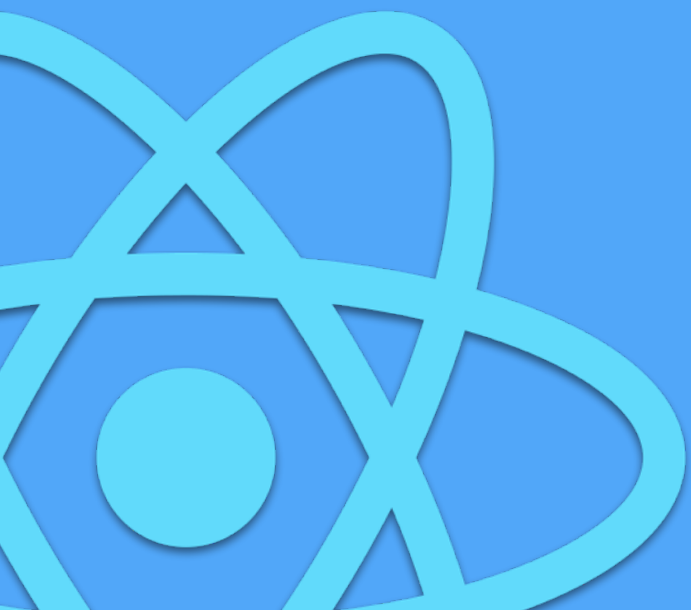


React+Flux in Action

實戰經驗分享



Jeremy Lu

@thecat | jeremy@pubulous.com

Founder and builder,

Lovelyreader | Pubulous | Visualmarks | Docknote

開發者

十年前後端開發經驗

講求實戰、其餘免談

顧問服務

社群媒體

証券業

房地產業

出版業

創辦人

Lovelyreader

Pubulous

Visualmarks

Docknote

React/Flux 是什麼？

- **Facebook** 於 2013 年中開源
- **React** 是內建 **VDOM** 的輕量顯示套件
- **Flux** 管理 **Data + Flow**
- **React** 負責 **view**，**Flux** 負責 **model**，兩者合作無間

誰在使用？

- Facebook
- Instagram
- Github
- Firefox
- Yahoo Mail
- ...

React 亮點

- 元件式架構
- **VDOM** 虛擬頁面模型
- **JSX** 標記語法

元件式架構

- 捨棄 **template** 直接用 **js** 控制版面邏輯
- 因為 **markup** 與背後掌管它的程式本來就是一體的
- 硬要分離反而削弱兩邊的力量
- 想想 **Handlebar** 裏要寫邏輯處理的痛苦

```
<EditorCore
```

```
  isFocusEditor={item.focused}
```

```
  content={item.content ? item.content :  
    'default texts'}
```

```
  ref={item.uid}
```

```
  onChange={this.handleChange}
```

元件可層層組合

- 以大化小，個個擊破
- 方便分工合作，最後再組合起來
- 小元件組合後可構成新功能

D file name 2

file name 1

file name 2

file name 333

222_1

222_2

222_3

+ -

+ -

[SECTION-- 222_2 --3]

- aaaa

- A Google employee is getting a week off work to spend with his daughter, thanks to an impassioned

- b1

- b2

- c1

- c2

- a2

- bbbb

- cccc

- ddddd

- eeeed

- ffff

- ###END###

- bbb

- eeeed

▶ file name 1
file name 2
file name 333

222_1

222_2

222_3

+ -

+ -

[SECTION-- 222_2 --3]

- aaaa

- A Google employee is get
off work to spend with
thanks to an impassioned

- b1

- b2

- c1

- c2

- a2

- bbbb

- cccc

- dddd

- eeed

- ffff

- ###END###

- bbb

- eeed



Tree



ToolBar

```
<FileList>
```

```
<Tree />
```

```
<ToolBar onAdd={this.add} />
```

```
</FileList>
```

In details

完整生命週期指令

componentWillMount()

componentDidMount()

shouldComponentUpdate()

render()

...

元件俱高度封裝性

- 可將 javascript/css/html 三位一體封裝於個別元件內
- 方便重覆使用

VDOM

- 虛擬文件模型在記憶體內進行高效能運算與批次更新
- 有效減少對 DOM 的操作因而加快反應速度
- 支援多種 backend，例如 DOM, Canvas, SVG 與 WebGL (famo.us)

JSX

- 類似 **HTML**，可直接寫在 **js** 內的標記語言
- 完美結合模板與程式語言的優點
- 設計師也看的懂，可以幫忙改
- 有工具可 **jsx** \leftrightarrow **html** 互轉

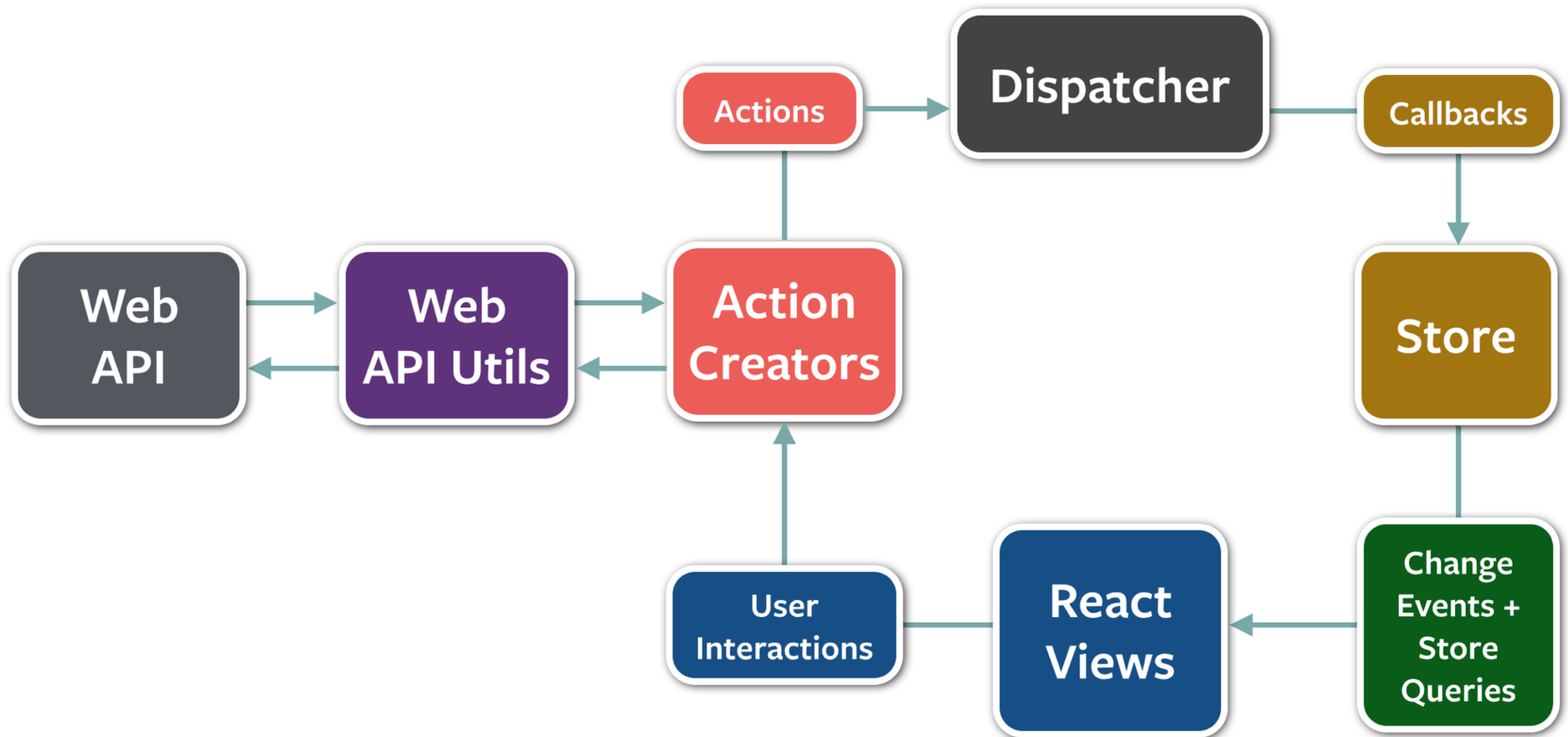
```
<div>Hello {this.props.name}</div>;
```

或

```
React.DOM.div(null, "Hello ",  
this.props.name);
```

Flux 亮點

掌管 Data + Flow



兩個基本概念

- 一律重繪
- 單向資料流

一律重繪

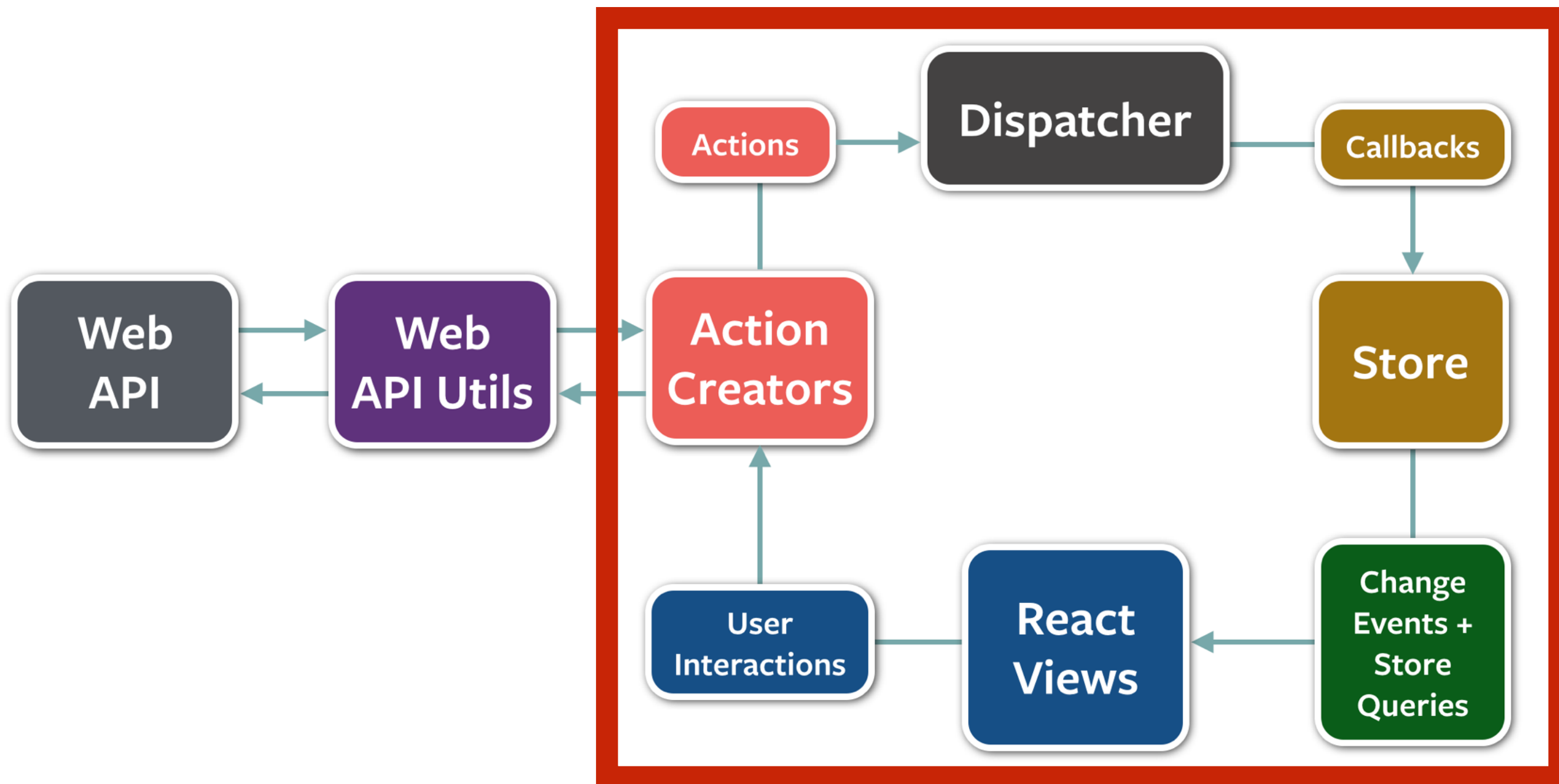
- **model** 是描述程式狀態的唯一真理
- 只要 **model** 有變動，一律重繪整個 **ui**，不需個別更新 **DOM**
- 簡言之，每次都當第一次繪圖，只是給定的初始值不同
- 所以開發流程就是一直做元件，然後餵 **model** 資料進去，畫面就保證正確

單向資料流

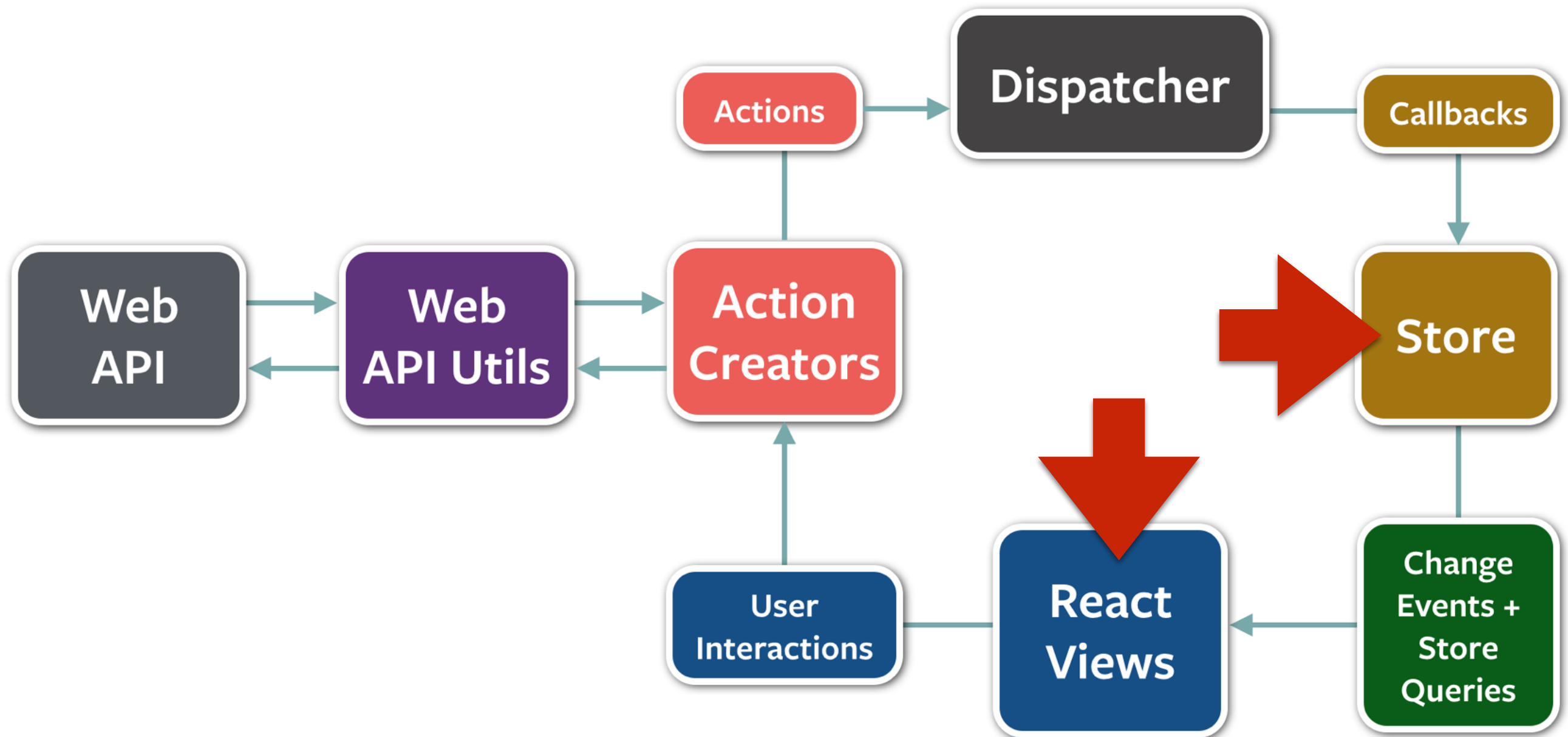
- **Model** 只讀不寫，是唯一真理之所在
- 讓系統複雜度大幅降低，程式長再大也不怕
- 一舉排除過往 **MVC** 掛一漏萬的問題，

***唯一真理 = Single Source of Truth**

掌管 Data + Flow

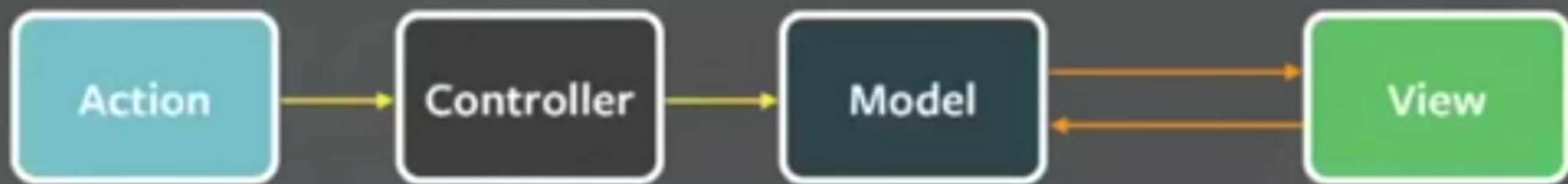


掌管 Data + Flow



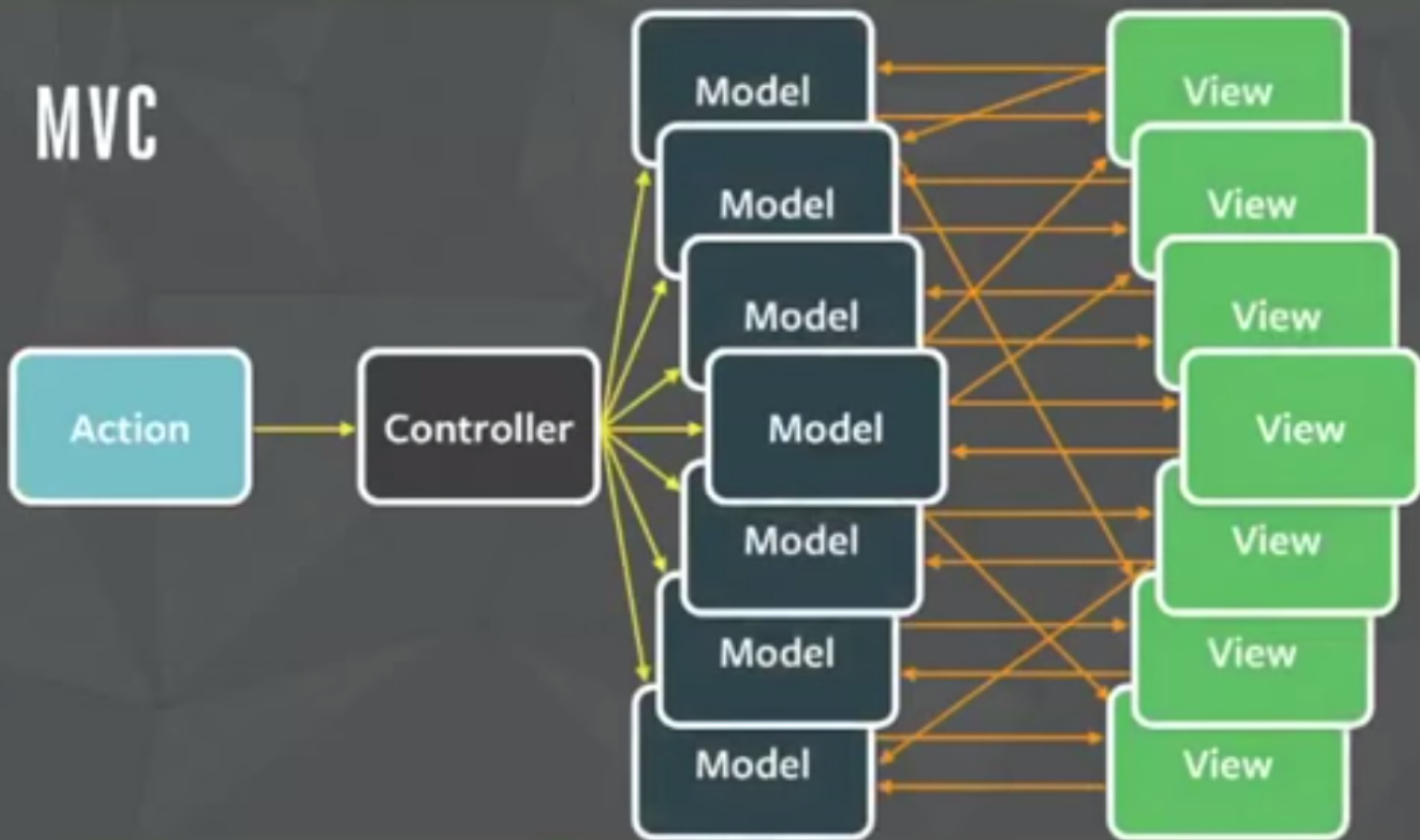
以往手法

MVC

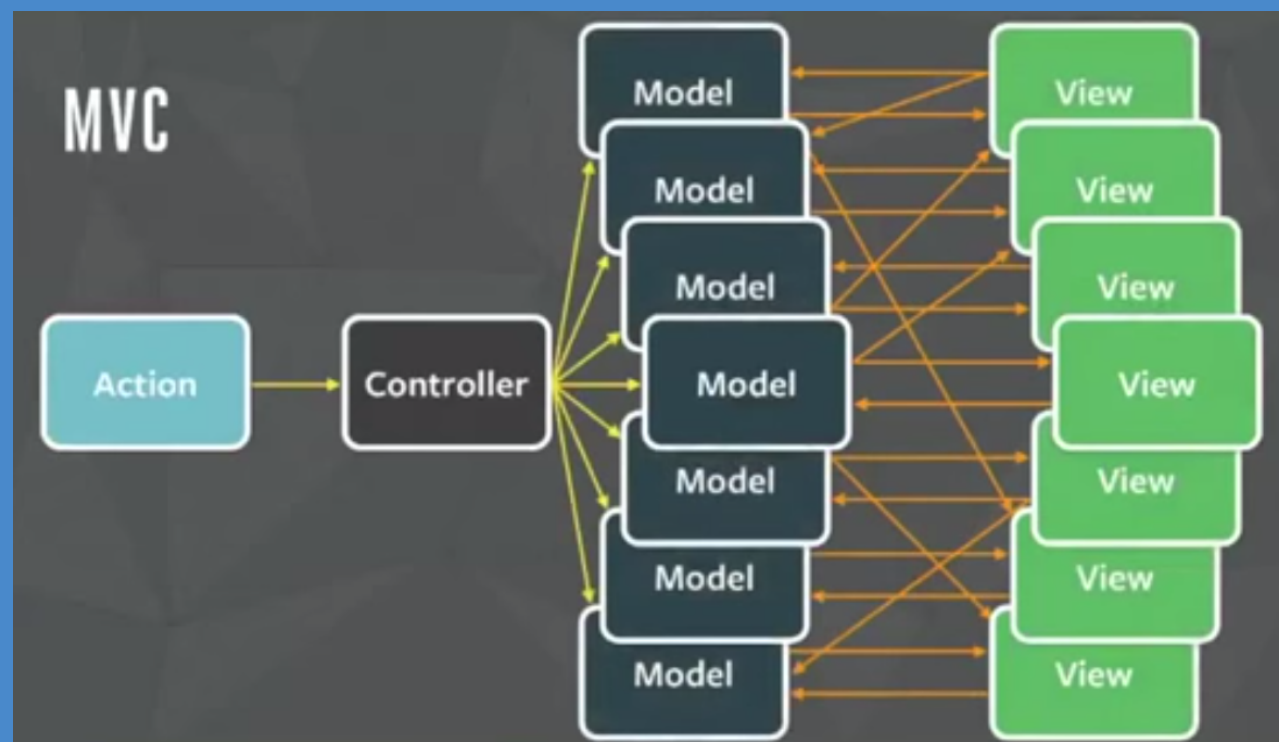
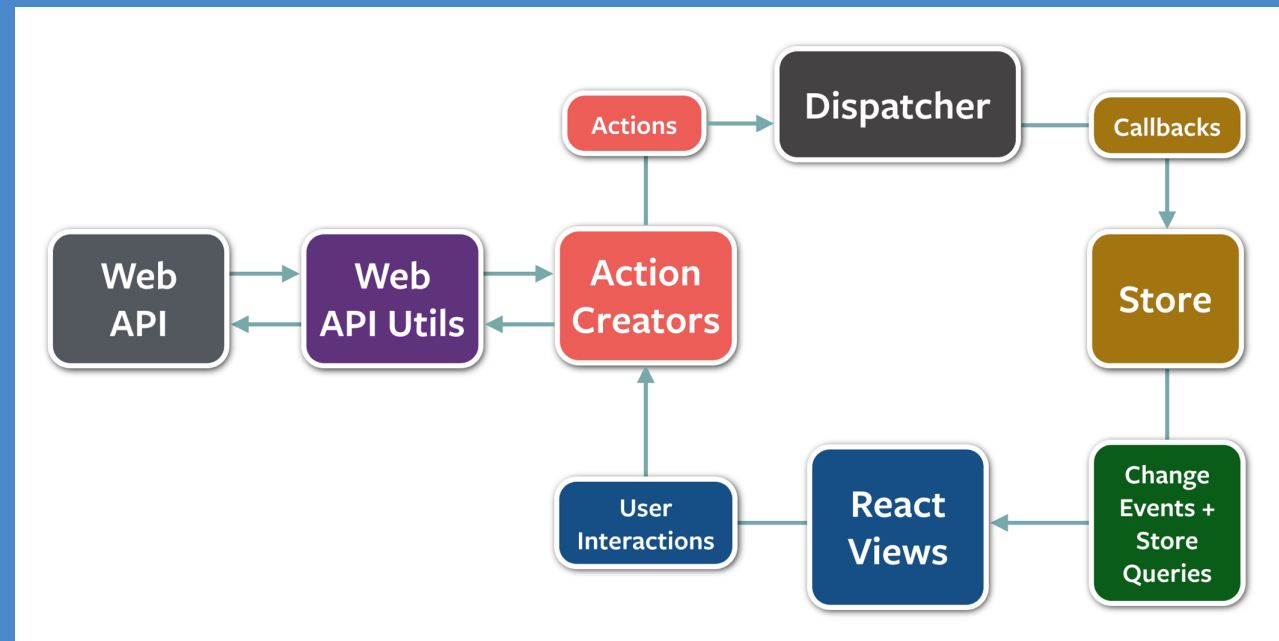


以往手法

MVC



兩相比較



Views -> Action Creators -> Dispatcher -> Stores ->
Views -> Action Creators -> Dispatcher -> Stores ->
Views -> Action Creators -> Dispatcher -> Stores ->
Views -> Action Creators -> Dispatcher -> Stores ->
Views -> Action Creators -> Dispatcher -> Stores ->
Views -> Action Creators -> Dispatcher -> Stores ->
Views -> Action Creators -> Dispatcher -> Stores ->
Views -> Action Creators -> Dispatcher -> Stores ->
Views -> Action Creators -> Dispatcher -> Stores ->
Views -> Action Creators -> Dispatcher -> Stores ->
Views -> Action Creators -> Dispatcher -> Stores ->



解決了哪些問題？

- 人才難覓
- 技能難學
- 框架複雜
- 手法岐異
- 引誘犯罪

人才難覓

→ **angular** 難學，會的人本就不多，高手更稀有

- 熟悉 **javascript/jquery** 並有一兩年 **web** 開發經驗者即是潛在招募對象
- 如果原本就會 **angular/backbone/ember** 等框架者更是人中之龍

技能難學

→ 概念複雜，新手至少要半年才能獨當一面

- 整體概念單純好學易懂
- 學習曲線極平滑，而且之後沒有隱藏陷阱
- 只要熟 **javascript** 大約兩天即可上手實戰



My Feelings About AngularJS Over Time

Angularjs 學習曲線



React/Flux 學習曲線

框架複雜

→ 除錯與客製化皆不易

- **React** 只做好元件系統，單純而不易出錯
- **Flux** 只是一個手法概念，容易理解，也好調整

引誘犯罪

→ 設計複雜容易引人犯錯，導致維護成本上升、客戶滿意度下降

- 程式結構單純，本不易出錯，長期維護成本就降低

- 當出錯時問題很容易一分為二，不是 **model** 就是 **view** 有問題，可快速修復

總的來說

- 讓系統架構更單純穩定，不易出錯且好維護
- 總體擁有成本就會下降
- 開發者寫的開心，臉上洋溢幸福笑容 **XDDD**

心法

專案最後都會搞大，如果一開始
手法單純，將來還有機會控制，
反之，則必然悲劇收場。

我的血淚教訓史

Case Study

Docknote

幫助工程師快速蒐集、整理與分享個人技術筆記

D file name 2

file name 1

file name 2

file name 333

222_1

222_2

222_3

+ -

+ -

[SECTION-- 222_2 --3]

- aaaa

- A Google employee is getting a week off work to spend with his daughter, thanks to an impassioned

- b1

- b2

- c1

- c2

- a2

- bbbb

- cccc

- ddddd

- eeeed

- ffff

- ###END###

- bbb

- eeeed

歷程

- 第一版使用 **angularjs** 開發
- 一半時間在與框架本身戰鬥(學習、除錯、熟練)
- 耗時四個月完成 **MVP**
- 框架本身太多黑魔法一旦出事很難找問題源
- 雙向綁定造成資料流向錯綜複雜控管不易
- 頁面效能不佳(**60FPS** 成為一種**謠傳**)
- 記憶體耗用極大(**dirty checking** 不穩定又佔用大量資源)

歷程

- 第二版使用 **react/flux** 開發
- 兩星期就重寫完成之前的進度
- 之後開始忍不住添加各種功能 (不要學
- 頁面反應快，意外問題減少
- 還能 **server pre-render**

導入 React/Flux 後的 具體好處

好教、易學、快上手

- 原本熟悉 js 的工程師大約 2 天即可學完整套概念
- 即戰力立即體現
- 終於不再怕缺人了

加快開發速度

- 程式可切成許多獨立小元件，每個人分頭開發，最後再組合
- 框架本身單純，撰寫元件容易且速度快
- 每個人手法一致，容易除錯與相互支援

提升程式穩定度

- 單向資料流 + 永遠重繪 兩大原則讓程式複雜度大幅降低
- 等於內建防呆措施因此不易製造 **bug**
- 更少的 **bug** 等於更早下班或開發更多新功能

降低維護成本

- 程式單純因此不易出錯, 自然也不需維護
- 行為高度可預測, 因此就算出錯也很容易排除
- 通常不是 **model** 就是 **view** 有問題

提升客戶滿意度

- 因為產品不易出包
- 又能即時交貨
- 客戶沒抱怨
- 自然滿意度++

完美實現 full stack 流程

- 從頭到尾都是 javascript
- 突顯 isomorphic javascript 優勢
- 與 node.js / Express 搭配良好
- 與 real-time API server 搭配良好

其它

- Accessibility (aria, role...)
- Immutable model
- Pre-rendering
- Server-side templating
- Gulp/Browserify
- Jest testing

晚上睡的好

中文社群 reactjs.tw

<https://www.facebook.com/groups/reactjs.tw/>

有問必答喔~

References

- **Future of MVC frameworks - <http://goo.gl/cUq1bh>**
- **React introduction - <http://goo.gl/7vbhl1>**
- **Flux introduction - <http://goo.gl/SZ10b2>**
- **Single source of truth - <http://goo.gl/FuwXv>**



Vojta Jína
@vojtajina



I like [@reactjs](#) a lot. Thinking how to make [@angularjs](#) more functional and yet preserve the dev experience. Any ideas? Please let me know.

1:35 AM - 29 Jun 2014

42 RETWEETS **45** FAVORITES



lead developer of angularjs

Questions?

jeremy@pubulous.com