# Functional Reactive Programming avec RxSwift

Cocoaheads Strasbourg Mai 2017

Nicolas VERINAUD

# Qui suis-je ?

- Freelance
- Formateur
- Développeur iOS depuis 2011
- Prog. Fonctionnelle (yay F# !)

@verinaud

# Functional Programming

- ≠ OOP
- Composition > Héritage
- Fonctions en paramètre (closures)
- Immuabilité

# Reactive Programming

- Réagir aux changements d'états
- Délégation
- Closures
- Notifications (😨)
- KVO (🤢)

# Reactive Programming

- Réagir aux changements d'états
- Délégation [implicite]
- Closures [implicite]
- Notifications (😰) [implicite]
- KVO (🤢) [implicite]

"Explicit is better than implicit."

-The Zen of Python

# Observable<T>

# Observable<T>

Une valeur qui change au cours du temps.

# Observable<T>

Push

≠

Pull (Sequence)

# Observable<T>

```
protocol Observable<T>
{
    func subscribe(_ observer: Observer<T>)
    -> Disposable
}
```

# Observable<T>

```
protocol Observer<T>
{
    func onNext(_ element: T)
    func onError(_ error: Error)
    func onCompleted()
}
```

# Observable<T>

Observable<A> -> Observable<B>

Observable<String> -> Observable<Bool>

# How ?

# Observable<T>

```
Observable<A> -> Observable<B>

Observable<String> -> Observable<Bool>
```

# Avec map !

*Monad in disguise...psshh*

# Observable<T>

```
Observable<A> -> Observable<B>

Observable<String> -> Observable<Bool>
```

```
func map(_ transform: (A -> B)) ->
            Observable<B>
```

# Observable<T>

```
// strings: Observable<String>

let isEmpty = strings.map { $0.isEmpty }
```

# Observable<T>

# Demo !

RxSwift
ou
ReactiveCocoa
?

# RxSwift

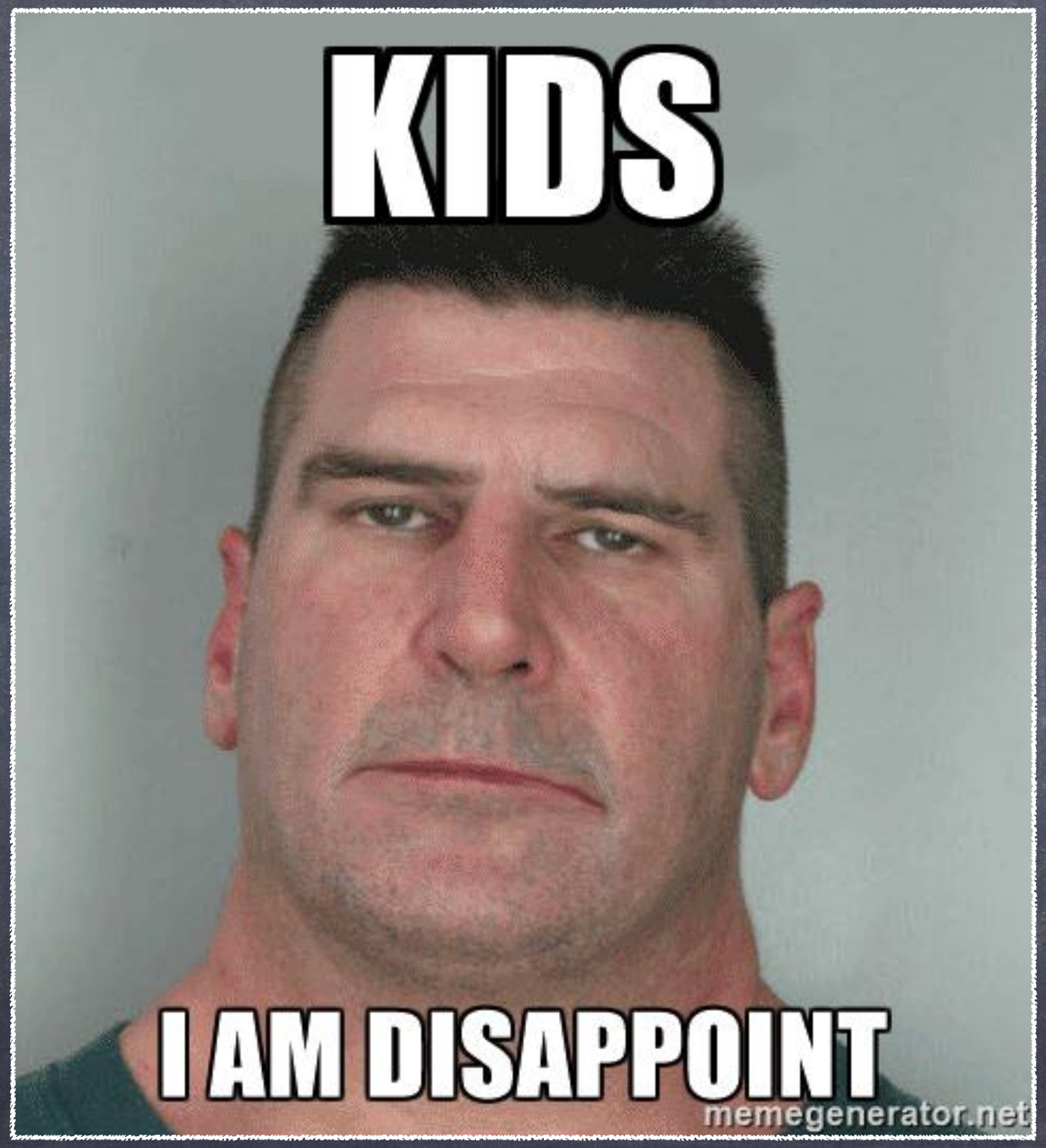
- 👍 Respectueux des interfaces Rx originales
- 👎 Pas de typage pour `Error`

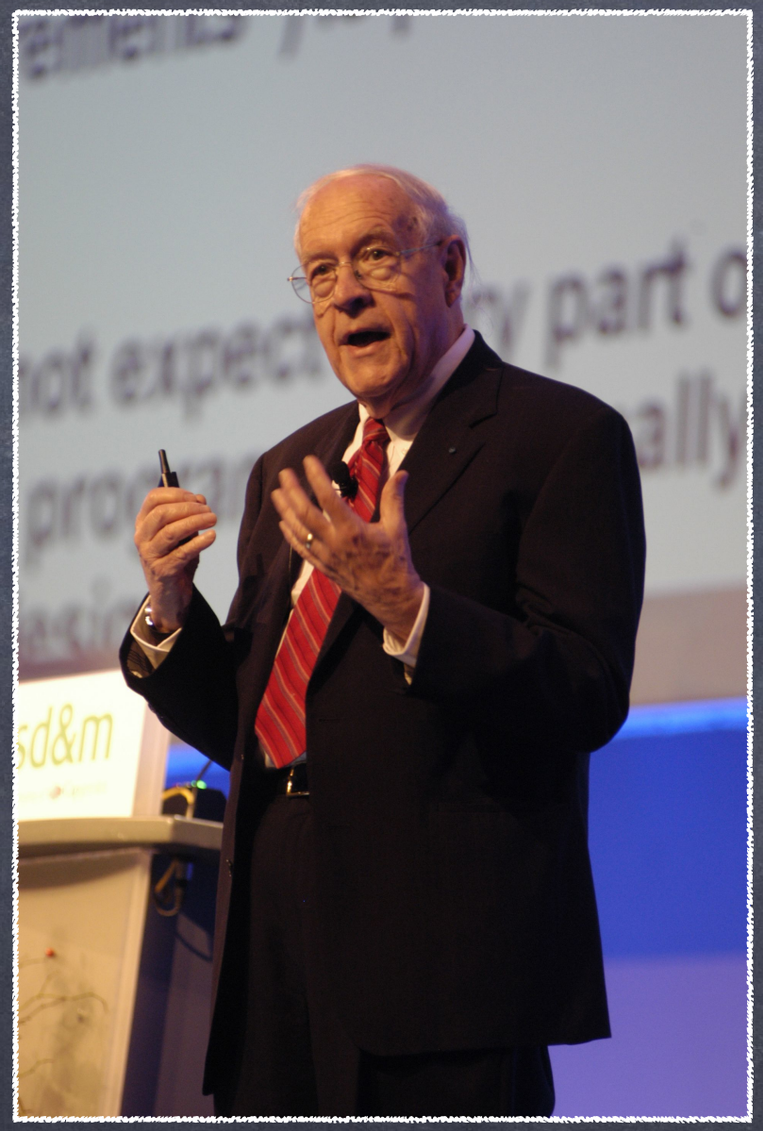
# ReactiveCocoa

- 👍 Typage de Error
- 👍 Distinction Hot et Cold
- 👎 Concepts ≠ Rx (Signal)

# One more thing...


REACTIVE
EVERYWHERE
memegenerator.net


KIDS
I AM DISAPPOINT
memegenerator.net



"No Silver Bullet"

-Fred Brooks


"Reactive is dead, long live tasteful composition of side effects."

-Erik Meijer

# Tasteful Side Effects

| | One | Many |
|---|---|---|
| Sync | T | Sequence |
| Async | Future<T> | Observable<T> |

# Tasteful Side Effects

```
Future<Option<Array<String>>>
```

```
Result<Observable<String>>
```

etc…

# Tasteful Side Effects

```
Future<Option<Array<String>>>
```

```
Result<Observable<String>>
```

etc…

Soyez explicites !

# Aller plus loin...

- ReactiveX.io
- IntroToRx.com
- React 2014 : Erik Meijer - What does it mean to be Reactive?

@nverinaud

n.verinaud@gmail.com

# Merci !

nverinaud.com

# Bibliographie


- Kaa - http://www.abadinte.com/wp-content/uploads/2014/09/Kaa-aie-confiance.jpg
- Fred Brooks Wikipedia - https://fr.m.wikipedia.org/wiki/Frederick_Brooks
- Erik Meijer Wikipedia - https://en.wikipedia.org/wiki/Erik_Meijer_%28computer_scientist%29
- React 2014 : Erik Meijer - What does it mean to be Reactive?
- Meme Generator - memegenerator.net