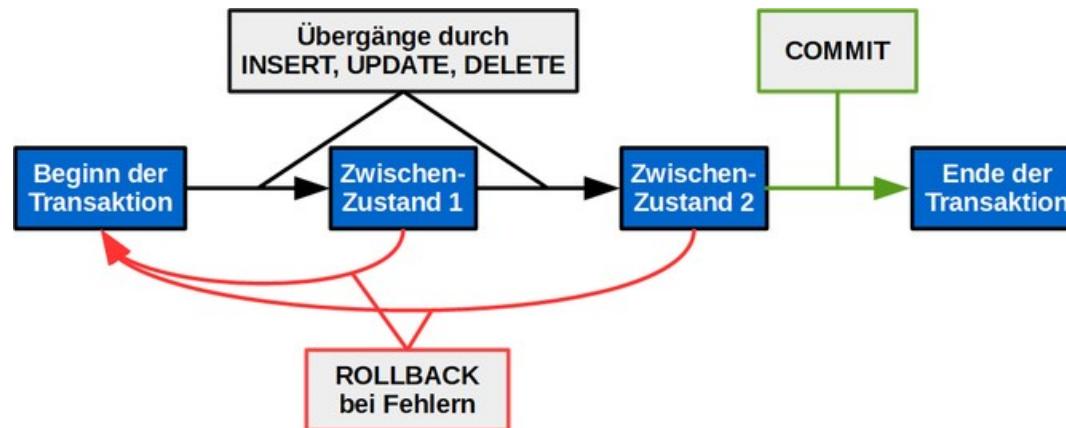
DBMS

MySQL, PostgreSQL, MongoDB, MariaDB, Microsoft SQL Server, Oracle, Sybase, SAP HANA, MemSQL, SQLite, IBM DB2.

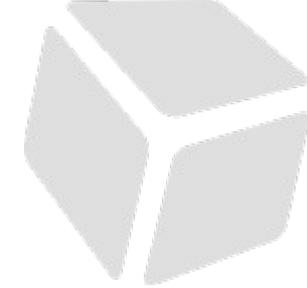


Transaktionen

- Ein wichtiger Teil der Datensicherheit ist das Transaktionskonzept.
- Die Datenbankverarbeitung erfolgt durch Transaktionen, welche aus einer oder mehreren Operationen bestehen.
- Aus logischer Sicht ist eine Transaktion eine Folge von Datenbankoperationen (INSERT, UPDATE, DELETE, ...) die als atomare Einheit ausgeführt werden.
- Sie hinterlassen den Datenbestand nach fehlerfreier und vollständiger Ausführung in einem konsistenten Zustand.
- Die Transaktion wird entweder fehlerfrei oder gar nicht ausgeführt.



ACID - Eigenschaften



- Bei einer Transaktion muss das Transaktionssystem die ACID-Eigenschaften garantieren und beherrschen.
- Diese Eigenschaften werden als ACID Regel bezeichnet. Sie ist ein zentraler Grundsatz der meisten klassischen relationalen DBMS (RDBMS).

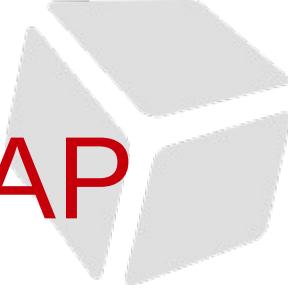
A	Atomarität (<i>atomicity</i>)
C	Konsistenz (<i>consistency</i>)
I	Isolation (<i>isolation</i>)
D	Dauerhaftigkeit (<i>durability</i>)

- 1) Atomarität (Alles oder nichts):** Eine Transaktion wird vollständig oder gar nicht ausgeführt.
- 2) Konsistenz:** Nach einer Transaktion muss der Datenbestand wieder konsistent ohne Anomalien oder Redundanzen.
- 3) Isolation:** Gleichzeitige Transaktionen dürfen sich gegenseitig nicht beeinflussen.
- 4) Dauerhaftigkeit:** Die Dauerhaftigkeit von erfolgreich beendeten Transaktionen wird garantiert.

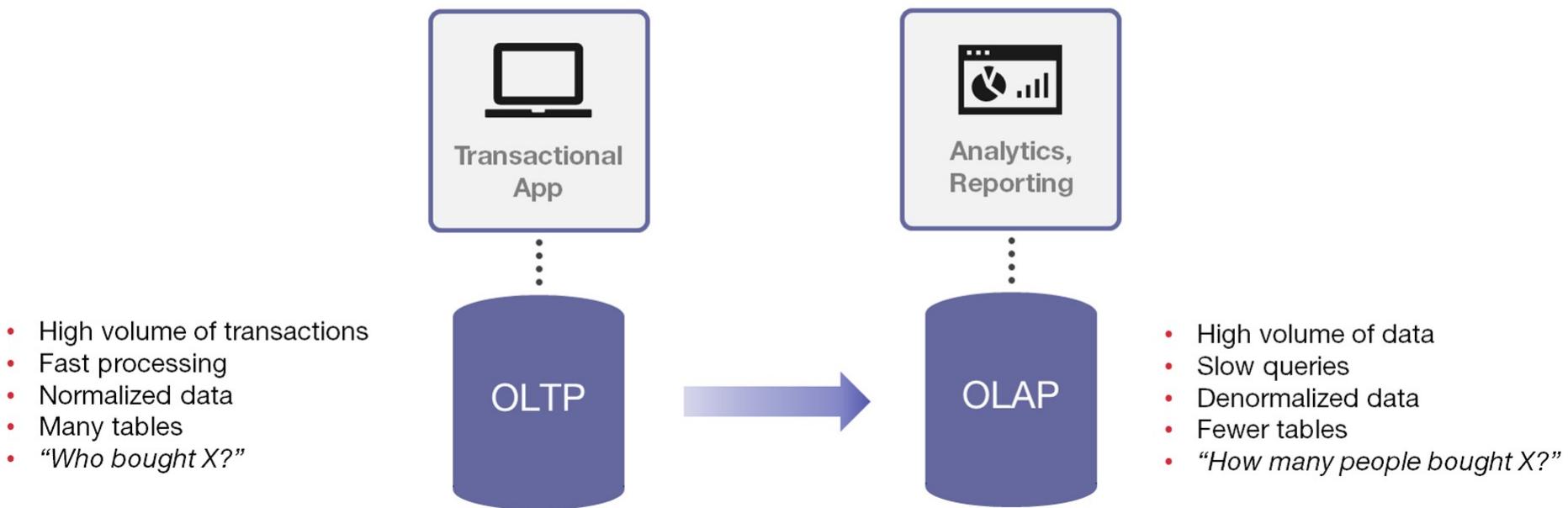
Transaktion nach dem ACID-Prinzip:

- 1) Verbindung zur Datenbank wird hergestellt
- 2) SQL-Statement wird ausgeführt
- 3) Commit oder Rollback
- 4) Verbindung zur Datenbank trennen

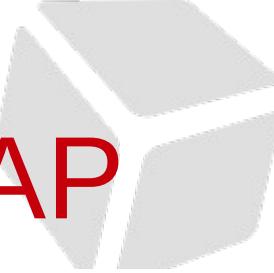
DBS-Einsatz: OLTP vs. OLAP



OLTP vs OLAP



DBS-Einsatz: OLTP vs. OLAP



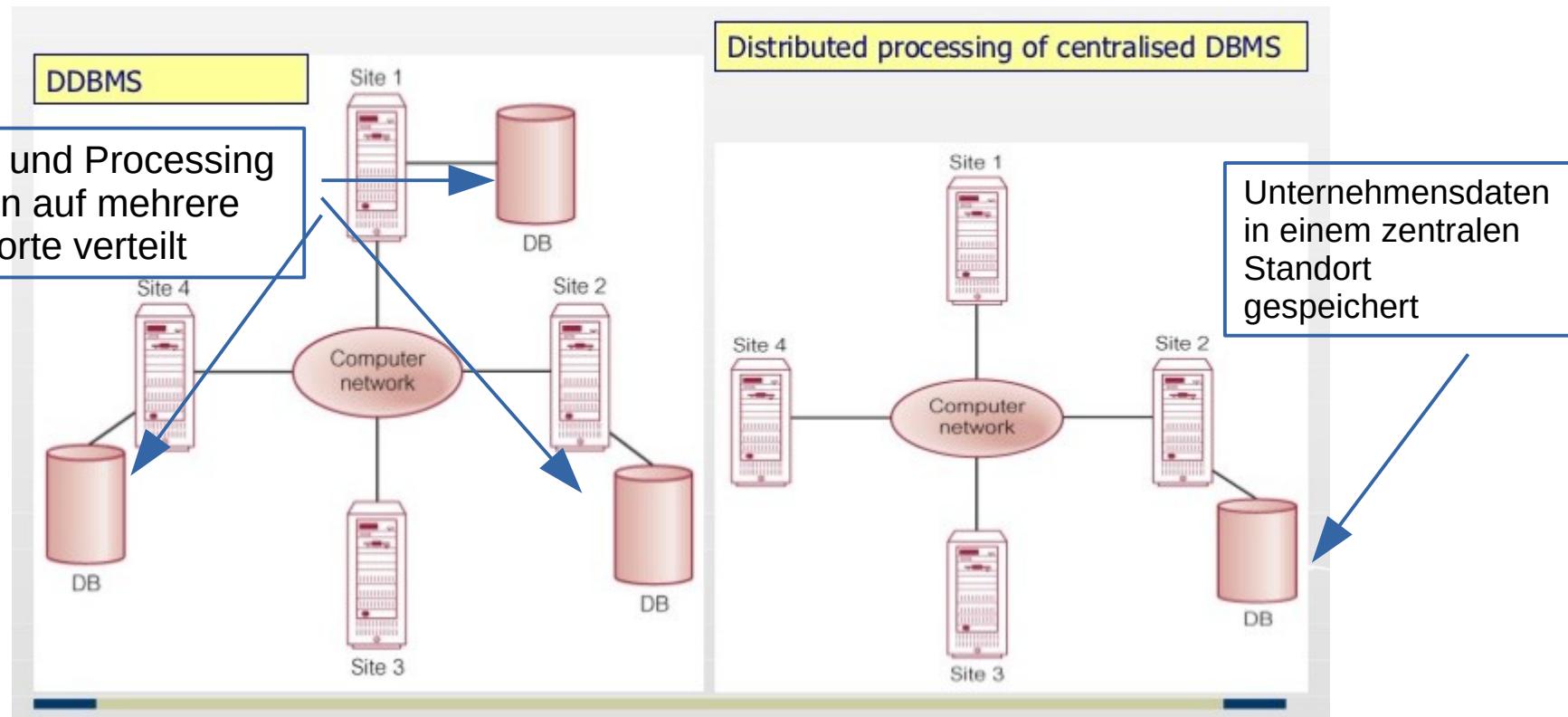
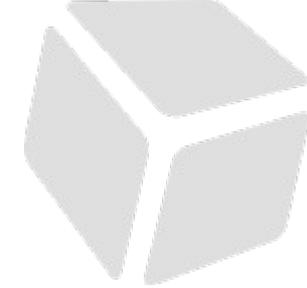
Vergleichsgrundlage	OLTP	OLAP
Basic	Es ist ein Online-Transaktionssystem und verwaltet Datenbankänderungen.	Es ist ein Online-Datenabruf- und Datenanalysesystem.
Fokus	Informationen aus der Datenbank einfügen, aktualisieren, löschen.	Extrahieren Sie Daten zur Analyse, die bei der Entscheidungsfindung helfen.
Daten	OLTP und seine Transaktionen sind die ursprüngliche Datenquelle.	Unterschiedliche OLTP-Datenbanken werden zur Datenquelle für OLAP.
Transaktion	OLTP hat kurze Transaktionen.	OLAP hat lange Transaktionen.
Zeit	Die Bearbeitungszeit einer Transaktion ist in OLTP vergleichsweise geringer.	Die Bearbeitungszeit einer Transaktion ist in OLAP vergleichsweise länger.
Anfragen	Einfachere Abfragen.	Komplexe Abfragen.
Normalisierung	Tabellen in der OLTP-Datenbank werden normalisiert (3NF).	Tabellen in der OLAP-Datenbank werden nicht normalisiert.
Integrität	Die OLTP-Datenbank muss die Integrität der Datenintegrität aufrechterhalten.	Die OLAP-Datenbank wird nicht häufig geändert. Daher ist die Datenintegrität nicht betroffen.



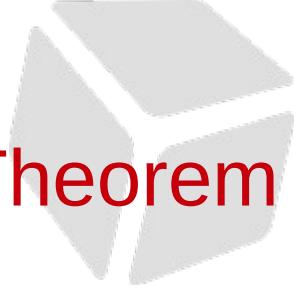
Verteilte DBMS (DDBMS)

- Ein verteiltes Datenbankverwaltungssystem (DDBMS) ist ein zentrales Softwaresystem, das eine verteilte Datenbank verwaltet.
- Beispiele für verteilte DBMS sind Master-Slave-Systeme, Client-Server-Systeme (gering verteilt) und Peer-to-Peer Systeme (stark verteilt).
- Eigenschaften
 - Es dient zum Erstellen, Abrufen, Aktualisieren und Löschen.
 - Es synchronisiert die Datenbank periodisch.
 - Es stellt sicher, dass die an jeder Stelle geänderten Daten allgemein aktualisiert werden.
 - Es wird in Anwendungsbereichen eingesetzt, bei denen grosse Datenmengen verarbeitet werden, und von zahlreichen Benutzern gleichzeitig zugegriffen.
 - Es ist für heterogene Datenbankplattformen.
 - Es behält die Vertraulichkeit und die Integrität der Daten der Datenbanken.

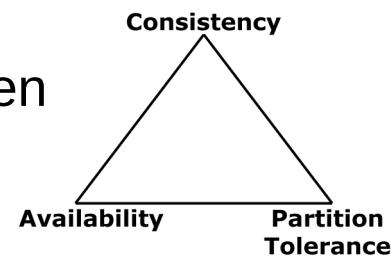
Verteilte DDBMS - Erfolgsfaktoren



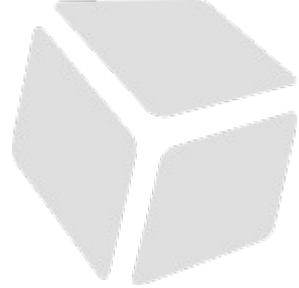
Trade-Off's verstehen – Brewer's CAP Theorem



- In einem verteilten System ist unmöglich, gleichzeitig die drei Eigenschaften Konzistenz (**Consistency**), Verfügbarkeit (**Availability**) und Ausfalltoleranz (**Partition Tolerance**) zu garantieren.
- Nur zwei Eigenschaften können gleichzeitig erfüllt werden



Consistency	Availability	Partition Tolerance
Konsistenz der gespeicherten Daten. Ein Update wird bei allen Usern gleichzeitig sichtbar.	Akzeptable Antwortzeiten. Das System ist jederzeit für jede gültige Anfrage verfügbar.	Das System arbeitet auch bei Ausfall eines Knotens.



BASE – Eigenschaft für verteilte DBMS

- In verteilten Datenbanken sollen alle ACID-Eigenschaften erfüllt werden?
- Diese Probleme wurden in dem CAP-Theorem von Brewer formuliert.
- Im Umfeld der NoSQL-Datenbanken wird daher häufig das BASE-Prinzip (Basically Available, Soft state, Eventual consistency) verfolgt.
- **Basically Available** (grundsätzlich Verfügbar): Das System garantiert die Verfügbarkeit im Sinne des CAP-Theorems.
- **Soft State** (loser Zustand): Der Zustand des Systems kann mit der Zeit ändern, auch ohne Eingabe. Dies ist wegen des schlussendlichen Konsistenzmodells.
- **Eventually Consistent** (schlussendlich Konsistent): Das System wird im Laufe der Zeit konsistent, angenommen dass, das System während dieser Zeit keine Eingabe erhält.
- Daten sind stets abrufbar aus der Datenbank, wobei nicht garantiert ist, dass sie sich im aktuellsten Zustand befinden, aber sie werden nach und nach aktualisiert, sodass in absehbarer Zeit Konsistenz erreicht wird.



NoSQL Datenbanken



NoSQL Datenbanken Entwicklung

- DBS fokussieren auf die effiziente und sichere Verarbeitung grosser Menge strukturierten Daten
- Der Grossteil der Daten (Web, soziale Netze, mobile Geräten) ist teilstrukturiert (E-Mail Nachrichten, Webseiten, Benutzerprofile, Produktangebote, etc.) oder unstrukturiert (z.B. Fotos, Videos)
- Die Verarbeitung solcher Daten (bzw. Generell von Big Data) mit RDBS ist ineffizient und unflexibel
- Entwicklung von sogenannten NoSQL-Systeme
- Challenges:
 - Data Volume
 - Data Velocity
 - Data Variety



NoSQL Datenbanken

Key/Value Store (Redis)

Column Store (**Cassandra, HBase**)

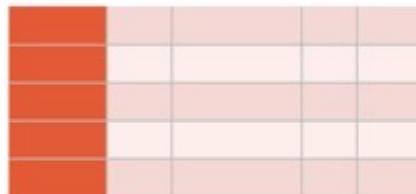
Document Store (**MongoDB, CouchDB**)

Graph Database (neo4j)

noSQL – not only SQL

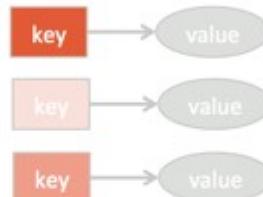
SQL-Datenbank-Modell

Relational:

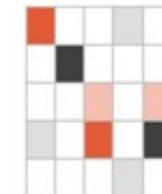


noSQL-Datenbank-Modelle

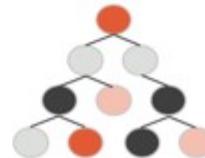
Key-Value:



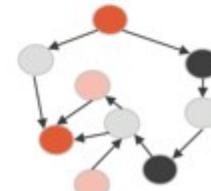
Wide-Colum:

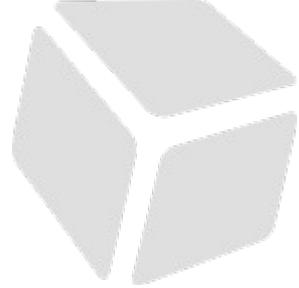


Document-Store:



Graph:

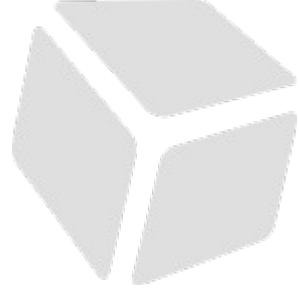




Key/Value Store

- Key-Value-Datenbanken sind die einfachste Form von NoSQL-Software.
- Sie paaren einzigartige Schlüssel mit den zugehörigen Werten in Datenelementen.
- Die grössten Vorteile sind die Skalierbarkeit und die effiziente und sehr schnelle Datenverarbeitung.
- Da der Zugriff auf einen Datensatz lediglich über einen Schlüssel erfolgt, ist ein bedeutender Nachteil die eingeschränkte Abfragemöglichkeit.

Key	Value
K1	AAA,BBB,CCC
K2	AAA,BBB
K3	AAA,DDD
K4	AAA,2,01/01/2015
K5	3,ZZZ,5623



Column Store

- Spaltenorientierte Datenbanken halten Daten in Tabellen vor, die sehr grosse Mengen an Spalten enthalten können.
- Mit einem spaltenorientierten Datenbankmanagementsystem (DBMS) können Daten mit sehr hoher Geschwindigkeit gelesen werden. Die Speicherung erfolgt spaltenweise und nicht wie bei herkömmlichen relationalen Datenbanken zeilenweise.

PERNR	NACHNAME	VORNAME	AUFGABE	GEHALT
1	Gustav	Gruber	DBA	3000
2	Frieda	Schulz	Controlling	3000
3	Peter	Schiller	Vertrieb	3600
4	Ulla	Heiner	Forschung	4000

Relational

1,Gustav,Gruber,DBA,3000;2,Frieda,Schulz,Controlling,3000;3,Peter,Schiller,Vertrieb,3600;4,
Ulla,Heiner,Forschung,4000;

Spaltenorientiert

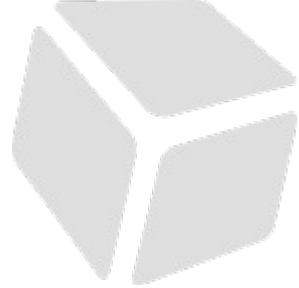
1,2,3,4;Gustav,Frieda,Peter,Ulla;Gruber,Schulz,Schiller,Heiner;DBA,Controlling,Vertrieb,
Forschung;3000,3000,3600,4000;



Document Store

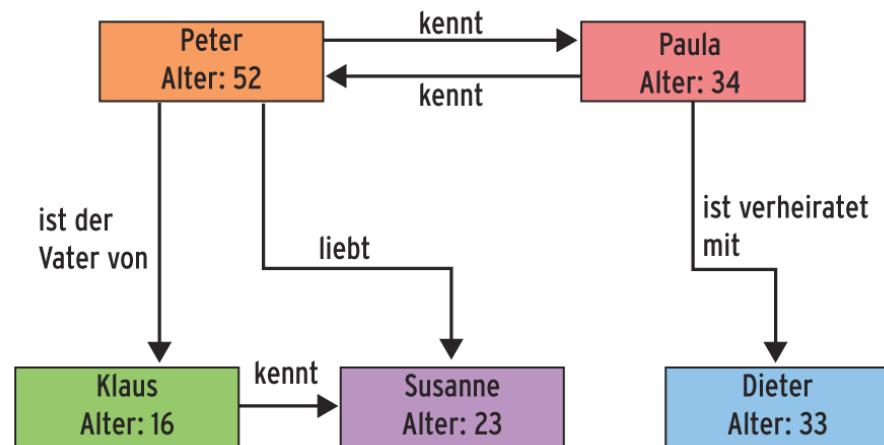
- Diese Datenbanken speichern Datenelemente in dokumentenähnlichen Strukturen, was in einigen Fällen sogar einfacher ist, da sie schemalos sind.
- Macht sinn, wenn die Daten Dokumente sind
- Sie werden zum Beispiel oftmals bei Content Management Systemen (CMS) eingesetzt. Außerdem sammelt und verarbeitet man damit Daten, die von Web- und mobilen Anwendungen mit hohem Traffic kommen, um sie überwachen.

```
{  
  name: "sue",  
  age: 26,  
  status: "A",  
  groups: [ "news", "sports" ]
```

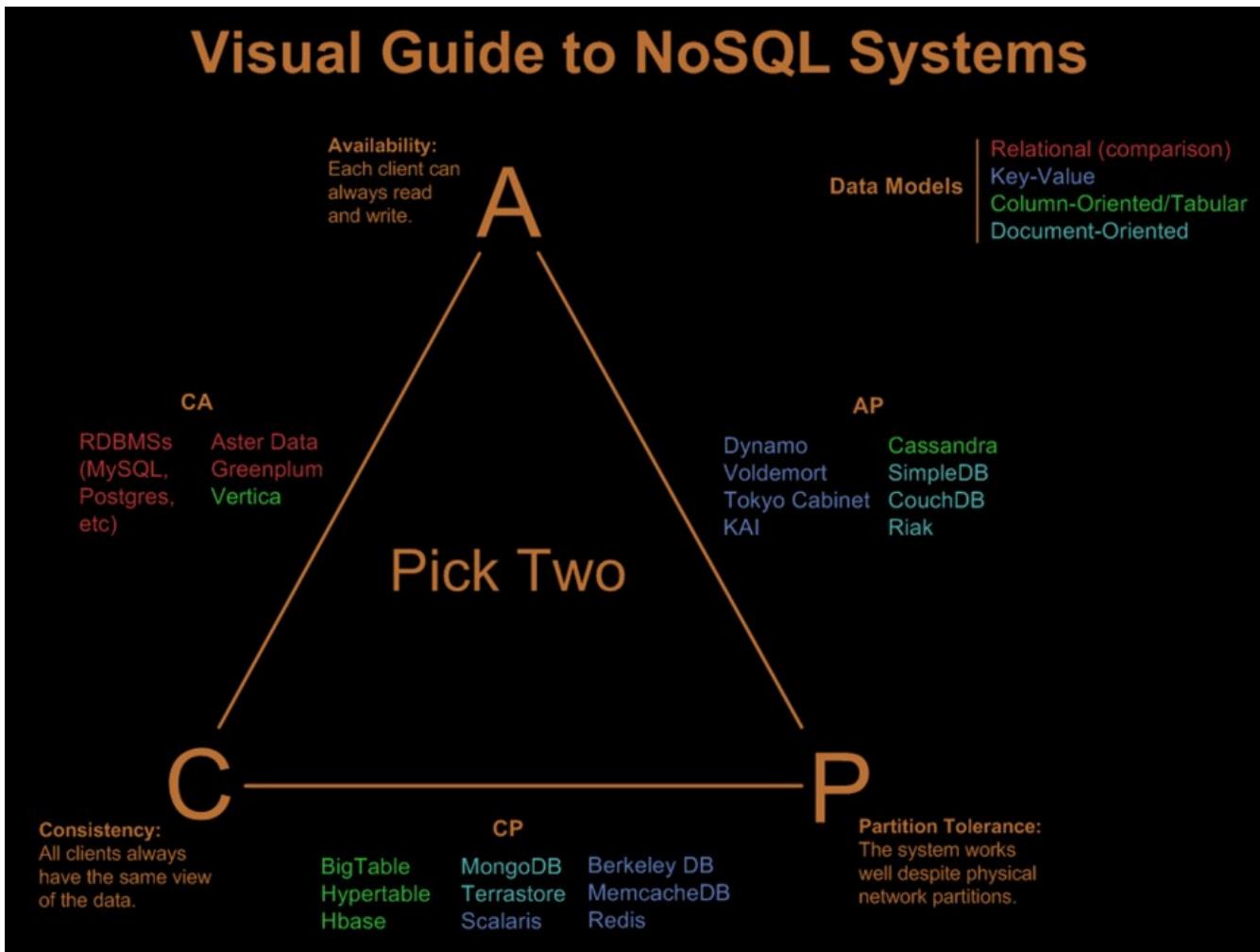


Graph Database

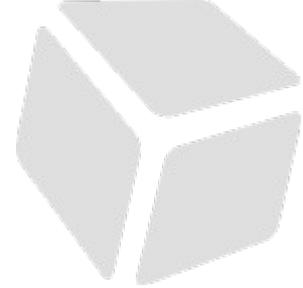
- Eine Graphdatenbank ist eine Datenbank, die Graphen benutzt, um stark vernetzte Informationen darzustellen und abzuspeichern. Ein solcher Graph besteht aus Knoten und Kanten, den Verbindungen zwischen den Knoten.
- Typisches Beispiel für den Einsatz von Graphdatenbanken ist die Analyse der Nutzerbeziehungen in sozialen Netzwerken oder des Kaufverhalten von Nutzern in Onlineshops.



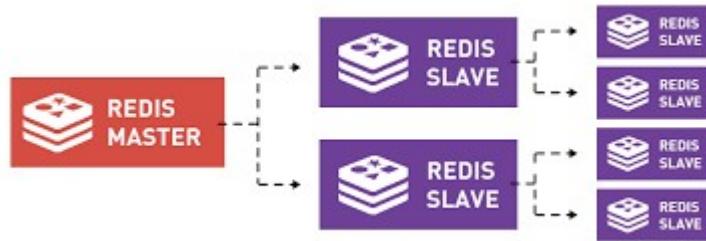
NoSQL-Datenbanken



Redis



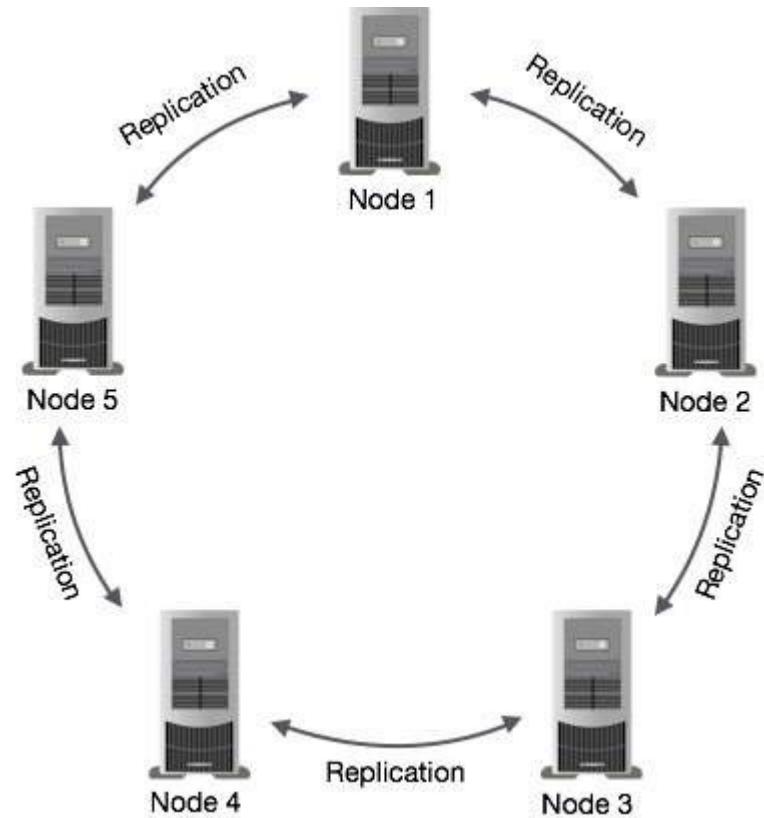
- **Key-Value Store**
- Redis ist ein Open Source, In-Memory-Datenstrukturspeicher, der als Datenbank, Cache und Message Broker verwendet wird.
- Unterstützt Datenstrukturen wie Strings, Hashes, Listen, Sets, Hyperloglogs, Geodaten und Streams. Es verfügt über integrierte Replikation, Transaktionen und verschiedene Ebenen der Persistenz auf der Festplatte und bietet **Hochverfügbarkeit** über Redis Sentinel und automatische Partitionierung mit RedisCluster.



Cassandra



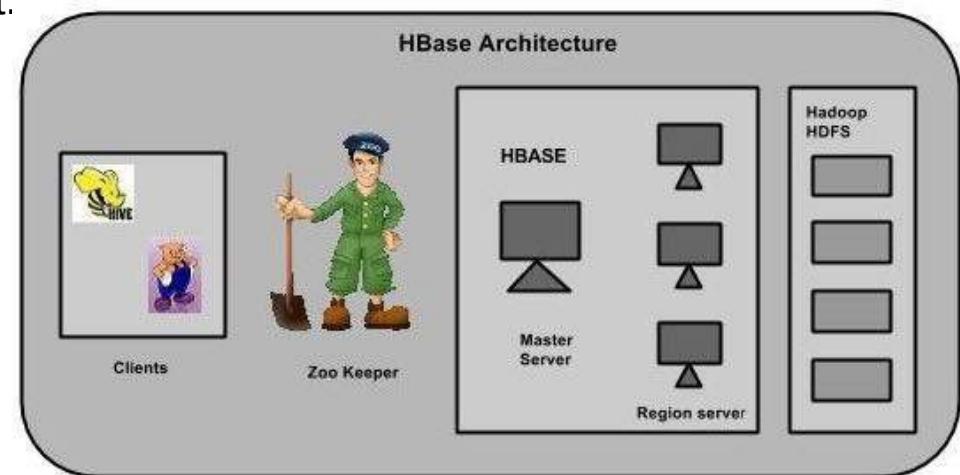
- **Column-Store, Stellt Verfügbarkeit über Konsistenz**
- Cassandra ist ein Wide-Column Store, der Daten auf mehrere Server verteilt hält. Fällt einer der Server aus, dann ist Cassandra so gebaut, dass das System weiterhin reibungslös läuft. Der Preis der für diese Hochverfügbarkeit gezahlt ist, ist die Konsistenz. Es ist dann möglich, dass während kurzer Zeit eine Anfrage an Cassandra ein nicht ganz aktuelles Ergebnis liefert.



HBase



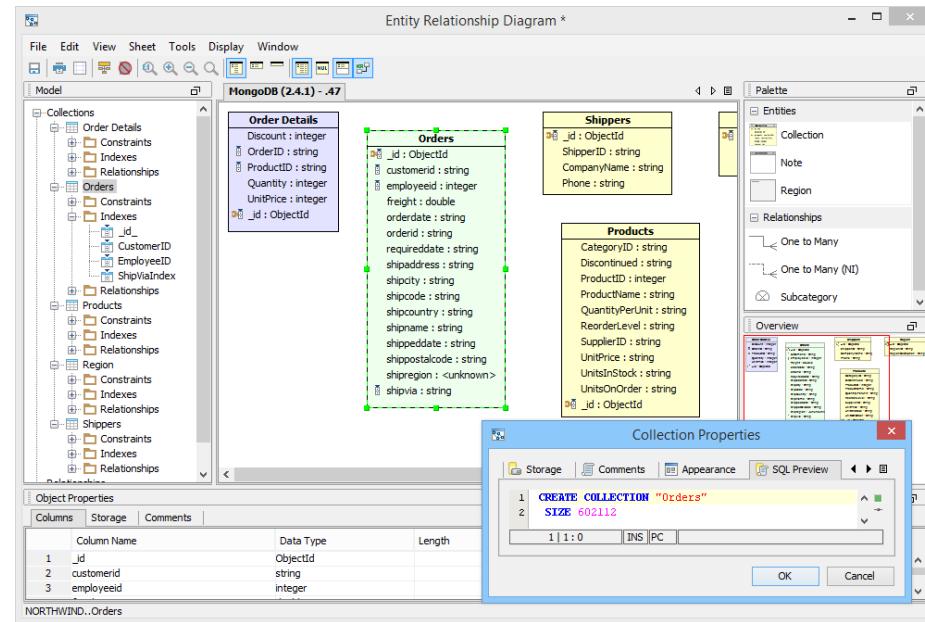
- **Column-Store**
- **Stellt Konzistenz über Verfügbarkeit**
- Hbase ist auch ein Wide-Column Store, der Daten auf mehrere Server verteilt hält. Er basiert auf HDFS und hat mit dem Namenode einen single point of failure. Fällt dieser aus, dann fällt das Gesamtsystem aus. Solange der Namenode läuft, werden die Daten in Hbase konsistent aufbewahrt.



MongoDB



- **Dokumentenorientierte NoSQL-Datenbank**
- **Stellt Verfügbarkeit über Konsistenz**
- Horizontale Skalierbarkeit
- Um Ausfälle zu kompensieren:
- Master-Slave-Replikation: Ein Slave kopiert die Daten vom Master (nur Lesezugriffe)
- Replica-Sets: ähnelt dem Master-Slaves Verhältnis, enthalten aber die Möglichkeit für die Slaves, einen neuen Master zu wählen, wen der derzeitige ausfällt



Data Lakes

Die üblichen Probleme eines Big-Data-Projekts

- Welche Datenquellen sind verfügbar?
- Wo sind die Daten, die ich brauche?
- Welche Schnittstellen bietet die Datenquellen?
- Mit welchem API kann ich effizient auf die Daten zugreifen?
- Wie kann ich meine Daten mit anderen Daten verknüpfen?
- Wie kann ich die Daten in meine gewünschte Struktur bringen?
- Wie kann man die Daten kontinuierlich analysieren?

Daten Zugriff & Verfügbarkeit

Data Lakes als universeller Datenspeicher

Daten-
quellen

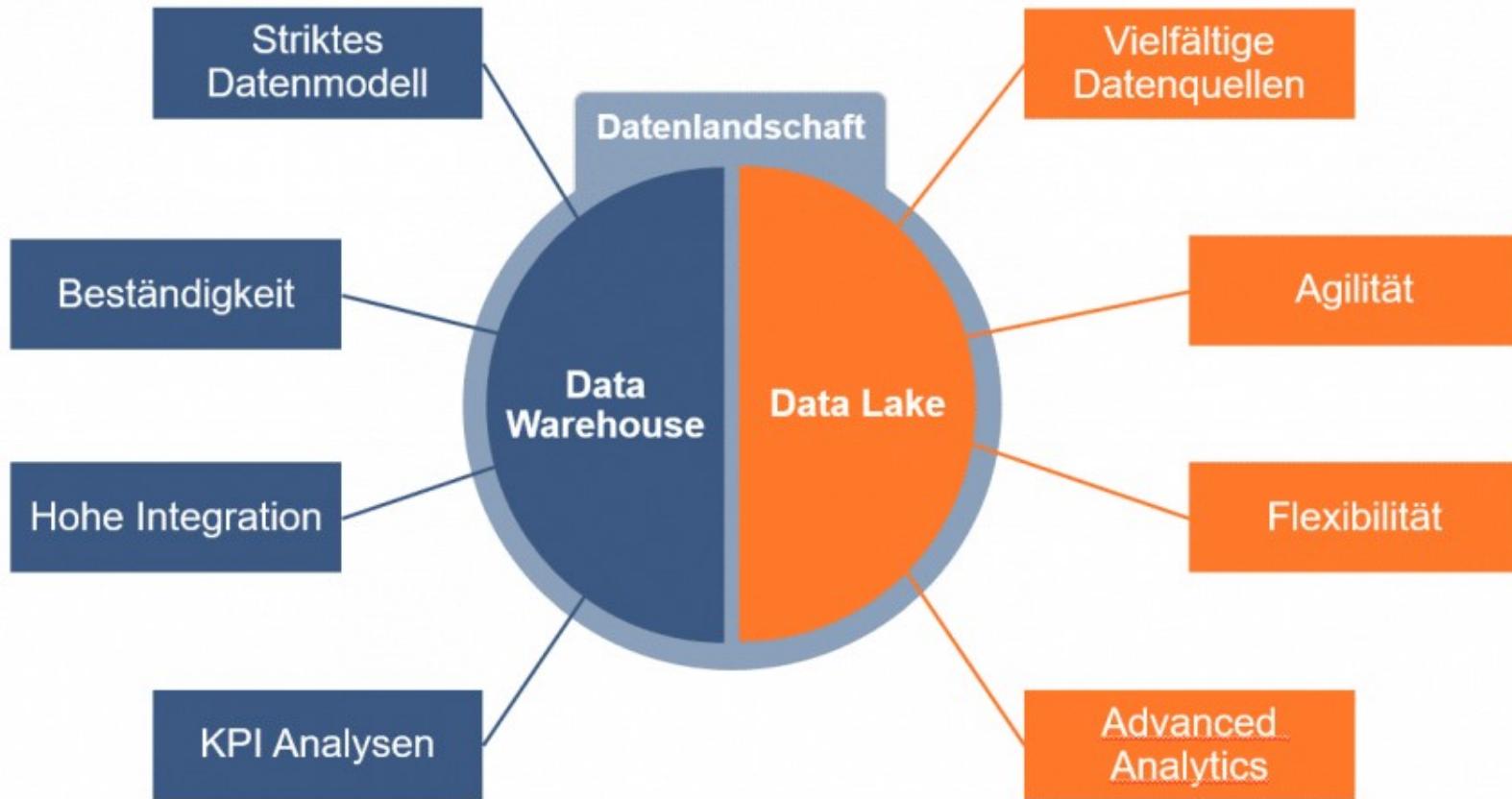


Heterogenität
Unsicherheit
Komplexität

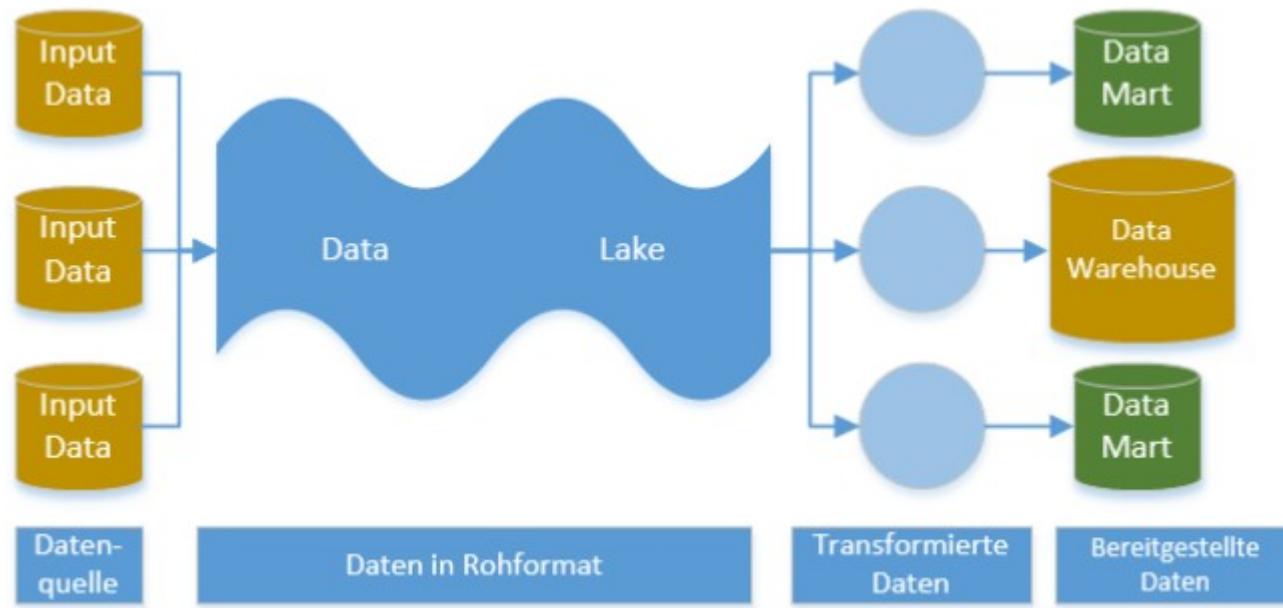


Daten-
nutzer

Data Lake vs Data Warehouse



Data Lake Architektur

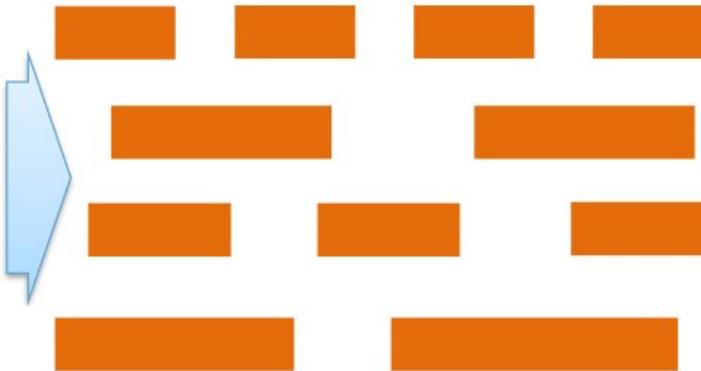


Batch vs. Stream

Data Producers

- Clickstream
- Transactions
- Machine Logs
- Sensor Data

Batches of Data



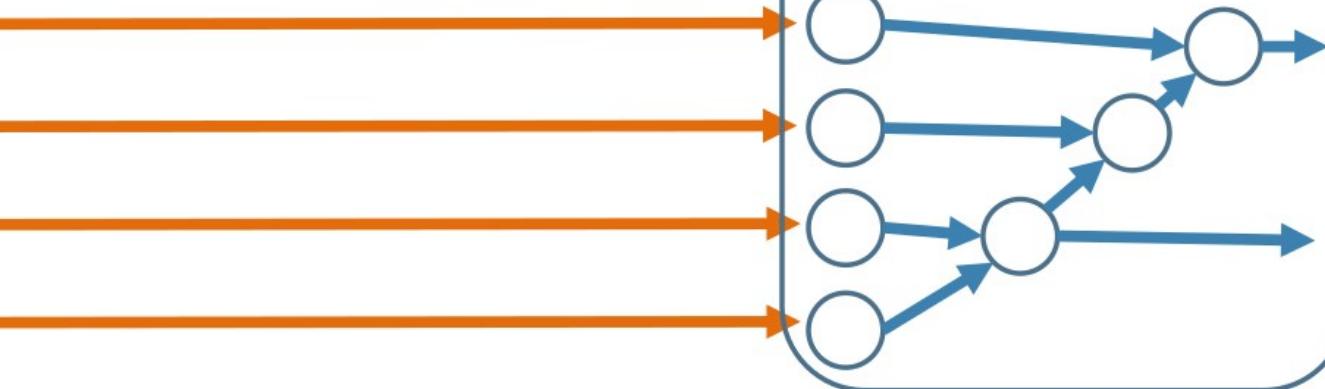
Batch Processing



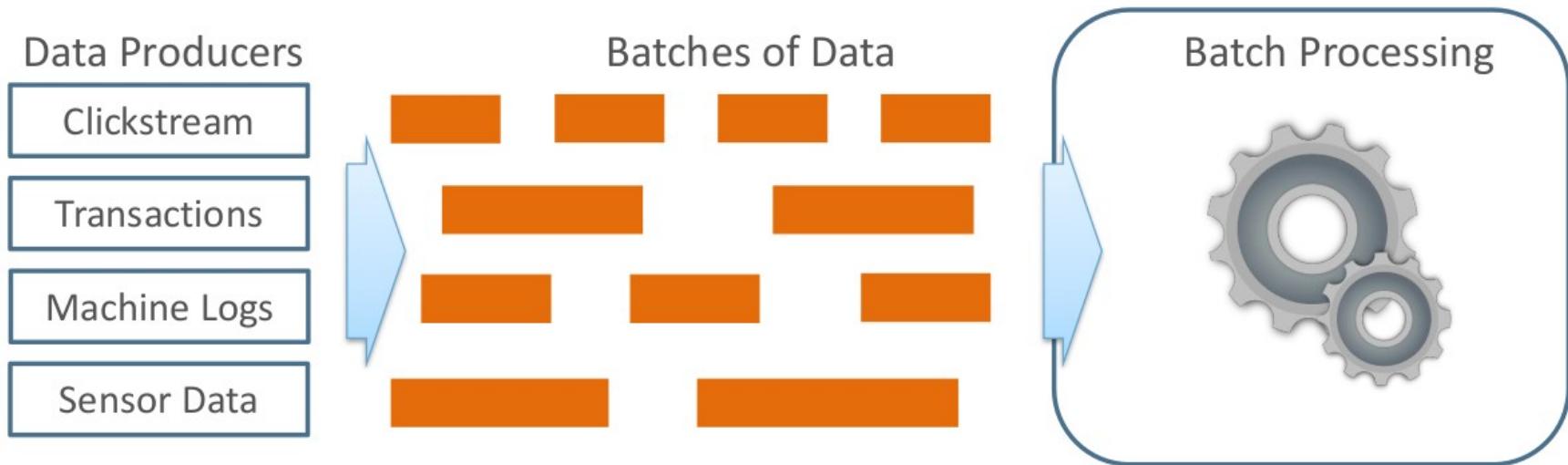
Streams of Data

- Clickstream
- Transactions
- Machine Logs
- Sensor Data

Stream Processing

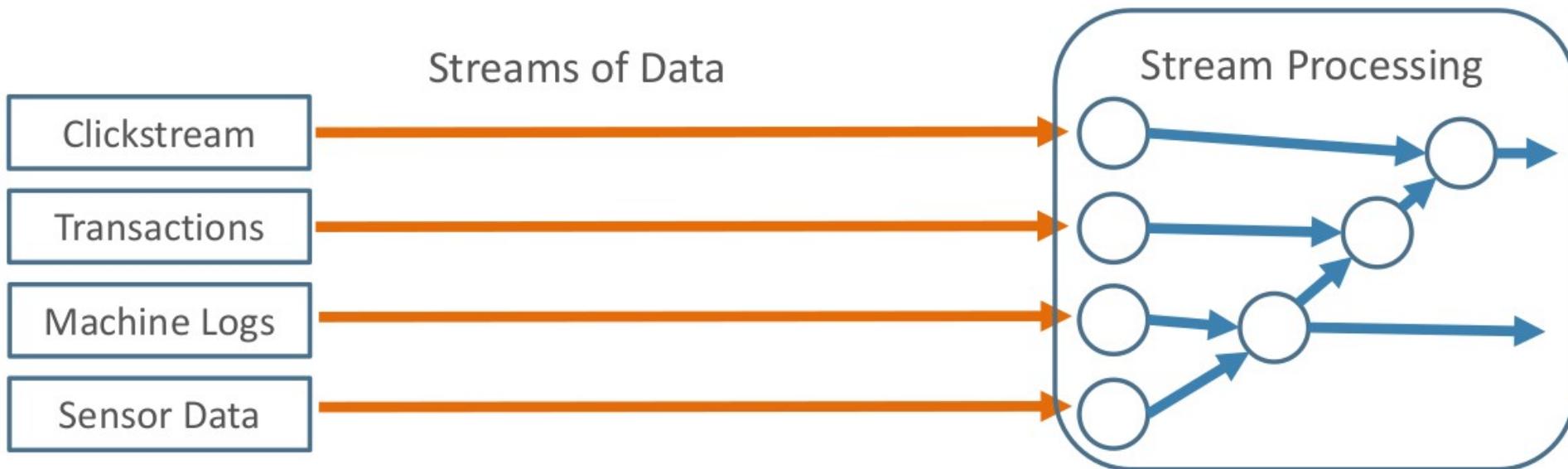


Batch Processing and Analysis - Data@Rest



- Zugriff auf alle Daten
- Split in Batches
- Verarbeitung aller Daten gleichzeitig
- Antwort am Ende
- Complex Analytics (z.B. Model Training)

Streaming Processing and FFHS Analysis – Data in Motion

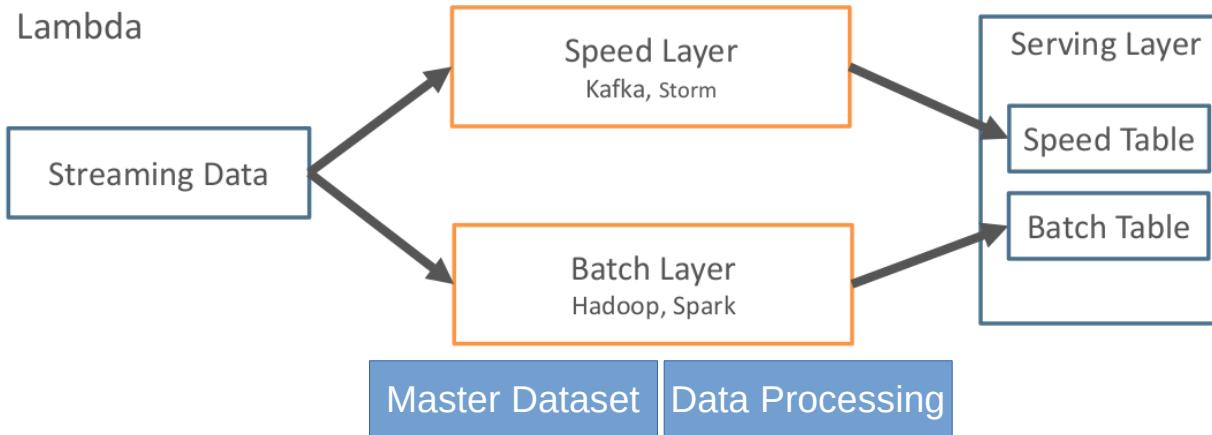


- Verarbeitung eines eingehenden Datenstroms
- Sofortige Antwort
- Die Ergebnisse basieren sich auf aktuellen Daten
- Optimierung der Latenzzeit (z.B. durchschnittliche Zeit für ein Record)
- Die Berechnung muss in Echtzeit abgeschlossen werden
- Berechnet etwas relativ Einfaches (z.B. die Verwendung vordefinierter Modelle zur Labeling eines Datensatzes)

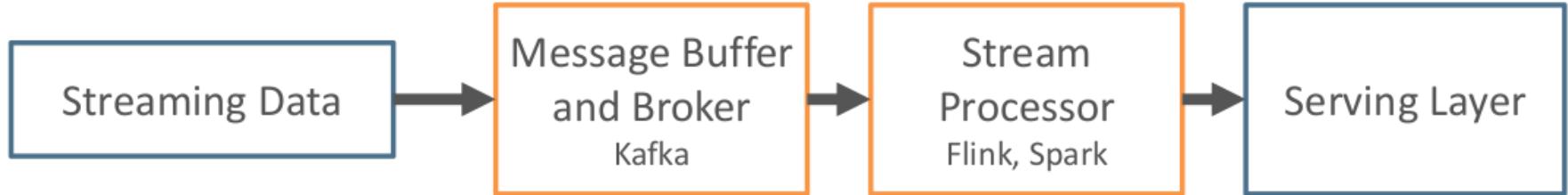
Lambda und Kappa Architektur

- Ziel: Schelle Datenauswertung
- Im Open-Source-Bereich gibt es viele Streamingtechnologien (grösstenteils betreut von der Apache Software Foundation).
- Dieser Technologien sind Teil einer Streamingarchitektur
- Im Big-Data-Bereich sind vor allem zwei Architekturen bekannt: Die Lambda-Architektur und die Kappa-Architektur
- Die Lambda-Architektur ist historisch gesehen älter

Lambda und Kappa Architektur

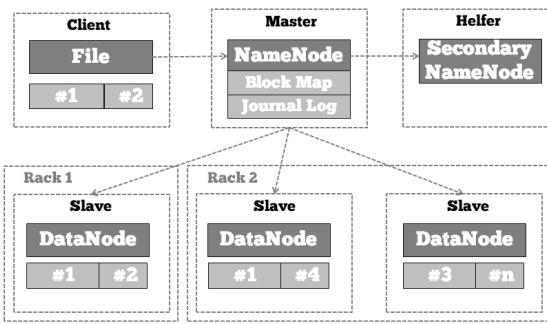


Kappa



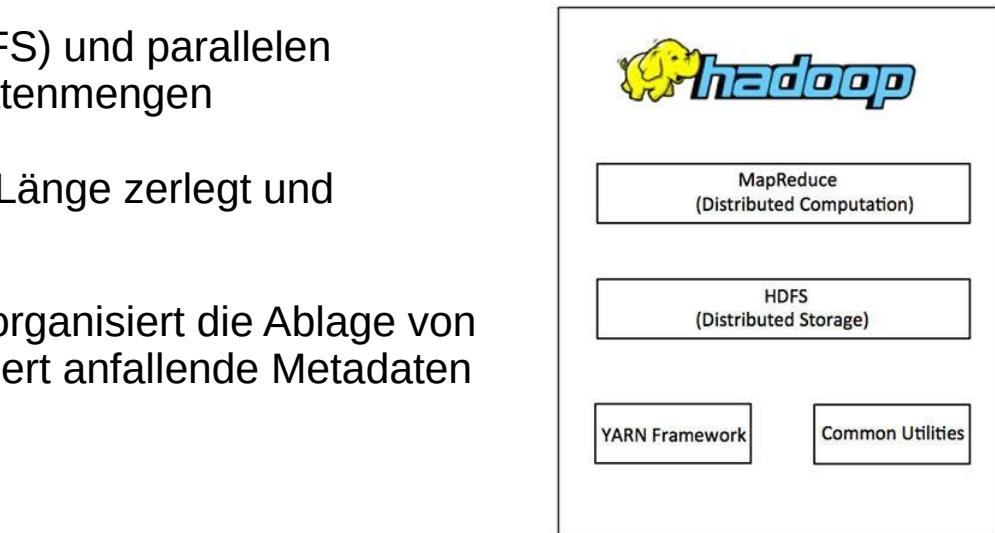
Hadoop-Batch Processing Framework

- Framework zum verteilten Speichern (HDFS) und parallelen Verarbeiten (MapReduce) von grossen Datenmengen
- Dateien werden in Datenblöcke mit fester Länge zerlegt und redundant auf die Knoten verteilt.
- Master: bearbeitet eingehende Anfragen, organisiert die Ablage von Dataiern in den Sklavenknoten und speichert anfallende Metadaten

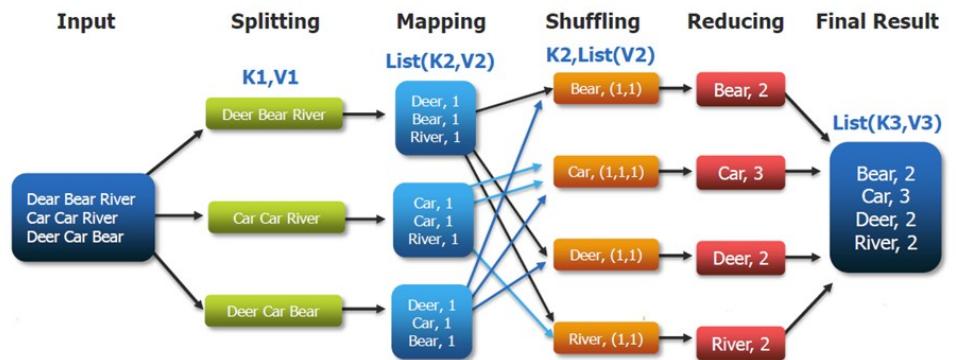


Stärken und Grenzen:

- Ausgereifte Basistechnologie
- Umfangreiches Ökosystem
- Breite Kompatibilität
- Scalability
- Speichert nicht in-memory
- MapReduce ist langsam (reading/writing)



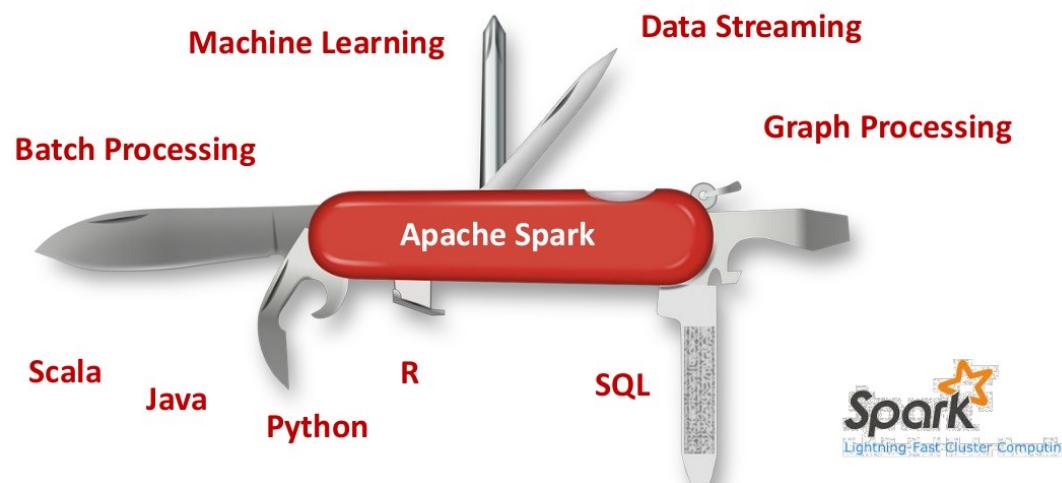
Divide-and-Conquer Ansatz The Overall MapReduce Word Count Process



Apache Spark

The Swiss Army Knife of Big Data

- Apache Spark ist ein Batch-Processing-Framework der nächsten Generation mit Stream-Verarbeitungsfunktionen.
- Wurde nach den gleichen Prinzipien wie die MapReduce-Engine von Hadoop entwickelt und konzentriert sich in erster Linie auf die **Beschleunigung des Batch-Processings** durch vollständige **In-Memory-Berechnung (durch RDDs)** und Prozessoptimierung.
- Spark kann als eigenständiger Cluster eingesetzt werden (wenn er mit einer geeigneten Speicherschicht gekoppelt ist) oder alternativ zur MapReduce-Engine in Hadoop eingebunden werden.
- Effiziente Entwicklung durch mächtige API (für Scala, Java und Python).
- Im Gegensatz zu MapReduce verarbeitet Spark alle **Daten in-Memory** und interagiert mit der Speicherschicht, nur um die Daten zunächst in-Memory zu laden und am Ende die Endergebnisse zu erhalten. Alle Zwischenergebnisse werden in-Memory verwaltet.



Streaming Frameworks

	Storm	Flink	Spark Streaming
Delivery Guarantees	at least once	exactly once	exactly once
Latency	very low	low	high
Throughput	medium	high	high
Processing Model	stream	stream	micro-batch
Resource Management	YARN	YARN	YARN
Functionality	stream-only	stream & batch	stream & batch

<http://data-artisans.com/high-throughput-low-latency-and-exactly-once-stream-processing-with-apache-flink/>

<https://www.digitalocean.com/community/tutorials/hadoop-storm-samza-spark-and-flink-big-data-frameworks-compared>

<https://databaseline.wordpress.com/2016/03/12/an-overview-of-apache-streaming-technologies/>

Zusammenfassung

- Die Batch-Verarbeitung ist für Anwendungsfälle geeignet, in denen es nicht auf die Aktualität der Daten ankommt und die Toleranz für langsamere Reaktionszeiten höher ist. Beispiel: Offline-Analyse historischer Daten.
- Die Stream-Verarbeitung hingegen ist für Anwendungsfälle notwendig, die Live-Interaktion und Echtzeit-Reaktion erfordern. Beispiele: Die Verarbeitung von Finanztransaktionen, die Erkennung von Betrug in Echtzeit und die Preisgestaltung in Echtzeit.
- Der Begriff "Microbatch" wird häufig verwendet, um Szenarien zu beschreiben, in denen Batches klein sind und/oder in kleinen Abständen verarbeitet werden. Auch wenn die Verarbeitung oft erfolgt, werden die Daten immer noch batch-weise verarbeitet. Beispiele: Spark-Streaming, Flink

Data Science und Ethik Diskussion

Big Data und Ethik

1. Daten und Identität von Privatkunden sollten privat bleiben
2. Gemeinsame personenbezogene Daten sollten vertraulich behandelt werden
3. Kunden sollten einen transparenten Überblick haben

Datenschutz

- Der Datenschutz ist ein in der Bundesverfassung festgeschriebenes Grundrecht
- **Art. 13: Schutz der Privatsphäre**

Jede Person hat Anspruch auf Achtung ihres Privat- und Familienlebens, ihrer Wohnung sowie ihres Brief-, Post- und Fernmeldeverkehrs.

Jede Person hat Anspruch auf Schutz vor Missbrauch ihrer persönlicher Daten.

- Konkretisierung dieses Anspruchs in Datenschutzgesetzen
 - Datenschutzgesetzt (DSG)
 - Kantonale Datenschutzgesetzt

Was ist Datenschutz?

- Datenschutzgesetz
- **Art 1 Zweck**

Dieses Gesetz bezweckt den Schutz der Persönlichkeit und der Grundrechte von Personen, über die Daten bearbeitet werden.

- Datenschutz = Personlichkeitsschutz
- Der DS schützt nicht die Daten sondern die betroffene Personen vor denjenigen, die ihre Daten bearbeiten.

Wozu braucht man Datenschutz?



Eigentlich hätten Angestellte warten sollen, bis ein Container mit alten Dossiers von Patienten von einer Datenvernichtungs-Firma abgeholt wird. Stattdessen lassen sie den Container unbewacht vor dem Gebäude stehen.



Da staunte Datenschutz-Anwalt Martin Steiger

Migros verschickt fremde Cumulus-Informationen

 Pascal Tischhauser

 20:02 Uhr
09.04.2018

 01:14 Uhr
01.10.2018

Internetanwalt Martin Steiger wollte wissen, was für Daten die Migros über ihn im Cumulus-Bonusprogramm gesammelt hatte. So forderte er diese beim orangen Riesen an. Doch die Migros schickte ihm nicht bloss seine Daten, sondern auch die einer wildfremden Person.

Personendaten

- Anonymisierung
 - Ist das Verändern von Personendaten derart, dass die Angaben über persönliche oder sachliche Verhältnisse nicht mehr oder nur mit einem unverhältnismässig grossen Aufwand Personen zugeordnet werden können
- Anonymisierte Daten sind keine Personendaten. Deren Bearbeitung unterliegt nicht den datenschutzrechtlichen Vorschriften.
- Pseudonymisierung
 - Ist das Ersetzen des Namens und anderer Identifikationsmerkmale durch ein Kennzeichen (z.B. Nummerncode) zu dem Zweck, die Bestimmung des Betroffenen auszuschliessen oder wesentlich zu erschweren.
- Pseudonymisierte Daten bleiben Personendaten im Sinne des Datenschutzrechts.

Analyse von Social Media Daten

Social Media Definition

Social Media ist ein Oberbegriff, der die verschiedenen Aktivitäten definiert, die Technologie, soziale Interaktion und die Konstruktion von Wörtern, Bildern, Videos und Audios integriert.

Oder: “Social media is people having conversations online”

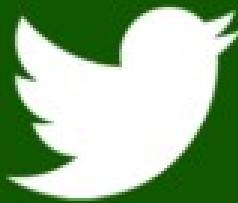


Social Media Statistics

Social Media Statistics



1.86B
monthly active
Daily: 1.23 billion



313M
monthly active



500M
registered members



700M
monthly active
Daily: 300 million



160M
daily active



1.2B
monthly active



150M
monthly active

UPDATED APRIL 2017

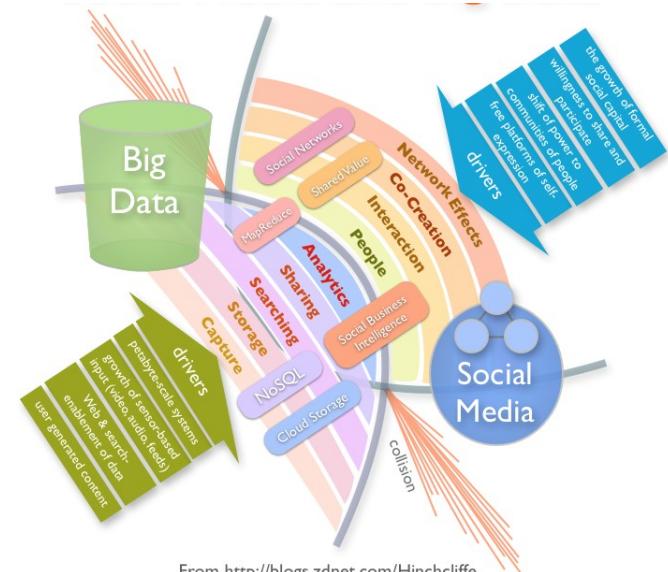
<https://neilpatel.com/blog/best-social-media-tools/>

2017 This Is What Happens In An Internet Minute



Herausforderungen

- Eine der grössten Herausforderungen bei so vielen Daten auf Social Media ist die Ableitung einer aussagekräftigen Kontextinformation.
- Social Media Daten sind unstrukturiert.



From <http://blogs.zdnet.com/Hinchcliffe>

Analyse von Social Media

- Social Network Analysis



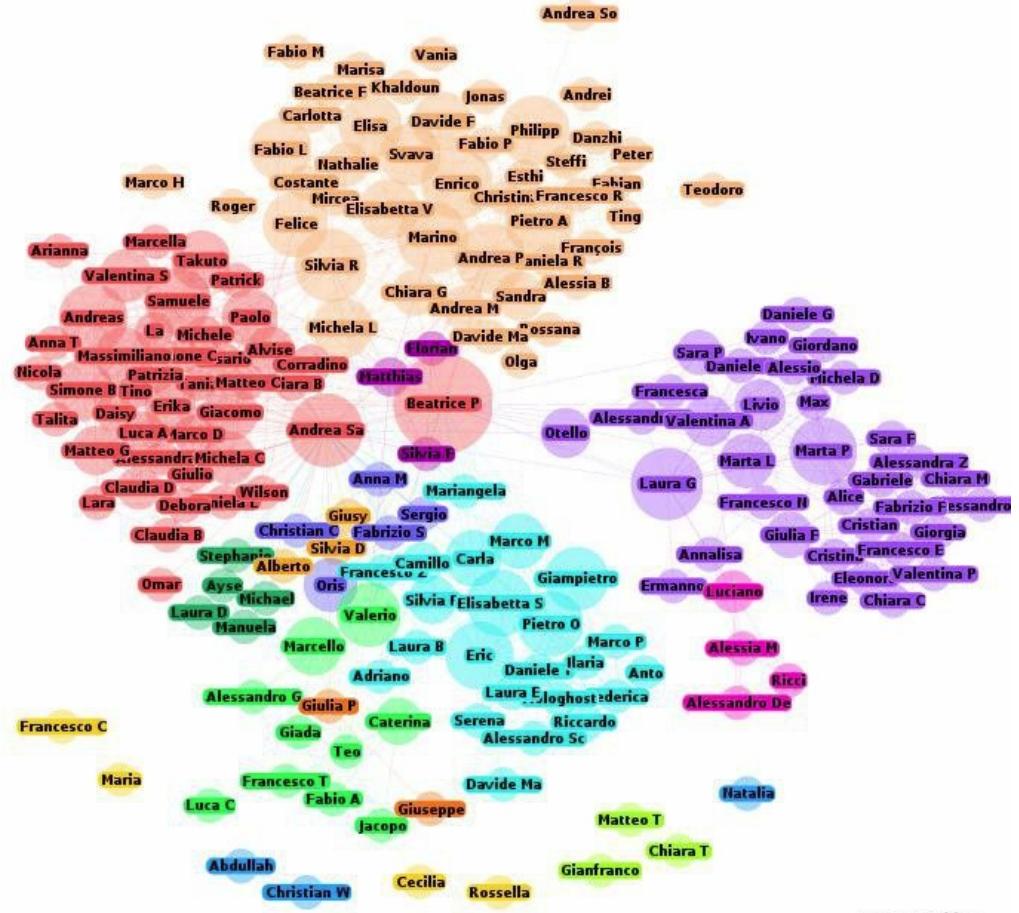
- Sentiment Analysis



Discovering people opinions, emotions and feelings about
a product or service

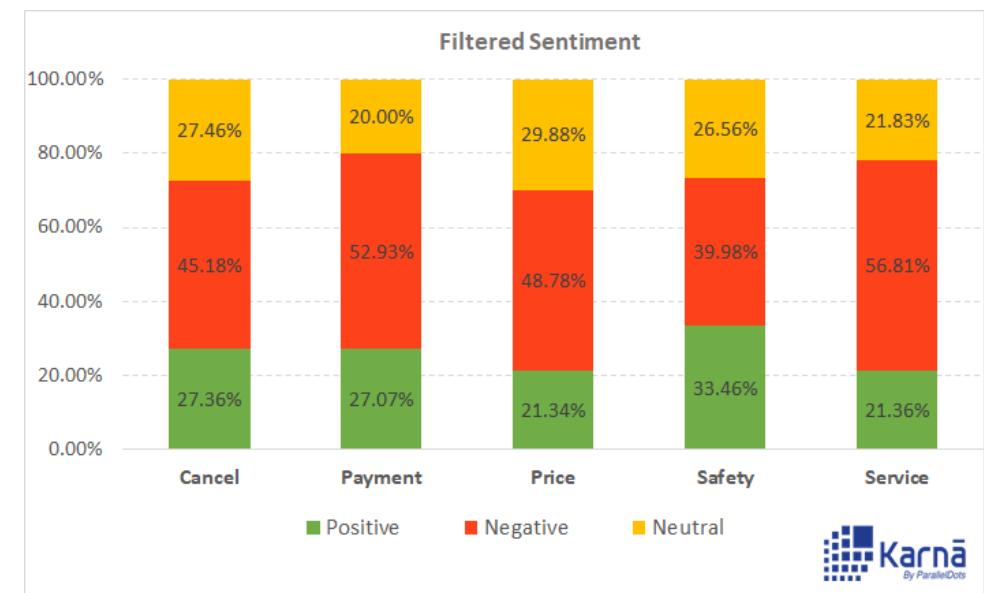
Social Networks Analysis

- Block 2
- Ein Netzwerk ist eine Sammlung von Entitäten, die durch eine Beziehung verbunden sind.
- People-People, Group-Group, innerhalb von Organisationen, Organisationsübergreifend
- z.B. Facebook verwendet die SNA, um potenzielle Freunde zu identifizieren und zu empfehlen, die auf Freunden von Freunden sich basieren.



Sentiment Analysis

- SA ist die Klassifizierung der Polarität eines bestimmten Textes oder Satz im Dokument.
- Ziel ist es, festzustellen, ob die im Text geäusserte Meinung positiv, negativ oder neutral ist.



Warum?

- Microblogging hat sich zu einem beliebten Kommunikationsmittel entwickelt.
- Die Meinung der Masse ist wichtig.
 - Politische Parteien möchten wissen, ob das Volk ihr Programm unterstützen oder nicht.
 - Bevor man in ein Unternehmen investiert, kann man die Stimmung der Menschen für das Unternehmen nutzen.
 - Ein Unternehmen möchte herausfinden, was die Leute von seinen Produkten hält.

Beispiel: Twitter

- Beliebte Microblogging-Seite
- Kurze Textnachrichten
- Millionen von aktiven Nutzern
- 500 Millionen Tweets täglich generiert
- Benutzer diskutieren oft über aktuelle Themen und teilen persönliche Ansichten zu verschiedenen Themen.
- Tweets sind klein in der Länge und damit eindeutig.

Challenges

- Tweets sind sehr unstrukturiert und nicht grammatikalisch.
- Out-of-Vocabulary Wörter
- Lexikalische Variationen
- Umfangreiche Verwendung von Akronymen wie asap, lol..... und Emoticons
- Sarcasm-Detection: positive Wörter mit negativem Smiley und umgekehrt

(mögliches) Verfahren

Tweet Downloader

Tokeniser

Preprocessing

Feature Extraction

Classification

Twitter API

Satz in Wörter
zerlegen

Hello Geeks how are you

StringTokenizer

Tokens

/ / | \ \
Hello Geeks how are you

z.B. Remove emoticons, replace
URLs, stop-words, Slang...

No	Data Field	Description
1	<code>id</code>	Unique id of the tweet
2	<code>created_at</code>	Tweet date and Time
3	<code>source</code>	Source of the tweet (Via web/Android/iPhone)
4	<code>text</code>	Tweet Text
5	<code>sentiment</code>	Sentiment of the tweet
	<code>polarity</code>	Separated the polarity from Sentiment
	<code>subjectivity</code>	Separated the subjectivity from Sentiment
6	<code>lang</code>	Language used in the tweet
7	<code>favorite_count</code>	Number of favorites per tweet
8	<code>retweet_count</code>	Number of retweets
9	<code>original_author</code>	Profile user name of the tweet's author
10	<code>possibly_sensitive</code>	Sensitivity of the message (Boolean true / false)
11	<code>Hashtags</code>	Extracted all the hashtags in the tweet
12	<code>User_mentions</code>	Any other profile mentions in the tweets
13	<code>Place</code>	User's location
14	<code>Place_coord_boundaries</code>	Coordinates of the tweet's location (if applicable)

z.B. binary (+/-)
3-stars
5-stars

Workshop

Workshop

Ziel

- Anhand von Twitter Daten eine einfache Analyse durchführen
- Weitere Begriffe klären
- Verstehen, wie beim Verfassen eines wissenschaftlichen Berichts vorgegangen werden kann.

Auftrag

Arbeiten Sie in Kleingruppen

Wählen Sie zwei Gebiete auf der Erde

- Untersuchen Sie, in welchem der Gebiete mehr trendige Topics auf Twitter erörtert werden (Siehe Example 1-2)
- Stellen Sie die Struktur eines wissenschaftlichen Berichts auf
- Zeigen Sie, welche Gegenden Sie gewählt haben und warum
- Wie gehen Sie vor zur Analyse
- Stellen Sie die Ergebnisse dar
- Interpretieren Sie die Ergebnisse
- Welche Schlussfolgerung ziehen Sie aus der Analyse
- Welches Fazit ziehen Sie insgesamt

To do's

- Nachbearbeitung
 - Semesterarbeit
- Vorbereitung Block2
 - Lektüre zu Netzwerkanalyse
 - Buch und Online Quellen
 - Notizen (Verständnisfragen)
 - Gephi installieren