

Learning Reactive Programming with Reactor



Esteban Herrera

Author | Developer | Consultant

@eh3rrera eherrera.net

Overview



Reactive Streams

- Interfaces

Project Reactor

- Flux
- Mono

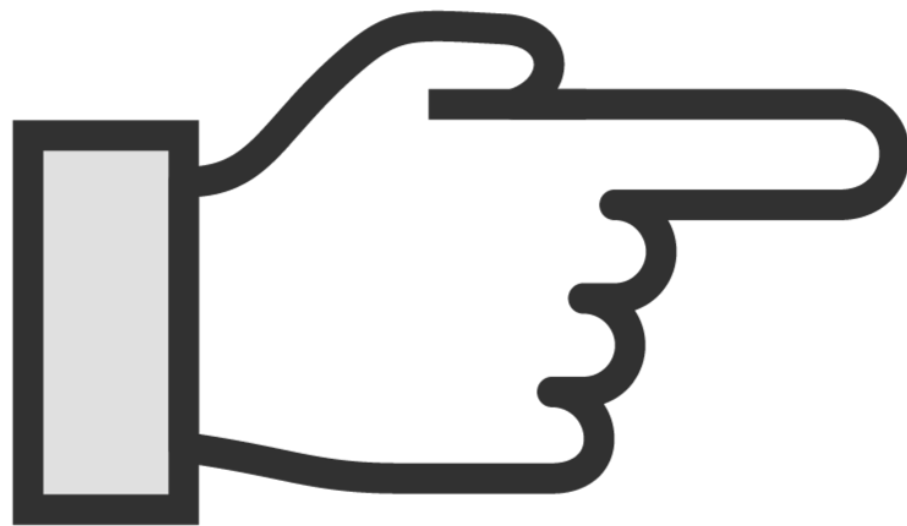
Setting up the demo

Demo: Flux and Mono

Demo: Operators

Reactive Streams

Reactive Streams



Contract

- Asynchronous
- Non-blocking
- Backpressure

Standard behavior

Only concerned with the stream of data

- No transformation (operators)

Reactive Streams



API



**Technology
Compatibility Kit
(TCK)**

Core Interfaces

Publisher

Subscriber

Subscription

Processor

Publisher

```
public interface Publisher<T> {  
    public void subscribe(Subscriber<? super T> s);  
}
```

Subscriber

```
public interface Subscriber<T> {  
    public void onSubscribe(Subscription s);  
    public void onNext(T t);  
    public void onError(Throwable t);  
    public void onComplete();  
}
```

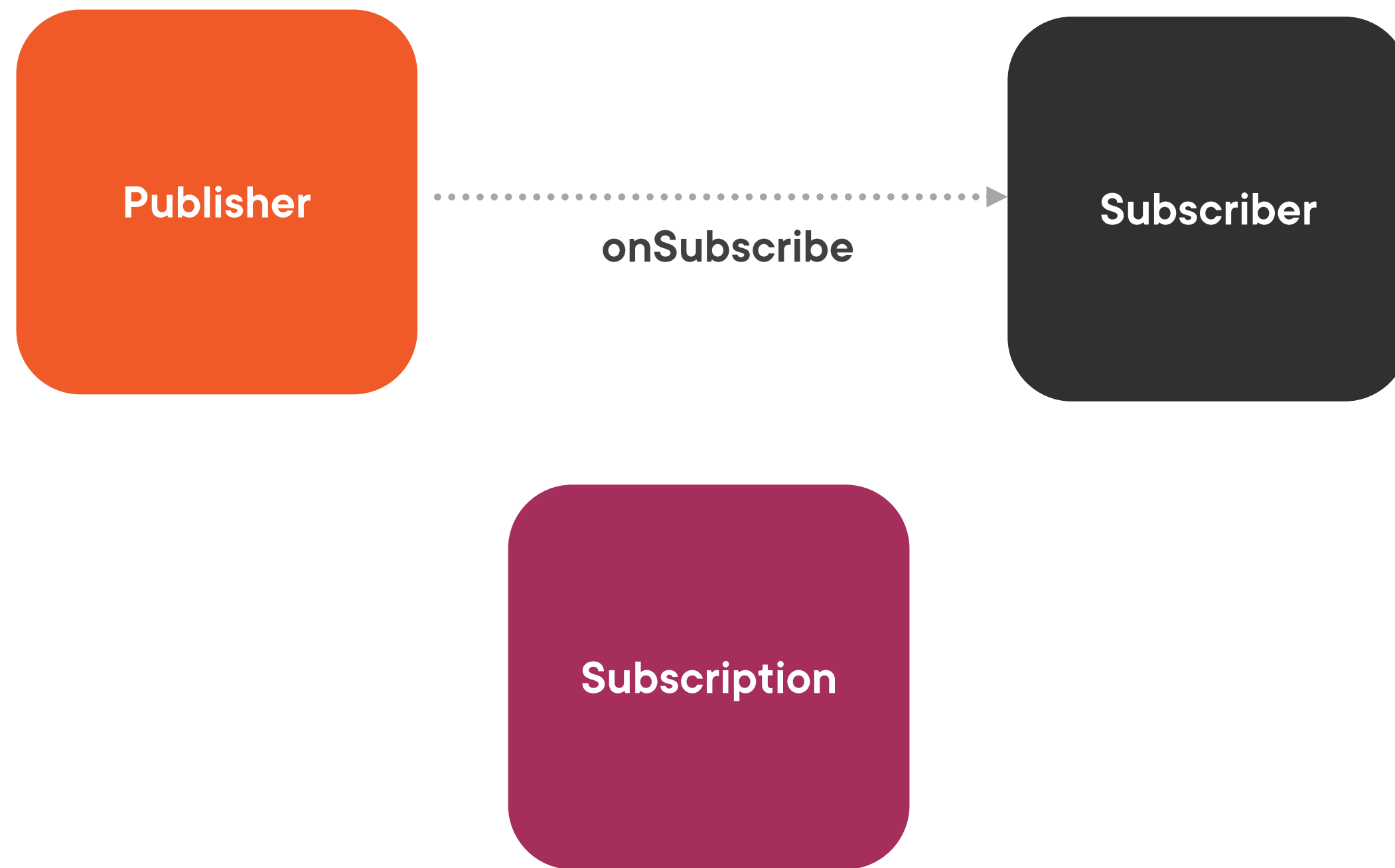

Subscription

```
public interface Subscription<T> {  
    public void request(long n);  
    public void cancel();  
}
```

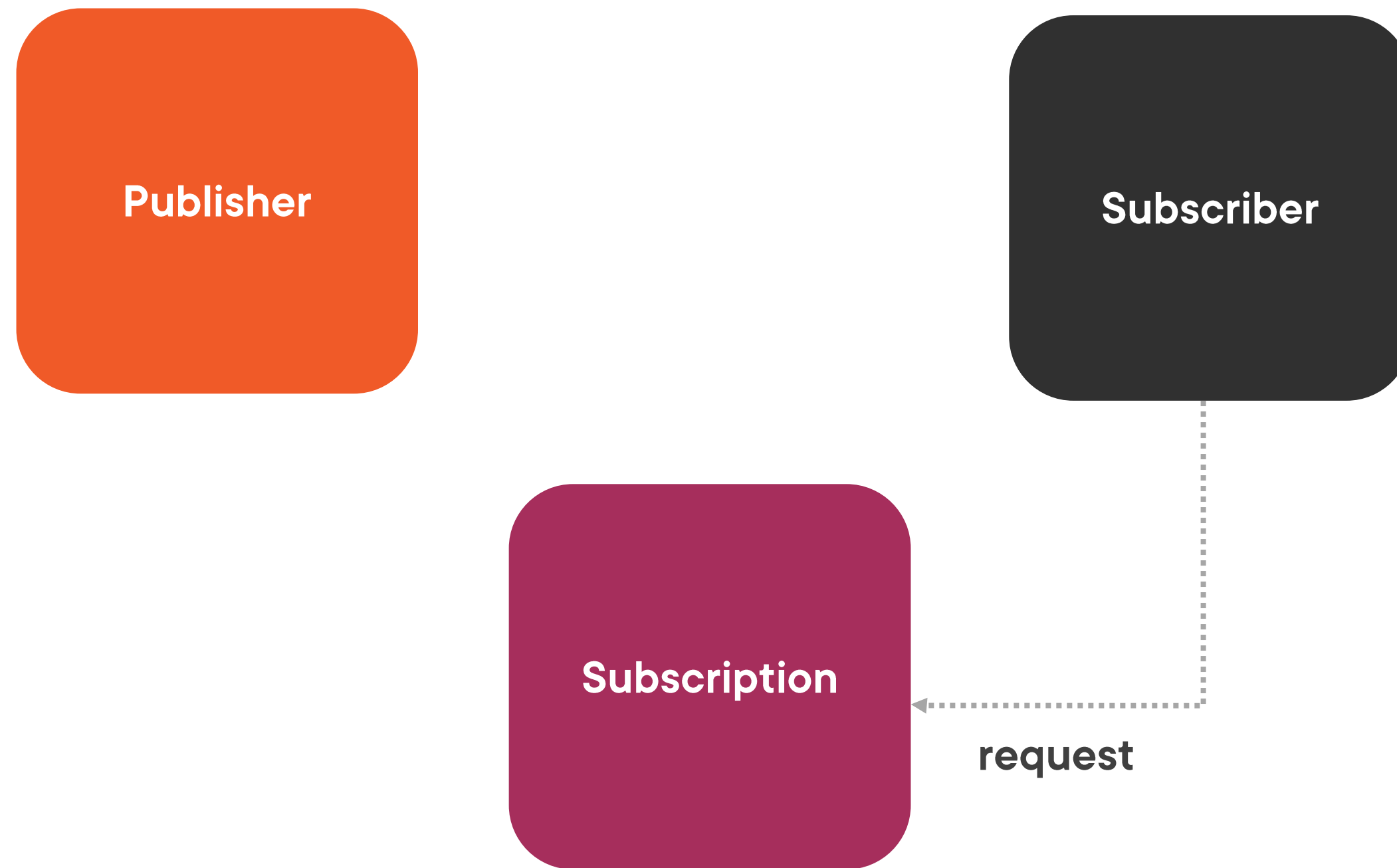
Data Flow



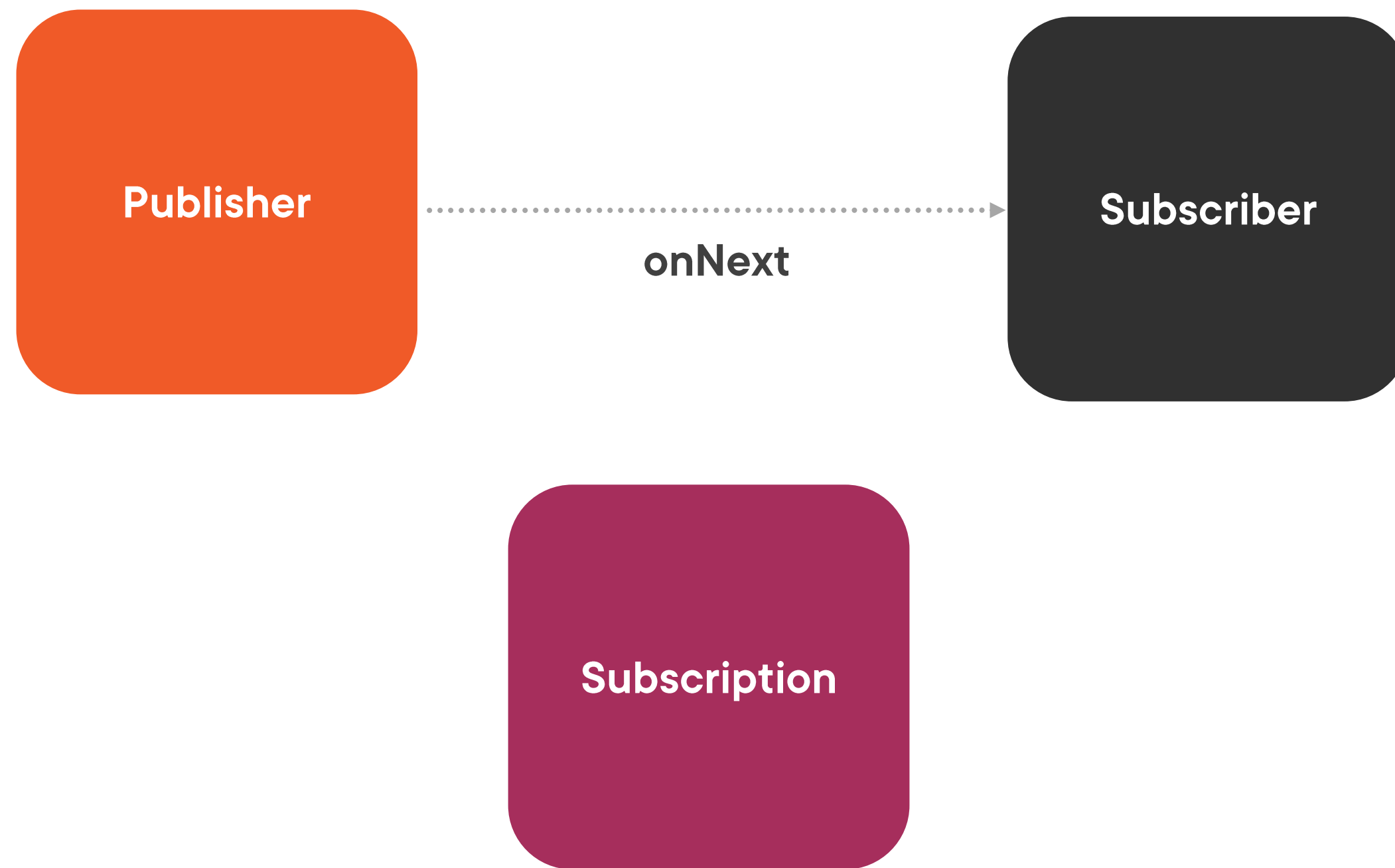
Data Flow



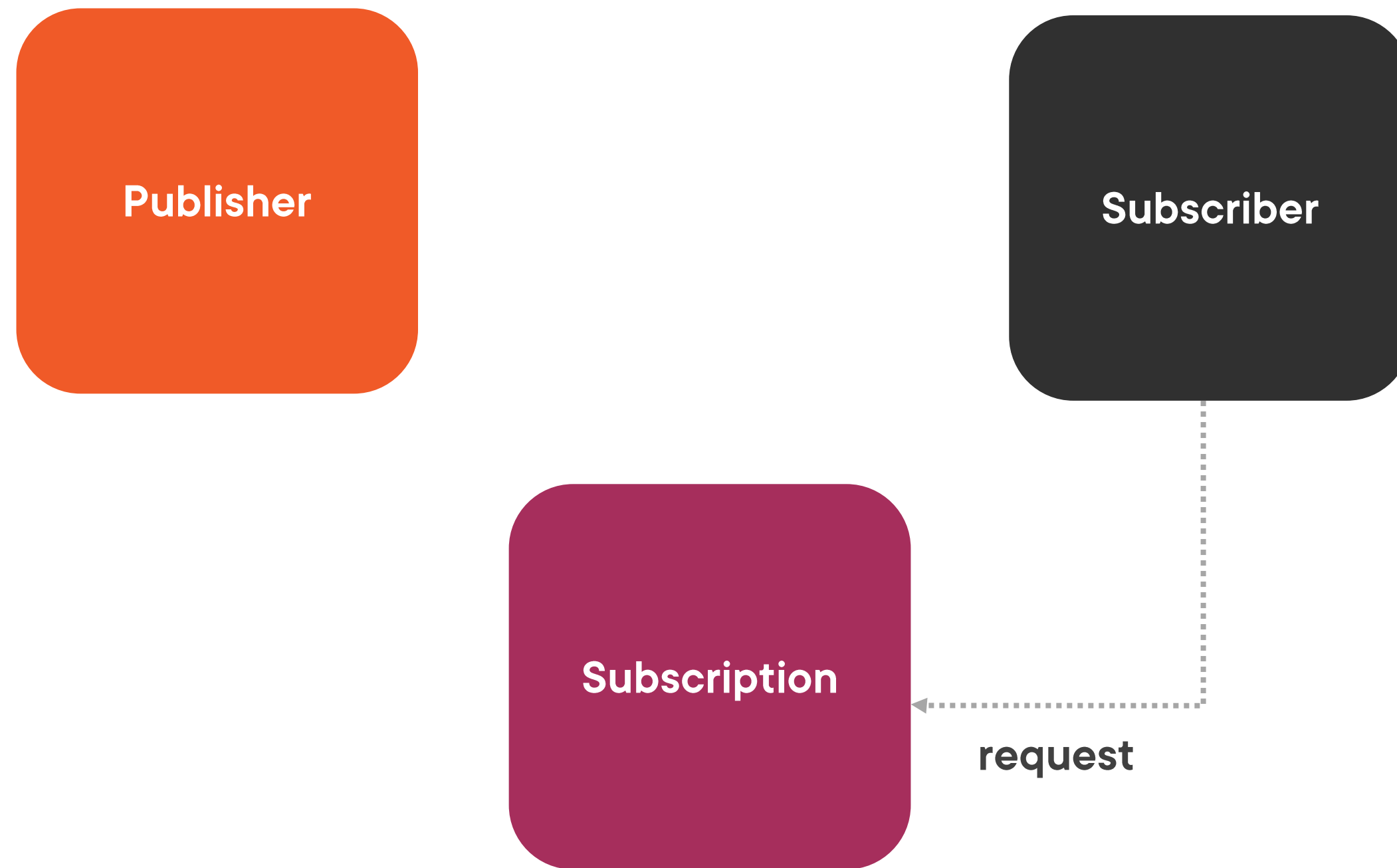
Data Flow



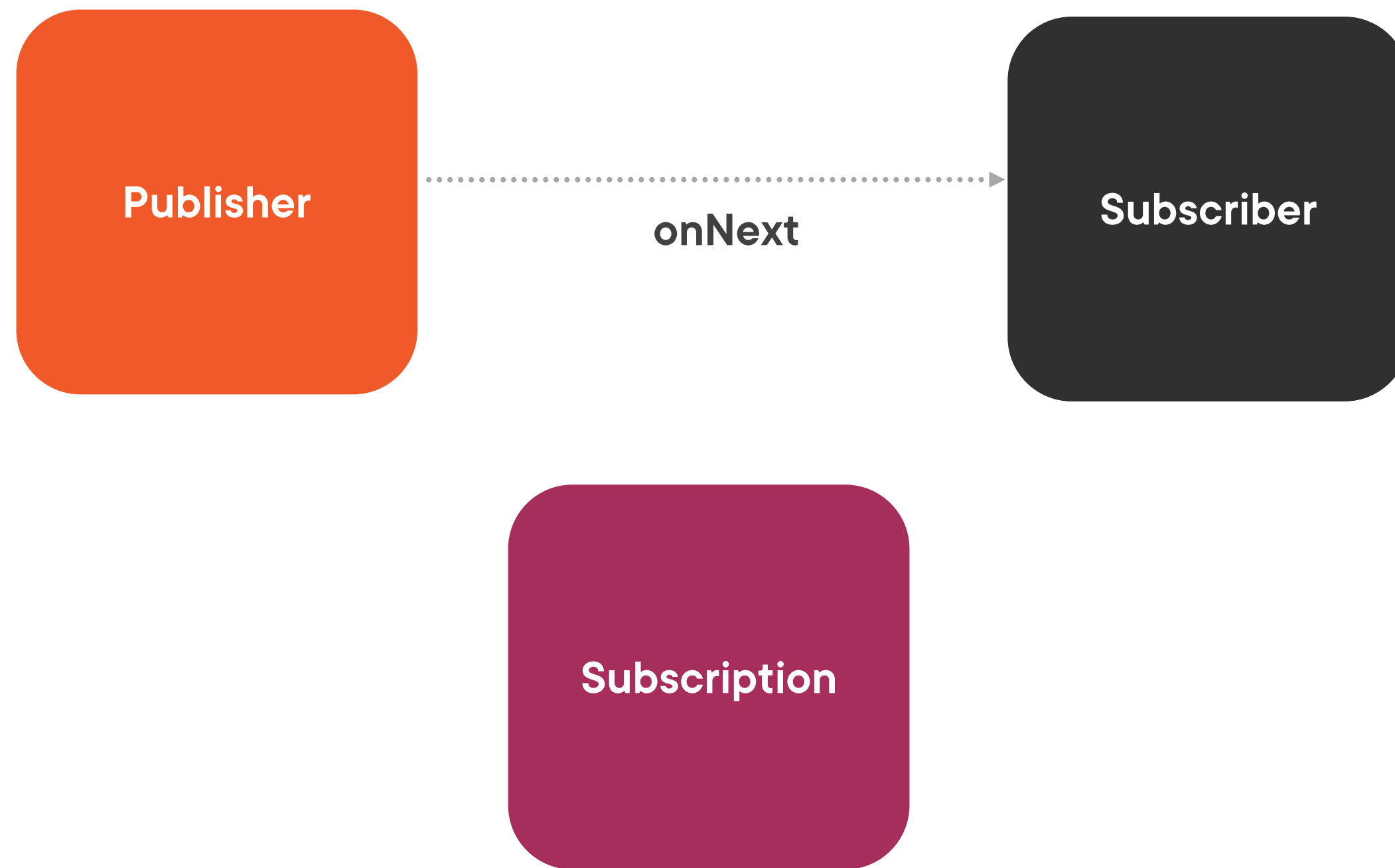
Data Flow



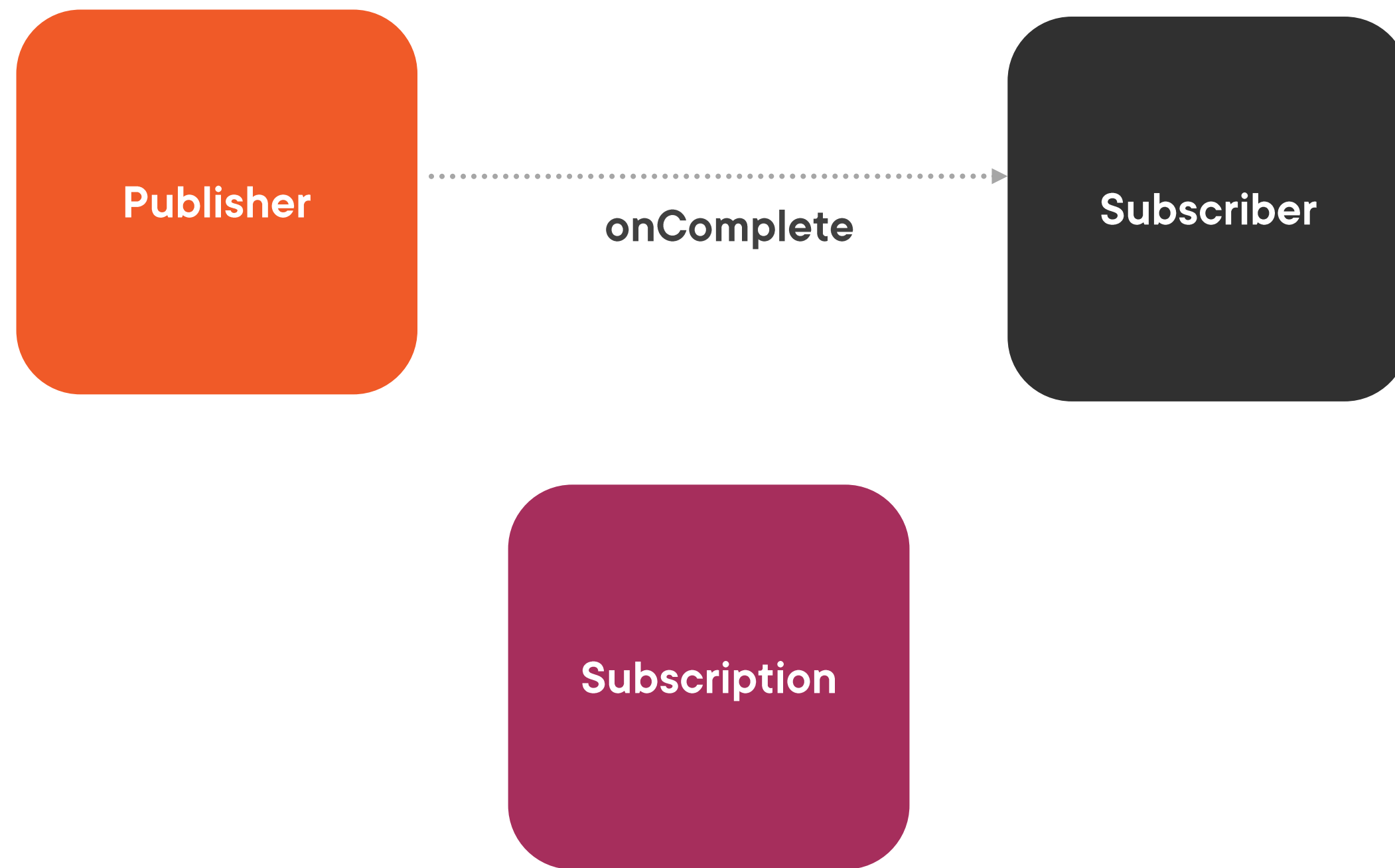
Data Flow



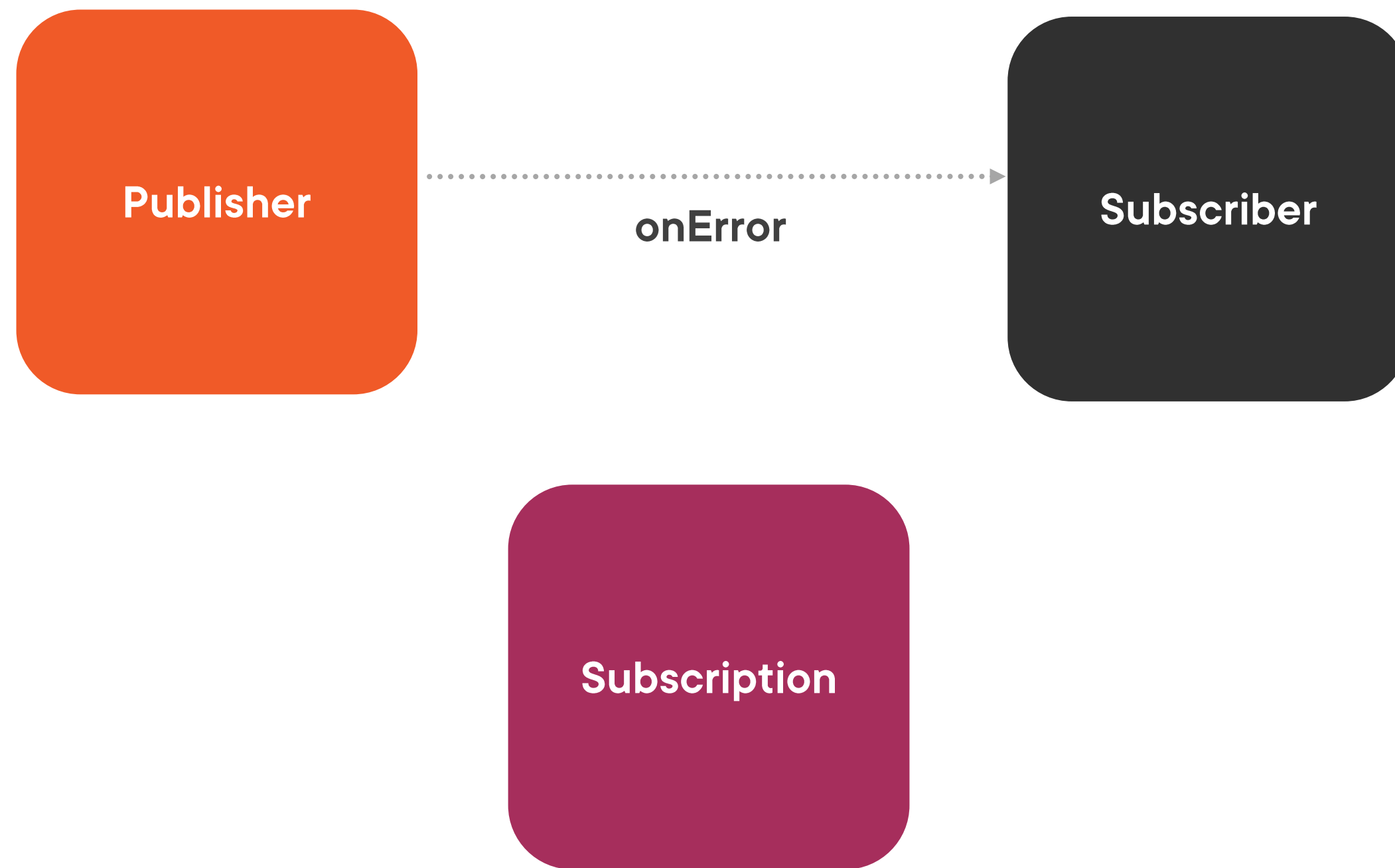
Data Flow



Data Flow



Data Flow



Project Reactor

Reactor Publishers

[0, 1]

Mono

[0 ... n]

Flux

Reactor Publishers

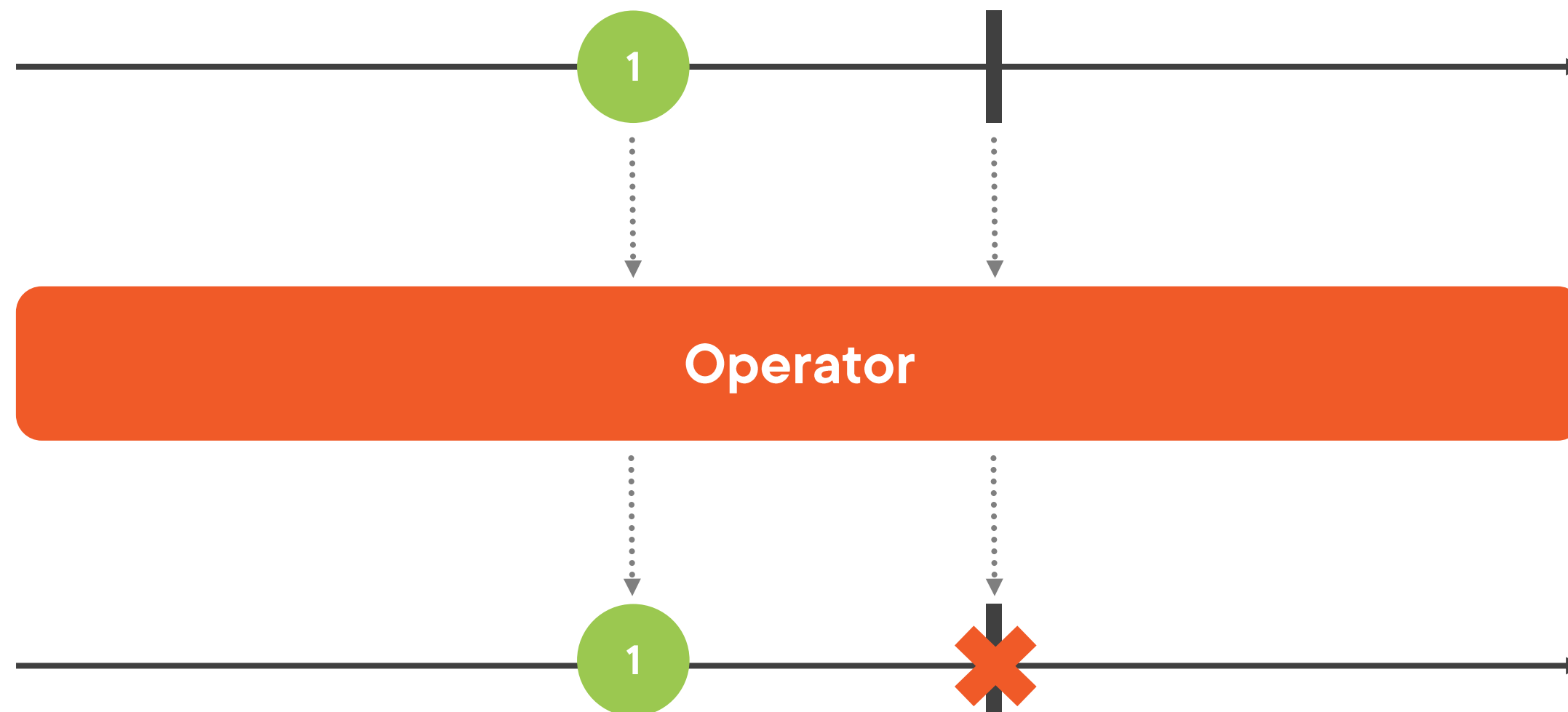


Mono is like the Optional type

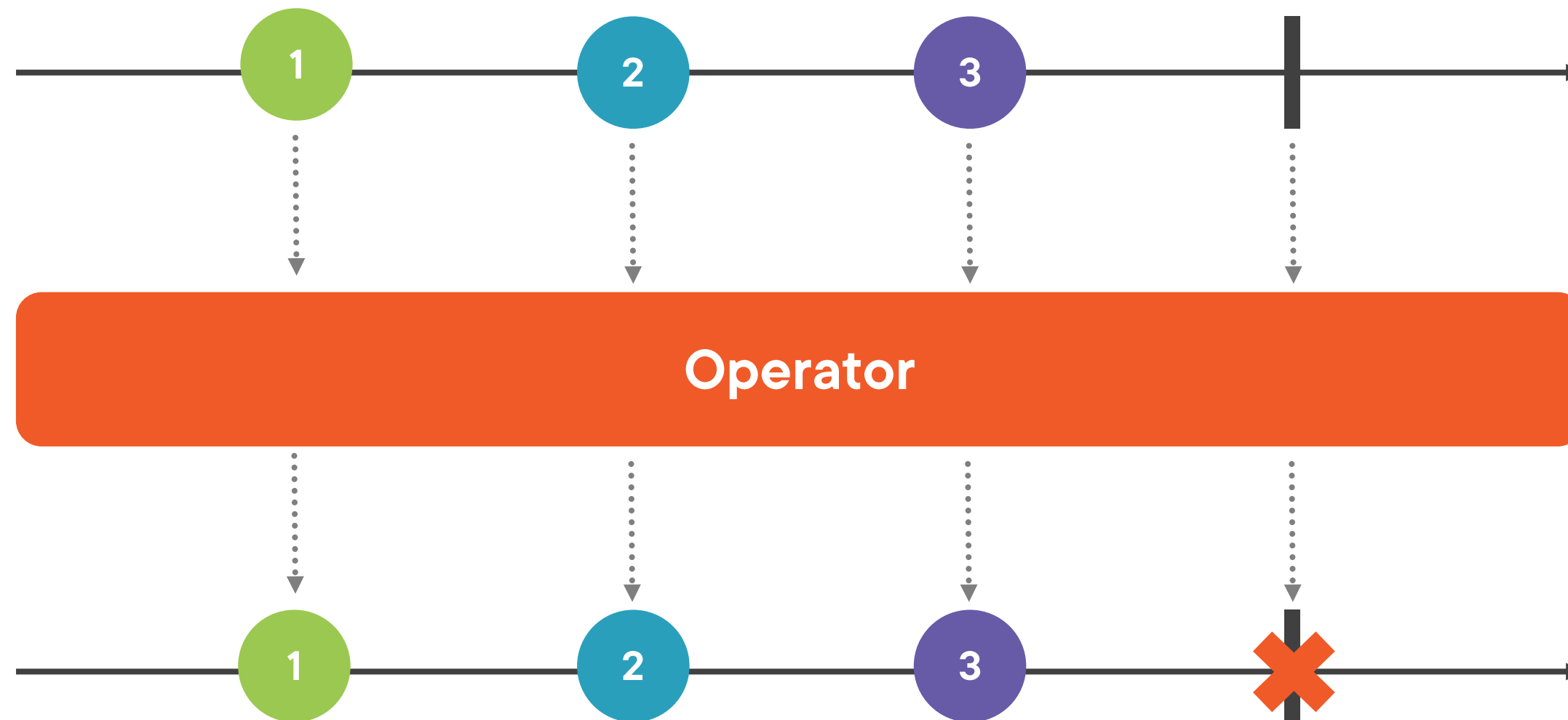
What to return?

- **Flux for lists**
- **Mono for single objects or void**

Mono



Flux



The Subscribe Method

```
subscribe()
```

```
subscribe(Consumer<? super T> consumer)
```

```
subscribe(Consumer<? super T> consumer,  
          Consumer<? super Throwable> errorConsumer)
```

```
subscribe(Consumer<? super T> consumer,  
          Consumer<? super Throwable> errorConsumer,  
          Runnable completeConsumer)
```

Demo



Setting up the project

Demo



Mono

- **Creation**
- **Subscription**
- **Exceptions**

Demo



Flux

- **Creation**
- **Subscription**
- **Request**

Demo



Operators

- **map**
- **flatMap**
- **concat and merge**
- **zip**

Things to Remember



Reactive Streams

- **Publisher**
- **Subscriber**
- **Subscription**
- **Processor**

Project Reactor

- **Flux**
- **Mono**
- **Subscription**
- **Operators**