



ManfredSteyer



Architectures for huge Angular based enterprise applications

Manfred Steyer

SOFTWAREarchitekt.at

About me...

- Manfred Steyer
- **SOFTWARE**architekt.at
- Angular Trainings and Consultancy
- Google Developer Expert (GDE)



Manfred Steyer



Public: Vienna, München, Stuttgart, Frankfurt
In-House: everywhere

<http://softwarearchitekt.at/workshops>



Kick-off



After some weeks



After some years

Ever started a project from the scratch?

Contents

1)
npm Packages

2)
Monorepo

3)
Micro Apps

4)
How to
choose?

The background of the slide is a close-up photograph of several wrapped gifts. In the foreground, a large, vibrant pink ribbon is tied into a bow on a light pink gift box. Behind it, a purple gift box with a silver ribbon and a yellow gift box with a blue ribbon are visible, though they are out of focus. The overall scene is festive and celebratory.

1) npm Packages

Why Packages?



Structure of an npm Package

- /node_modules
- your-stuff
- package.json

<https://goo.gl/hjt7G3>



Angular Package Format

Lots of
DETAILS





Angular CLI 6+
supports
Packages

Create npm-Packages with CLI >= 6

```
ng new lib-project
```

```
cd lib-project
```

```
ng generate library flight-booking-lib
```

```
ng generate application playground-app
```

```
ng serve --project playground-app
```

```
ng build --project flight-booking-lib
```

```
cd dist/flight-booking-lib
```

```
npm publish --registry http://...
```


npm Registries

Nexus

Artifactory

Team
Foundation
Server

Verdaccio

*npm i -g verdaccio
verdaccio*



Advantages

- Distribution
- Versioning

Disadvantages

- Distribution
- Versioning

;-)

Disadvantages

Distribution

- Annoying within project











Versioning

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



2) Monorepos

Workspace

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

Advantages

Everyone uses the latest versions

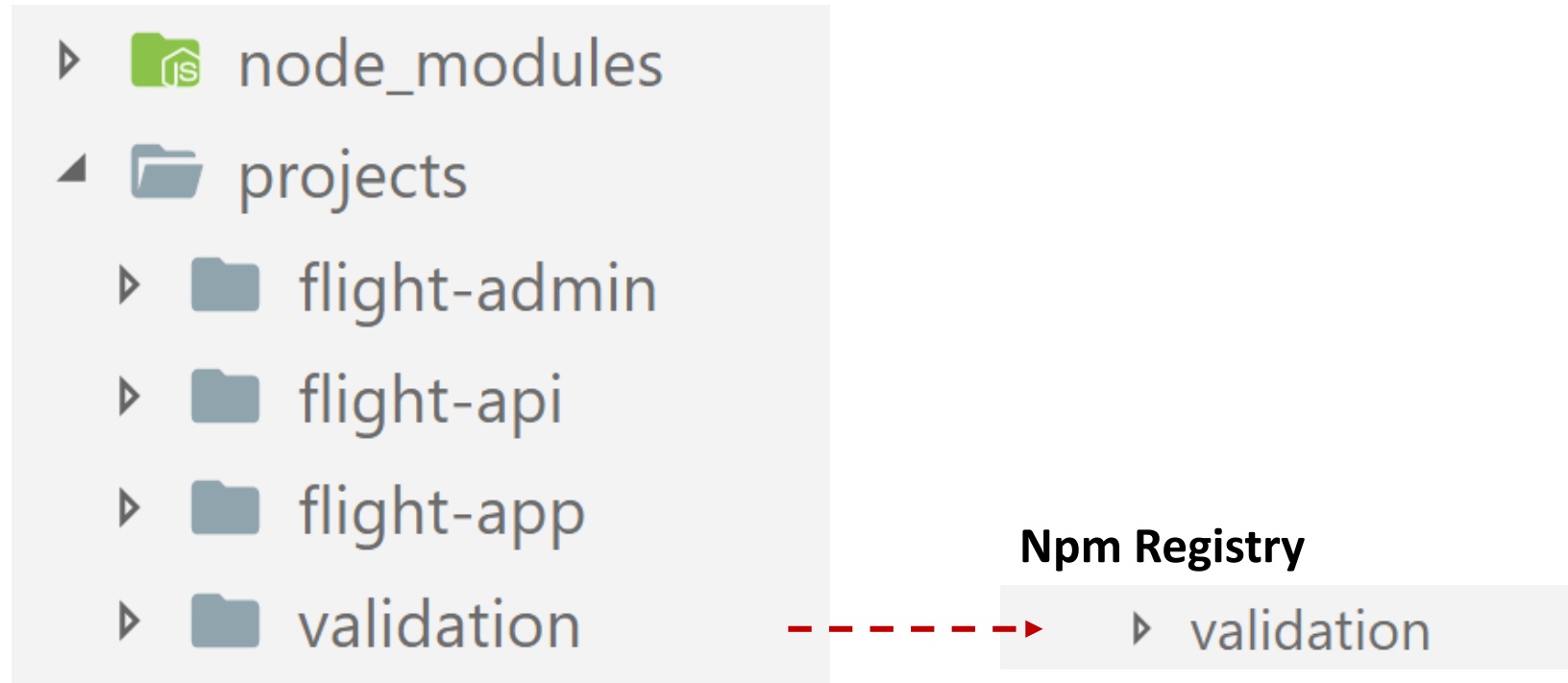
No version conflicts

No burden with distributing libs

Creating new libs: Adding folder

Experience: Successfully used at Google, Facebook, ...

Moving back and forth



DEMO

Tooling & Generator

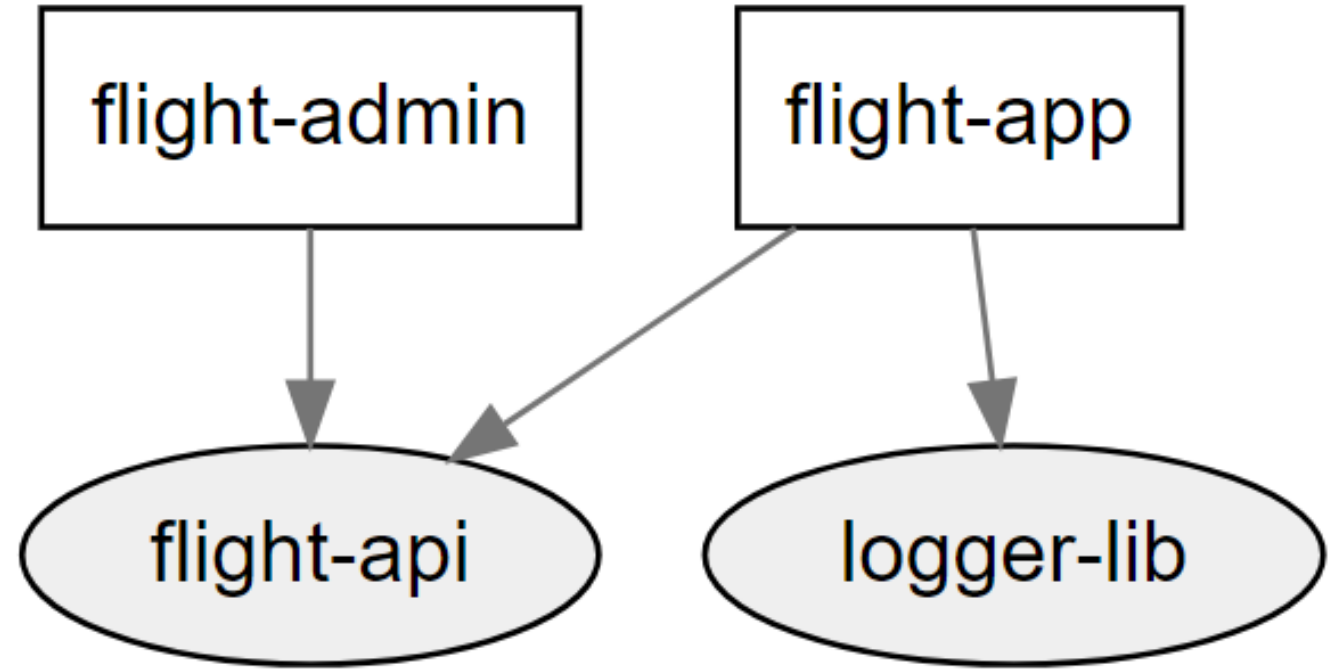
<https://nrwl.io/nx>



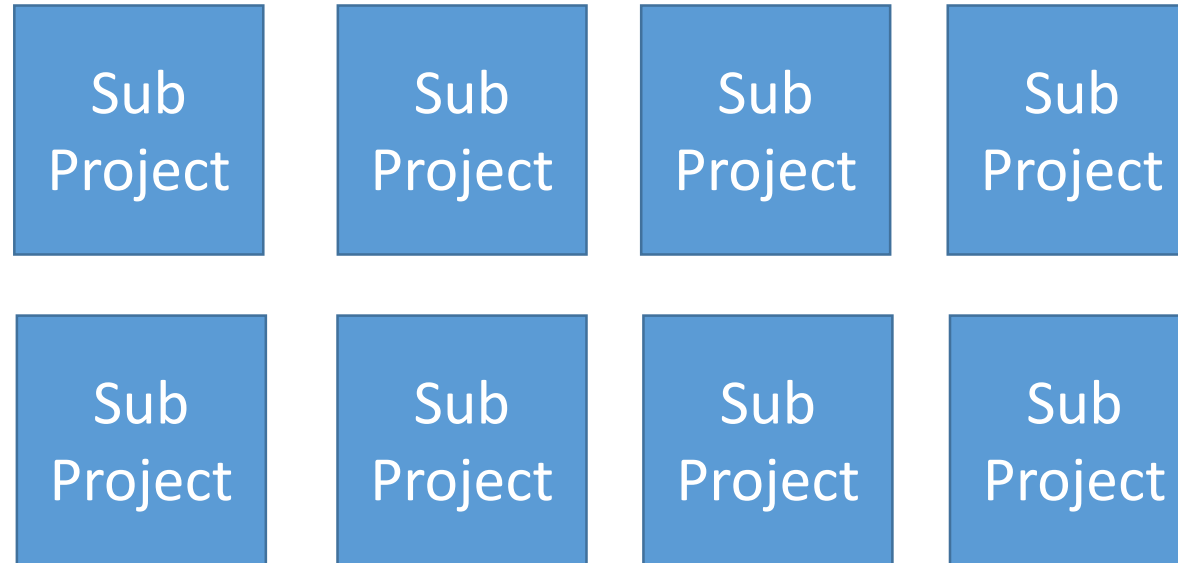
Nrwl Extensions for Angular

An open source toolkit for enterprise Angular applications.

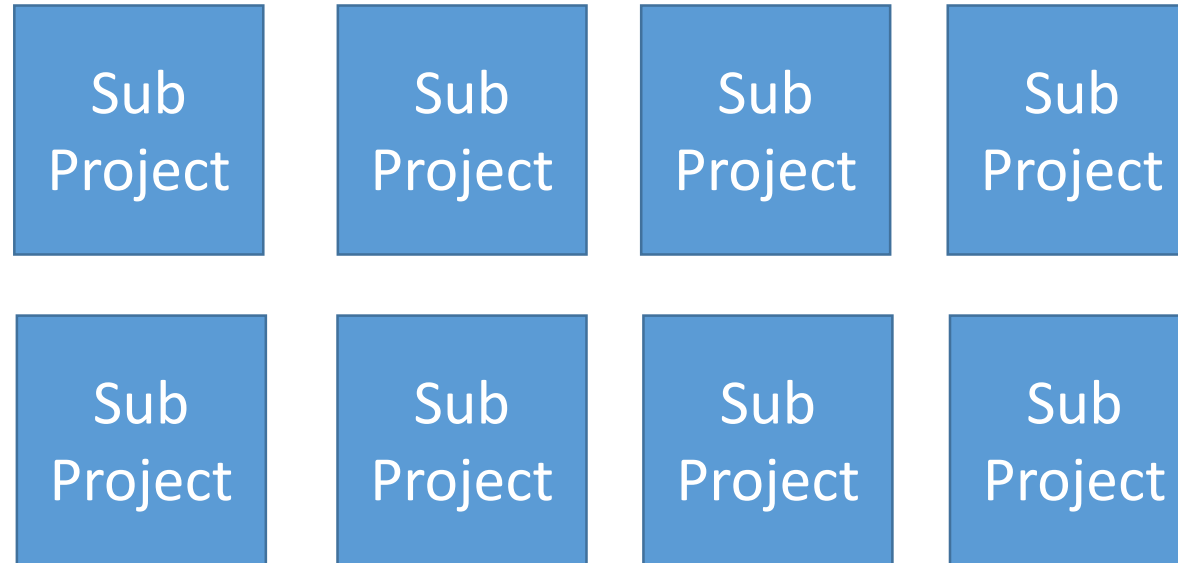
Visualize
Module
Structure



Advantages



Disadvantages

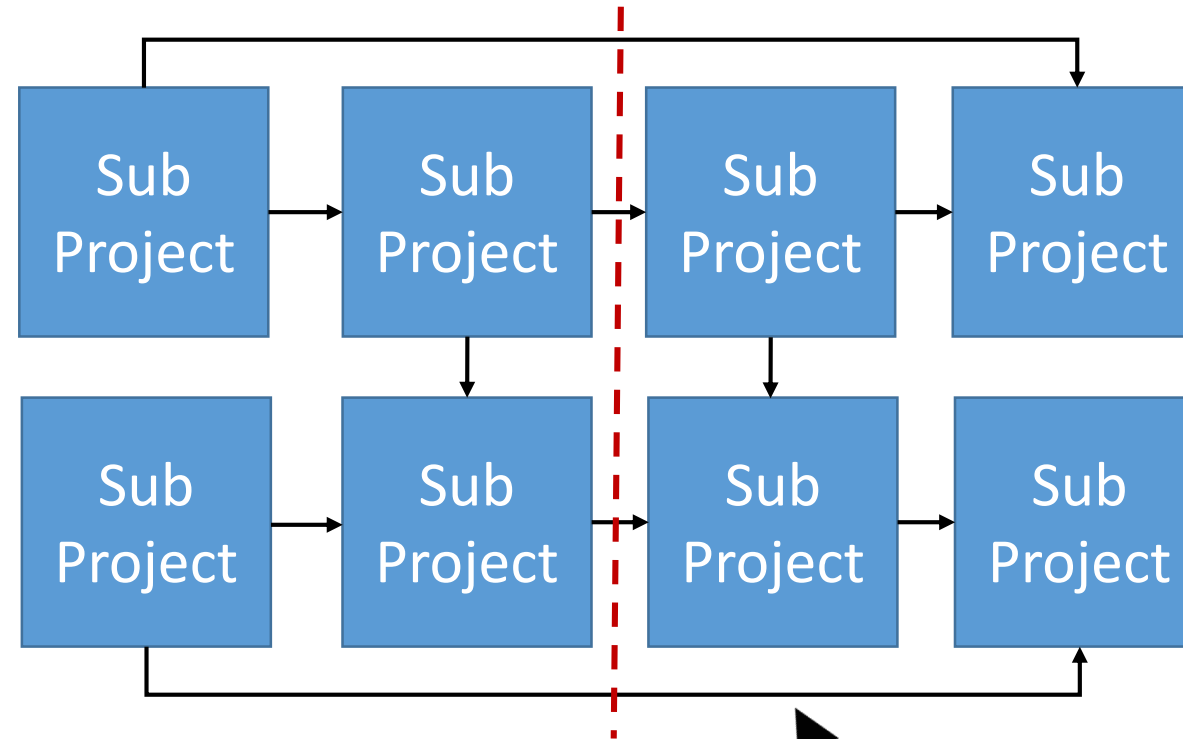


Disadvantages

Maintainability?



Team B



Team A



Contract

One Architecture/ One Framework



3) Micro Apps

Idea



System

Idea

μService

μService

μService

Idea

μApp

μApp

μApp

Advantages

No Contracts
b/w Apps

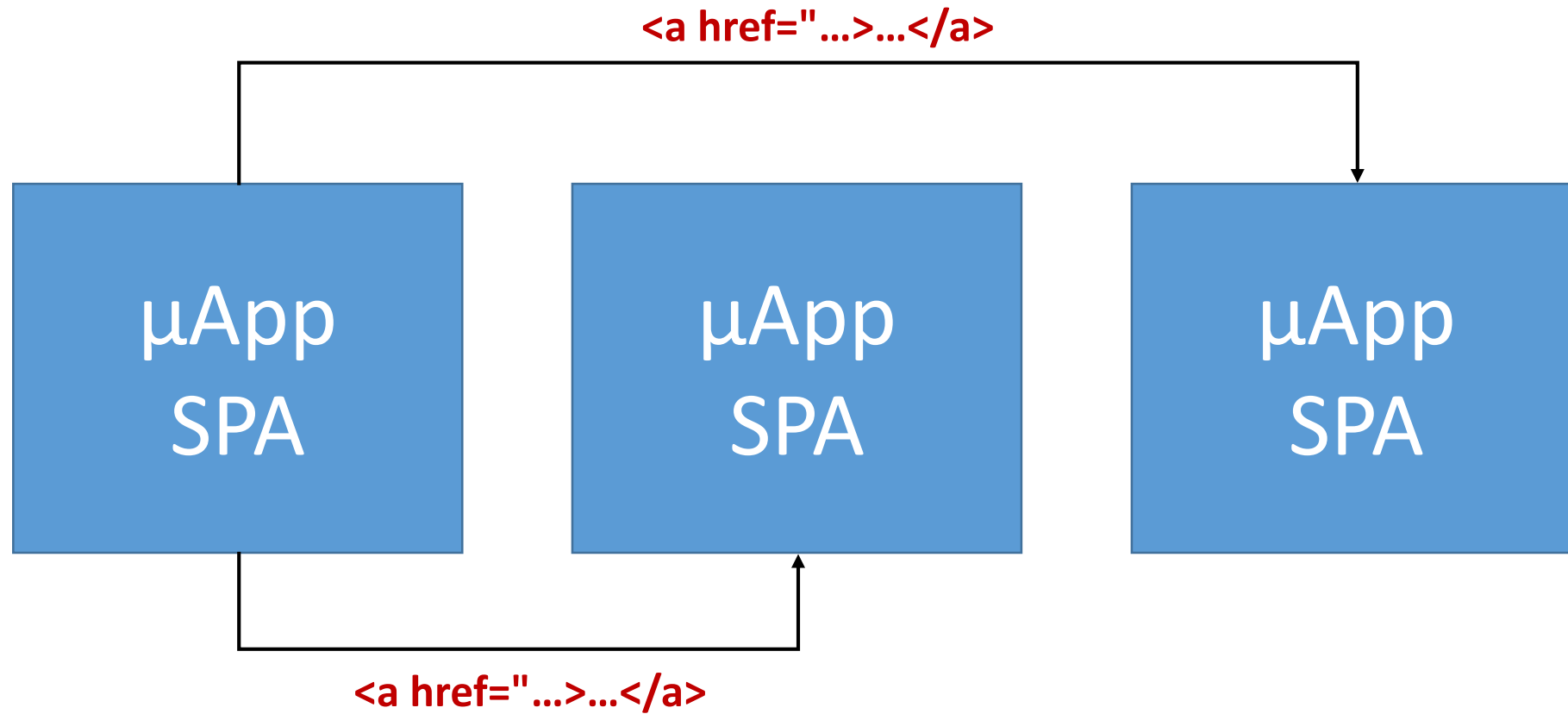
Separate
Development

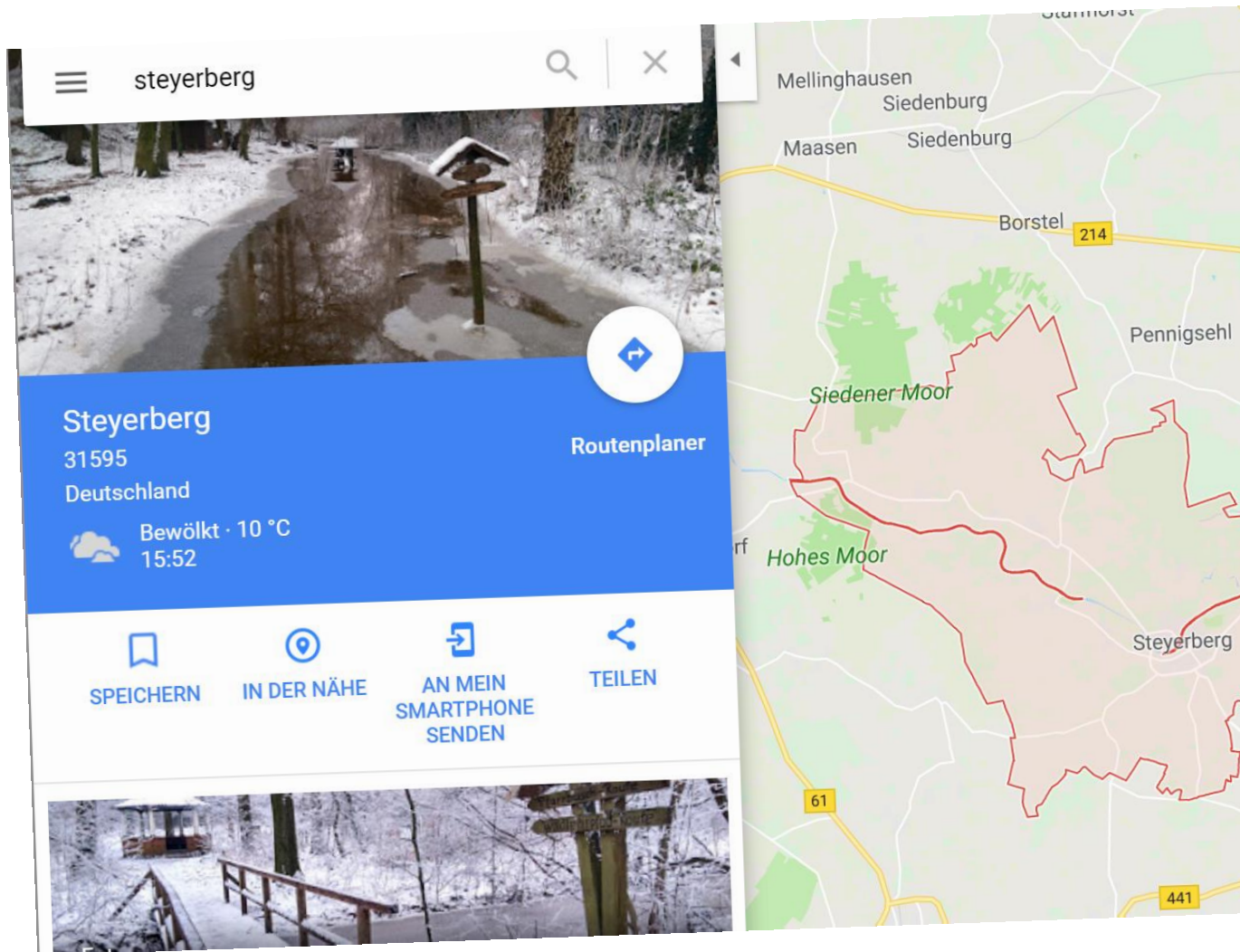
Separate
Deployment

Mixing
Technologies

Mixing
Architectures

UI Composition w/ Hyperlinks

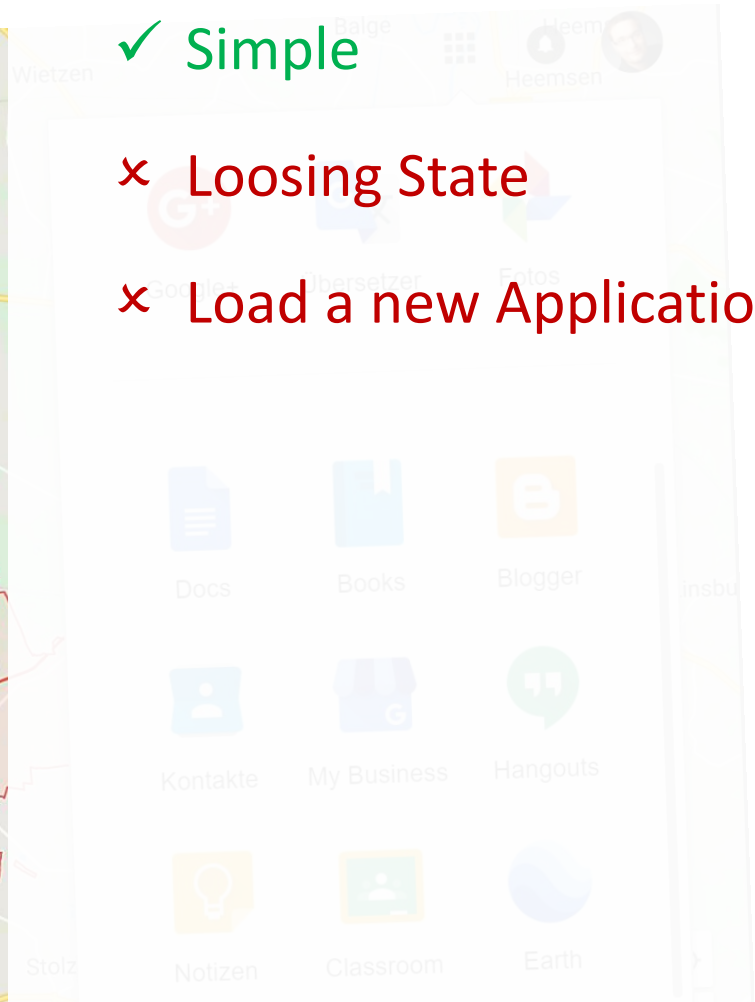


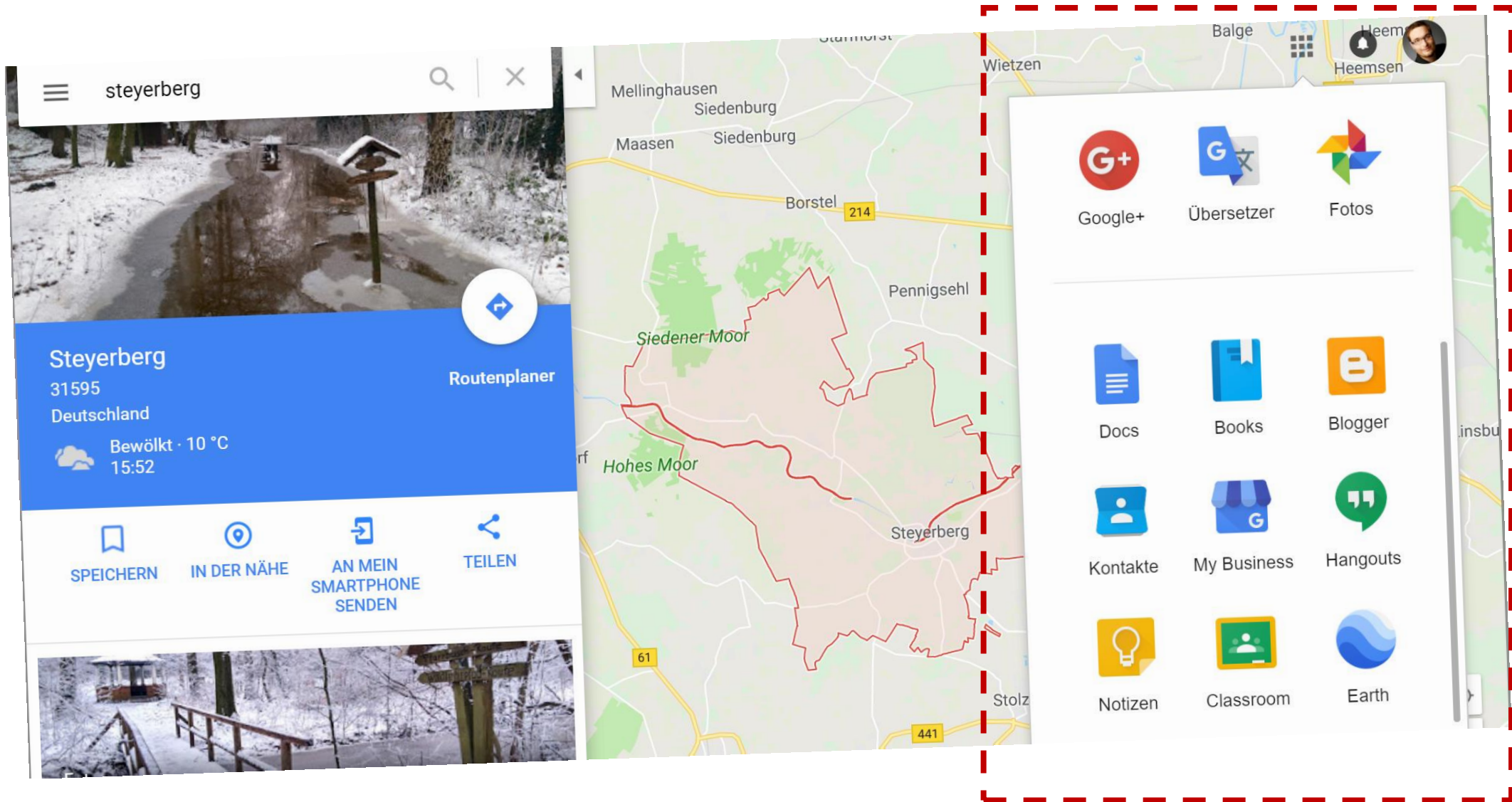


✓ Simple

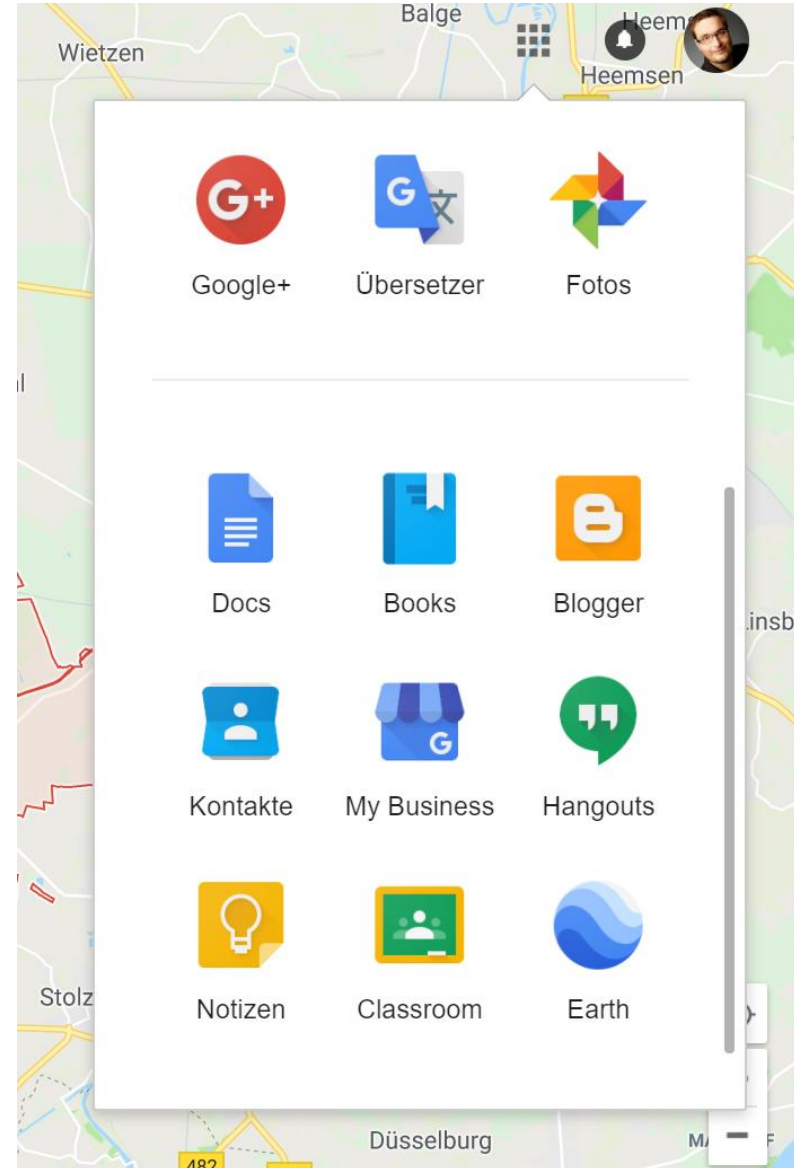
✗ Loosing State

✗ Load a new Application

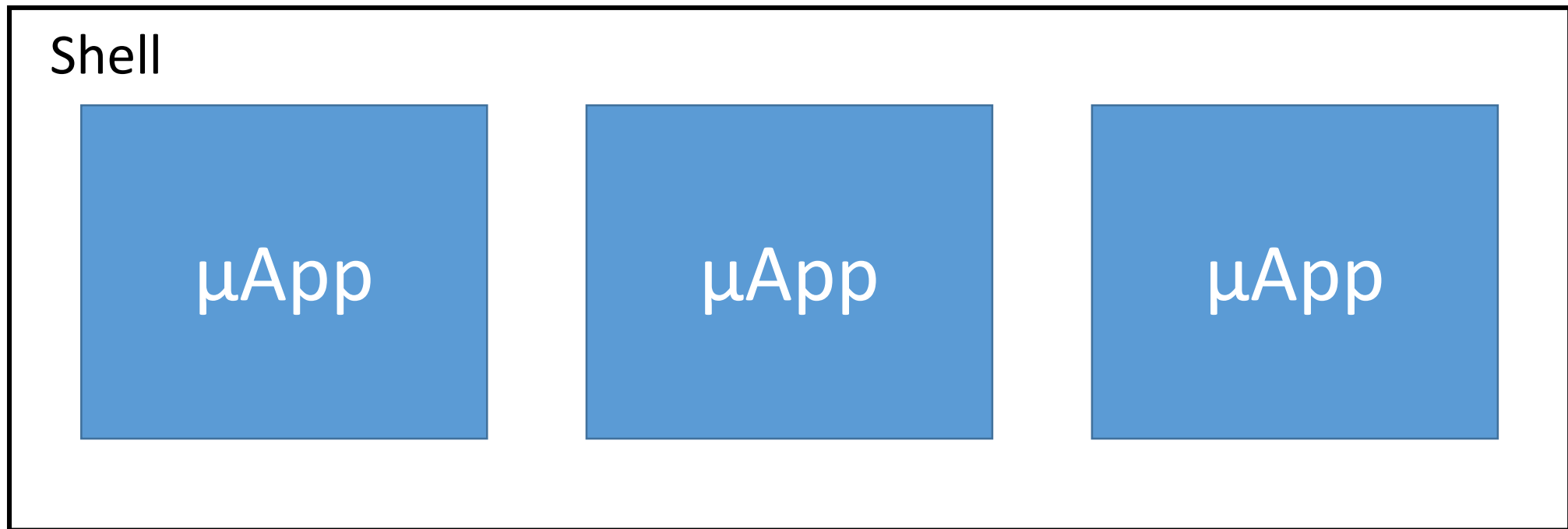




Web Components for Shared Widgets?

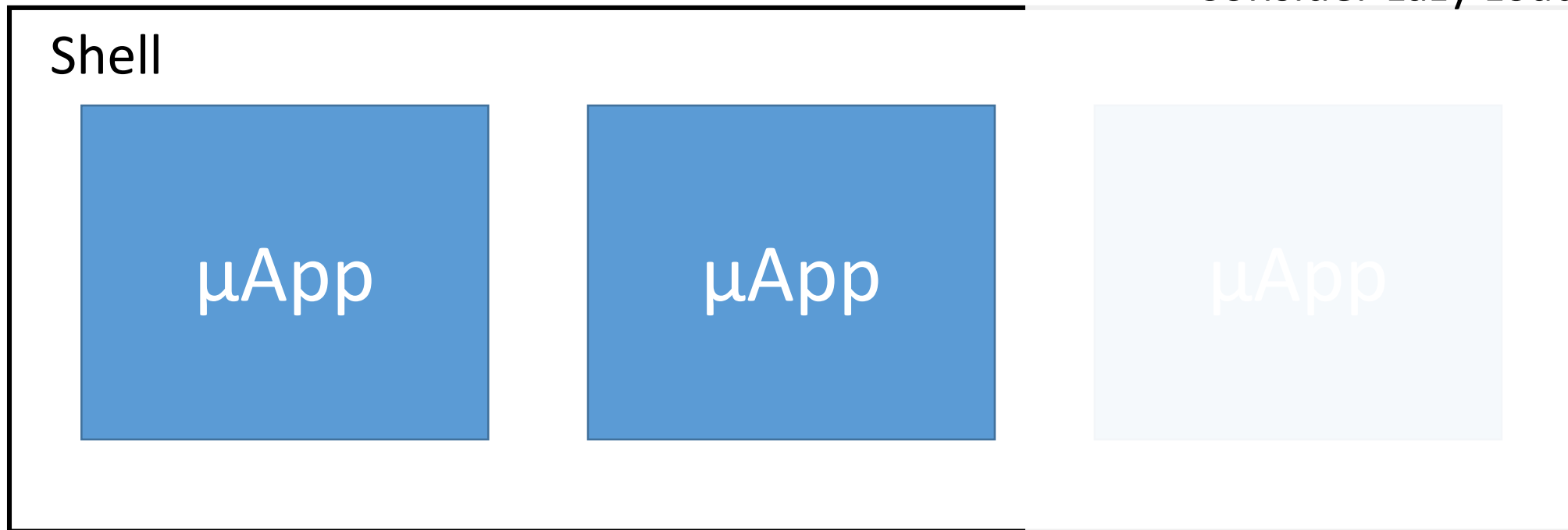


Providing a (SPA based) Shell



Providing a (SPA based) Shell

- iframes
- Bootstrapping several SPAs
- Consider Lazy Loading!





Wrap μ Apps into Web Components

- **Not a typical web component!**
- But: Unified API for Shell
- And: Nesting

DEMO

Advantages

- Several small and decoupled Projects
- Autarkic Teams
- Seperate development/ deployment
- Own decisions (technology, architecture)

Disadvantages ;-)

- Several small and decoupled Projects
- Autarkic Teams
- Seperate development/ deployment
- Own decisions (technology, architecture)

Disadvantages ;-)

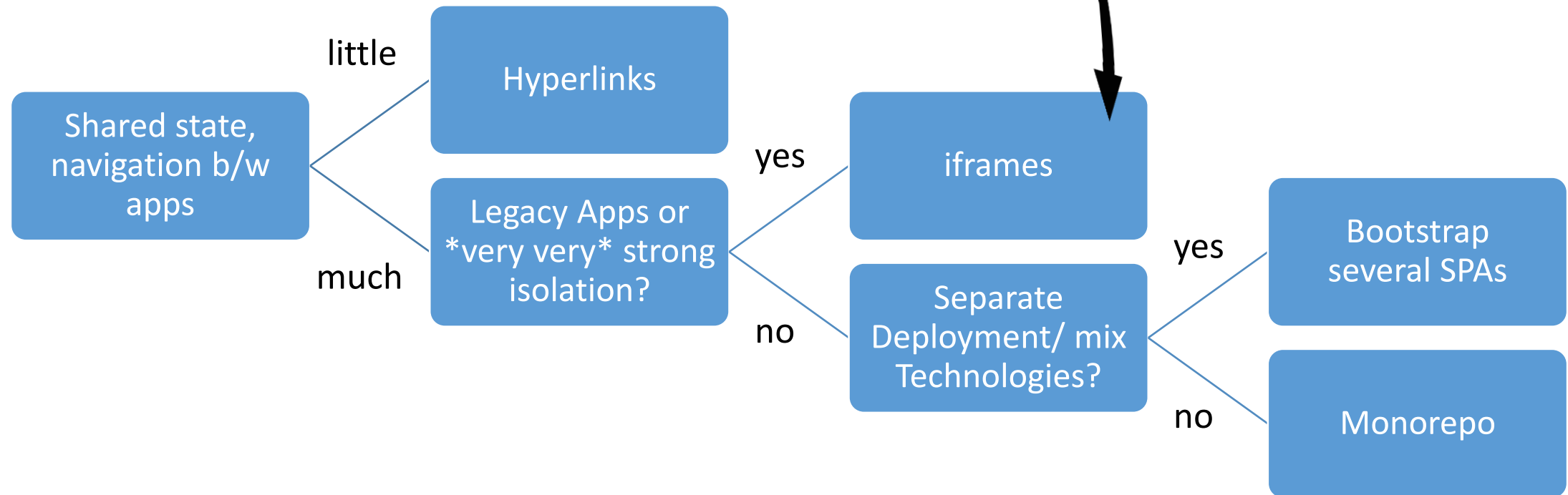
- More Bundles
- Less optimized Bundles/ duplicate Code?
- UI Consistency



4) Choosing a Solution

Some General Advice

Not a good fit for public web sites



Blog > SOFTWAREarchitekt.at

- A Software Architect's Approach Towards Using Angular (And SPAs In General) For Microservices Aka Microfrontends
- A Lightweight And Solid Approach Towards Micro Frontends (Micro Service Clients) With Angular And/Or Other Frameworks
- Micro Apps With **Web Components** Using Angular Elements

Conclusion

Packages:
Reuse

Monorepo:
Structure

Micro Apps:
Decoupling

CLI 6+

CLI 6+/ Nx

Links,
iframes,
Several SPAs



Beware the
Frankenstein-
Monster!!

Contact and Downloads

[mail] manfred.steyer@SOFTWAREarchitekt.at

[web] SOFTWAREarchitekt.at

[twitter] ManfredSteyer

Slides & Examples

