

Building Reactive Applications



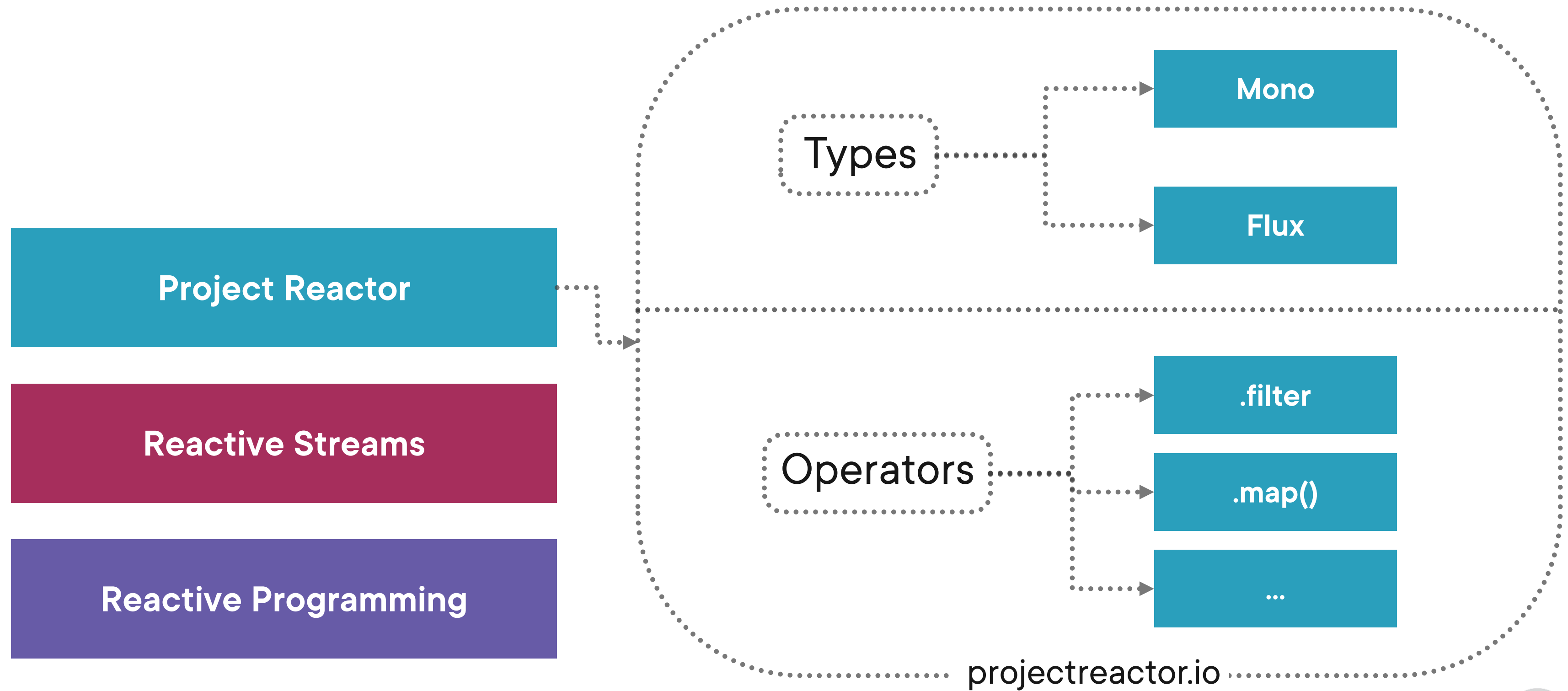
Bogdan Sucaciu

Tech Lead

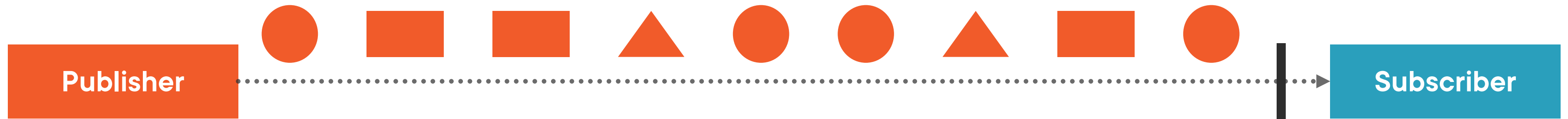
@bsucaciu bsucaciu.com



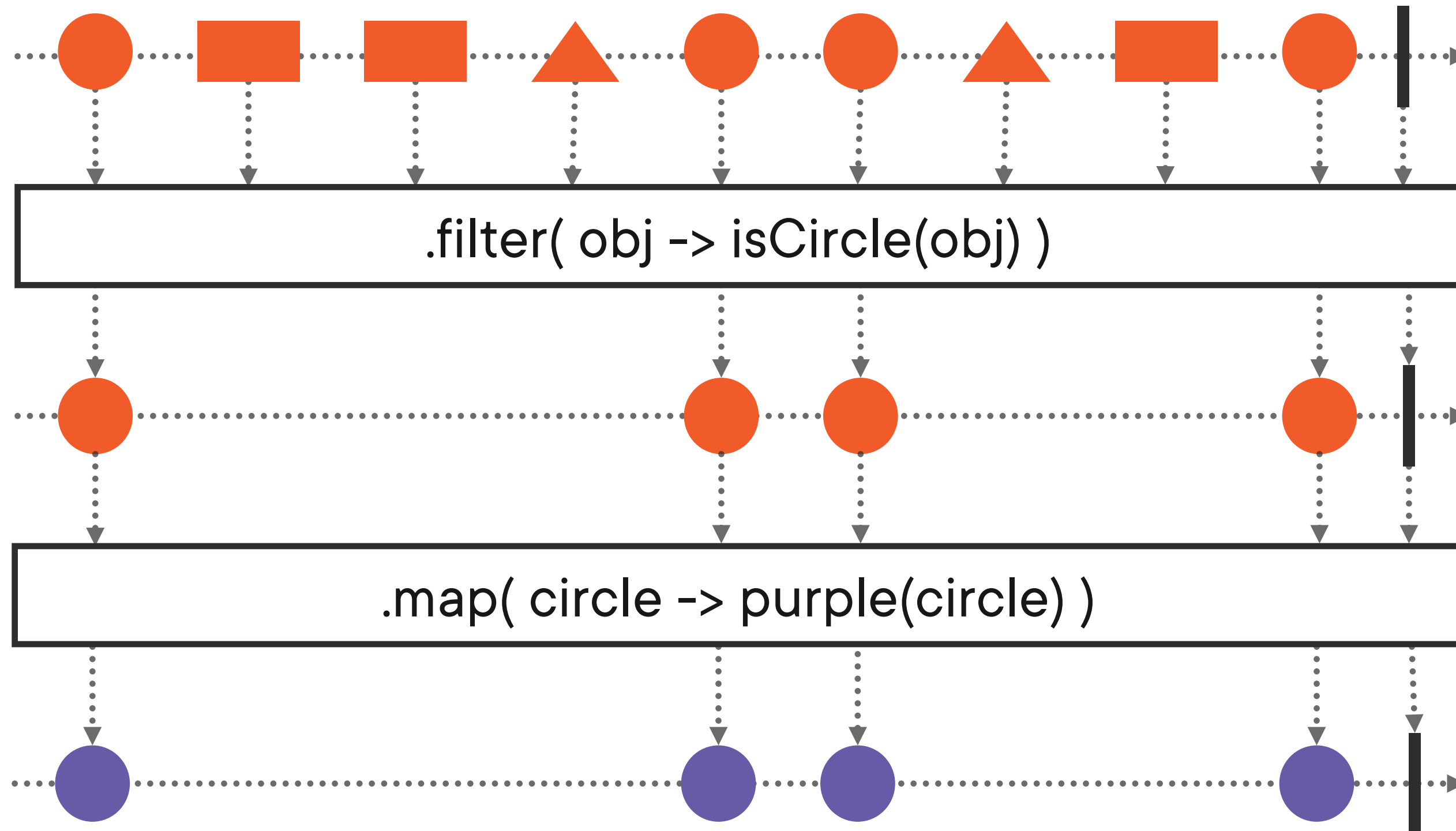
Spring WebFlux



Project Reactor Operators



Project Reactor Operators



Operator Types

Create

Peek

Split

Filter

Error Handling

Sync

Transform

Time Manipulation

Multicast



Operator Types

Create

Mono

`.just()`

`.justOrEmpty()`

`.fromSupplier()`

`.fromCallable()`

`.fromRunnable()`

`.fromFuture()`

`.empty()`

`.error()`

`.never()`

`.defer()`

`.using()`

`.create()`

Flux

`.just()`

`.fromArray()`

`.fromIterable()`

`.fromStream()`

`.range()`

`.generate()`

`.empty()`

`.error()`

`.never()`

`.defer()`

`.using()`

`.create()`



Operator Types

Create

Peek

Split

Filter

Error Handling

Sync

Transform

Time Manipulation

Multicast

bit.ly/project-reactor-operators



Demo



Reactive data access

Map operator

Filter operator



Error Handling



Error Handling

```
dangerousMethod() {
```



```
}
```

```
@ExceptionHandler(SomeException.class)
```



Error Handling

```
try {
```



```
} catch(SomeException ex) {
```



```
} finally {
```



```
}
```

`Mono.error(new SomeException())`

`.onErrorReturn(▲)`

`.onErrorResume(ex -> ●)`

`.onErrorMap(ex -> otherEx)`

`.doFinally(signtype -> doStuff())`



Demo



Global error handling mechanism

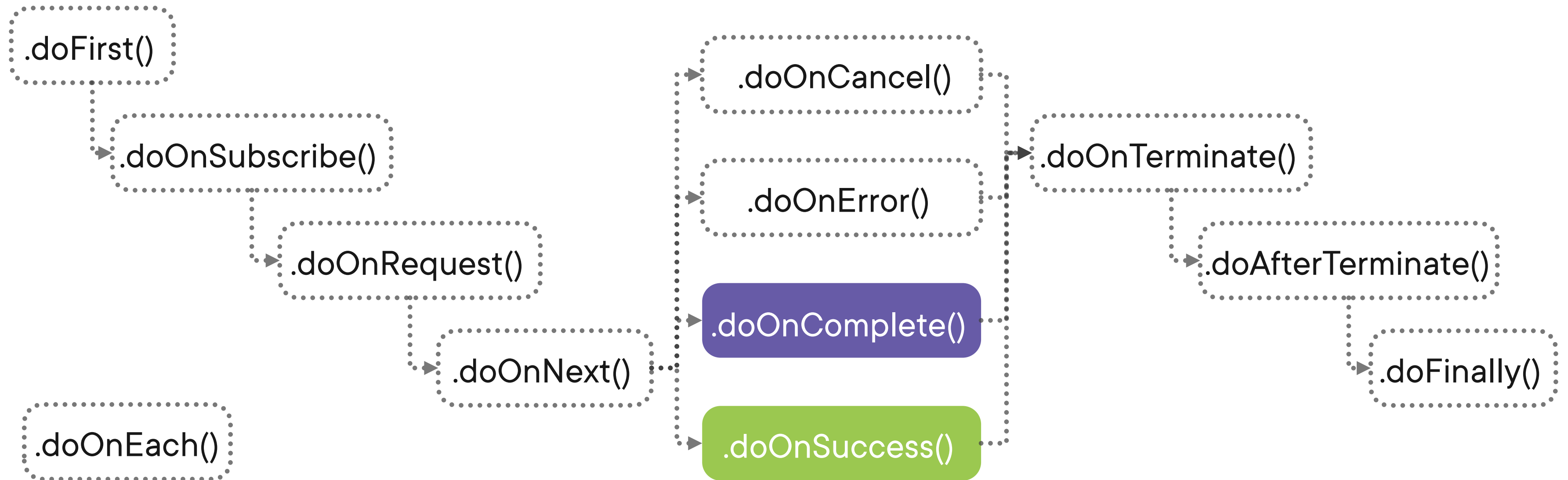
Explore onError operators



Peek Operators



Peek Operators



 - Mono|Flux

 - Mono

 - Flux



Demo



Explore peek operators



Up Next:
Calling REST APIs with WebClient

