

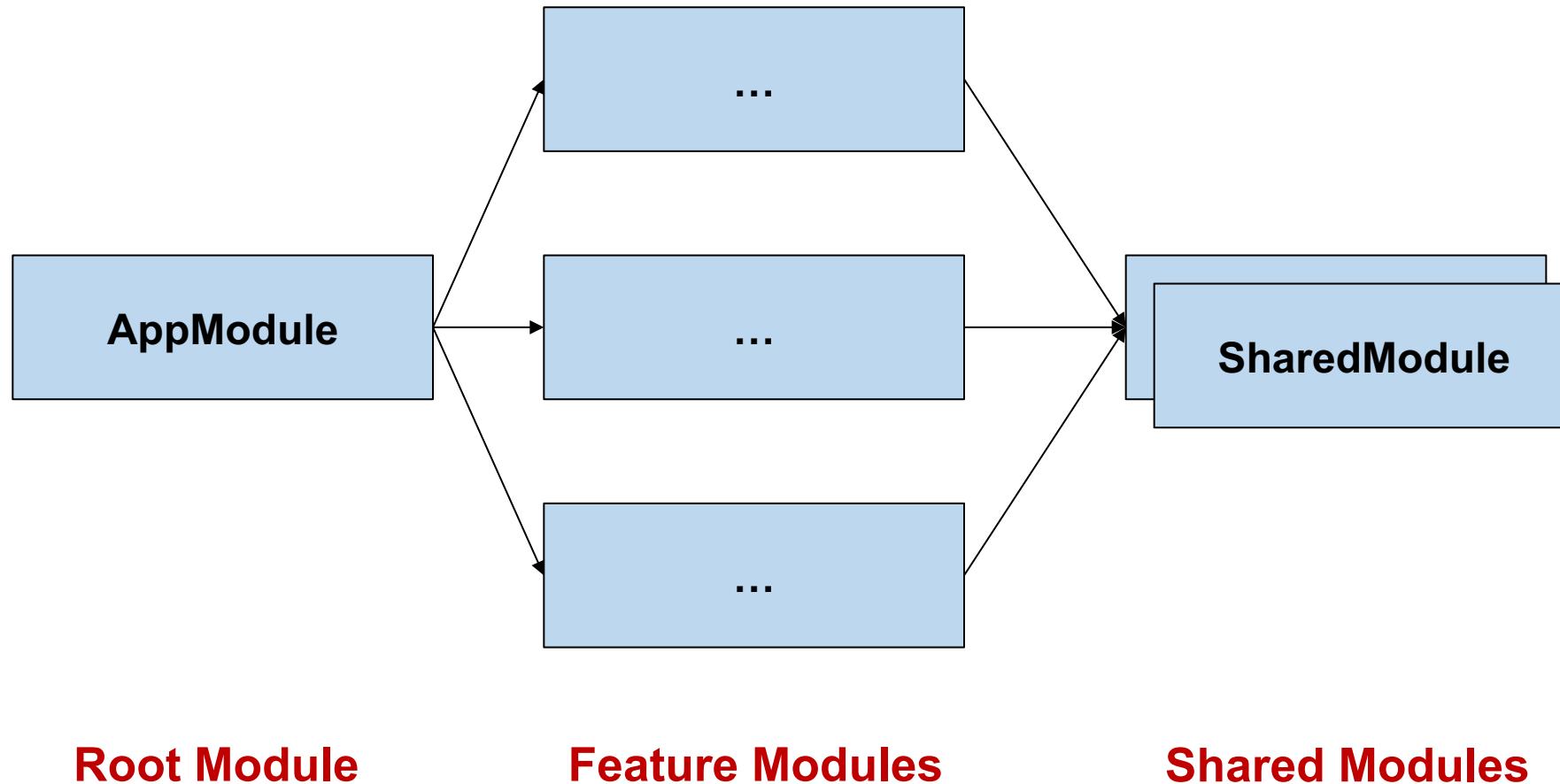


ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Angular Architektur-Workshop: Strategic Design and Micro Frontends

ANGULARarchitects.io

Typical Module Structure



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
```



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/
```

```
@cmp:registry=http://my-server:4871/  
@cmp-ui:registry=http://my-server:4872/
```

npm Registries

Nexus

Artifactory

Team
Foundation
Server

Verdaccio

*npm i -g verdaccio
verdaccio*



@a_awada

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?



@a_awada



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



Two Flavors

Project Monorepo

- Like Workspaces/Solutions in different IDEs

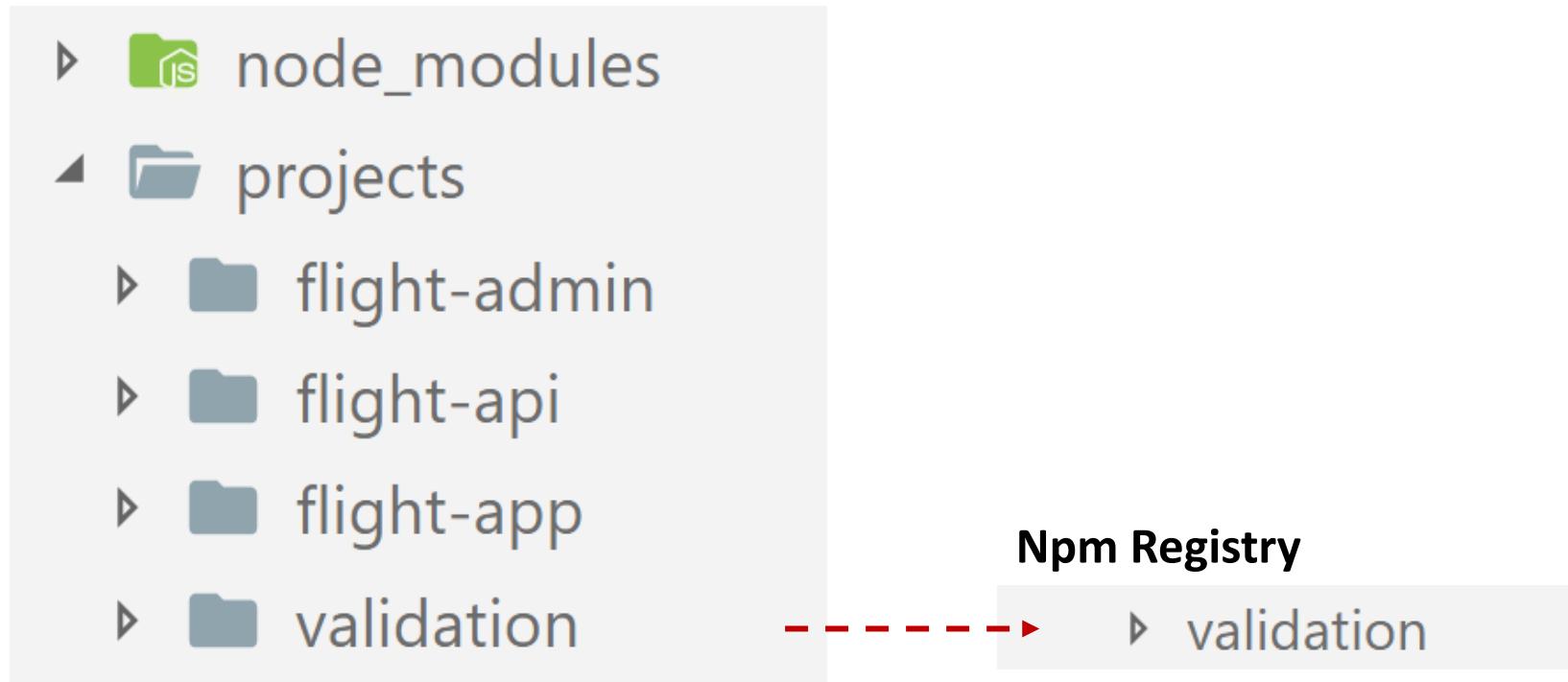
Company-wide Monorepo

- E. g. used at Google or Facebook



@a_awada

Moving back and forth



Tooling & Generator

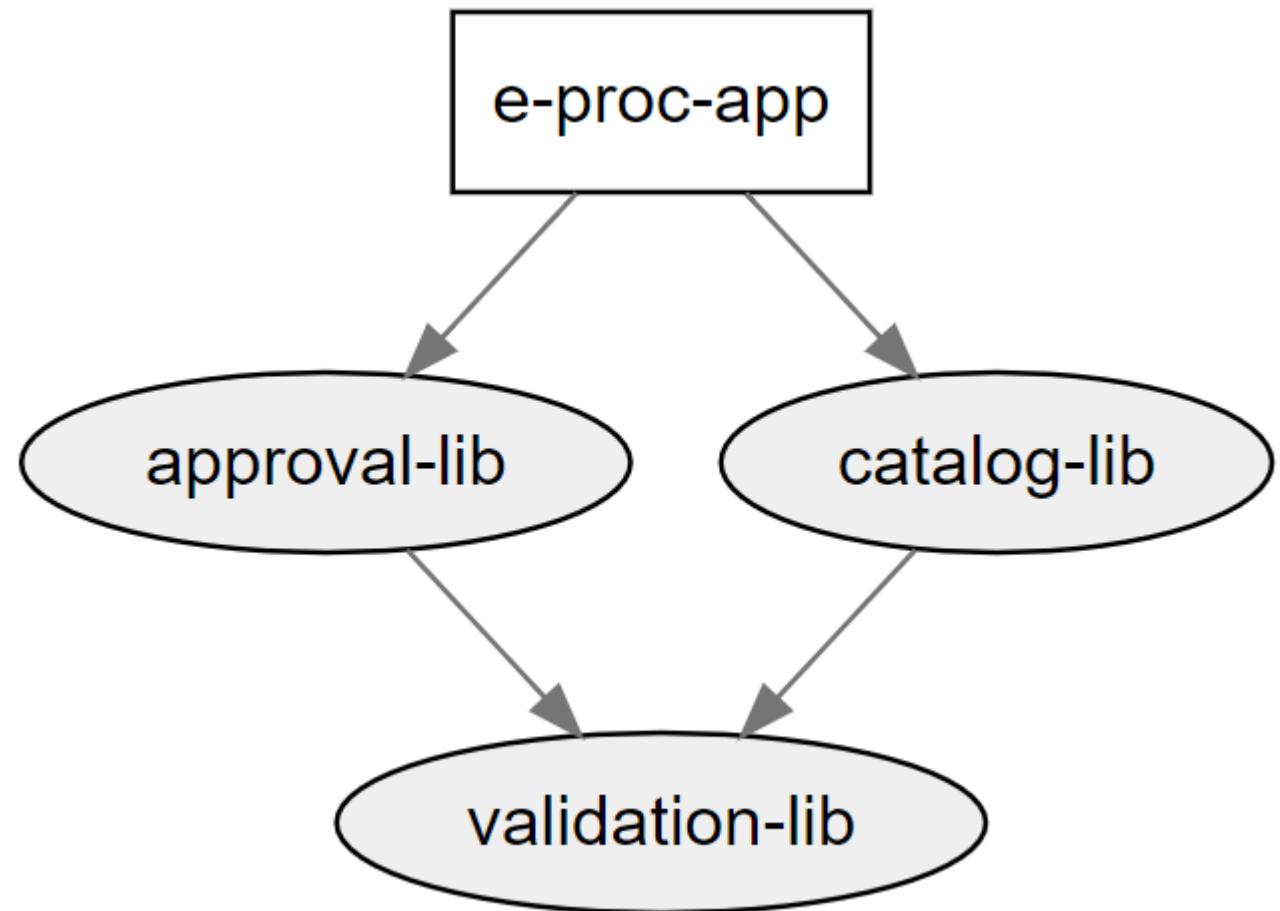
<https://nrwl.io/nx>



Nrwl Extensions for Angular

An open source toolkit for enterprise Angular applications.

Visualize Module Structure



@a_awada

Further Features of Nx

- Define folders for libraries
- Restrict which apps/libs can access which other libs
- Just recompile changed apps
- Visualize module structure and dependencies
- Scaffold Boilerplate

Creating a Workspace

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

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

```
cd workspace
```

```
nx generate app my-app
```

```
nx generate lib my-lib --buildable
```

```
nx serve --project my-app
```

```
nx build --project my-app
```



@a_awada

DEMO



@a_awada

LAB

Domain-Driven DESIGN

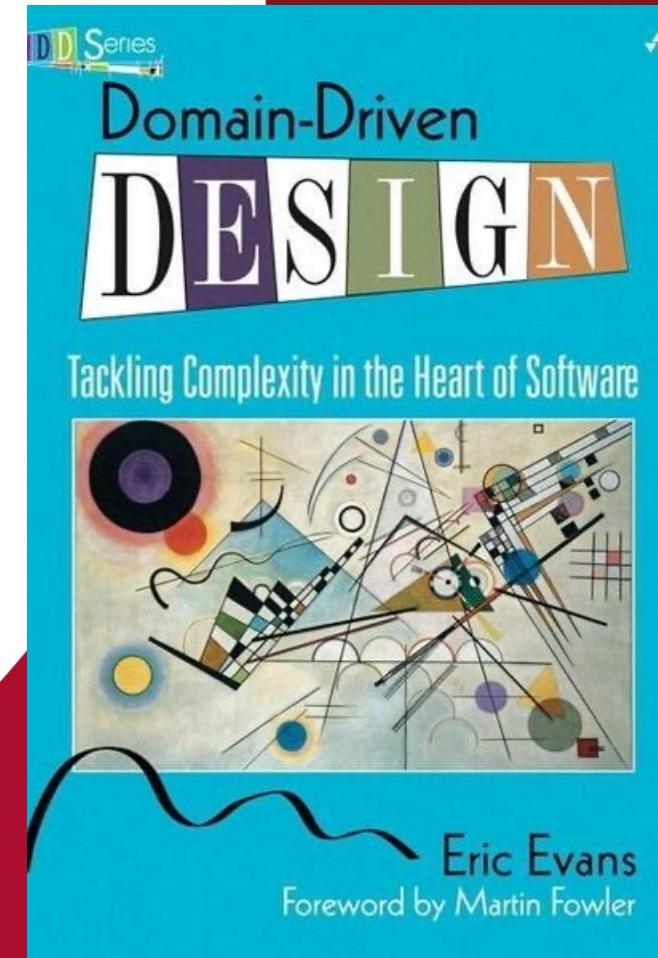
Tackling Complexity in the Heart of Software



Eric Evans
Foreword by Martin Fowler

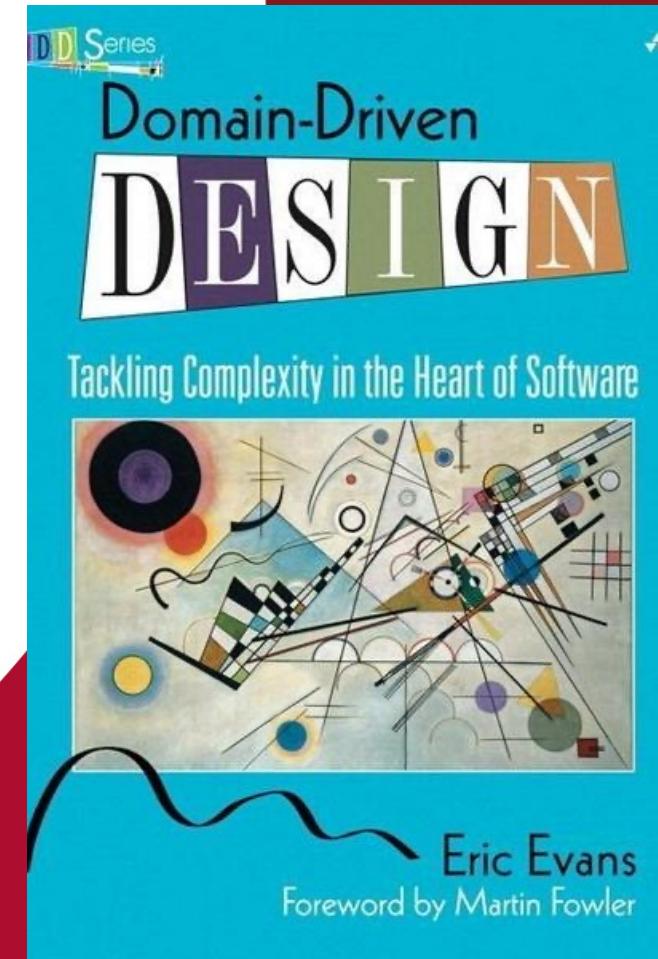
Methodology for
bridging the gap b/w
requirements and
architecture/ design

How to create sustainable frontend architectures with ideas from DDD?



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

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



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Domain Driven Design

Decomposing a System



Strategic Design

Design Patterns
& Practices



Tactical Design

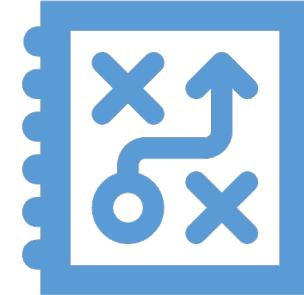
Domain Driven Design

Decomposing a System

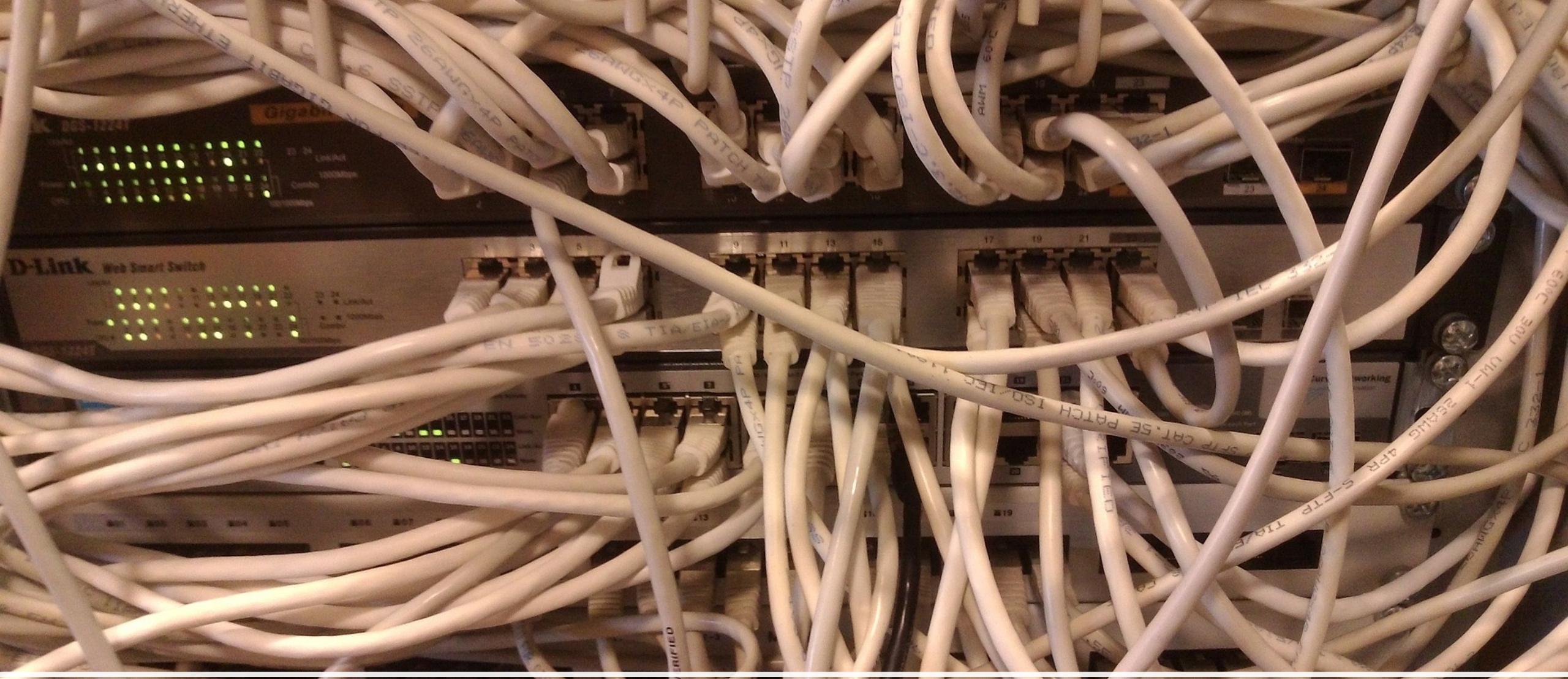


Strategic Design

Design Patterns
& Practices



Tactical Design



This is what Strategic DDD prevents

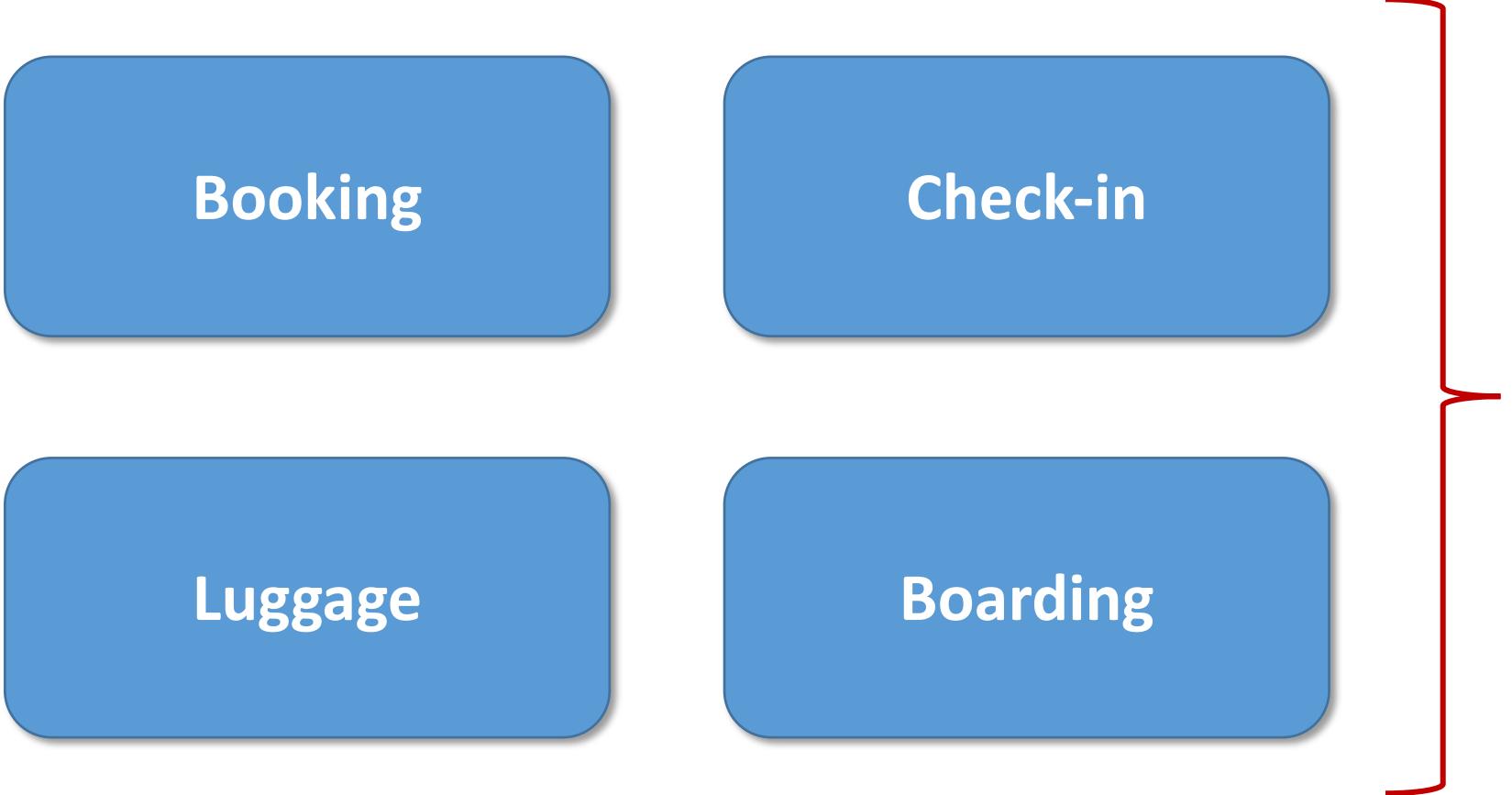
Example

Flight System

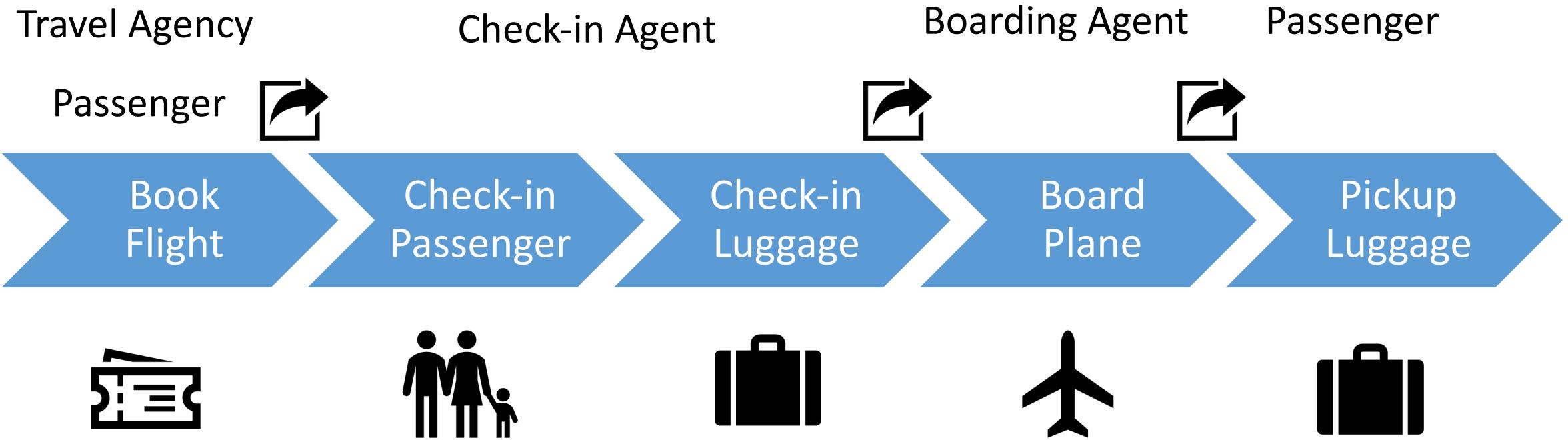


@a_awada

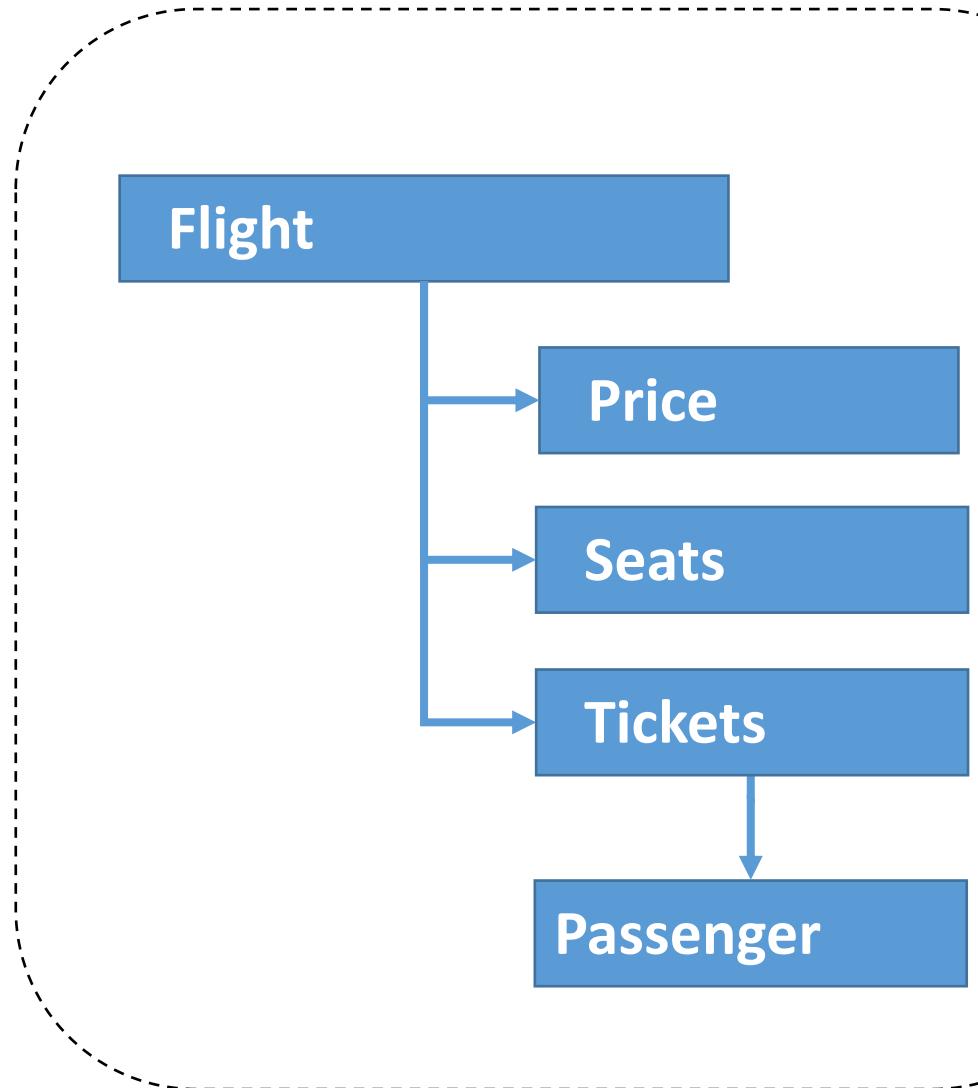
Example



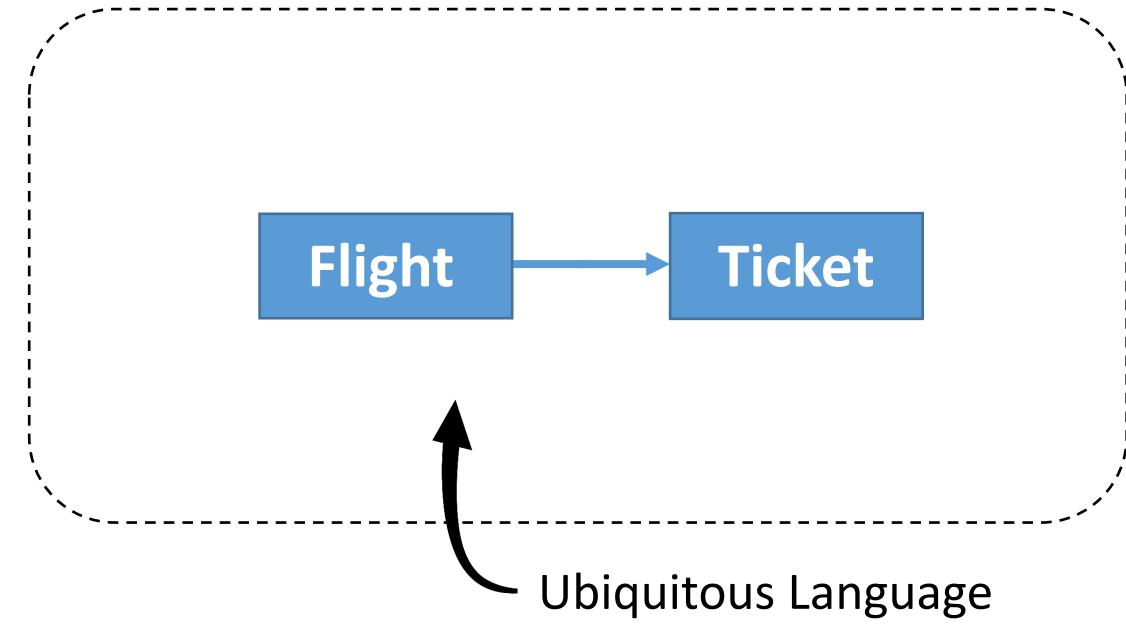
Finding Sub-Domains



Booking

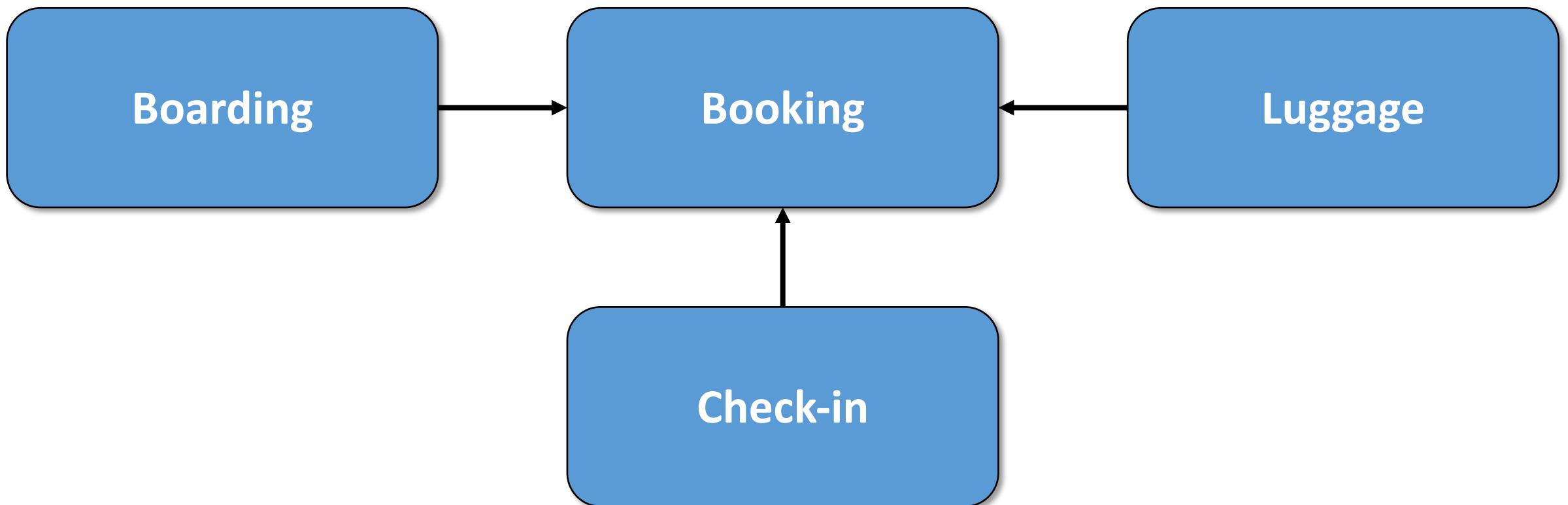


Boarding

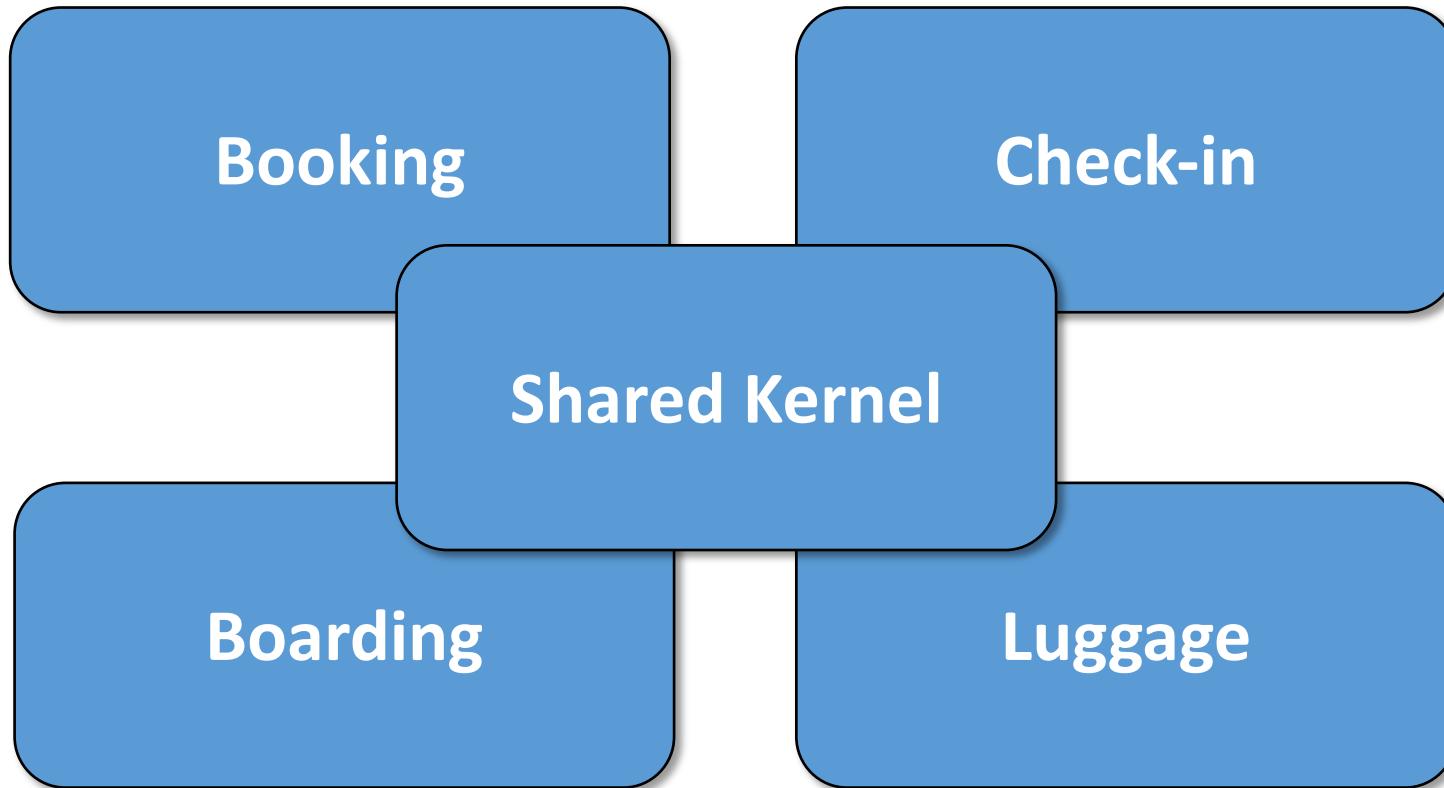


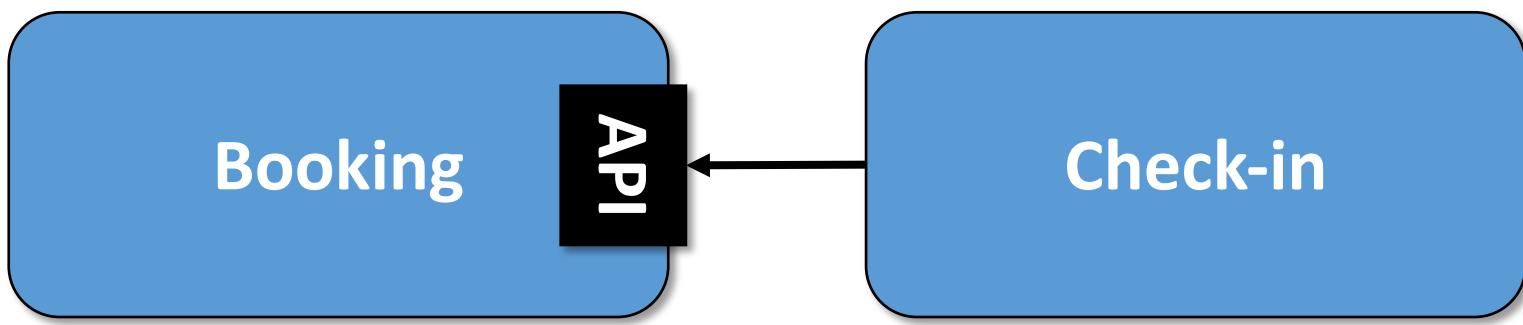
Bounded Context

Context Map



Context Map





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.

Booking

Boarding

Shared

Feature

Feature

Feature

Feature

Feature

UI

UI

UI

UI

UI

UI

UI

UI

UI

Domain

Domain

Domain

Domain

Domain

Domain

Util

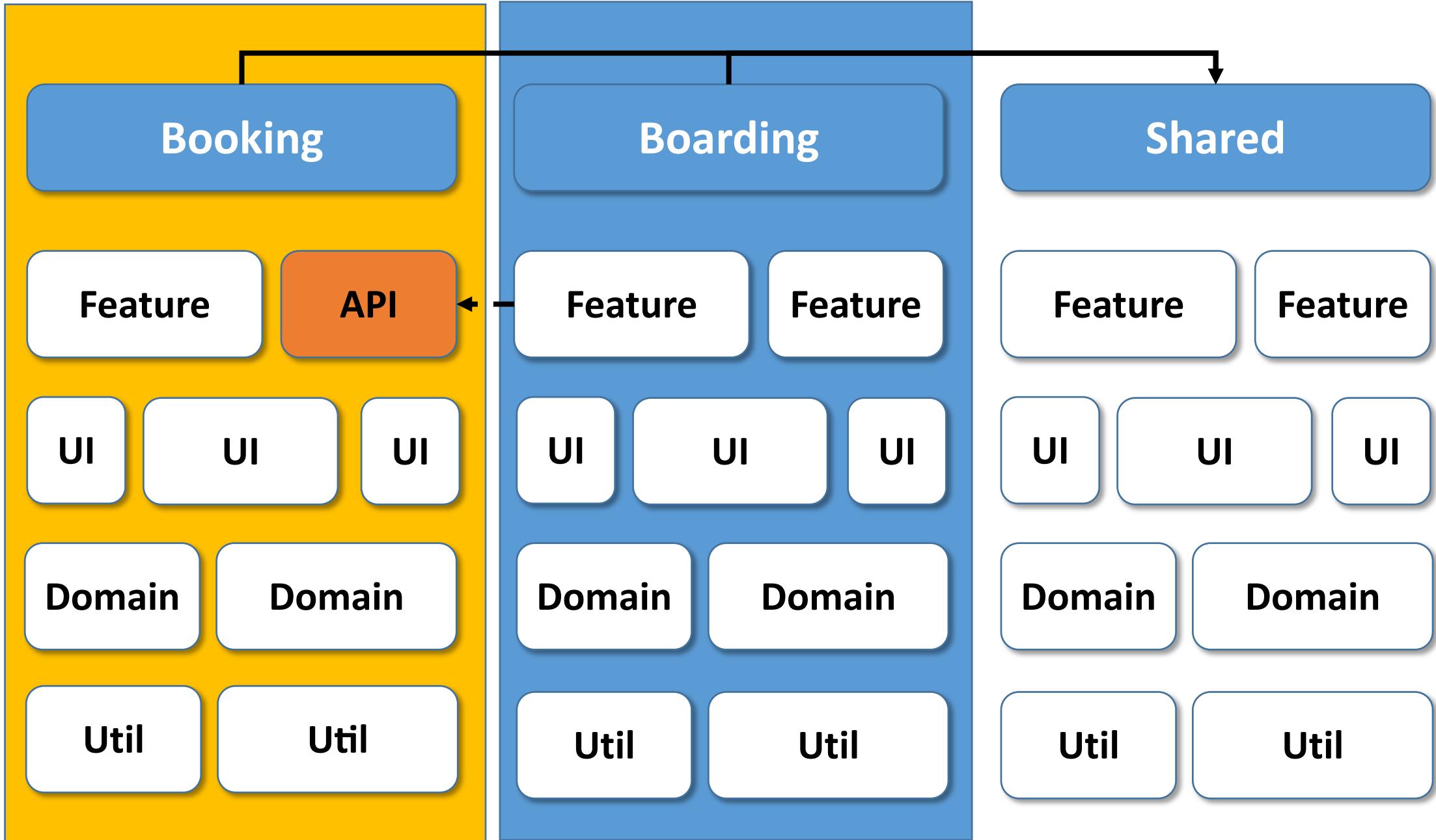
Util

Util

Util

Util

Util



@a_awada

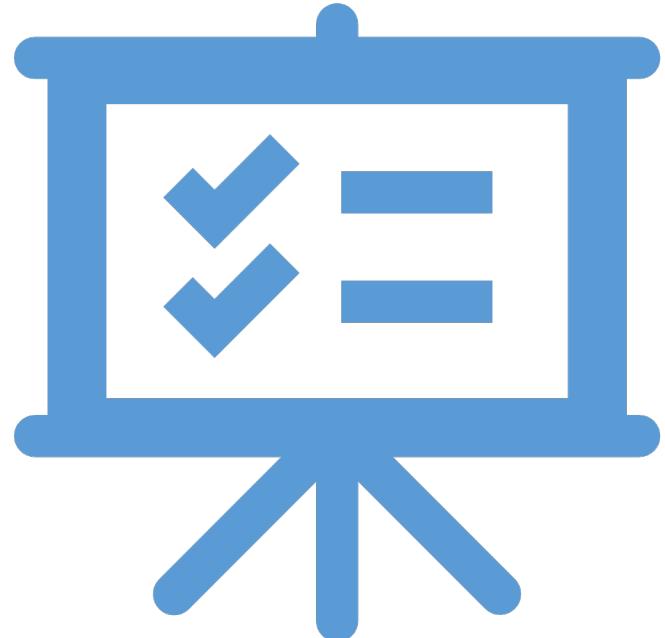


Alternatives to layering

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

DEMO

LAB



Finegrained Libraries

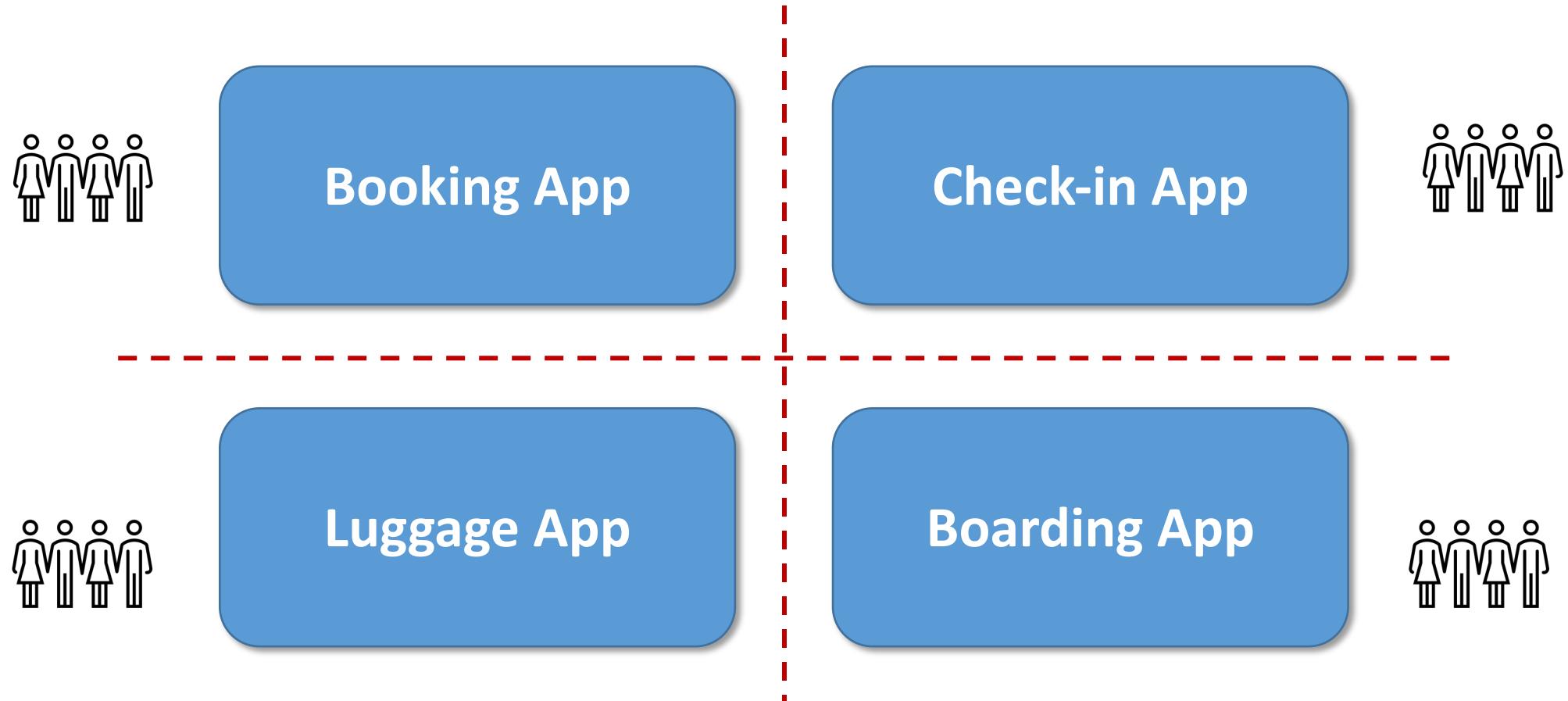
- Unit of recompilation
- Unit of retesting
- Access restrictions
- Information Hiding
- Easy: Just *ng g lib ...*
- Future replacement for NgModules?



Micro Frontends?

Short outlook

Microfrontends



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



Deployment Monolith

Booking

Boarding

Shared

Flight App

Feature

Feature

Feature

Feature

Feature

...

...

...

...

...

...

...

...

...



@a_awada

Microfrontends

Booking

Boarding

Shared

Booking App

Boarding App

Feature

Feature

Feature

Feature

Feature

...

...

...

...

...

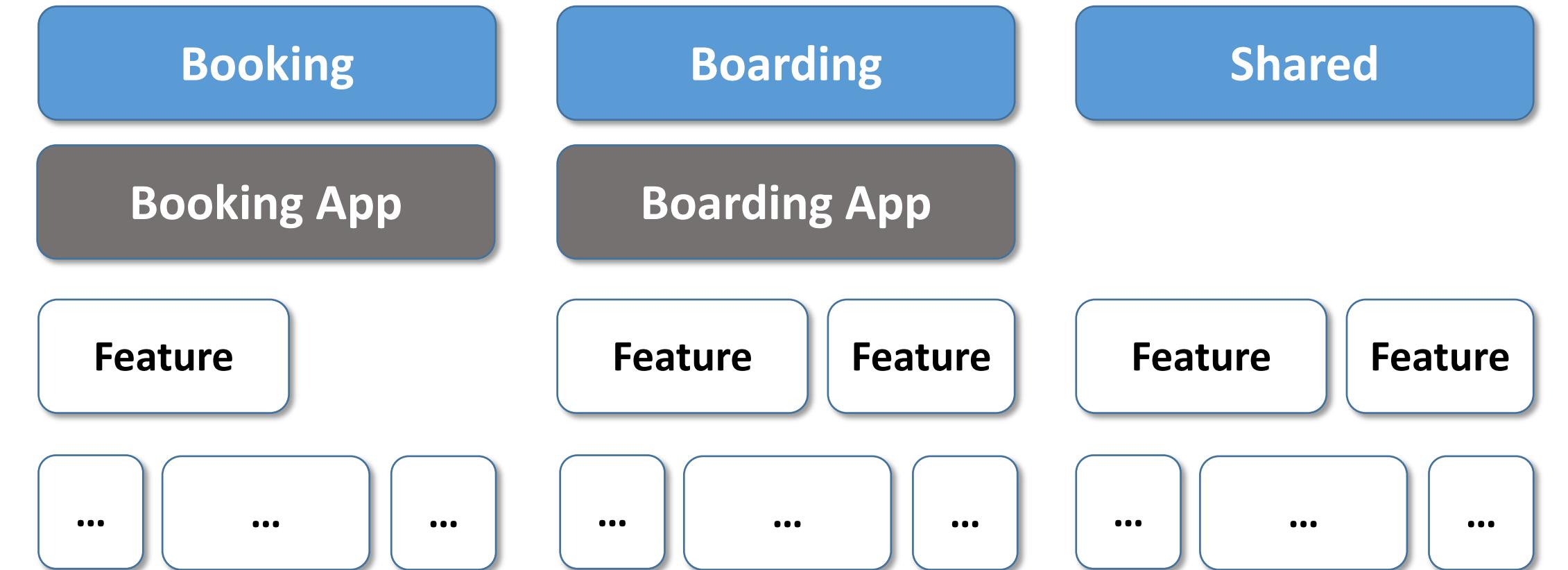
...

...

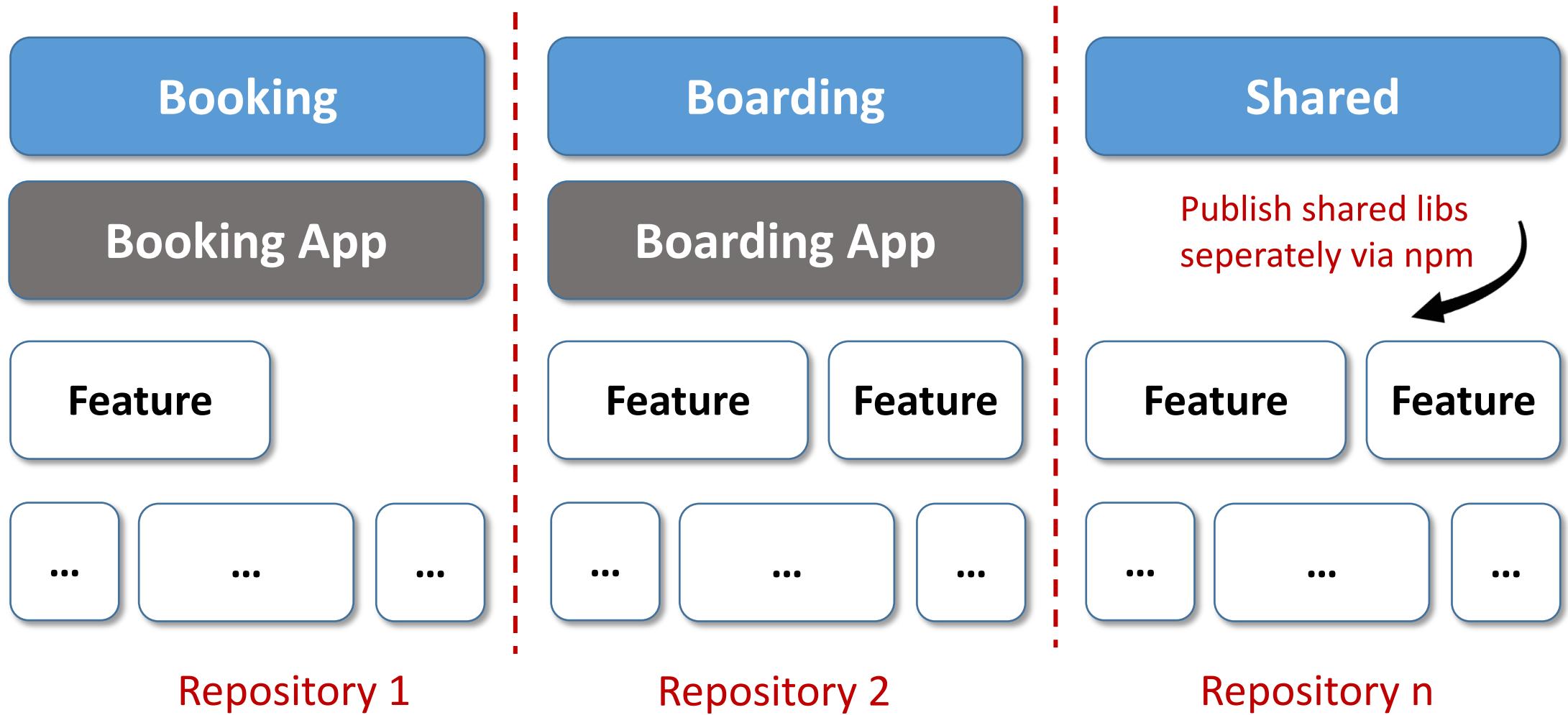
...

...

Option 1: One App per Domain



Option 2: One Monorepo per Domain



Benefits

Autonomous Teams

Separate Development

Separate Deployment

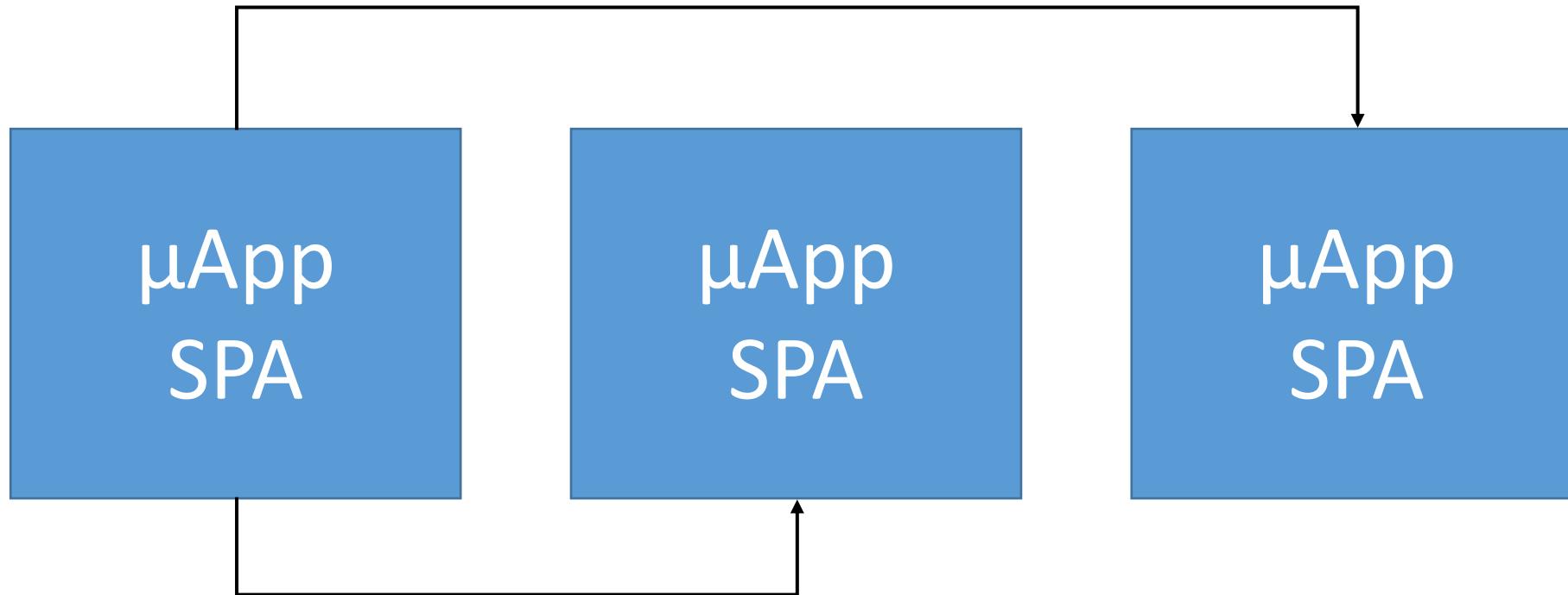
Own architecture decisions

Own technology decisions



Integration via
Hyperlinks

UI Composition w/ Hyperlinks



Word Online

Document - Auf OneDrive gespeichert.

Manfred S... Abmelden

DATEI START EINFÜGEN SEITENLAYOUT ÜBERPRÜFEN ANSICHT Was möchten Sie tun? IN WORD BEARBEITEN

Rückgängig Einfügen Zwischenablage

Calibri (Textkörper) 16 AaBbCc Standard AaBbCc Kein Leerra... AaBbCc Überschrift 1

Schriftart Absatz Formatvorlagen Bearbeiten

Hello World!

This screenshot shows the Microsoft Word Online interface. The top navigation bar includes 'Word Online', the document title 'Document - Auf OneDrive gespeichert.', and user information 'Manfred S... Abmelden'. Below the title, the ribbon tabs are visible: DATEI, START (selected), EINFÜGEN, SEITENLAYOUT, ÜBERPRÜFEN, and ANSICHT. A search bar says 'Was möchten Sie tun?' and a 'IN WORD BEARBEITEN' button is present. The ribbon icons include 'Rückgängig', 'Einfügen', 'Zwischenablage', 'Schriftart' (with Calibri selected), 'Absatz' (with Standard selected), 'Formatvorlagen', and 'Bearbeiten'. The main content area displays the text 'Hello World!'. On the right side, there is a vertical scroll bar.

Word Online

Document - Auf OneDrive gespeichert.

S Manfred S... Abmelden

The screenshot shows the Microsoft Word Online interface. At the top, there's a blue header bar with the title "Word Online" and the message "Document - Auf OneDrive gespeichert.". On the right side of the header, there's a user profile icon, the name "Manfred S...", and a "Abmelden" (Logout) button. Below the header is a ribbon bar containing icons for various Microsoft services: Outlook.com (blue square), Kontakte (orange square), Kalender (purple square), OneDrive (blue square), Word Online (dark blue square), Excel Online (green square), PowerPoint Online (orange square), OneNote Online (purple square), Sway (teal square), Skype (light blue square), Office Online (red square), and Flow (blue square). To the right of the ribbon, there's a "IN WORD BEARBEITEN" (Edit in Word) panel with options for font, style, and search.

A large, light-colored seashell, possibly a cone shell, is positioned on a sandy beach. The shell is oriented vertically, pointing upwards. In the background, the ocean with white-capped waves meets a clear blue sky.

Integration via
Shell

Providing a (SPA based) Shell

Shell

μApp

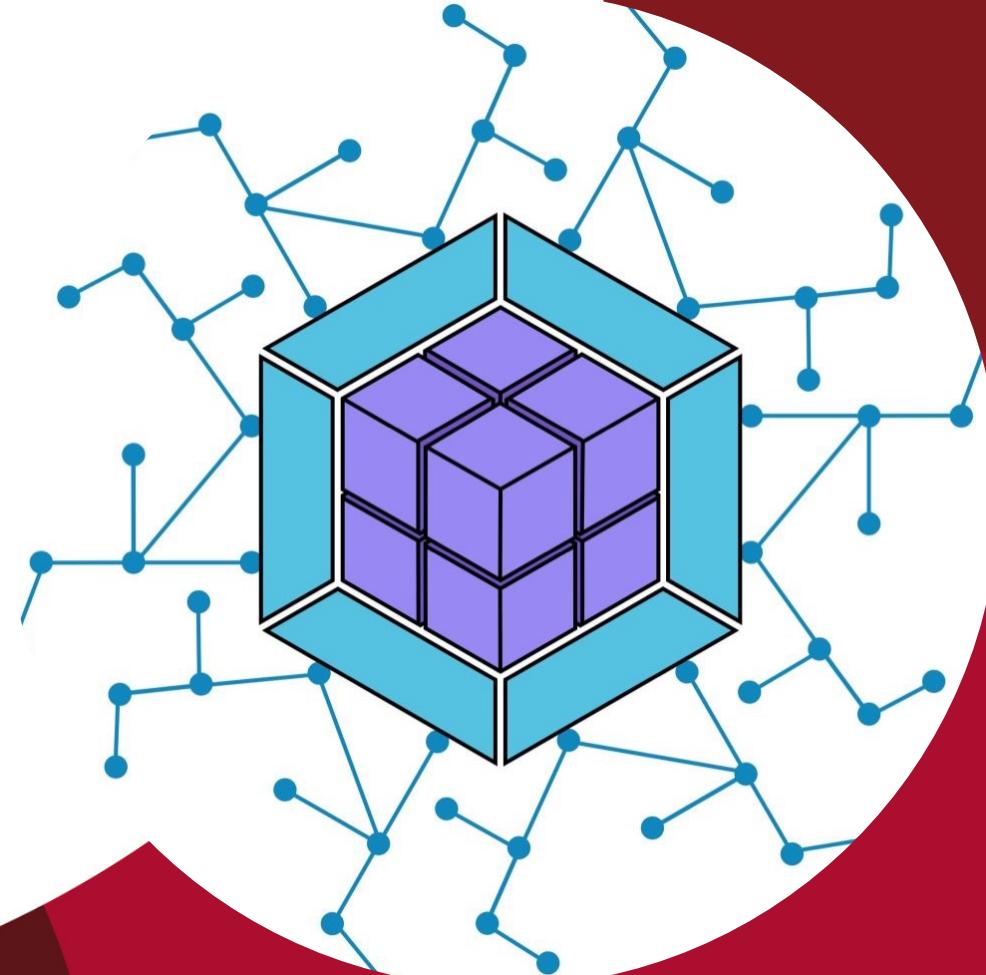
μApp

μApp



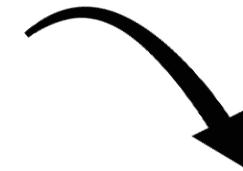
@a_awada

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)

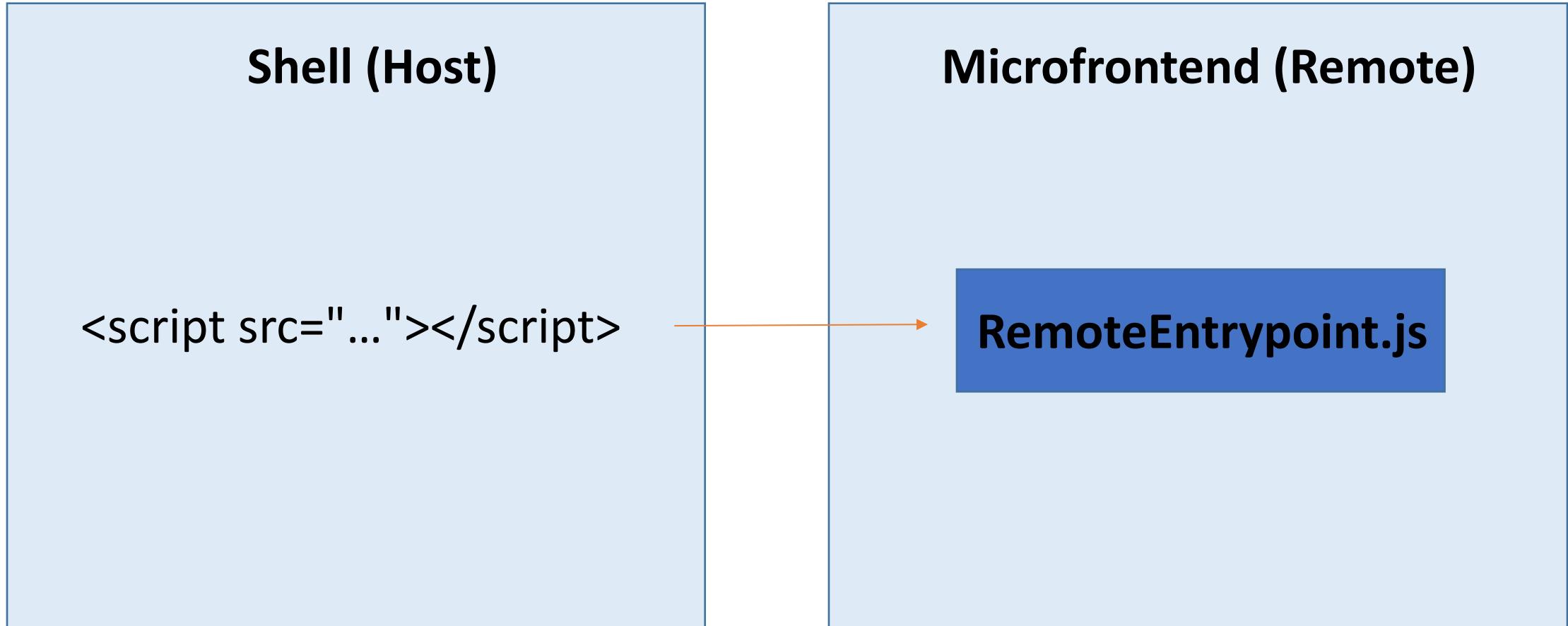
```
import('mfe1/Cmp')  
  
// MapsUrls in  
// webpack config  
remotes: {  
  mfe1: "http://..."  
}
```

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~~
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 



@a_awada

Configuring Singletons

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

11.0  10.1 



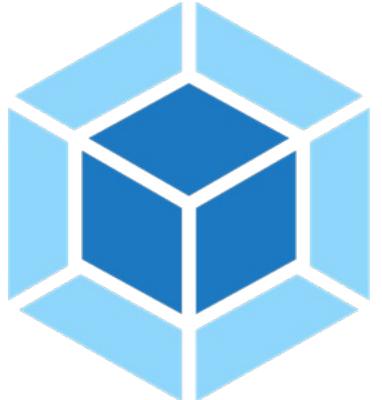
Relaxing Version Requirements

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

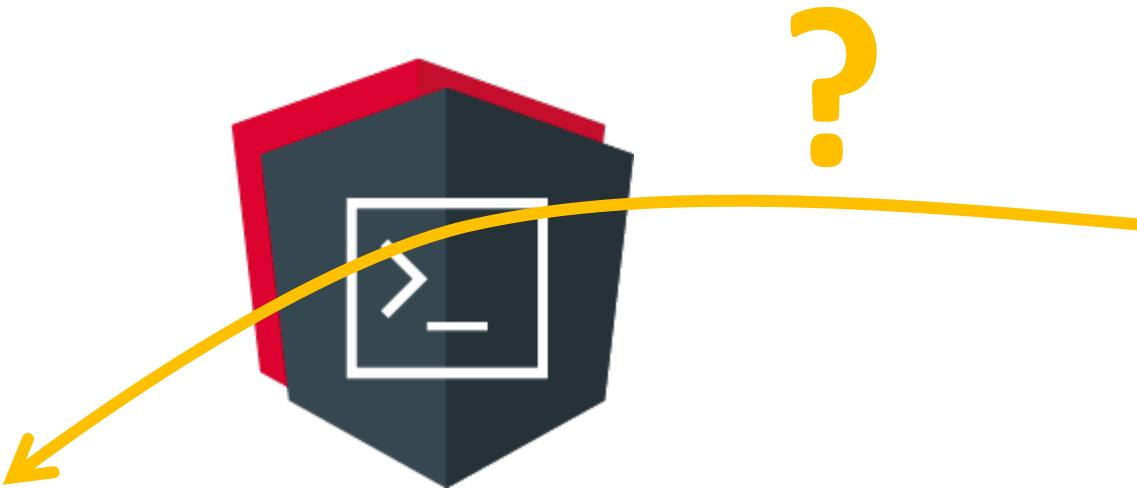
Federated Angular: Angular, CLI, & Module Federation



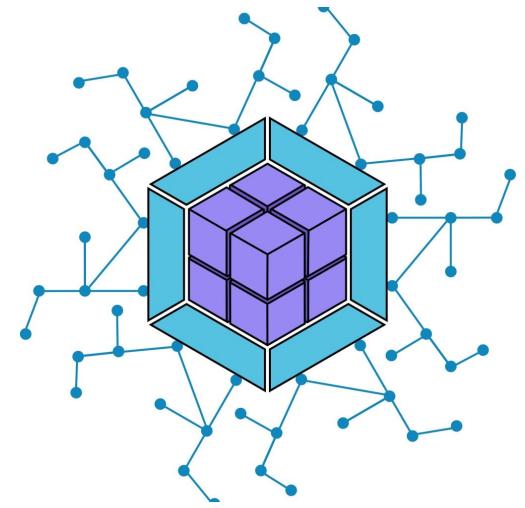
ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



webpack



Angular CLI

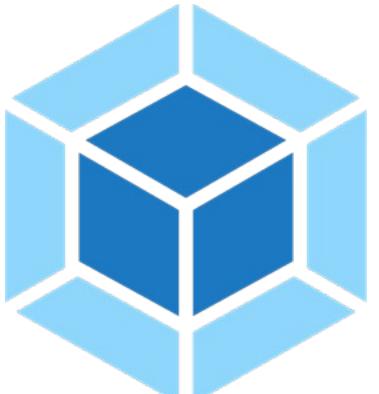


Module Federation
Configuration

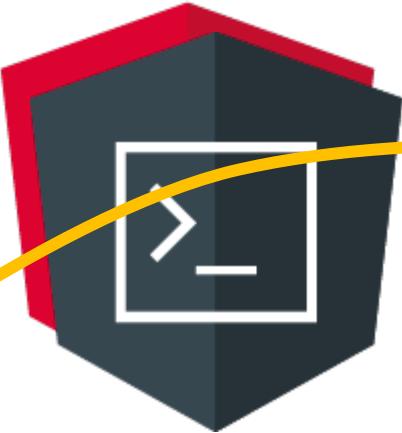


@a_awada

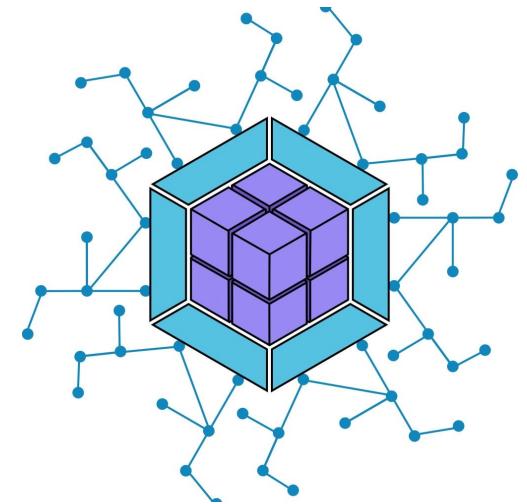
Custom Builder



webpack



Angular CLI



Module Federation
Configuration



@a_awada

@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.



@a_awada

Usage

- 1) CLI
 - 1) ng add @angular-architects/module-federation
- 2) Nx
 - 1) npm install @angular-architects/module-federation –D
 - 2) nx generate @angular-architects/module-federation:init --project <project-name>
- 3) Adjust generated configuration
- 4) ng serve

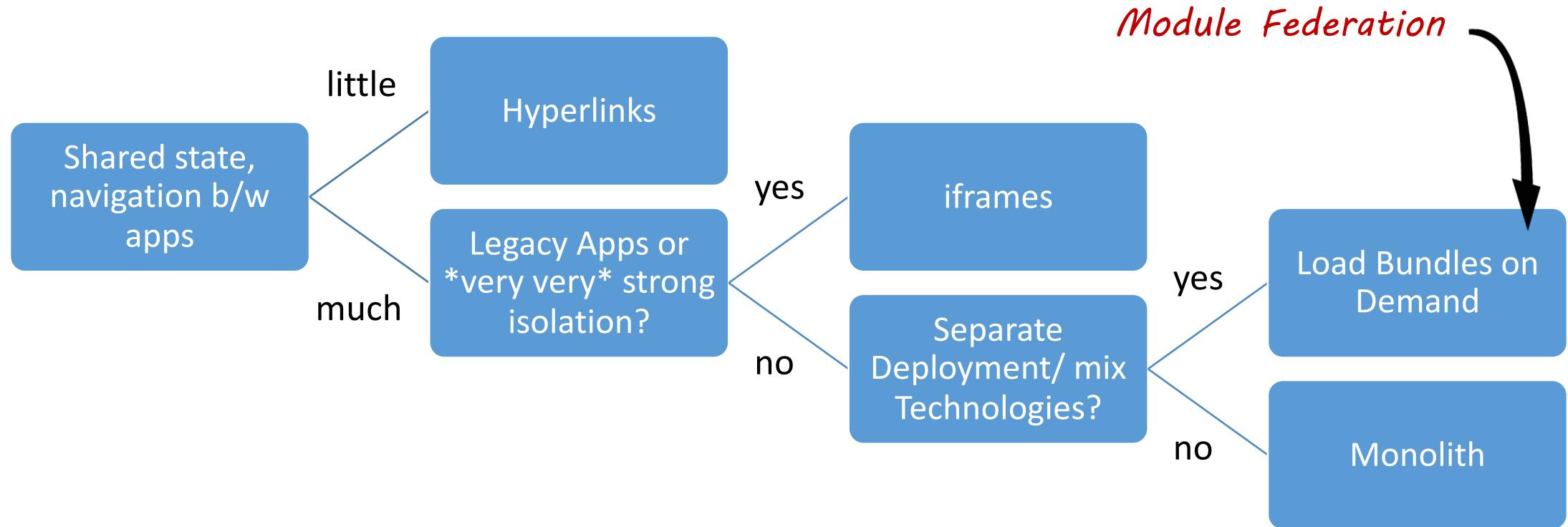
DEMO

LAB

The background of the image is a dark, atmospheric room. The wall is covered in a repeating pattern of dark, ornate, floral or damask motifs. In the center of the frame, there is a row of several closed doors, all of which appear to be made of a light-colored wood or similar material. The doors are evenly spaced and extend from the foreground towards the background, creating a sense of depth. The floor is made of dark, polished wooden planks that reflect the ambient light.

Choosing a Solution

Some General Advice



Summary

Libs: Subdividing big solution into tiny parts

Monorepo: No burden with distributing libs

Strategic Design: Cut system into loosely coupled domains

Nx: Access restrictions, visualization, incremental compilation/ testing, etc.

Architecture Matrix: Orientation and access restrictions

Microfrontends: Autarkic Teams, separate deployment, challenges