Crossing the flux with Redis and Spring Webflux

DevDay Benevento - Davide Cerbo - @davide_cerbo

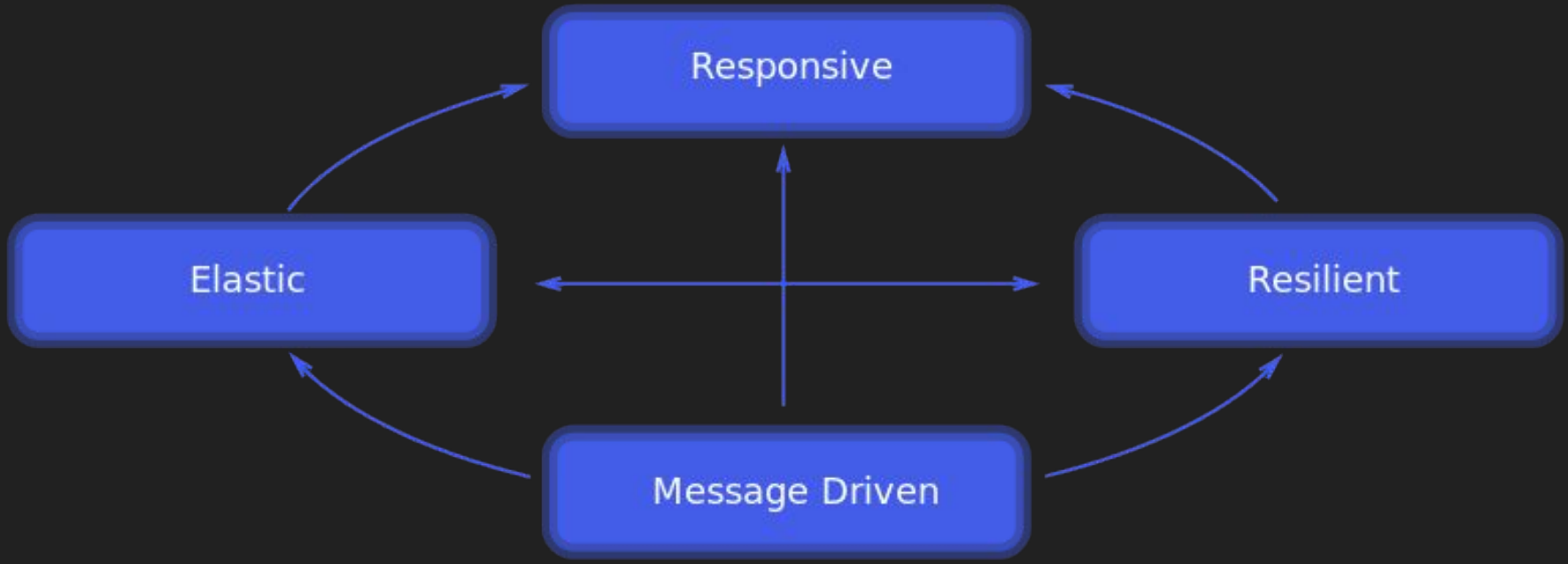
Davide Cerbo

15+ years as developer

@davide_cerbo

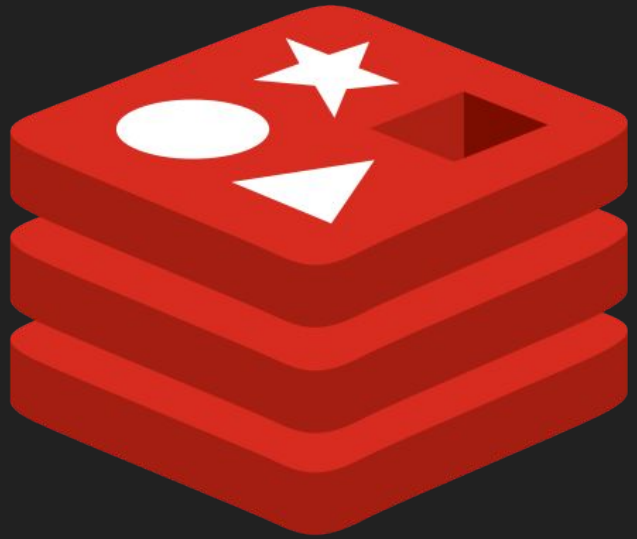
https://medium.com/@davidecerbo

Reactive System



Reactive Programming

Reactive programming, that is a non-blocking alternative to traditional programming solutions, working entirely with event driven data streams and functional programming concepts to manipulate these streams and don’t forget to consider the failure as first citizen, in any kind of scenario.



redis

Redis collections

- Strings
- Lists
- Hashes
- Sets
- Sorted sets

https://www.cheatography.com/tasjaevan/cheat-sheets/redis/

Publish / subscribe

publish key value

subscribe key

psubscribe key*

Keyspace notification

config set notify-keyspace-events KEA

PUBLISH __keyspace@0__:mykey del

PUBLISH __keyevent@0__:del mykey

EXPIRE and SETEX

set name davide

expire name 10

setext name 10 davide

EXPIRE + Keystore notification = timer

PSUBSCRIBE __keyspace@0__:timer:*

SETEX timer:first 10 test

Redis as durable persistence system

- Append only file
- Snapshot
- fsynch

http://oldblog.antirez.com/post/redis-persistence-demystified.html

Spring framework

Kotlin

Step by step

- **Declarative/imperative**:

  for (String a : list)
- **Functional**:

  ```
  list.stream()
      .forEach(...)
  ```
- **Reactive**:

  ```
  Flux.from(list)
      .doOnNext(...)
      .doOnError(...)
      .doOnSuccess(...)
      .subscribe()
  ```

How do I…?

https://projectreactor.io/docs/core/release/reference/#faq

Demo

https://github.com/jesty/reactiveredis

Frontend

```
const url ='http://localhost:8080/...';
this.eventSource = new EventSource(url );
this.eventSource.onmessage = (event => {
  …
 });
 this.eventSource.onerror = (error => { … });
```

Testing

https://github.com/kstyrc/embedded-redis

Thanks!

?