Handling Events & Using Efficient Data Operations

Grant Little

www.grantlittle.me

Overview

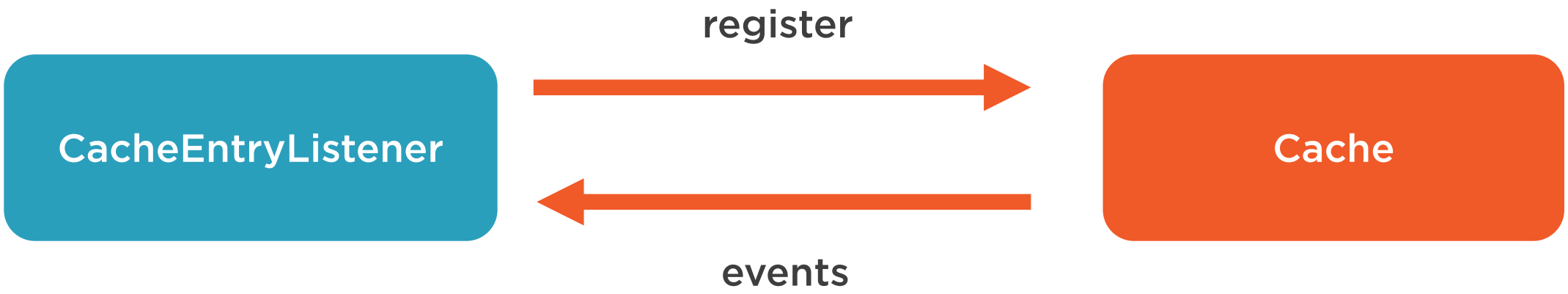
Cache events

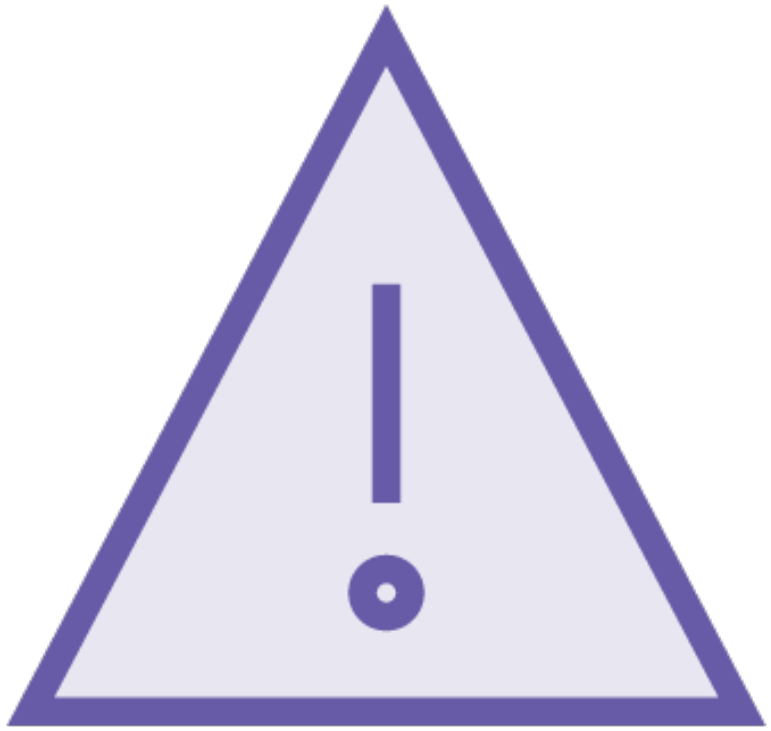
- Added
- Updated
- Removed
- Expired

EntryProcessors

Listening to Cache Events

Cache Entry Listeners





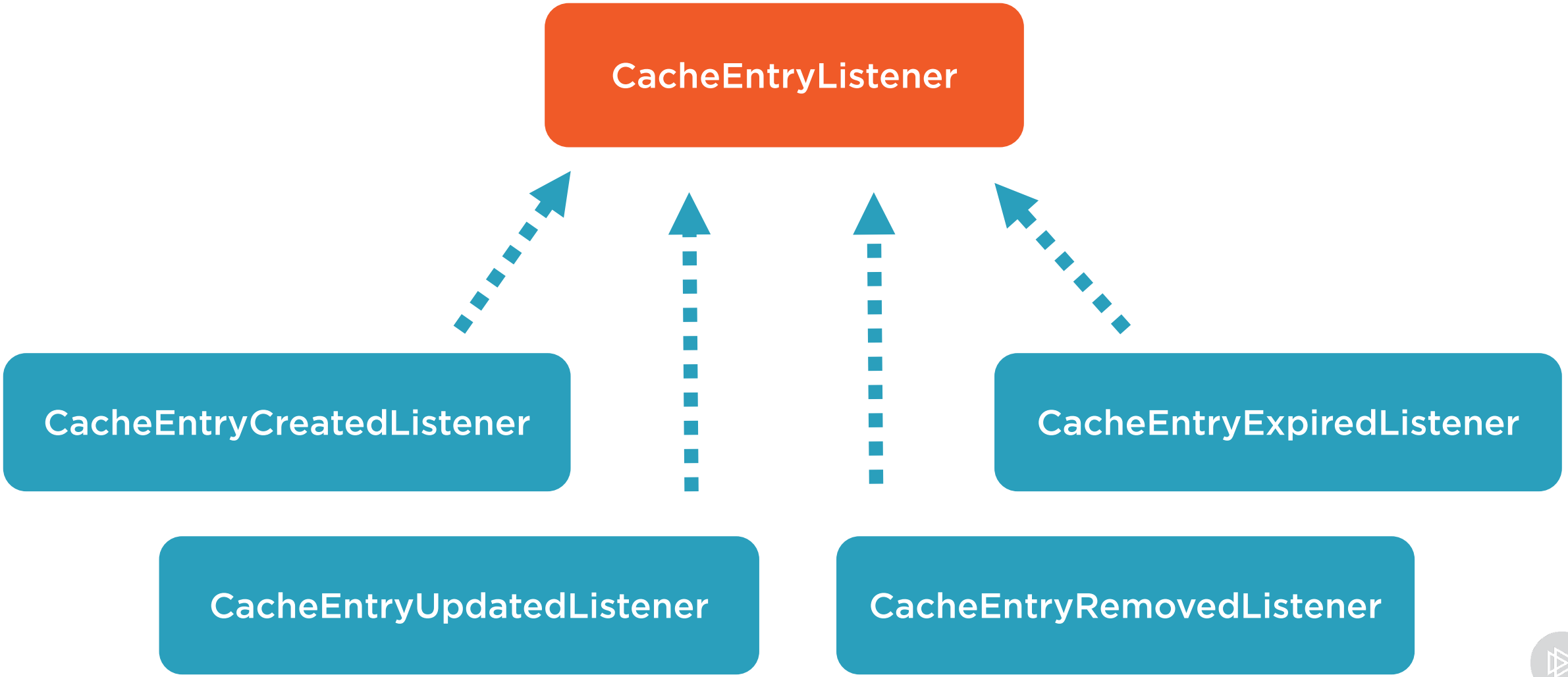
Created (added) to cache

Modified in the cache

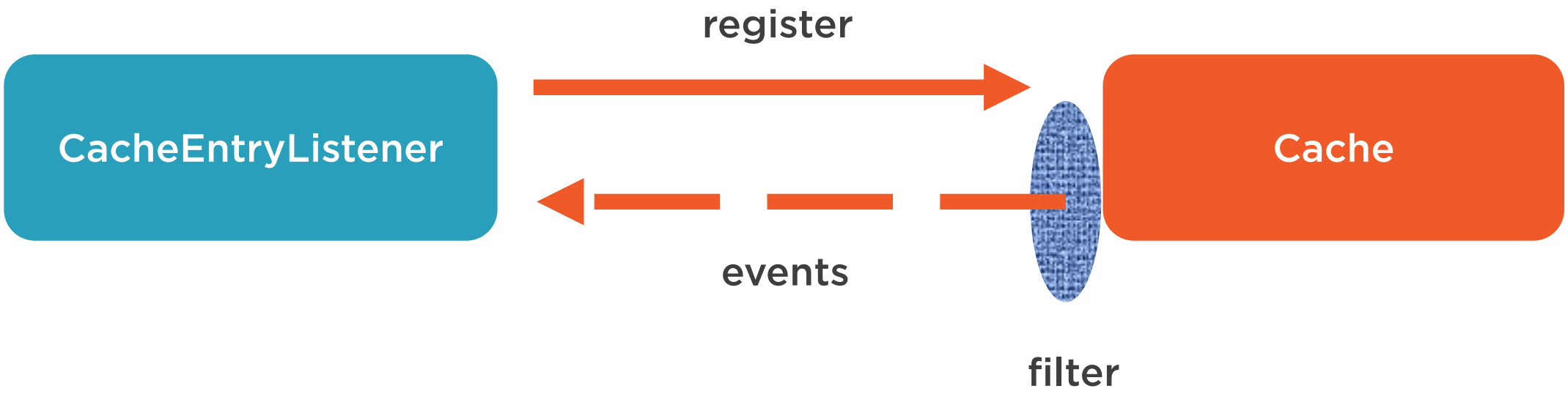
Removed from the cache

Expired from the cache

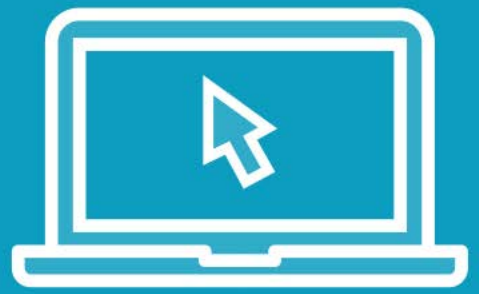
Class Hierarchy



Cache Entry Listener Filter



Demo



Defining CacheEntryListeners at Configuration Time (Startup)

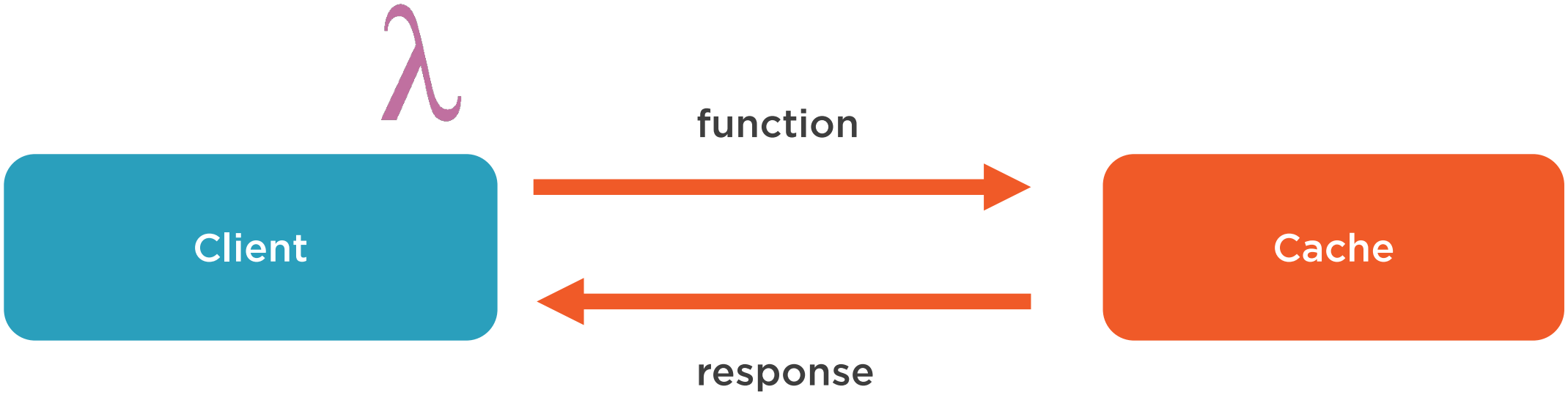
Efficient Processing of Data

Some operations involve locking

Don’t pull the data to the worker

Send the worker to the data

Entry Processor



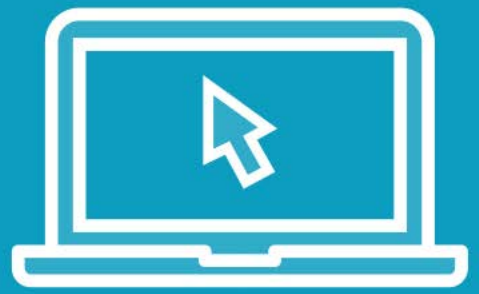
EntryProcessors

- Read Data
- Mutate Data
- Delete Data

Atomic Operation

Cache Entry Listeners invoked once

Demo



CustomerEntryProcessor

- Change last name of customer
- Calculate & return age of customer

Why not use this approach?

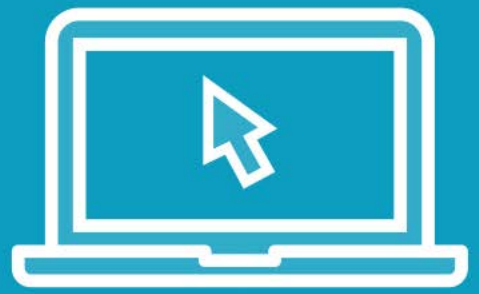
EntryProcessor possibly sent to remote nodes

- Network Latency
- Multiple network requests

Individually invoking the EntryProcessor for each key

- Synchronous

Demo



Exceptions in EntryProcessors

(cache.invokeAll(....))

Review

Cache Entry Listeners

Cache Entry Listener Filter

Entry Processors