ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Angular Basics & A first Component

Alex Thalhammer

Outline

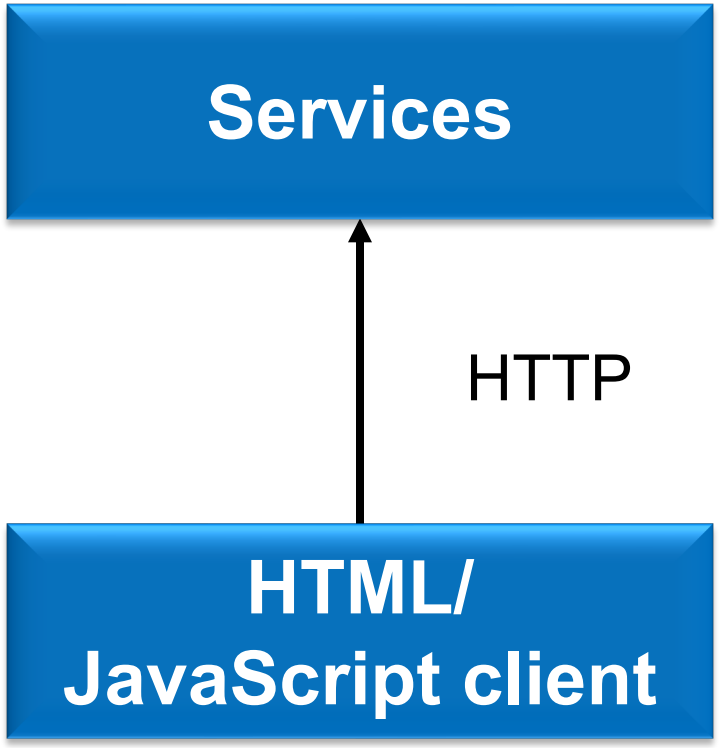
- Motivation
- First steps
- HTTP access with Angular HTTP client
- Your first Component
- Built-in Angular Directives

Motivation

Platforms and Usability

HTML + JavaScript

Single Page Application (SPA)



HTML + JavaScript = Complexity

Frameworks make SPA manageable

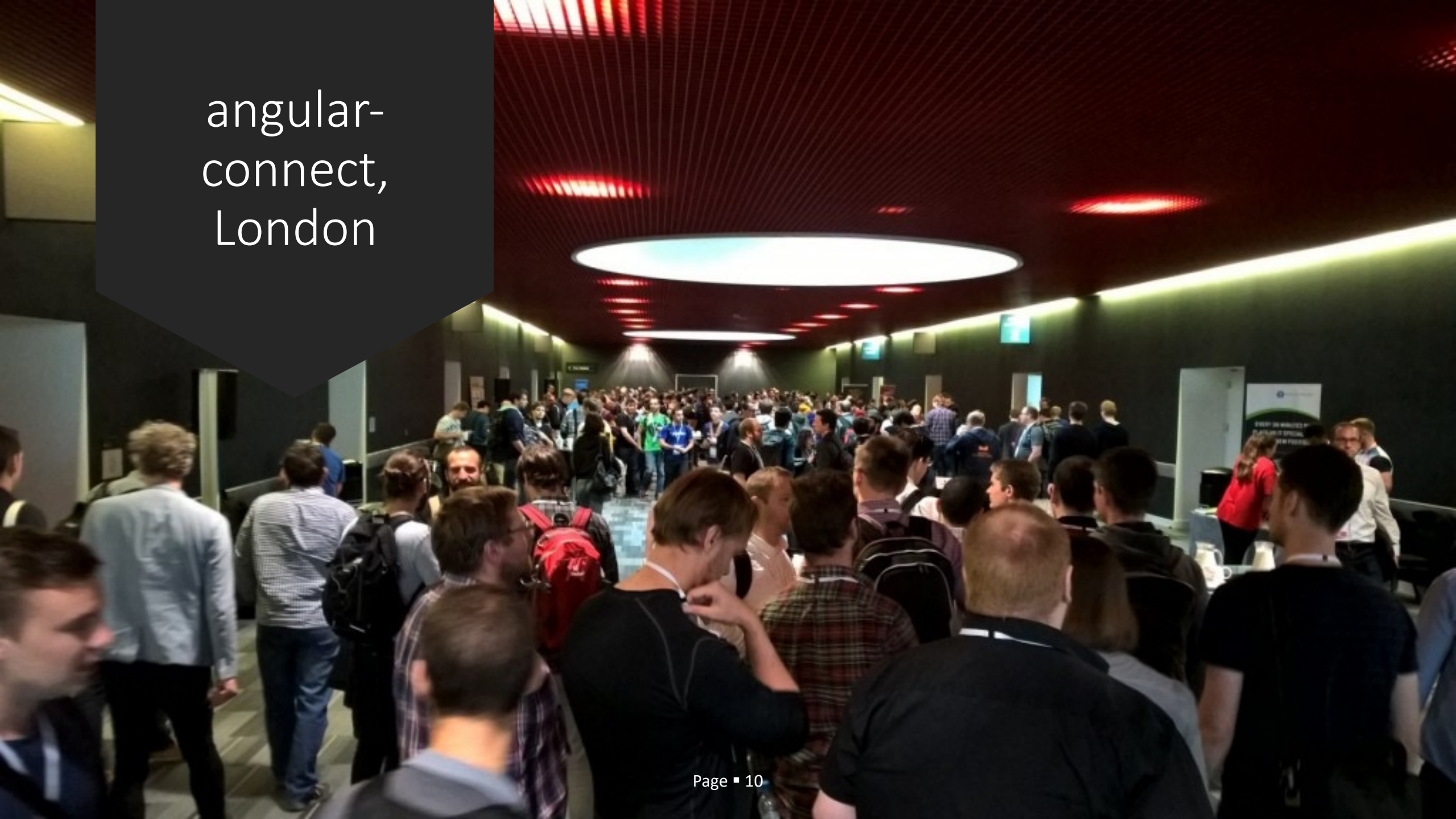
Google

Community

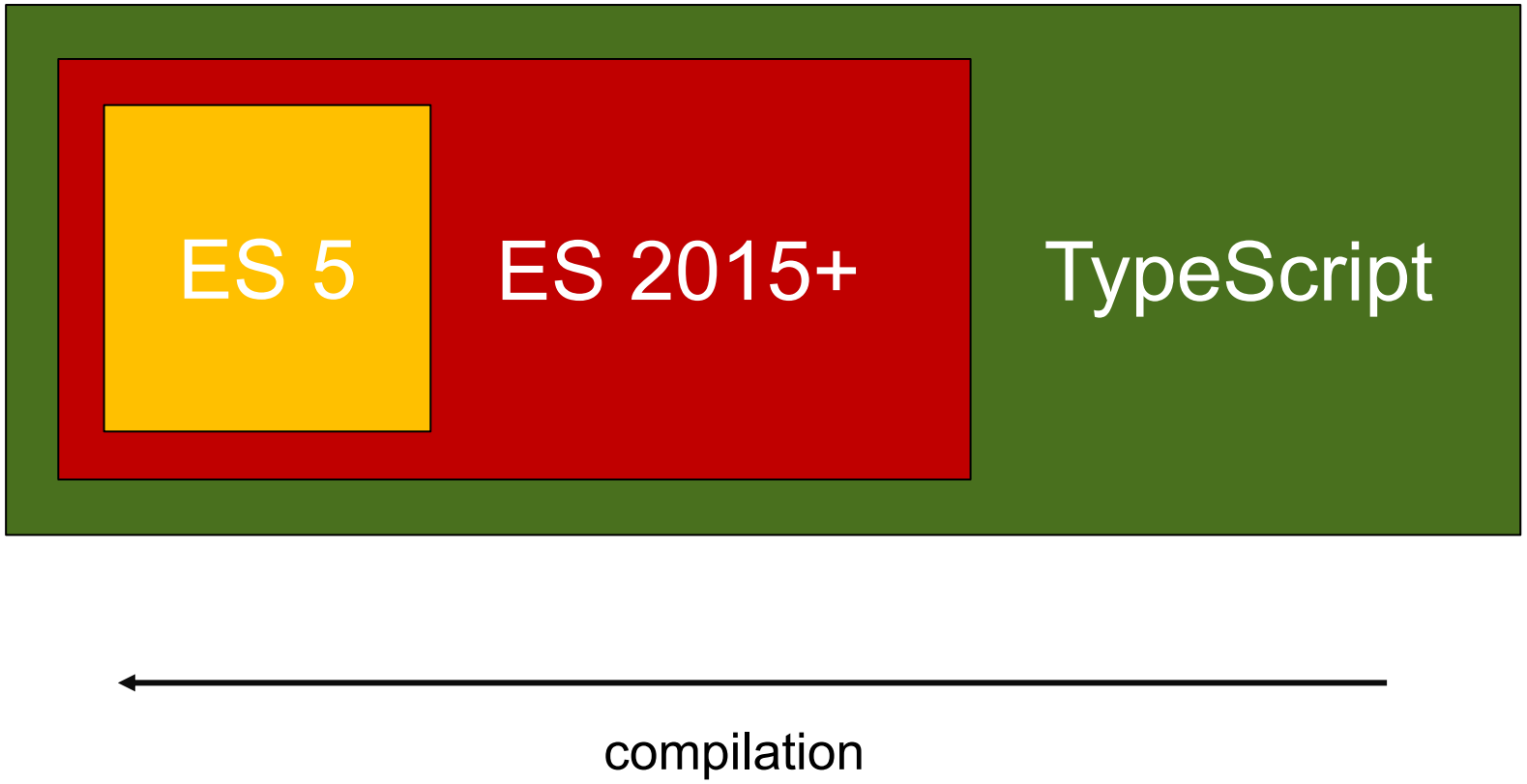
Angular
>2M Devs

angular-connect, London

angular-connect, London

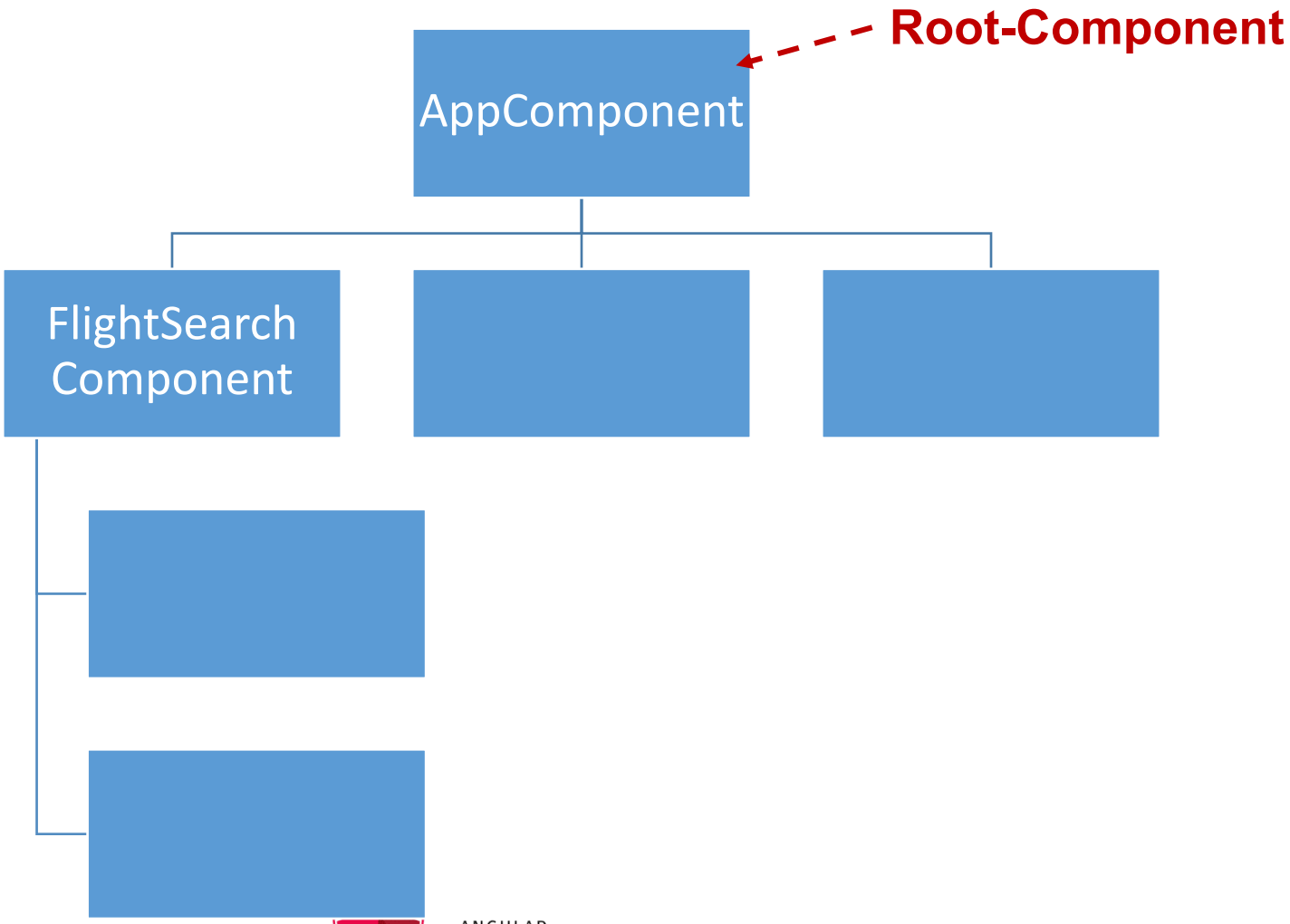


JavaScript vs TypeScript



First steps with Angular

App == component tree



Angular Dev Tools

- https://chrome.google.com/webstore/detail/angular-devtools/ienfalfjdbdpebioblfackkekamfmbnh

DEMO

AppComponent

```
@Component({
    selector: 'flight-app',
    templateUrl: './app.component.html'
})
export class AppComponent {
    title = 'Hello World!';
}
```

AppComponent

```
import { Component } from '@angular/core';

@Component({
    selector: 'flight-app',
    templateUrl: './app.component.html'
})
export class AppComponent {
    title = 'Hello World!';
}
```

Library
E.g.: @angular/core

Own project
E.g.: ../entities/flight
No ending ".ts"

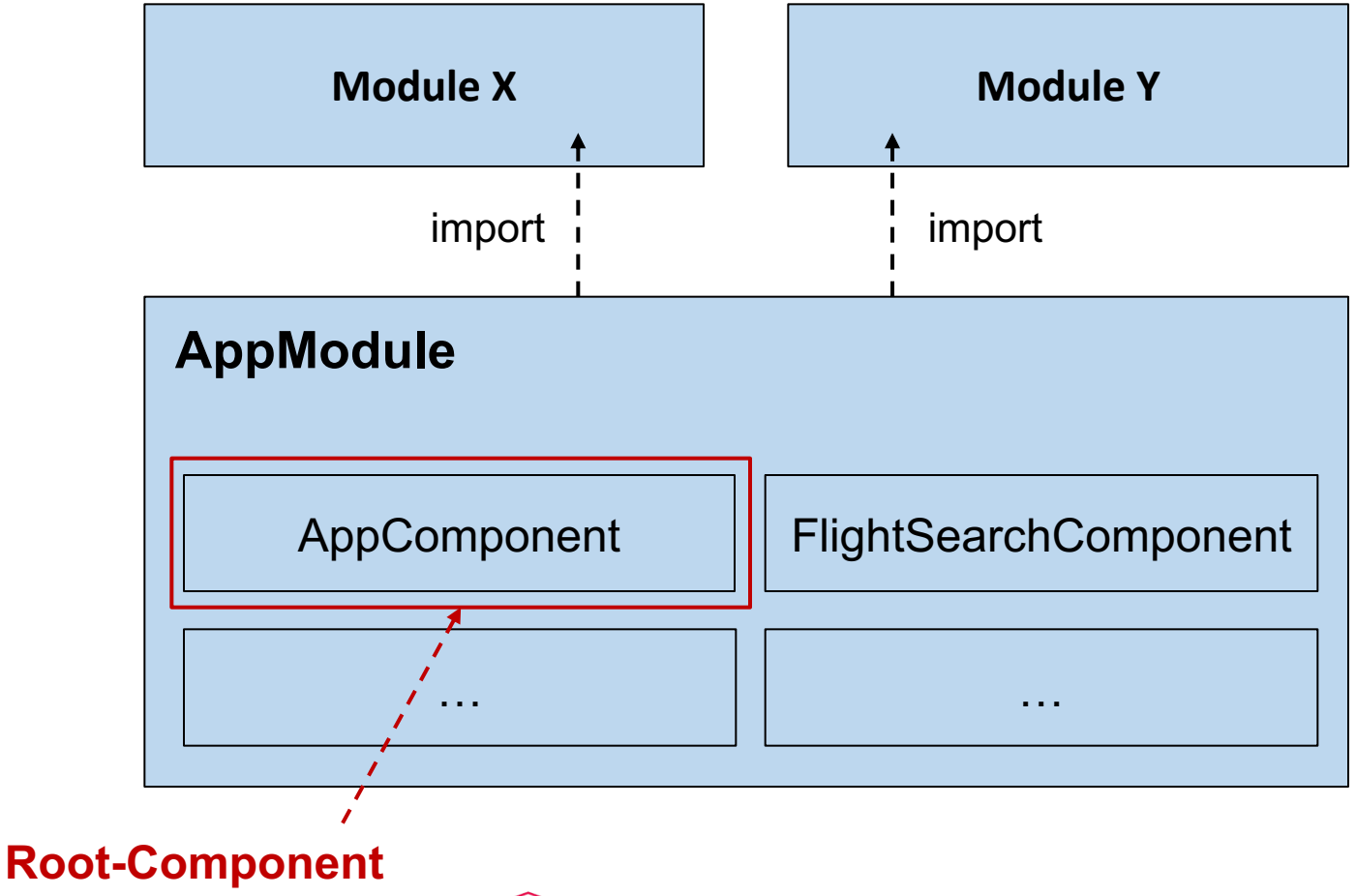
AppComponent

```
import { Component } from '@angular/core';

@Component({
    selector: 'flight-app',
    templateUrl: './app.component.html'
})
export class AppComponent {
    title = 'Hello World!';
}
```

```
<h1>{{title}}</h1>
<div class="container">
    <flight-search></flight-search>
</div>
```

Module



AppModule

```
@NgModule({
    imports: [
        BrowserModule, HttpClientModule, FormsModule
    ],
    declarations: [
        AppComponent, FlightSearchComponent
    ],
    bootstrap: [
        AppComponent
    ]
})
export class AppModule {
}
```

Bootstrapping

- Starting Angular
- Loading
  - RootModule/AppModule with
    - RootComponent/AppComponent

Bootstrapping

```
platformBrowserDynamic().bootstrapModule(AppModule);
```

index.html

```
[…]

<body>

  <flight-app></flight-app>

  <script src="..."></script>

</body>

[…]
```

A new Angular project

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

Angular CLI

A command line interface for Angular

GET STARTED

Angular CLI

Our Starterkit

- ng new starter
- cd starter
- npm i bootstrap –save
- Adding global styles in *angular.json*

```
[…]
"styles": [
    "styles.css",
    "../node_modules/bootstrap/dist/css/bootstrap.css",
   […]
],
[…]
```

DEMO

Let’s get it on

- Pull the repo https://github.com/L-X-T/mtrail-essentials/
- Please get started with lab 00_getting_started
- Yarn (or npm i) and then Yarn start (or npm start)
- Take a closer look at the starter kit
- VS Code: You may add the plugins mentioned

Your first component

Component as TypeScript class

```
@Component({
    selector: 'flight-search',
    templateUrl: './flight-search.html'
})
export class FlightSearchComponent {

    from: string;
    to: string;
    flights: Flight[];

    search(): void { [...] }
    select(flight: Flight): void { [...] }
}
```

Template

Two Way Binding

```
<input [(ngModel)]="from">
<input [(ngModel)]="to">

<button [disabled]="!from || !to" (click)="search()">
    Search
</button>

<table>
    <tr *ngFor="let flight of flights">
        <td>{{flight.id}}</td>
        <td>{{flight.date}}</td>
        <td>{{flight.from}}</td>
        <td>{{flight.to}}</td>
    </tr>
</table>
```

Event (/Output) Binding

Property (/Input) Binding

Template

DEMO

Access HTTP ressources

HttpClient

- get(url, options)
- post (url, body, options)
- put(url, body, options)
- delete(url, options)
- …

HttpClient

- get<T>(url, options)
- post<T>(url, body, options)
- put<T>(url, body, options)
- delete<T>(url, options)
- …

Inject HttpClient

```
@Component({
    selector: 'flight-search',
    templateUrl: './flight-search.html'
})
export class FlightSearchComponent {

    from: string;
    to: string;
    flights: Flight[];

    constructor(private http: HttpClient) { […] }

    search(): void { [...] }
    select(flight: Flight): void { [...] }
}
```

Use HttpClient (I)

```
const url = 'http://www.angular.at/api/flight';
```

Use HttpClient (II)

```
const url = 'http://www.angular.at/api/flight';

const params = new HttpParams().set('from', this.from).set('to', this.to);
```

Use HttpClient (III)

```
const url = 'http://www.angular.at/api/flight';

const params = new HttpParams().set('from', this.from).set('to', this.to);

const headers = new HttpHeaders().set('Accept', 'application/json');
```

Use HttpClient (IV)

```
const url = 'http://www.angular.at/api/flight';

const params = new HttpParams().set('from', this.from).set('to', this.to);

const headers = new HttpHeaders().set('Accept', 'application/json');

this.http.get<Flight[]>(url, { params: params, headers: headers })
```

Use HttpClient (V)

```
const url = 'http://www.angular.at/api/flight';

const params = new HttpParams().set('from', this.from).set('to', this.to);

const headers = new HttpHeaders().set('Accept', 'application/json');

this.http.get<Flight[]>(url, { params, headers })
```

Short hand

Use HttpClient (VI)

```
const url = 'http://www.angular.at/api/flight';

const params = new HttpParams().set('from', this.from).set('to', this.to);

const headers = new HttpHeaders().set('Accept', 'application/json');

this.http.get<Flight[]>(url, { params, headers })
    .subscribe(
        function(flights) { […] }
    );
```

Use HttpClient (VII)

```
const url = 'http://www.angular.at/api/flight';

const params = new HttpParams().set('from', this.from).set('to', this.to);

const headers = new HttpHeaders().set('Accept', 'application/json');

const that = this;
this.http.get<Flight[]>(url, { params, headers })
    .subscribe(
        function(flights) {
              that.flights = flights;
        }
);
```

Use HttpClient (VIII)

```
const url = 'http://www.angular.at/api/flight';

const params = new HttpParams().set('from', this.from).set('to', this.to);

const headers = new HttpHeaders().set('Accept', 'application/json');

this.http.get<Flight[]>(url, { params, headers })
    .subscribe(
        (flights) => {
              this.flights = flights;
        }
    );
```

Use HttpClient (IX)

```
const url = 'http://www.angular.at/api/flight';

const params = new HttpParams().set('from', this.from).set('to', this.to);

const headers = new HttpHeaders().set('Accept', 'application/json');

this.http.get<Flight[]>(url, { params, headers })
    .subscribe({
        next: (flights) => { this.flights = flights; },
        error: (err) => { console.error('Error loading', err); }
    });
```

DEMO

LAB

Use HttpClient (X)

```
const url = 'http://www.angular.at/api/flight';

const params = new HttpParams().set('from', this.from).set('to', this.to);

const headers = new HttpHeaders().set('Accept', 'application/json');

this.http.get<Flight[]>(url, { params, headers })
    .subscribe({
        next: (flights) => { this.flights = flights; },
        error: (err) => { console.error('Error loading', err); }
    });
```

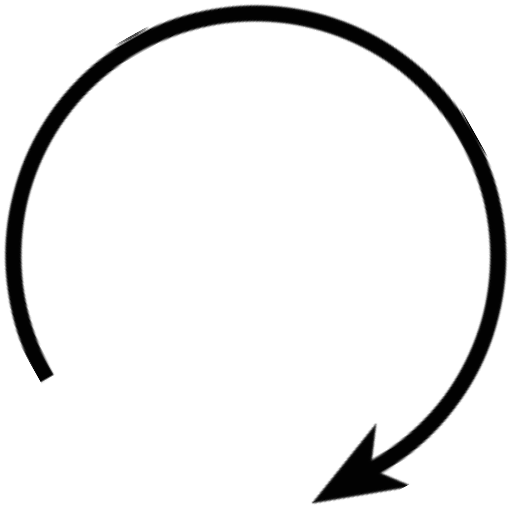
Observable

Observable
„Source“

Operator
(z. B. map)

Observer
„Target“

Observable



Observable

```
.subscribe({
    next: (result) => { … },
    error: (error) => { … },
    complete: () => { … }
});
```

Observer

Use Angular Directives

What are Directives?

- Add behaviour to html elements
- Are used as html attributes

- Examples:
  - <input **[(ngModel)]**="from">
  - <div ***ngFor**="let flight of flights">…

Types of Directives?

- Structural Directives
  - *ngIf="statement"
  - *ngFor="let element of array"
  - *ngSwitch="something"

- Attribute directives
  - Built-ins
    - [(ngModel)]
    - [ngClass] or [ngStyle]
  - Custom ones

Examples (I)

```
<tr *ngFor="let flight of flights">
    <td>{{flight.id}}</td>
</tr>
```

```
<table *ngIf="flights.length > 0">
…
</table>
```

```
<tr [ngClass]="{ 'active': flight === selectedFlight }">
…
</tr>
```

```
<tr [ngStyle]="{ 'background-color':
     (flight === selectedFlight) ?
     'orange' : 'blue' }">
</tr>
```

Examples (II)

```
<tr *ngFor="let flight of flights">
    <td>{{flight.id}}</td>
</tr>
```

```
<table *ngIf="flights.length > 0">
…
</table>
```

```
<tr [ngClass]="{ 'active': flight === selectedFlight }">
…
</tr>
```

```
<tr [class.active]="flight === selectedFlight">
</tr>
```

```
<tr [ngStyle]="{ 'background-color':
     (flight === selectedFlight) ?
     'orange' : 'blue' }">
</tr>
```

DEMO