Our Observable Past Present And Future

github.com/mattpodwysocki/rxjs-live-2019

Matthew Podwysocki
@mattpodwysocki

@mattpodwysocki

@mattpodwysocki
@mpodwysocki

BluerThanBlueFalcon


MICROSOFT

Visual Studio App Center

appcenter.ms

@VSAppCenter


2005

Microsoft Live Labs

Volta

```
public partial class VoltaPage1 : Page
{
    public VoltaPage1()
    {
        var output = new Div();

        var b = new Input();
        b.Type = "button";
        b.Value = "Get Message";
        b.Click += () => output.InnerHtml = Handler.GetMessage();

        Document.Body.AppendChild(output);
        Document.Body.AppendChild(b);
    }
}

class Handler
{
    [RunAtOrigin]
    public static string GetMessage()
    {
        return "Hello, World";
    }
}
```

The Assembly Language of the Web

IL2JS

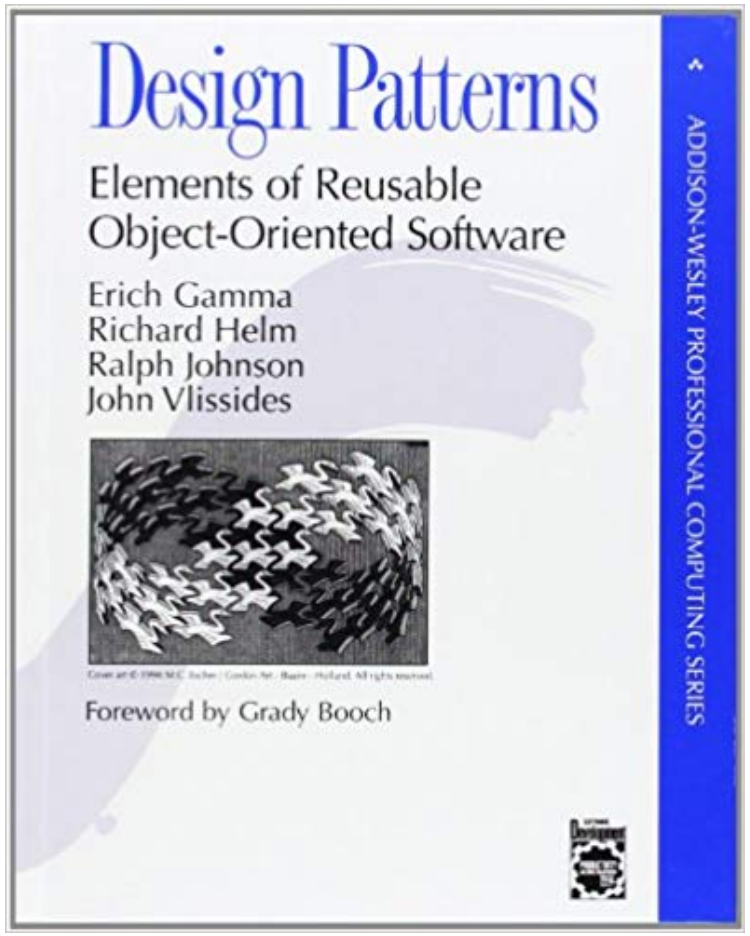
https://github.com/reactive-extensions/il2js

CrockScript

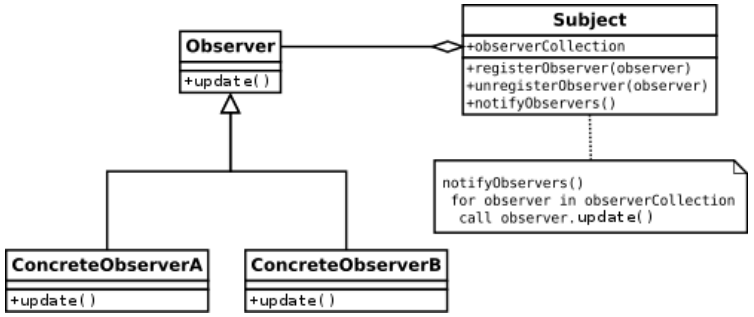
```
var mousemove$ =
  from ev in document.mousemove
  select new {
    x: ev.target.clientX,
    y: ev.target.clientY
  };
```


Async is
Awful

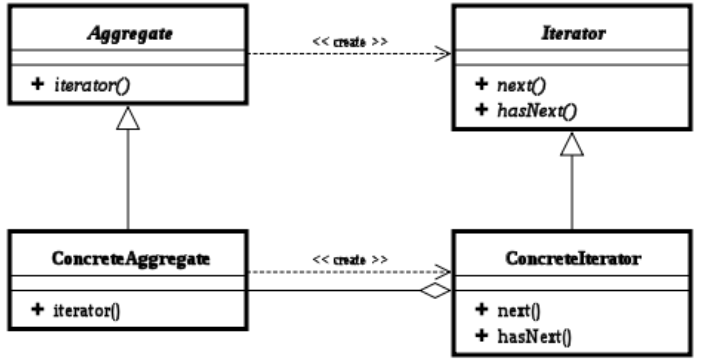
1994 - Design Patterns Emerge



Subject/Observer

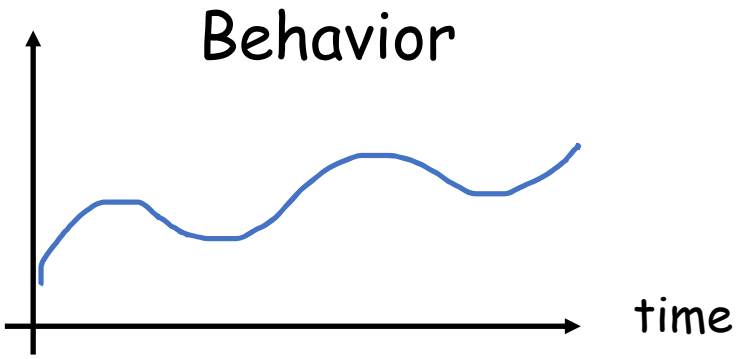


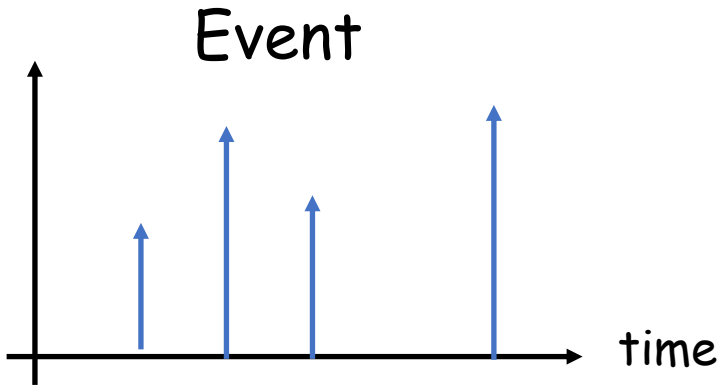
Iterator



Duality of Push to Pull Collections

1997 - Functional Reactive Programming





Functional Reactive Animation
Conal Elliott / Paul Hudak
http://conal.net/papers/icfp97/

Your Mouse is a Database

Web and mobile applications are increasingly composed of asynchronous and realtime streaming services and push notifications.

Erik Meijer

https://queue.acm.org/detail.cfm?id=2169076

Joe Armstrong @joeerl · Apr 4

One on the disadvantages of having a PhD in computer science is that I get asked really difficult questions.

Like - "In gmail on my iPhone I press archive - can I get my mail back?"

and "Why have they changed the interface?"

Why no easy questions like what's a monad?

59 677 3.8K

Ahmad @sudoreality · Apr 4

What's a monad?

4 15

Joe Armstrong
@joeerl

Follow

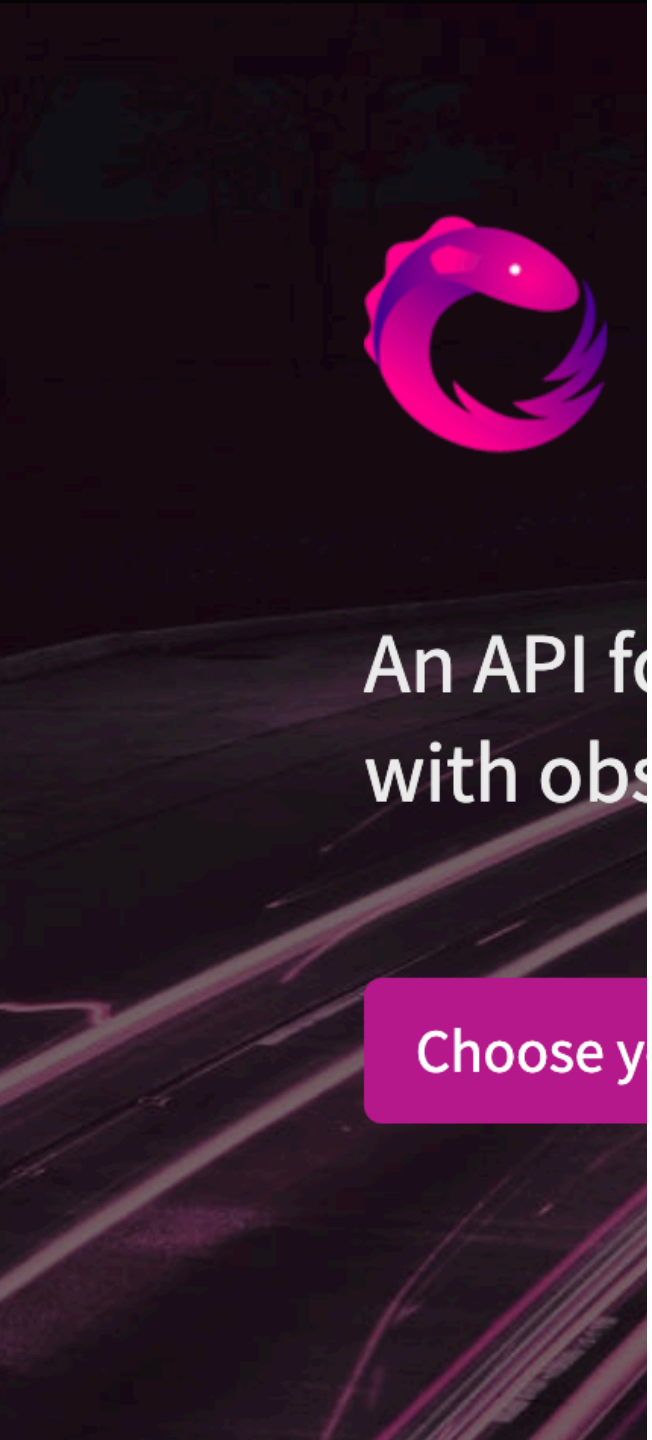
Replying to @sudoreality

it's a thingy - see
en.wikipedia.org/wiki/Monad

11:59 AM - 4 Apr 2019

3 Retweets **60** Likes

Monads???

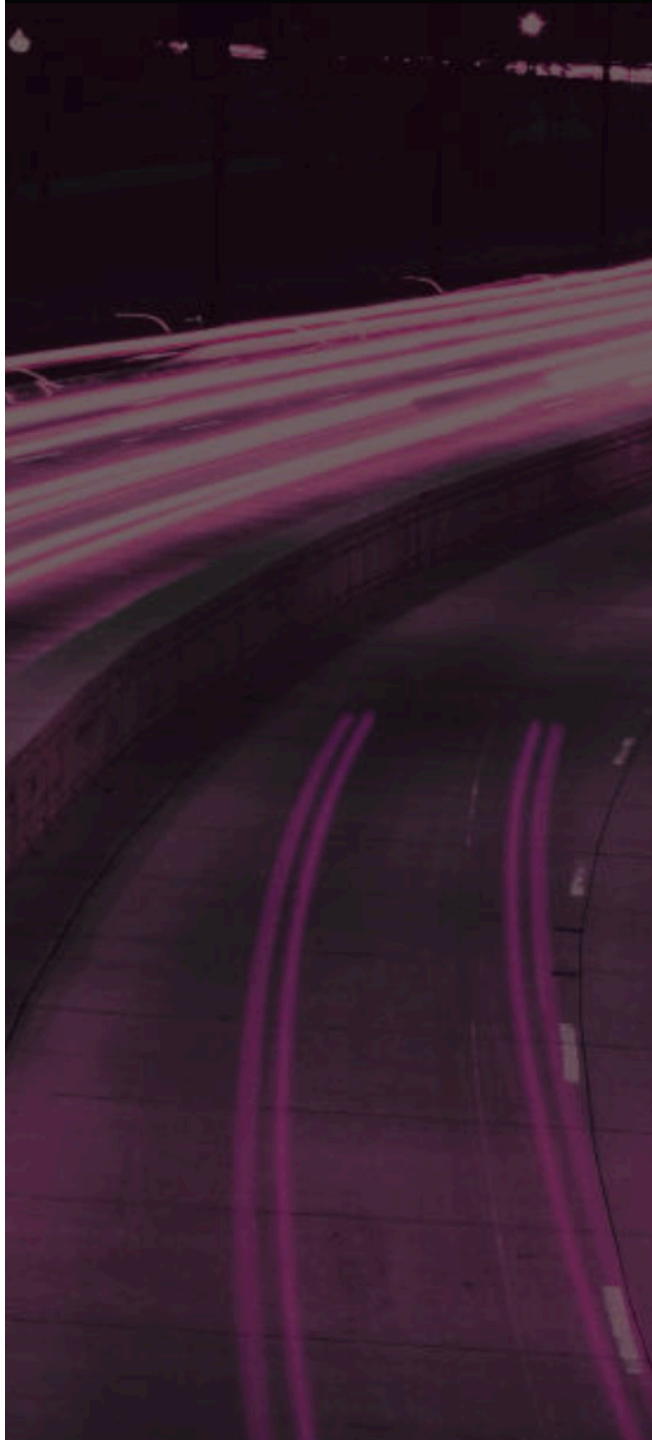


Languages

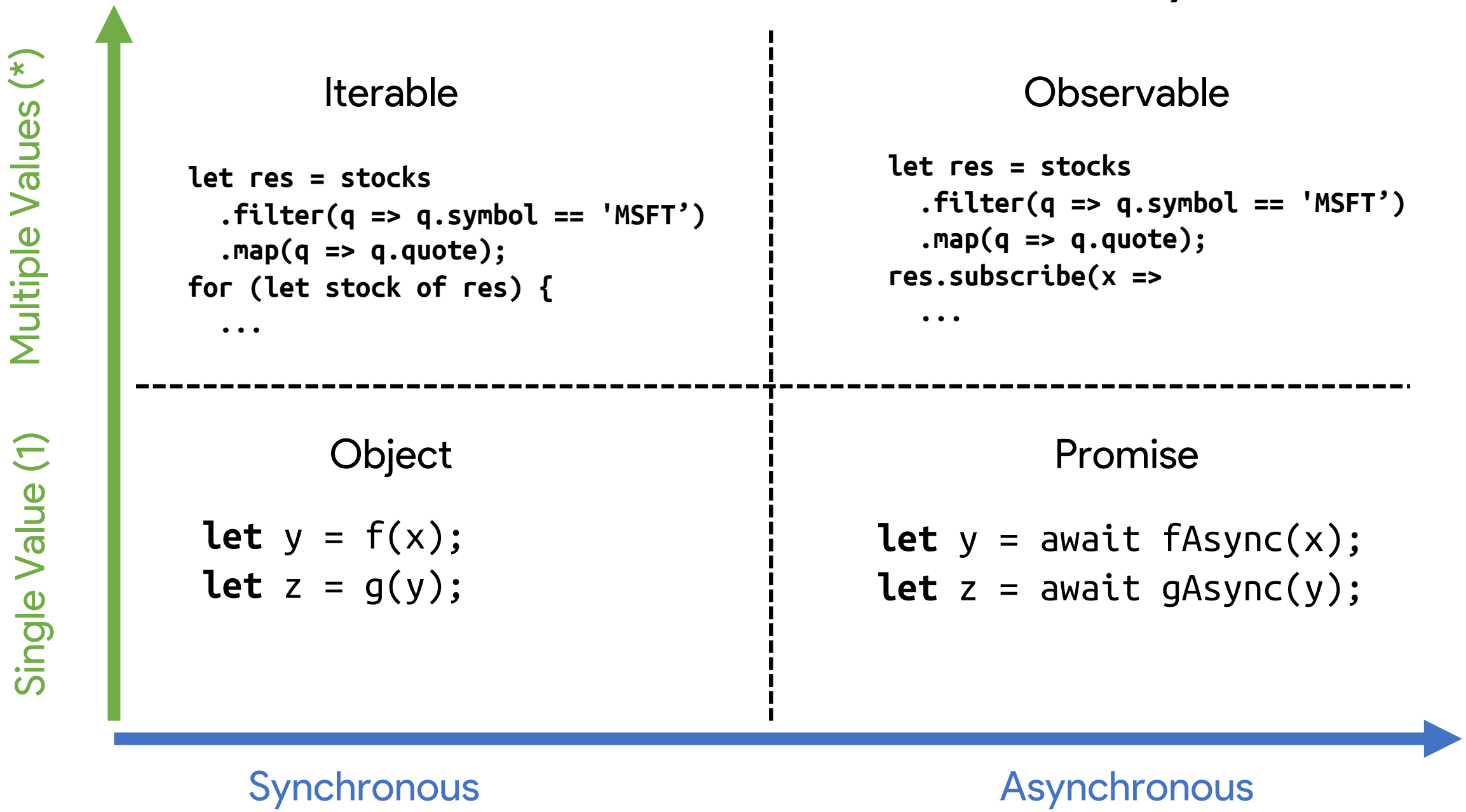
- Java: RxJava
- JavaScript: RxJS
- C#: Rx.NET
- C#(Unity): UniRx
- Scala: RxScala
- Clojure: RxClojure
- C++: RxCpp
- Lua: RxLua
- Ruby: Rx.rb
- Python: RxPY
- Go: RxGo
- Groovy: RxGroovy
- JRuby: RxJRuby
- Kotlin: RxKotlin
- Swift: RxSwift
- PHP: RxPHP
- Elixir: reaxive
- Dart: RxDart

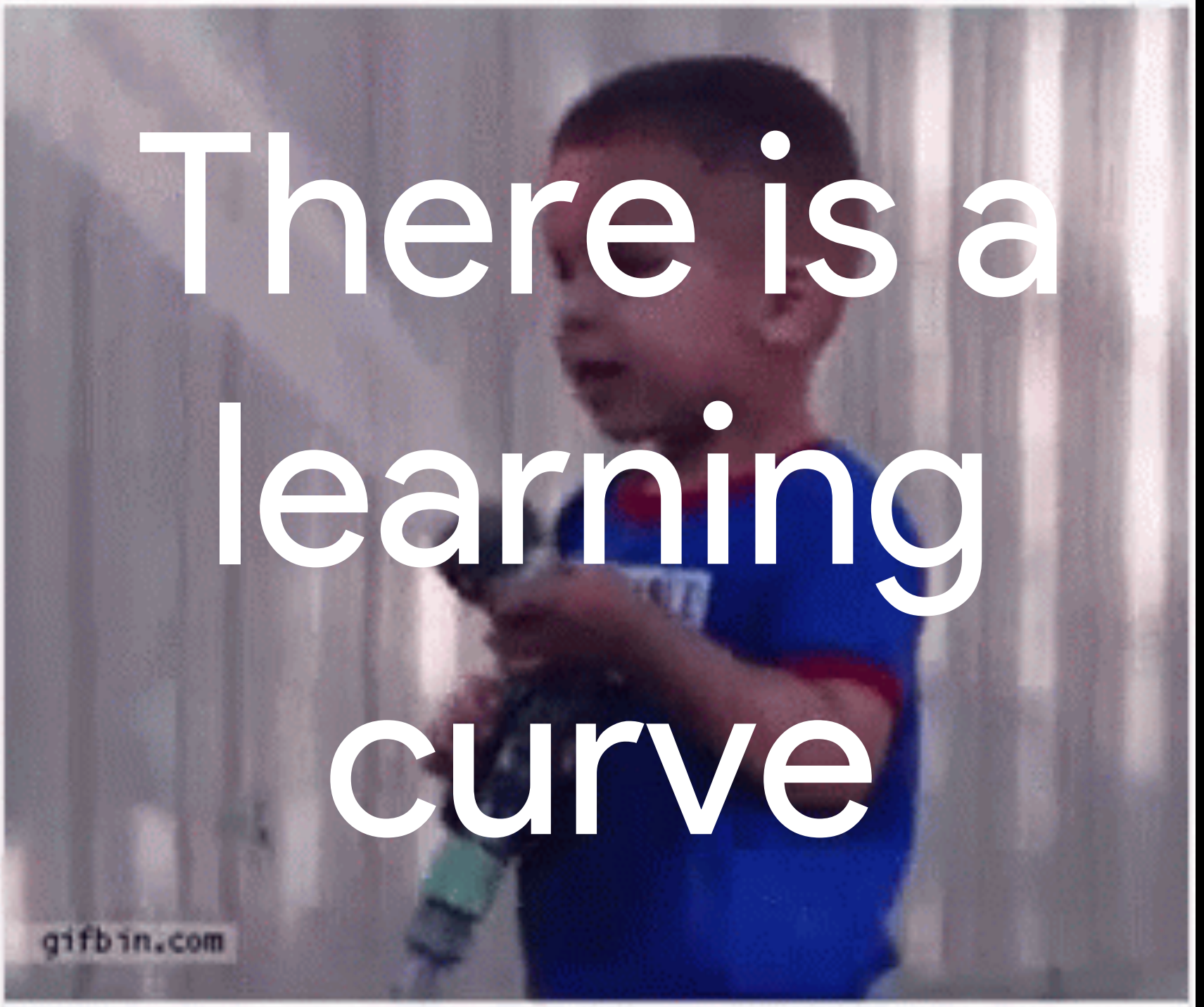
ReactiveX for platforms and frameworks

- RxNetty
- RxAndroid
- RxCocoa



General Theory of Reactivity




There is a learning curve
gifbin.com

interval
throwError
filter
concat
mergeAll
defer
empty
bufferCount
groupJoin
never
debounceTime
groupBy
retry
switchMap
toMap
of
timer
onErrorResumeNext
join
first
merge
race
withLatestFrom
exhaustMap
throwError
map
toSet
catchError
timeout
iif
filter
asObservable
last
zip
buffer
scan
throttle
from
fromEvent
single
debounce
combineLatest
windowCount
mergeMap
timer
toPromise
concatAll
throttleTime
fromEventPattern
window
switch
pluck
sum
max
min
concat
reduce
minBy

jQuery

HTML

dojo toolkit

ANGULARJS by Google

Ext JS

yui

MOOTOOLS



RxJS Evolved with the Times...

```
import * as Rx from 'rxjs';

Rx.Observable.fromEvent(document, 'mousemove')
    .map(ev => ev.target)
    .scan(calculateDelta, { x: 0, y: 0 })
    .subscribe(render);
```

RxJS Evolved with the Times...

```
import { Observable } from 'rxjs';
import 'rxjs/add/operator/map';
import 'rxjs/add/operator/scan';

Observable.fromEvent(document, 'mousemove')
    .map(ev => ev.target)
    .scan(calculateDelta, { x: 0, y: 0 })
    .subscribe(render);
```

RxJS Evolved with the Times...

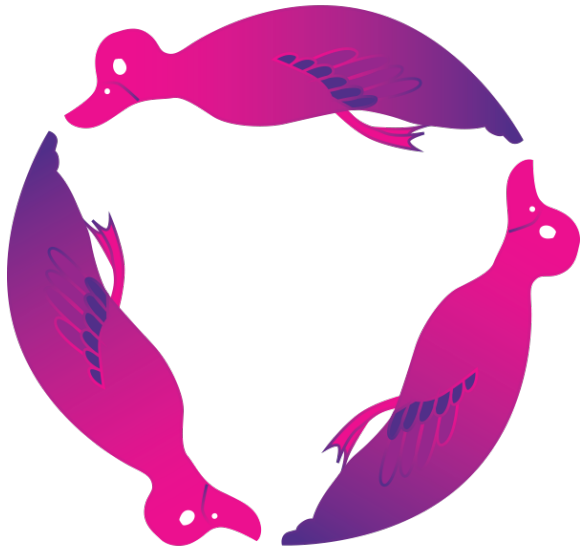
```
import { fromEvent } from 'rxjs';
import { map, scan } from 'rxjs/operators';

fromEvent(document, 'mousemove')
    .pipe(
        map(ev => ev.target),
        scan(calculateDelta, { x: 0, y: 0 })
    )
    .subscribe(render);
```

A predictable state container for JavaScript apps.

GET STARTED

```
$action
    .pipe(scan(reducer, initialState))
    .subscribe(renderer);
```

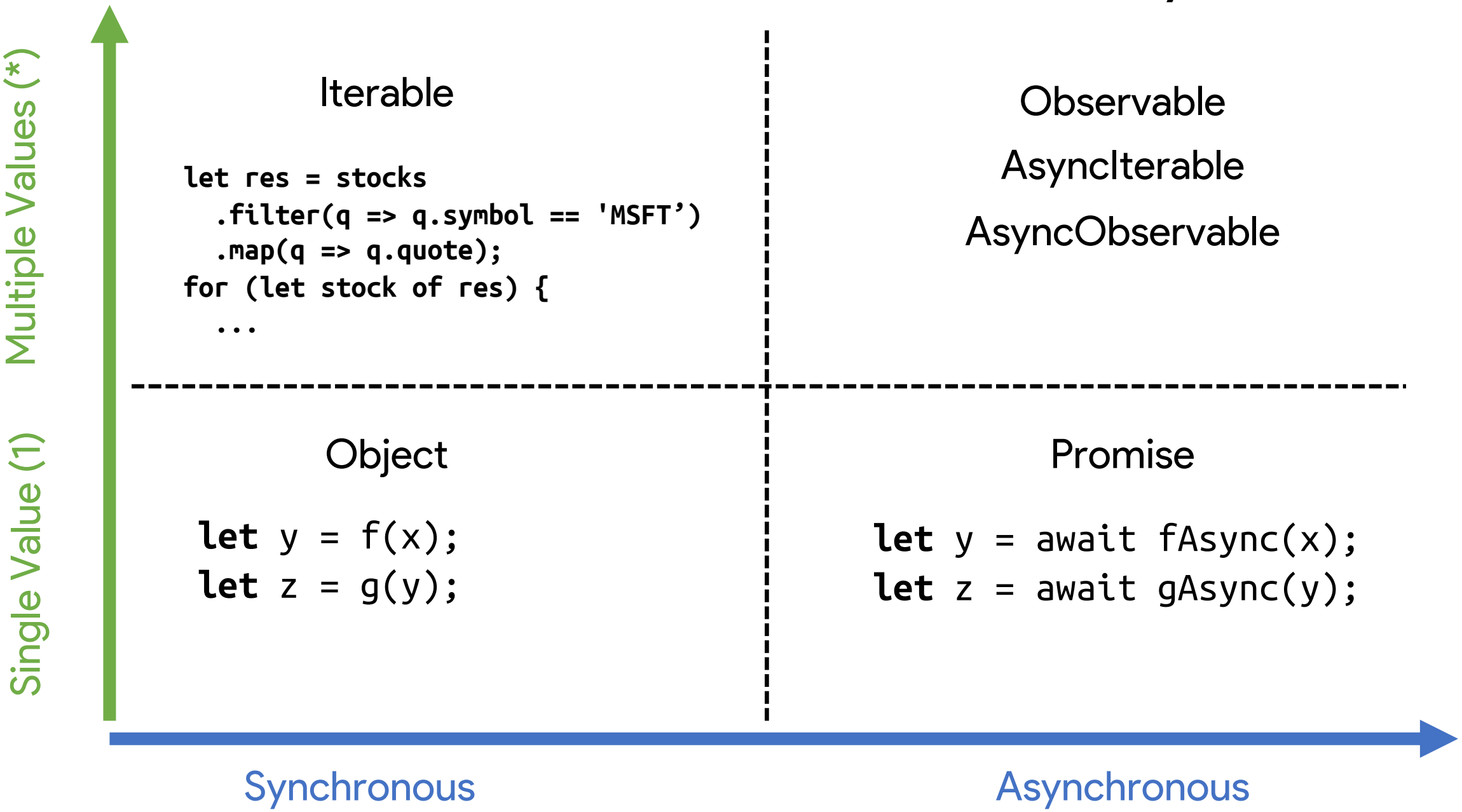
State Management using RxJS



NGXS

Akita

General Theory of Reactivity



Things can overload

Pull/Push/Pull Collections with AsyncIterable

```
import { as } from 'ix/asynciterable';
import { map } from 'ix/asynciterable/operators';

async function* getData() { ... }

let data = getData()
  .pipe(map(transformData))
  .pipe(domEncodeStream);

for await (let item of data) {
  ...
```

https://github.com/reactivex/ixjs

Pull/Push/Pull Collections with AsyncObservable

```
import { filter, map } from 'ix/asyncobservable/operators';

const stream = getData().pipe(
  map(transformData),
  filter(filterData)
);

const subscription = await stream.subscribeAsync({
  next: item => { await processItem(item); },
  error: err => { await processError(err); }
});
```

https://github.com/reactivex/ixjs


Where will
we go
next?

```
function map<T, R>(
  selector: (source: T, index: number, signal: AbortSignal) => Promise<R>
): AsyncObservable<R>

function mergeMap<T, R>(
  selector: (source: T, index: number, signal: AbortSignal) => AsyncIterable<R>
): AsyncIterable<R>

function forEach<T>(
  selector: (source: T, index: number, signal: AbortSignal) => Promise<void>
): Promise<void>
```

Let's Build Something Amazing Together!

github.com/mattpodwysocki/rxjs-live-2019

Matthew Podwysocki
@mattpodwysocki