

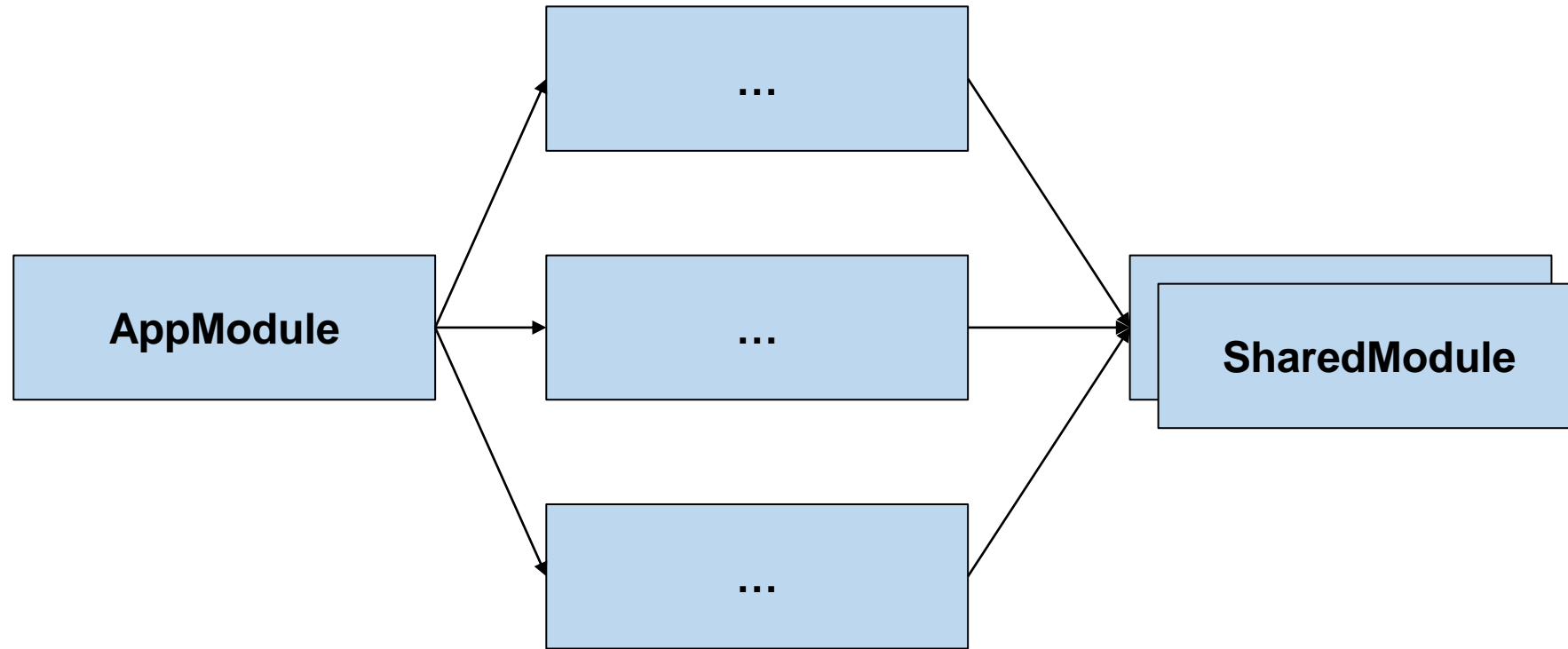


ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Angular Architecture Workshop: Modulith to Micro Frontends

ANGULARarchitects.io

Typical Module Structure



Root Module

Feature Modules

Shared Modules

Contents

- (npm-)Packages
- Nx Monorepos
- Strategic Design and DDD
- Microfrontends



Manfred Steyer

About me...

Manfred Steyer, **ANGULAR**architects.io

Angular Trainings and Consultancy



Frankfurt, Munich, Vienna
In-House: everywhere

<https://ANGULARarchitects.at/workshops>



Google Developer Expert
for Angular













Trusted Collaborator
in the Angular Team





Monorepos

Monorepo Structure

- ▶  node_modules
- ◀  projects
 - ▶  flight-admin
 - ▶  flight-api
 - ▶  flight-app
 - ▶  validation
-  .gitignore
-  angular.json
-  package-lock.json
-  package.json

Advantages

Everyone uses the latest versions

No version conflicts

No burden with distributing libs

Tooling & Generator

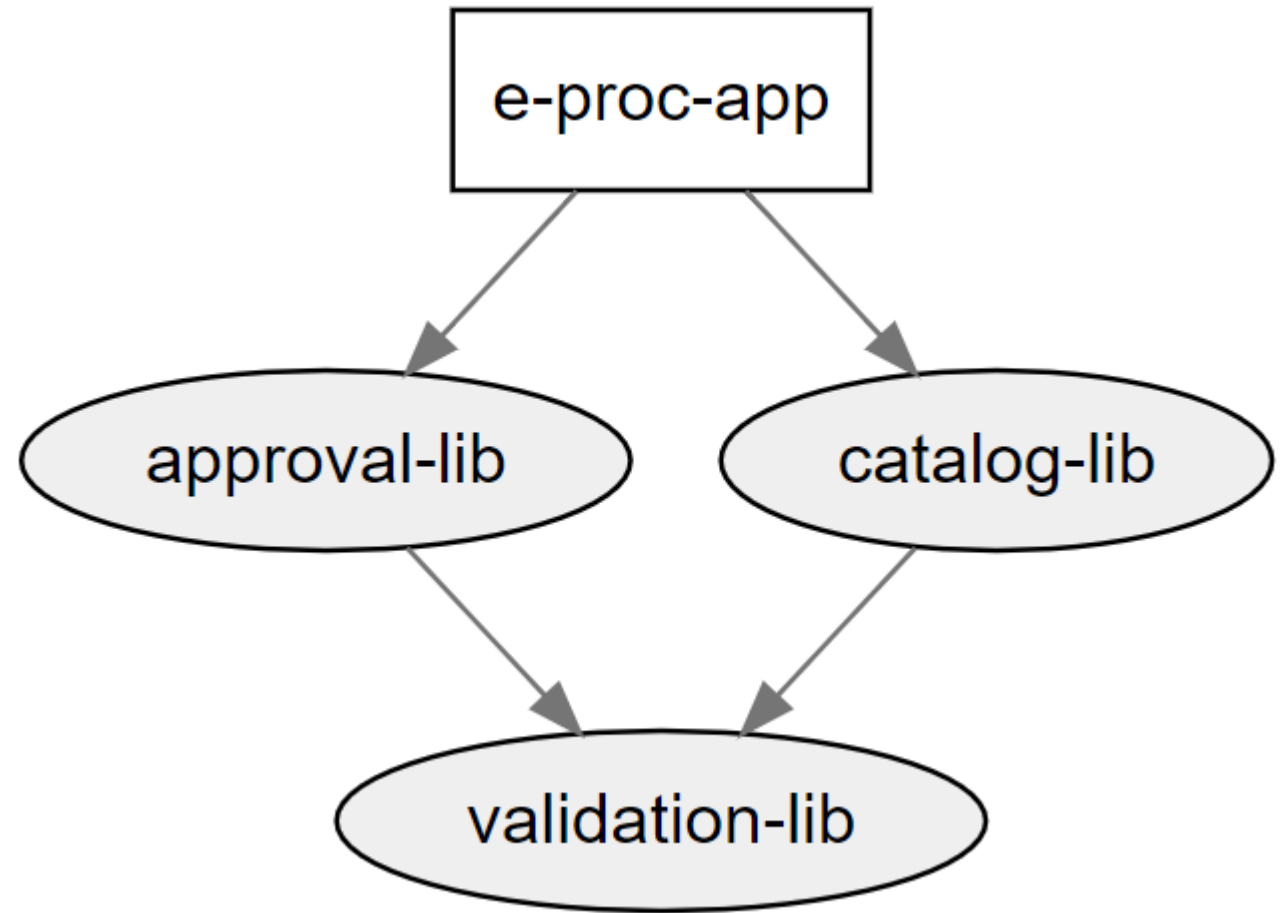
<https://nrwl.io/nx>



Nrwl Extensions for Angular

An open source toolkit for enterprise Angular applications.

Visualize Module Structure



Creating a Workspace

```
npm install -g @angular/cli
```

```
ng new workspace
```

```
cd workspace
```

```
ng generate app my-app
```

```
ng generate lib my-lib
```

```
ng serve --project my-app
```

```
ng build --project my-app
```

Creating a Workspace

```
npm install -g @angular/cli
```

```
npm init nx-workspace workspace
```

```
cd workspace
```

```
ng generate app my-app
```

```
ng generate lib my-lib
```

```
ng serve --project my-app
```

```
ng build --project my-app
```

DEMO



DDD

in a nutshell

Domain-Driven

DESIGN

Tackling Complexity in the Heart of Software



Eric Evans

Foreword by Martin Fowler

Methodology for
bridging the gap b/w
requirements and
architecture/ design

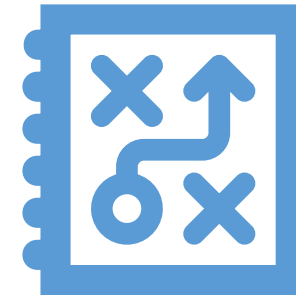
Domain Driven Design

Decomposing a System



Strategic Design

Design Patterns
& Practices



Tactical Design

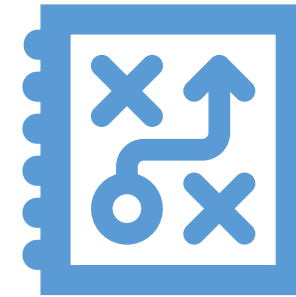
Domain Driven Design

Decomposing a System



Strategic Design

Design Patterns
& Practices



Tactical Design

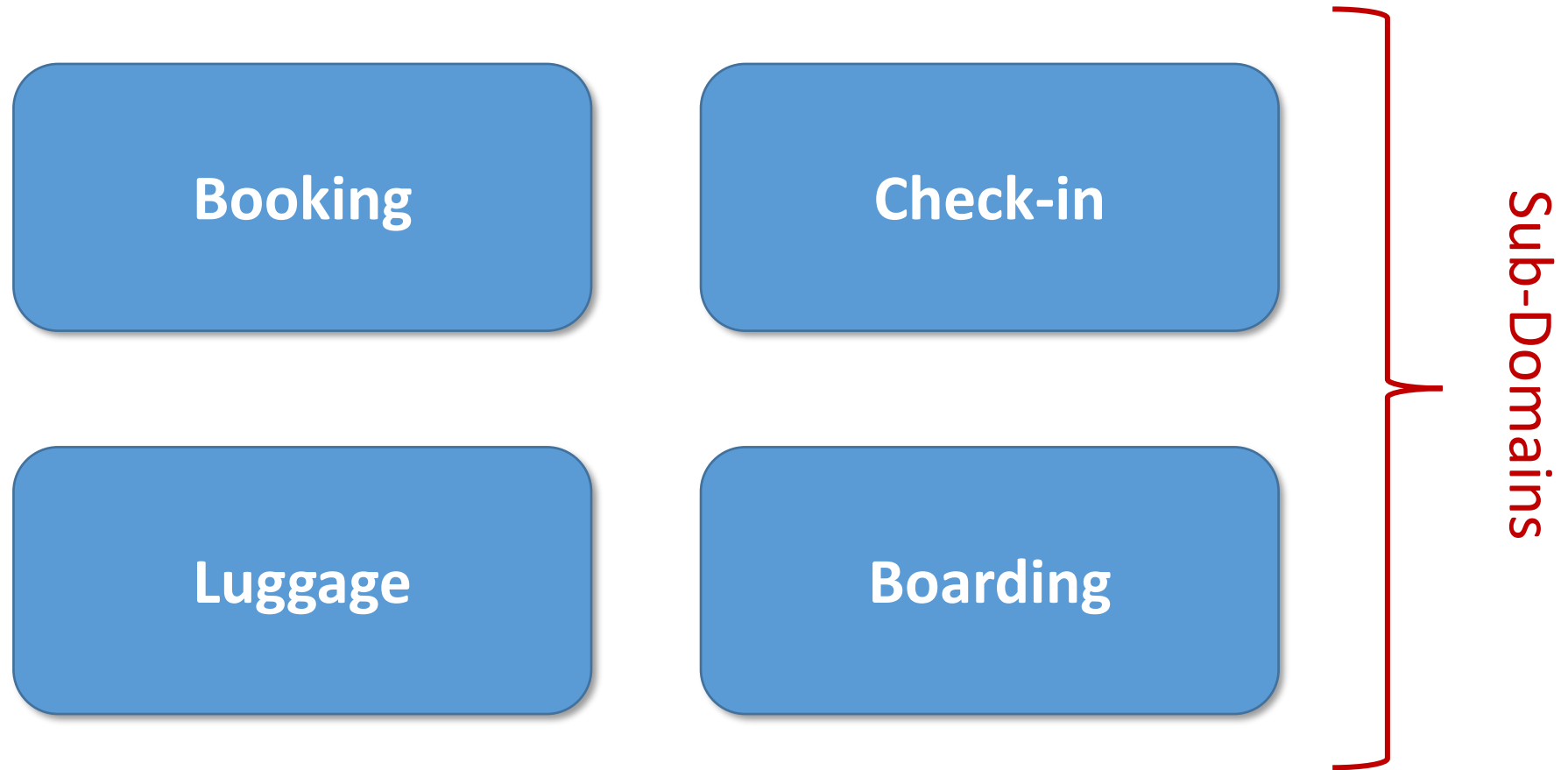
A photograph showing a dense, chaotic mess of white Ethernet cables plugged into network switches. The cables are tangled and disorganized, illustrating a lack of structured cabling. A semi-transparent white banner is overlaid at the bottom of the image with the text "This is what Strategic DDD prevents".

This is what Strategic DDD prevents

Example

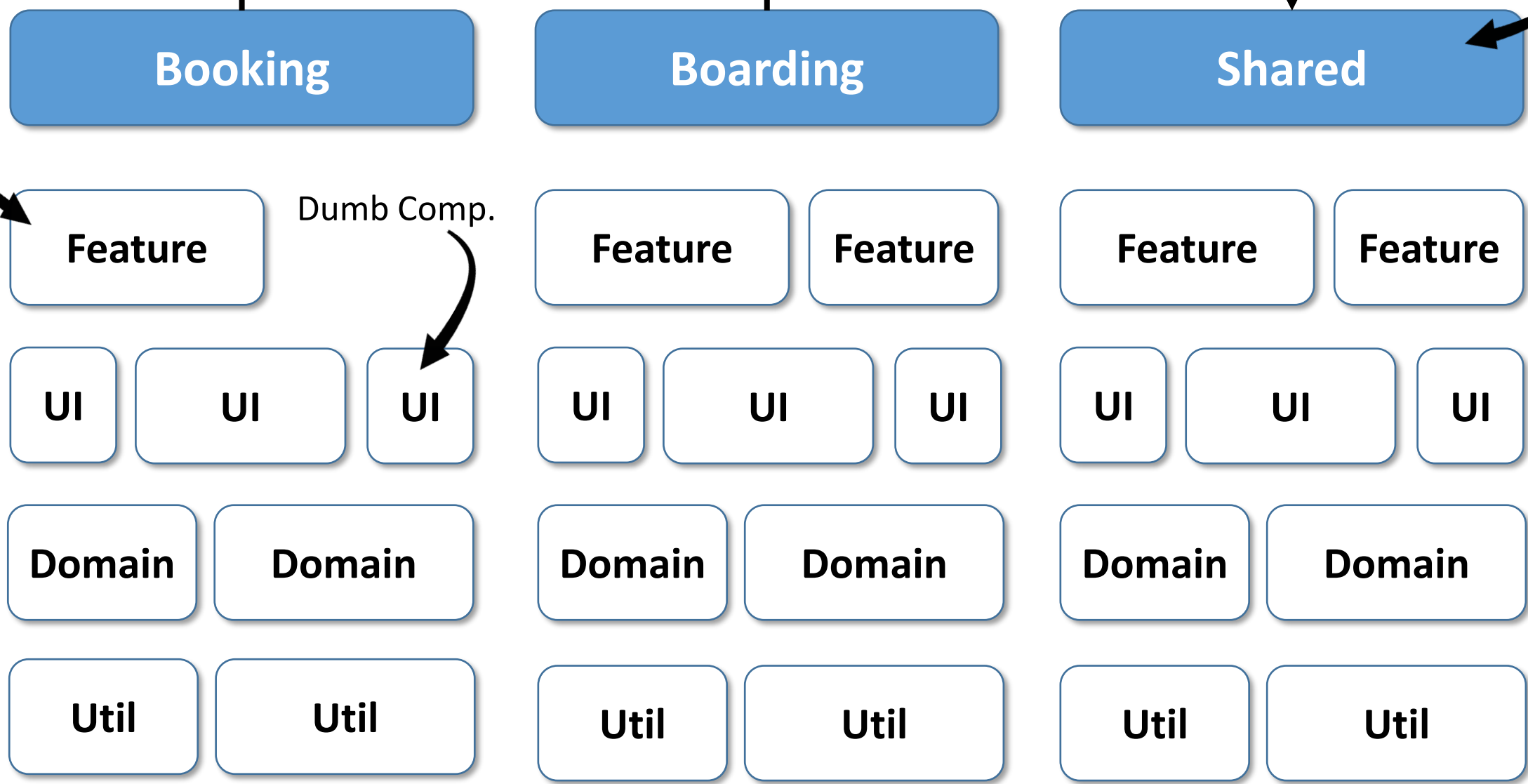
Flight System

Example



Smart
Comp.

Dumb Comp.



DEMO

Build Cache

**Best Performance:
Several Apps!**



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



Finegrained Libraries

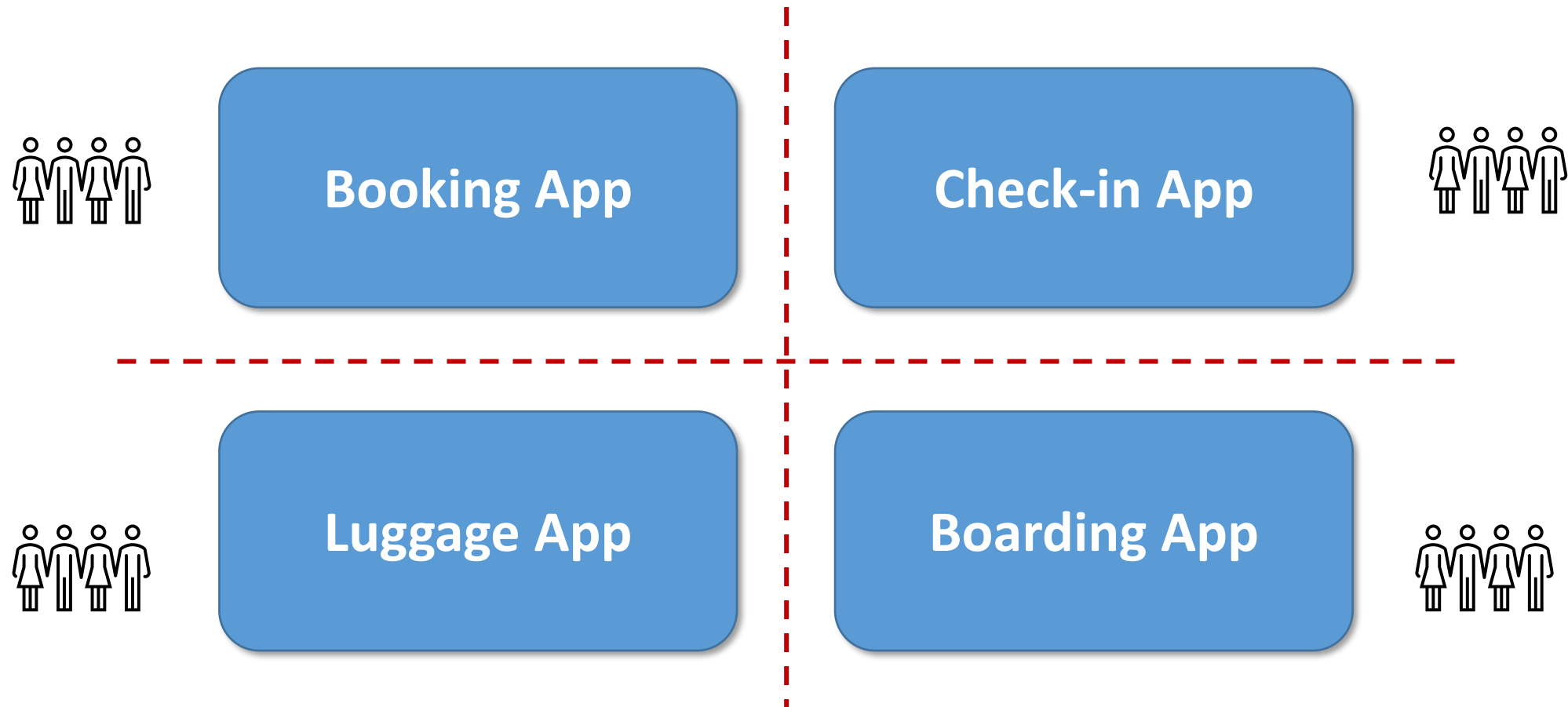
- Unit of recompilation
- Unit of retesting
- Access restrictions
- Information Hiding
- Easy: Just *ng g lib ...*
- Future replacement for NgModules?



Micro Frontends?

Short outlook

Microfrontends

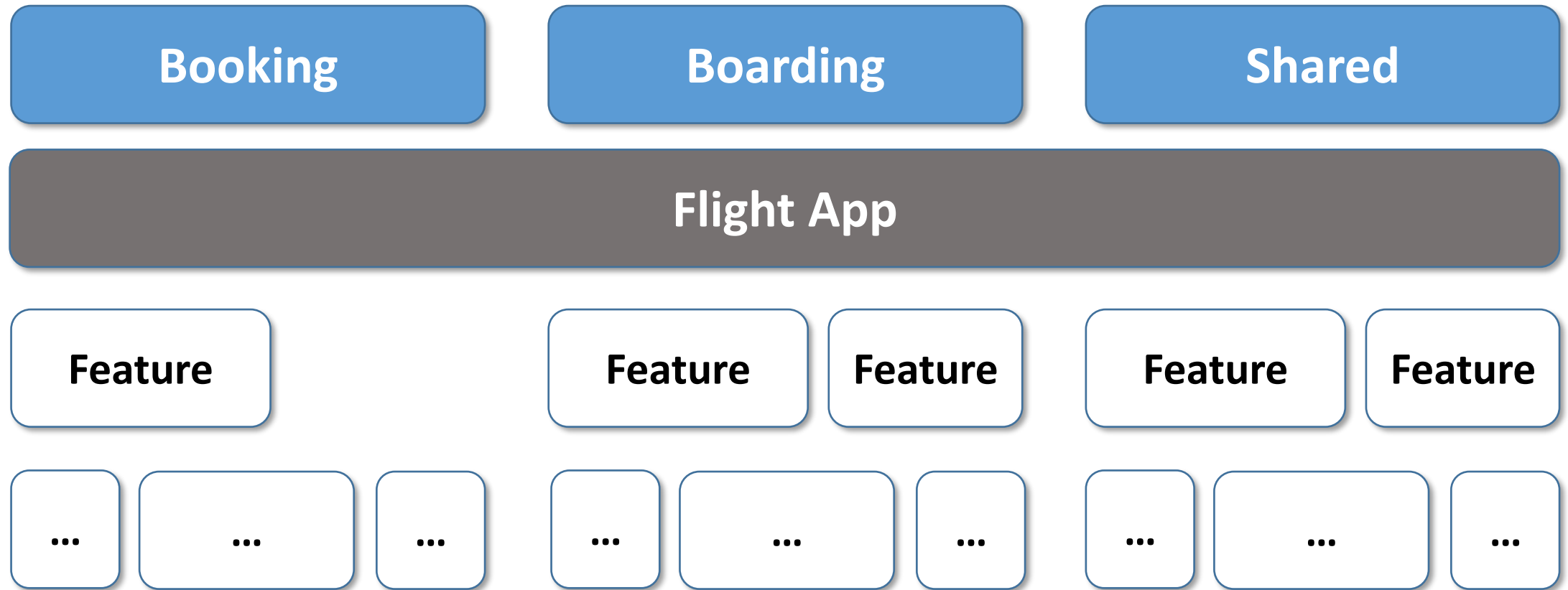


Microfrontends
are first and foremost
about **scaling teams!**

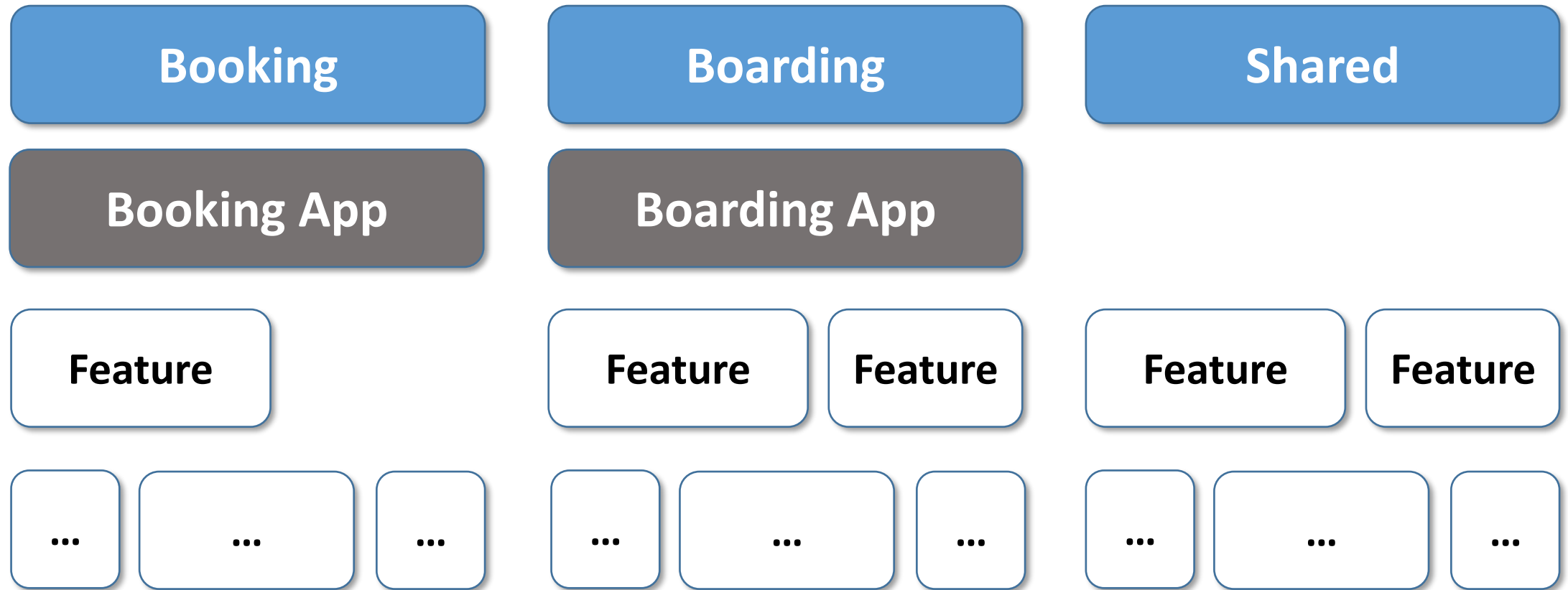


ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

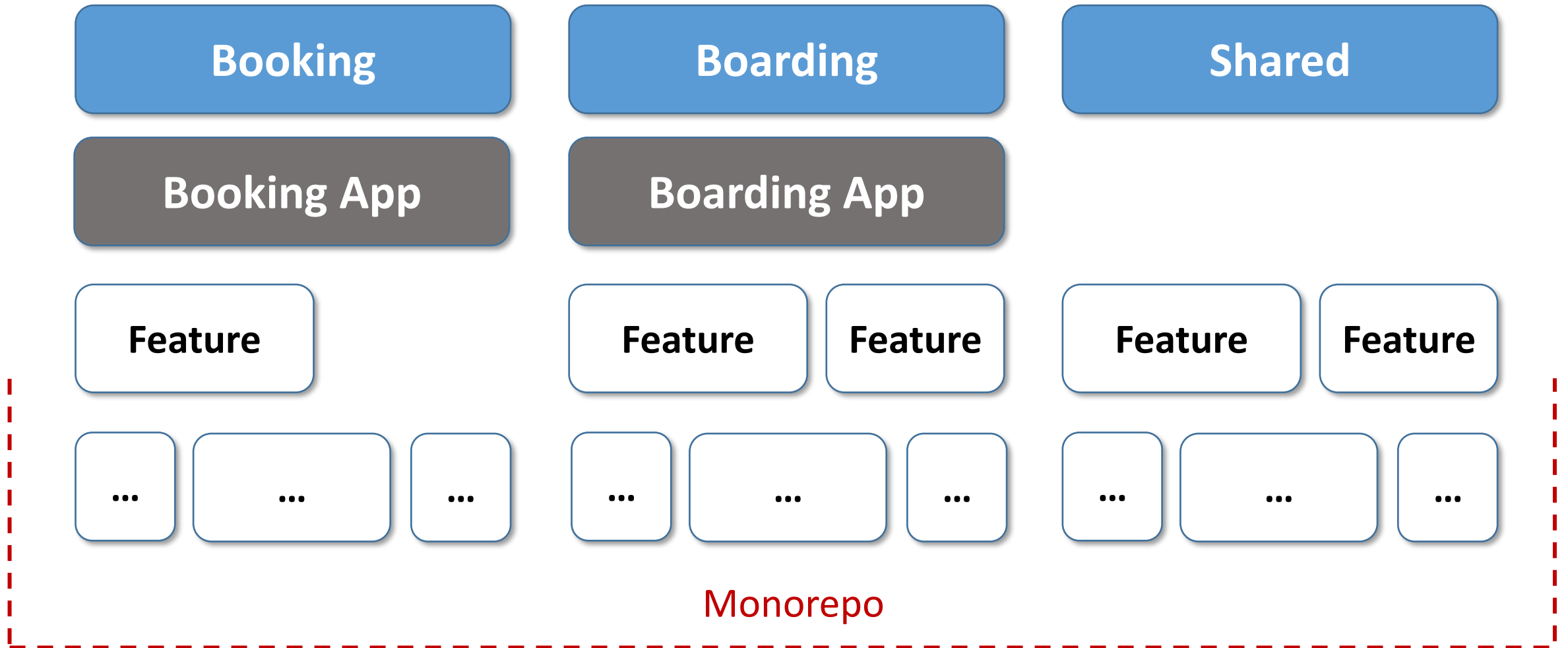
Deployment Monolith



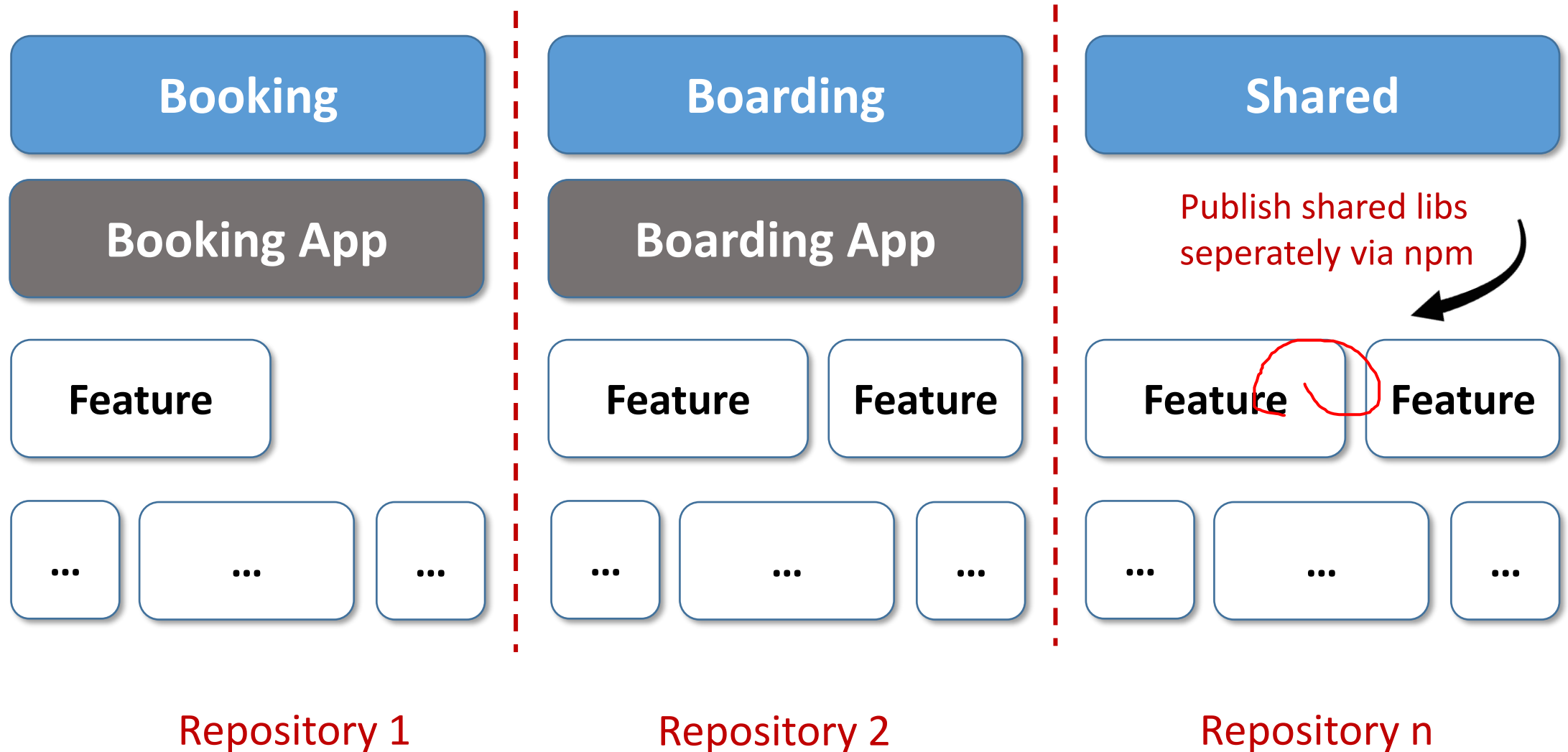
Microfrontends



Option 1: One App per Domain



Option 2: One Monorepo per Domain



Benefits

Autonomous Teams

Separate Development

Separate Deployment

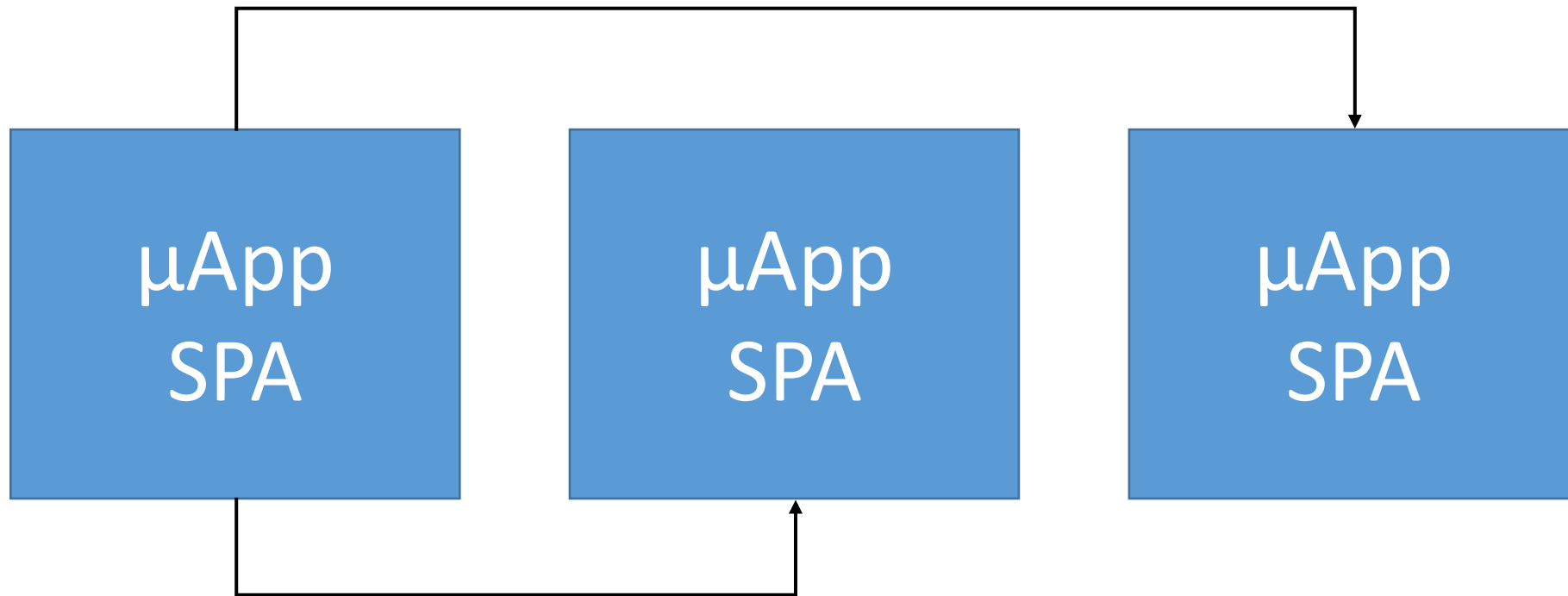
Own architecture decisions

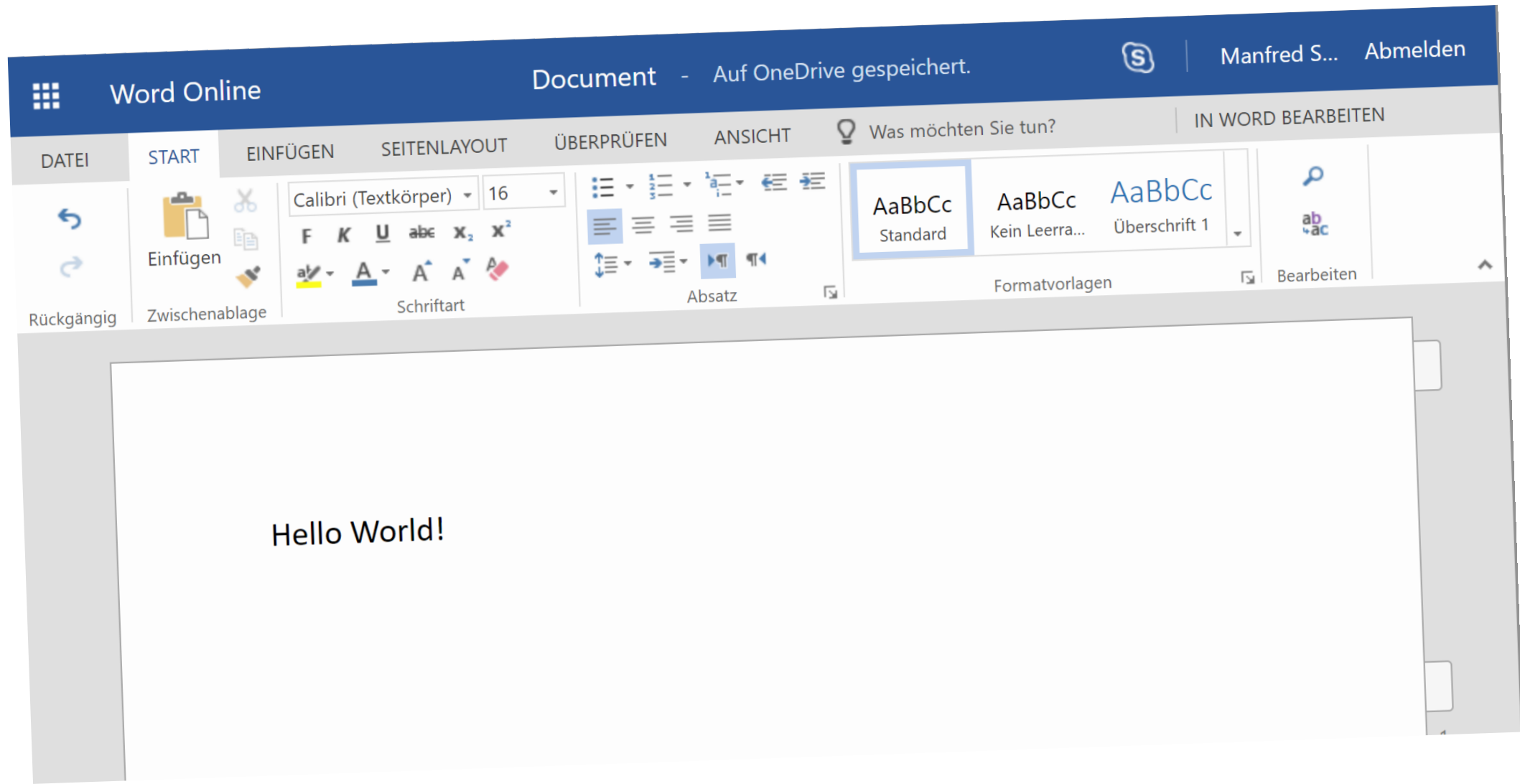
Own technology decisions

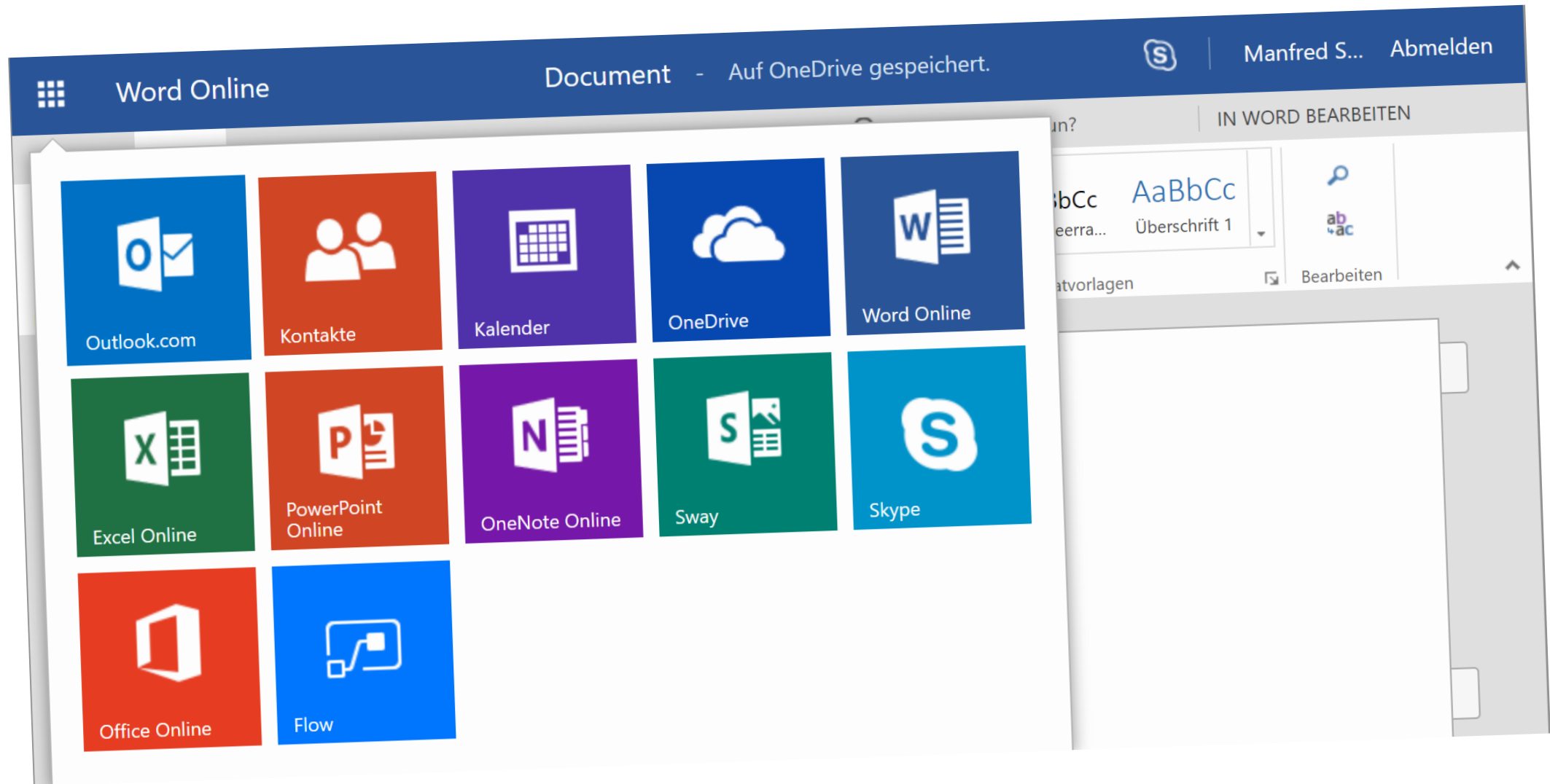


Integration via Hyperlinks

UI Composition w/ Hyperlinks



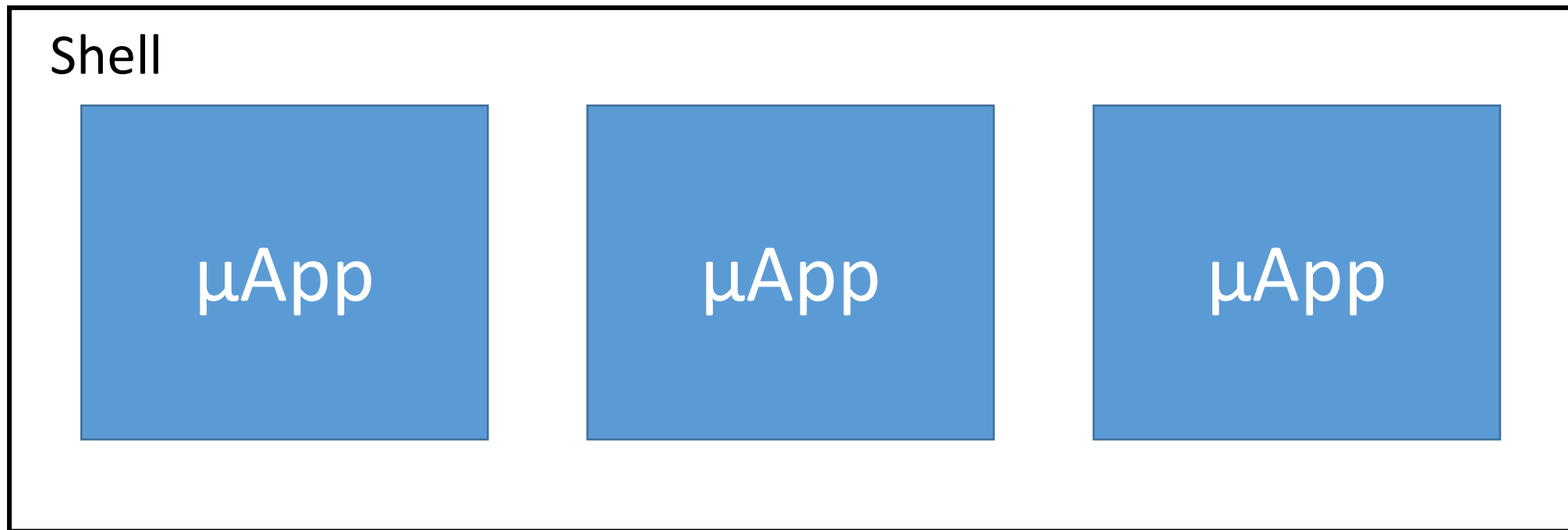




A photograph of a seashell on a sandy beach with the ocean in the background. A semi-transparent circular overlay is positioned on the right side of the image, containing the text 'Integration via Shell'.

Integration via
Shell

Providing a (SPA based) Shell



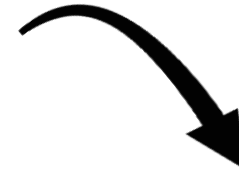
Webpack 5

Module Federation



Idea

Does not work with
webpack/ Angular CLI



```
const Component = import('http://other-app/xyz')
```

Even lazy parts must be
known at compile time!



Webpack 5 Module Federation

Shell (Host)

```
import('mfe1/Cmp')
```

```
// Maps Urls in  
// webpack config
```

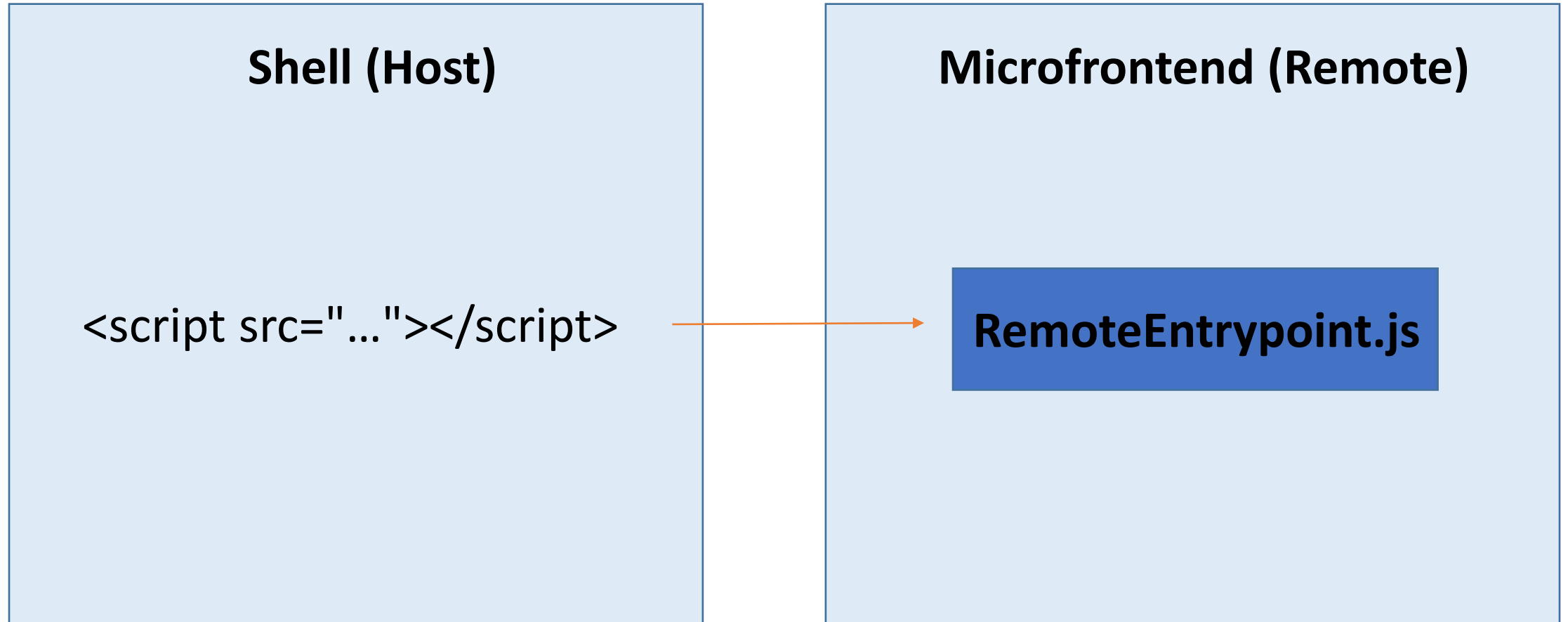
```
remotes: {  
  mfe1: "http://..."  
}
```

Microfrontend (Remote)

```
// Expose files in  
// webpack config
```

```
exposes: {  
  Cmp: './my.cmp.ts'  
}
```

How to Get the Microfrontend's URL?



How to Share Libs?

Shell (Host)

```
shared: [  
  "@angular/core", "..."  
]
```

Microfrontend (Remote)

```
shared: [  
  "@angular/core", "..."  
]
```

Dealing with Version Mismatches



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Default Behavior

Selecting the highest compatible version

~~10.0~~

10.1



Default Behavior

Conflict: No highest compatible version

11.0 ✓ 10.1 ✓

Example

- Shell: my-lib: ^10.0
- MFE1: my-lib: ^10.1
- MFE2: my-lib: ^9.0
- MFE3: my-lib: ^9.1

Result:

- Shell and MFE1 share ^10.1
- MFE2 and MFE3 share ^9.1

Configuring Singletons

```
shared: {  
  "my-lib": {  
    singleton: true  
  }  
}
```

11.0 ✓

~~10.1~~

Configuring Singletons

```
shared: {  
  "my-lib": {  
    singleton: true,  
    strictVersion: true // Error instead of warning!  
  }  
}
```

11.0 ✓ 10.1 ✗

Relaxing Version Requirements

```
shared: {  
  "my-lib": {  
    requiredVersion: ">=1.0.1 <11.1.1"  
  }  
}
```

Federated Angular: Angular, CLI, & Module Federation



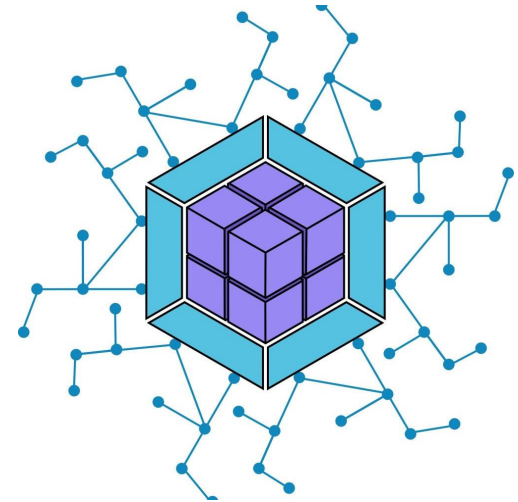
ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



webpack



Angular CLI

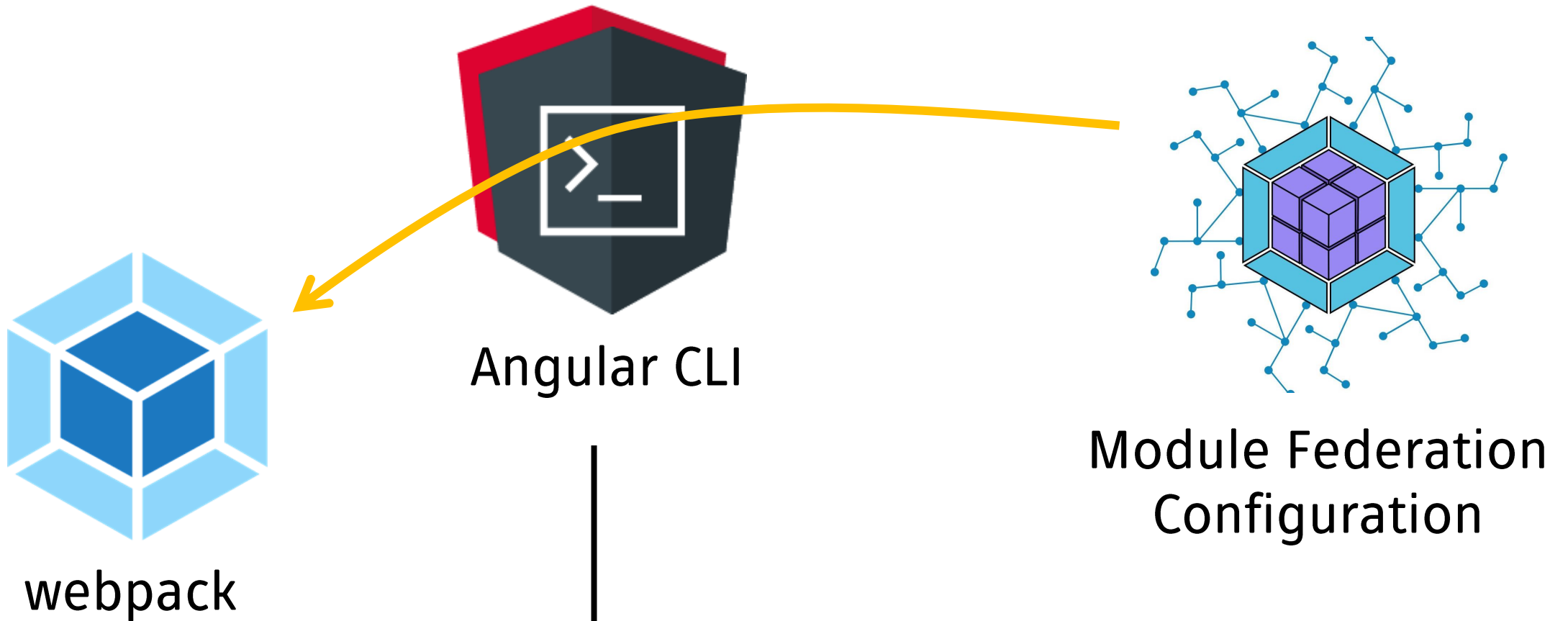


Module Federation
Configuration




@ManfredSteyer

Custom Builder



@angular-architects/module-federation

1.0.2 • Public • Published 18 hours ago

 [Readme](#)

 [Explore](#) BETA

 [3 Dependencies](#)

Features

- ✓ Generates the skeleton for a Module Federation config.
- ✓ Installs a custom builder to enable Module Federation.
- ✓ Assigning a new port to serve (`ng serve`) several projects at once.

Usage

- 1) `ng add @angular-architects/module-federation`
- 2) Adjust generated configuration
- 3) `ng serve`

Usage

- 1) `npm i @angular-architects/module-federation -D`
- 2) `ng g @angular-architects/module-federation:init`
- 3) Adjust generated configuration
- 4) `ng serve`

DEMO



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Multi Framework/ Version Solutions



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Abstracting Differences b/w SPA Frameworks

Wrap them into Web Components



Loading Web Components via Module Federation



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Module Federation

```
await import('other-app/web-cmp');
```

Module Federation

```
const rootElm = document.createElement('web-cmp')  
document.body.appendChild(rootElm);
```

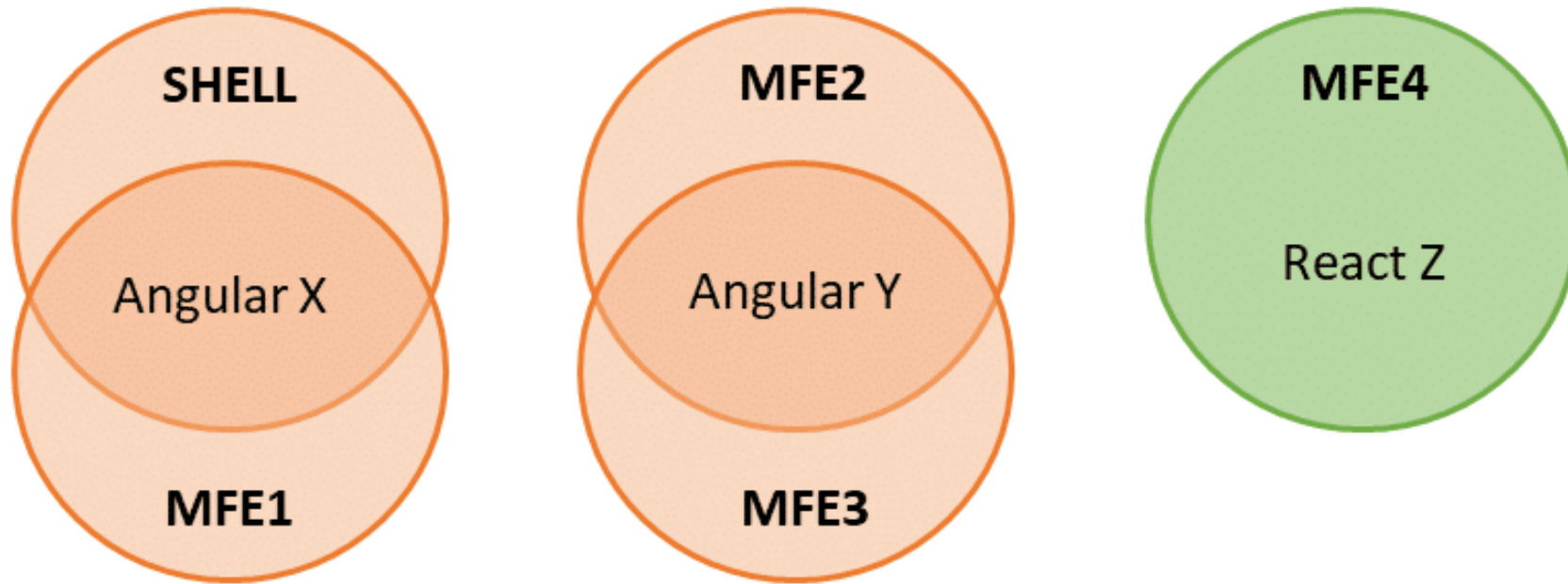
```
await import('other-app/web-cmp');
```

Routing to Another SPA?

WrapperComponent

```
const rootElm = document.createElement('web-cmp')  
document.body.appendChild(rootElm);  
  
await import('other-app/web-cmp');
```

Result



DEMO

<https://red-ocean-0fe4c4610.azurestaticapps.net>



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Challenges

- Bundle Size
- Multiple Routers
- Bootstrapping Several Angular Instances
 - Share Platform-Object when same version is reused
 - Share ngZone

Challenges & Solutions

@angular-architects/module-federation-tools TS

12.5.3 • Public • Published 21 days ago



Readme



Explore

BETA



1 Dependency



@ManfredSteyer



Choosing a Solution

Some General Advice

