

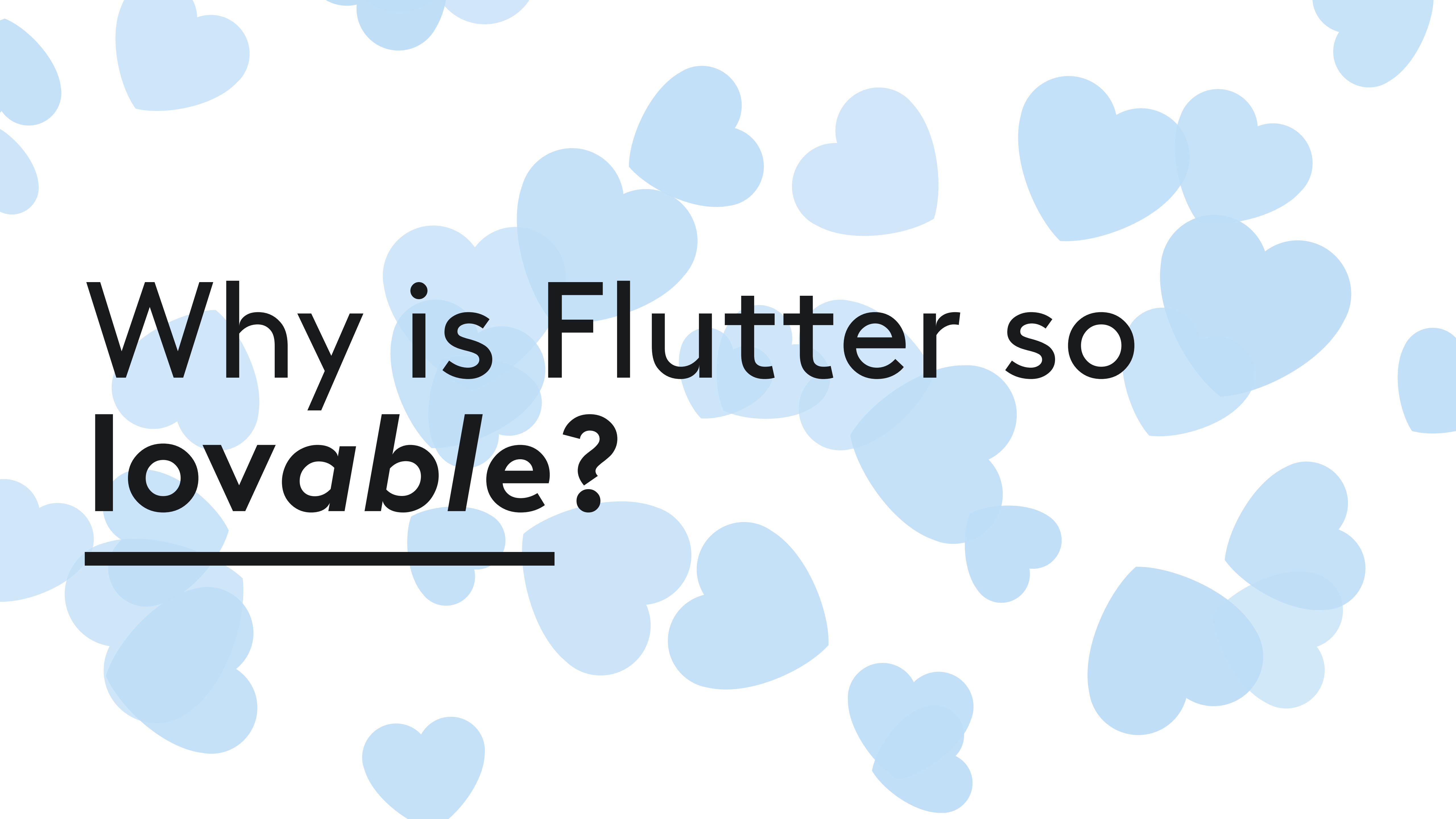
# Flutter

a modern declarative  
UI toolkit

Iain Smith

 @b09913





# Why is Flutter so *lovable*?

---

# Hi, I'm Iain Smith!



# Hi, I'm Iain Smith!

Lead Mobile Developer  
Screenmedia  
@b099I3



Take you on a journey  
of what I discovered...

Lots of developers love  
Flutter... 

---



Pawan Kumar  
@imthepk

Lots of developers like  
Flutter...

I LOVE Flutter!  
- - - - -  
(\\_) ||  
(•ㅅ•) ||  
/ づ

And Dart!  
- - - - -  
(\\_) ||  
(•ㅅ•) ||  
/ づ

#flutter

5:55 AM · Apr 26, 2020 · Twitter for Android



Daye  
@ib\_daye

I love flutter ❤️ ❤️

3:12 AM · May 8, 2020 · Twitter for iPhone

# Lots of developers | Flutter...



Pawan Kumar  
@imthepk

I LOVE Flutter!  
- - - - -  
(\\_) ||  
(•ㅅ•) ||  
/ づ

And Dart!  
- - - - -  
(\\_) ||  
(•ㅅ•) ||  
/ づ

#flutter

5:55 AM · Apr 26, 2020 · Twitter for Android



Daye  
@ib\_daye

I love flutter ❤️ ❤️

3:12 AM · May 8, 2020 · Twitter for iPhone

# Lots of developers | Flutter...



Pawan Kumar  
@imthepk

I LOVE Flutter!

(\\_/\\_) ||  
(•ㅅ•) ||  
/ づ

And Dart!

(\\_/\\_) ||  
(•ㅅ•) ||  
/ づ

#flutter

5:55 AM · Apr 26, 2020



Hicham Baghedoud  
@baghedoudhicham

btw #Flutter, I love you ❤️ ❤️

8:02 PM · Mar 3, 2021 · Twitter Web App



Daye  
@ib\_daye

I love flutter ❤️ ❤️

• 8, 2020 · Twitter for iPhone



Pawan Kumar  
@imthepk

I LOVE Flutter!

(\\_/\\_) ||  
(•ㅅ•) ||  
/ づ



Danish Ansari  
@md\_is\_coding

Once you try #Flutter, either you  
love it or hate it with your heart,  
there's nothing in between.

7:49 AM · Mar 2, 2021 · Twitter for Android

# Developers

#flutter

5:55 AM · Apr 26, 2020



Hicham Baghdoud  
@baghedoudhicham

btw #Flutter, I love you ❤️ ❤️

8:02 PM · Mar 3, 2021 · Twitter Web App



Daye  
@ib\_daye

I love flutter ❤️ ❤️  
2020 · Twitter for iPhone



Pawan Kumar  
@imthepk

I LOVE Flutter!  
- - - - -  
(\\_)/ ||  
(•ㅅ•) ||  
/ づ

3 · Danish Ansari  
@md\_is\_coding

Once you try #Flutter, either you  
love it or hate it with your heart,  
there's nothing in between.

7:49 AM · Mar 2, 2021 · Twitter for Android

# Developers

And Dart!  
- - - - -  
(\\_)/ ||  
(•ㅅ•) ||  
/ づ

#flutter

5:55 AM · Apr 26, 2020



Hicham Baghedoud  
@baghedoudhicham

btw #Flutter, I love you ❤️ ❤️  
8:02 PM · Mar 3, 2021 · Twitter Web App



Luis Ge Erre Te  
@LuisGrt

I love #Flutter

2:02 AM · Mar 2, 2021 · Twitter Web App



Daye  
@ib\_daye

I love flutter ❤️ ❤️

8,2020

· Twitter for iPhone



Pawan Kumar  
@imthepk

I LOVE Flutter!

(\\_/\\_) ||  
(•ㅅ•) ||  
/ づ



Danish Ansari  
@md\_is\_coding

Once you try **#Flutter**, either you  
love it or hate it with your heart,  
there's nothing in between.

3,

· Mar 2, 2021 · Twitter for Android



Saqlain  
@saqlain\_inc

Flutter is taking over app  
development. I love it. **#flutter**  
**#Flutter2**  
1:04 AM · Mar 4, 2021 · Twitter Web App

# Developers

#flutter

5:55 AM · Apr 26, 2020



Hicham Baghedoud  
@baghedoudhicham

btw **#Flutter**, I love you ❤️ ❤️  
8:02 PM · Mar 3, 2021 · Twitter Web App



Luis Ge Erre Te  
@LuisGrt

I love **#Flutter**

2:02 AM · Mar 2, 2021 · Twitter Web App



**Daye**  
@ib\_daye

I love flutter ❤️ ❤️

· Mar 8, 2020 · Twitter for iPhone



**Danish Ansari**  
@md\_is\_coding

Once you try **#Flutter**, either you  
love it or hate it with your heart,  
there's nothing in between.

· Mar 2, 2021 · Twitter for Android



**Saqlain**

@saqlain\_inc

Flutter is taking over app  
development. I love it. **#flutter**  
#Flutter2  
1:04 AM · Mar 4, 2021 · Twitter Web App



**Pratik Butani #AndroidDev**  
**#FlutterEngage**  
@pratik13butani

Since 2013, I was fell in love  
with **#Android** but Now you  
**#Flutter** ❤️

**#FlutterEngage #FlutterDev**  
**#love #FlutterWeb**



6:24 PM · Mar 3, 2021 · Twitter for Android



**Luis Ge Erre Te**  
@LuisGrt

I love **#Flutter**

2:02 AM · Mar 2, 2021 · Twitter Web App



**Pawan Kumar**  
@imthepk

I LOVE Flutter!

(\\_/\\_) ||  
(•ㅅ•) ||  
/ づ

And Dart!

(\\_/\\_) ||  
(•ㅅ•) ||  
/ づ

**#flutter**

5:55 AM · Apr 26, 2020

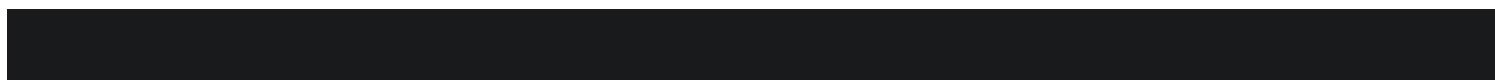


**Hicham Baghedoud**  
@baghedoudhicham

btw **#Flutter**, I love you ❤️ ❤️

8:02 PM · Mar 3, 2021 · Twitter Web App

Why do we  
love it?



Why do we  
love it?

Not just hot reload 

# Flutter:

What

Where

Why

and How

**What**  
is a modern  
declarative UI  
toolkit?

**What**  
is a modern  
**declarative UI**  
toolkit?

"imperative is how you do it, and  
declarative is what you want to do."



# Imperative

Fill the machine with 5 litres of water

Heat the water to 30 degrees

Dispense detergent

Spin for 30 mins

Drain water

Fill the machine with 2 litres of water

Spin for 15 mins

Wait for 5 mins

Play very annoying beep noise

# Imperative

Describes computation in  
terms of statements that

change state

# Declarative

I want my clothes washed on an Eco-Cycle

# Declarative

Describes what the  
program must accomplish  
in terms of the problem  
domain

# Imperative

```
override func viewDidLoad()
{
    super.viewDidLoad()
    var label = UILabel(frame: CGRectMake(0, 0, 200, 21))
    label.text = "I'm a test label"
    self.view.addSubview(label)
}
```

# Declarative

```
Widget build(BuildContext context) {
    return Text('Hello, Flutter Scotland!');
}
```

# Declarative

Describes what the  
program must accomplish  
in terms of the problem  
domain

# Domain = Abstraction



# Domain Specific Language

```
Widget build(BuildContext context) {  
  return Text('Hello, Flutter Scotland!');  
}
```

build Text Widget

Why do we  
love declarative?

---

Why do we  
*love declarative?*

---

Readability

# No side effects

---

# Imperative

Fill the machine with 5 litres of water

# Imperative

Fill the machine with 5 litres of water

Fill the machine with 5 litres of water

# Imperative

Fill the machine with 5 litres of water

Fill the machine with 5 litres of water

Fill the machine with 5 litres of water

# Imperative



# Declarative

I want my clothes washed on an Eco-Cycle

# Declarative

I want my clothes washed on an Eco-Cycle

I want my clothes washed on an Eco-Cycle



**Immutable  
state**



# How could you change the text on a Text Widget

```
Widget build(BuildContext context) {  
  return Text('Hello, Flutter Scotland!');  
}
```

# There is not a setter for the text on a Text Widget

```
Widget build(BuildContext context) {  
  return Text('Hello, Flutter Scotland!');  
}
```

There is not a setter for the text on a Text Widget

```
Widget build(BuildContext context) {  
  return Text('Hello, Flutter Scotland!');  
}
```

You would have to call build again

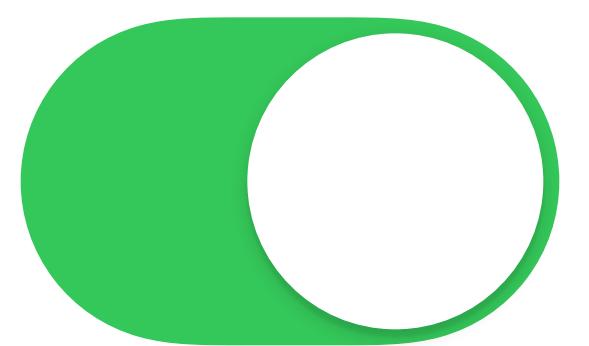
# There is a setter for the text on a UILabel

```
override func viewDidLoad()
{
    super.viewDidLoad()
    var label = UILabel(frame: CGRectMake(0, 0, 200, 21))
    label.text = "I'm a test label"
    self.view.addSubview(label)
}
```

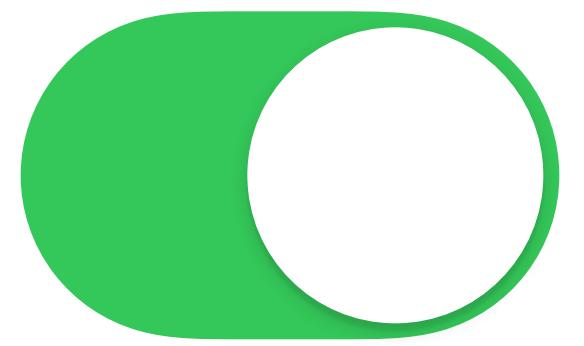
Widgets are  
*simple objects*

---

# Switch

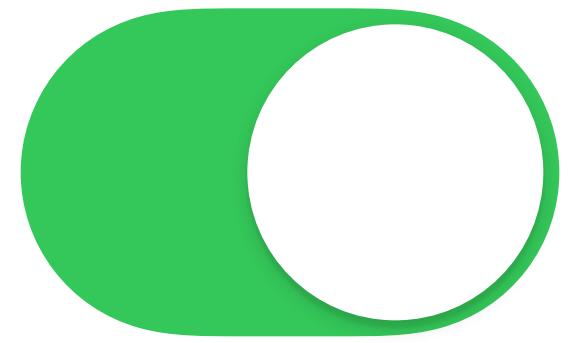


# Switch



```
let switchOnOff = UISwitch()  
switchOnOff.isOn = false  
switchOnOff.setOn(true, animated: false)  
switchOnOff.addTarget(  
    self,  
    action:#selector(switchStateChanged),  
    for: .valueChanged)  
  
@objc func switchStateChanged(sender UISwitch)  
{  
    print(sender.isOn ? "on" : "off")  
}
```

# Switch



```
let switchOnOff = UISwitch()  
switchOnOff.isOn = false  
switchOnOff.setOn(true, animated: false)  
switchOnOff.addTarget(  
    self,  
    action:#selector(switchStateChanged),  
    for: .valueChanged)  
  
@objc func switchStateChanged(sender UISwitch)  
{  
    print(sender.isOn ? "on" : "off")  
}
```

```
bool isSwitched = false;  
  
Switch(  
    value: isSwitched,  
    onChanged: (value) {  
        setState(() {  
            isSwitched = value;  
            print(isSwitched);  
        });  
    },  
);
```

*Single* source of truth

# Why do we *love*

---

No side effects,  
Immutable state,  
Simple Objects,  
Single source of truth?

# Why do we *love*

---

No side effects  
Immutability  
Simple objects,  
Single source of truth?

# Reliability

**What**  
is a modern  
declarative UI  
toolkit?

# Reactive

Reactive programming is  
programming with **asynchronous**  
**data flows.**

# Its not a new concept



# Who is responsible for the change?



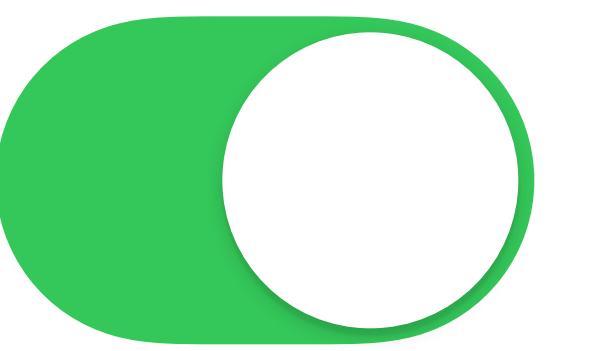
# Who is responsible for the change?



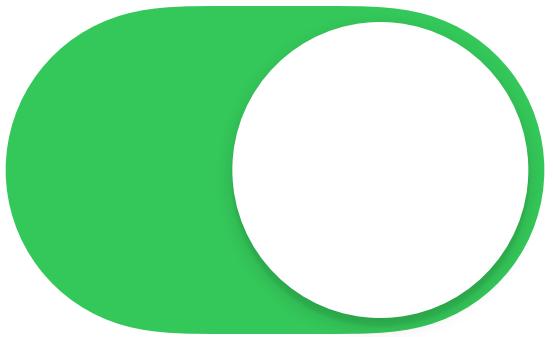
Cell A1 is responsible for reacting

Reactive programming is when  
components are responsible for  
reacting to change.

On



On



```
var label = UILabel()

override func loadView() {
    let view = UIView()

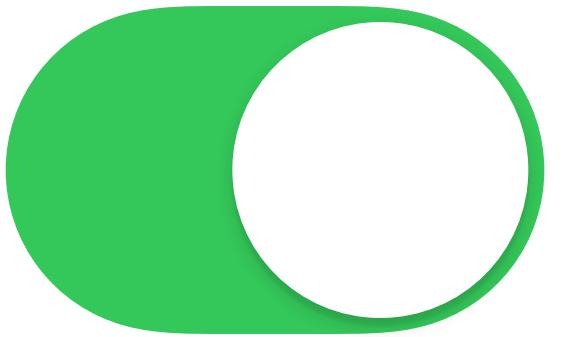
    label.frame = CGRect(x: 60, y: 5, width: 200, height: 20)
    label.text = "on";

    let switchOnOff = UISwitch()
    switchOnOff.isOn = true
    switchOnOff.addTarget(
        self,
        action: #selector(switchStateChanged),
        for: .valueChanged)

    view.addSubview(label)
    view.addSubview(switchOnOff)
    self.view = view
}

@objc func switchStateChanged(sender:UISwitch)
{
    label.text = sender.isOn ? "on" : "off";
}
```

On

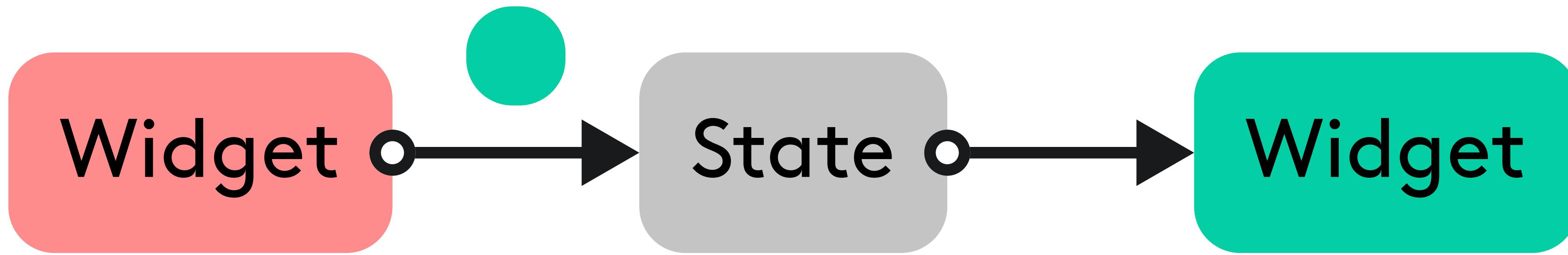


```
bool isSwitched = false;

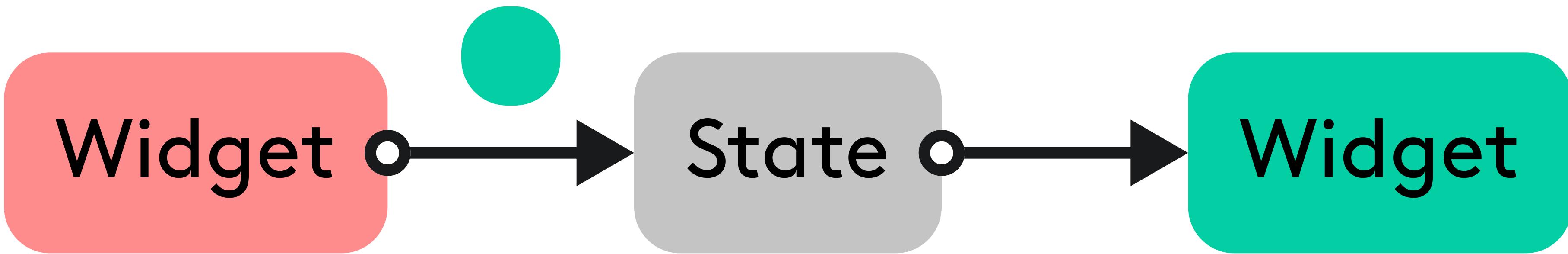
@Override
Widget build(BuildContext context) {
    return Scaffold(
        body: Row(children: [
            Switch(
                value: isSwitched,
                onChanged: (value) {
                    setState(() {
                        isSwitched = value;
                    });
                },
            ),
            Text(isSwitched ? 'on' : 'off'),
        ])
    );
}
```

# Why do we care about this?

# Why do we care about this?



# Why do we care about this?



## Uni directional data flow

# Why do we *love*

---

## Reactive & Uni directional dataflow?

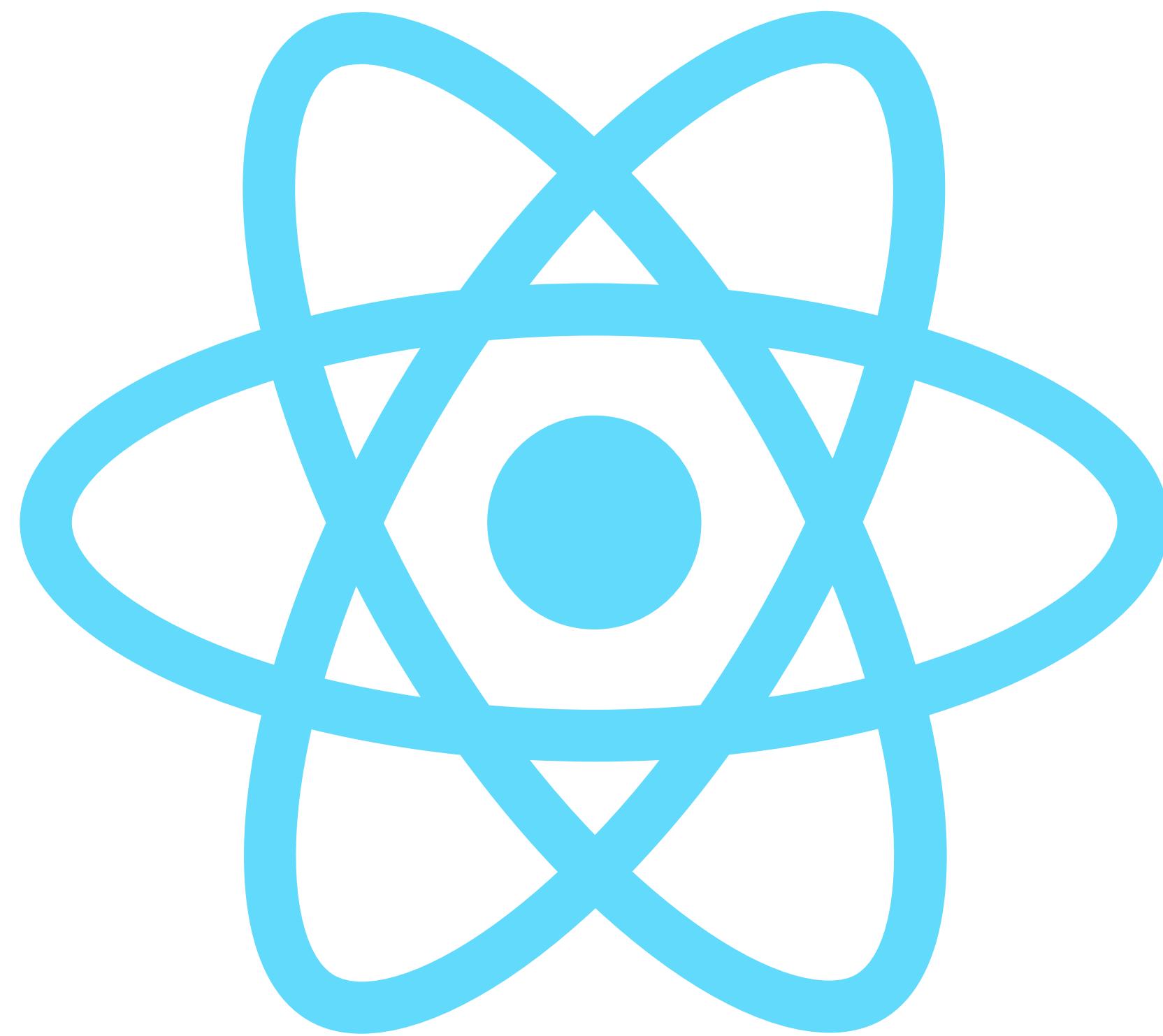
# Why do we *love*

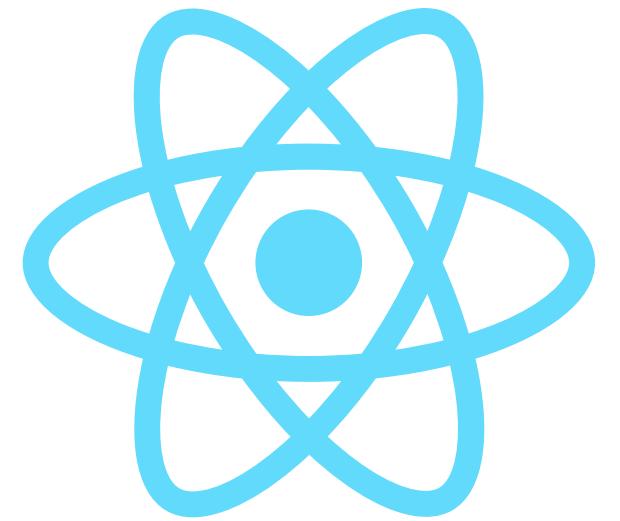
---

Reactive &  
Unpredictable  
**Predictable**  
Pattern?

**Where**  
did declarative UI come  
from?

# React

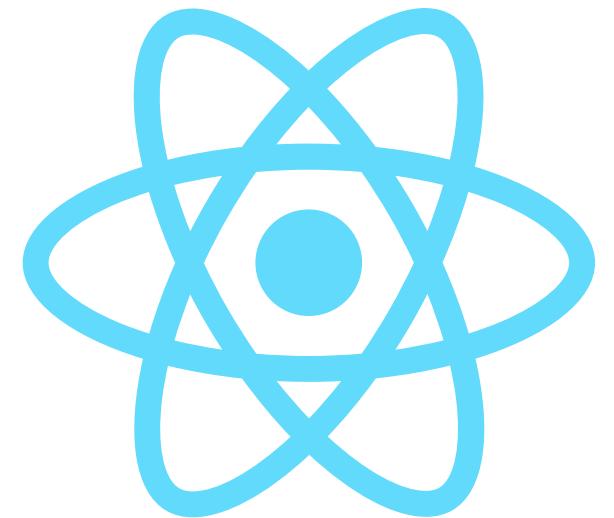




# React



Jordon Walke



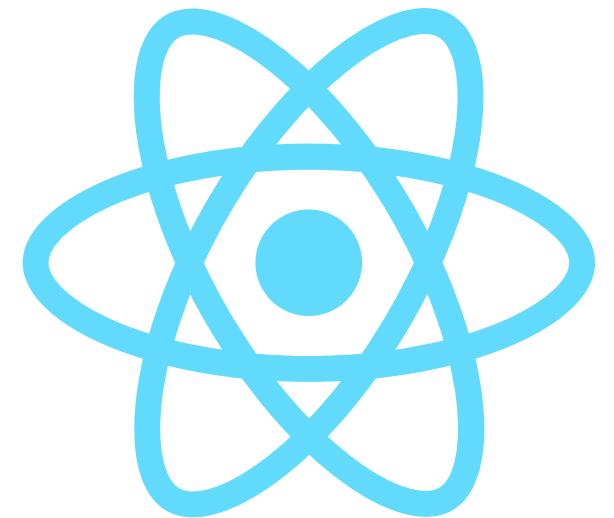
# React



Jordon Walke



XHP



# React

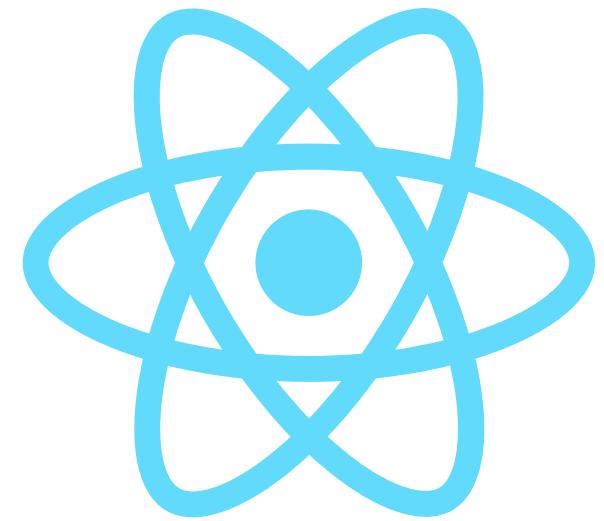


## Jordon Walke

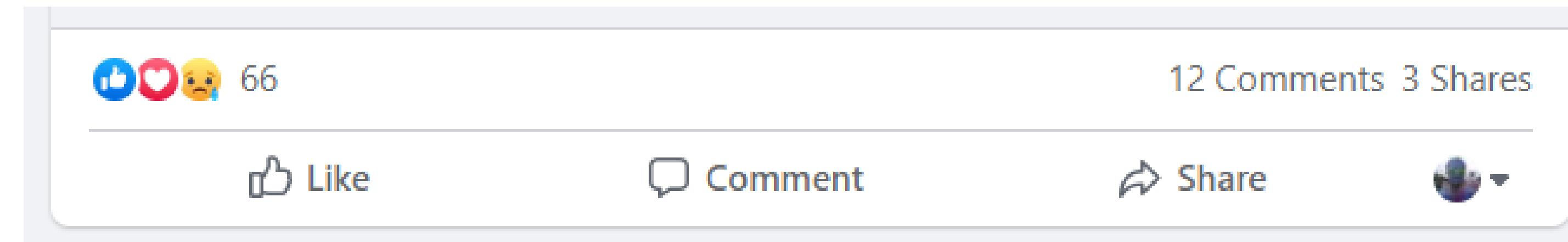


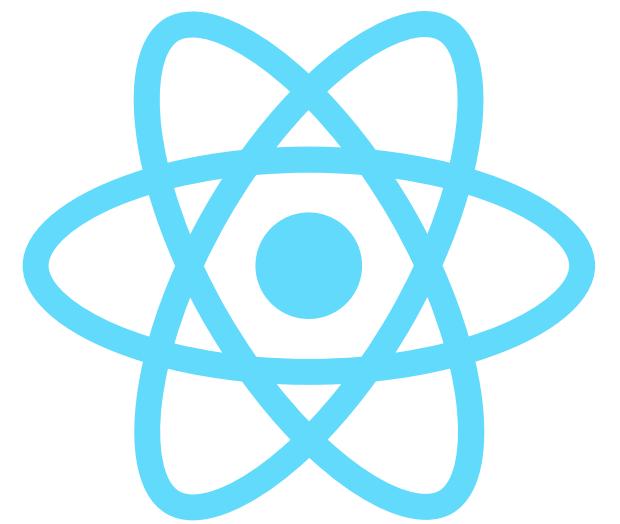
XHP





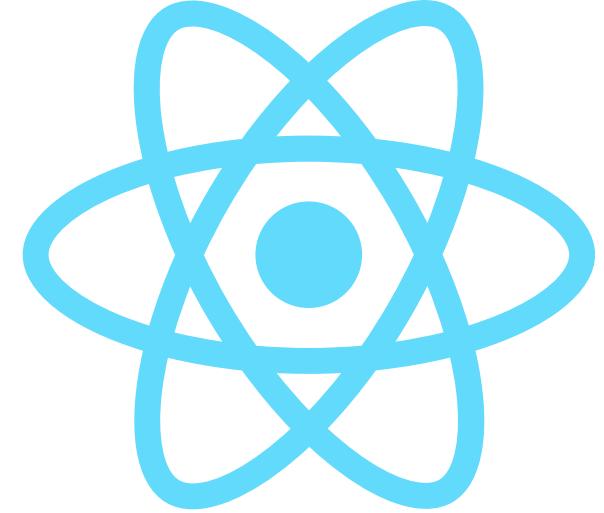
# React 2011





# React 2012





# React

- Declarative UI
- Building UI with React Components
- Uni-directional Dataflow
- Re-render entire application on  
data changes

```
class MyButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = {value: 0};
    this.buttonClicked = this.buttonClicked.bind(this);
  }

  buttonClicked(event) {
    this.setState({value: this.state.value+1});
  }

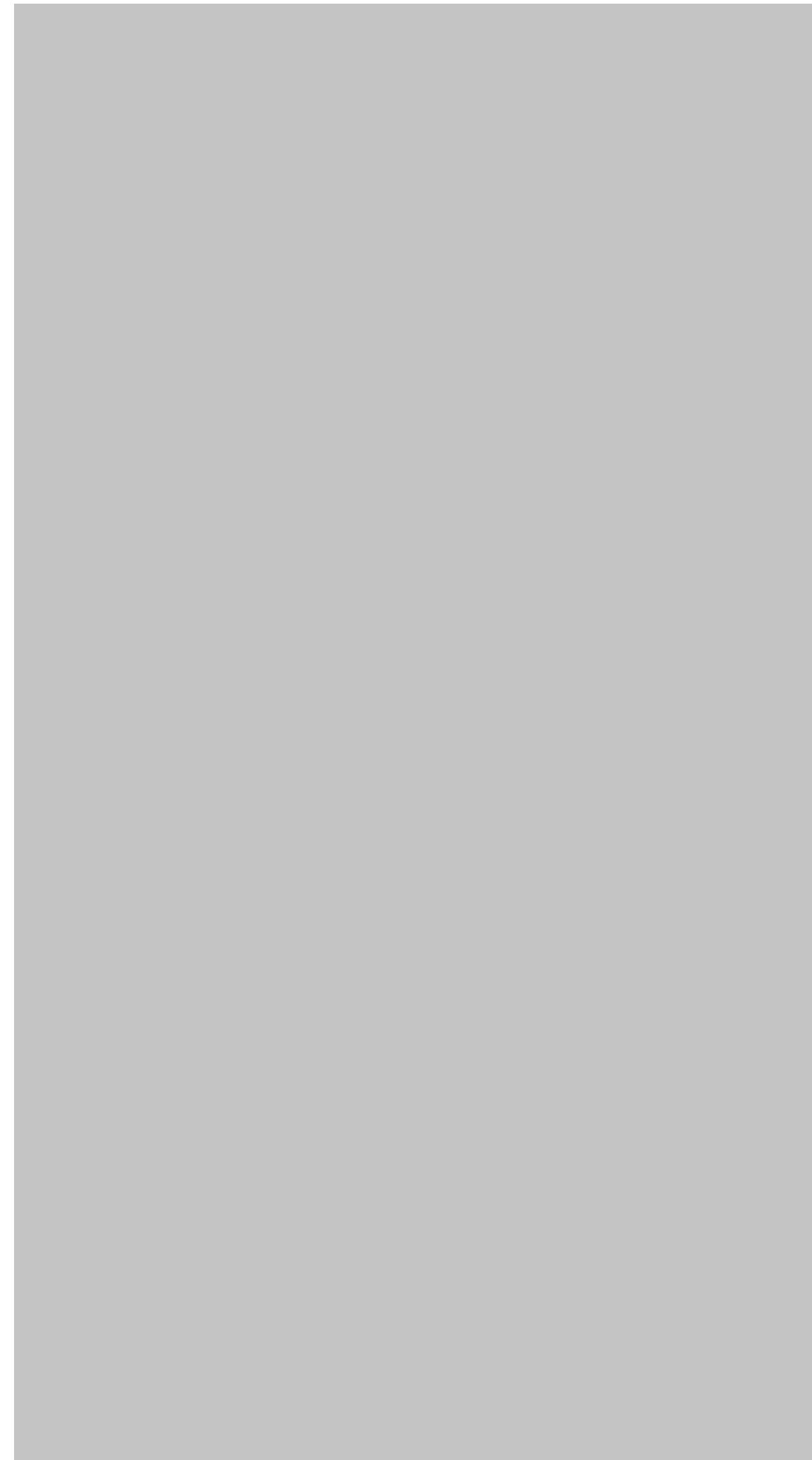
  render() {
    return (
      <div>
        <div>{this.state.value}</div>
        <button onClick={this.buttonClicked}>+</button>
      </div>
    );
  }
}

ReactDOM.render(
  <MyButton />,
  document.getElementById('root')
);
```

# 2011



# 2011



2014

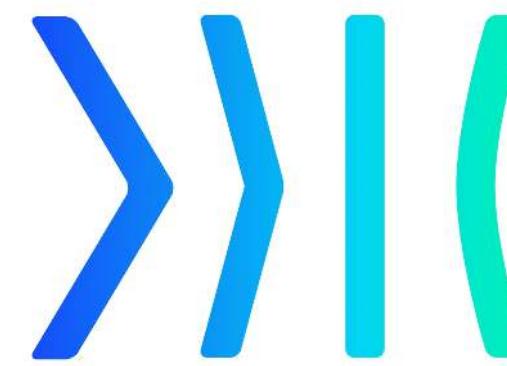
Async  
DisplayKit



# 2015

# Texture





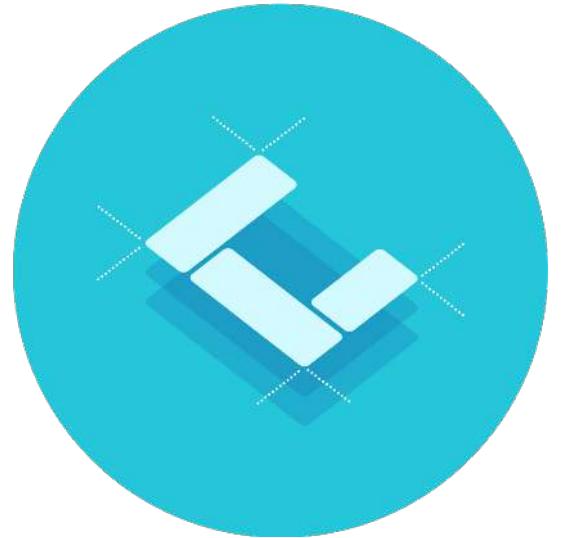
# Texture

- First Declarative UI toolkit for iOS
- Interactive UI
- Improving on UIKit for modern devices

2015

# Component Kit



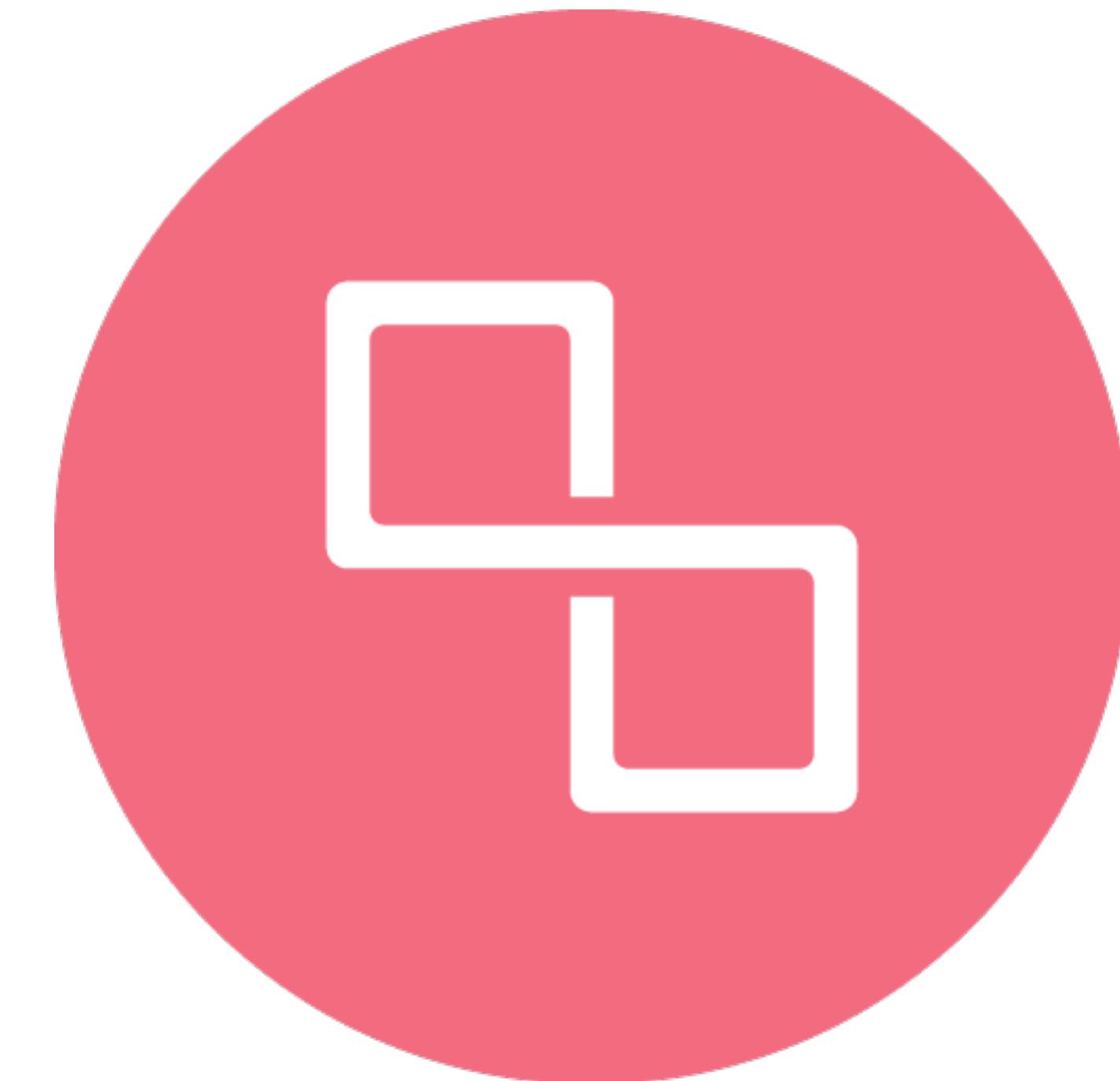


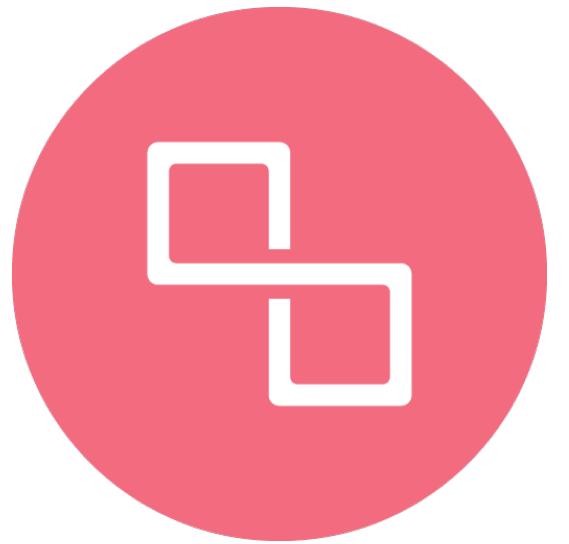
# Component Kit

- Functional and Declarative UI toolkit
- More of React's features

2017

Litho



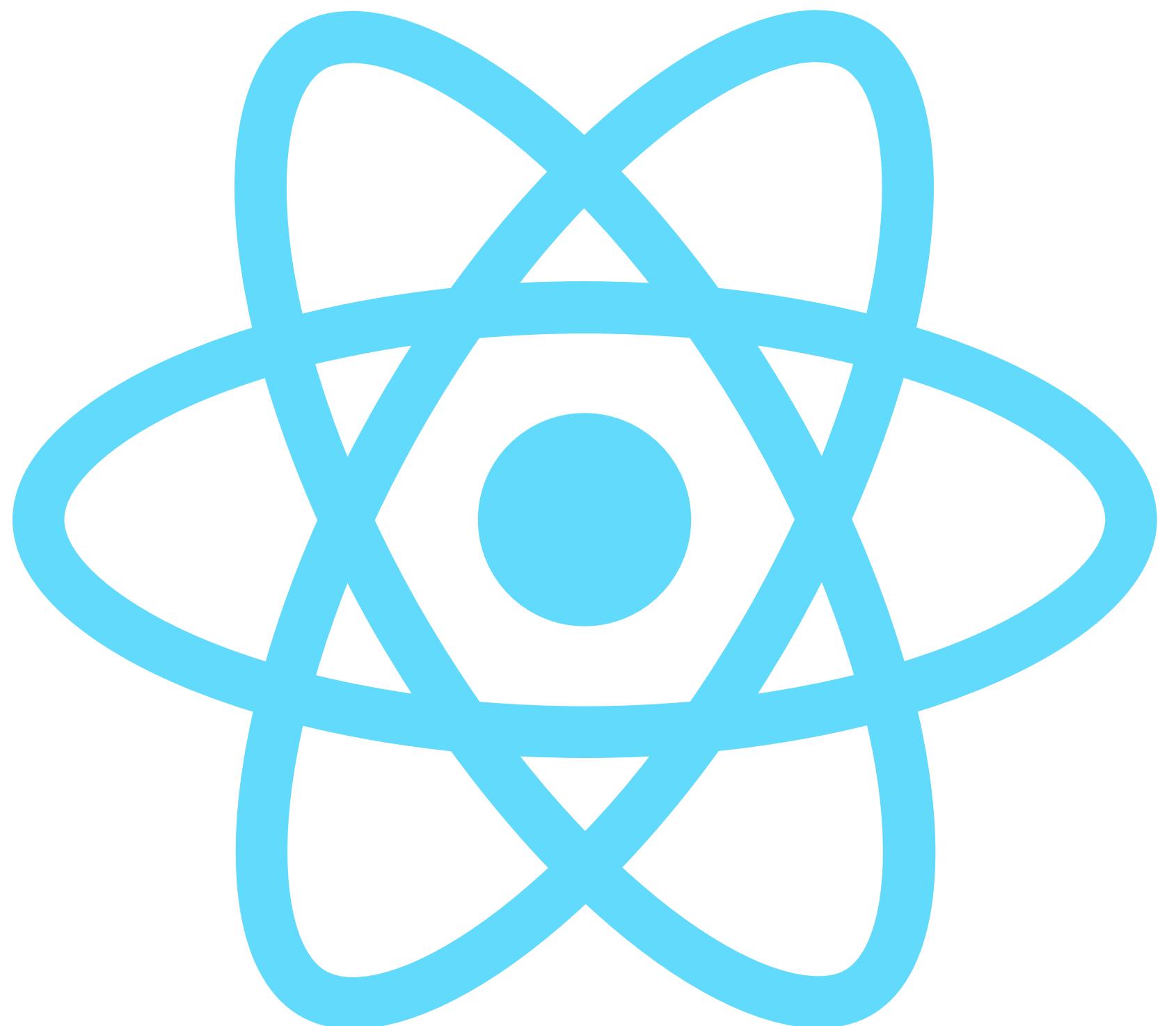


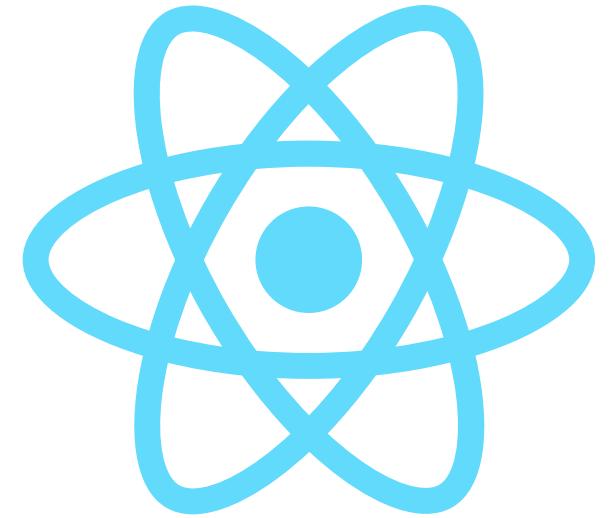
# Litho

- Declarative UI components
- More of React's features
- Improving on current UI toolkit for modern devices

2015

React-Native





# React-Native

- React's concepts to mobile
- Uses a JavaScript bridge to native code
- Highly popular

2015

Flutter





# Flutter

- No bridge to native code
- Skia render engine (no dependency on native controls)
- Performance was top priority

2019

Jetpack  
Compose



SwiftUI





# Jetpack Compose / SwiftUI

- Great potential to bring native UI toolkits into a modern approach
- Focused only on their ecosystem

Why do we love Flutter?

# Why do we love Flutter?

Inspired  
by the best solutions

**Why**  
are we using declarative  
UI toolkits?

# Old Devices



# Old UI Kits

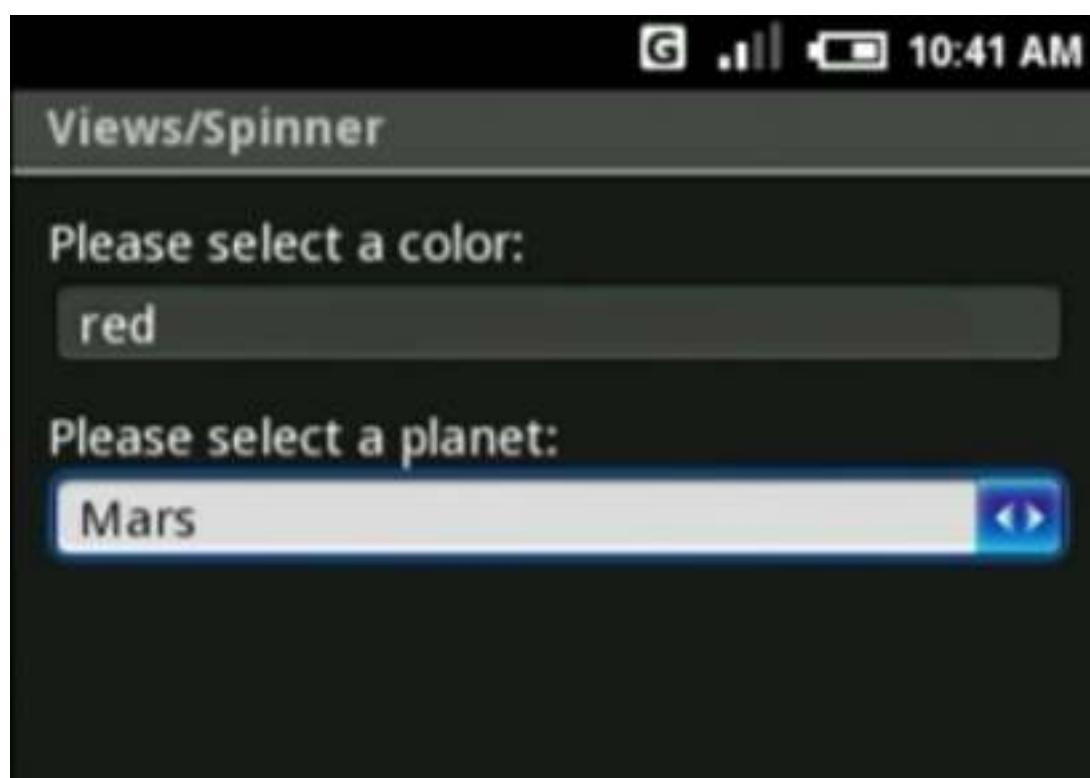


android

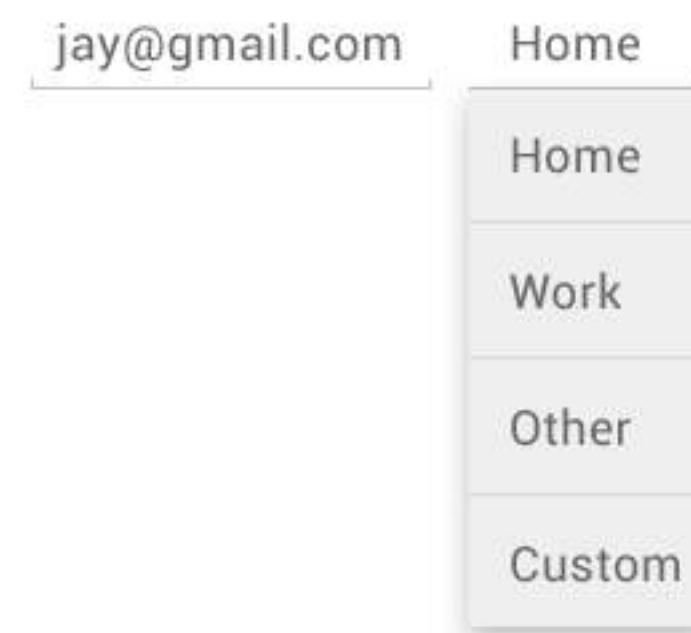
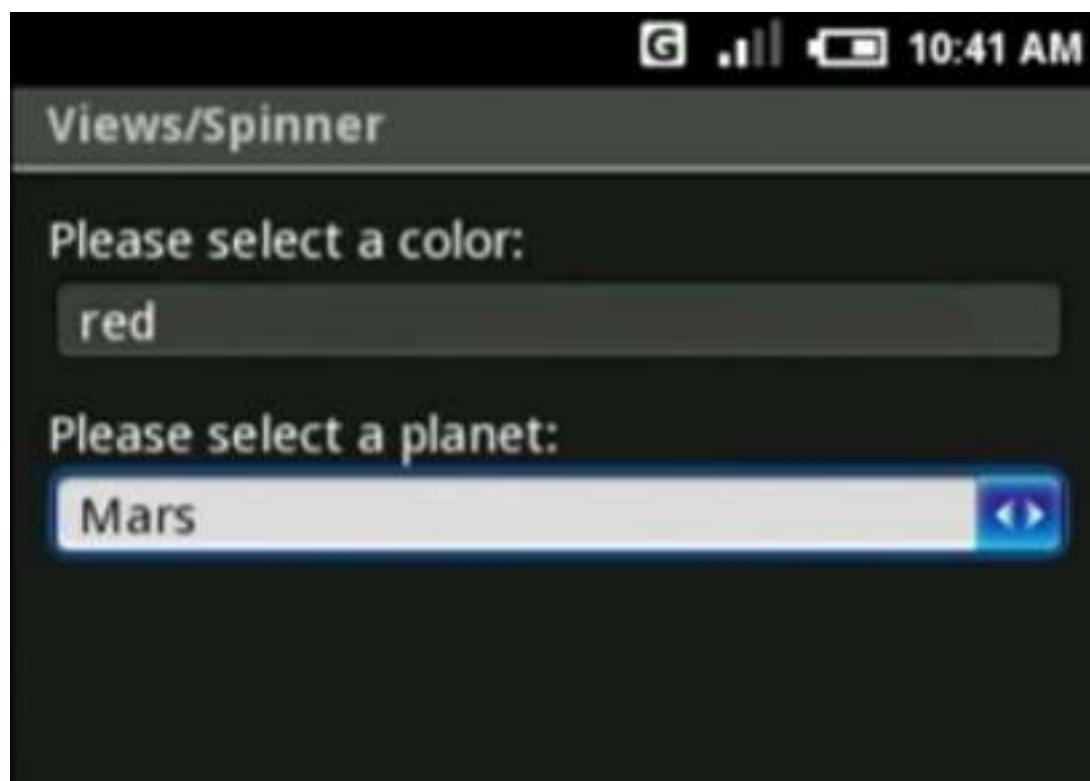


# API regret - Spinner

# API regret - Spinner



# API regret - Spinner



Why do we love Flutter?

# Why do we love Flutter?

- built for modern hardware
- considered UI toolkit

**How**  
does this impact our  
development?

# Paradigm Shift

Imperative

Declarative      Functional

# Paradigm Shift

Changes how you  
solve problems & think

# Why do we *love* Flutter?

---

# Why do we *love* Flutter?

---

Gives us new tools  
to solve problems

---

# Recap

What

Where

Why

and How

# Recap

What

Where

Why

and How

Readable

Reliable

Predictable

# Recap

What  
Where  
Why  
and How

*Inspired*  
by the best  
solutions

# Recap

**What**  
**Where**  
**Why**  
and **How**

built for  
*modern*  
devices

# Recap

What

Where

Why

and How

new tools to  
*solve problems*

**Next time some one asks you:**

Next time some one asks you:

Why do you love  
Flutter?

---

# I do know why I love Flutter



## Let me tell you the facts!

# Questions?

@b09913

iainsmith.me