Digital Innovation Hub Süd

Datenbanken - Security Aspekte
Teil 2: BigData & NoSQL

Agenda

- Einführung in Big Data & NoSQL
- Eigenschaften von NoSQL Systemen
- Vergleich zu relationalen Sytemen
- NoSQL-Security im Vergleich mit relationalen Systemen
- Zusammenfassung & Ausblick

BigData & NoSQL

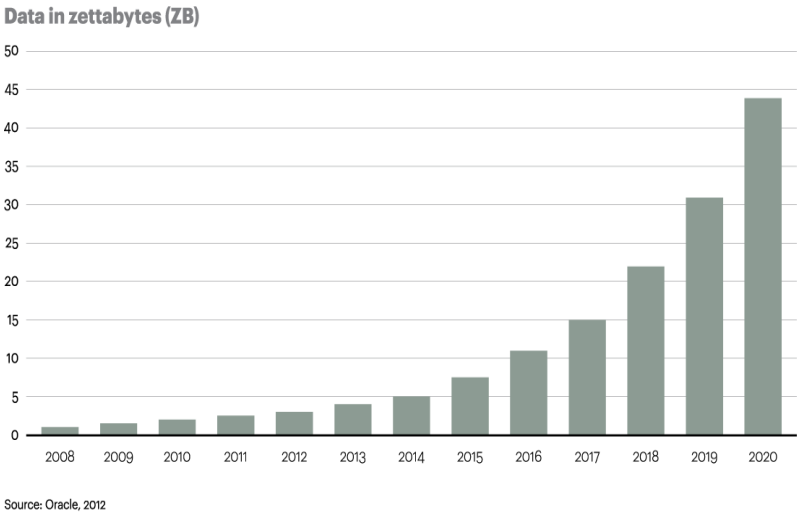
Globales Datenwachstum

Typische Treiber:

- Cloud
- IOT
- Sensoren
- Smart meters
- Social Networks
- Online Shopping
- Mobile

Figure 1

Data is growing at a 40 percent compound annual rate, reaching nearly 45 ZB by 2020

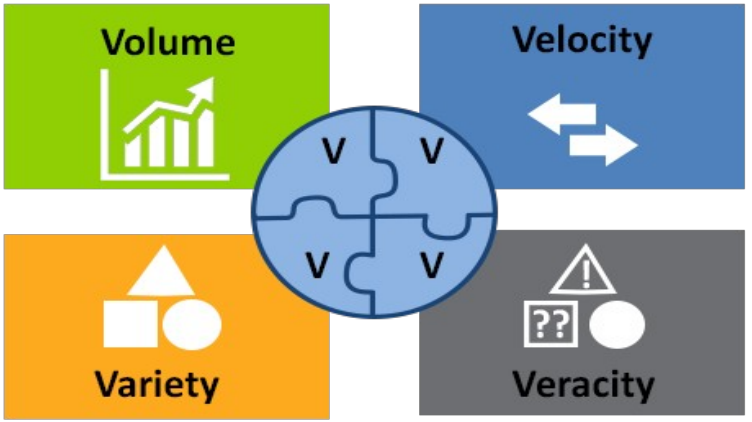


Source: Oracle, 2012

http://www.atkearney.com/strategic-it/ideas-insights/article/-/asset_publisher/LCcgOeS4t85g/content/big-data-and-the-creative-destruction-of-today-s-business-models/10192

Definition Big Data

- Es gibt keine einheitliche Definition
- Wird beschrieben durch die 3-9 V's:
  - Volume
  - Velocity
  - Variability



- Validity
- Value
- Variety
- Velocity
- Veracity
- Viability
- Visibility
- Volatility
- Volume

Big Data ist ein Paradigmenwechsel: Technology & Methoden Change

- Storage Systems
  - alt: Relational Database Systems
  - neu: NoSQL Database Systems
- Processing
  - alt: Serial Transactional Processing
  - neu: Parallel Data Processing
- Analytics
  - alt: Statistische Analyse
  - neu: Machine Learning Algorithms

Insbesondere: Das Aufkommen von NoSQL Systemen

- 2004: Forscher von Google entwickeln einen neuen Datenbanktyp den sie „BigTable“ nennen.
  -> Dieser war nur für den internen Einsatz bei Goolge gedacht.
  -> Die zugehörige Publikation wurde von den unabhängigen Programmieren Doug Cutting and Mike Cafarella aufgegriffen.
- Diese überzeugten Yahoo, dass dies die Lösung für deren Such- und Indizierungsprobleme sei.
- 2006: Yahoo erstellt einen ersten Prototyp namens ‘Hadoop’
- 2008: Yahoo veröffentlicht die erste kommerzielle Implementierung von Hadoop.
- Facebook, Twitter, eBay, und andere Big Player übernehmen diese Technologie.

- Der Begriff NoSQL war der Name einer Open-Source Datenbank von Carlo Strozzi.
  Daten wurden in Textdateien gespeichert auf die ohne SQL zugegriffen wurde.
  Dies war keine NoSQL Datenbank im heutigen Sinn.

- 2009: „NoSQL“ im Weblog von Eric Evans als Namens-Vorschlag für ein von
  Johan Oskarsson organisiertes Event zu „distributed data storage“ .

Eigenschaften von NoSQL Systemen

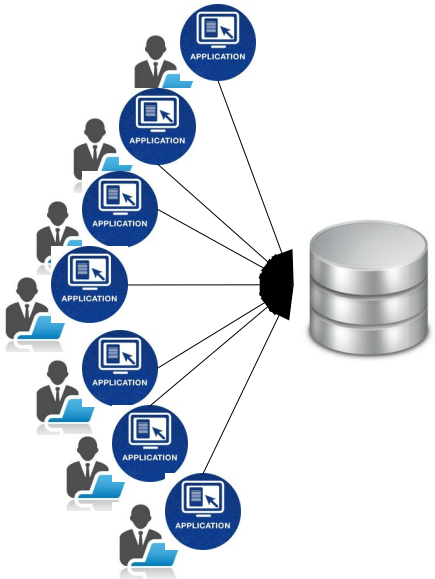
NoSQL ist grundsätlich optimiert für

- Extreme Datenmengen
- Hohe Geschwindigkeit (Datenstreaming)
- Unstrukturierte Daten
- Vielzahl unterschiedlicher Datenquellen

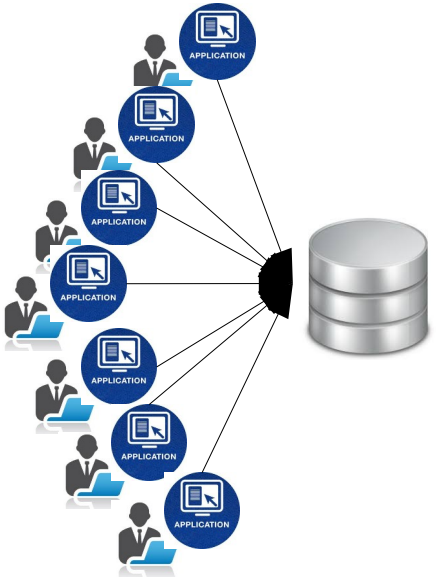
Ab 2000er DAS Thema: Skalierbarkeit

Hohe Last durch viele Read-Operationen

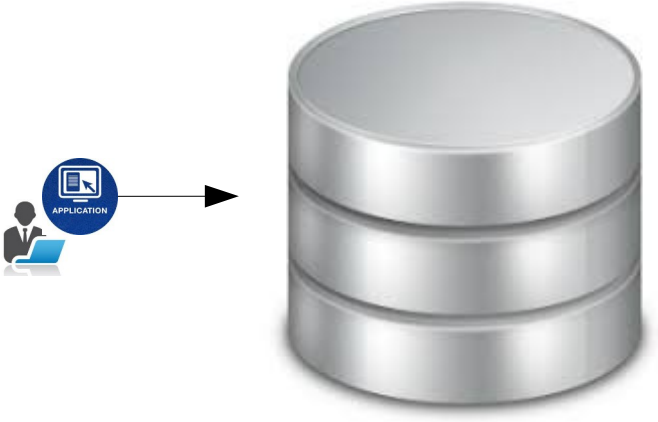
-> 3 Szenarien



Hohe Last durch viele Write-Operationen



Hohe Last durch einzelne Operation
auf extrem großer Datenmenge

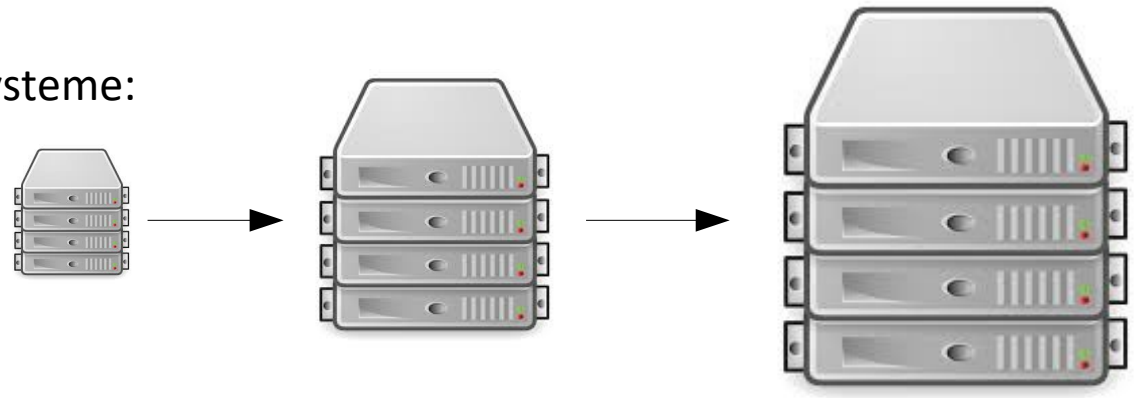


2 Techniken

Bevorzugt für relationale Systeme:

Vertical Scaling

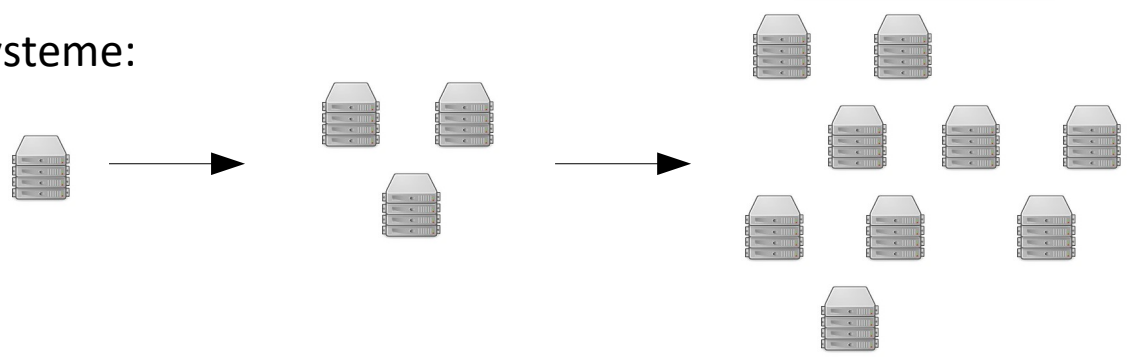
(scale-up)



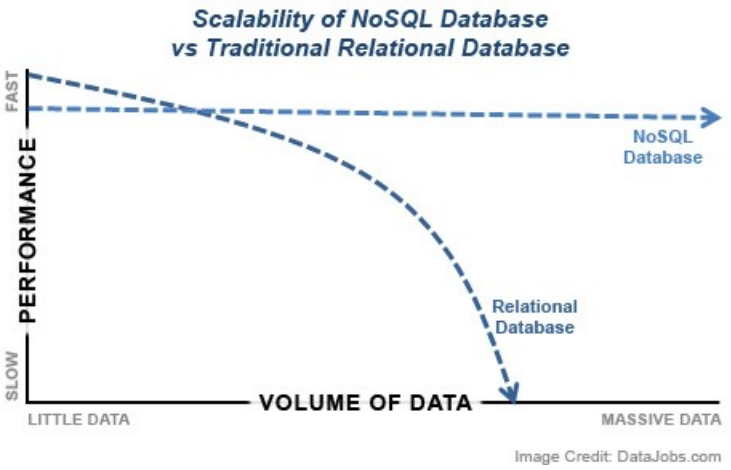
Einsatzszenario der NoSQL Systeme:

Horizontal Scaling

(scale-out)

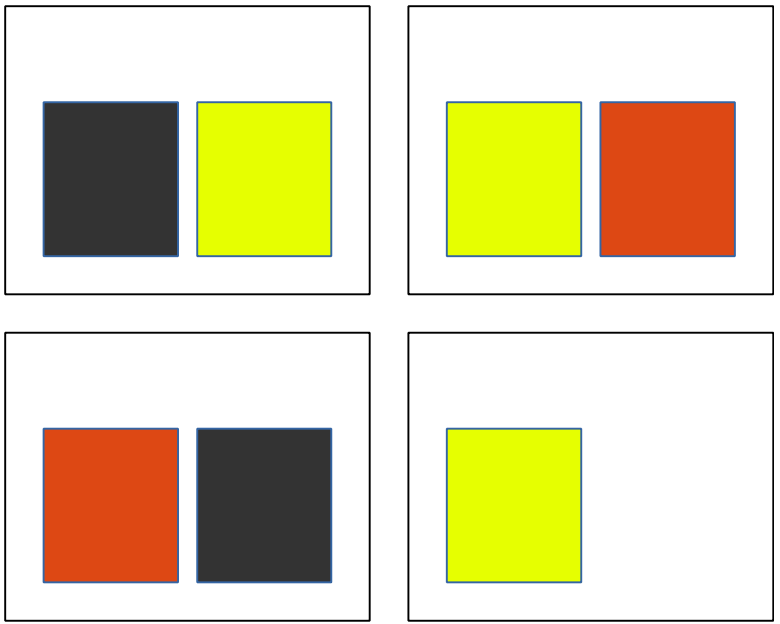


Scalierbarkeit: Scale Out vs. Scale Up



Quelle: DataJobs.com

Sharding: Aufteilung der Daten auf Knoten



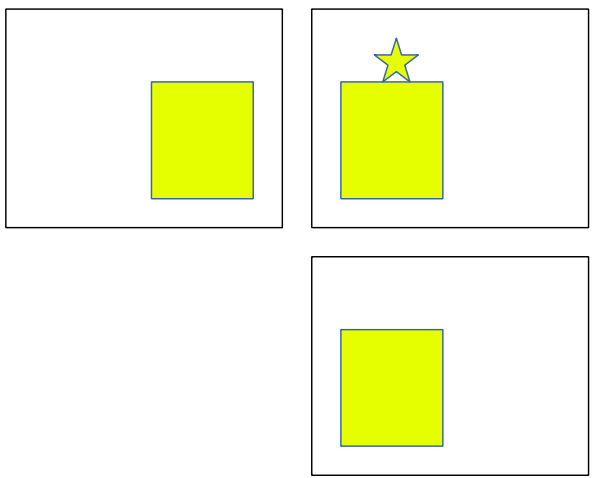
Beispiel: 3 Shards - 4 Knoten

Ermöglicht parallel Computing:

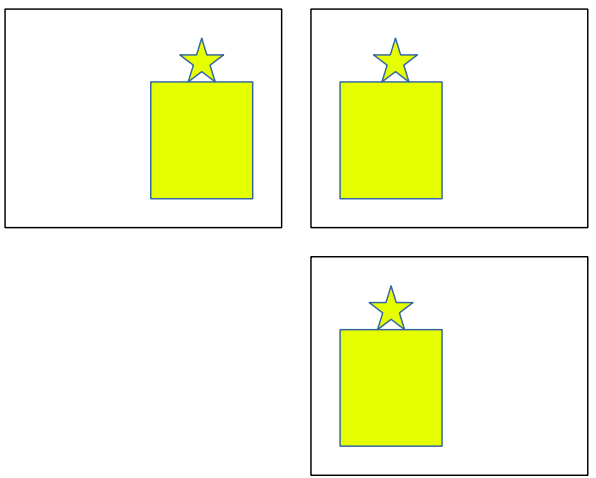
1 Query wird auf viele Knoten und Shards aufgeteilt

Replikation: Verteilung von Datenkopien auf Knoten

Master-Slave Replication



Peer 2 Peer Replication



Vorteile:

Partition Tolerance

Load Balancing für

-> Leselast (Master-Slave)

-> Schreiblast (Peer 2 Peer)

bedeutet Schreibzugriff

NoSQL: Grundlegende Eigenschaften

- Auf vergleichsweise billiger Standardhardware verfügbar
- Für Scale-Out ausgelegt, ermöglicht einfachen Einsatz in der Cloud
- Parallel Computing, Distributed Queries - Berechnungen auf vielen Knoten gleichzeitig
- Ausfallssicher - das Gesamtsystem verkraftet den Ausfall einzelner Knoten
- Flexible Datenmodelle und Datentypen
- Schemalos im Gegenzug kaum Unterstützung für Integritätsbedingungen
- Abfragen nur bedingt flexibel, müssen bei Datenmodellierung berücksichtigt werden
- Kein ACID dafür BASE
- Häufig Open Source

Aber: Diese Eigenschaften werden von den jeweiligen Systemen unterschiedlich stark erfüllt

NoSQL: Ein weiteres von vielen Werkzeugen

FH JOANNEUM University of Applied Sciences

Data Analytics & Modelling

MATLAB · Apache Spark · R · GNU Octave · ANACONDA

SQL-Databases

Microsoft SQL Server · MySQL · ORACLE DATABASE · PostgreSQL · MariaDB

Multidimensional Databases

Microsoft SQL Server Analysis Services · pentaho · ORACLE DATABASE · jedox

NoSQL-Databases

redis · mongoDB · APACHE HBASE · cassandra · neo4j

Distributed Data Storage

hadoop · ceph

Container Technologies

kubernetes · docker

Virtualisation

KVM · Xen · vmware vSphere

Vergleich: Relationale DBMS vs NoSQL Systeme

Relationales Datenmodell

Filialen

Filiale	Adresse	#Mitarbeiter
Graz	Wiener-Straße	12
Kapfenberg	Werk-VI-Straße	8
Leoben	EHZ-Johann Str.	8

Produkte

ID	Name	Kategorie
10	Kronprinz Rudolf	1
20	Karotten	2
30	Beiried	3
40	Golden Delicious	1
50	Karee	3

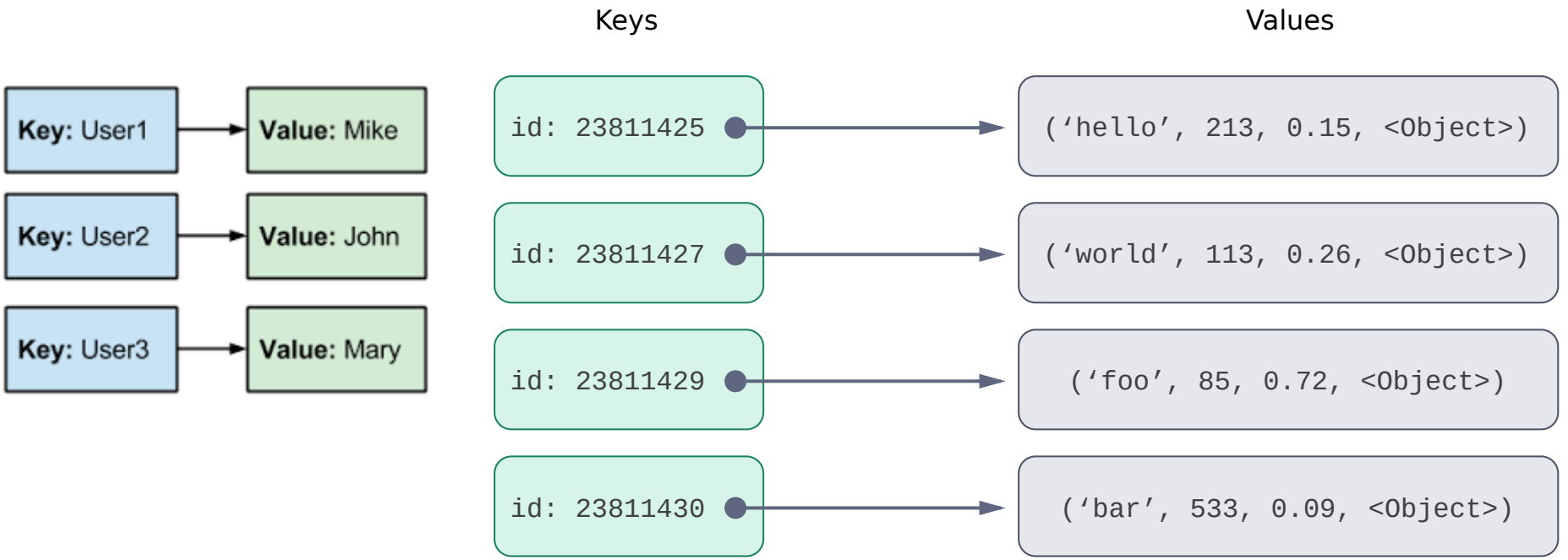
Produktkategorien

ID	Name
1	Obst
2	Gemüse
3	Fleisch

Umsätze

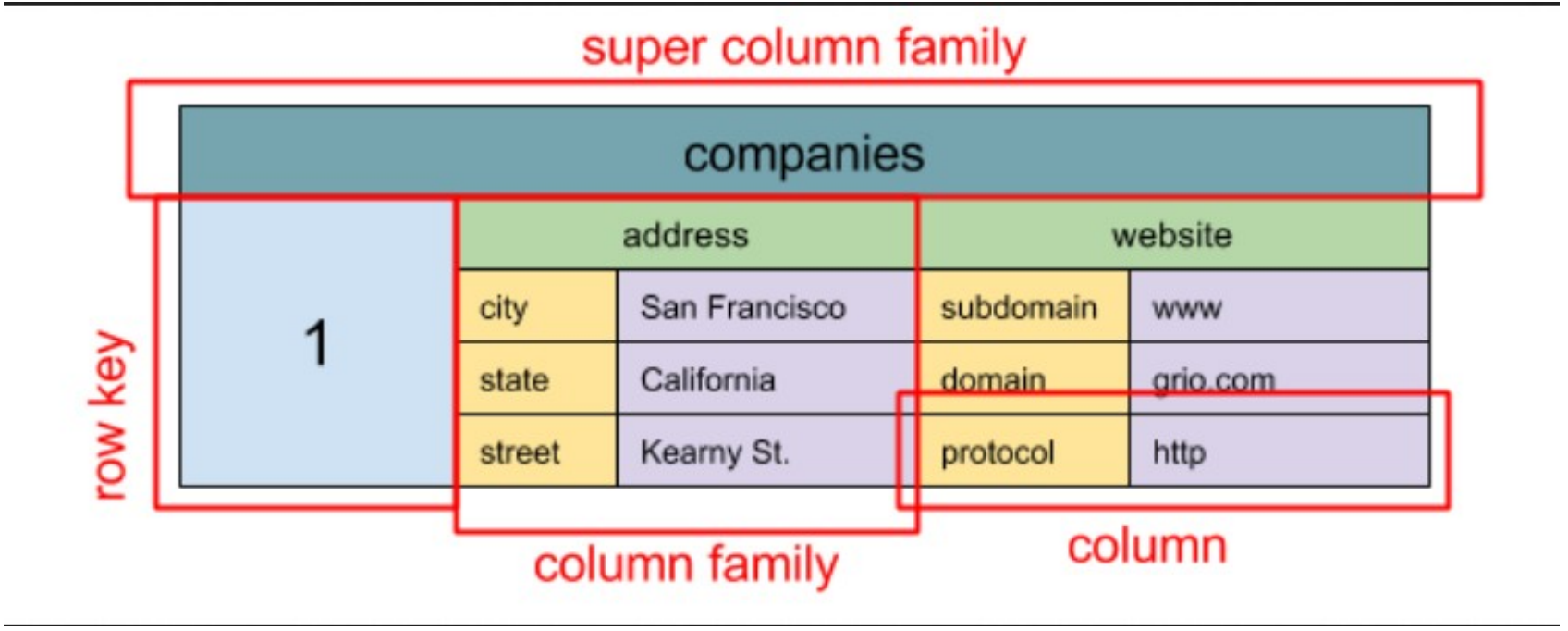
Betrag	Produkt	Filiale
24035	10	Graz
12280	20	Graz
13566	30	Kapfenberg

1) Key Value Datenbanken

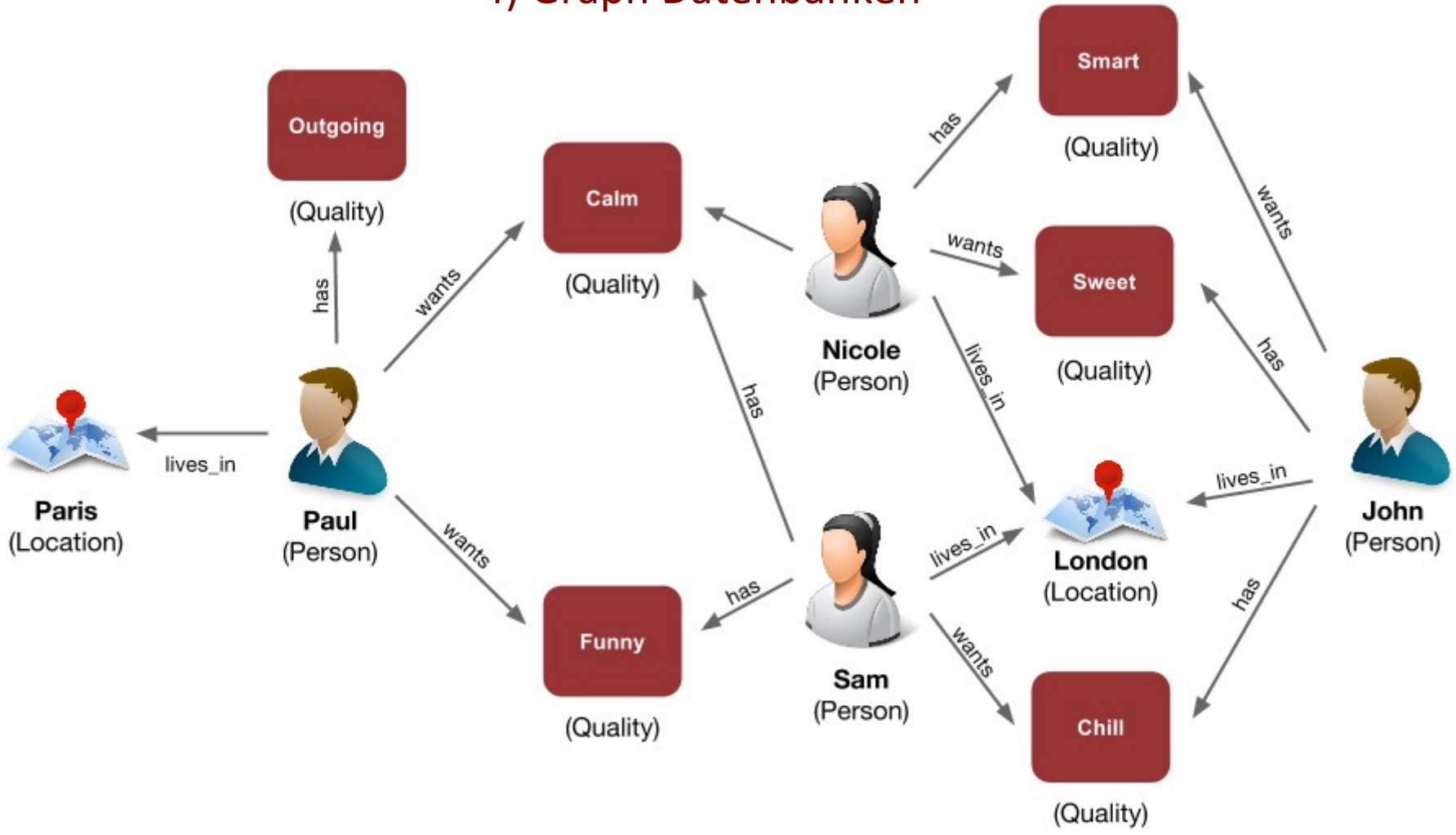


2) Dokumenten-Datenbanken

3) Wide Column Store



4) Graph Datenbanken



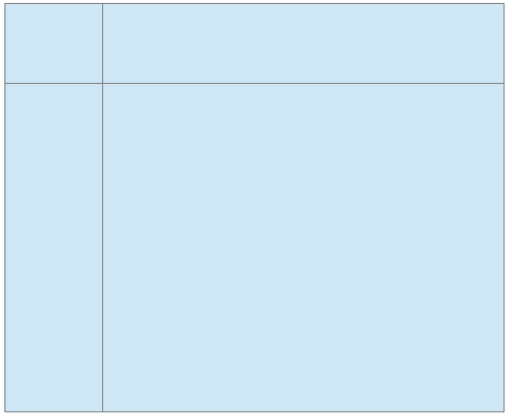
Problem des relationalen Modells: Impedance Mismatch

Objektorientierte Programmierung

```
Objekt:
{
        name: "Sue",
        age: 26,
        status: "married",
        groups: ["news", "sports"],
        child: {
                    name: "Joe",
                    age: "3"
                }
}
```

?

Tabelle



NoSQL: Speicherung von Objekten

Dokumente, Graphen und Wide Column Stores erlauben das Speichern von Objekten.

Daher sind diese Systeme ideal zur Anbindung an Objekt-Orientierte Software.

```
{
        name: "Sue",
        age: 26,
        status: "married",
        groups: ["news", "sports"],
        child: {
                name: "Joe",
                age: "3"
            }
}
```

C – A – P:

Consistency

Darunter versteht man Cluster-Konsistenz, nicht Transaktions-Konsistenz.

„Jede Read Operation liefert den Zustand nach der letzten Write-Operation“.

Availability

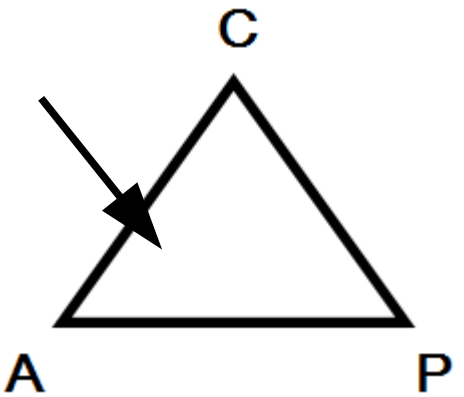
„Jede Read Operation bekommt eine fehlerfrei Antwort
- ohne Garantie auf die aktuellsten Daten“.

Partition Tolerance

„Das verteilte System funktioniert auch noch wenn Nachrichten zwischen den Knoten durch Netzwerkfehler verloren gehen“.

CAP Theorem

„Man kann höchstens zwei der drei Eigenschaften Consistency, Availability und Partition Tolerance mit einem verteilten System erfüllen.“



Dies wurde erstmals 2000 von Eric Brewer vermutet,
2002 folgte der axiomatische Beweis durch Gilbert und Lynch.

CA-Systeme (Relationale Datenbanken)

Solange es zu keiner Partition kommt, kann C und A garantiert werden.

CP-Systeme

CP-Systeme opfern im Angesicht einer Netzwerkpartition
die Availability um konsistent zu bleiben.

AP-Systeme (Vielzahl der NoSQL Systeme)

AP-Systeme opfern im Angesicht einer Netzwerkpartition
die Konsistenz um hochverfügbar zu bleiben.

(relationale) ACID vs. BASE (NoSQL)

Atomicity

Consistency

Isolation

Durability

Basically Available

Soft State

Eventually Consistent

Relationale Systeme → Datenverwaltungssprache: SQL

Flexibilität: Abfragen sind von Datenmodellierung unabhängig, jederzeit neue Abfragen möglich

Relationen erzeugen: CREATE TABLE

Daten einfügen: INSERT

Daten ändern: UPDATE

Daten löschen: DELETE

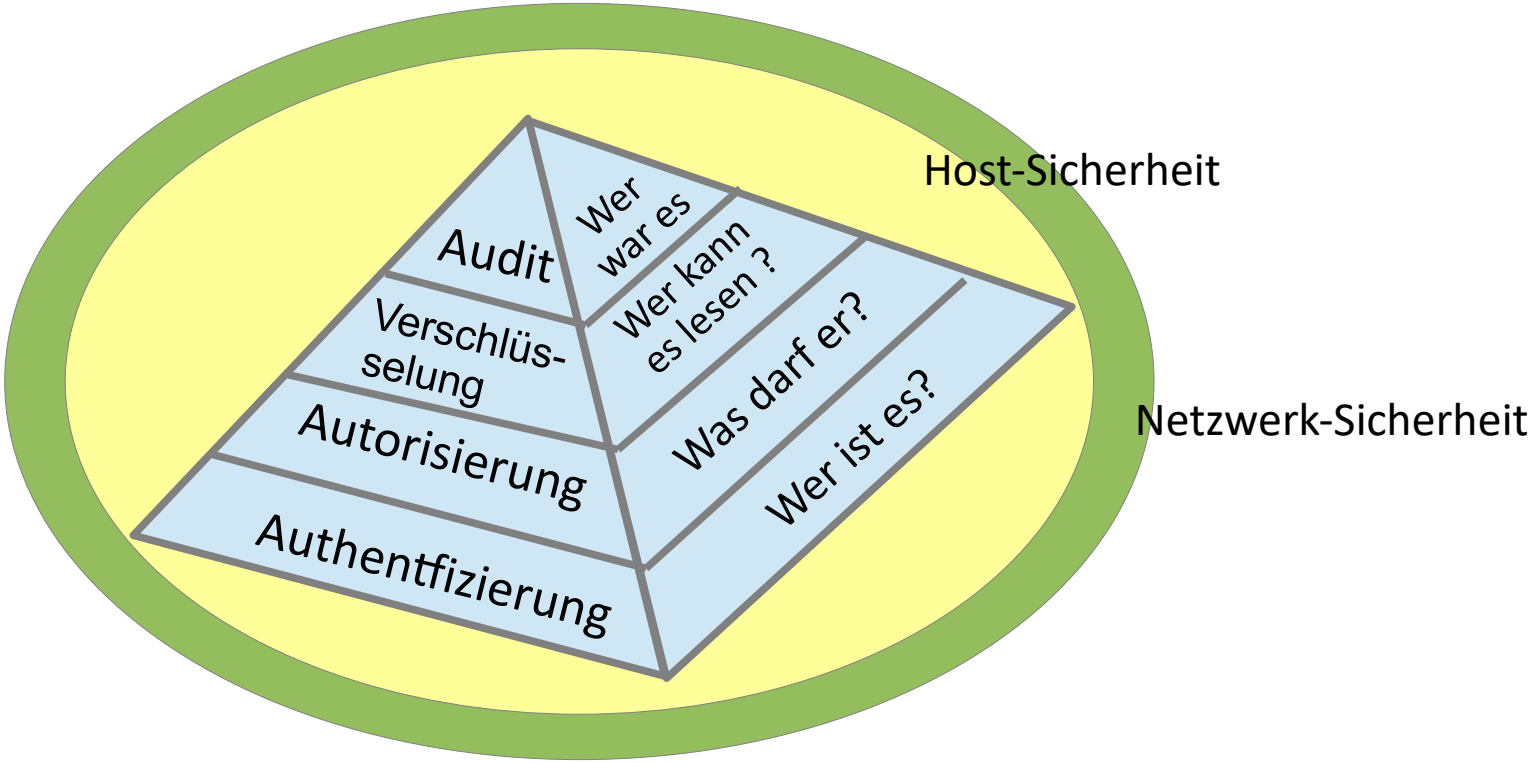
Daten suchen: SELECT

NSQL → Proprietäre Abfragesprachen

Wichtigsten Abfragen sollten zur Design-Time bekannt sein.

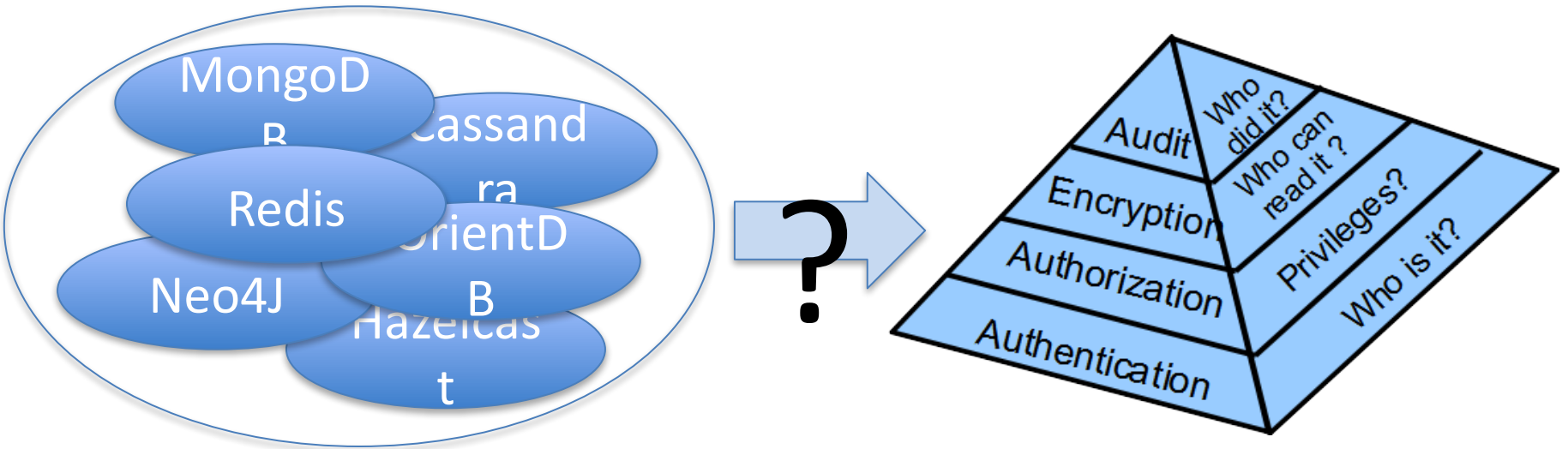
NoSQL Security

Sicherheitheitspyramide Relationaler DBMS



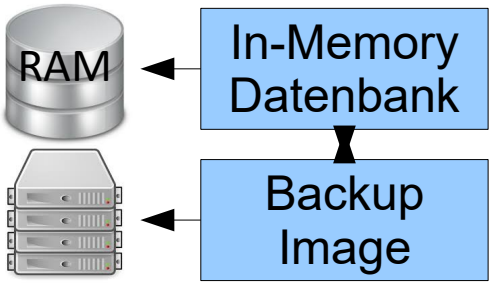
Es sind über 225 NoSQL Systeme verfügbar

(Edlich, S.: NoSQL – List of NoSQL databases, http://nosql-database.org.



redis

Architektur



Redis ist Single-Threaded

-> Empfehlung: Eine Instanz pro Prozessorkern

Memory Footprint:

Empty instance:	~ 3MB
1 Million Small-Keys + String Wert:	~ 85 MB
1 Million Keys + 5 Value Fields:	~ 160 MB

Datenmodel

Redis verwaltet Daten in einem Key-Value Modell:

Beispiele:

Kndn1 -> "Maier"

Kndn2 -> "Huber"

Kndn3 -> "Hauser"

Prdn1 -> "Allzweckreiniger"

Prdn2 -> "Staubsauger"

...

Schlüssel (Key) besteht aus

- Binärdaten bis 512 MB

Werte (Value) sind u.a.:

- Strings
- Listen von Strings (sorted/unsorted)
- Score-Lists
- Hashes
- BitArrays
- Hyperloglogs


cassandra

DataModel

Keyspace: Kundenverwaltung

Zeitstempel

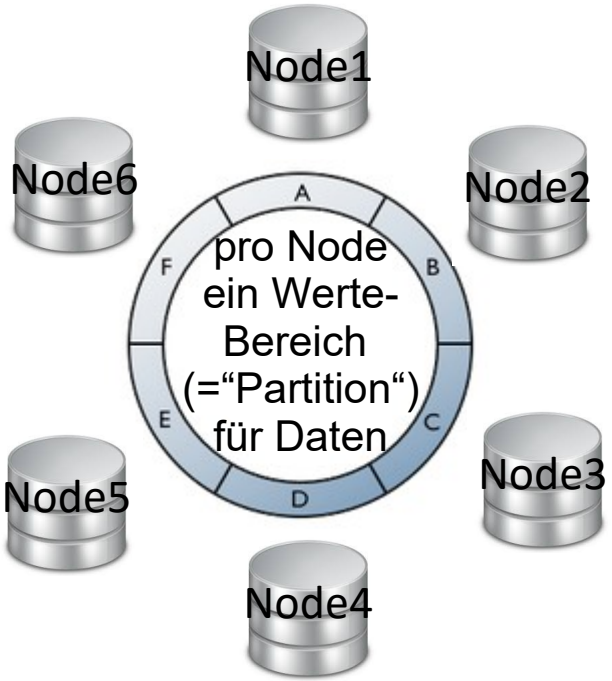
UserID	month	time	text	posted_by
4711	8	12:15	Hallo! Ich ...	2286
1277	6	13:10	Guten M...	
1330	6	13:15		
1330	6	13:16	Servus, w...	1277

Ein Keyspace enthält viele Tabellen

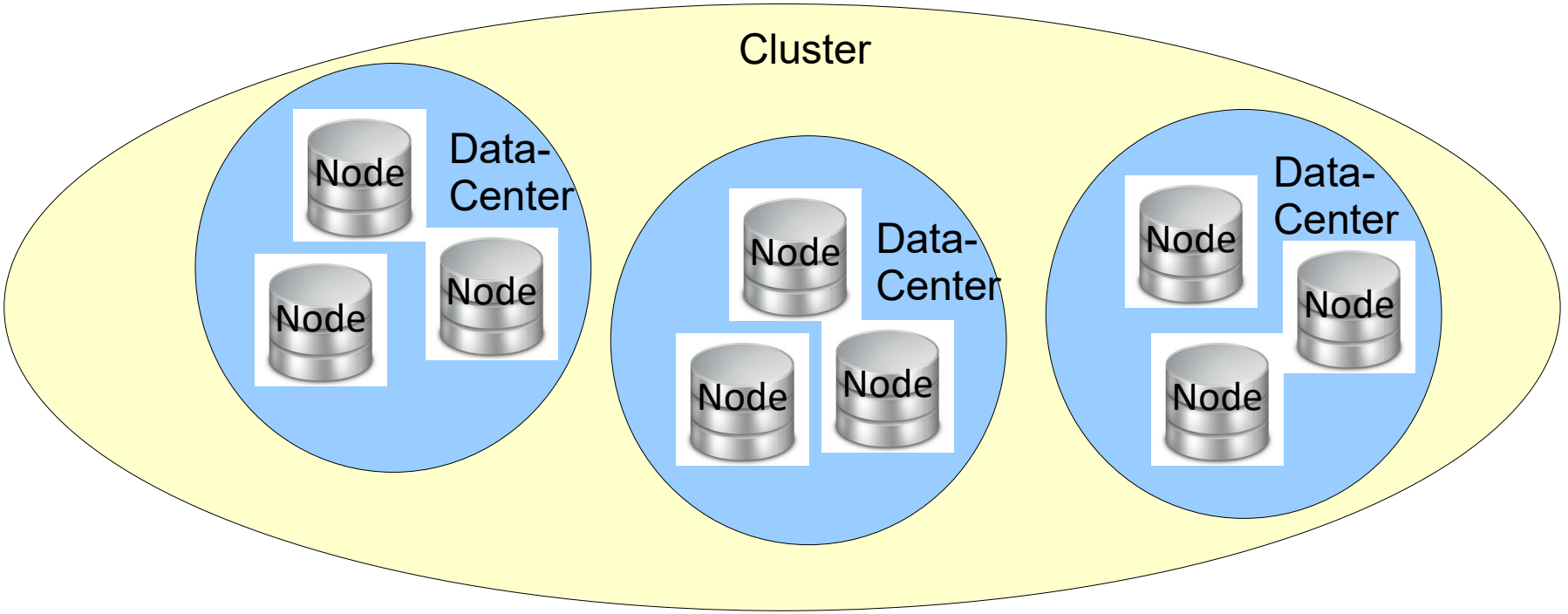
Zeitstempel

Kndnr	Alter	Produktnummern
4711	8	[11201, 33440, 17209, 16520]
1277	6	[33440, 89730, 12020]

Architektur: Verteilung der Daten auf Nodes



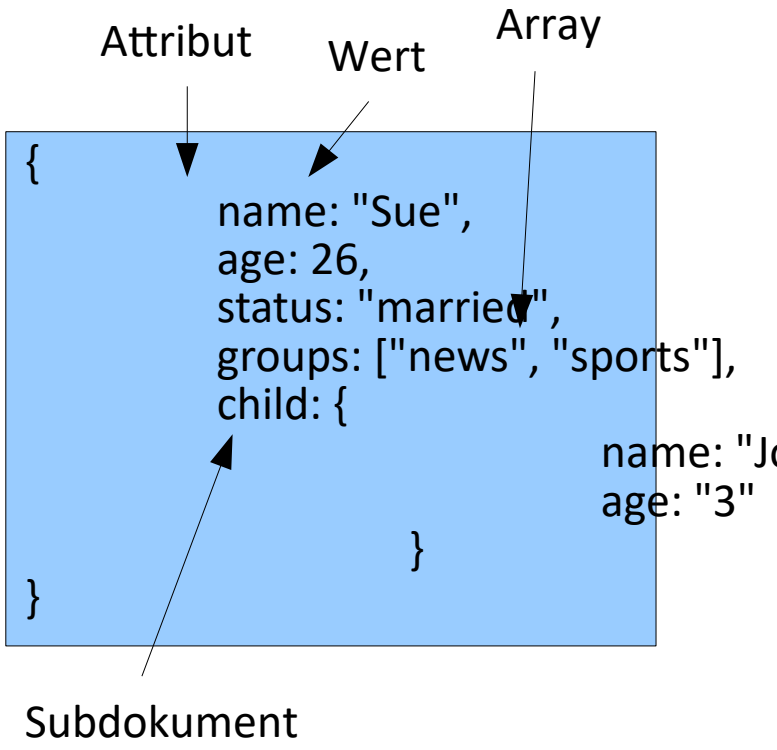
Auf Nodes können auch mehrere Partitionen gespeichert werden.


Cluster
Data-
Center
Node
Node
Node
Data-
Center
Node
Node
Node
Data-
Center
Node
Node
Node


mongoDB.

MongoDB

- Dokumentenorientierte Datenbank
- von engl. „humongous“
- Erstveröffentlichung 2009
- Speichert Collections von JSON-Artigen Dokumenten
- Eine Collection kann völlig unterschiedliche Dokumente enthalten



Architektur

Treiber für:
C/C++/C#
Hadoop
Haskell
Java
node.js
PHP
Perl
Python
Ruby
Scala
...

Client-
Server
Architektur

Mongo DB

Mongo DB

Server Host

Mongo DB

HTTP/REST-API

Mongo-Shell

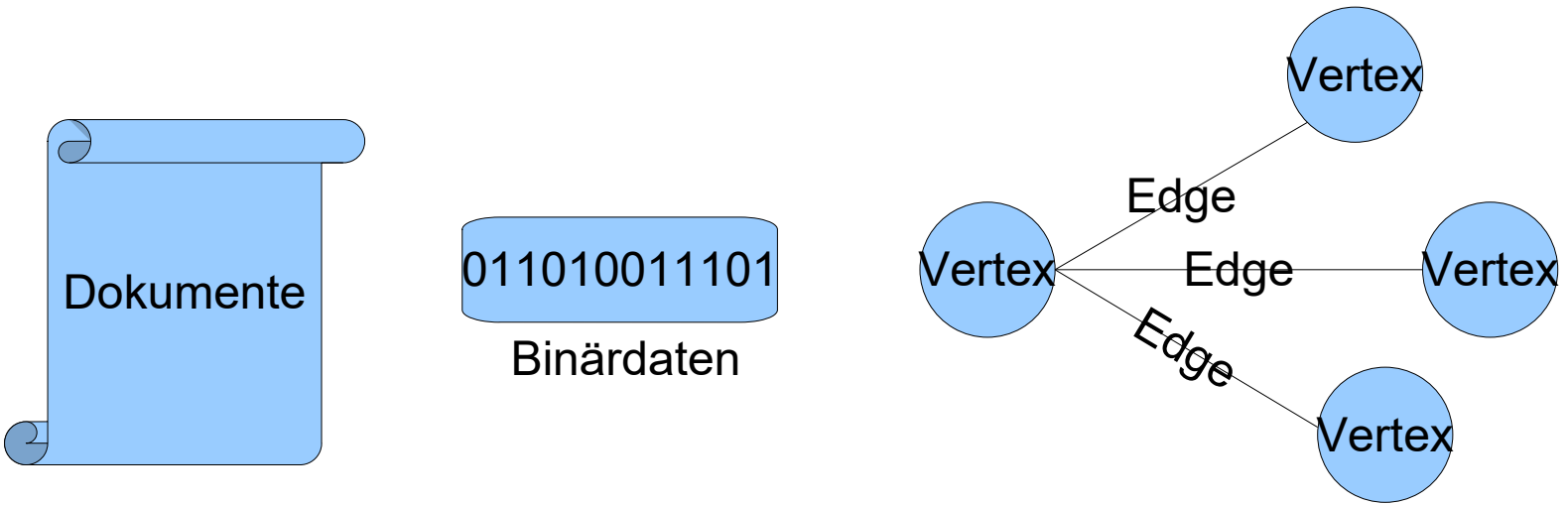

OrientDB®

OrientDB ist eine MultiModel Datenbank

- DocumentModel
- GraphModel
- Key Value Model
- Object Model

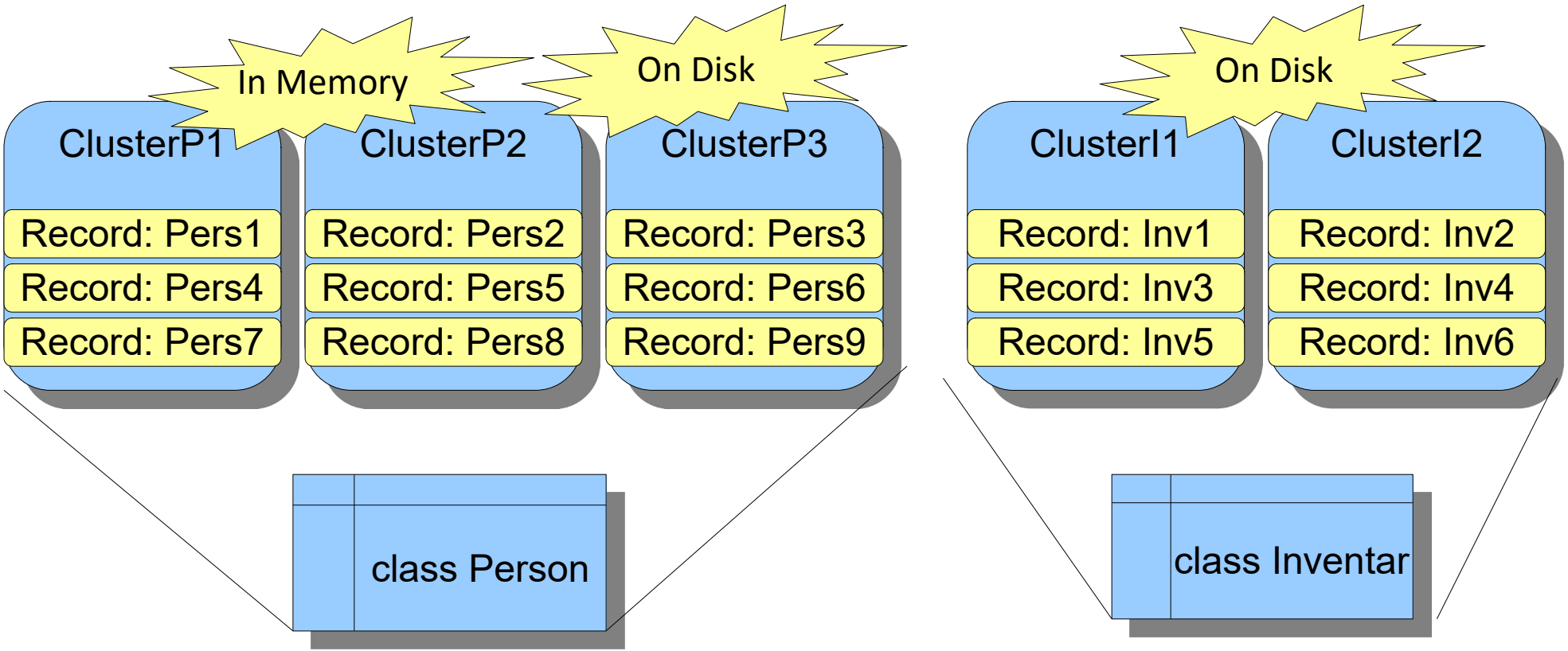
Records

OrientDB speichert Daten in Records, diese können einen von 4 Typen annehmen:

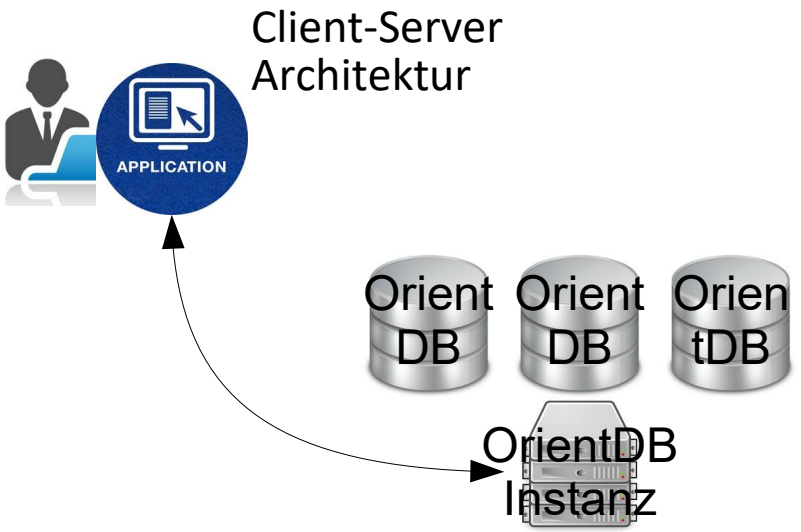


Jeder Record verfügt über eine eindeutige RID über die er im Cluster gefunden werden kann

Zusammenhang von Class, Record & Cluster



Architektur



multiple Datenbanken pro Instanz

HTTP-API	Treiber für: JDBC Gremlin NodeJS PHP .NET Python GO JavaScript Ruby Scala Perl ...
JAVA-API	

	Authentication		User Administration	
	Client to server	**Server to server**	**Role Options**	**Role Scope**
Redis	Disabled by default; Custom user/password	Shared keyfile	Not supported	Not supported
Cassandra	Disabled by default; Default super user; Custom user / password	SSL	Built-in roles and custom roles	Keyspace, table and function
MongoDB	Disabled by default; Custom user/password, X509, SCRAM-SHA-1, MongoDB-CR and SSL; Enterprise Edition: LDAP proxy and Kerberos authentication	Shared keyfile and SSL; X.509 certificates to verify membership in a cluster	Multiple roles per user; built-in roles and custom roles	Cluster, database and collection
OrientDB	Disabled by default; Admin user per database; Custom user/password and SSL; Enterprise edition: symmetric key authentication	Cluster group identifier and password	Multiple roles per user, role inheritance, built-in roles and custom roles	Cluster, database, collection, object and field

	Authorization	Password security	Audit & Log management
Redis	Not supported (default) Optional ACL since Redis 6	Password is set in clear text and sent unencrypted by the auth command.	Not supported Enterprise Edition: Log Files
Cassandra	Role-based authorization	Default password for admin user; Credentials are stored encrypted.	Community edition: configurable event logging or write authenticator and add logging where desired. Commercial version offers auditing features.
MongoDB	Role-based authorization	Using default authentication option (SCRAM-SHA-1): providing per-user random salts, strong hash function and SHA-1 usage	Community edition: logging server activity to unencrypted file, monitoring utilities and reporting statistics. Enterprise edition: auditing authentication, authorization and CRUD operations; more advanced security logging
OrientDB	Security model based on concept of users and roles; LDAP import of users and roles is a standard feature.	Default password for admin user; Database user passwords are stored using the PBKDF2 hash algorithm.	Community edition: not supported. Enterprise edition: configurable event logging and audit log

	Encryption	
	Data-in-transit	**Data-at-rest**
Redis	Not supported; Users are encouraged to encrypt traffic between servers and clients by using an SSL tunnelling software.	Not supported
Cassandra	Disabled by default; Via SSL: client-to-node and node-to-node encryption; Supported protocols and cipher suites are configured in the YAML configuration.	Disabled by default; Currently, two file types are supported.
MongoDB	Disabled by default; Via SSL: OpenSSL libraries are used; Only SSL ciphers with a minimum of 128-bit key length are allowed for connections.	Enterprise feature only; Multiple storage engines supported. Default encryption mode is AES256-CBC (or 256-bit Advanced Encryption Standard in Cipher Block Chaining mode) via OpenSSL.
OrientDB	Disabled by default; Via SSL	Disabled by default; Supported algorithms: AES and DES. Encryption key must be provided at run-time. Operates at database or cluster level.

NoSQL Security zusammengefasst…

- Sehr unterschiedlich – je nach System
- Security Features sind limitiert
  - Oft nur in Enterprise Edition
  - Manchmal auch gar nicht vorhanden
  - Auch Open Source RDBMS haben nicht immer Audit/Verschlüsselung
- Default Setups nicht verwenden!
- Bauen Sie NoSQL Security Expertise auf

Referenz:

Zugaj, W., & Beichler, A. (2018).
Towards a NoSQL security map.

In B. Andersson, B. Johansson, S. Carlsson,
C. Barry, M. Lang, H. Linger, & C. Schneider (Eds.),
Designing Digitalization (ISD2018 Proceedings).
Lund, Sweden: Lund University.

ISBN: 978-91-7753-876-9.

http://aisel.aisnet.org/isd2014/
proceedings2018/ISDevelopment/11

DB-Engines: Überblick & Verbreitung

https://db-engines.com/de/ranking
https://statisticsanddata.org/data/most-popular-databases-2006-2022

Fragen, Diskussion, Feedback

