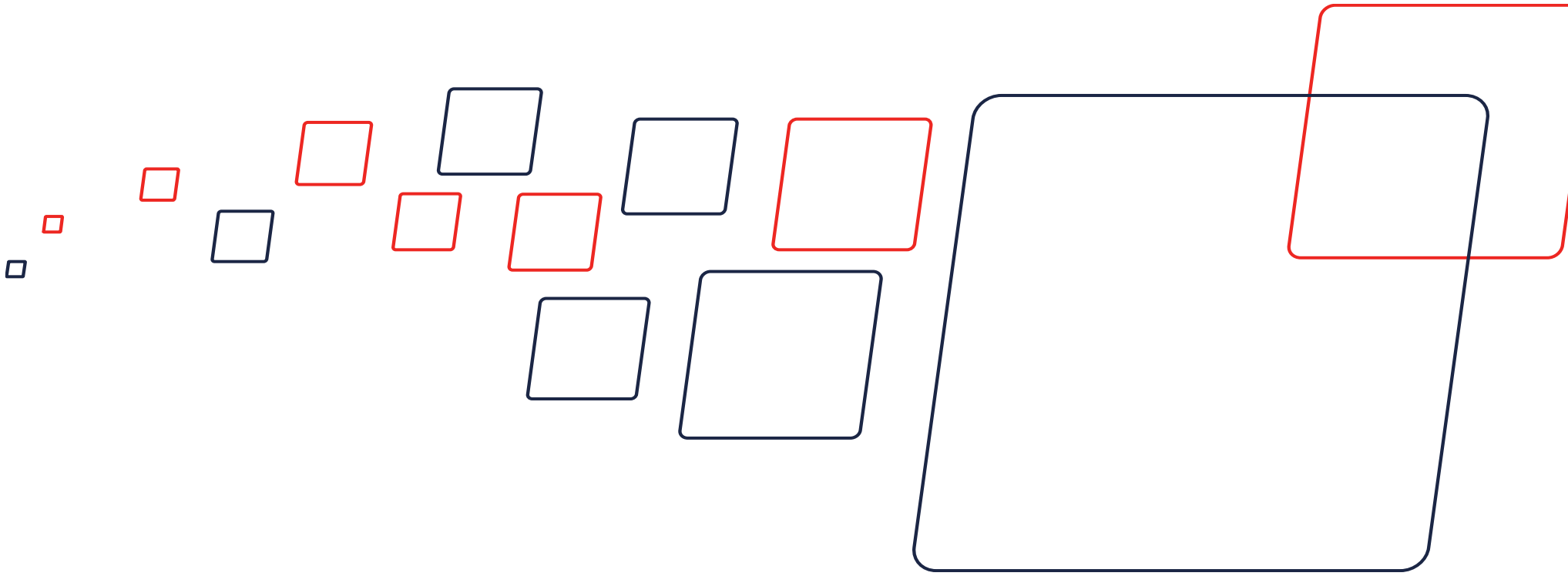
digiterre agility

Modern javascript development

Evaluating AngularJS and React patterns and practices



Introduction – Digiterre Agility

- We provide bespoke software solutions to three sectors:
  - Capital Markets, Energy Trading and Digital Marketing / AdTech

- We also provide technical consulting in the following ways:
  - Agile adoption
  - Framework & technical training
  - Architecture design & build
  - Software testing strategy
  - UX requirements

- We have been delivering software for 15 years across a wide client base

M&G INVESTMENTS
rlam royal london asset management
Man
SECURIS INVESTMENT PARTNERS
RWE The energy to lead
MILLENNIUM GLOBAL INSTITUTIONAL CURRENCY INVESTMENT MANAGER
TRAYPORT
GLENCORE INTERNATIONAL AG
EDF
e·on Energy Trading
GAZPROM MARKETING & TRADING

Agenda

digiterre agility

- Will cover:
  - Introduction to Javascript development landscape in 2016
  - Build an example application with different frameworks to better understand the differences and pros and cons
  - Deep code samples as well as slide material
  - Future view into this space

- Won't cover:
  - Introduction to Javascript / HTML5 development
  - CSS / SASS, design-related subject matter
  - Isomorphism / deep diving into any one framework or approach
  - Redux – we're using Alt
  - ES6 syntax

Evolution of the web

Early HTML specs
XML
Early Flash
Basic cookies / auth
SSL

Early Safari
Firefox hits 1.0
SVG, Early Canvas

07: iPhone released
08: HTML5 finalised
(geolocation, etc)
08: Chrome 1.0
09: Angular released

CSS3 improvements

Deep system APIs, Audio,
Full screen, etc
CSP
Electron / Containers /
OpenFin

2016: Building client-side web applications

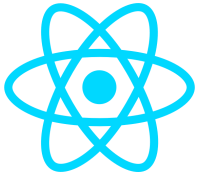
digiterre agility

- Large selection of javascript frameworks and components to use
  - >15 highly used MV* frameworks to choose from
- Nodejs & package management has driven web forward
- Electron and other app container packages replicate desktop apps
  - Including elevation & many other once-thought “desktop only” features

- Landscape accelerating still, but key patterns settled
  - This makes framework selection still very difficult as switching is costly

- This presentation will delve in to two potential choices:

AngularJS

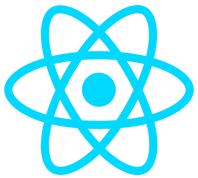


React

Not quite like-for-like…

AngularJS

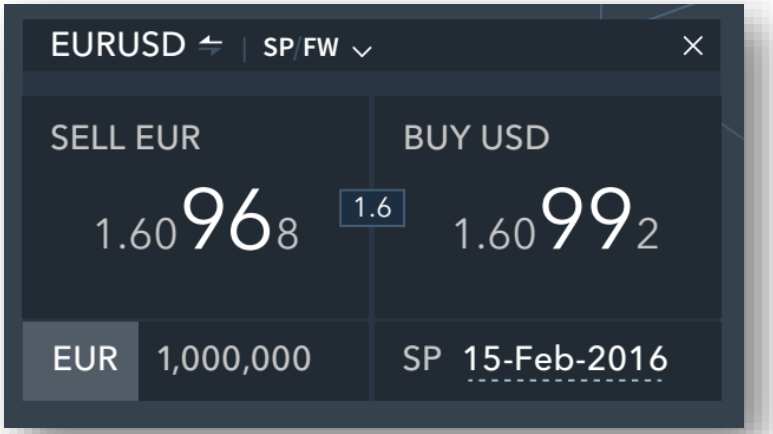
- Provides an entire framework for developing javascript applications
- Provides conventions based on architectural patterns
- View engine, data flow, routing built in



React

- Provides a library for building user interfaces
- Componentises views into logical sections
- Suggests but does not enforce a unidirectional data flow

Sample app: FX Trade Tile

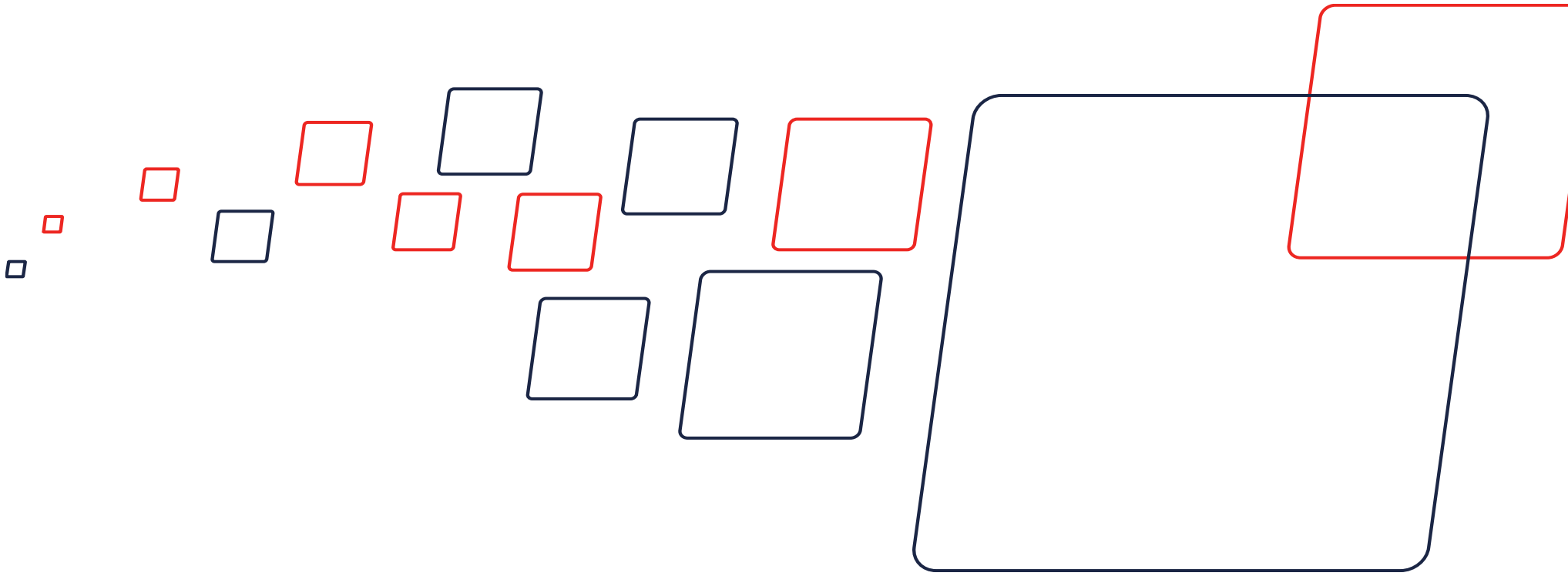


- The objective is to build a simple functional FX trade tile in HTML with Javascript

- The functionality will be simple, supported by a pre-built back-end:
  - Real-time streaming rates (provided via websockets)
  - Ability to execute a trade on single click (submission via API)

Example Digiterre UX flow

Quick view into the Digiterre UX flow



Basic application principles

1. Communication:
   - Trade booking
   - Live rates

2. UI / Interaction:
   - Live rate updates
   - Live spread updates + calculation
   - Trade on click functionality
   - Basic highlighting on hover
   - Notification of trade booking

Trading / Rates Service
REST API integration
Web Socket integration

UI requirements
Real-time UI – DataBinding?
Hover interaction
Fairly Rich UI experience
Notification / toast service

Vanilla JS / JQuery

digiterre agility

- Remember – frameworks aren't always needed
  - Small applications can easily be written without a (sometimes) large and cumbersome framework or library.
- Ideal for initial scratch-pad or spike
  - Very quick to get started
  - Testing UI or back-end
- If / when necessary, can evolve into a larger application

Demo: Vanilla JS implementation

Simple starts don’t mean simple ends

- Increased complexity
  - If we wanted to add another trade tile, we may end up replicating code
  - In order to build this properly, custom implementations of design patterns would be needed
- No convention or build strategy
  - Harder to add developers to the project
- No adherence to architectural design patterns
- These are the reasons JS evolved into making use of frameworks and libraries

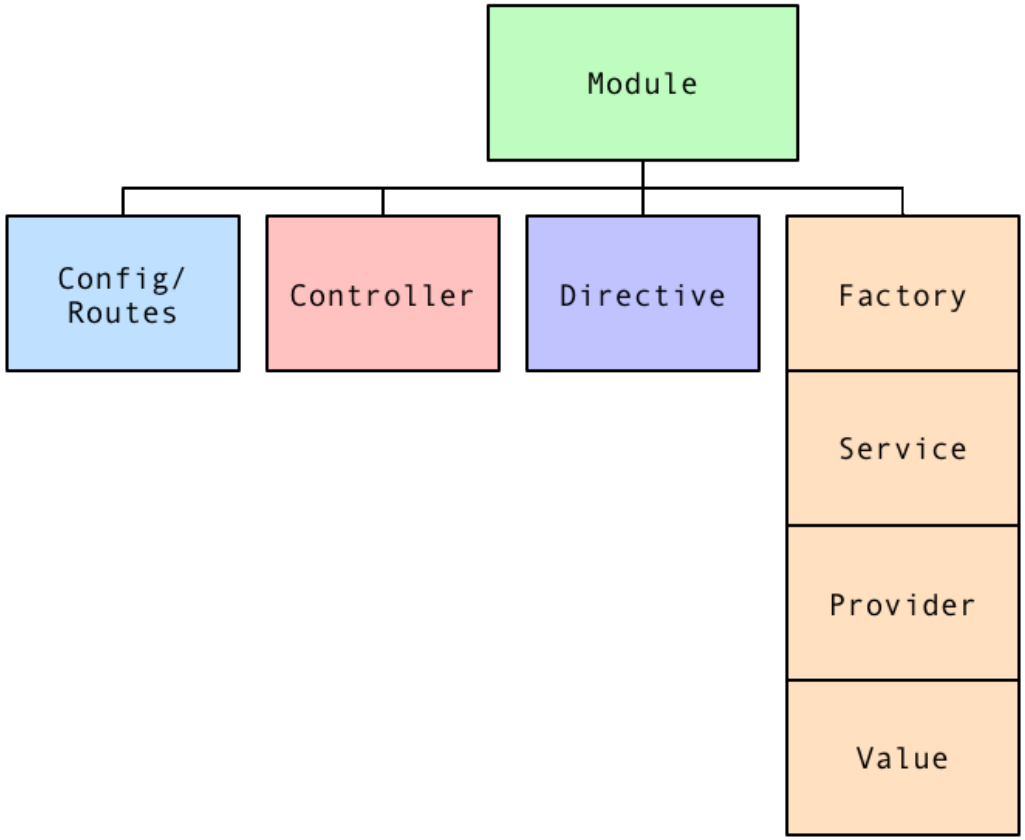
We need a framework!

digiterre agility

AngularJS

- Extensible Javascript framework created by Google in 2009
- Opinionated and standard / configuration-based
- MVC/MVVM pattern
  - View is bound to properties on the Controller
  - Model provided to controller
- Well-documented testing strategy
  - Agreed patterns for testing controllers and services
  - UI and unit testing made simple

Angular: Concepts



Demo: AngularJS

The good, the bad, and the Angular

digiterre agility

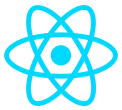
Pros

- Angular is a fully-fledged opinionated framework
  - This takes a lot of time away from decision-making and allows developers to get productive from the start
- Angular re-uses many familiar concepts from statically typed languages
  - Approach helps to apply SOLID principles to code written in this format
- Significant support for external modules
  - Everything from shared UI to internationalisation

Cons

- There is a significant learning curve
  - An initial investment of time by all team members is necessary
- Some say Angular detours from standard Javascript concepts

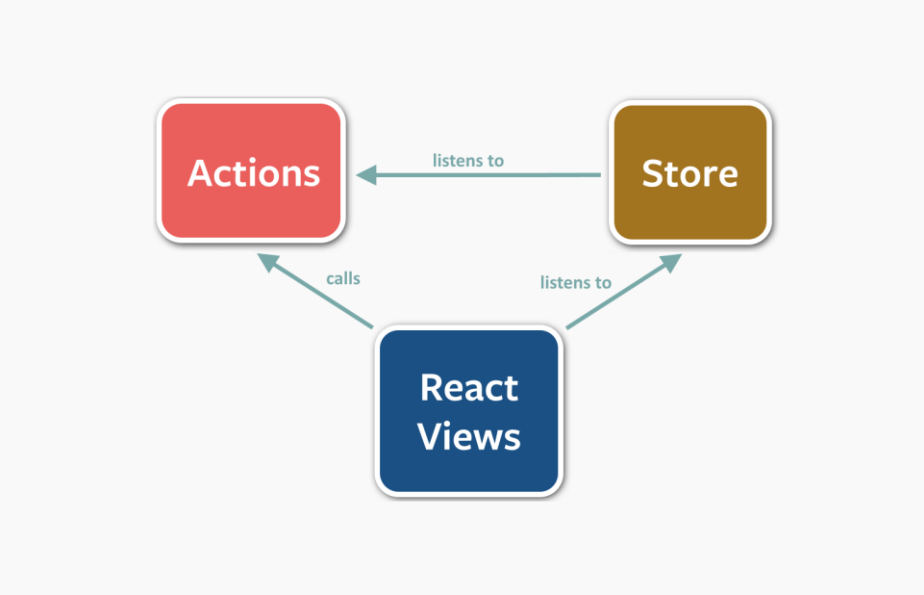
Introducing React & Flux



digiterre agility

- React is a *library* for building user interfaces for web applications
- React has been growing in popularity for the last 2 years, and provides a way for building native mobile interfaces with React Native
- React can integrate with MVC/MVVM frameworks (such as Angular), but has a recommended data architecture - **Flux**

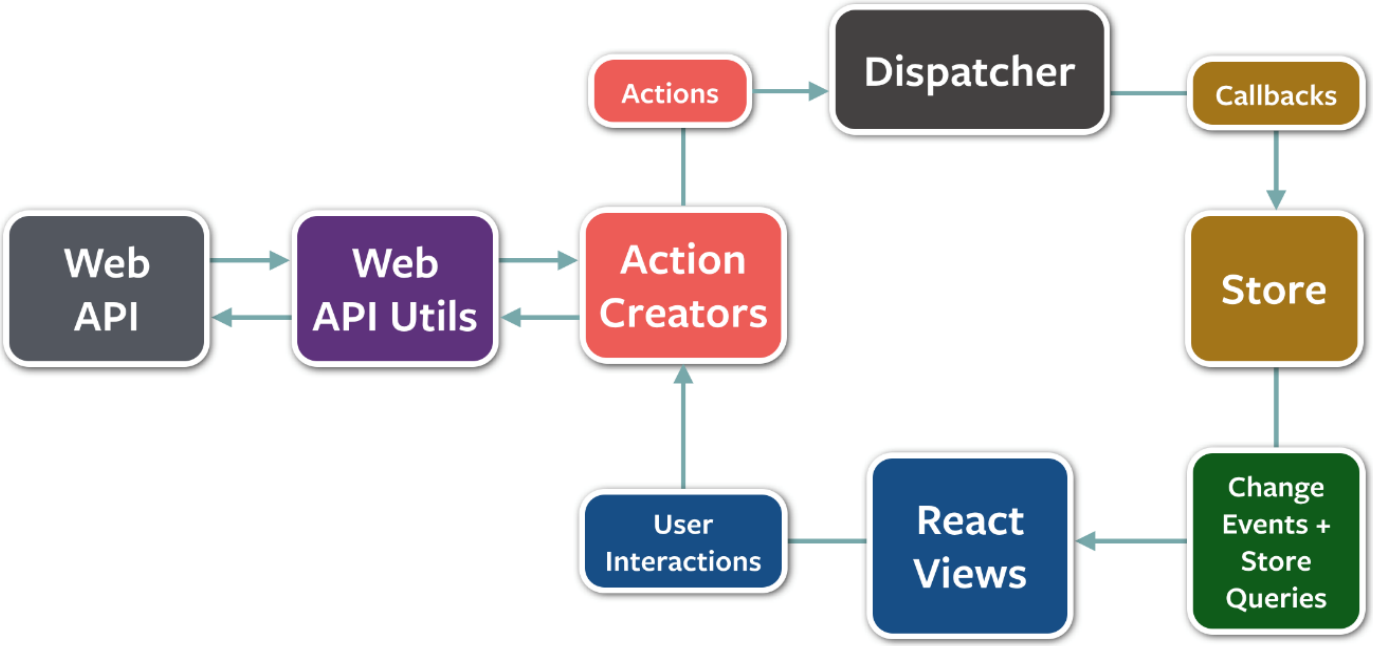
- Makes use of "virtual DOM", speeding up visual tree changes (no databinding)



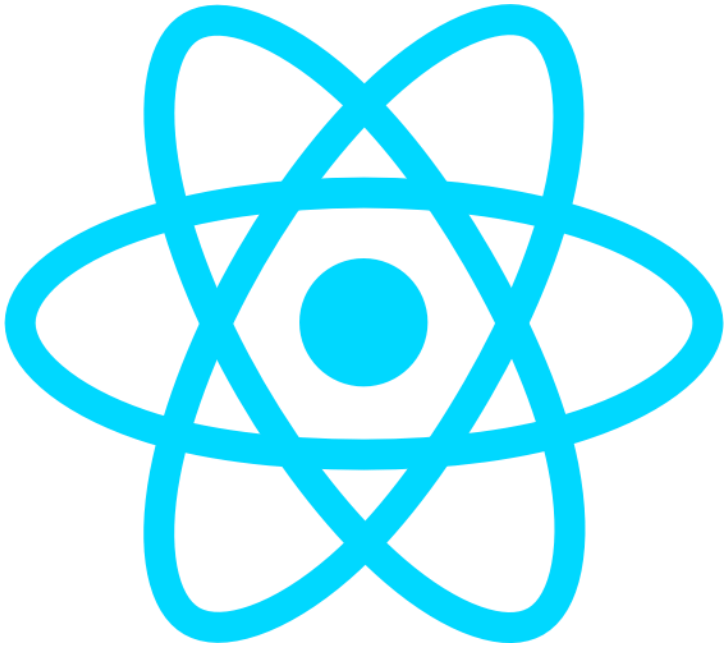
React: Components

- Every logically separated piece in the UI can be broken down into a React "component"
  - This provides separation from the outset
- Components can use JSX, a React-specific way to define HTML in javascript files
  - This keeps all of the interaction code around a component in one file
- Component-based design promotes re-use
  - Some repositories of components online – reuse Bootstrap-specific ones, for example

React: Flux



- Key concepts: **Dispatcher**, **Stores**, **Views** and **Actions**
- Composable, re-renderable views simplify interaction logic



Demo: React & Flux (simple)

React: First thoughts

digiterre agility

Pros

- Easy to use – simple javascript
  - Very simple modular structure makes writing components easy
- Logic and HTML for components in one file
  - Hard to become unwieldy due to component principles
- Flux allows easy consumption of price events
  - Rate changes simply cause action creation

Cons

- Flux causes significant **boilerplate code**
- No two-way binding!
  - Differs from Angular significantly in this way
  - Hard to grip initially
- Lack of opinionated test strategy

Recap: Angular, React & Flux

digiterre agility

- Angular is an opinionated framework
  - Provides lots of direction over how to architect your application
  - Both good and bad – easy to get started & pick up, but difficult to stray away from the pack
  - Good for teams with a .Net or Java background
- React feels more like Javascript
  - React is fluid and due to it's small surface area can integrate with many other packages
  - React is quickly rising in popularity but is stable
  - React Native provides a quick way to get started with Mobile
- Flux is independent of React, but impressive
  - Provides a simple way to understand data flow through an application
  - Flux frameworks such as Alt and Redux simplify the process

Looking forward

- How is this landscape going to change over the next year?
  - Angular2 – TypeScript
  - React & React Native reaching stable
- Angular2
  - Second iteration of Angular
  - Significant change – much closer to React
  - Written in TypeScript, transpiled to Javascript

Recommended Resources

digiterre agility

- "You don't know JS!" – free book series by Kyle Simpson
  - https://github.com/getify/You-Dont-Know-JS
- Online video training:
  - Pluralsight – https://www.pluralsight.com/
  - Egghead – https://egghead.io/
  - FrontEndMasters - https://frontendmasters.com/
- React components:
  - http://www.react-components.com/
  - http://react.parts/

End