Cache Abstraction: JCache (JSR-107) Annotations Support

Spring's caching abstraction is available as from Spring 3.1 and it was about time to show it some more love. In this post, I want to walk you through the major improvement in that area which is the JCache (JSR-107) annotations support.

As you may have heard, JSR-107 went final after all, 13 years after the initial proposal. For those who are familiar with Spring's caching annotations, the following table describes the mapping between the Spring annotations and the JSR-107 counterpart:

Spring	JSR-107
`@Cacheable`	`@CacheResult`
`@CachePut`	`@CachePut`
`@CacheEvict`	`@CacheRemove`
`@CacheEvict(allEntries=true)`	`@CacheRemoveAll`

JCache annotations

Let's first look at each annotation and describe how they can be used. This will be a chance to better understand what they support with regards to what you've been used to with the Spring annotations and more importantly the **new** features that these annotations bring.

@CacheResult

`@CacheResult` is fairly similar to `@Cacheable`, the following rewrites the original example using the `@CacheResult` annotation:

```
@CacheResult(cacheName = "books")
public Book findBook(ISBN isbn) {...}
```

COPY

Keys generation can be customized using the `CacheKeyGenerator` interface. If no