

ORACLE®

NOSQL DATABASE

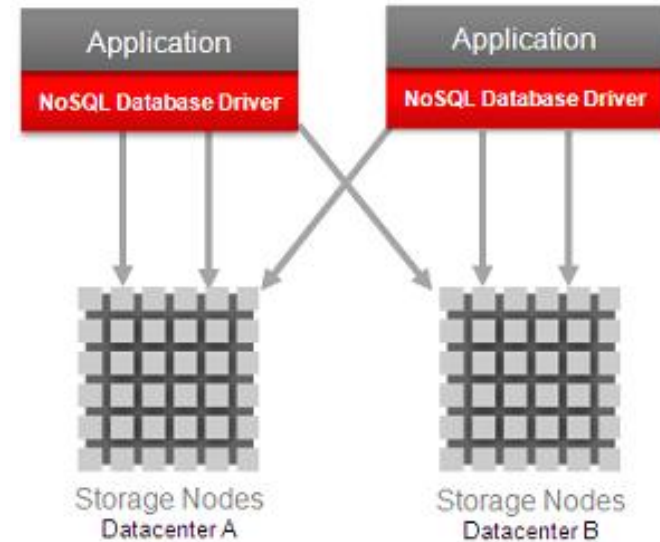
Profil der Oracle NoSQL Database

Carsten Czarski

Oracle Deutschland B.V. & Co KG

Oracle NoSQL Database

- **Key-Value Datenhaltung**
 - Basiert auf BerkeleyDB JE HA
 - Konsistenz und Persistenz konfigurierbar
 - ACID-Transaktionen möglich
- Extrem skalier- und parallelisierbar
 - Hochverfügbarkeit per Replikation
 - Transparentes Loadbalancing
- Community- und Enterprise Edition
- Einfache Installation und Einrichtung
 - Aktuelle Version: 2.1.24



Was ist ein "Key-Value-Store"?

Key-Value Store CUSTOMERS

Key = Index

Zeilen

Key	Value				
010101010
010101011
...

Datenstrukturen sind *nicht* selbstbeschreibend

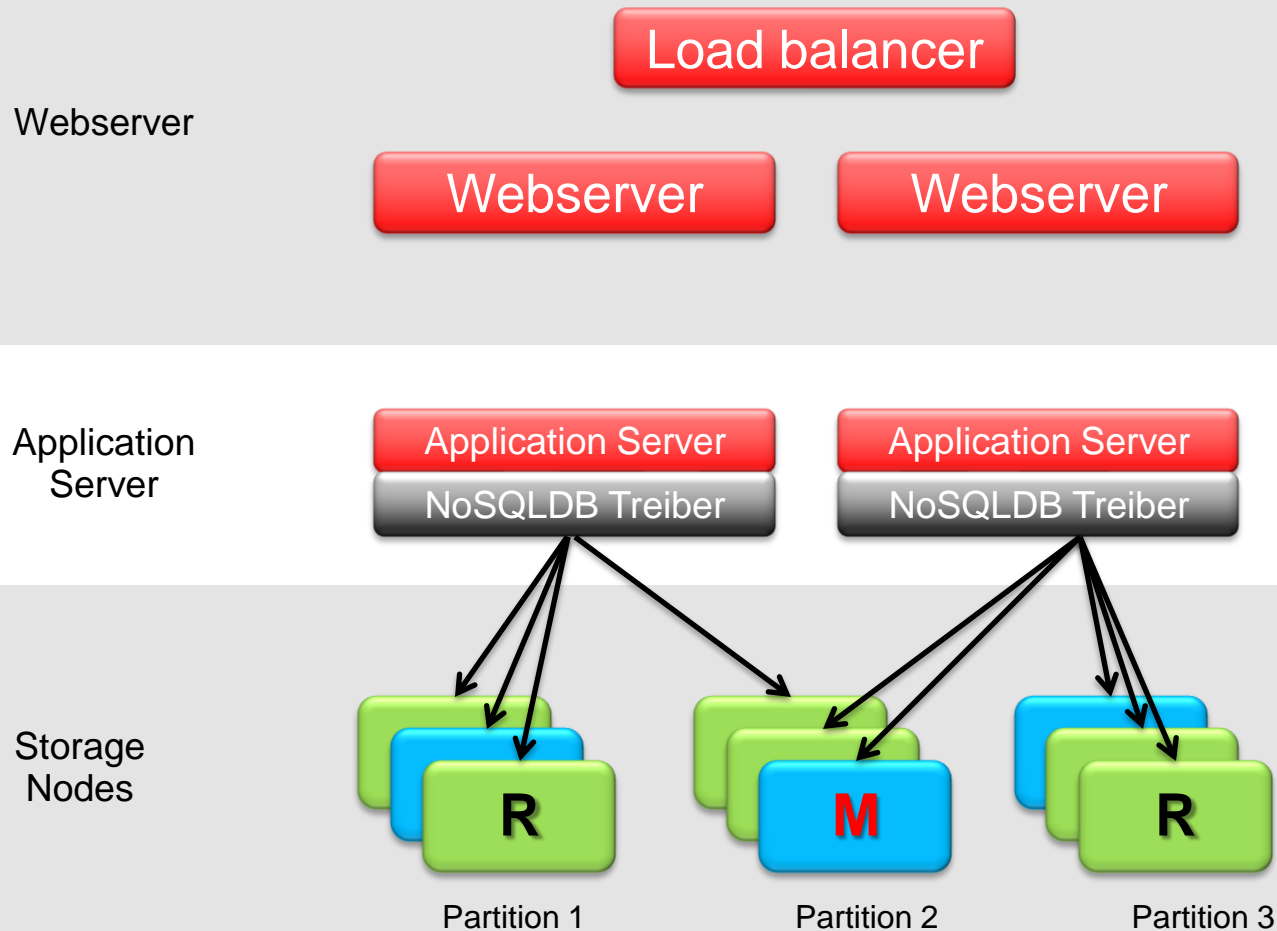
- Im wesentlichen eine zweiseitige Tabelle – "KEY" und "VALUE"
 - VALUE kann auch ein komplexes Objekt sein
 - Die Anwendung kennt die Datenstrukturen – nicht selbstbeschreibend
 - Joins zu anderen Key-Value Stores allein durch die Anwendung
- Einfache Zugriffe: **GET, PUT, DELETE**
- Einfach parallelisierbar

Oracle NoSQL Database

Ein Codebeispiel: Java

```
:  
config = new KVStoreConfig("carstenstore", "sccloud032:5000");  
store = KVStoreFactory.getStore(config);  
  
store.put(  
    Key.createKey("EMP_7839_ENAME"),  
    Value.createValue("KING".getBytes())  
);  
store.put(  
    Key.fromByteArray(new String("EMP_7839_SAL").getBytes()),  
    Value.createValue("5000".getBytes())  
);  
  
store.close();  
:
```

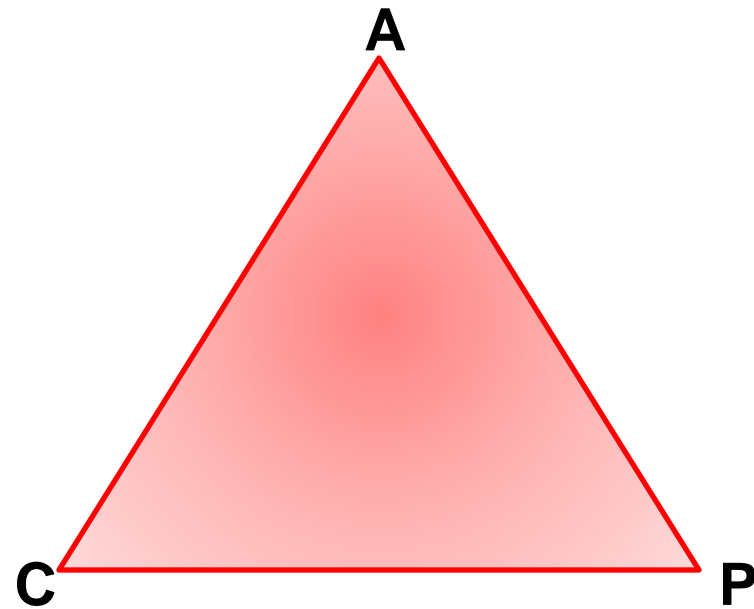
Topologie einer NoSQL Installation



Exkurs: CAP Theorem

Anforderungen an Datenhaltungssysteme

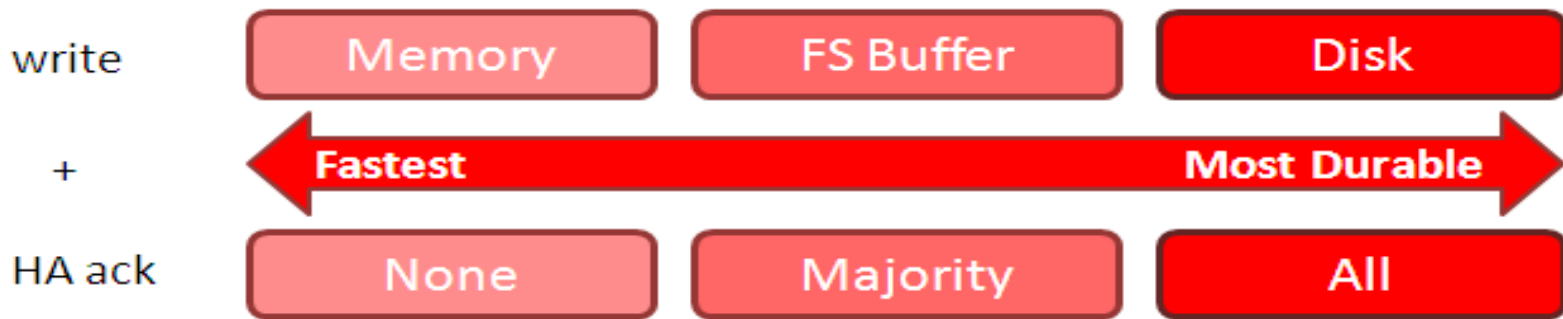
- **Zwei** Anforderungen sind gleichzeitig erfüllbar
 - C = Consistency
 - A = Availability
 - P = Partition Awareness
- RDBMS Oracle ist **CA**
 - Prio 1: Konsistenz und Verfügbarkeit
- NoSQL DB is **CP** oder **AP**
 - Prio 1: Verteilte Systeme



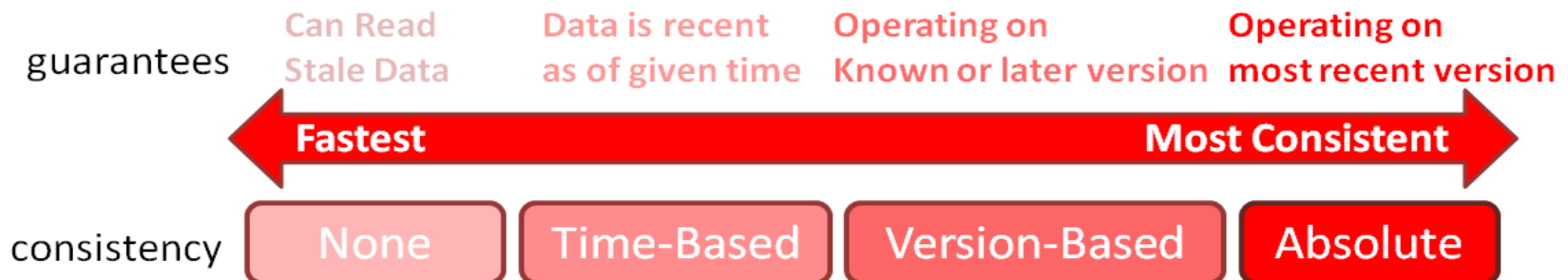
CAP Theorem in der Praxis

Schreib- / Lesekonsistenz wird „verhandelbar“

- Schreibkonsistenz: “C” vs. “A”

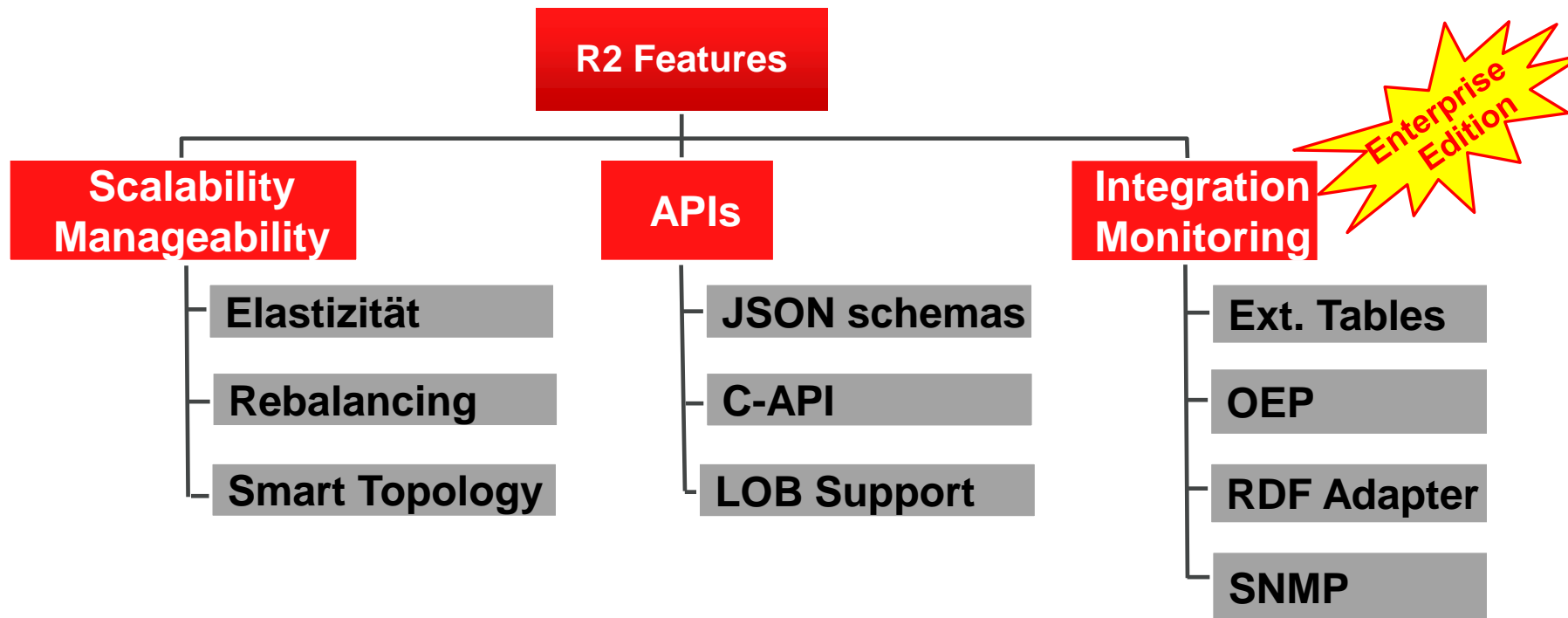


- Lesekonsistenz: “C” vs. “A”



NoSQL DB Version 2

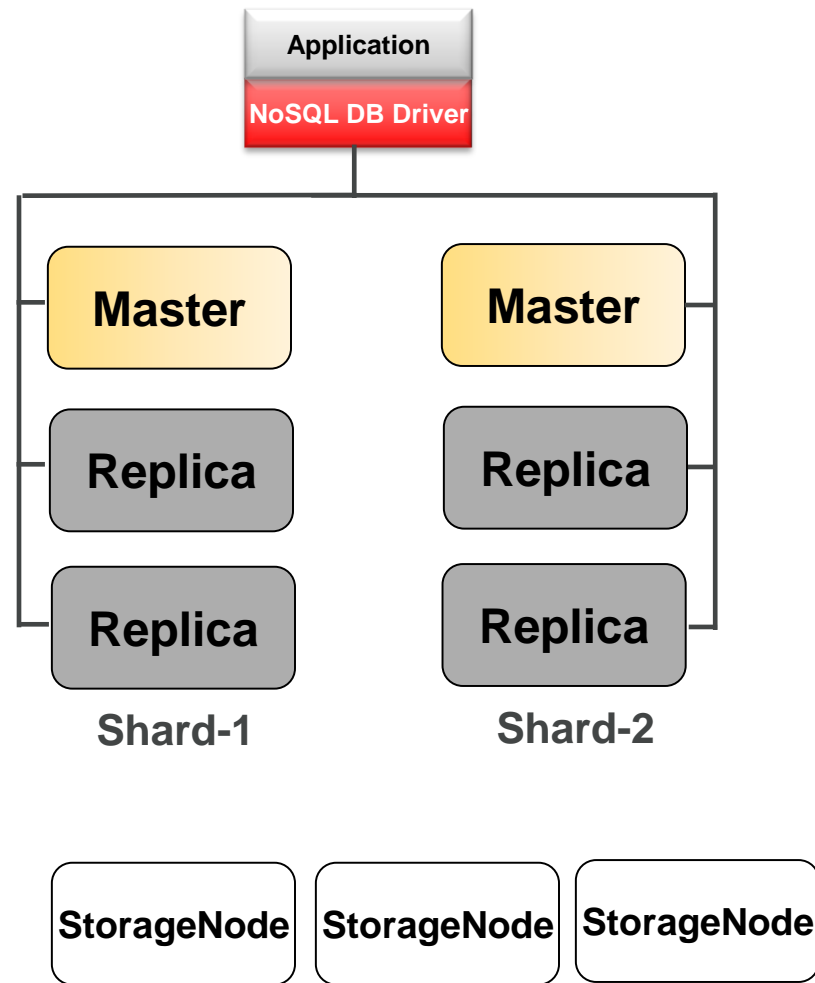
- Aktuelle Version 2.1.24 – November 2013



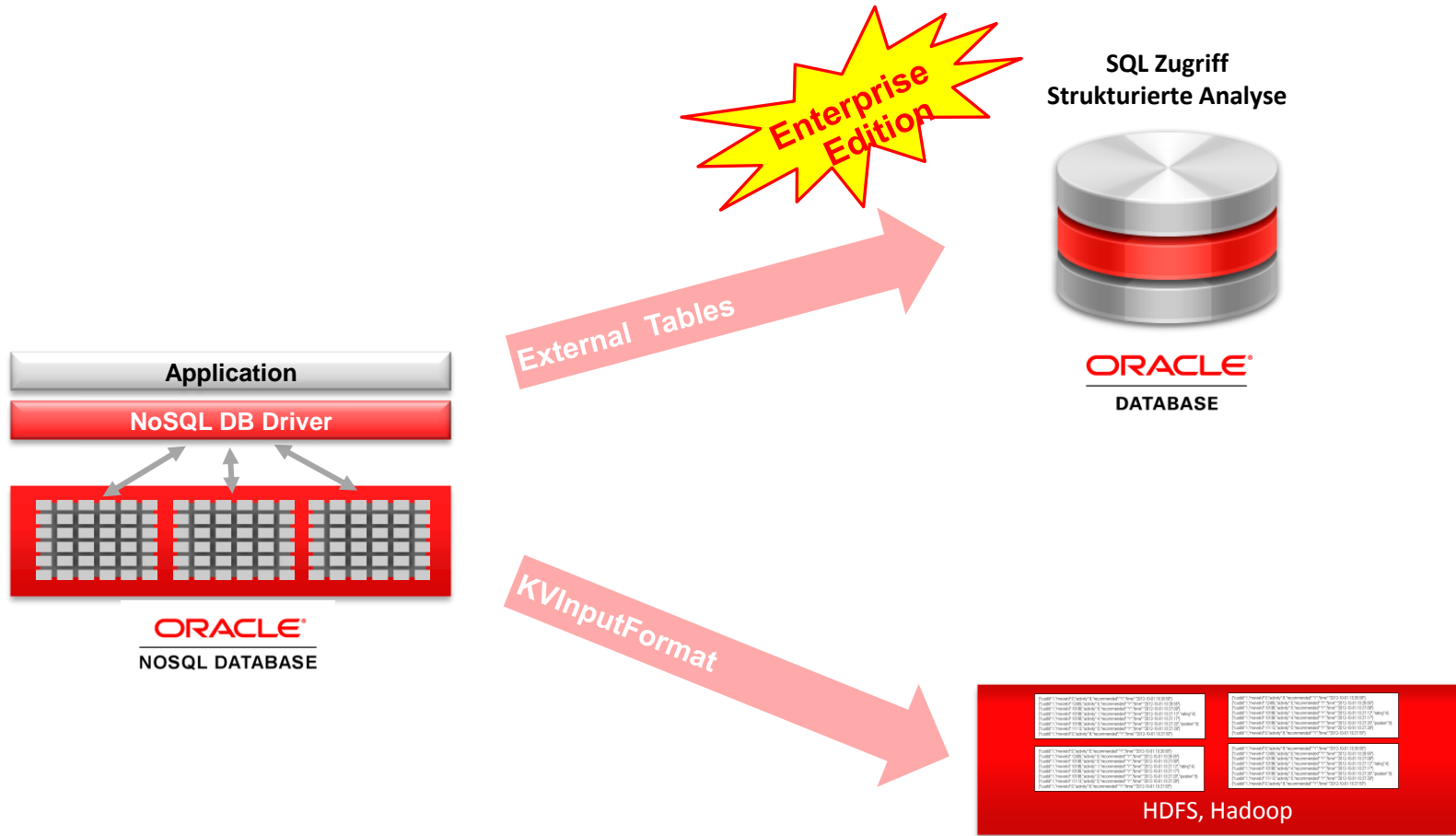
NoSQL DB 2: Elastizität

Änderungen an der Topologie

- Dynamisches Hinzufügen von Storage Nodes
- Dynamisches Ändern der Partitionen

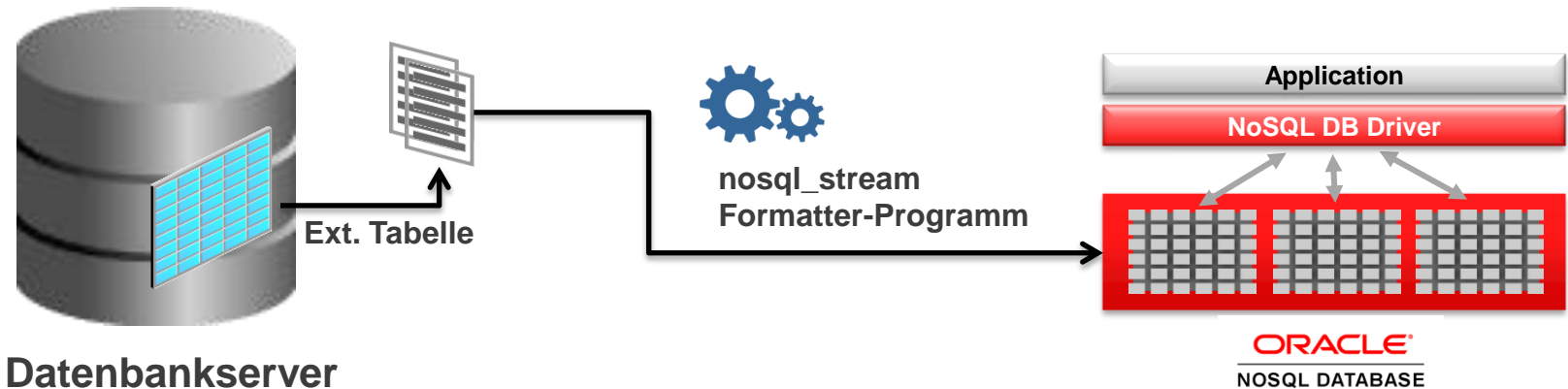


NoSQL DB im Rechenzentrum



Externe Tabellen auf eine NoSQL DB

- Grundlage: Preprocessing external tables
 - Preprocessor-Utility: **nosql_stream**
 - Zusammenspiel mit *Formatter-Programm*
 - Wandelt KV-Paare der NoSQL DB in Text um
 - Übergabe an SQL*Loader Driver der Datenbank



Weitere Informationen im OTN

Overview Downloads Documentation Community Learn More

ORACLE
NOSQL DATABASE

Oracle NoSQL Database

The Oracle NoSQL Database is a distributed key-value database. It is designed to provide **highly reliable, scalable and available** data storage across a configurable set of systems that function as storage nodes.

Data is stored as key-value pairs, which are written to particular storage node(s), based on the hashed value of the primary key. Storage nodes are replicated to ensure high availability, rapid failover in the event of a node failure and optimal load balancing of queries. Customer applications are written using an easy-to-use Java/C API to read and write data.

Oracle NoSQL Driver links with the customer application, providing access to the data via appropriate storage node for the requested key. A web based console as well as command line interface is available for easy administration of the cluster.



Oracle NoSQL Database
NoSQL and the Enterprise

News!

- [The Oracle NoSQL Database Community Edition](#), now has commercial support

<http://www.oracle.com/technetwork/products/nosqldb/overview/index.html>



Carsten.Czarski@oracle.com

<http://tinyurl.com/apexcommunity>

<http://sql-plsql-de.blogspot.com>

<http://oracle-text-de.blogspot.com>

<http://oracle-spatial.blogspot.com>

<http://plsqlxecoscomm.sourceforge.net>

<http://plsqlmailclient.sourceforge.net>

Twitter: @cczarski @oraclebudd