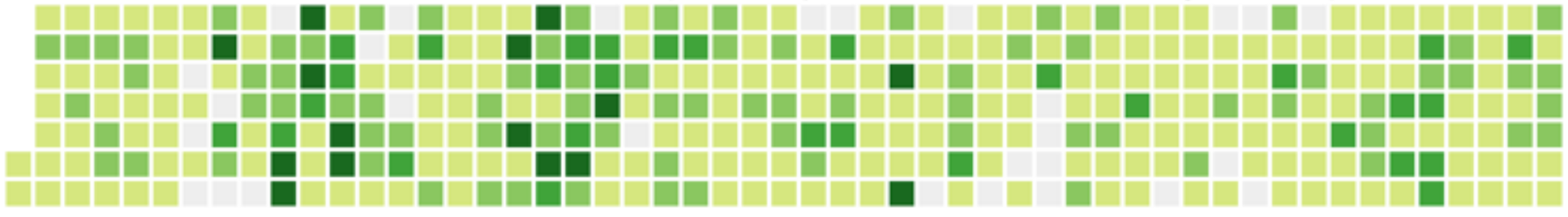
Building Universal Applications with

Angular 2

Minko Gechev

github.com/mgechev
twitter.com/mgechev
blog.mgechev.com

github.com/mgechev
twitter.com/mgechev
blog.mgechev.com



Agenda

Agenda

What is Angular 2?

Agenda

What is Angular 2?

What are the core concepts of Angular 2?

Agenda

What is Angular 2?
What are the core concepts of Angular 2?
What’s wrong with Angular 2?

Agenda

What is Angular 2?
What are the core concepts of Angular 2?
What’s wrong with Angular 2?
Why Angular 2 is awesome?

Agenda

What is Angular 2?
What are the core concepts of Angular 2?
What’s wrong with Angular 2?
Why Angular 2 is awesome?
Should I use Angular 2 in my next project?

What is Angular 2?

Angular is a development platform for building mobile and desktop applications

Angular is a development platform for building mobile and desktop applications

Core concepts of Angular 2

- Directives
- Components
- Pipes
- Services
- Dependency Injection

Directives

Primitive used for encapsulation of basic UI related logic

tooltip.ts

```
@Directive({
  selector: '[tooltip]',
  inputs: ['tooltip'],
  host: {
    '(mouseenter)': 'onMouseEnter()',
    '(mouseleave)': 'onMouseLeave()'
  }
})
export class Tooltip {
  private overlay:Overlay;
  private tooltip:string;
  constructor(private el:ElementRef, manager: OverlayManager) {
    this.el = el;
    this.overlay = manager.get();
  }
  onMouseEnter() {
    this.overlay.open(this.el, this.tooltip);
  }
  onMouseLeave() {
    this.overlay.close();
  }
}
```

tooltip.ts

```
@Directive({
  selector: '[tooltip]',
  inputs: ['tooltip'],
  host: {
    '(mouseenter)': 'onMouseEnter()',
    '(mouseleave)': 'onMouseLeave()'
  }
})
export class Tooltip {
  private overlay:Overlay;
  private tooltip:string;
  constructor(private el:ElementRef, manager: OverlayManager) {
    this.el = el;
    this.overlay = manager.get();
  }
  onMouseEnter() {
    this.overlay.open(this.el, this.tooltip);
  }
  onMouseLeave() {
    this.overlay.close();
  }
}
```

tooltip.ts

```
@Directive({
  selector: '[tooltip]',
  inputs: ['tooltip'],
  host: {
    '(mouseenter)': 'onMouseEnter()',
    '(mouseleave)': 'onMouseLeave()'
  }
})
export class Tooltip {
  private overlay:Overlay;
  private tooltip:string;
  constructor(private el:ElementRef, manager: OverlayManager) {
    this.el = el;
    this.overlay = manager.get();
  }
  onMouseEnter() {
    this.overlay.open(this.el, this.tooltip);
  }
  onMouseLeave() {
    this.overlay.close();
  }
}
```

What a minute...

this looks like Java

Angular 2 is written in TypeScript


Microsoft

TypeScript is...

superset of ECMAScript

optional type checking

open source

...you can still use ES5...

tooltip.js

```
var Tooltip = ng.Directive({
  selector: '[tooltip]',
  inputs: ['tooltip'],
  host: {
    '(mouseenter)': 'onMouseEnter()',
    '(mouseleave)': 'onMouseLeave()'
  }
})
.Class({
  constructor: [ng.Inject(ng.ElementRef),
   ng.Inject(OverlayManager),
   function (tooltip, el, manager) {
    this.el = el;
    this.overlay = manager.get(tooltip);
  }],
  onMouseEnter() {
    this.overlay.open(this.el, this.tooltip);
  },
  onMouseLeave() {
    this.overlay.close();
  }
});
```

tooltip.js

```
var Tooltip = ng.Directive({
  selector: '[tooltip]',
  inputs: ['tooltip'],
  host: {
    '(mouseenter)': 'onMouseEnter()',
    '(mouseleave)': 'onMouseLeave()'
  }
})
.Class({
  constructor: [ng.Inject(ng.ElementRef),
   ng.Inject(OverlayManager),
   function (tooltip, el, manager) {
    this.el = el;
    this.overlay = manager.get(tooltip);
  }],
  onMouseEnter() {
    this.overlay.open(this.el, this.tooltip);
  },
  onMouseLeave() {
    this.overlay.close();
  }
});
```

tooltip.js

```
var Tooltip = ng.Directive({
  selector: '[tooltip]',
  inputs: ['tooltip'],
  host: {
    '(mouseenter)': 'onMouseEnter()',
    '(mouseleave)': 'onMouseLeave()'
  }
})
.Class({
  constructor: [ng.Inject(ng.ElementRef),
   ng.Inject(OverlayManager),
   function (tooltip, el, manager) {
    this.el = el;
    this.overlay = manager.get(tooltip);
  }],
  onMouseEnter() {
    this.overlay.open(this.el, this.tooltip);
  },
  onMouseLeave() {
    this.overlay.close();
  }
});
```

Components

Directives with views

hello-world.ts

```
@Component({
  selector: 'hello-world'
})
@View({
  template: '<h1>Hello, {{this.target}}!</h1>'
})
class HelloWorld {
  target:string;
  constructor() {
    this.target = 'world';
  }
}
```

hello-world.ts

```
@Component({
  selector: 'hello-world'
})
@View({
  template: '<h1>Hello, {{this.target}}!</h1>'
})
class HelloWorld {
  target:string;
  constructor() {
    this.target = 'world';
  }
}
```

hello-world.ts

```
@Component({
  selector: 'hello-world'
})
@View({
  template: '<h1>Hello, {{this.target}}!</h1>'
})
class HelloWorld {
  target:string;
  constructor() {
    this.target = 'world';
  }
}
```

Pipes

Encapsulate the data transformation logic

lowercase-pipe.ts

```
@Pipe({
  name: 'lowercase'
})
class LowerCasePipe implements PipeTransform {
  transform(value: string): string {
    if (!value) return value;
    if (typeof value !== 'string') {
      throw new Error('Invalid pipe value', value);
    }
    return value.toLowerCase();
  }
}
```

Services

Simply, ES2015 classes which can be wired together with...

Dependency Injection

di.ts

```
import { provide, Injector, Injectable }
    from 'angular2/angular2';

@Injectable()
class DataMapper {
  constructor(private http: Http) {}
}

class Http {}

let injector = Injector.resolveAndCreate([
  provide(Http).toValue(new Http()),
  DataMapper
]);

injector.get(DataMapper);
```

di.ts

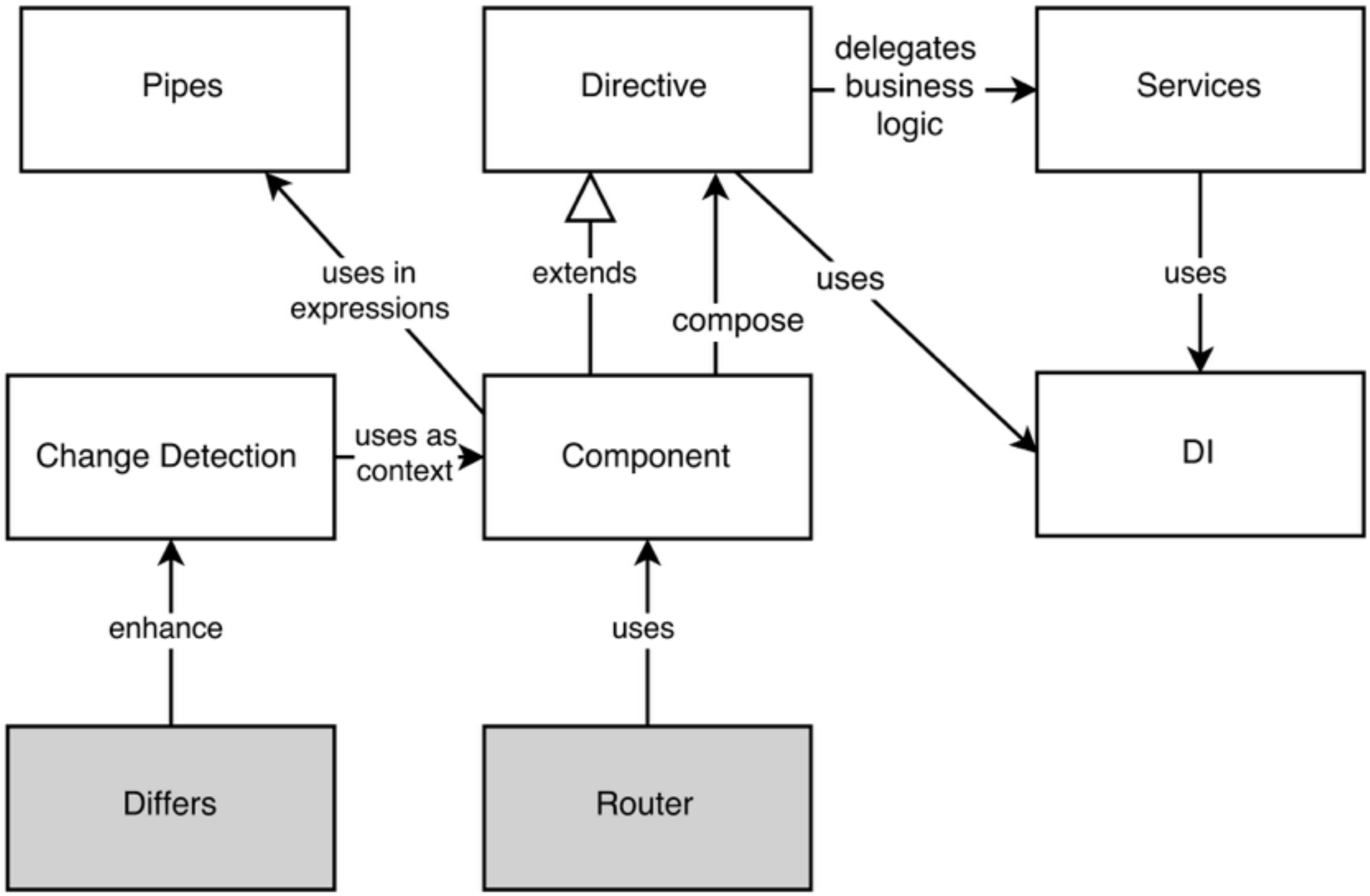
```
import { provide, Injector, Injectable }
    from 'angular2/angular2';

@Injectable()
class DataMapper {
  constructor(private http: Http) {}
}

class Http {}

let injector = Injector.resolveAndCreate([
  provide(Http).toValue(new Http()),
  DataMapper
]);

injector.get(DataMapper);
```

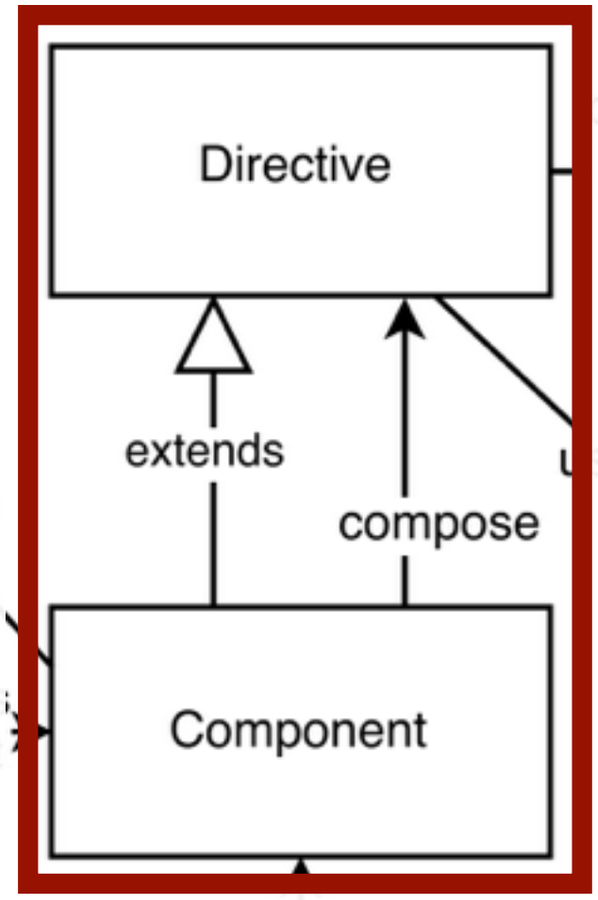
di.ts

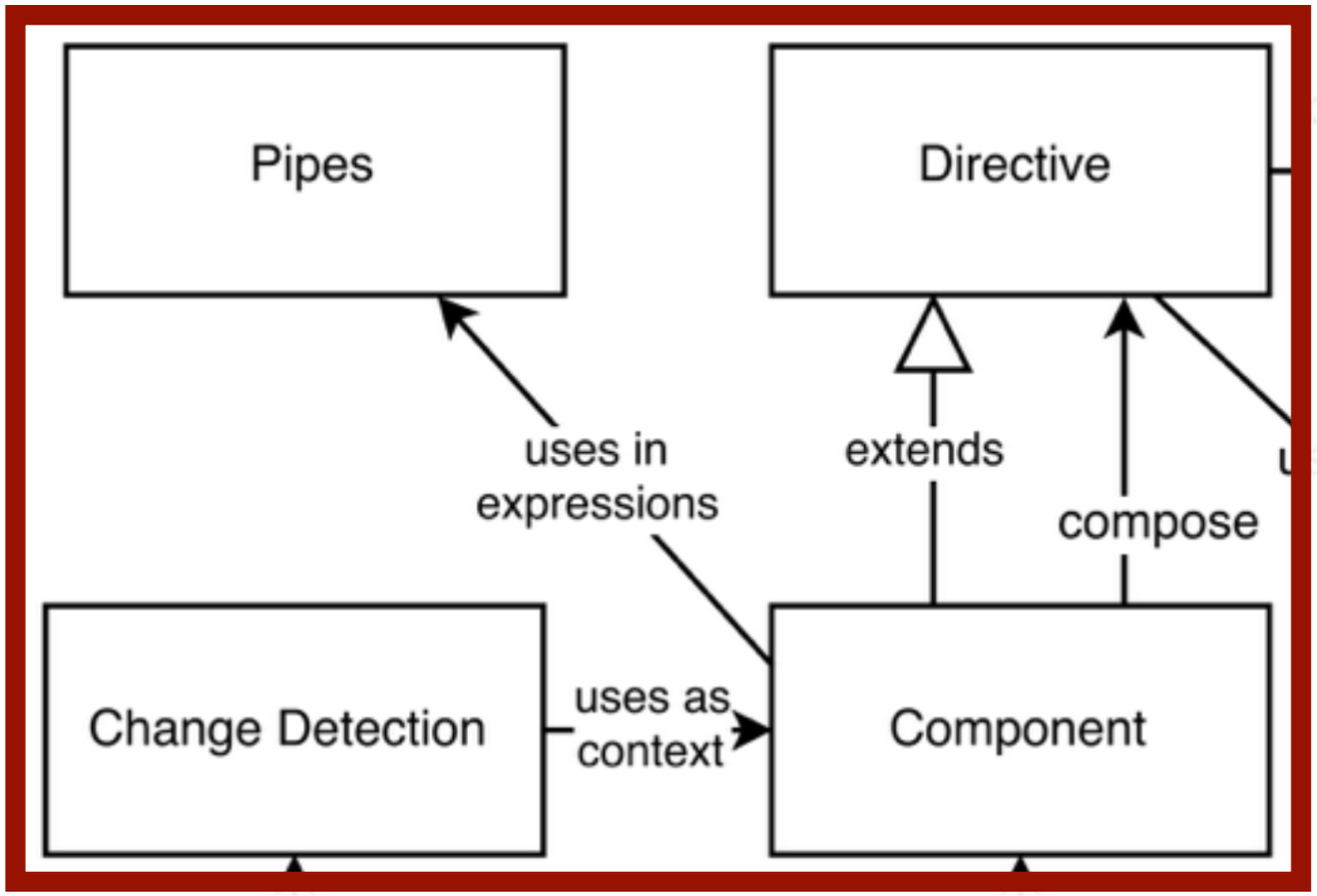
```
import { provide, Injector, Injectable }
    from 'angular2/angular2';

@Injectable()
class DataMapper {
  constructor(private http: Http) {}
}

class Http {}

let injector = Injector.resolveAndCreate([
  provide(Http).toValue(new Http()),
  DataMapper
]);

injector.get(DataMapper);
```

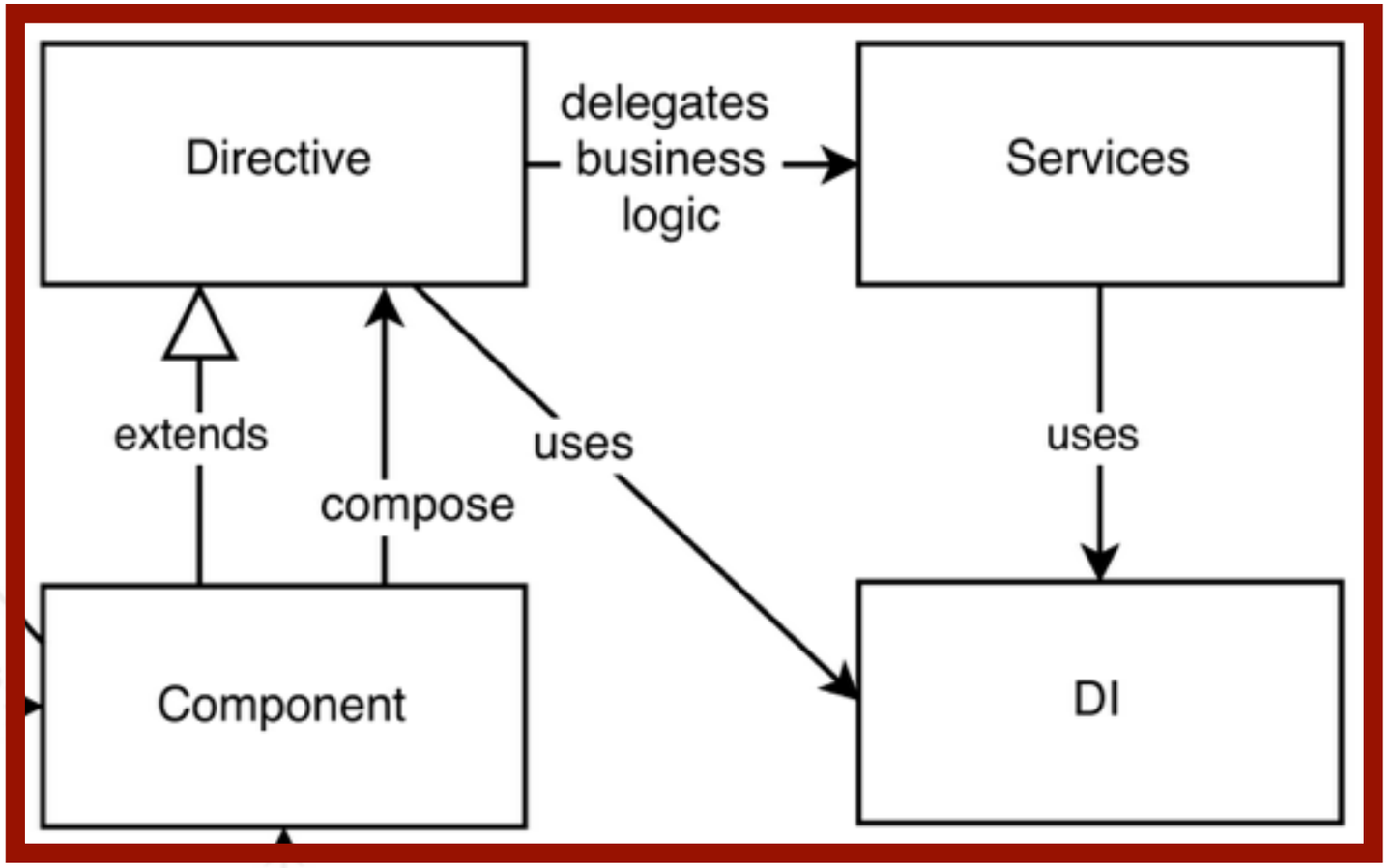
di.ts

```
import { provide, Injector, Injectable }
    from 'angular2/angular2';

@Injectable()
class DataMapper {
  constructor(private http: Http) {}
}

class Http {}

let injector = Injector.resolveAndCreate([
  provide(Http).toValue(new Http()),
  DataMapper
]);

injector.get(DataMapper);
```


Pipes
Directive
delegates
business
logic
Services
uses in
expressions
extends
compose
uses
uses
Change Detection
uses as
context
Component
DI
enhance
uses
Differs
Router


Directive
extends
compose
Component


Pipes
Directive
uses in
expressions
extends
compose
Change Detection
uses as
context
Component


Directive
delegates
business
logic
Services
extends
compose
uses
uses
Component
DI

Now you know the basics!

Lets take a step back...

Angular is a development platform for building mobile and desktop applications

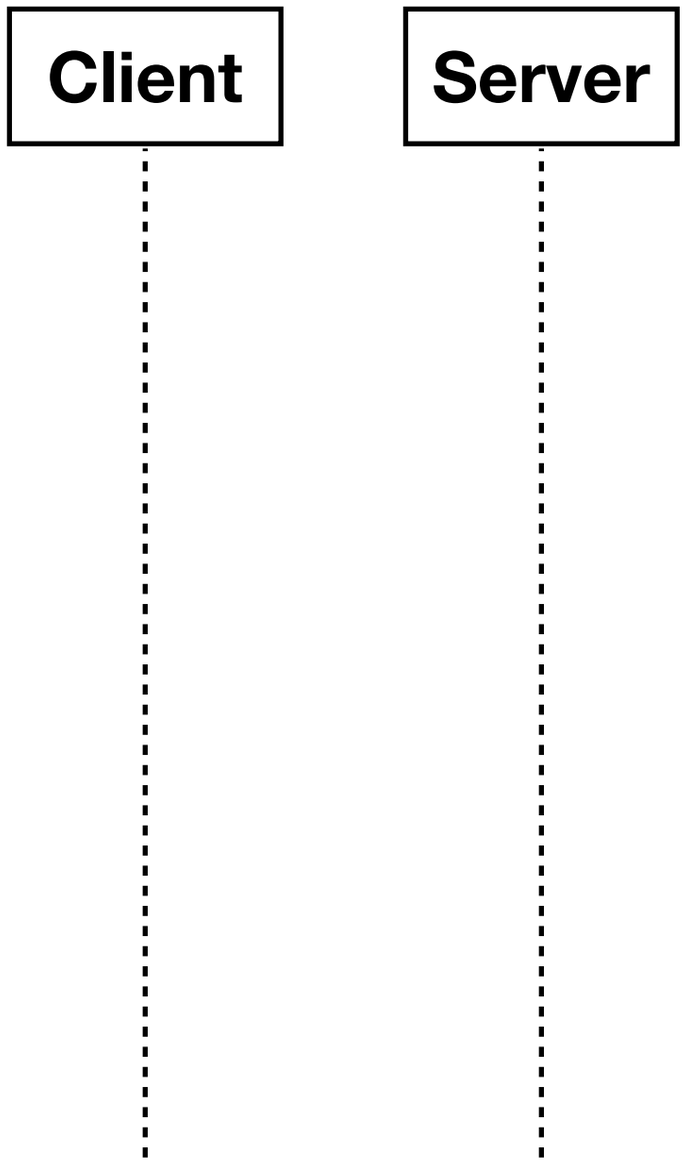
Angular is a development platform for building mobile and desktop applications

Angular 2 is platform agnostic

Not coupled with DOM, even HTML

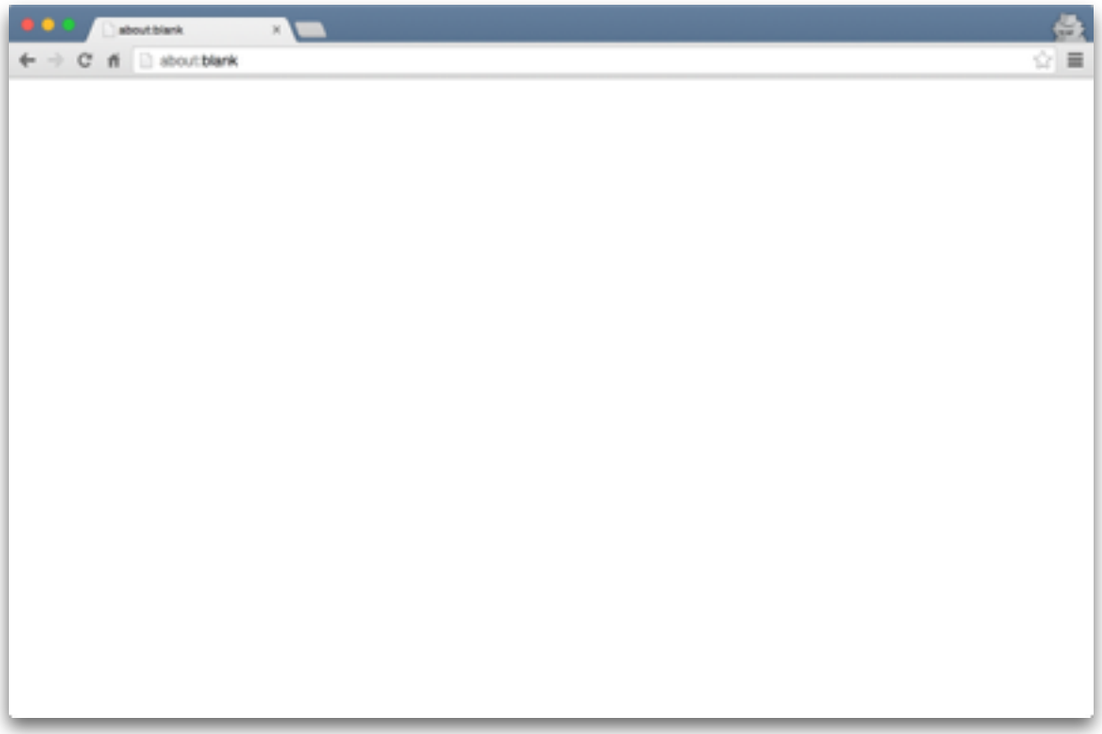
NativeScript

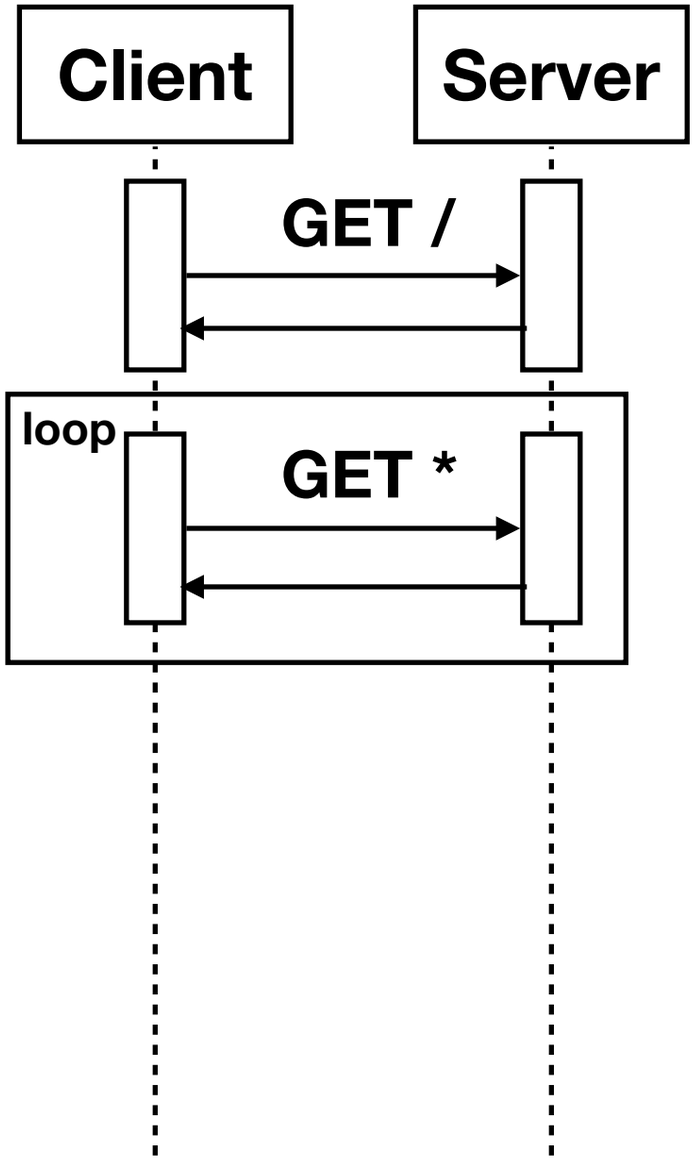
...and even more...

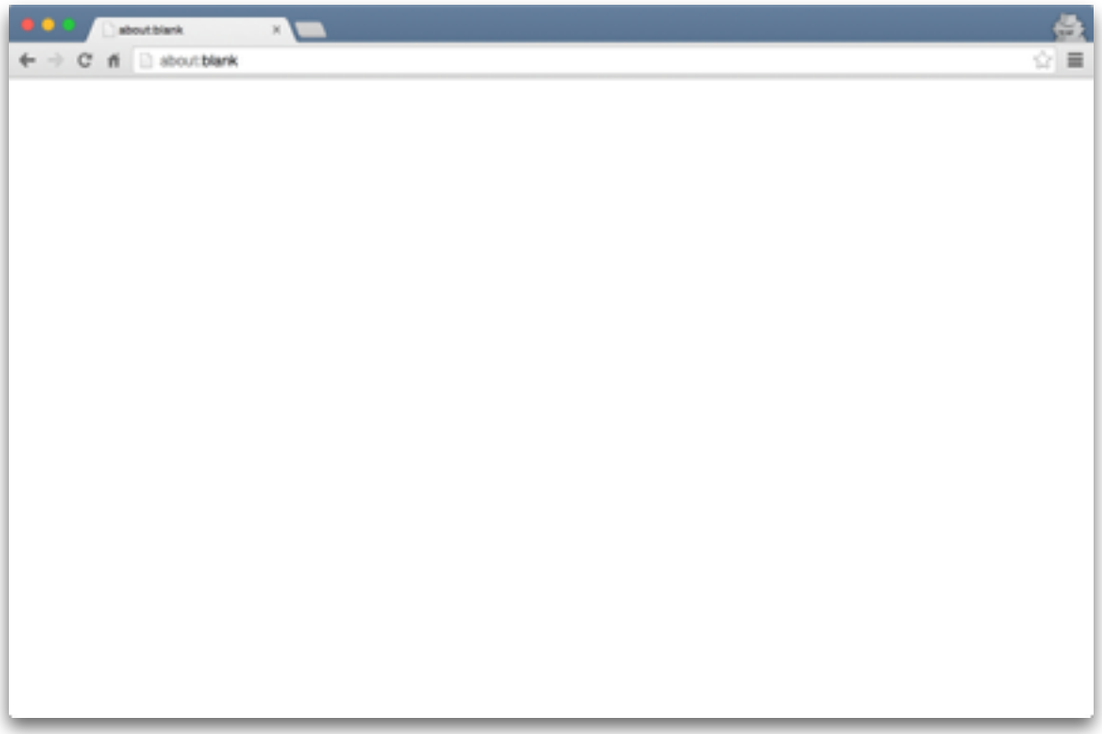

Client
Server

Client

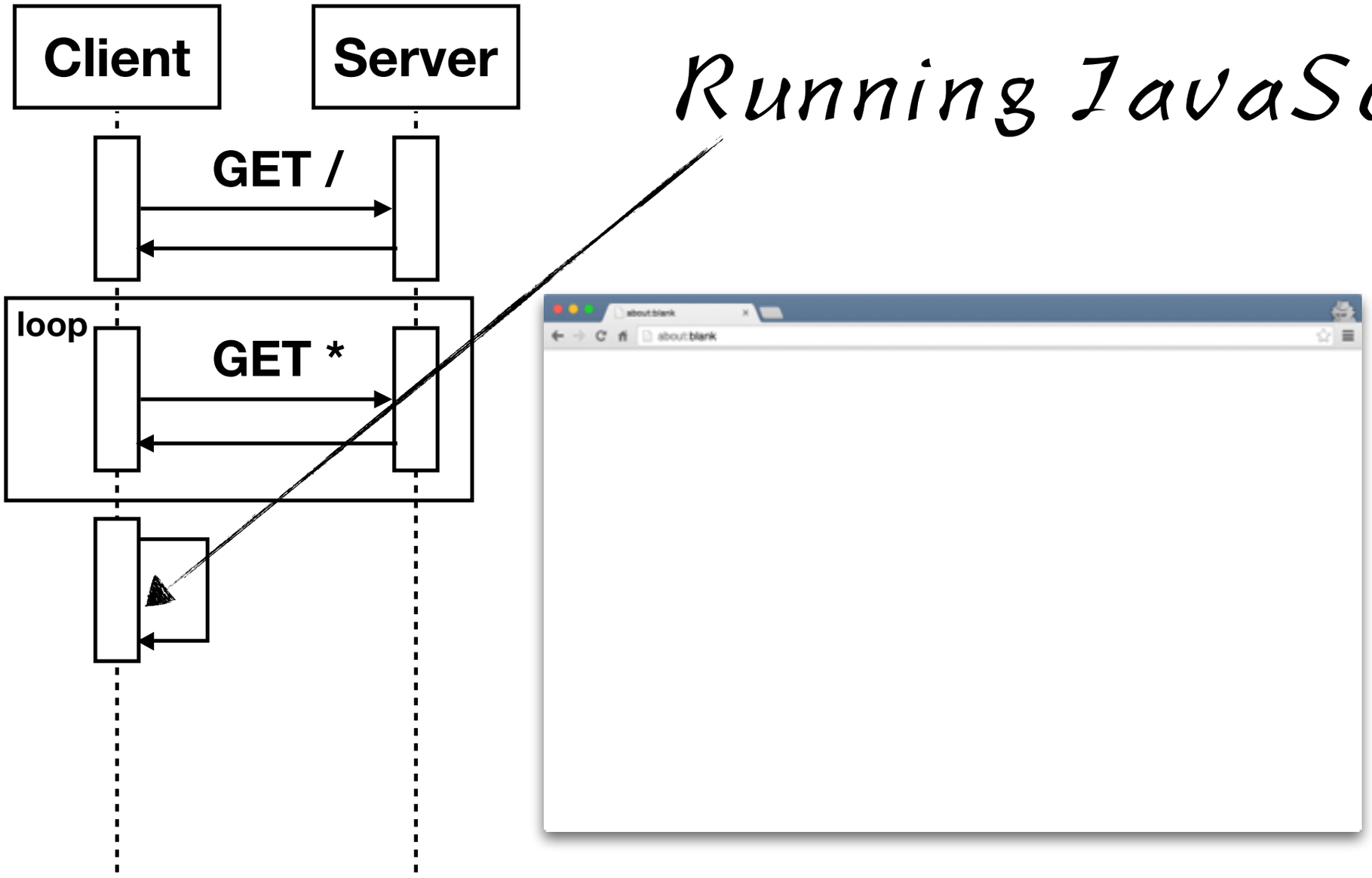
Server

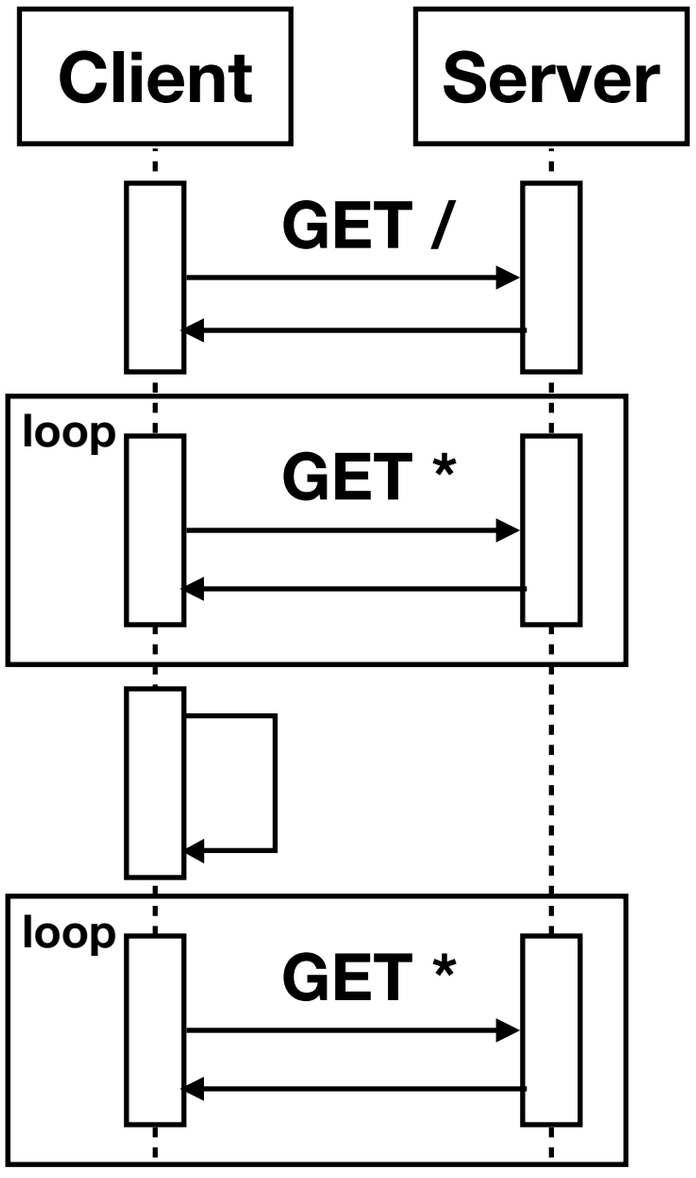


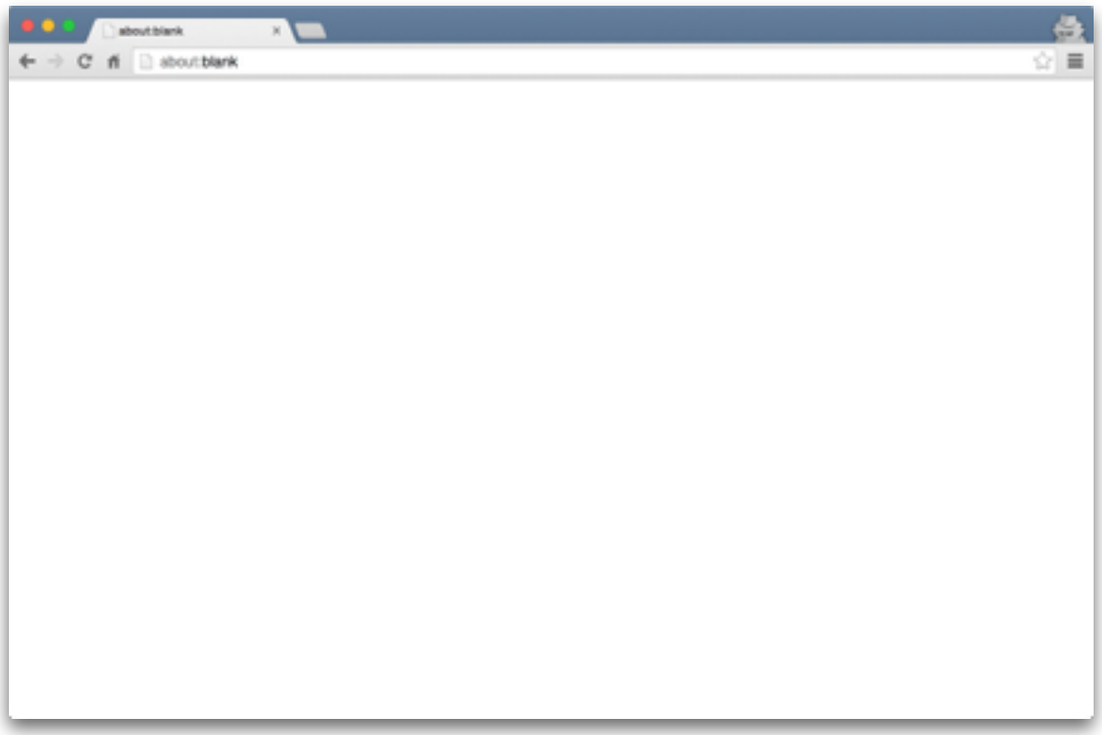

Client
Server
GET /
loop
GET *

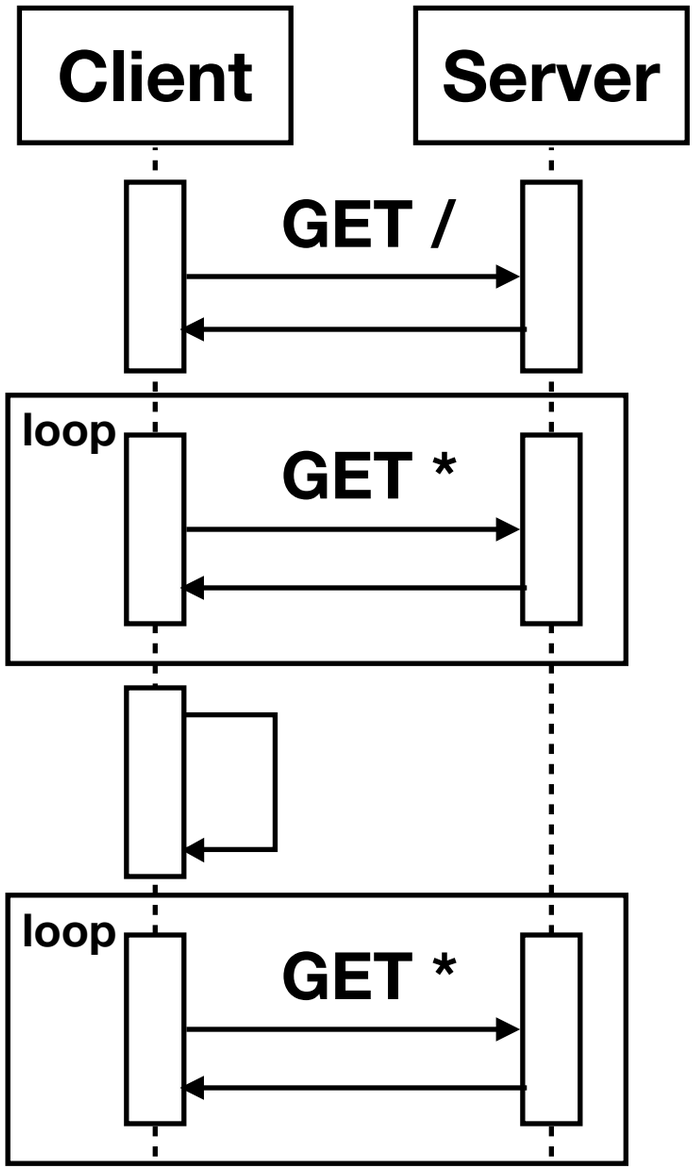

about:blank
about:blank

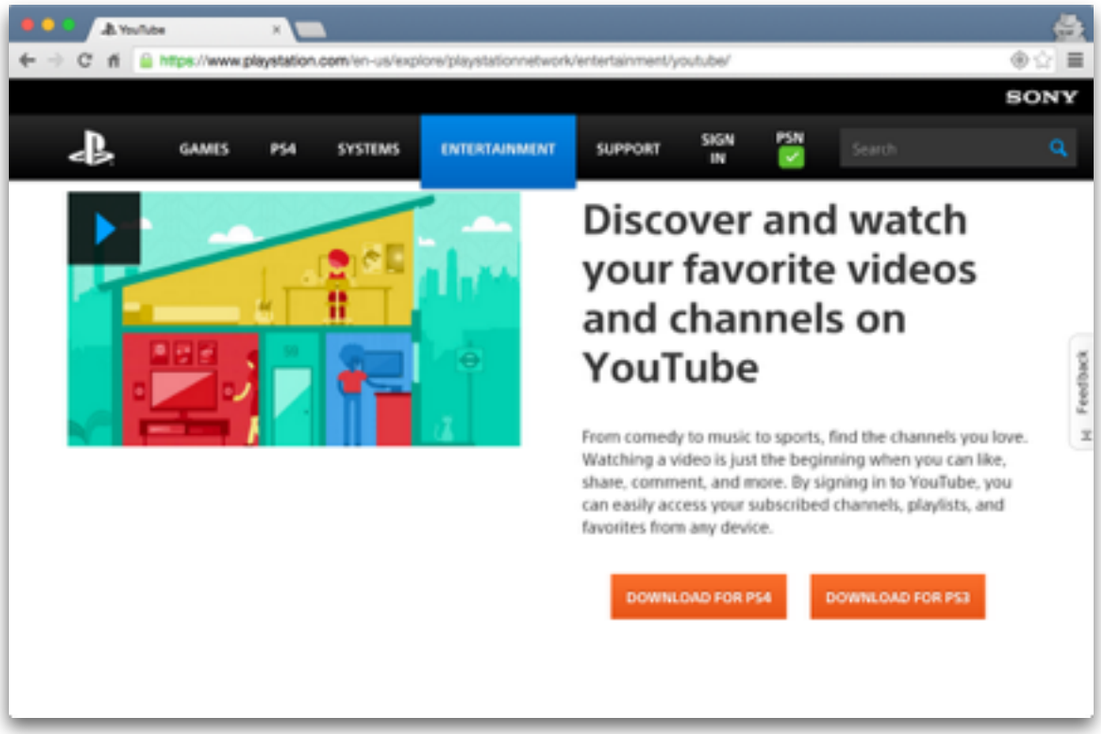
Running JavaScript



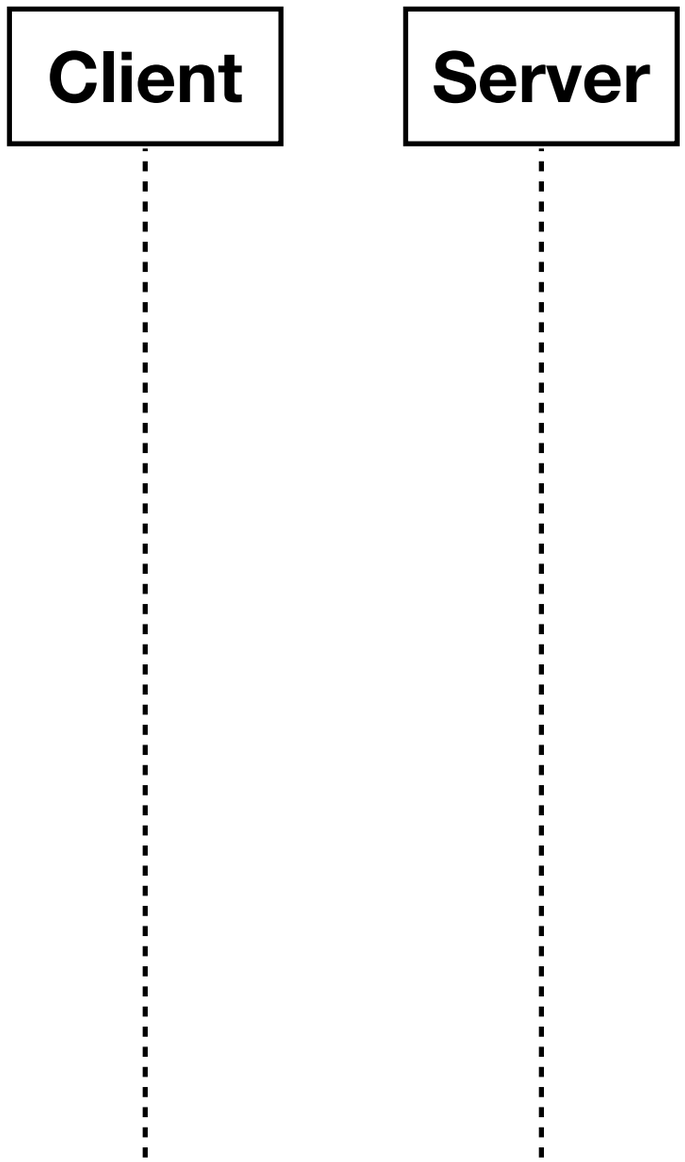

Client
Server
GET /
loop
GET *
loop
GET *


about:blank
about:blank


Client
Server
GET /
loop
GET *
loop
GET *

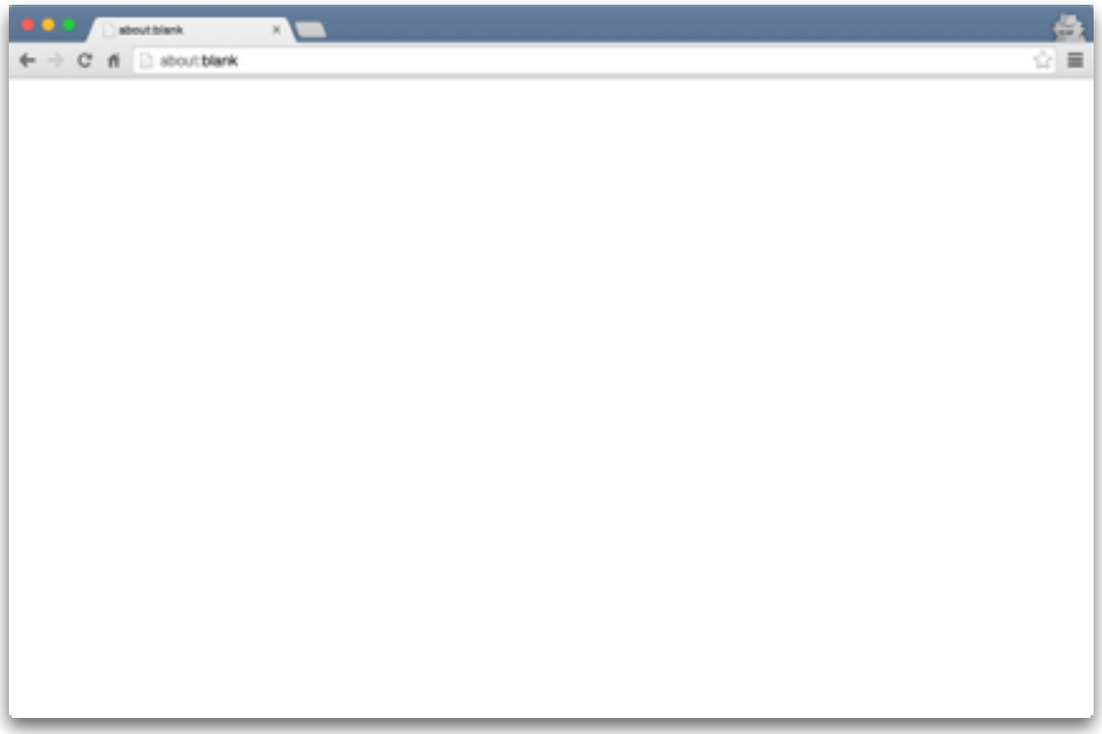

YouTube
https://www.playstation.com/en-us/explore/playstationnetwork/entertainment/youtube/
SONY
GAMES
PS4
SYSTEMS
ENTERTAINMENT
SUPPORT
SIGN IN
PSN
Search
Discover and watch your favorite videos and channels on YouTube
From comedy to music to sports, find the channels you love. Watching a video is just the beginning when you can like, share, comment, and more. By signing in to YouTube, you can easily access your subscribed channels, playlists, and favorites from any device.
DOWNLOAD FOR PS4
DOWNLOAD FOR PS3
Feedback

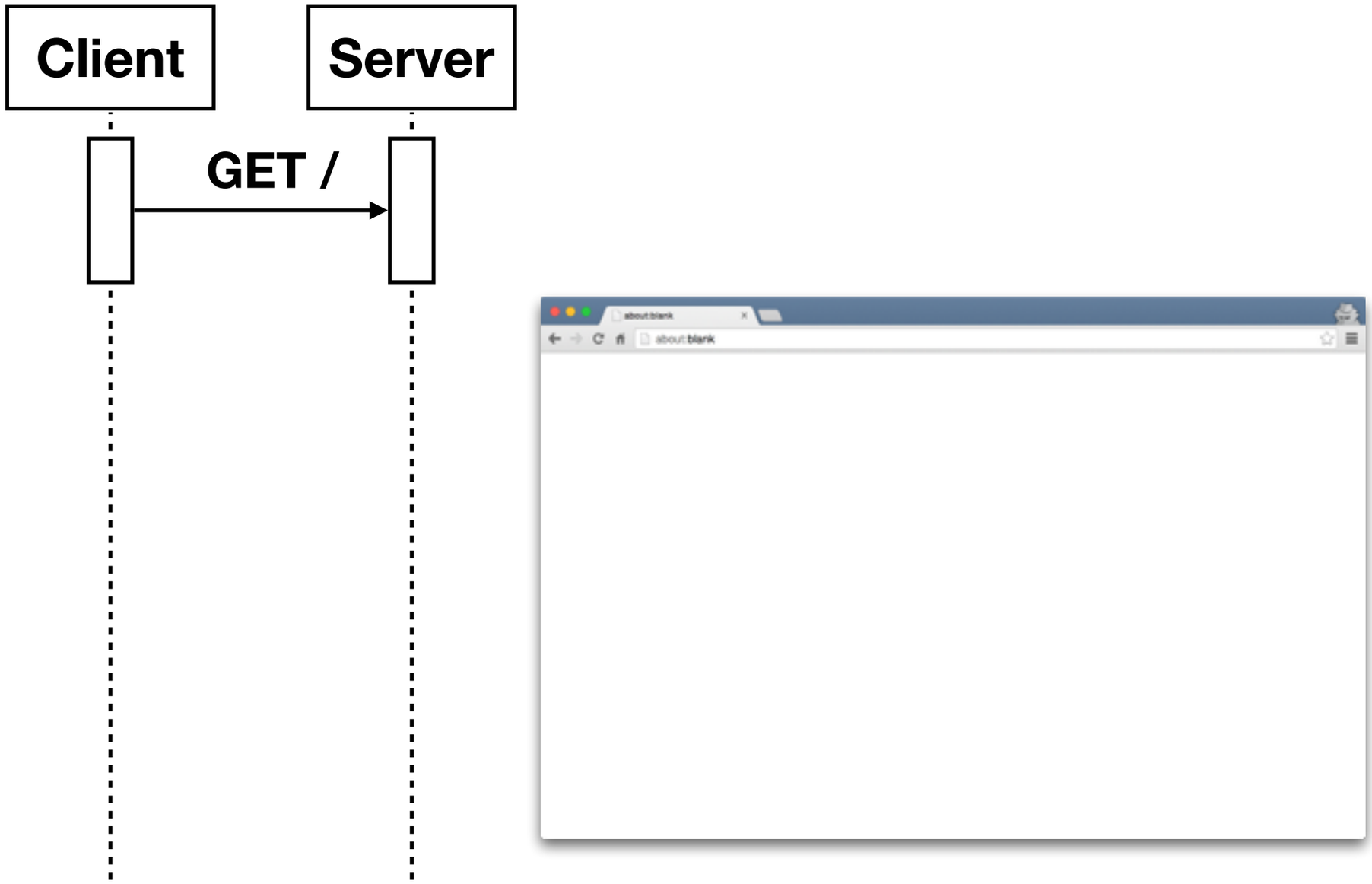
server-side rendering

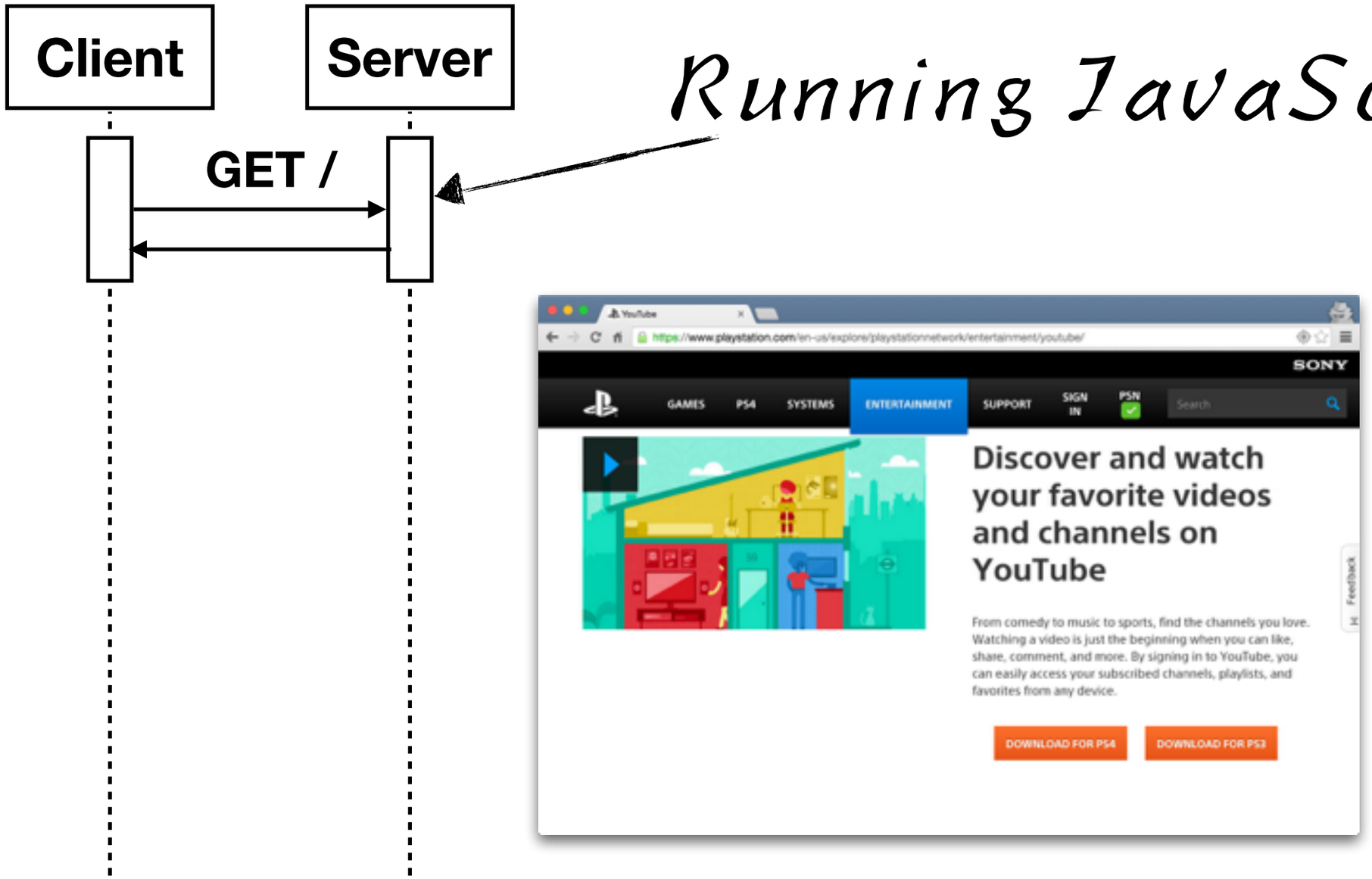

Client
Server

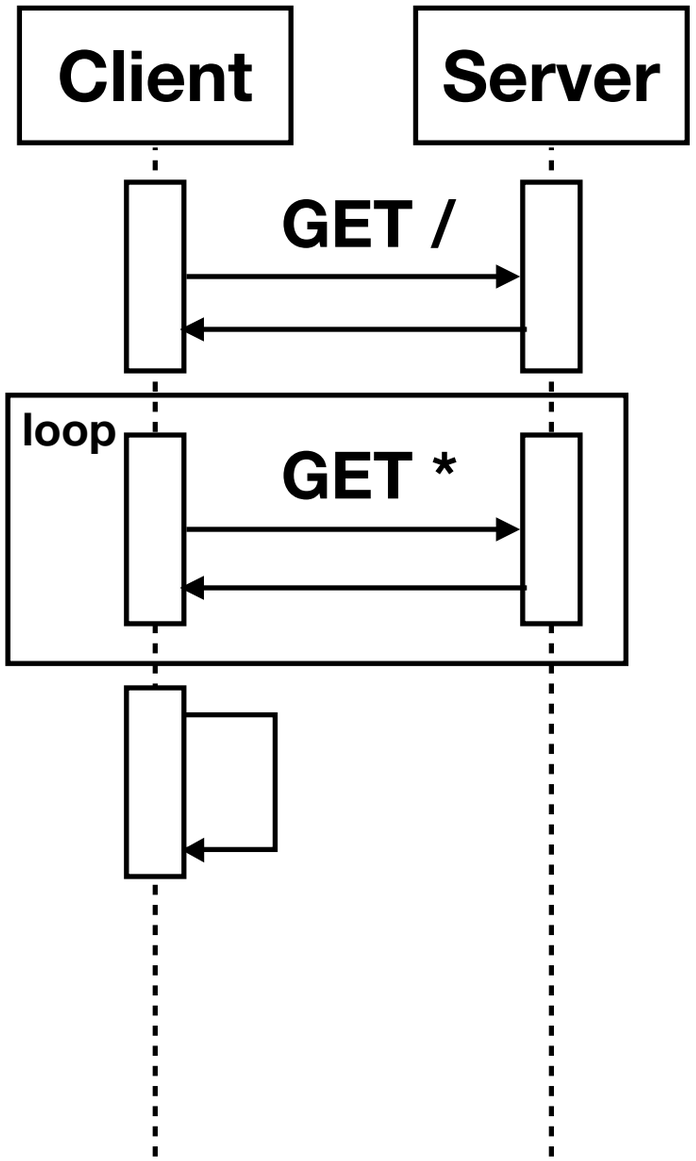
Client

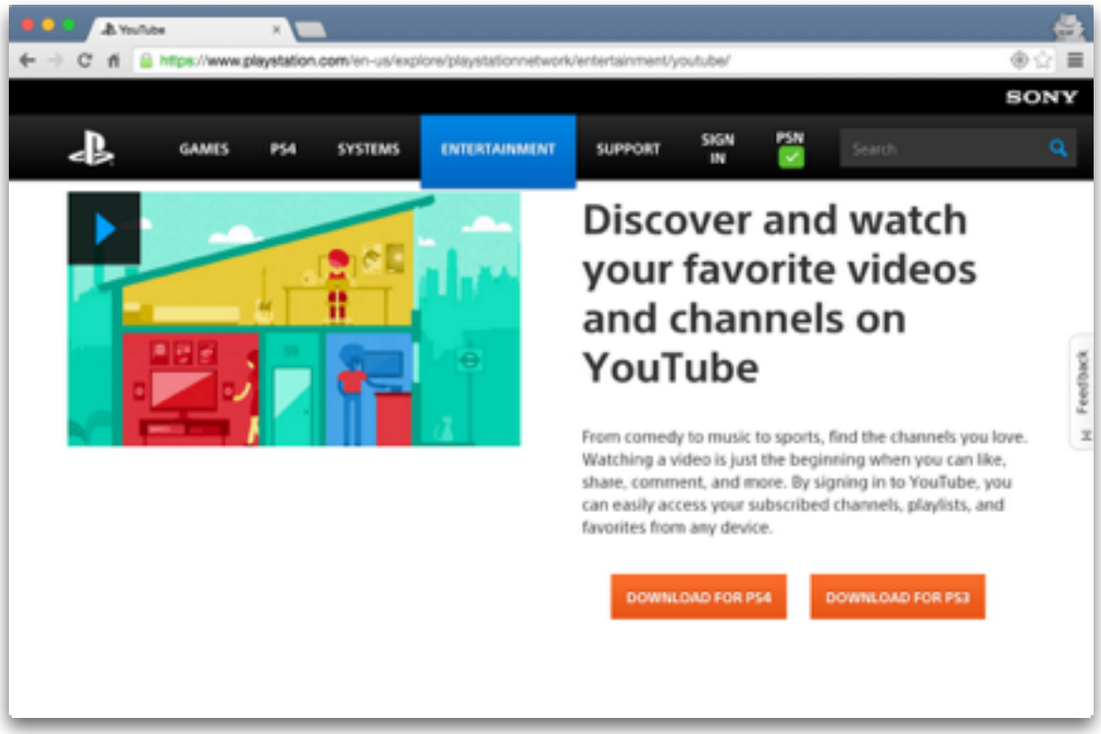
Server



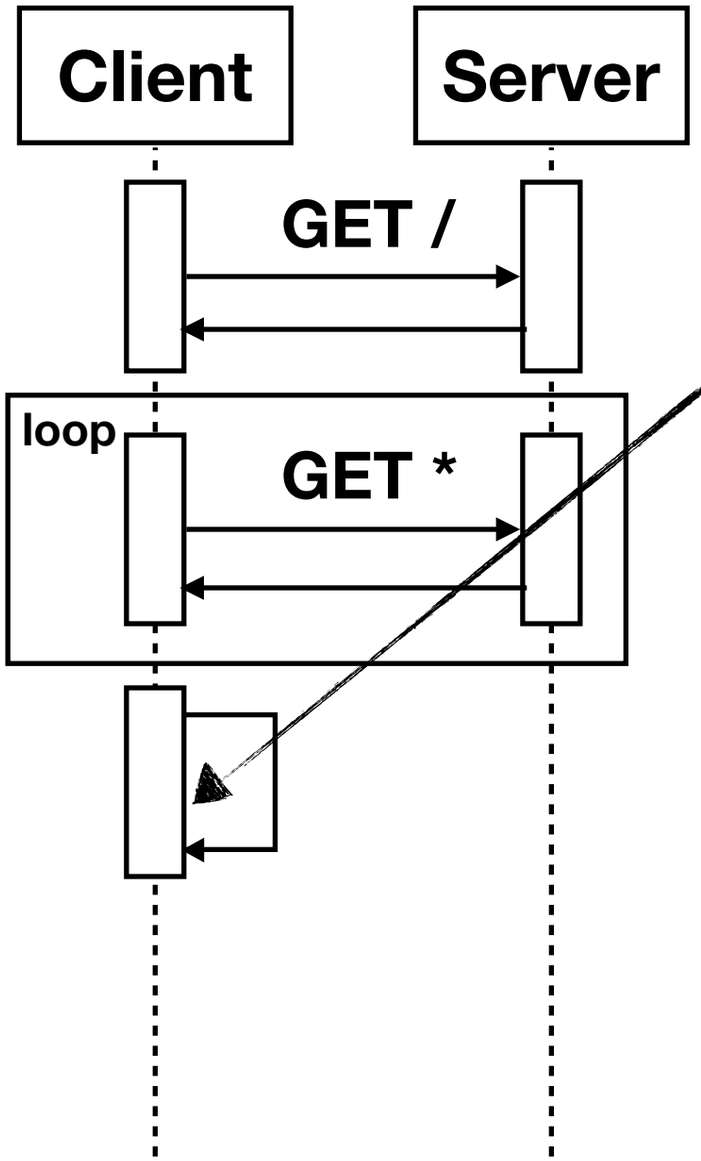

Client
Server
GET /
about:blank
about:blank

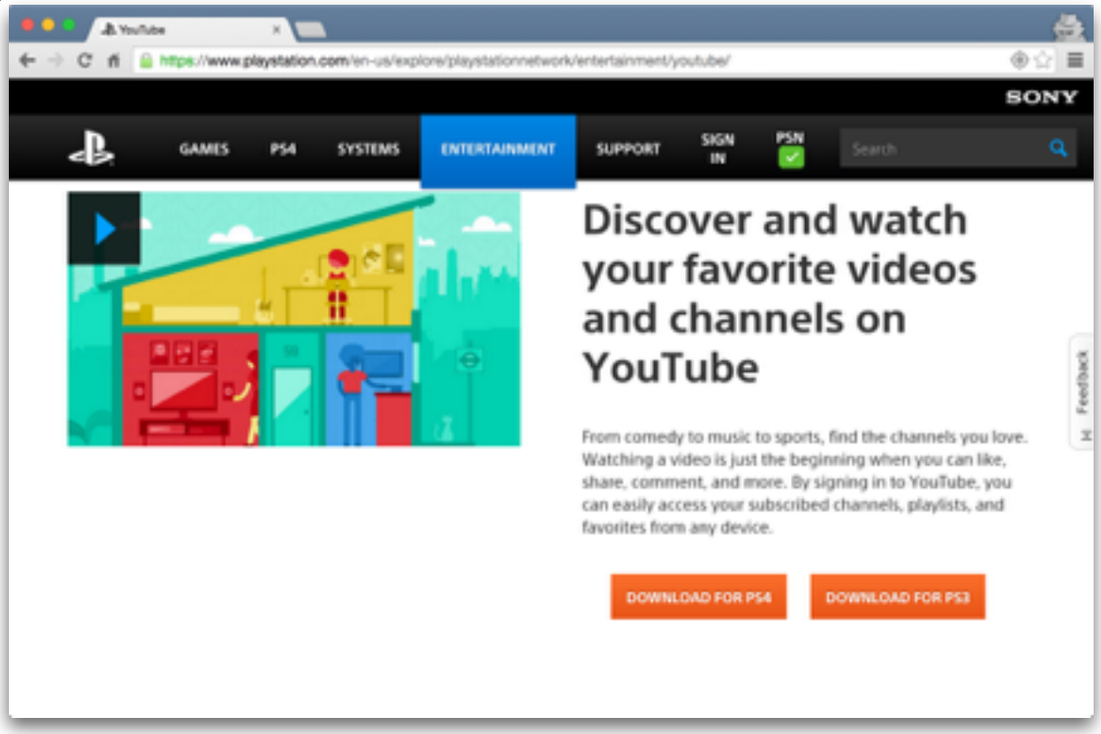

Client
Server
GET /
Running JavaScript
YouTube
https://www.playstation.com/en-us/explore/playstationnetwork/entertainment/youtube/
SONY
GAMES
PS4
SYSTEMS
ENTERTAINMENT
SUPPORT
SIGN IN
PSN
Search
Discover and watch your favorite videos and channels on YouTube
From comedy to music to sports, find the channels you love. Watching a video is just the beginning when you can like, share, comment, and more. By signing in to YouTube, you can easily access your subscribed channels, playlists, and favorites from any device.
DOWNLOAD FOR PS4
DOWNLOAD FOR PS3
Feedback

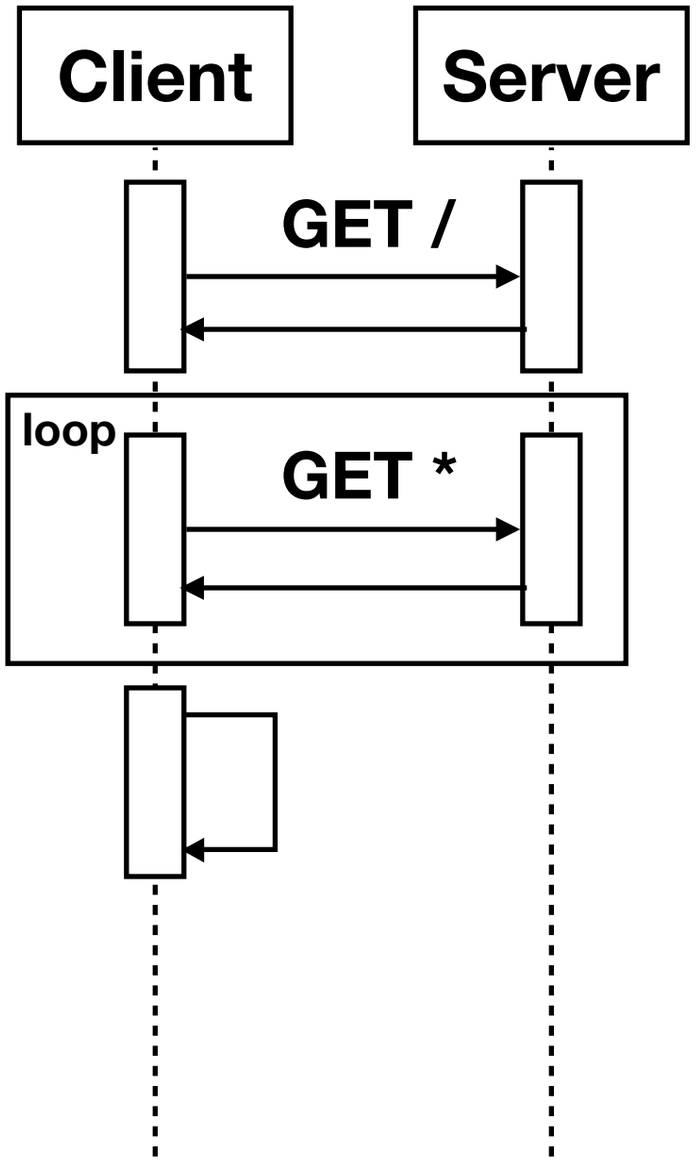

Client
Server
GET /
loop
GET *

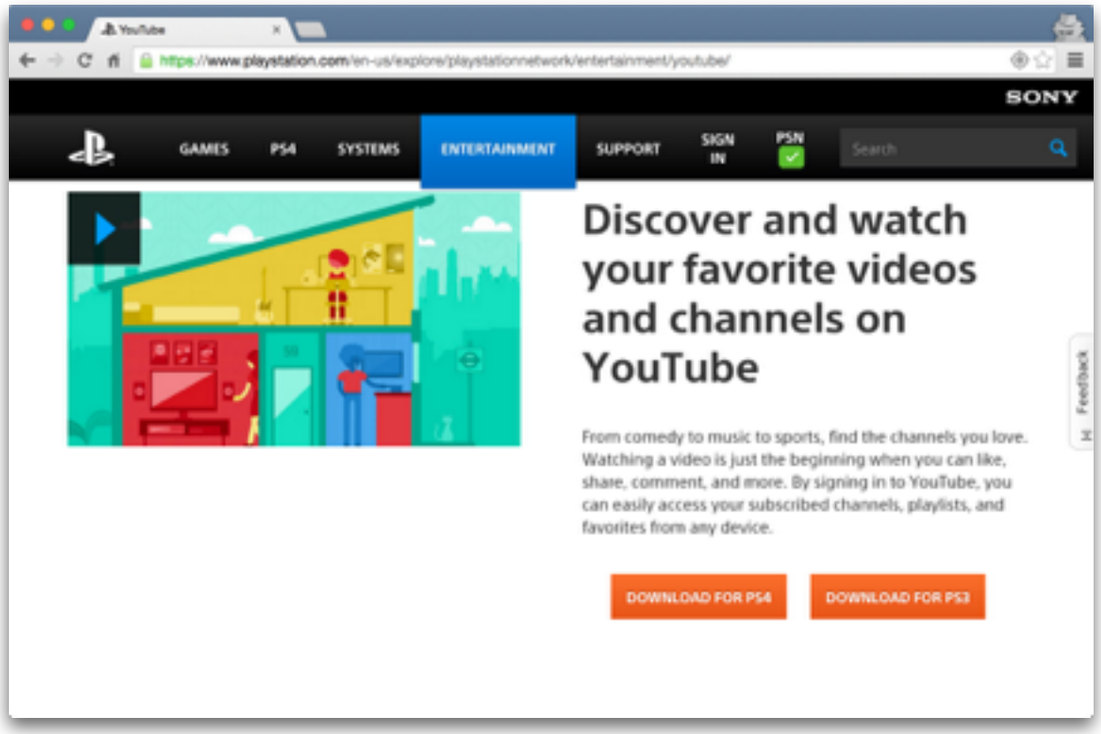

https://www.playstation.com/en-us/explore/playstationnetwork/entertainment/youtube/
SONY
GAMES
PS4
SYSTEMS
ENTERTAINMENT
SUPPORT
SIGN IN
PSN
Search
Discover and watch your favorite videos and channels on YouTube
From comedy to music to sports, find the channels you love. Watching a video is just the beginning when you can like, share, comment, and more. By signing in to YouTube, you can easily access your subscribed channels, playlists, and favorites from any device.
DOWNLOAD FOR PS4
DOWNLOAD FOR PS3
Feedback

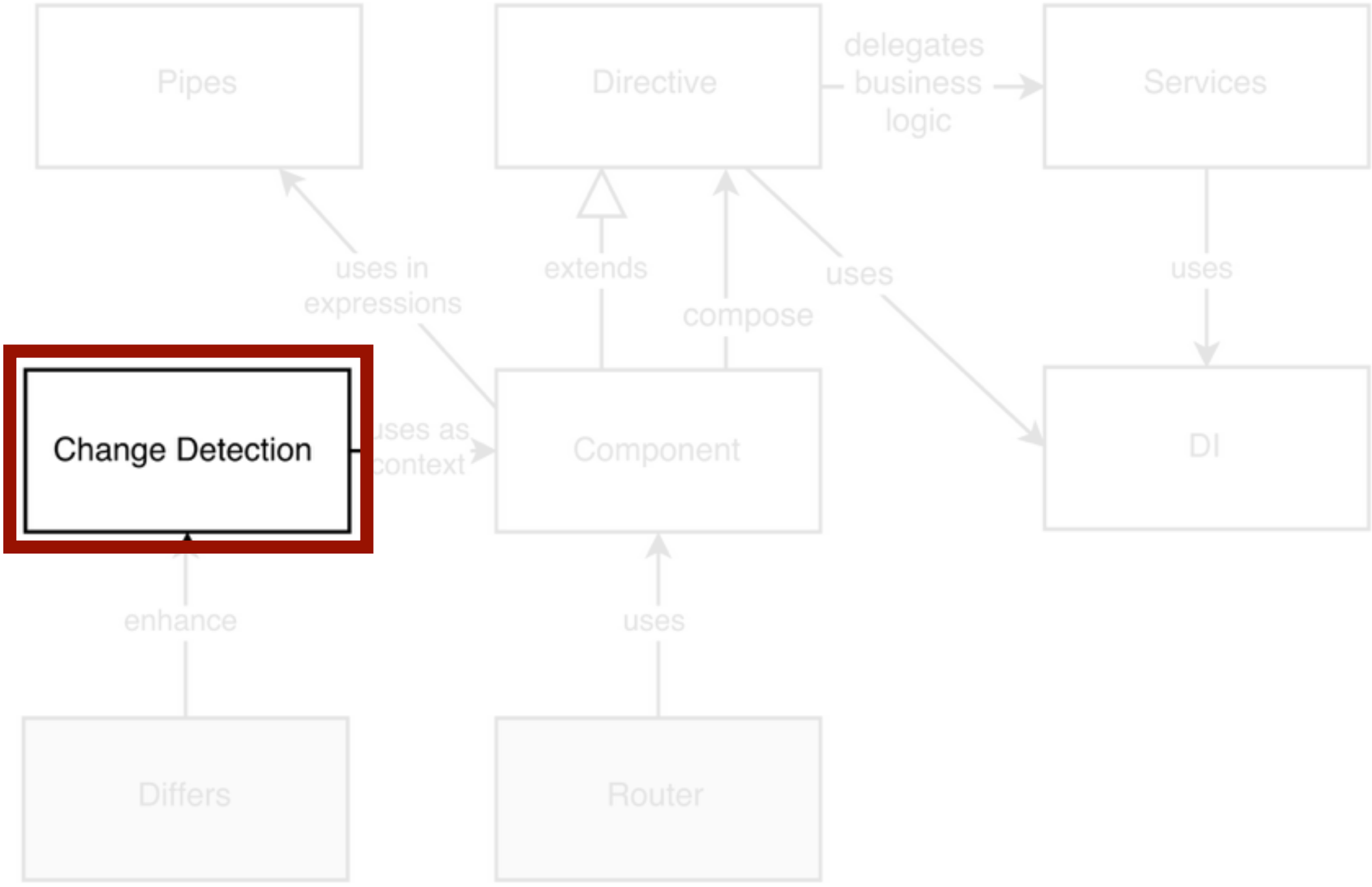
Running JavaScript






Client
Server
GET /
loop
GET *


https://www.playstation.com/en-us/explore/playstationnetwork/entertainment/youtube/
SONY
GAMES
PS4
SYSTEMS
ENTERTAINMENT
SUPPORT
SIGN IN
PSN
Search
Discover and watch your favorite videos and channels on YouTube
From comedy to music to sports, find the channels you love. Watching a video is just the beginning when you can like, share, comment, and more. By signing in to YouTube, you can easily access your subscribed channels, playlists, and favorites from any device.
DOWNLOAD FOR PS4
DOWNLOAD FOR PS3
Feedback


Pipes
Directive
delegates
business
logic
Services
uses in
expressions
extends
compose
uses
uses
Change Detection
uses as
context
Component
DI
enhance
uses
Differs
Router

Change detection can be run into separate process

Benefits of WebWorkers

Application logic doesn't block render thread

Run application across multiple windows or frames

Better compete performance-wise with native mobile apps

Test without the browser

Works with all modern browsers (IE 10+)

4:40 / 25:42

» Auto Replay

Using Web Workers for more responsive apps – Jason Teplitz

AngularConnect 2015

Subscribe 1,767

927 views

Add to Share More

21 0

so...what's wrong with Angular 2?

HTML enhanced for web apps!

View on GitHub

Download (1.4.7 / 1.2.29)

Design Docs & Notes

Follow +AngularJS on

Follow @angularjs 93.7K followers

Tweet 4,942

Learn Angular in your browser for free!

HTML enhanced for web apps!

View on GitHub

Download (1.4.7 / 1.2.29)

Design Docs & Notes

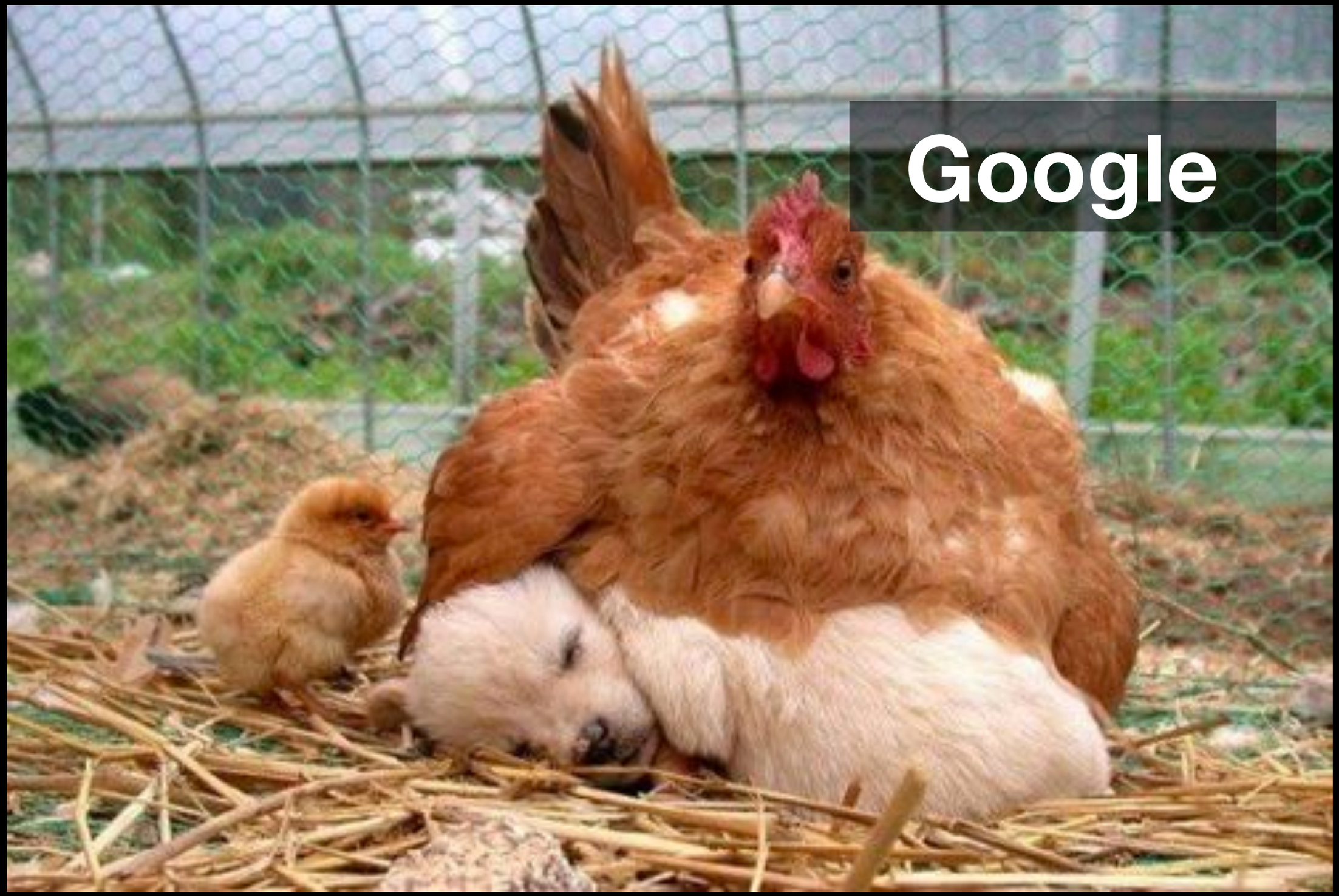
Follow +AngularJS on

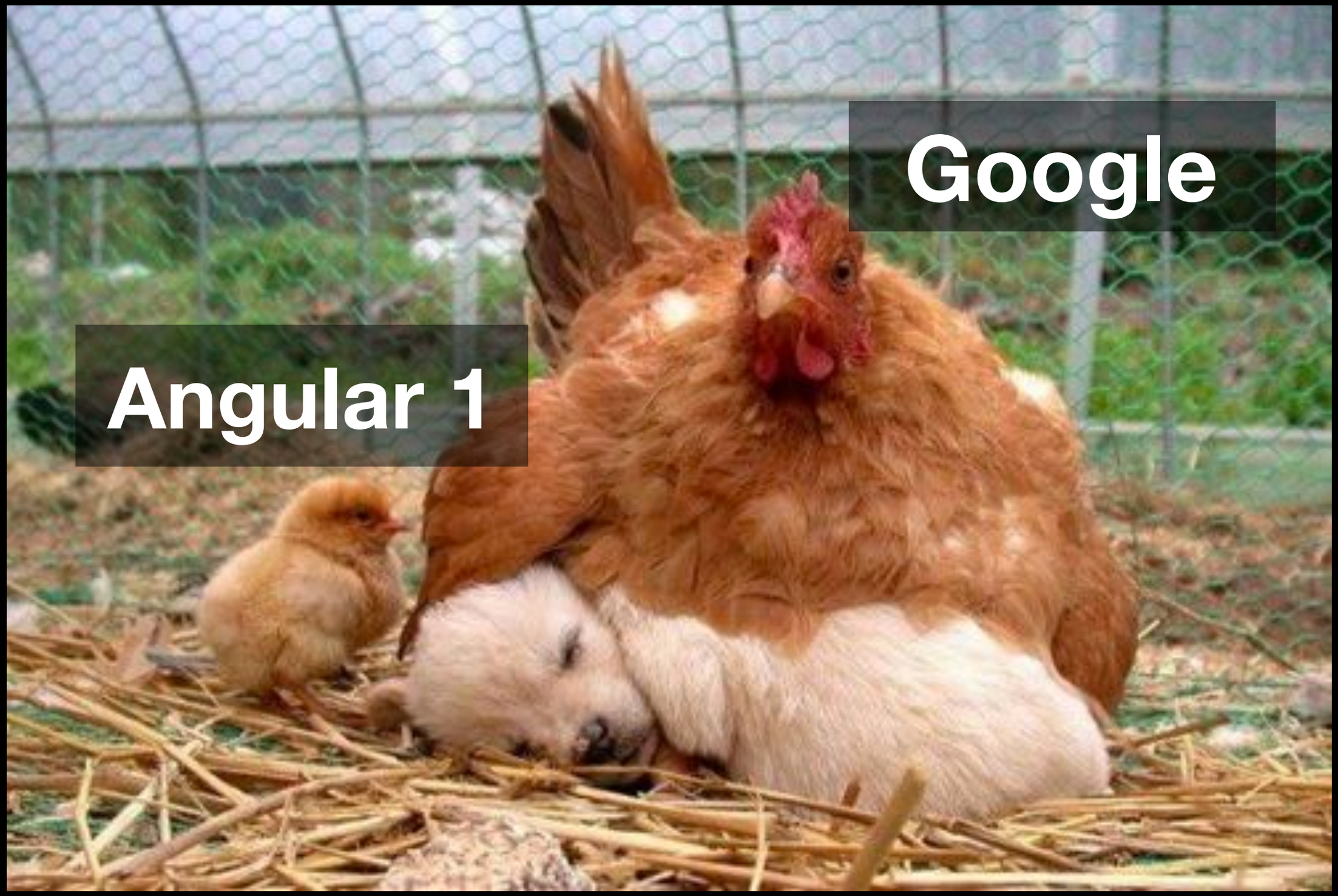
Follow @angularjs 93.7K followers

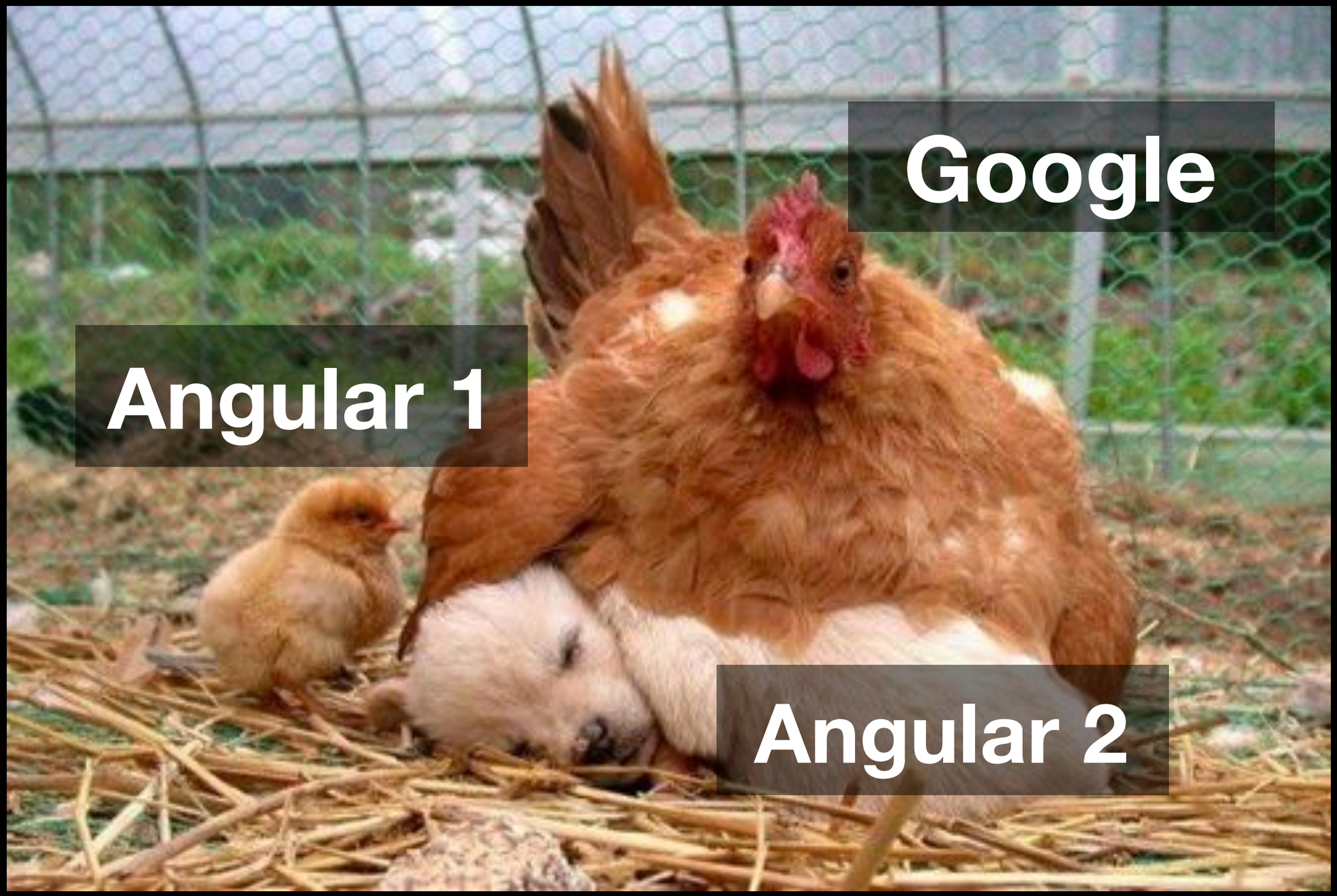
Tweet 4,942

Learn Angular in your browser for free!

Angular is a development platform for building mobile and desktop applications


Google


Google
Angular 1


Google
Angular 1
Angular 2

Angular 2 is...

rewritten from scratch

...developed in a

different language

...completely

incompatible API

...brand

new building blocks

Why?

Web have moved forward

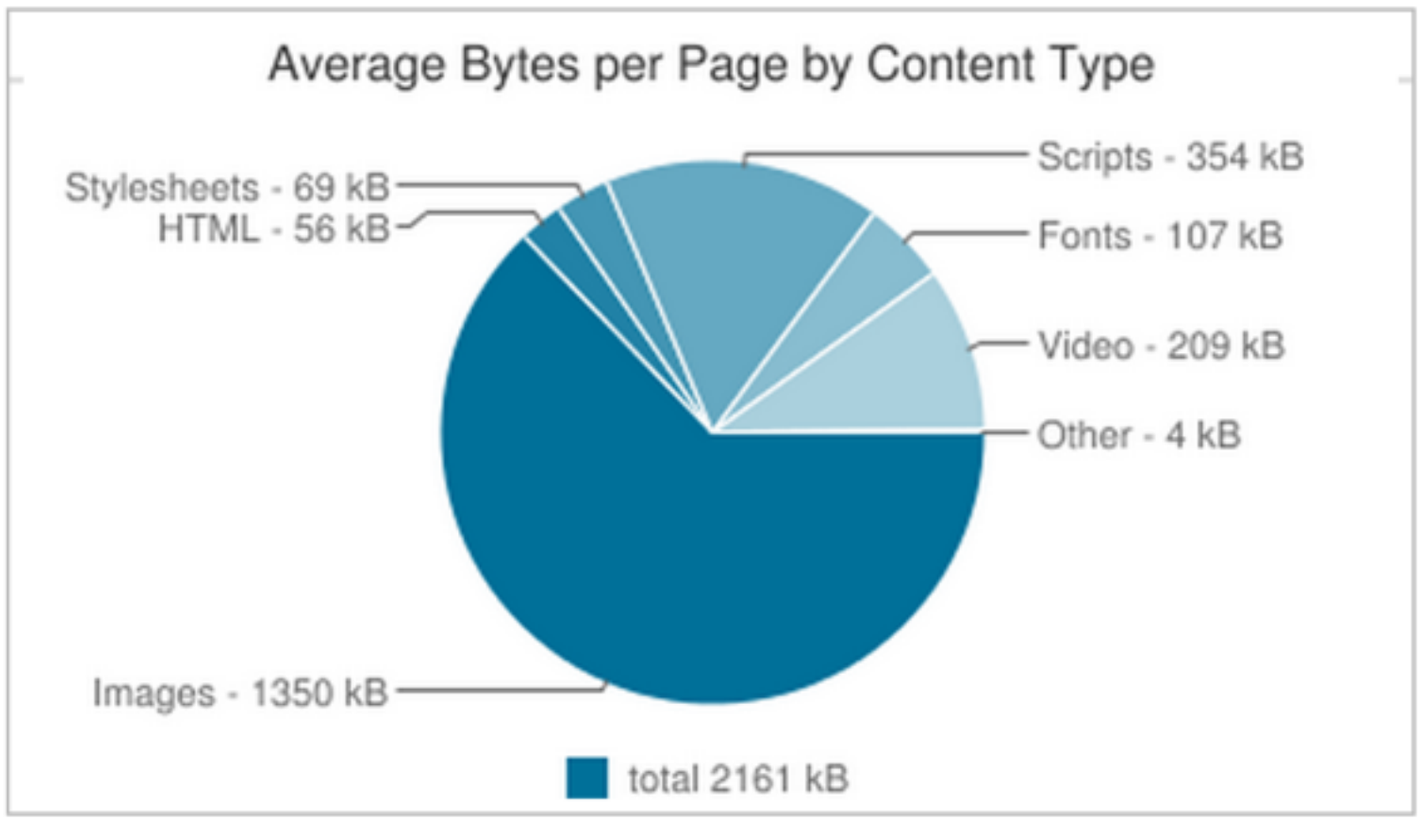
WebWorkers

Reactive Extensions (Rx)

Interesting stats

Got a stat you'd like to see? Suggest it! New feature: Compare two runs

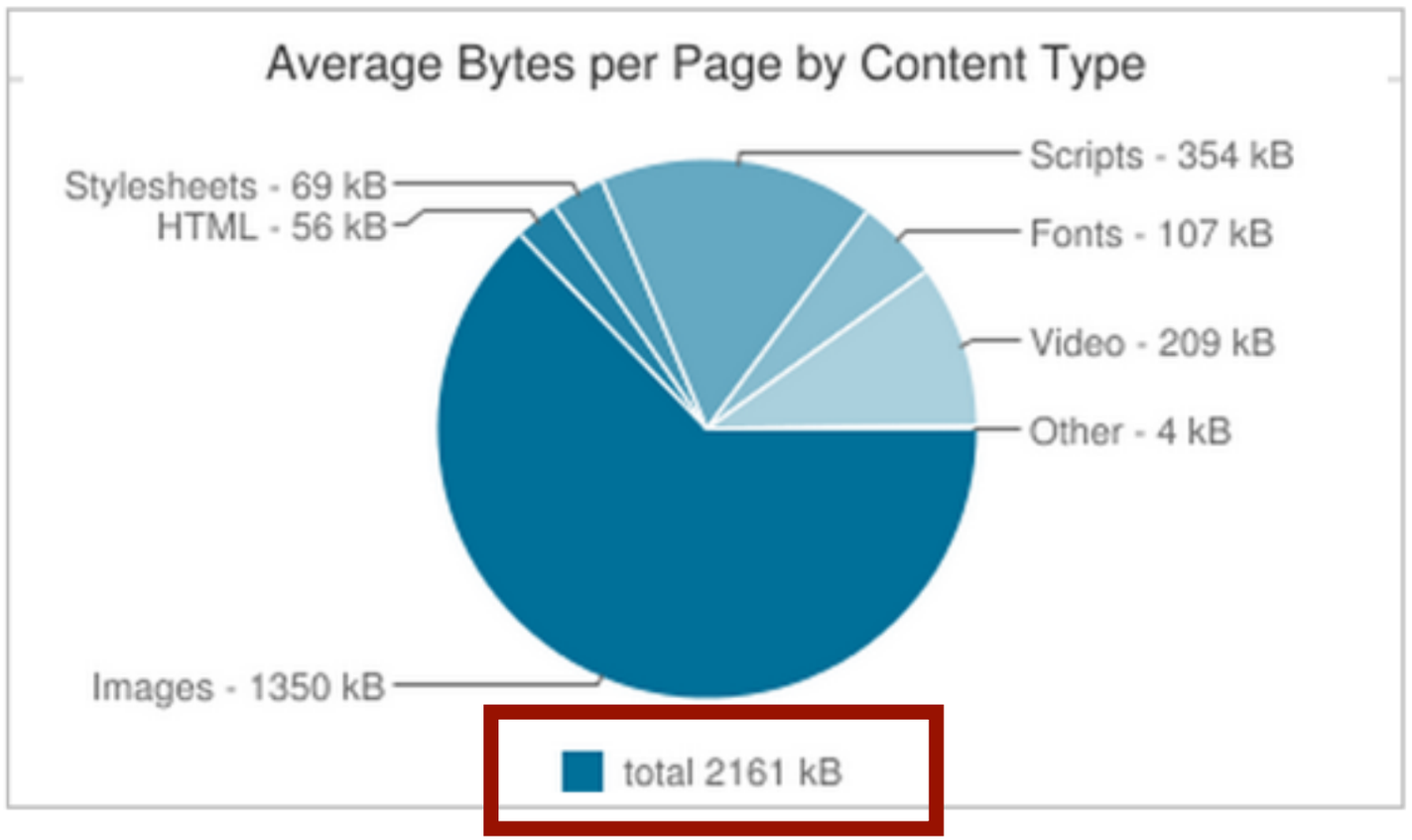
Choose a run: Aug 1 2015 Choose URLs: All



Interesting stats

Got a stat you'd like to see? Suggest it! New feature: Compare two runs

Choose a run: Aug 1 2015 Choose URLs: All

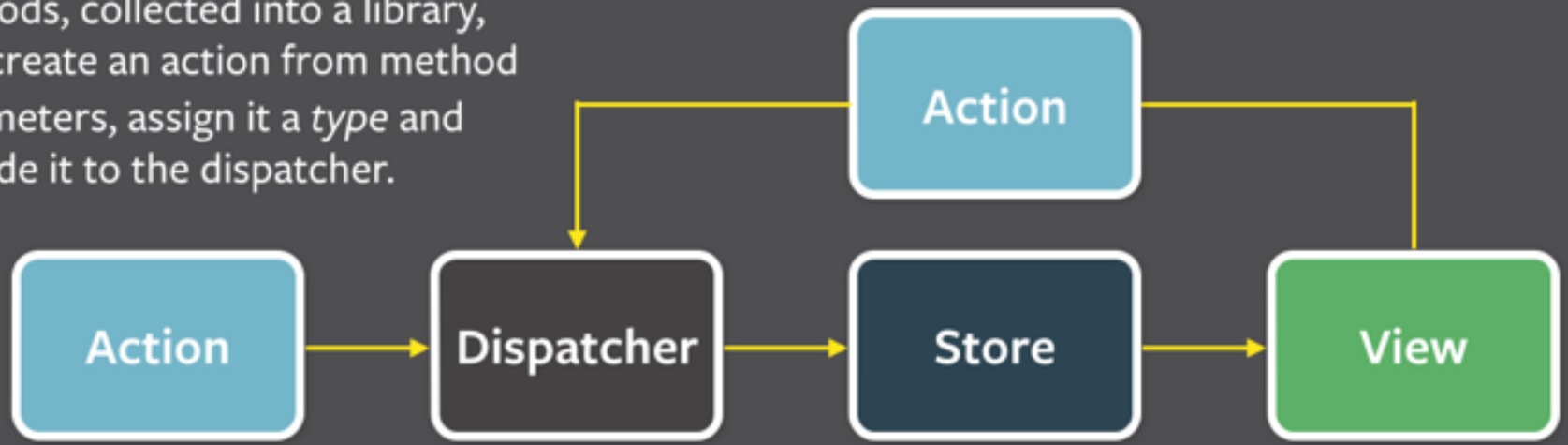


Learnt lessons


Mutable state
4GIFs.com


Tangled data-flow

Action creators are helper methods, collected into a library, that create an action from method parameters, assign it a *type* and provide it to the dispatcher.



Every action is sent to all stores via the *callbacks* the stores register with the dispatcher.

After stores update themselves in response to an action, they emit a *change* event.

Special views called *controller-views*, listen for *change* events, retrieve the new data from the stores and provide the new data to the entire tree of their child views.

…we will make the
transition process boring…

Pascal Precht ʕ◔ϖ◔ʔ
@PascalPrecht

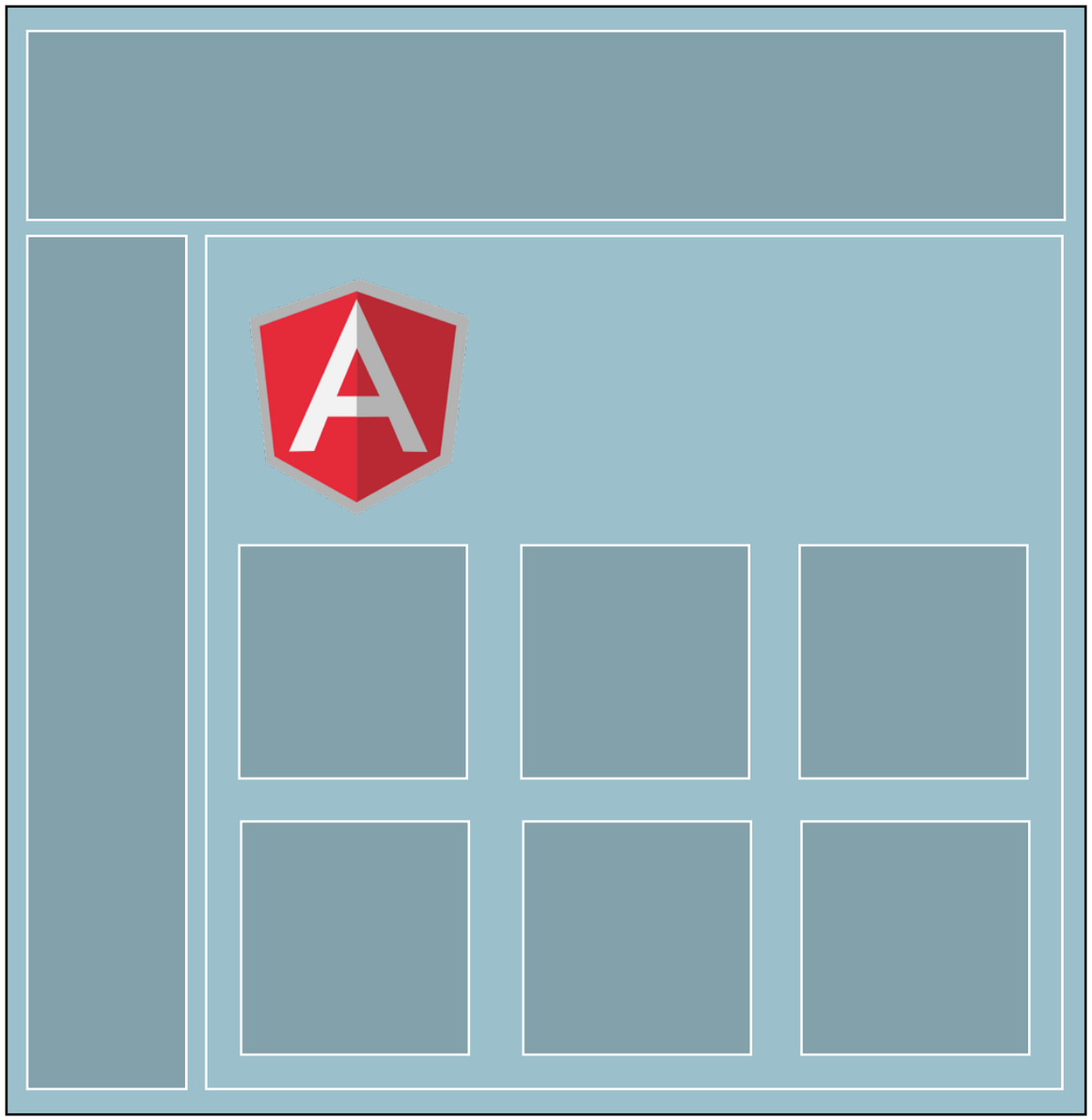
Remember when @mhevery said at @angularu that upgrading to Angular 2 will be boring? Well.. Lemme tell you sth. He didn't lie.

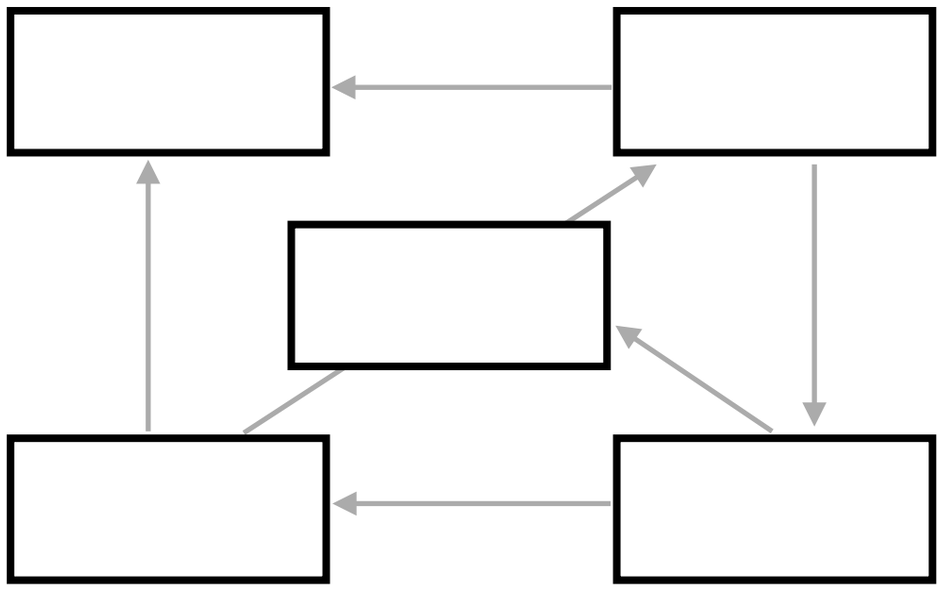
ngUpgrade

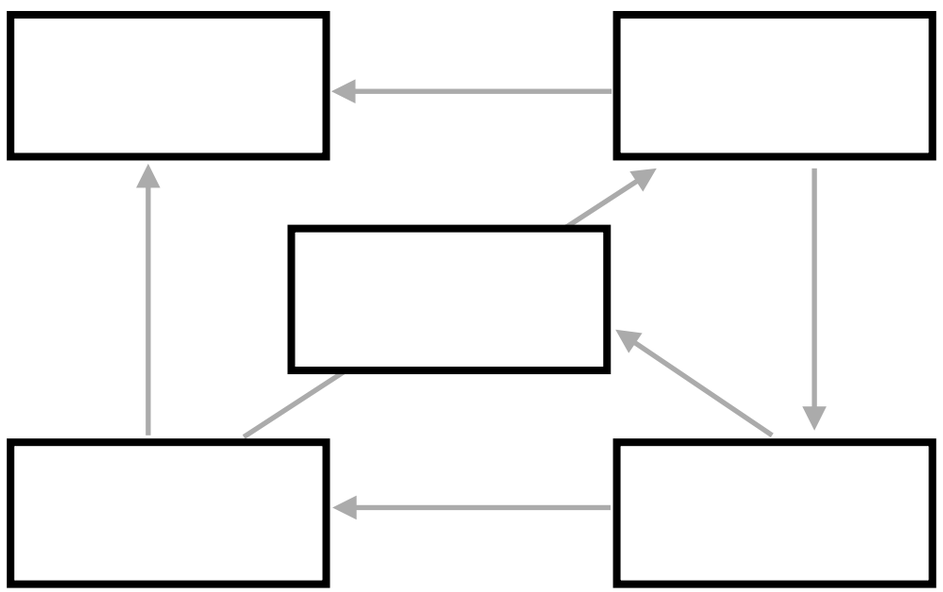
How to migrate to Angular 2?

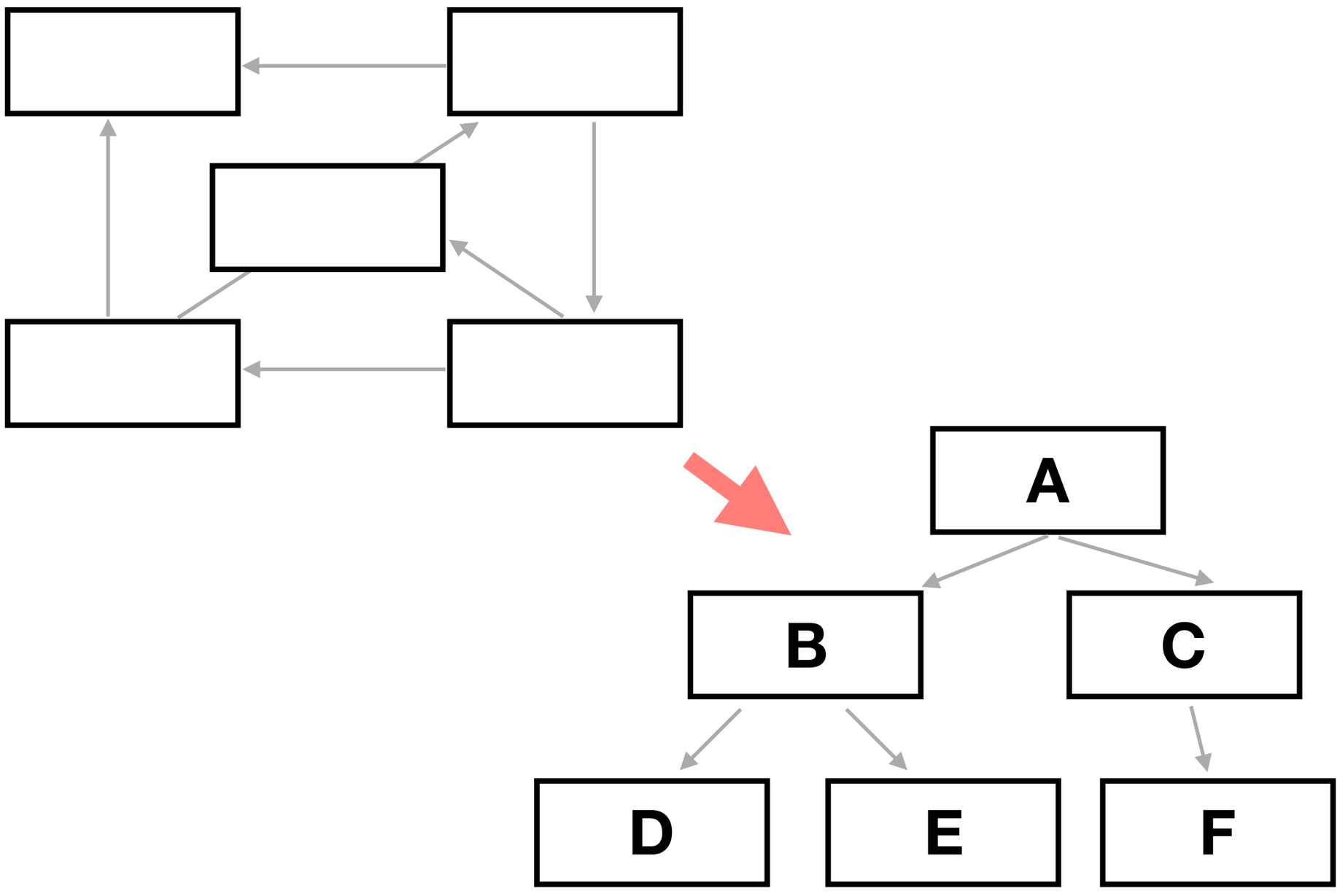
(2 easy steps)

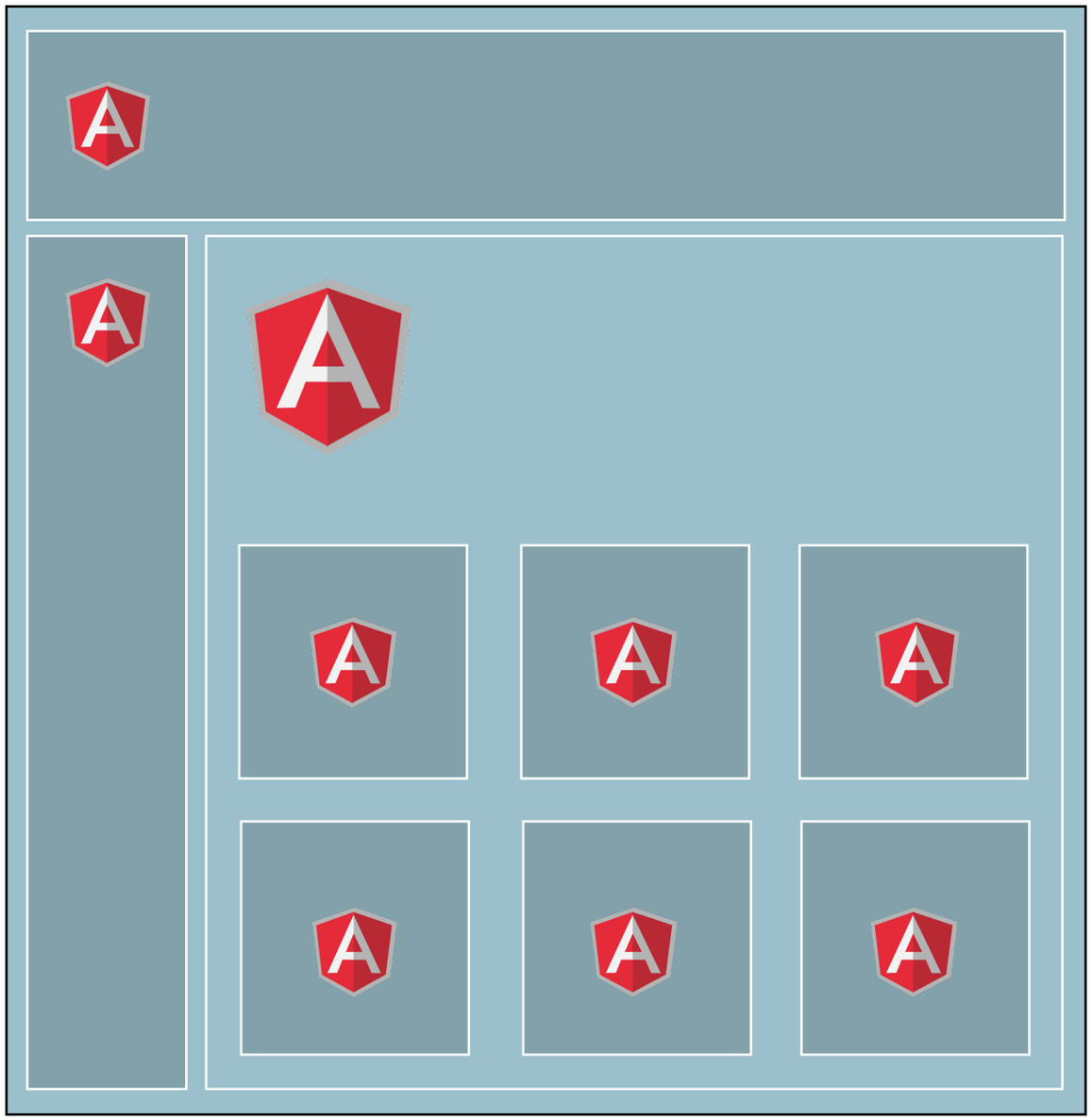
Step 1



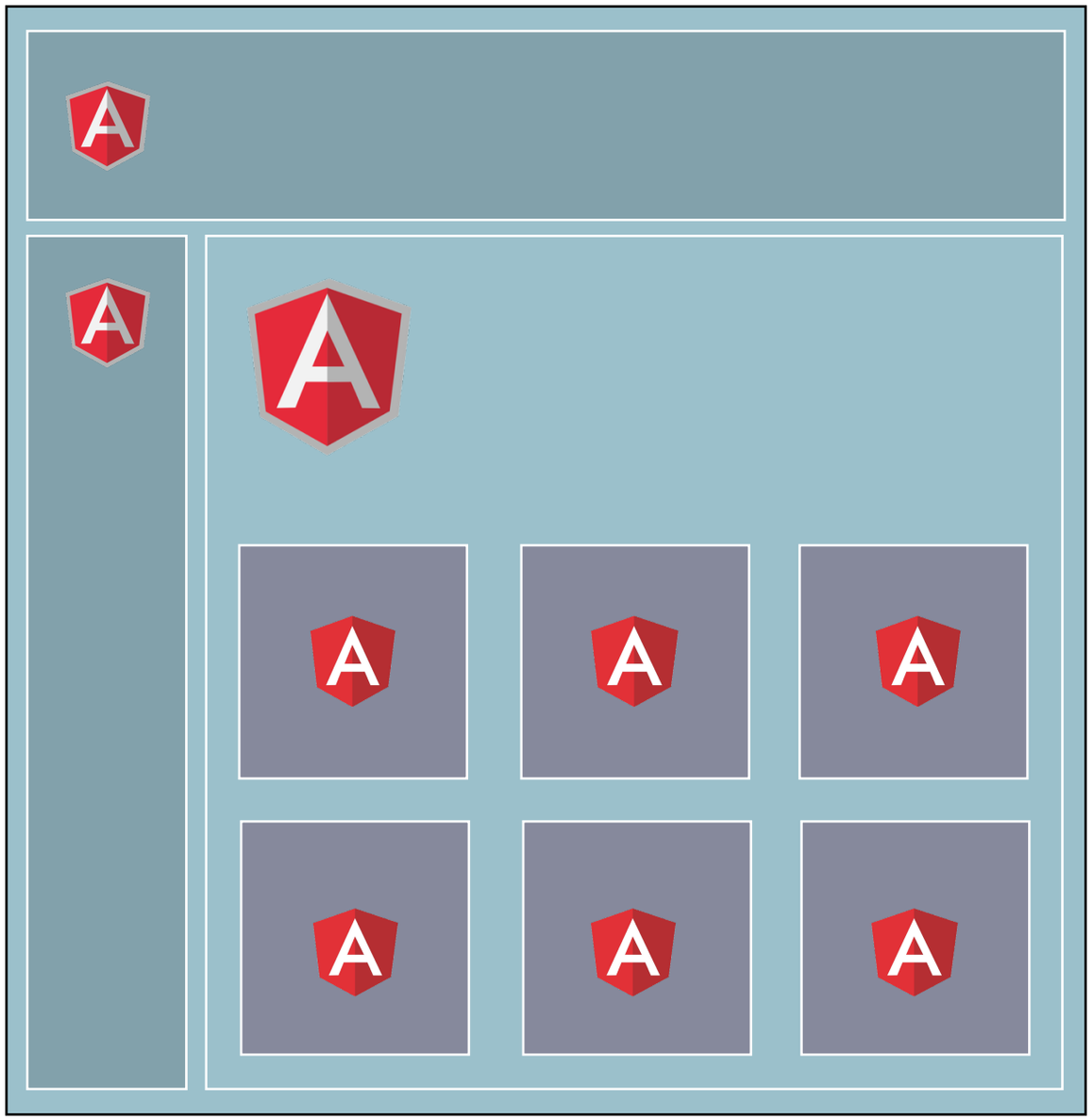


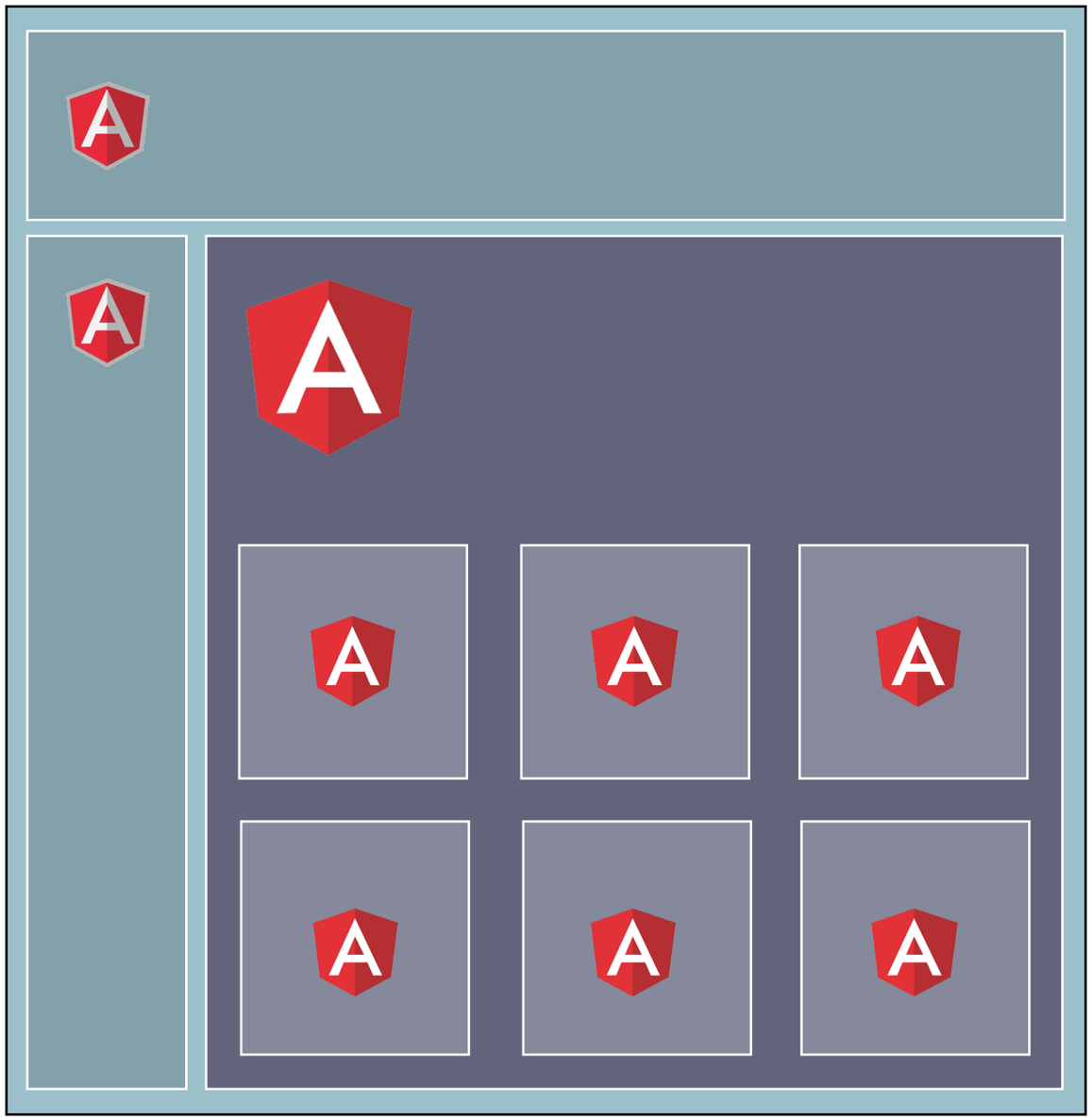


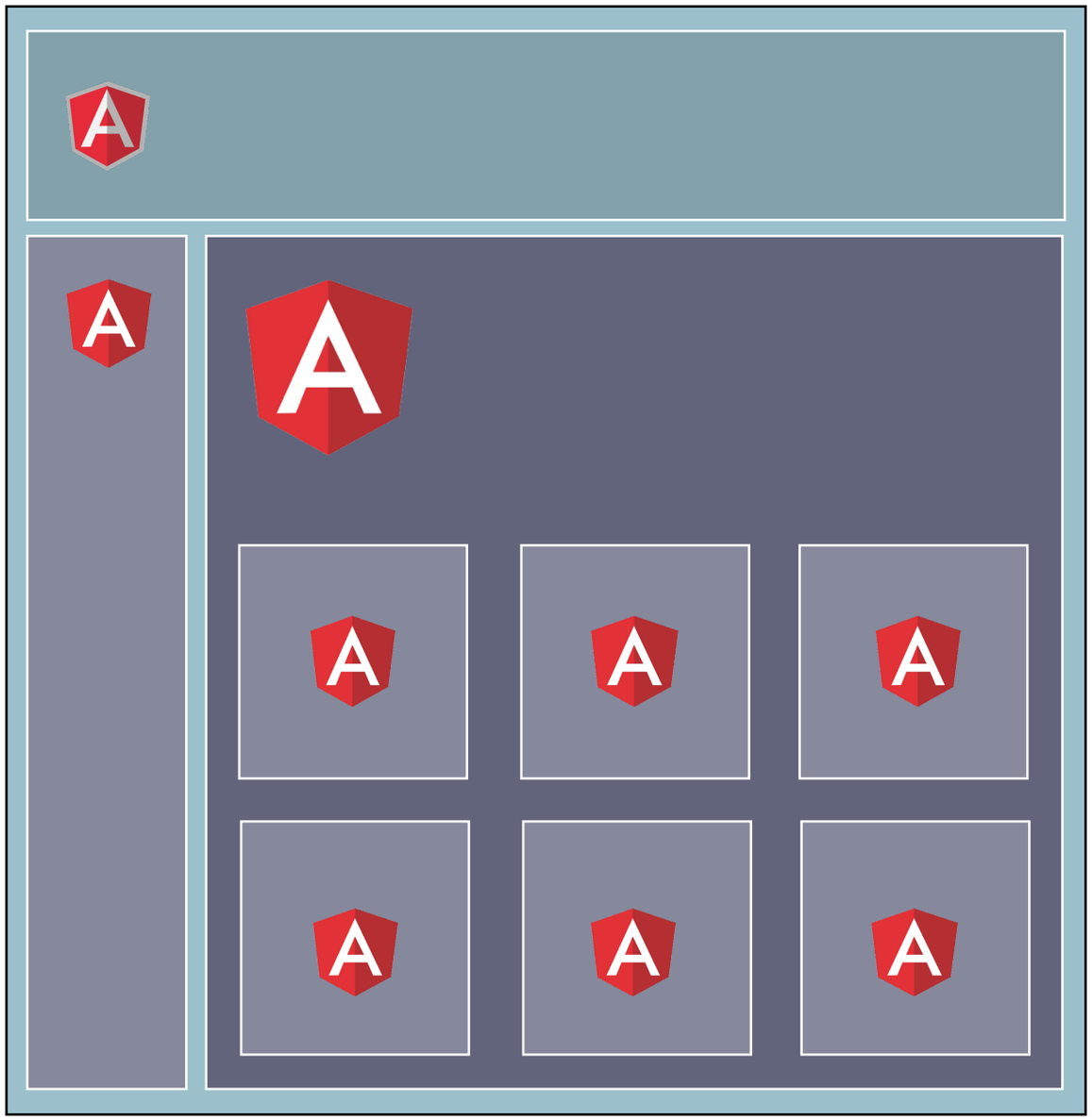

A
B
C
D
E
F

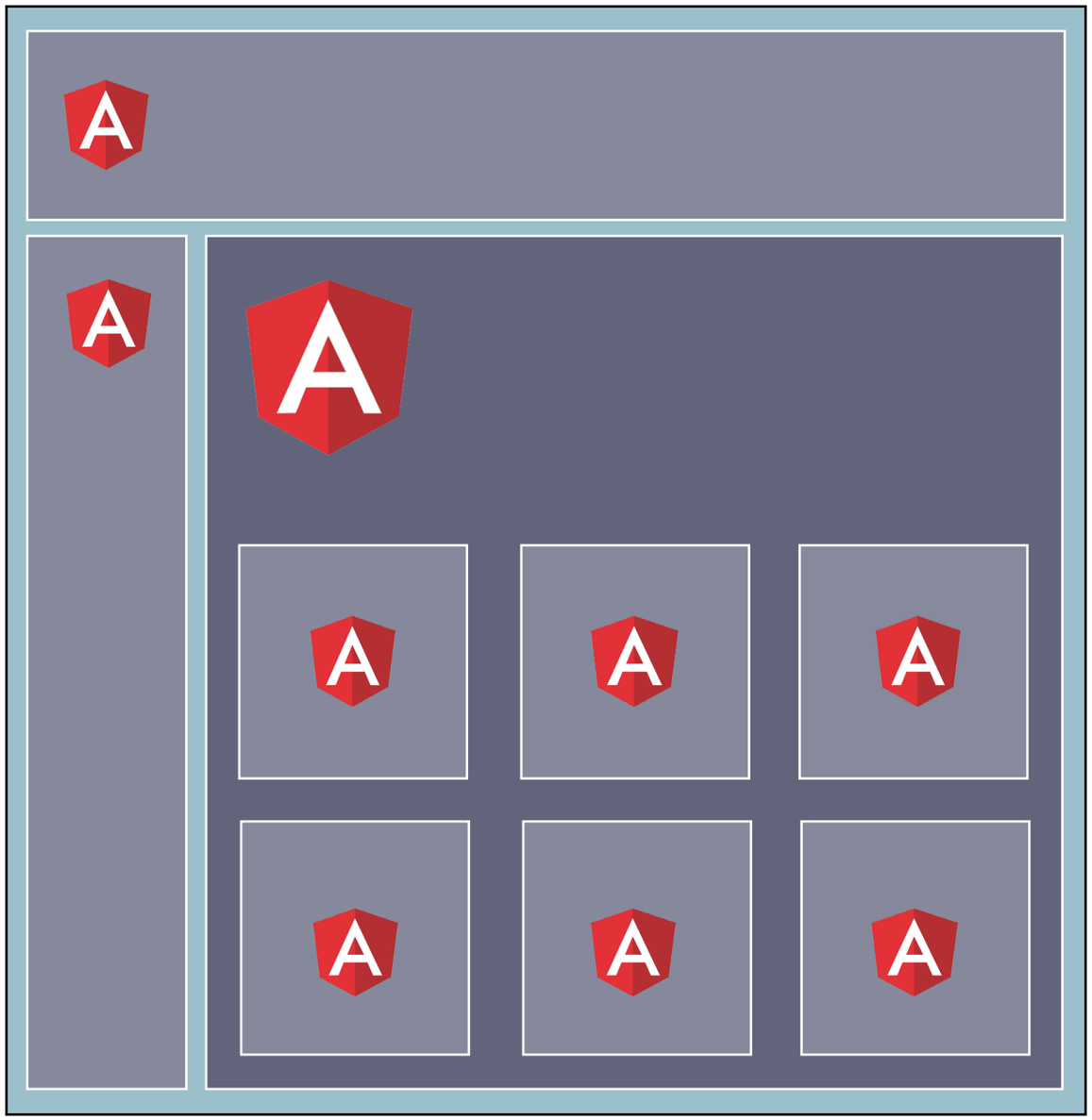


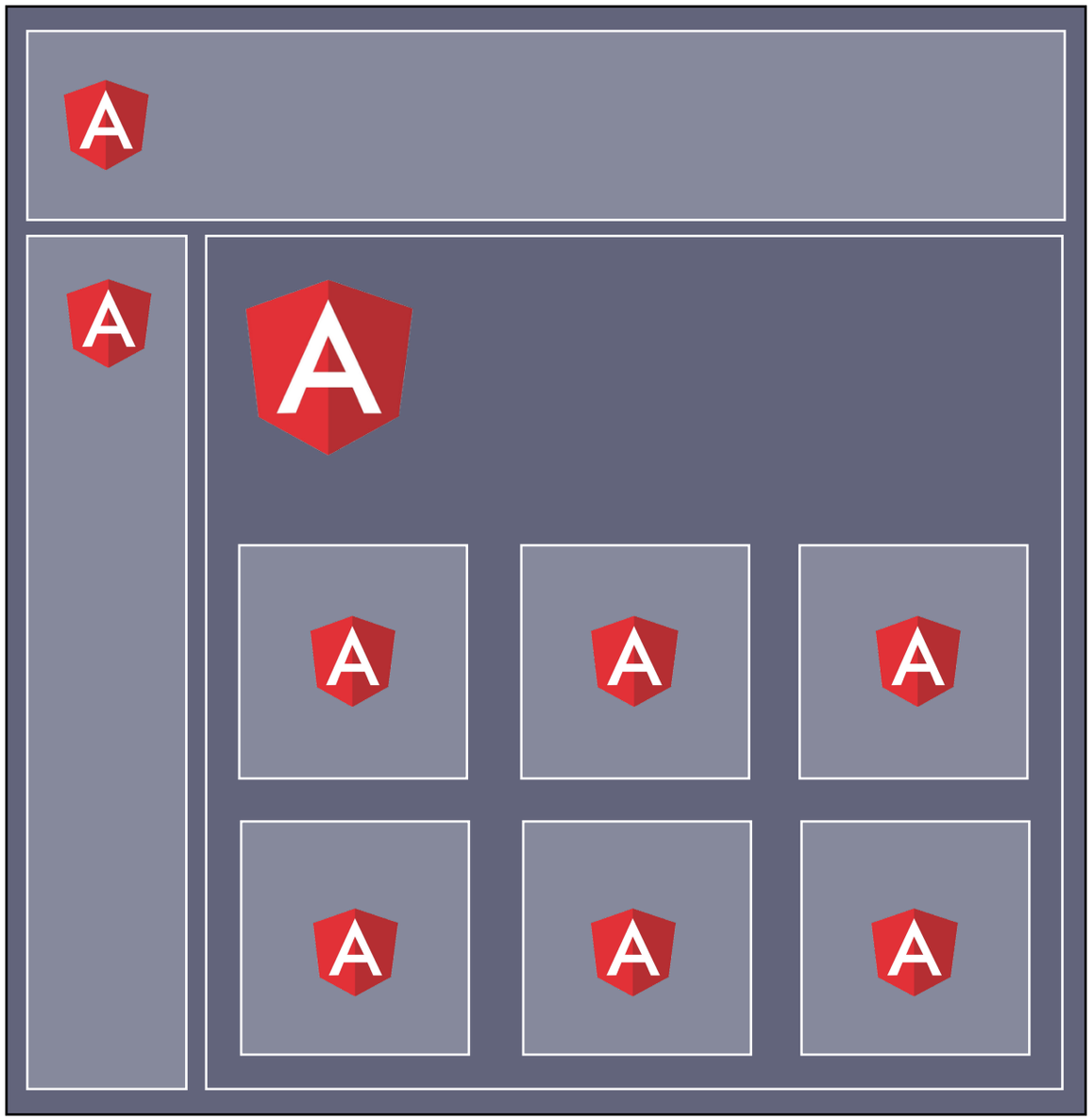
Step 2

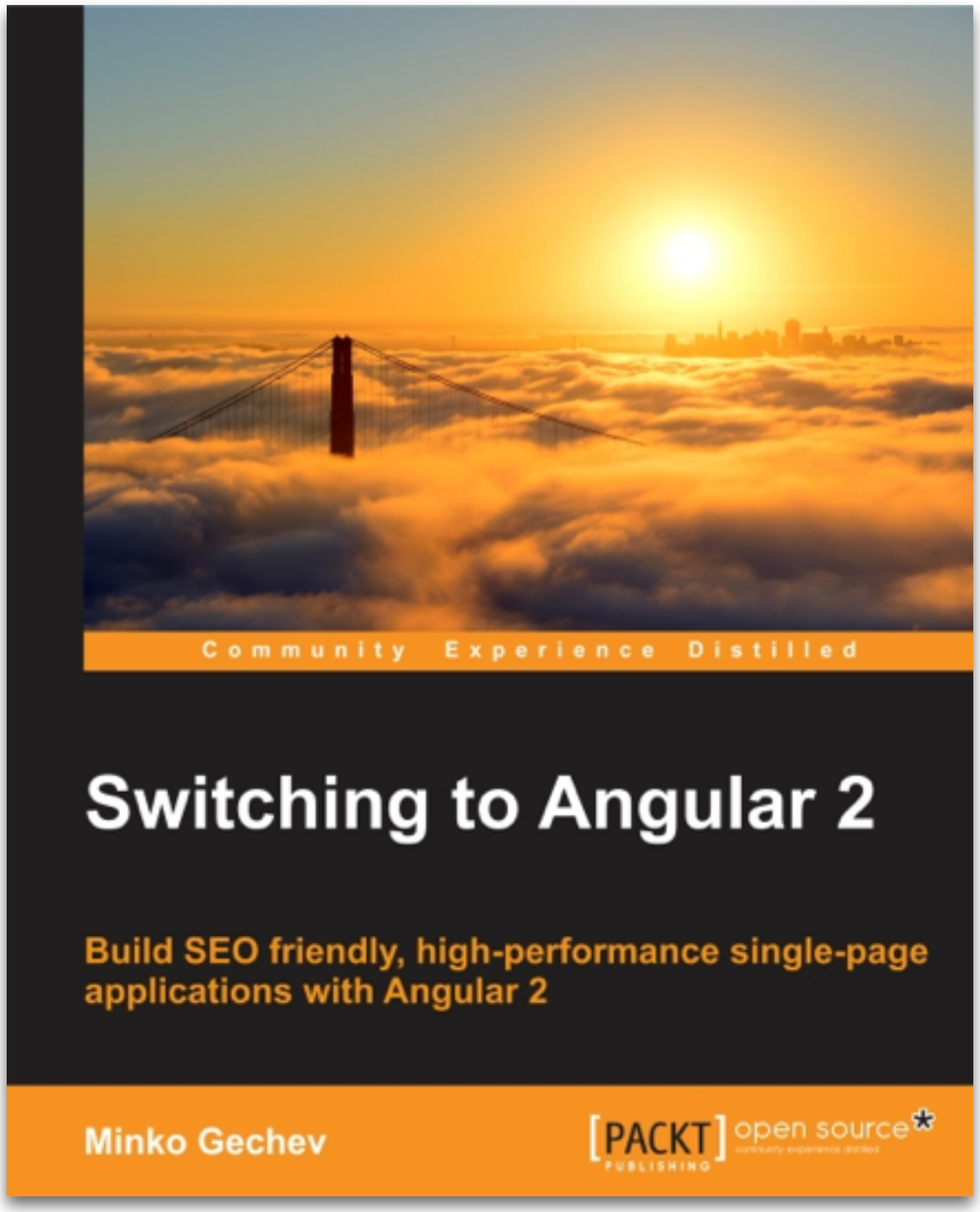











Community Experience Distilled
Switching to Angular 2
Build SEO friendly, high-performance single-page applications with Angular 2
Minko Gechev
[PACKT] open source
PUBLISHING

Thank you!

github.com/mgechev
twitter.com/mgechev
blog.mgechev.com