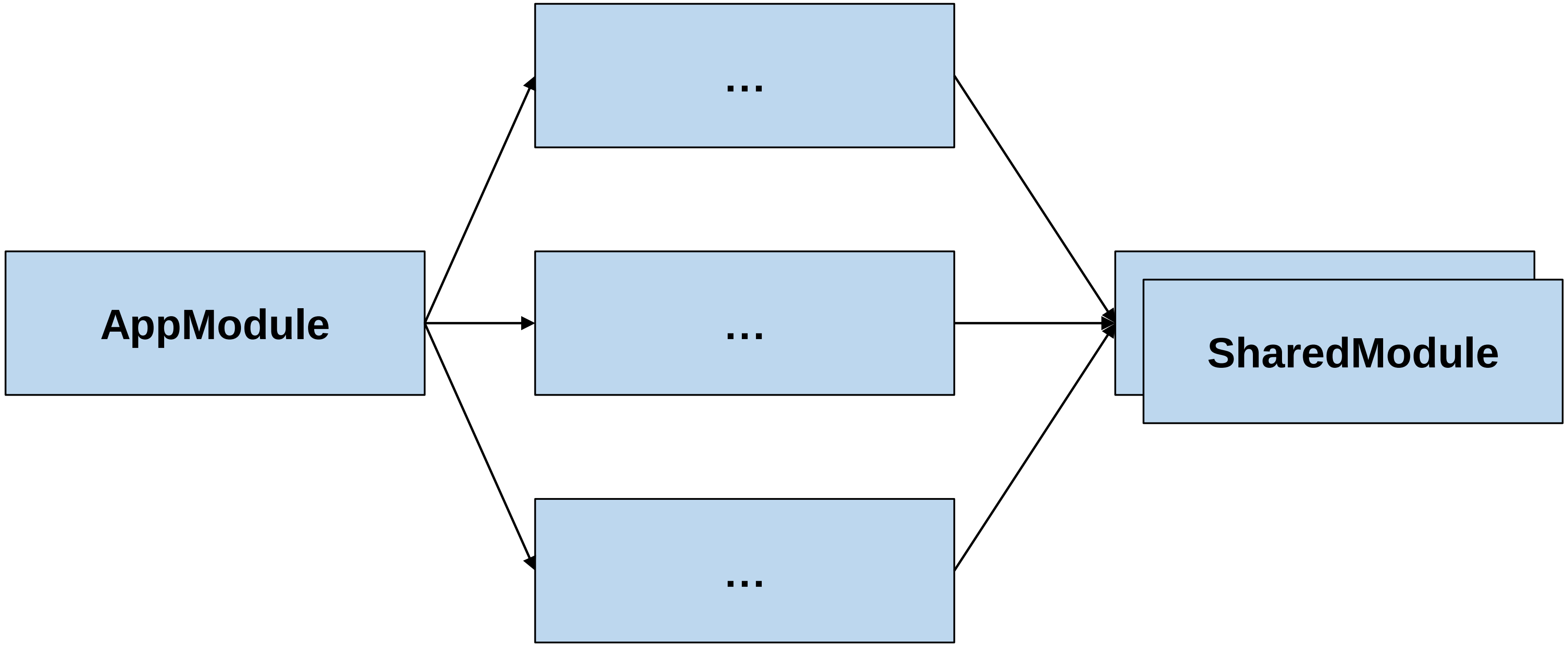
ANGULAR ARCHITECTS
INSIDE KNOWLEDGE

Angular Architektur-Workshop: Strategic Design and Micro Frontends

ANGULARarchitects.io

Typical Module Structure



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

Contents

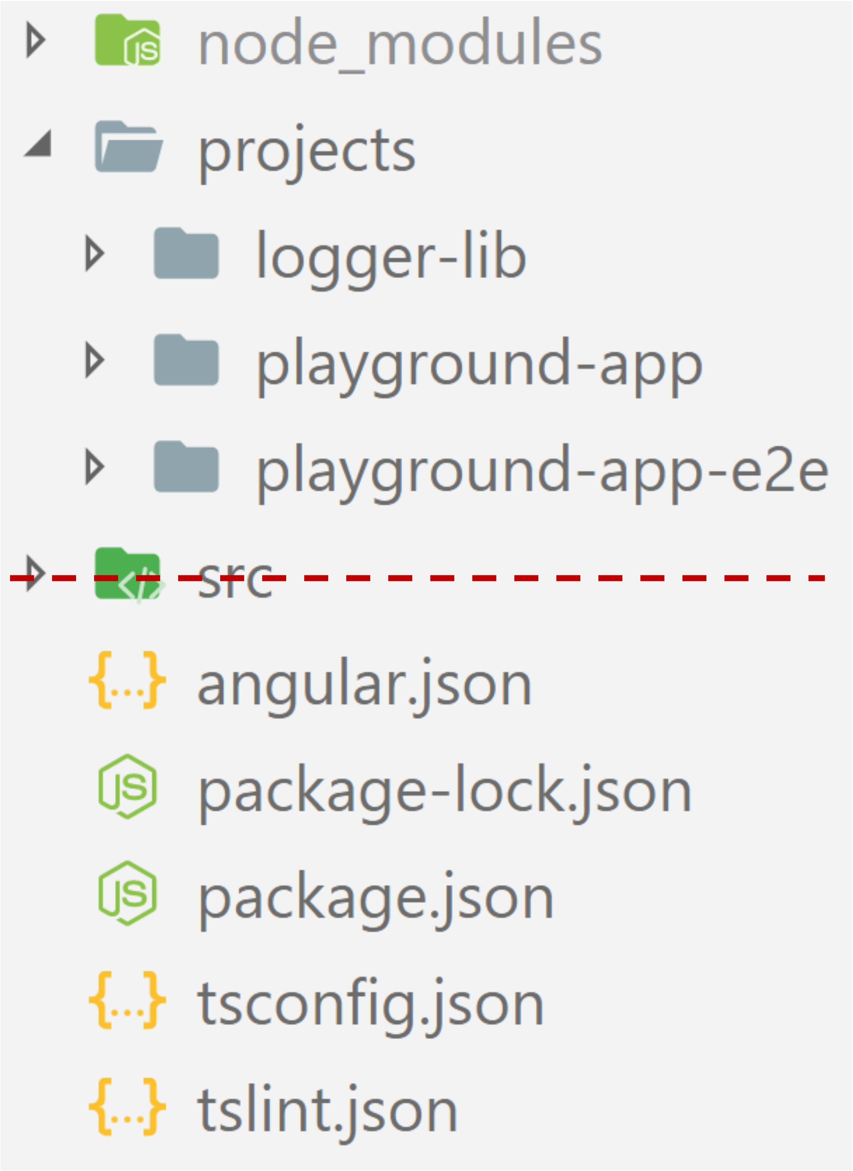
- (npm-)Packages
- Nx Monorepos
- Strategic Design and DDD
- Microfrontends

npm Packages

Create Library with CLI >= 6

```
npm install -g @angular/cli

ng new lib-project

cd lib-project

ng generate library logger-lib
ng generate application playground-app

ng serve --project playground-app
ng build --project logger-lib
```

Folder Structure



- node_modules
- projects
  - logger-lib
  - playground-app
  - playground-app-e2e
- ~~src~~
- angular.json
- package-lock.json
- package.json
- tsconfig.json
- tslint.json

Create Library with CLI >= 6

```
npm install -g @angular/cli

ng new lib-project --create-application false

cd lib-project

ng generate library logger-lib
ng generate application playground-app

ng serve --project playground-app
ng build --project logger-lib
```

Leerungszeiten
Deutsche Post
Tagesleerungen
Spätleerung
Nachtleerung
Montag - Freitag
08:00
Samstag
08:00
Sonntag
Service-Telefon der Briefkastenleerung:
0228 4333112
Briefkästen mit späteren Leerungen:
letzte Tagesleerung
Spätleerung
Nachtleerung
Montag - Freitag
Samstag
Sonntag

Publishing

Publishing to npm Registry

- Increment version in package.json
- ng build logger-lib --prod
- npm publish *dist/logger-lib* --registry http://localhost:4873
- npm install logger-lib --registry http://localhost:4873

Alternatives for setting the Registry

- Global: npm set registry http://localhost:4873
  - Default: registry.npmjs.org
  - npm get registry

- Project: .npmrc in project root

```
registry=http://localhost:4873/
```

```
@my-company:registry=http://my-server:4873/
```

npm Registries

Nexus

Artifactory

Team Foundation Server

Verdaccio

```
npm i -g verdaccio
verdaccio
```

DEMO

Advantages

- Distribution
- Versioning

Disadvantages

- Distribution
- Versioning

;-)

Disadvantages

Distribution

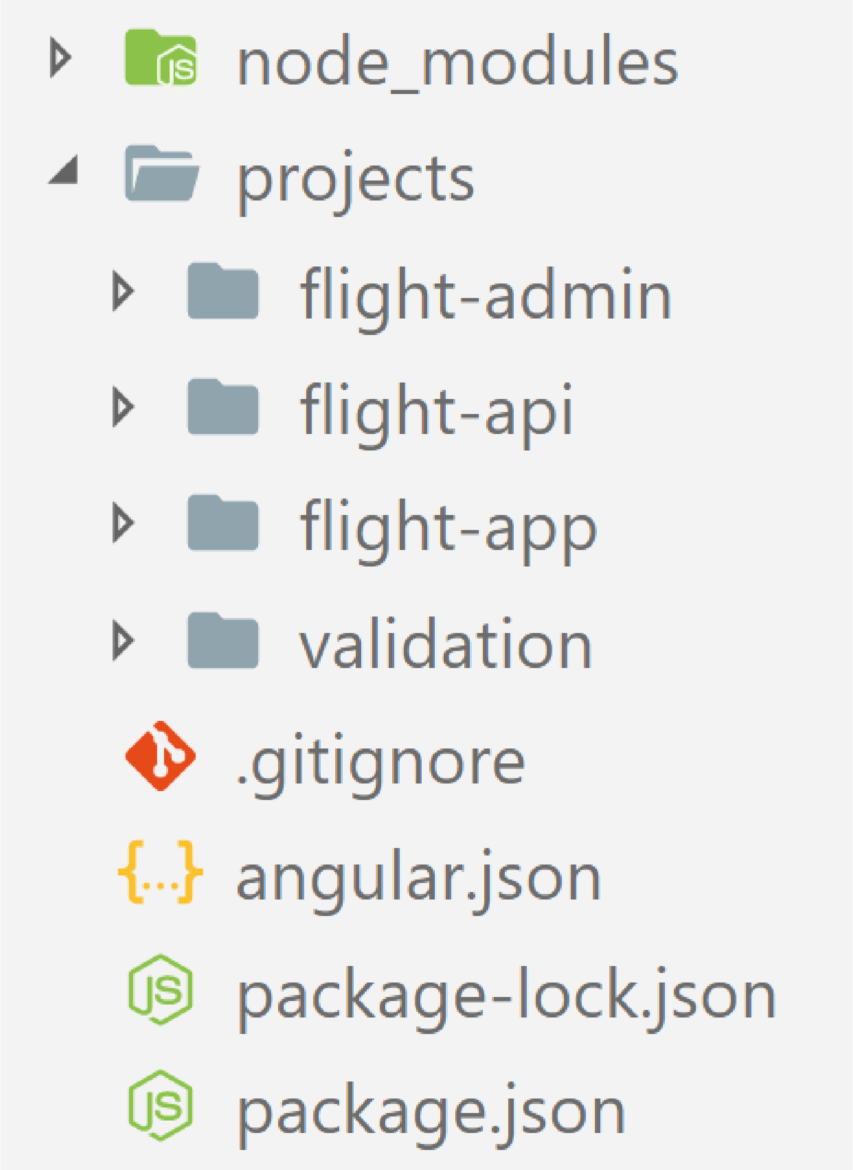
- Annoying within project
- Prevents gritting further libs

Versioning

- Old versions
- Conflicts
- How to force devs to use latest version?

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
- Creating new libs: Adding folder
- Experience: Successfully used at Google, Facebook, …

Two Flavors

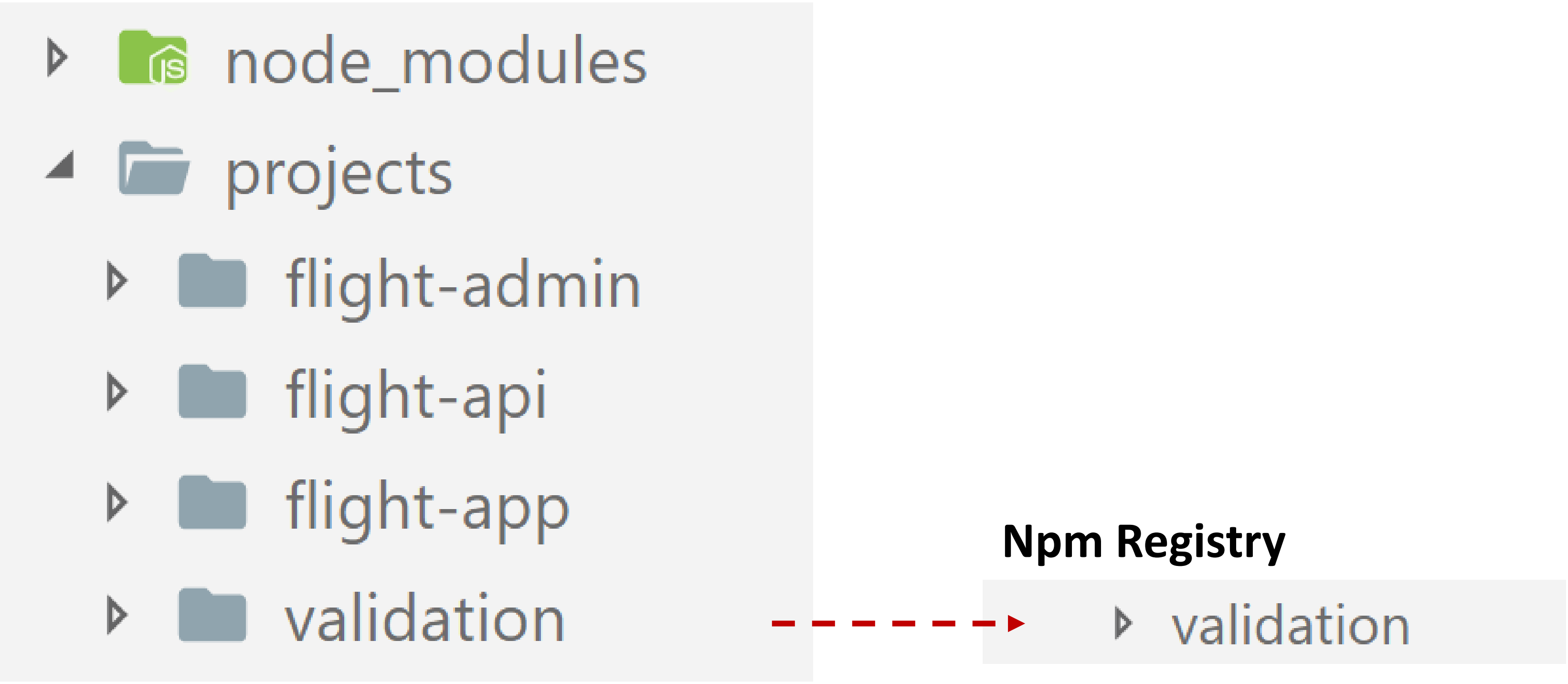
Project Monorepo

- Like Workspaces/Solutions in different IDEs

Company-wide Monorepo

- E. g. used at Google or Facebook

Moving back and forth



- node_modules
- projects
  - flight-admin
  - flight-api
  - flight-app
  - validation

Npm Registry

- validation

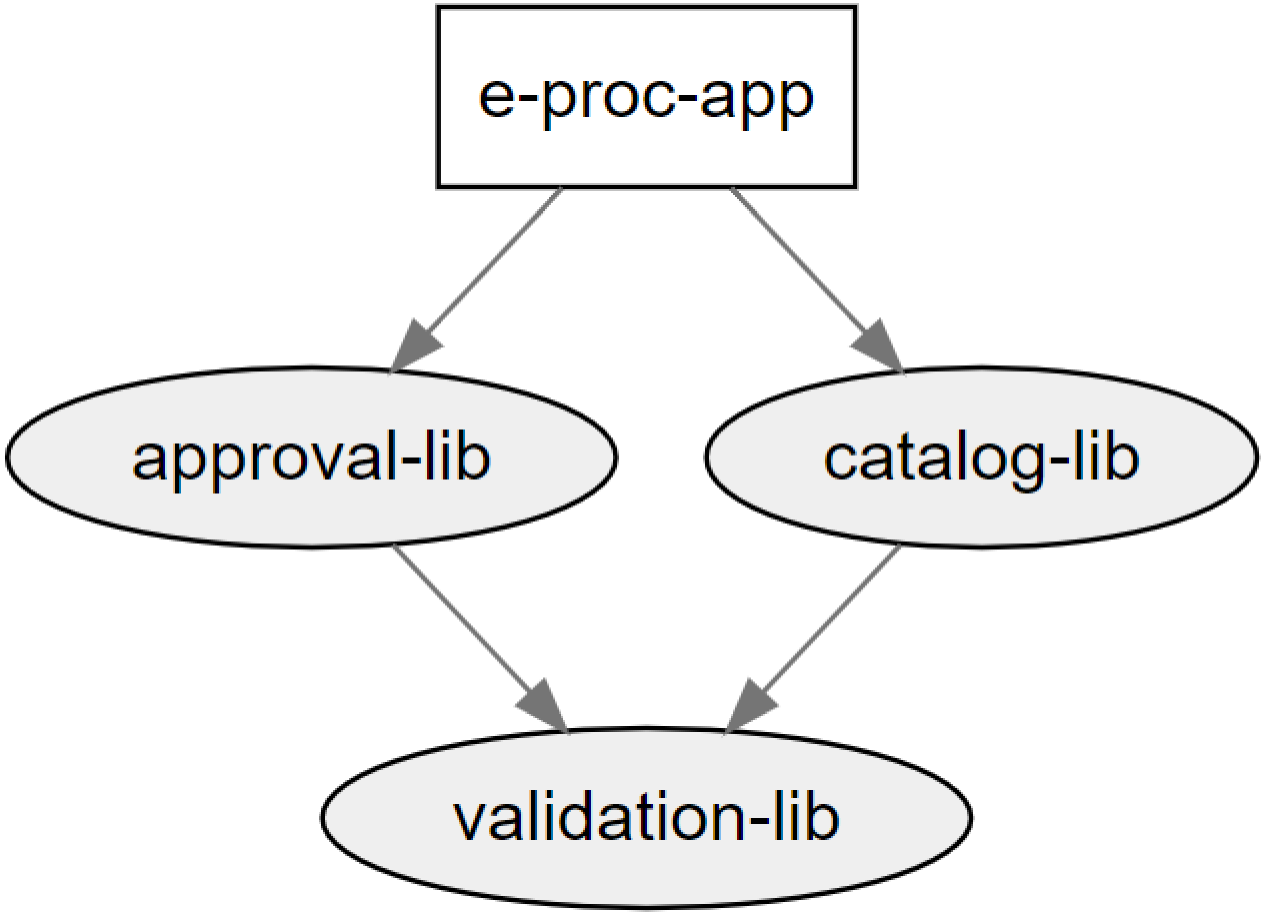
Tooling & Generator

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

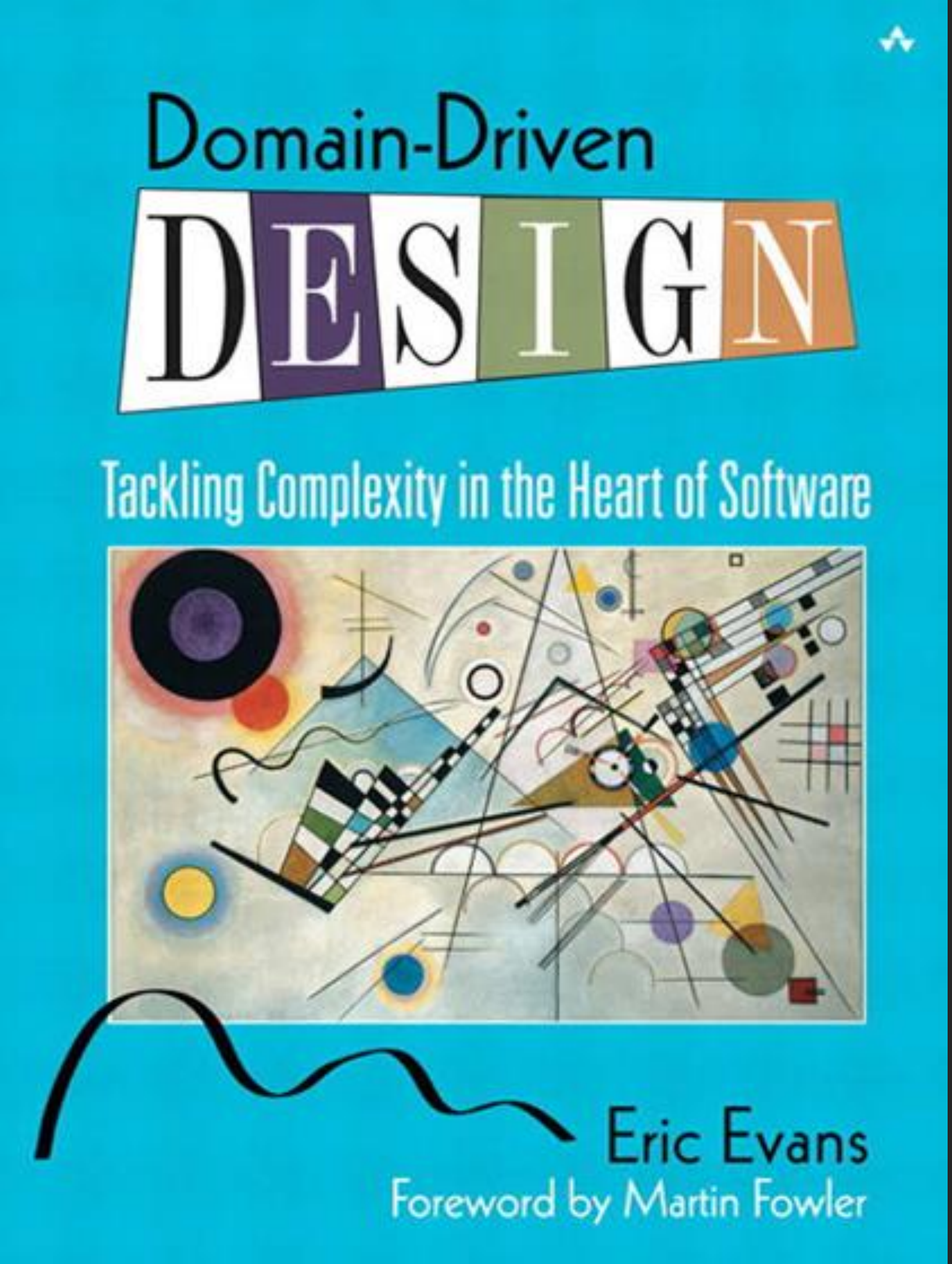
```
npm install -g @angular/cli

npm init nx-workspace workspace

cd workspace

ng generate app my-app
ng generate lib my-lib --buildable

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

DEMO

LAB

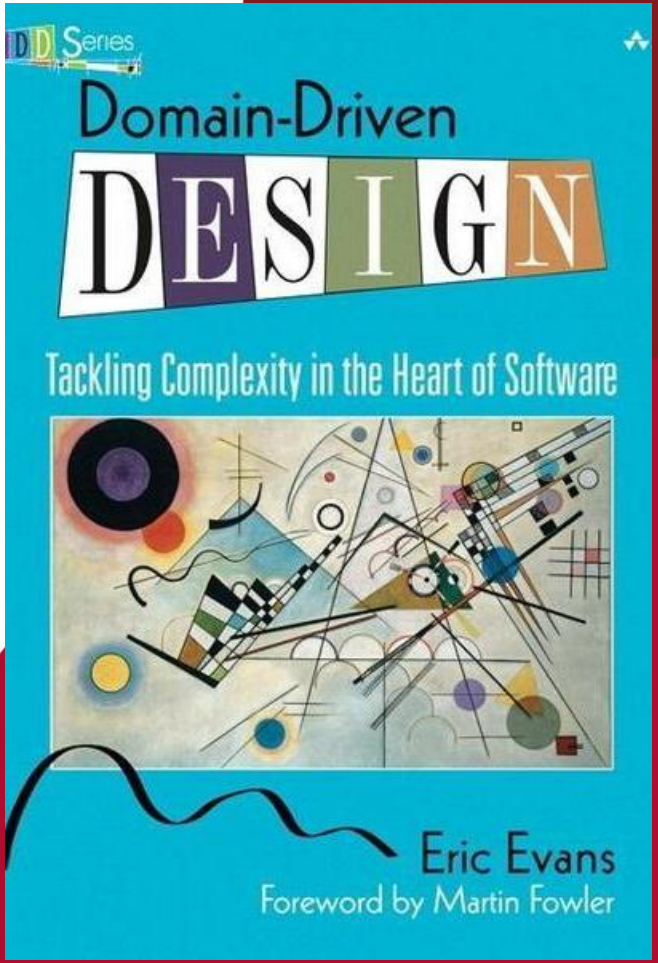
DDD

in a nutshell

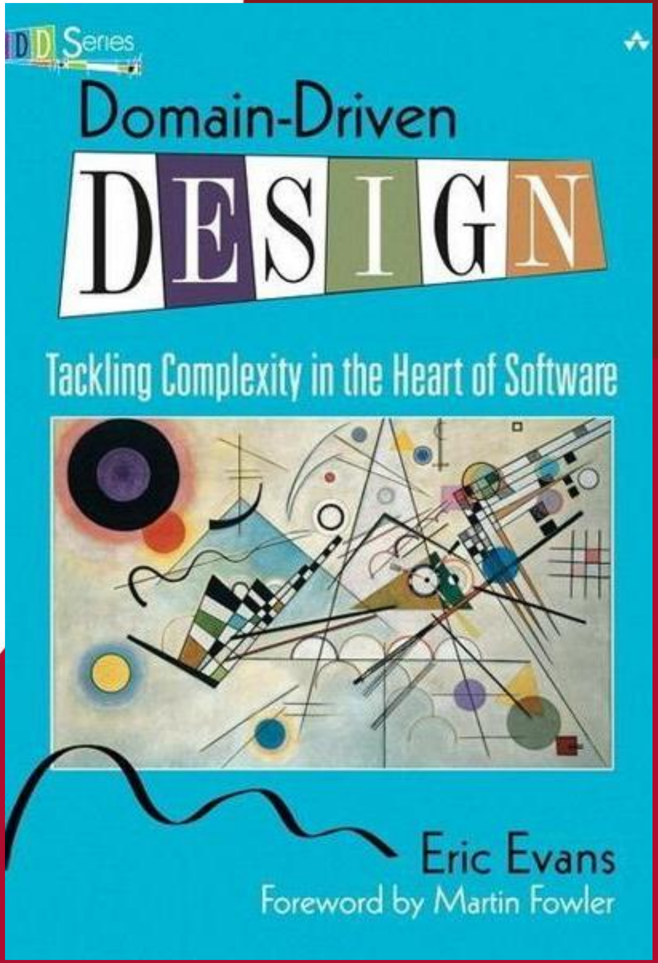

Domain-Driven
DESIGN
Tackling Complexity in the Heart of Software
Eric Evans
Foreword by Martin Fowler

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

How to create sustainable frontend architectures with ideas from DDD?


DDD Series
Domain-Driven
DESIGN
Tackling Complexity in the Heart of Software
Eric Evans
Foreword by Martin Fowler

How to create **sustainable** frontend architectures with **ideas from DDD**?


DDD Series
Domain-Driven
DESIGN
Tackling Complexity in the Heart of Software
Eric Evans
Foreword by Martin Fowler

Domain Driven Design

Decomposing a System

Design Patterns
& Practices

Strategic Design

Tactical Design

Domain Driven Design

Decomposing a System

Design Patterns
& Practices

Strategic Design

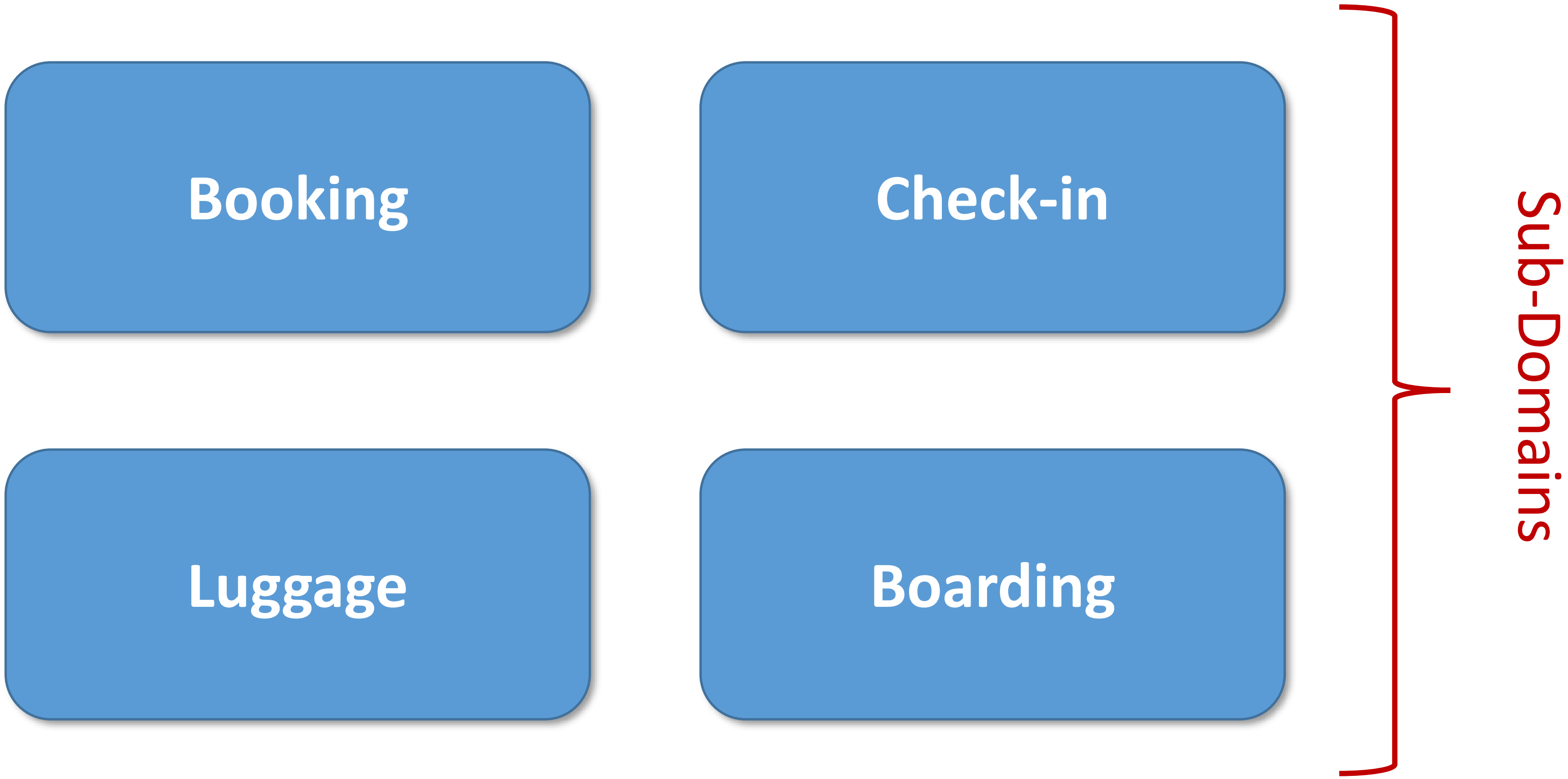
Tactical Design

This is what Strategic DDD prevents

Example

Flight System

Example



Booking

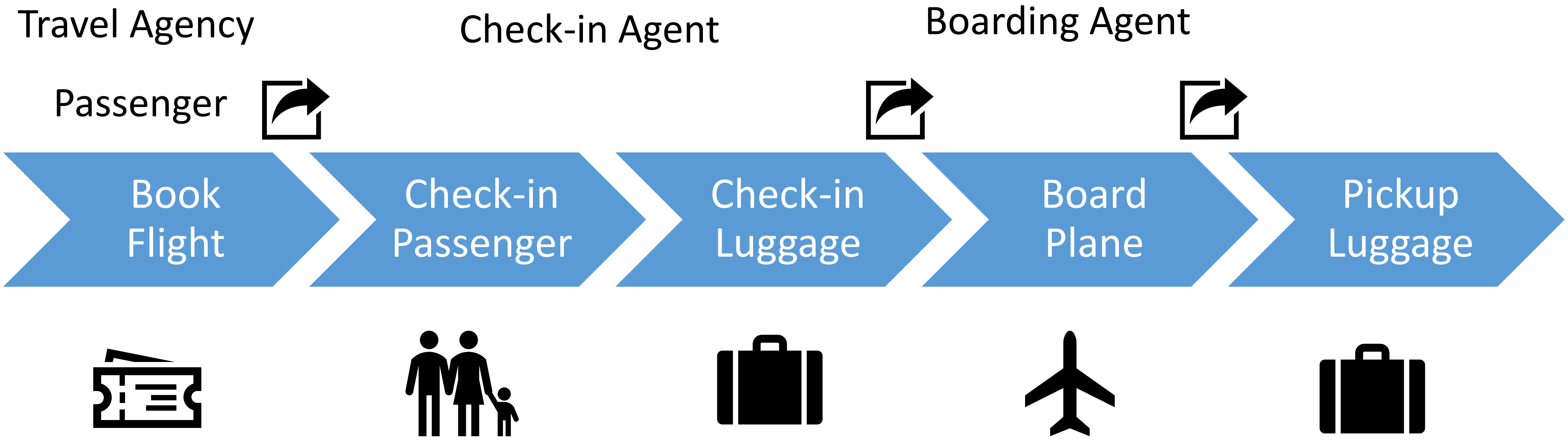
Check-in

Luggage

Boarding

Sub-Domains

Finding Sub-Domains



Travel Agency

Passenger

Check-in Agent

Boarding Agent

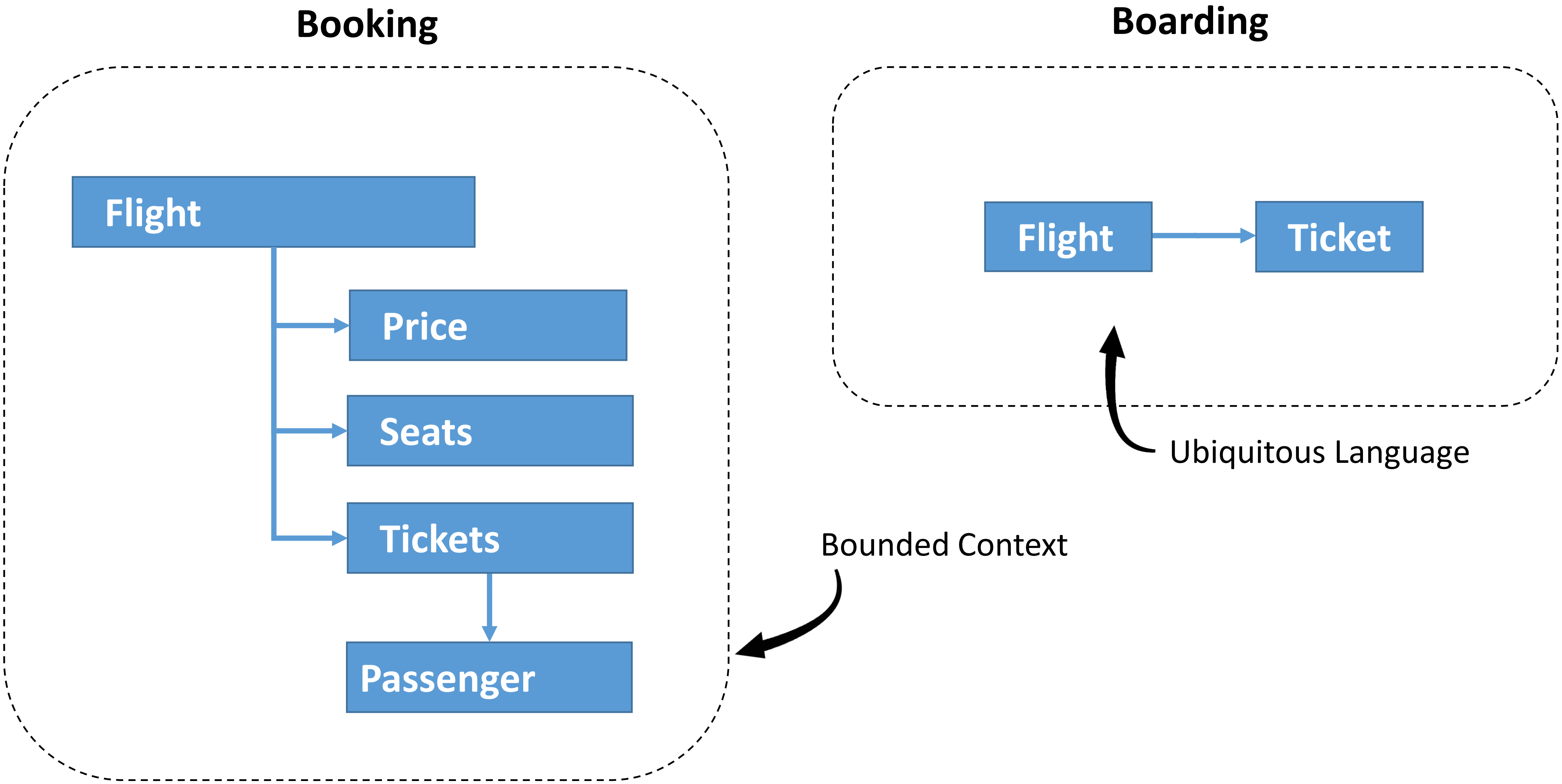
Book Flight

Check-in Passenger

Check-in Luggage

Board Plane

Pickup Luggage



Booking

Flight

Price

Seats

Tickets

Passenger

Boarding

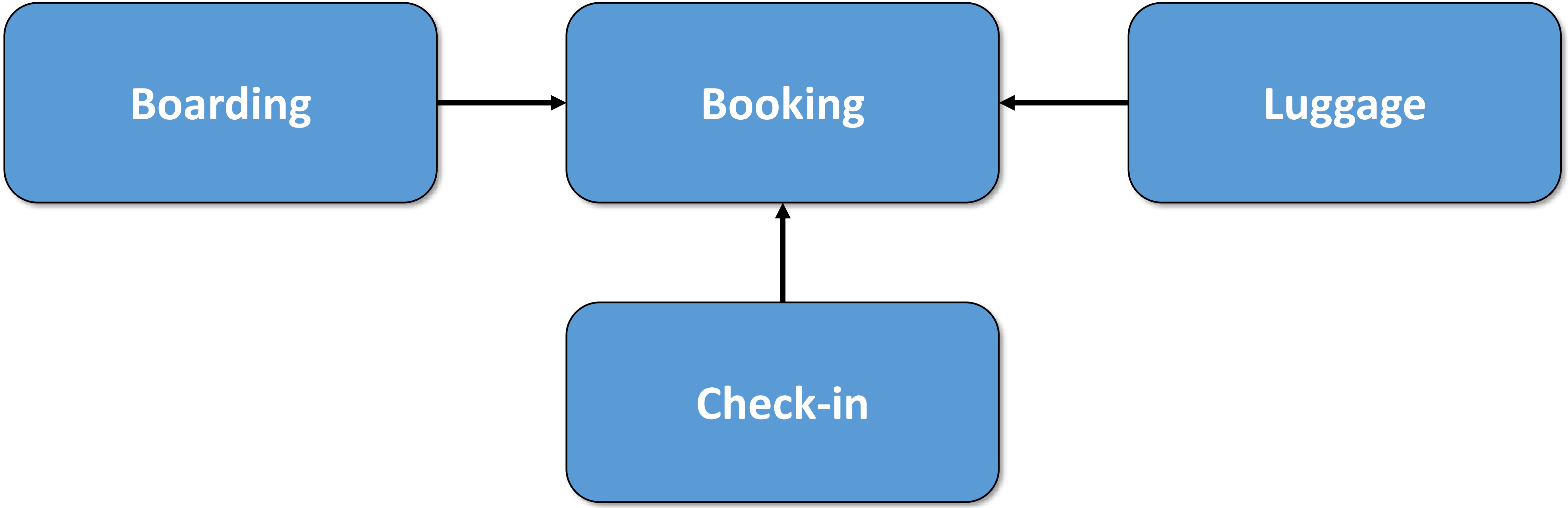
Flight

Ticket

Ubiquitous Language

Bounded Context

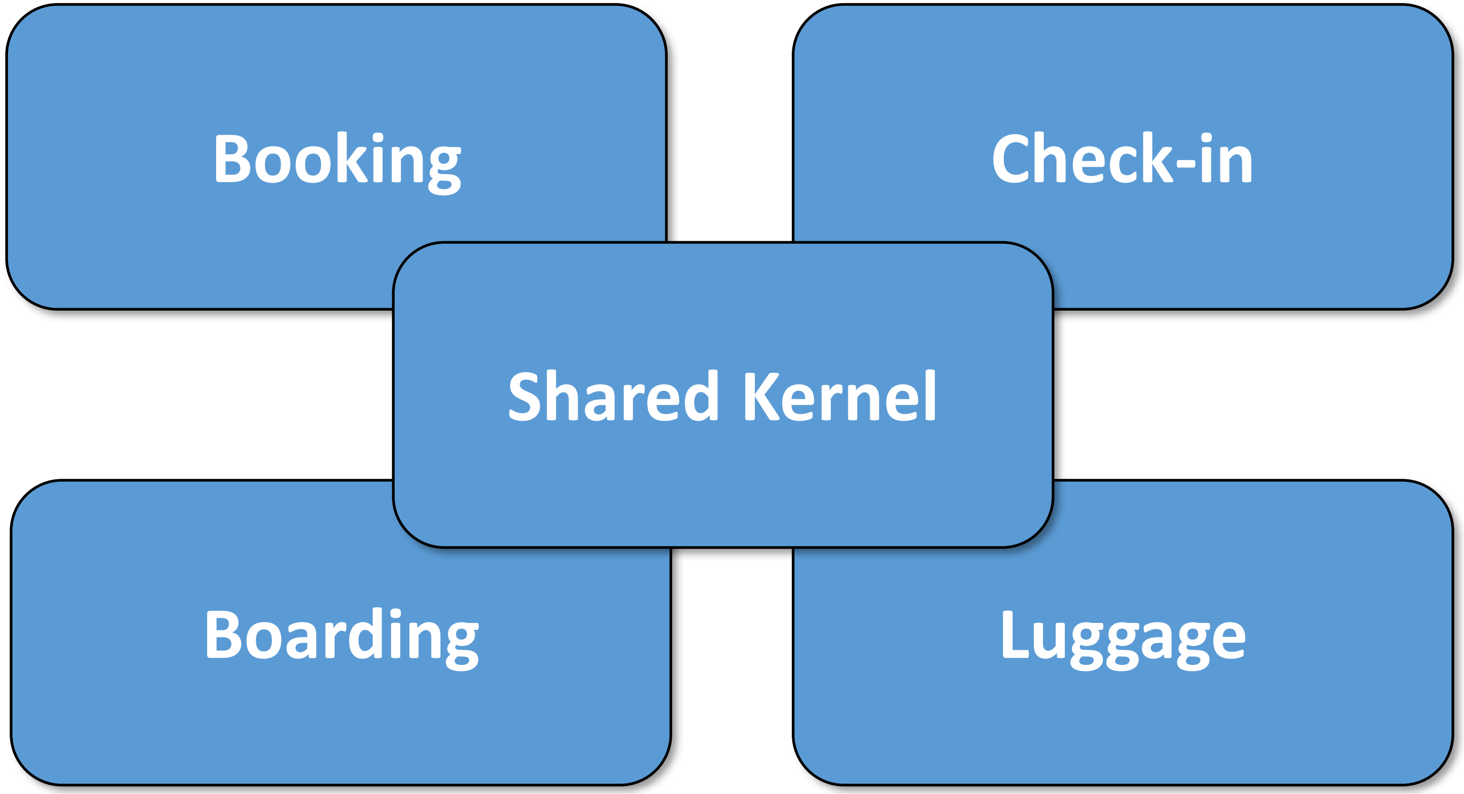
Context Map

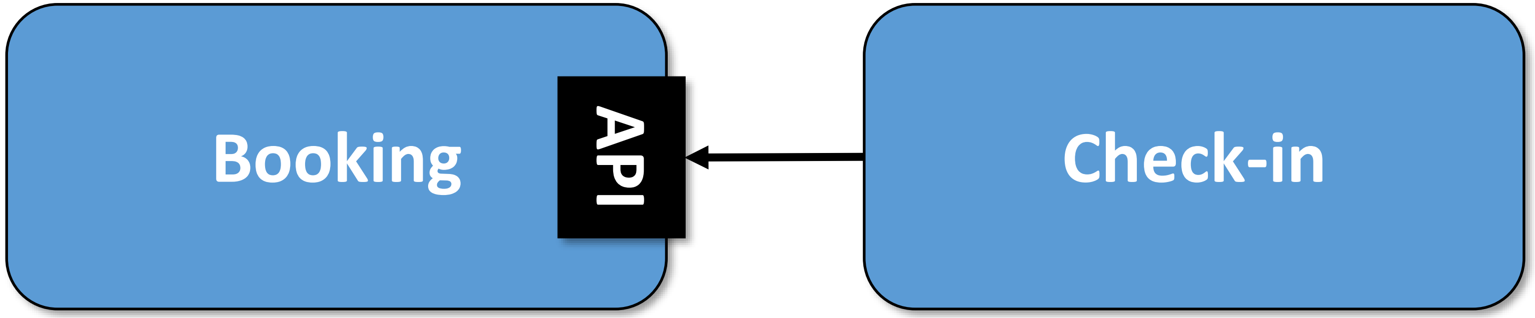


Context Map

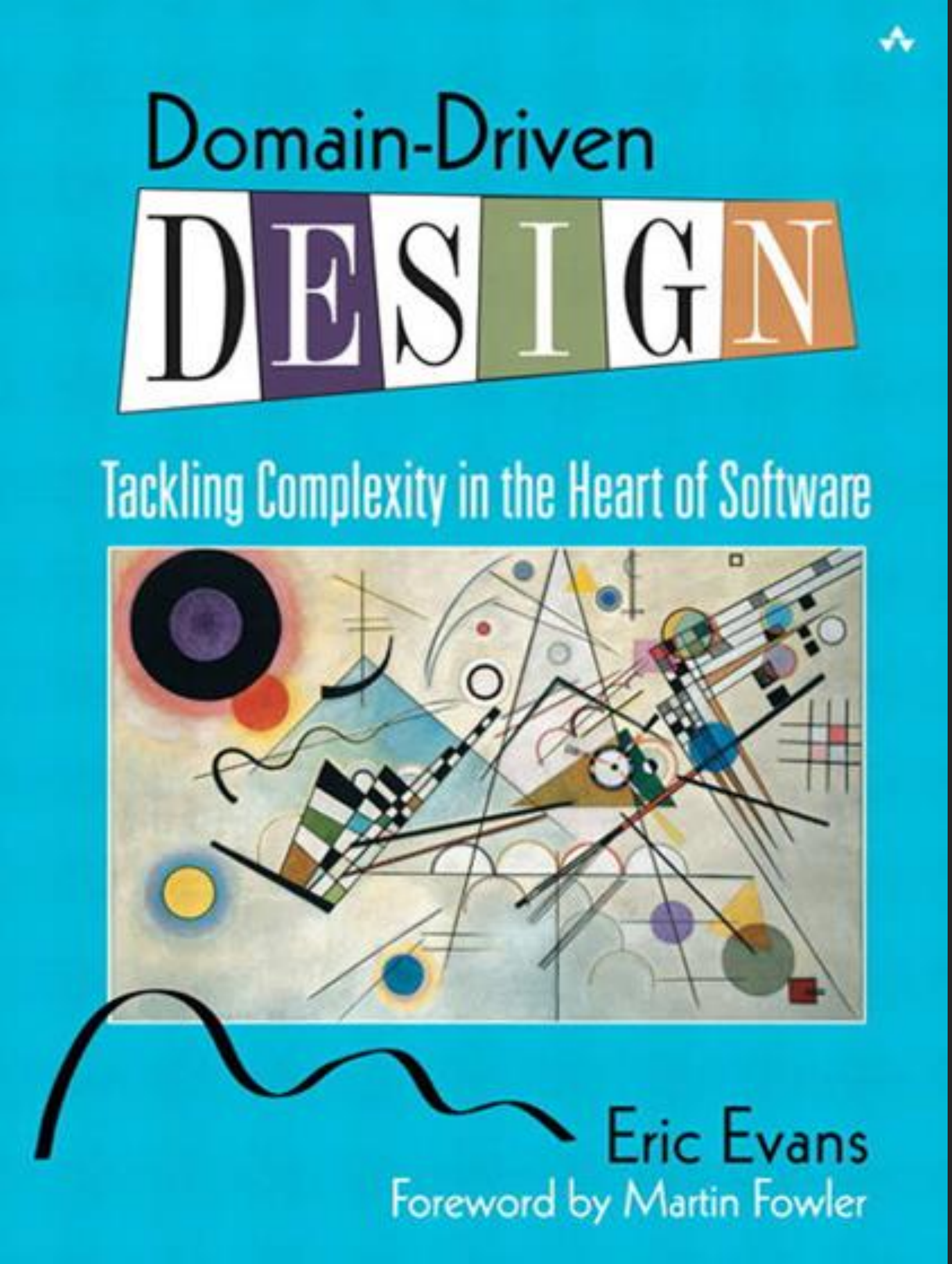
Responsibilities?

Breaking Changes?

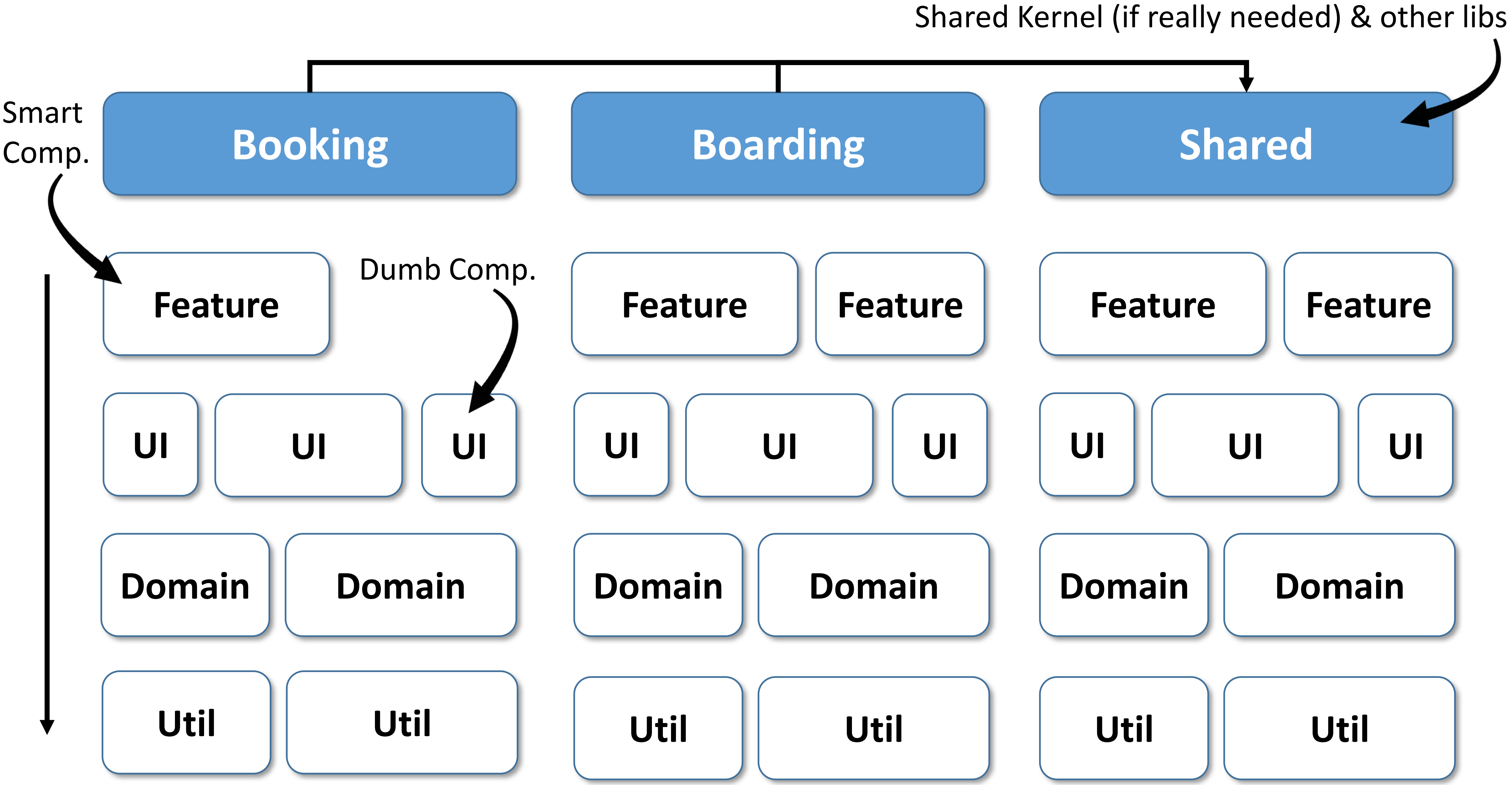




Open-/Host-Service


Domain-Driven
DESIGN
Tackling Complexity in the Heart of Software
Eric Evans
Foreword by Martin Fowler

Lots of approaches for cross-domain communication and more ...



Shared Kernel (if really needed) & other libs

Smart Comp.

Dumb Comp.

Booking

Feature

UI UI UI

Domain Domain

Util Util

Boarding

Feature Feature

UI UI UI

Domain Domain

Util Util

Shared

Feature Feature

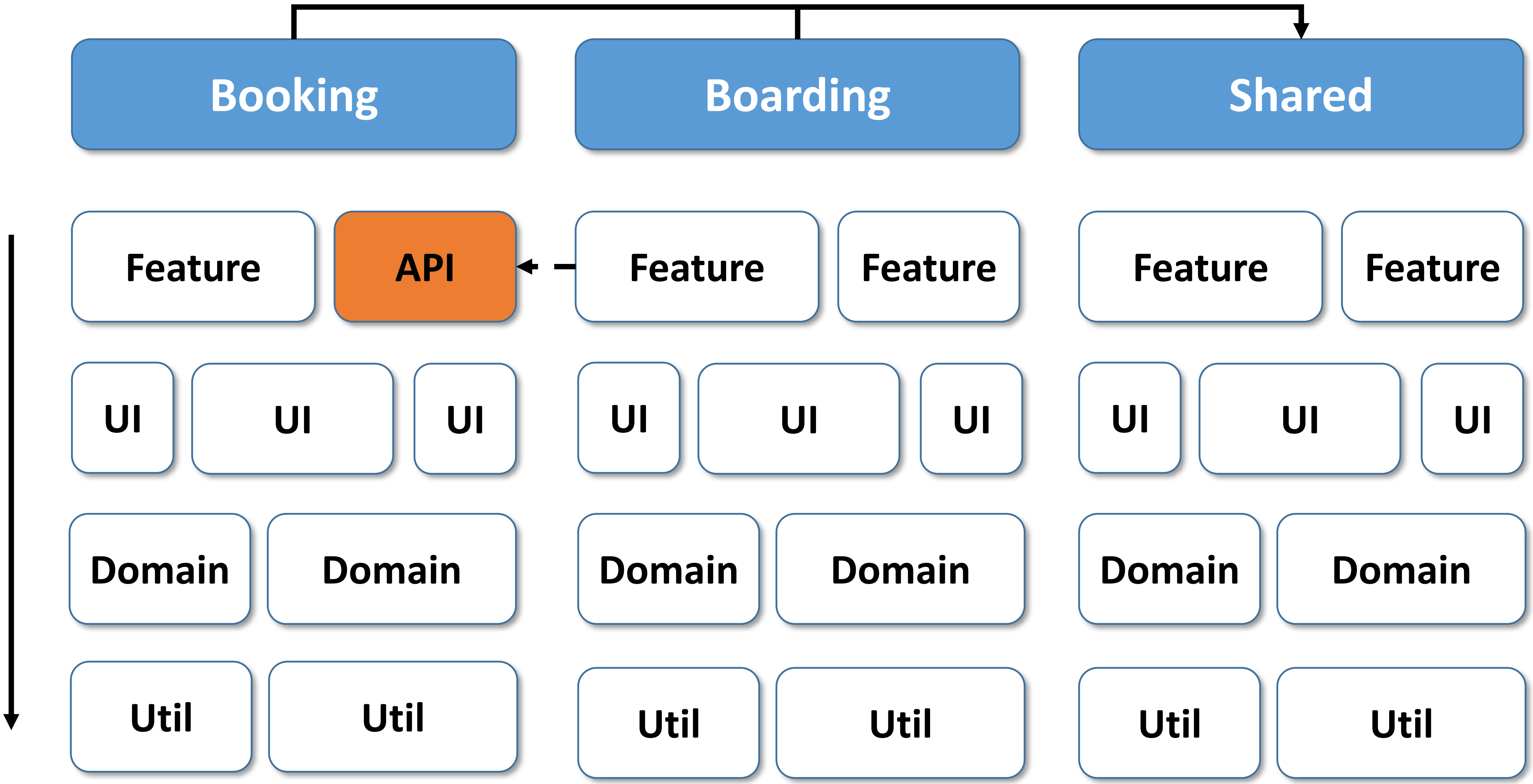
UI UI UI

Domain Domain

Util Util

@ManfredSteyer

Enterprise Monorepo Patterns, Nrwl 2018: https://tinyurl.com/y2jjxld7



Booking

Boarding

Shared

Feature API

Feature Feature

Feature Feature

UI UI UI

UI UI UI

UI UI UI

Domain Domain

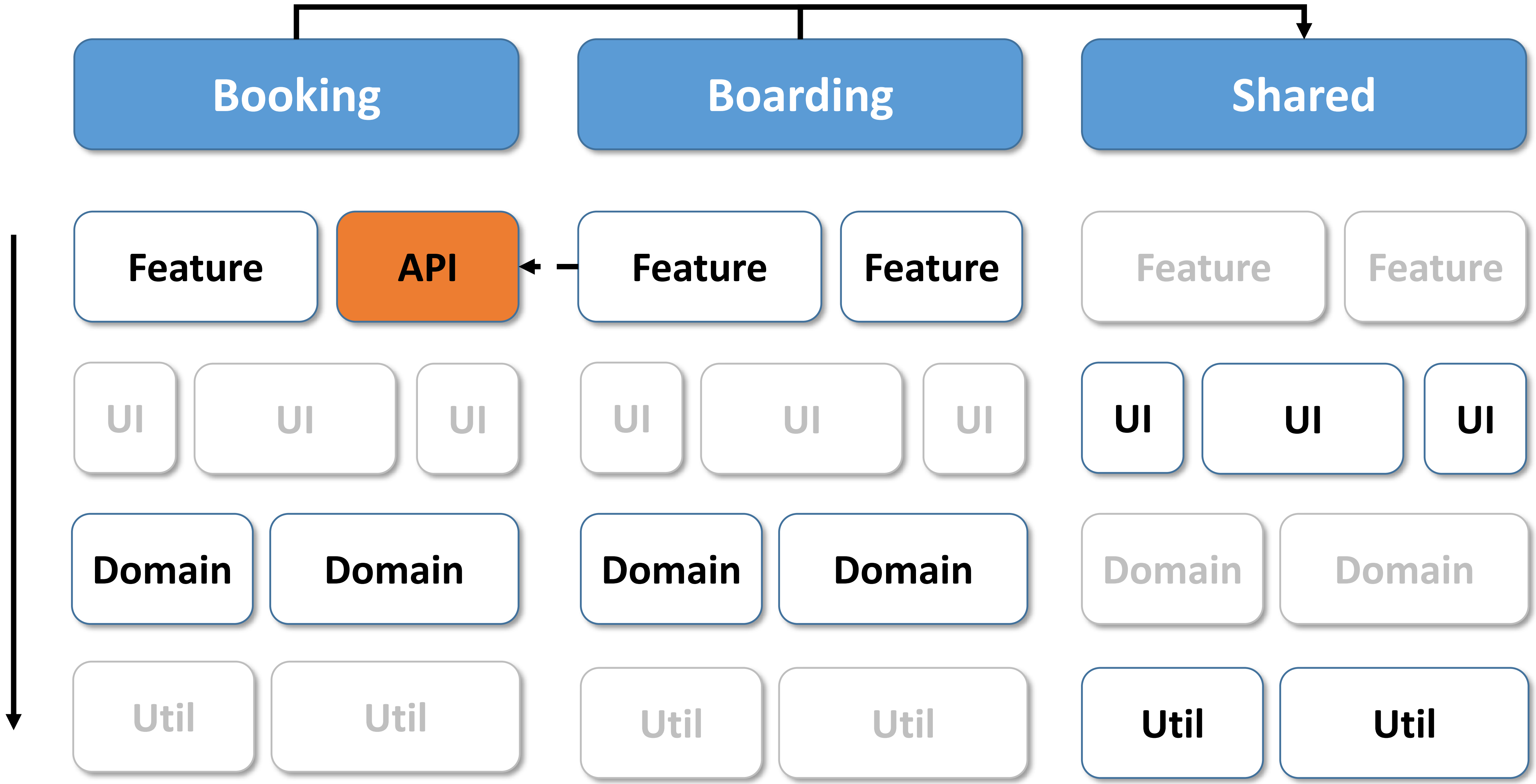
Domain Domain

Domain Domain

Util Util

Util Util

Util Util



Booking

Boarding

Shared

Feature

API

Feature

Feature

Feature

Feature

UI

UI

UI

UI

UI

UI

UI

UI

UI

Domain

Domain

Domain

Domain

Domain

Domain

Util

Util

Util

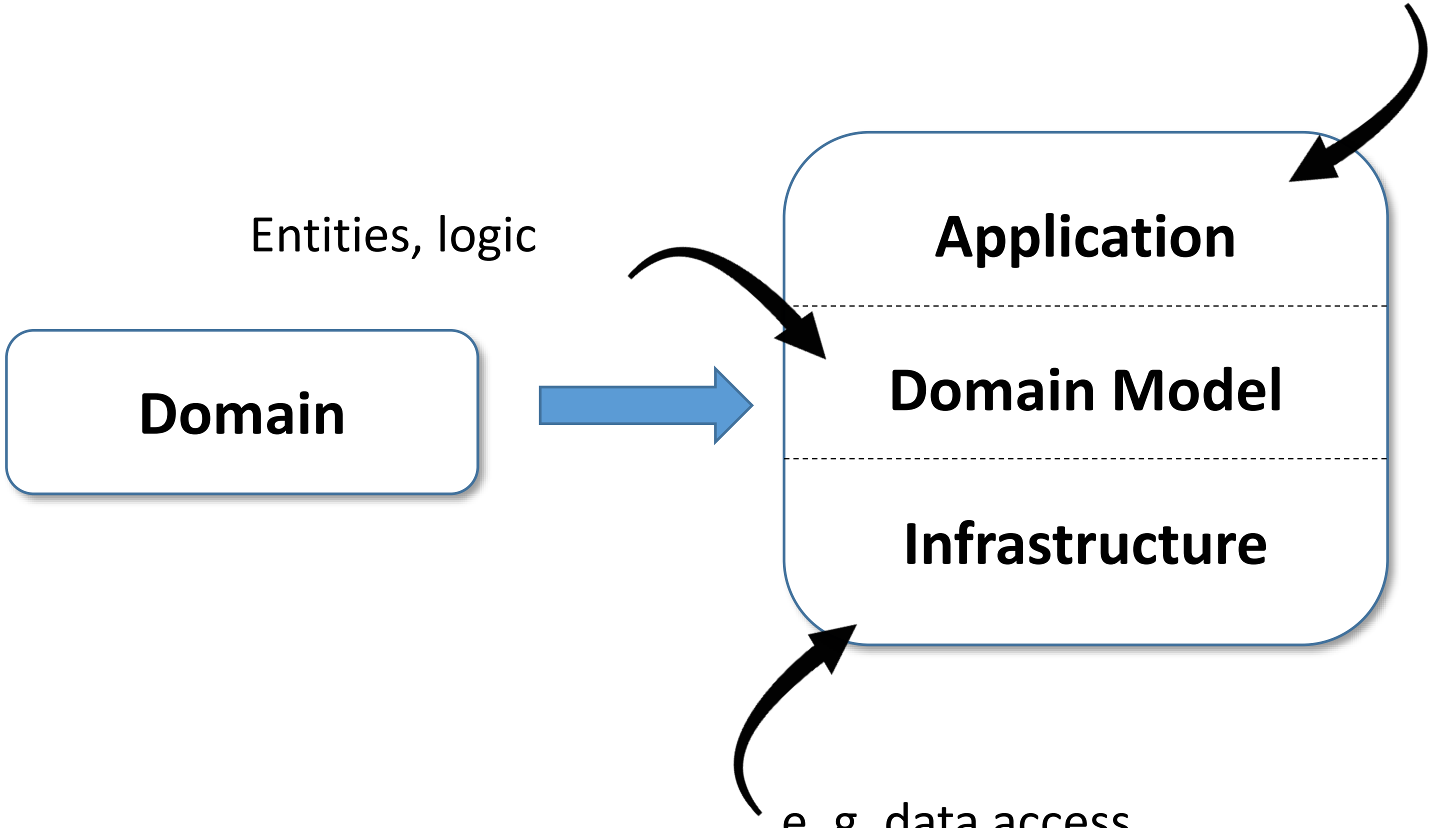
Util

Util

Util

@ManfredSteyer

Isolate your domain!



Use case specific facades, state management

e. g. data access

Alternatives to layering

- e. g. Hexagonal Architecture, Clean Architecture
- Anyway: We need to **restrict access** b/w libraries

DEMO

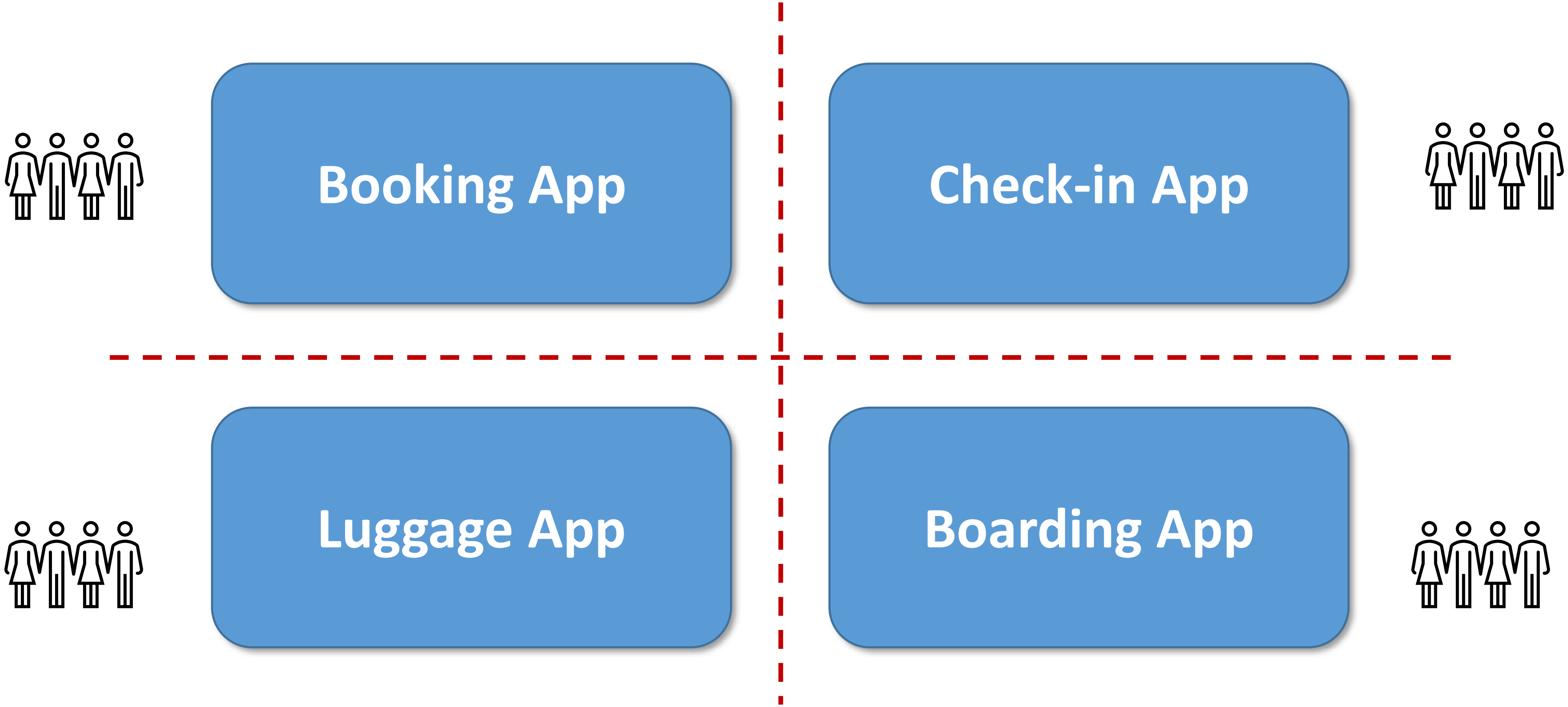
Finegrained Libraries

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

Micro Frontends?

Short outlook

Microfrontends



Booking App

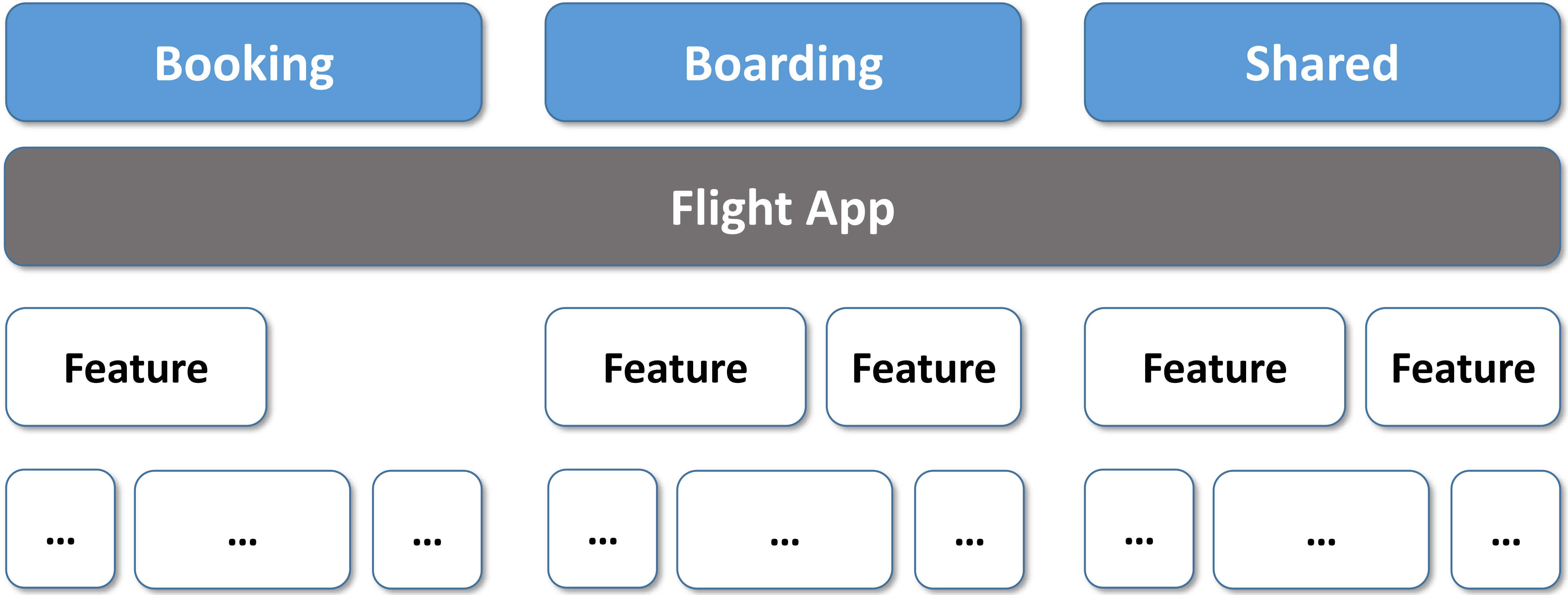
Check-in App

Luggage App

Boarding App

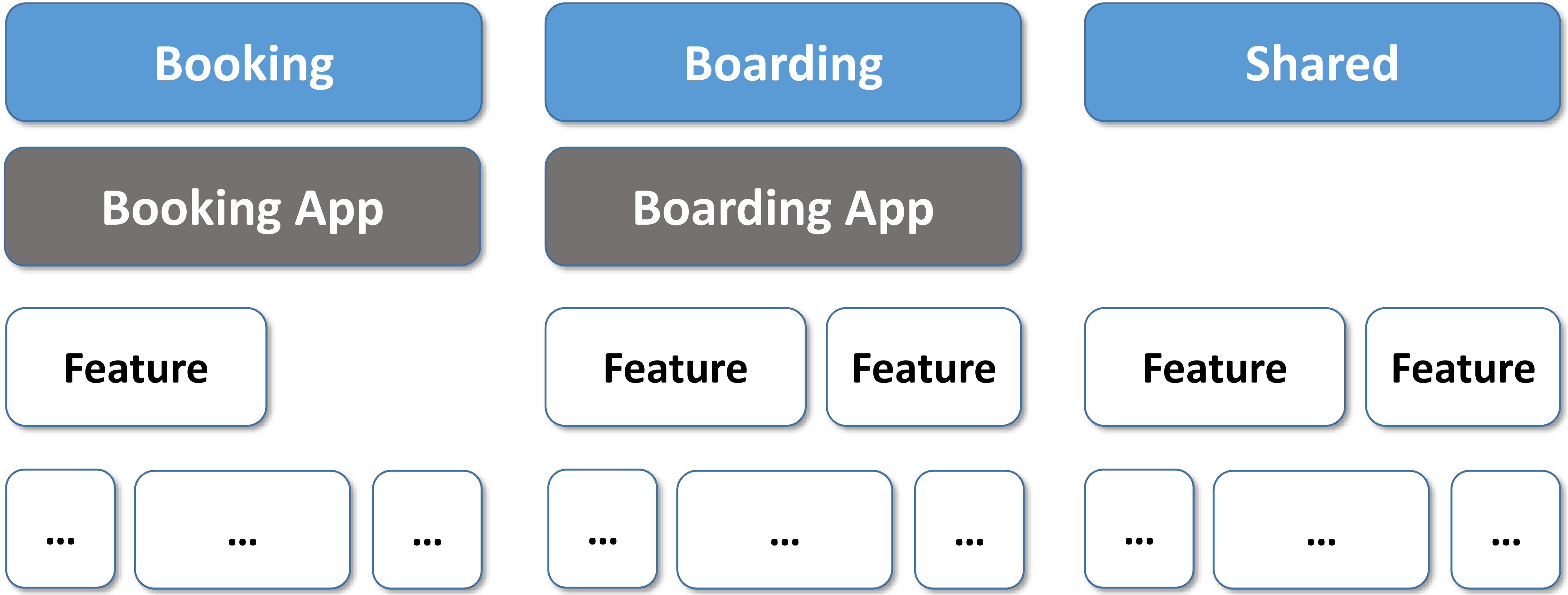
Microfrontends
are first and foremost
about **scaling teams!**

Deployment Monolith



@ManfredSteyer

Microfrontends



Booking

Booking App

Feature

... ...

Boarding

Boarding App

Feature Feature

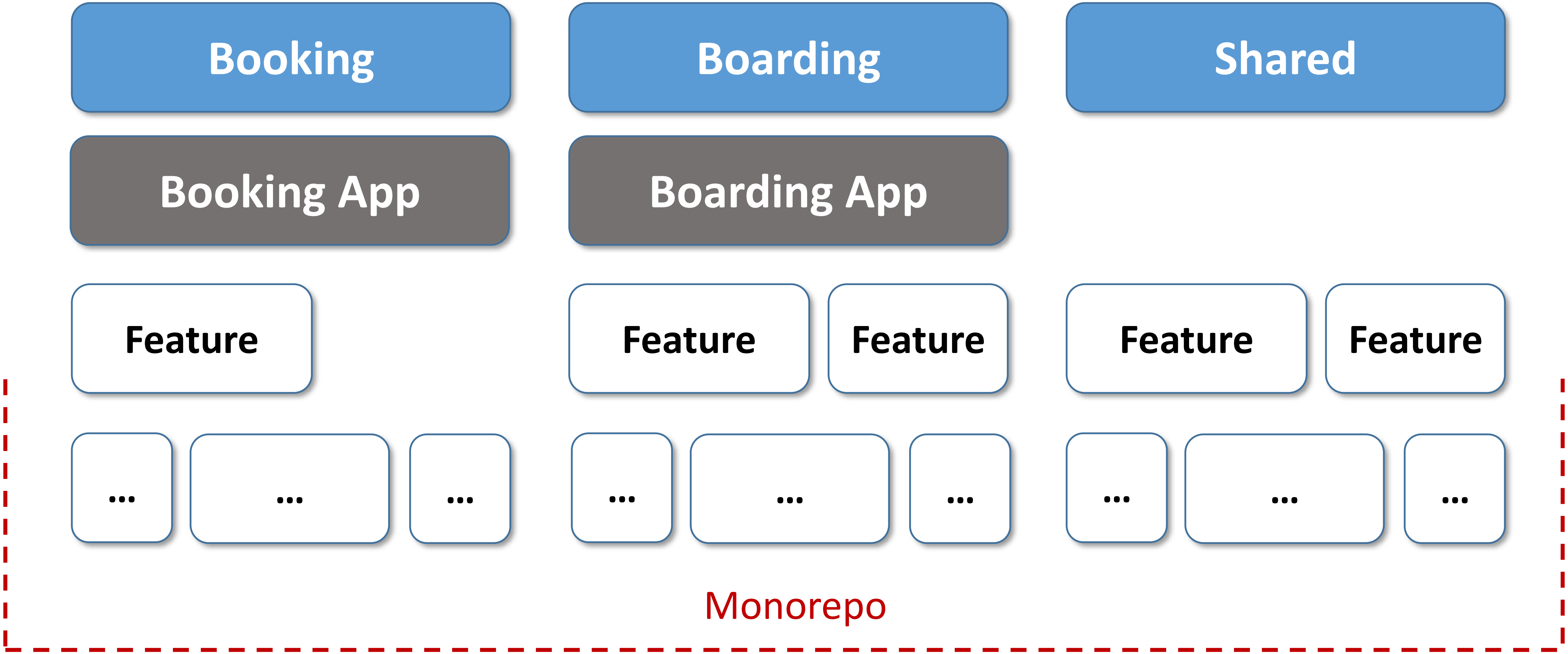
... ...

Shared

Feature Feature

... ...

Option 1: One App per Domain



Booking

Boarding

Shared

Booking App

Boarding App

Feature

Feature

Feature

Feature

Feature

...

...

...

...

...

...

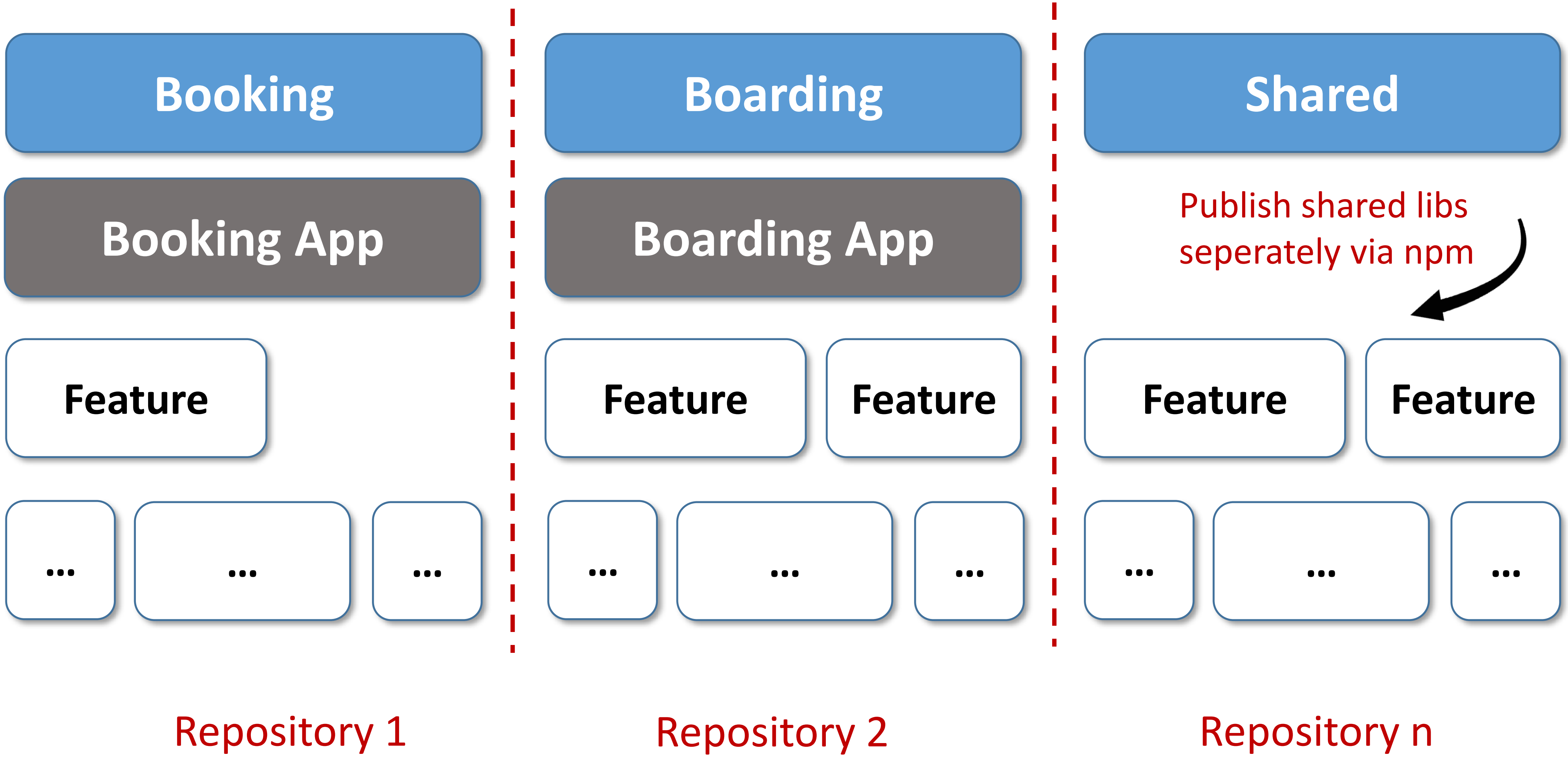
...

...

...

Monorepo

Option 2: One Monorepo per Domain



Publish shared libs seperately via npm

Repository 1

Repository 2

Repository n

Benefits

Autonomous Teams

Separate Development

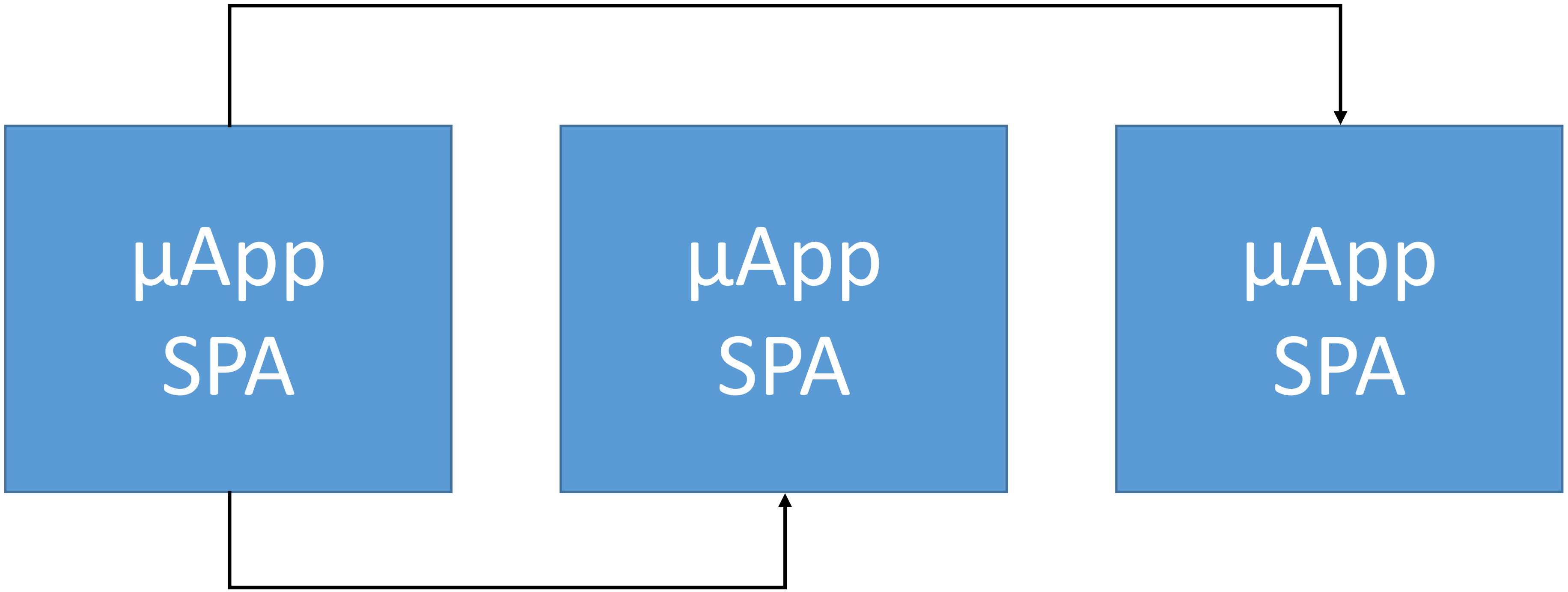
Separate Deployment

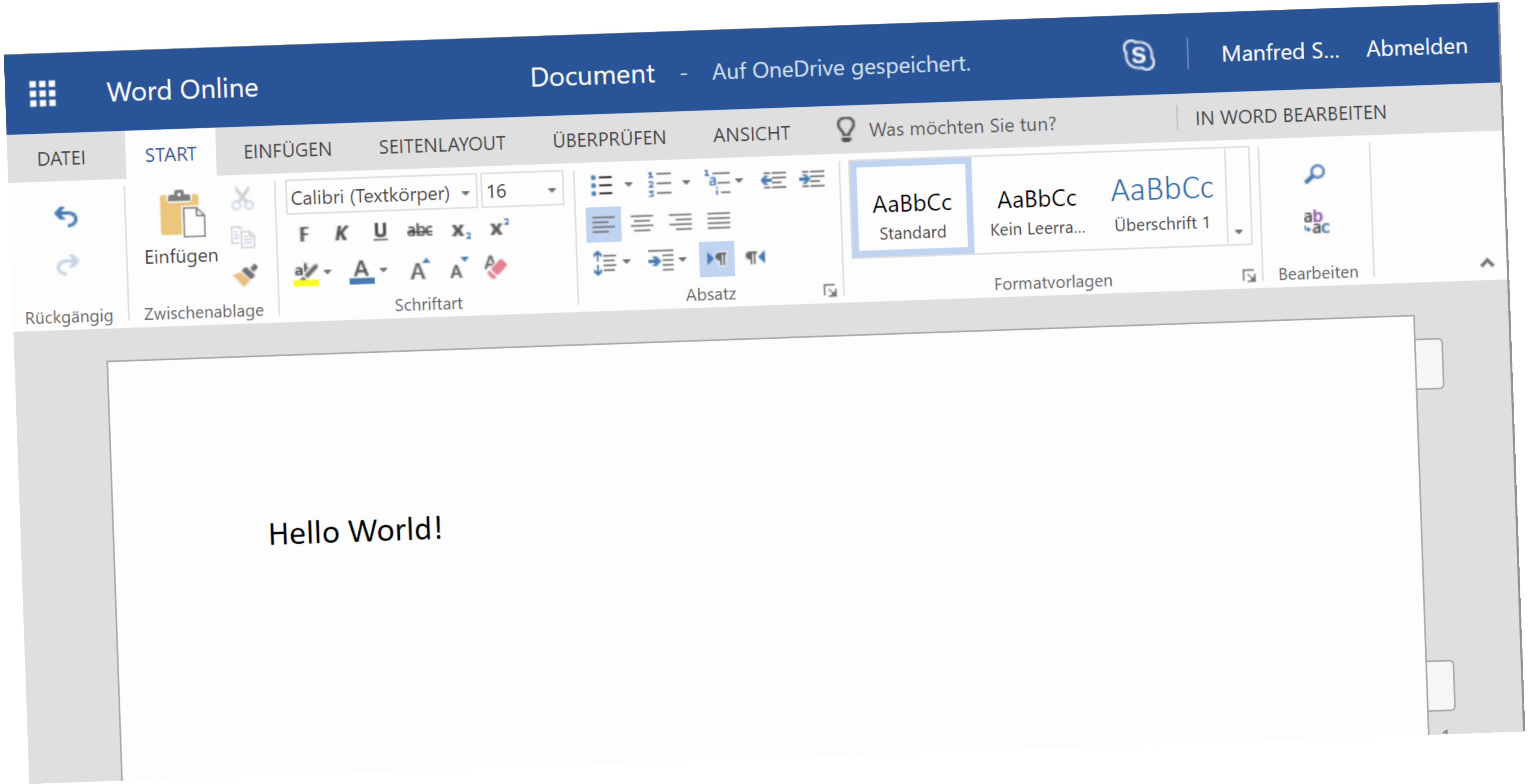
Own architecture decisions

Own technology descisions

Integration via Hyperlinks

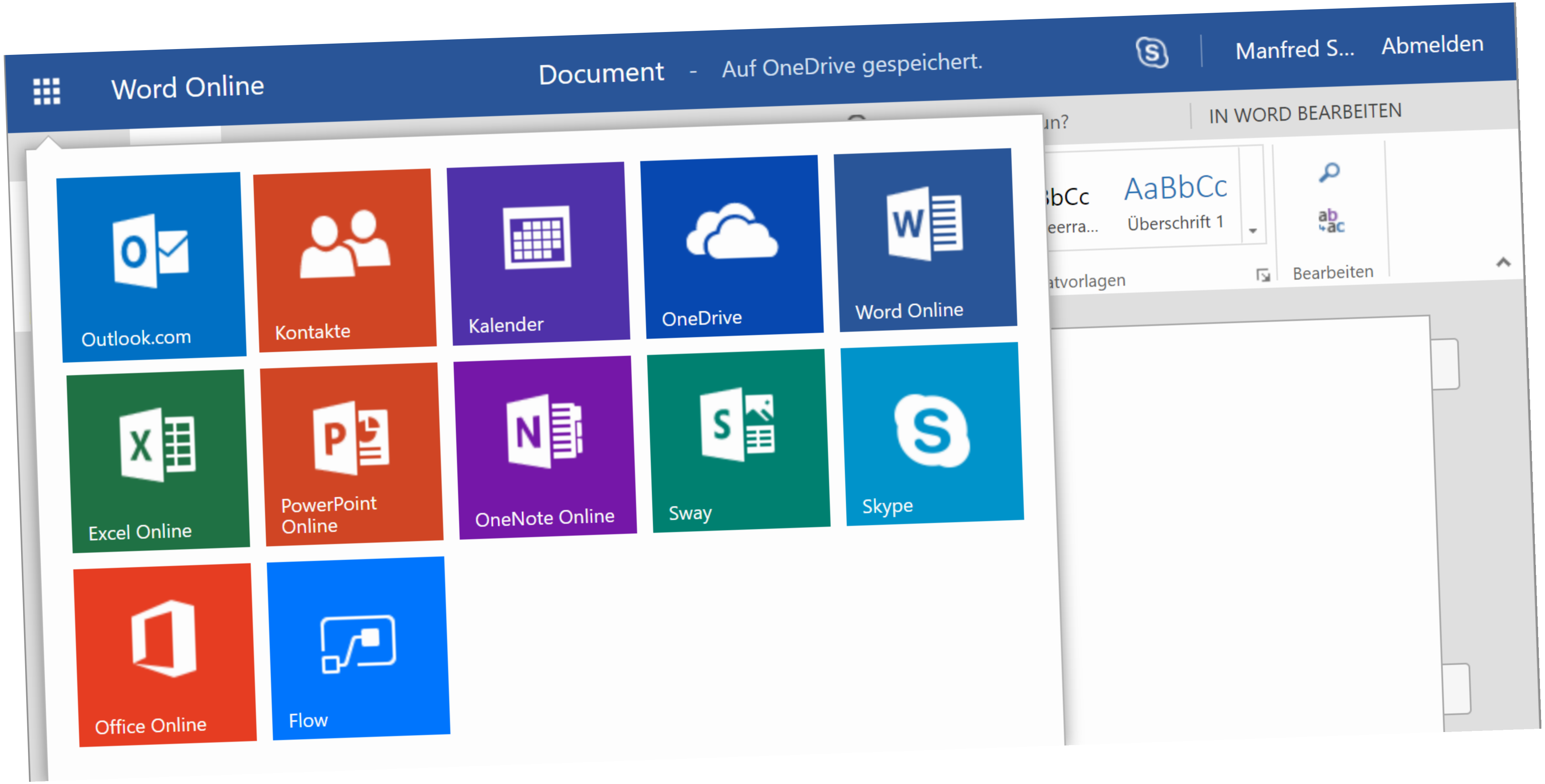
UI Composition w/ Hyperlinks





Hello World!

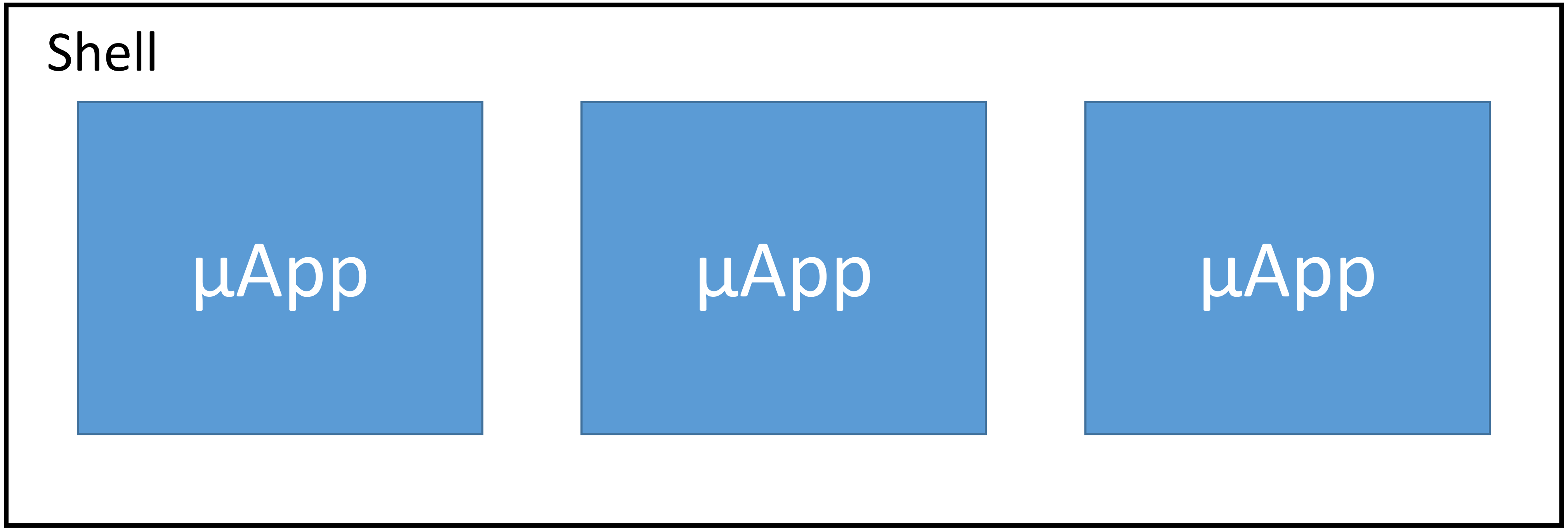
@ManfredSteyer



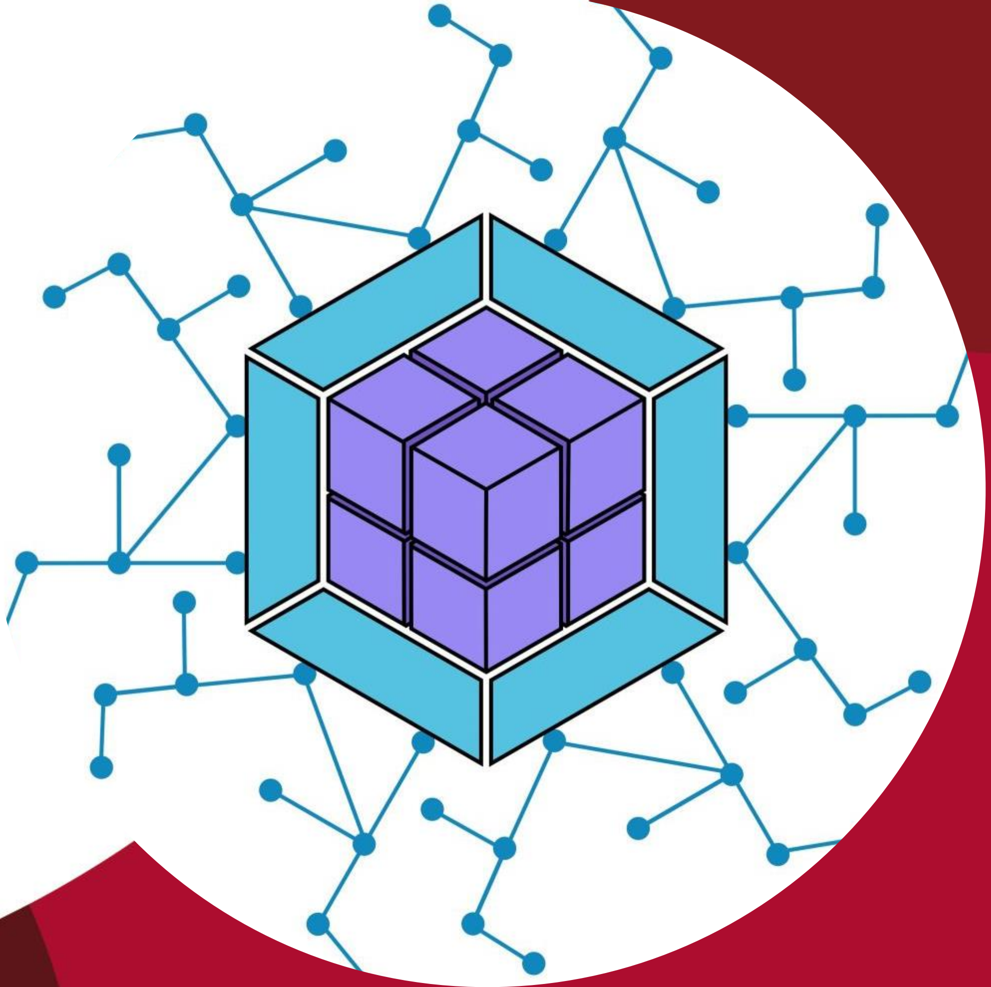
@ManfredSteyer

Integration via Shell

Providing a (SPA based) Shell



Webpack 5
Module Federation



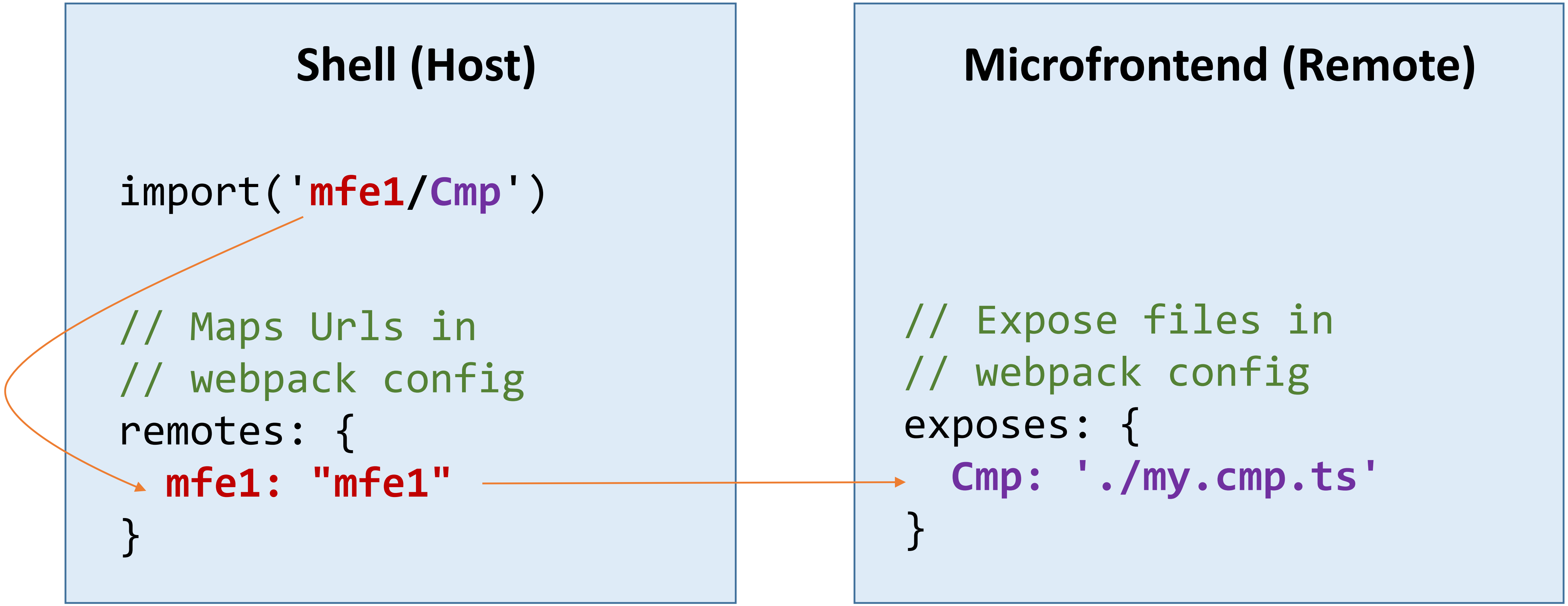
Idea

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



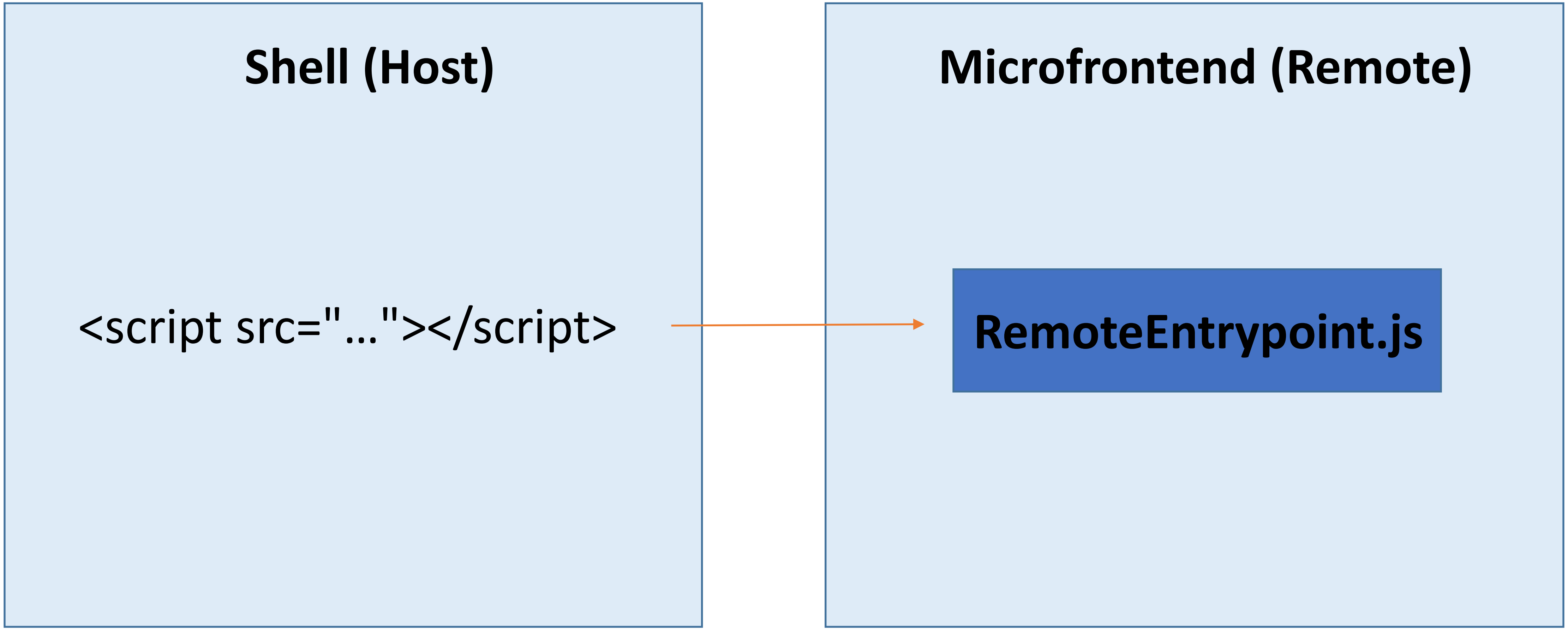
Shell (Host)

```
import('mfe1/Cmp')


// Maps Urls in
// webpack config
remotes: {
  mfe1: "mfe1"
}
```

Microfrontend (Remote)

```
// Expose files in
// webpack config
exposes: {
  Cmp: './my.cmp.ts'
}
```

How to Get the Microfrontend's URL?



Shell (Host)

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

Microfrontend (Remote)

RemoteEntrypoint.js

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!
  }
}
```

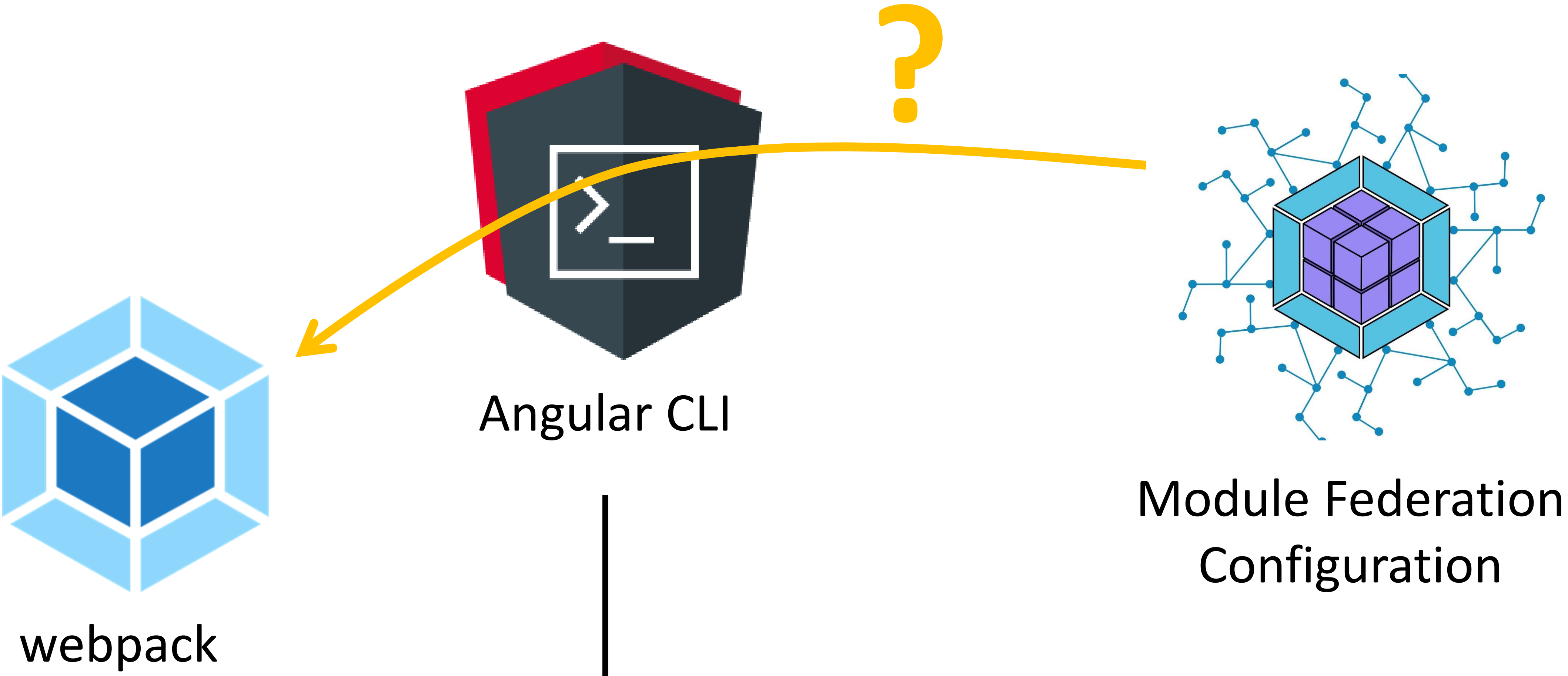
11.0 ✓ 10.1 ✗

Relaxing Version Requirements

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

Federated Angular: Angular, CLI, & Module Federation

ANGULAR ARCHITECTS
INSIDE KNOWLEDGE

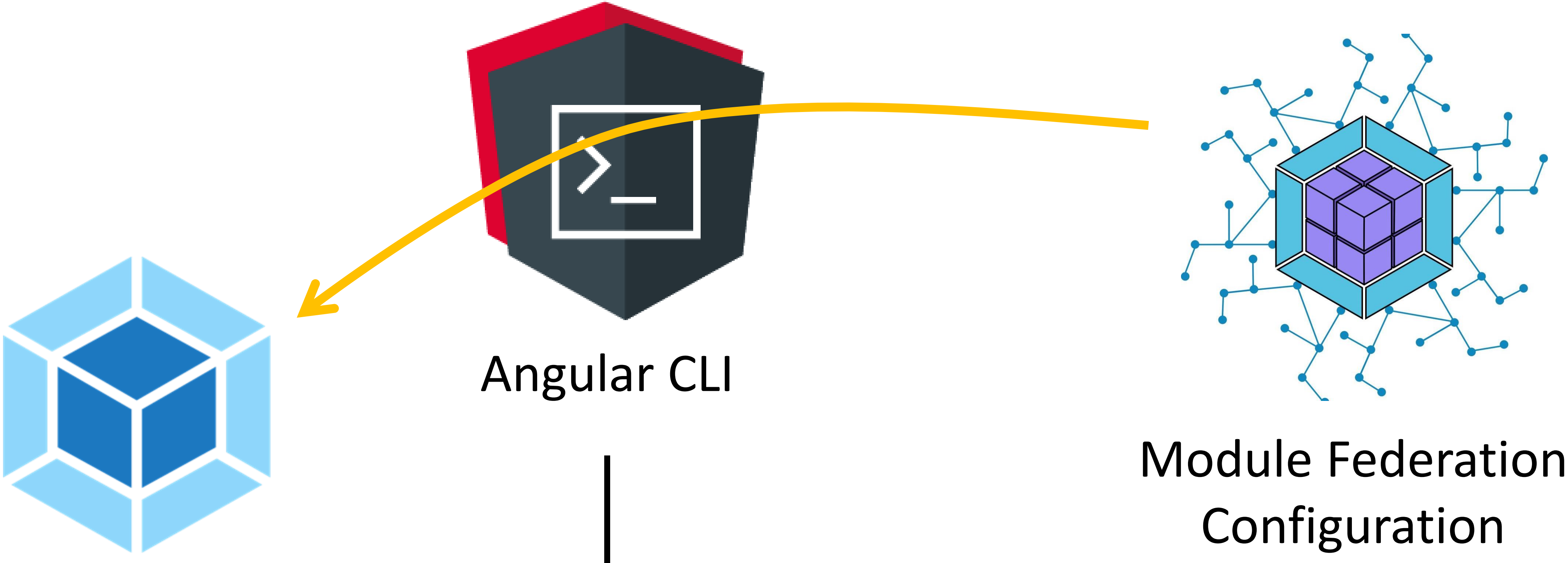


webpack

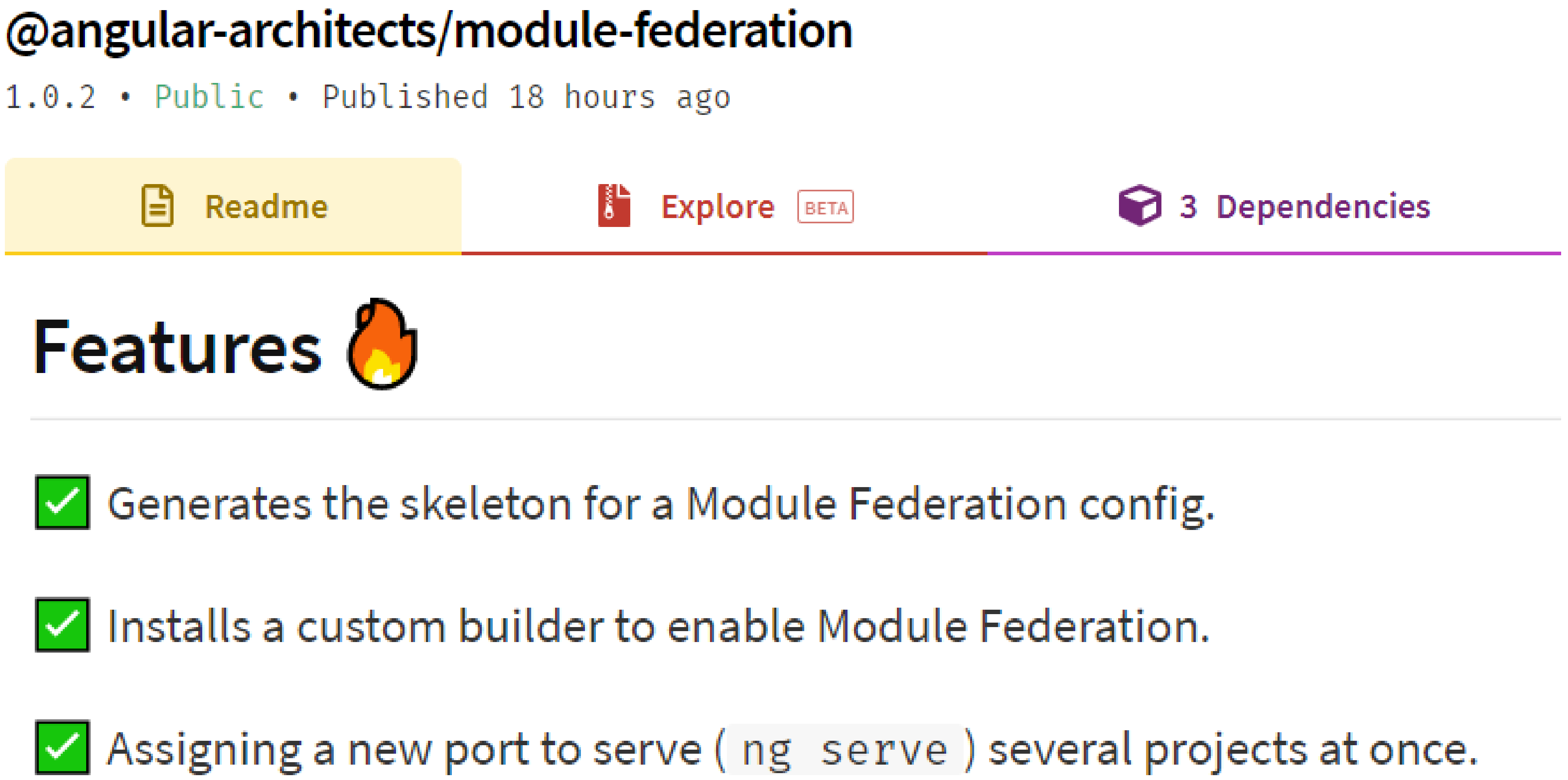
Angular CLI

?

Module Federation Configuration



Custom Builder
Angular CLI
webpack
Module Federation Configuration



@angular-architects/module-federation

1.0.2 • Public • Published 18 hours ago

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

DEMO

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Multi Framework/ Version Solutions

ANGULAR ARCHITECTS
INSIDE KNOWLEDGE

Module Federation

```
const Component = await import('other-app/cmp');
```

Module Federation

```
const main = await import('other-app/main');

main.bootstrap();
```

Module Federation

```
const rootElm = document.createElement('app-root')
document.body.appendChild(rootElm);

const main = await import('other-app/main');

main.bootstrap();
```

Module Federation

```
const rootElm = document.createElement('app-root')
document.body.appendChild(rootElm);

await import('other-app/main'); // Self-Bootstrapping
```

Routing to Another SPA?

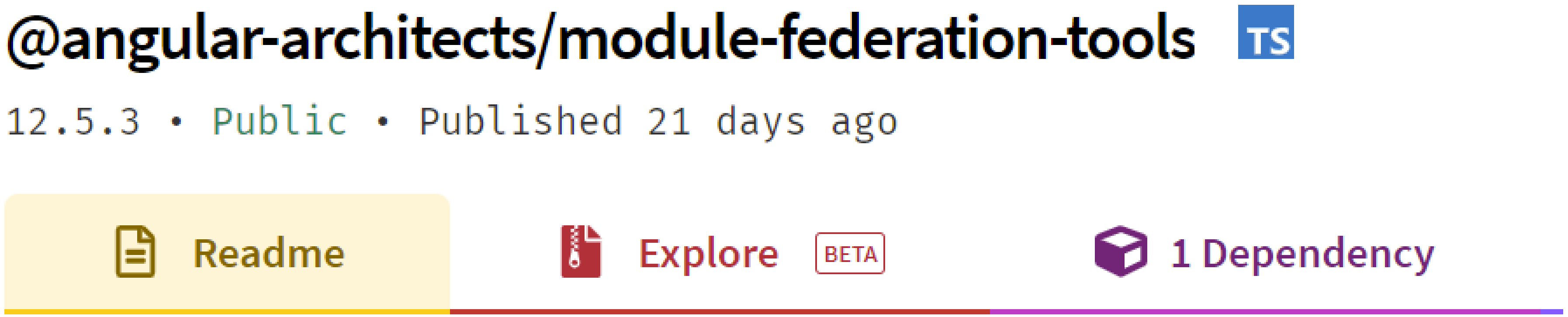
WrapperComponent

```
const rootElm = document.createElement('app-root')
document.body.appendChild(rootElm);

await import('other-app/main');
```

Challanges

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

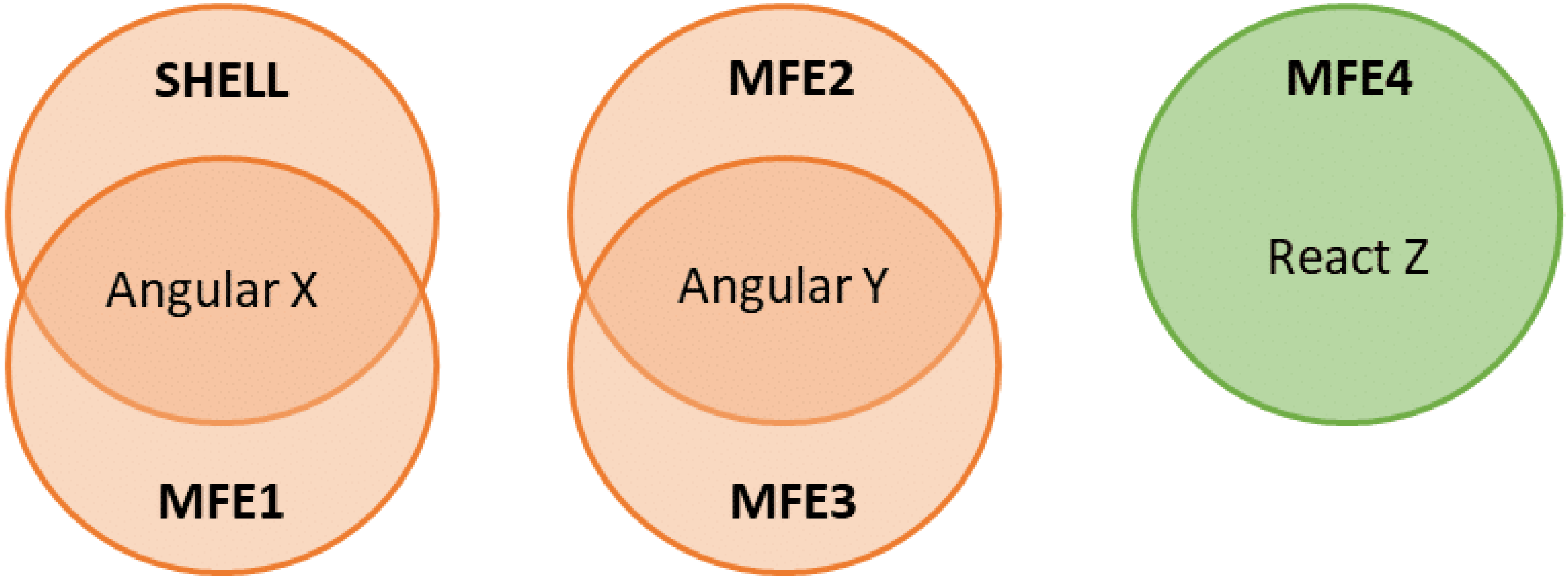
Challanges & Solutions



@angular-architects/module-federation-tools TS

12.5.3 • Public • Published 21 days ago

Result

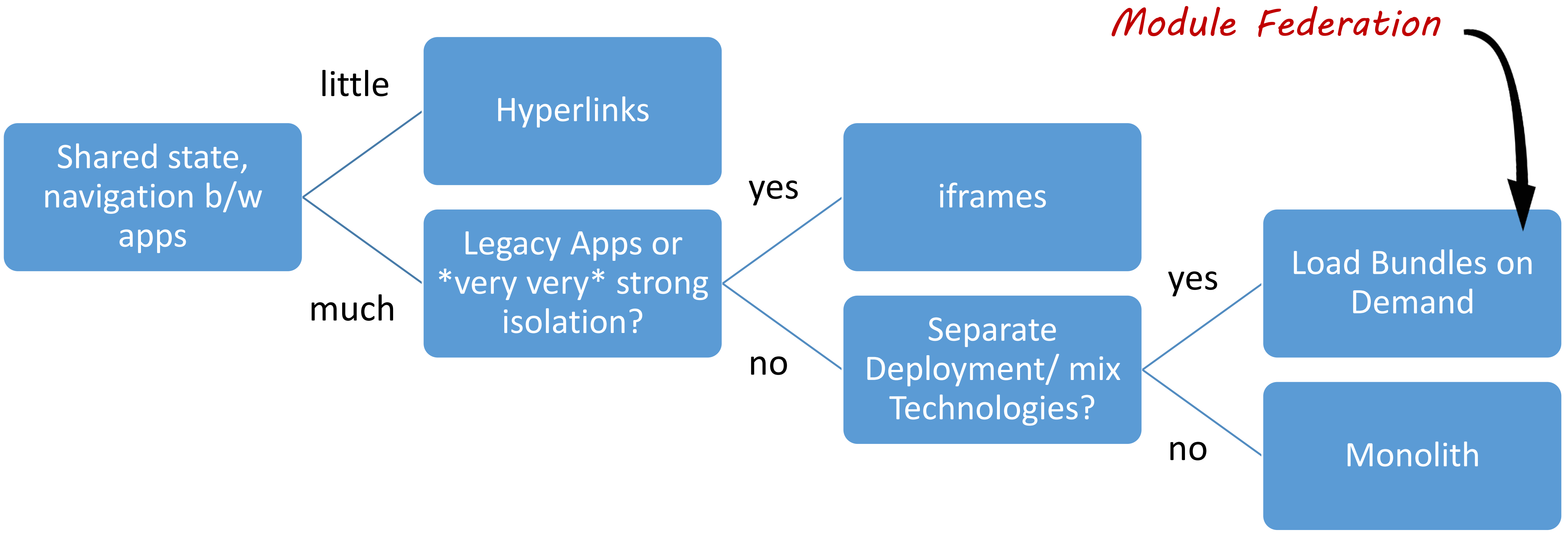


DEMO

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

Choosing a Solution

Some General Advice


Module Federation
Shared state, navigation b/w apps
little
much
Hyperlinks
Legacy Apps or *very very* strong isolation?
yes
no
iframes
Separate Deployment/ mix Technologies?
yes
no
Load Bundles on Demand
Monolith