

The JBoss logo consists of a cluster of red dots of varying sizes arranged in a circular pattern to the left of the text. The text "JBoss" is in a bold, sans-serif font, with "JB" in red and "oss" in black. A registered trademark symbol (®) is located to the upper right of "JBoss". Below "JBoss" is the text "by Red Hat" in a smaller, red, sans-serif font.

Hochverfügbarkeit mit der **JBoss[®] Enterprise Application Platform**

Eigenschaften Java Enterprise Edition



EAP7 != EAP5 +2

Clustering

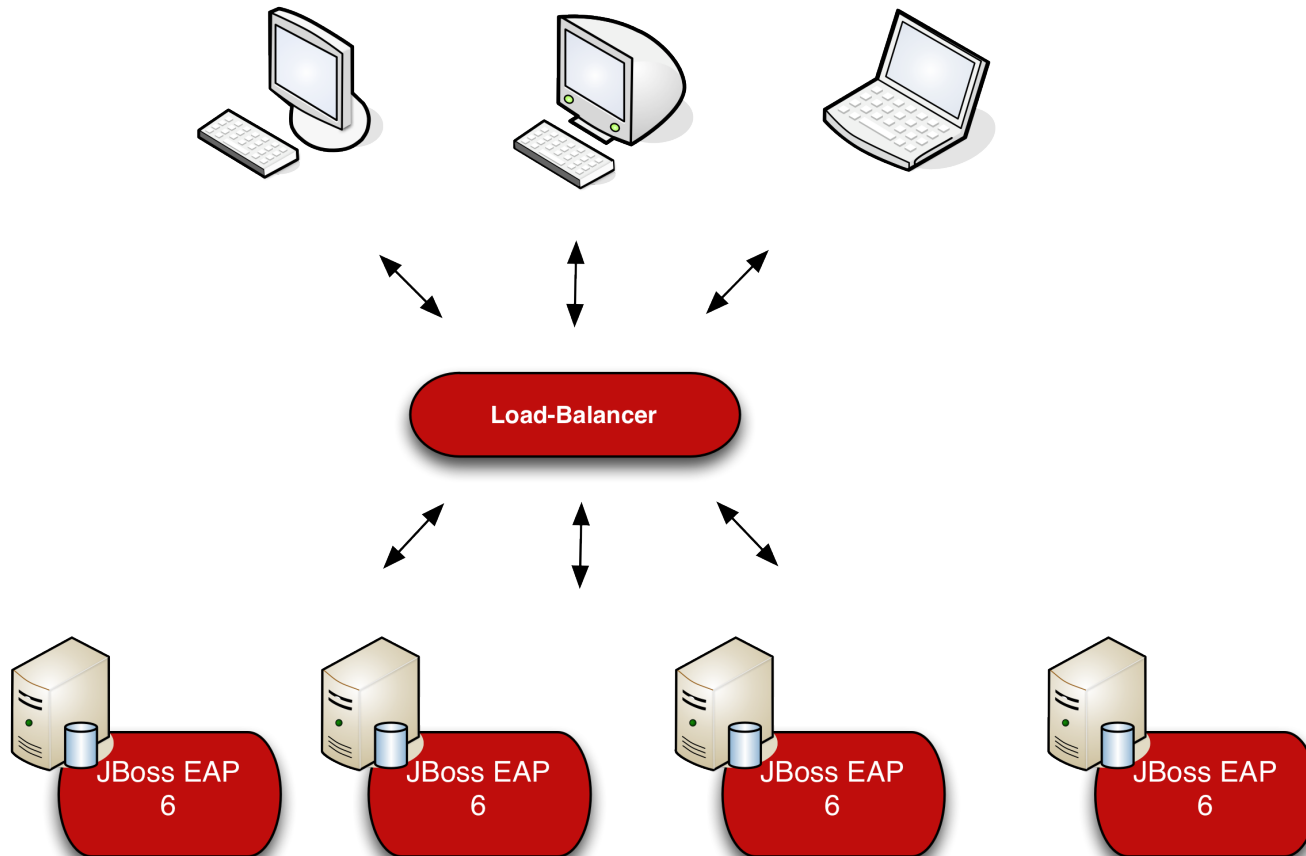
Gruppe von Servern, die gleiche Dienste erbringen.
Client sieht nur den Cluster nicht die Server.

Ziele:

- Fehlertoleranz durch Ausblenden fehlerhafter Server
(\Rightarrow Hochverfügbarkeit, HA)
- Lastausgleich durch Verteilung an Server Knoten
(\Rightarrow Skalierbarkeit)

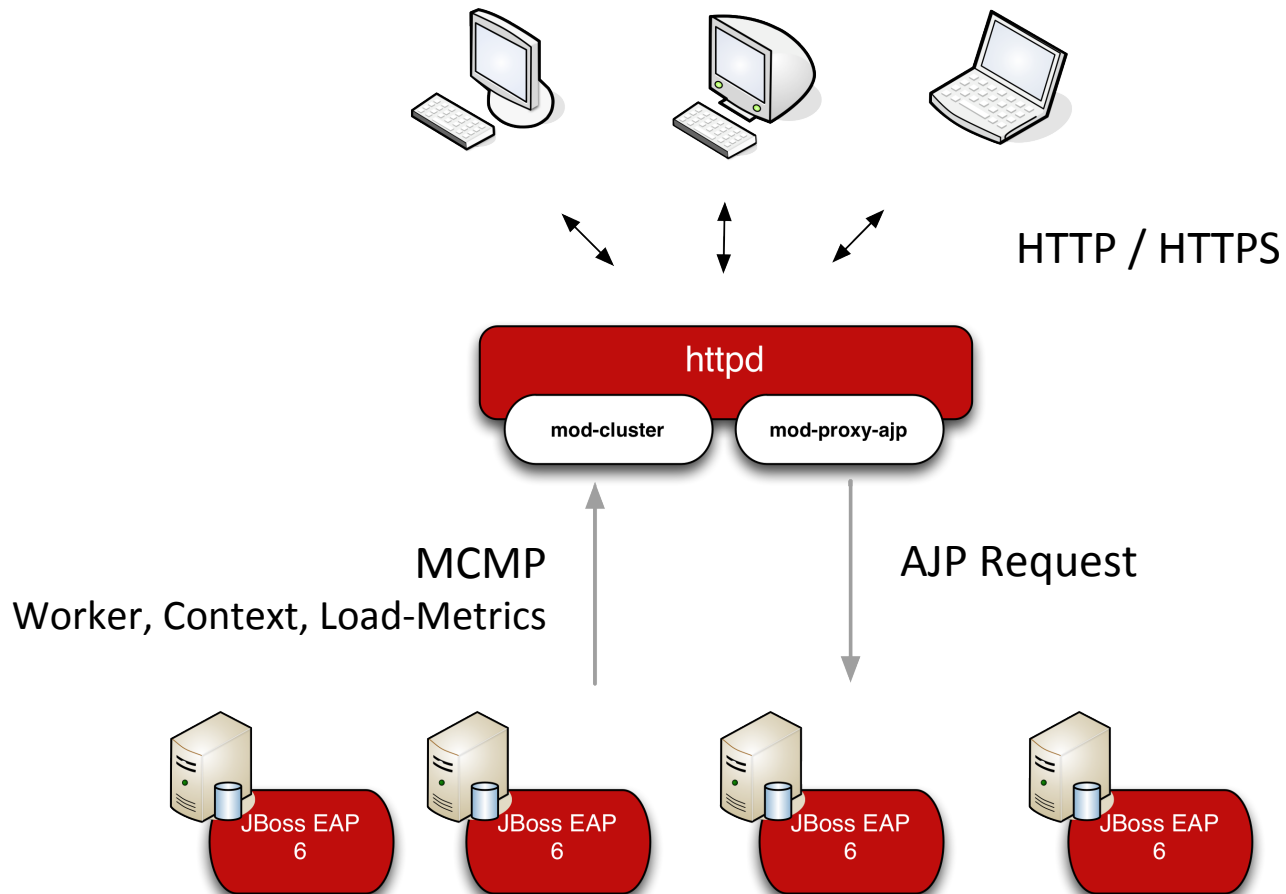


Lastverteilung ohne Session Replikation



Skalierung

mod_cluster



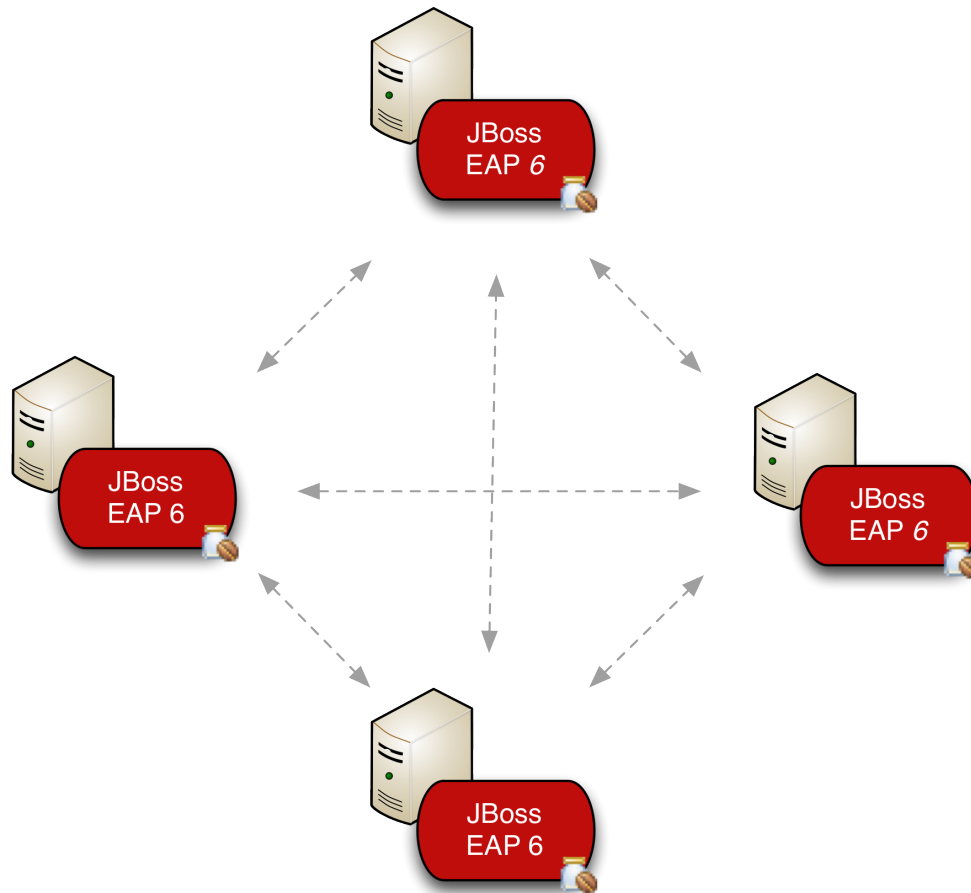
mod_cluster

- Dynamische Konfiguration der HTTPD worker
Advertise mittels Multicast
- Intelligente Lastverteilung
Einbezug von Lastverteilungsmetriken
(cpu, mem, heap, sessions, receive-traffic, send-traffic, requests, busyness)
- Load-Balancing Groups (aka. Domäne)

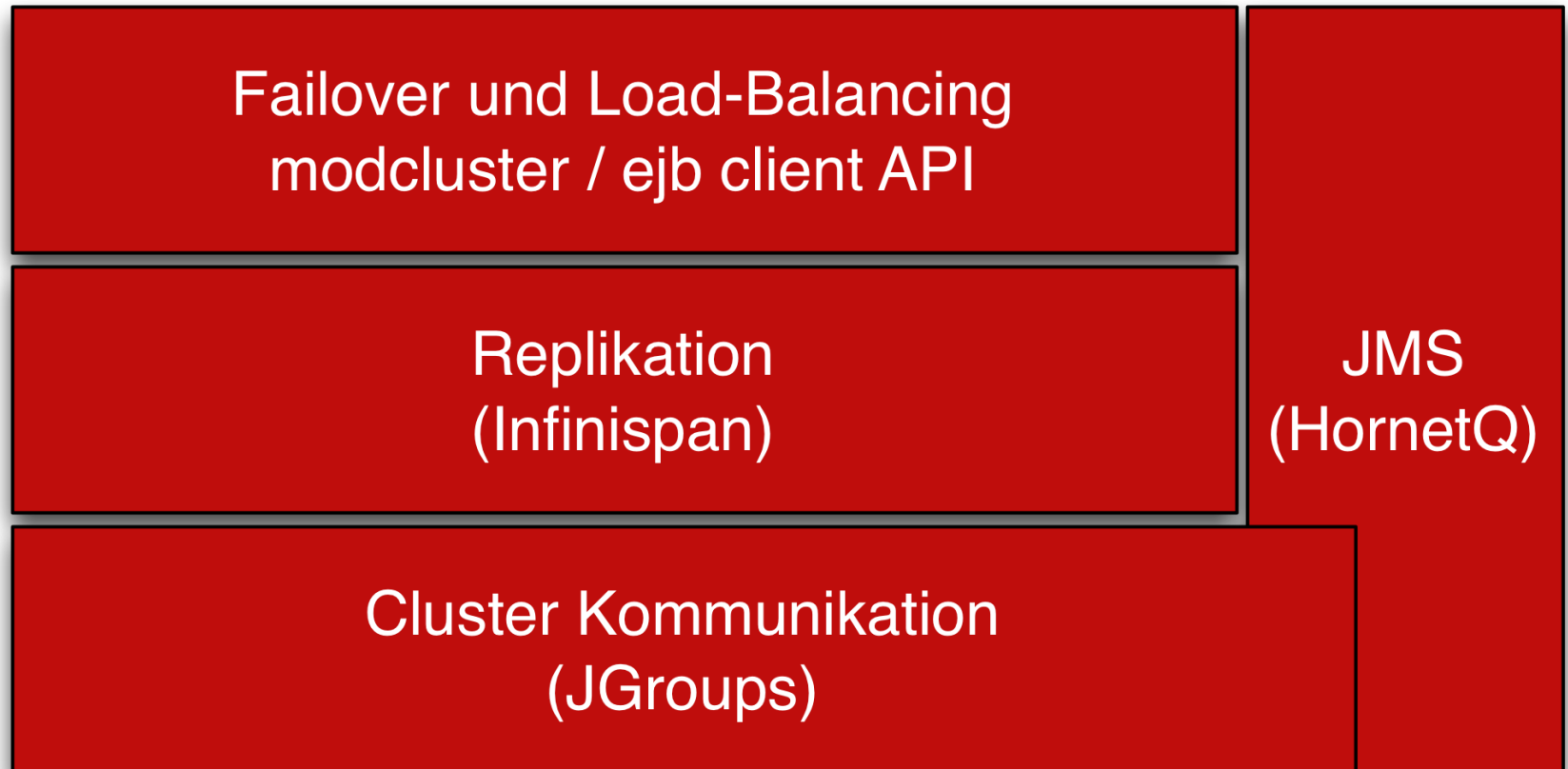
mod_cluster vs. mod_jk

- mod_jk benötigt statische Konfiguration
- Cluster Knoten hinzufügen oder entfernen (worker.properties)
- Anwendung hinzufügen oder entfernen (uriworkermap.properties)

Session Replikation



Subsysteme



Cluster-fähige Applikationen

Clustering einer EJB Session-Bean

```
@Stateless
@Remote(ClusteredStateless.class)
@org.jboss.ejb3.annotation.Clustered
public class ClusteredStatelessBean implements ClusteredStateless
{
    ...
}
```

Zustandslose Session Beans

→ dynamische Lastverteilung auf Clusterknoten

Zustandsbehaftete Session Beans

→ Session Affinity

→ Failover

Alternativ: Deployment-Descriptor (META-INF/jboss-ejb3.xml)

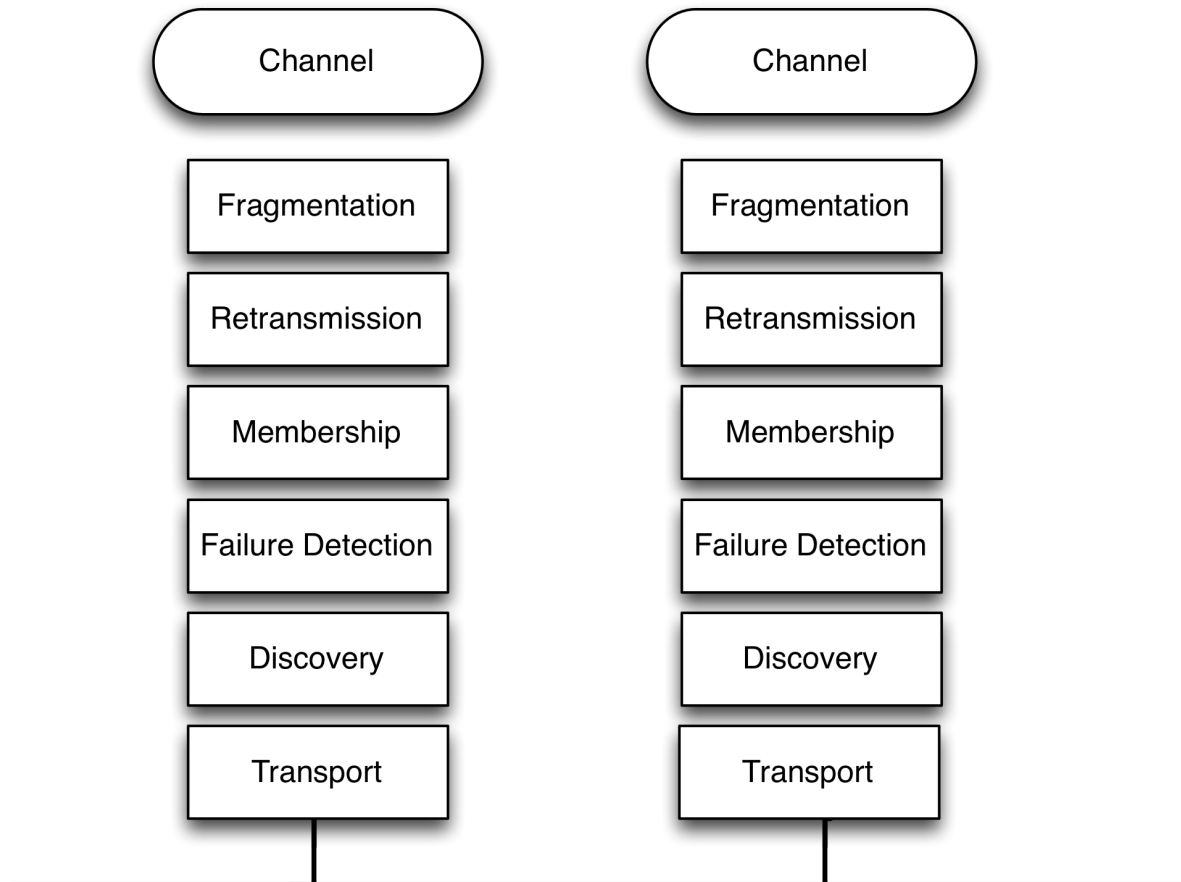
HTTP Session Replikation

Replikation der HTTP-Session stellt sicher, dass die Sessions der Klienten auf anderen Cluster-Knoten verfügbar sind

```
<web-app>  
    <distributable/>  
</web-app>
```

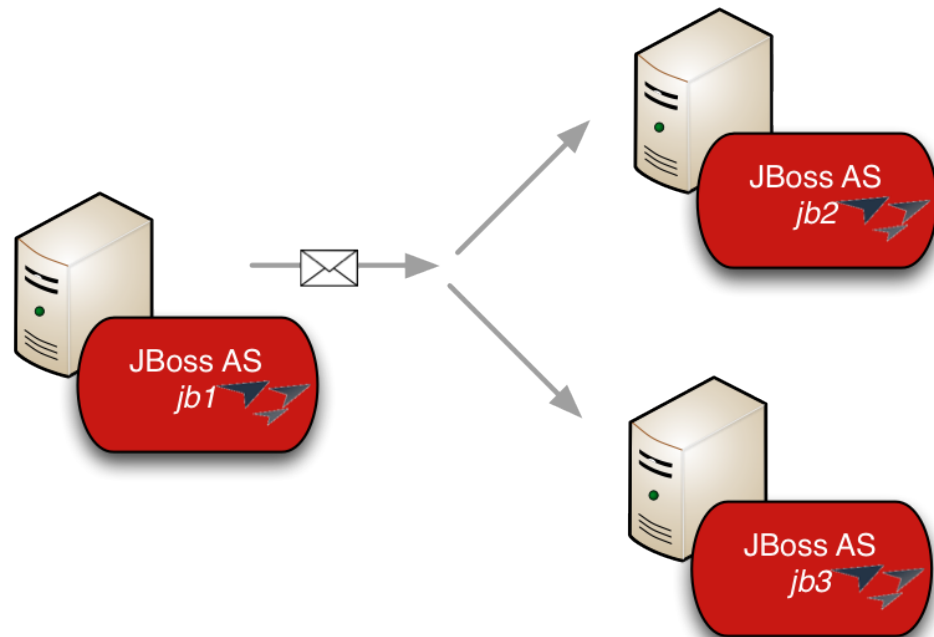
→ WEB-INF/web.xml

Cluster Kommunikation

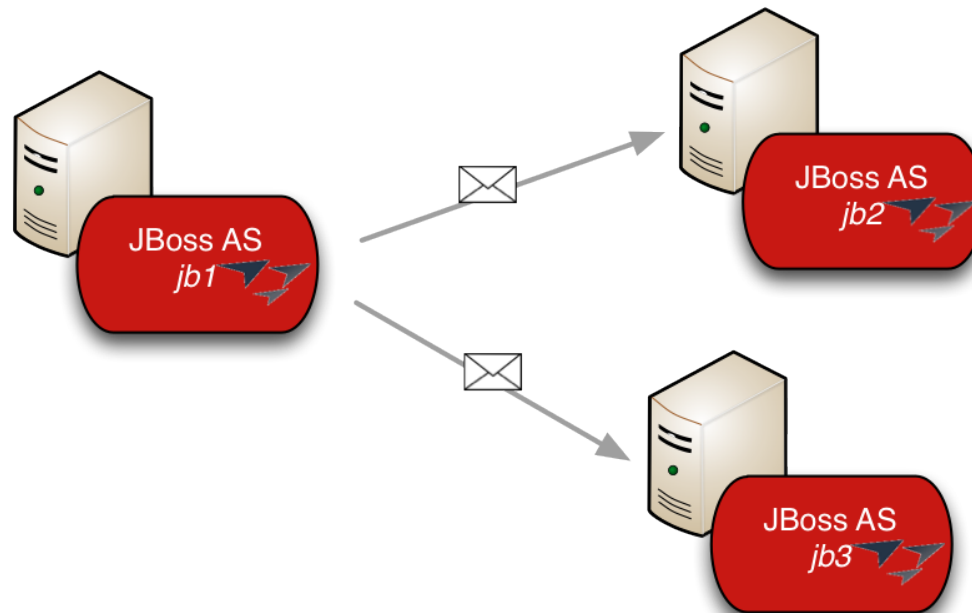


JGroups Transport

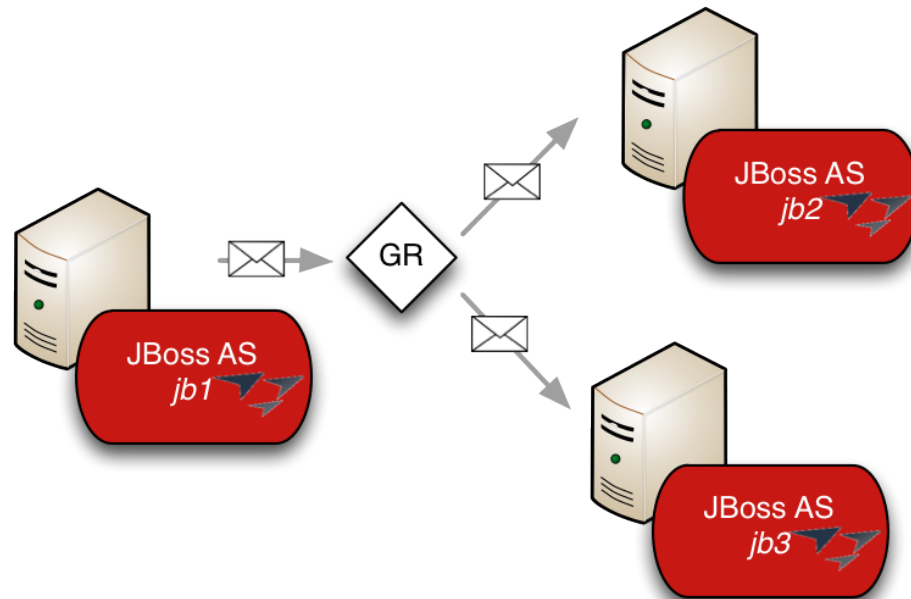
UDP



TCP



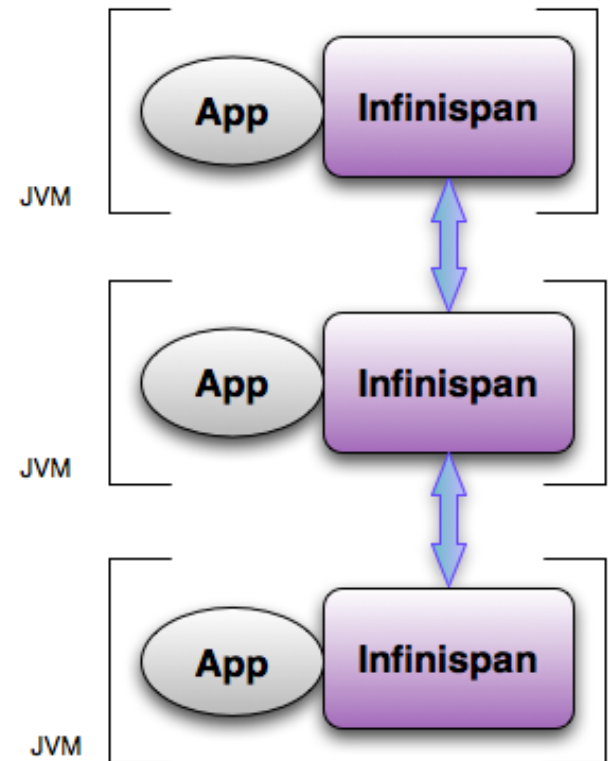
TUNNEL



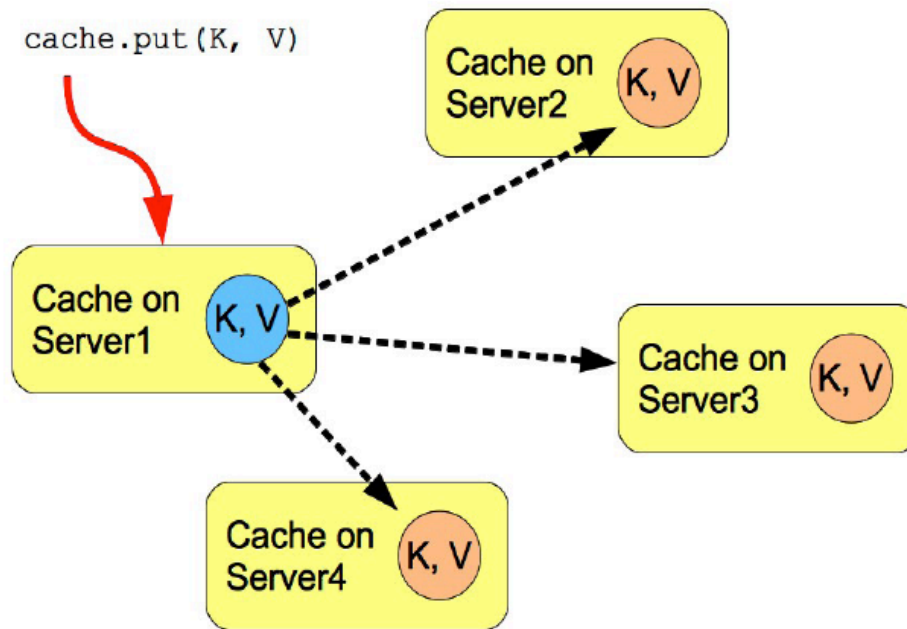
Replication

Um Hochverfügbarkeit zu unterstützen, müssen die Daten innerhalb des Clusters repliziert werden

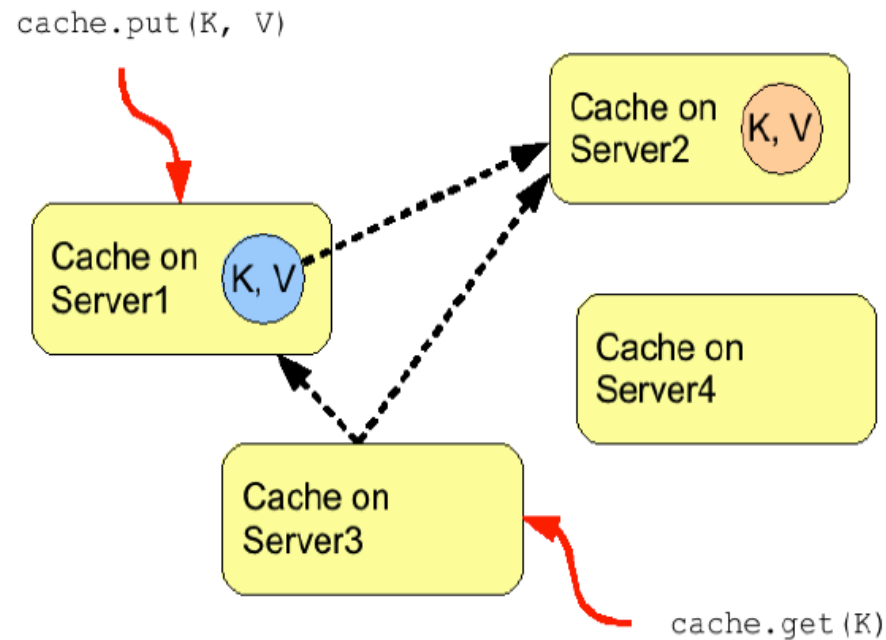
Infinispan wird intern für die Replikation verwendet



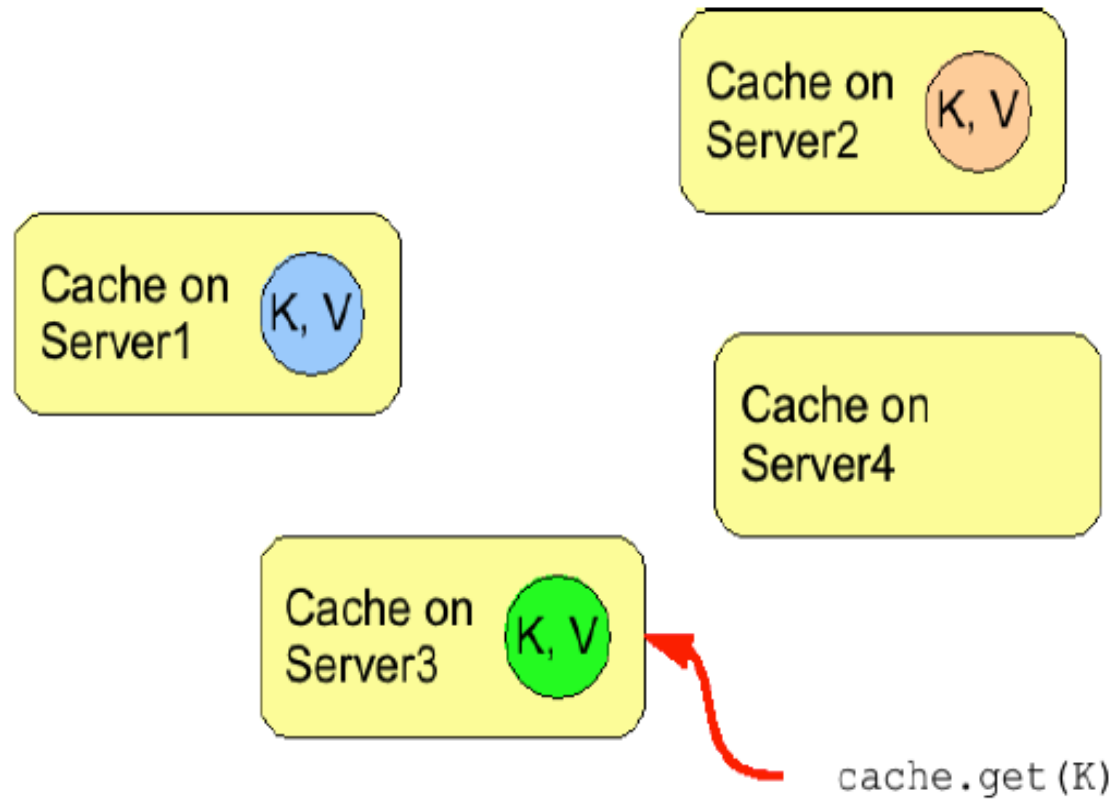
Replication - Cache mode



Distribution- Cache mode

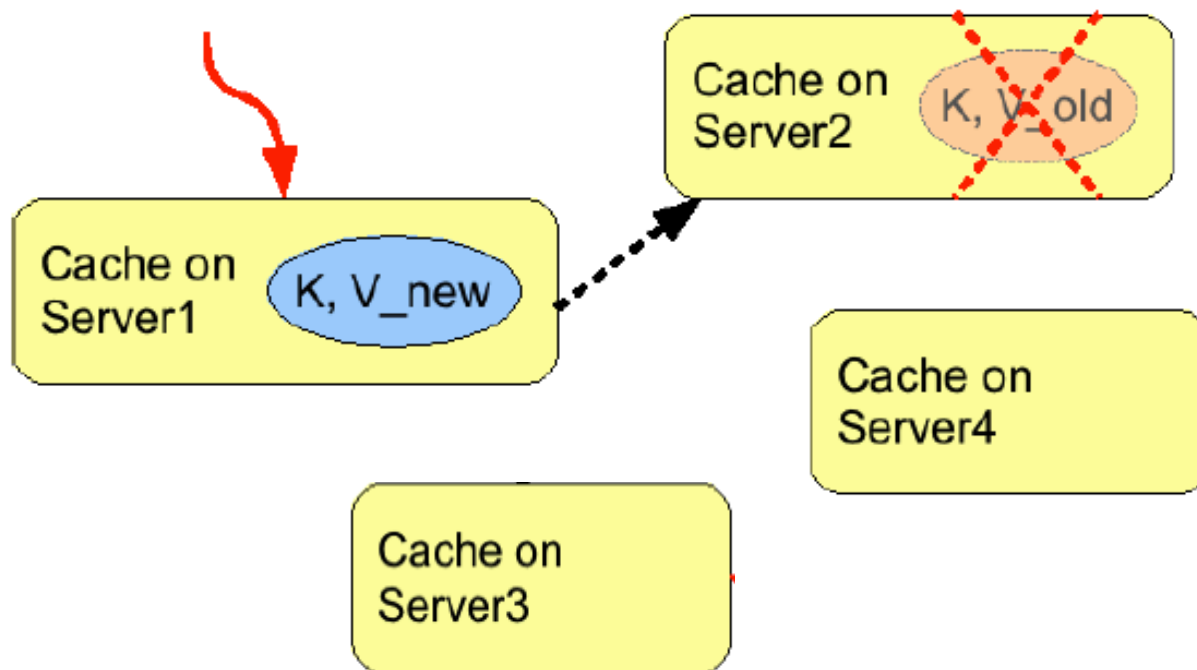


L1 - Cache mode

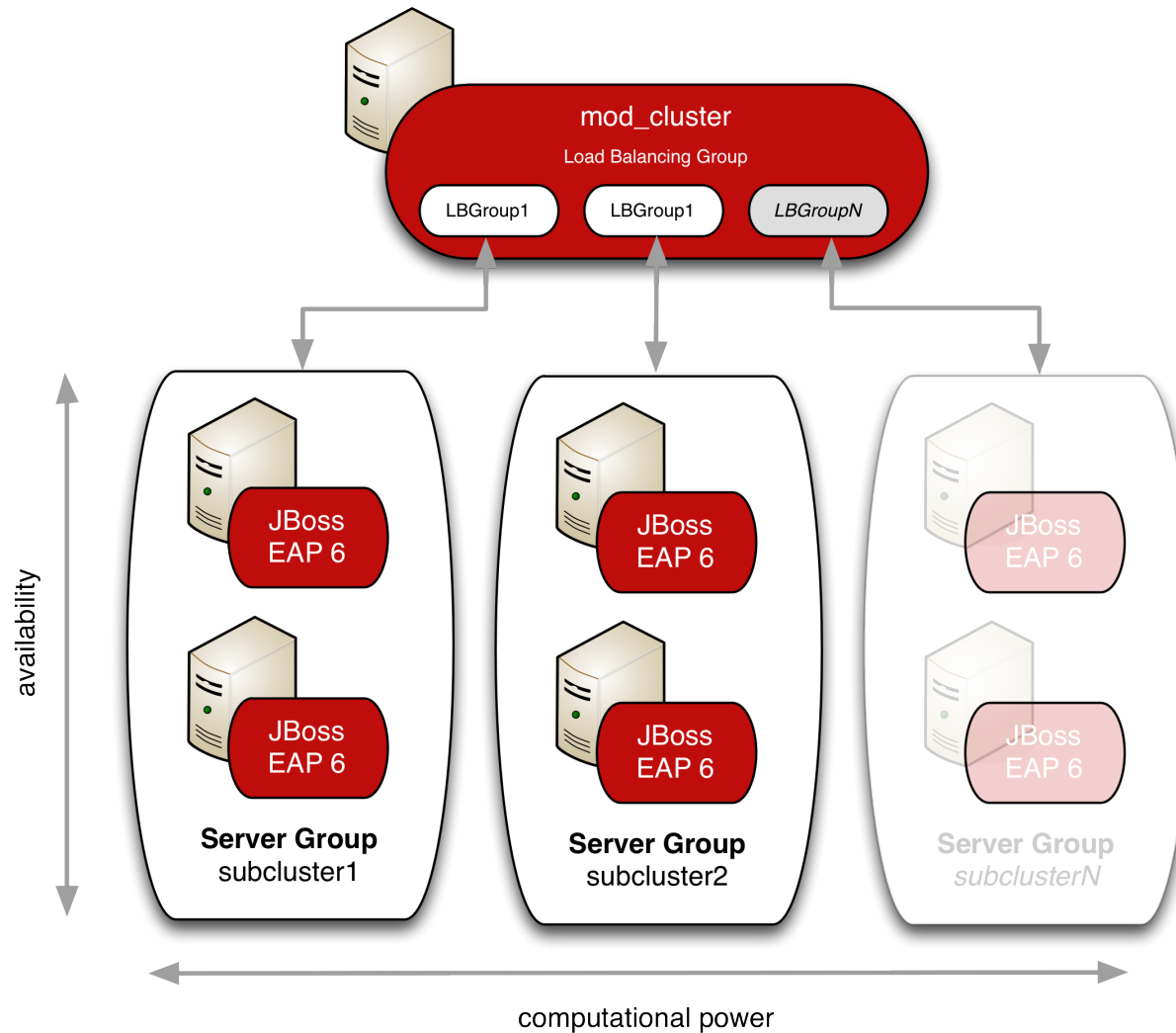


Invalidation - Cache mode

`cache.put(K, V_new)`



Cluster Partitionen



Optimierung

- JGroups Distribution enthält vorkonfigurierte Stacks für unterschiedliche Cluster Topologien
 - UDP vs. TCP
- Infinispan
 - ASYNC vs. SYNC
 - HTTP Session Replication
 - Replication Trigger
 - Replication Granularity

Messaging (HornetQ)

1. Zuverlässigkeit

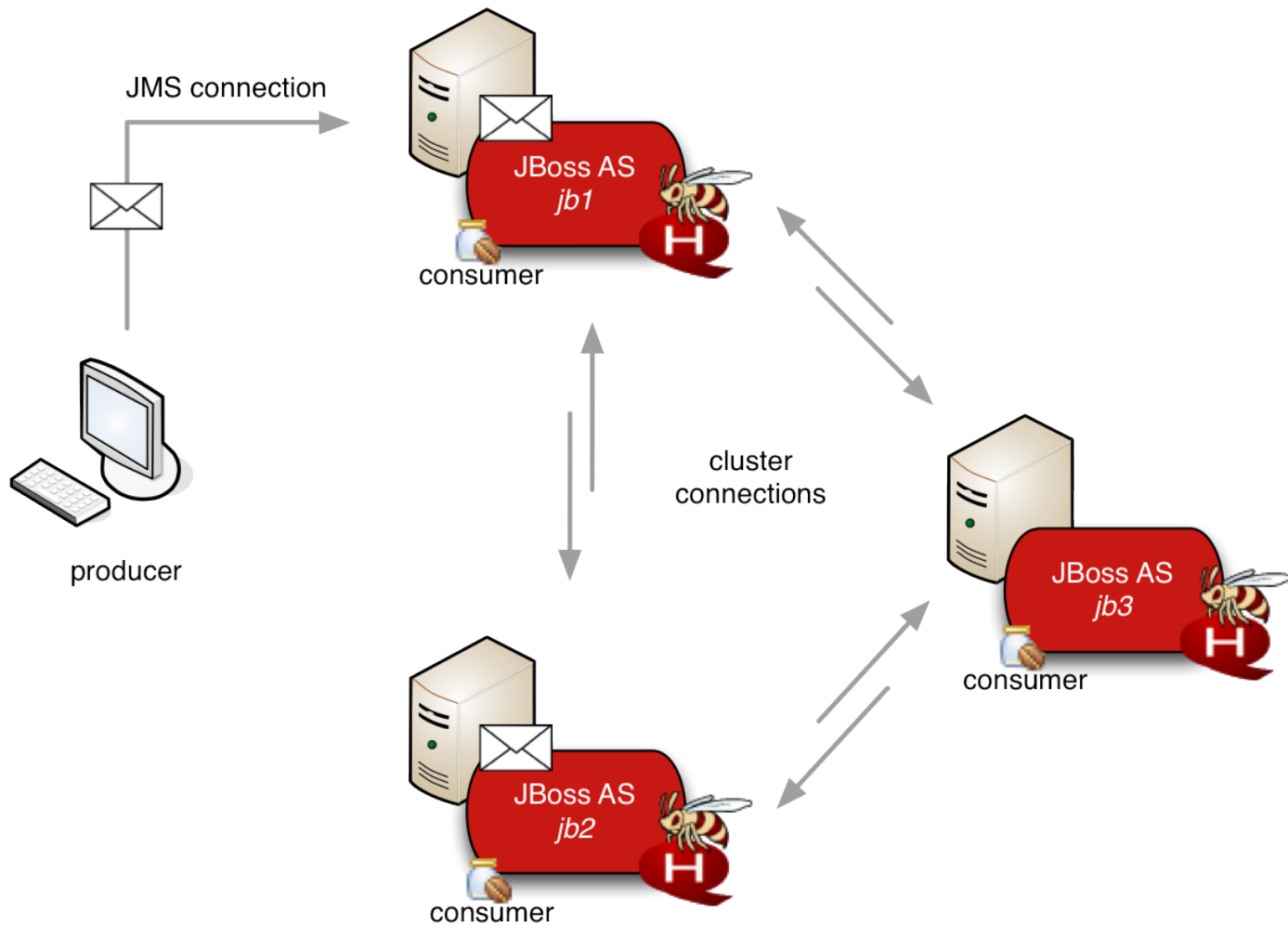
bereits empfangene Nachrichten werden garantiert zugestellt (gehen nicht verloren)

2. höherer Nachrichtendurchsatz

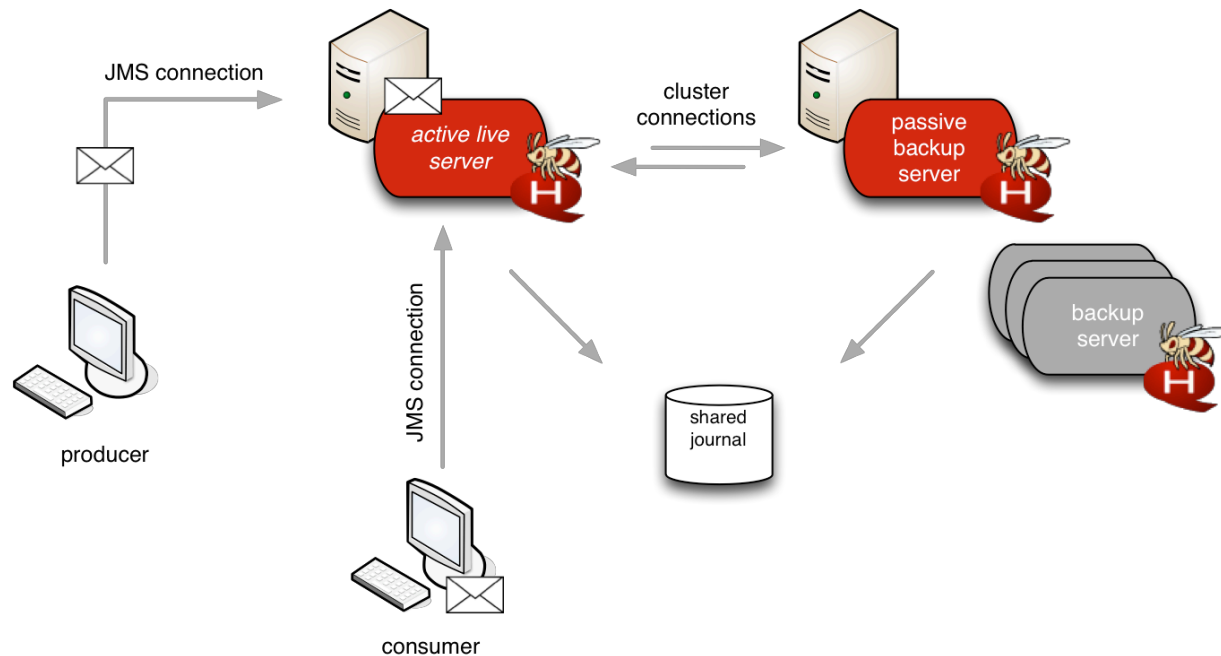
3. Lastverteilung

4. automatisches Failover bestehender Verbindungen

HornetQ - Lastverteilung



HornetQ - Failover



- Ab JBoss EAP 6.1 (HornetQ 2.3) - In-Memory Message Replication



7.2.0.Final



QA

6.1.0

alpha

6.1.0

beta

6.1.0

Final

7.x

Free

Zero-Dollar-Subscription
for Development

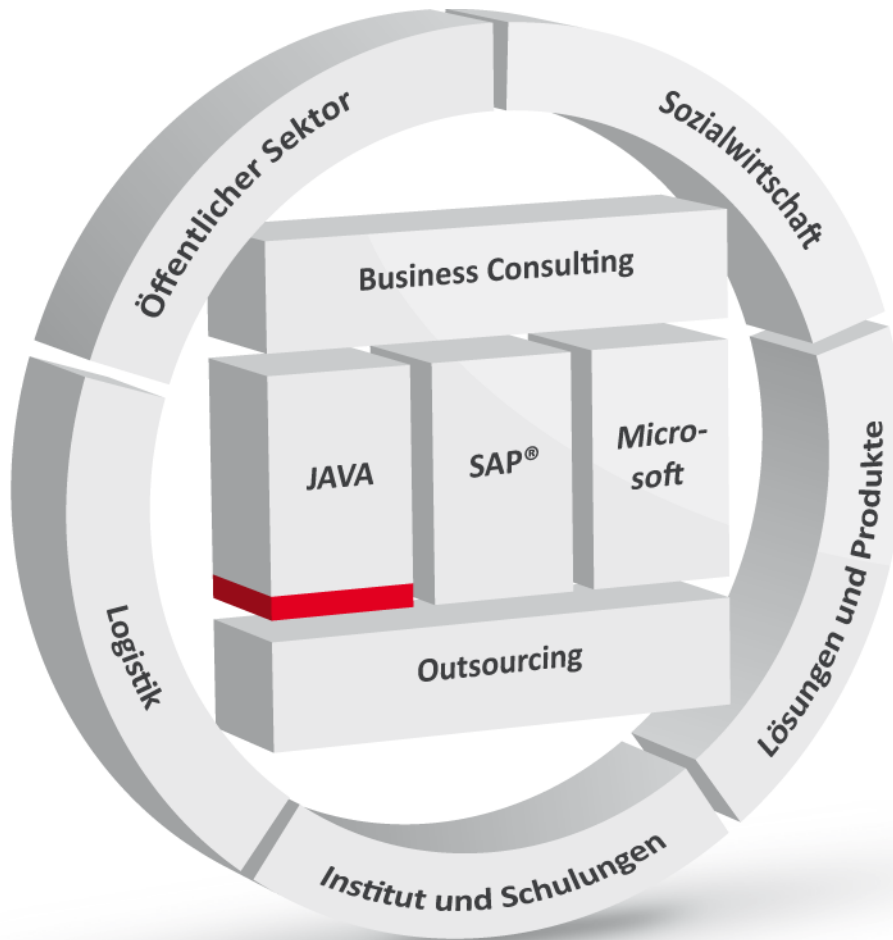
Payed-Subscription with
Support for
Production



8.0.0

alpha1

akquinet AG – Standort Berlin



JAVA

Beratung und
Coaching

Individuelle
Lösungen

Betriebsführung

Schulungen

Produktberatung



**PREMIER
BUSINESS
PARTNER**