




Infinispan - NoSQL für den Enterprise Java Alltag

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
- Ausblick

© 2013 Orientation in Objects GmbH

Infinispan - NoSQL für den Enterprise Java Alltag | 2

Gliederung

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




© 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

CAP Theorem und JTA

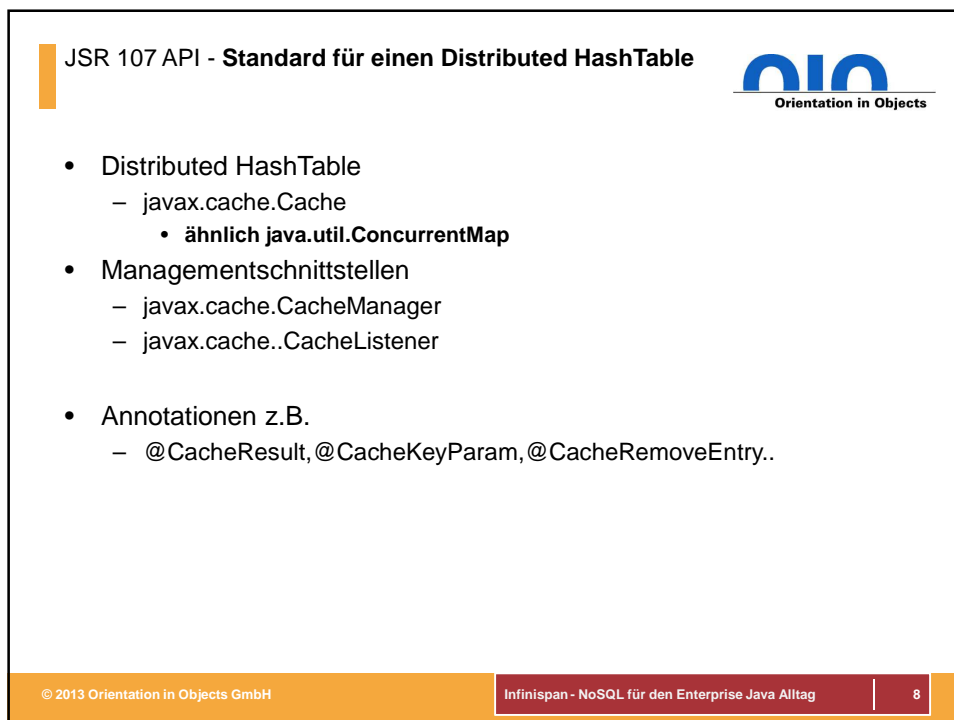
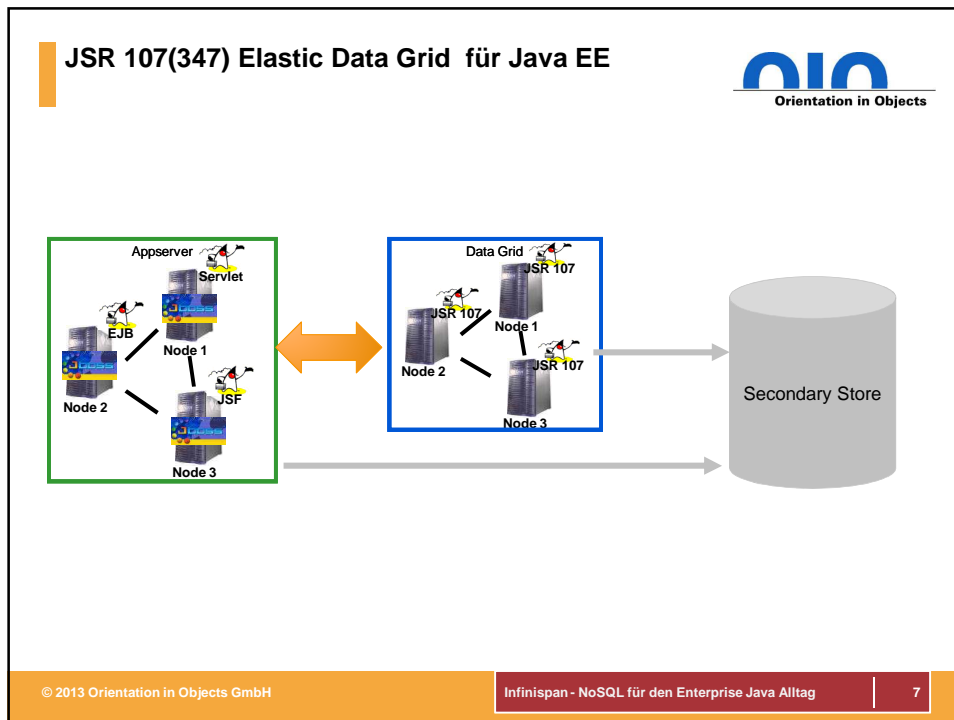
A Venn diagram illustrating the CAP theorem. It features three overlapping circles: an orange circle at the top labeled 'Consistency', a green circle on the left labeled 'Availability', and a blue circle on the right labeled 'Partition Tolerance'. The intersection of 'Consistency' and 'Availability' is labeled 'JTA'. The intersection of 'Consistency' and 'Partition Tolerance' is also labeled 'JTA'. The intersection of 'Availability' and 'Partition Tolerance' is labeled 'BASE'. The central intersection of all three circles is labeled 'n.a.'. The logo 'nio Orientation in Objects' is in the top right corner.

© 2013 Orientation in Objects GmbH | Infinispan - NoSQL für den Enterprise Java Alltag | 5

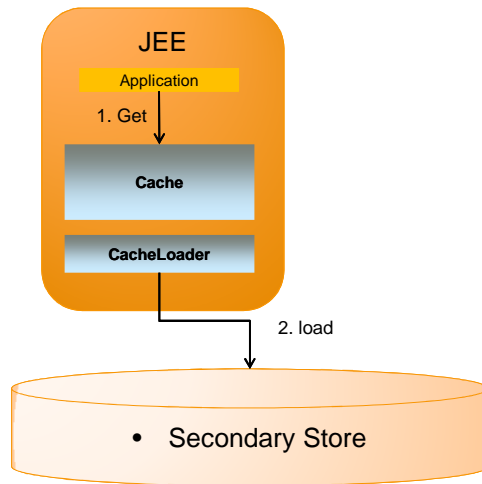
Data Grid vs. Distributed DBMS

A Venn diagram comparing Data Grid and Distributed DBMS against the CAP theorem. It features three overlapping circles: an orange circle at the top labeled 'Consistency', a green circle on the left labeled 'Availability', and a blue circle on the right labeled 'Partition Tolerance'. The intersection of 'Consistency' and 'Availability' is labeled 'Data Grid'. The intersection of 'Consistency' and 'Partition Tolerance' is labeled 'Distributed DBMS'. The intersection of 'Availability' and 'Partition Tolerance' is labeled 'BASE'. The central intersection of all three circles is labeled 'n.a.'. The logo 'nio Orientation in Objects' is in the top right corner.

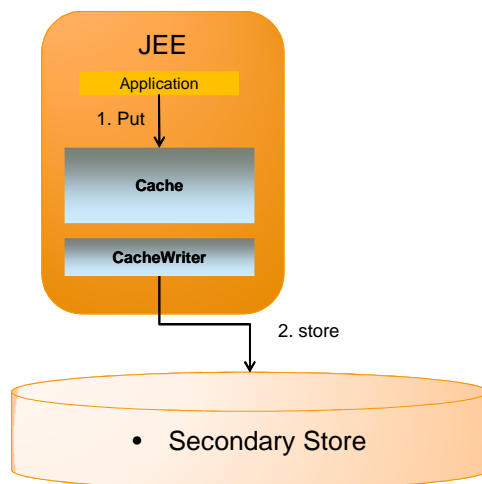
© 2013 Orientation in Objects GmbH | Infinispan - NoSQL für den Enterprise Java Alltag | 6



javax.cache.CacheLoader



javax.cache.CacheWriter



Expiration

The diagram shows a 'Cache' box containing items M, X, Y, and I. An arrow labeled 'get' points to item Y. An arrow labeled 'put' points to item D. Item F is shown in a trash bin, indicating expiration. A clock icon with a red arrow indicates the passage of time.

© 2013 Orientation in Objects GmbH

Infinispan - NoSQL für den Enterprise Java Alltag

11

Gliederung


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

© 2013 Orientation in Objects GmbH

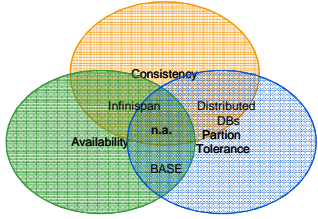
Infinispan - NoSQL für den Enterprise Java Alltag

12

Infinispan



- 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

13

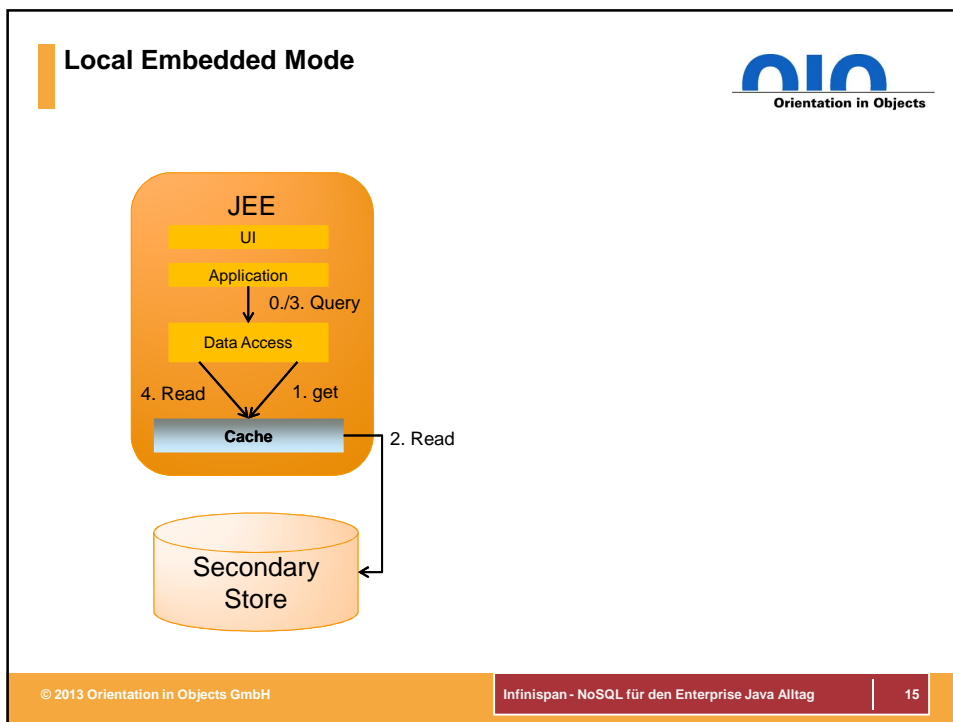
Demo




© 2013 Orientation in Objects GmbH

Infinispan - NoSQL für den Enterprise Java Alltag

14



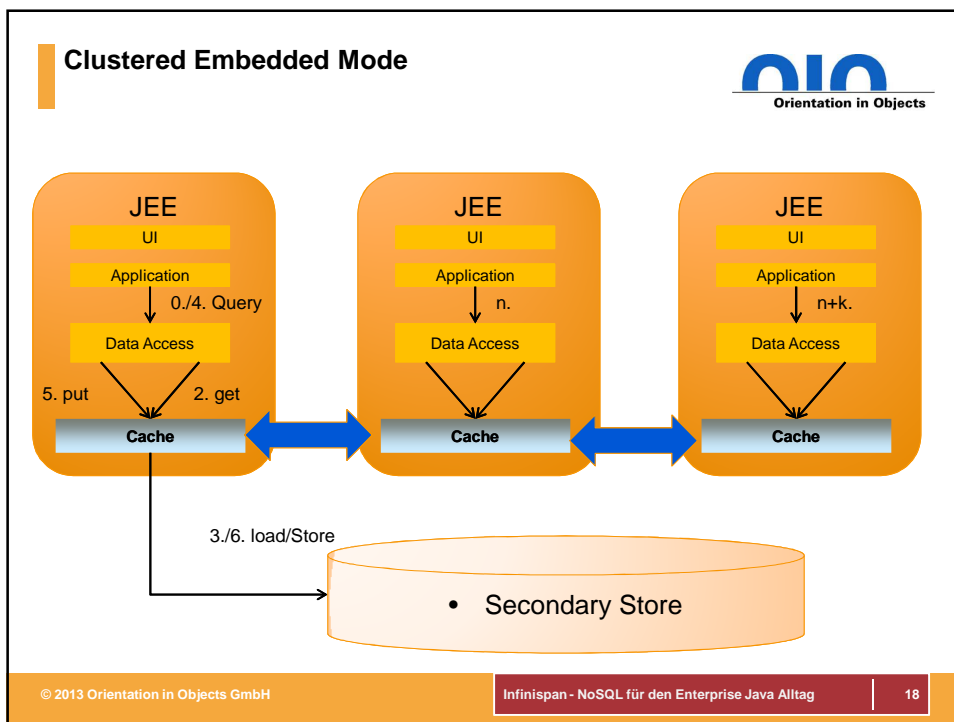
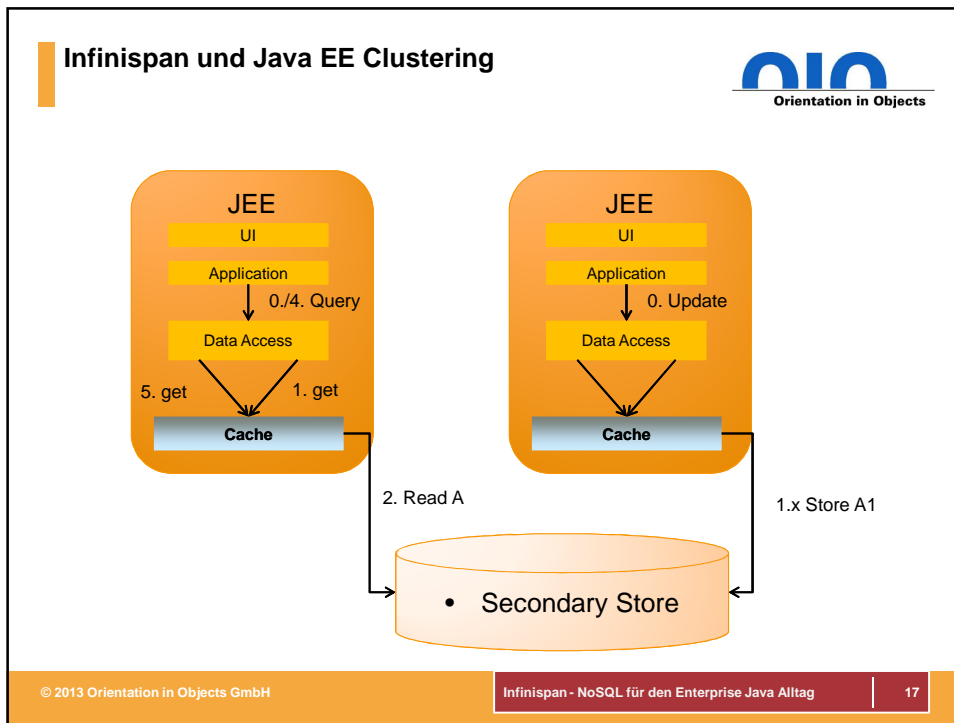
Local Embedded Mode II

- Code in der gleiche VM wie infinispan
- 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("cluster.xml");
Cache<String, String> cache = cacheManager.getCache();
```

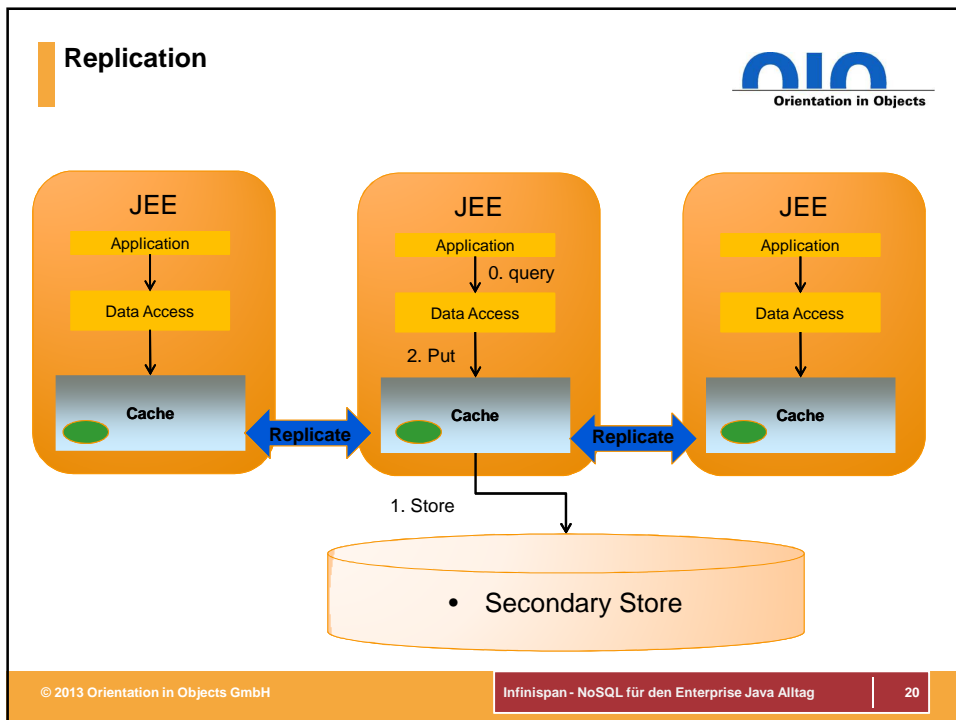
© 2013 Orientation in Objects GmbH | Infinispan - NoSQL für den Enterprise Java Alltag | 16

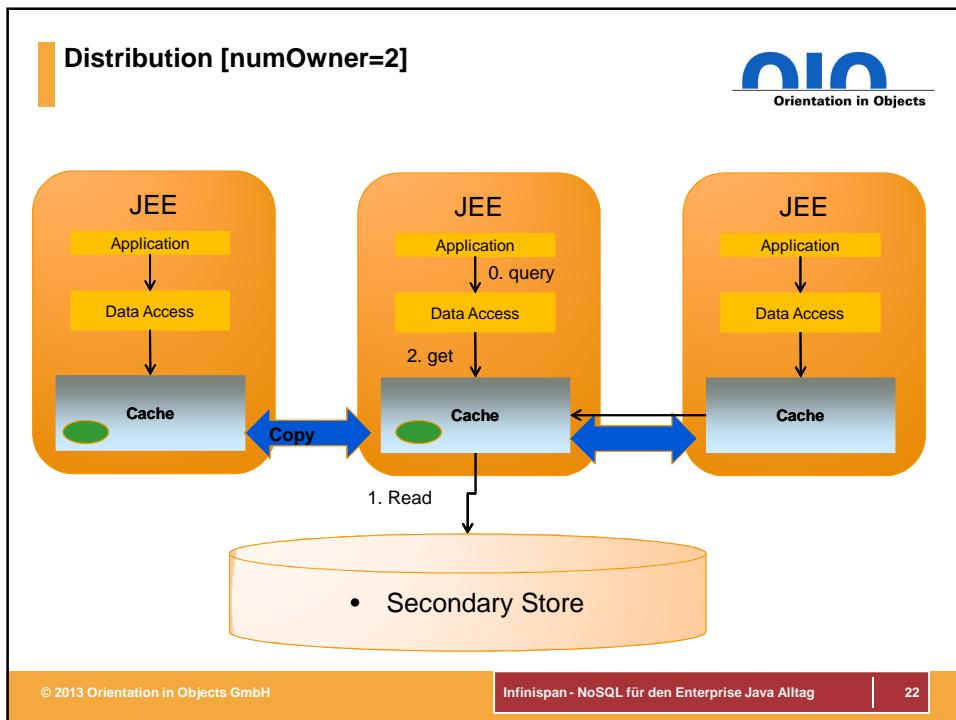
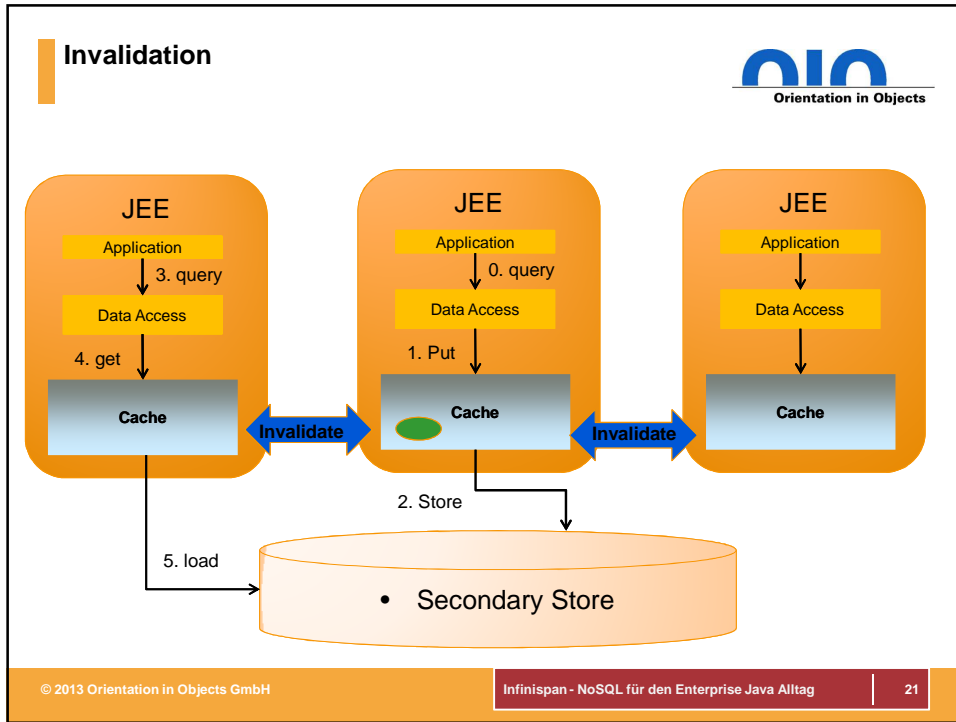


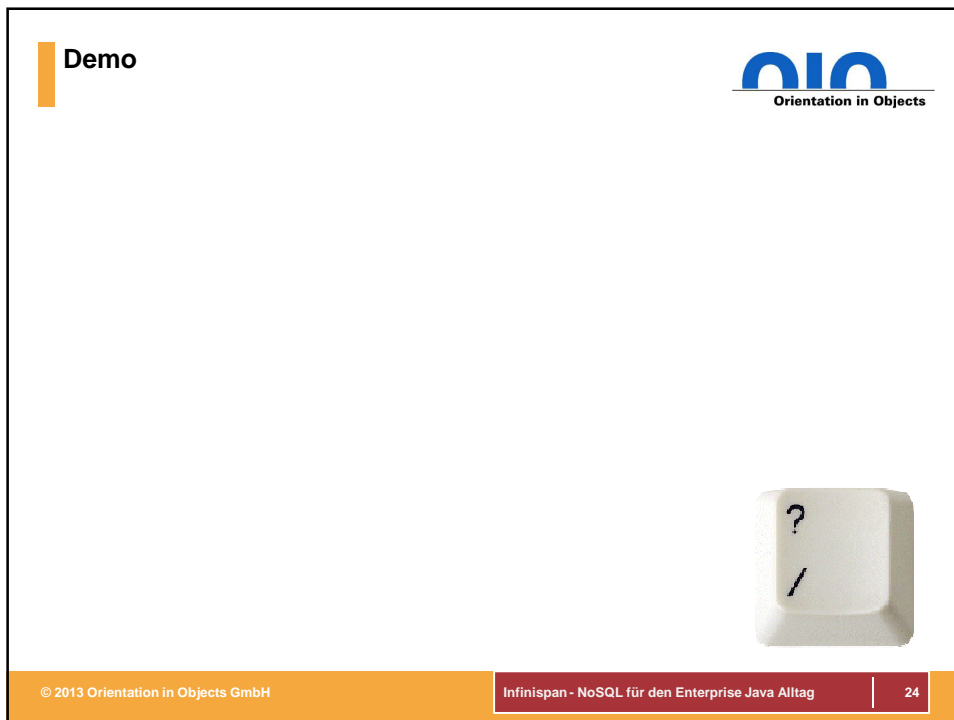
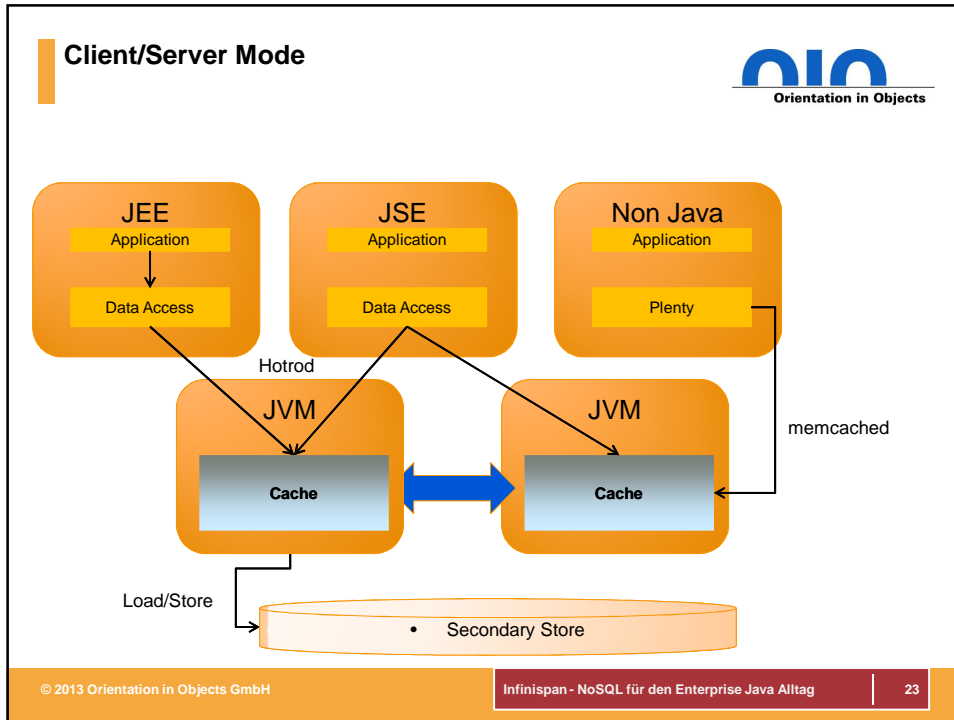
Demo



© 2013 Orientation in Objects GmbH | Infinispan - NoSQL für den Enterprise Java Alltag | 19







Client/Server Mode II



- Starten des Servers erfordert die Angabe des Protokolls

```
– bin/clustered.sh -Djboss.socket.binding.port-offset=xxx -
  Djboss.node.name=nodeX
```

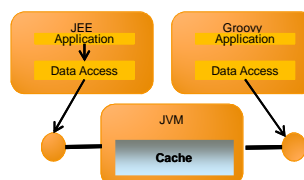
- org.infinispan.client.hotrod.RemoteCacheManager

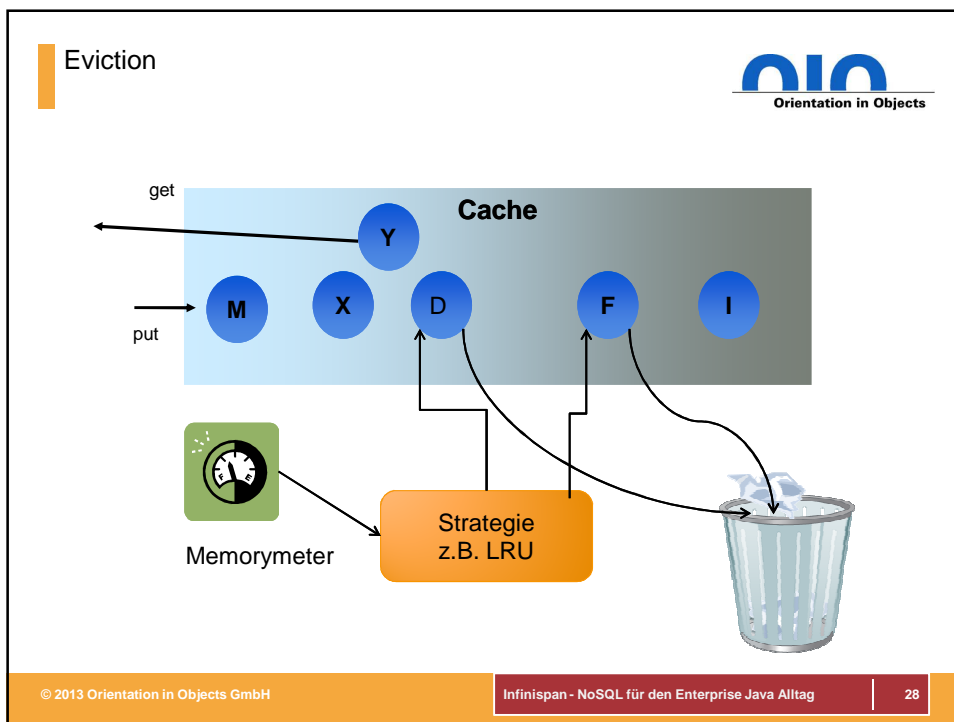
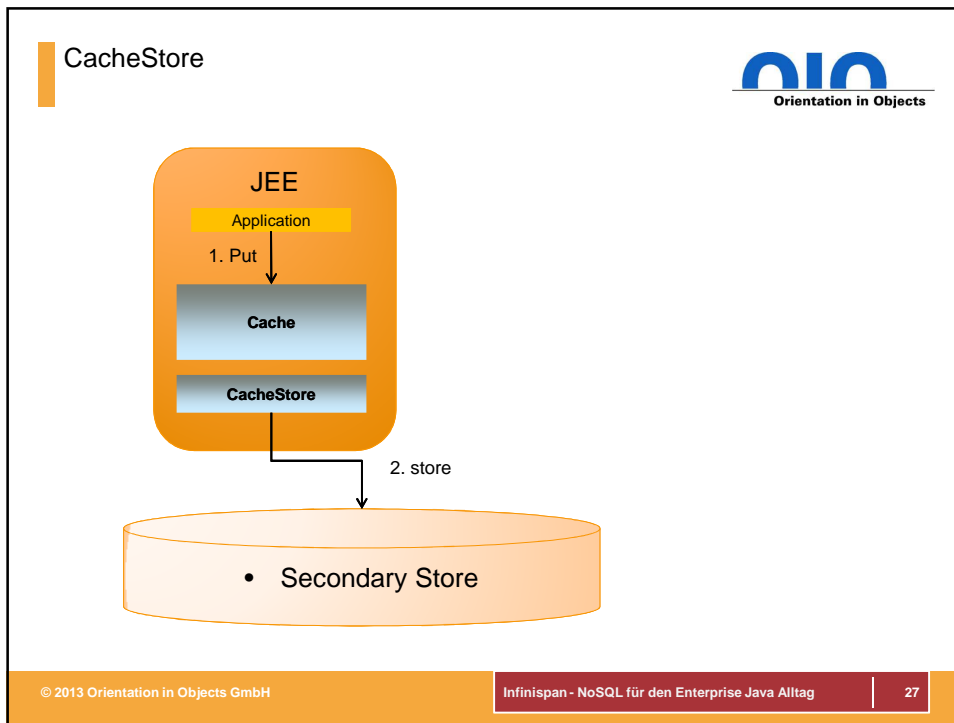
```
RemoteCacheManager rcm = new
  RemoteCacheManager("192.168.3.120");
RemoteCache remoteCace = rcm.getCache();
String key = "hello";
remoteCace.put(key, "world");
```

Client/Server Remoting



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





Storage JPA Cache Store



- Implementierung mit JPA 2
- Speicherung mit applikationsspezifischem Datenmodell
- Key muss Primärschlüssel 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());
```

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



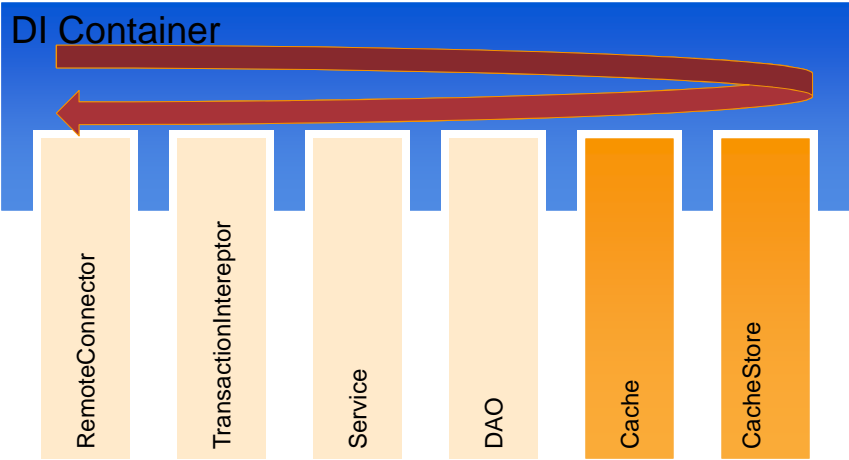

- NoSQL und Java EE
- Infinispan
- **Integrationszenarien**
- Ausblick

© 2013 Orientation in Objects GmbH

Infinispan - NoSQL für den Enterprise Java Alltag

31

DI-Container Integration



DI Container

RemoteConnector

TransactionIntereptor

Service

DAO

Cache

CacheStore

© 2013 Orientation in Objects GmbH

Infinispan - NoSQL für den Enterprise Java Alltag

32

Spring Integration



- Spring 3.1 Cache Abstraktion
- Infinispan Provider
- Integration mit Spring DI für CacheManager und Cache

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

- Aktivieren des Cache
 - <cache:annotation-driven />

Spring Integration



- Cache Annotation

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

```
@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
 - Cache Injizierung

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

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

- Partielle Unterstützung vom JSR 107 Annotationen

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

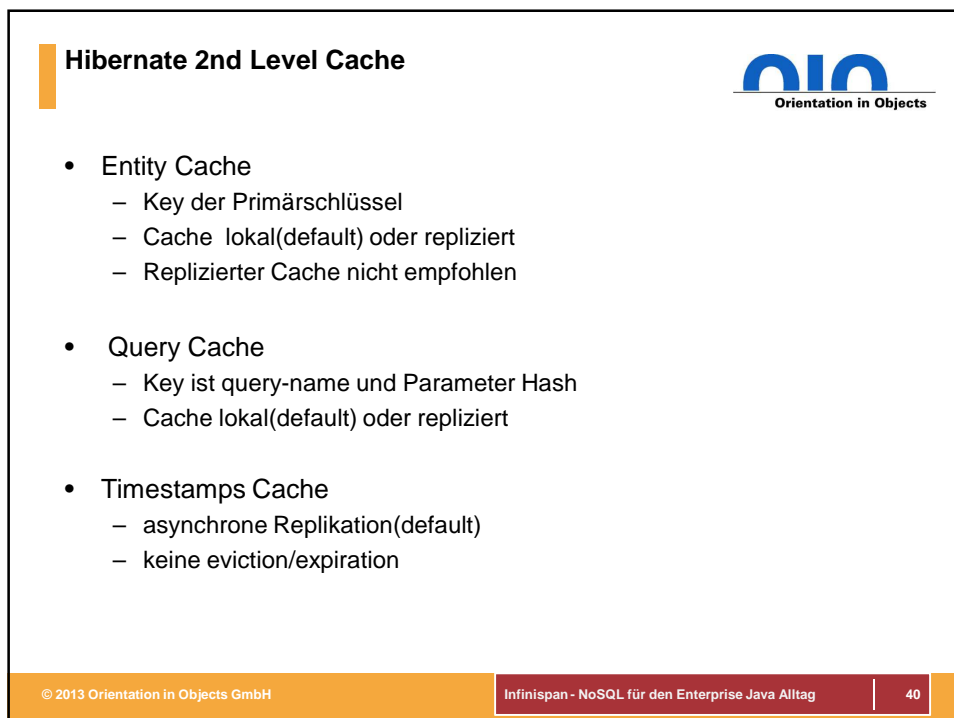
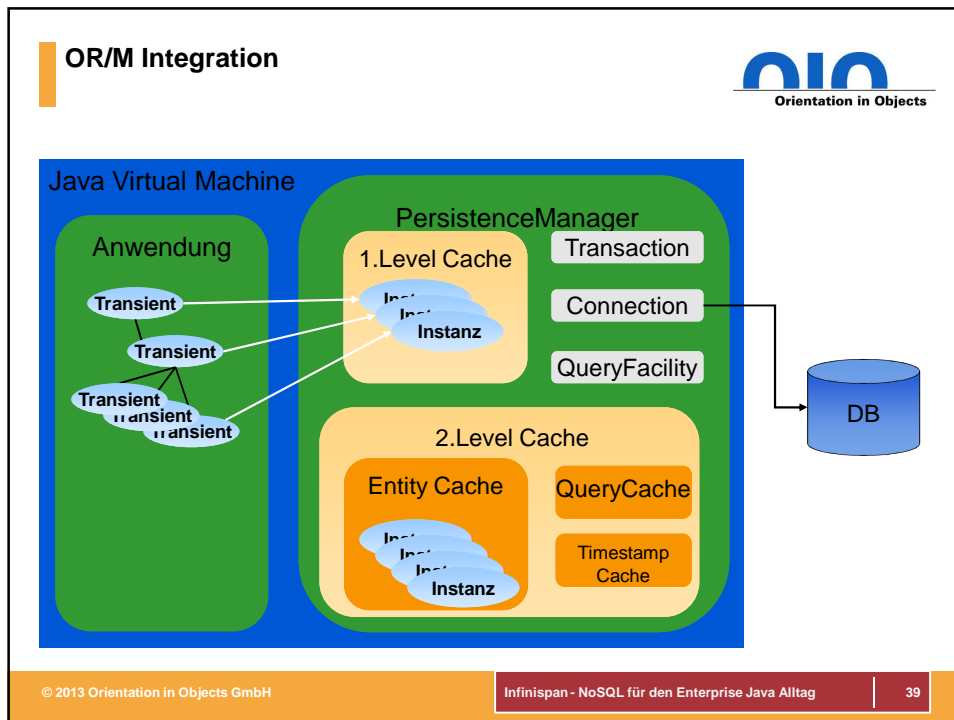


- Injizierung des eigenen Caches

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

Demo





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
- **Ausblick**

Infinispan Ausblick



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

© 2013 Orientation in Objects GmbH

Infinispan - NoSQL für den Enterprise Java Alltag

43



Fragen ?



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



**Vielen Dank für ihre
Aufmerksamkeit !**

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de