



BigData

Buzzword, Hype oder Zukunftsthema?

BigData – Definition (komparativ und architekturell)

In einer McKinsey-Veröffentlichung¹⁾ wird Big Data als “***datasets whose size is beyond the ability of typical database software tools to capture, store, manage, and analyze.***” bezeichnet.

Diese Definition grenzt den Begriff Big Data vergleichend gegen herkömmliche Daten(banken) ab.

Das National Institute of Standards and Technology (NIST) definiert Big Data aus einer Architektursicht folgendermaßen:

“Big data is where the data volume, acquisition velocity, or data representation limits the ability to perform effective analysis using traditional relational approaches or requires the use of significant horizontal scaling for efficient processing.”

Konkret wird hier noch in *Big data science* (“... techniques covering the acquisition, conditioning, and evaluation of big data,”) und *Big data frameworks* (“software libraries along with their associated algorithms that enable distributed processing and analysis of big data problems across clusters of computer units”) unterschieden.

¹⁾ J. Manyika et al., Big data: The Next Frontier for Innovation, Competition, and Productivity. San Francisco, CA, USA: McKinsey Global Institute, 2011, pp. 1–137.

BigData - Anwendungsfelder

- **Marketing und Vertrieb**

- Web Analytics (Empfehlungsdienste, Analyse von Marketingkampagnen)
- Kundenbindungs- und Kundenabwanderungsanalyse
- Social Data
- Echtzeitauswertung von Webstatistiken und Anpassung von Online-Werbemaßnahmen

- **Produktion und Logistik**

- Maschinendatenerfassung/Messdaten für die Produktionssteuerung
- RFID/Bewegungsaufzeichnungen
- Planung von Lager- und Warenwirtschaftsdaten
- Materialverbrauch bzw. Fertigwarenerfassung (Gewicht, Maße, Qualität, Bilder)

- **Medizin**

- Erkennen von Zusammenhängen in der medizinischen Diagnostik, z.B. in der personalisierten Medizin zum Finden individueller Therapiemöglichkeiten

- **Finanzbereich**

- Entdeckung von Unregelmäßigkeiten bei Finanztransaktionen (Fraud-Detection)

- **Personaldatenmanagement**

- Personal-Recruiting
- Personaleinsatz

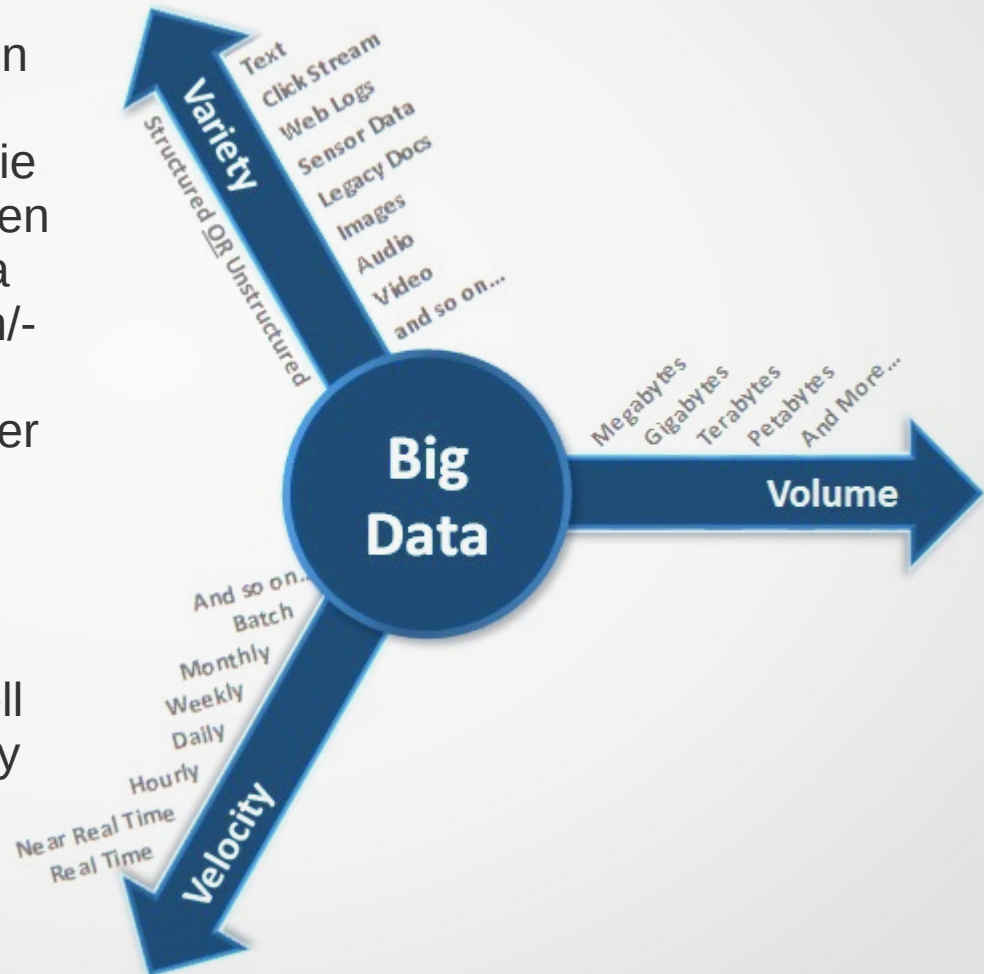
Die gesteigerte Bedeutung von Big Data lässt sich ua damit erklären, dass der *betriebswirtschaftliche Wert von Unternehmensdaten* zunehmend erkannt wird.

Informationen, die sich aus Rohdaten erschließen lassen, stellen oft einen Wettbewerbsvorteil dar. Beispiele dafür sind etwa sogenannte Forecasts oder Nowcasts zur Vorhersage zukünftiger Ereignisse oder Simulationen von Verkaufszahlen oder Nutzerverhalten.

BigData – Definition (attributiv)

Big Data Technologien beschreiben eine neue Generationen von Technologien und Architekturen, die entworfen wurden um geschäftlichen Nutzen aus großen Datenvolumina mit unterschiedlichsten Datenarten/-typen zu ziehen und unterstützen schnelle Erfassung, Suche und/oder Analyse.

Die attributive Definition wird oft auch als **3-V-Modell** (Volume, Variety, Velocity) oder 4/5-V-Modell (zusätzlich Value und/oder Veracity = Unschärfe/Unsicherheit) bezeichnet.



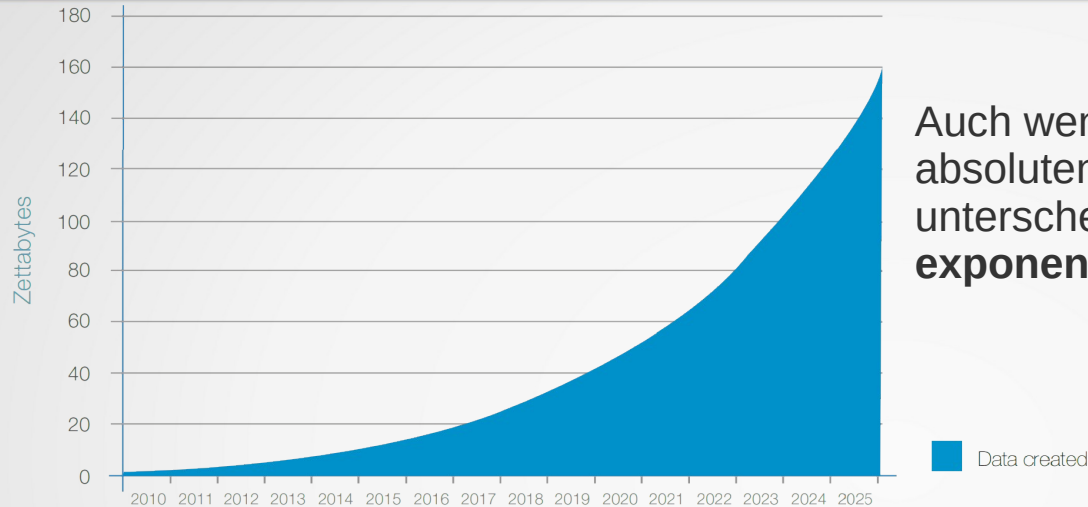
BigData - Volume

- Big Data beschreibt Datenbestände, die aufgrund ihres **Umfangs**, **Unterschiedlichkeit** oder ihrer **Schnellebigkeit** nur begrenzt durch traditionelle Datenbanken und Daten-Management-Tools verarbeitet werden können
- Big Data ist ein Synonym für die Bedeutung großer Datenvolumen in verschiedensten Anwendungsbereichen sowie der damit verbundenen Herausforderung, diese verarbeiten zu können
- Lt. IDC-Report¹⁾ **verdoppelt sich** die global erzeugte **Datenmenge alle 2 Jahre** – das bedeutet einen Anstieg um Faktor 300 von 2005 bis 2020 von 130 Exabytes auf 40,000 Exabytes²⁾
 - Der enorme Datenzuwachs in Wirtschaft, Forschung und privatem Umfeld ergibt sich aus der Digitalisierung von Inhalten, der Erfassung von Daten in digitalen Mess-, Steuer- und Kommunikationssystemen sowie dem Austausch dieser Daten in „intelligenten Umgebungen“. Dazu zählen z.B. die permanente Generierung von Daten durch die Nutzung von Mobiltelefonen, intelligenten Steuerungen im Heimbereich, oder Sensorsysteme zur Steuerung von Produktionsanlagen.

²⁾ 1 Exabyte = eine Milliarde Gigabyte, eine Million Terabyte, Tausend Petabyte

¹⁾ J. Gantz and D. Reinsel, "The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east," in Proc. IDC iView, IDC Anal. Future, 2012.

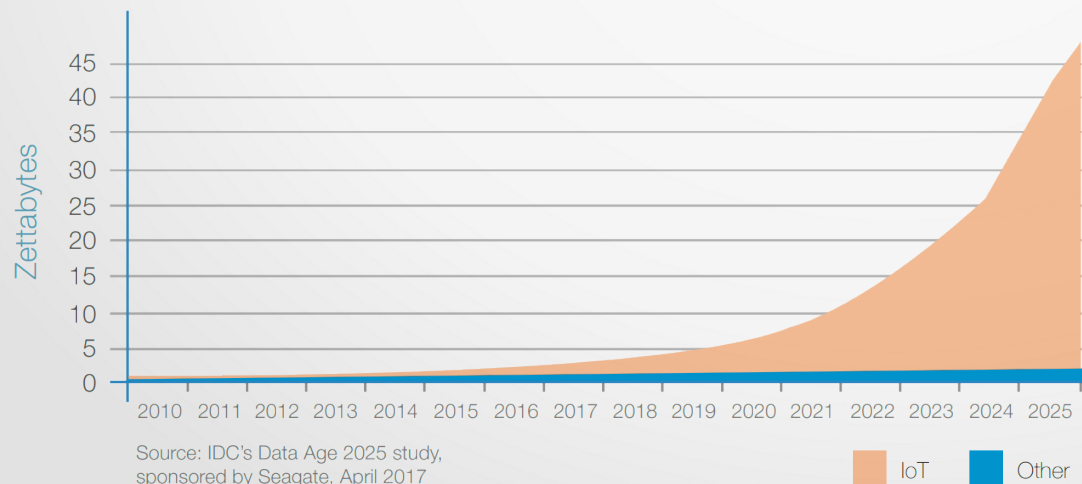
BigData – Volume



Source: IDC's Data Age 2025 study, sponsored by Seagate, April 2017

Auch wenn sich viele Studien in den absoluten Werten (geringfügig) unterscheiden, so zeigt sich in allen der **exponentielle** Zuwachs an Daten!

Volume (Umfang, Datenvolumen): Die anfallende Datenmenge steigt überproportional, man geht im Moment von einer Verdopplung im Zeitraum von zwei Jahren aus

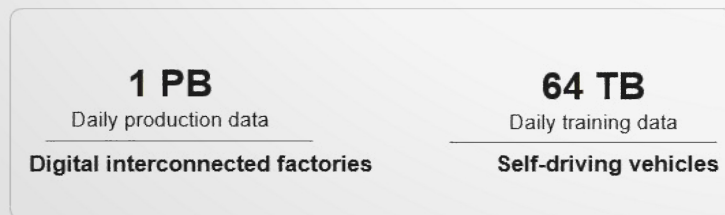
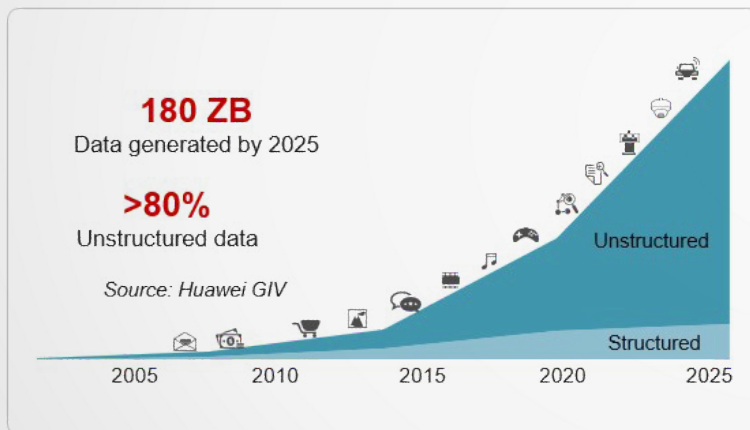


Source: IDC's Data Age 2025 study, sponsored by Seagate, April 2017

Der Datenzuwachs entfällt zu einem Großteil auf das Internet of Things (smart devices, smart home/city, Industrie 4.0, ...) – damit steigt in Summe auch die Unsicherheit (**Veracity**) der Daten (bspw durch falsche/ ungenaue Sensorwerte)

BigData – Variety

Variety (Vielfalt): Die Anzahl unterschiedlicher Datenquellen (und vor allem unstrukturierter Daten) nimmt stark zu (mit Einführung des 5G Mobilfunkstandards rechnet Huawei, dass sich das Datenaufkommen für die Netzbetreiber etwa verachtfacht)



2021 This Is What Happens In An Internet Minute



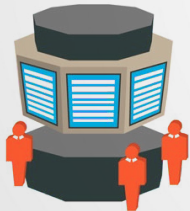
BigData – Variety (Datenstrukturklassen)

- Daten werden von ihrem Strukturierungsgrad her in drei Klassen untergliedert:
 - Strukturierte Daten
 - Unstrukturierte Daten
 - Semistrukturierte Daten
- **Strukturierte Daten:**
 - Daten mit einem vorgegebenen Format/Schema => speichereffizient (aber unflexibel!)
 - können leicht durchsucht/analysiert und wiederverwendet werden
 - Bsp.: Tabelle in einer relationalen Datenbank
- **Unstrukturierte Daten:**
 - Keine formalisierte Struktur => sehr flexibel und gut geeignet für heterogene Daten
 - Bsp.: Bilder, Videos, Texte
- **Semistrukturierte Daten:**
 - haben i.d.R. eine implizite bzw. “versteckte” Struktur, die aber nicht notwendigerweise immer eingehalten werden muss
 - Bsp.: XML-Dokument (ohne Schema/DTD)

BigData - Velocity

Die **Velocity** (Geschwindigkeit) mit der die Daten einerseits anfallen und andererseits verarbeitet werden steigt massiv an.

The New York Stock Exchange captures
1 TB OF TRADE INFORMATION
during each trading session



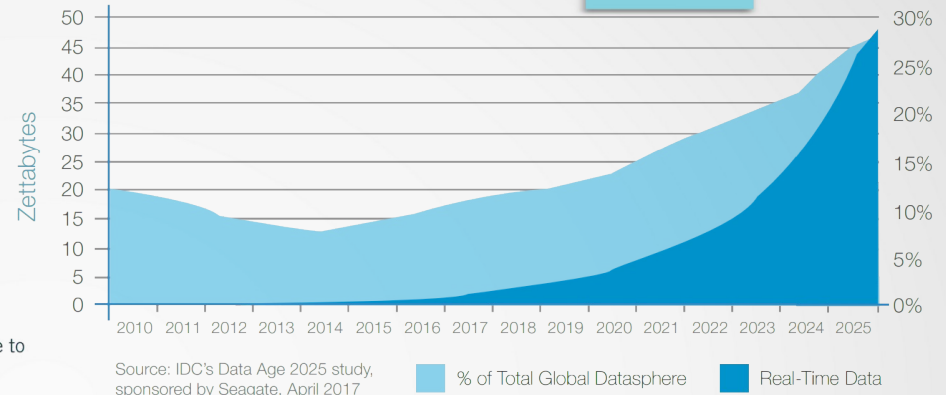
By 2016, it is projected there will be
18.9 BILLION NETWORK CONNECTIONS

– almost 2.5 connections per person on earth

Velocity
ANALYSIS OF
STREAMING DATA



Modern cars have close to
100 SENSORS
that monitor items such as fuel level and tire pressure



Beispiele: Entscheidungen auf Finanzmärkten, Anpassung von Produktionsparametern aufgrund von Messdaten, Steuerung eines autonomen Fahrzeugs auf Basis der Sensordaten, etc.

<https://www.ibmbigdatahub.com/infographic/extracting-business-value-4-vs-big-data>

BigData – Historische Entwicklung: Database Machine → Shared Nothing Architecture → GFS & MapReduce

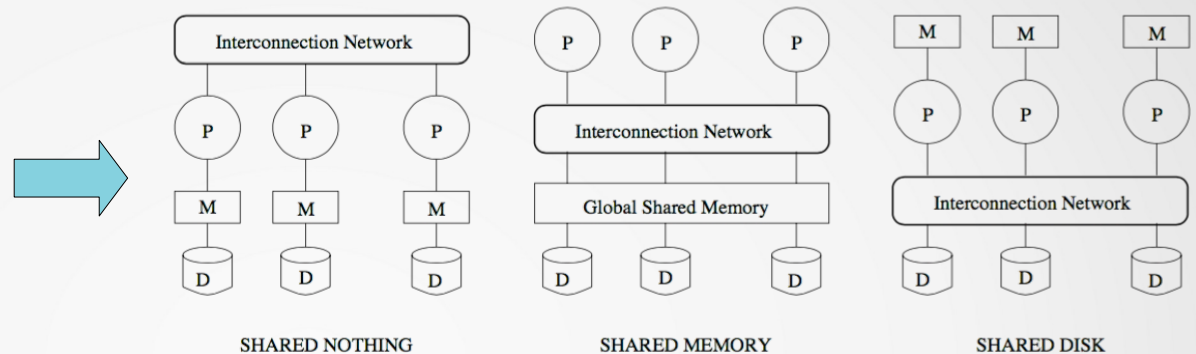
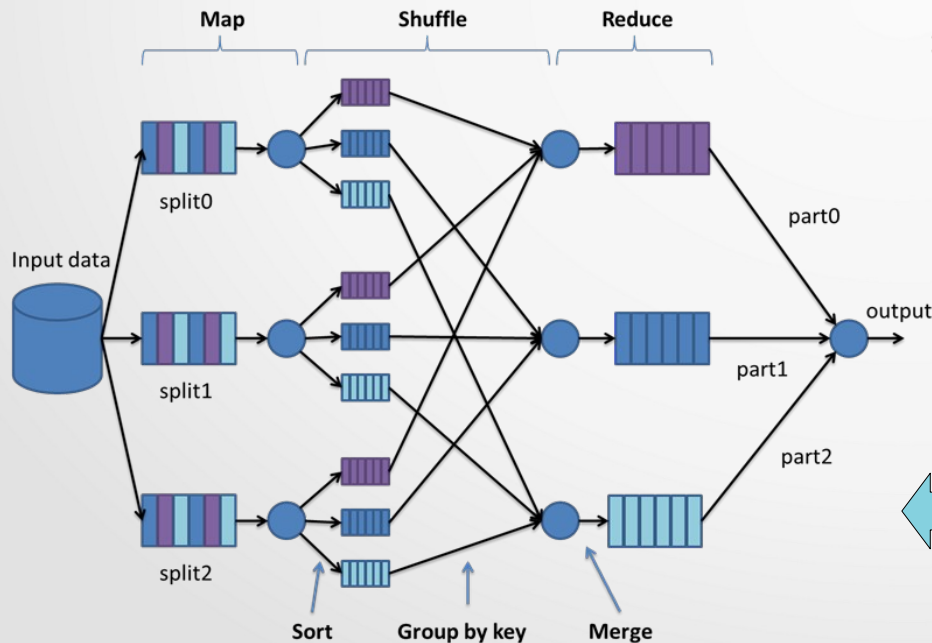
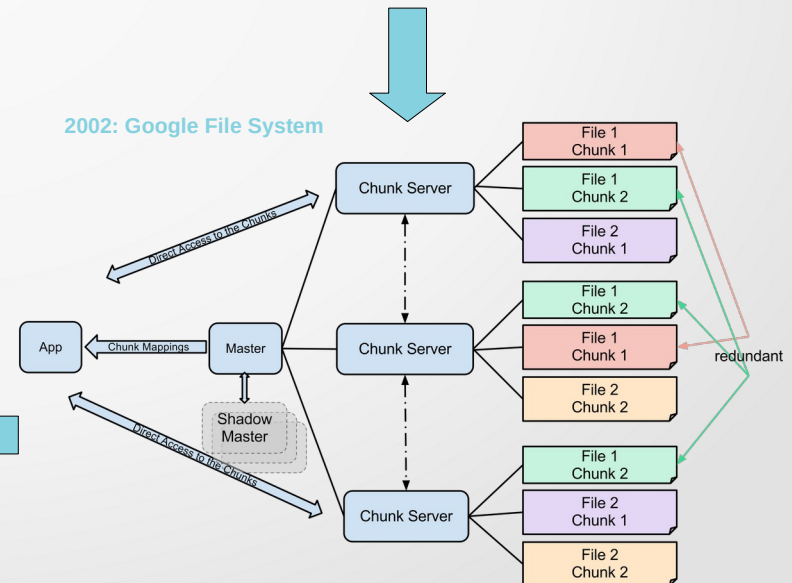


Figure 21.1 Physical Architectures for Parallel Database Systems

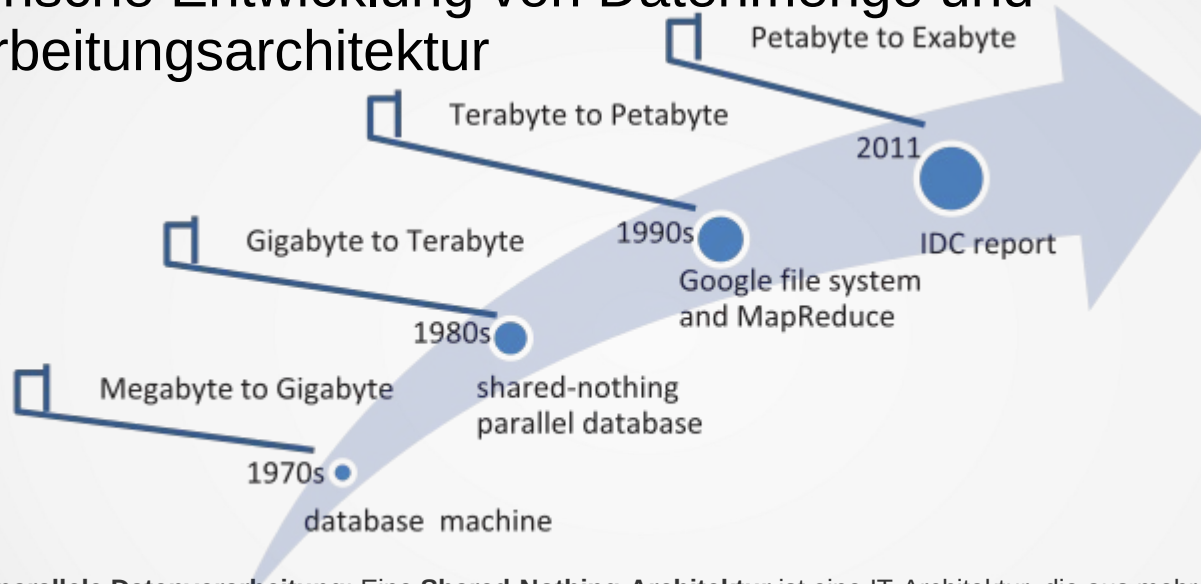


2002: Google File System



BigData – Historische Entwicklung

- Historische Entwicklung von Datenmenge und Verarbeitungsarchitektur

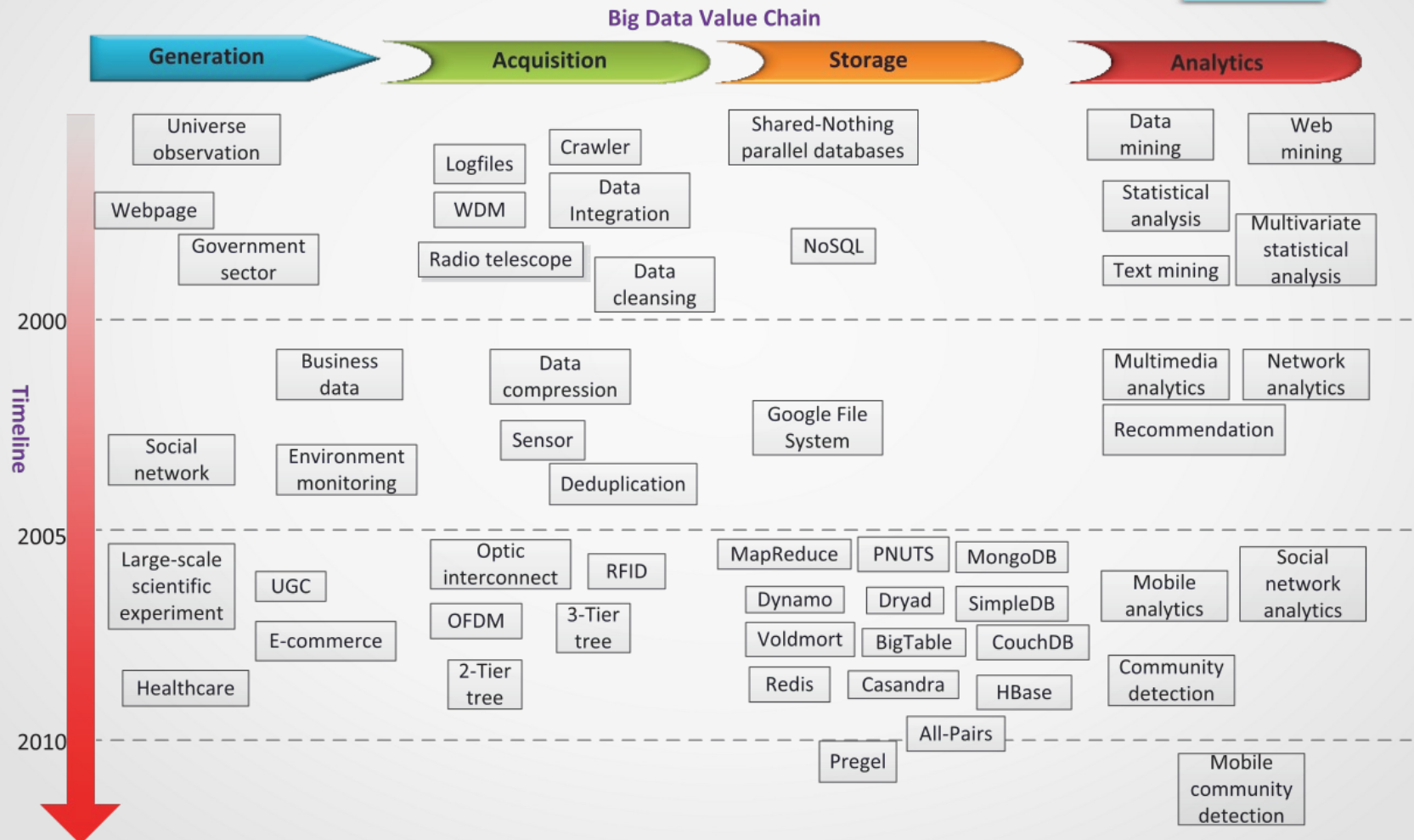


Architekturen für parallele Datenverarbeitung: Eine **Shared-Nothing-Architektur** ist eine IT-Architektur, die aus mehreren autonomen Knoten besteht, die ein verteiltes System bilden. Dabei hat jeder Knoten eigene Ressourcen wie Prozessorleistung, Speicherplatz, Betriebssystem und Arbeitsspeicher und ist dadurch in der Lage, Aufgaben selbstständig – ohne Beteiligung anderer Knoten – zu bearbeiten. Durch dieses Funktionsprinzip ist es möglich, viele einzelne Aufgaben oder Teilaufgaben gleichzeitig zu erledigen. Diese Architektur erlaubt eine praktisch grenzenlose horizontale Skalierung.

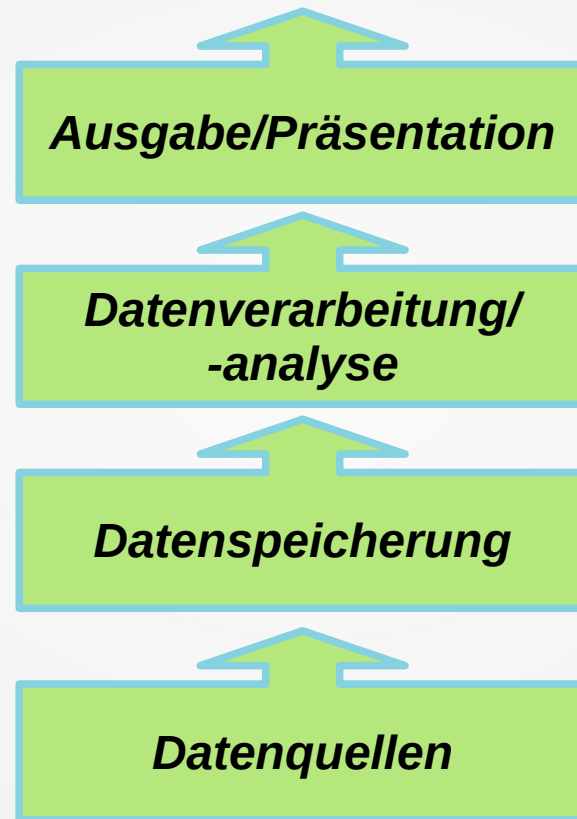
In einer **Shared-Disk-Umgebung** hat zwar jedes Teilsystem einen eigenen Prozessor und Hauptspeicher, es wird aber auf gemeinsame Laufwerke zugegriffen. In dieser Umgebung ist es nicht möglich, dass einzelne Knoten die gleichen Daten auf dem Laufwerk parallel bearbeiten. Um die Konsistenz der Daten sicherzustellen, hat die Shared-Disk-Architektur Mechanismen vorzusehen, die gleichzeitiges Verändern von Daten unterbindet.

Das prinzipiell gleiche Problem entsteht in einer **Shared-Memory-Architektur**. Hier existiert nur ein gemeinsamer Arbeitsspeicher. Auf gleiche Datenbereiche innerhalb des Speichers kann immer nur ein Prozess zugreifen. In der Shared-Disk- oder der Shared-Memory-Architektur bilden Disk oder Memory die jeweils leistungsbegrenzende Ressource für das Gesamtsystem und schränken die Skalierbarkeit ein.

BigData – Historische Entwicklung: Technologien entlang der Wertschöpfungskette



BigData – logische Schichten



BigData – logische Schichten

- **Datenquellen-Schicht:**

- Enthält alle (notwendigen/relevanten) Datenquellen
- Bsp.: Sensordaten, operative Daten, Dokumente, Kundendaten, Geodaten, Medizinische Daten (CT-/MR/Röntgen-Aufnahmen), Daten aus unterschiedlichsten Kommunikationskanälen, ...

- **Datenspeicherungsschicht:**

- Hier werden die gesammelten Daten abgelegt
- Dazu werden nicht-relationale Systeme (oftmals auf der NoSQL-Architektur basierend) kombiniert mit verteilten Speichersystemen (Hadoop FS, GFS/Colossus) verwendet
- Bsp.: HBase (zB auf Basis Hadoop FS), MongoDB, Cassandra (entwickelt von/für Facebook – dzt. zB verwendet von Twitter), Dynamo DB (Amazon), ...

BigData – logische Schichten

- **Datenverarbeitungs/-analyse-Schicht:**

- Hier werden die Daten aufbereitet und verarbeitet
 - Bei der *Stream-basierten Verarbeitung* werden (große) laufend ankommende Datenmengen (z.B. von Sensoren) permanent analysiert (real-time processing)
 - dabei können bspw wichtige Informationen gefiltert werden ohne den gesamten Datenstrom speichern zu müssen und
 - man kann auf bestimmte Informationen unmittelbar reagieren
 - Bei der *Batch-basierten Verarbeitung* werden die Daten zuerst gespeichert und danach analysiert (bekanntester Vertreter dieses Paradigmas ist die MapReduce-Vorgehensweise)
- Ziel ist neue wertvolle Informationen aus dem Datenberg ableiten zu können
- Bsp.: Apache Hive (transformiert SQL-Abfragen in MapReduce-Code), Apache Pig (einfache Skriptsprache zur Datenanalyse – wird automatisch in MapReduce-Code transformiert), ...

- **Ausgabe-/Präsentationsschicht:**

- Hier werden die gewonnenen Informationen in geeigneter Form (Diagramme, Berichte, ...) für die jeweilige Zielgruppe dargestellt

BigData - Herausforderungen

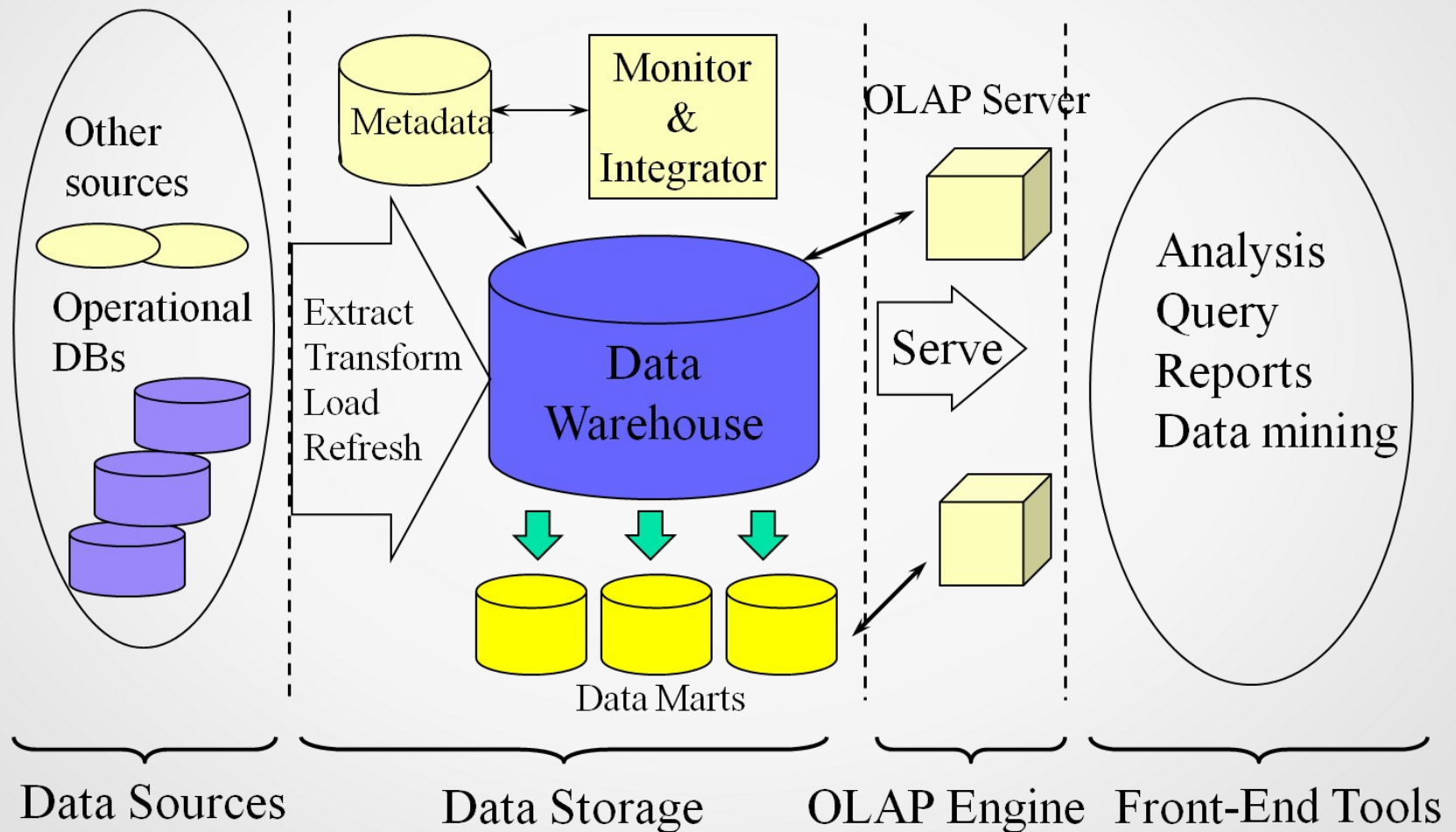
- **Heterogene Daten:** Die verfügbaren Daten unterscheiden sich in Struktur, Semantik, Organization, Granularität und Zugriffsmöglichkeiten
- **Vermeidung von Redundanzen**
- Datenkomprimierung ohne wesentliche Daten zu verlieren
- **Priorisierung der Daten** nach Aktualität und Relevanz
- **Datensicherheit und -schutz** (Privacy by Design)
- Approximative Analysen als mögliche Lösung für real-time Datenverarbeitung
- **Energieeffizienz** und ökonomischer Betrieb der Hardware-Cluster (GreenIT)
- ...

BigData – Exkurs: Abgrenzung zu traditioneller Datenhaltung

- Relationale Datenbanken
 - sind weitverbreitet und stellen einen etablierten und traditionellen Ansatz zur Verwaltung von Daten dar, verwalten Daten v.a. in strukturierter Form (in statischen – also nicht dynamisch anpassbaren, dh unflexiblen - Tabellen/Schemata) und bieten **wenig wertvolle Unterstützung für unstrukturierte** bzw. semi-strukturierte **Daten** an
 - Bieten ein hohes Maß an **Datenkonsistenz** (ACID-Prinzip, Constraints, Transaction[s] [Isolation]) => diese Eigenschaft ist für viele Big-Data-Anwendung nur eingeschränkt erforderlich
 - Werden va vertikal skaliert - **horizontale Skalierung** ist aber schwierig/aufwändig (insbesondere bei vollständiger Gewährleistung der Datenkonsistenz) => Big Data wird oft über Rechner(knoten)netzwerke verteilt gespeichert/verwaltet
 - Bei (der Skalierung kommerzieller) RDBMS-Lösungen können hohe Lizenz- und Wartungskosten entstehen – BigData-Lösungen basieren daher oft auf einem Open-Source-Stack (unter Verwendung sogenannter Commodity-Hardware)
- In Abgrenzung zu existierenden Business Intelligence (BI) und Data Warehouse Systemen, arbeiten Big Data Anwendungen in der Regel ohne aufwändige Aufbereitung (ETL-Prozess¹⁾) der Daten

¹⁾ ETL steht für Extrahieren, Transformieren und Laden von Daten aus einem oder mehreren Quellsystemen in einen Zieldatenbestand inkl. Data Cleansing. ETL-Systeme bilden beim Data Warehousing die Datenschnittstelle zwischen operativen / externen Datenbeständen und Data Warehouse / Data Marts.

BigData – Exkurs: Data Warehouse



BigData – Exkurs: NoSQL-Datenbanken

- Für BigData-Anwendungen kommen oftmals **NoSQL** (= Not-only SQL) Datenbanken zum Einsatz
- Daten werden hier **nicht** mehr **normalisiert** und in der Regel wird auf ein starres Datenbankschema verzichtet
- Fokus liegt auf **Skalierbarkeit** und hoher **Performance**
 - Konsistenz der Daten wird (oft) nicht garantiert (ACID-Prinzip¹⁾)
 - Transaktionen werden nur eingeschränkt oder überhaupt nicht unterstützt
- Sind idR mit verteilten Dateisystemen kompatibel bzw. **“partition tolerant”** (dh die Systeme funktionieren auch über/auf physische/n Netzwerkpartitionen)
- Können in 4 Klassen unterteilt werden: spaltenorientiert, Schlüssel-Wert-orientiert, dokumentorientiert und graphenbasiert

¹⁾ **Atomicity:** Eine Transaktion ist eine Folge von Datenbank-Operationen, die entweder ganz oder gar nicht (Rollback) ausgeführt wird

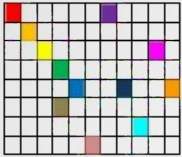
Consistency: heißt, dass eine Transaktion nach Beendigung einen konsistenten Datenbankzustand hinterlässt, falls die Datenbank davor auch konsistent war (alle Integritätsbedingungen müssen erfüllt sein)

Isolation: verhindert bzw schränkt die gegenseitige Beeinflussung sich glz in Ausführung befindlicher Transaktionen ein (Isolationsniveaus wie READ_COMMITTED, REPEATABLE_READ oder SERIALIZABLE)

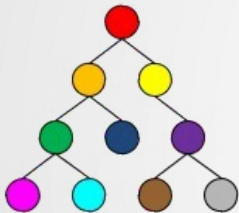
Durability: sagt aus, dass Daten nach dem erfolgreichen Abschluss einer Transaktion garantiert dauerhaft in der Datenbank gespeichert sind (bspw auch bei einem schwerwiegenden Fehler des DBMS oder einem Hauptspeicherproblem)

BigData – Exkurs: NoSQL-Datenbanken / CAP-Theorem

Column-Family



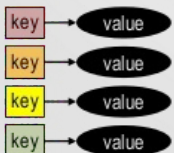
Document



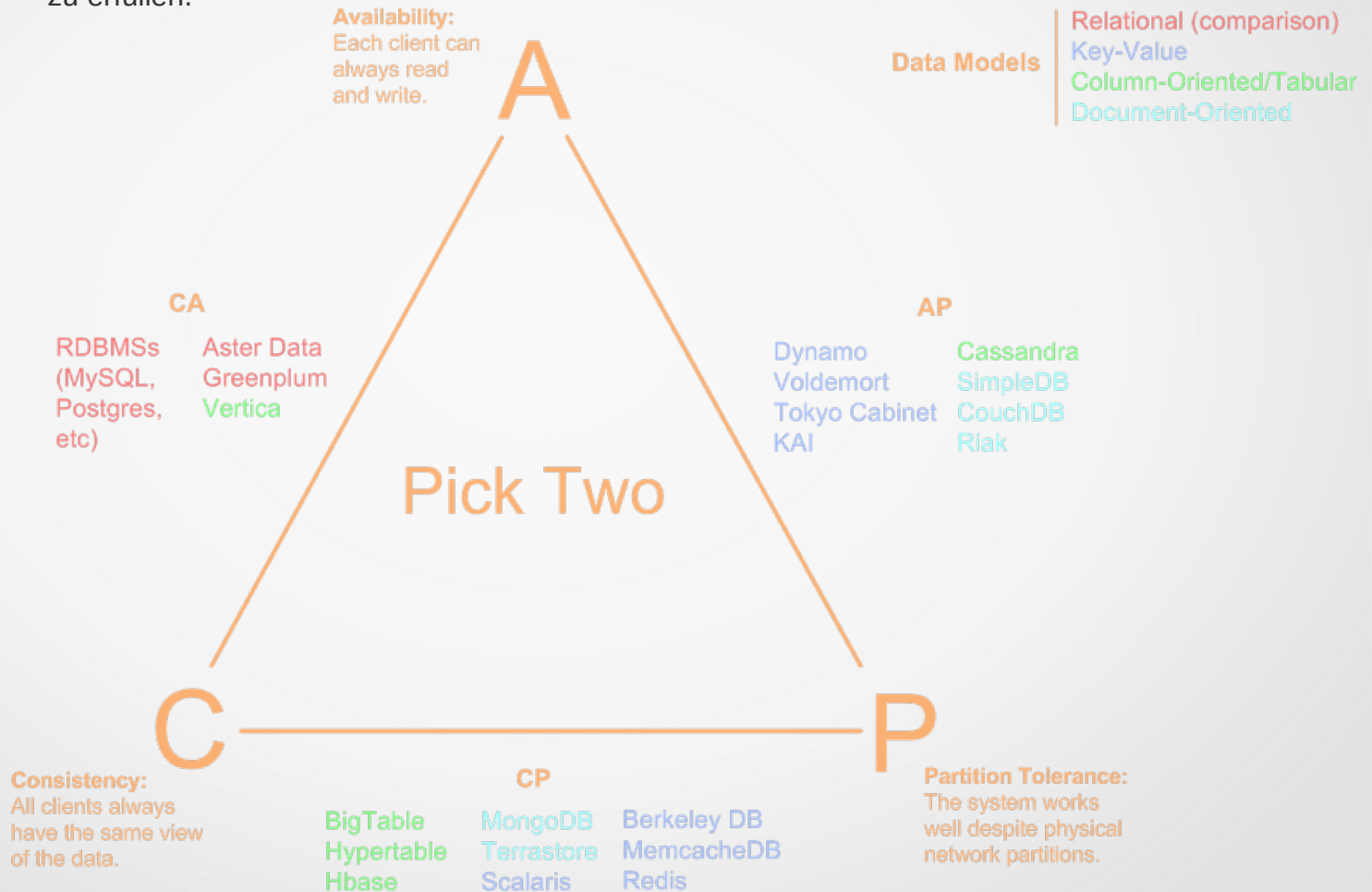
Graph



Key-Value



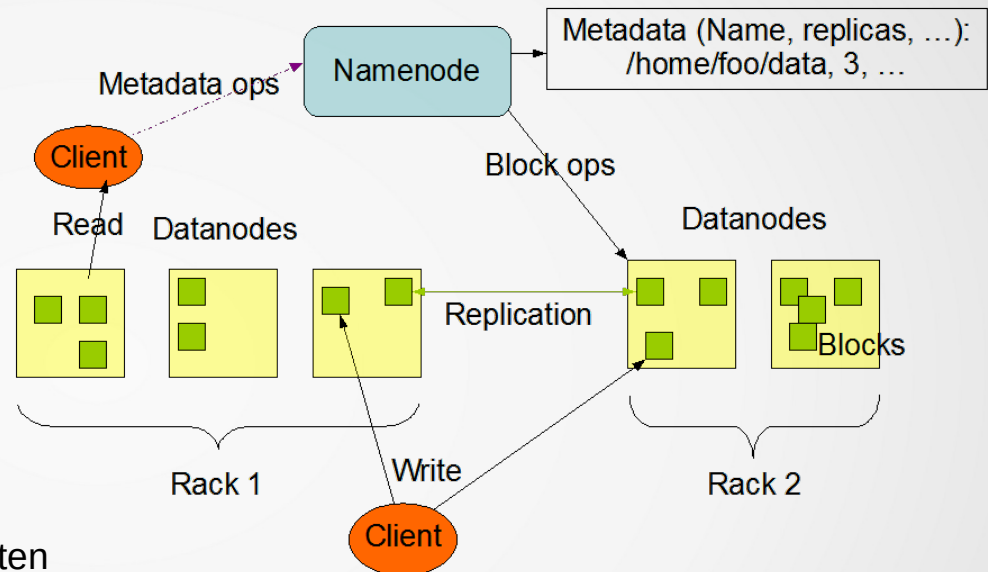
Das **CAP-Theorem** (nach Eric Brewer) besagt, dass es für eine verteilte Datenbank unmöglich ist gleichzeitig alle 3 unten angeführten Garantien zu erfüllen.



BigData – Exkurs: Hadoop File System (HDFS)

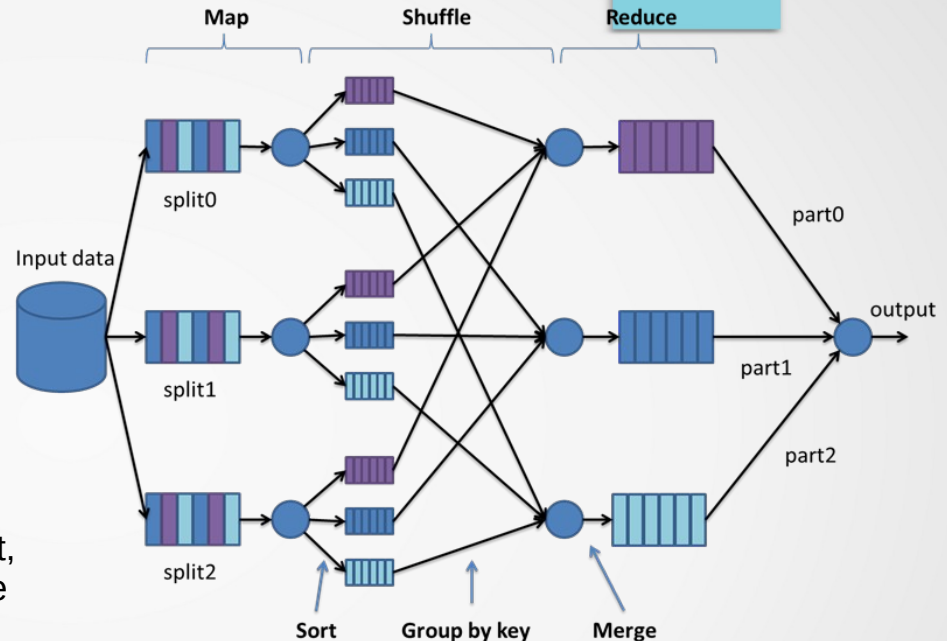
HDFS Architecture

- **Verteilte Dateisysteme** speichern Daten über viele/mehrere Knoten eines Netzwerks
- Dazu wird eine Datei in **Blöcke** bestimmter Größe geteilt und auf den einzelnen **Knoten** (sog. Datanodes) verteilt
- Ein **Namenode** verwaltet sämtliche Metadaten (bspw. auf welchen Datanodes sind die Blöcke einer Datei aufgeteilt) und kümmert sich zusätzlich auch um eine **redundante Datenhaltung** (idR wird ein Block auf 3 verschiedenen Datanodes abgelegt). Weiters kümmert sich der Namenode auch darum laufend zu prüfen, ob die Datanodes überhaupt noch im Netzwerk verfügbar sind (und sendet dazu sog. **Heartbeats** aus) und sorgt für eine gleichmäßige Aufteilung der Daten auf den einzelnen Datanodes.
- Namenodes stellen einen Single-Point-of-Failure dar und sind daher in der Regel redundant ausgeführt



BigData – Exkurs: MapReduce

- **MapReduce** ist ein wichtiges/zentrales **Datenverarbeitungsparadigma** im Bereich BigData
- **Parallelisiert** die Verarbeitung von Daten, wobei Rechenkapazitäten nahe am Entstehungsort der Daten zugeteilt werden, um den Übertragungs-overhead möglichst gering zu halten.
- Arbeitet in **3 Phasen**:
 - **Map**: Die Eingabedaten werden (in chunks) auf eine Menge von (idealerweise parallel ablaufenden) Map-Prozessen verteilt, welche jeweils die vom Nutzer bereitgestellte Map-Funktion ausführen/berechnen (Ergebnisse sind *Schlüssel-Wert-Paare*)



- **Shuffle & Sort**: Es werden nun die Werte eines Schlüssels zu Listen zusammengefasst/aggregiert (es entstehen also *Schlüssel-Wertelisten-Paare*) und die Schlüssel sortiert
- **Reduce**: Die Werte der Wertelisten werden hier (je nach vom Nutzer bereitgestellter Reduce-Funktion) bspw. aggregiert und in der Form *Schlüssel-Wert_aggregiert* ausgegeben (auch hier kann/soll es zu einer parallelisierten Verarbeitung kommen)
- Das MapReduce-Modell ist einfach und in unterschiedlichsten Bereichen (Bioinformatik, WebMining, machine learning, etc.) weitverbreitet.

BigData – Exkurs: MapReduce - Beispiel

Ziel: Berechnung der Gesamtumsätze je Region/Standort

Chunk 1

2012-01-01 – 12:01 – San Jose – Music – 12.99 – Amex

2012-01-01 – 12:05 – NYC – Clothes – 76.99 – Visa

Chunk 2

2012-01-01 – 13:01 – Miami – Electronics – 24.45 – Mastercard

2012-01-01 – 14:01 – NYC – Music – 12.99 – Amex

Chunk 3

2012-01-01 – 14:03 – NYC – Music – 13.99 – Diner's Club

2012-01-01 – 12:01 – Miami – Music – 12.99 – Amex

....

Map-Job #1

San Jose – 12.99

NYC – 76.99

Miami – 24.45

NYC – 12.99

NYC – 13.99

Miami – 12.99

....

Map-Job #2

San Jose – 12.99

Miami – 7336.99

Miami – 24.45

LA – 12.99

NYC – 13.99

LA – 12.99

....

...

Map-Job #n

Tampa – 12.99

LA – 76.99

LA – 24.45

Miami – 12.99

Tampa – 13.99

NYC – 12.99

....

Shuffle & Sort

LA – 12.99, 12.99, 3.45, ...

Miami – 76.99, 2.66, 84.30

NYC – 24.45, 121.77, 55.33

San Jose – 12.99, 445.99, 33.09

Tampa – 13.99, 11.77, 345.09

Reduce-Job #1

LA – 12343454354.99

Miami – 14324242.36

Reduce-Job #2

NYC – 13666532.20

...

Reduce-Job #n

San Jose – 124324234.99

Tampa – 1323122313.99

Phase 1 "Map":

Die Eingabedaten (Rohdaten zB aus einem Log) werden (in Chunks) auf eine Menge von (idealerweise parallel ablaufenden) Map-Prozessen verteilt, welche jeweils die vom Nutzer bereitgestellte Map-Funktion ausführen/berechnen (Ergebnisse sind Schlüssel-Wert-Paare, hier: <Standort>: <Umsatz>)

Phase 2 "Shuffle & Sort":

Es werden nun die Umsätze jedes Standorts (zu einer Liste) zusammengefasst (es entstehen also Schlüssel-Wertelisten-Paare) und die Schlüssel (=Standorte) sortiert

Phase 3 "Reduce":

Die Umsätze je Standort werden hier (parallelisiert) aggregiert (nämlich addiert)