



RxSwift

Introduction to reactive programming

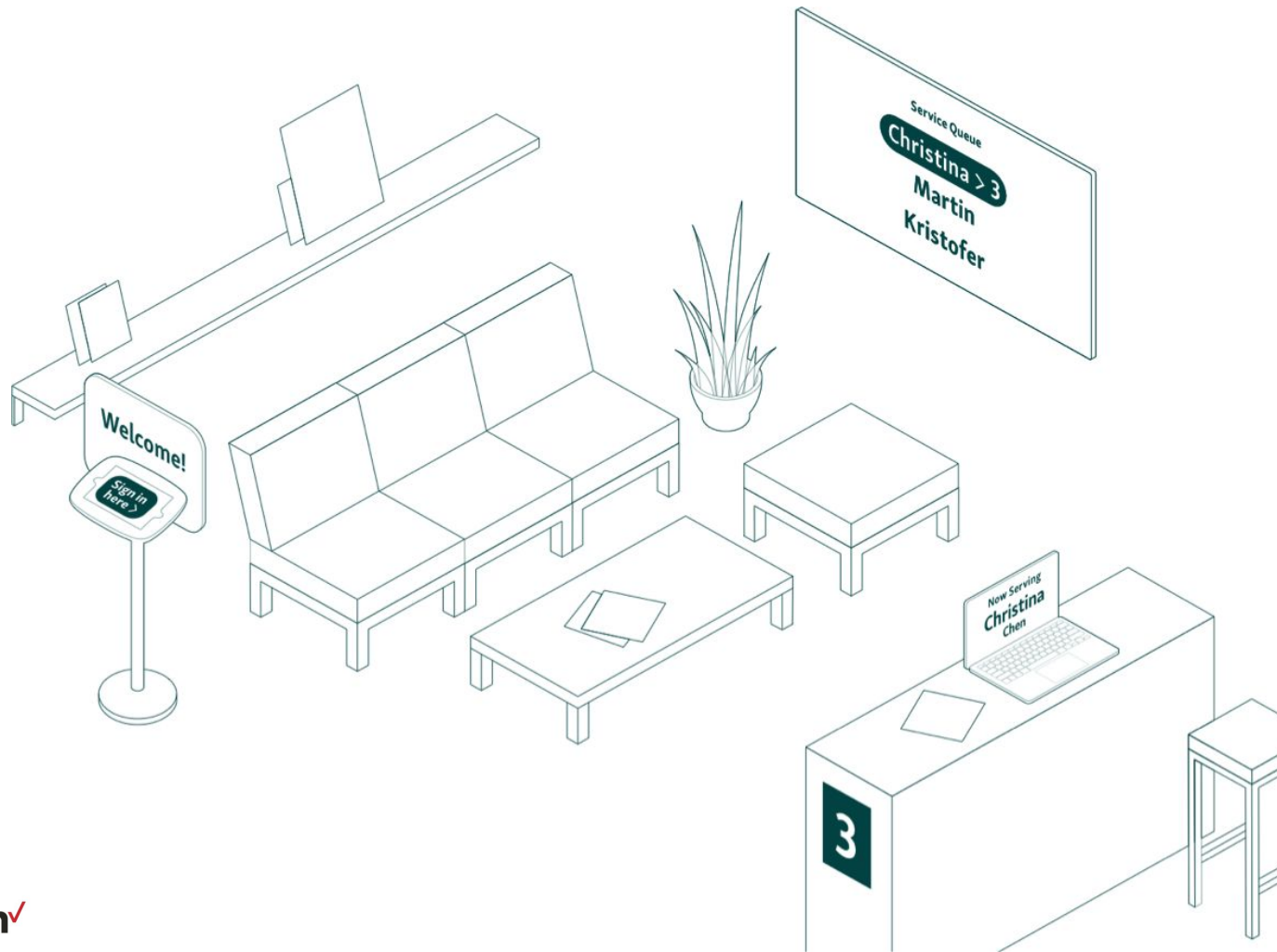
**RxSwift isn't KVO, ReactNative or
ReactJS**

About me

- Swift developer at Qminder **E E**
- **BCompSci** & **MComSci** unfinished
DmCompSci
- > 15 years in IT
- love open source



Qminder



Premium
Reseller

lyft



Uber

FINANCIAL TIMES

verizon✓

Qminder



Uber



FINANCIAL TIMES



NOT REACTIVE



REACTIVE

Agenda

- What is Rx
- RxSwift and building blocks
- Examples
- Pros and Cons

Imperative programming

- Saved state
- Order of execution
- Side effects

In computing, reactive programming is a programming paradigm oriented around data flows and the propagation of change.

Wikipedia

An API for asynchronous programming
with observable streams

Reactive programming is programming with
asynchronous data streams

Reactive programming

- Asynchronous
- No saved state
- Functional
- Responsive

ReactiveX

An API for asynchronous programming with observable streams.

- Microsoft (Rx.NET) - 2009
- Supports many languages
- RxSwift - 2015

Rx components

- Observable
- Observers
- Schedulers

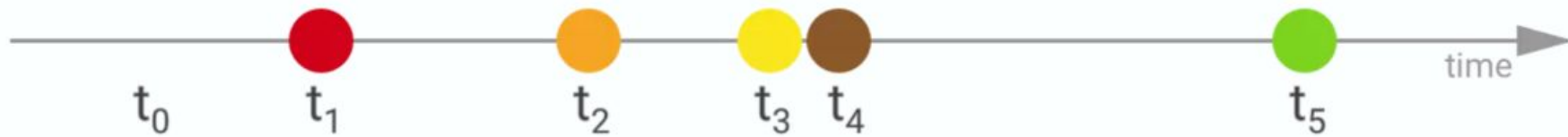
EVERYTHING



CAN BE A SEQUENCE



Array – elements in time



Observable is an array with time dimension

Observables

Completed and Error will
terminate the stream

Next

Completed

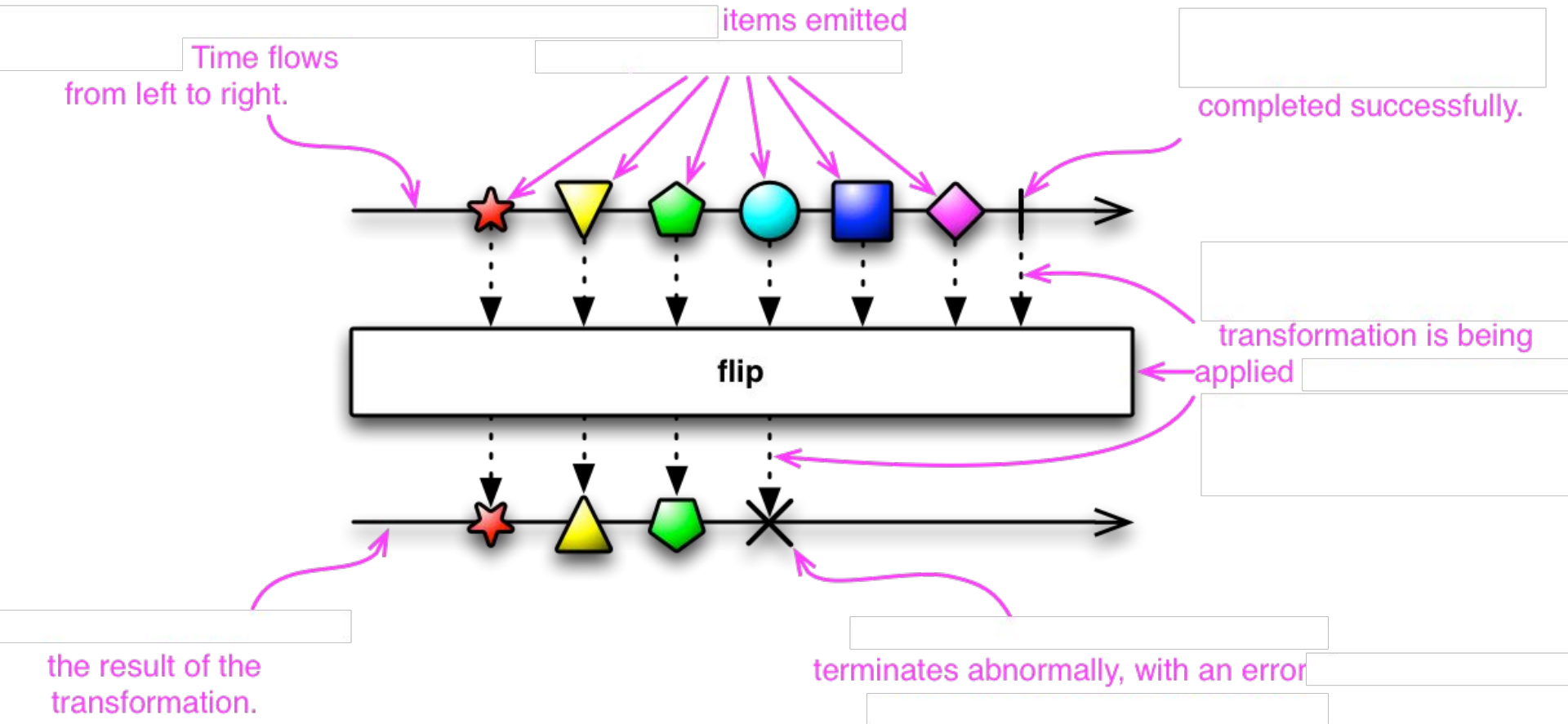
Error

Operators

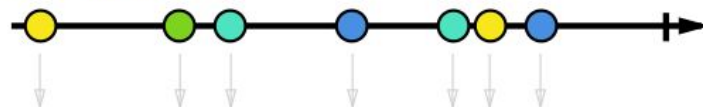
Filtering

Transforming

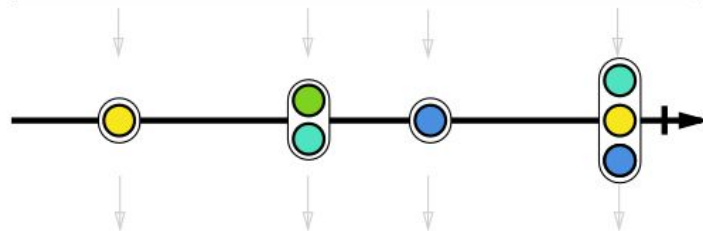
Combining



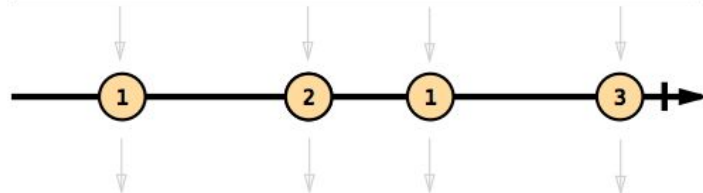
Click stream



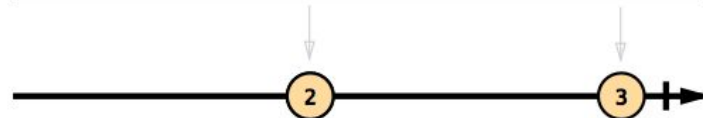
`buffer(clickStream.throttle(250ms))`



`map('get length of list')`



`filter(x >= 2)`



Multiple clicks stream

RxJS Marbles

CREATION OBSERVABLES

[from](#)
[interval](#)
[of](#)
[timer](#)

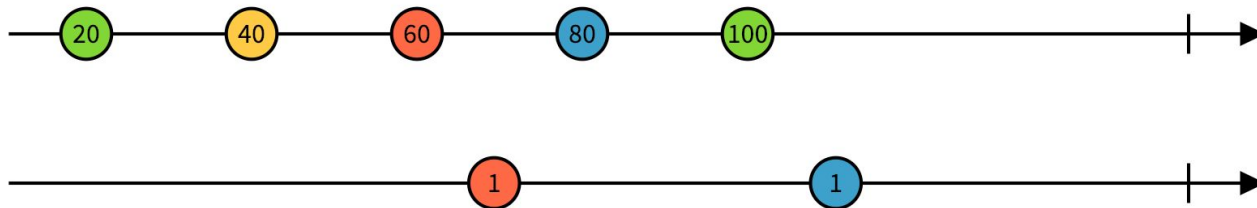
CONDITIONAL OPERATORS

[defaultIfEmpty](#)
[every](#)
[sequenceEqual](#)

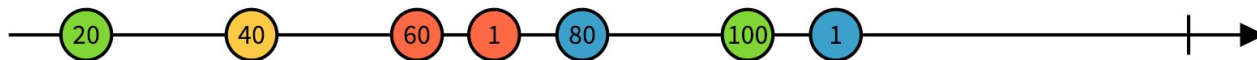
COMBINATION OPERATORS

[combineLatest](#)
[concat](#)
[merge](#)
[race](#)
[startWith](#)
[withLatestFrom](#)
[zip](#)

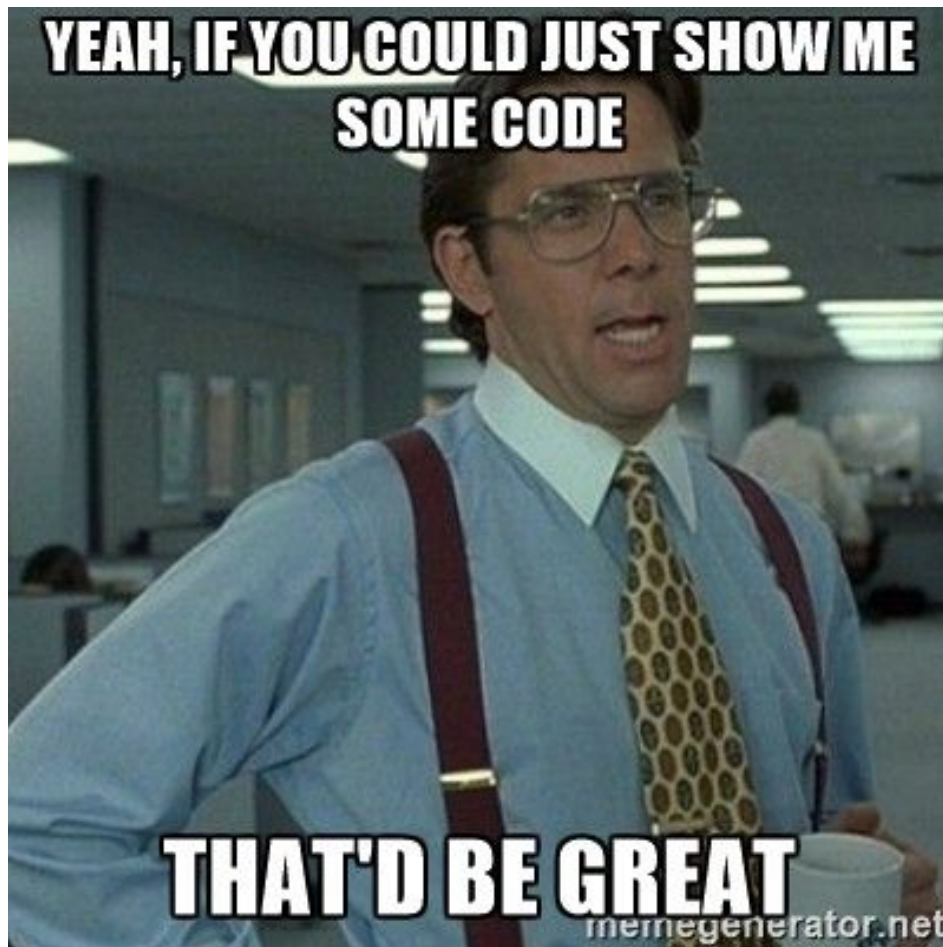
Interactive diagrams of Rx Observables



merge



**YEAH, IF YOU COULD JUST SHOW ME
SOME CODE**



THAT'D BE GREAT

memegenerator.net

UI bindings

RxCocoa

RXCOCOA

The glue that connects **Observable streams** to iOS, macOS, watchOS and tvOS UI Components



RX + MOBILE UI

Streams of UI components



RxCocoa (part of RxSwift) lets you “subscribe” to events/properties on most standard UI components as Observable streams under the **rx** namespace, as well as most delegates. Easily customizable.

```
button.rx.tap  
textField.rx.text  
slider.rx.value  
collection.rx.itemSelected  
datePicker.rx.date
```



Rx extensions

Almost for everything

3RD PARTY LIBRARIES

RxSwift's Community is thriving with amazing Open-Source projects, be it wrappers around libraries you know, or entirely new Libraries.

To find an Rx wrapper for a library or component you use, **simply add Rx before**, and you'll probably find it available online ;-)



RxSwiftCommunity @ GitHub



RxAlamofire



RxRealm



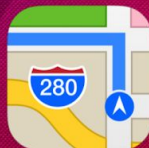
RxGoogleMaps



RxMoya



RxStarscream



RxMKMapView

Other recommended
libraries:

RxSwiftExt

RxDataSources

Action

RxAnimated

RxFeedback

ReactorKit

And many many more...

Testing

- Async XCTest with expectations
 - RxBlocking making it synchronous
 - Using RxTest and custom Schedulers
-
- <https://www.raywenderlich.com/7408-testing-your-rxswift-code>
 - <https://www.youtube.com/watch?v=FgbTenGH-P0>

+

Agile and reusable code

Thread safety

Eliminate state from code

Testing

Well tested

Error handling

Large community and extensions

Extends MVVM

—

Steep learning curve

Change previous practices

Debugging

Testing

Over-reactive

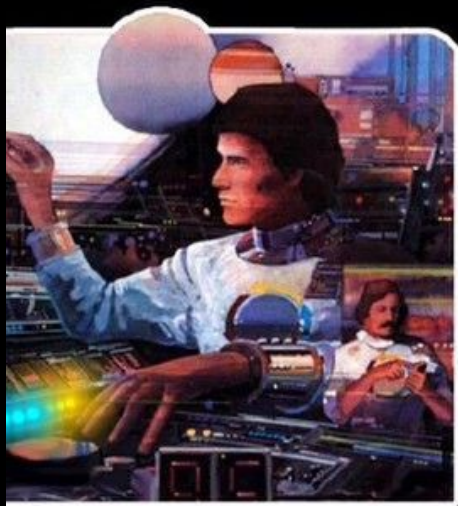
Overhead

COMPANIES USING RX

Rx is mature and has been in use by some of the biggest tech companies



THE TWO STATES OF EVERY PROGRAMMER



I AM A GOD.



**I HAVE NO IDEA
WHAT I'M DOING.**

Resources

- <http://reactivex.io/>
- <https://github.com/ReactiveX/RxSwift>
- <https://github.com/RxSwiftCommunity>
- <http://adamborek.com/>
- <http://rx-marin.com/>
- [RayWenderlich RxSwift book](#)
- Shai Mishali presentations

**THANK YOU
FOR YOUR ATTENTION!**



AND GOOD LUCK!