The Future of Reactive Cocoa

@jspahrsummers

RAC3.0

Hotor Cold?

.NET offers: IEnumerable IEnumerator IObservable IObserver

Observer Event -> ()

Observable (Event -> ()) -> ()

Flip the arrows!

() -> Event

Enumerator () -> Event

Enumerable () -> (() -> Event)

Push:

```
Event -> ()
(Event -> ()) -> ()
       Pull:
    () -> Event
() -> (() -> Event)
```

But why is enumeration

biocking

in a reactive framework?

Enumerator v1 () -> Event

Enumerator V2 () -> Promise Event

Push:

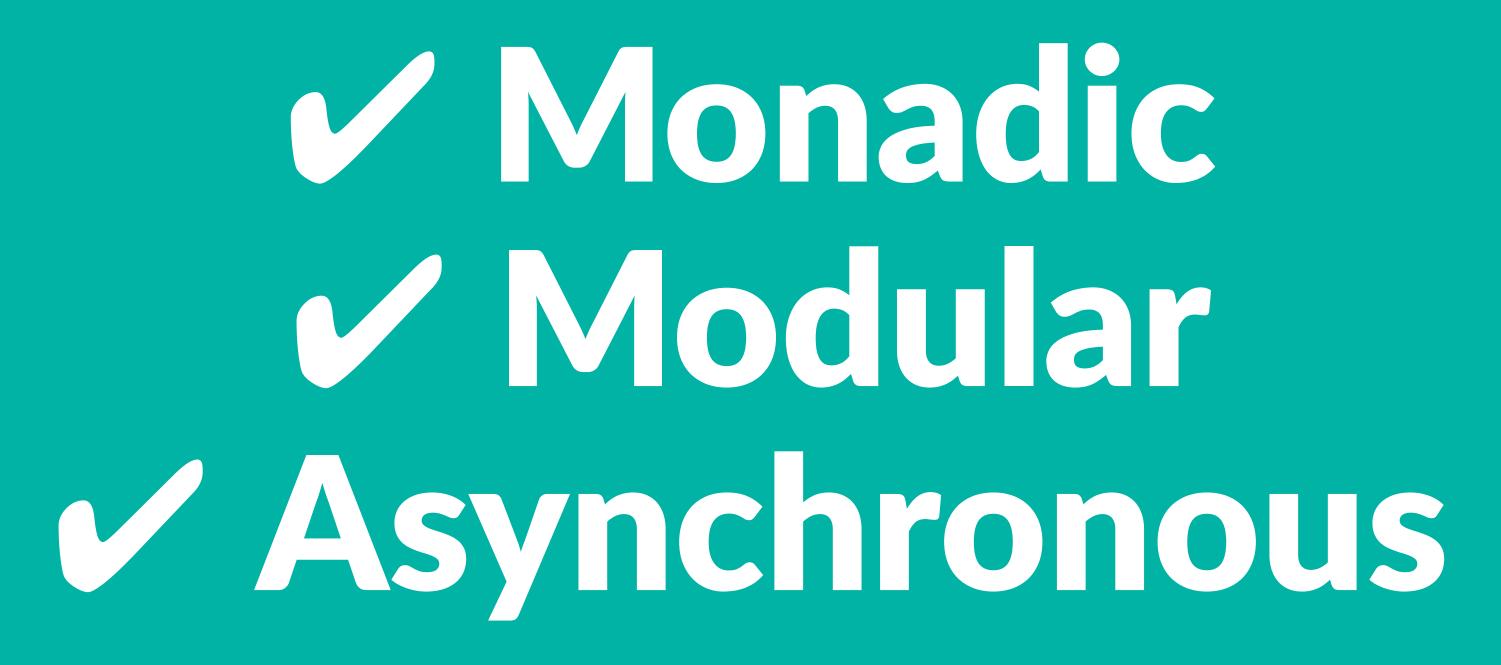
```
Event -> ()
(Event -> ()) -> ()
Pull:
```

() -> Promise Event

() -> (() -> Promise Event)

Promises allow work to be performed out-of-order & asynchronously when and where the caller wants

Observables and Enumerables are:



Observables are driven by the producer

Enumerables are controlled by the consumer

Observables are the same to all observers

Enumerables are enumerated independently

Observables are always live Enumerables start new work with each enumeration

Hot signals Observables Cold signals Enumerables

Questions?