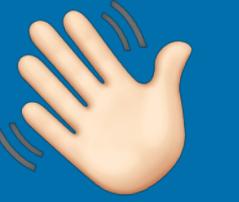


REINVENTING RXJS

HI 

I'M MAX GALLO

ABOUT ME:        

PRINCIPAL ENGINEER @ DAZN



TWITTER: @_MAXGALLO

MORE: MAXGALLO.IO



HERE'S THE AGENDA

1. INTRODUCTION
2. REINVENTING RXJS
3. DEEP DIVE SCHEDULERS

INTRODUCTION

INTRODUCING RXJS

PART OF THE REACTIVE X FAMILY

API FOR ASYNCHRONOUS PROGRAMMING
WITH OBSERVABLE STREAMS

REINVENTING THE WHEEL

BY

TAKING THINGS APART



REINVENTING RXJS



```
1 const { from } = require('rxjs');
2 const { map, filter } = require('rxjs/operators');
3
4 const observable = from([1, 2, 3, 4, 5])
5   .pipe(
6     map(x => x + 1),
7     filter(x => x % 2 === 0),
8     map(x => x - 1),
9   );
10
11 observable.subscribe(
12   val => console.log('odd: ', val),
13   error => console.error(error),
14   () => console.log('Completed!'),
15 );
16 // odd: 1, odd: 3, odd: 5, Completed!
```

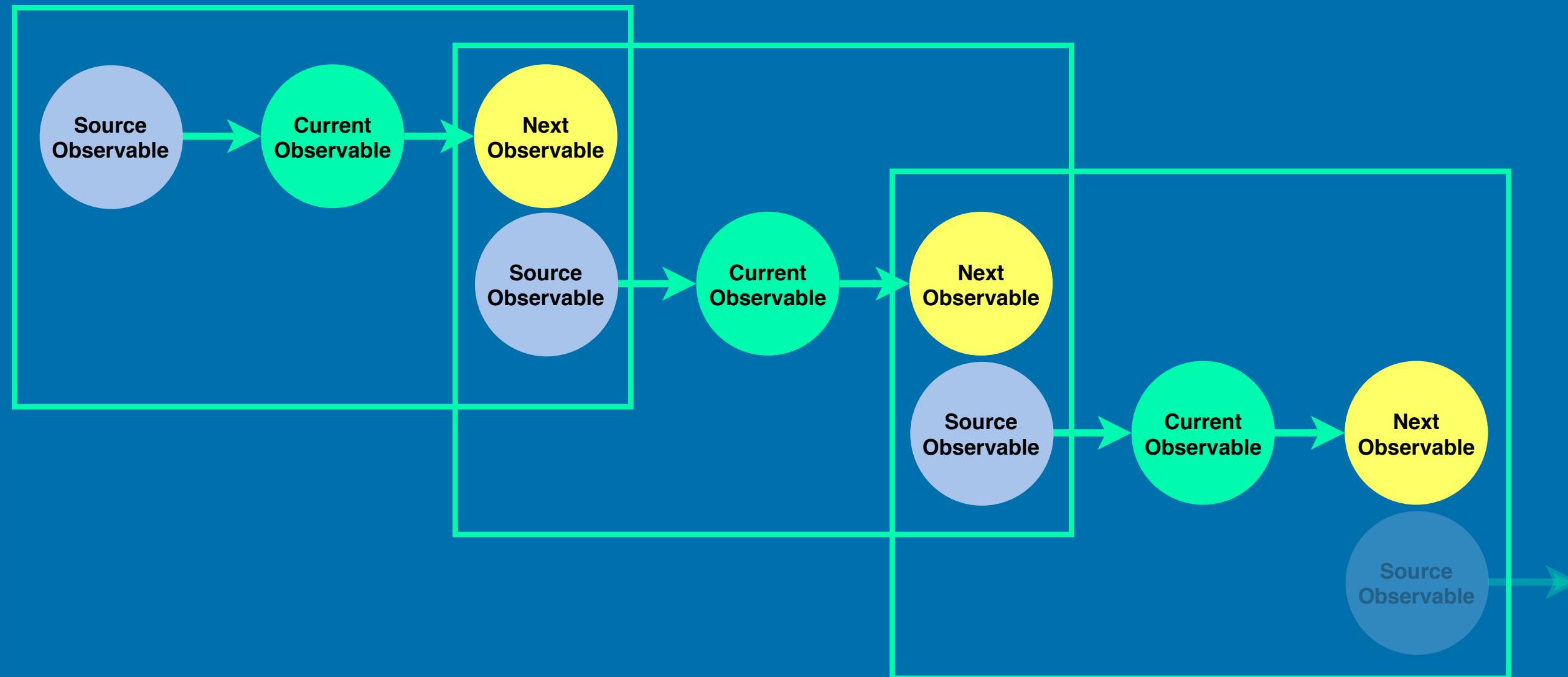
RXJS CODE FIRST IMPRESSIONS

- › SYNTAX IS LIBRARY SPECIFIC
- › EXPLICIT SUBSCRIPTION
- › OBSERVABLE TC39 STAGE 1
- › PIPELINE OPERATOR TC39 STAGE 1



LET'S REINVENT RXJS

RXJS OPERATORS



OPERATOR 1 → OPERATOR 2 → OPERATOR 3

RXJS FROM THE INSIDE

- MADE OF **REUSABLE PARTS** ➤ **STREAMS**
 - CUSTOM OPERATORS
 - LAZY EVALUATION
- STANDARD CONTRACT BETWEEN PARTS
- SYNCHRONOUS BY DEFAULT ➤ **SCHEDULERS**

DEEP DIVE SCHEDULERS

**SCHEDULERS IN RXJS ARE THINGS
THAT CONTROL THE ORDER OF EVENT
EMISSIONS (TO OBSERVERS) AND THE
SPEED OF THOSE EVENT EMISSIONS.**

- ANDRÉ STALTZ

ORDER OF EVENT EMISSIONS
SPEED OF EVENT EMISSIONS

```
1 const {  
2   from, merge,  
3   asyncScheduler,    // ASYNC scheduler  
4   queueScheduler,   // QUEUE scheduler  
5   asapScheduler,    // ASAP scheduler  
6 } = require('rxjs')  
7  
8 const asyncObservable = from(['async'], asyncScheduler)  
9 const queueObservable = from(['queue'], queueScheduler)  
10 const asapObservable = from(['asap'], asapScheduler)  
11  
12 const observable = merge(  
13   asyncObservable  
14   queueObservable,  
15   asapObservable,  
16 );  
17  
18 observable.subscribe(console.log)  
19  
20 console.log('→ Subscribed!')
```

CONSOLE

```
1 const {  
2   from, merge,  
3   asyncScheduler,    // ASYNC scheduler  
4   queueScheduler,   // QUEUE scheduler  
5   asapScheduler,    // ASAP scheduler  
6 } = require('rxjs')  
7  
8 const asyncObservable = from(['async'], asyncScheduler)  
9 const queueObservable = from(['queue'], queueScheduler)  
10 const asapObservable = from(['asap'], asapScheduler)  
11  
12 const observable = merge(  
13   asyncObservable  
14   queueObservable,  
15   asapObservable,  
16 );  
17  
18 observable.subscribe(console.log)  
19  
20 console.log('→ Subscribed!')
```



- 1 queue
- 2 → Subscribed!
- 3 asap
- 4 async

QUEUE / ASAP / ASYNC / ANIMATIONFRAME

QUEUE / ASAP / ASYNC / ANIMATIONFRAME

```
requestAnimationFrame(task)  
|  
Promise.resolve().then(task)  
|  
|
```

```
|  
|  
synchronous  
|  
|  
setInterval(task, delay)
```

ORDER OF EVENT EMISSIONS

QUEUE / ASAP / ASYNC / ANIMATIONFRAME

ORDER OF EVENT EMISSIONS

QUEUE / ASAP / ASYNC / ANIMATIONFRAME

SPEED OF EVENT EMISSIONS

/ VIRTUALTIME /

VIRTUAL TIME SCHEDULER

(HANDS ON  EXAMPLE)

WE MADE IT





DISASSEMBLE

TAKE THINGS APART TO UNDERSTAND
WHAT'S GOING ON



REINVENT

BUILD YOUR OWN VERSION



LEARN

BUILD MEANS LEARN



SHARE

TELL OTHERS WHAT YOU DID

DISASSEMBLE / REINVENT / LEARN / SHARE

TWITTER @_MAXGALLO
OTHER MAXGALLO.IO

THANK YOU



SLIDES [GITHUB.COM/MAXGALLO/TALK-REINVENTING-RXJS](https://github.com/maxgallo/talk-reinventing-rxjs)