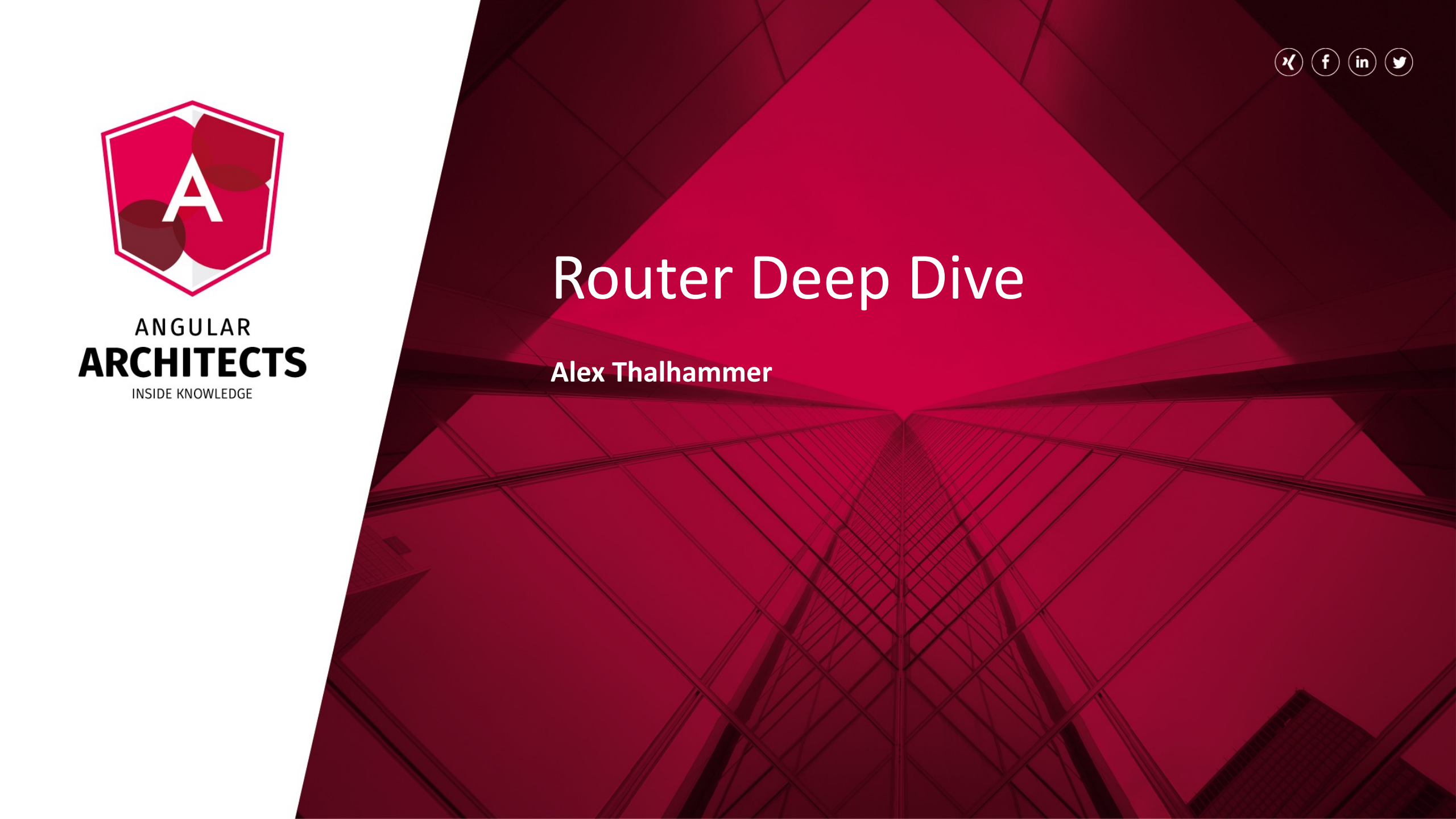

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE
Router Deep Dive
Alex Thalhammer

Motivation

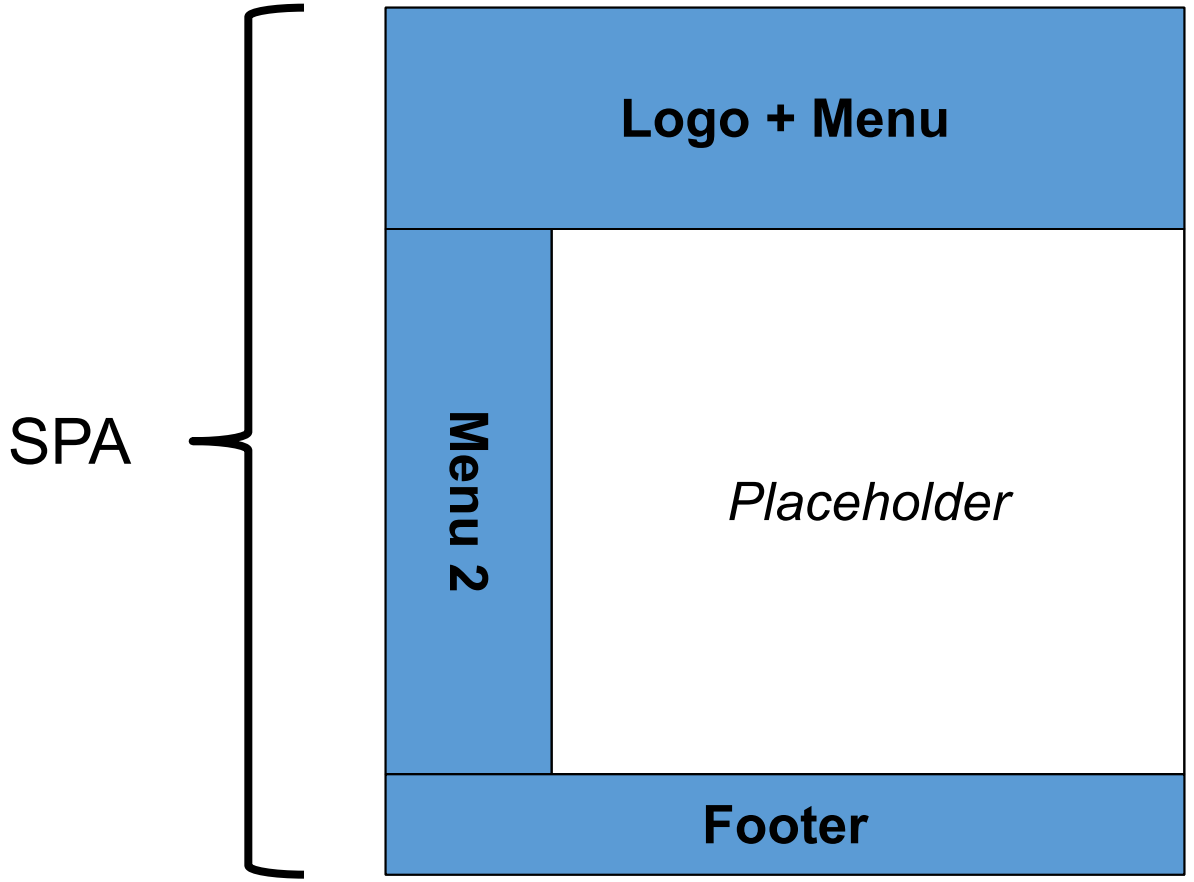
- SPAs → single page application
- Simulate pages → routes
- URL should direct to the routed component
  - Menus & bookmarks (today → Google ☺)
  - Sharing (social platforms)
  - **The Back button!**
  - **For development**

Contents

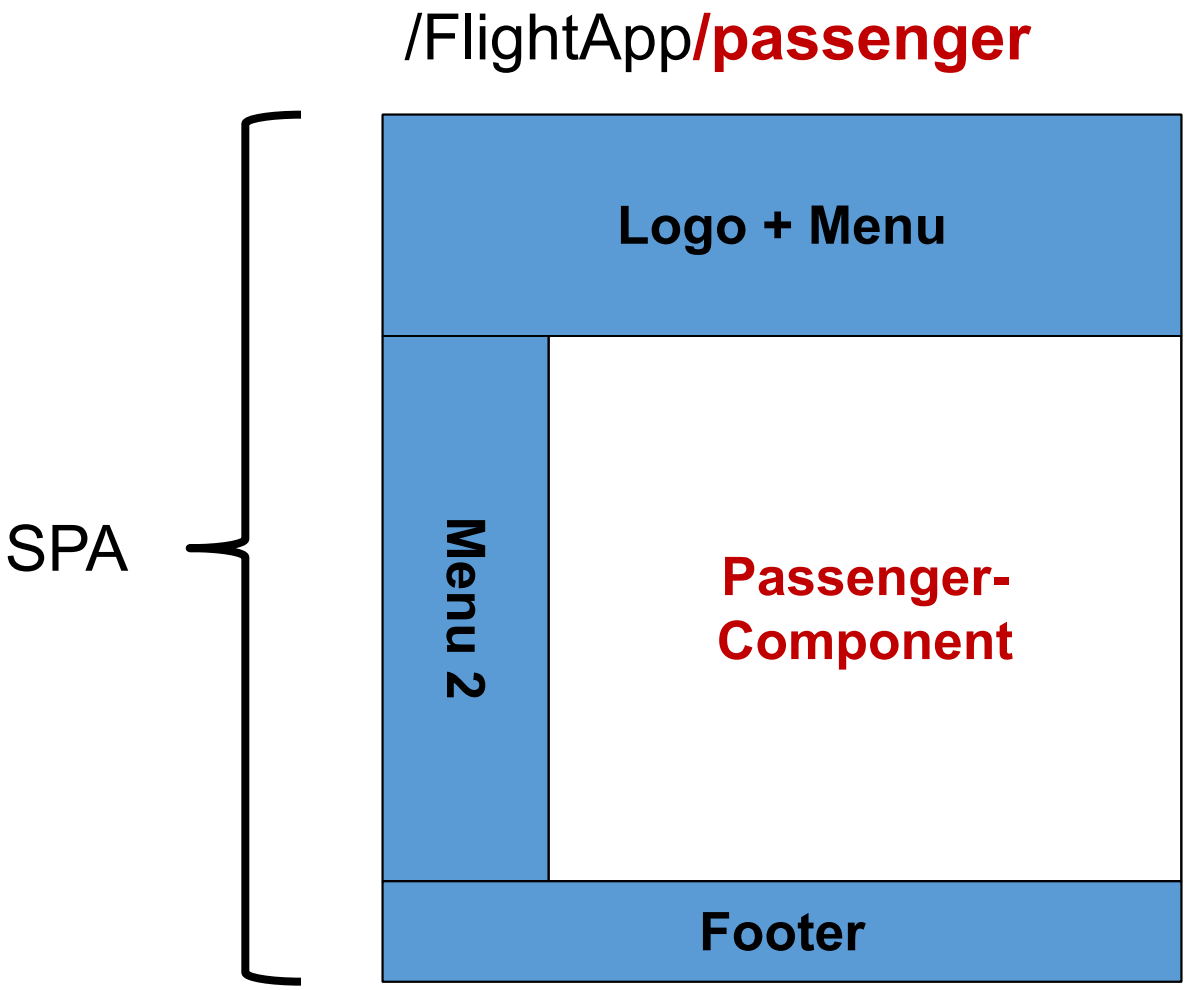
- Basics & Parameters
- Child Routes
- Aux Routes
- Guards (new as functions in NG14!)
- Resolver
- Lazy Loading

Angular Router

Routing in Angular



Routing in Angular



Configuration

```
const APP_ROUTES: Routes = [
    {
        path: 'home',
        component: HomeComponent
    },
    {
        path: 'flight-search',
        component: FlightSearchComponent
    },
    {
        path: '**',
        redirectTo: 'home'
    }
]
```

Configuration

```
// app.module.ts
@NgModule({
    imports: [
        BrowserModule,
        HttpModule,
        FormsModule,
        RouterModule.forRoot(ROUTE_CONFIG)
    ],
    […]
})
export class AppModule {}
```

For Root-Module

For Feature-Module: forChild

AppComponent

```
<a routerLink="/home">Home</a>
<a [routerLink]="'/flight-search'">Flight Search</a>

<div>
   <router-outlet></router-outlet>
</div>
```

Parameters

- passenger
- passenger/7
- passenger/7/flights
- passenger/flights
- ~~passenger/7?details=true&page=7~~
- passenger/7;details=true;page=7
- passenger/7;details=true;page=7/flights

Parameters

```
const APP_ROUTES: Routes = [
    […]
    {
        path: 'flight-search',
        component: FlightSearchComponent
    },
    {
        path: 'flight-edit/:id',
        component: FlightEditComponent
    }
}
```

Reading Parameters

```
export class FlightEditComponent {
    id = '';

    constructor(private route: ActivatedRoute) {
        route.params.subscribe(
            (params) => {
                this.id = params['id'];
                […]
            }
        );
    }
    […]
}
```

Reading Parameters

```
export class FlightEditComponent {
    id = '';

    constructor(private route: ActivatedRoute) {
        route.paramMap.subscribe(
            paramMap => {
                this.id = paramMap.get('id');
                […]
            }
        );
    }
    […]
}
```

Reading Parameters

```
export class FlightEditComponent {
    id?: number;

    constructor(private route: ActivatedRoute) {
        route.paramMap.subscribe(
            paramMap => {
                this.id = +paramMap.get('id'); // or
                this.id = Number(paramMap.get('id'));
                […]
            }
        );
    }
    […]
}
```

Links for Routes with Parameters

```
<a [routerLink]="['/flight-edit', flight.id, { showDetails: "true" }]">Edit</a>
```

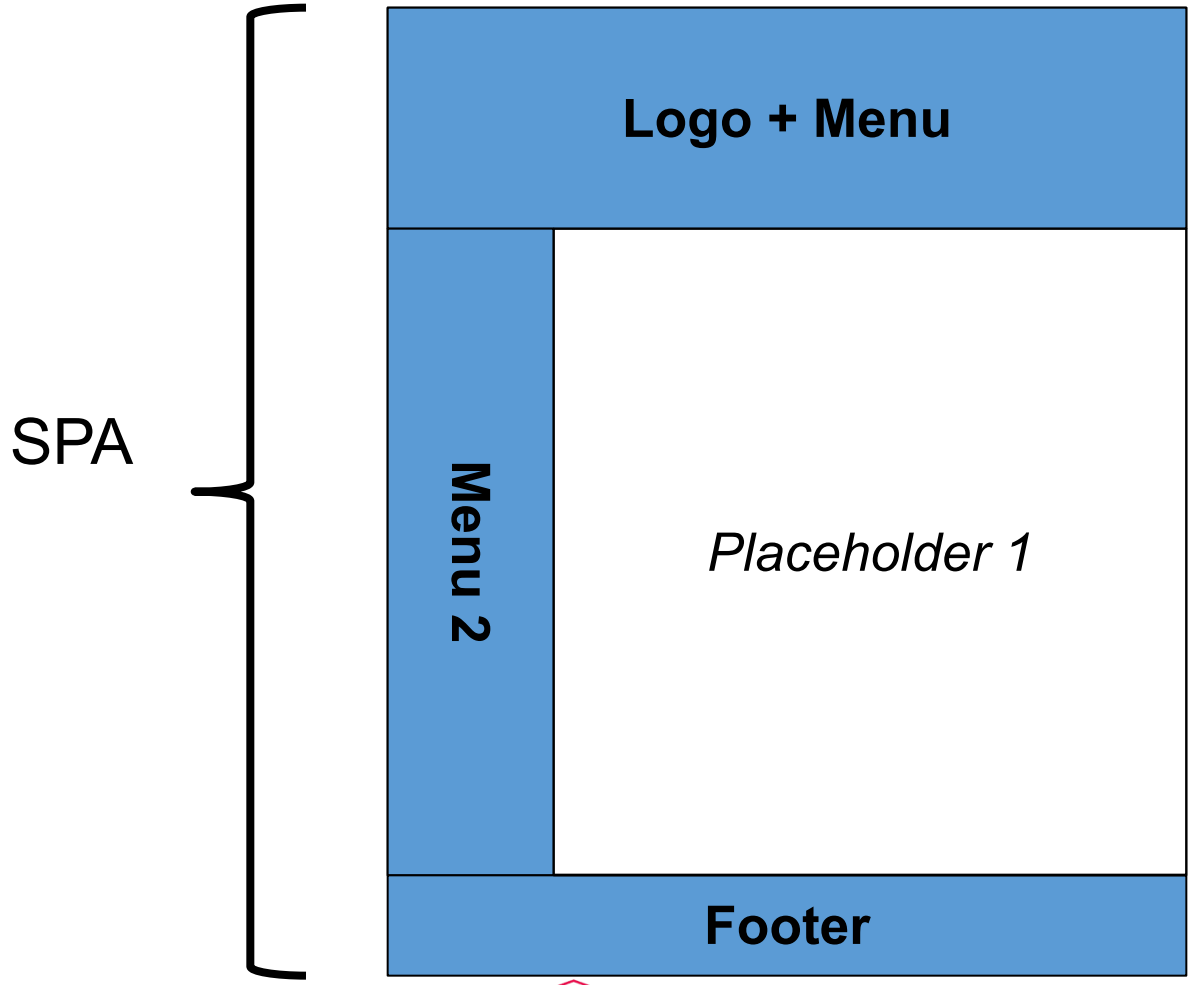
DEMO

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

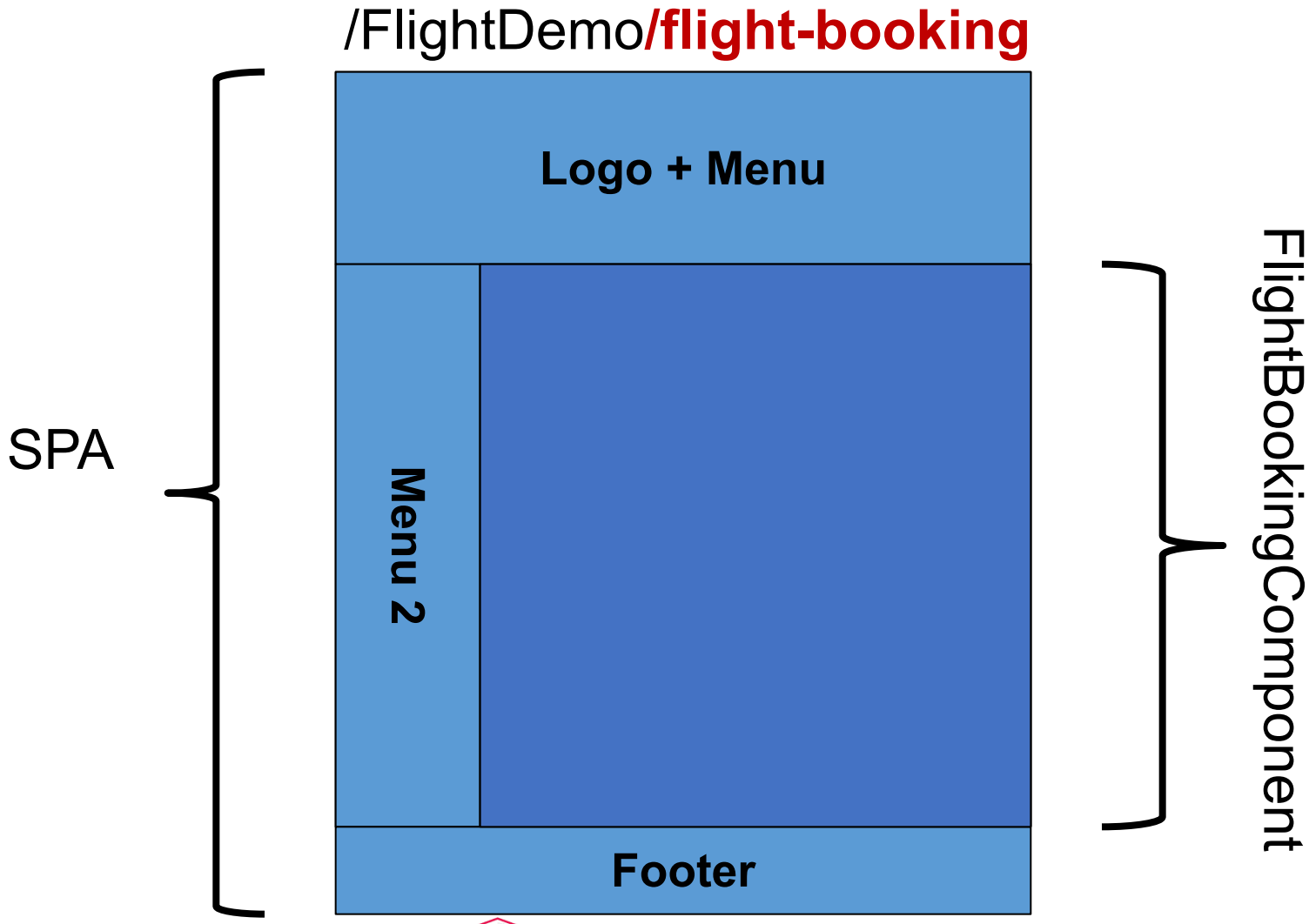
SOFTWARE
ARCHITECT

Hierarchical Routing

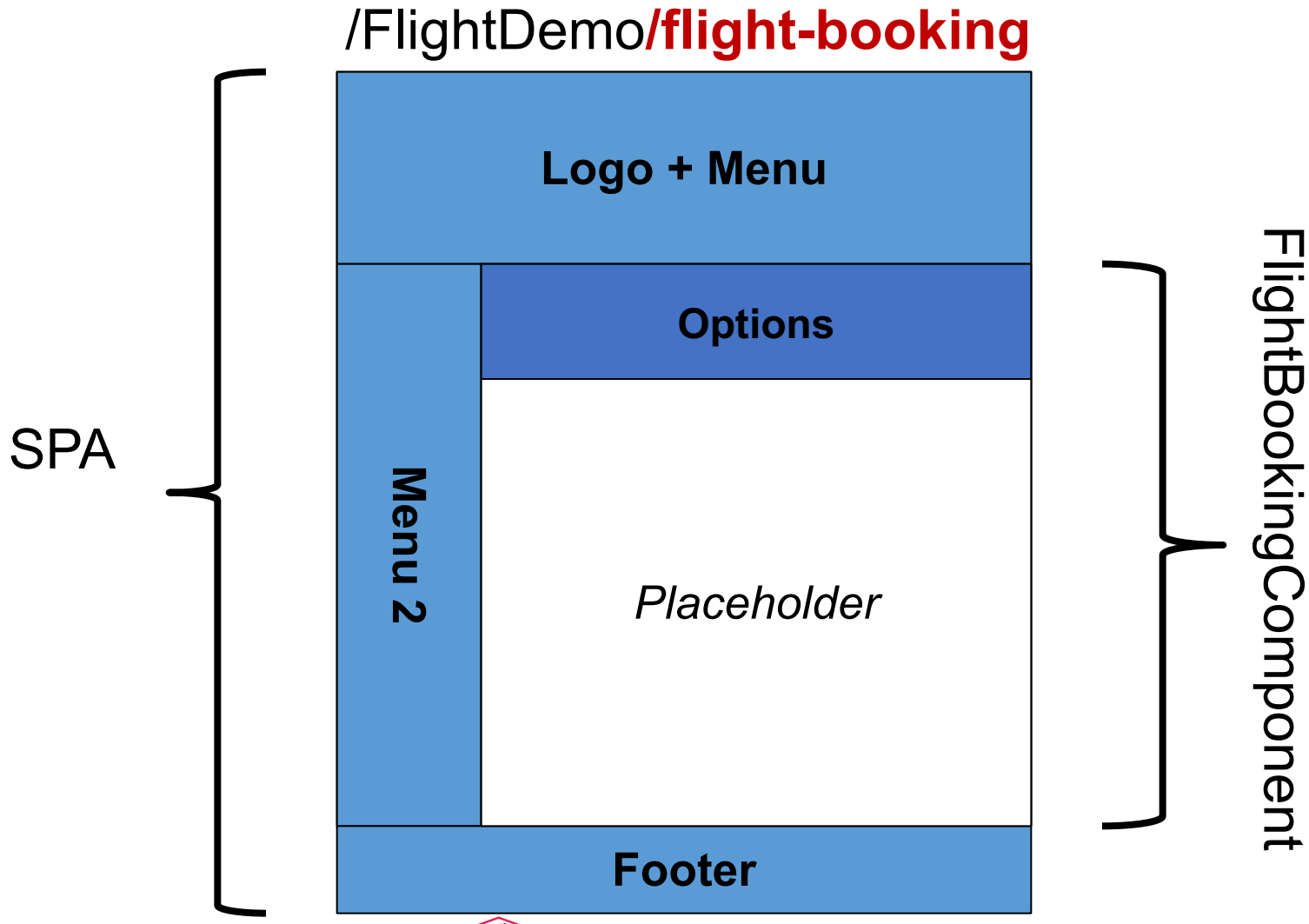
Hierarchical Routing



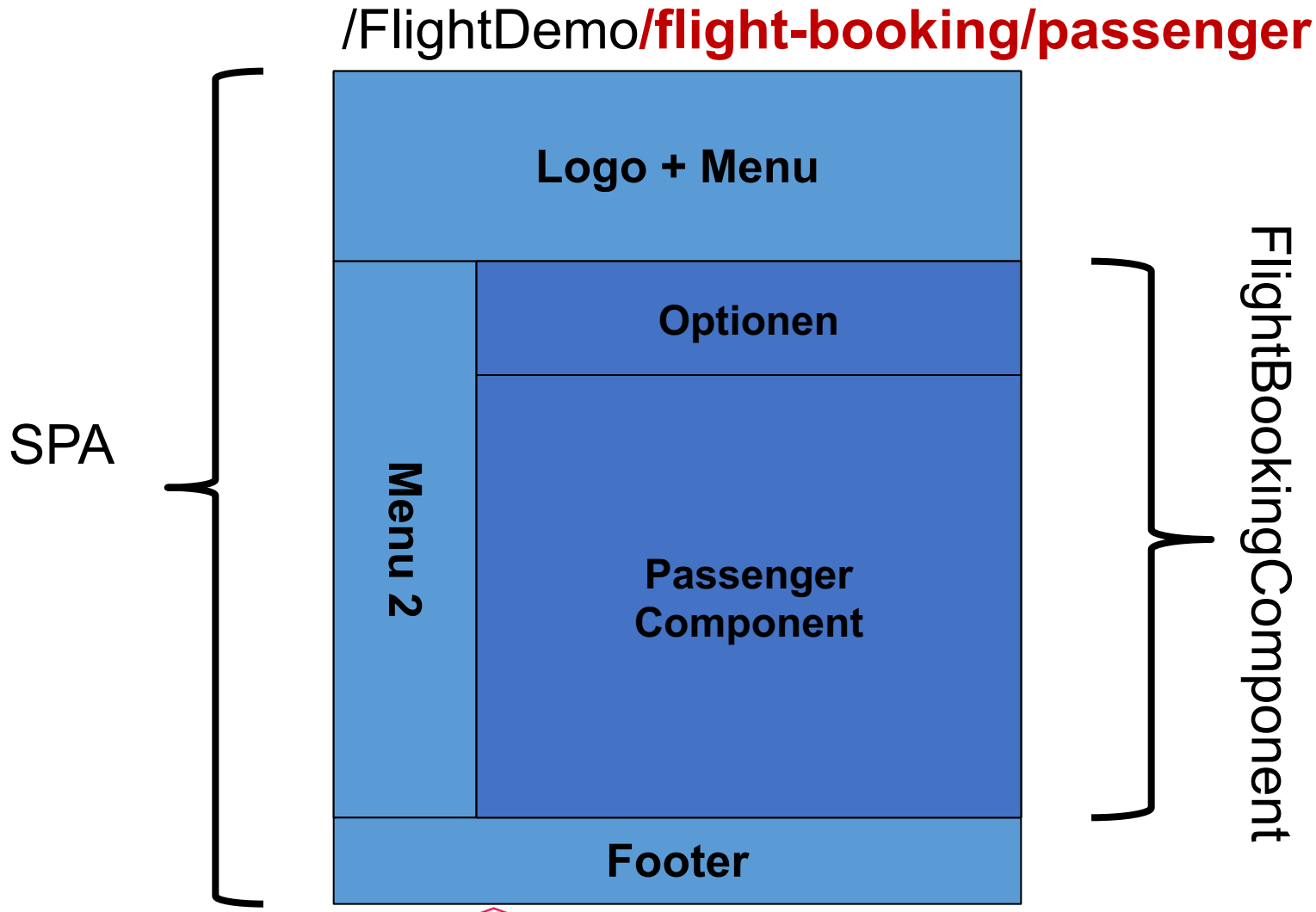
Hierarchical Routing



Hierarchical Routing



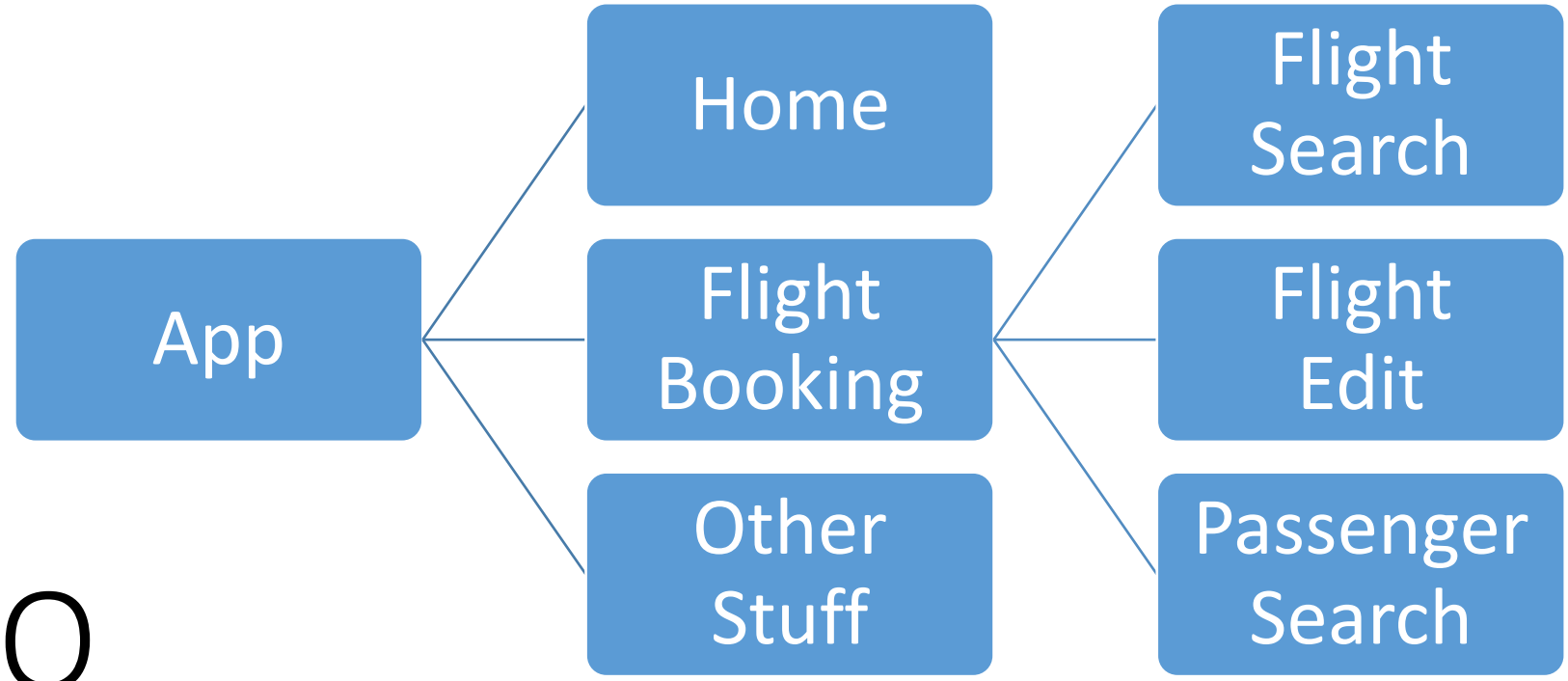
Hierarchical Routing



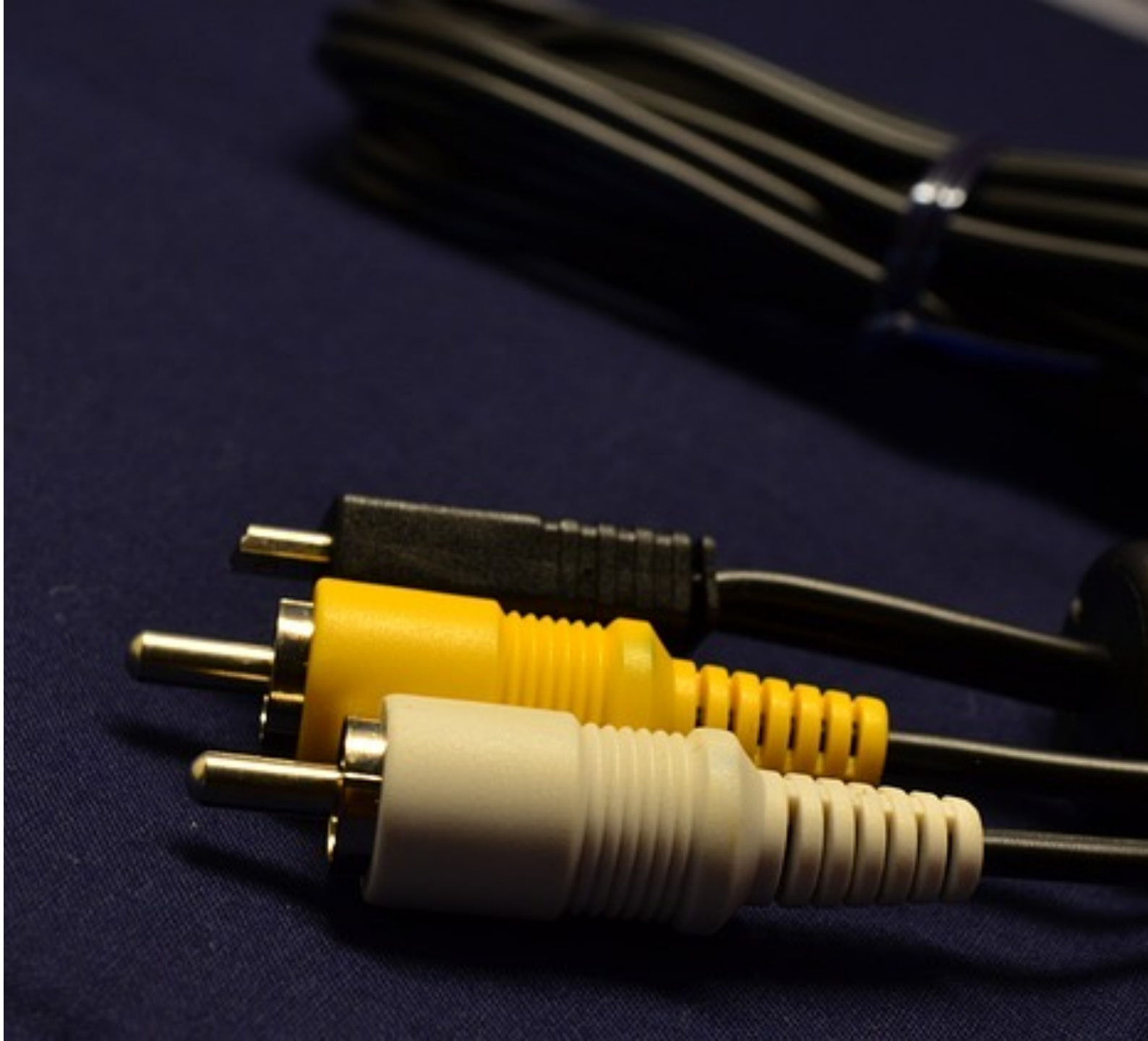
Configuration

```
const APP_ROUTES: Routes = [
    {
        path: '',
        component: HomeComponent
    },
    {
        path: 'flight-booking',
        component: FlightBookingComponent,
        children: [
            {
                path: 'flight-search',
                component: FlightSearchComponent
            },
            […]
        ]
    }
];
```

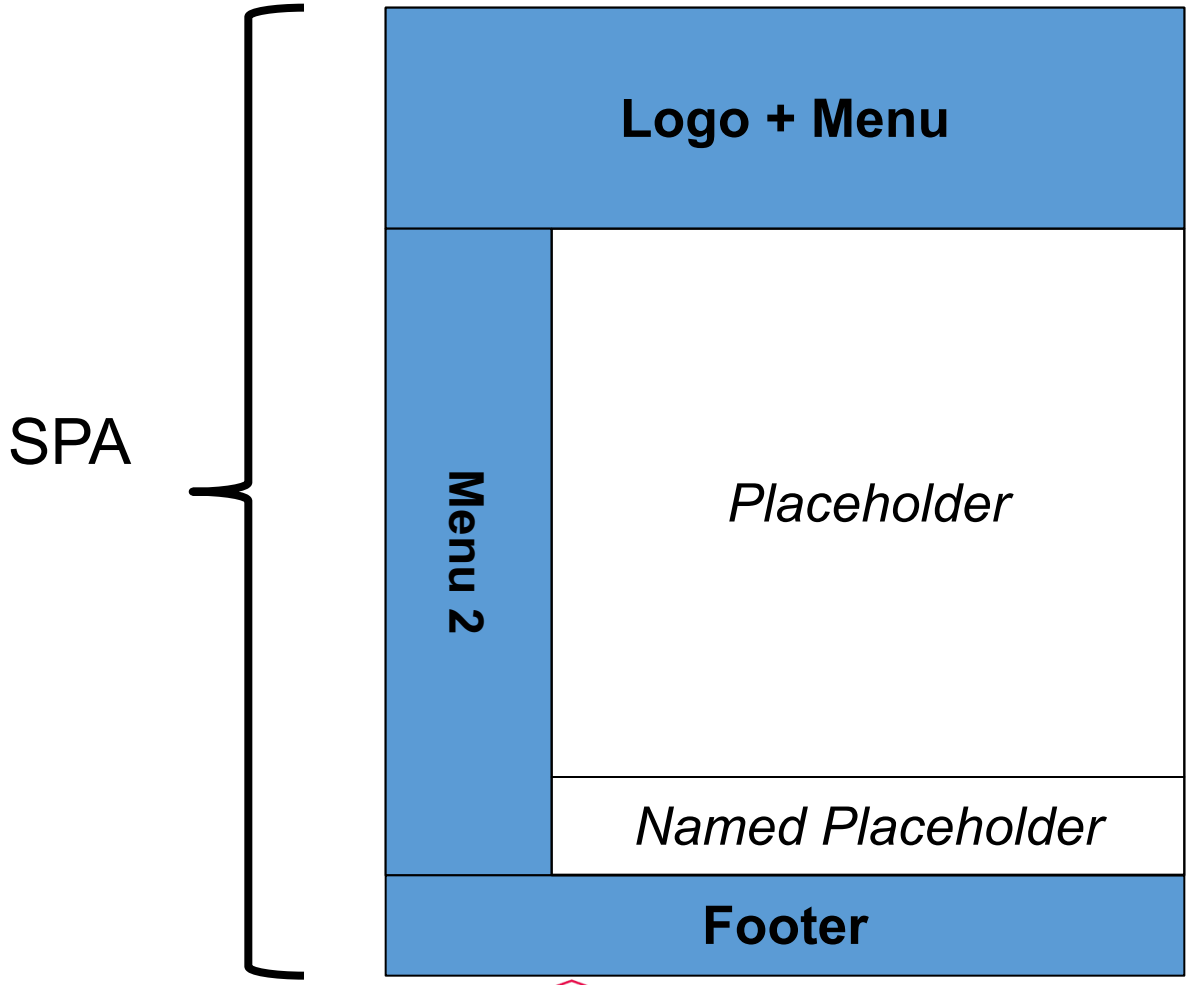
DEMO



Aux Routes

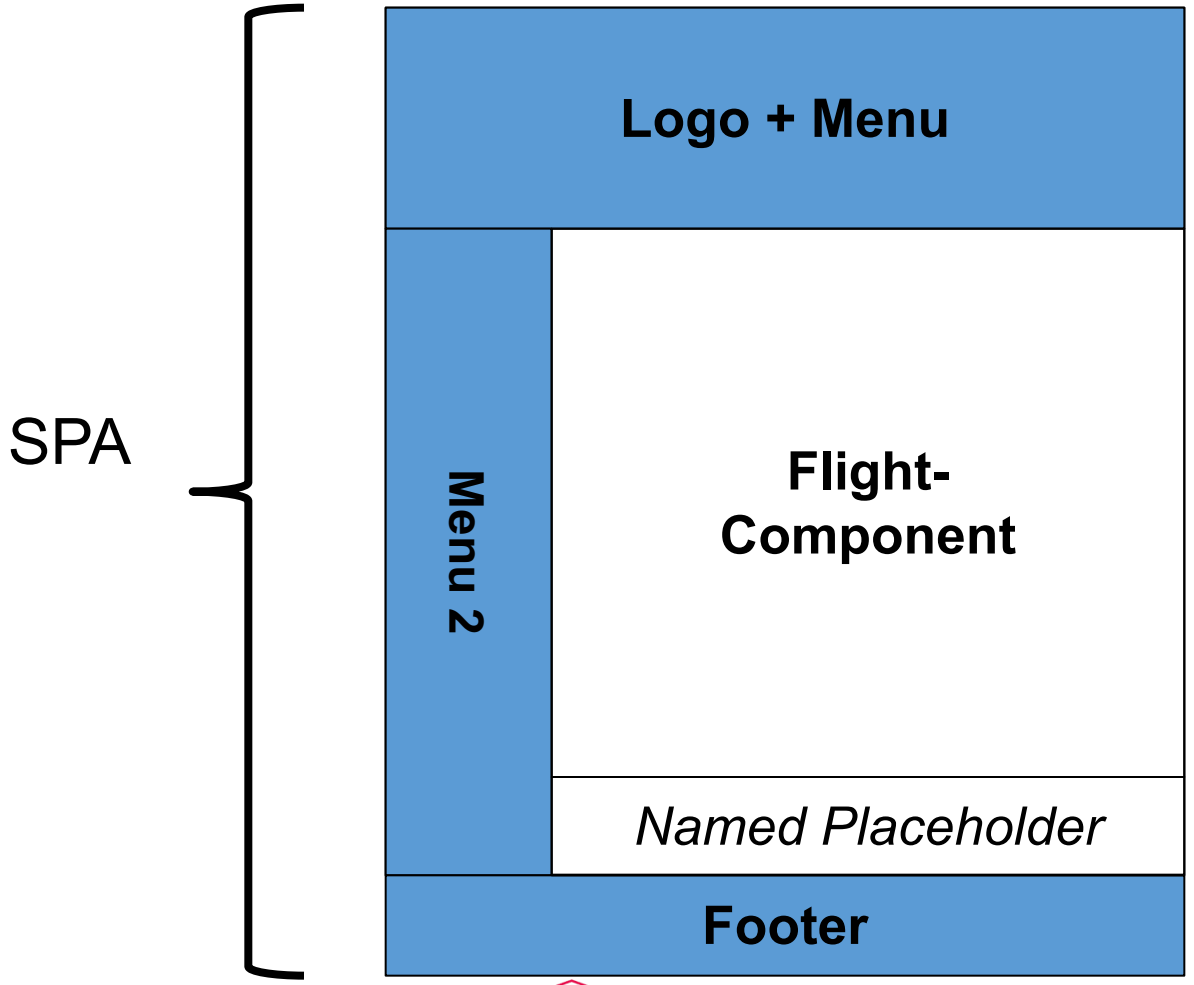


Aux-Routes



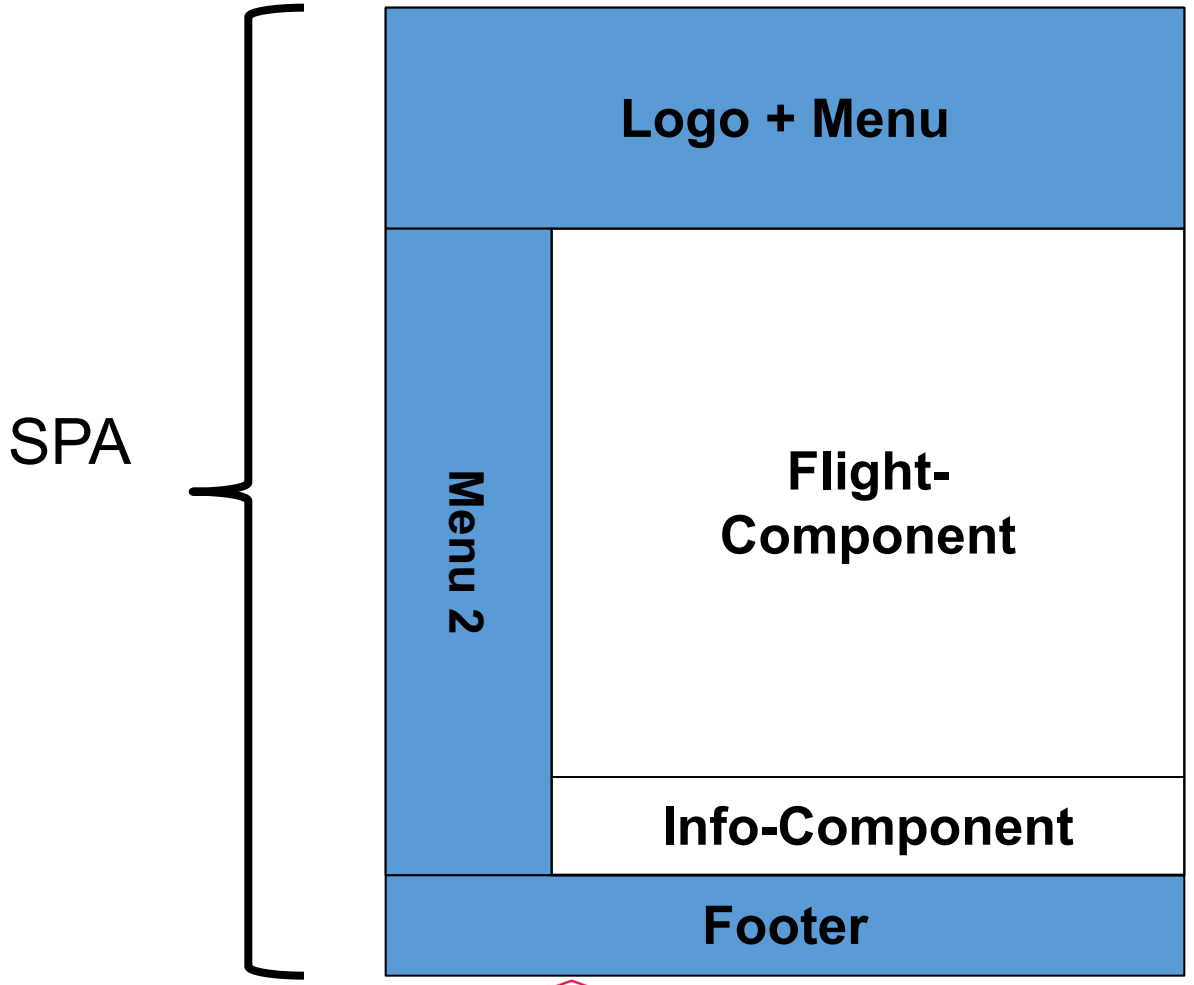
Aux-Routes

/FlightApp/**flights**



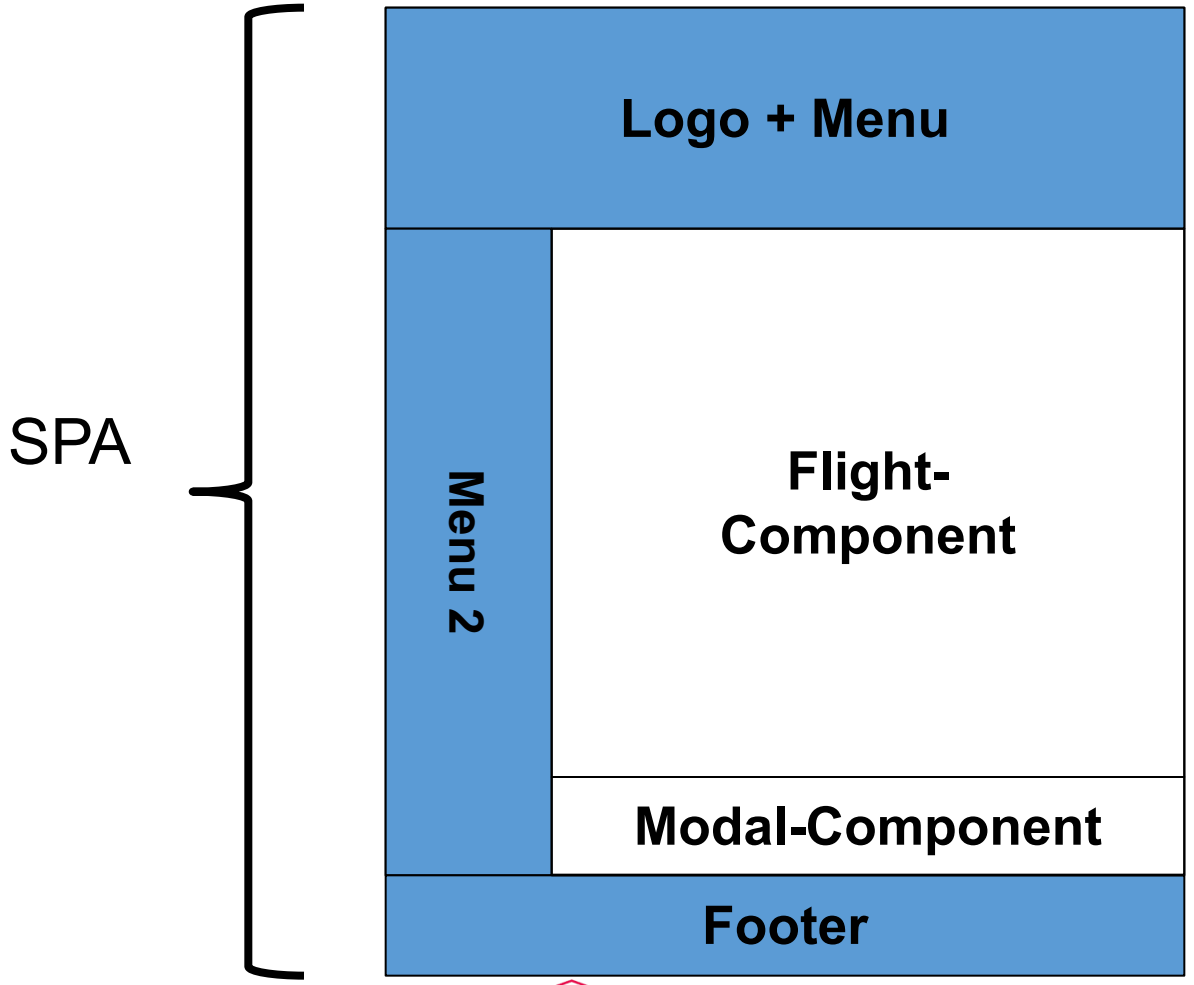
Aux-Routes

/FlightApp/**flights(aux:info)**



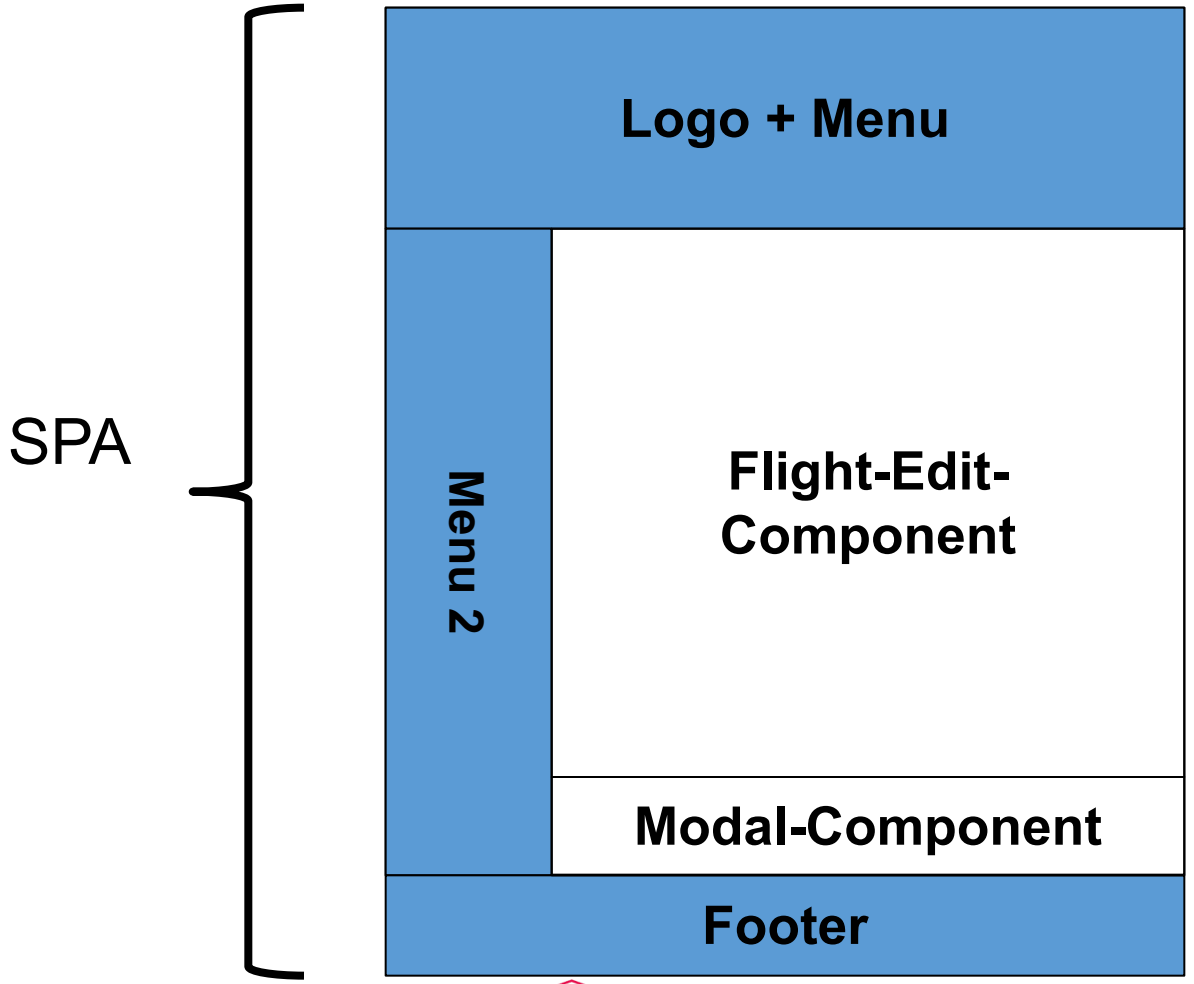
Aux-Routes

/FlightApp/**flights(aux:info/modal)**



Aux-Routes

/FlightApp/**flights(aux:info/modal)/edit/17**



Use Cases

- Partly autonomous parts of an application
- „Norton Commander Style“
- (CSS-based) Popups and Modals

Define Outlets

Default Name: primary

```
<router-outlet></router-outlet>

<hr>

<router-outlet name="aux"></router-outlet>
```

Configuration

```
export const APP_ROUTES: Routes = [
    {
        path: 'home',
        component: HomeComponent
    },
    {
        path: 'info',
        component: InfoComponent,
        outlet: 'aux'
    },
    {
        path: 'dashboard',
        component: DashboardComponent,
        outlet: 'aux'
    }
]
```

Activating Aux-Routes

```
<a [routerLink]="[{outlets: { aux: 'info' }}]">
    Activate Info
</a>

<a [routerLink]="[{outlets: { aux: null }}]">
    Deactivate Info
</a>
```

Activating Several Aux Routes at Once

```
<a [routerLink]="[{outlets: {
                    aux: 'basket',
                    primary: 'flight-booking/flight-search' }}]"> … </a>


<a [routerLink]="[{outlets: {
                    aux: 'basket',
                    primary: ['flight-booking', 'flight-search'] }}]"> … </a>


<a [routerLink]="[{outlets: {
                    aux: 'basket',
                    primary: ['flight-booking', 'flight-edit', 17] }}]"> … </a>
```

Code-based Routing

```
export class AppComponent {
    constructor(private router: Router) {}

    activateInfo(): void {
        this.router.navigate([{outlets: { aux: 'info' }}]);
    }

    deactivateInfo(): void {
        this.router.navigate([{outlets: { aux: null }}]);
    }
}
```

DEMO

Lab

Guards

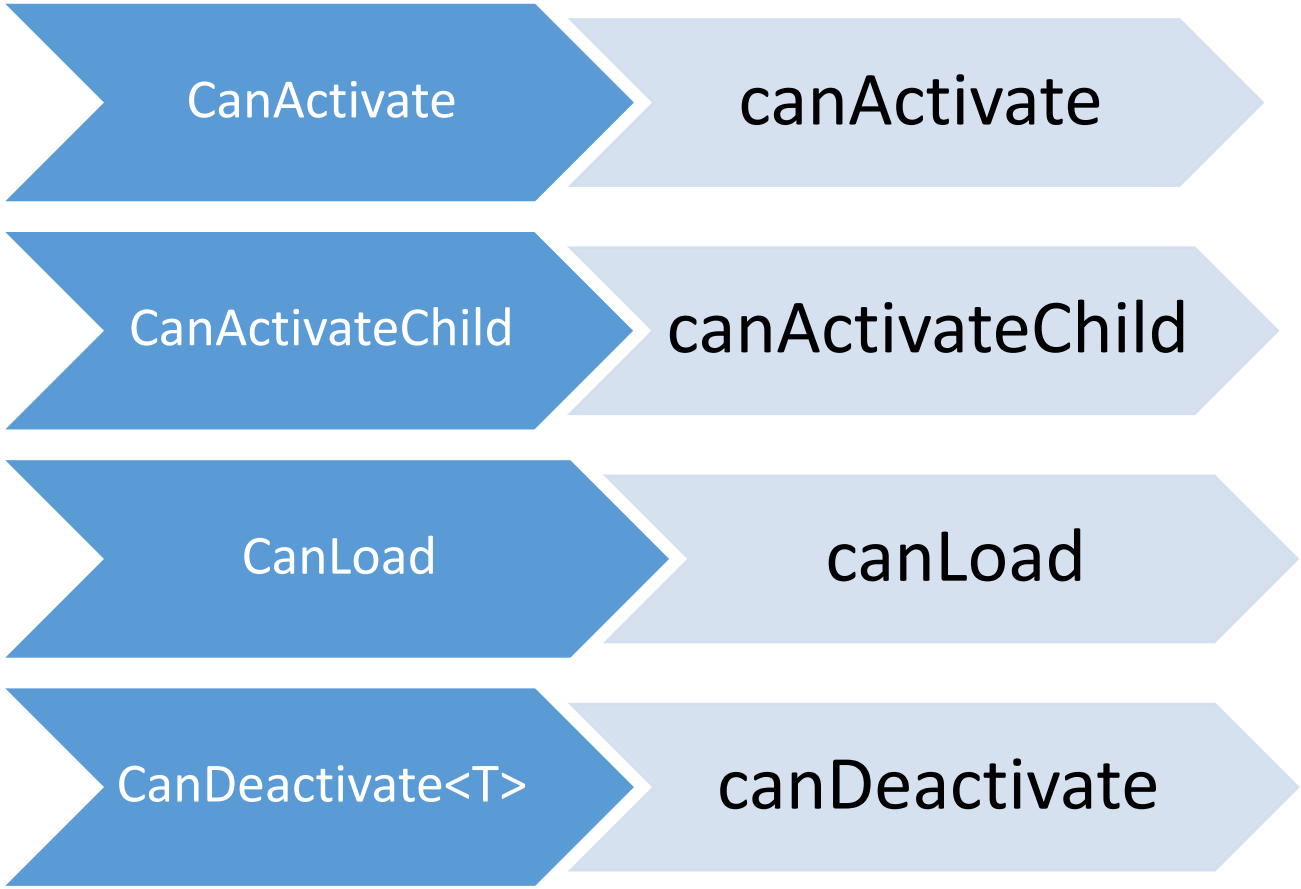
What are Guard?

- ~~Services (deprecated)~~

- const function (since NG 14)

- Can prevent the Activation or Deactivation of a Route

Guards



Result: boolean | Observable<boolean> | Promise<boolean>

Example

```
export const authGuard = () => {
  const authService = inject(AuthService);
  const router = inject(Router);

  if (authService.userName) {
    return true;
  }

  // Redirect to the login page
  return router.navigate(['/home', { needsLogin: true }]);
};
```

DEMO

Resolver

ANGULAR ARCHITECTS
INSIDE KNOWLEDGE

SOFTWARE ARCHITECT

What are Resolver?

- Services
- Are activated when the Router switches over to another route
- Can load needed data
- Postpone activation of target route until data is loaded
- Meanwhile, a loading indicator can be shown

Resolver

```
@Injectable()
export class FlightResolver implements Resolve<Flight> {
    constructor(private flightService: FlightService) {}

    resolve(route, state):
          Observable<Flight> | Promise<Flight> | any {

        return […]
    }
}
```

Register Resolver

```
const FLIGHT_BOOKING_ROUTES: Routes = [
    […]

    {
        path: 'flight-edit/:id',
        component: FlightEditComponent,
        resolve: {
            flight: FlightResolver
        }
    }

];
```

Token

Receive Data in Component

```
@Component({ … })
export class FlightEditComponent {
    flight?: Flight;

    constructor(private route: ActivatedRoute) {}

    ngOnInit(): void {
        this.route.data.subscribe(
            data => {
                this.flight = data['flight'];
            }
        );
    }
}
```

DEMO

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

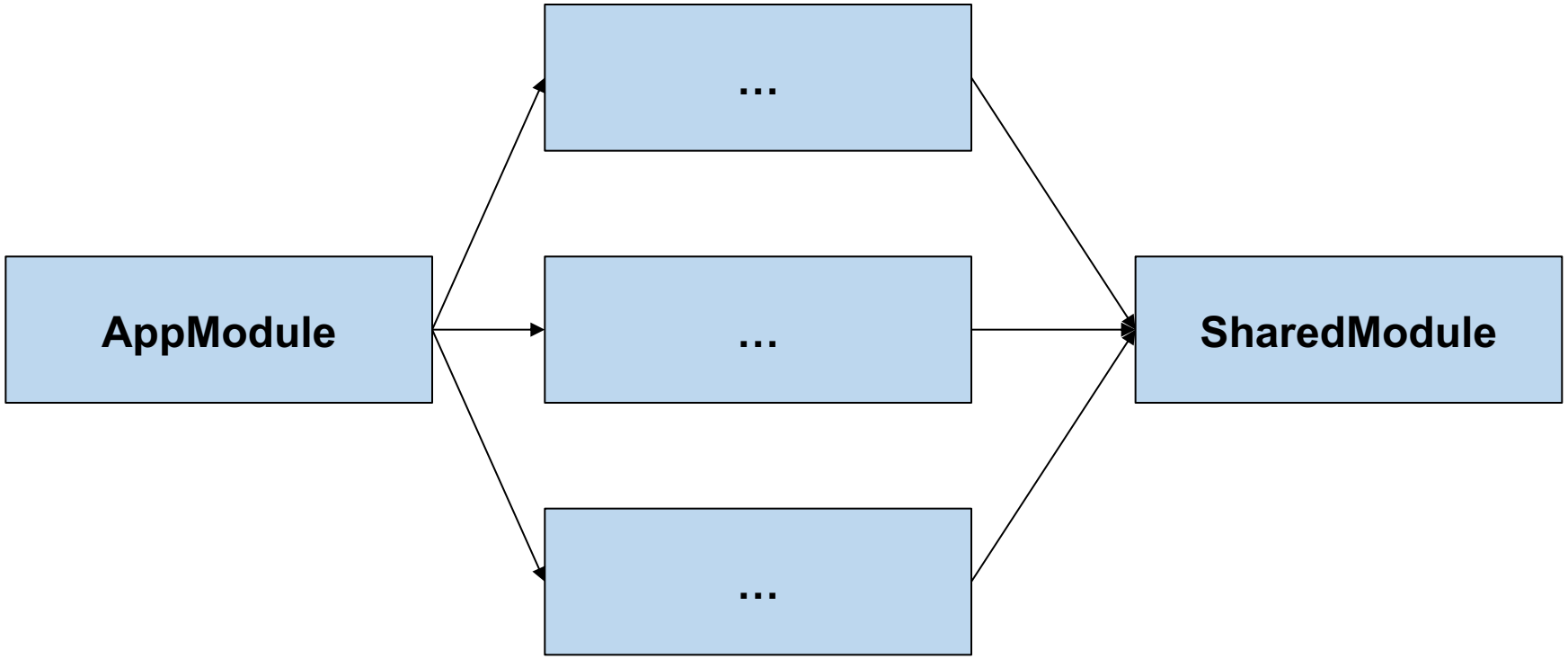
SOFTWARE
ARCHITECT

Lab

Why Lazy Loading?

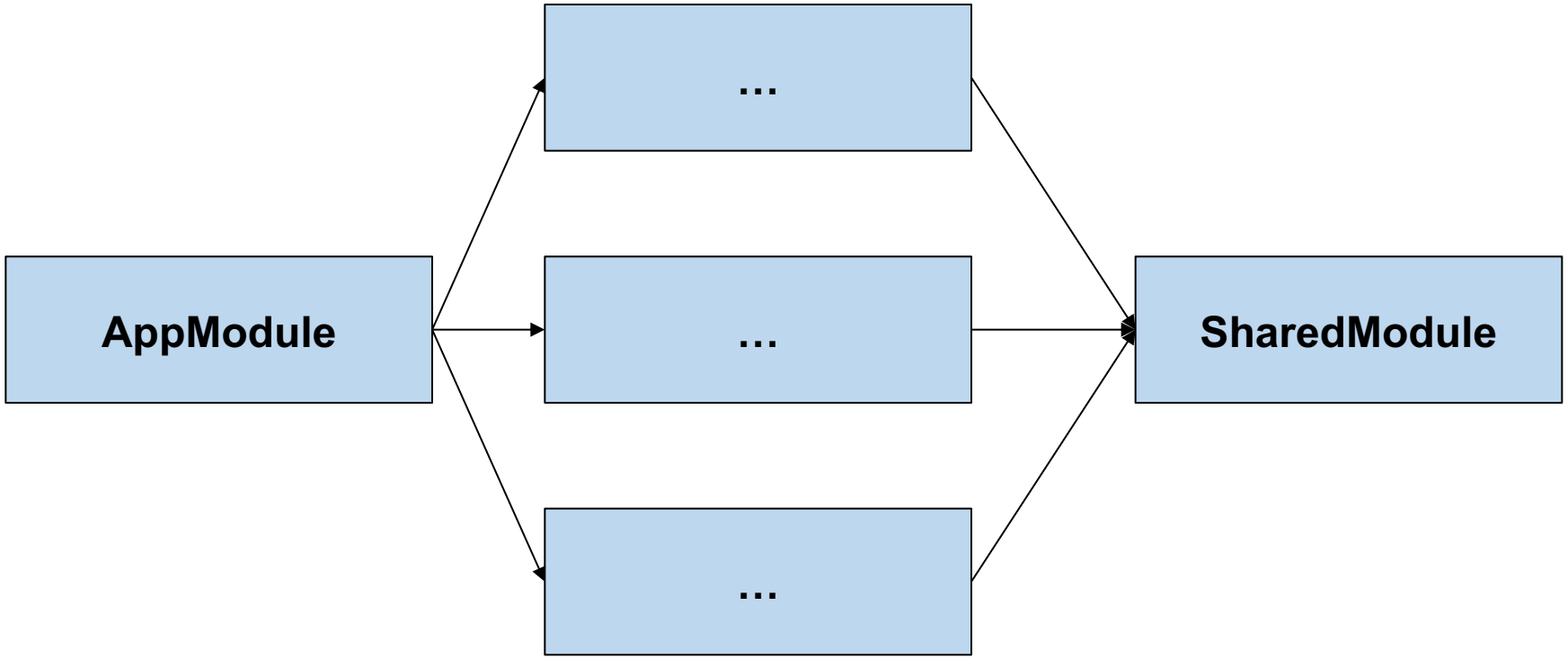
- Improve initial load time (performance → very important!)

Module Structure



Root Module **Feature Modules** **Shared Module**

Lazy Loading



Root Module **Feature Modules** **Shared Module**

Root Module with Lazy Loading

```
const APP_ROUTES: Routes = [
    {
        path: 'home',
        component: HomeComponent
    },
    {
        path: 'flight-booking',
        loadChildren: () => import('./flight-booking/flight-booking.module')
                              .then(m => m.FlightBookingModule)
    }
];
```

Routes for "lazy" Feature Module

```
const FLIGHT_BOOKING_ROUTES: Routes = [
    {
        path: 'flight-search',
        component: FlightSearchComponent,
        […]
    },
    […]
}
```

Routes for "lazy" Feature Module

```
const FLIGHT_BOOKING_ROUTES: Routes = [
    {
        path: 'flight-search',
        component: FlightSearchComponent,
        […]
    },
    […]
}
```

flight-booking/flight-search

Triggers Lazy Loading w/ loadChildren

DEMO

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

SOFTWARE
ARCHITECT

Lazy Loading

- Lazy Loading means: Load it later, after startup
- Better initial load performance
- But: Delay during execution for loading on demand

Preloading

Idea

- Once the initial load (the important one) is complete load the lazy loaded modules (before they are even used)

- Once the module will come into use it's immediately accessable

Activate Preloading

```
…
imports: [
    […]
    RouterModule.forRoot(
        APP_ROUTES,
        { preloadingStrategy: PreloadAllModules }
    );
]
…
```

DEMO

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

SOFTWARE
ARCHITECT

Intelligent Preloading with ngx-quicklink

```
…
imports: [
    […]
    QuicklinkModule,
    RouterModule.forRoot(
        ROUTE_CONFIG,
        { preloadingStrategy: QuicklinkStrategy }
    );
]
…
```

https://web.dev/route-preloading-in-angular/

https://www.npmjs.com/package/ngx-quicklink

Or CustomPreloadingStrategy

```
…
imports: [
    […]
    RouterModule.forRoot(
        ROUTE_CONFIG,
        { preloadingStrategy: CustomPreloadingStrategy }
    );
]
…
```

LAB

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

SOFTWARE
ARCHITECT

Conclusion

Child Routes & Parameters

Aux Routes

Guards

Lazy Loading & Preloading

Summary

- Child Routes & Parameters
- Aux Routes
- Guards and Resolvers
- Lazy Loading & Preloading

Homework for this evening

1. Check at least one of your teams Angular projects
2. Find out what Angular Forms are being used there
3. Report your findings tomoro morning to our group