Reactive Programming in Java 12 with RxJava 2

RXJAVA CORE CONCEPTS

Russell Elledge
ENTERPRISE ARCHITECT

@mc2ftw mastercraftcoding.com

RxJava Core Concepts

Reactive Manifesto

Observable Events and Cardinality

Working with Observables

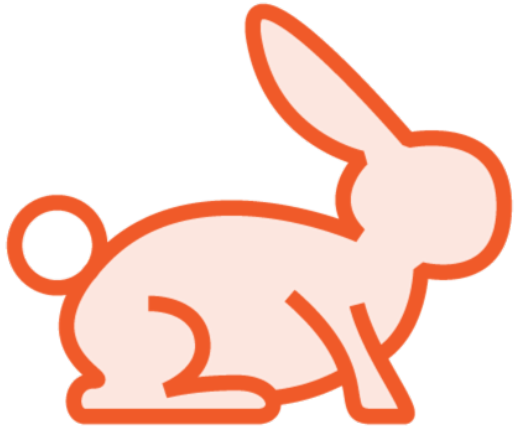
Observable Types and Backpressure

Attributes of Reactive Applications

Event Driven

Scalable

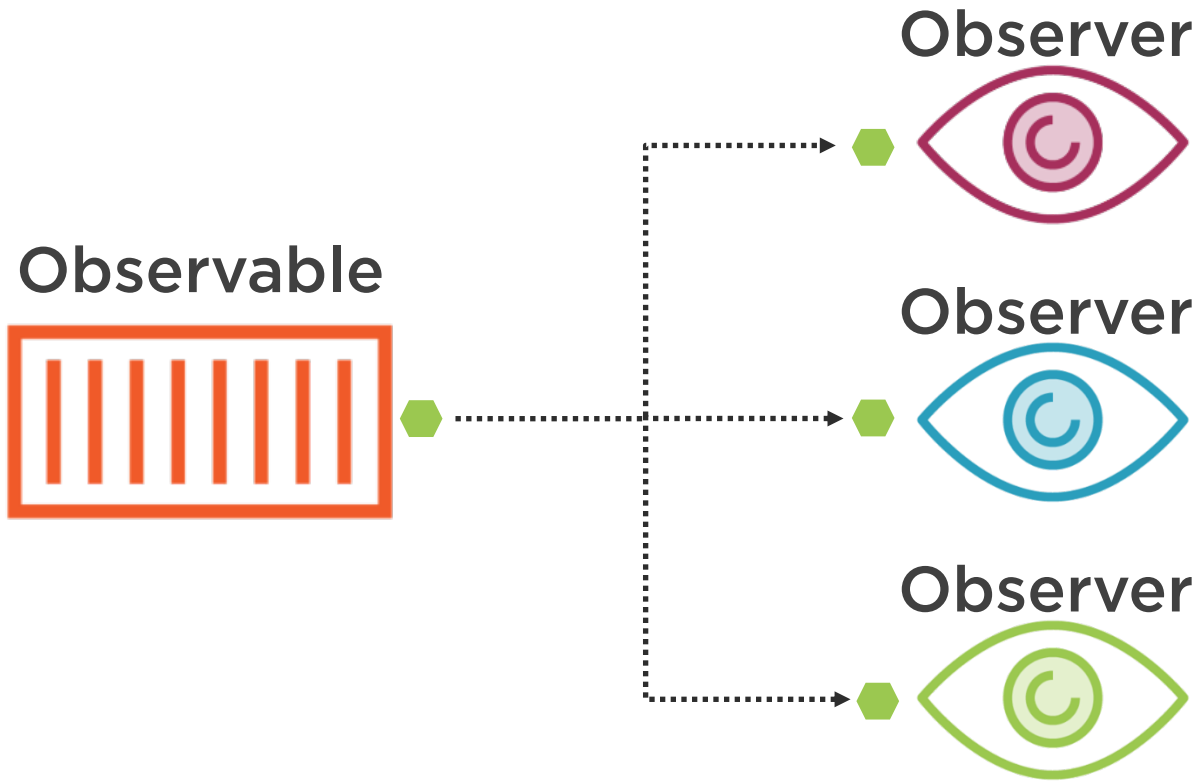
Resilient



Responsive

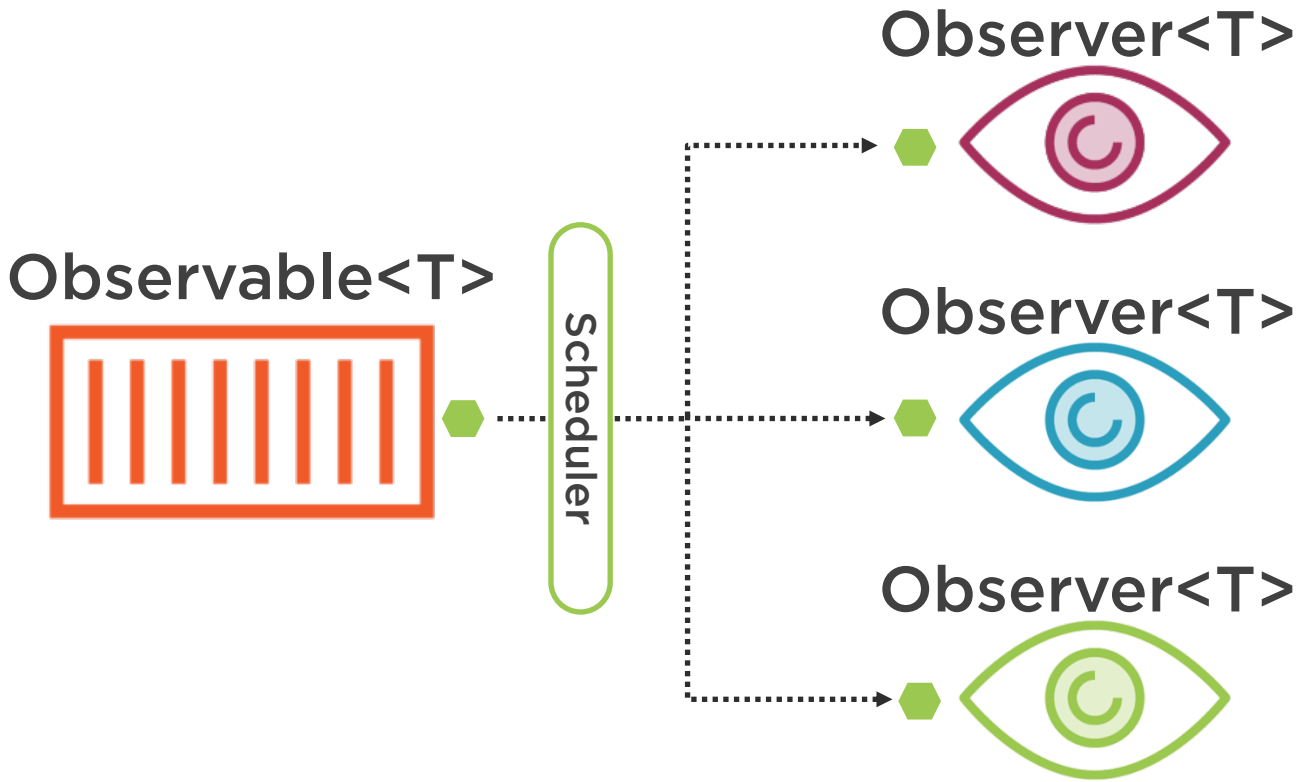
Event Driven

Observer Pattern



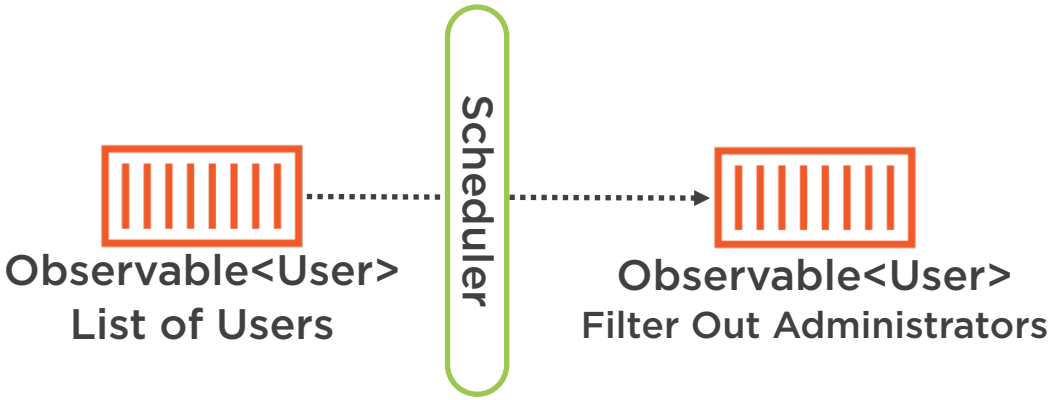
Event Driven

RxJava Observables



Event Driven

Composing Observables

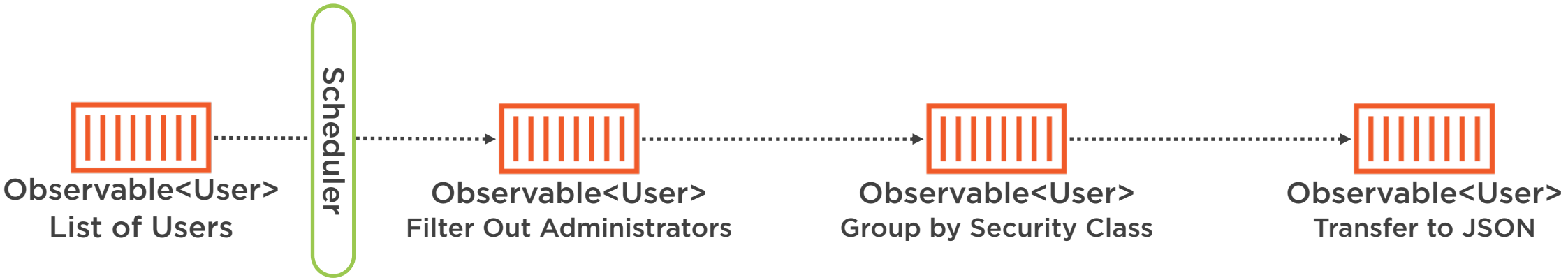


RxJava is single threaded by default.

Even when you specify a scheduler, in most cases RxJava will still only work on a single thread of the designated scheduler.

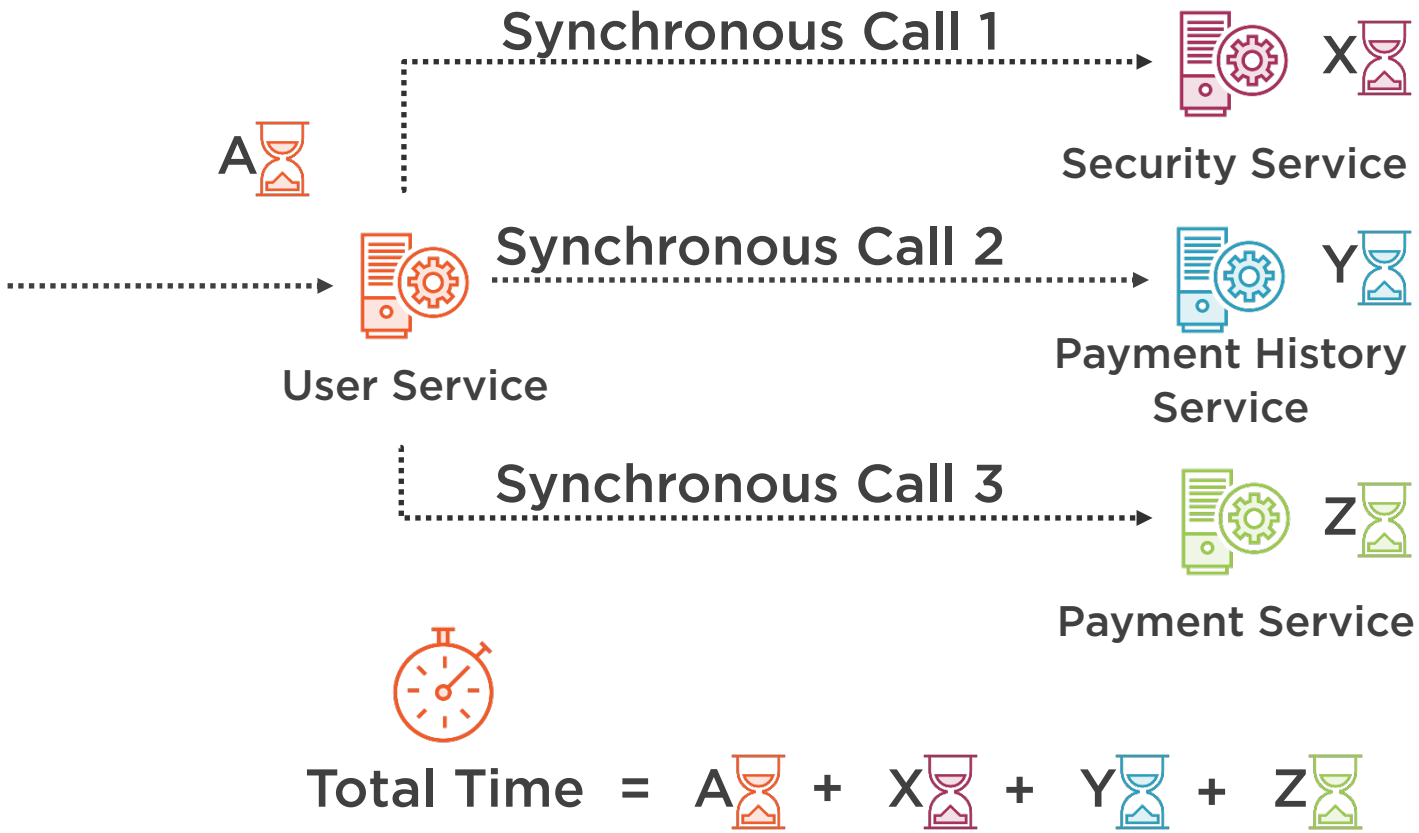
Event Driven

Composing Observables



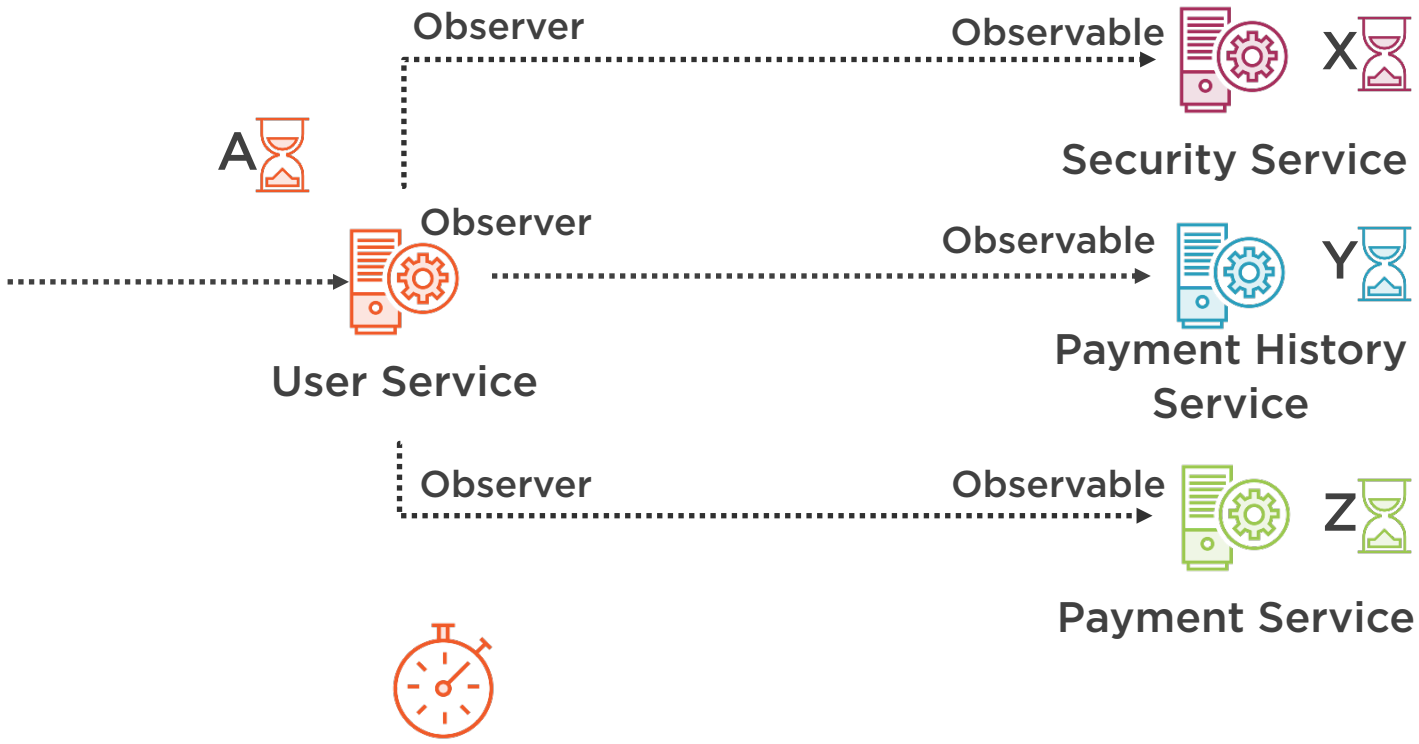
Scalable

Improved Performance with Concurrency



Scalable

Improved Performance with Concurrency



Total Time = A + MAX(X , Y , Z)

Resilient

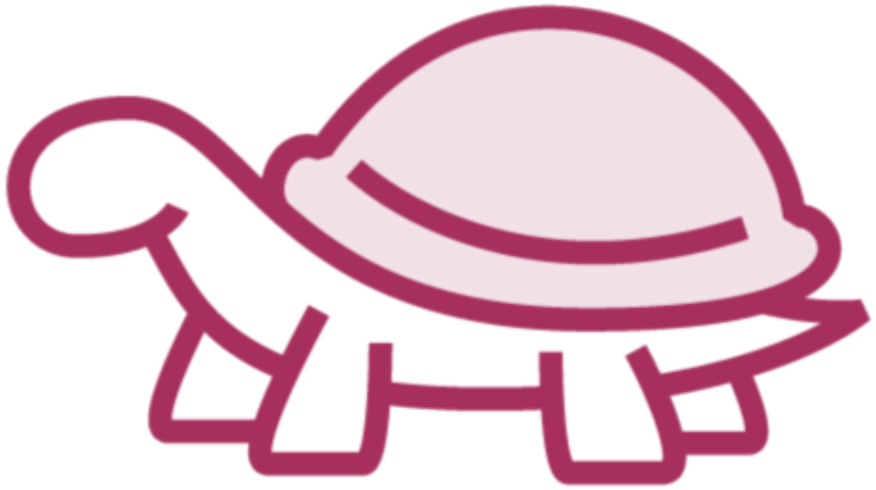
Graceful Error Handling

- Handle more than just the “happy” path.
- RxJava onError Handlers

Manage Failure

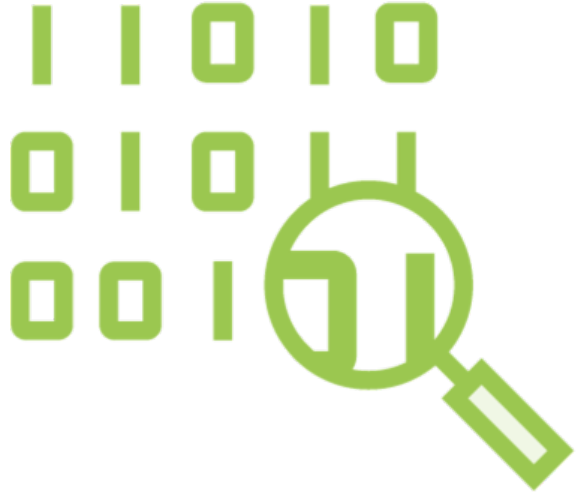
- “Bulkhead” components that could fail
- Purpose-built fallback code

Responsive



Slow Responses == Failure

- Metrics to actively monitor performance
- Define desired levels of performance



Observable Data Models

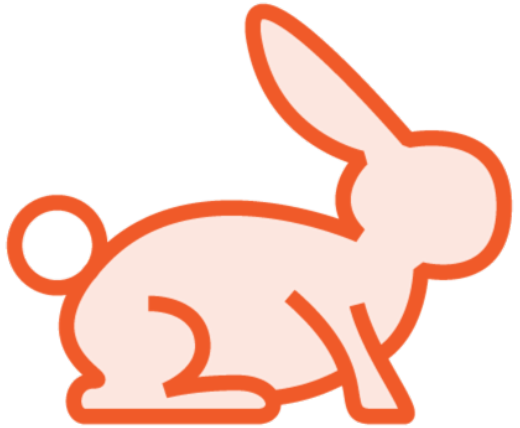
- Observable data models (MVVM)
- Stream of events alters model, UI updates automatically

Attributes of Reactive Applications

Event Driven

Scalable

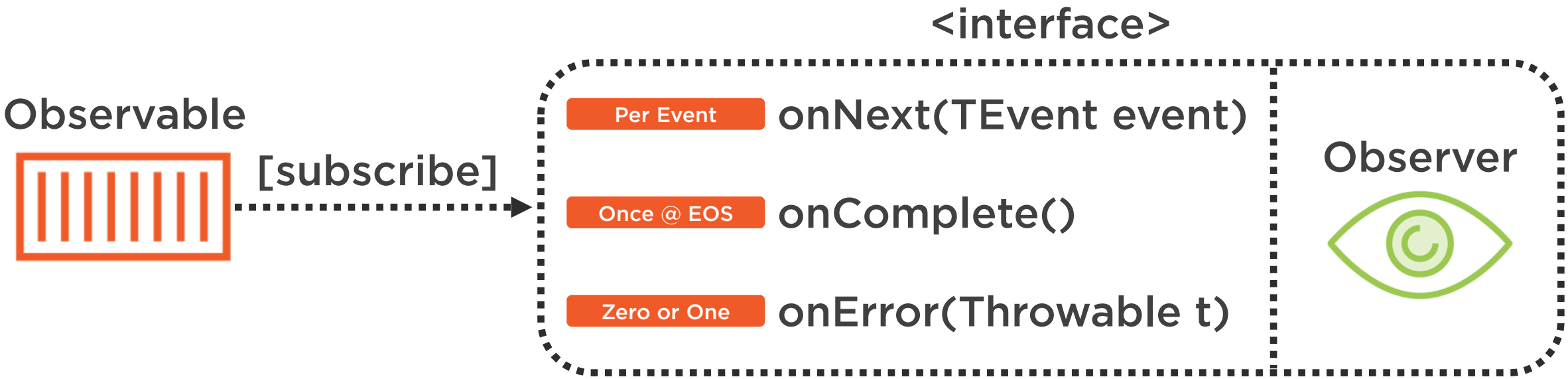
Resilient



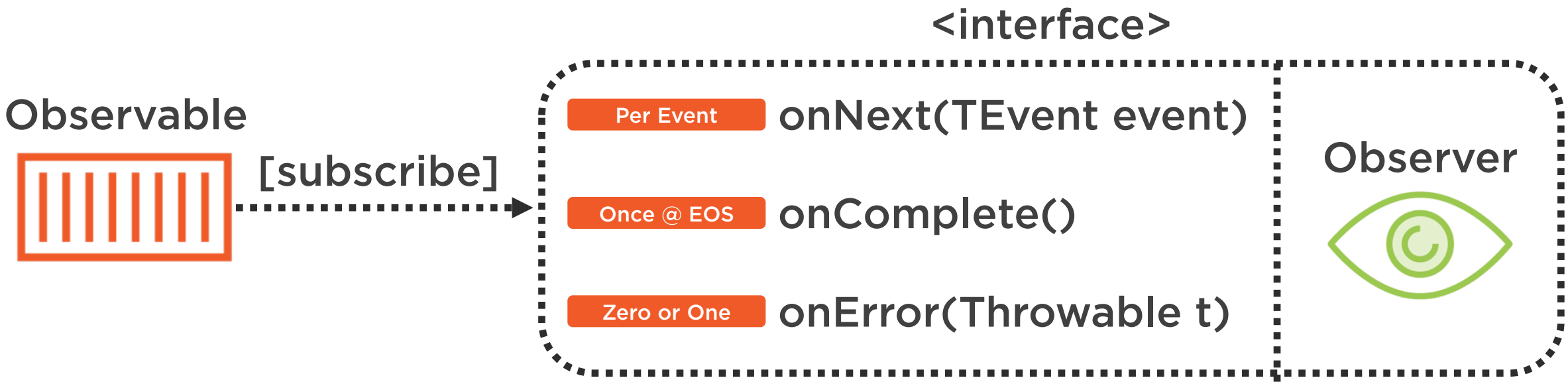
Responsive

Observable Events and Cardinality

Observable Events

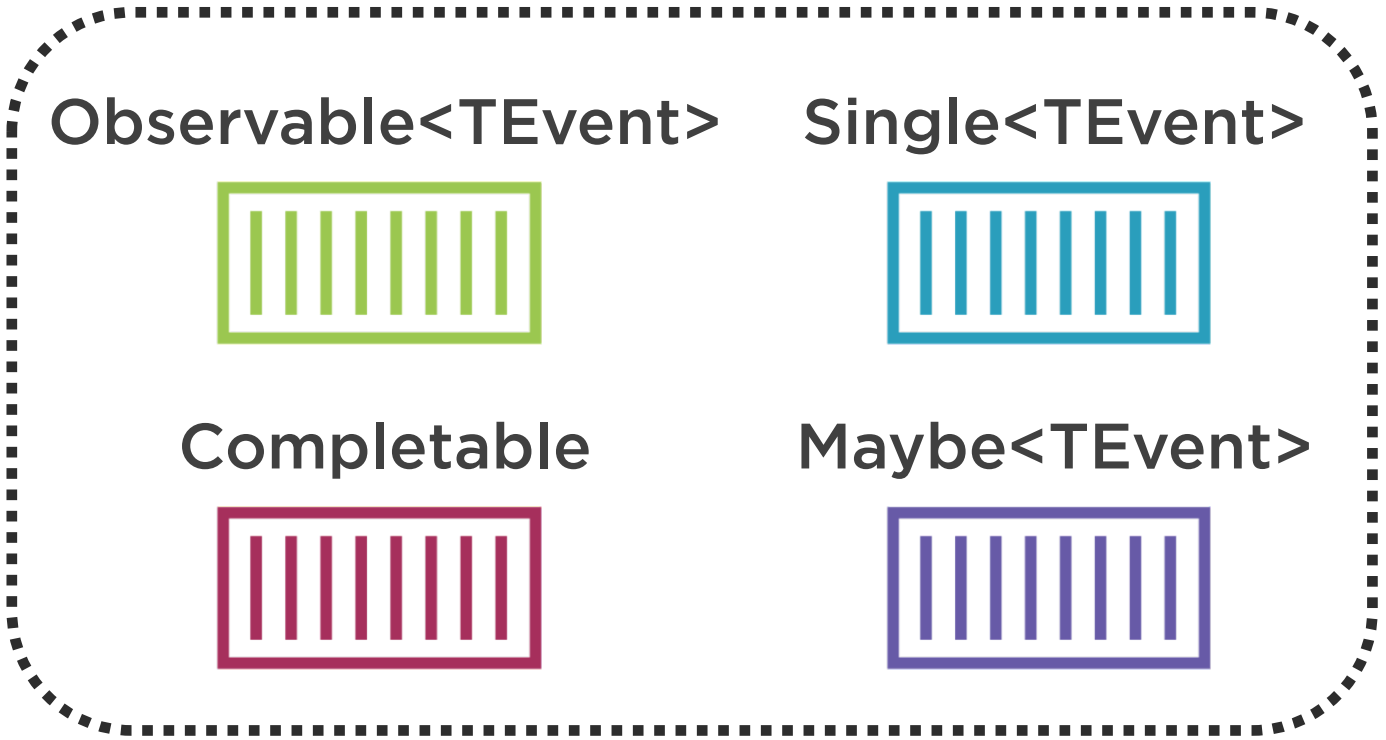


Observable Events



Observable Cardinality

Observable



Demo

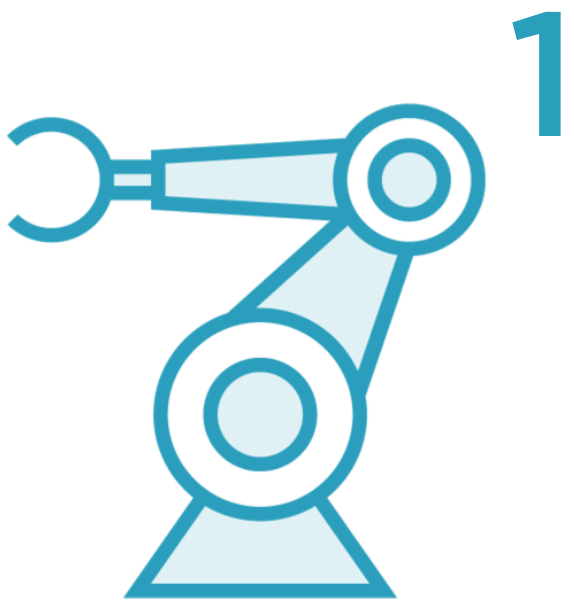
Simple Observable Creation

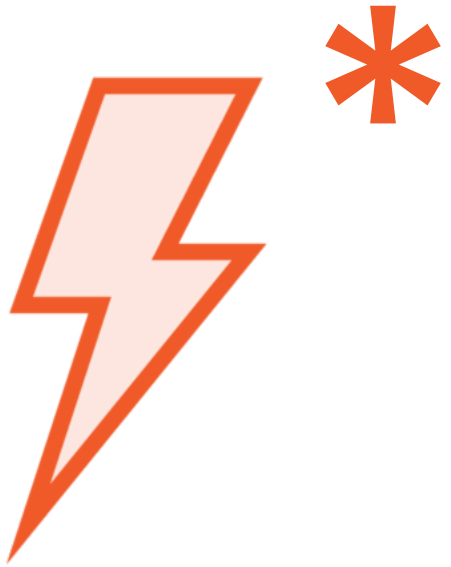
Observable Events

Observable Cardinality

Lifecycle and Error Handling

Observable Lifecycle





Assembly

Setup Observable

Subscription

“subscribe” to Observable

Generation

Active emission of events

Error Handling

Observable.onError* Methods

OnErrorResumeNext

OnErrorReturn

OnErrorReturnItem

Unsubscribing

Stop Event Flow

Resource Disposal

Idempotent and Thread Safe

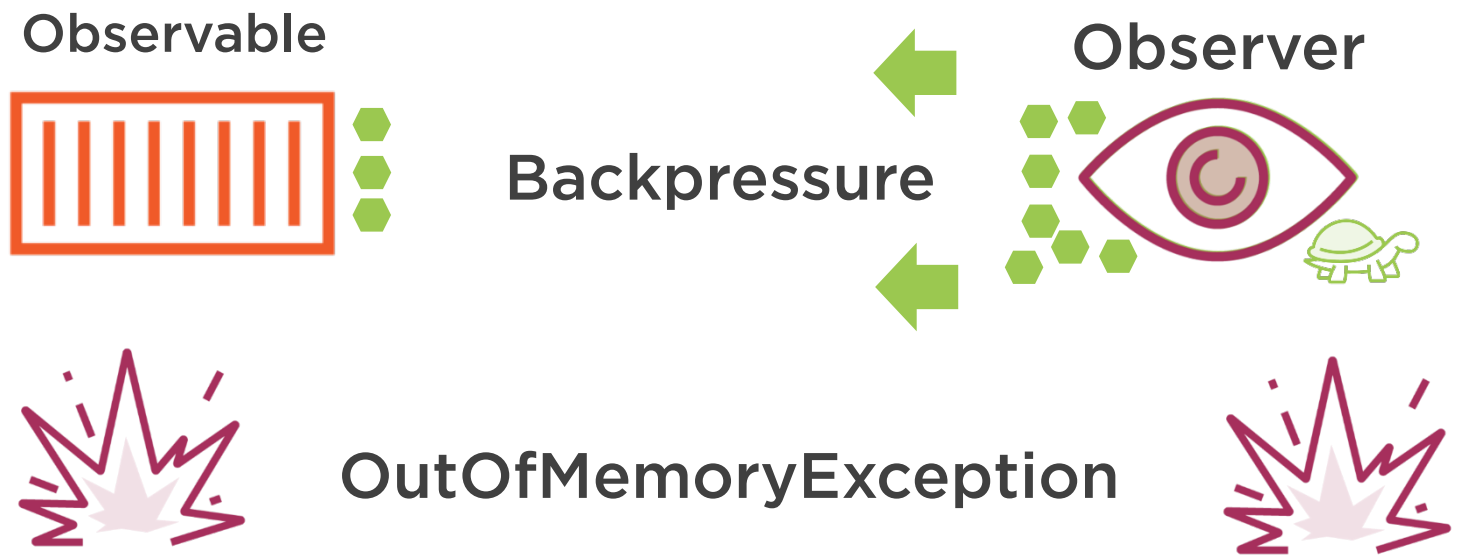
Demo

Observable Lifecycle

Error Handling

Unsubscribing

Backpressure and Flowables


Observable
Backpressure
Observer
OutOfMemoryException

Flowable

Observable

Unaware of Backpressure

OutOfMemoryException

Flowable

Backpressure Aware

Extra “Request” Requirement for FlowableSubscribers

Demo

Backpressure

Flowable