Angular 4 and more...

TypeScript, Reactive Programming, and more.

Agenda

1. Live demo of real Angular4 App in DNN
2. Background: This is Angular, CLI and TS
3. Live demo building your first Angular App
4. Background: Developing JS in 2017
5. Live deep-dive into the DNN-App
6. Background: Reactive Programming
7. Real integration into DNN w/security
8. Background: Angular vs. React vs. Others

Who’s talking

Daniel the iJungleboy – est ‘78

- Founded 2sic in 1999
- Architect of 2sxc since 2012
- Angular since 2014
- Blogger, daddy, nerd, ceo, checklist-freak, world-traveler, …

Christoph

- Joined 2sic 2016
- Worked on complex AngularJS 1.x solutions
- Lead Dev on new Quick-Dialog in 2sxc 9

2sxc 9 with Angular 4

Demo content

Click the *pencil-icon* to edit content. This is just a demo-text with Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed sit amet gravida magna. Proin in porttitor ante. Vestibulum varius pretium libero nec maximus. Nulla ac nulla purus.

Phasellus condimentum vehicula felis, quis porttitor enim aliquet in. Praesent consequat neque eget nisl dictum malesuada. Nam id efficitur arcu. Donec id mauris viverra nibh volutpat dapibus. Vivamus molestie sem urna, commodo laoreet nisl fermentum rutrum. Vivamus maximus, est ac gravida mattis, lacus orci gravida ipsum, non viverra leo ligula iaculis libero.

Image / Text 1:2 (showing *Basic content*)

The image is about 1/3rd of the available width...

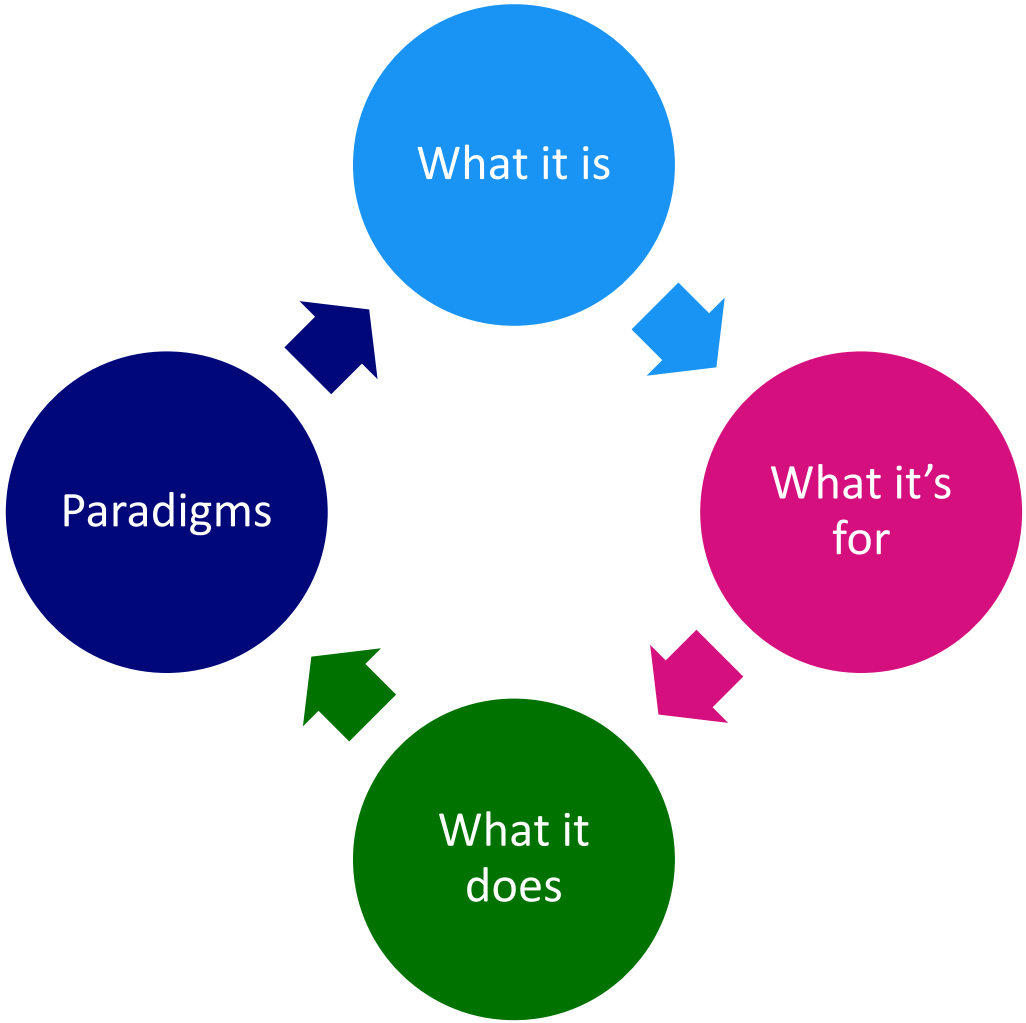
Content Change View (8)

Angular4 Directory App in DNN

Live Demo

Background: Angular

Angular



Angular is a Way to Build Solutions

- JavaScript based
- Google backed
- Completely redone (like .net core)

- Angular = new → Platform
- AngularJS = old → Framework

Angular is for creating…

- Any kind of application
- Usually with visual output und user interaction
- Cross platform & device
  - Usually in a browser
  - …or on a server
  - …or on a native device

Angular does

- Standardize structures
  - Files / folders
  - Parts / Components
- Standardize how components relate and communicate
- Recommends
  - libraries like Rx
  - tools like TS, WebPack
  - dev workflow like CLI

Write Angular with style.

Looking for an opinionated guide to Angular syntax, conventions, and application structure? Step right in! This style guide presents preferred conventions and, as importantly, explains why.

Contents

Angular Paradigms

- Very pattern oriented
- Dependency Injection
- Separation of Concerns
- Every feature is an independent part
- Strong testability, both unit and e2e

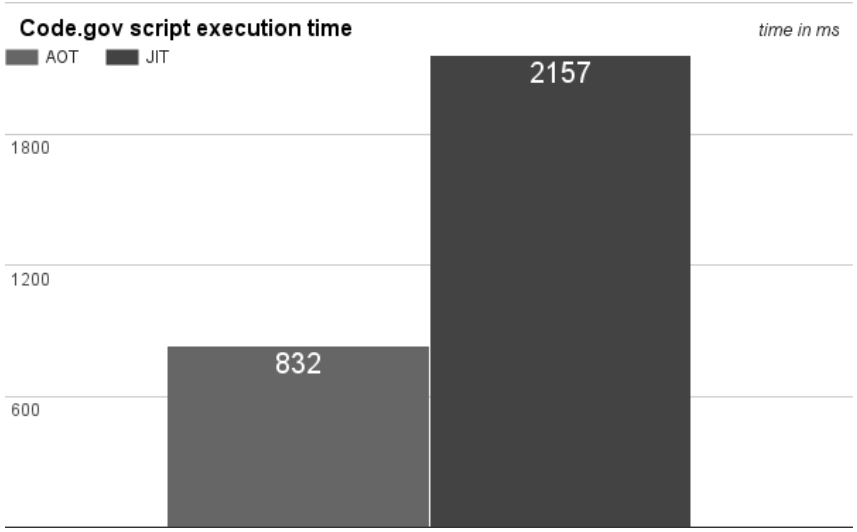
Things are done very ground-up. Then tools are added, to make it faster to develop.

- Functional / Reactive Programming with Observables
- JS Type Safety with TypeScript

Developers must understand what they do and why

Angular has…lots of optional stuff

- Dependency Injector
- Various view engines
  - like browser-engine
- Lots of pre-build parts for optional use
- AOT-Compiler
- Server Runtime
- Native Runtime



TypeScript & Angular CLI

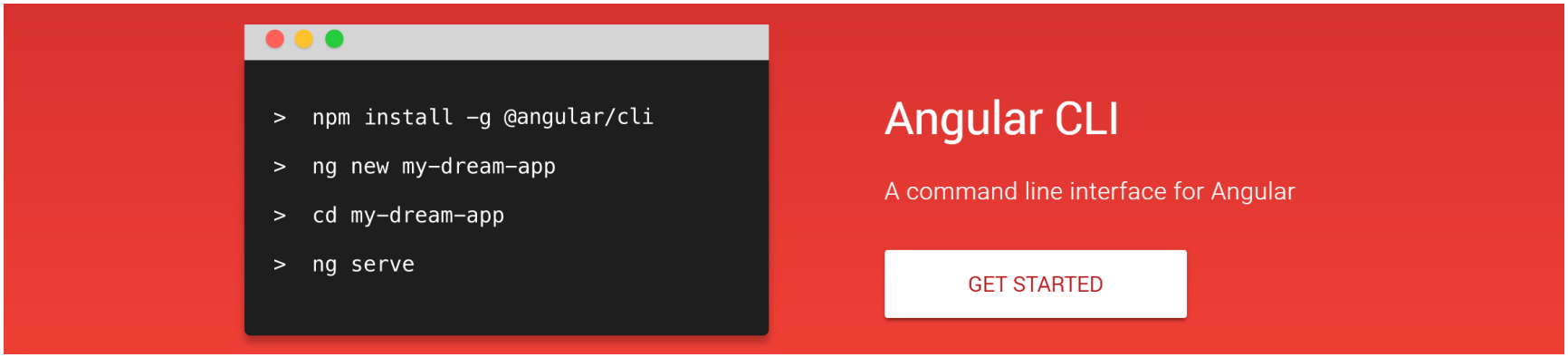
Background

A word on TypeScript

- JS / ECMA 7++
- Provides Types, Interfaces and Classes in JS
- …and Intelisense!
- Usually supports JS which not all browsers support…
- …and will be transpiled to simpler JS

The Angular CLI

- Standardizes…
- …and streamlines a lot of work steps
- You basically cannot develop Angular without the CLI
- Does project structure, compiling, WebPacking and way more automatically
- Requires node



Let’s Build with Angular

Developing JS in 2017

This is not Your Grandpa's JS

- @-annotations
- modules
- exports
- constructors
- let x = 7;
- Arrows (lambda) =>
- Classes & Interfaces
- Rx & Observables
- Multi-line strings

Learn if you're a Web Dev in 2017

- node & npm
- git
- TypeScript and ES7
- Patterns and Architecture
  - Dependency Injection, SoC, SRP, …
  - Reactive (Stream-based) programming
  - One way data flow
  - Functional programming…

Optionally learn this:

- Gulp / Grunt (so 2015)
- WebPack (so 2016)

Slowly unlearn this:

- Two-way data binding
- Promises
- jQuery
- Server rendered HTML
- MVC

Deep Dive into Real App

Reactive Programming

Stop making promises

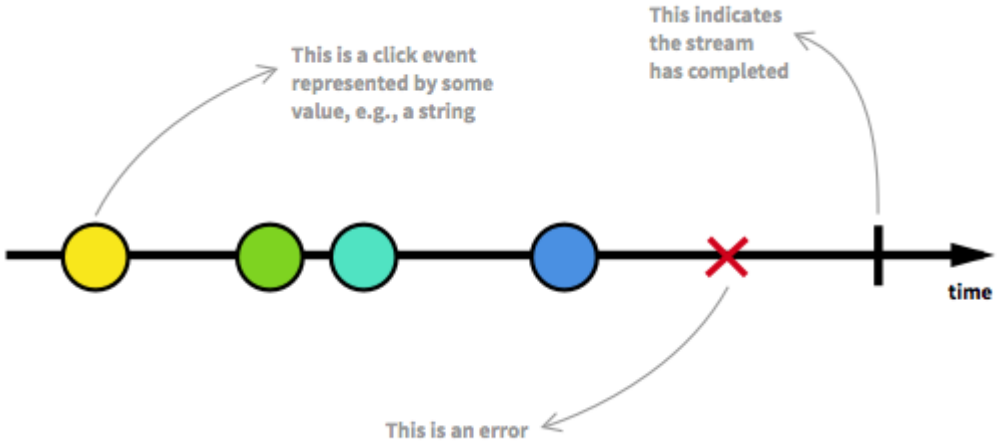
Events happen…in crazy sequence

- New data arrives
- Clicks happen
- Settings/filters change

These events are not coordinated. One can defeat the other…

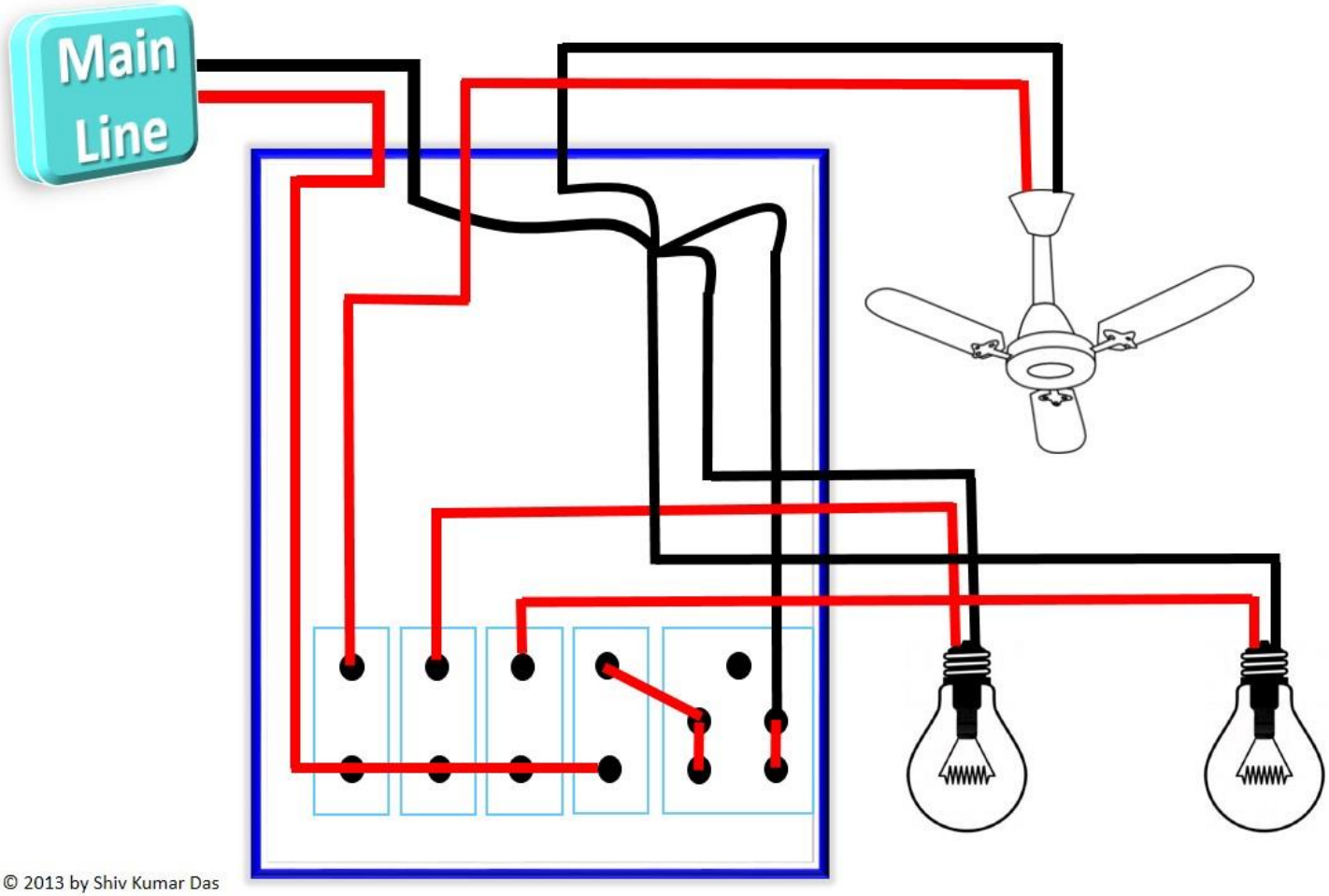
We need a better way to model this kind of real-world problem

Streams, Data, Observables

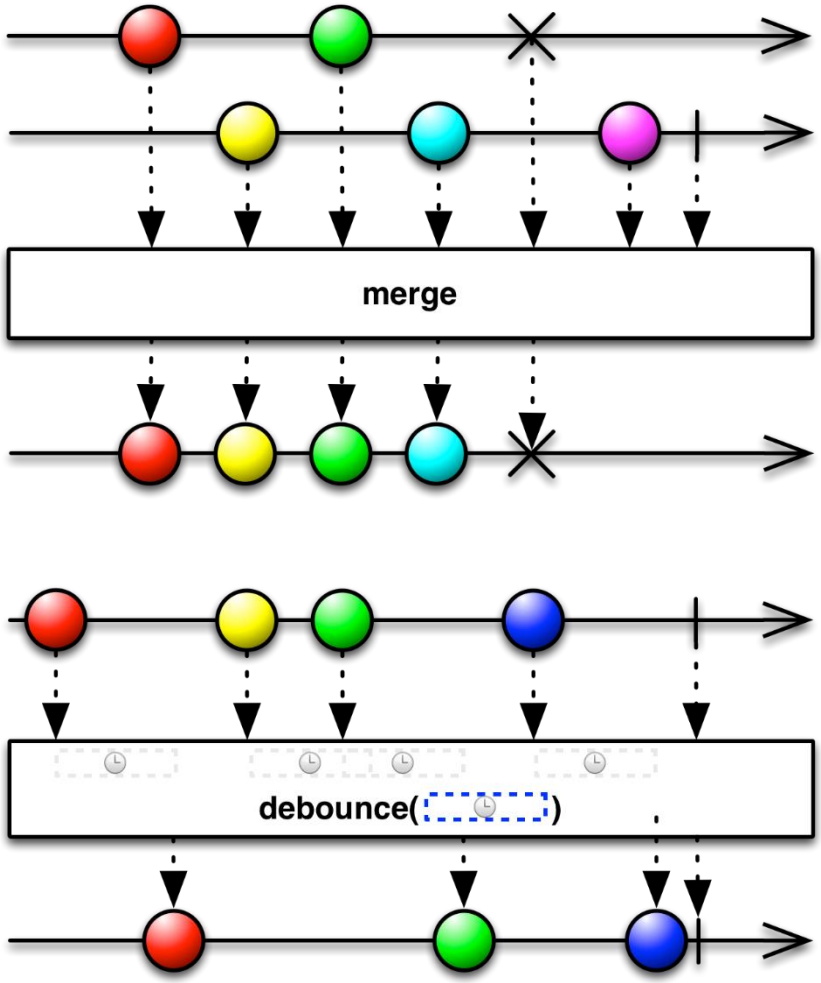


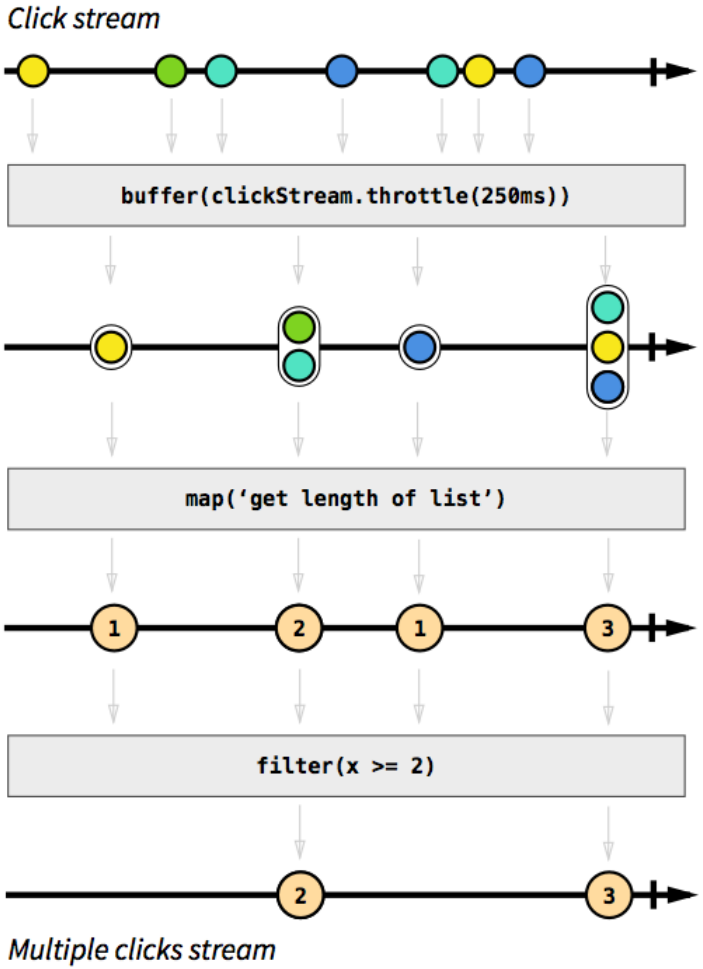
Google: The introduction to Reactive Programming you've been missing

Reactive programming is wiring…



Commands like map / reduce





RxJS, RxNet, RxJava, …

- The same model to describe a problem in all languages
- Like LINQ for events

- …but complete change of paradigms

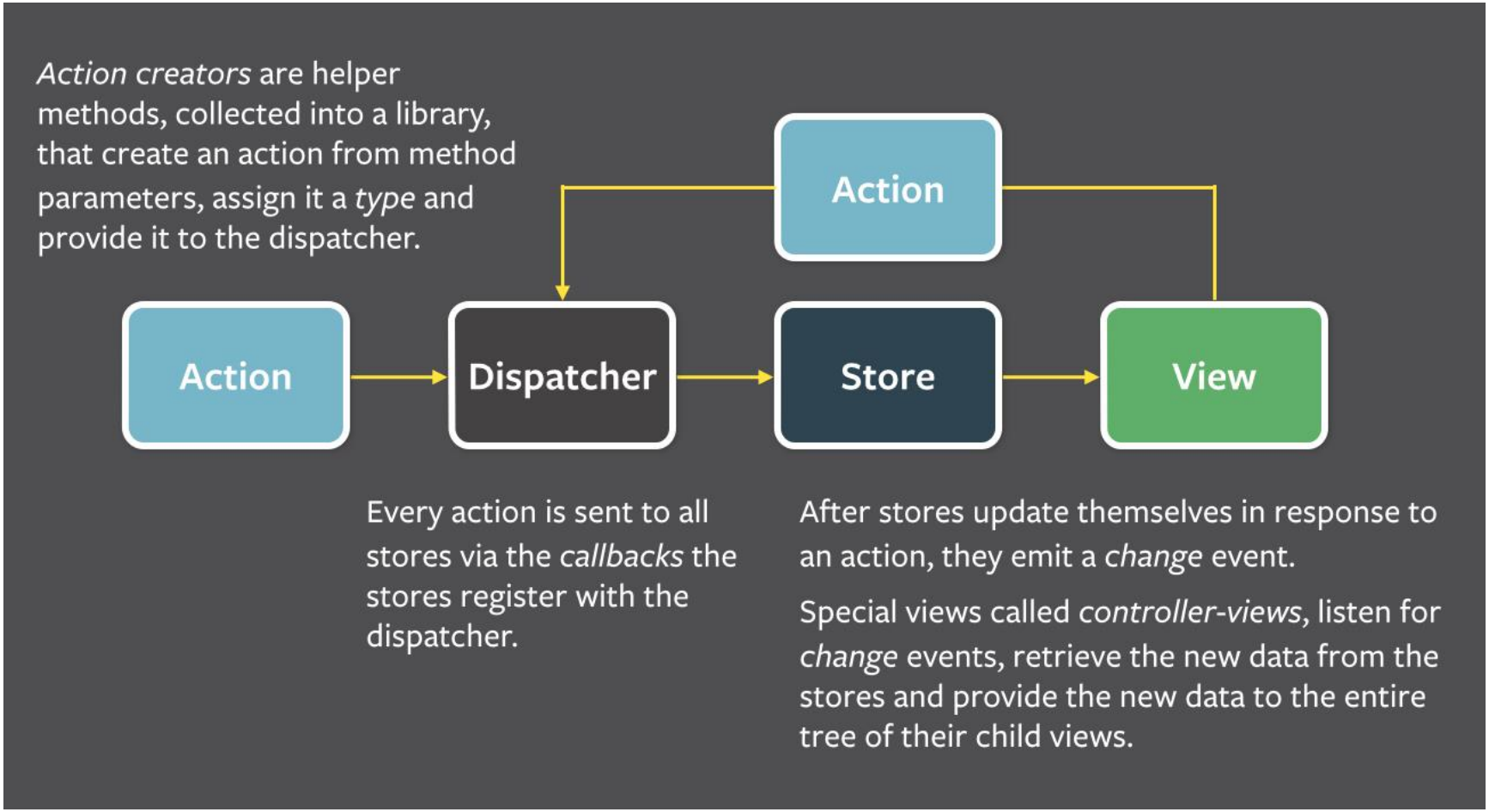
Example of Events in App

- See observables code in our app

One Way Data Flow

Stop two-way bindings, and probably stop MVC

Two Way Data Binding is Dead



DNN Integration

Challenges

- Runtime needs DNN dependencies
  - Module/tab ID and security token
- Initial base path
- Development workflow
  - Change / recompile / reload time
- Package distribution
- Multiple Apps per page

Our Solution is 80% complete

Goal

1. stay within Angular conventions
    1. IoC
    2. npm
2. loose coupling
3. ensure that environment changes don't require a rebuild
4. provide optional overrides for ng-serve mode

- Auto-detect the context
    - moduleID, TabID, security token etc.
- Replace Http-Service with a dnn-version
    - dependency injection

Let's look at some code ☺

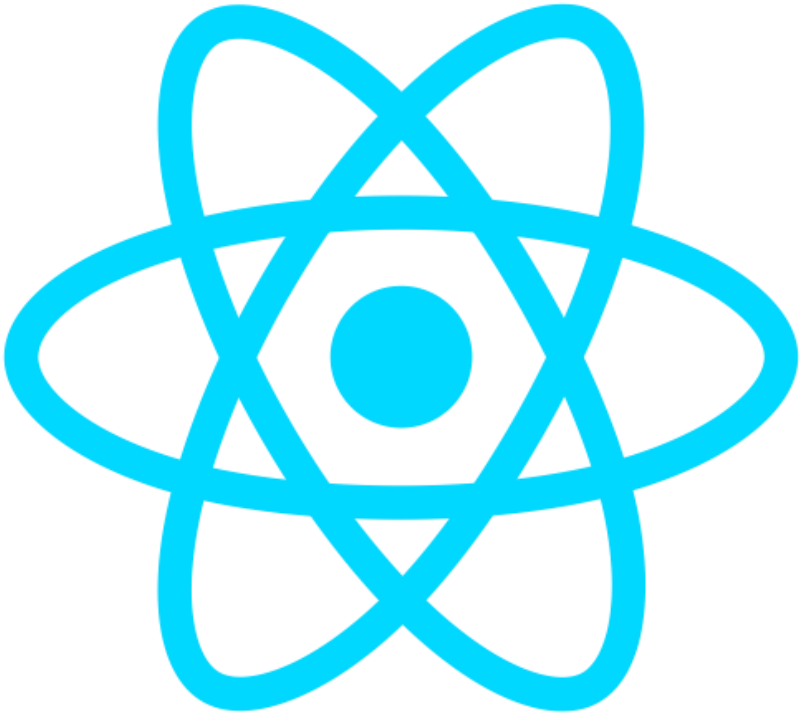
- NPM Packages (beta, should get done next week)
- Dependency Injection for all Http
  - See the DnnHttpProvider in the Provider section
- Root component must do auto-detect
  - See AppComponent.ts
    - …extends DnnAppComponent
    - Super(…, …);

Angular vs. React vs. XYZ

Every fan can prove that he’s right

- Search stats
- Performance numbers
- Some brand using or backing XYZ
- Superior paradigms
- Anecdotal evidence

The leading frameworks 2017



Inspiring Contenders

Pick your poison ☺

- Anything with a large community will thrive
- …and any really good idea will be copied by the others
- Angular CLI – inspired by ember
- Angular view & virtual DOM – inspired by React / JSX

Quick Recap!

Angular and JS 2017

- JS is really big today
- both Angular (platform) & React (with Redux) are good for 2017
- Observable Streams
- One-Way data flow
- Patterns everywhere
  - no more room for people who ignore theory / background
- Angular 4 is stable and works in productions
- Angular Material is not complete yet
- Tools are good and productive
- Correct DNN integration is hard, but easy when we're done

Thank you!

Questions?