

Please

Ask questions
through the app



Rate Session

Thank you!



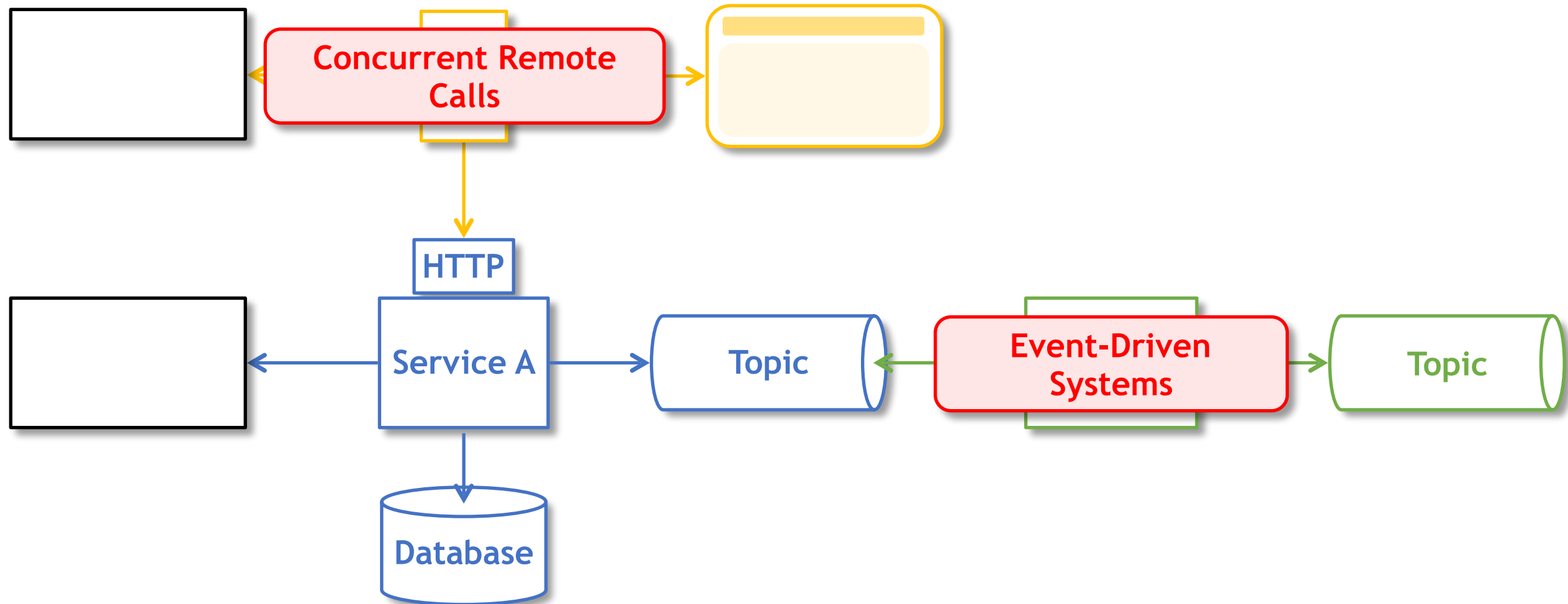
Reactive Streams in the Web

Florian Stefan | eBay Classifieds Group | GOTO Berlin 2017

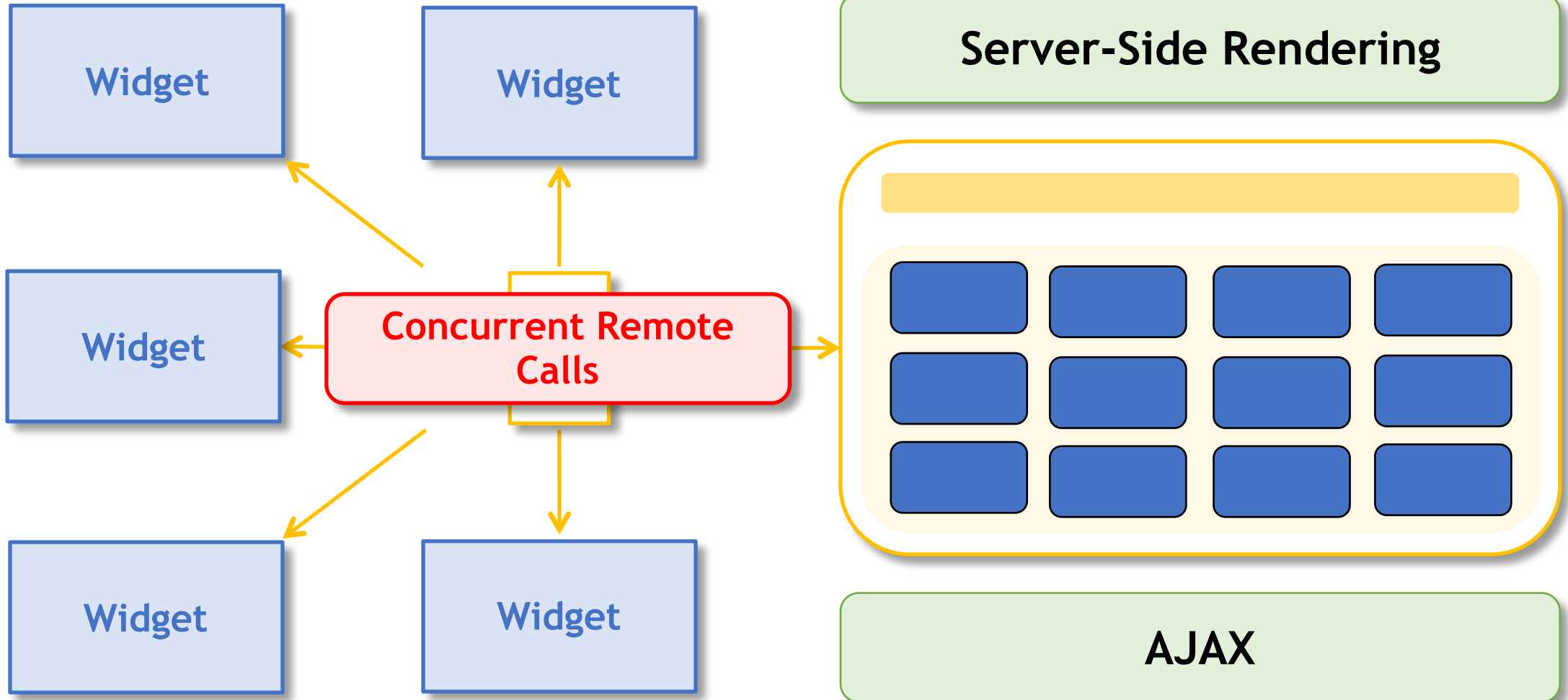
Who am I?

- Florian Stefan
- mobile.de (eBay Classifieds Group)
- <https://ebaytech.berlin/>
- fstefan@ebay.com | [@f_s_t_e_f_a_n](#)
- <https://github.com/florian-stefan/>

Why are we here?



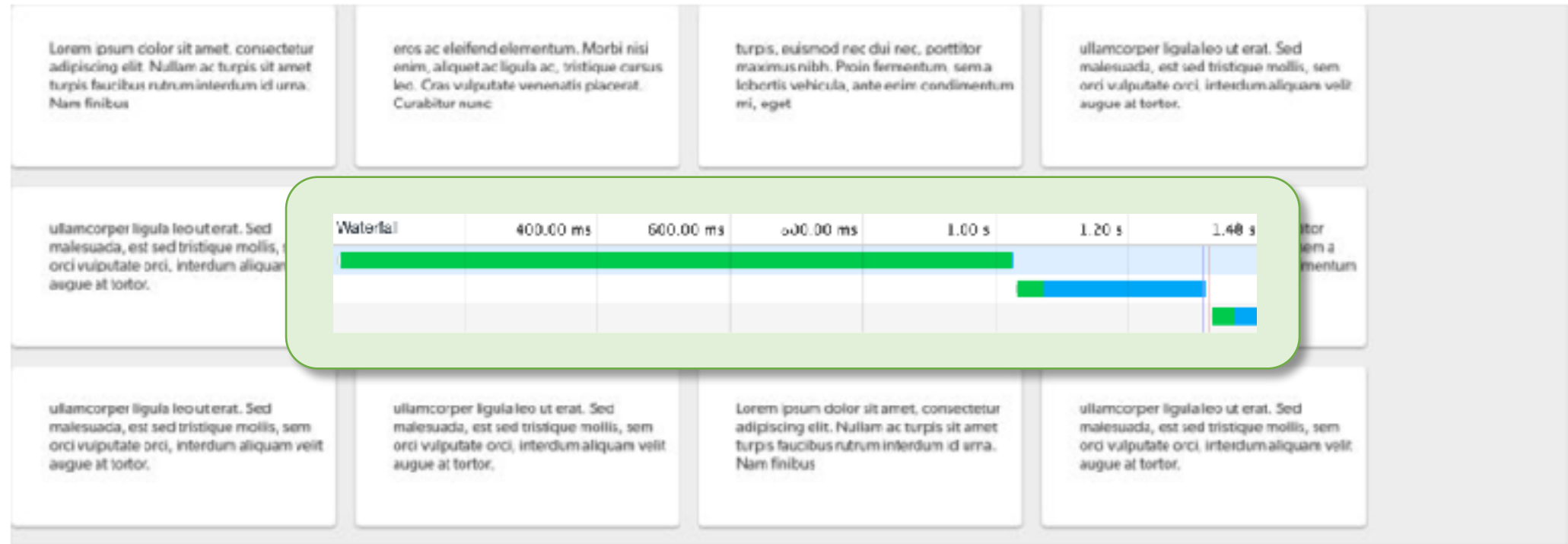
HTML Rendering



HTML Rendering

Server-Side Rendering

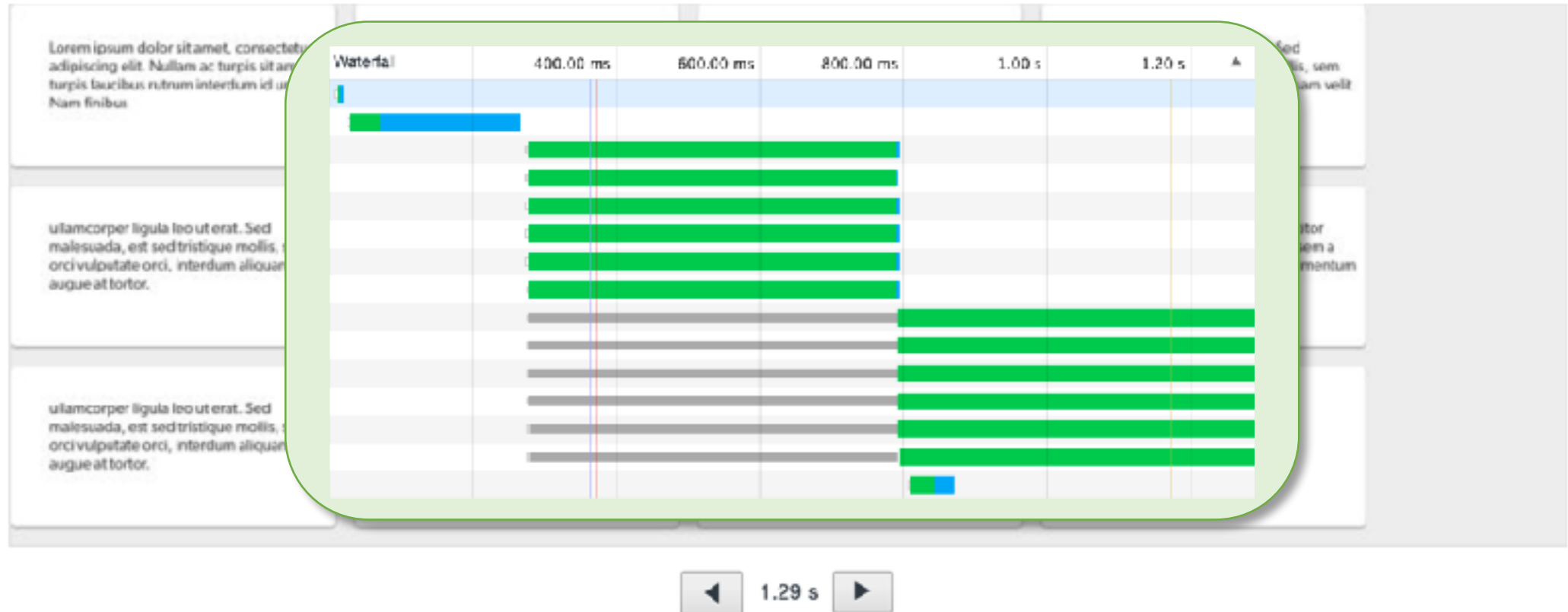
- Time To First Byte
- Prerendering / Caching



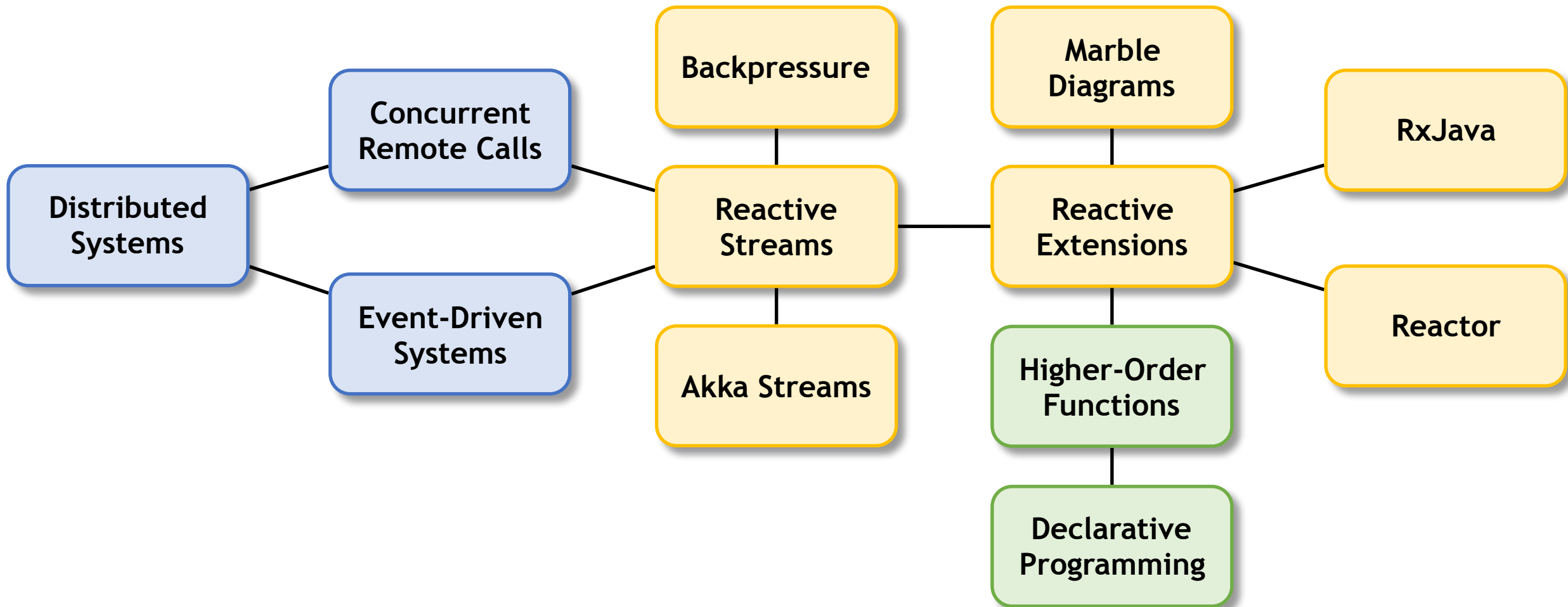
HTML Rendering

AJAX

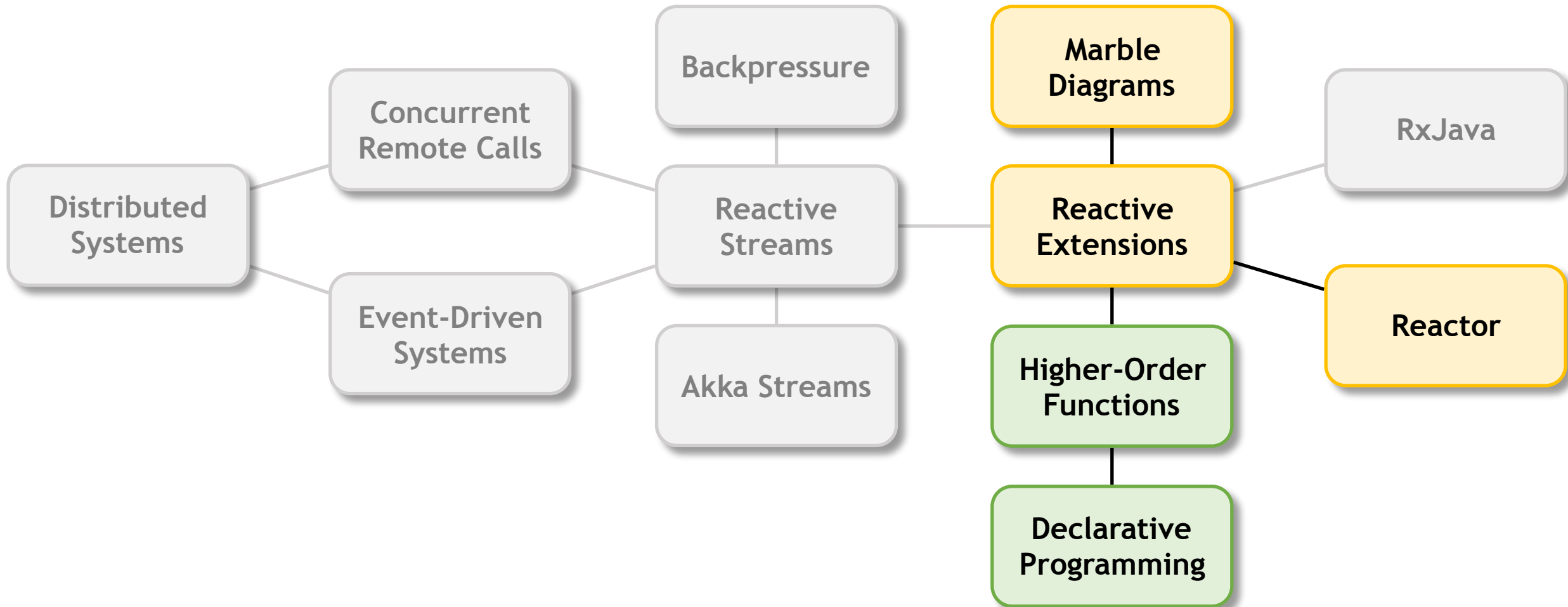
- Head-of-line blocking
- Multiplexing with HTTP/2



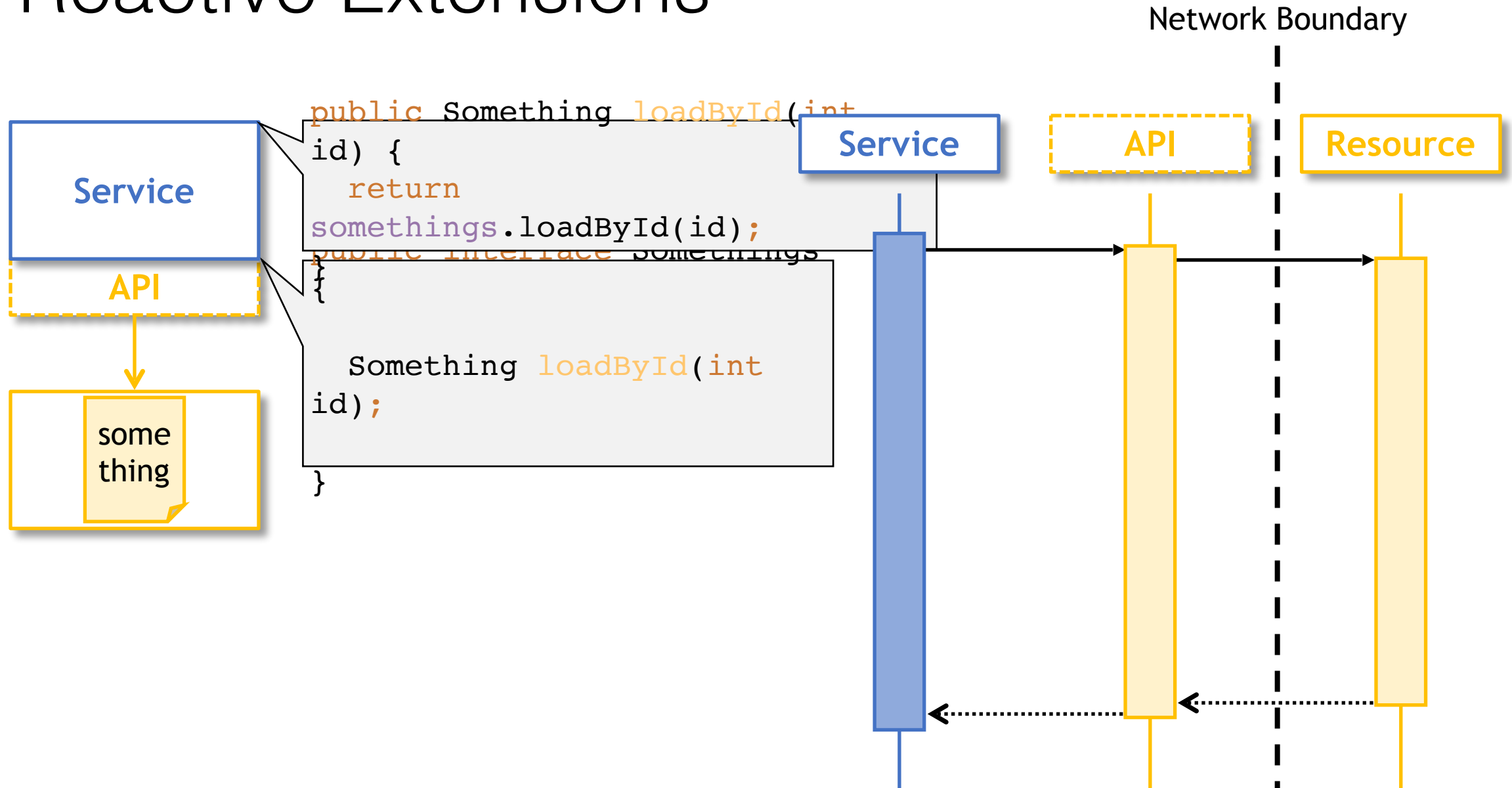
The Reactive Landscape



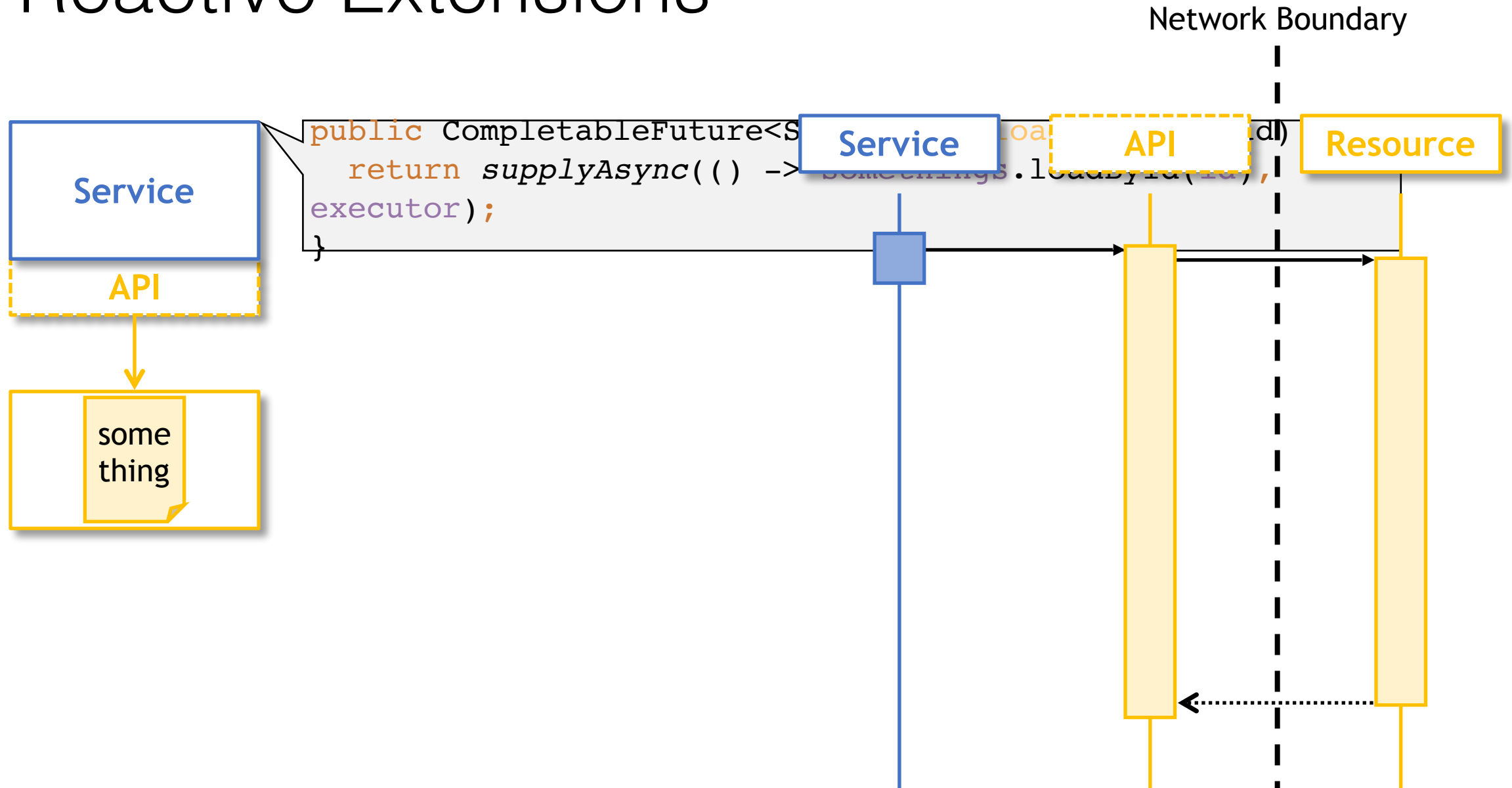
The Reactive Landscape



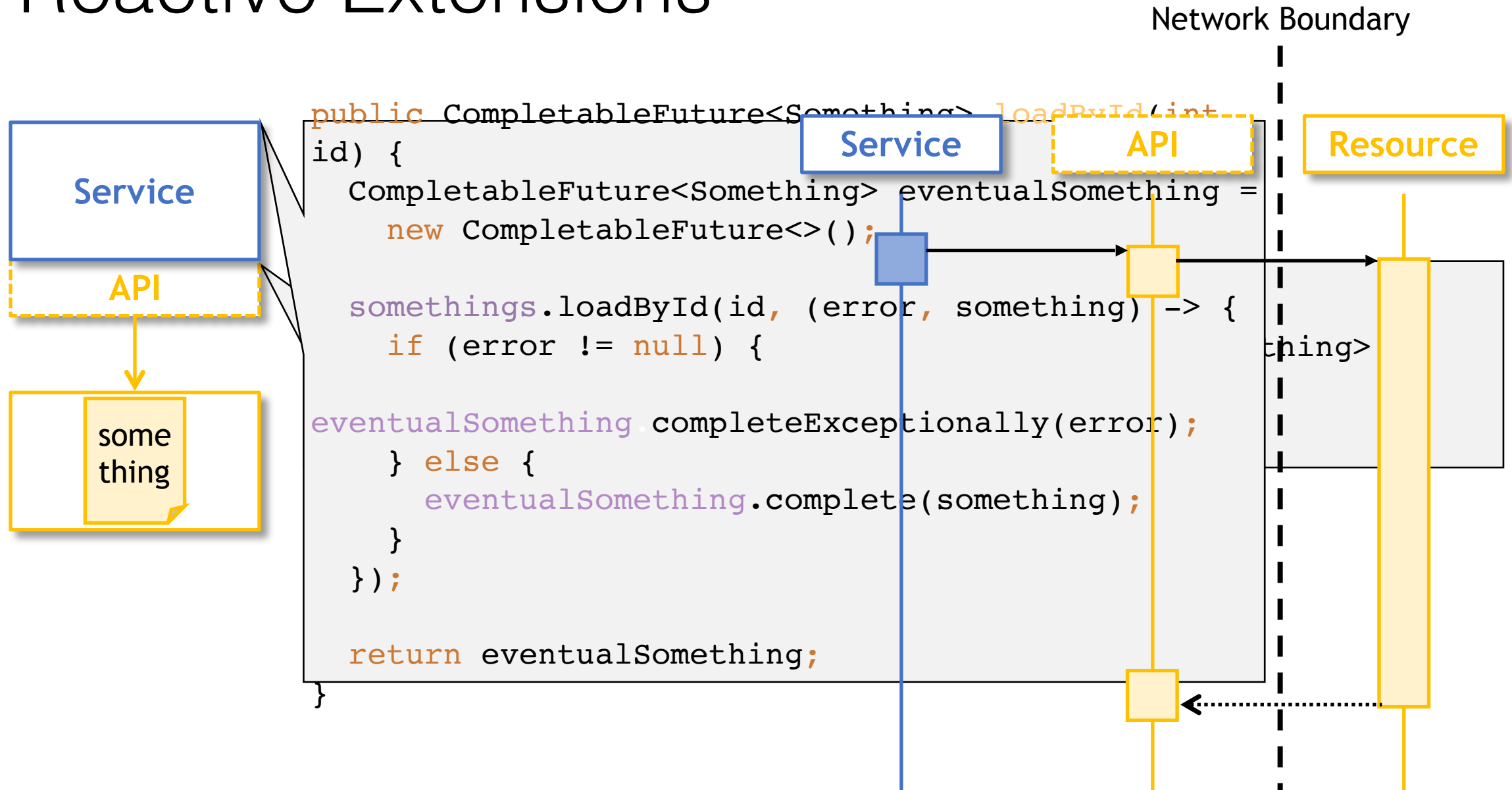
Reactive Extensions



Reactive Extensions



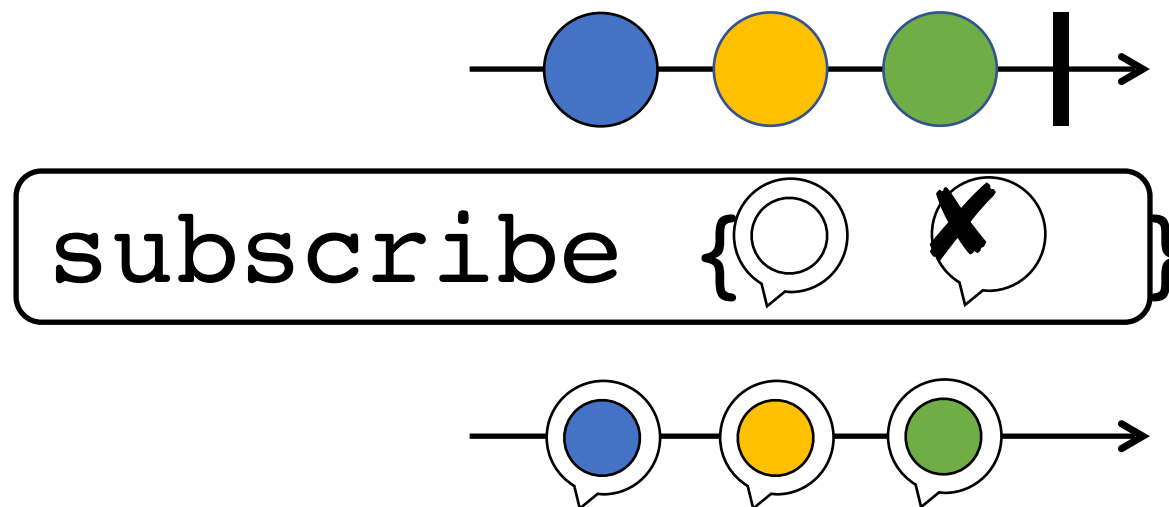
Reactive Extensions



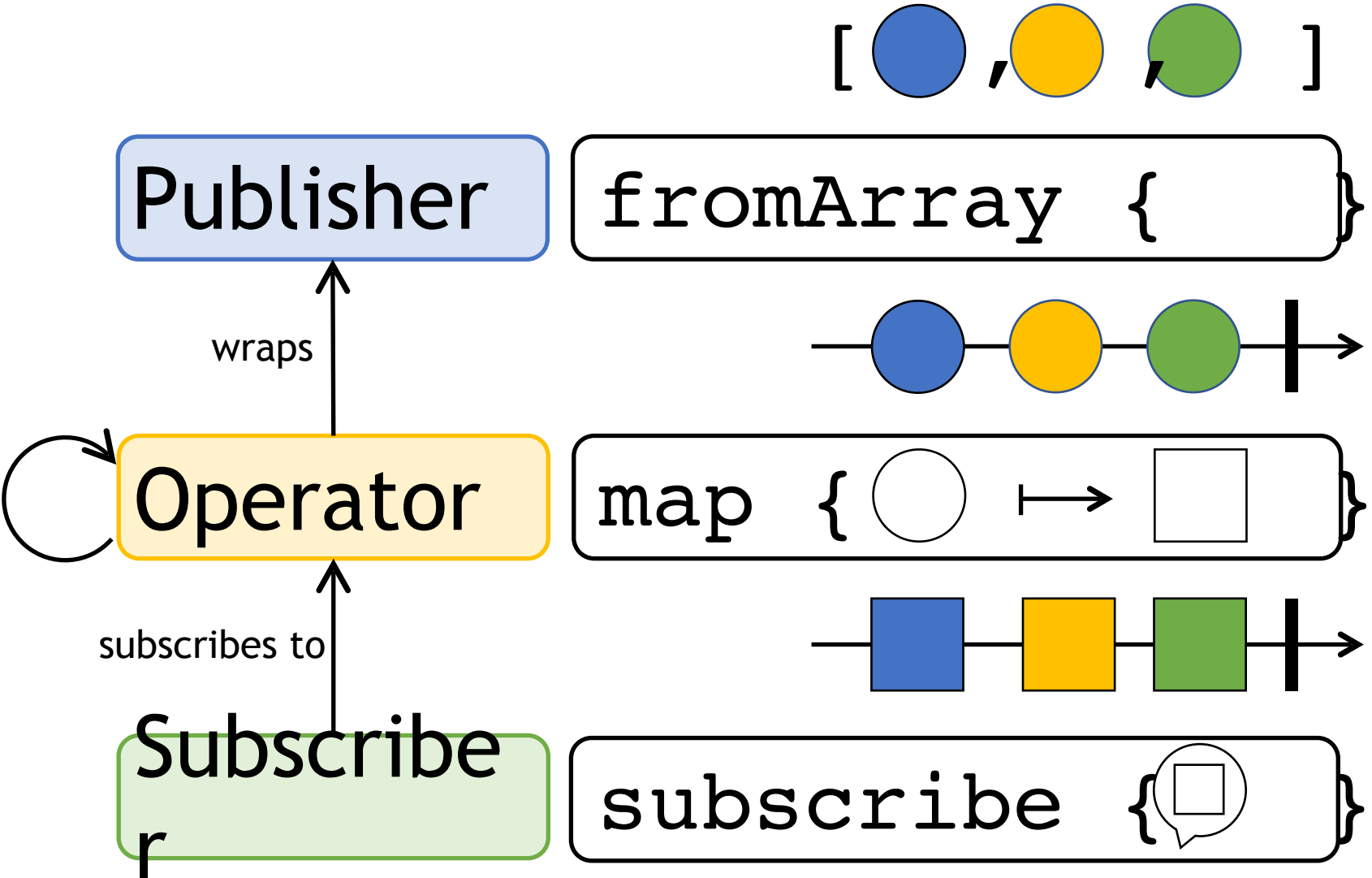
Reactive Extensions



CompletableFuture<T>	Mono<T>	Flux<T>
represents an already scheduled task	represents a building plan (caller has to subscribe)	represents a building plan (caller has to subscribe)
completes only once	emits at most one value	emits many values



Reactive Extensions



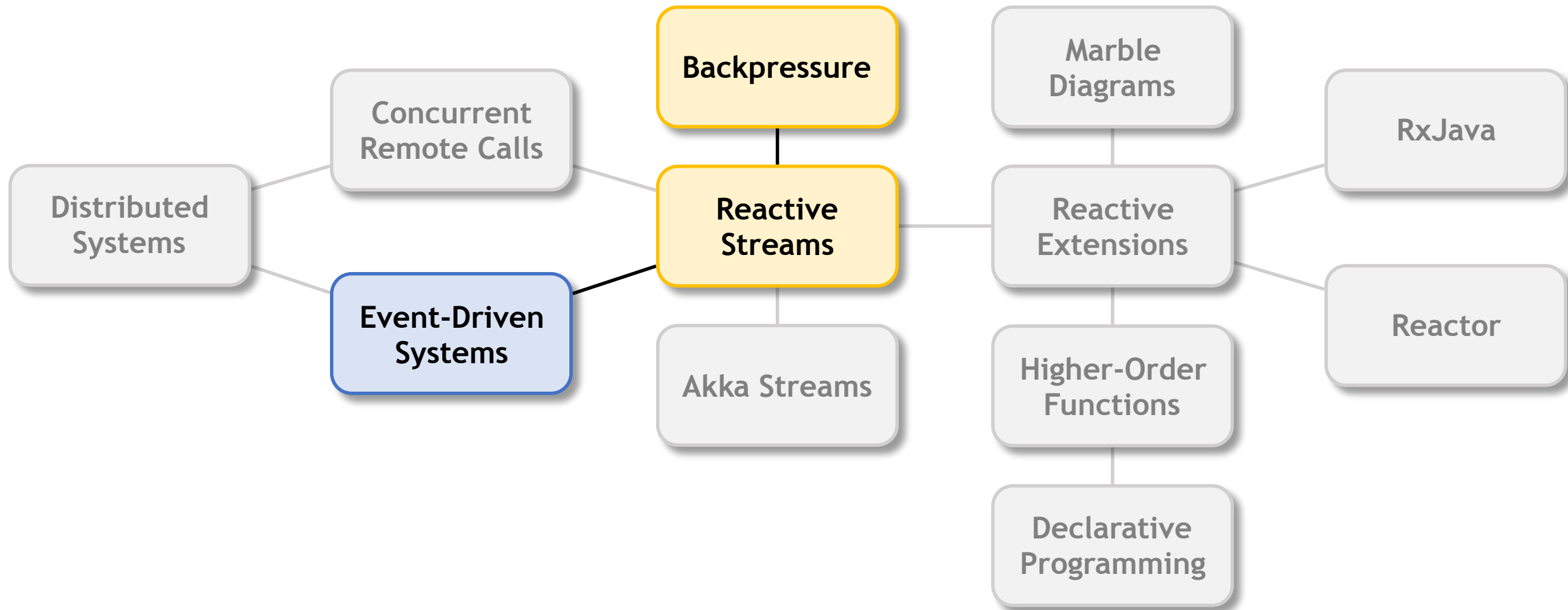
Reactive Extensions



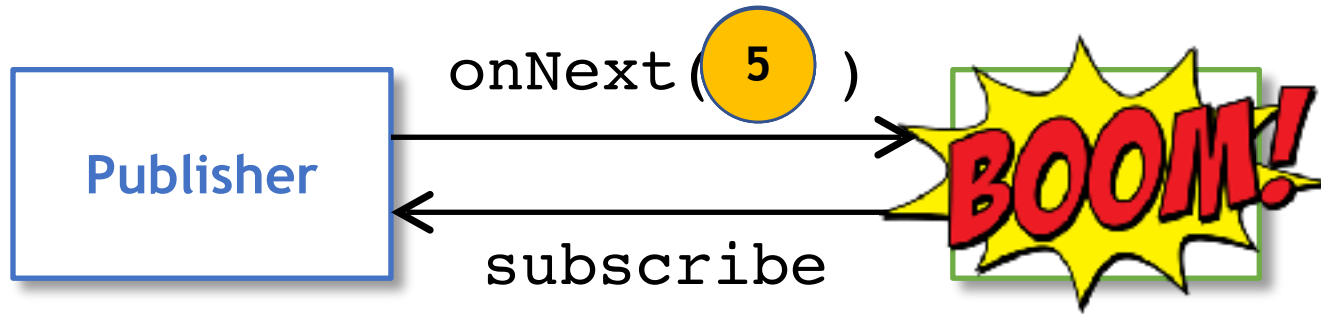
```
Circle[] circles = { Circle.of(BLUE), ... };  
  
Flux.fromArray(circles)  
    .map(circle -> Square.of(circle.getColor()))  
    .log()  
    .subscribe(...);
```

```
[main] INFO - | onSubscribe()  
[main] INFO - | ?  
[main] INFO - | onNext(Square(BLUE))  
[main] INFO - | onNext(Square(YELLOW))  
[main] INFO - | onNext(Square(GREEN))  
[main] INFO - | onComplete()
```

The Reactive Landscape

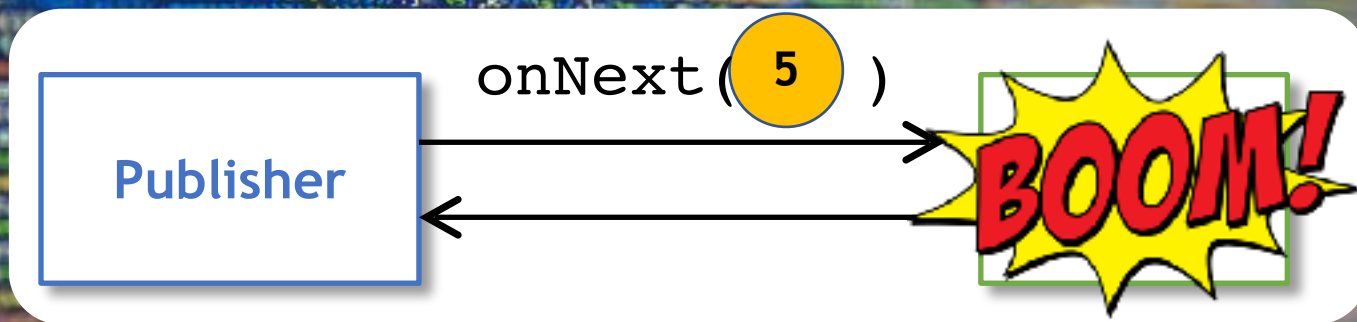


Backpressure



Backpressure

```
java.util.concurrent.RejectedExecutionException:  
Task rejected from ThreadPoolExecutor [  
    Running, pool size = 2, active threads = 2,  
    queued tasks = 4, completed tasks = 0  
]
```



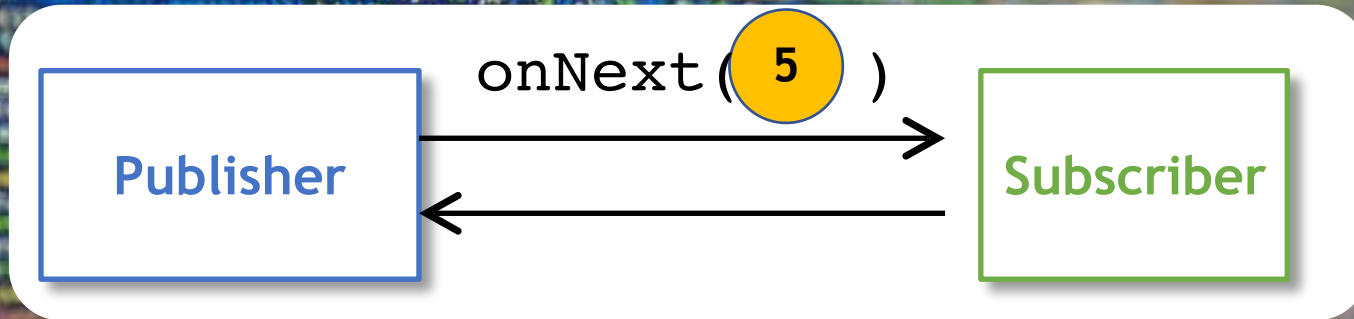
```
ExecutorService executorService =  
    new ThreadPoolExecutor(  
        nThreads, nThreads,  
        0L, TimeUnit.MILLISECONDS,  
        new LinkedBlockingQueue<>(capacity)  
    );
```

```
ids.stream()  
    .map(id -> executorService  
        .submit(() -> somethings.loadById(id)))  
    .collect(toList())  
    .stream()  
    .map(waitForSomething())  
    .forEach(handleSomething());
```

Backpressure



Backpressure



```
ExecutorService executorService =  
    new ThreadPoolExecutor(  
        nThreads, nThreads,  
        0L, TimeUnit.MILLISECONDS,  
        new LinkedBlockingQueue<>(capacity)  
    );
```

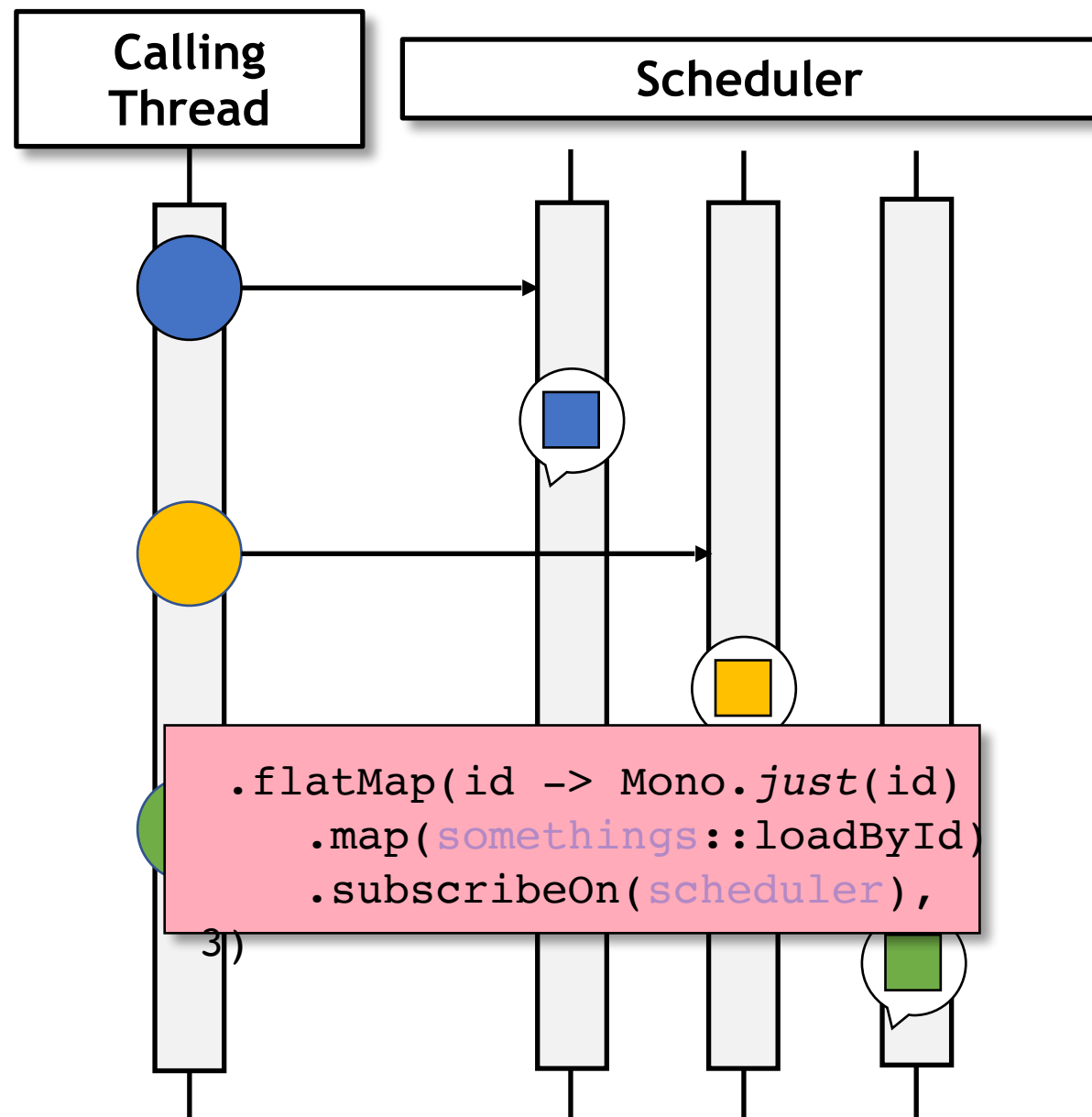
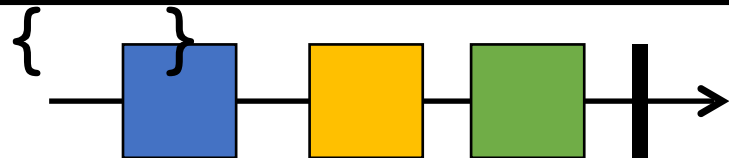
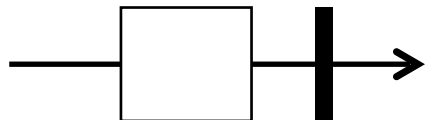
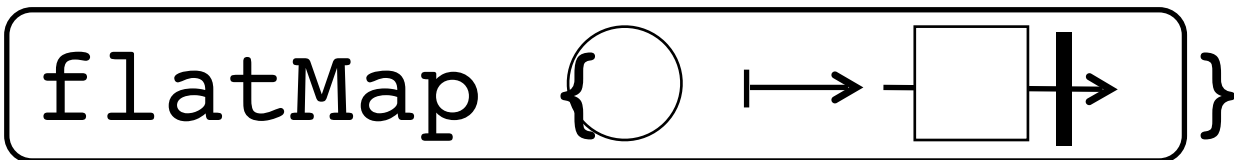
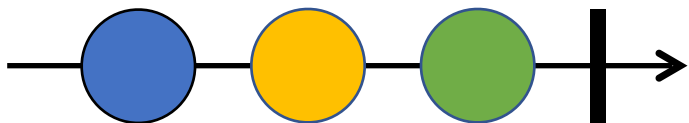
```
Scheduler scheduler = Schedulers  
    .fromExecutorService(executorService);
```

```
Flux.fromIterable(ids)  
    .log()
```

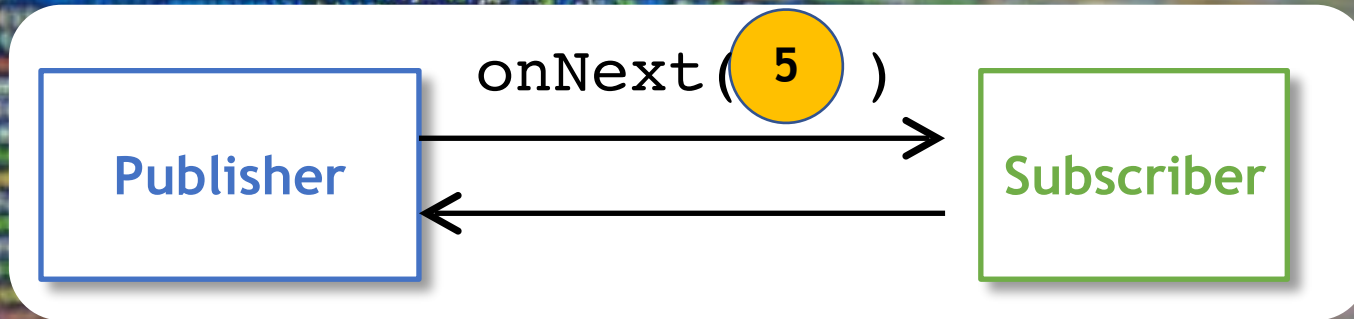
```
    .flatMap(id -> Mono.just(id)  
        .map(somethings::loadById)  
        .subscribeOn(scheduler),
```

```
3)    .doOnNext(handleSomething())  
        .subscribe();
```

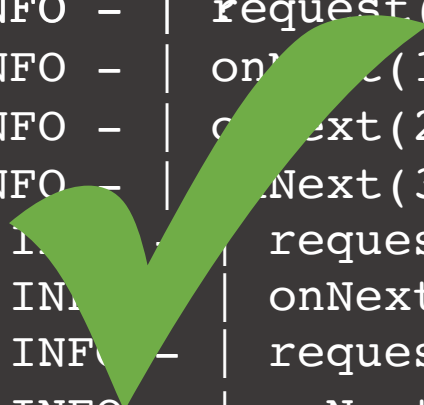
Backpressure



Backpressure



```
[main] INFO - | onSubscribe()
[main] INFO - | request(3)
[main] INFO - | onNext(1)
[main] INFO - | onNext(2)
[main] INFO - | onNext(3)
[pool-1] INFO - | request(1)
[pool-1] INFO - | onNext(4)
[pool-2] INFO - | request(1)
[pool-2] INFO - | onNext(5)
[pool-2] INFO - | onComplete()
```



```
ExecutorService executorService =
    new ThreadPoolExecutor(
        nThreads, nThreads,
        0L, TimeUnit.MILLISECONDS,
        new LinkedBlockingQueue<>(capacity)
    );
```

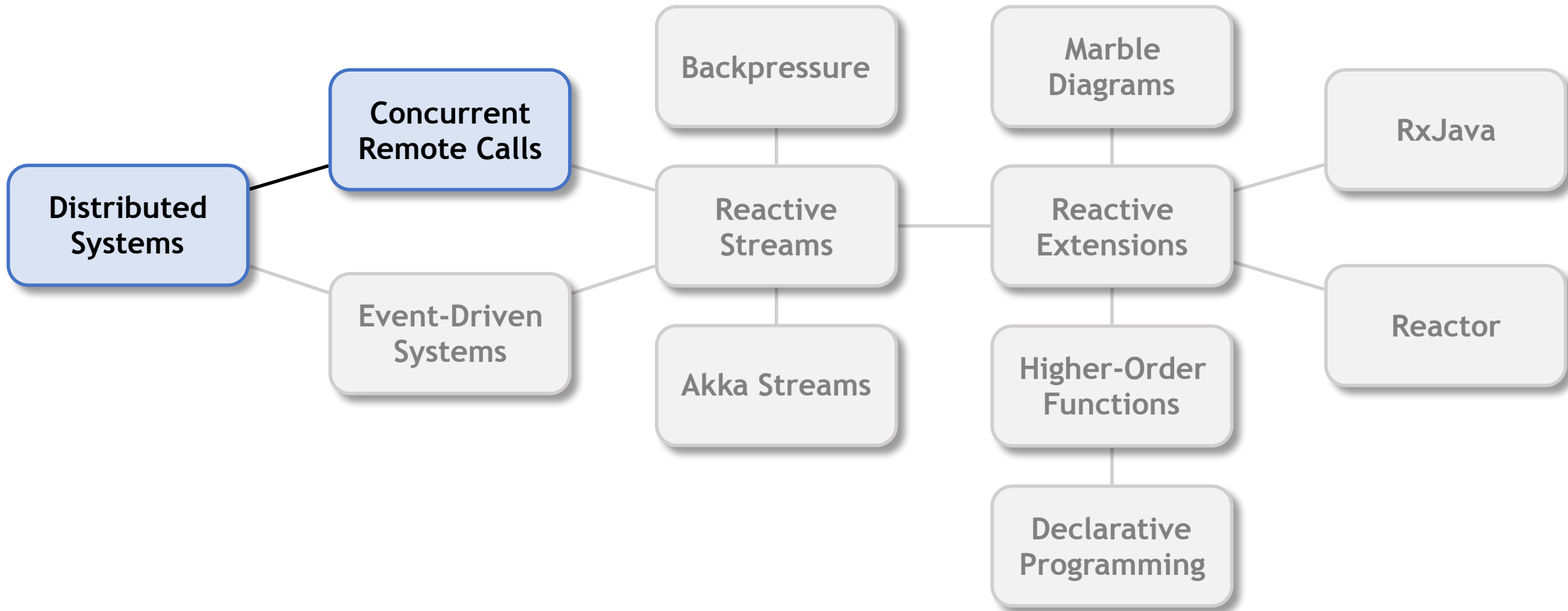
```
Scheduler scheduler = Schedulers
    .fromExecutorService(executorService);
```

```
Flux.fromIterable(ids)
    .log()
```

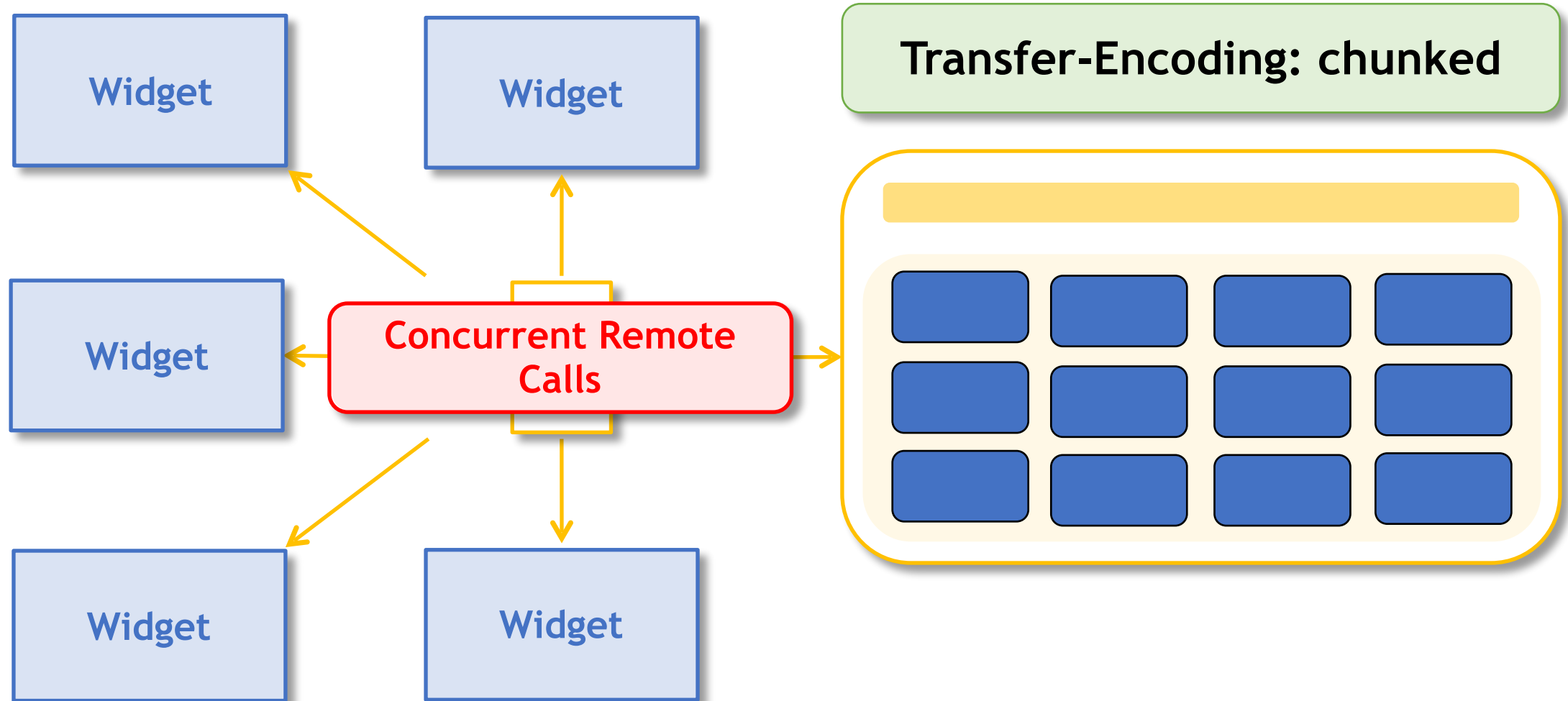
```
.flatMap(id -> Mono.just(id)
    .map(somethings::loadById)
    .subscribeOn(scheduler),
```

```
3)
    .doOnNext(handleSomething())
    .subscribe();
```

The Reactive Landscape



Progressive HTML Rendering



Progressive HTML Rendering

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>

<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-webflux</
artifactId>
</dependency>
```

Progressive HTML Rendering

```
curl -i -N localhost:8080/hello
```

```
HTTP/1.1 200
```

```
Content-Type: text/plain
```

```
Transfer-Encoding: chunked
```

```
Hello
```

```
Hello
```

```
Hello
```

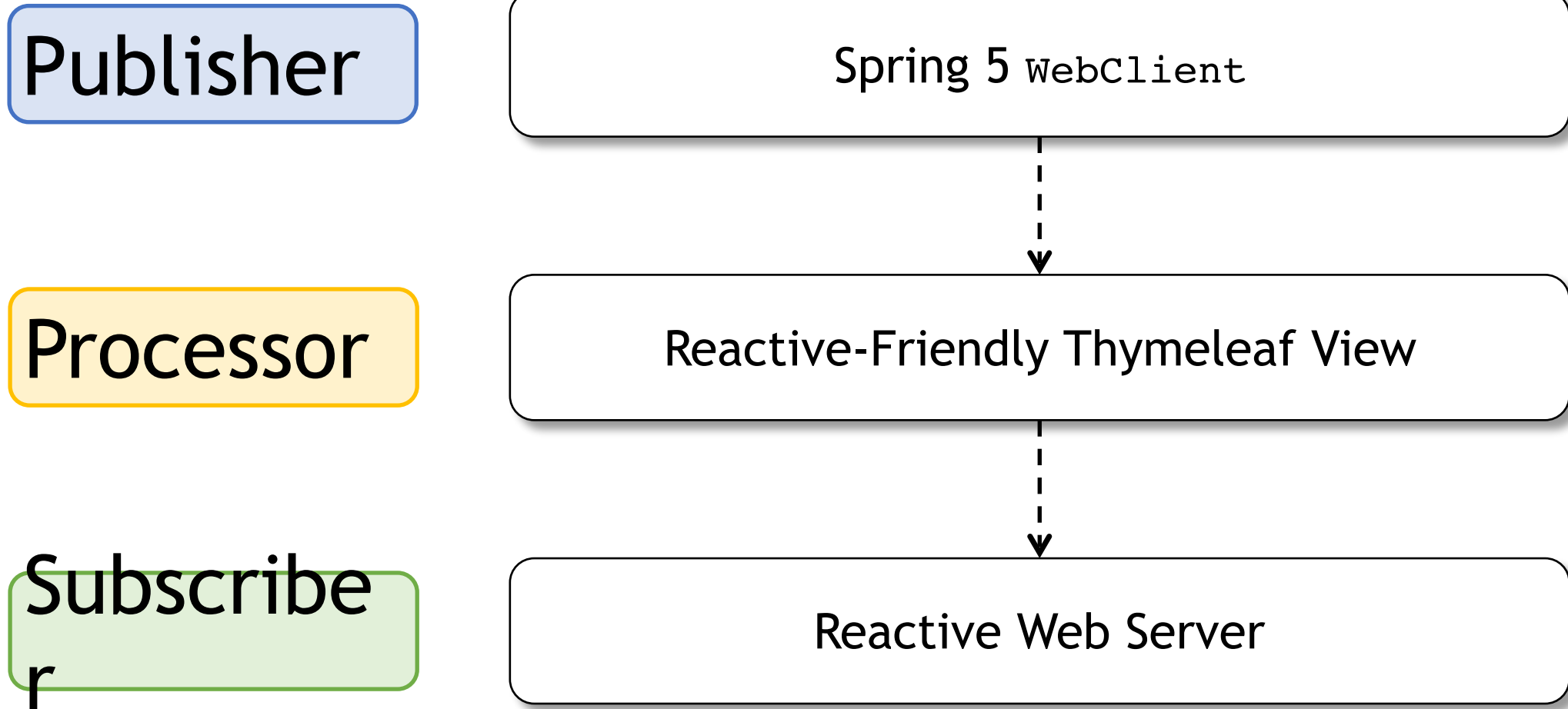
```
Hello
```

```
@SpringBootApplication
@RestController
public class Application {

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }

    @GetMapping(value = "hello")
    public Flux<String> sayHello() {
        return Flux.interval(Duration.ofMillis(500)).map(tick ->
"Hello\n");
    }
}
```

Progressive HTML Rendering



Progressive HTML Rendering

Publisher

Spring 5 WebClient

```
URI uri = ...  
String name = ...  
  
Mono<Pagelet> pageletMono =  
    webClient.get().uri(uri).exchange()  
        .flatMap(res -> res.bodyToMono(String.class))  
        .map(body -> new Pagelet(name, body));  
  
List<Mono<Pagelet>> pageletMonos = ...;  
  
... = Flux.merge(pageletMonos);
```

Progressive HTML Rendering

Processor

Reactive-Friendly Thymeleaf View

```
@GetMapping(value = "/widgets")
public String widgets(Model model) {
    List<String> pageletNames = widgetService.getPageletNames();
    Flux<Pagelet> pagelets = widgetService.loadPagelets();

    model.addAttribute("pageletNames", pageletNames);
    model.addAttribute("pagelets", pagelets);
    ReactiveDataDriverContextVariable<Pagelets> pageletsVar =
        new ReactiveDataDriverContextVariable<Pagelets>(pagelets, 1);

    return "widgets";
}
```

ReactiveDataDriverContextVariable

Progressive HTML Rendering

Processor

Reactive-Friendly Thymeleaf View

```
<article th:each="pagelet: ${pagelets}" class="...">
  <section class="...">
    <div class="...">
      <div th:utext="${pagelet.content}"
class="..."></div>
    </div>
  </section>
</article>
```

```
List<Mono<Pagelet>> pageletMonos = ...;
```

```
... Flux.merge(pageletMonos) ;
```

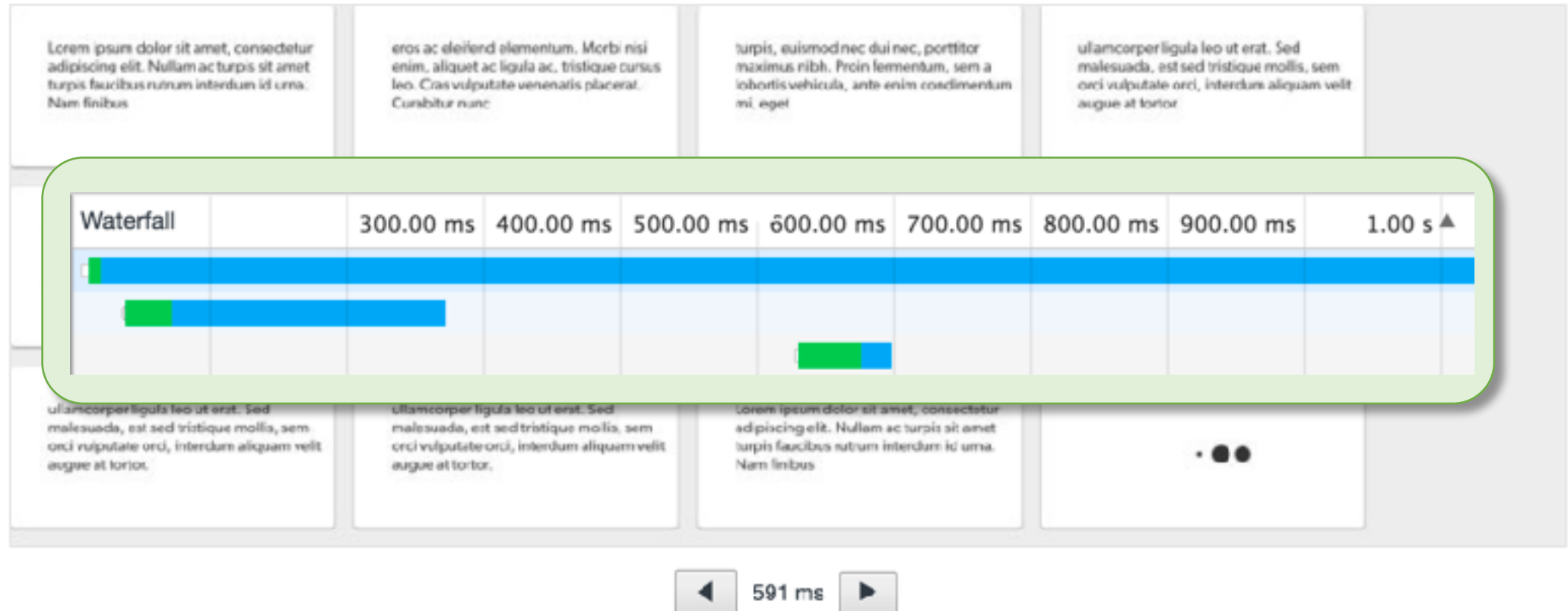
Progressive HTML Rendering

Processor

Reactive-Friendly Thymeleaf View

```
<script><article th:each="pageletName: ${pageletNames}"
class="...">
  <section class="...">
    <div th:id="${pageletName}" class="...">
      <div class="...">...</div>
    </div>
  </section>
</article>
</script>
```

Progressive HTML Rendering



What's more?

- Reactive Systems
- Cold Streams vs. Hot Streams
- Reactive Database Drivers
- Backpressure Strategies
- Threaded vs. Evented Server Model
- WebSockets and Server-Sent Events

Thank you!

Questions?

Please

**Remember to
rate this session**

Thank you!

