Enterprise Angular Applications with NgRx and Nx

Install dependencies

Dependency	Source	Version check
Node v8+	https://nodejs.org	node -v
git	https://git-scm.com/downloads	git --version
Angular CLI v6+	npm install -g @angular/cli	ng -v
Nx (Nrwl Extensions)	npm install -g @nrwl/schematics	

NOTE: Please uninstall angular cli before updating:

```
npm uninstall -g @angular/cli
npm uninstall -g @nrwl/schematics
```

Duncan Hunter

@DuncHunter

Nx workspaces and angular cli

Components and nx libs

RxJS, services and HTTP module

Third party dependencies

Reactive forms

Routing

Ngrx and state libs

Entity adapter and router store

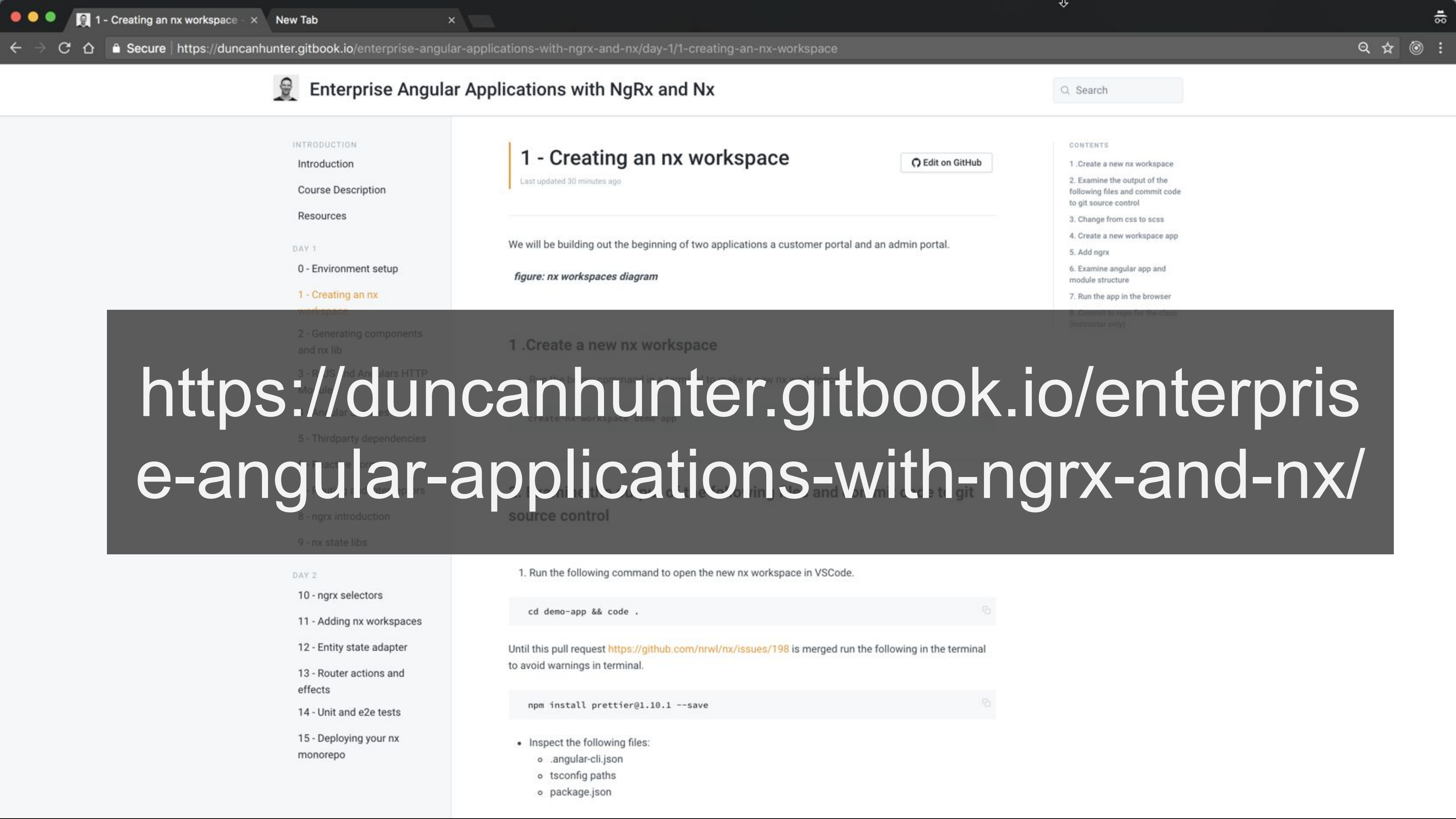
Unit tests

Deploying nx monorepo

Housekeeping

1. Facilities
2. Break times
3. Parking lot
4. Course Materials

https://duncanhunter.gitbook.io/enterprise-angular-applications-with-ngrx-and-nx/


1 - Creating an nx workspace
New Tab
Secure
https://duncanhunter.gitbook.io/enterprise-angular-applications-with-ngrx-and-nx/day-1/1-creating-an-nx-workspace
Enterprise Angular Applications with NgRx and Nx
Search
INTRODUCTION
Introduction
Course Description
Resources
DAY 1
0 - Environment setup
1 - Creating an nx workspace
2 - Generating components and nx lib
5 - Thirdparty dependencies
8 - ngrx introduction
9 - nx state libs
DAY 2
10 - ngrx selectors
11 - Adding nx workspaces
12 - Entity state adapter
13 - Router actions and effects
14 - Unit and e2e tests
15 - Deploying your nx monorepo
Edit on GitHub
CONTENTS
1 .Create a new nx workspace
2. Examine the output of the following files and commit code to git source control
3. Change from css to scss
4. Create a new workspace app
5. Add ngrx
6. Examine angular app and module structure
7. Run the app in the browser

https://duncanhunter.gitbook.io/enterprise-angular-applications-with-ngrx-and-nx/

1 - Creating an nx workspace

Last updated 30 minutes ago

We will be building out the beginning of two applications a customer portal and an admin portal.

figure: nx workspaces diagram

1 .Create a new nx workspace

source control

1. Run the following command to open the new nx workspace in VSCode.

```
cd demo-app && code .
```

Until this pull request https://github.com/nrwl/nx/issues/198 is merged run the following in the terminal to avoid warnings in terminal.

```
npm install prettier@1.10.1 --save
```

- Inspect the following files:
  - .angular-cli.json
  - tsconfig paths
  - package.json

Who has used angular 1?

Who has used angular 2+?

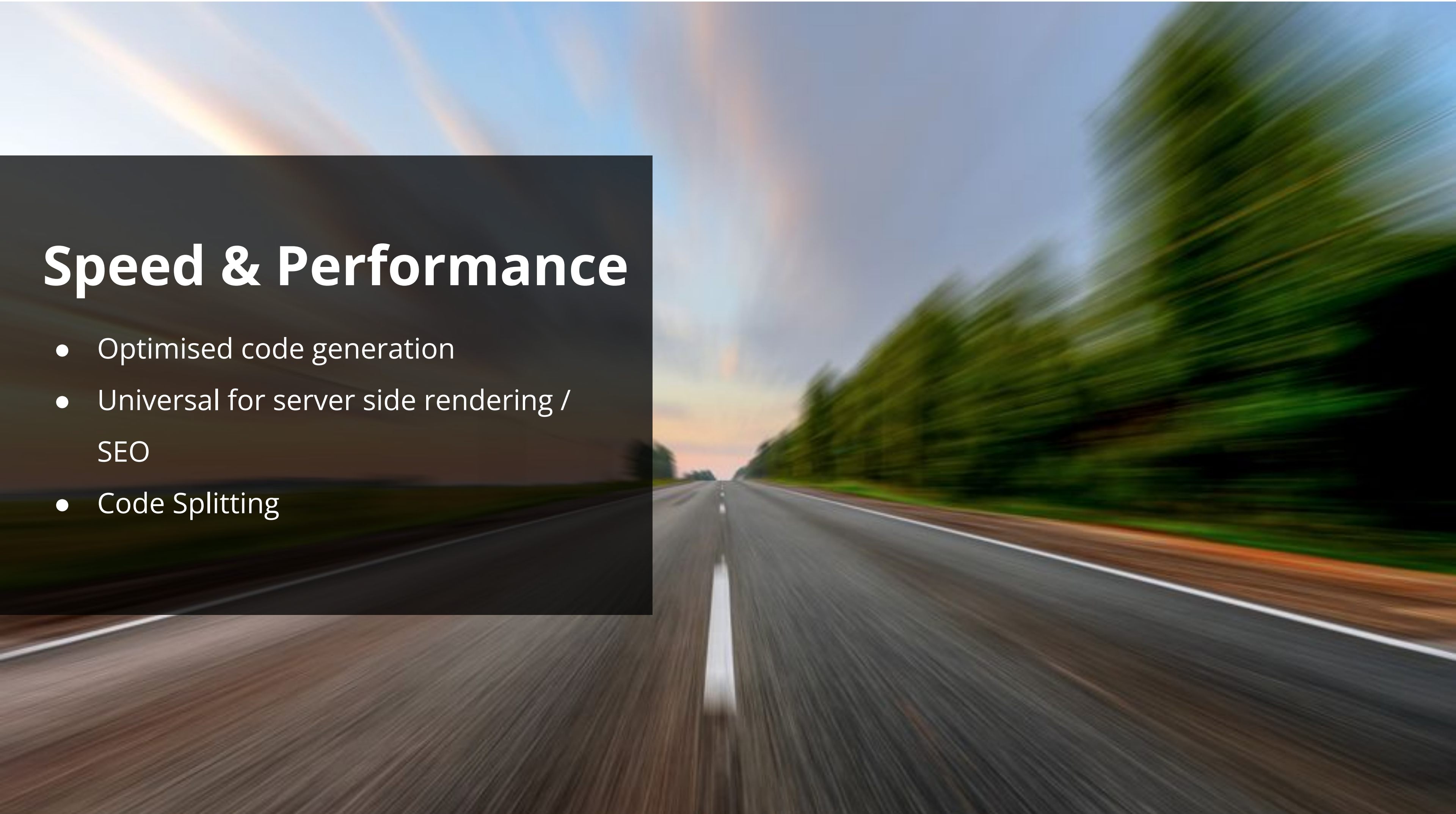
Who has using ngrx ?

Who has used nx ?

Who has written angular tests ?

Introductions

Name?
Angular experience?
What you would like to get from the workshop?

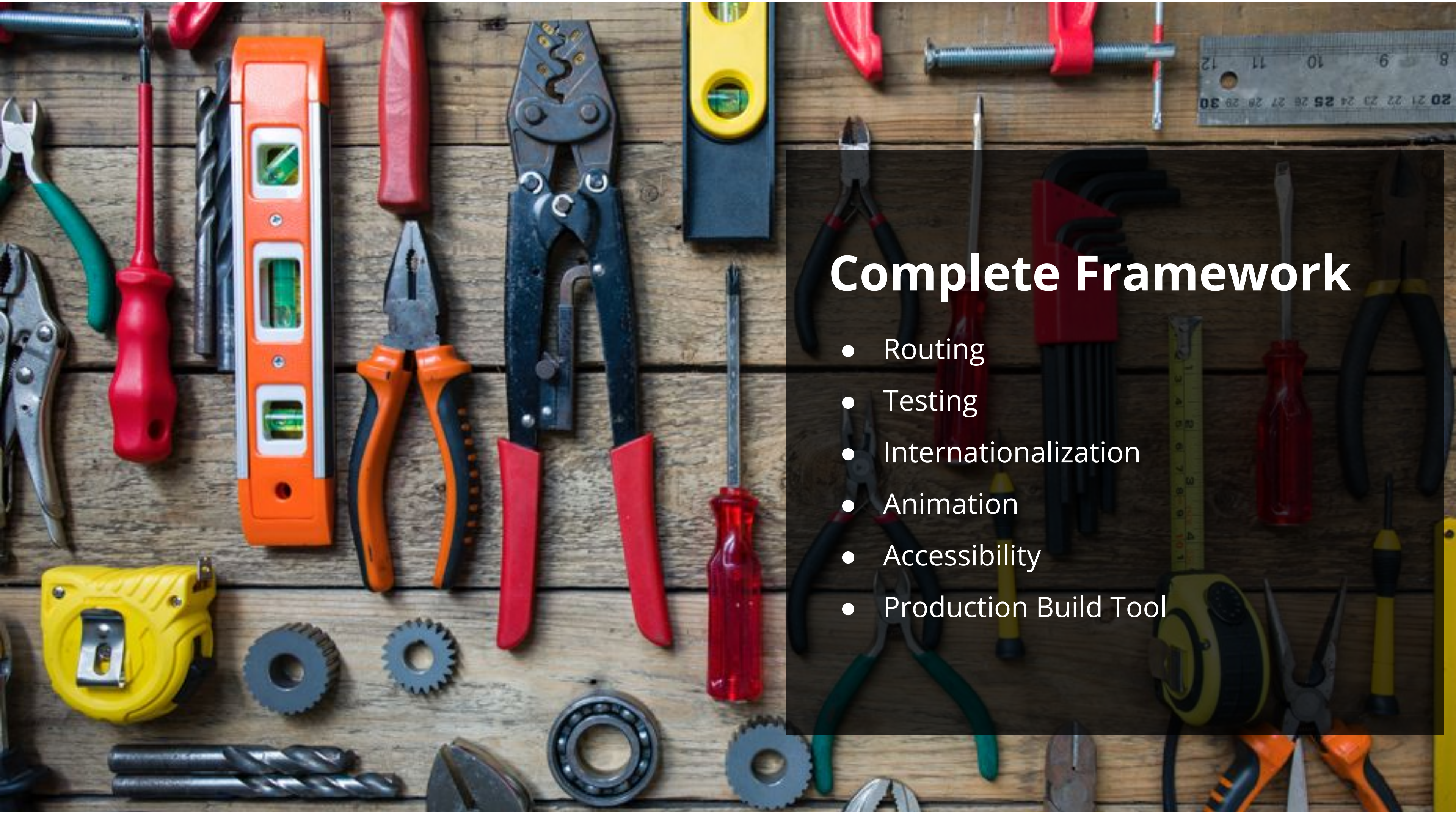


Speed & Performance

- Optimised code generation
- Universal for server side rendering / SEO
- Code Splitting

Productivity

- Component Model
- Template Syntax
- TypeScript
- Angular CLI



Complete Framework

- Routing
- Testing
- Internationalization
- Animation
- Accessibility
- Production Build Tool

Universal

Angular Elements

Progressive Web App

ngrx

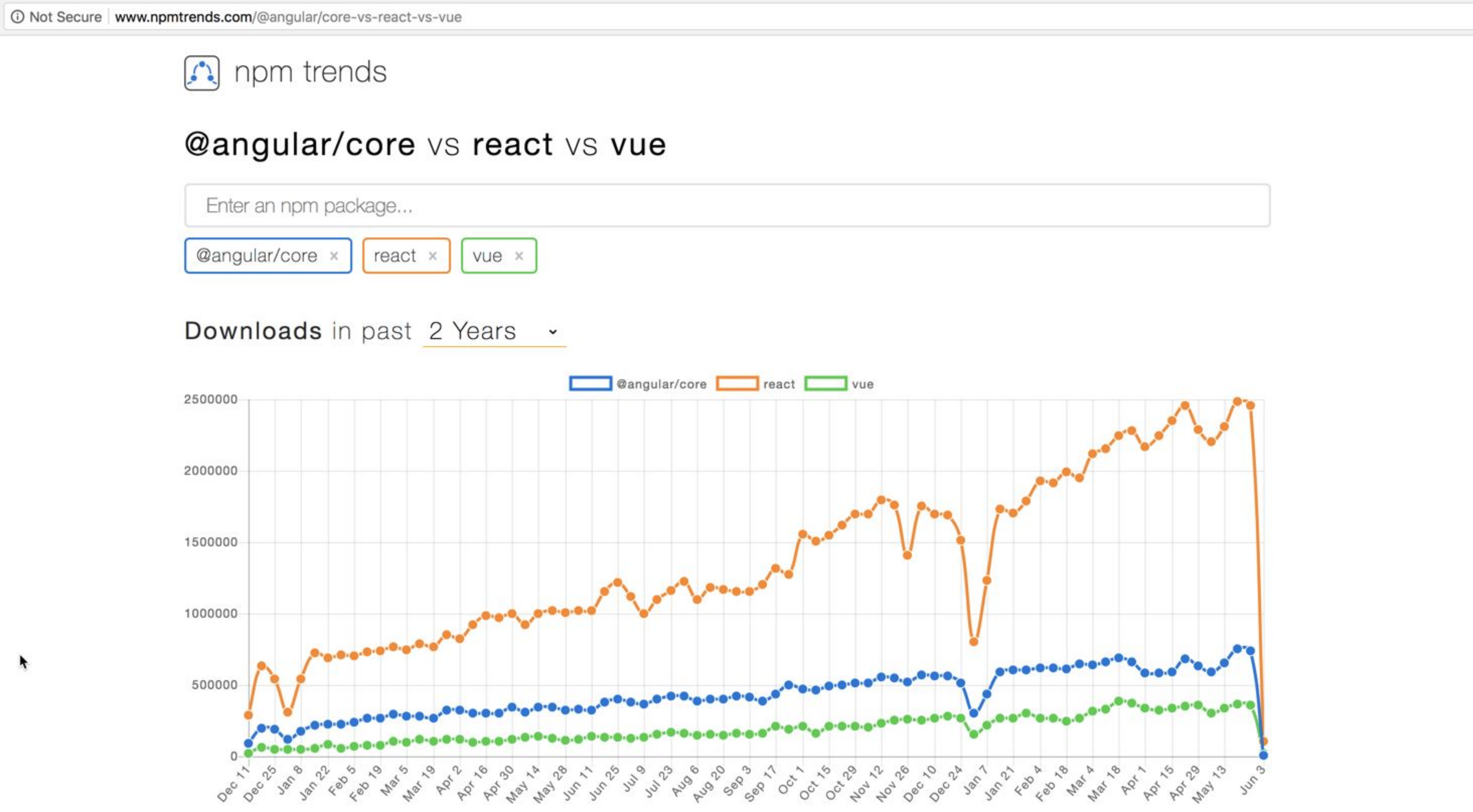
cli

Material

Component Dev Kit

Animations

i18n



ⓘ Not Secure | www.npmtrends.com/@angular/core-vs-react-vs-vue

npm trends

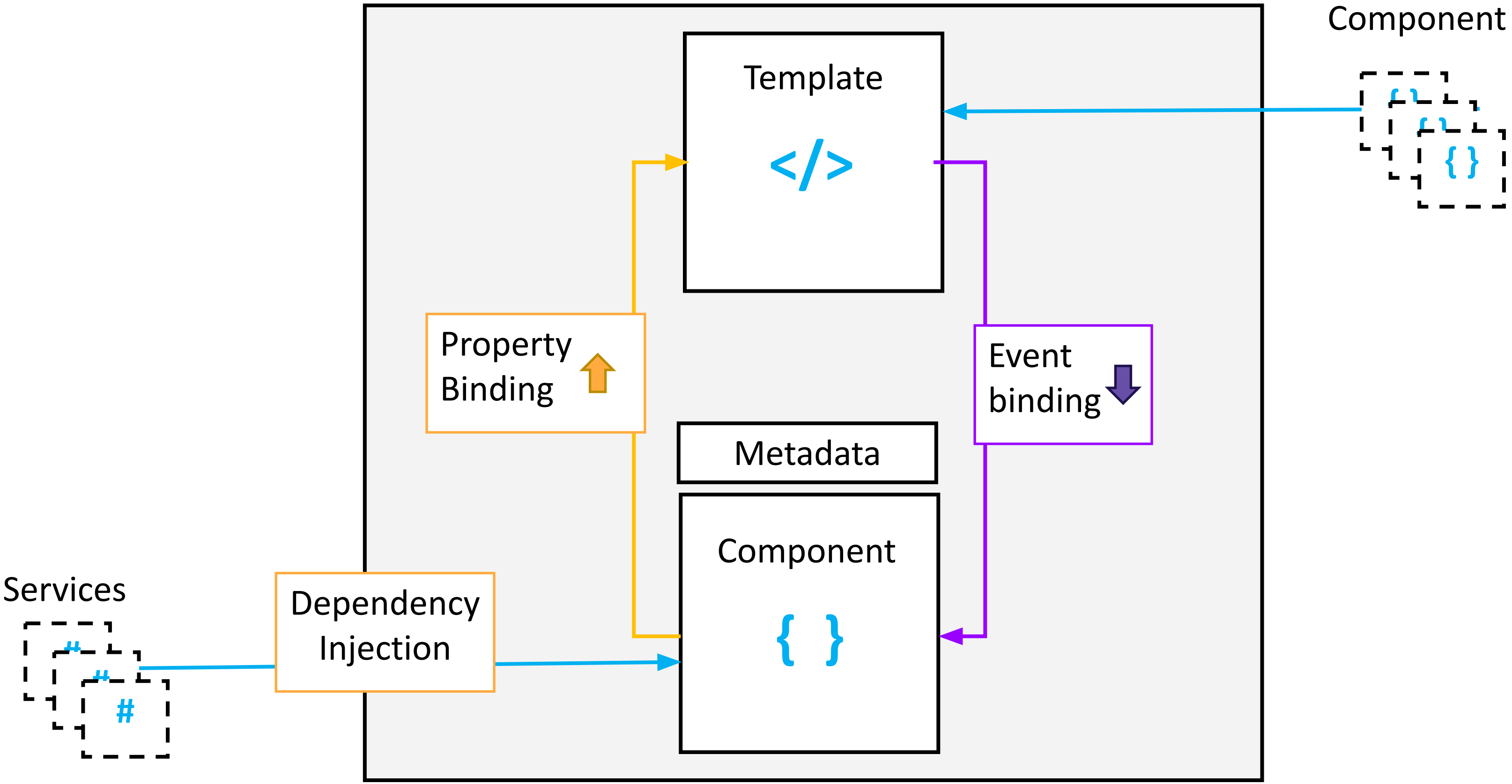
@angular/core vs **react** vs **vue**

Enter an npm package...

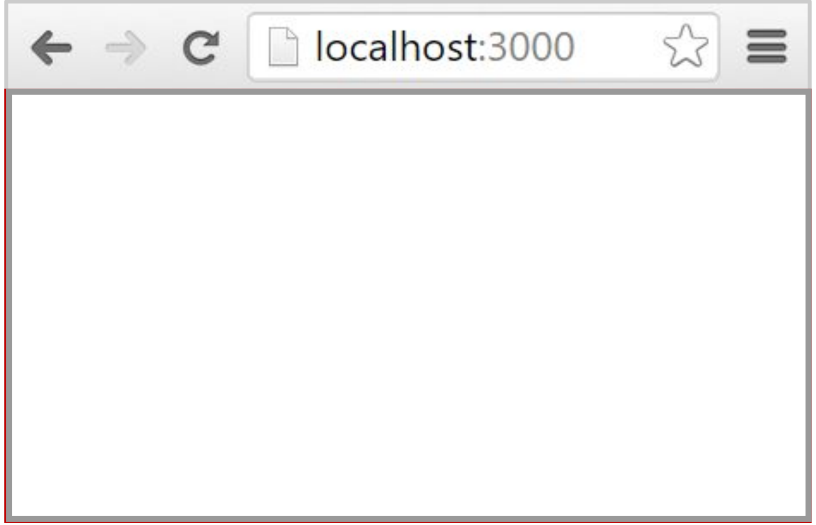
@angular/core × react × vue ×

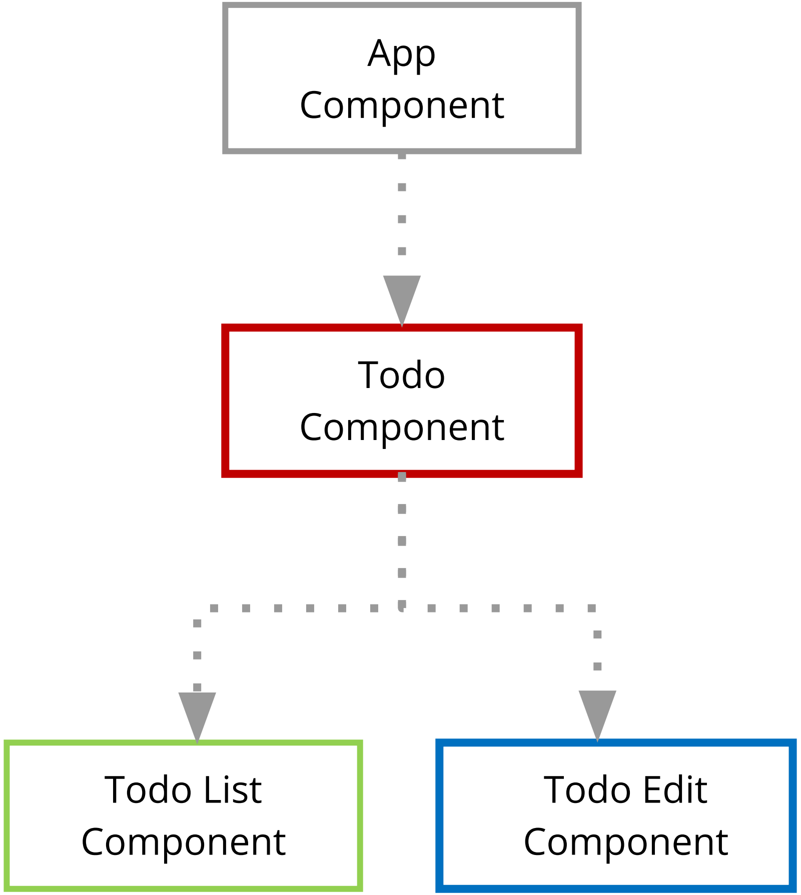
Downloads in past 2 Years

Component Architecture



Component Tree





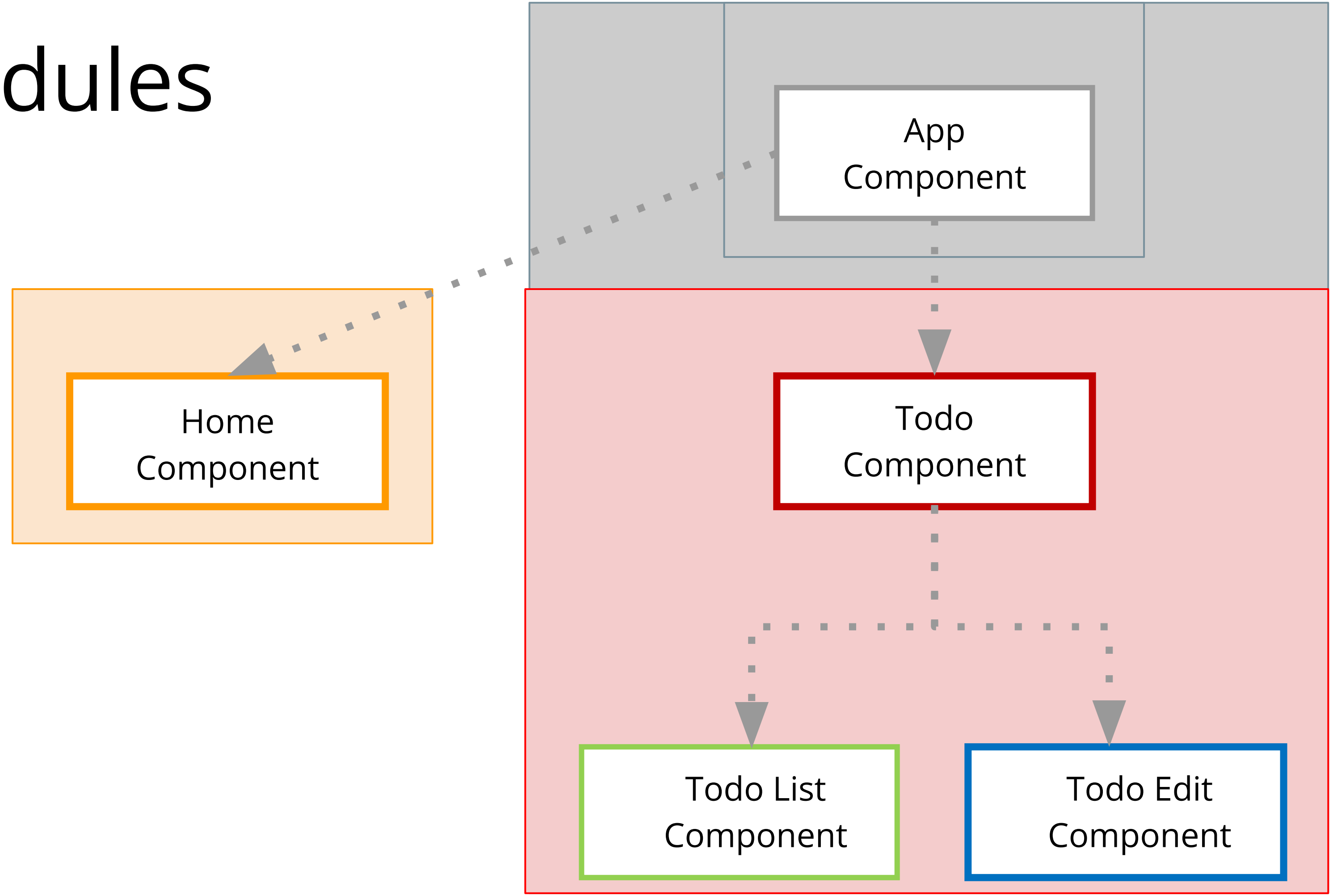
App Component

Todo Component

Todo List Component

Todo Edit Component

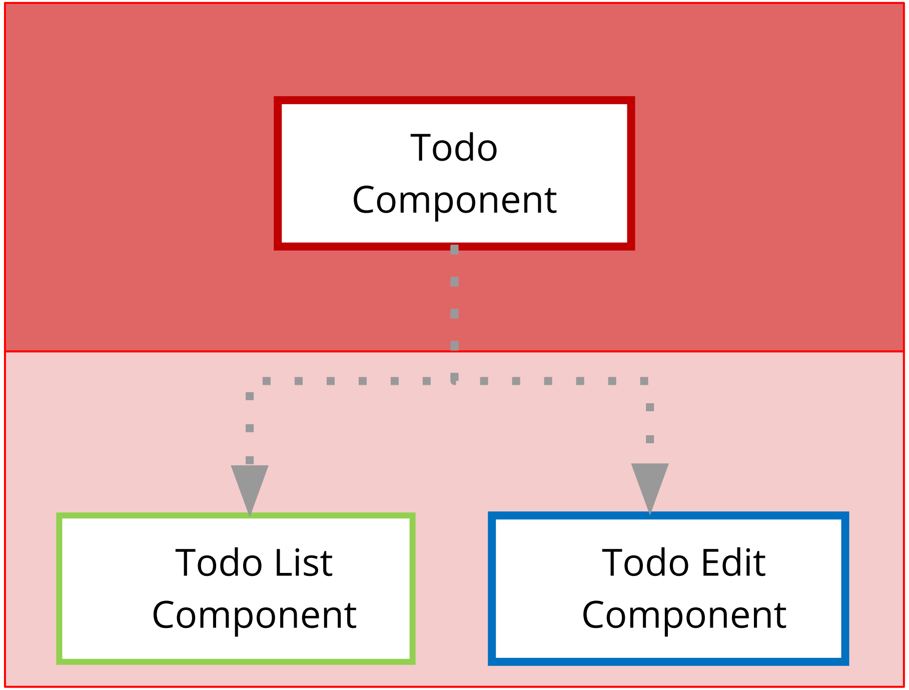
NgModules

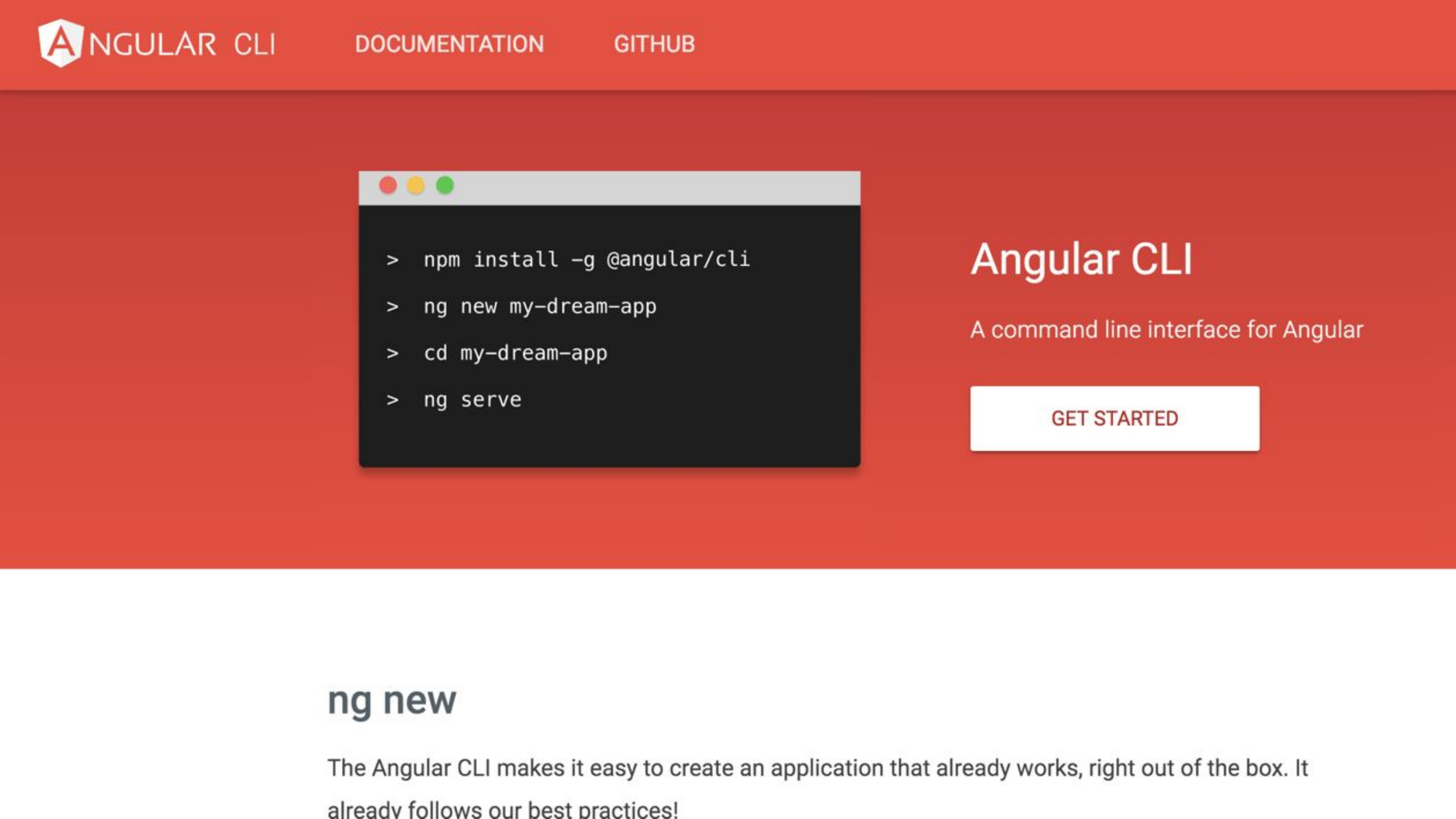

App Component
Home Component
Todo Component
Todo List Component
Todo Edit Component

Presentational and Container Components

Container components

Presentational components





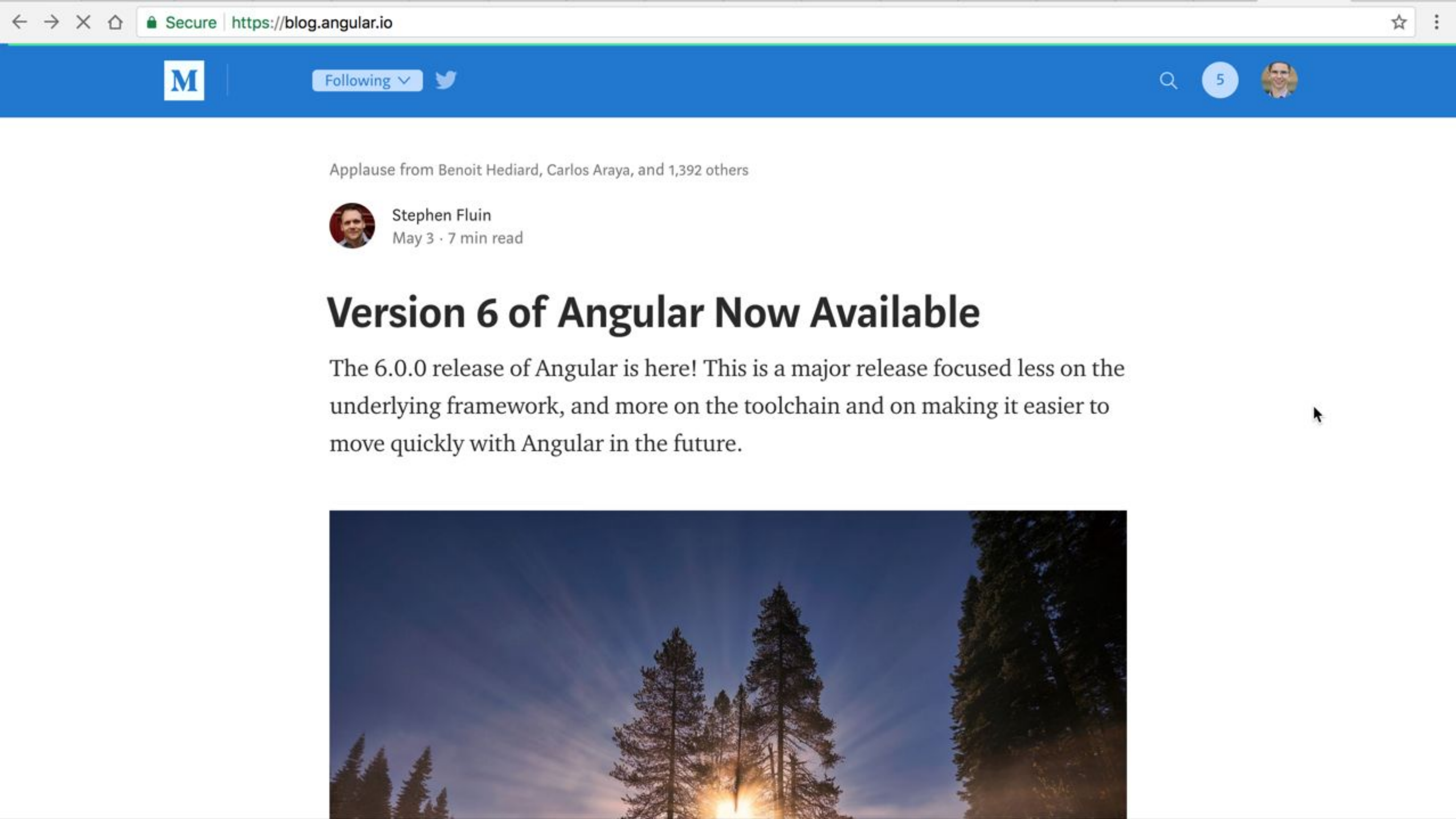
```
>  npm install -g @angular/cli
>  ng new my-dream-app
>  cd my-dream-app
>  ng serve
```

Angular CLI

A command line interface for Angular

ng new

The Angular CLI makes it easy to create an application that already works, right out of the box. It already follows our best practices!



Applause from Benoit Hediard, Carlos Araya, and 1,392 others

Stephen Fluin
May 3 · 7 min read

Version 6 of Angular Now Available

The 6.0.0 release of Angular is here! This is a major release focused less on the underlying framework, and more on the toolchain and on making it easier to move quickly with Angular in the future.

New in V6

ngUpdate

ngAdd

Angular elements

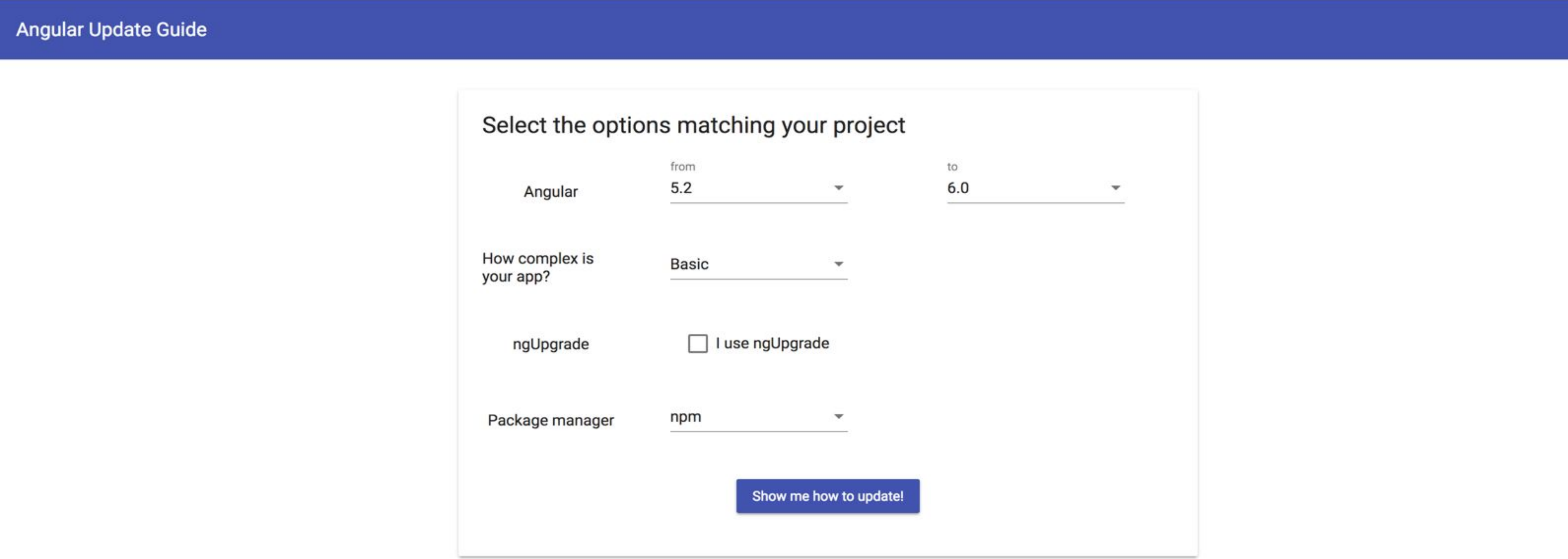
Angular material, CDK and generated templates

CLI workspaces

Library support

RxJS v6

LTS, Ivy renderer, Animation perf, Treeshackable providers



Angular Update Guide

Select the options matching your project

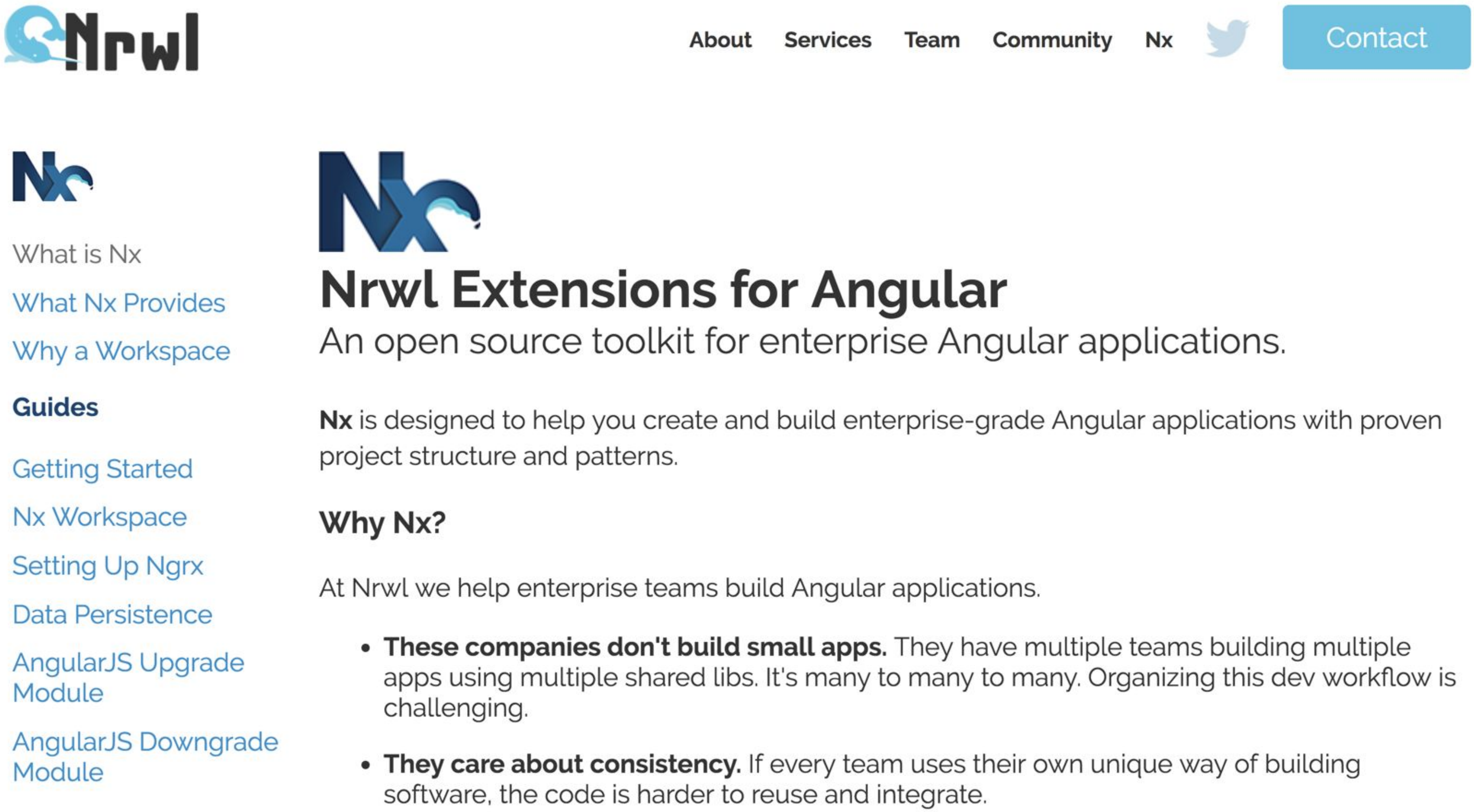
Angular from 5.2 to 6.0

How complex is your app? Basic

ngUpgrade □ I use ngUpgrade

Package manager npm

We are using v6



Nrwl Extensions for Angular

An open source toolkit for enterprise Angular applications.

Nx is designed to help you create and build enterprise-grade Angular applications with proven project structure and patterns.

Why Nx?

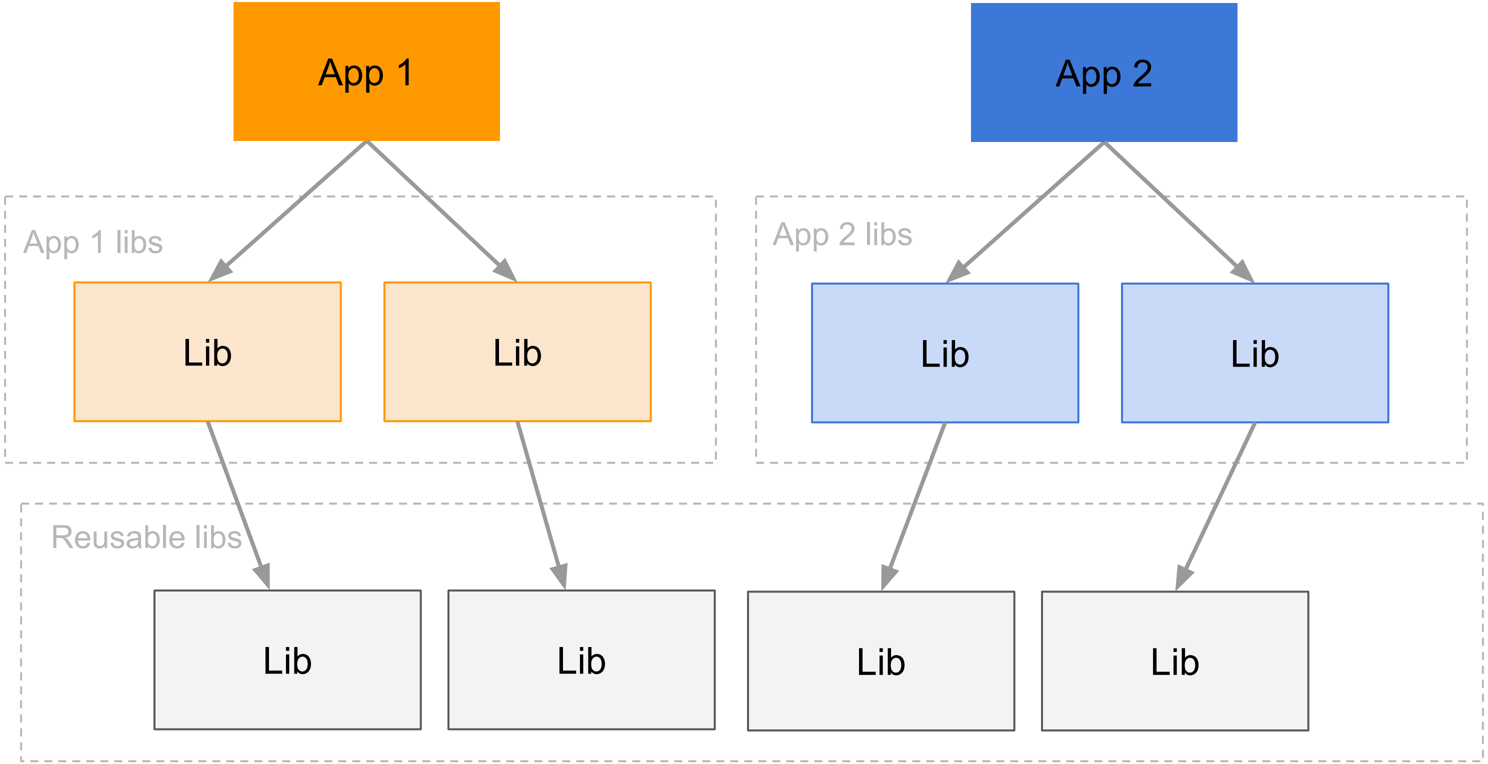
At Nrwl we help enterprise teams build Angular applications.

- **These companies don't build small apps.** They have multiple teams building multiple apps using multiple shared libs. It's many to many to many. Organizing this dev workflow is challenging.
- **They care about consistency.** If every team uses their own unique way of building software, the code is harder to reuse and integrate.

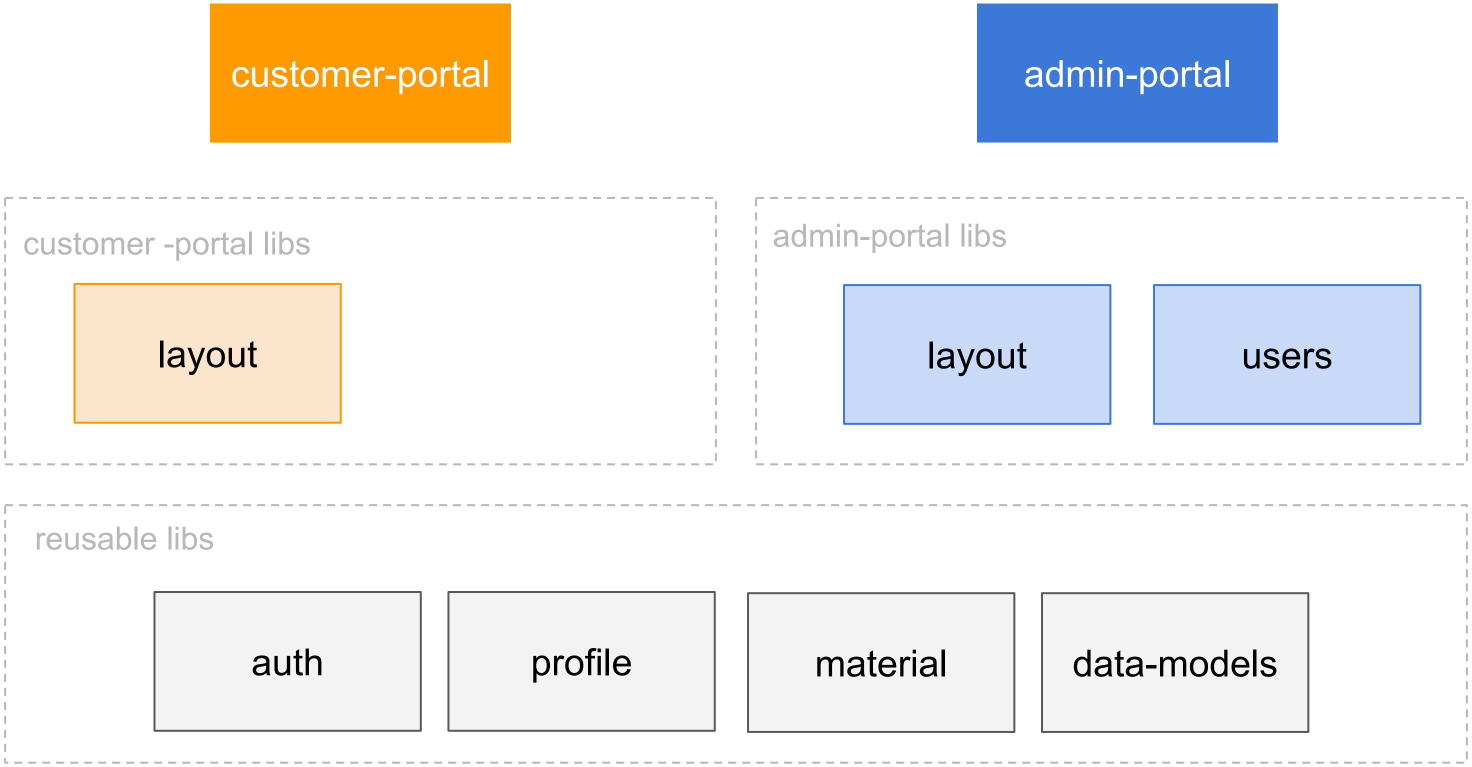
Nx = Supercharged Angular CLI

Nx is about composing apps from libs

Nx Workspaces



Nx Workspaces



customer-portal

admin-portal

customer -portal libs

layout

admin-portal libs

layout

users

reusable libs

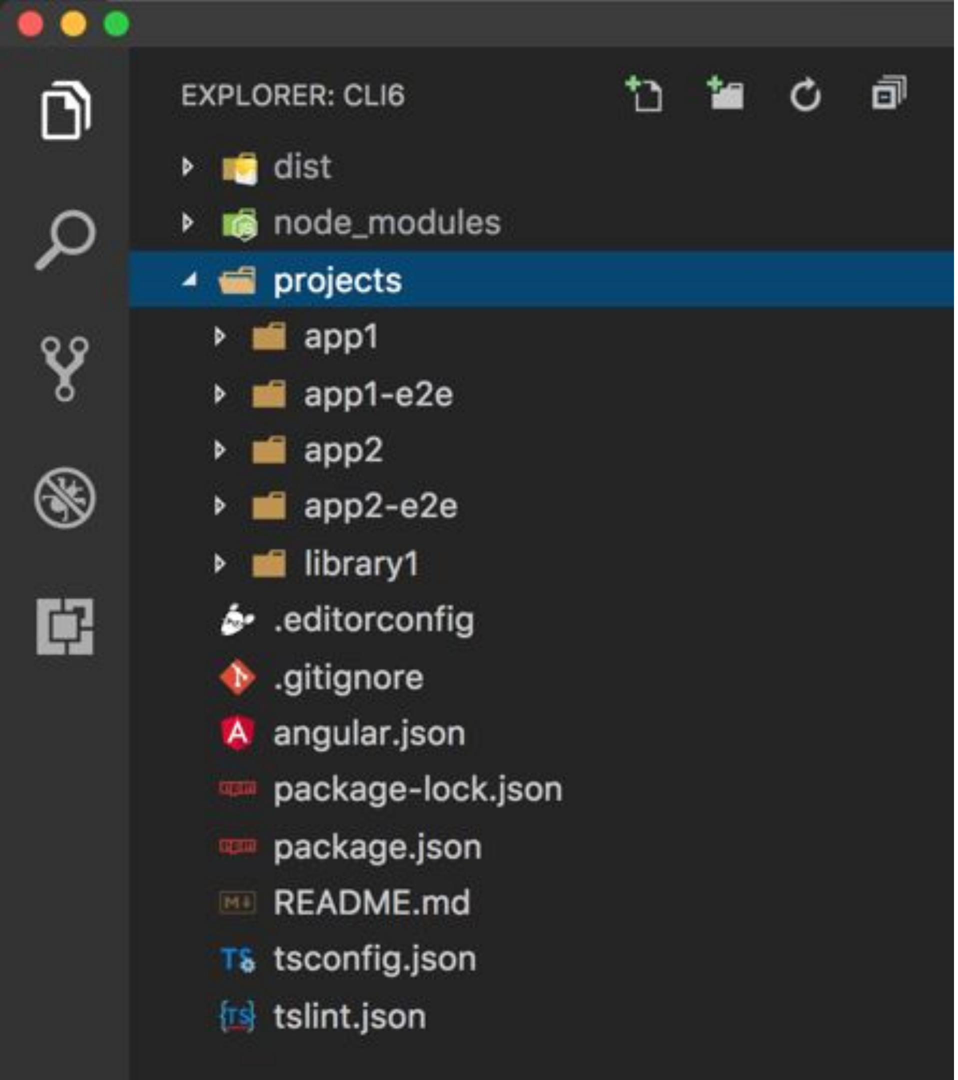
auth

profile

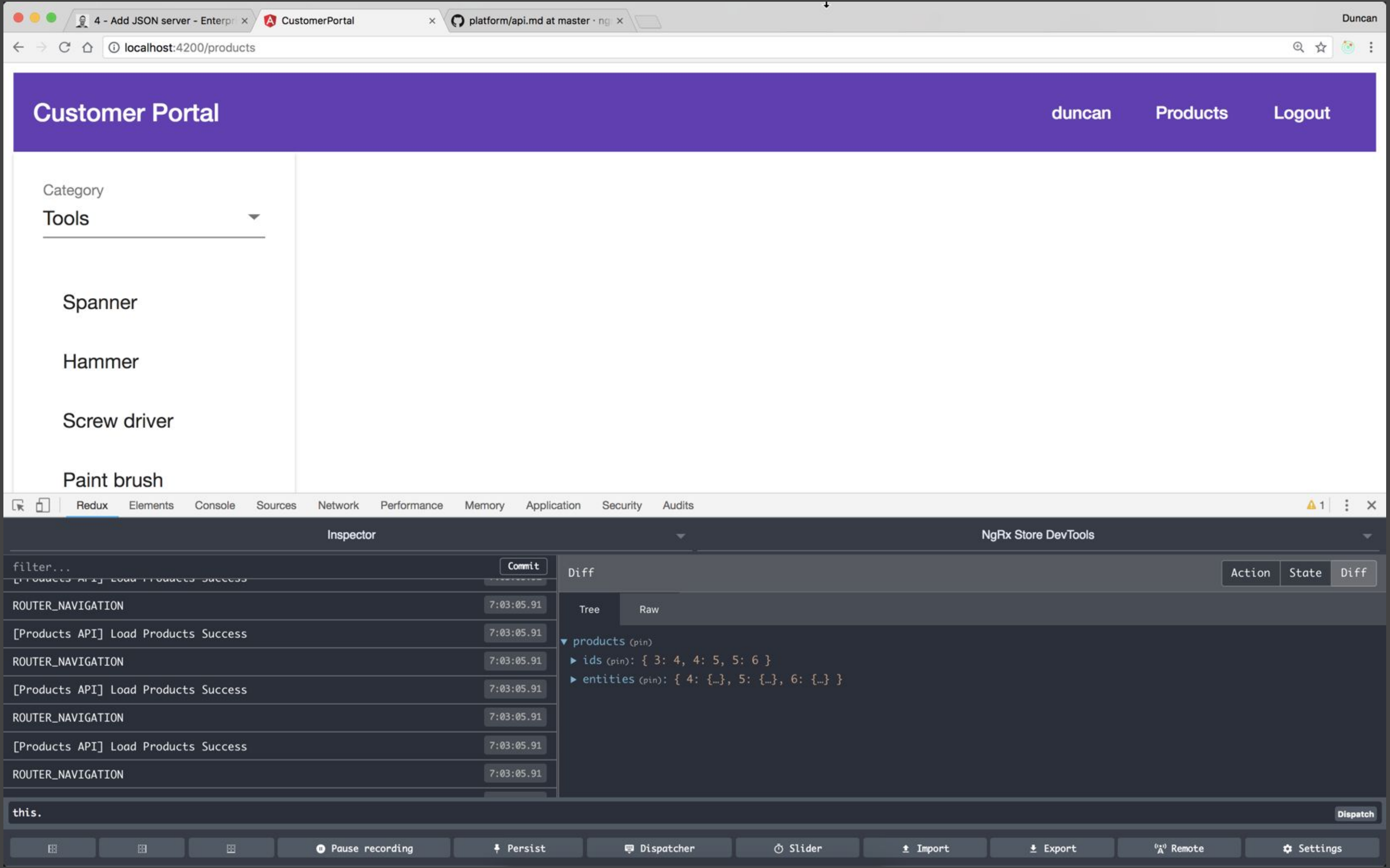
material

data-models

Why not just use the new CLI workspaces?



No libs sharing
No mono repo support
No advanced linting
No opinionated scaffolding for NgRx



Category

Tools

Spanner

Hammer

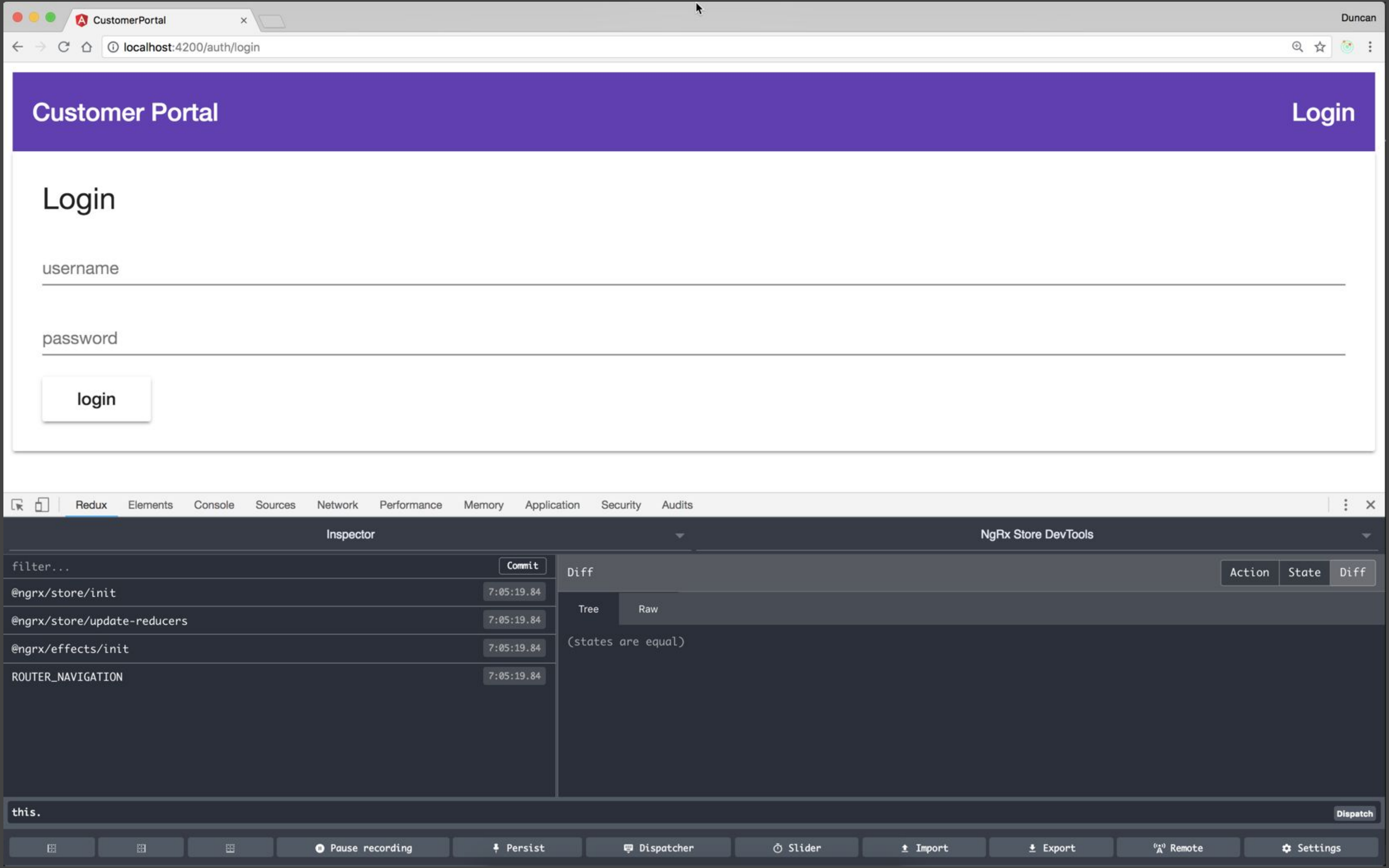
Screw driver

Paint brush

```
ROUTER_NAVIGATION                          7:03:05.91
[Products API] Load Products Success       7:03:05.91
ROUTER_NAVIGATION                          7:03:05.91
[Products API] Load Products Success       7:03:05.91
ROUTER_NAVIGATION                          7:03:05.91
[Products API] Load Products Success       7:03:05.91
ROUTER_NAVIGATION                          7:03:05.91
```

```
▼ products (pin)
  ▶ ids (pin): { 3: 4, 4: 5, 5: 6 }
  ▶ entities (pin): { 4: {…}, 5: {…}, 6: {…} }
```

```
this.
```



Customer Portal

Login

Login

username

password

login

filter...

```
@ngrx/store/init                7:05:19.84
@ngrx/store/update-reducers     7:05:19.84
@ngrx/effects/init              7:05:19.84
ROUTER_NAVIGATION               7:05:19.84
```

Diff

(states are equal)

this.

Angular itself is easy

… but it forces you to confront modern web development.

TypeScript

RxJS

Modules

Redux

Summary - Why Angular

Latest web tools and patterns

Super fast

Huge community

Enterprise friendly

Summary - Why Nx

Mono Repo Support

NgRx takes logic out of components.

Presentational and Container Components

Division of components into two categories

Presentational and Container Components

Presentational	Container
Concerned with how things look	Concerned with how things work
HTML markup and CSS styles	Have little to no HTML and CSS styles
No dependencies on the rest of the app	Have injected dependencies
Don’t specify how data is loaded or changed but emit events via @Outputs	Are stateful and specify how data is loaded or changed
Receive data via @Inputs	Top level routes
May contain other components	May contain other components

Benefits of Presentational and Container Components

Performance

Composability

Easier to test

Benefits of Container and Presentational Components

Performance

Composability

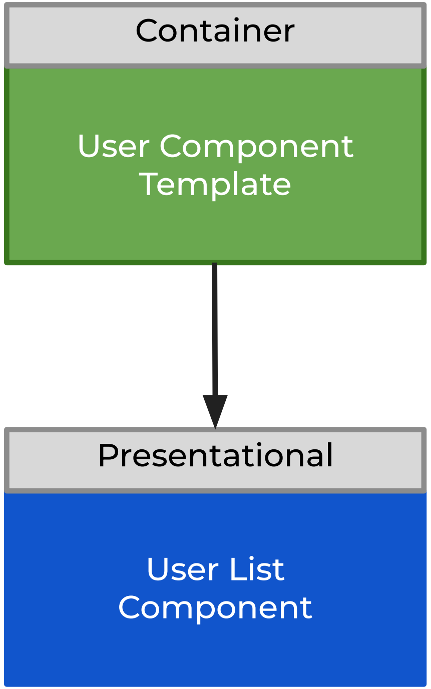
Easier to test

ChangeDetectionStrategy.OnPush

OnPush means that the change detector's mode will be initially set to CheckOnce

ChangeDetectionStrategy.OnPush

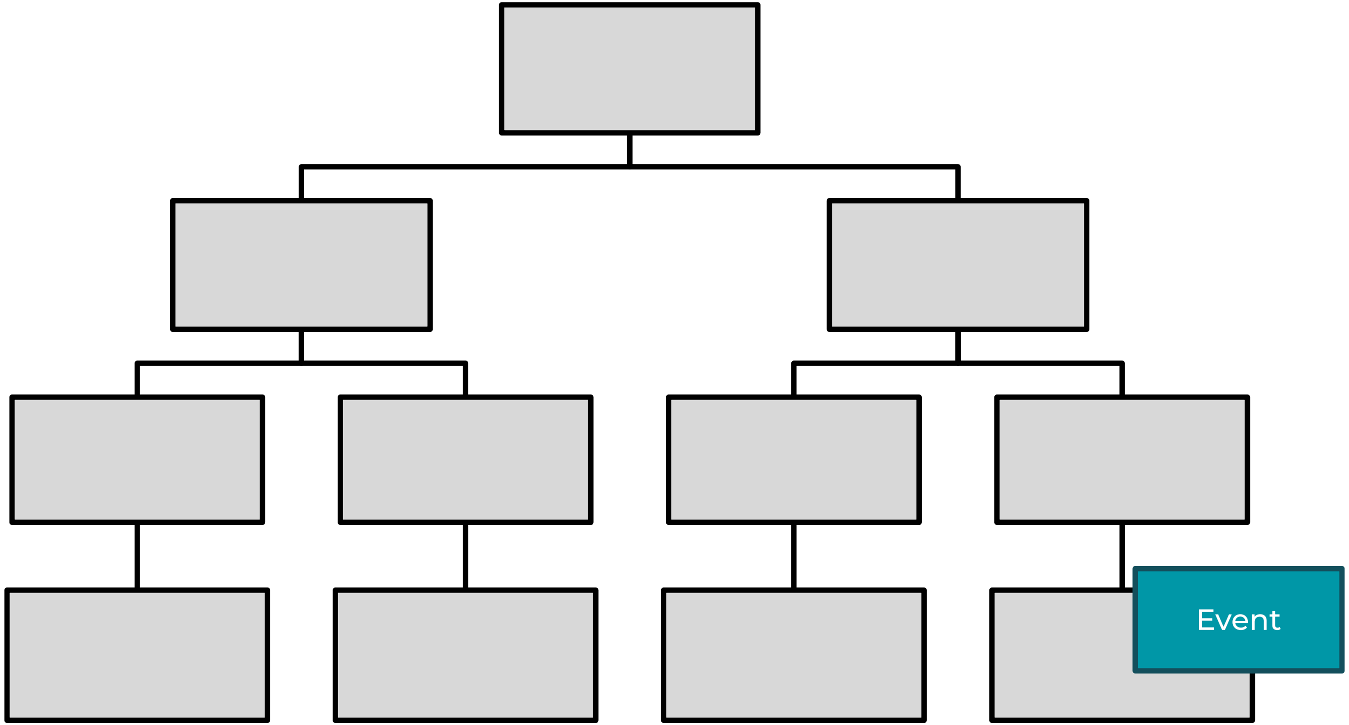
```
<app-user
   [users]="users$ | async">
</app-user>
```



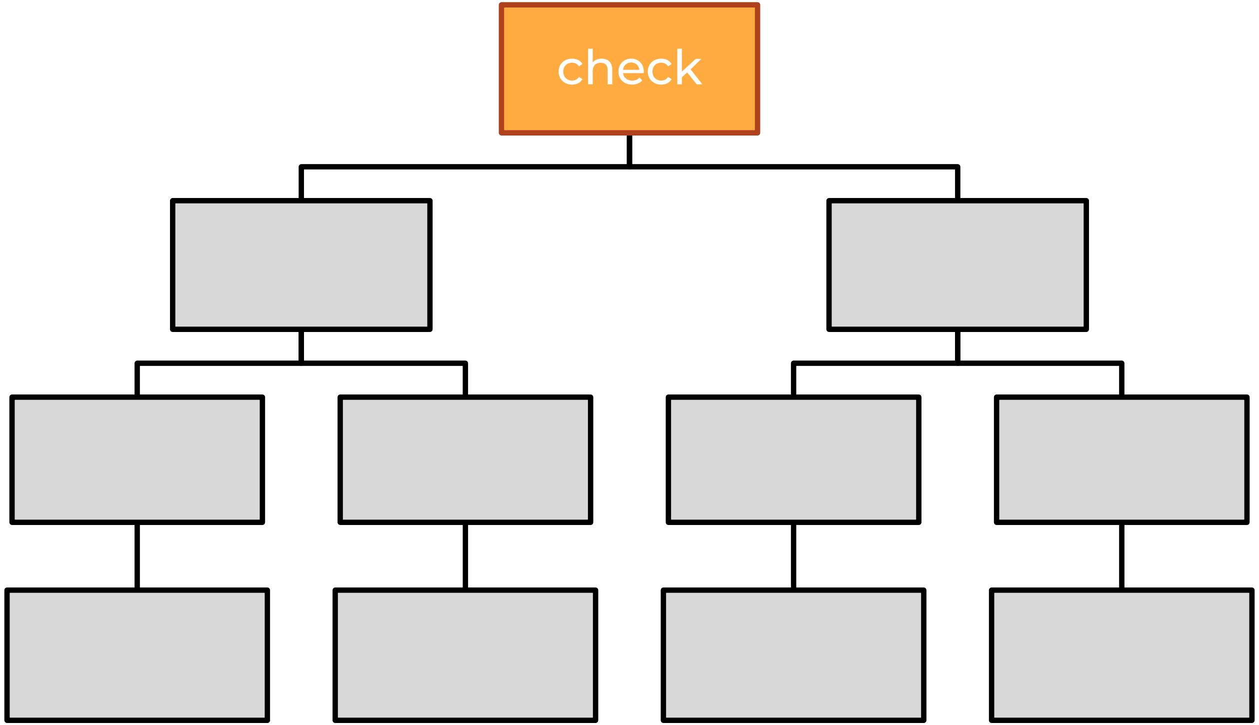
```
export class UserListComponent {
   @Input() users: User[];
}
```

ChangeDetectionStrategy.Default

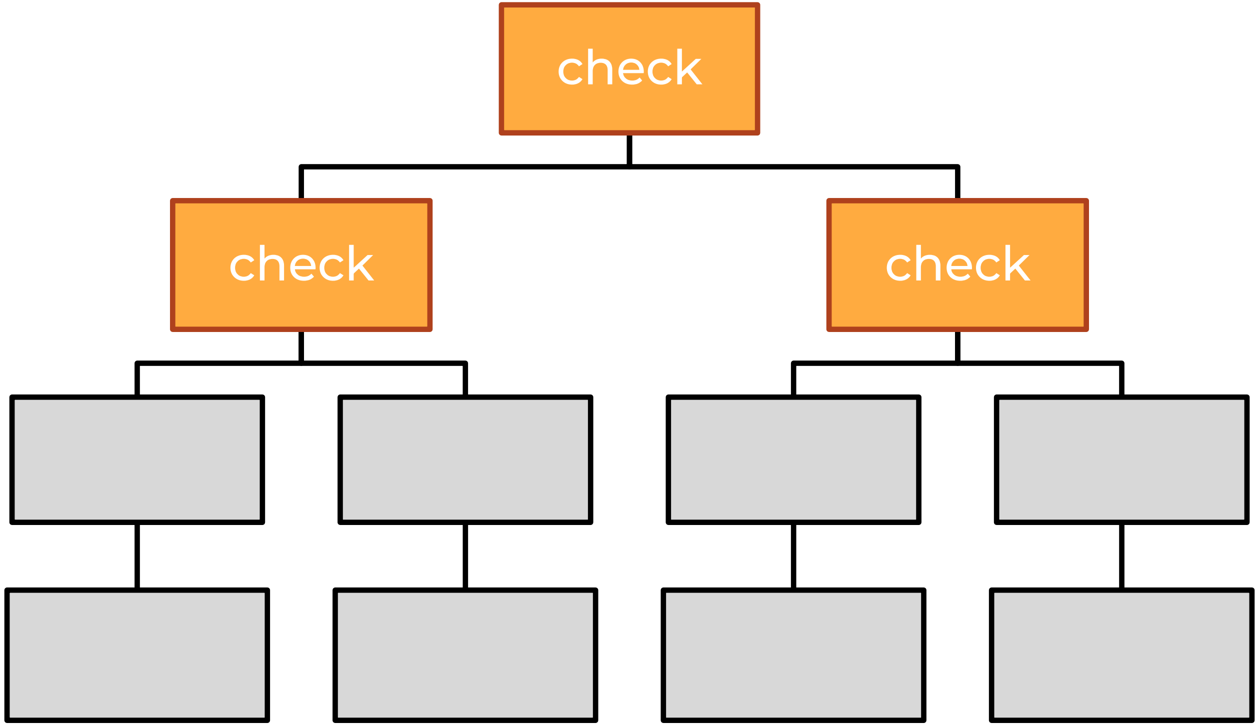
Change Detection



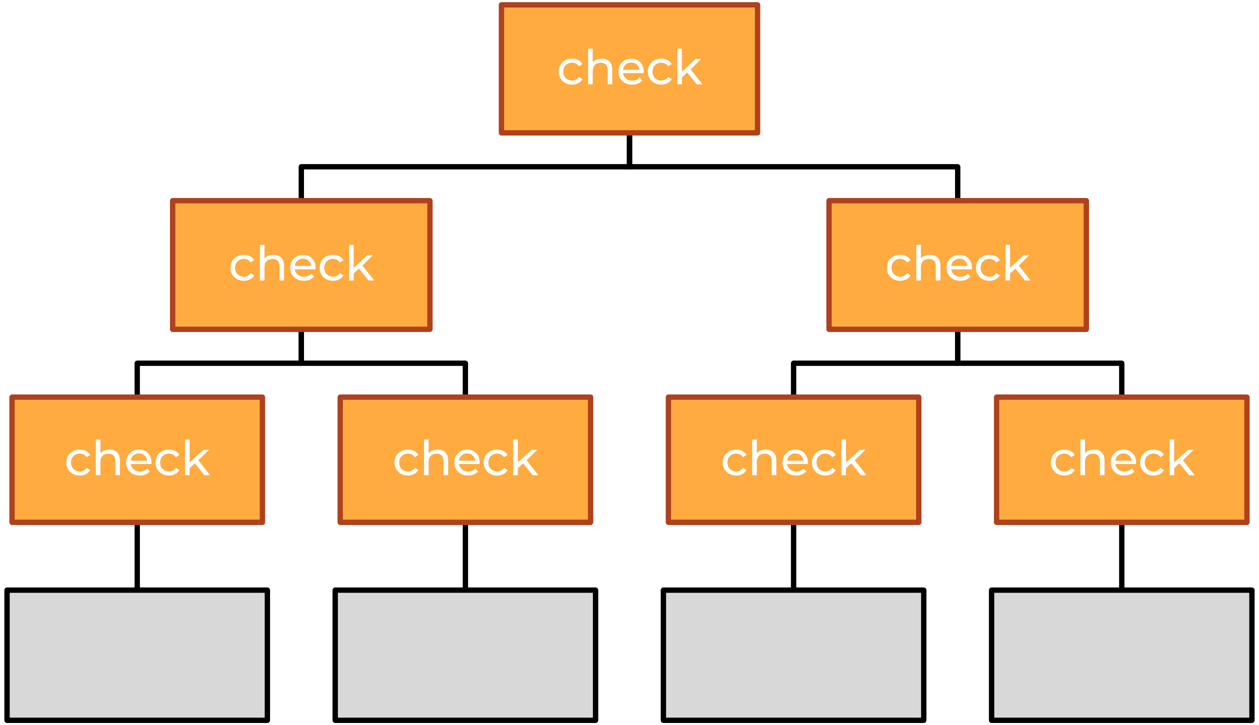
Change Detection



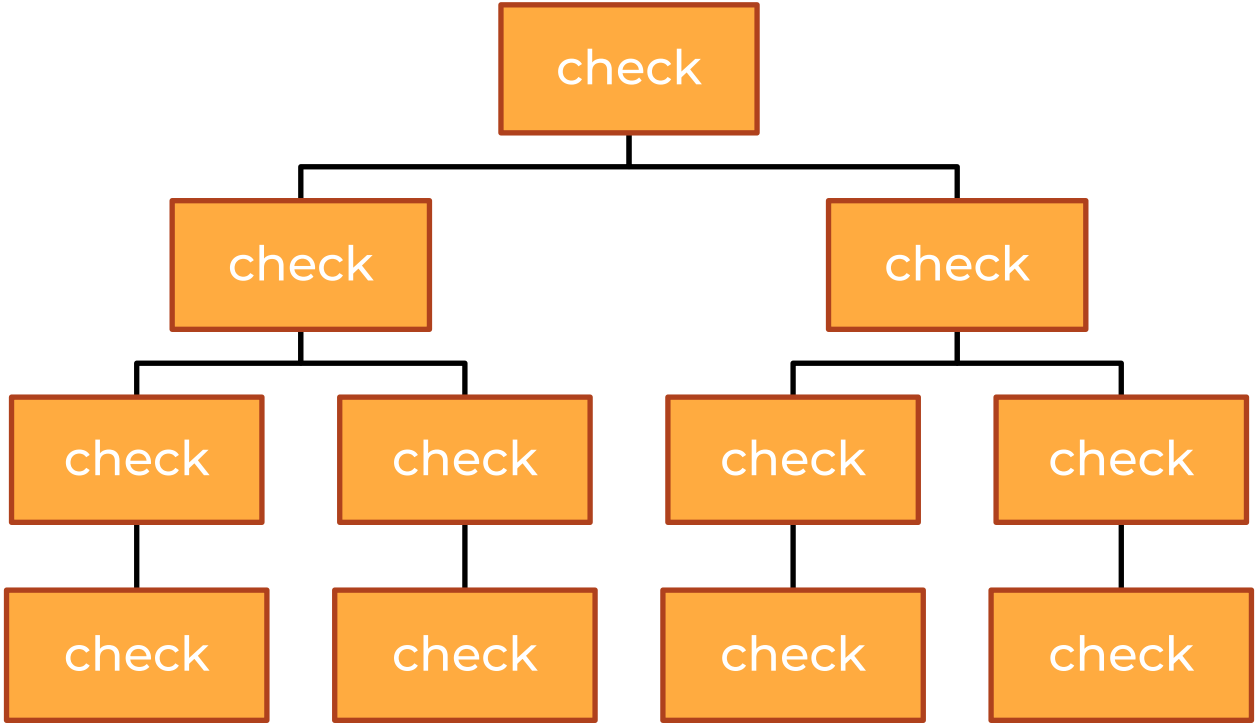
Change Detection



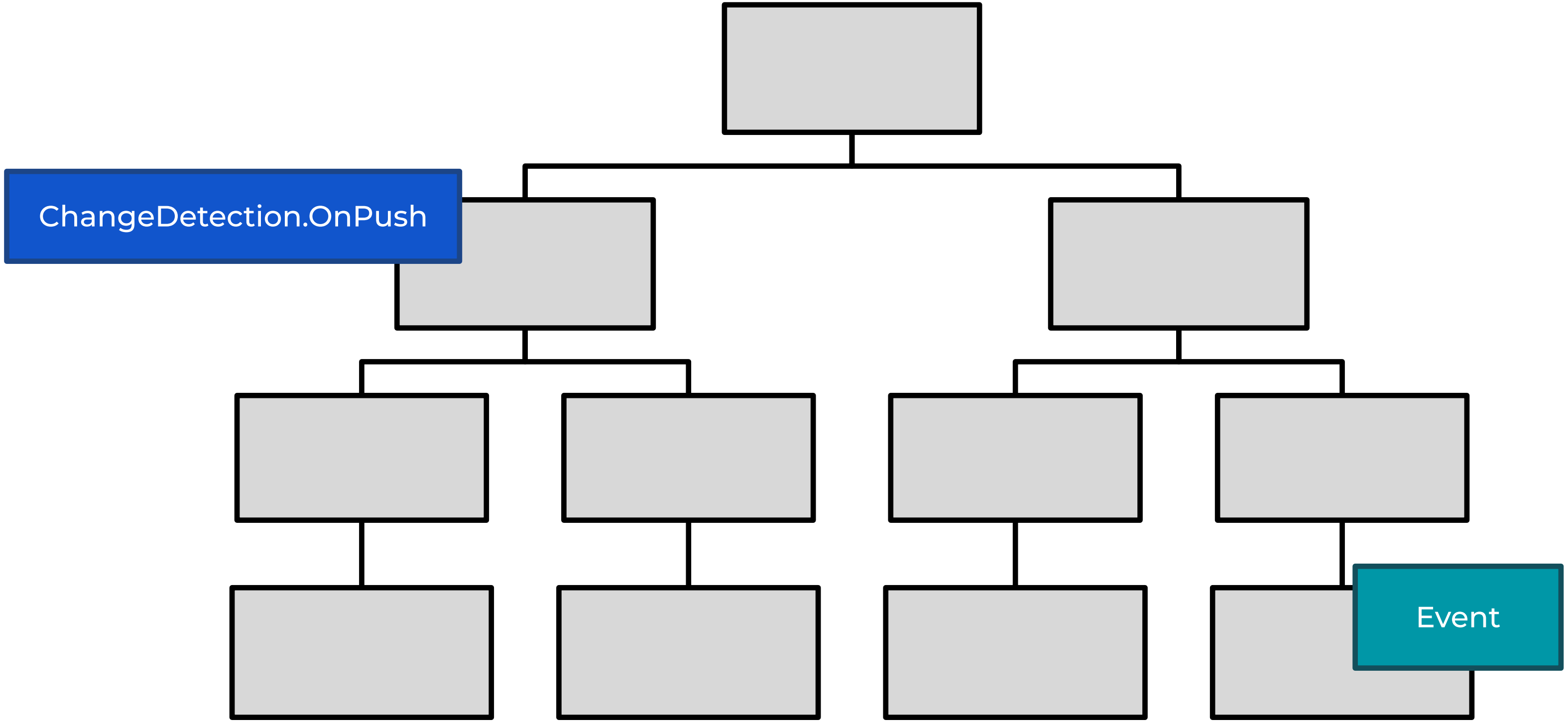
Change Detection



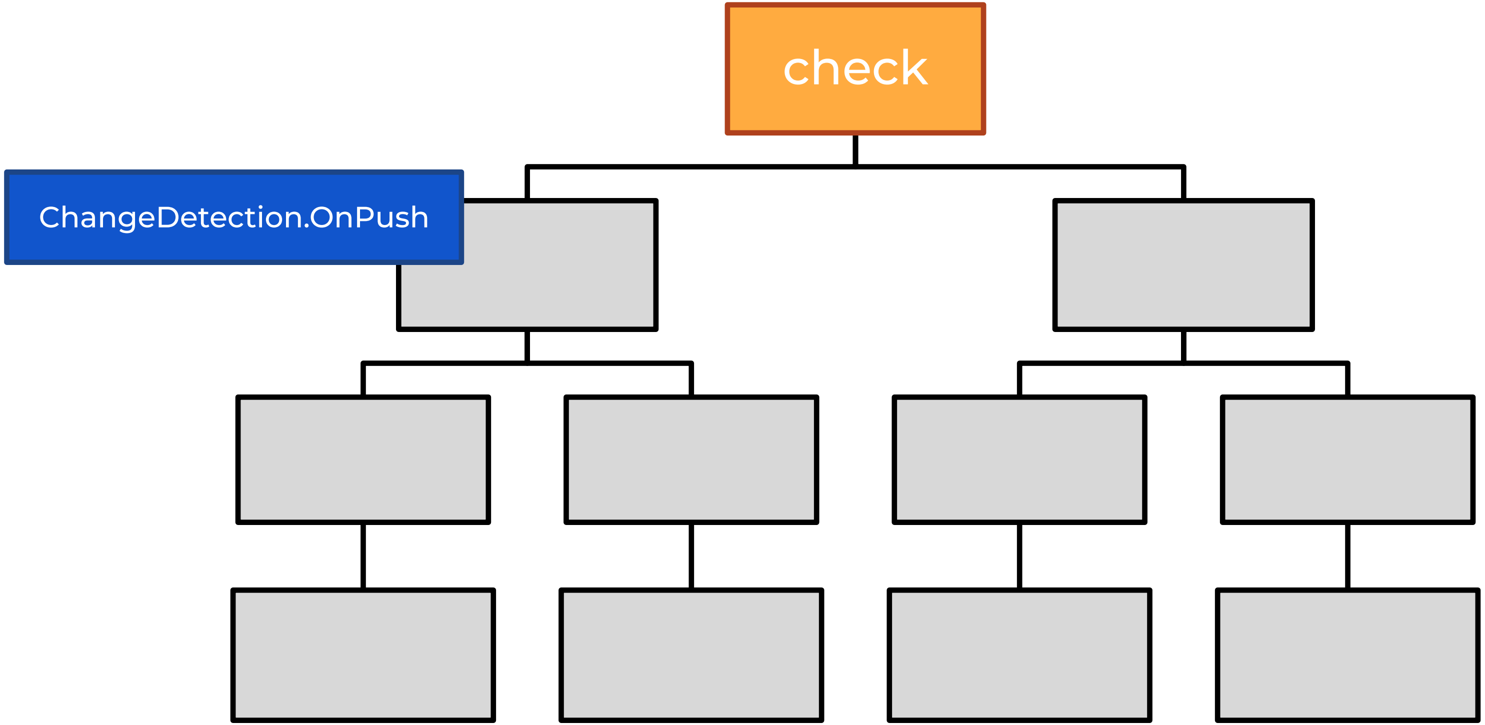
Change Detection



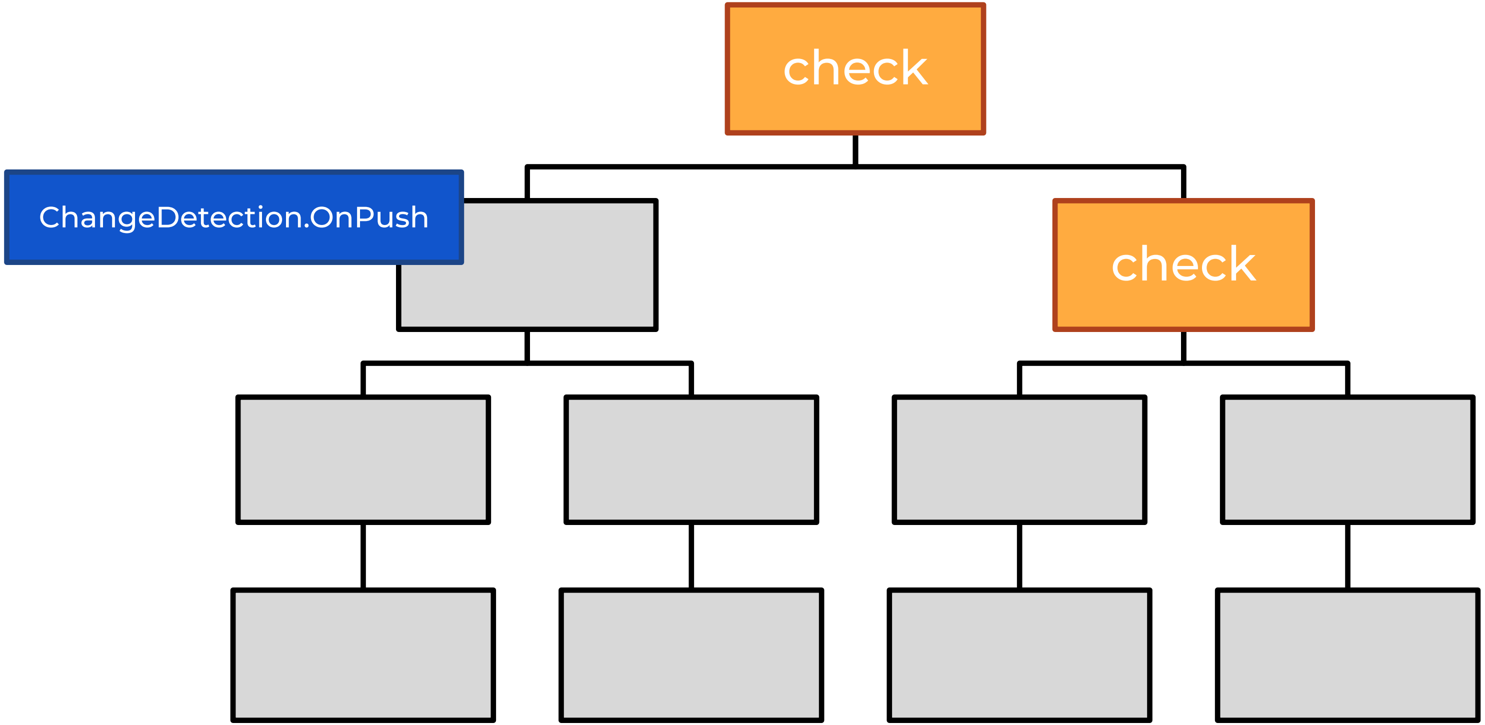
Change Detection - OnPush



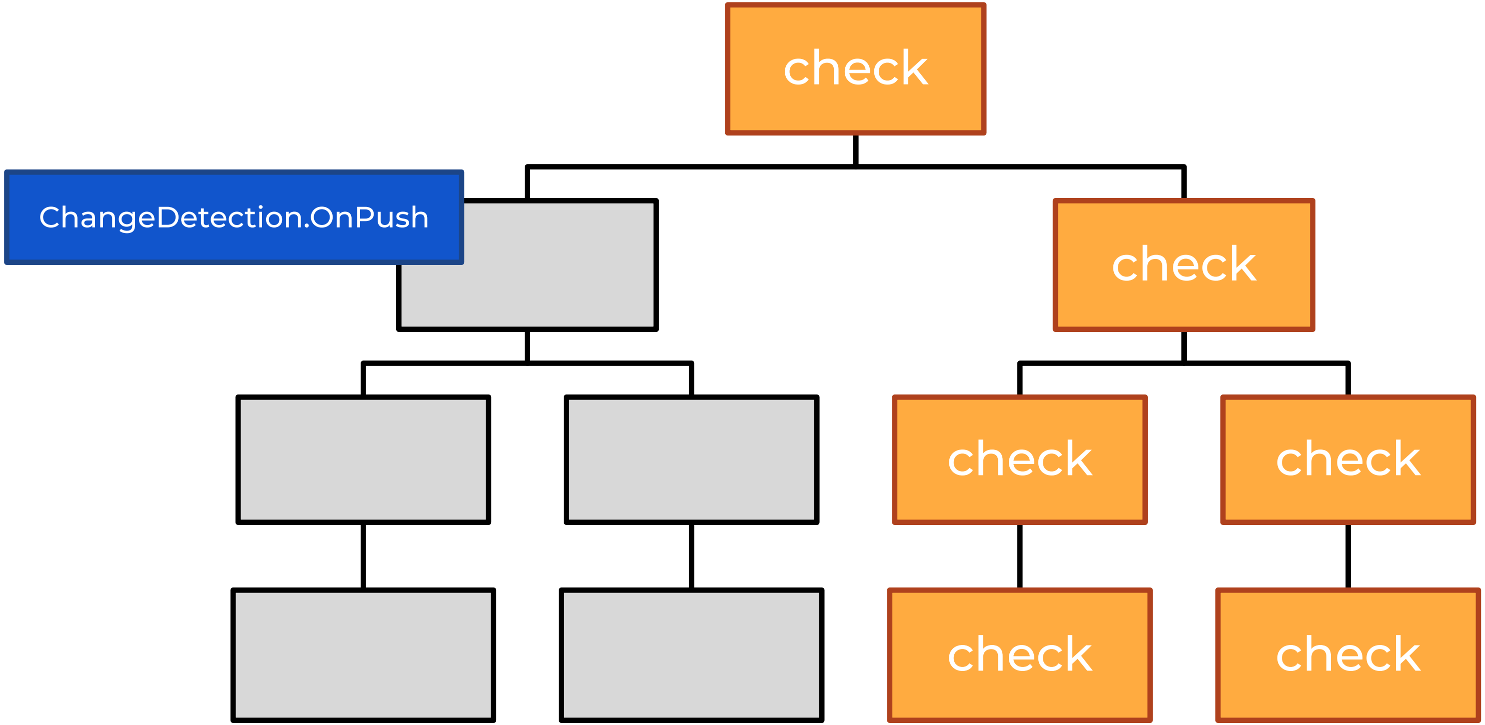
Change Detection - OnPush

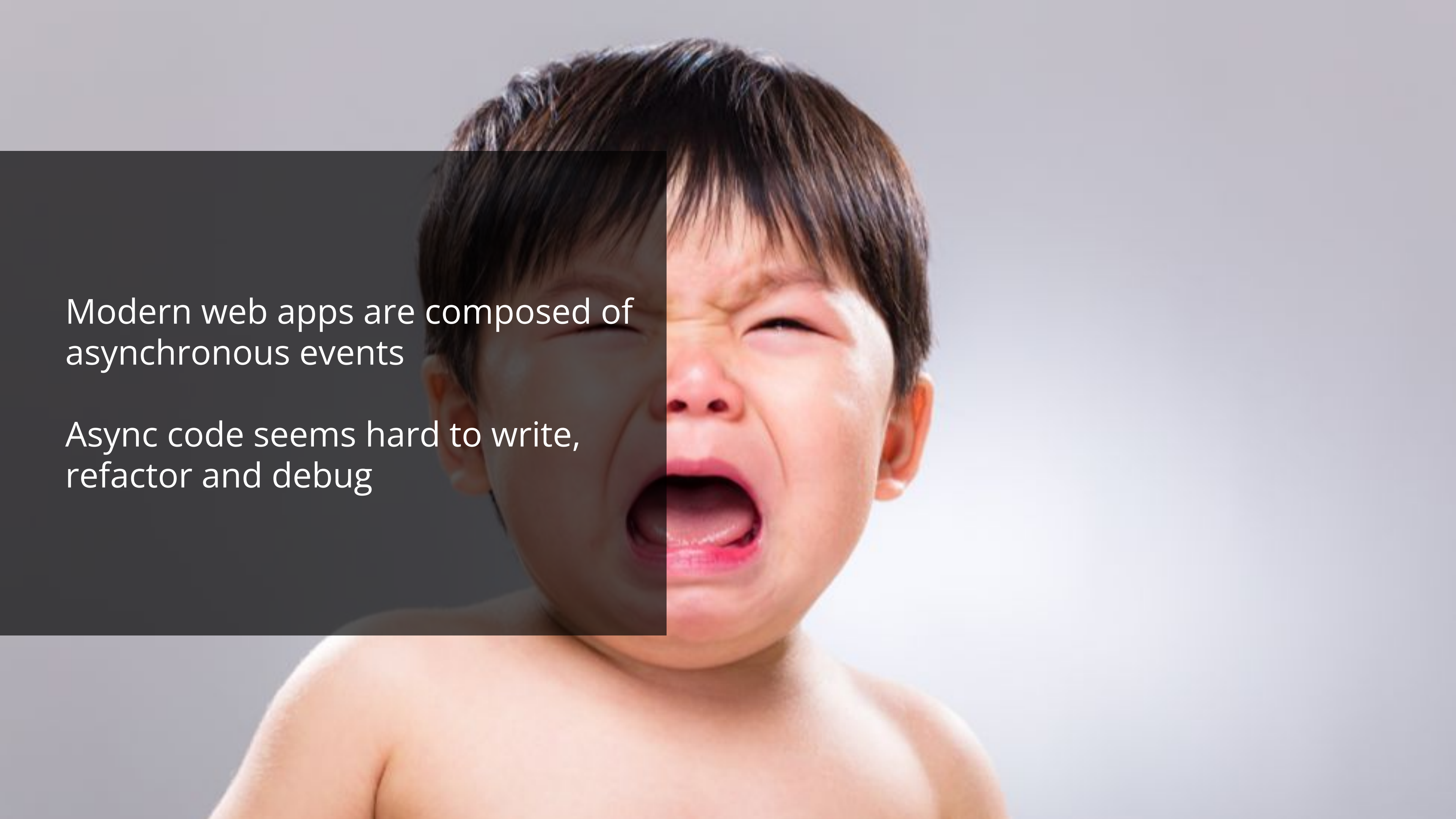


Change Detection - OnPush



Change Detection - OnPush





Modern web apps are composed of asynchronous events

Async code seems hard to write, refactor and debug

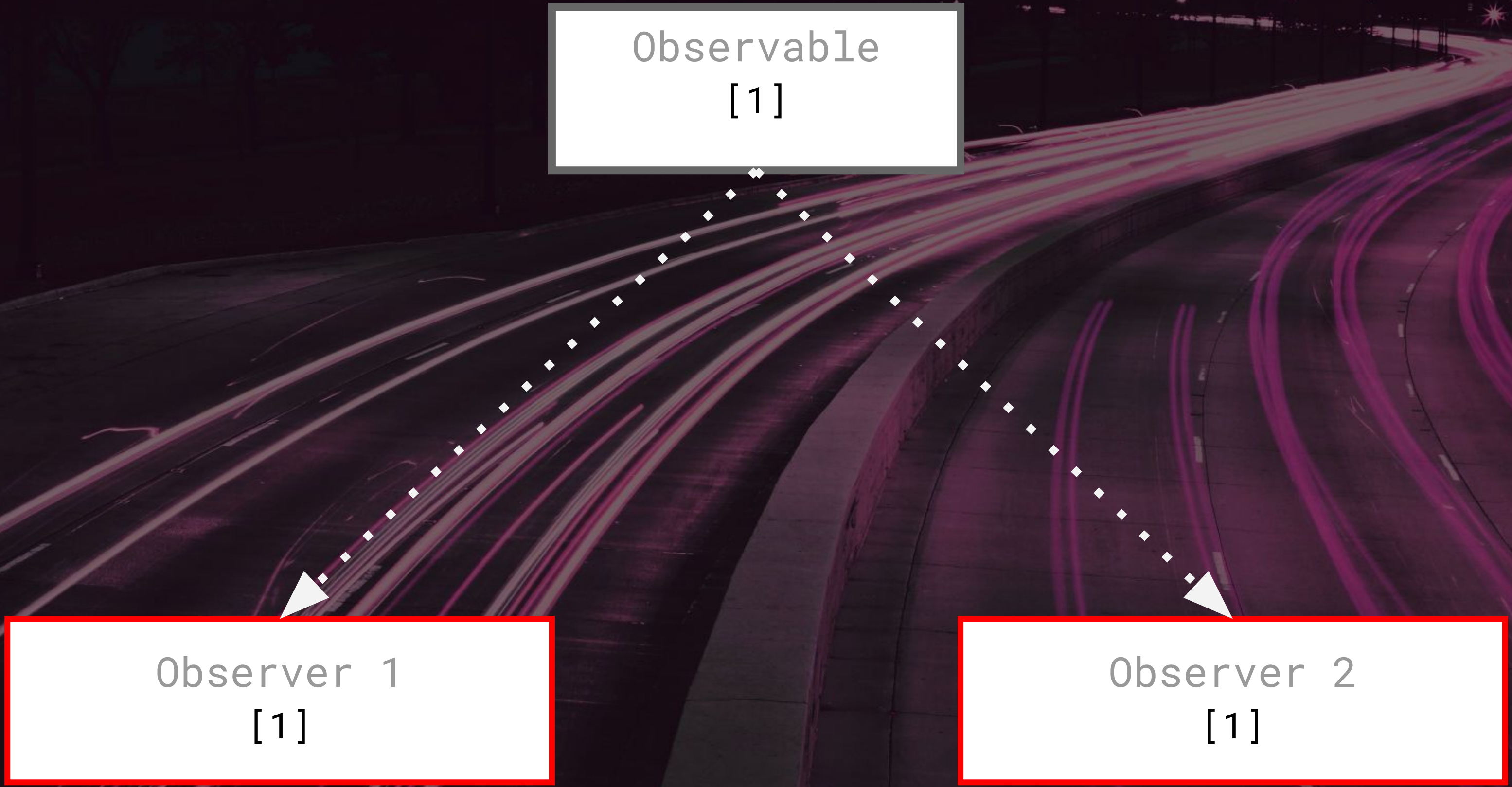
RxJS simplifies async code

RxJS

RxJS

An API for asynchronous programming

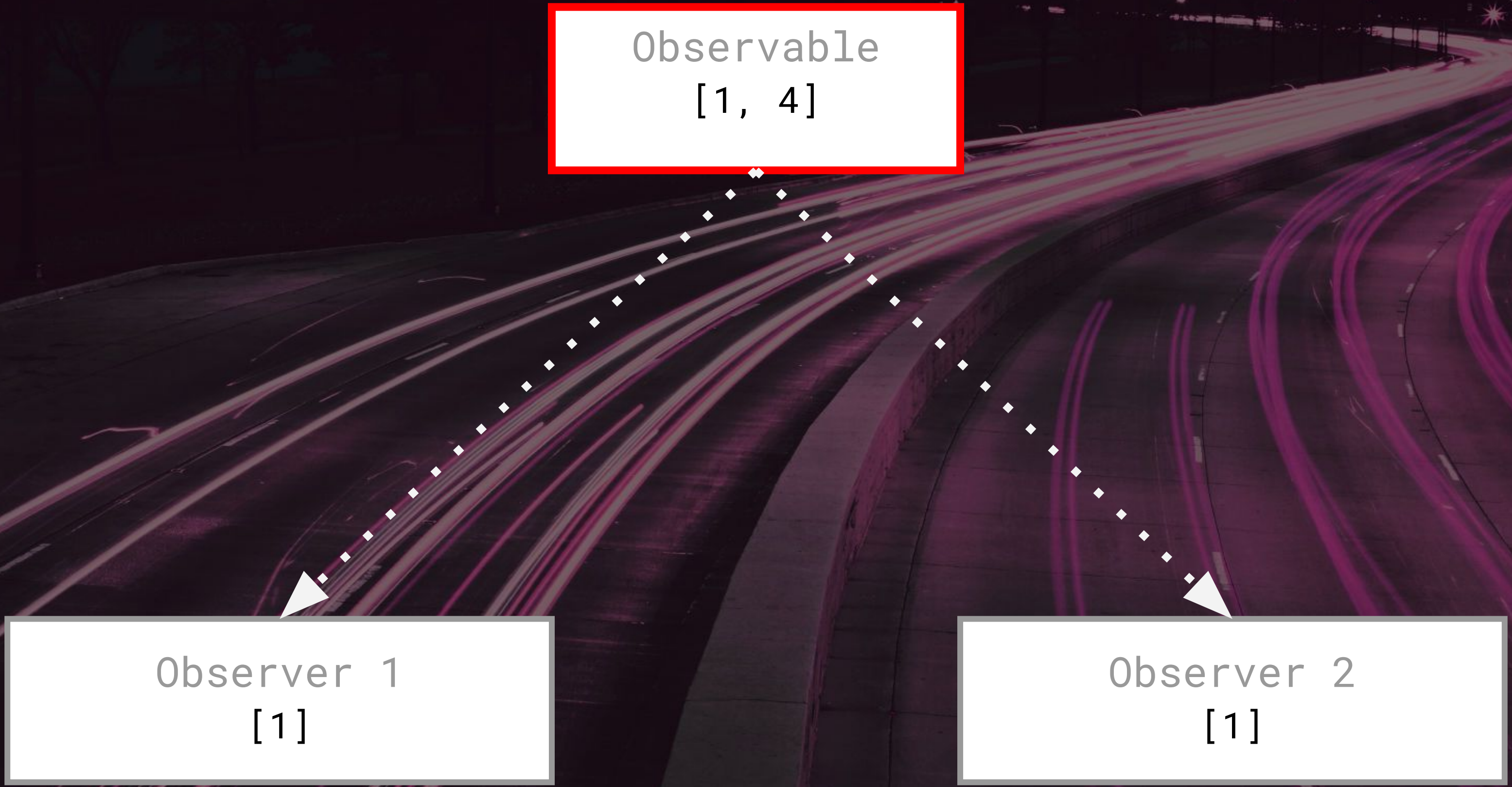
with observable streams



Observable
[1]

Observer 1
[1]

Observer 2
[1]



Observable

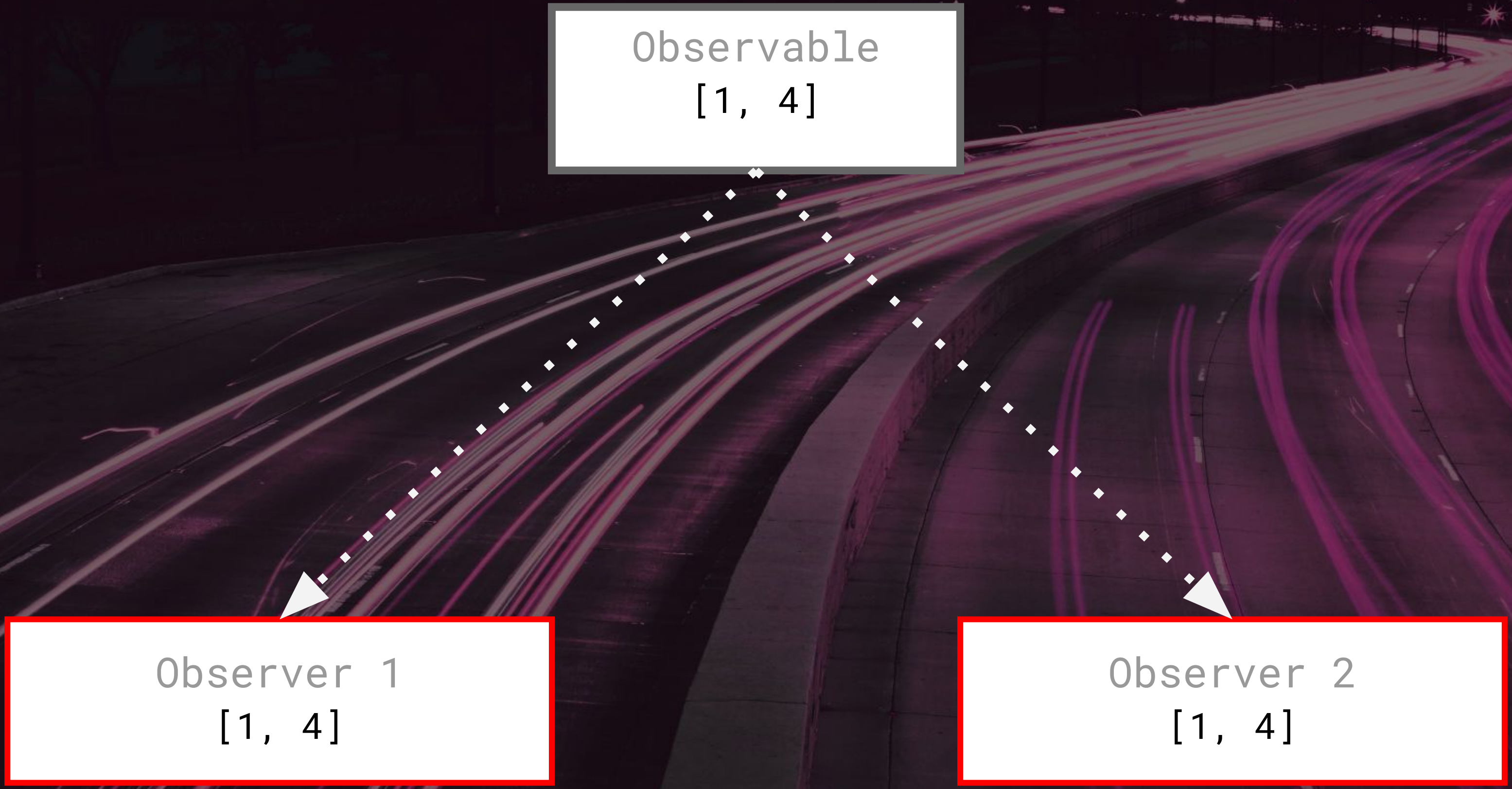
[1, 4]

Observer 1

[1]

Observer 2

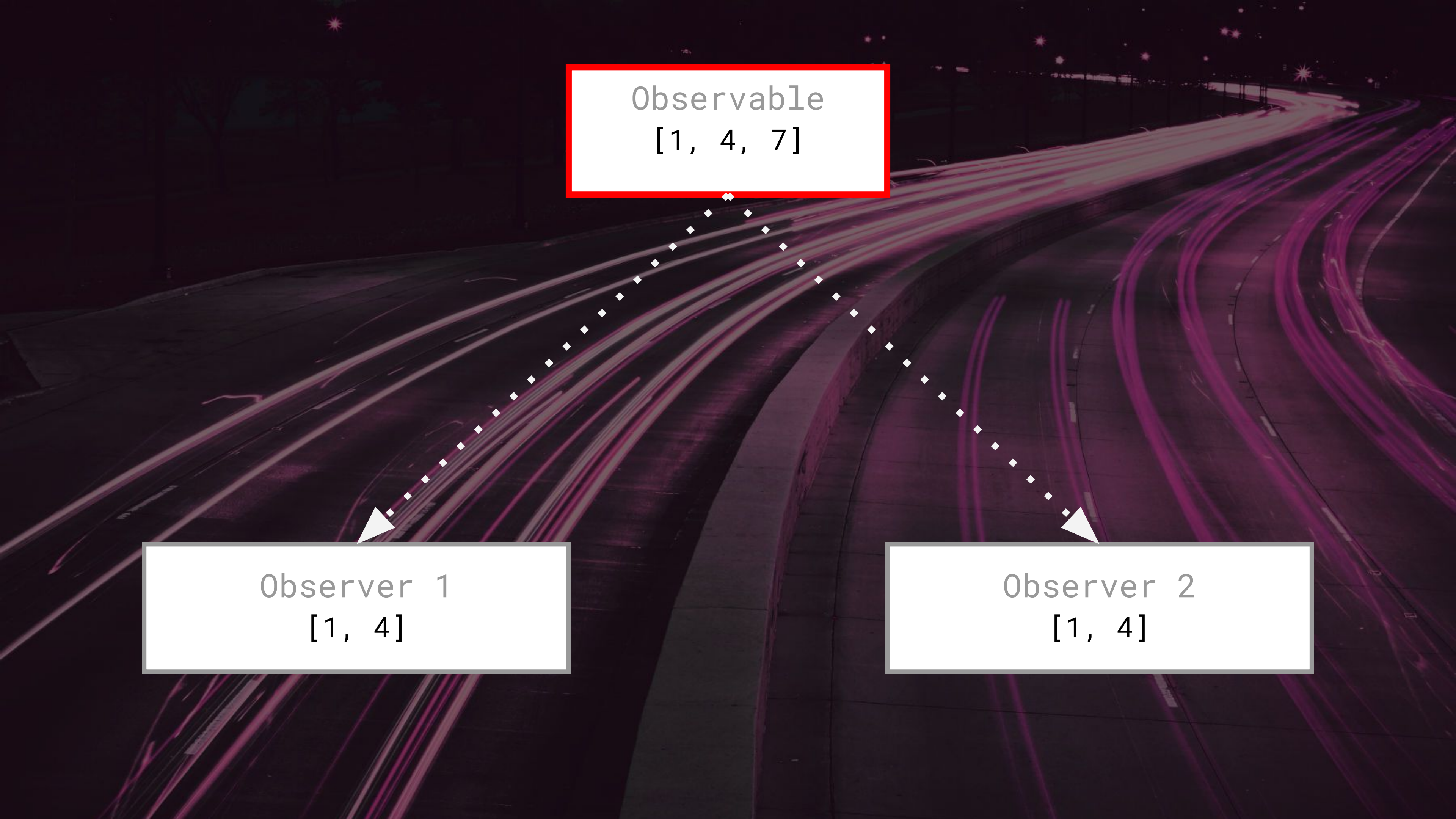
[1]



Observable
[1, 4]

Observer 1
[1, 4]

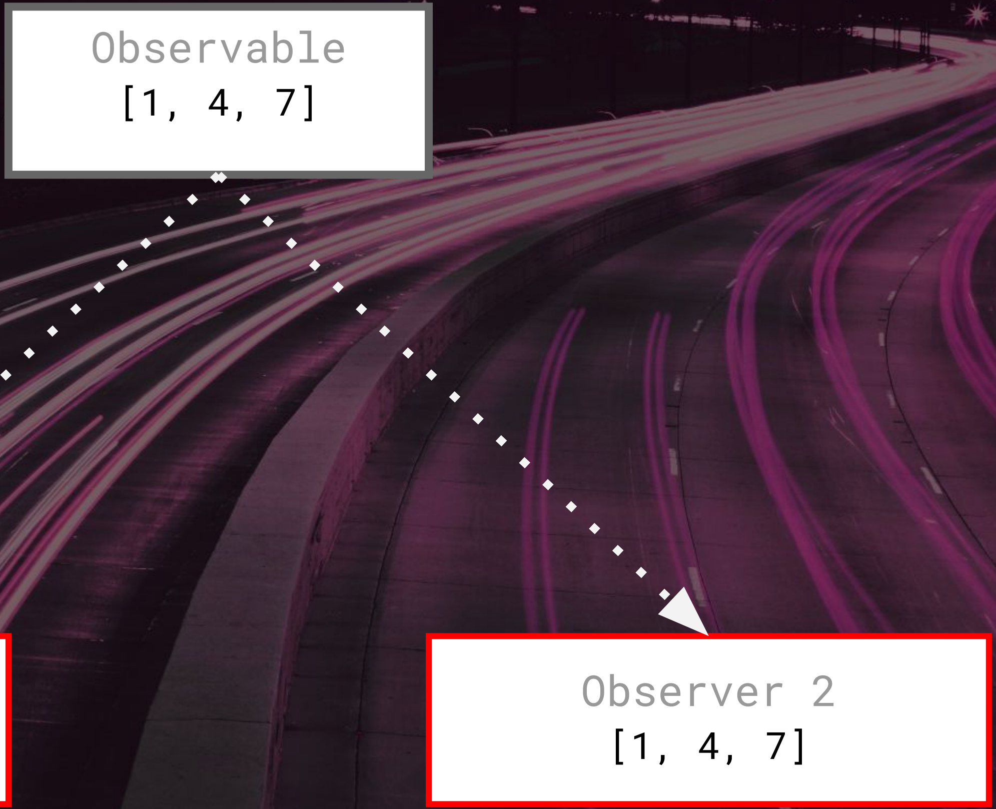
Observer 2
[1, 4]



Observable
[1, 4, 7]

Observer 1
[1, 4]

Observer 2
[1, 4]



Observable

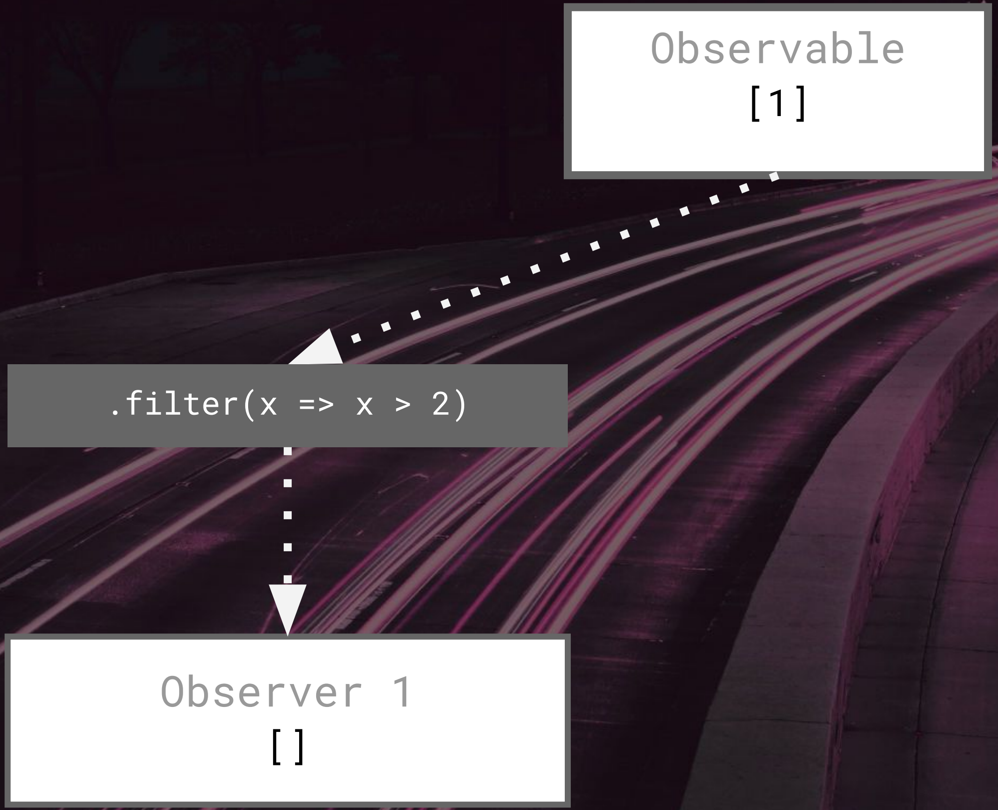
[1, 4, 7]

Observer 1

[1, 4, 7]

Observer 2

[1, 4, 7]



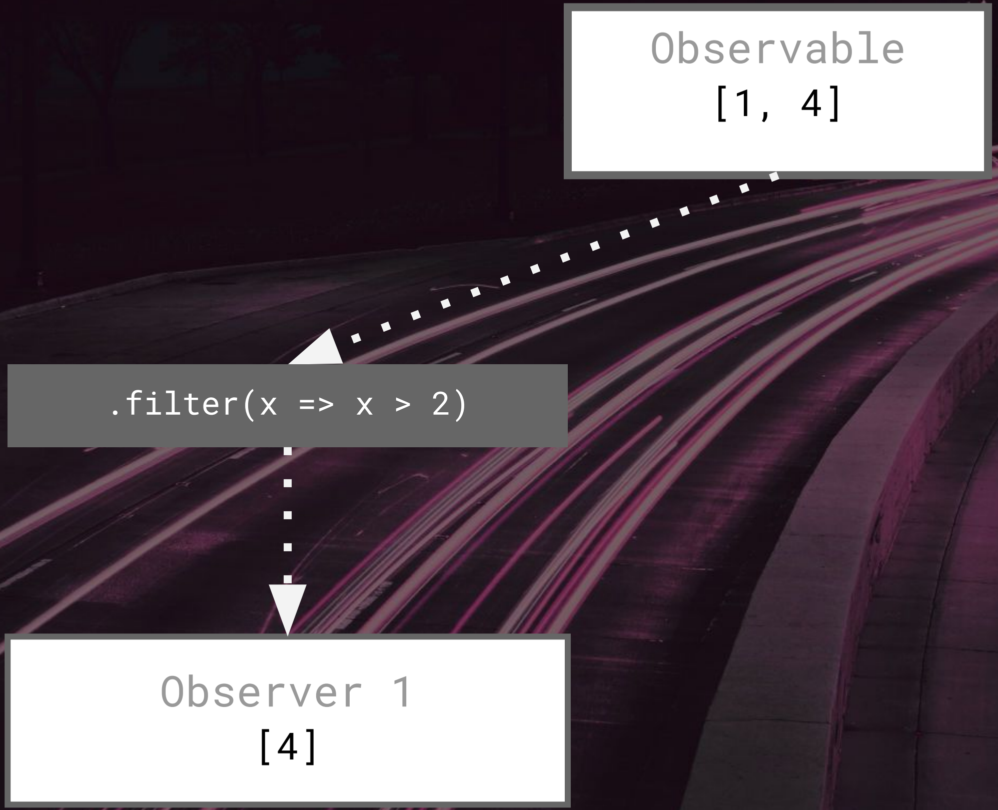
Observable

[1]

.filter(x => x > 2)

Observer 1

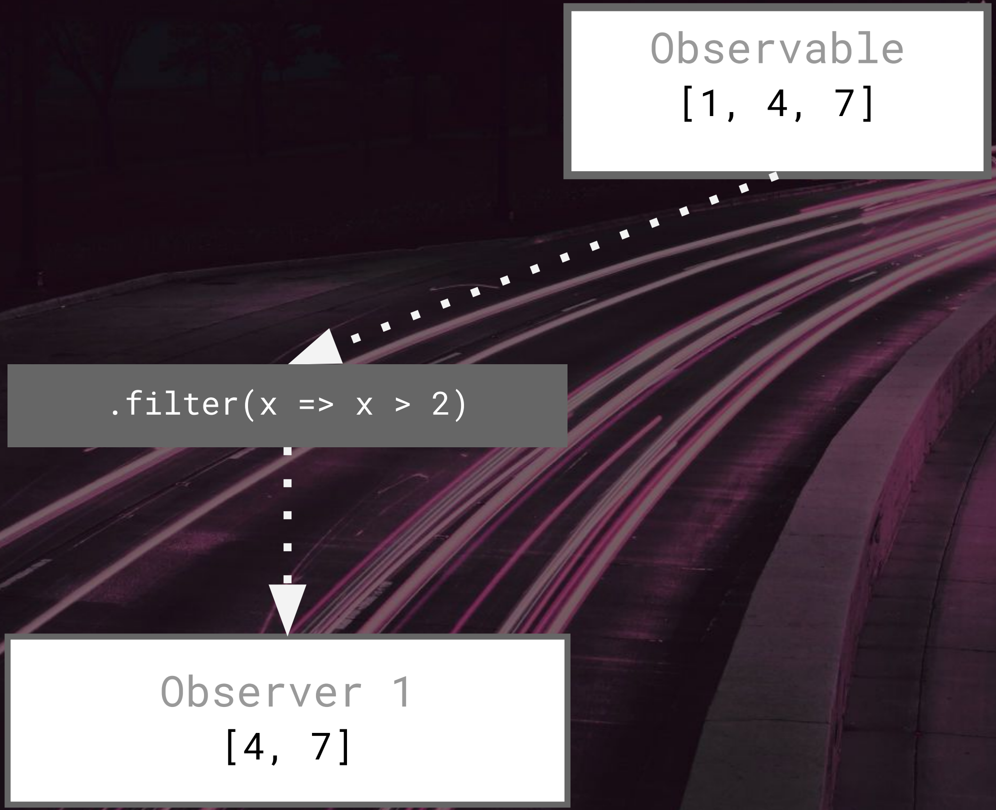
[]



Observable
[1, 4]

```
.filter(x => x > 2)
```

Observer 1
[4]



Observable

[1, 4, 7]

.filter(x => x > 2)

Observer 1

[4, 7]

Combination

combineAll

combineLatest

concat

concatAll

exhaust

forkJoin

merge

mergeAll

race

startWith

switch

withLatestFrom

zip

Creation

ajax

bindCallback

bindNodeCallback

create

defer

empty

from

fromEvent

fromEventPattern

fromPromise

generate

interval

never

Error Handling

catch

retry

retryWhen

Utility

do

delay

delayWhen

dematerialize

finally

let

materialize

observeOn

Conditional and Boolean

defaultIfEmpty

every

find

findIndex

isEmpty

Mathematical and Aggregate

count

max

min

reduce

Transformation

buffer

bufferCount

bufferTime

bufferToggle

bufferWhen

concatMap

concatMapTo

exhaustMap

expand

groupBy

map

mapTo

mergeMap

Filtering

debounce

debounceTime

distinct

distinctKey

distinctUntilChanged

distinctUntilKeyChanged

elementAt

filter

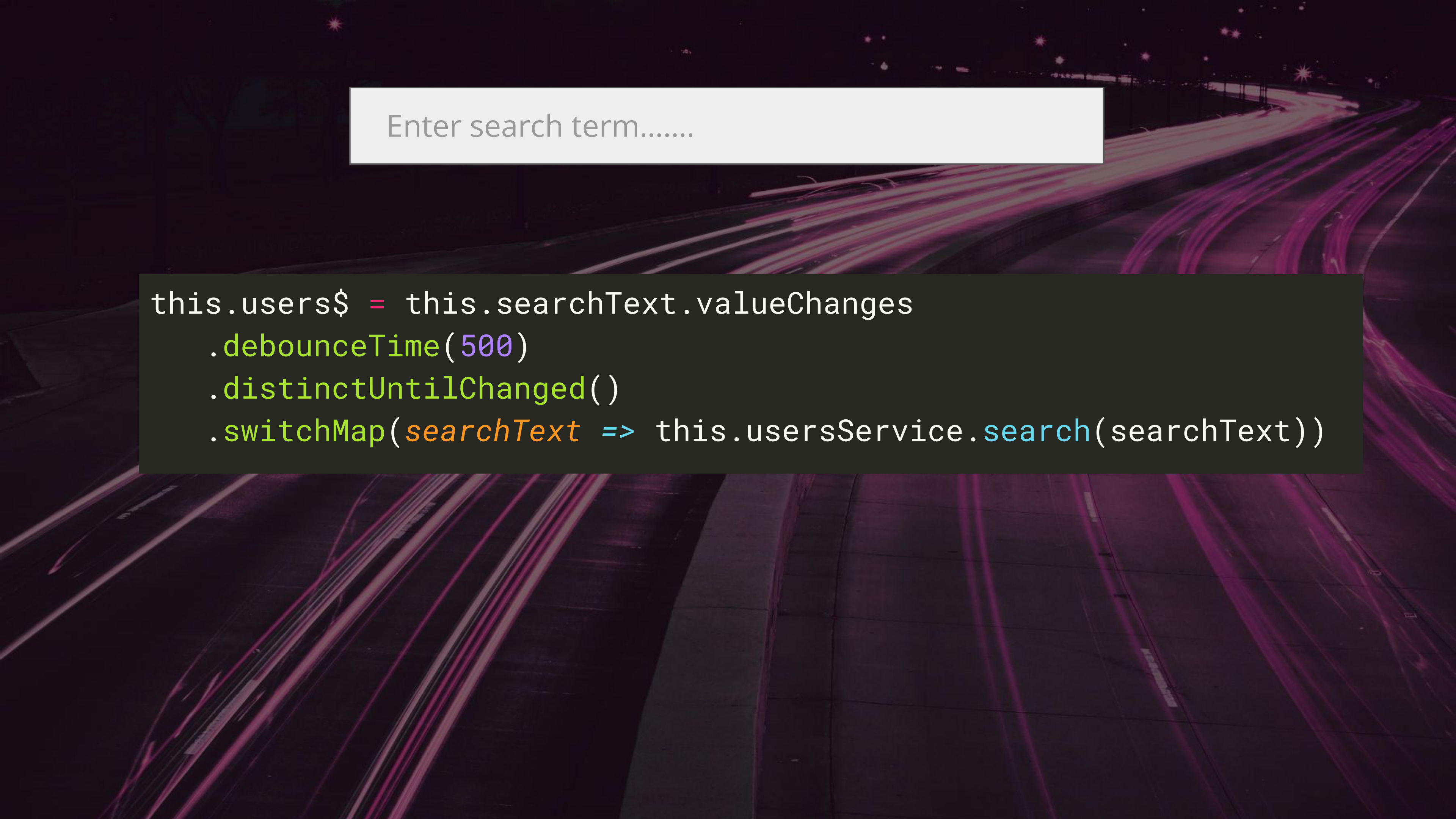
first

ignoreElements

audit

auditTime

last



Enter search term.......

```
this.users$ = this.searchText.valueChanges
    .debounceTime(500)
    .distinctUntilChanged()
    .switchMap(searchText => this.usersService.search(searchText))
```

RxJS is a journey…

```
@Injectable()
export class PeopleService {
  constructor(private httpClient: HttpClient) { }

   getPeople() {
     return this.httpClient.get(`http://swapi.co/api/people/`)
      .subscribe(people => console.log(people));
  }
}
```

```
@Component()
export class SearchComponent implements OnInit {
  searchControl = new FormControl();
  ngOnInit() {
    this.searchControl.valueChanges.subscribe(value => {
      // do something with value here
    });
  }
}
```

```
ngOnInit() {
   this.route.params
    .subscribe(params => this.id = params['id'])
  }
```

```
ngOnInit() {
   this.route.params.pipe(
     map(params => params['id']),
     switchMap(id => this.contactsService.getContact(id)),
    )
    .subscribe(contact => this.contact = contact);
  }
```

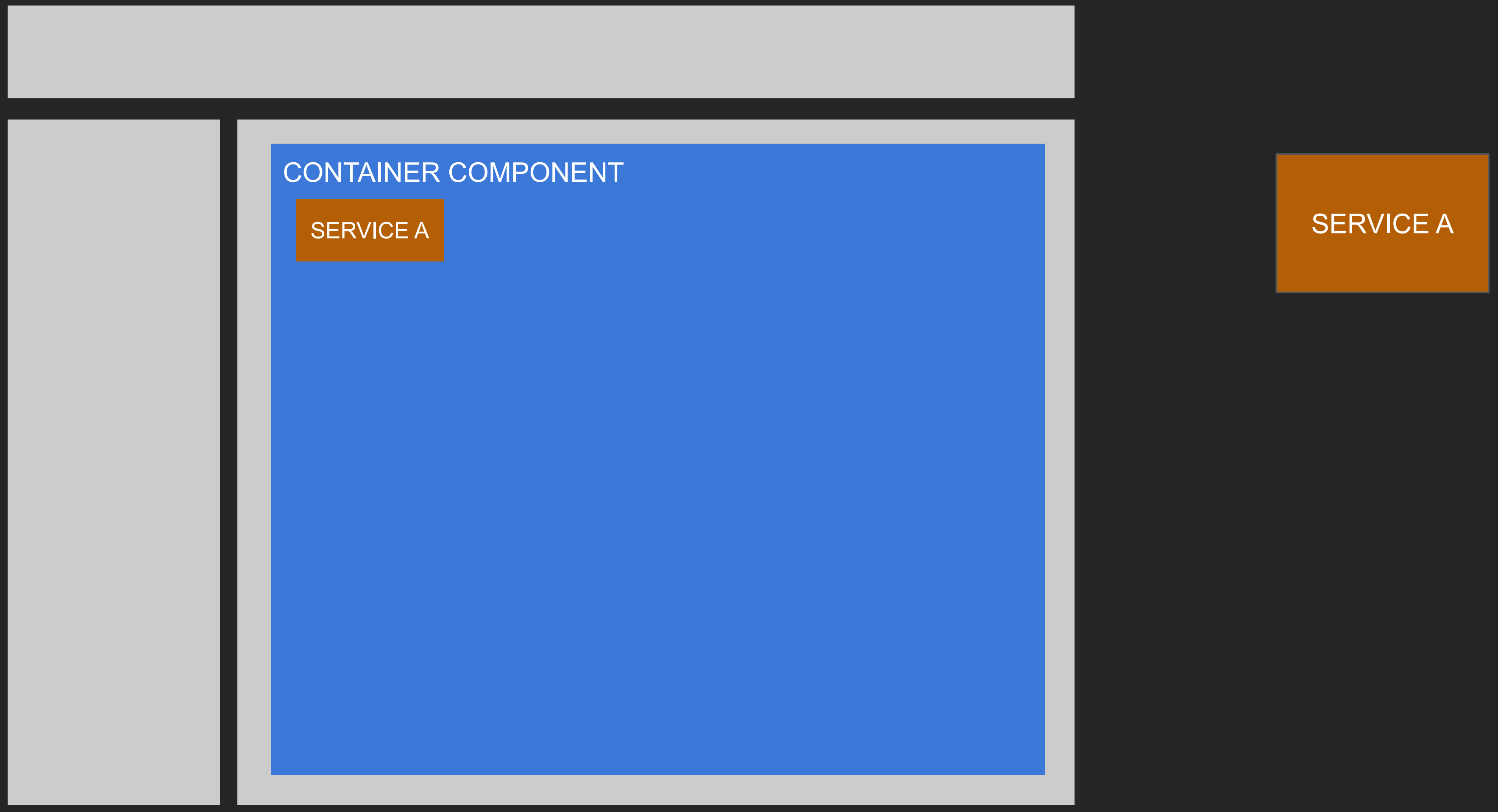
Summary - RxJS

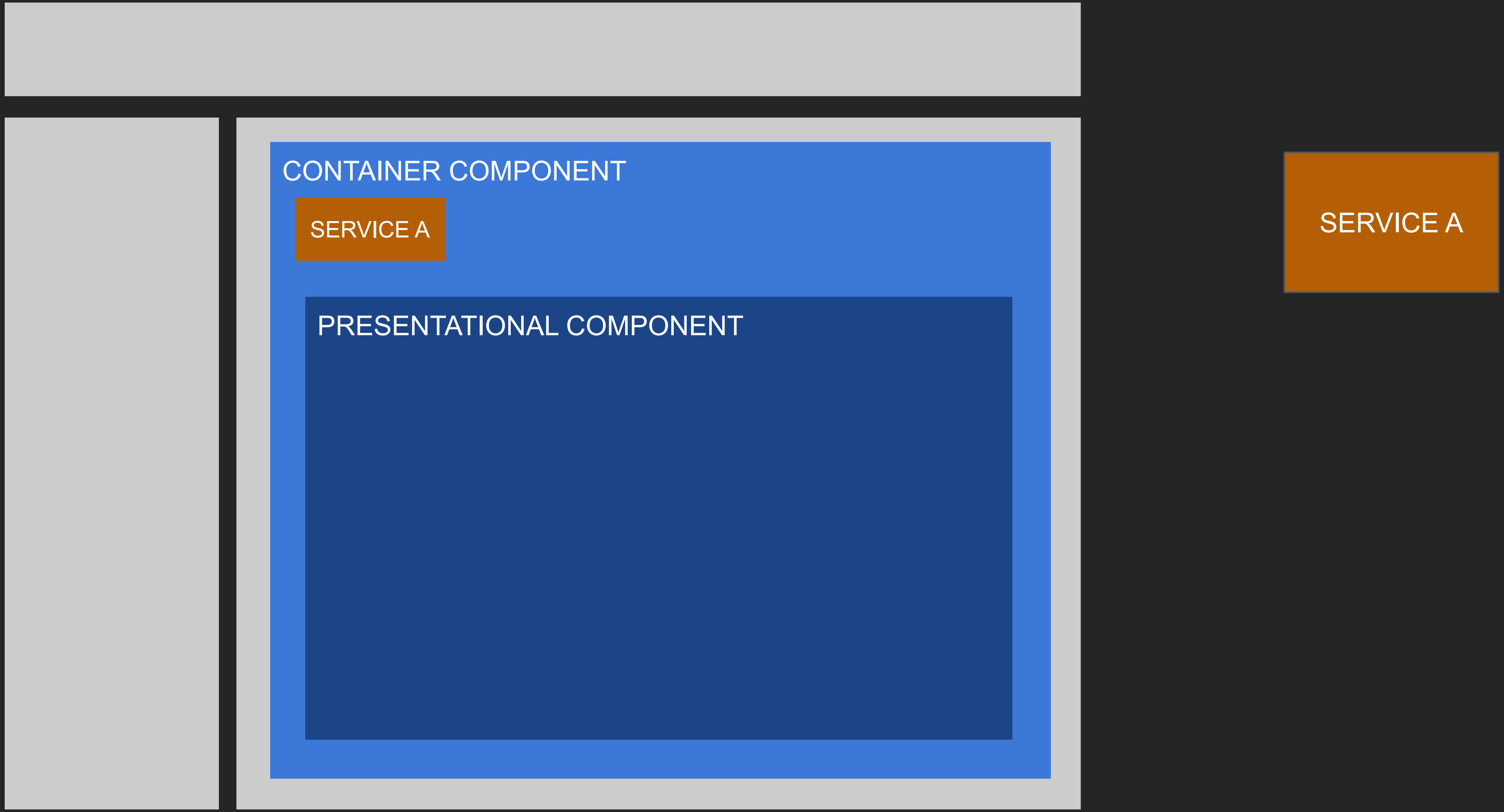
Modern web development is asynchronous

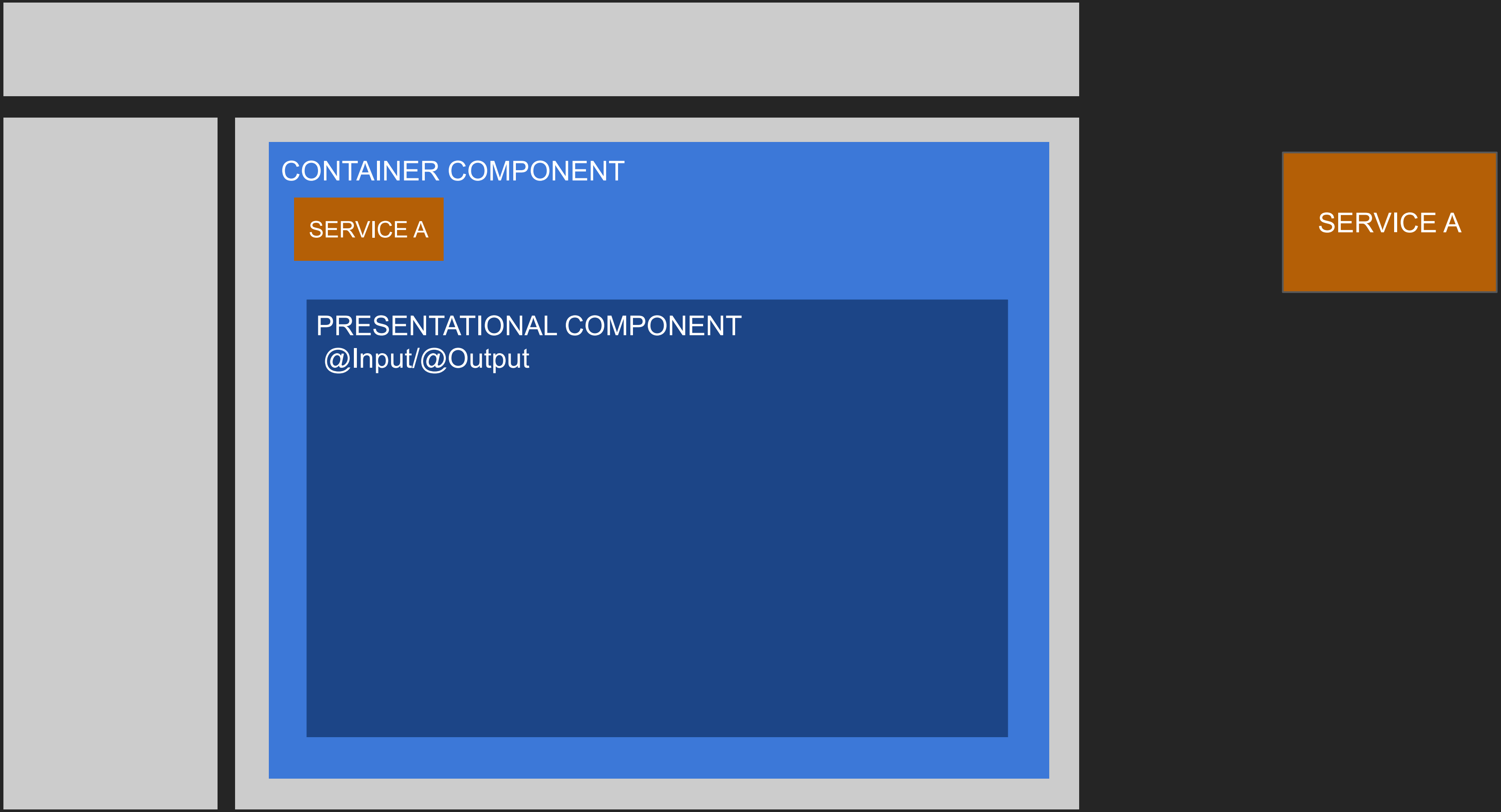
RxJS takes some effort to fully understand, but it simplifies writing async code

RxJS is baked into Angular

NgRx


CONTAINER COMPONENT
SERVICE A
SERVICE A


CONTAINER COMPONENT
SERVICE A
PRESENTATIONAL COMPONENT
SERVICE A

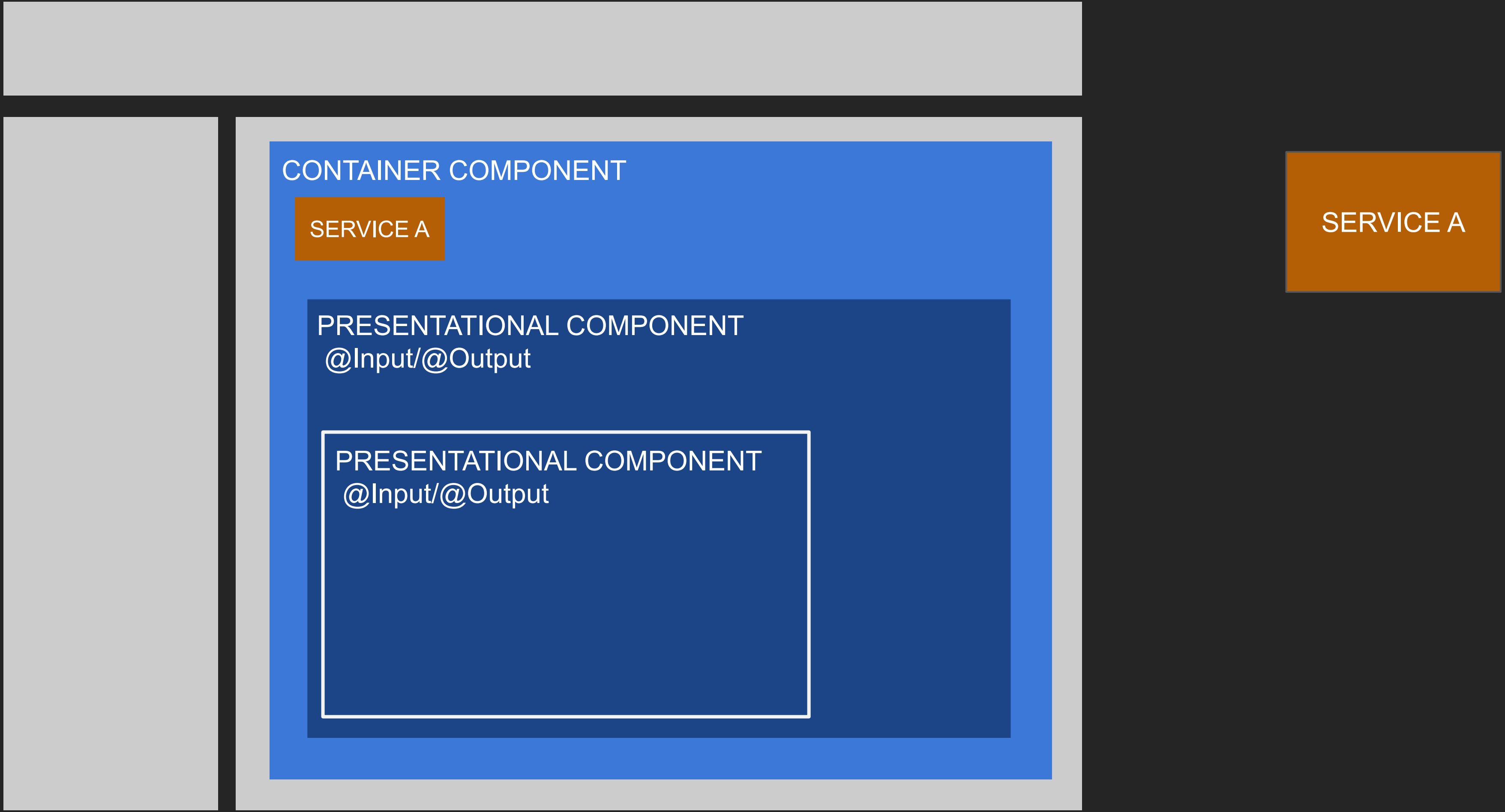


CONTAINER COMPONENT

SERVICE A

PRESENTATIONAL COMPONENT
@Input/@Output

SERVICE A



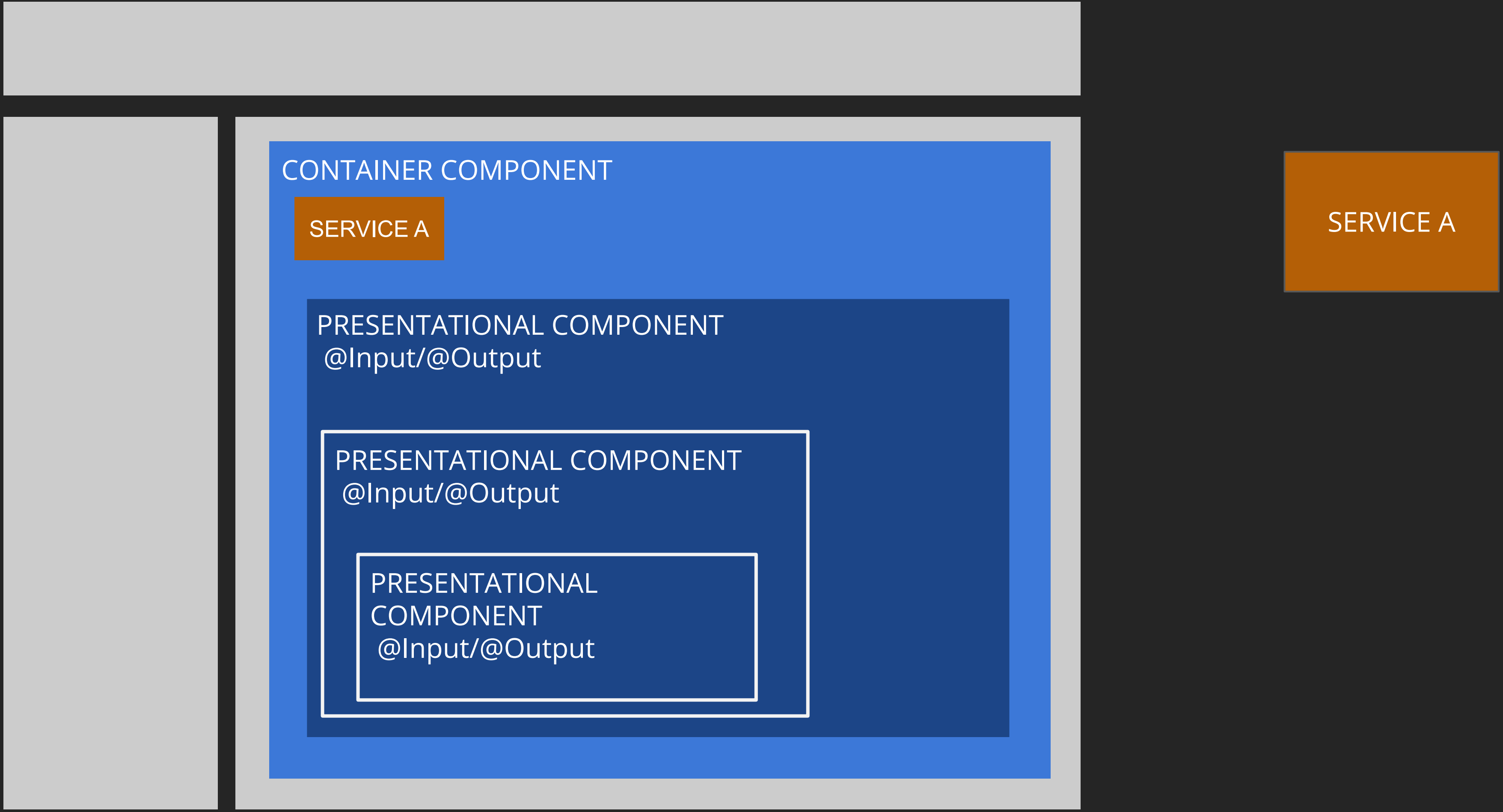
CONTAINER COMPONENT

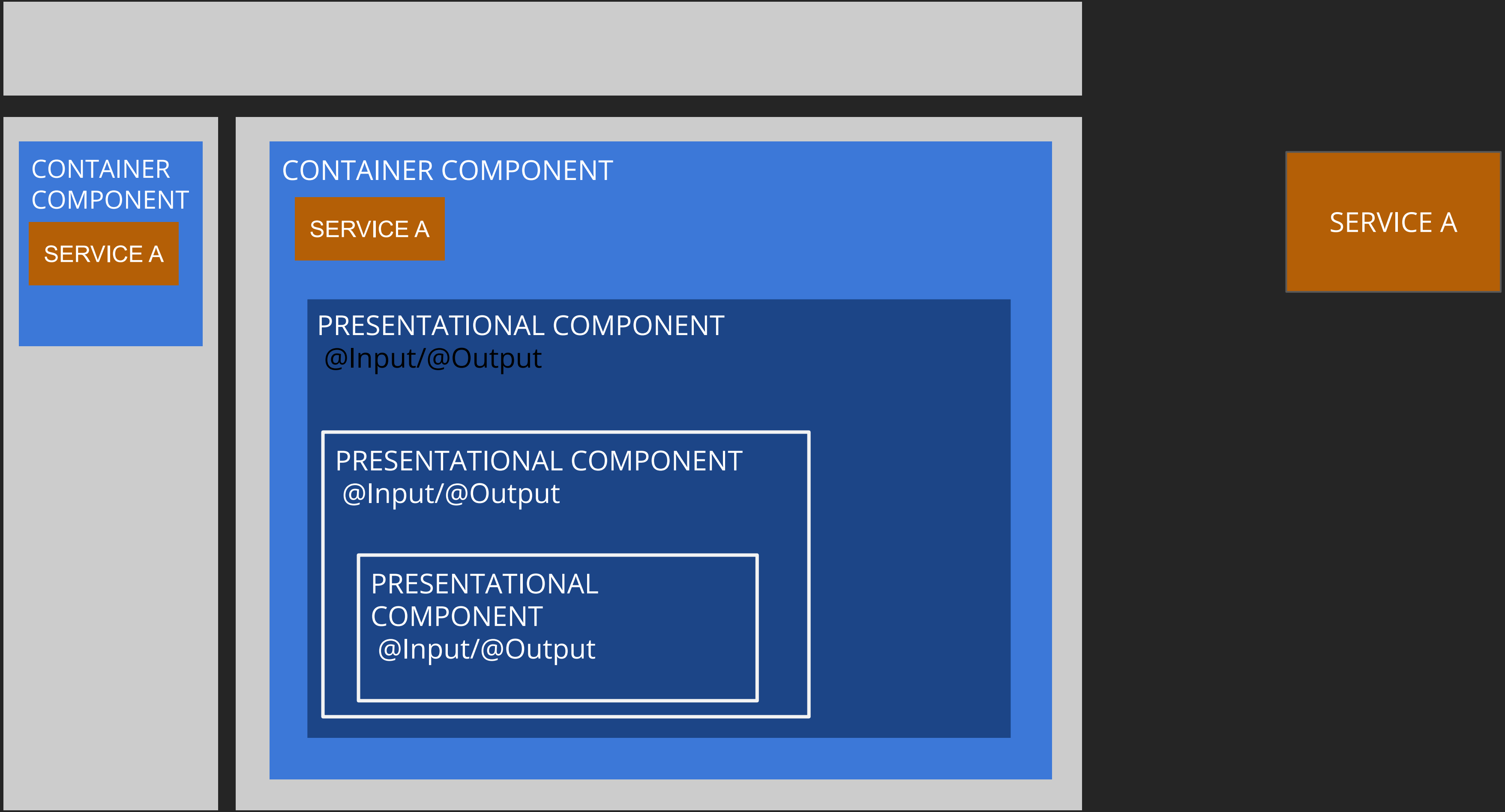
SERVICE A

PRESENTATIONAL COMPONENT
@Input/@Output

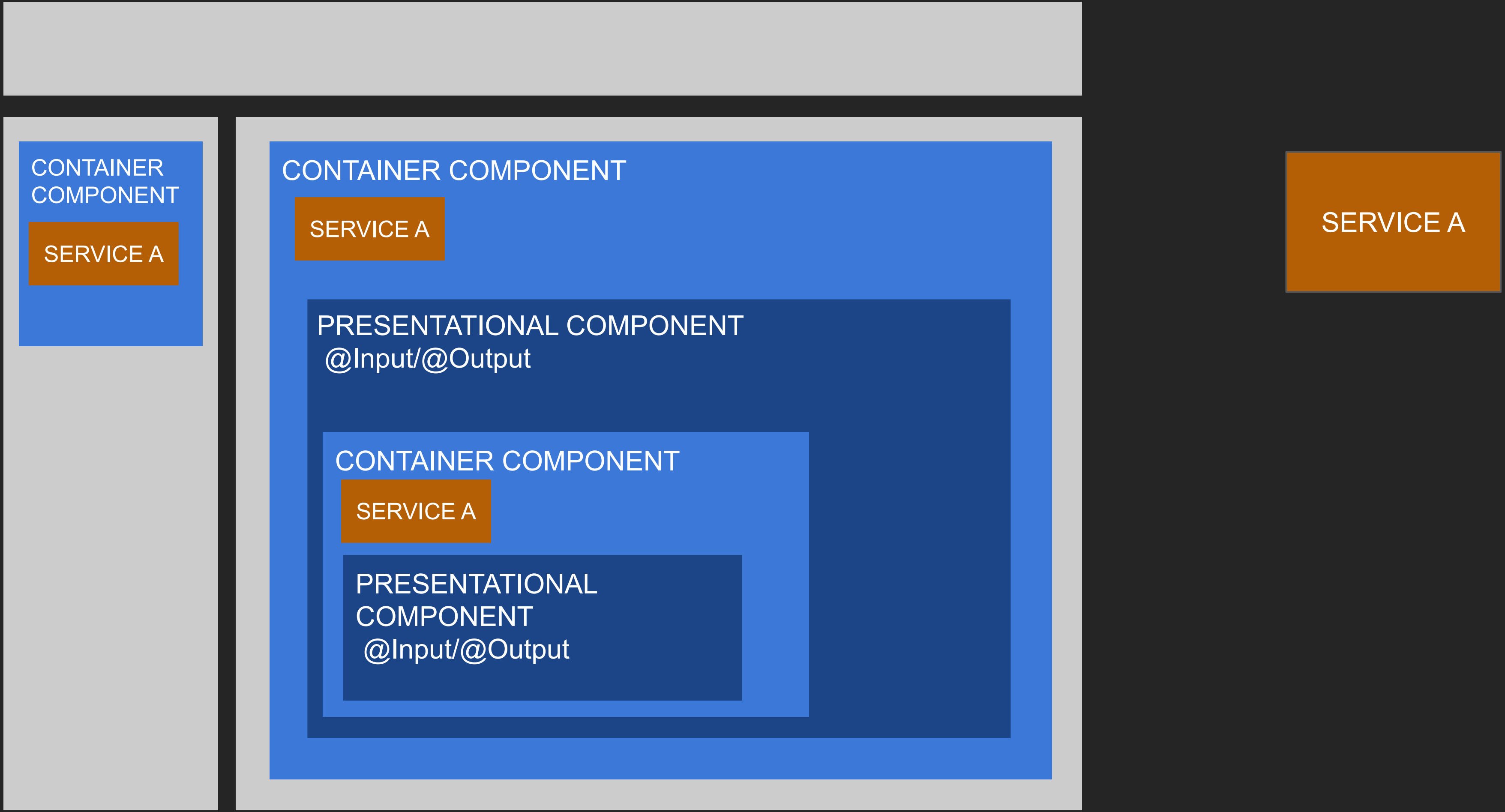
PRESENTATIONAL COMPONENT
@Input/@Output

SERVICE A


CONTAINER COMPONENT
SERVICE A
PRESENTATIONAL COMPONENT
@Input/@Output
PRESENTATIONAL COMPONENT
@Input/@Output
PRESENTATIONAL COMPONENT
@Input/@Output
SERVICE A



CONTAINER COMPONENT
SERVICE A
CONTAINER COMPONENT
SERVICE A
PRESENTATIONAL COMPONENT
@Input/@Output
PRESENTATIONAL COMPONENT
@Input/@Output
PRESENTATIONAL COMPONENT
@Input/@Output
SERVICE A



CONTAINER COMPONENT

SERVICE A

CONTAINER COMPONENT

SERVICE A

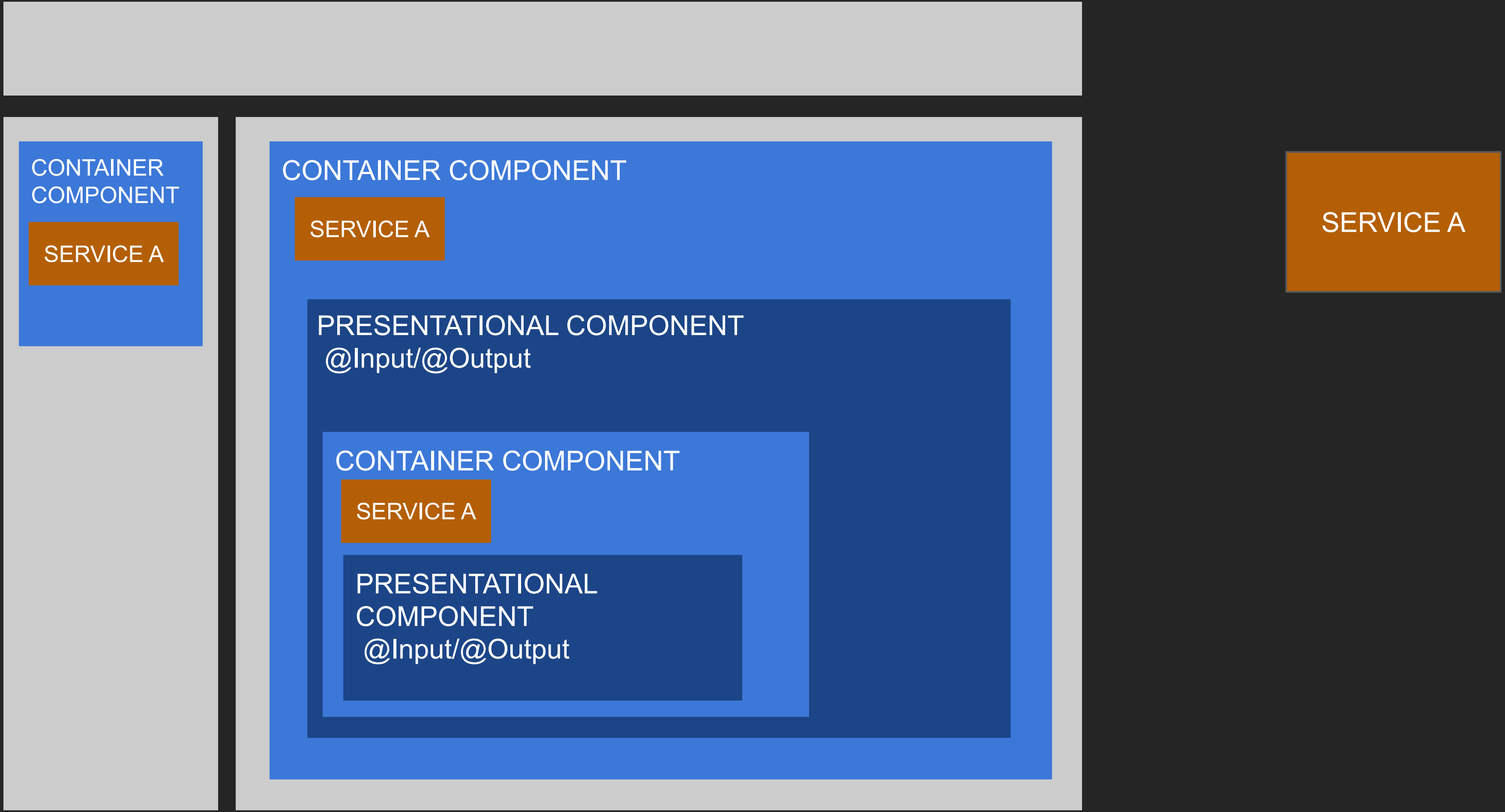
PRESENTATIONAL COMPONENT
@Input/@Output

CONTAINER COMPONENT

SERVICE A

PRESENTATIONAL COMPONENT
@Input/@Output

SERVICE A



CONTAINER COMPONENT

SERVICE A

CONTAINER COMPONENT

SERVICE A

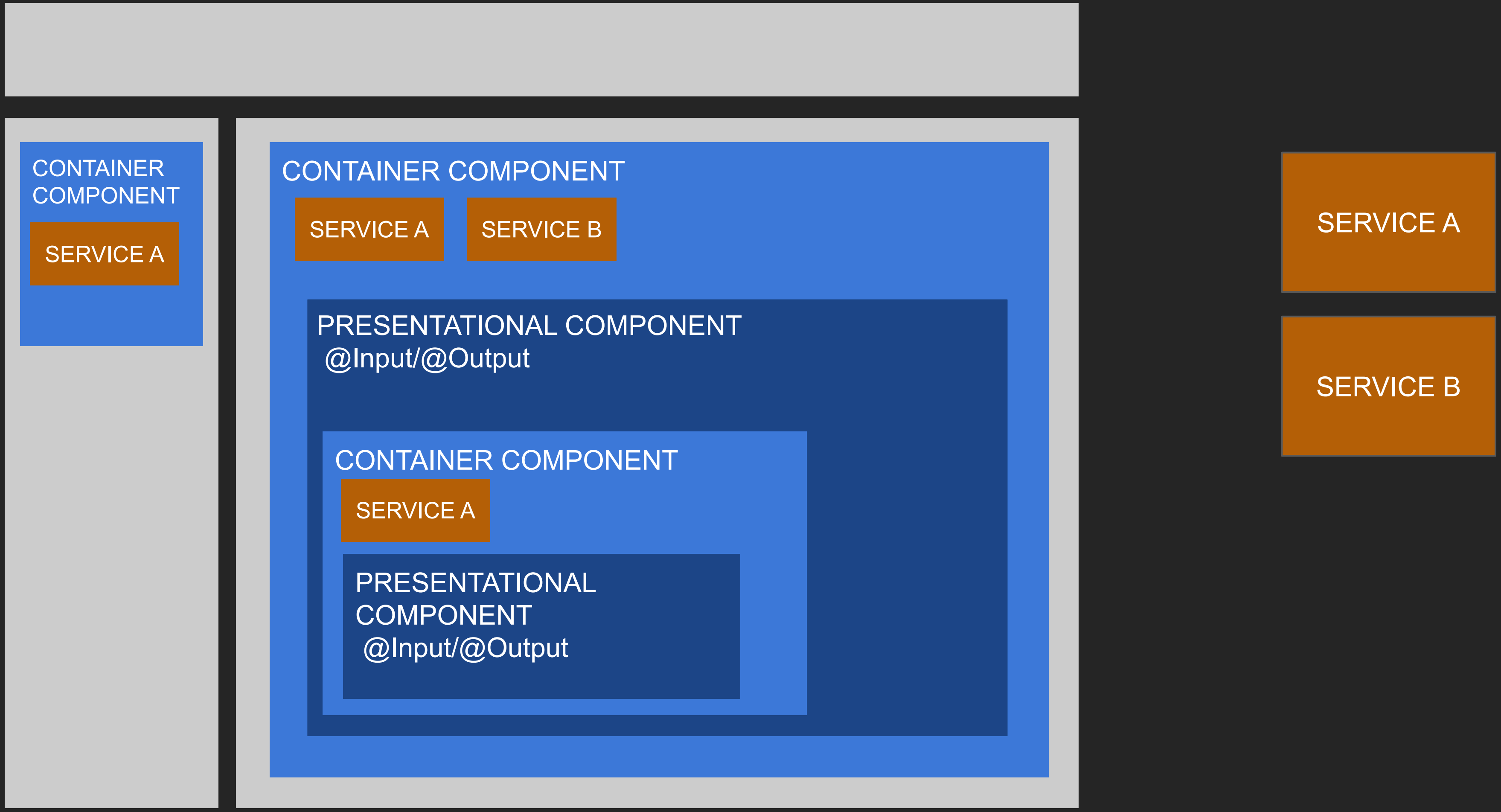
PRESENTATIONAL COMPONENT
@Input/@Output

CONTAINER COMPONENT

SERVICE A

PRESENTATIONAL COMPONENT
@Input/@Output

SERVICE A



CONTAINER COMPONENT

SERVICE A

CONTAINER COMPONENT

SERVICE A

SERVICE B

PRESENTATIONAL COMPONENT
@Input/@Output

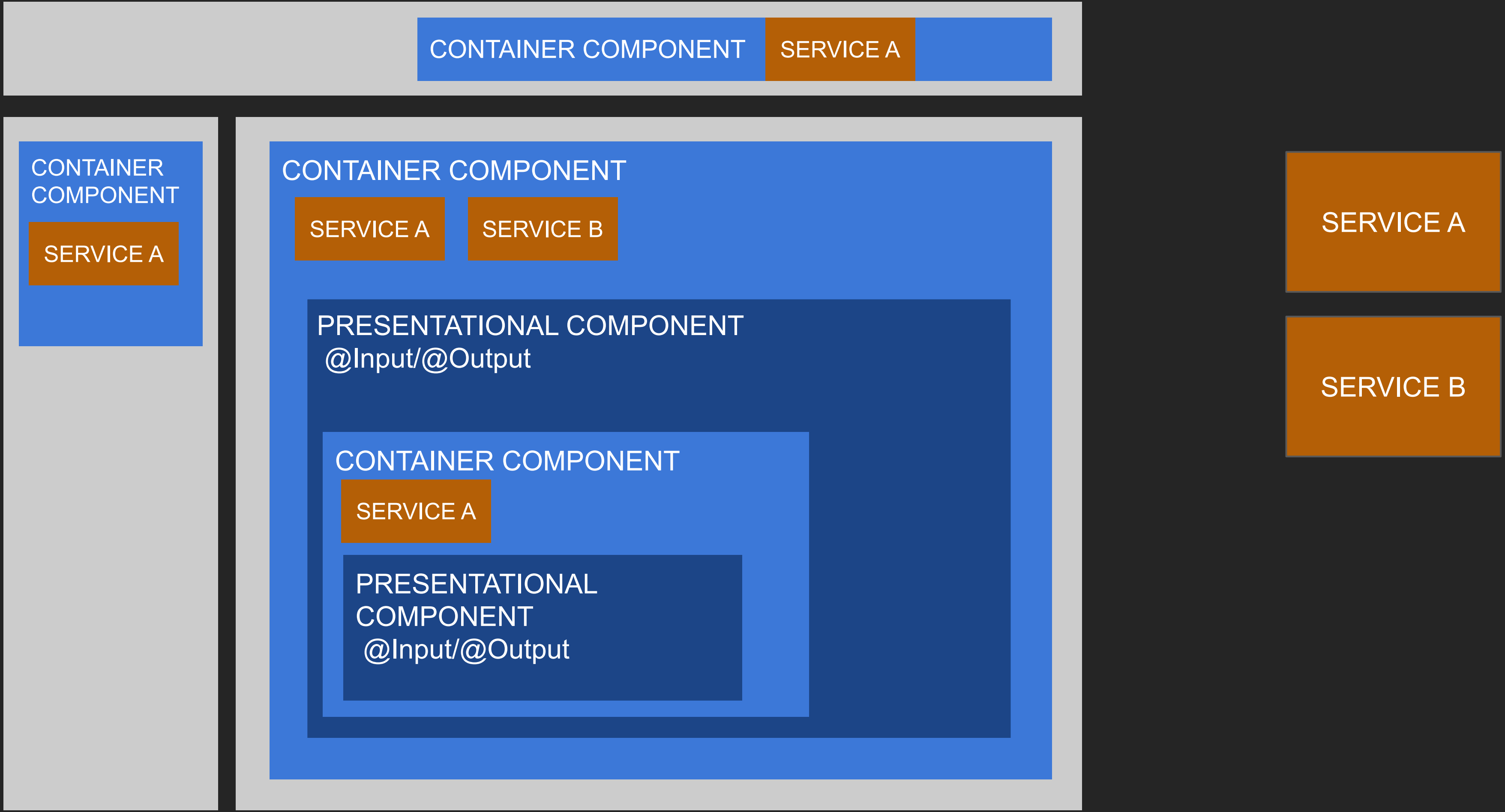
CONTAINER COMPONENT

SERVICE A

PRESENTATIONAL COMPONENT
@Input/@Output

SERVICE A

SERVICE B



CONTAINER COMPONENT

SERVICE A

CONTAINER COMPONENT

SERVICE A

CONTAINER COMPONENT

SERVICE A

SERVICE B

PRESENTATIONAL COMPONENT
@Input/@Output

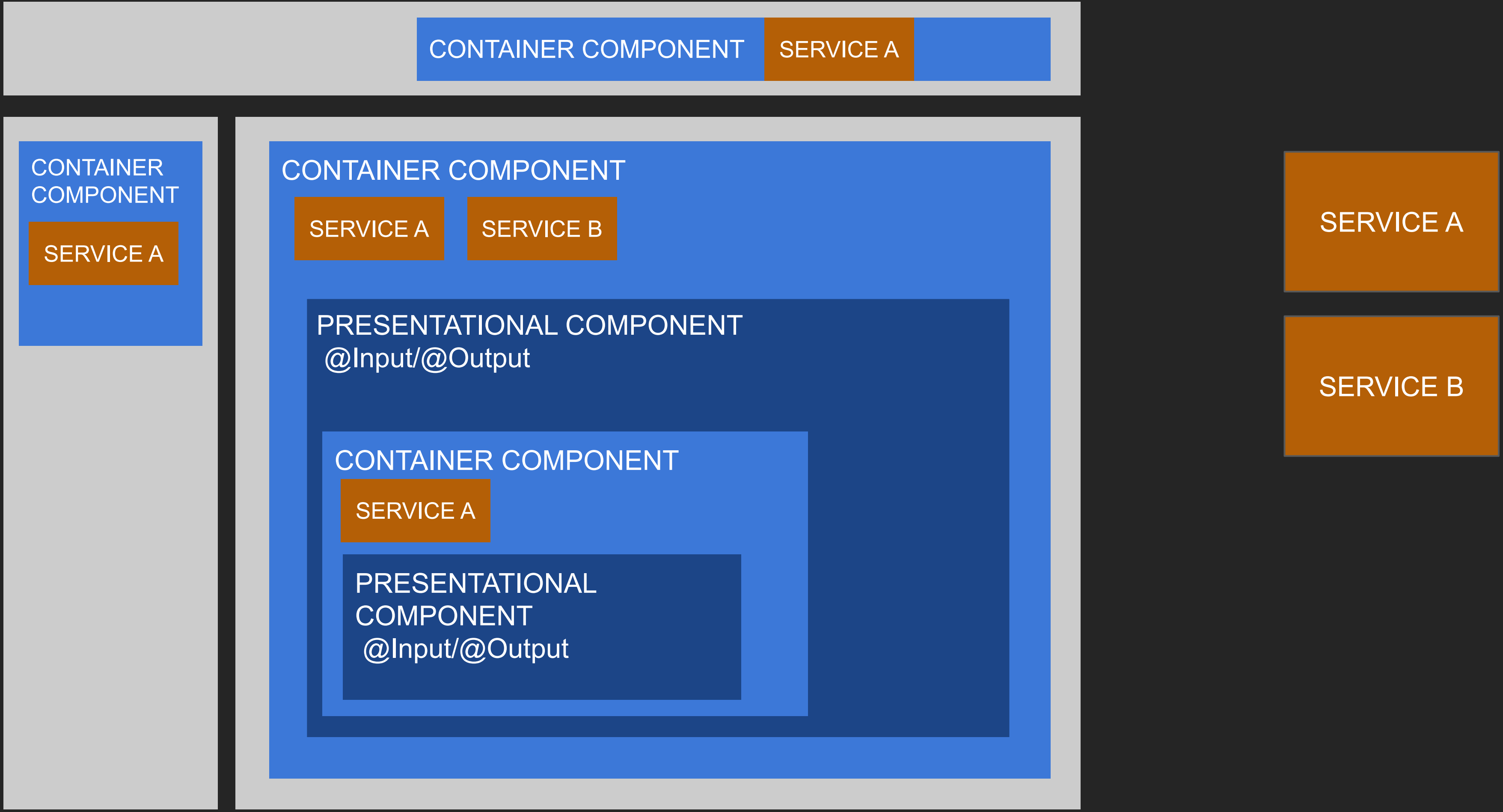
CONTAINER COMPONENT

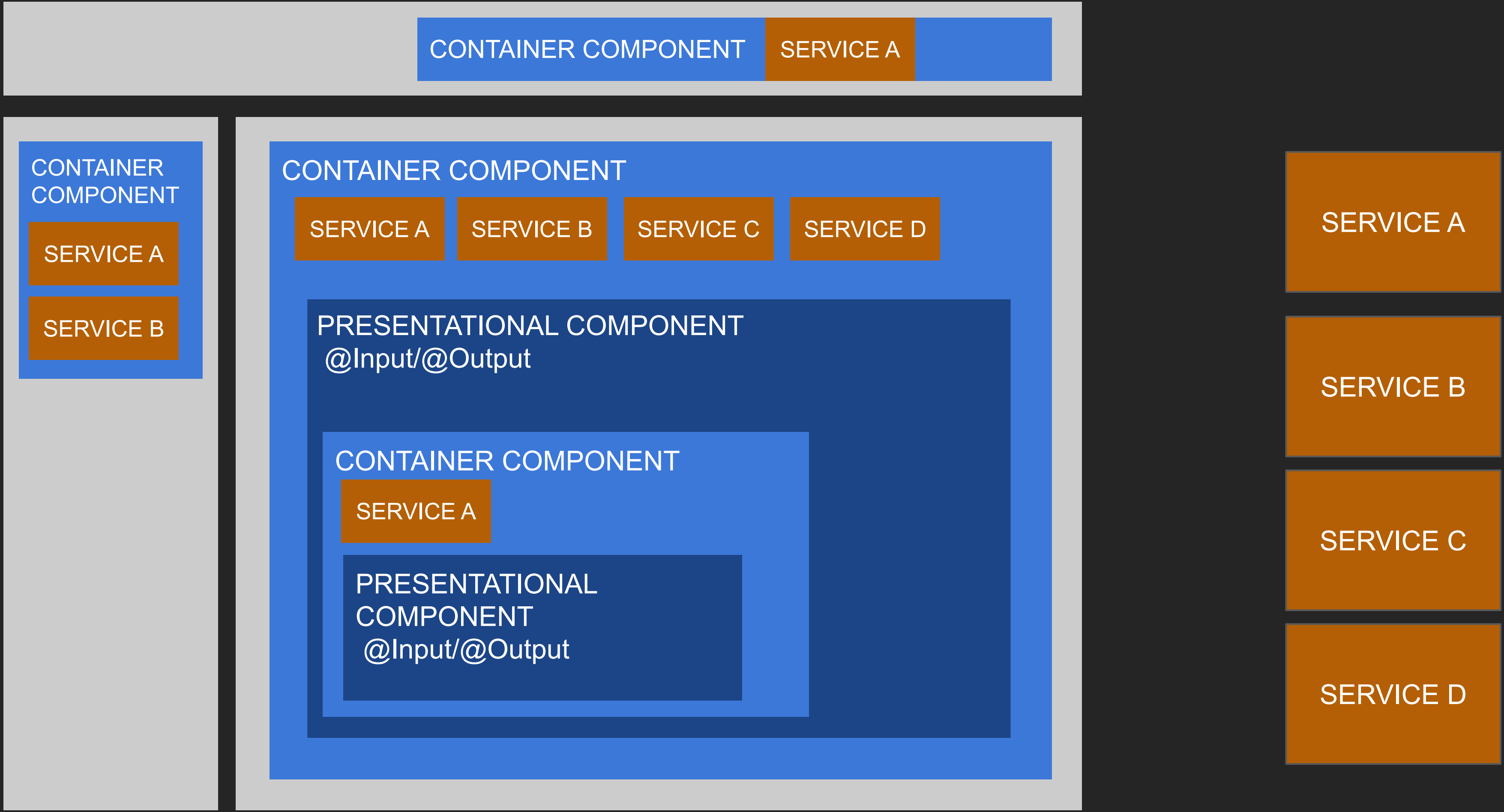
SERVICE A

PRESENTATIONAL COMPONENT
@Input/@Output

SERVICE A

SERVICE B


CONTAINER COMPONENT
SERVICE A
CONTAINER COMPONENT
SERVICE A
CONTAINER COMPONENT
SERVICE A
SERVICE B
PRESENTATIONAL COMPONENT
@Input/@Output
CONTAINER COMPONENT
SERVICE A
PRESENTATIONAL COMPONENT
@Input/@Output
SERVICE A
SERVICE B



CONTAINER COMPONENT

SERVICE A

CONTAINER COMPONENT

SERVICE A

SERVICE B

CONTAINER COMPONENT

SERVICE A

SERVICE B

SERVICE C

SERVICE D

PRESENTATIONAL COMPONENT
@Input/@Output

CONTAINER COMPONENT

SERVICE A

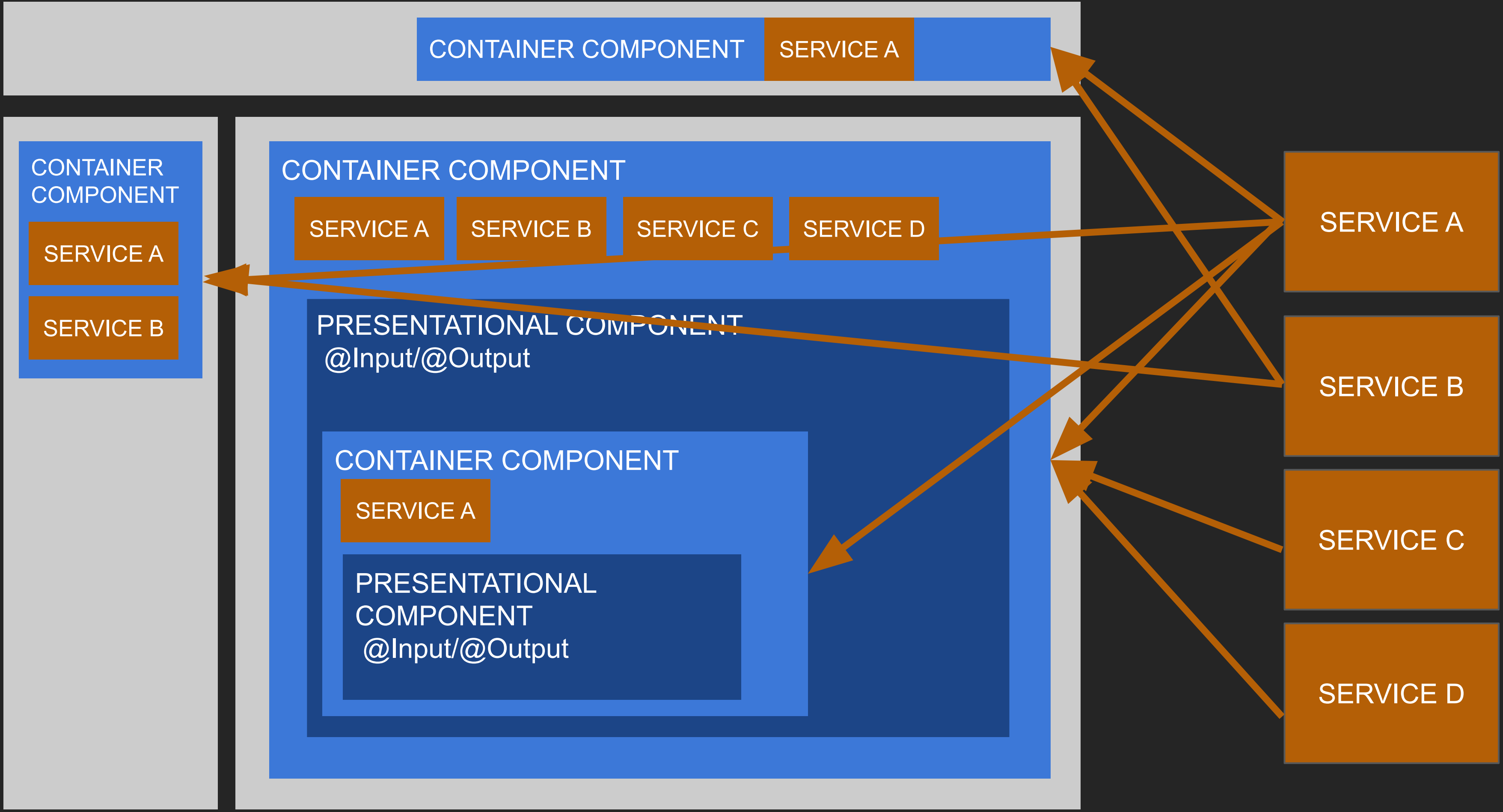
PRESENTATIONAL COMPONENT
@Input/@Output

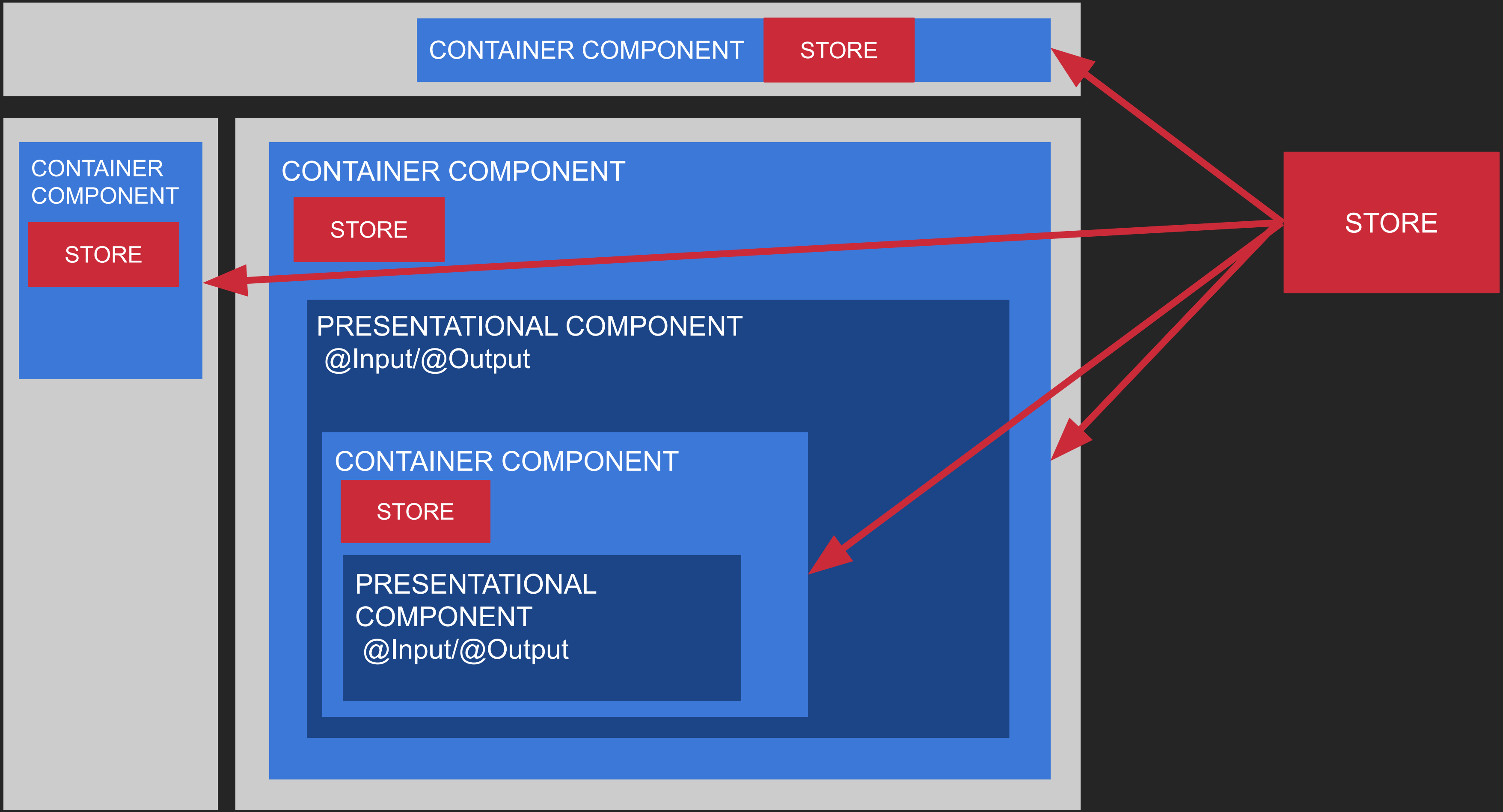
SERVICE A

SERVICE B

SERVICE C

SERVICE D


CONTAINER COMPONENT
SERVICE A
CONTAINER COMPONENT
SERVICE A
SERVICE B
CONTAINER COMPONENT
SERVICE A
SERVICE B
SERVICE C
SERVICE D
PRESENTATIONAL COMPONENT
@Input/@Output
CONTAINER COMPONENT
SERVICE A
PRESENTATIONAL COMPONENT
@Input/@Output
SERVICE A
SERVICE B
SERVICE C
SERVICE D



CONTAINER COMPONENT

STORE

CONTAINER COMPONENT

STORE

CONTAINER COMPONENT

STORE

PRESENTATIONAL COMPONENT
@Input/@Output

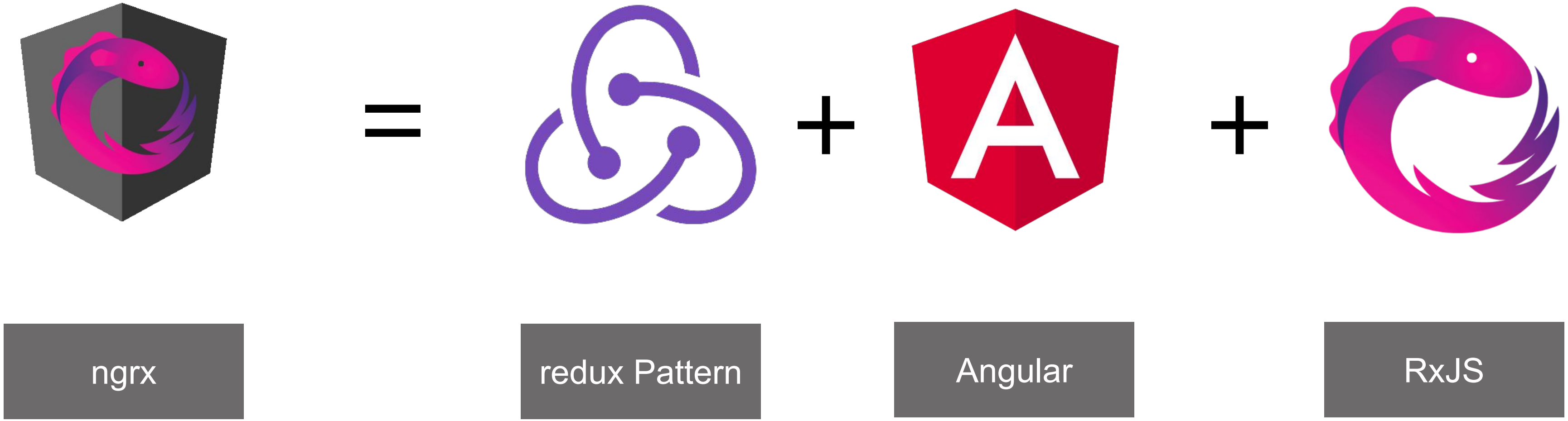
CONTAINER COMPONENT

STORE

PRESENTATIONAL COMPONENT
@Input/@Output

STORE

ngrx is the redux pattern for Angular



ngrx = redux Pattern + Angular + RxJS

action

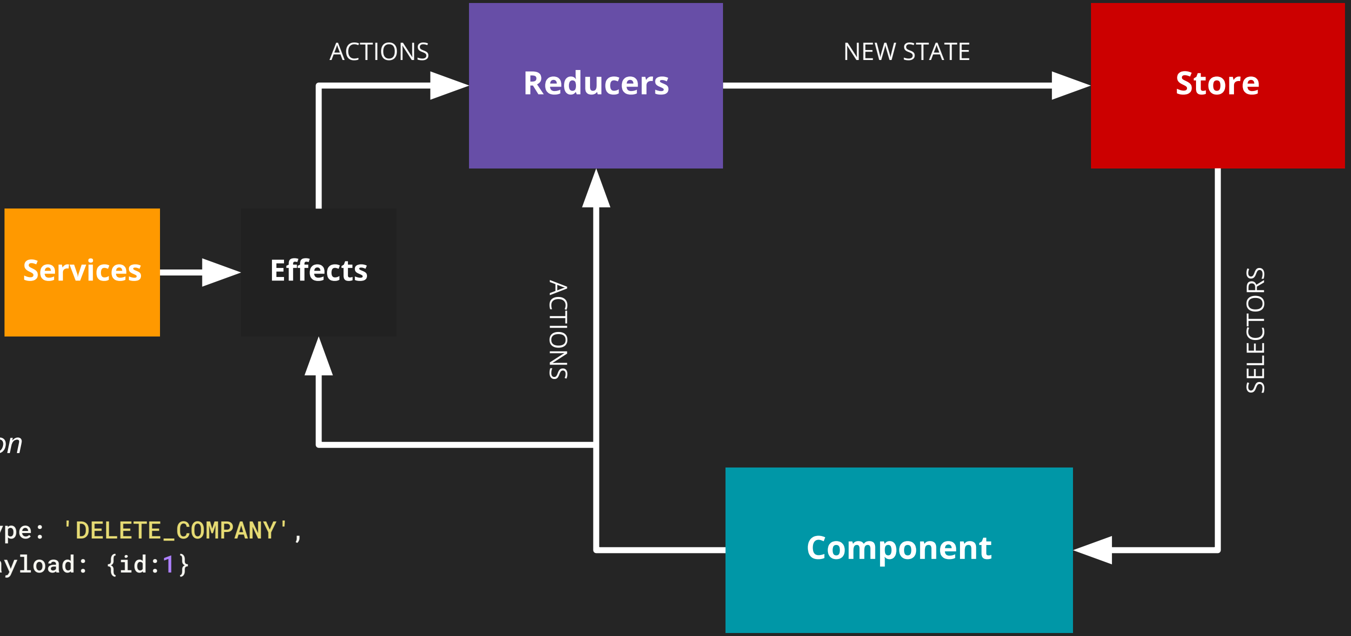
```
{
  type: 'DELETE_COMPANY_SUCCESS',
  payload: {id:1}
}
```

new state

```
companies: [
    {id:2, name:'Microsoft'}
];
```

store state

```
{
  companies: [
   {id:2, name:'Microsoft'}
  ],
  contacts: [
   {id:1, name:Duncan}
  ]
};
```



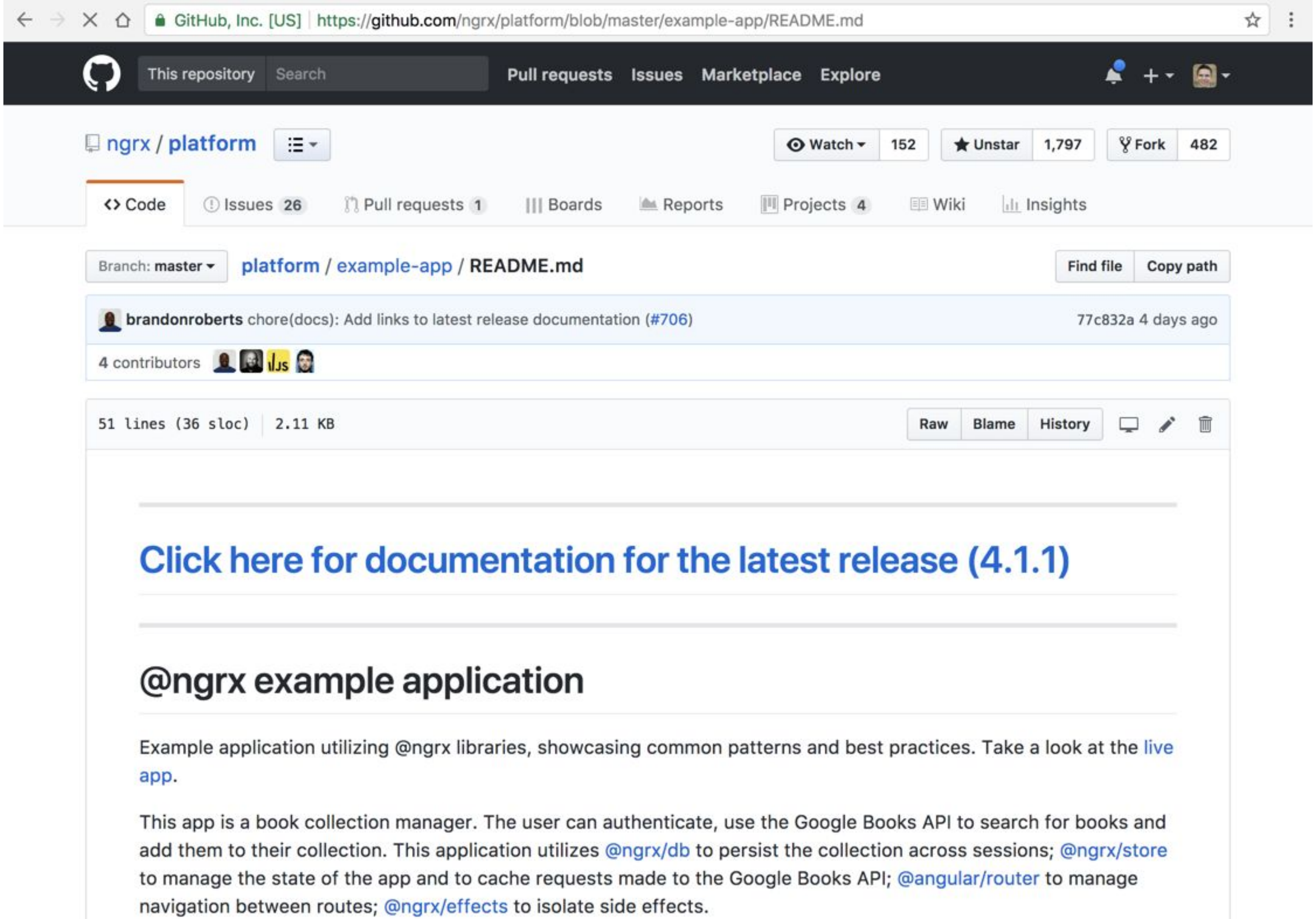
action

```
{
  type: 'DELETE_COMPANY',
  payload: {id:1}
}
```

Play by Play Angular and ngrx

by Lars Klint and Duncan Hunter

In this course, you'll explore using ngrx with Angular to implement the redux pattern.



brandonroberts chore(docs): Add links to latest release documentation (#706) 77c832a 4 days ago

4 contributors

51 lines (36 sloc) 2.11 KB

Click here for documentation for the latest release (4.1.1)

@ngrx example application

Example application utilizing @ngrx libraries, showcasing common patterns and best practices. Take a look at the live app.

This app is a book collection manager. The user can authenticate, use the Google Books API to search for books and add them to their collection. This application utilizes @ngrx/db to persist the collection across sessions; @ngrx/store to manage the state of the app and to cache requests made to the Google Books API; @angular/router to manage navigation between routes; @ngrx/effects to isolate side effects.

ngrx

- ☑ Simplify complex apps
- ☑ Better tooling and
- ☑ ecosystem
- ☑ Performance
- ☑ Easier to test

… or not

- ☒ Small apps
- ☒ Learning curve
- ☒ Boilerplate

Unit tests

Test specific methods or units of code.

Jasmine

KARMA

e2e test

Tests an app running in a real browser, interacting with it as a user would.

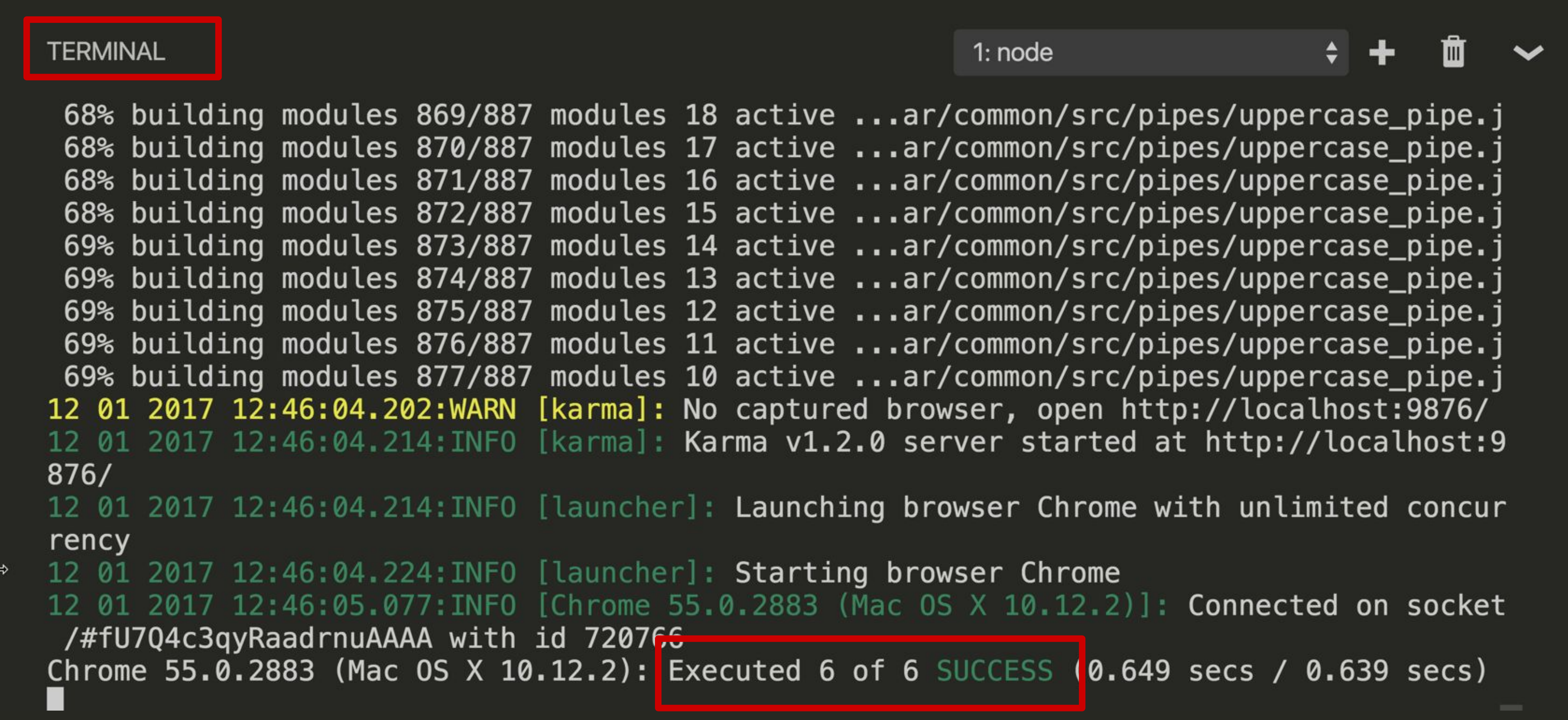
Jasmine

Protractor

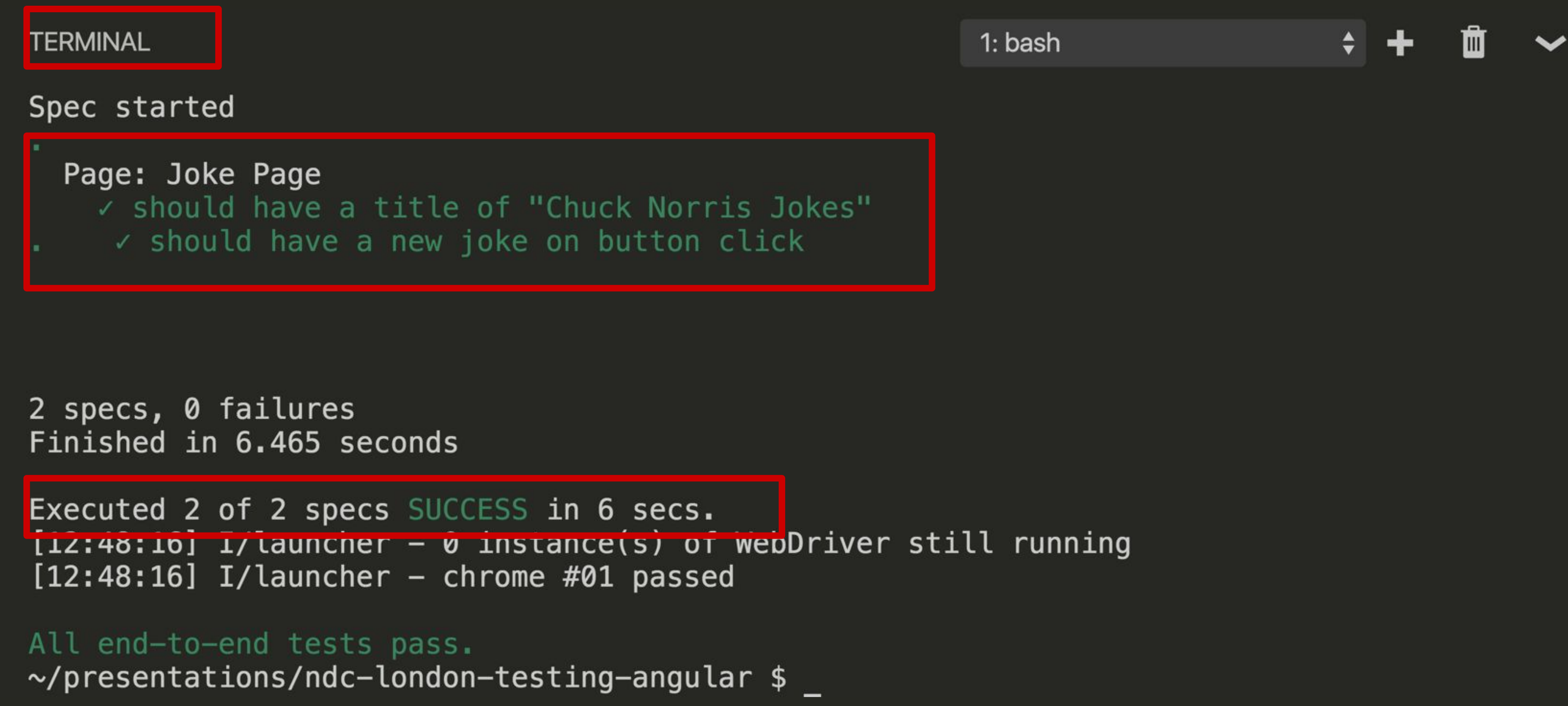
end to end testing for AngularJS

```
describe(`Component: JokeComponent`, () => {

    it(`should add 1 + 1 `, () => {
      expect(1 + 1).toEqual(2);
    });


    it(`should add 1 + 1 `, () => {
      expect(1 + 1).toEqual(2);
    });


});
```

Jasmine



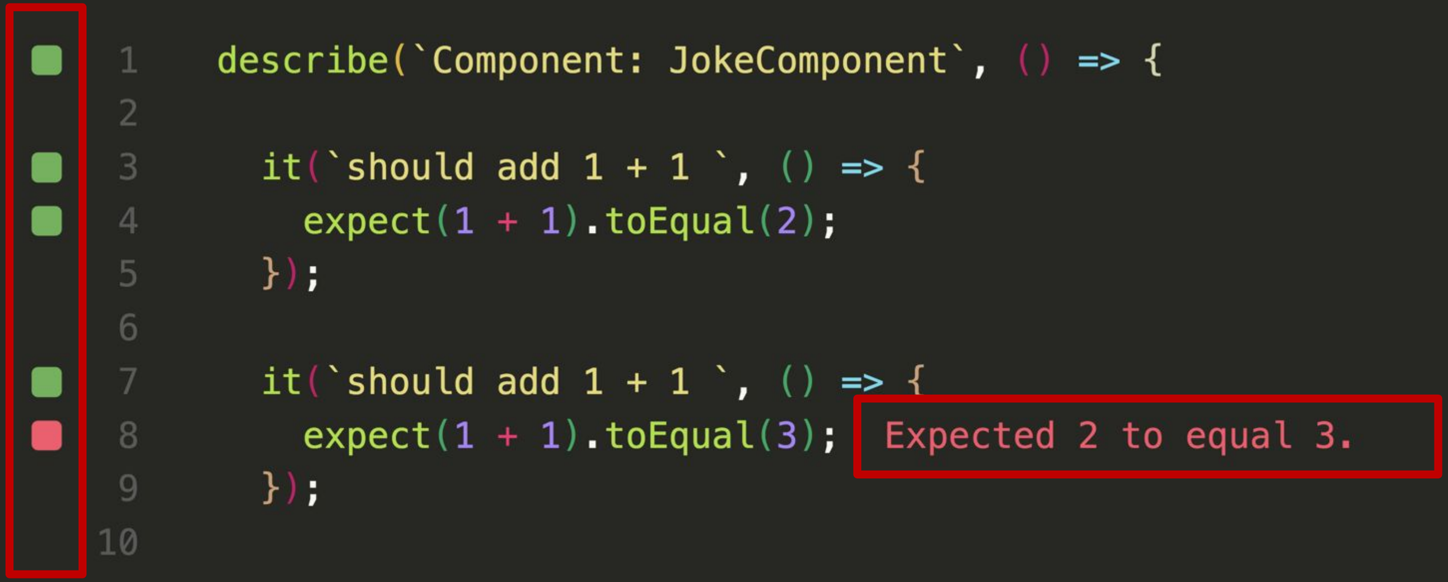
```
 68% building modules 869/887 modules 18 active ...ar/common/src/pipes/uppercase_pipe.j
 68% building modules 870/887 modules 17 active ...ar/common/src/pipes/uppercase_pipe.j
 68% building modules 871/887 modules 16 active ...ar/common/src/pipes/uppercase_pipe.j
 68% building modules 872/887 modules 15 active ...ar/common/src/pipes/uppercase_pipe.j
 69% building modules 873/887 modules 14 active ...ar/common/src/pipes/uppercase_pipe.j
 69% building modules 874/887 modules 13 active ...ar/common/src/pipes/uppercase_pipe.j
 69% building modules 875/887 modules 12 active ...ar/common/src/pipes/uppercase_pipe.j
 69% building modules 876/887 modules 11 active ...ar/common/src/pipes/uppercase_pipe.j
 69% building modules 877/887 modules 10 active ...ar/common/src/pipes/uppercase_pipe.j
12 01 2017 12:46:04.202:WARN [karma]: No captured browser, open http://localhost:9876/
12 01 2017 12:46:04.214:INFO [karma]: Karma v1.2.0 server started at http://localhost:9
876/
12 01 2017 12:46:04.214:INFO [launcher]: Launching browser Chrome with unlimited concur
rency
12 01 2017 12:46:04.224:INFO [launcher]: Starting browser Chrome
12 01 2017 12:46:05.077:INFO [Chrome 55.0.2883 (Mac OS X 10.12.2)]: Connected on socket
 /#fU7Q4c3qyRaadrnuAAAA with id 720766
Chrome 55.0.2883 (Mac OS X 10.12.2): Executed 6 of 6 SUCCESS (0.649 secs / 0.639 secs)
```



```
Spec started
.
  Page: Joke Page
    ✓ should have a title of "Chuck Norris Jokes"
.     ✓ should have a new joke on button click




2 specs, 0 failures
Finished in 6.465 seconds

Executed 2 of 2 specs SUCCESS in 6 secs.
[12:48:16] I/launcher - 0 instance(s) of WebDriver still running
[12:48:16] I/launcher - chrome #01 passed

All end-to-end tests pass.
~/presentations/ndc-london-testing-angular $ _
```

Protractor
end to end testing for AngularJS

wallaby

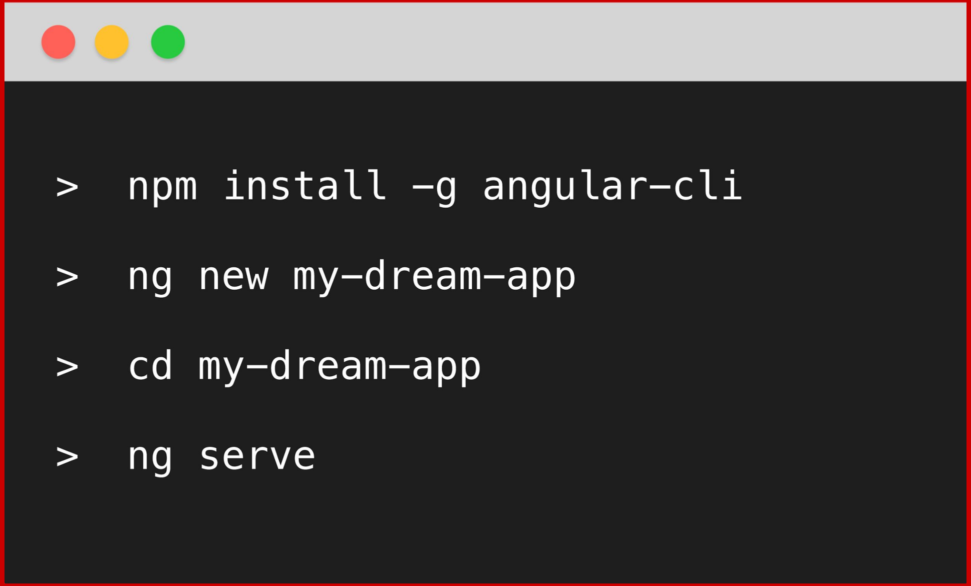


```
describe(`Component: JokeComponent`, () => {

  it(`should add 1 + 1 `, () => {
    expect(1 + 1).toEqual(2);
  });

  it(`should add 1 + 1 `, () => {
    expect(1 + 1).toEqual(3);
  });

```

```
Expected 2 to equal 3.
```

wallaby

Angular CLI



```
>   npm install -g angular-cli

>   ng new my-dream-app

>   cd my-dream-app

>   ng serve
```

Nx workspaces and angular cli

Components and nx libs

RxJS, services and HTTP module

Third party dependencies

Reactive forms

Routing

Ngrx and state libs

Entity adapter and router store

Unit tests

Deploying nx monorepo

Thanks

@dunchunter
duncan@duncanhunter. com.au