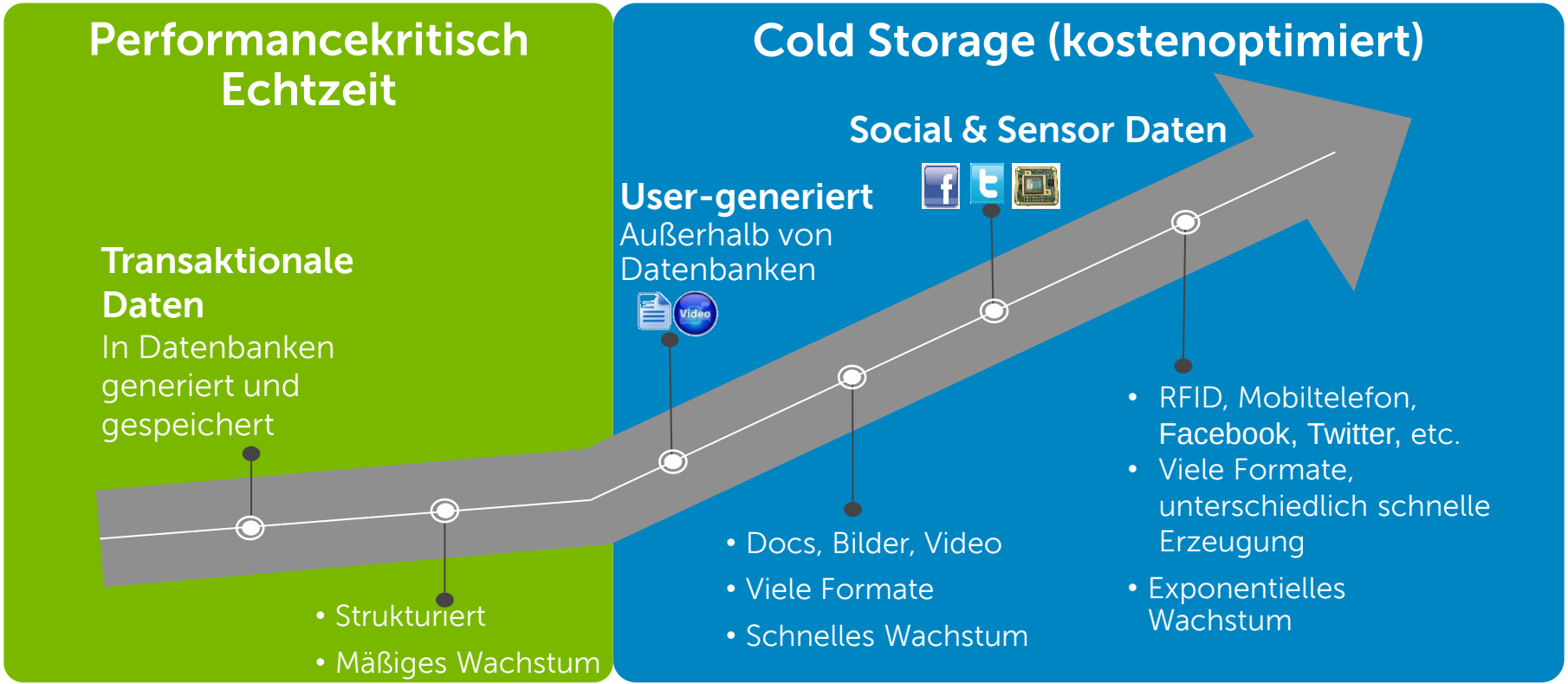
Neues aus der nicht-, semi- und relationalen Welt

Information Management

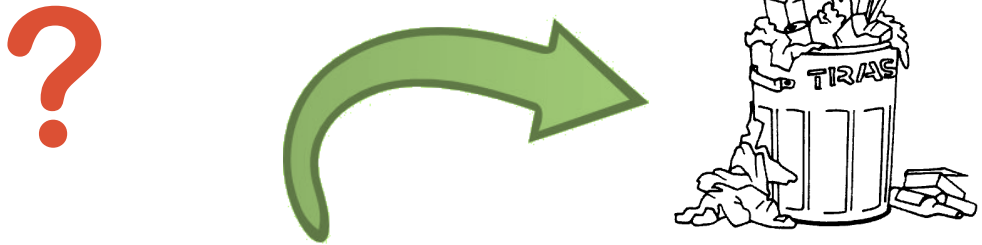
Thomas Klughardt
Senior System Consultant

Das „Big Data“ Problem

Was bedeutet Big Data?



Big Data Analytics
Informationen aus Daten gewinnen



Unstrukturierte Daten

Big Data Analytics
Informationen aus Daten gewinnen

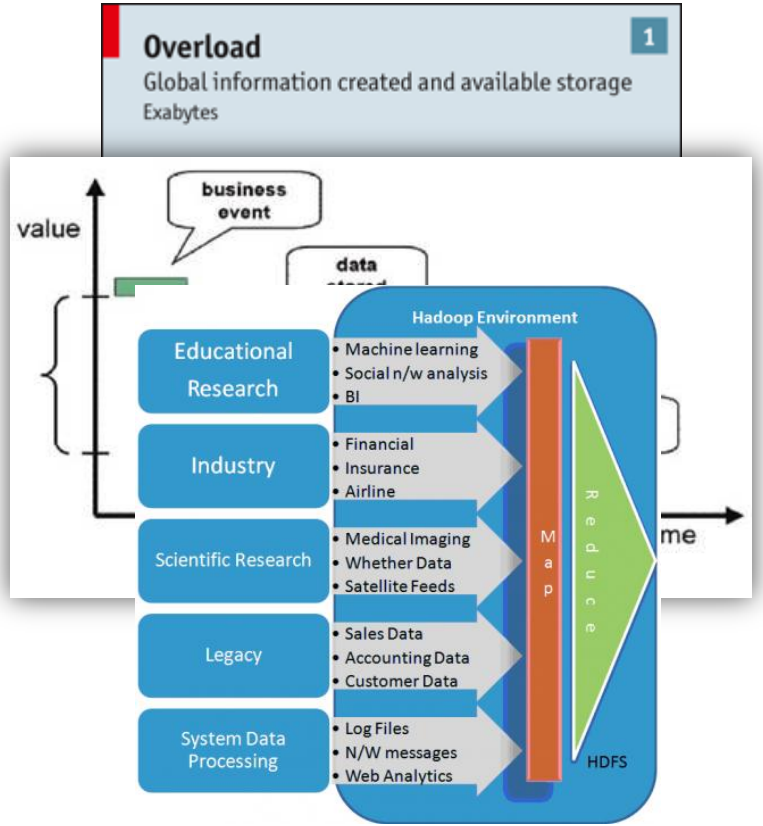
Informationen

Unstrukturierte Daten

Big Data – was ist das?
Die berühmten drei V

Big Data hat eine oder mehrere der Charakteristiken:

- **Volume** – Große, schnell wachsende Datenmengen
- **Velocity** – Die Geschwindigkeit, in der Daten anfallen, verarbeitet und ausgewertet werden müssen
- **Variety** – Die Vielfalt an Datentypen, Strukturen und Formaten



Neue Erkenntnisse – wo möchten wir hin?

Zunehmende Reife

Prescriptive Analytics		"Welche Schritte sind notwendig, um ein bestimmtes Ziel zu erreichen?" (Optimierung)
Predictive Analytics		"In welche Richtung wird sich unser Geschäft wahrscheinlich entwickeln?"
Geschäftsberichte und Analysen		"Wie schneiden wir ab, verglichen mit unseren Geschäftszielen?"
Datenverknüpfung und Konsoliodierung		"In welchen Regionen verkaufen wir am besten/meisten?"
Sammeln von Daten Basisabfragen		"Wer sind meine Kunden?"

Plattformen

NoSQL Systeme

Arten von NoSQL Systemen (Auszug)

- Wide Column Store / Column Families
- Document Store
- Key Value / Tuple Store
- Graph Databases
- Multimodel Databases
- Object Databases
- XML Databases
- Grid & Cloud Database Solutions
- Multidimensional Databases
- Multivalue Databases
- Event Sourcing
- Andere
  - z.B. Lotus Notes Domino

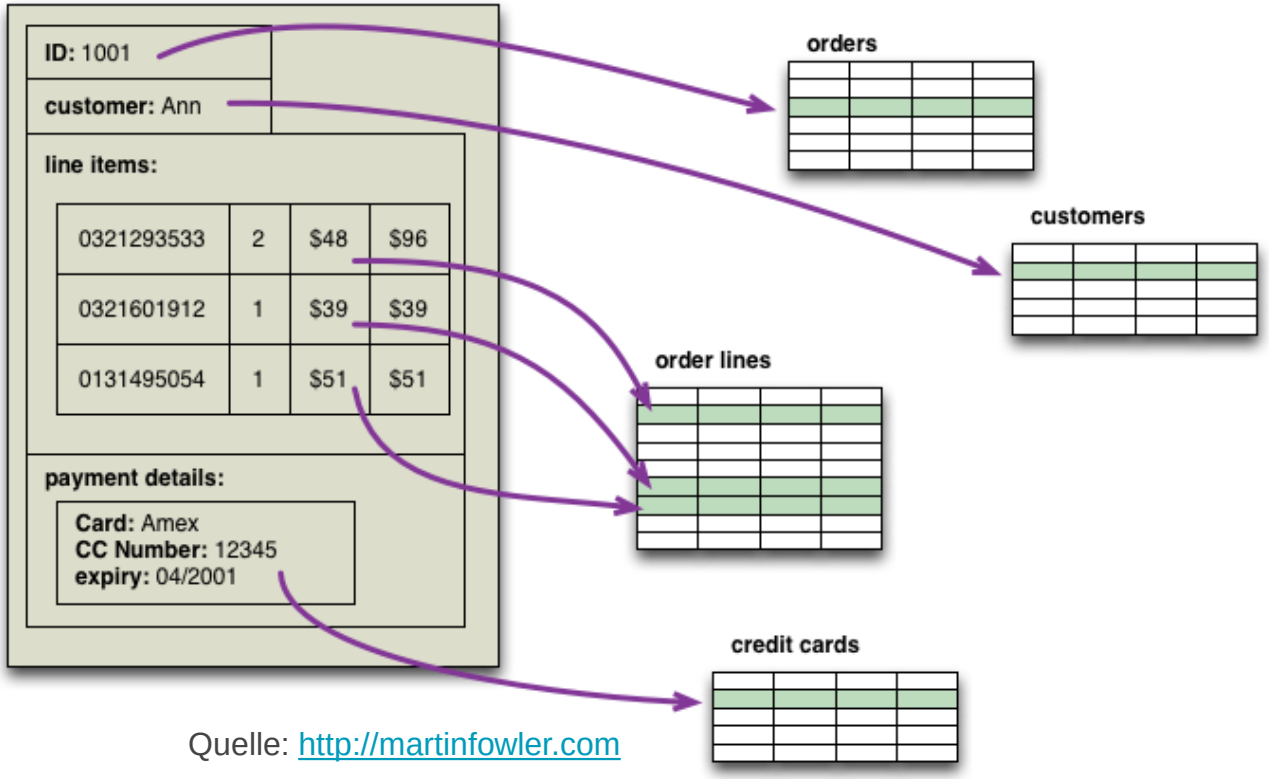
Weiterführende Informationen: http://nosql-databases.org/

Aggregatorientierte Datenbanken

- Wide Column Stores
- Document Stores
- Key Value Stores

- Denormalisiert
- Schnell und skalierbar

- Daten sind Aggregate



Quelle: http://martinfowler.com

Hadoop, Spark und co

- Aggregatorientierte Datenbanken sind denormalisiert
  - Aggregate können nicht zerteilt werden
- Wie machen wir Joins?
  - Daten in ein anderes System laden und dort verarbeiten
    - Zum Beispiel Hadoop
- Das passiert im Batch Betrieb.
  - Langsam und es müssen immer alle Daten betrachtet werden -> teuer.
  - Dafür große Datenmengen und beliebige Auswertungen.

Plattformen

NewSQL Systeme

NewSQL Systeme – semirelationale Datenbanken

- ACID-konforme Transaktionen, hochskalierbare und ständig verfügbare Datenbanken – kann das funktionieren?
  - Nein! -> Das widerspricht dem CAP Theorem
- Und doch gibt es diese Datenbanken
  - Google's Spanner Datenbank
  - NuoDB
- Wie geht das?
  - Versionierung statt Sperren
  - Asynchrone Replikation mit Konflikterkennung

NewSQL Systeme – semirelationale Datenbanken

- Vorteile:
  - Verstehen meist SQL Syntax
  - ACID konforme Semantik
    - ACID Verletzungen werden erkannt
  - Schnell, skalierbar, ausfallsicher

- Nachteil:
  - Speziell, nicht für jede Anwendung geeignet.

- Ungünstig, wenn Updates auf den gleichen Datensatz von verschiedenen Stellen ausgeführt werden.

Plattformen

In-Memory Column Stores

In-Memory Column Stores

- „Neue“ Art der relationalen Datenverarbeitung
- Derzeit hauptsächlich im Gespräch:
  - Oracle 12c In-Memory Option
  - SAP HANA
- Was bedeutet In-Memory für uns?
  - „Normale“ relationale Datenbanken arbeiten größtenteils im Speicher.
  - Praktisch das Gleiche, als würde man die gesamte Datenbank in den Buffer Cache pinnen…
  - … nichts wirklich Neues.

(In-Memory) Column Stores

- „Neue“ Art der relationalen Datenverarbeitung
  - Column Stores gibt es, seit es relationale Datenbanken gibt.
- Andere Column Stores
  - IBM DB2 with BLU Acceleration
  - Microsoft SQL Server 2012 mit Column Store Index
  - Sybase IQ
  - Und viele weitere…
- Wozu Daten spaltenweise speichern?

(In-Memory) Column Stores

Personalnr	Nachname	Vorname	Gehalt
1	Schmidt	Josef	40000
2	Müller	Maria	50000
3	Meier	Julia	44000

- Zeilenweise Speicherung:

```
1,Schmidt,Josef,40000;2,Müller,Maria,50000;3,Meier,Julia,44000;
```

- Spaltenweise Speicherung:

```
1,2,3;Schmidt,Müller,Meier;Josef,Maria,Julia;40000,50000,44000;
```

(In-Memory) Column Stores

- Vorteile:
  - Wenn nur bestimmte Spalten gelesen werden.
  - Wenn Aggregate gebildet werden
    - Üblicherweise werden Min, Max, Count, Average alle x Zeilen gespeichert.
  - Besser komprimierbar (wenn Werte sich wiederholen)

- Nachteile
  - Wenn nur bestimmte (komplette) Zeilen gelesen werden.
  - Wenn oft kleine Insert Operationen durchgeführt werden.

- Column Stores funktionieren gut im Data Warehouse Umfeld.

Die Mischung macht‘s

Verschiedene Plattformen für verschiedene Dinge

- Relationale Datenbank
  - Auftragsverwaltung
  - ERP System
- Hadoop Cluster
  - Sensordaten
  - Datenhalde und Rechencluster
- Aggregatorientierte NoSQL Datenbank
  - CRM
  - Webanwendungen
- In-Memory Column Store
  - Data Warehouses
  - Große, analytische Abfragen

Traditioneller Ansatz vs. Big Data Architektur

- Relationale Datenbank
  - Strukturiertes Schema; normalisierte Daten
  - Schema on Write
  - Verknüpfbare Daten
  - Konsistentes Modell
- Big Data Architektur
  - Mischung aus relationalen und nicht-relationalen Datenbanken
  - Erfassung und Speicherung von unstrukturierten und strukturierten Daten
  - Direkte Auswertung oder Aggregation in relationale Daten
  - Schema on Read; nach Aggregation meist Schema on Write
- Big Data ≠ NoSQL
  - NoSQL Systeme normalerweise nur ein Bestandteil einer Big Data Lösung.

Fazit

Fazit

- Es gibt keine eierlegende Wollmilchsau.
  - Vermeintliche Allheilmittel werden schnell entzaubert.
  - Man kann Plattformen schlecht „verbiegen“.
- Big Data Plattformen erfordern zusätzliches Wissen.
  - Es ist ein weiter Weg bis zur kompletten Plattform.
- Die Anforderungen sind schon da und werden weiter kommen.
  - Besser verknüpfte Daten sind ein Wettbewerbsvorteil.
  - Deshalb auch besser jetzt schon damit beschäftigen.
  - Es gibt viele „Experten“, die den Fachbereich beraten.