

# Caching mit Spring über JCache (JSR-107) hinaus

created by Andreas Wirth 03/2017

# Andreas Wirth

## Bitech AG

mail: [andreas.wirth@koeln.bitech.de](mailto:andreas.wirth@koeln.bitech.de)

- IT- und Fachberatung
- Individual-Softwareentwicklung
- regionaler Schwerpunkt
- unabhängig und inhabergeführt

# Agenda

- Zusicherung von JCache (JSR-107)
- Wie verwende ich JCache?
- Spring Cache Abstraction
- Demo - Spring-Cache
- Weitere Möglichkeiten durch AOP

# Begriff schärfen

*"In the context of the Java Caching API the term Caching describes the technique whereby Java developers use a Caching Provider to **temporarily** cache **Java objects**."*

[Greg LUCK, Brian OLIVER: JSR107FinalSpecification 1.2]

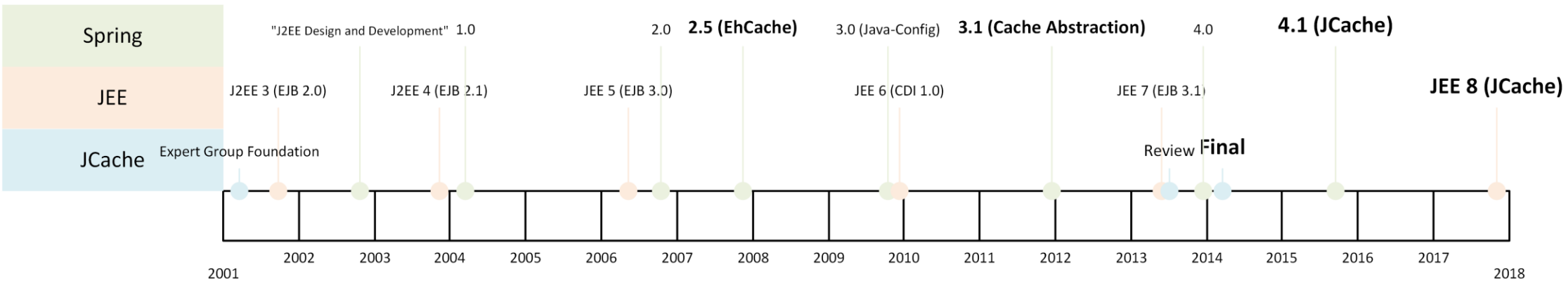
# Anwendungsfälle

*"It is often assumed that information from database is being cached. This however is not a requirement of caching. Fundamentally **any information that is expensive or time consuming to produce or access** can be stored in a cache. Some common use cases are:"*

- client side caching of Web service calls
- caching of expensive computations such as rendered images
- caching data
- servlet response caching
- caching of domain object graphs

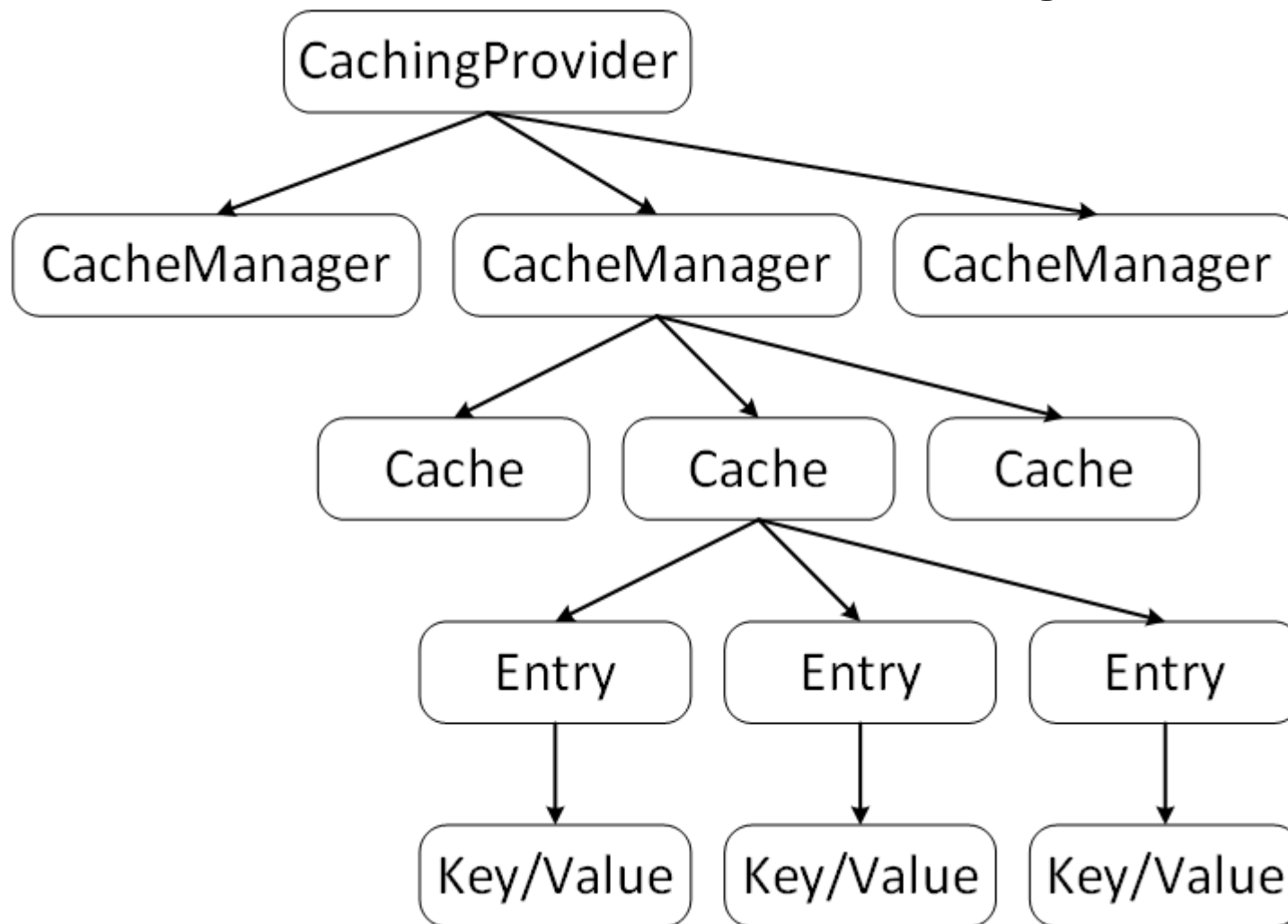
[Greg LUCK, Brian OLIVER: JSR107FinalSpecification 1.2]

# Timeline



[Andreas WIRTH, Javaland 03-2017]

# JCache Core Concepts



[Abhishek GUPTA, <https://dzone.com/refcardz/java-caching>

# Demo - JCache



# Demo JCache

## Zusammenfassung

- 5 JCache Core Concepts
- Listener
- Expiry Policy

## beachte

- footprint
- annotations → context

# Programmatisch vs. Annotation

programmatisch	Annotation
cache.put	@CachePut
cache.get	@CacheResult
cache.remove	@CacheRemove
cache.removeAll	@CacheRemoveAll

# Spring Cache Abstraction

seit Version 3.1 (Ende 2011)

JCache	Spring
@CachePut	@CachePut
@CacheResult	@Cacheable
@CacheRemove	@CacheEvict
@CacheRemoveAll	@CacheEvict(allEntries=true)
@Cache Defaults	@CacheConfig

# Spring Cache Abstraction II

## über JCache hinaus

- Adapter für EhCache, Caffeine, ...
- xml config (programmatisch, annotation)
- synchronized caching
- conditional caching
- key generator

# Demo - spring-jcache

# Demo spring-jcache

## Zusammenfassung

- 100% JCache
- spring jdk-proxy
- cache condition

## beachte

- cache condition unless → result
- synchronized caching
- spring aop-proxy

# Demo - spring-jcache-aop-proxy

# Demo spring-jcache-aop-proxy

## Zusammenfassung

- aop-proxy → call Reihenfolge

## beachte

- aop → instrument



# Demo - spring-jcache-aspect

# Demo spring-jcache-aspect

## Zusammenfassung

- **spring-context im Aspekt**
- Keygenerator

## beachte

- Aspekt auch für JCache-Caching möglich

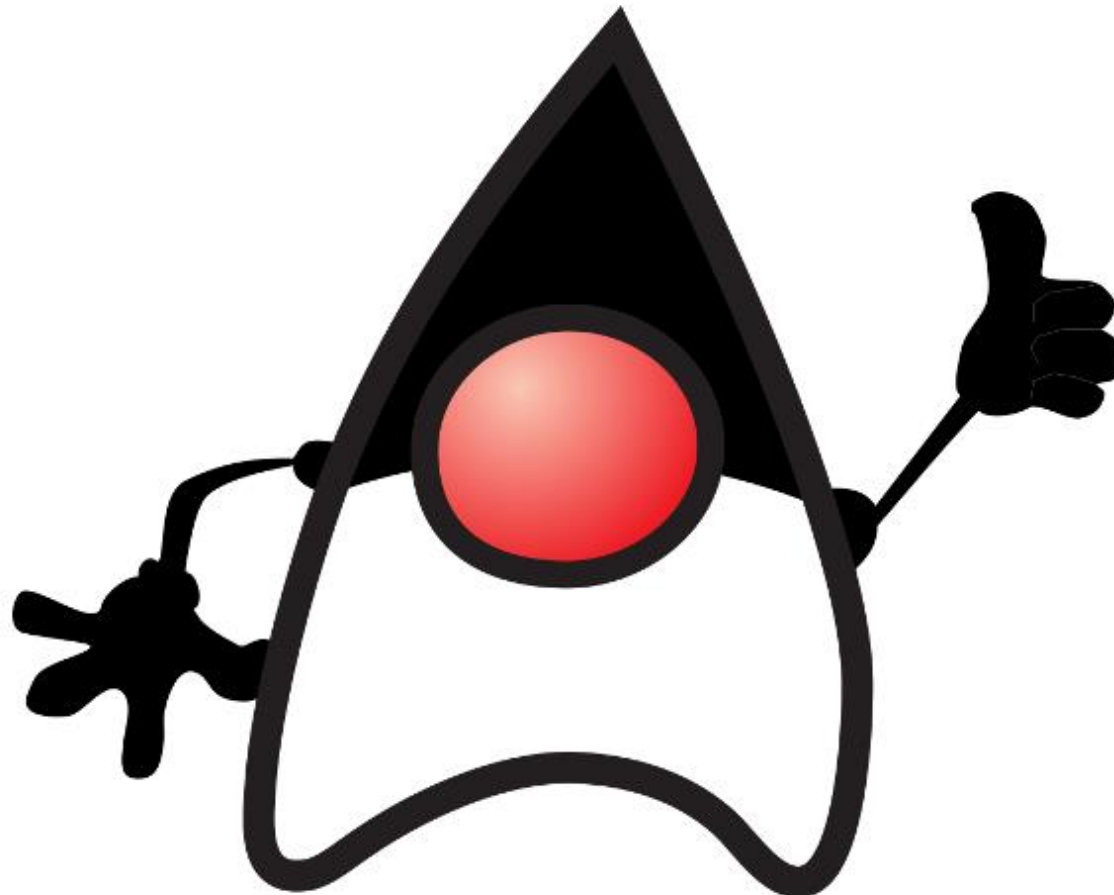
# Zusammenfassung

1. JCache/Spring-Caching sind einfach zu verwenden
2. herstellerunabhängig
3. Spring bietet kleine (aber feine) zusätzliche Features
4. **Aspekte werden mit Spring-Context noch mächtiger**
5. **Caching-Strategy nicht** Bestandteil von JCache

# Quellen

- <https://jcp.org/en/jsr/detail?id=107>
- <https://github.com/jsr107/RI>
- <http://docs.spring.io/spring/docs/current/spring-framework-reference/htmlsingle/>
- <https://dzone.com/refcardz/java-caching>
- <http://hazelcast.com/content/Using-JCache-with-Hazelcast-Slides.pdf>
- <http://www.tomitribe.com/blog/2015/06/using-jcache-with-cdi/>
- <http://www.ehcache.org/documentation/3.0/107.html>
- [http://www.ehcache.org/blog/2016/05/18/ehcache3\\_jsr107\\_spring.html](http://www.ehcache.org/blog/2016/05/18/ehcache3_jsr107_spring.html)
- <https://www.slideshare.net/SpringCentral/spring-one2gx-caching-with-spring?qid=8fd5ecbc-f5e7-4c84-b984-ce4aa5a9fd4e&v>

# Danke für eure Aufmerksamkeit



# Q & A