Big Data Anwendungen

Technische Lösungen

Ausblick

- Ziel
  - Verständnis der Datenhaltung um Auswertungen „effizient“ zu realisieren
  - Übersicht über Big Data Technologien

- Herausforderung
  - Ursprung des Themas: 1970er [für Informatik Themen uralt]
  - Ursprüngliche Konzepte und Grundlagen beeinflussen moderne Systeme
  - Schnellüberblick über verschiedene Lösungen
    (würde auch ausreichen, um mehrere Vorlesungen zu füllen)

- Themen
  - Relationale Datenbanken
    - Speicherung der Daten in Tabellen (ähnlich Excel)
    - SQL als populäre Anfragesprache (auch für Big Data)
  - Data Warehouse
    - Konkurrent zu Big Data Clustern
    - Versuch der Abgrenzung zu Big Data
  - Hadoop

Agenda

- Einführung
- Deskriptive Methoden zur Datenexploration
- Datenqualität
- Klassifikation
- Recommender Systems
- Clusteringverfahren
- Stream Mining
- Social Network Analysis

- Technische Lösungen
  - Relationale Datenbanken
  - Data Warehouse
  - Datenanalyse und Reporting
  - MapReduce
  - Hadoop
  - Big Data Warehouse

- Datenschutz und gesellschaftliche Aspekte

Relationale Datenbank – Idee

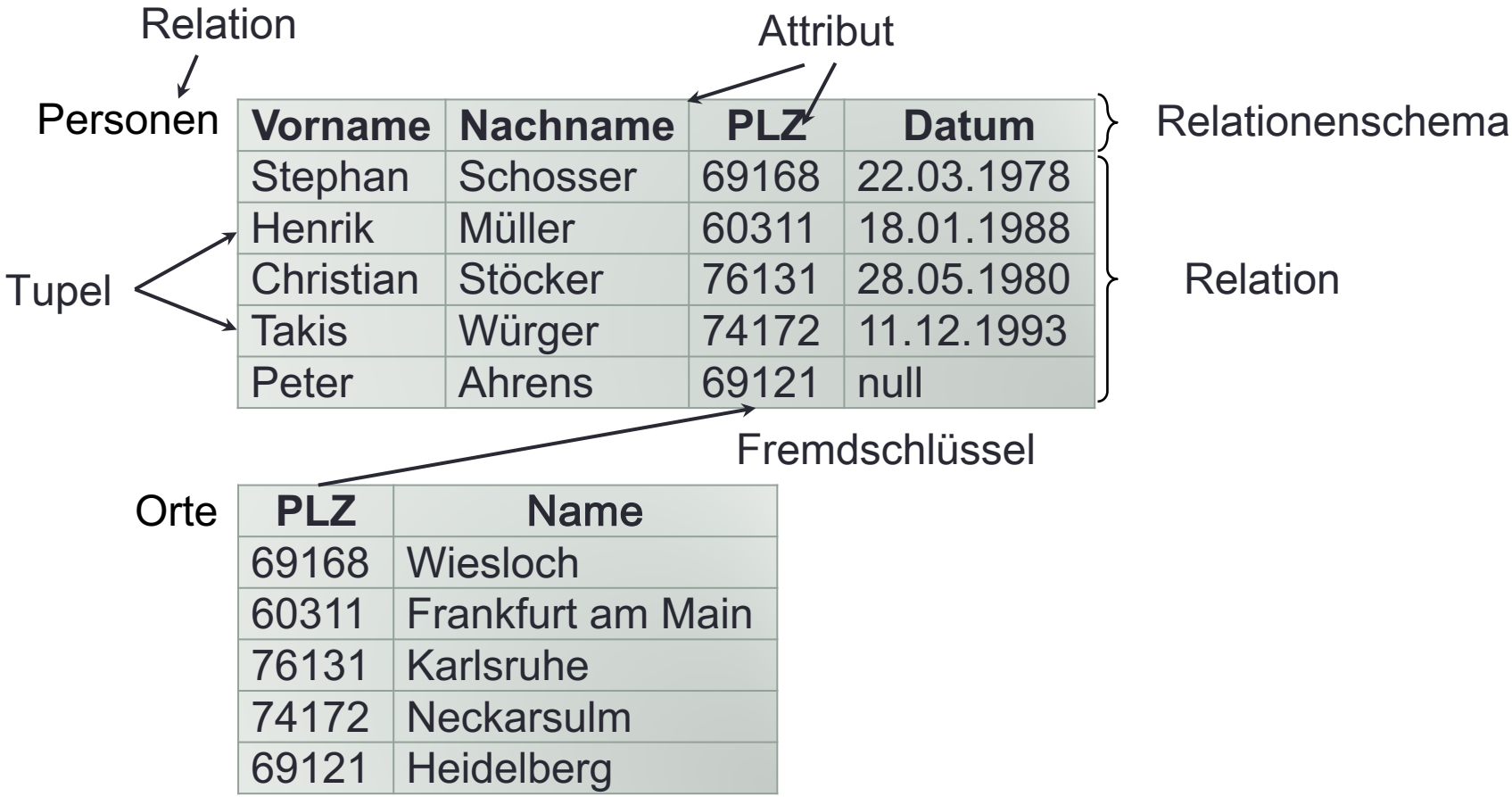
- Aufgabe einer Datenbank...
  ... analog zu einer „richtigen“ Bank
  - Langfristige Aufbewahrung von Werten (hier: Daten)
  - Sicherheit vor Verlusten der Werte
  - Dienstleistung um effizient mit Werten zu arbeiten

- Verbreitung
  - Relationale Datenbanken existieren seit den 1970ern
  - Operative Daten in Unternehmen meist in Datenbanken
  - Erlauben insbesondere für strukturierte Daten effiziente Speicherung
  - Grundprinzipien der Datenbanken in allen modernen Systemen enthalten

- Ursachen für Verbreitung
  - Abfrage von Daten standardisiert (und relativ einfach)
  - Migration zwischen unterschiedlichen Lösungen möglich
  - Viele Lösungen Open Source verfügbar
    (z.B. PostgreSQL oder MariaDB)

Relationen

- Datenbanken bestehen aus Relationen



Anfragesprache – Projektion

- Definition
  Auswahl einzelner Spalten
- SQL-Anfrage
  SELECT Nachname, PLZ FROM Personen
- Beispiel

Personen

Vorname	Nachname	PLZ	Datum
Stephan	Schosser	69168	22.03.1978
Henrik	Müller	60311	18.01.1988
Christian	Stöcker	76131	28.05.1980
Takis	Würger	74172	11.12.1993
Peter	Ahrens	69121	null

Nachname	PLZ
Schosser	69168
Müller	60311
Stöcker	76131
Würger	74172
Ahrens	69121

Anfragesprache – Selektion

- Definition
  Auswahl einzelner Zeilen nach einem Filter
- SQL-Anfrage
  SELECT Nachname, PLZ FROM Personen WHERE PLZ < 70000
- Beispiel

Personen

Vorname	Nachname	PLZ	Datum
Stephan	Schosser	69168	22.03.1978
Henrik	Müller	60311	18.01.1988
Christian	Stöcker	76131	28.05.1980
Takis	Würger	74172	11.12.1993
Peter	Ahrens	69121	null

Nachname	PLZ
Schosser	69168
Müller	60311
Ahrens	69121

Anfragesprache – Verbund

- Definition
  Verknüpfung zweier Tabellen
- SQL-Anfrage
  SELECT Nachname, Ort FROM Personen JOIN Orte
- Beispiel

Personen

Vorname	Nachname	PLZ	Datum
Stephan	Schosser	69168	22.03.1978
Henrik	Müller	60311	18.01.1988
Christian	Stöcker	76131	28.05.1980
Takis	Würger	74172	11.12.1993
Peter	Ahrens	69121	null

Orte

PLZ	Name
69168	Wiesloch
60311	Frankfurt am Main

Nachname	Name
Schosser	Wiesloch
Müller	Frankfurt am Main

Anfragesprache – Kartesisches Produkt

- Definition
  Verknüpfung zweier Tabellen
- SQL-Anfrage
  SELECT Nachname, Ort FROM Personen, Orte
- Beispiel

Personen

Vorname	Nachname	PLZ	Datum
Stephan	Schosser	69168	22.03.1978
Henrik	Müller	60311	18.01.1988
Christian	Stöcker	76131	28.05.1980
Takis	Würger	74172	11.12.1993
Peter	Ahrens	69121	null

Orte

PLZ	Name
69168	Wiesloch
60311	Frankfurt am Main

Nachname	Name
Schosser	Wiesloch
Schosser	Frankfurt am Main
...	...

Anfragesprache – Umbenennung

- Definition
  Umbenennung von Attributen
- SQL-Anfrage
  SELECT Nachname AS Name FROM Personen
- Beispiel

Personen

Vorname	Nachname	PLZ	Datum
Stephan	Schosser	69168	22.03.1978
Henrik	Müller	60311	18.01.1988
Christian	Stöcker	76131	28.05.1980
Takis	Würger	74172	11.12.1993
Peter	Ahrens	69121	null

Name	PLZ
Schosser	69168
Müller	60311
Stöcker	76131
Würger	74172
Ahrens	69121

Anfragesprache – Vereinigung

- Definition
  Anhängen von Tabellen an andere Tabellen
- SQL-Anfrage
  SELECT Nachname AS Name FROM Personen UNION SELECT Name FROM Orte
- Beispiel

Personen

Vorname	Nachname	PLZ	Datum
Stephan	Schosser	69168	22.03.1978
Henrik	Müller	60311	18.01.1988
Christian	Stöcker	76131	28.05.1980
Takis	Würger	74172	11.12.1993
Peter	Ahrens	69121	null

Orte

PLZ	Name
69168	Wiesloch
60311	Frankfurt am Main

Name
Schosser
Müller
...
Frankfurt am Main

Anfragesprache – Differenz

- Definition
  Entfernen von Elementen einer Tabelle aus anderer Tabellen
- SQL-Anfrage
  SELECT PLZ FROM Personen MINUS SELECT PLZ FROM Orte
- Beispiel

Personen

Vorname	Nachname	PLZ	Datum
Stephan	Schosser	69168	22.03.1978
Henrik	Müller	60311	18.01.1988
Christian	Stöcker	76131	28.05.1980
Takis	Würger	74172	11.12.1993
Peter	Ahrens	69121	null

Orte

PLZ	Name
69168	Wiesloch
60311	Frankfurt am Main

PLZ
76131
74172
69121

Agenda

- Einführung
- Deskriptive Methoden zur Datenexploration
- Datenqualität
- Klassifikation
- Recommender Systems
- Clusteringverfahren
- Stream Mining
- Social Network Analysis

- Technische Lösungen
  - Relationale Datenbanken
  - Data Warehouse
  - Datenanalyse und Reporting
  - MapReduce
  - Hadoop
  - Big Data Warehouse

- Datenschutz und gesellschaftliche Aspekte

Abgrenzung zu Relationalen Datenbanken

	OLTP (relationale DB)	OLAP (Data Warehouse)
Fokus	Lesen, Schreiben, Modifizieren, Löschen	Lesen, Wiederkehrendes Schreiben
Art der Transaktionen	Kurze Lese und Schreibe Transaktionen	Lange Lesezugriffe
Struktur der Anfragen	Einfach	Komplex
Datenvolumen pro Anfrage	Wenige Datensätze	Viele Datensätze
Datenvolumen pro System	Gigabyte (GB) – Terabyte (TB)	Terabyte (TB) – Petabyte (PB)
Alter der Daten	Aktuell bis wenige Wochen	Aktuell bis mehrere Jahre
Datenmodell	Flexibles Datenmodell	Fokus auf Analysen
Verfügbarkeit	Sehr hoch	Hoch
Anzahl der Nutzer	Hoch	Gering

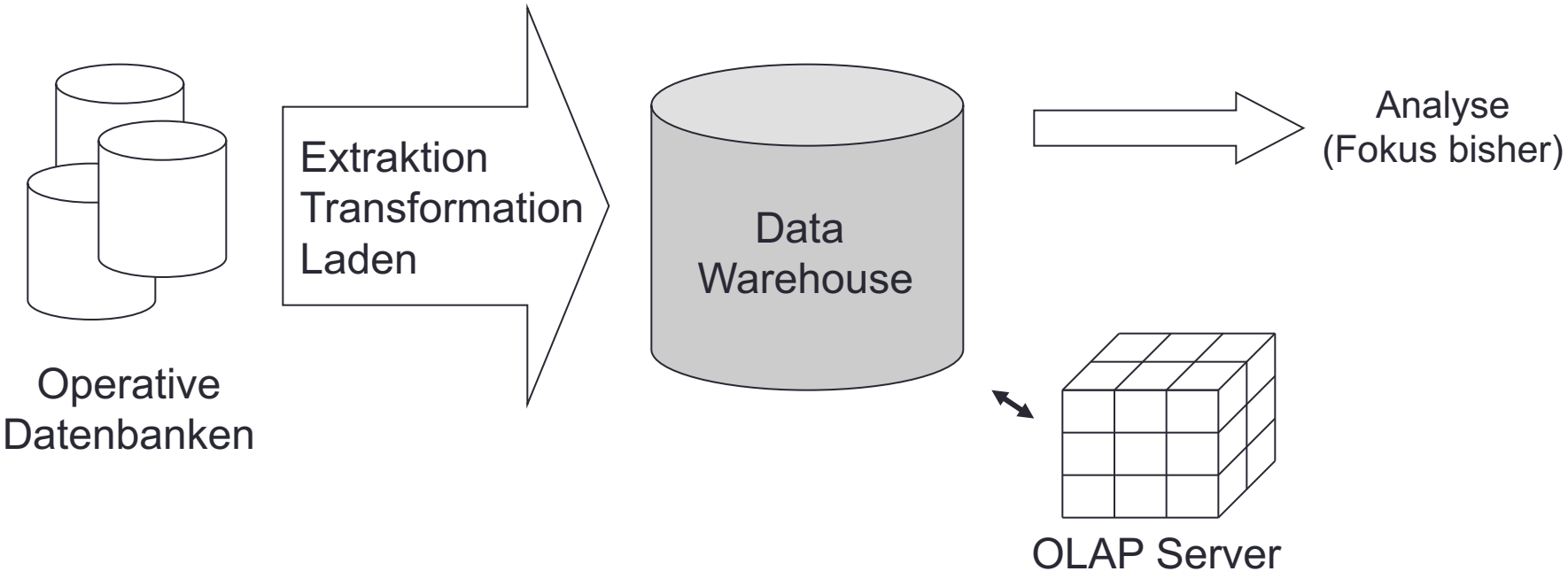
Eigenschaften eines Data Warehouse

- Integrierte Sicht auf beliebige Daten
  - … aus verschiedenen Datenbeständen
  - … Integration von Schemata und Daten aus Quellen
- Analyseaspekt
  - … multidimensionales Datenmodell
  - … Online Analytical Processing (OLAP)
- Stabile Datenbasis
  - Eingebrachte Daten werden nicht mehr modifiziert
  - Neue Daten können aufgenommen werden
- Data Warehouse System
  - Komponenten zur Integration und Analyse + Data Warehouse

Data Warehouse Prozess

- Monitoring
  Entdecken und melden von Änderungen in den Quellen
- Extraktion
  Selektion und Transport von Daten aus den Quellen in den Arbeitsbereich
- Transformation
  Vereinheitlichung, Bereinigung, Integration, Konsolidierung, Aggregation und Ergänzung der Daten im Arbeitsbereich
- Laden
  Laden der Daten aus dem Arbeitsbereich in die Basisdatenbank bzw. ins Data Warehouse
- Analyse
  Analyse und Präsentation der Daten im Data Warehouse

Vereinfachte Sicht auf die Referenzarchitektur



Agenda

- Einführung
- Deskriptive Methoden zur Datenexploration
- Datenqualität
- Klassifikation
- Recommender Systems
- Clusteringverfahren
- Stream Mining
- Social Network Analysis

- Technische Lösungen
  - Relationale Datenbanken
  - Data Warehouse
  - Datenanalyse und Reporting
  - MapReduce
  - Hadoop
  - Big Data Warehouse

- Datenschutz und gesellschaftliche Aspekte

Datenanalyse – Ansätze

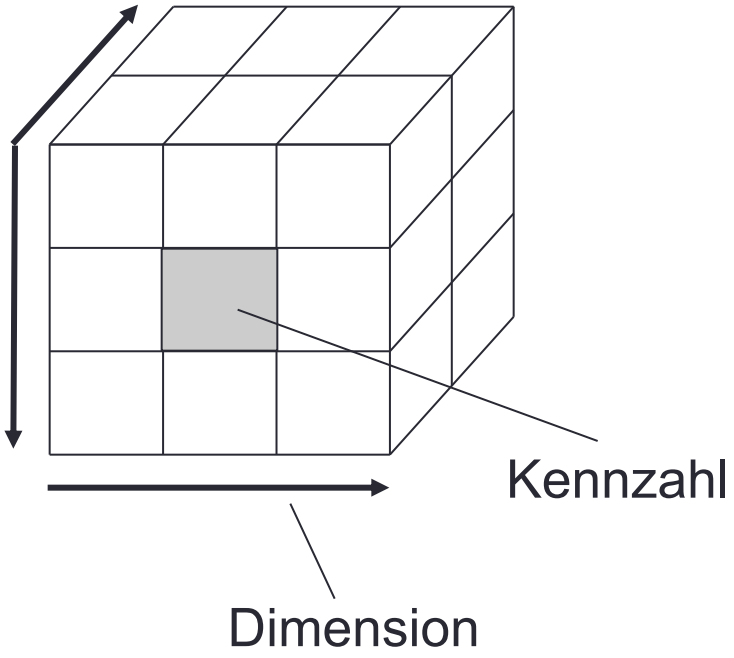
- Online Transactional Processing (OLTP)
  - Zugriff auf vorhandenen Datenbestand
  - Nutzung von Datenmanipulationssprachen (z.B. SQL)
  - „Wieviele Einheiten von Artikel X wurden in Filiale Y im Jahr Z verkauft?“
- Online Analytical Processing (OLAP)
  - Anpassung des Datenbestands an die Analyse
  - Suche nach neuen oder unerwarteten Beziehungen zwischen Variablen
  - „In welcher Stadt macht Produktgruppe X den größten Umsatz?“
- Data Mining
  - Suche nach Mustern im Datenbestand
  - „Wie ist die Entwicklung des Absatzes der Produktgruppen im Jahresverlauf?“

Multidimensionales Datenmodell - Begriffe

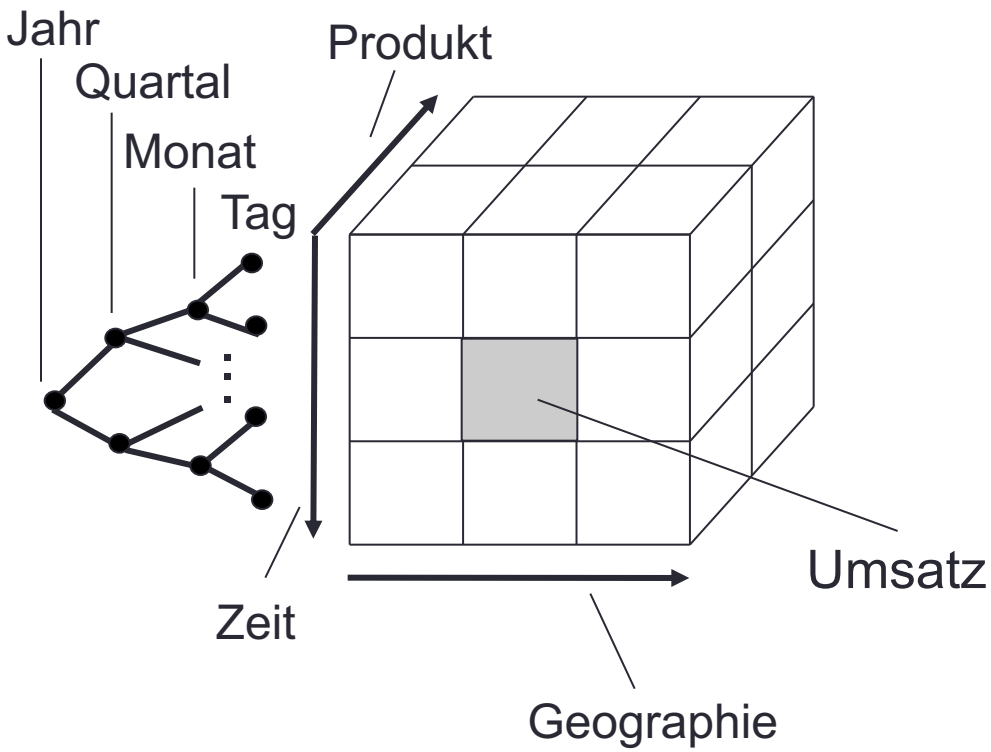
- Hilfsmittel zur Veranschaulichung von Daten
- verschiedene Aspekte auf gleiche Weise zugreifbar
- Einsatz bei OLAP Anwendungen

- Kennzahlen
  - Elemente eines Würfels

- Dimensionen
  - Beschreiben Daten
  - Ermöglichen Zugriff auf Kennzahlen
  - Können Hierarchien sein



Multidimensionales Datenmodell – Beispiel

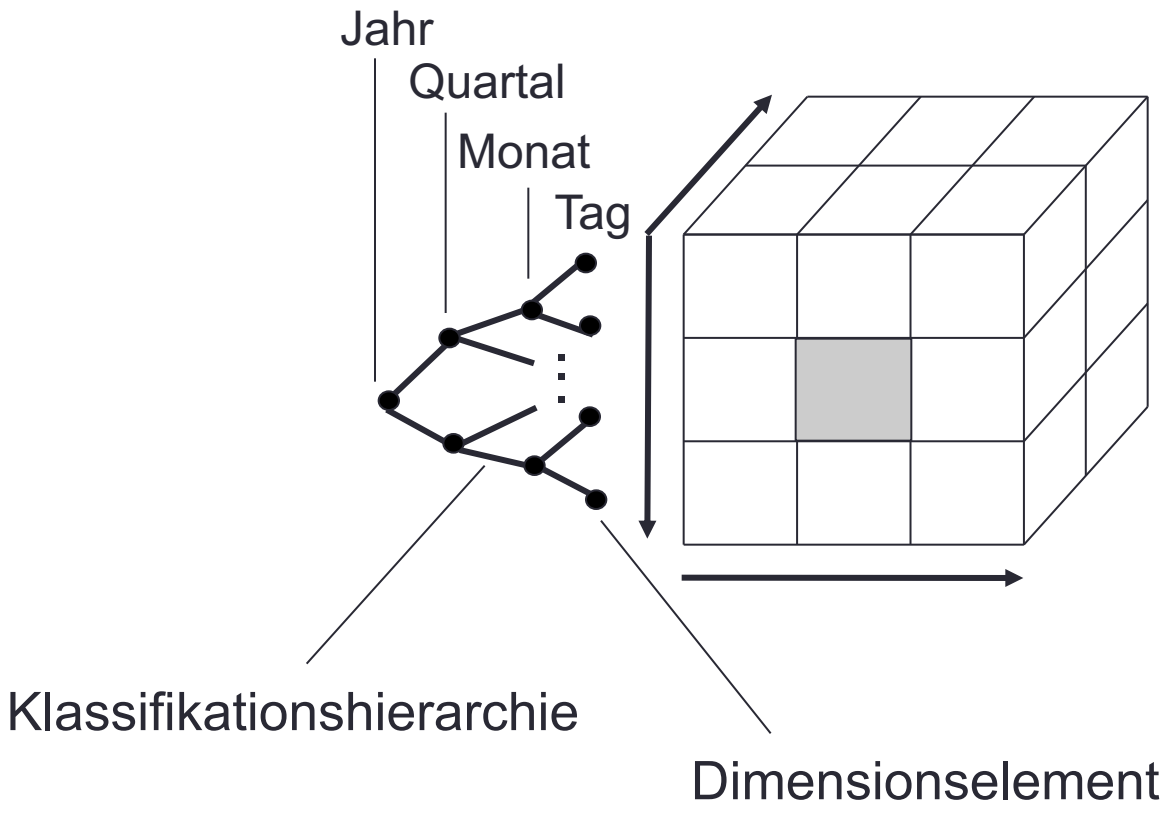


Dimensionen

- Einordnung
  - Bewertung der Analysedaten durch Kenngrößen (z.B. Umsatz, Kosten)
  - Untersuchung der Kenngrößen aus verschiedenen Perspektiven (z.B. Stadt, Bundesland, Zeitachse)
  - Betrachtungsperspektive heißt Dimension

- Eigenschaften
  - Mindestens 2 Dimensionselemente
  - Dimensionselemente Bilden Blätter eines Baums (sog. Klassifikationshierarchie)

Dimensionen – Beispiel Zeit

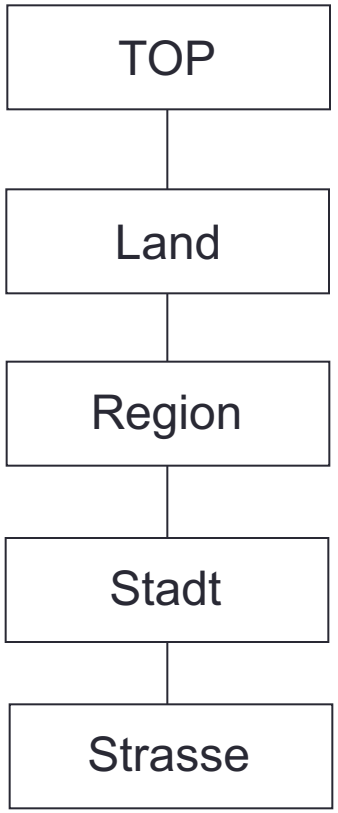


Arten von Klassifikationshierarchien

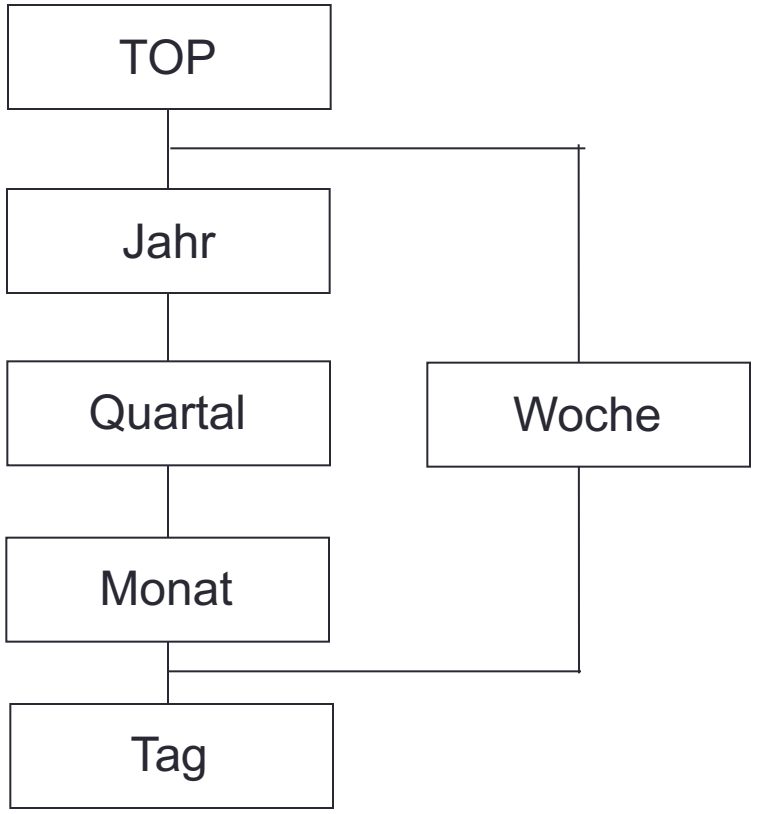
- Einfache Hierarchien
  - Höhere Hierarchieebenen enthalten die aggregierten Werte der jeweils niedrigeren Ebenen
  - Oberster Knoten: Gesamtknoten
    - Verdichtung aller Werte einer Dimension

- Parallele Hierarchien
  - Entstehen bei unterschiedlicher Art der Gruppierung
  - Parallele Äste ohne Beziehung
  - Betrachtung eines Teilaspekts der Hierarchie pro Ast

Klassifikationshierarchie Beispiele

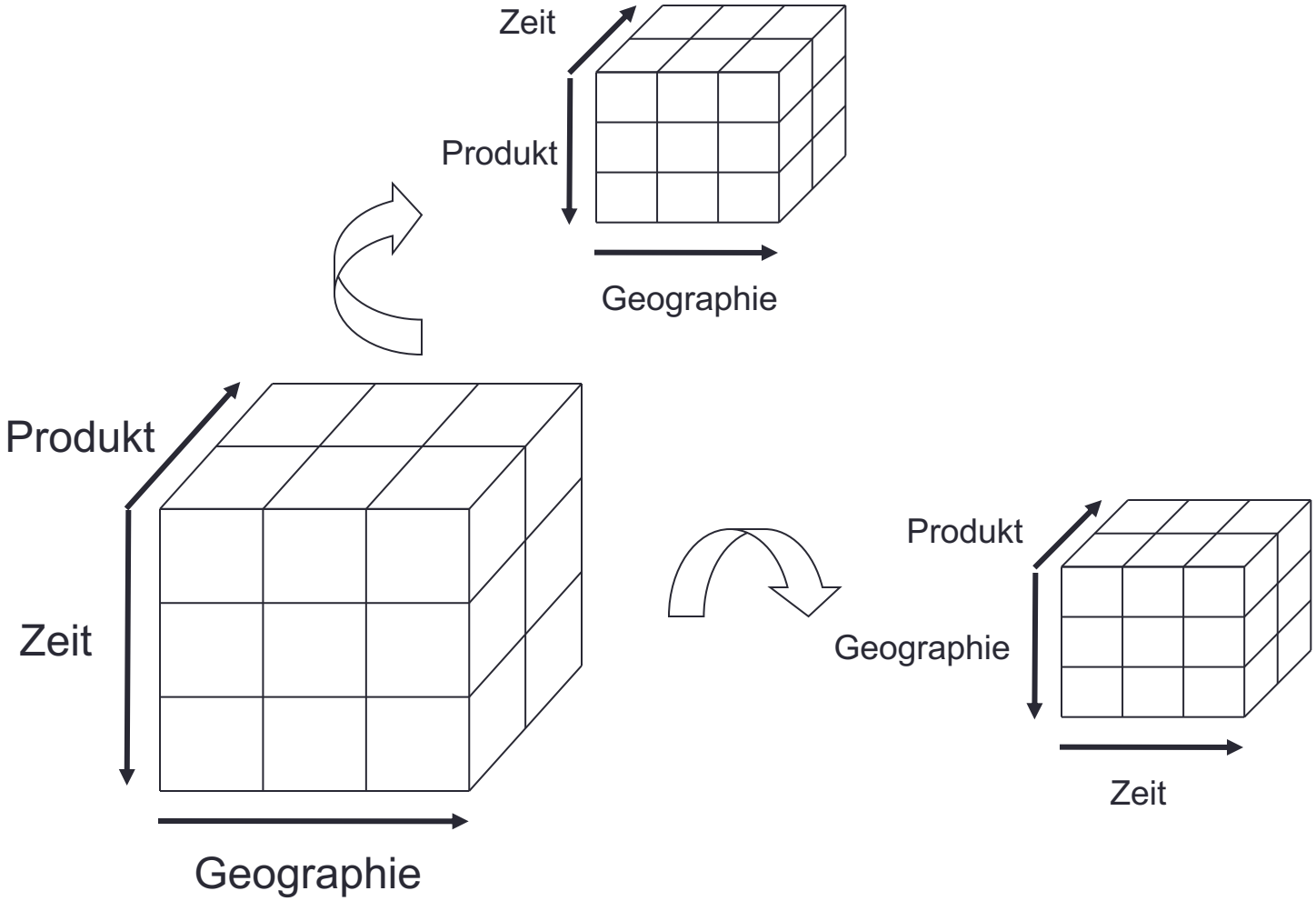


Einfache Hierarchie

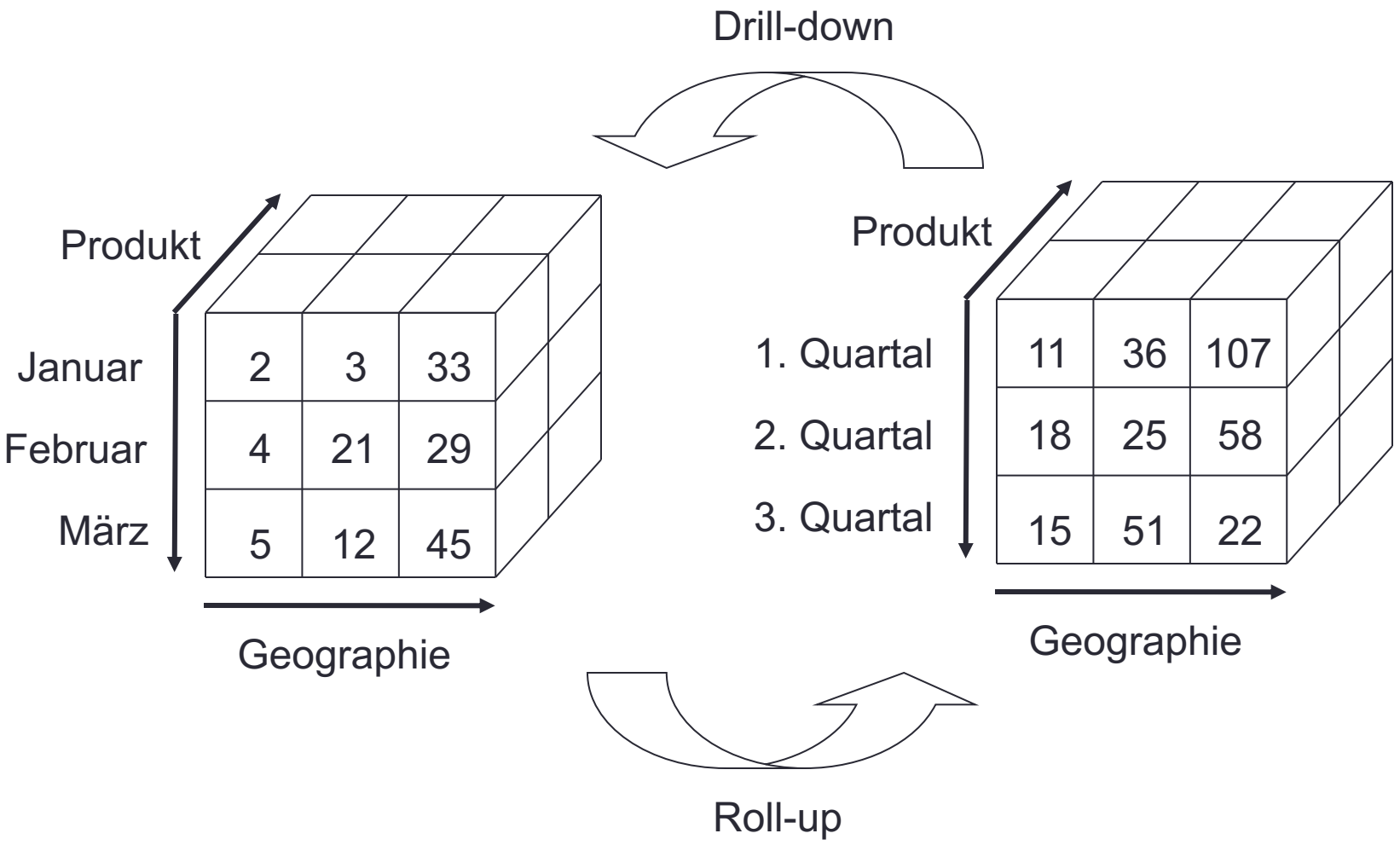


Parallele Hierarchie

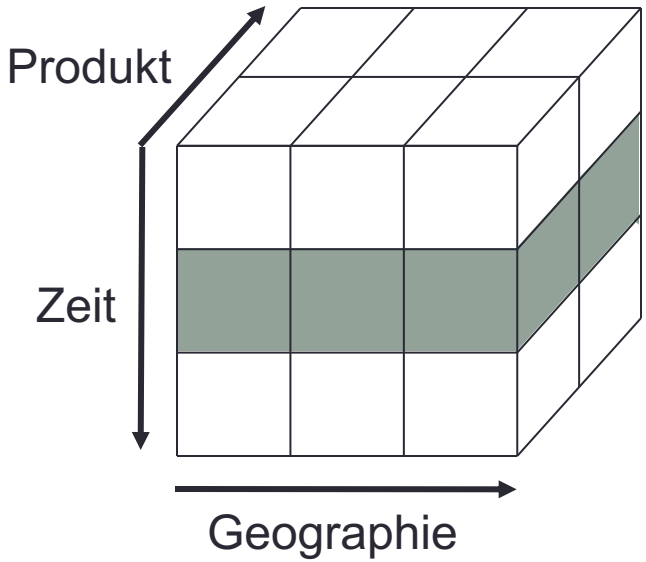
Pivotieren

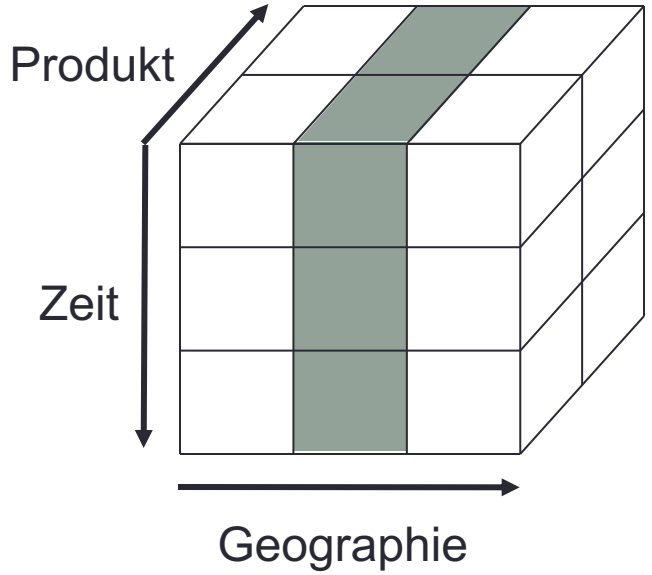

Zeit
Produkt
Geographie
Produkt
Zeit
Geographie
Produkt
Geographie
Zeit

Roll-up und Drill-down



Slicing und Dicing





Agenda

- Einführung
- Deskriptive Methoden zur Datenexploration
- Datenqualität
- Klassifikation
- Recommender Systems
- Clusteringverfahren
- Stream Mining
- Social Network Analysis

- Technische Lösungen
  - Relationale Datenbanken
  - Data Warehouse
  - Datenanalyse und Reporting
  - MapReduce
  - Hadoop
  - Big Data Warehouse

- Datenschutz und gesellschaftliche Aspekte

MapReduce

- Beobachtung Big Data
  - Datenmenge deutlich größer als bei traditionellen Systemen
  - Datenmenge steigt mit hoher Geschwindigkeit an
- Konsequenzen
  - Verteilung der Daten auf mehrere Systeme
  - Berechnung verteilt
  - Zusammenführung der Ergebnisse
- Herausforderungen Architektur
  - Analysen dürfen bei Fehlern einzelner Systeme nicht abbrechen
  - Speicherort der Daten und Ort der Berechnung müssen unabhängig sein
  - Für Berechnungen muss Lastbalancierung erfolgen
- Idee: Zwei Schritte für Berechnung
  - Map: Durchführung von Auswertungen auf Beobachtungen
  - Reduce: Aggregieren der Auswertungen auf zentralen Systemen

Map-Funktion

- Eingabe: List aus (Schlüssel, Wert)-Paaren
- Ausgabe: Neue Liste

- Beispiel
  - Ziel: Ermittlung Anzahl der Kunden älter 25
  - Funktion
    data.map((ID, Alter) => (ID, Alter > 25))
  - Ausführung (zeilenweise!)
    - (572839412, 22) => (572839412, 0)
    - (239025234, 34) => (239025234, 1)
    - (789434884, 98) => (789434884, 1)

Eingabe	ID	Alter
	572839412	22
	239025234	34
	789434884	98

- Vorteil
  - Map-Funktion (bei Verfügbarkeit der Daten) stark parallelisierbar
  - Quell-Code (mit etwas Übung) leicht verständlich

Reduce-Funktion

- Eingabe: Ergebnis aus Map-Funktion
- Vorgehen:
  - Initialisierung eines „Akkumulators“
  - Aktualisierung des Akkumulators mit jeder Zeile der Eingabe
- Ausgabe: Aggregat über den Ergebnissen
- Beispiel
  - Ziel: Ermittlung Anzahl der Kunden älter 25
  - Funktion
    data.map((ID, Alter) => (ID, Alter > 25))
    .reduce((x, y) => x + y)
  - Ausführung (zeilenweise!)
    - (572839412, 0) => 0
    - (239025234, 1) => 1
    - (789434884, 1) => 2

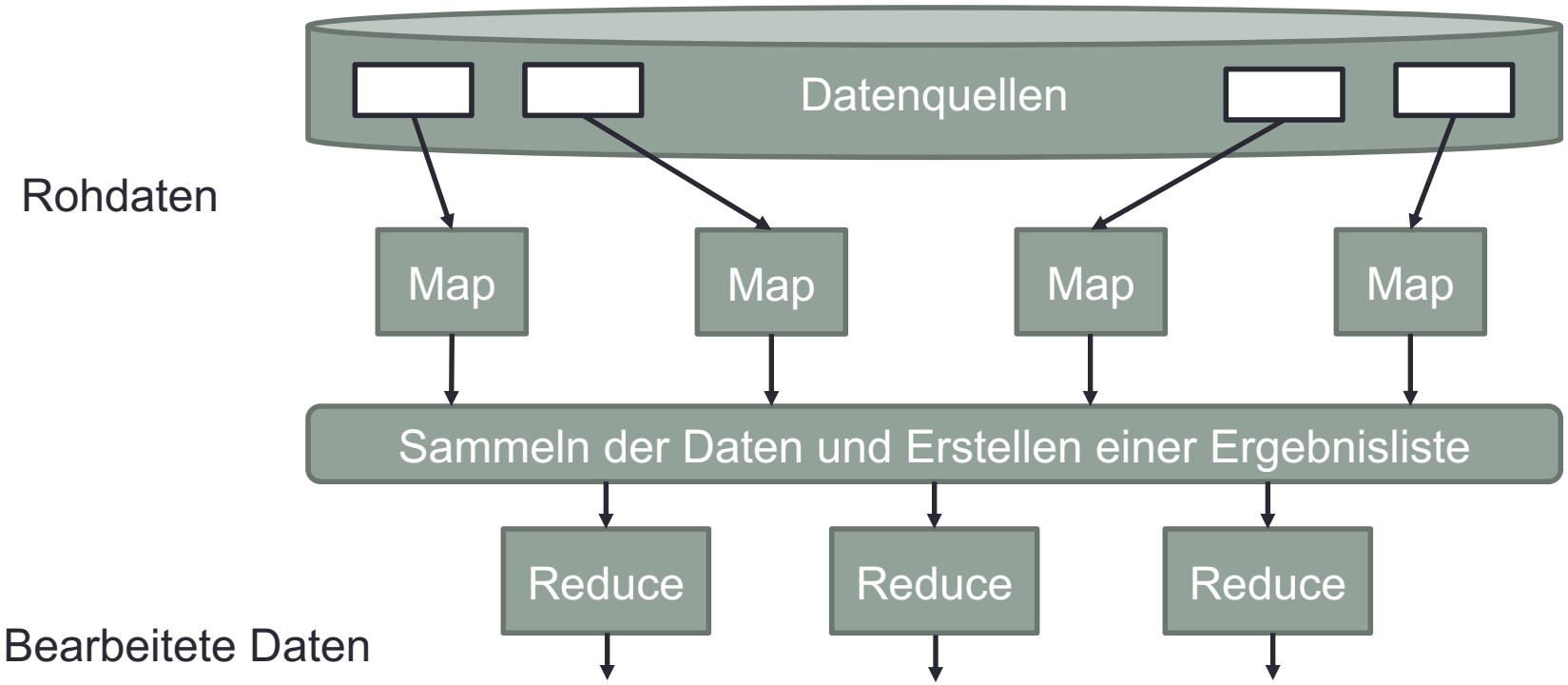
Eingabe

ID	Wert
572839412	0
239025234	1
789434884	1

- Vorteil
  - Reduce-Funktion (bei Verfügbarkeit der Daten) stark parallelisierbar
  - Quell-Code (mit etwas Übung) leicht verständlich

Kombination von Map- und Reduce-Funktion

- Typischerweise Kombination mehrerer Operationen
  - *val result = data.map((x, y) => (x, y, x*y))*
    .map((x, y, z) => (x, y, z, 1))
    .reduce((a, b) => (a._3 + b._3, a._4 + b._4))
  println("Das Ergebnis ist: " + result._1 / result._2)



Grundlegendes Verhalten

- Scheduling
  - Map-Schritte (Tasks) werden vor Reduce-Schritten ausgeführt
  - Verteilung bei mehr Map-(bzw. Reduce-) Schritten als Systemen
- Synchronisation
  - Monitoring des Status der Schritte (abgeschlossen/offen)
  - Übertragung der Daten zwischen den Systemen (shuffle and sort)
- Code/Daten Colocation
  - Hohe Performanz nur bei Speicherung...
    .. von Daten und Code auf gleichem System
  - Zusammenführen von Code und Daten
- Fehlerbehandlung
  - Fehler treten auf und werden behandelt
  - Typische Fehler
    - Zu viel Arbeitsspeicher vergeben
    - Zwischenergebnisse wachsen zu schnell an

Technische Optimierung

- Hardware / Netzwerktopologie
  - Schnellste Hardware ermöglicht schnellste Berechnung, ...
    MapReduce lässt sich aber auch auf günstiger Hardware verteilen
  - Wichtig: Hohe Netzwerkperformance zwischen den Knoten

- Synchronisation
  - Zwischenergebnisse werden direkt nach der Berechnung...
    ... auf weiterverarbeitende Knoten kopiert (Netzwerkperformance!)

- Dateisystem
  - Verteiltes Dateisystem (d.h. Daten auf unterschiedlichen Systemen)
  - Client-Server (d.h. ein System verteilt die Arbeit)
  - Anforderungen
    - Zweiter Master in warmem Standby
    - Große Dateien verbessern Performanz
    - Netzwerkbandbreite wichtiger als CPU Speed
    - Sicherheit verringert Performanz
      (also lieber System einschließen als Nutzerrechte vergeben)

Agenda

- Einführung
- Deskriptive Methoden zur Datenexploration
- Datenqualität
- Klassifikation
- Recommender Systems
- Clusteringverfahren
- Stream Mining
- Social Network Analysis

- Technische Lösungen
  - Relationale Datenbanken
  - Data Warehouse
  - Datenanalyse und Reporting
  - MapReduce
  - Hadoop
  - Big Data Warehouse

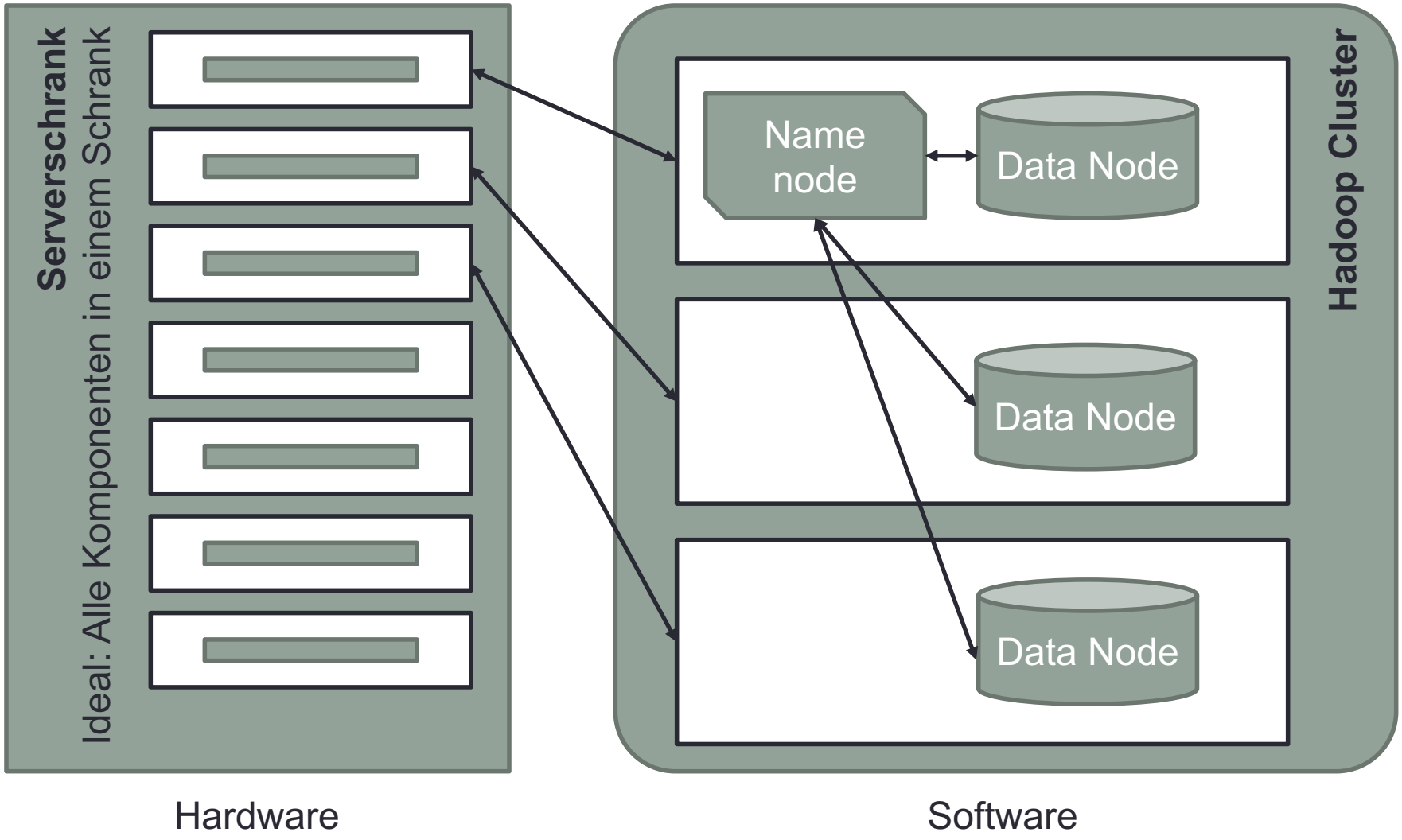
- Datenschutz und gesellschaftliche Aspekte

Hadoop

- Grobinformationen
  - Initiale Entwicklung: Yahoo! (Doug Cutting)
  - Open Source (Apache Software Foundation, Apache License v2.0)
  - Bietet Implementierung von verteiltem Dateisystem, MapReduce Engine
  - Fokus
    - Verarbeitung von Terabytes bis Petabytes von Daten
    - Selbstheilung (d.h. Hardwareausfälle werden erkannt und behandelt)

- Hadoop Distributed File System (HDFS)
  - Optimierung auf Speicherung von Daten
    (kein nachträgliches Bearbeiten wie bei OLTP)
  - Zwei Typen von Systemen
    - Name node
      - System zur Verwaltung der übrigen Knoten
      - Verteilt Ressourcen / Anfragen
    - Data node
      - System zur Speicherung von Daten
      - Meist hohe Anzahl von Data Nodes pro Hadoop Cluster

Architekturbild



Name Node

- Aufgaben
  - Zerlegen großer Dateien in „Blöcke“
  - Zuordnung der Blöcke zu Data Nodes
  - Management von Zugriffen
    (Lesen, Schreiben, Erstellen, Löschen, Replizieren)
  - Rebalancierung (Umverteilen der Zugriffe bei Ungleichverteilung)
  - Wichtig: Single Point of Failure
- Zusammenarbeit mit Data Nodes
  - Regelmäßige Abfrage der Data Nodes beim Name Node
    - Frage nach weiteren Aufgaben
    - Übergabe der Auslastung
  - Name Node kennt
    - Verfügbare Data Nodes
    - Auslastung des Gesamtsystems
- Zusammenarbeit zwischen Data Nodes
  - Austausch von Daten während Bearbeitung
  - Datenaustausch ohne Name Node

Data Node

- Aufgaben
  - Funktionieren am Besten innerhalb eines Racks
  - Replikation von Datenblöcken
  - Senden von „Heartbeats“ an den Name Node
  - Sicherstellen von Datenintegrität
    - Monitoring von Transaktions Logs
      (Jede einzelne Datenoperation wird „isoliert“ ausgeführt)
    - Berechnung von Checksummen
      (Kennzahl wird vor und nach Datenübertragung ermittelt und verglichen)

- Datenhaltung
  - Daten werden auf lokalen Festplatten abgelegt
    (Nicht in NAS oder SAN)
  - Unterstützt Pipelining
    (Daten werden nach Schreiben an nächsten Data Node weitergereicht)
  - Replikationsgrad ist konfigurierbar
    (Trade off zwischen Performance und Speicherverbrauch)

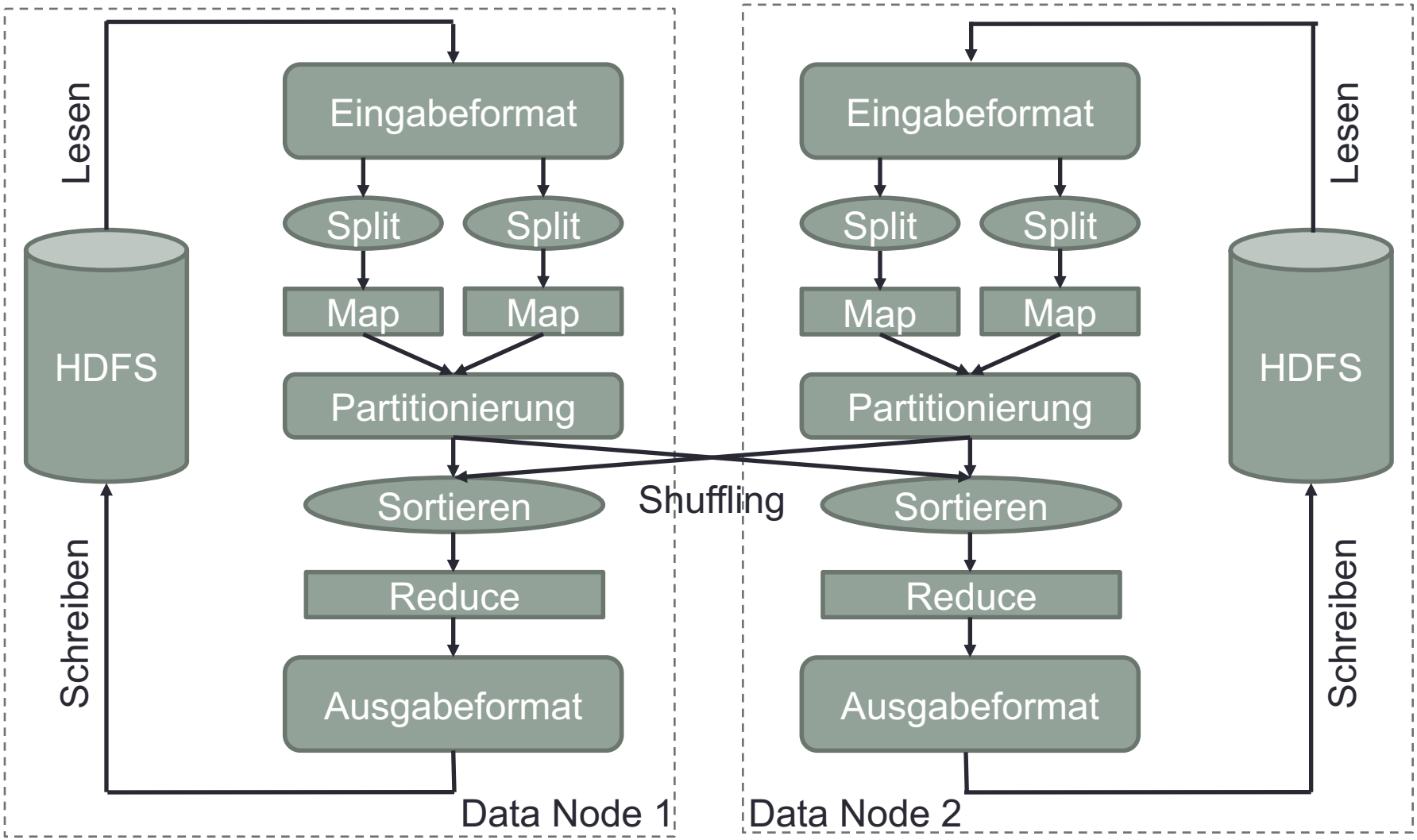
Metadaten

- Idee: Daten über Daten

- Metadaten im HDFS
  - Zeitpunkte: Erstellung, Zugriff, Löschen, ... von Dateien
  - Zugriffsrechte: Lese- und Schreibrechte pro Datei und Nutzer
  - Datenmanagement
    - Verteilung der Dateien auf Blöcke / Data Nodes
    - Anzahl der Dateien im Cluster
  - Clustermanagement
    - Anzahl der Data Nodes im Cluster
    - Ort des Transaktions-Logs für den Cluster

- Verwaltung der Metadaten
  - Speicherung auf dem Name Node
    (Großer Hauptspeicher, Einsatz von SSDs wünschenswert)

Hadoop MapReduce auf zwei Data Nodes



Agenda

- Einführung
- Deskriptive Methoden zur Datenexploration
- Datenqualität
- Klassifikation
- Recommender Systems
- Clusteringverfahren
- Stream Mining
- Social Network Analysis

- Technische Lösungen
  - Relationale Datenbanken
  - Data Warehouse
  - Datenanalyse und Reporting
  - MapReduce
  - Hadoop
  - Big Data Warehouse

- Datenschutz und gesellschaftliche Aspekte

Unterschied Big Data Cluster zu Data Warehouse

- Überschneidungen
  - Beide Systeme halten Daten
  - Beide Systeme dienen der Analyse
  - Ähnliche Nutzergruppen
- Unterschiede
  - Speicherung operationaler und transaktionaler Daten im Data Warehouse
  - Ablage von stark strukturierten Daten im Data Warehouse
  - (Meist) starke Zentralisierung im Data Warehouse
  - Rückspielen von Daten in operative Systeme bei Big Data Cluster
- Im Unternehmen
  - Parallele Existenz von Big Data Cluster und Data Warehouse
  - Data Warehouse (meist schon vorhanden) übernimmt obige Aufgaben
  - Big Data Cluster als Ergänzung für unstrukturierte und Event- Daten
- Idee: Kombination beider Ansätze (möglicher Architekturvorschlag)

Anwendungsbeispiel – Onlineshop I

- Unternehmen bietet viele verschiedene Leistungen
  - Oberbekleidung (d.h. Pullover, Hosen, Hemden, ...)
  - Schuhe
  - Accessoires (d.h. Taschen, Schmuck, ...)

- Unternehmen ist „Multi-Channel“
  - Verkauf über Webseite
  - Verkauf über „Marktplätze“ (z.B. Amazon Marketplace)
  - Unterschiedliche Angebote für verschiedene Länder

- Status Quo (vor Einführung eines Big Data Clusters)
  - Operationale und transaktionale Daten im Data Warehouse (Verkäufe, Lieferantendaten, Logistikbewegungen, ...)
  - Web Traffic wird 3 Monate archiviert und nicht intern untersucht
  - Zukauf eines externen Tools zum Web Tracking (z.B. google Analytics)

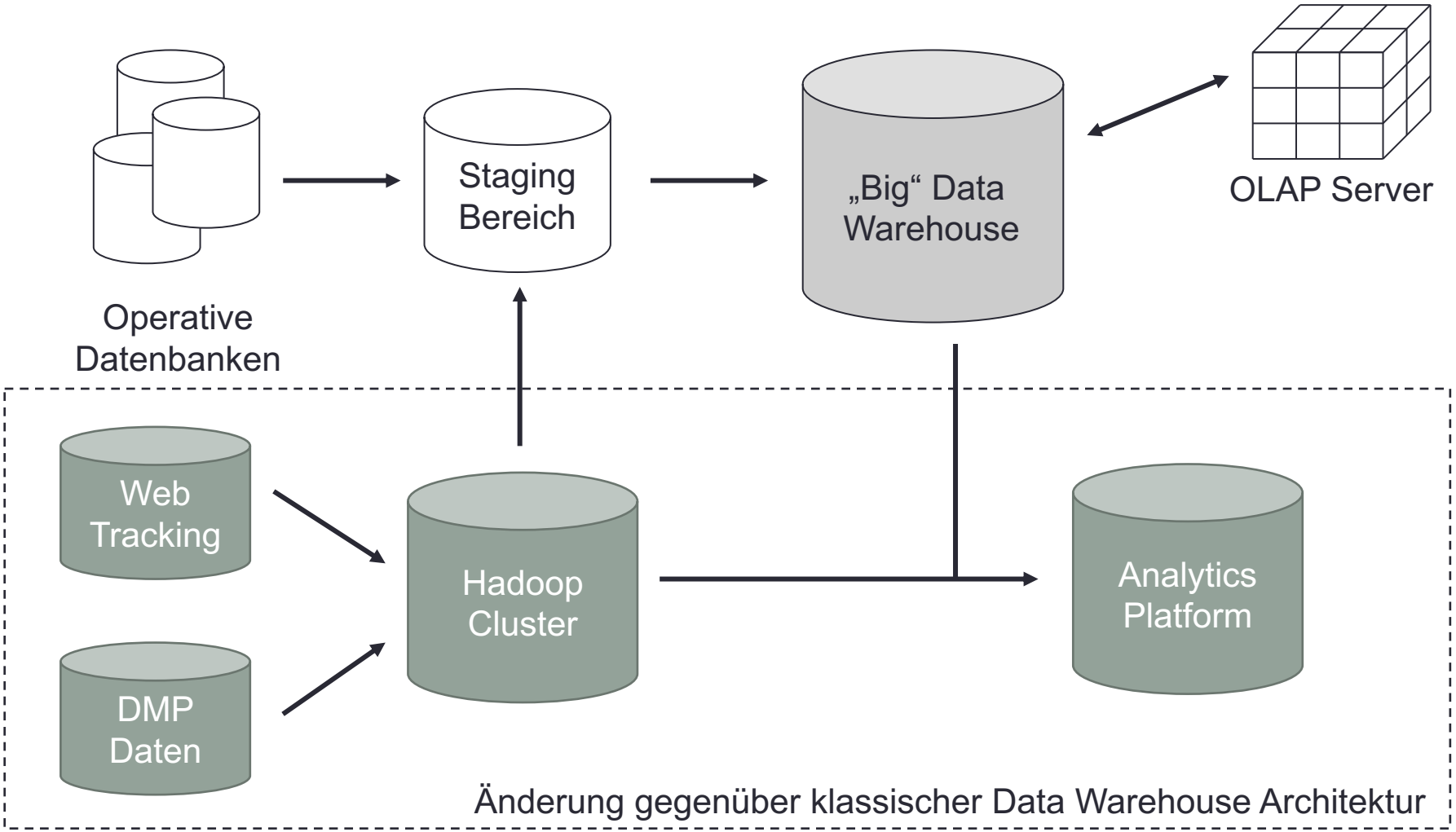
- Problem
  Externes Tool erlaubt nur vorgegebene Analysen – viel Wissen geht verloren!

Anwendungsbeispiel – Onlineshop II

- Entscheidungen in der Geschäftsführung
  - Wegwerfen der Daten nach 3 Monaten ist nicht hilfreich
  - Daten werden in Hadoop Cluster abgelegt
  - Daten werden im Hadoop Cluster aggregiert...
    ... und aus dem Hadoop Cluster in Data Warehouse zurückgespielt

- Vorteile des Ansatzes
  - Auswertung des Web Traffics mit individuellen Reports möglich
    - Demographische Änderungen in der Kundengruppe werden erkannt
    - Änderungen im Kundenverhalten werden erkannt
    - Erfolgsmessung von Onlinemarketingkampagnen ist möglich
  - Neue Anwendungsmöglichkeiten
    - Rückspielen der aggregierten Daten ins Data Warehouse und Abgleich
    - Auswertungen auf unaggregierten Daten in Near Real Time
  - Vorverarbeitungsschritte bereits im Big Data Cluster möglich
    - Große Datenmengen berücksichtigbar
    - Existierende Architektur kann weitergenutzt werden

Erweiterte Referenzarchitektur



Einordnung

- Je nach Art der Analyse sind unterschiedliche Methoden sinnvoll:

Art der Analyse	Beschreibung
Standard Analyse für Erkenntnisse	Slicing und Dicing der Daten, Reporting, einfache Visualisierungen [Data Warehouse]
Fortgeschrittene Analysen	Komplexere Verfahren wie Pattern-Matching oder Predictive Modelling [Data Mining]
Operationalisierte Analysen	Analysen werden direkt Teil der Wertschöpfungskette
Monetarisierte Analysen	Analysen werden direkt genutzt um Mehrwert zu generieren

- Fortgeschrittene Analysen Gewinnen durch „Big Data“ an Wert
- Insbesondere für operationalisierte und monetarisierte Analysen Big Data
  - Implementierung der Anwendung im Big Data Cluster möglich
  - Analysen basieren auf Erkenntnissen aus vorhergehenden Schritten
  - Analysen nicht auf „Samples“ möglich