

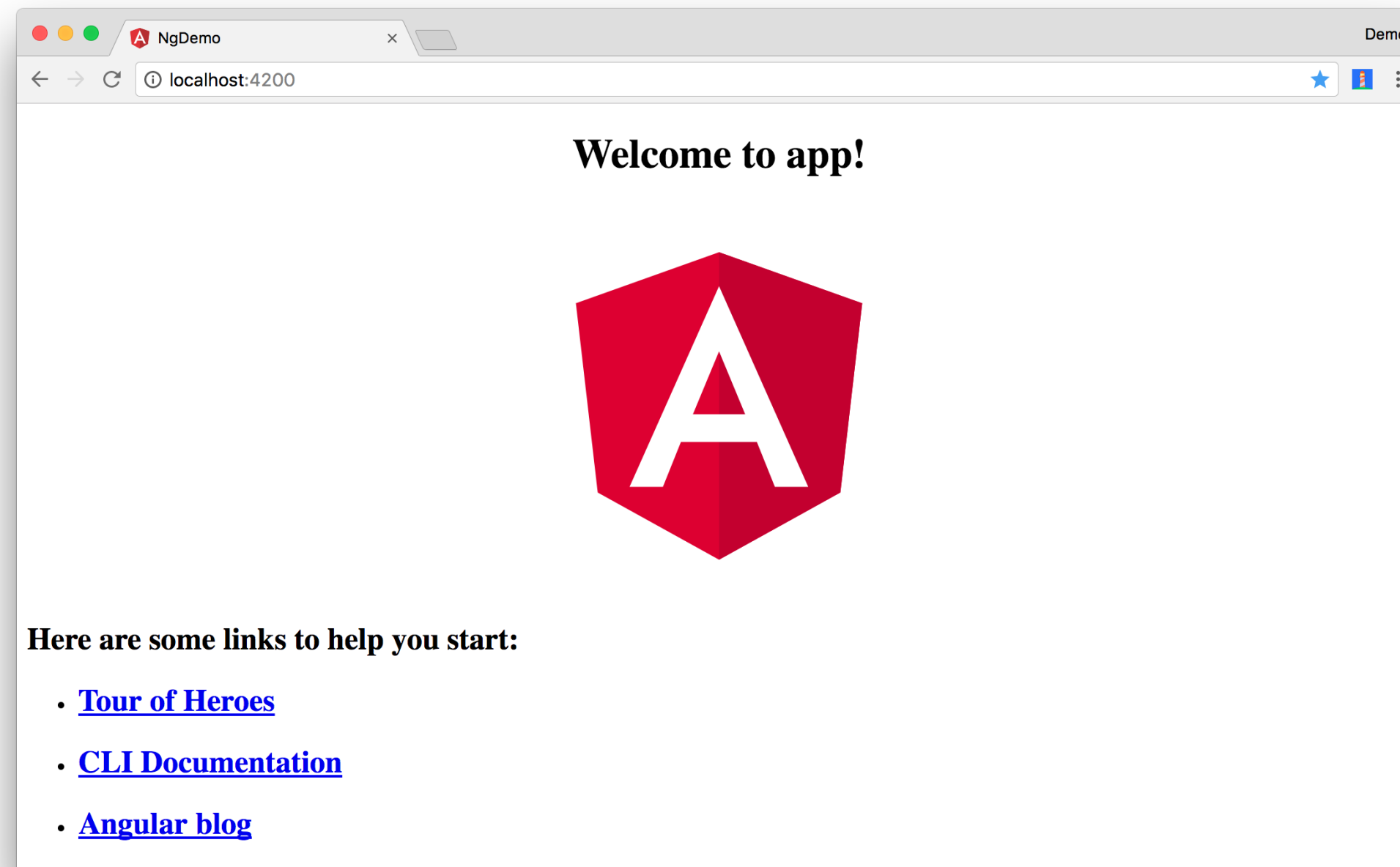
## COMMON APPLICATION FRAMEWORK

---

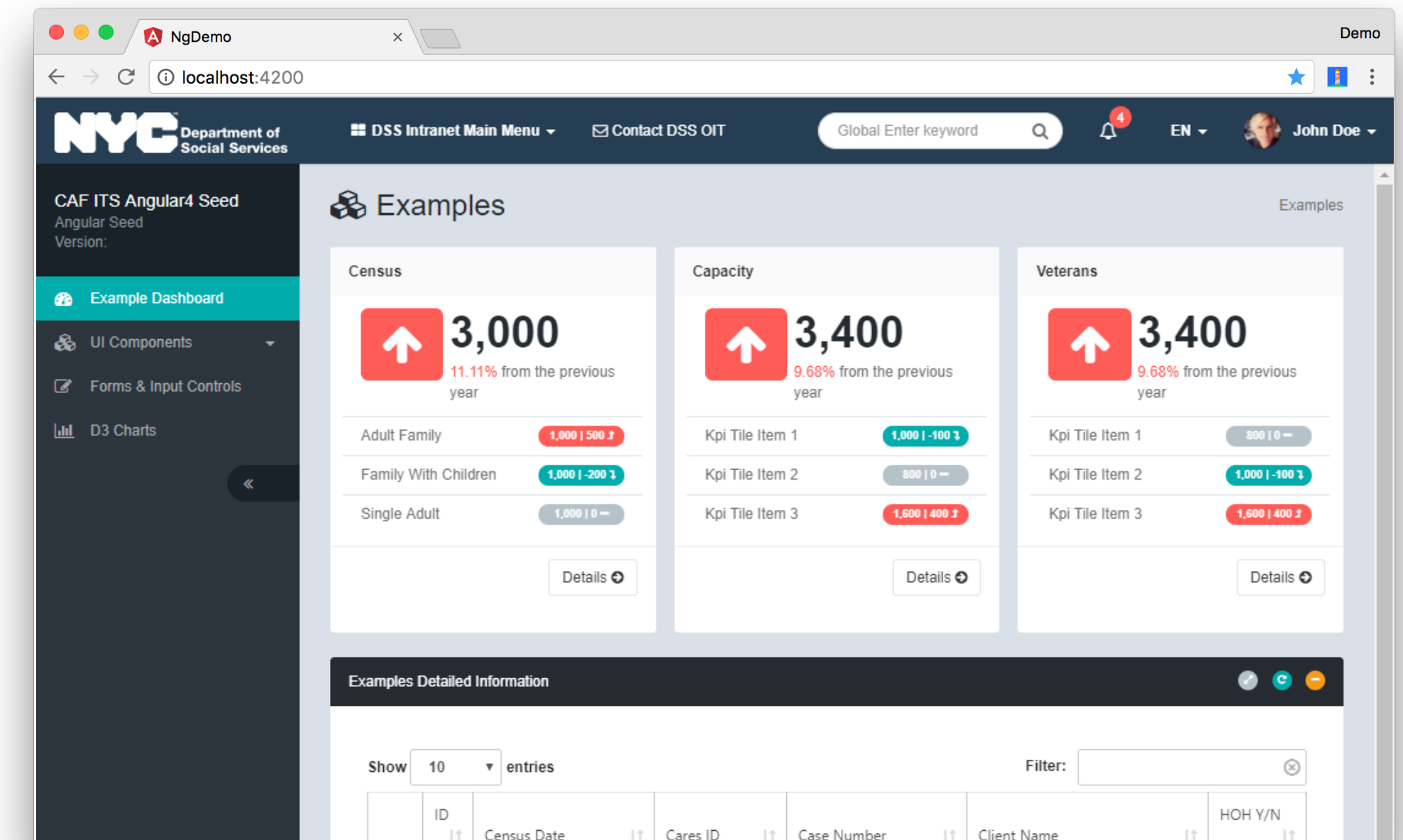
# Angular 4 Seed Walkthrough

	Today	Angular 4 Common Application Framework
UX Consistency	UX Design Inconsistencies	100% DSS/DHS Standard UX Design
Deployment	Inflexible Deploy Options (client + server = 1 repo)	Separate Unit of Deployment from Development
Platform	Older Platforms(MVC, Angular1)	Well-supported, Rapidly growing Platform (Angular 4+)
Latency	Long Start-Up Latency	Instant-On Low Latency
Performance	Performance Issues due to Chattiness, Large Payloads	High Performance: Fewer AJAX calls, Smaller Payloads
Reusability	Reuse primarily via Copy/Paste	Component-Bases Reuse
Code length	1000s of lines of code	100s of lines of code
Linting	Inconsistent Code quality, Formatting	Linting Built-In
Automated Testing	No Automated Tests	Fully Automated Acceptance/Unit Tests
Production Deployment	Challenges	Deployment Simplified, Automated, CI/CD Built-In
Production Debugging	Challenges	Rich State Debugging Tools: Augury, Redux Dev Tools

## Angular 4 Application created with Angular CLI



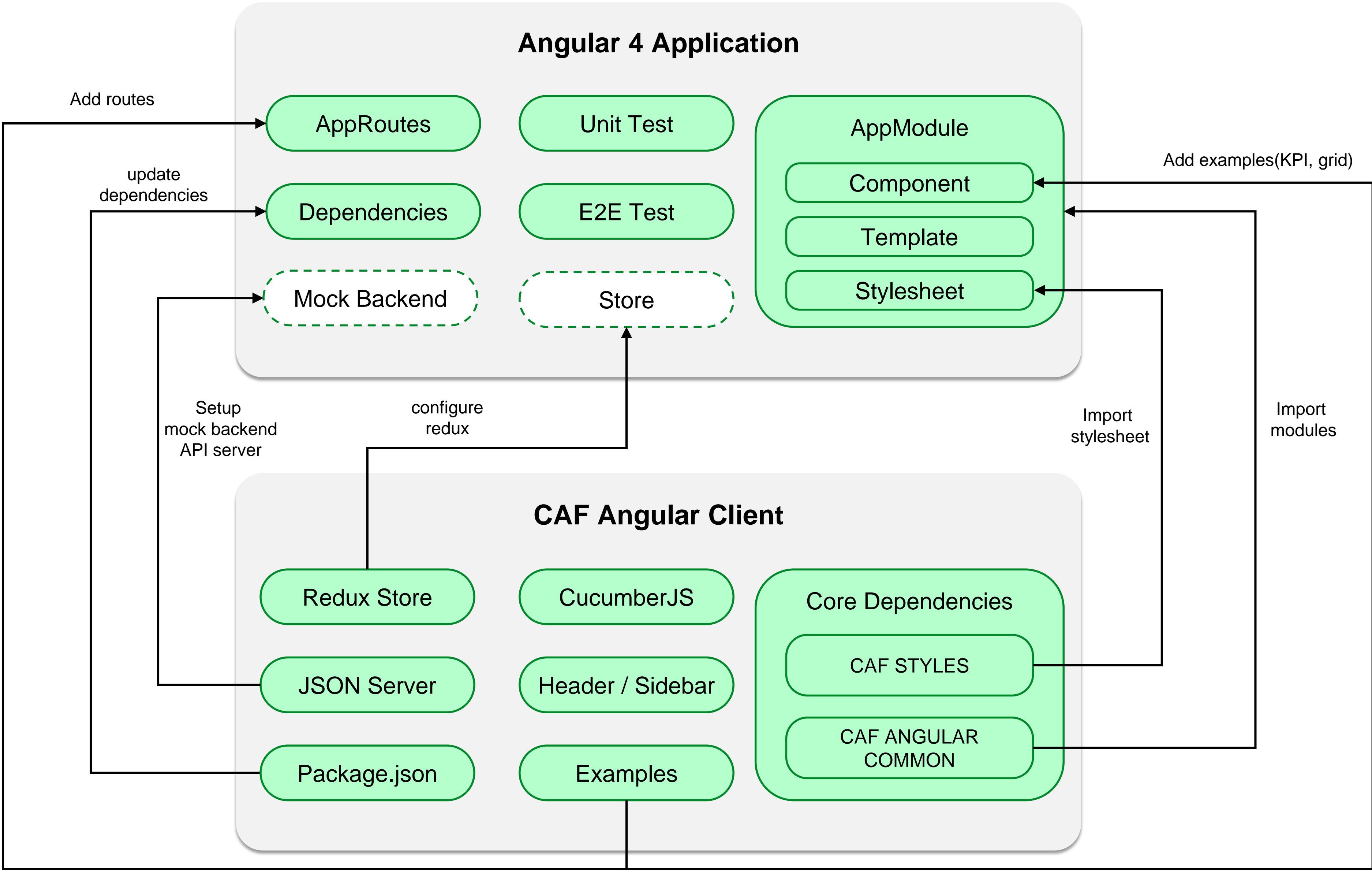
## CAF Angular Client built with CAF Angular CLI



```
npm install -g @angular/cli
ng new my-app
cd my-app
npm install
ng serve --open
```

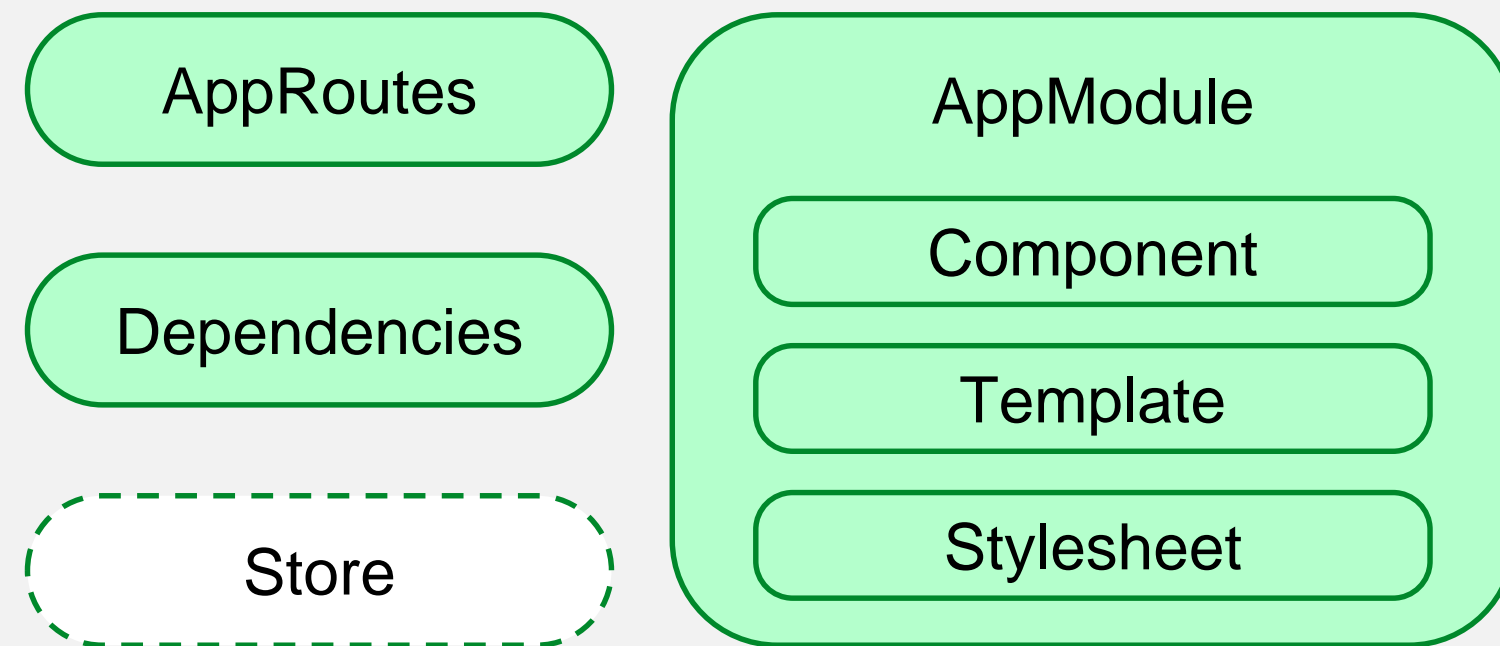
```
npm install -g @angular/cli
npm install -g https://bitbucket.org/dhsit/caf-angular-cli
caf new my-app
cd my-app
npm start
```

# How CAF Builds on Angular 4(Architecture)

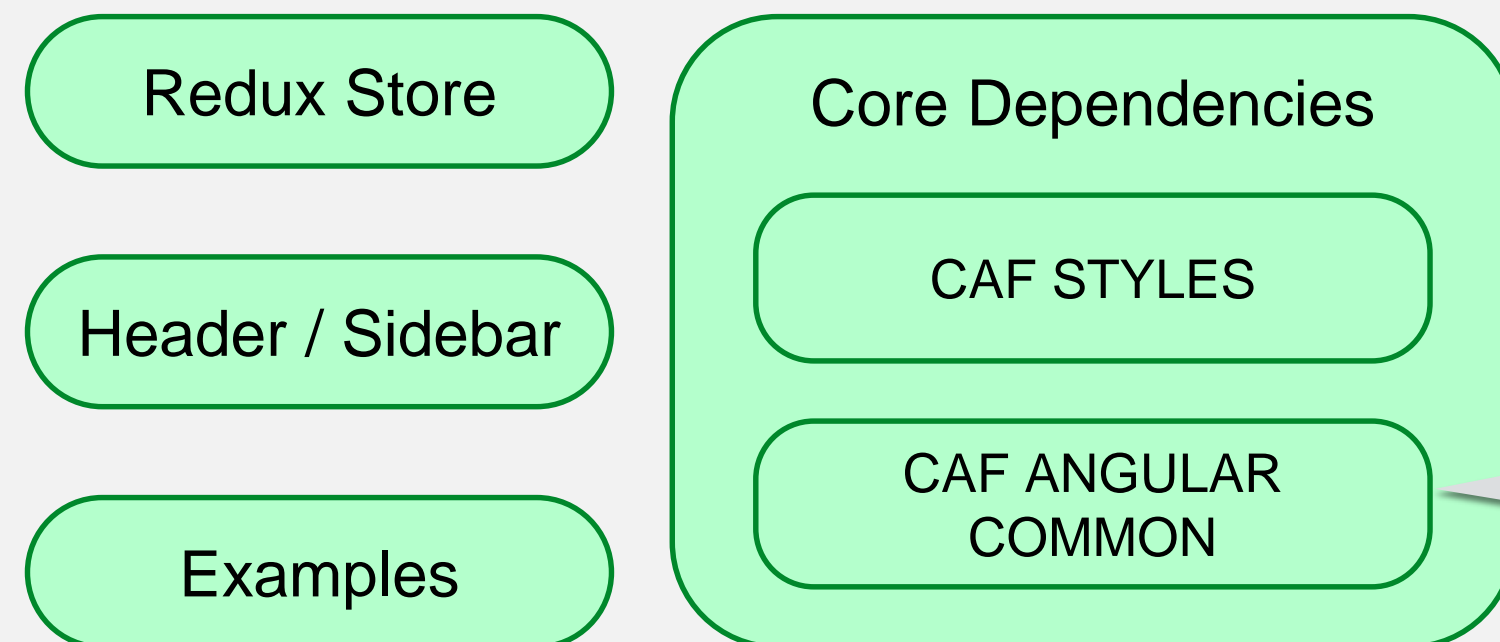


# How CAF Builds on Angular 4(Common Modules)

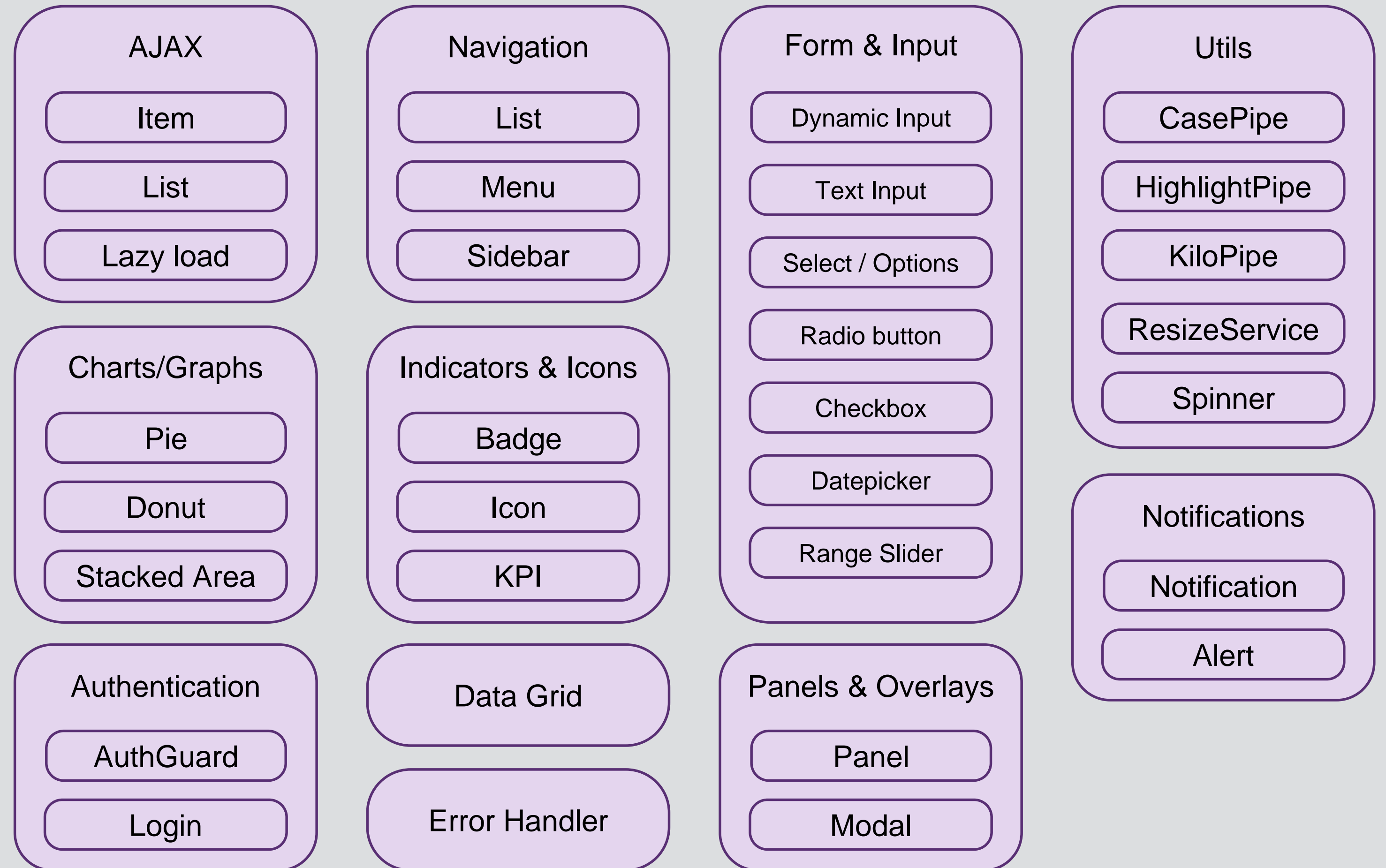
## Base Angular 4 Application



## CAF Angular Client



## CAF Angular Common Modules

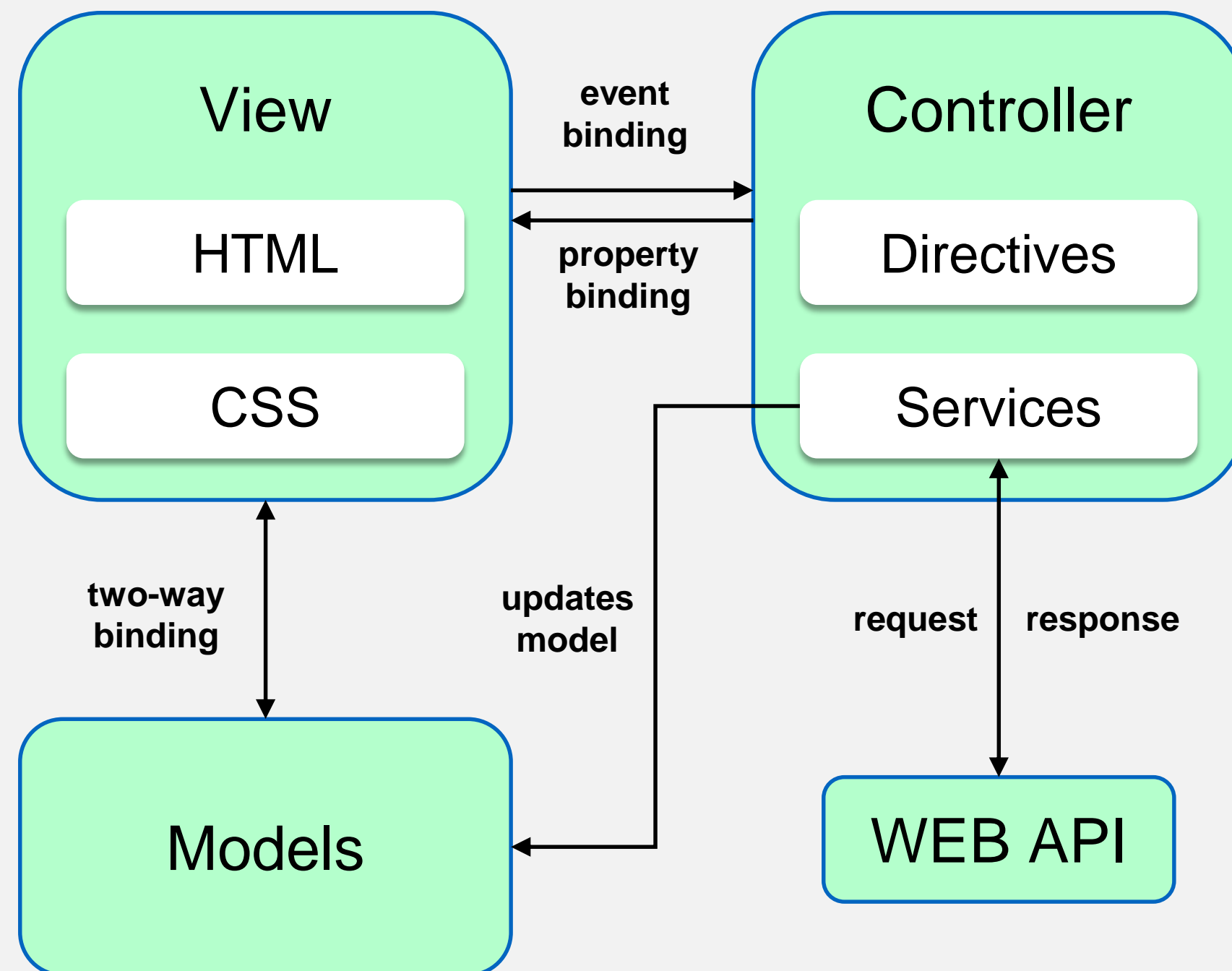




Language	CLI tool	Unit Testing	E2E Testing
Angular 1.x	X	Supported but not out of the box	Supported but not out of the box
Angular 2 +	<div><div><div>webpack</div><div>embercli</div></div><div>(Angular CLI)</div></div>	<div><div>Jasmine</div><div>KARMA</div><div>+</div><div>Istanbul Test coverage</div></div>	<div><div>Protractor <small>end to end testing for AngularJS</small></div><div><div>Selenium Web driver</div><div>+</div><div>cucumber</div></div></div>
<ul style="list-style-type: none"><li>• Avoid painful common bugs by type-checking the code.</li><li>• Reports issues without even saving file.</li><li>• Leverage the type system to help you write code even faster.</li></ul>	<ul style="list-style-type: none"><li>• Serve and build application for both prod/dev mode.</li><li>• Run testing frameworks.</li><li>• AOT and JIT compilation.</li><li>• All modern frameworks now use CLI: dotnet.exe, rake, etc.</li></ul>	<ul style="list-style-type: none"><li>• Testing frameworks are included out of the box.</li><li>• Testing helper functions.</li><li>• Additional plugins may be added in karma.config.js</li></ul>	<ul style="list-style-type: none"><li>• Testing frameworks are included out of the box.</li><li>• Protractor uses Selenium Server to output feature testing logs.</li><li>• Cucumber is used for Automated Acceptance testing</li></ul>



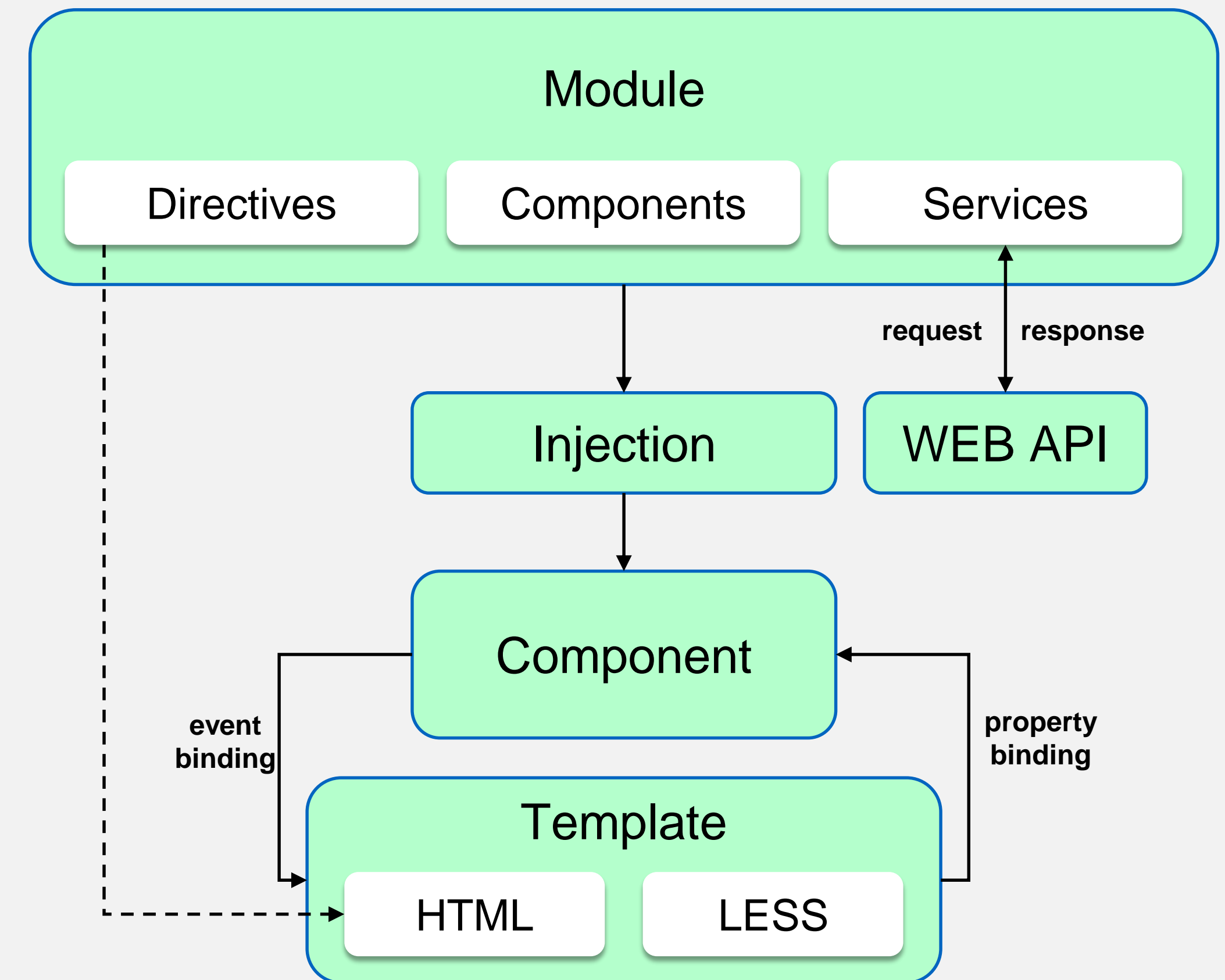
## AngularJS(version 1)



The architecture of Angular 1 is based on the **Model View Controller**.

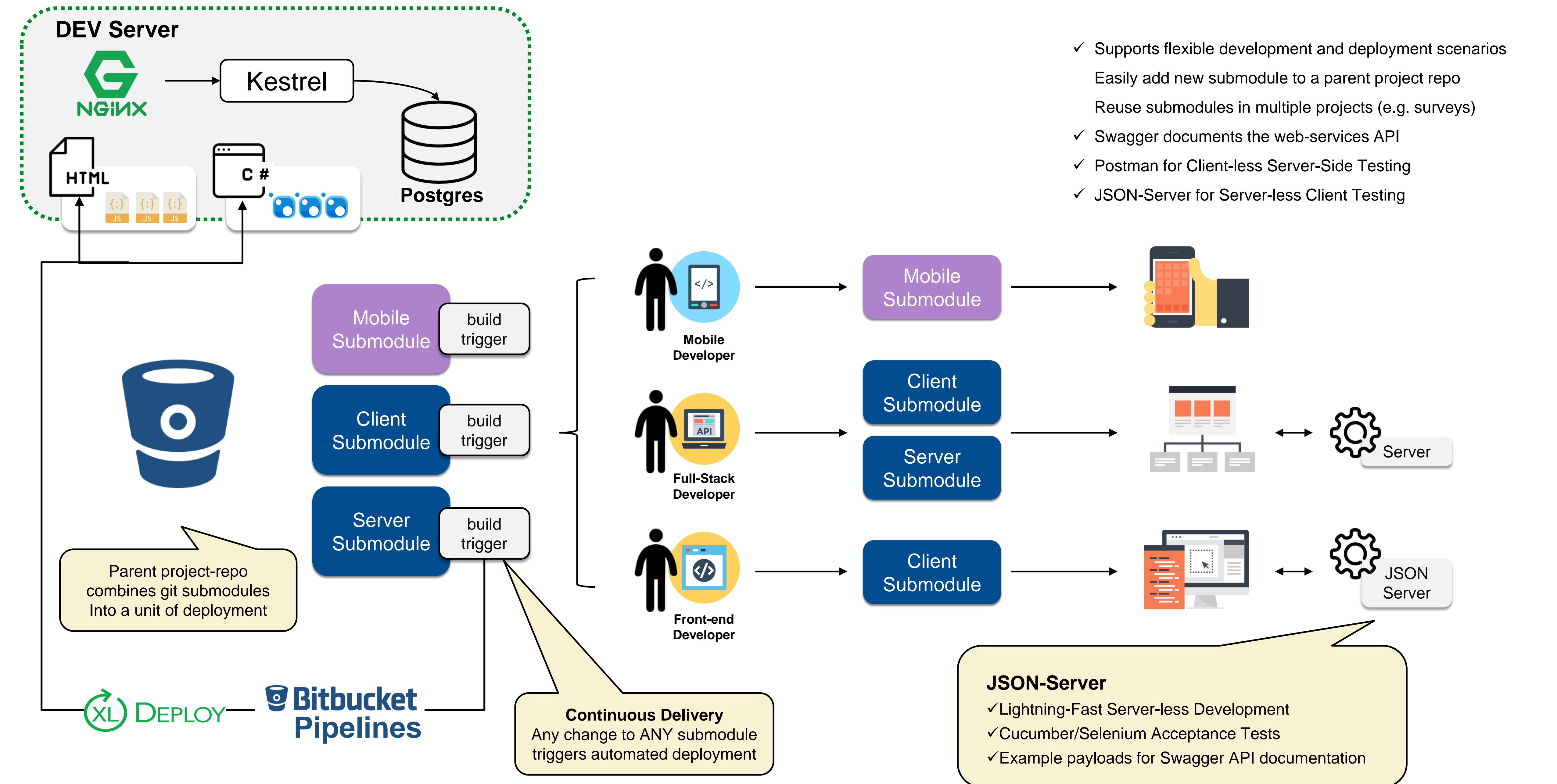


## Angular(version 2 +)



Angular 2 is based on a **Component** structure, like what we see in React.js.

# Git Submodules Separate Unit of Dev from Unit of Deployment



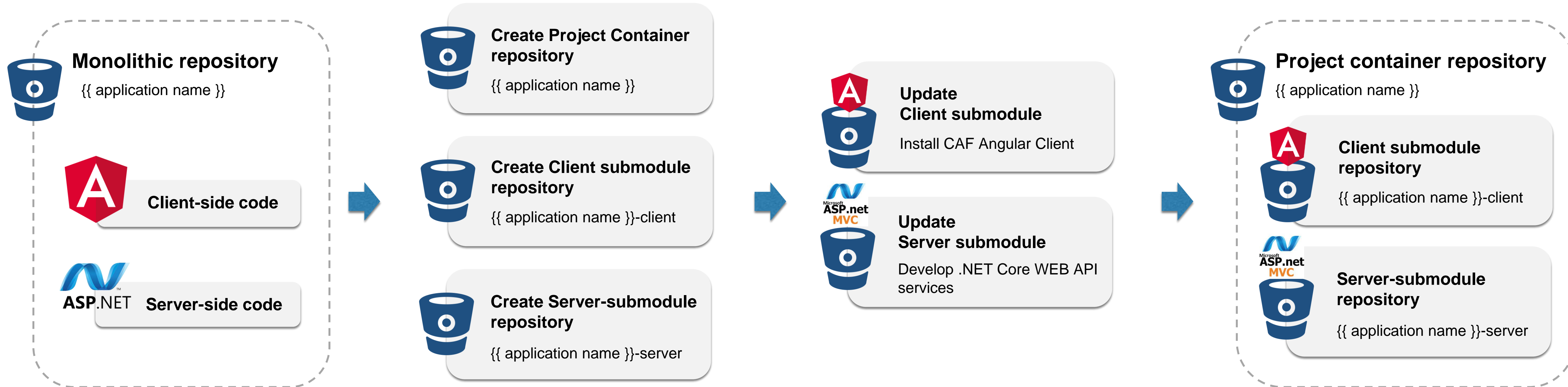


# TLDR; So.. How Do I Get Started?

Migrate from Monolithic to Micro Services Architecture

## START

- ❑ Angular 1 or ASP.NET WebForms
- ❑ .NET 4.5.1



### Step 1.

Create separate repositories for each container, client, server

### Step 2.

Update client-side with CAF angular client, and server-side with .NET core WEB API services.

### Step 3:

Add client & server repository as submodules of project container using 'git submodule' command

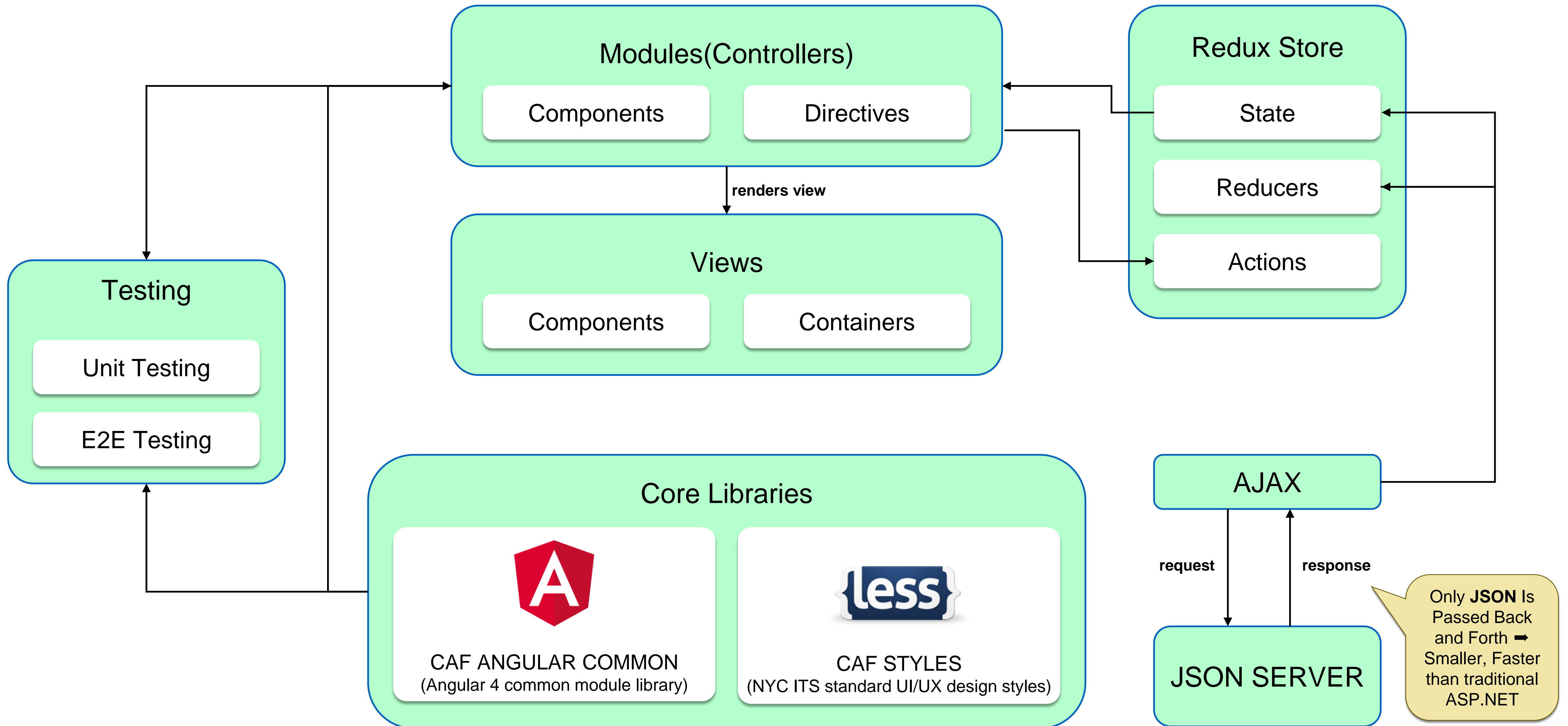
[Git submodule documentation](#)

- Sample usage of CAF Angular Common
- Production and Development builds
- Hot module reload
- State management using Redux
- Sample unit test with Jasmine and Karma
- Sample end-to-end test using Cucumber
- Continuous integration using Bitbucket Pipelines

# COMMON APPLICATION FRAMEWORK

---

# DEMO



## Component Render Example

app works

### Component(Typescript)

```
import {Component, OnInit} from '@angular/core';
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.less']
})
export class AppComponent {
  title = 'app works!';
}
```

### Template(HTML)

```
<div>{{title}}</div>
```

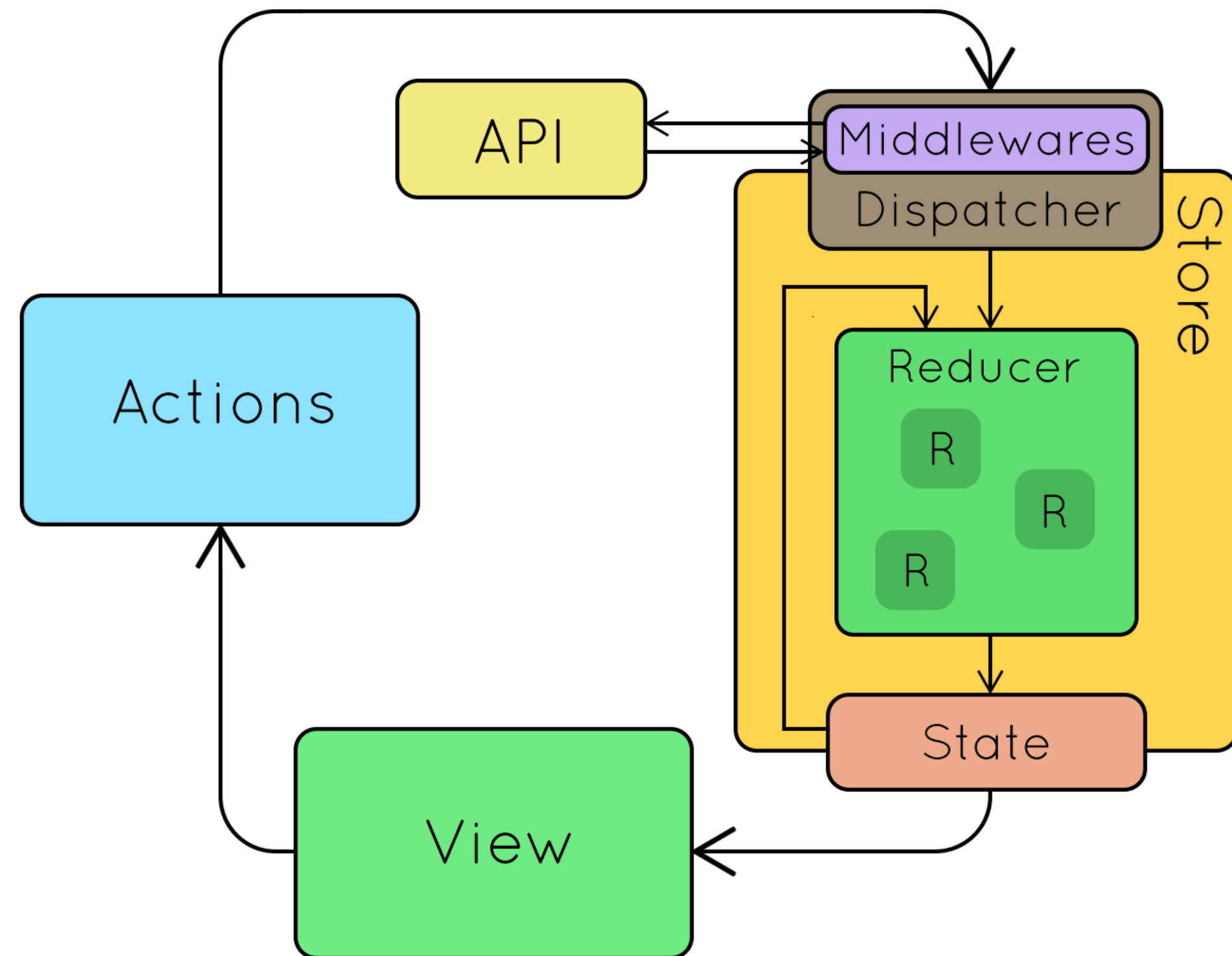
### Style(LESS)

```
:host > div {
  background: 'red';
  color: 'white';
  padding: 10px;
}
```



# Benefits of Redux Approach to State Management

## Why is Redux worth learning?



- ✓ **Performance** JSON data cached in store, no need to re-retrieve on user navigation
- ✓ **Maintainability** All state centralized, keeps code neat and clean
- ✓ **Testability** Significantly simplified unit testing
- ✓ **Flexibility** Can set eviction policies to decide when to flush cache (c.f. redux-cache)
- ✓ **Enables Powerful Use Cases** Centralized error handling, Recreate production scenario via export-import state



CAF ANGULAR CLIENT  
(Angular 4 seed project)

<https://bitbucket.org/dhsit/caf-angular-client>



CAF ANGULAR CLI  
(Angular 4 seed generator)

<https://bitbucket.org/dhsit/caf-angular-cli>



CAF ANGULAR COMMON  
(Angular 4 common module library)

<https://bitbucket.org/dhsit/caf-angular-common>



CAF STYLES  
(NYC ITS standard UI/UX design styles)

<https://bitbucket.org/dhsit/caf-styles>

Categories	Title and URL	Publisher	Size	Price
.NET Core Angular5	<a href="#">Build an app with ASPNET Core and Angular from scratch</a> A Practical example of how to build an application with ASP.NET Core WebAPI (v2) and Angular (v5) from start to finish	Udemy	20.5hr video	\$10
Automated Testing	<a href="#">Writing Great Specifications: Using Specification By Example and Gherkin</a> An example-rich tutorial that teaches you how to write good Gherkin specification documents that take advantage of the benefits of specification by example– for Automated Acceptance Tests	Amazon or Manning	400 pages	\$36.81
Angular4	<a href="#">Angular 4 (formerly Angular 2) - The Complete Guide</a> Master Angular (both Angular 4 and Angular 2) and build awesome, reactive web apps with the successor of Angular.js	Udemy	22hr video 17 articles 73 resources	\$192 (\$10 after discount)
Redux Angular4	<a href="#">Redux in Angular (2 and 4+)</a> Better manage the state of your Angular apps, improve their testability and use awesome debugging tools	Udemy	1.5hr video	\$70 (\$10 after discount)
Angular1 → Angular2	<a href="#">Upgrade Angular 1 app to Ang4 in 5 simple steps</a>	Online	10 pages	FREE
Protractor & CucumberJS	<a href="#">Tutorial: Creating maintainable E2E tests with Protractor and Cucumber</a>	Medium		FREE
Redux Observable	<a href="#">Understanding RxJS and Redux-Observable</a>	Medium		FREE

Title and URL	Description
<a href="#">Angular 5</a>	Front-end MVC Framework
<a href="#">Redux</a>	State container for javascript
<a href="#">Redux Observable</a>	Redux middleware with RxJS Observable
<a href="#">Angular CLI</a>	CLI tool for Angular
<a href="#">Angular Redux</a>	Redux store implementation in Angular
<a href="#">Typescript</a>	Language
<a href="#">RXjs</a>	The ReactiveX library for JavaScript
<a href="#">Protractor</a>	Protractor is an end-to-end test framework for Angular
<a href="#">Jasmine</a>	Unit testing framework
<a href="#">JSON-server</a>	Mock-backend for CAF

## Additional Components

**Risks:** Low

**Benefits:** High

**Effort:** Moderate

We are adding in-cell editing for tables, login integration with Active Directory and form input controls supporting template driven forms and reactive forms as well as a global error handling service

## Angular v4 to 5 Upgrade

**Risks:** Low

**Benefits:** Moderate

**Effort:** Small

Angular5 incorporates Angular Universal for optional server-side rendering, and performance improvements. Very few API changes

## Bootstrap v3 to 4 Upgrade

**Risks:** High

**Benefits:** High

**Effort:** Very Large

Requires migration from LESS to SCSS. Entire UX catalog must be migrated. Significant CAF Component Refactoring. Much better horizontal fitting for tables

Immediate  
*Now*

Short-term  
*Feb*

Long-term  
*TBD*



thank  
you