




Infinispan - NoSQL für den Enterprise Java Alltag

Wer schnell und einfach in bestehende Enterprise Java Anwendungen eine NoSQL-Datstore integrieren möchte dem bietet JBoss mit dem Projekt Infinispan eine Lösung an.

Die Session zeigt wie man mit NoSQL als komplementärem Denkansatz und Infinispan als elegantem Werkzeug auch im "relational dominierten JEE-Umfeld" Performanz und Verfügbarkeit von Anwendungen verbessern kann. Infinispan ist das NoSQL-Datstore-Projekt aus dem Hause JBoss und als distributed transactional Cache gemäß JSR-107 umgesetzt.

Orientation in Objects GmbH
 Weinheimer Str. 68
 68309 Mannheim
 www.oio.de
 info@oio.de

Version: 1.1



Gliederung

- NoSQL und Java EE
- Infinispan
- Integrationsszenarien
- Zusammenfassung

© 2013 Orientation in Objects GmbH

Infinispan - NoSQL für den Enterprise Java Alltag | 2

Gliederung

- **NoSQL und Java EE**
- Infinispan
- Integrationsszenarien
- Zusammenfassung




© 2013 Orientation in Objects GmbH

Infinispan - NoSQL für den Enterprise Java Alltag

3

NoSQL vs. Java EE

- **NoSQL**
 - Datenspeicher ohne festgelegte Tabellenschemata
 - skalieren horizontal
 - keine strenge Spezifikation(viele gut diskutierte/erprobte Konzepte)
 - viele Implementierungen auf verschiedenen Konzepten basierend
- **Java EE**
 - strenge Spezifikation einer Softwarearchitektur
 - transaktionsbasierte Ausführung von Java-Komponenten
 - auf transaktionsbasiertem Konzept beruhende Teilstandards
 - **JTA/JCA/JPA/JMS/JDBC**
 - Horizontale Skalierung als Konzept für High Availability
 - **Antwortzeiten**
 - **Ausfall von Knoten**
 - (insbesondere gedacht für Web-Anwendungen)



© 2013 Orientation in Objects GmbH

Infinispan - NoSQL für den Enterprise Java Alltag

4

NoSQL

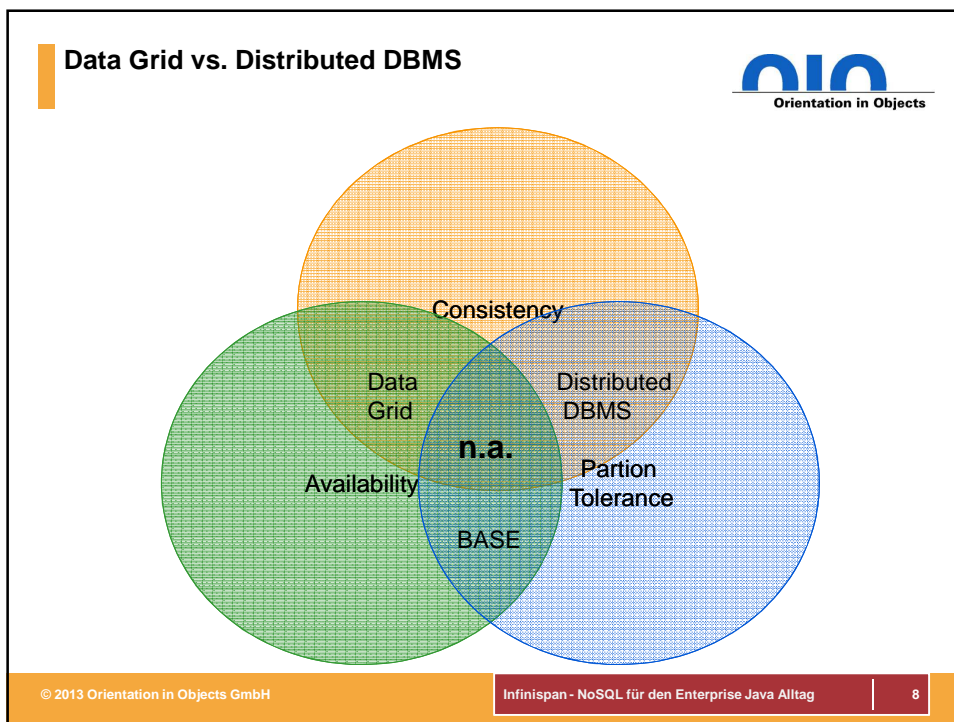
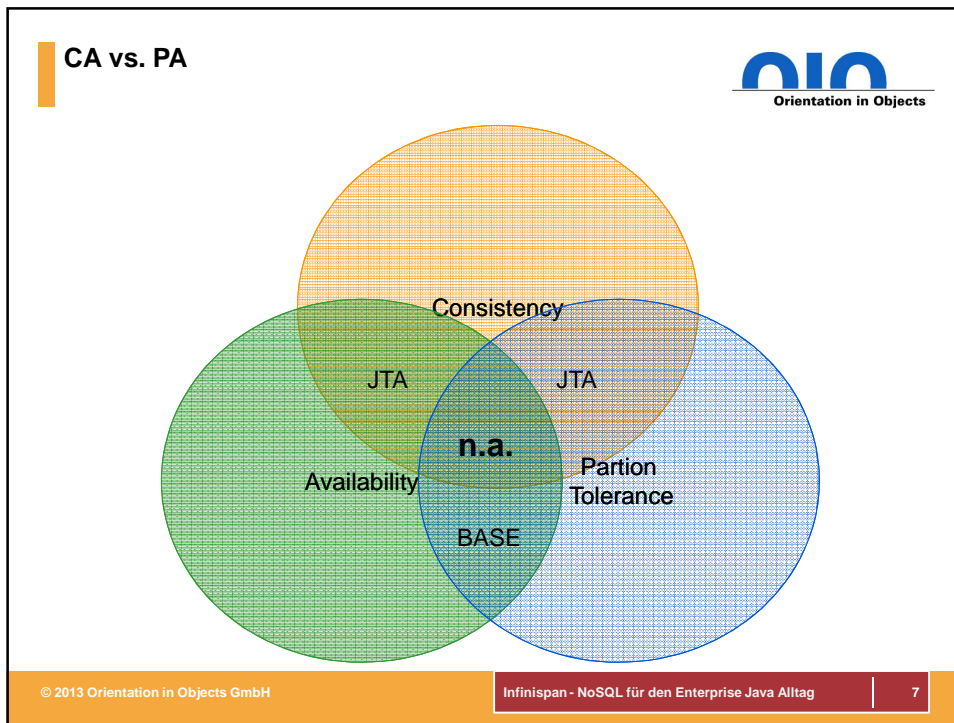


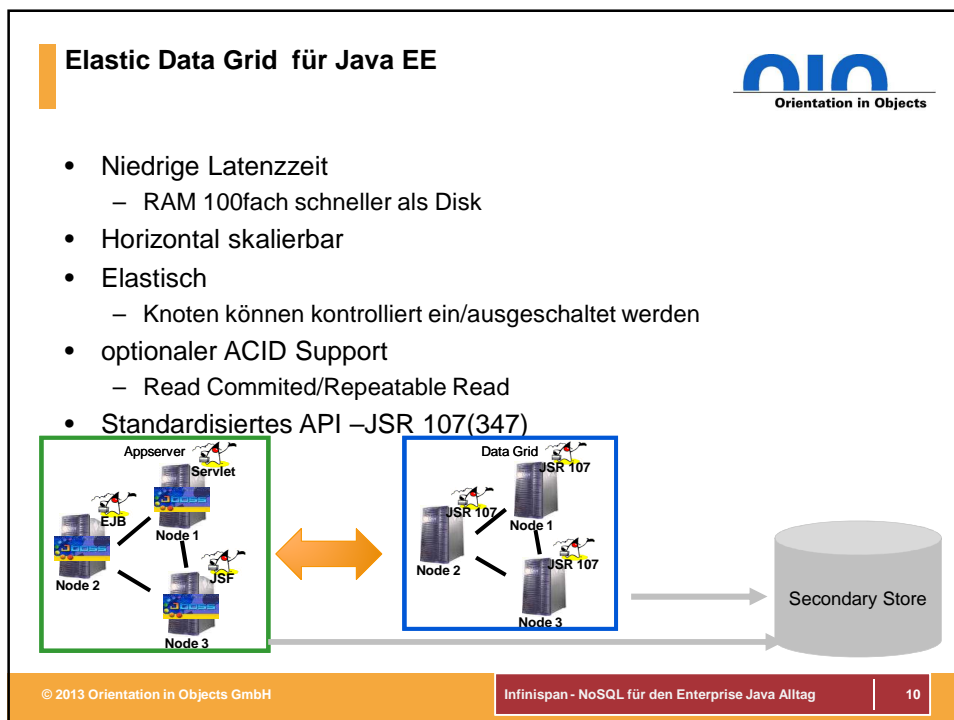
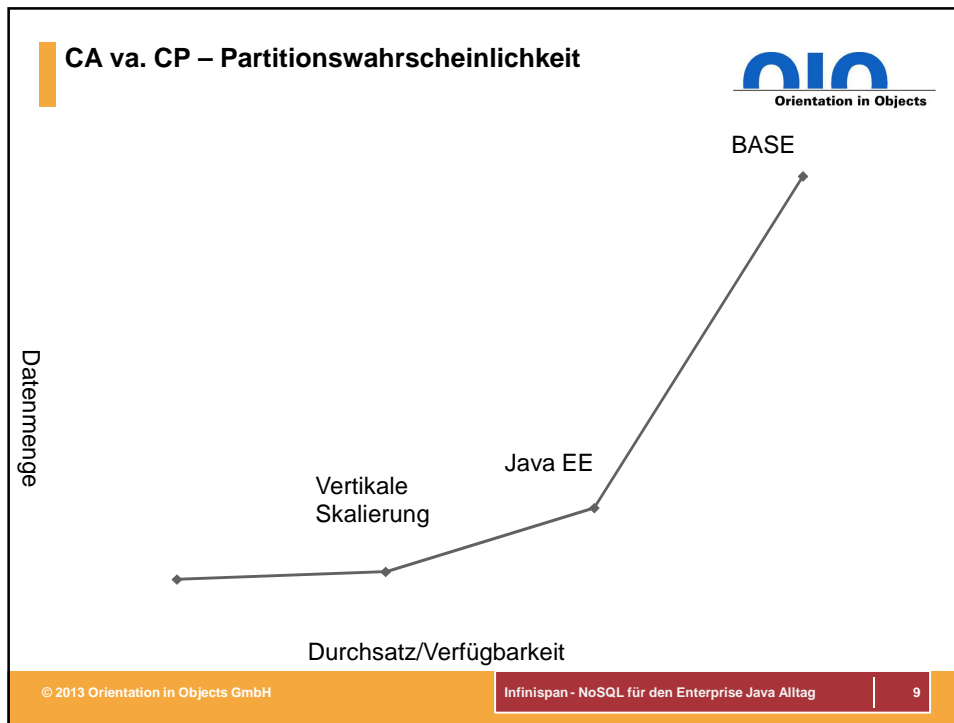
- Dokumentenorientierte Datenbanken
- Graphendatenbanken
- Key-Value-Stores
 - Diskbasiert
 - RAM-Cache
 - Sortierte Key-Value-Stores
- Eventually Consistent Stores
- MultivalueDatenbanken
- Objektdatenbanken
- Spaltenorientierte Datenbank

CAP-Theorem (nach E. Brewer PODC Symposium 2000)



- **Consistency**
 - roughly meaning that all clients of a data store get responses to requests that 'make sense'. For example, if Client A writes 1 then 2 to location X, Client B cannot read 2 followed by 1.
- **Availability**
 - all operations on a data store eventually return successfully. We say that a data store is 'available' for, e.g. write operations.
- **Partition tolerance**
 - *"The network will be allowed to lose arbitrarily many messages sent from one node to another"*
 - if the network stops delivering messages between two sets of servers, will the system continue to work correctly?





Java Distributed HashTable Standard –JSR 107



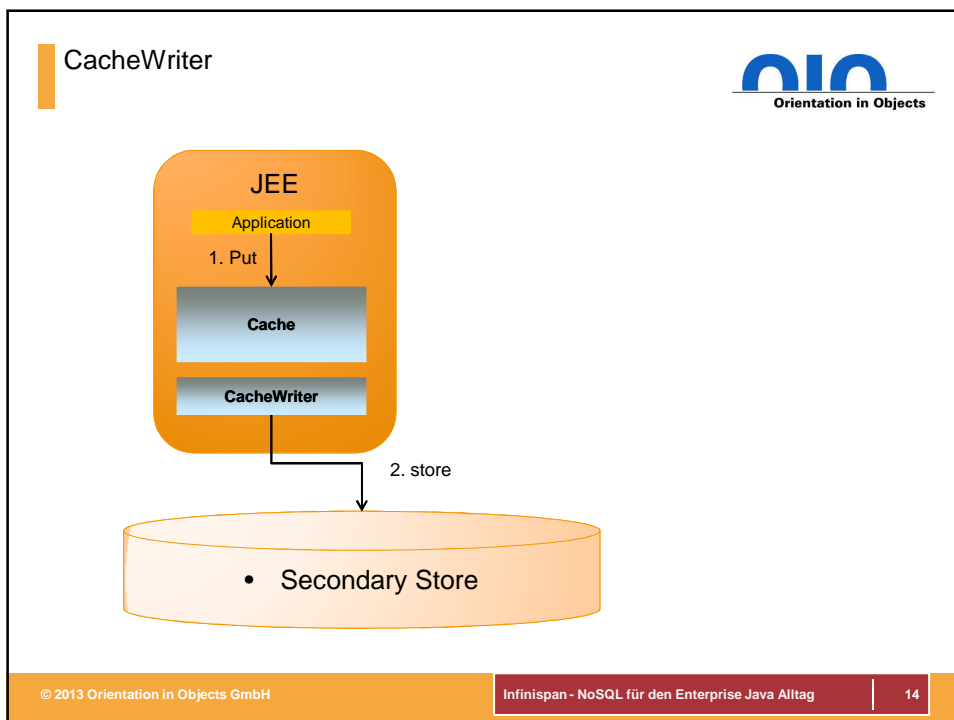
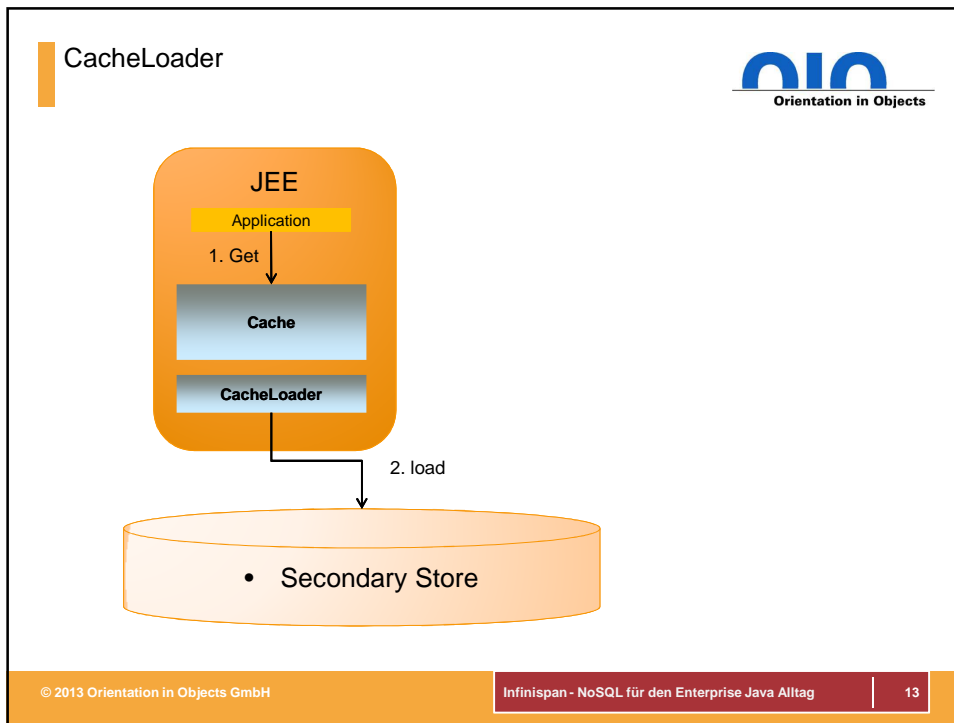
- javax.cache.CacheManager
- javax.cache.Cache
 - ähnlich java.util.ConcurrentMap
 - javax.cache.CacheLoader
 - javax.cache.CacheWriter
- javax.cache..CacheListener

- Annotationen z.B.
 - @CacheResult
 - @CacheKeyParam
 - @CacheRemoveEntry
 - @CacheRemoveAll
 - @CacheDefaults..

Distributed HashTable – nicht in JSR 107



- Standardisierte Konfiguration
- Asynchrone Bearbeitung mit Futures
 - Nonblocking add/remove
- Clustering Details der Zustandsverteilung
 - Replikation
 - Synchron/Asynchron



Expiration

The diagram shows a 'Cache' box containing items M, X, Y, and I. Item D is outside the cache. A clock shows time passing, and a trash can labeled 'F' indicates expiration. Arrows labeled 'get' and 'put' show data flow.

© 2013 Orientation in Objects GmbH

Infinispan - NoSQL für den Enterprise Java Alltag

15

Gliederung


- NoSQL und Java EE
- **Infinispan**
- Integrationsszenarien
- Zusammenfassung

© 2013 Orientation in Objects GmbH

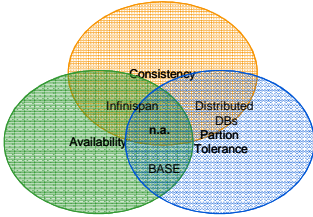
Infinispan - NoSQL für den Enterprise Java Alltag

16

Infinispan



Orientation in Objects

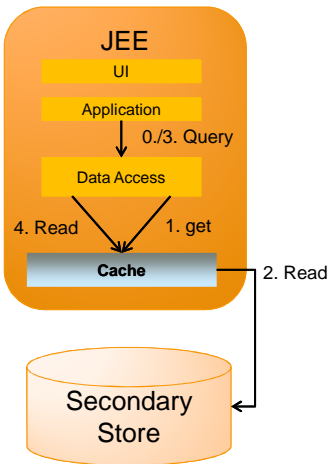
- Key-Value-Store
 - Distributed Hash Table(JSR 107/347)
 - Transactional (JTA)
 - Low-latency (RAM)
 - Optional persist to disk
- Open Source Data Grid Plattform (LGPL)
 - JBoss/Red Hat
- Java and Scala
- Nicht nur für die JVM gedacht
- in verschiedenen Modi einsetzbar
 - Embedded
 - Client/Server(Data Grid)



© 2013 Orientation in Objects GmbH Infinispan - NoSQL für den Enterprise Java Alltag | 17

Local Embedded Mode


Orientation in Objects



© 2013 Orientation in Objects GmbH Infinispan - NoSQL für den Enterprise Java Alltag | 18

Local Embedded Mode II



- Code in der gleiche VM wie infinispán
- wenig Konfigurationsarbeit

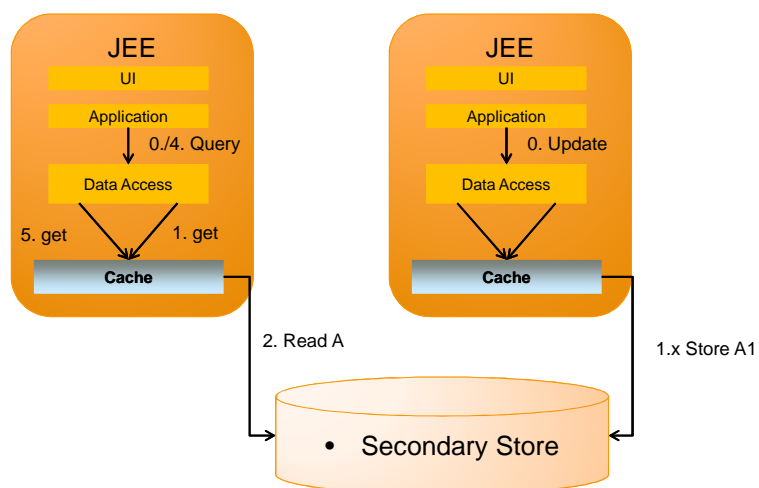
- Mit default Einstellungen

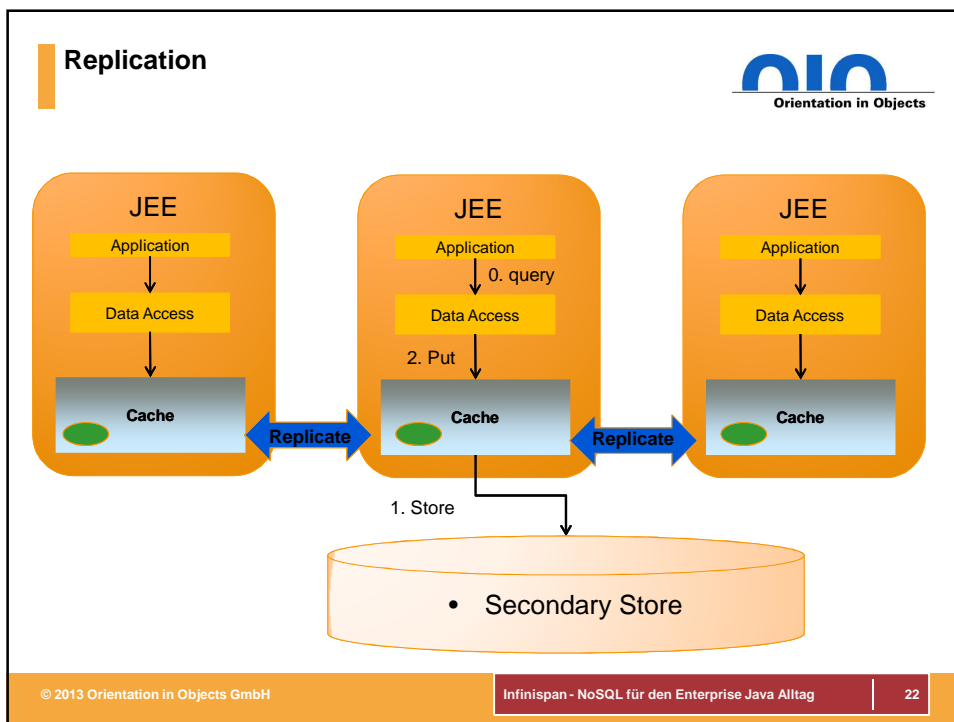
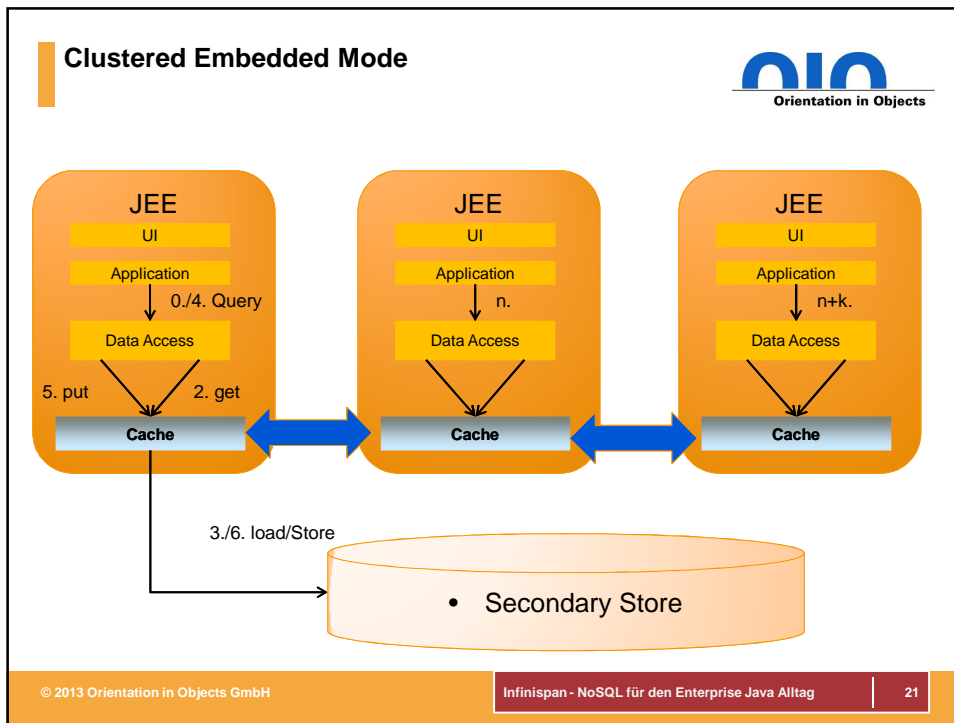
```
DefaultCacheManager cacheManager = new DefaultCacheManager();
Cache<String, String> cache = cacheManager.getCache();
```

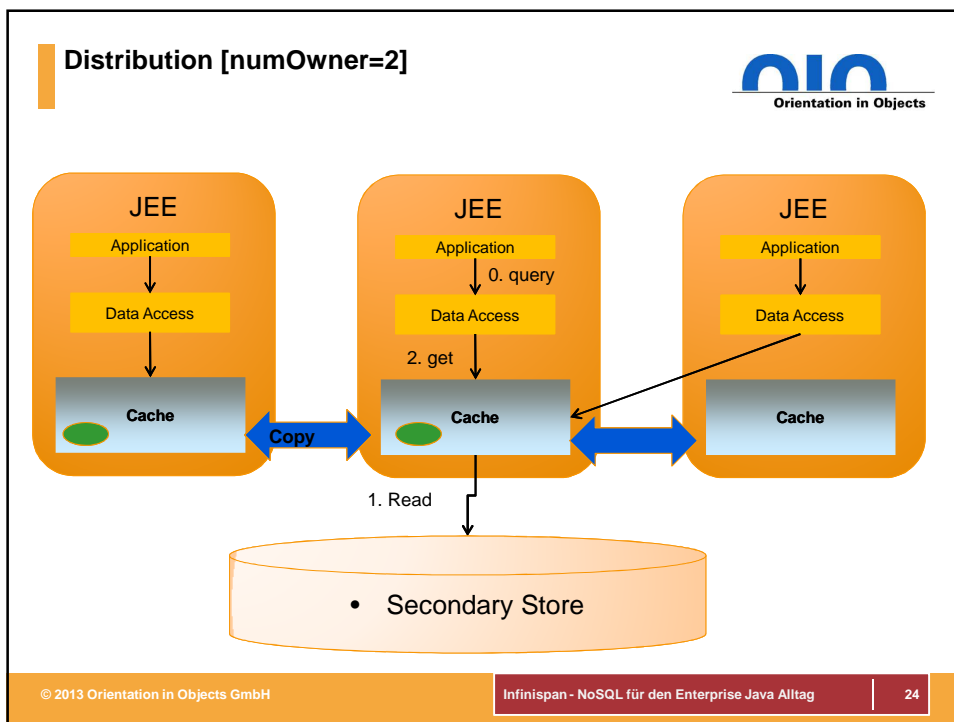
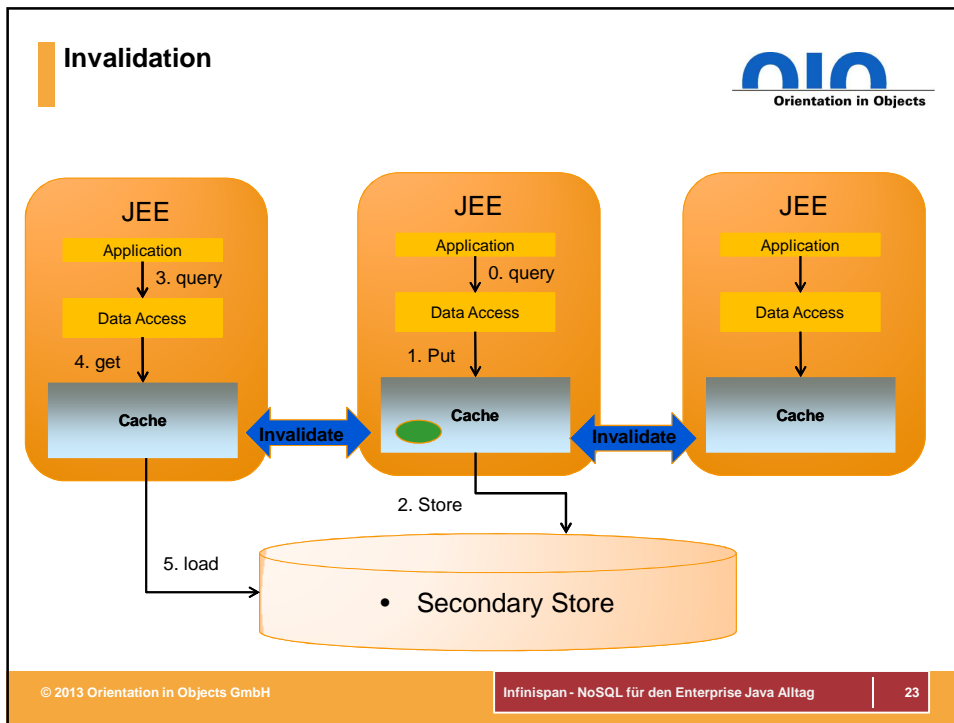
- Mit eigenen Einstellungen z.B: Clustereinstellungen

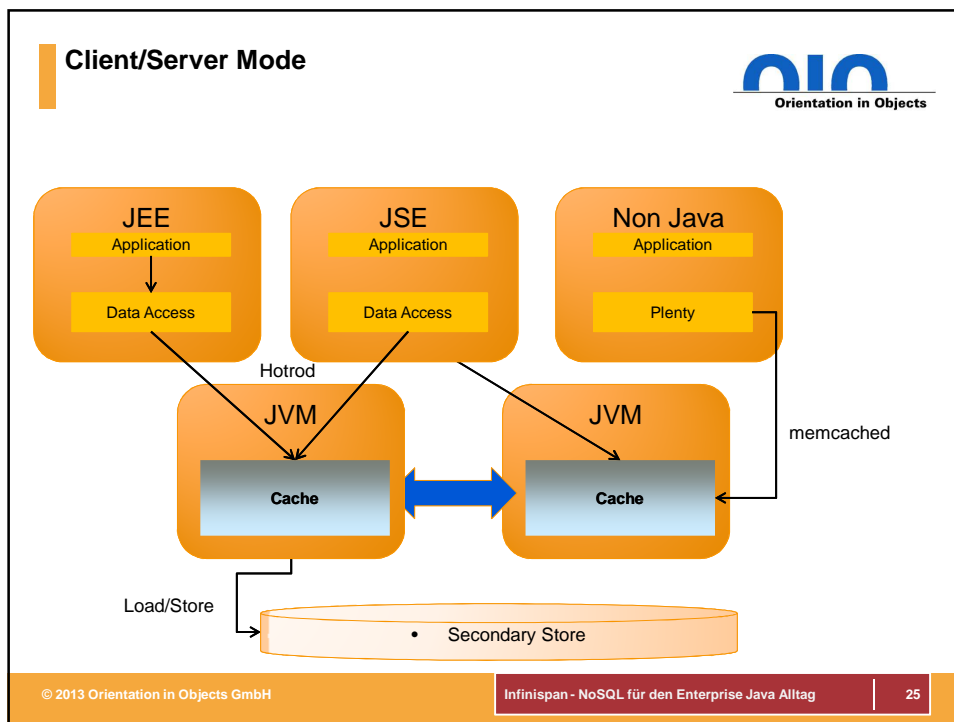
```
DefaultCacheManager cacheManager = new DefaultCacheManager(„oio-
infinispán-config.xml“);
Cache<String, String> cache = cacheManager.getCache();
```

Infinispán und Java EE Clustering









Client/Server Mode II

- Schneller starten und beenden
- Bessere tuning von Hardware und JVM
- Data Grid kann von mehreren Applikation benutzt werden
- Starten des Servers erfordert die Angabe des Protokolls
 - C:\infinispan-5.1.2.FINAL\bin>startServer.bat -r hotrod
- Unterstützte Protokolle
 - Memcached
 - Hot Rod
 - Infinispan REST interface
- org.infinispan.client.hotrod.RemoteCacheManager



```
RemoteCacheManager rcm = new
RemoteCacheManager("192.168.3.120");
RemoteCache remoteCace = rcm.getCache();
String key = "hello";
remoteCace.put(key, "world");
...
```

© 2013 Orientation in Objects GmbH

Infinispan - NoSQL für den Enterprise Java Alltag

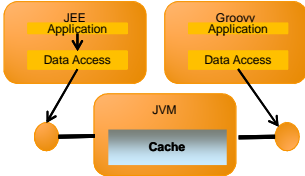
26

Client/Server Remoting




Orientation in Objects

Remote Endpoint	Protokoll	Smart Routing	Load Balancing
Memcached	Text	-	Vordefinierte Liste
ReSTful	Text	-	http Loadbalancer
HotRod	Binary	+	Dynamische Topologie

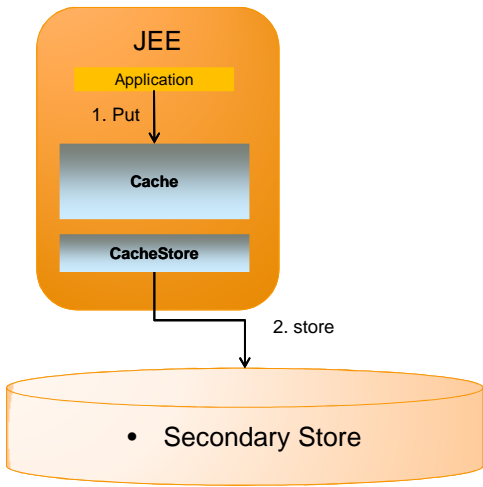


© 2013 Orientation in Objects GmbH
Infinispan - NoSQL für den Enterprise Java Alltag
27

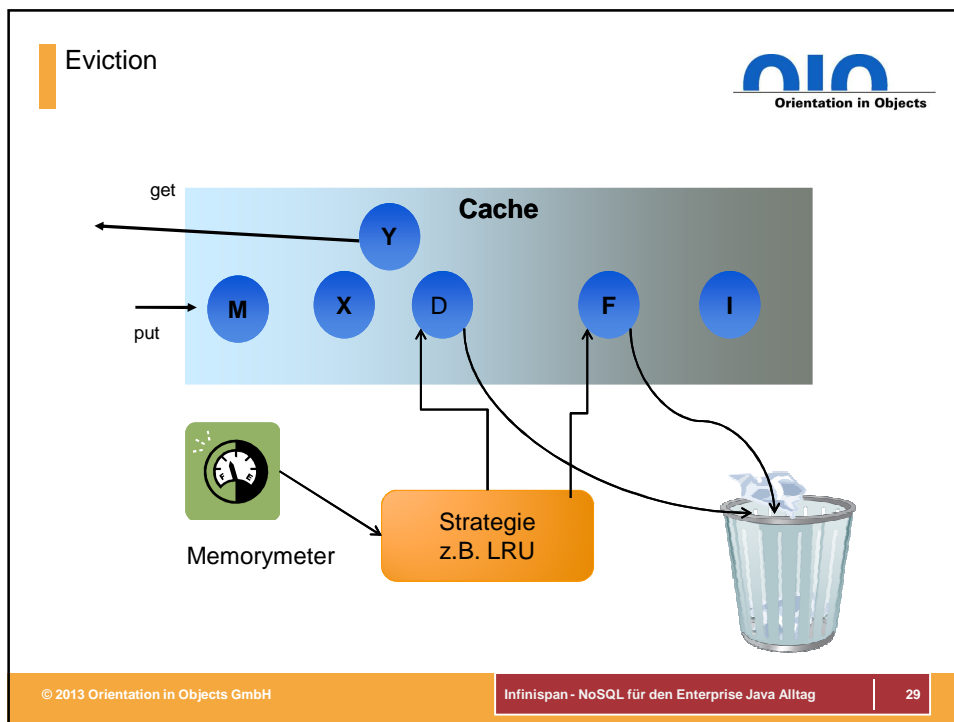
CacheStore



Orientation in Objects



© 2013 Orientation in Objects GmbH
Infinispan - NoSQL für den Enterprise Java Alltag
28



Storage JPA Cache Store

- Implementierung von JPA 2.0
- Speicherung auf eine DB mit definiertem Schema
- Key muss die Id der Entität sein
 - @Id, @EmbeddedId
- Keine generierten Ids

```
Cache<String, User> usersCache =
    cacheManager.getCache("myJPACache"); // configured for User
    entity class
usersCache.put("raytsang", new User());
Cache<Integer, Teacher> teachersCache =
    cacheManager.getCache("myJPACache"); // cannot do this when
    this cache is configured to use a JPA cache store
teachersCache.put(1, new Teacher());
```

© 2013 Orientation in Objects GmbH

Infinispan - NoSQL für den Enterprise Java Alltag

30

Storage JPA Cache Store

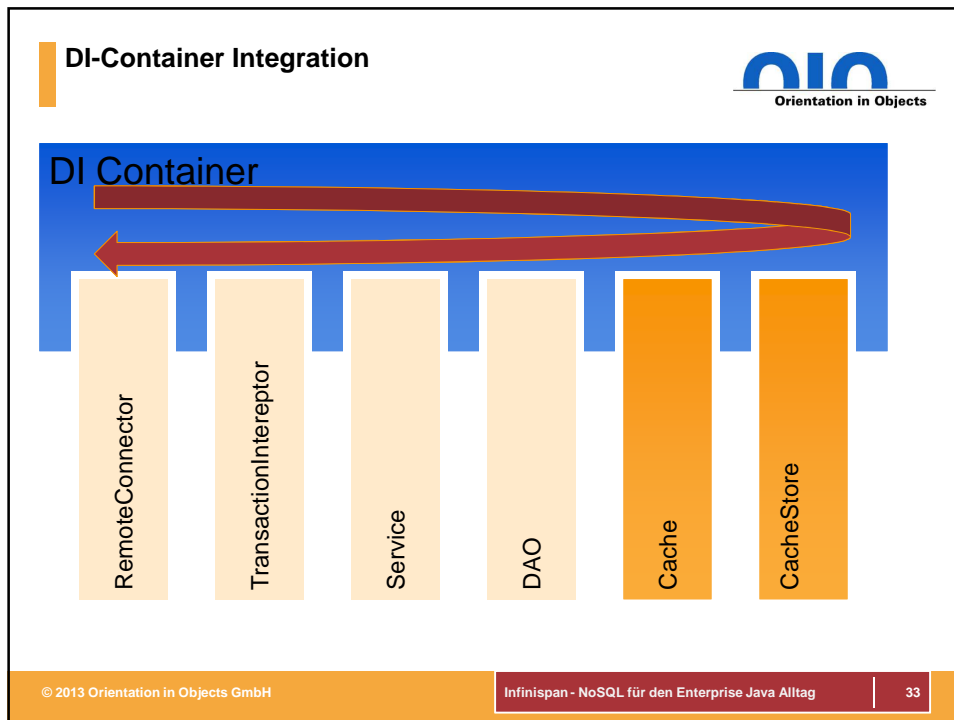


```
<namedCache name="vehicleCache">
  <loaders passivation="false" shared="true" preload="true">
    <jpaStore
      persistenceUnitName="org.infinispan.loaders.jpa.configurationTest"
      entityClassName="org.infinispan.loaders.jpa.entity.User"
    />
  </loaders>
</namedCache>
```

Gliederung



- NoSQL und Java EE
- Infinispan
- **Integrationsszenarien**
- Zusammenfassung



Spring Integration

- seit Spring 3.1 Cache Abstraktion
- Infinispan Provider vorhanden
- Integration mit Spring DI für cache managers und caches

```
<bean id="cacheManager"
class="org.infinispan.spring.provider.SpringEmbeddedCacheManagerFactoryBean"
p:configurationFileLocation="classpath:META-INF/oiio-infinispan-config.xml">
</bean>
```

- Aktive Cache Unterstützung
 - <cache:annotation-driven />
- Deklarative Cache Annotation

```
@Cacheable("nachrichtenCache")
public List<Nachricht> findAlleUngeleseneNachrichten() {...}
```

© 2013 Orientation in Objects GmbH | Infinispan - NoSQL für den Enterprise Java Alltag | 34

Spring Integration



```
@Cacheable("nachrichtenCache", key="#projektId")
public List<Nachricht> findNachrichtenFuerProjekt(BigInteger
    projektId) {...}
```

- Deklarative Eviction

```
@CacheEvict(value = "nachrichtenCache", allEntries = true)
public void nachrichtMarkieren(BigInteger nachrichtId, boolean
    ungelesen) {...}
```

CDI



- Konfiguration und Injizierung des Infinispan Cache
- Partielle Unterstützung vom JSR 107 Annotationen
- Cache Injizierung

```
@Inject
    private Cache<String, String> cache;
```

```
@Inject
    RemoteCache<String, String> remoteCache;
```

CDI



- Eigene Cache Qualifier

```
@javax.inject.Qualifier
@Target({ElementType.FIELD, ElementType.PARAMETER, ElementType.
METHOD})
@Retention(RetentionPolicy.RUNTIME)
@Documented
public @interface NachrichtenCache {}
```

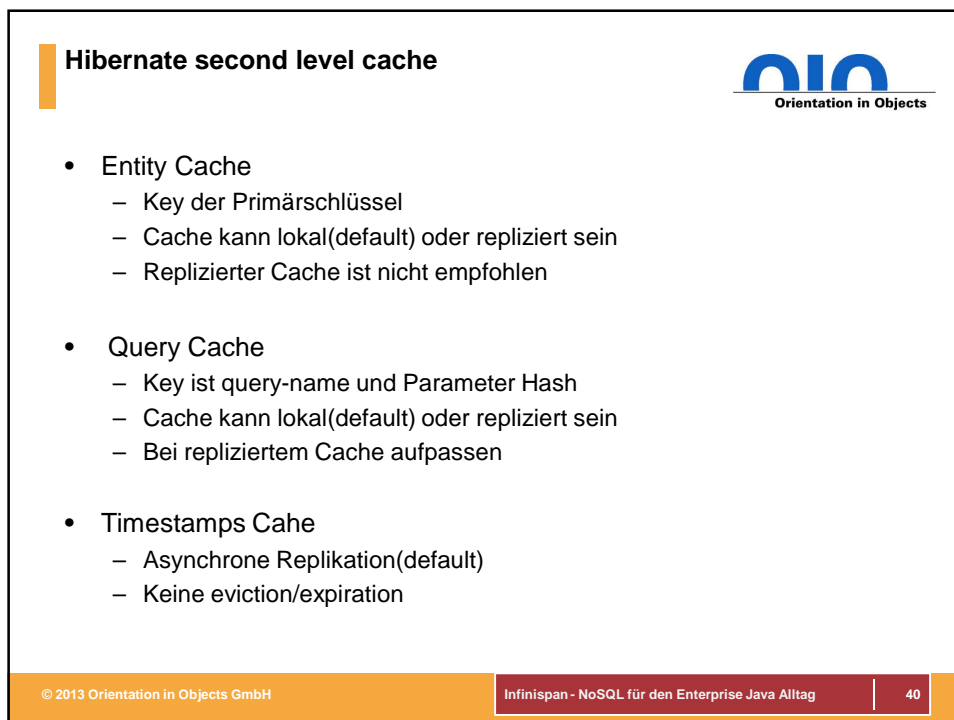
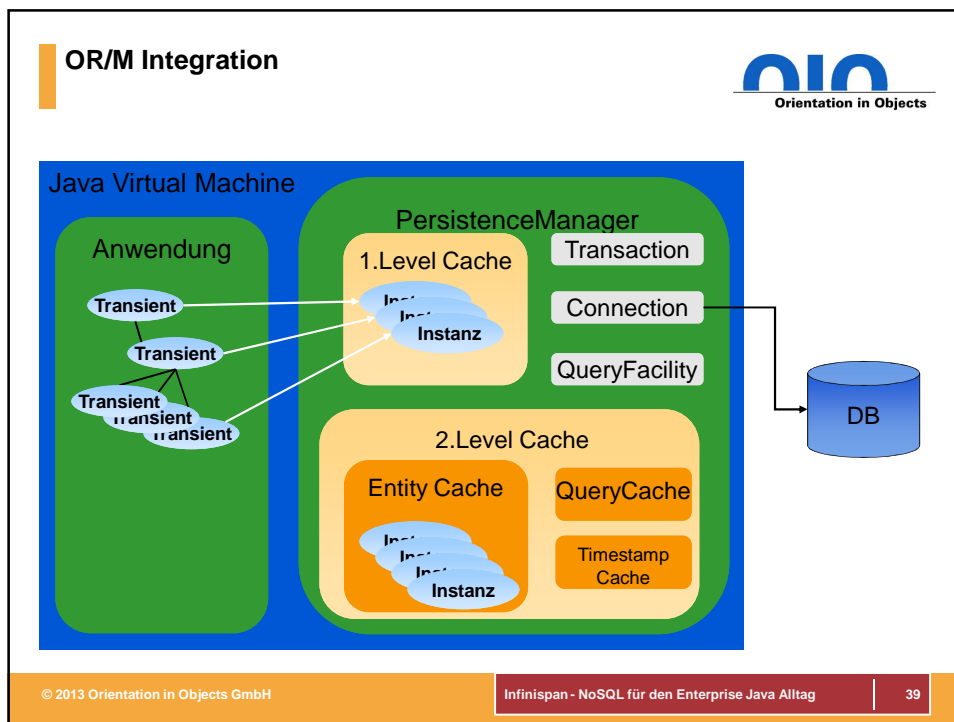
```
public class CacheCreator {
@ConfigureCache("nachrichtencache") // cache name.
@NachrichtenCache // cache qualifier.
@Produces
public Configuration specialCacheCfg() {
return new ConfigurationBuilder()
.eviction()
.strategy(EvictionStrategy.LRU)
.maxEntries(10)
.build();
}
}
```

CDI



- Injizierung des eigenen Caches

```
public class MyCDIComponent {
• @Inject @NachrichtenCache
• Cache<String, String> nachrichtenCache;
• }
```



Hibernate second level cache



- JTA Transaktion Manager
 - Hibernate soll/muss JTA Manager nutzen.

```
<property name="hibernate.cache.use_second_level_cache"
  value="true" />
<property name="hibernate.cache.use_query_cache" value="true" />
<property name="hibernate.cache.region.factory_class"
  value="org.hibernate.cache.infinispan.InfinispanRegionFactory" /
>

<property name="hibernate.cache.infinispan.cfg" value="/oio-
infinispan-config.xml" />
<property name="hibernate.transaction.manager_lookup_class"
  value="org.hibernate.transaction.JBossTSStandaloneTransactionMa
nagerLookup" />
```

Gliederung



- NoSQL und Java EE
- Infinispan
- Integrationsszenarien
- **Zusammenfassung**

NoSQL-Design Prinzipien



- Schnelles, horizontal skalierendes Speichern und Skalieren von ständig wachsenden Immutable-Data (Fakten)
- Die primäre Schreiboperation ist das Hinzufügen von Fakten
- Vermeidung der Komplexität des CAP Theorems durch Neuberechnung aller Queries aus den Fakten
- Verwendung von inkrementellen Query-Algorithmen um die Antwortzeit zu verringern

Infinispan Ausblick



- Map/Reduce Support
- Versioned Entries
- Support Eventual Consistency (7.0 ?)
- Optimierungen:
 - Transaktionen
 - Locking
 - RPC



Vielen Dank für ihre Aufmerksamkeit !

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de



Fragen ?



Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de