

JUDCon

JBoss Users & Developers Conference

London:2011

Extreme Performance and Scalability with Near Caches

Galder Zamarréño
Senior Software Engineer
Red Hat, Inc

31st October 2011

Galder Zamarréño

- R&D Engineer, Red Hat Inc.
- Infinispan developer
- 5+ years exp. with distributed data systems
- Twitter: @galderz
- Blog: zamarreno.com

Agenda

- Introduction to Infinispan
- Near caches
- Clustered near caches
- Near caches with JMS
- Demo

Introducing Infinispan

What is Infinispan?

An in-memory, highly available, elastic, and open source (LGPL) data grid platform

Infinispan can be used as...

In-memory cache

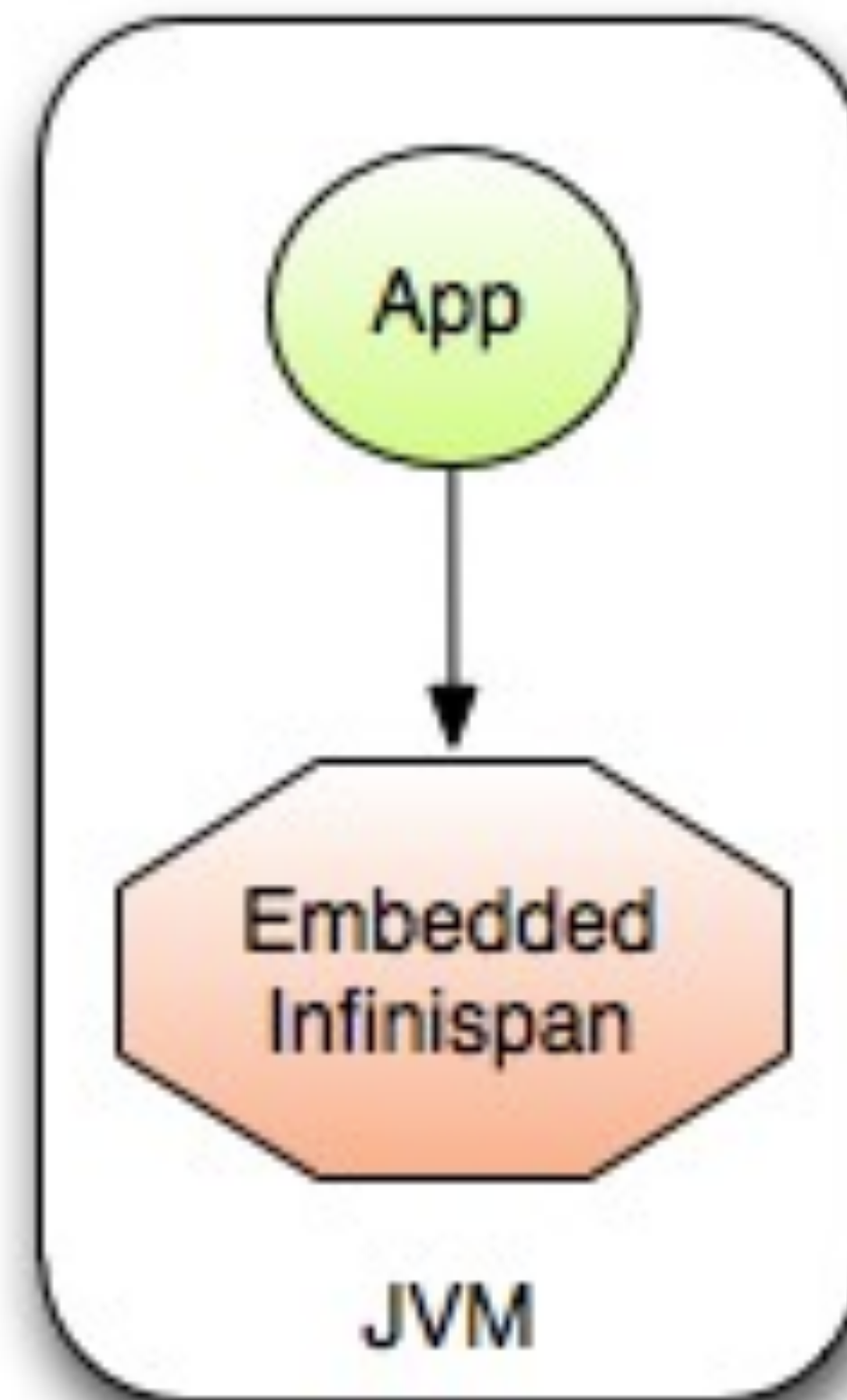
Boost performance caching
data that's frequently
accessed, or hard to
calculate...etc

Perfectly suited for it!

Highly concurrent thanks to MVCC, has built-in **eviction**, pluggable **persistence...etc**

Embedded access

- Client application and Infinispan live in same JVM
- **Zero Latency Access!**



Infinispan meets CDI...

```
...
import javax.ejb.Stateless;
import javax.inject.Inject;
import org.infinispan.Cache;

@Stateless
class FooEJB {
    @Inject @MyCacheQualifier
    Cache<String, String> myCache;
}
```

```
...
import javax.enterprise.inject.Produces;
import org.infinispan.cdi.ConfigureCache;
import org.infinispan.config.Configuration;

class Config {
    @ConfigureCache("my-cache-name")
    @MyCacheQualifier
    @Produces
    Configuration myCacheConfiguration() {
        return new Configuration().fluent()
            .eviction()
            .strategy(FIFO)
            .maxEntries(2048)
            .build();
    }
}
```


... and JSR-107

```
...
import javax.cache.interceptor.CacheResult;
import javax.cache.interceptor.CachePut;
import javax.cache.interceptor.CacheValue;
import javax.cache.interceptor.CacheRemoveEntry;
import javax.cache.interceptor.CacheRemoveAll;

class MyDAO {
    @CacheResult(cacheName = "user-cache")
    User getUser(long id) {...};

    @CachePut(cacheName = "user-cache")
    void storeUser(long id, @CacheValue User user) {...};

    @CacheRemoveEntry(cacheName = "user-cache")
    void removeUser(long id) {...};

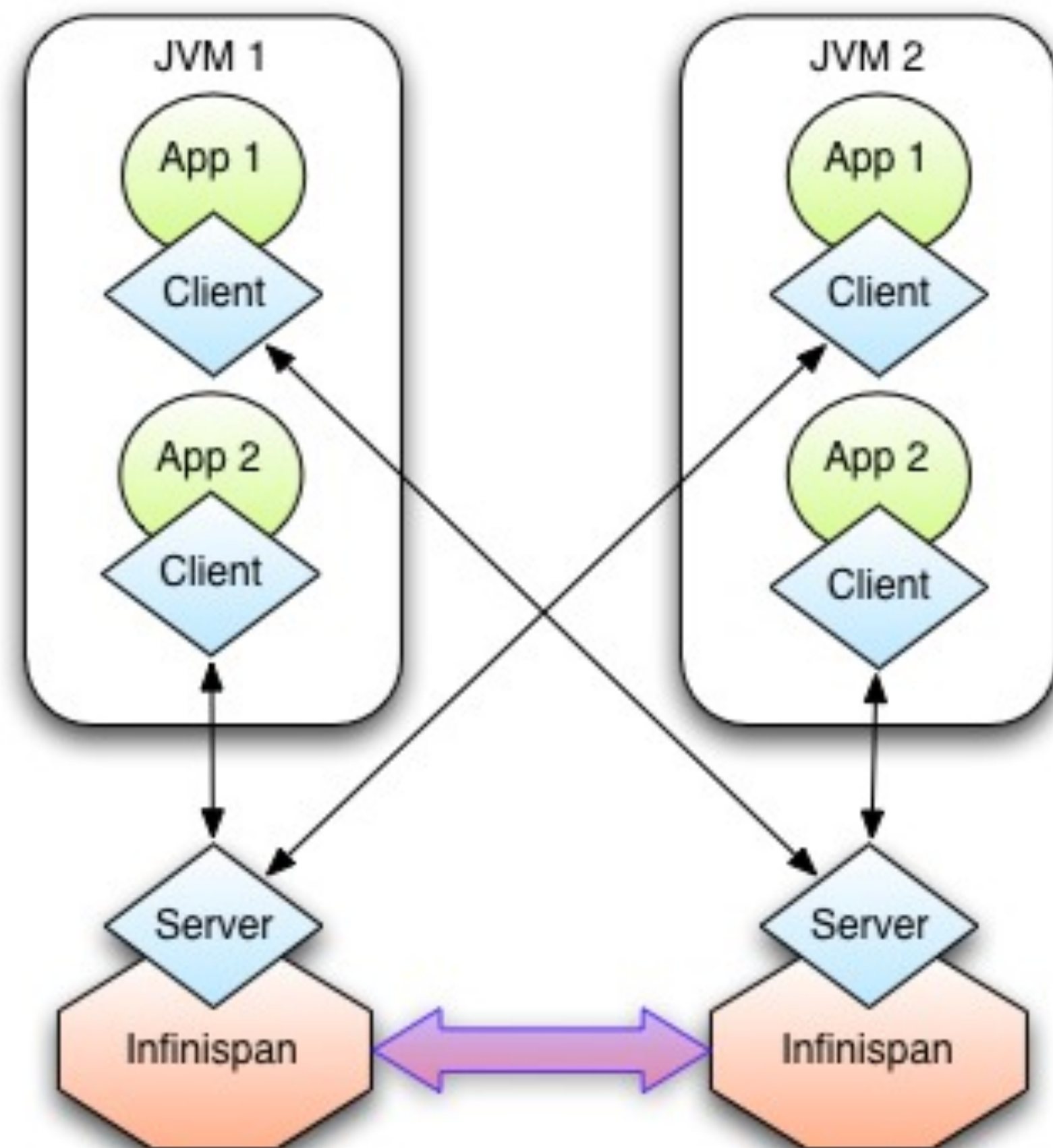
    @CacheRemoveAll(cacheName = "user-cache")
    void removeAllUser() {...};
}
```


Infinispan is not just a cache!

In-memory data grid

It's a **Fast, Available,
Distributed, Elastic** data store,
not just a cache!

Remote access



- Via protocols
 - REST
 - Hot Rod

Why separate data tier?

Manage/tune data tier independently, helps build stateless application tiers, and is easily scalable!

Data tier latency

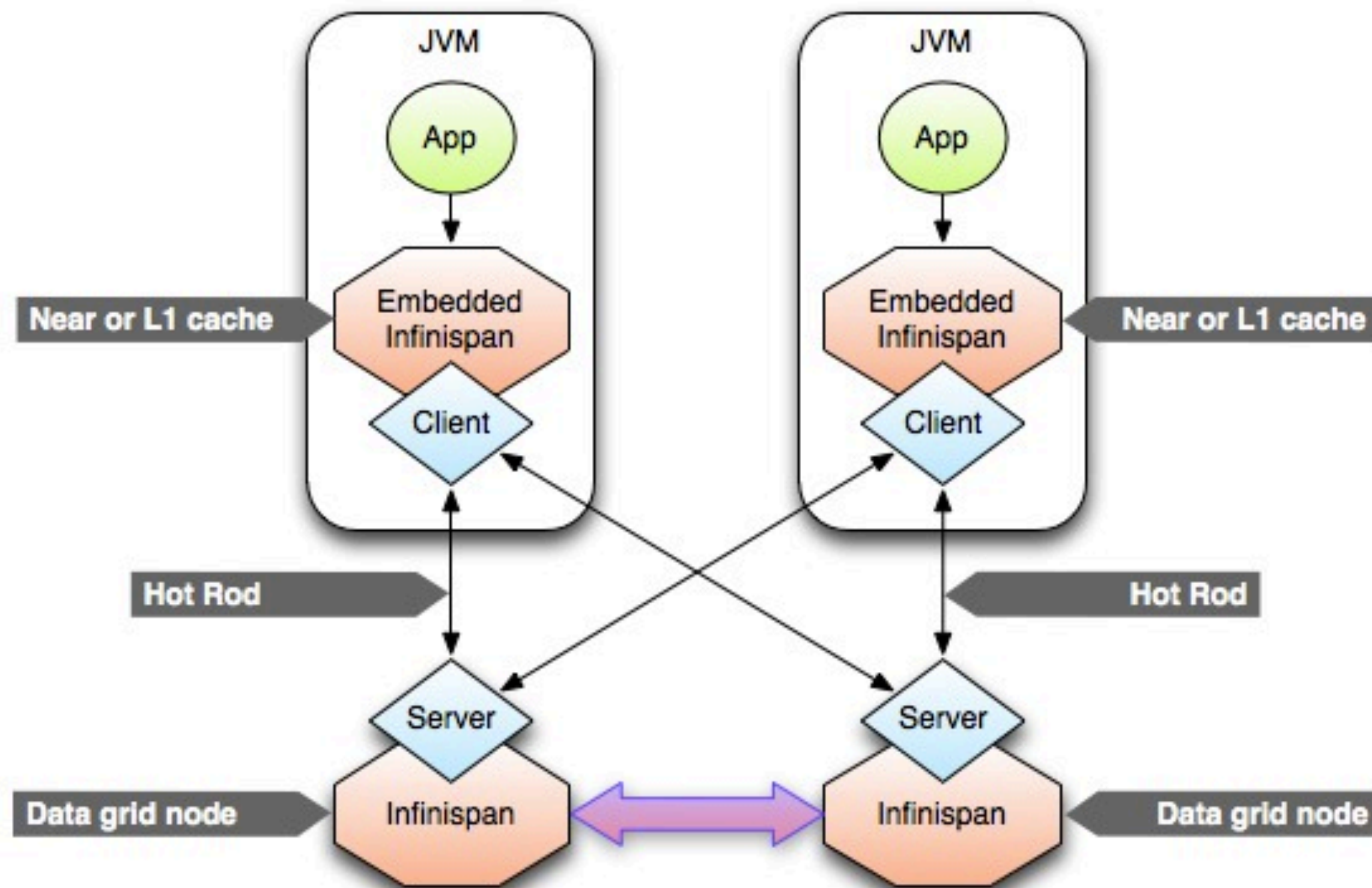
Network increases data grid
access latency

Have your Cake
and eat it Too!

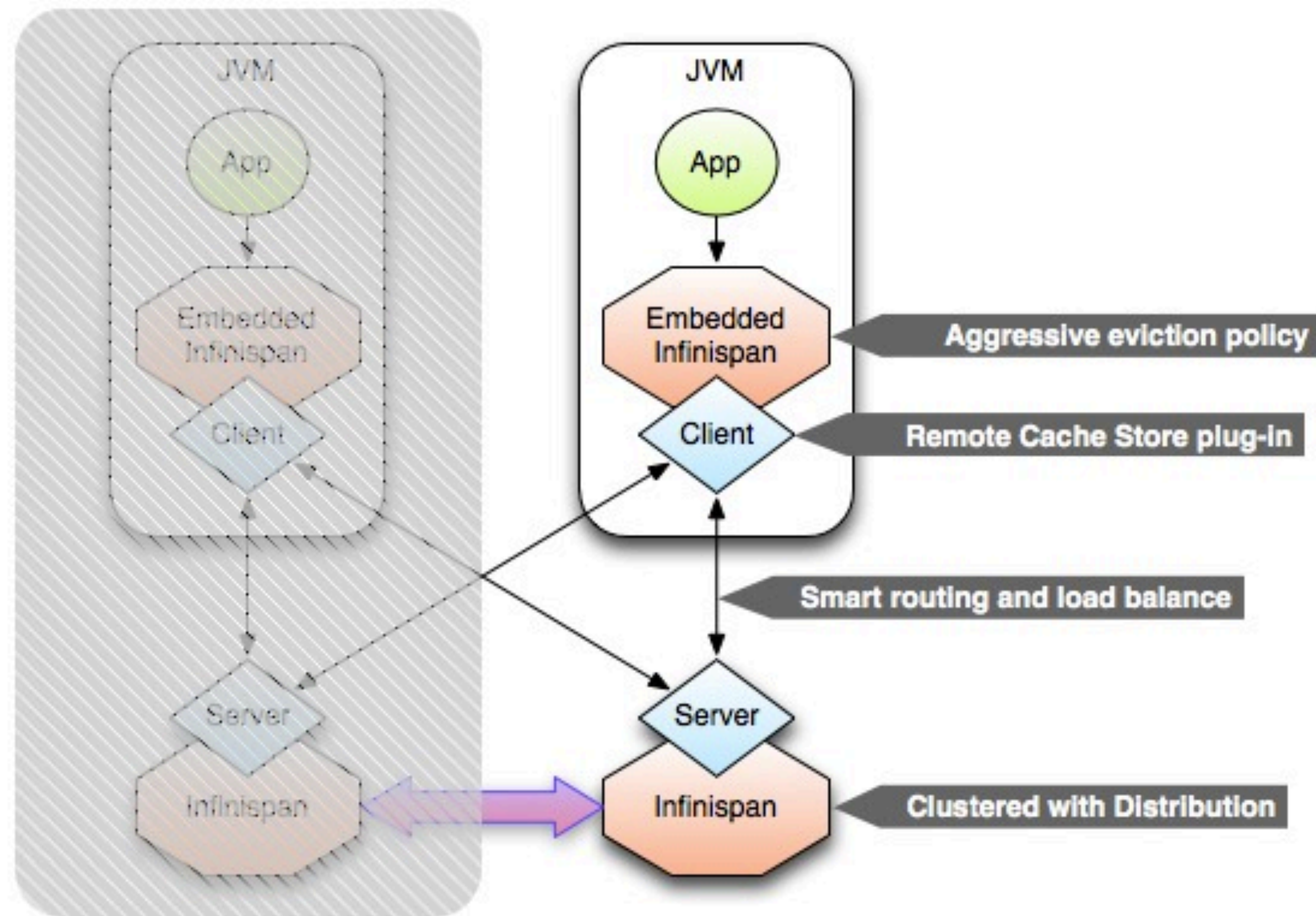


Data tier + zero latency?

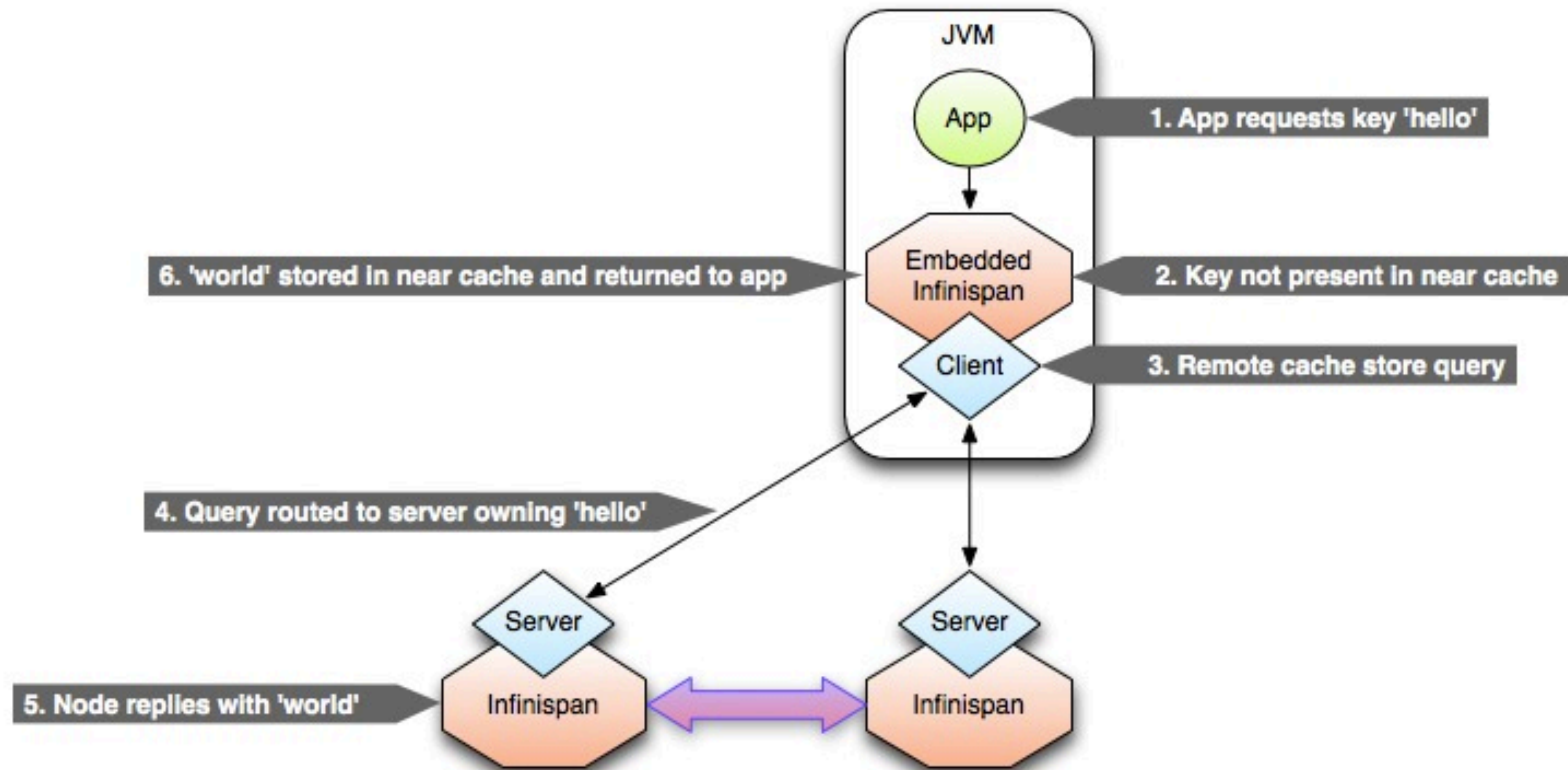
Near caches



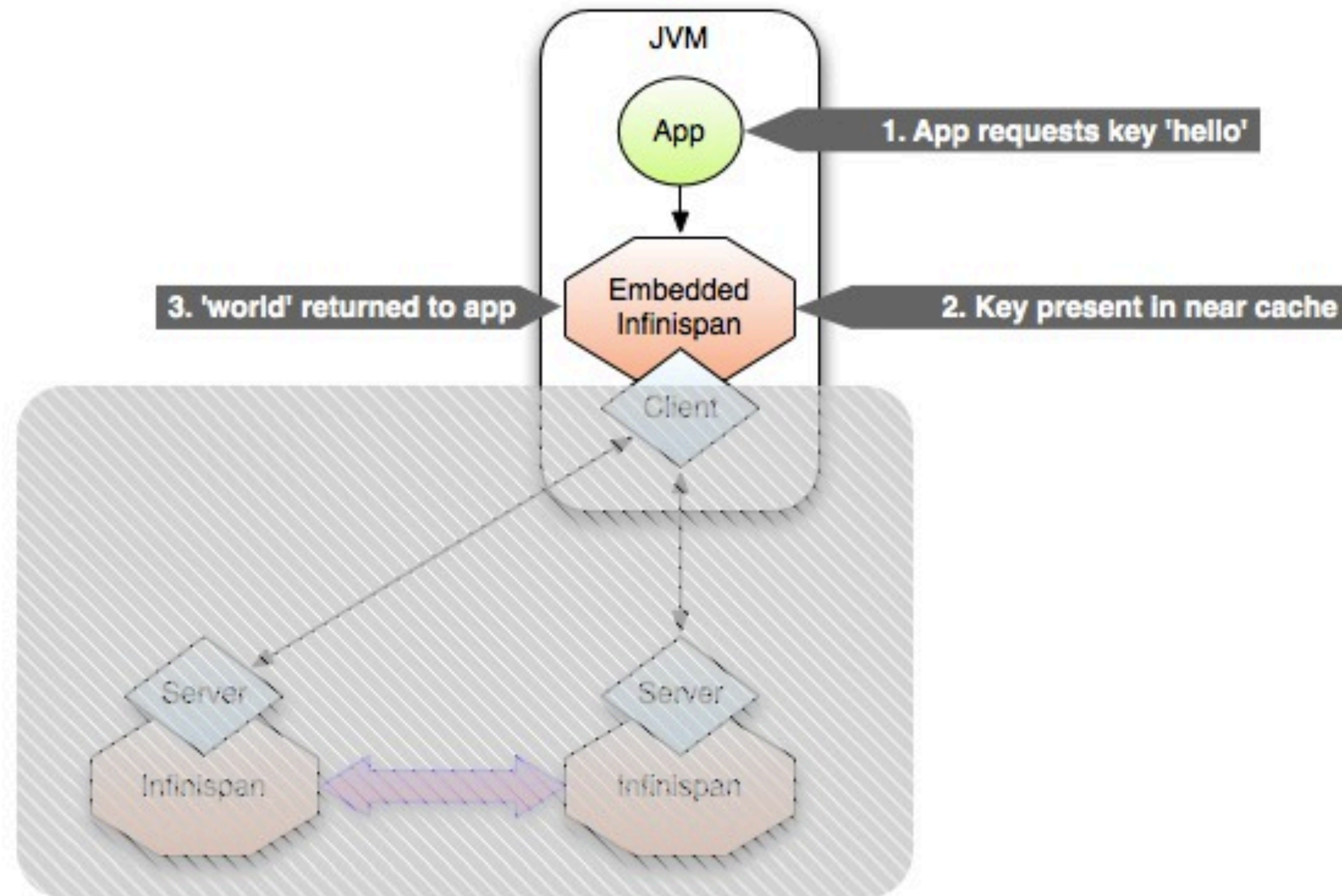
Tiers in more detail



Near cache miss

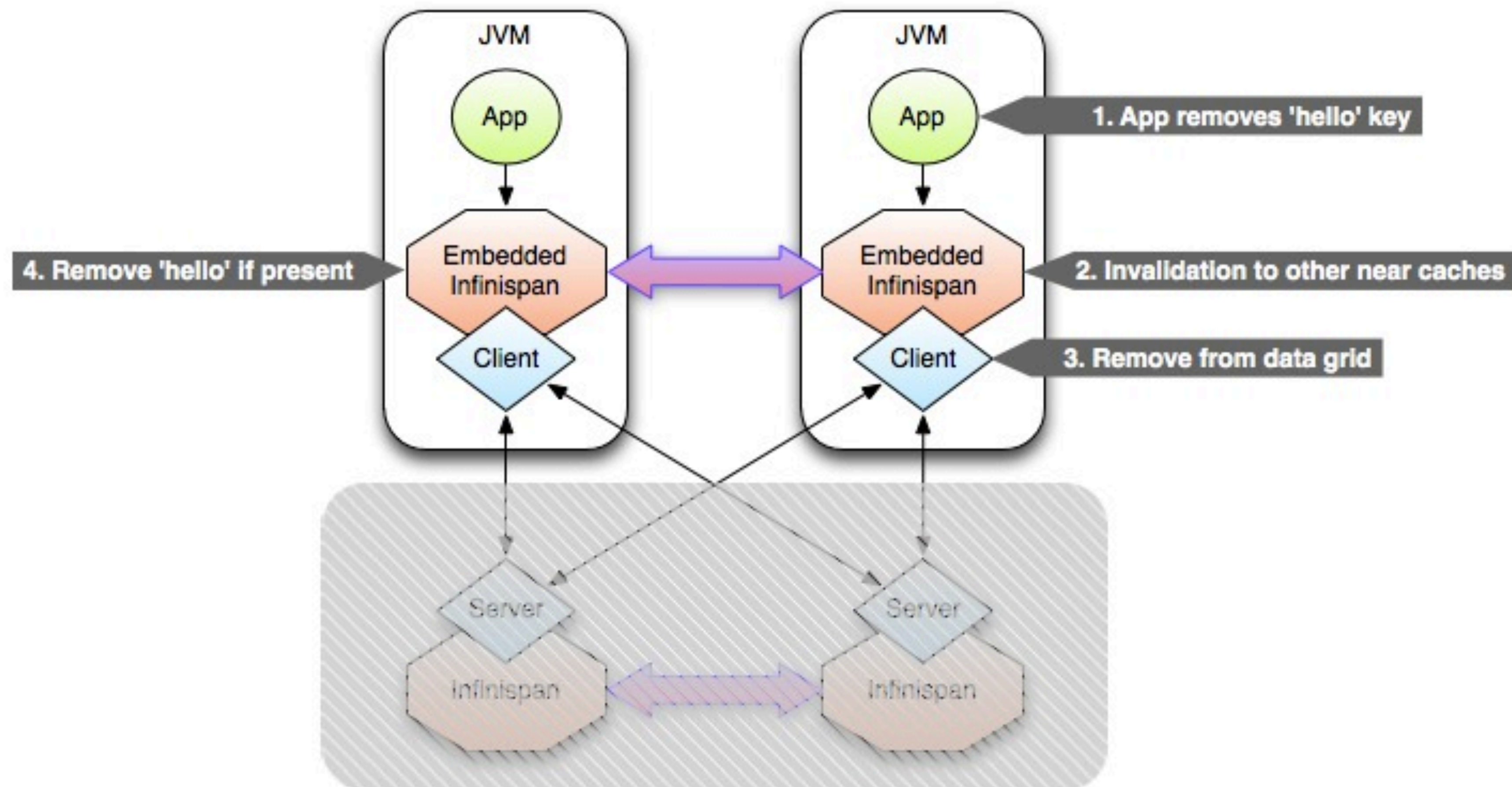


Near cache hit



How to keep near caches consistent?

Cluster near caches



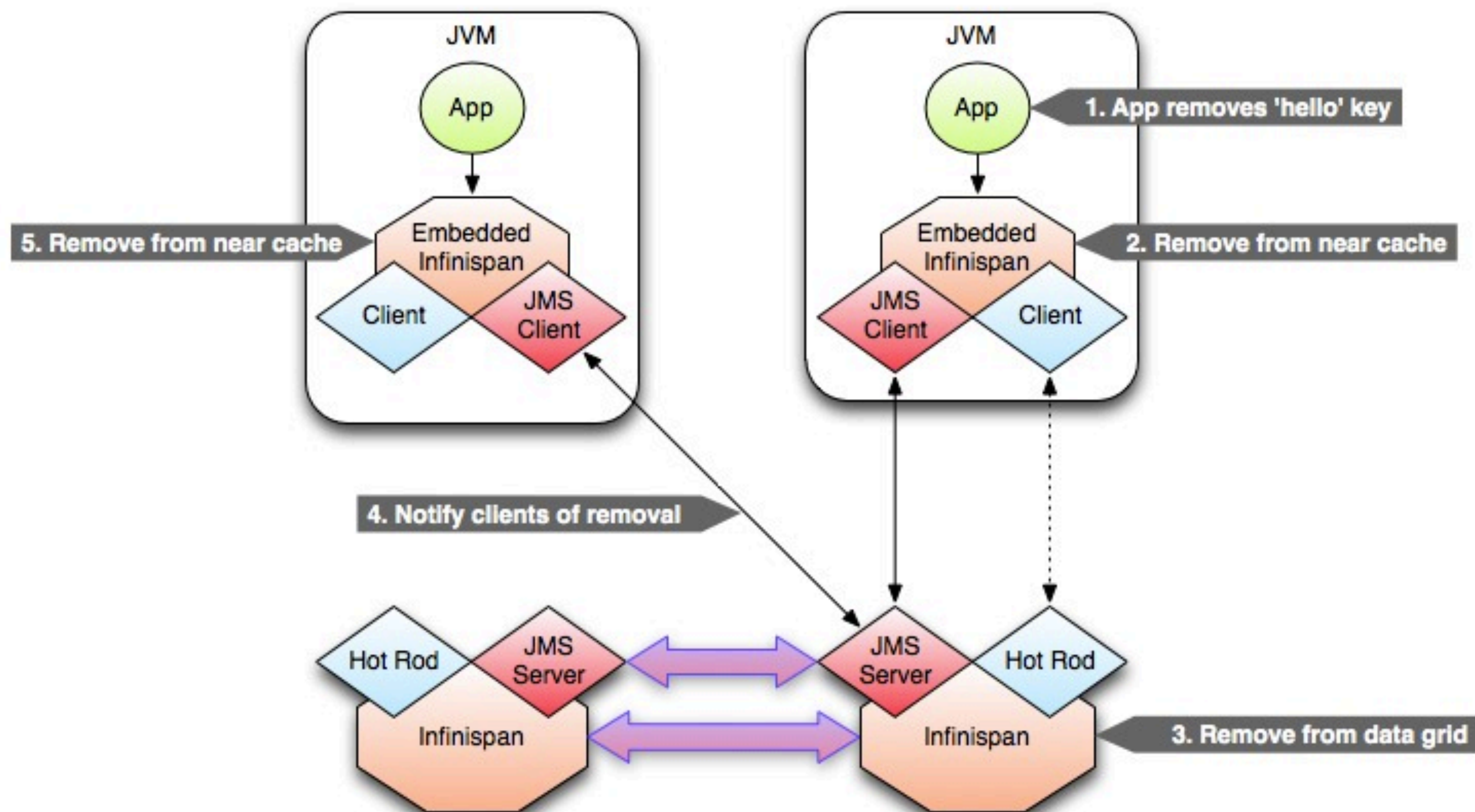
Server notifications

Clustering near caches not always possible, so server to client notifications needed

JMS to the rescue!

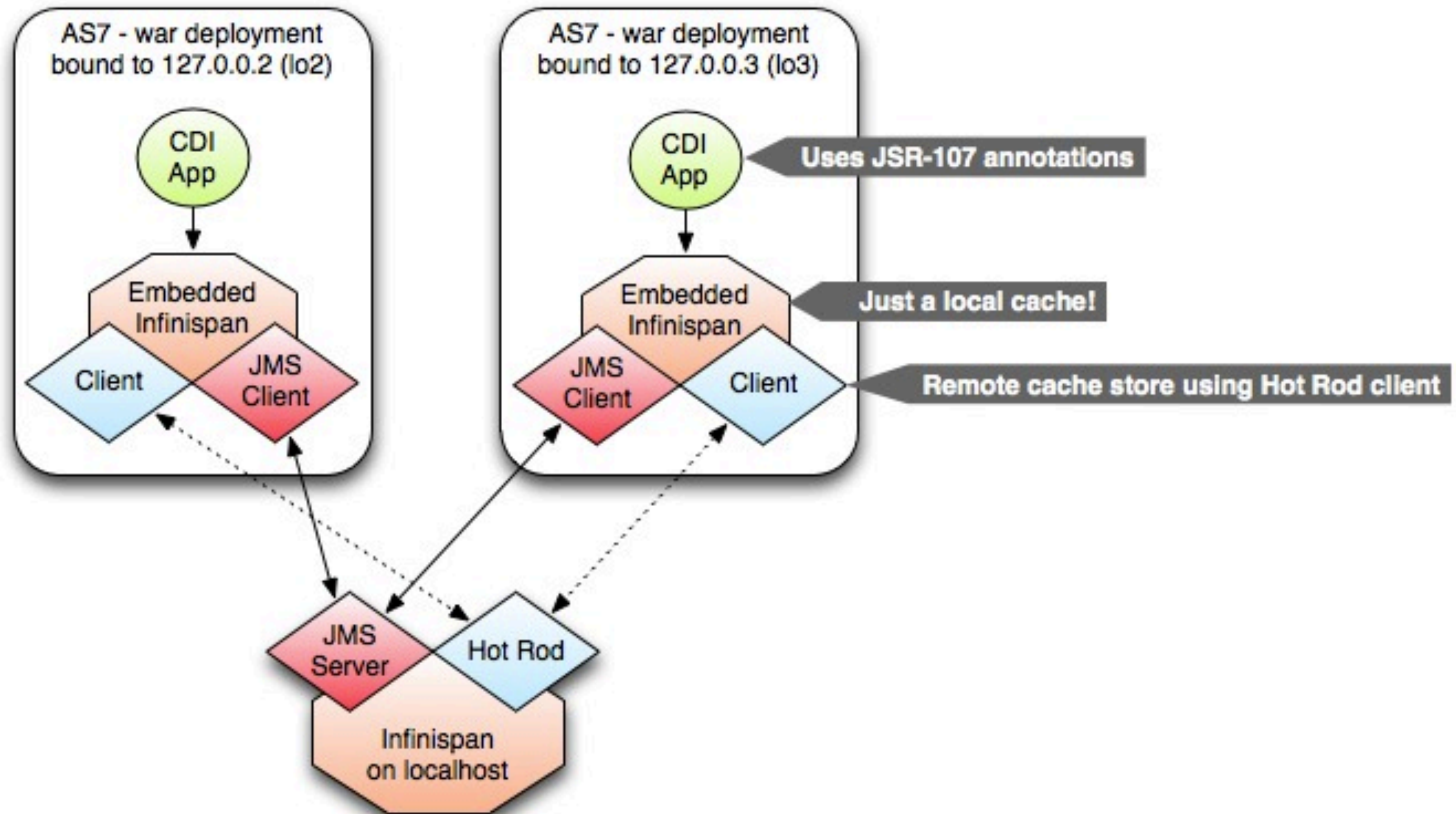
Hot Rod protocol not there yet,
but **JMS** can easily fill the gap!

Near caches with JMS



Demo

Demo architecture



Summary

Infinispan can be used as a fast, powerful local cache to boost performance

Summary

Infinispan can also be used as
a **Fast, Available, Distributed,
Elastic** data grid

Summary

Infinispan caches and data grids can be combined into **performant, scalable near caches**

Questions?

<http://infinispan.org>

<http://blog.infinispan.org>

<http://twitter.com/infinispan>

Rate this talk!

<http://speakerrate.com/galder>