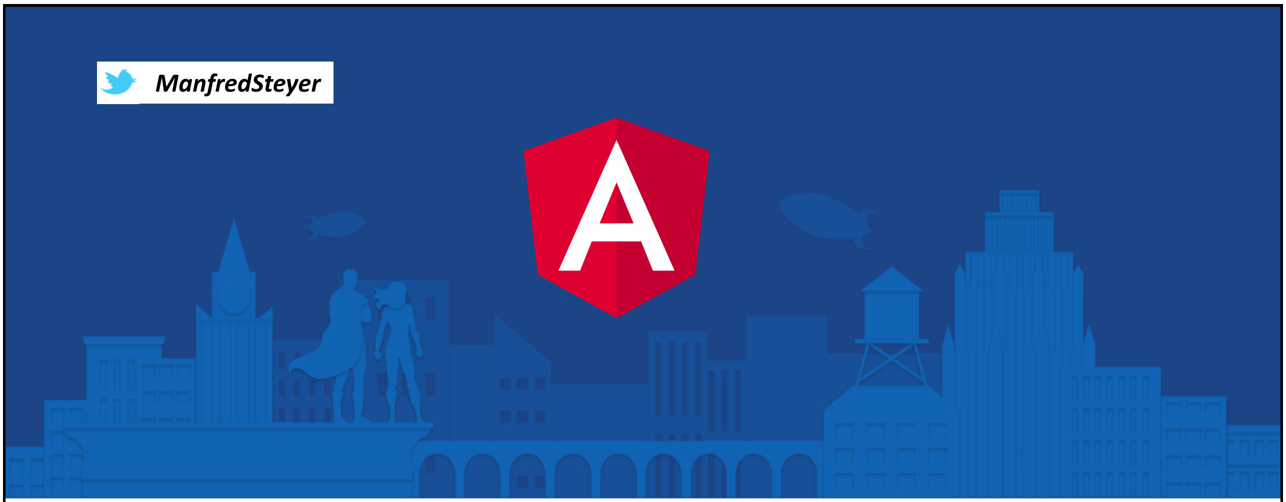


ManfredSteyer

Angular Architecture

Manfred Steyer

SOFTWARE*architekt.at*

Questions

- How to structure an Angular application?
- How to subdivide an Angular application into smaller parts?

SOFTWARE*architekt.at*

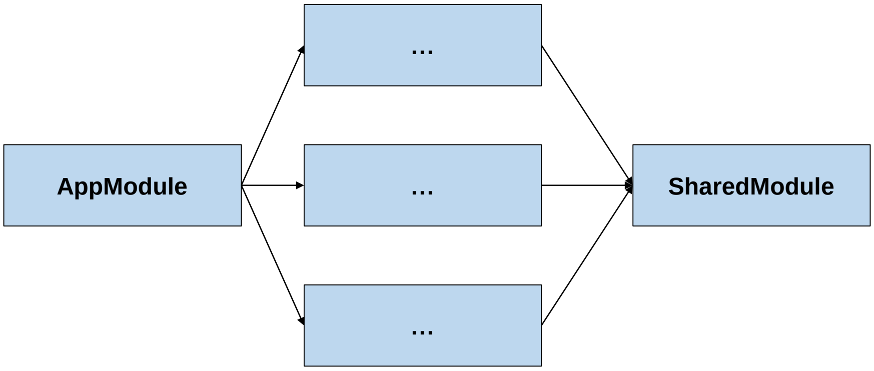
Contents

- Modules
- (npm-)Packages
- Monorepos
- Microservices

SOFTWARE architekt.at

Modules

Typical Module Structure



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

Page ▪ 5

SOFTWARE*architekt.at*

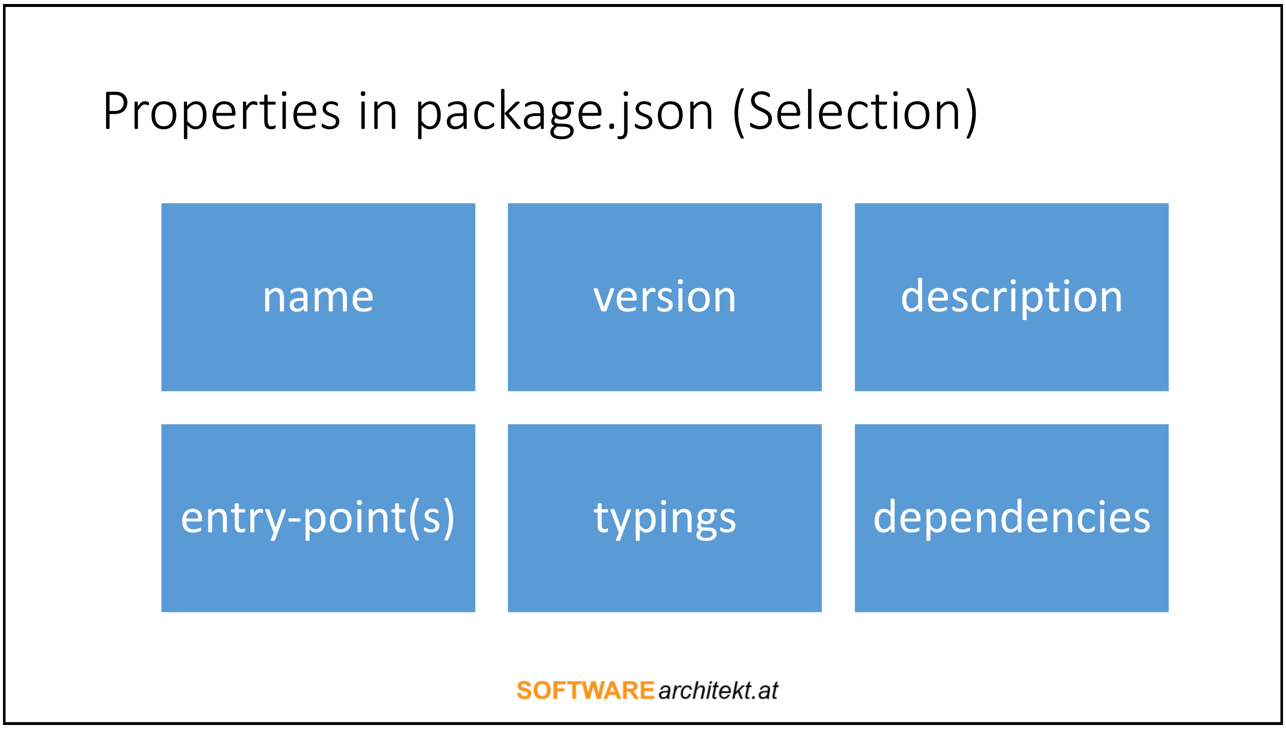
npm Packages

Why npm Packages?

- Reusable Logic
- Structuring big Applications

Structure of an npm Package

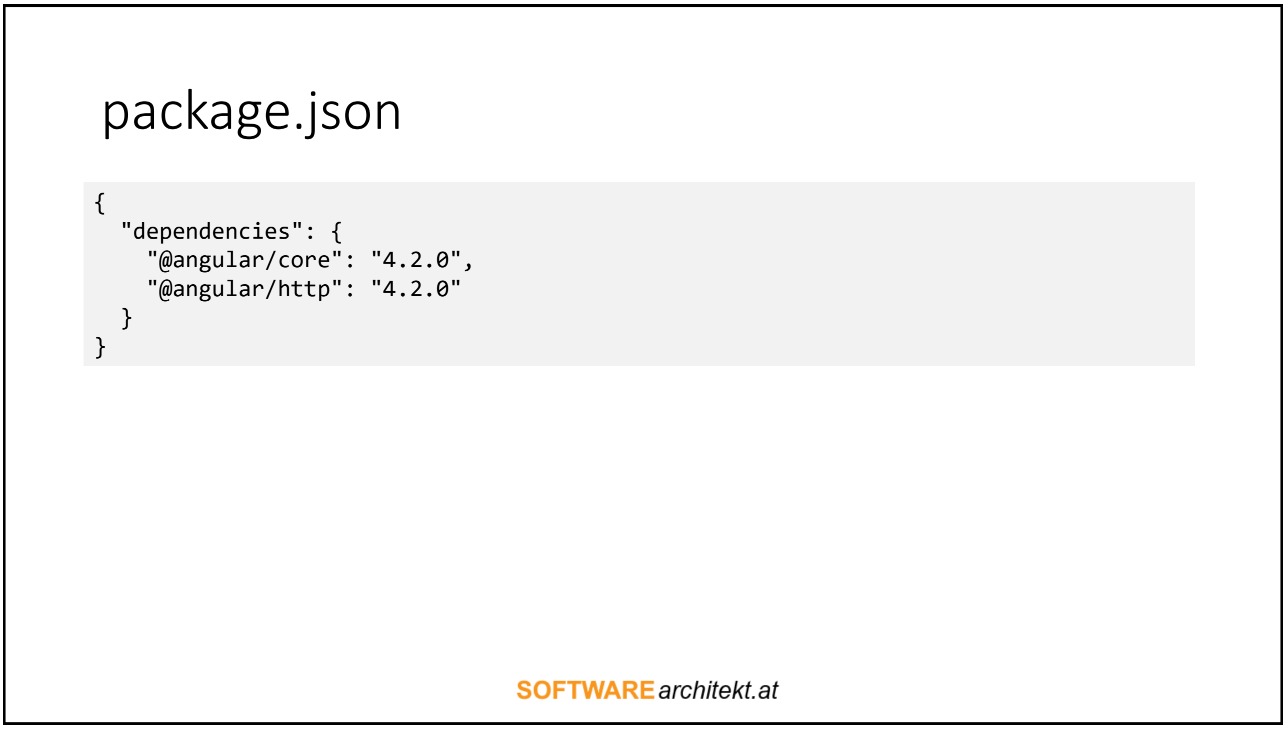
- /node_modules
- your-stuff
- package.json



Properties in package.json (Selection)

name	version	description
entry-point(s)	typings	dependencies

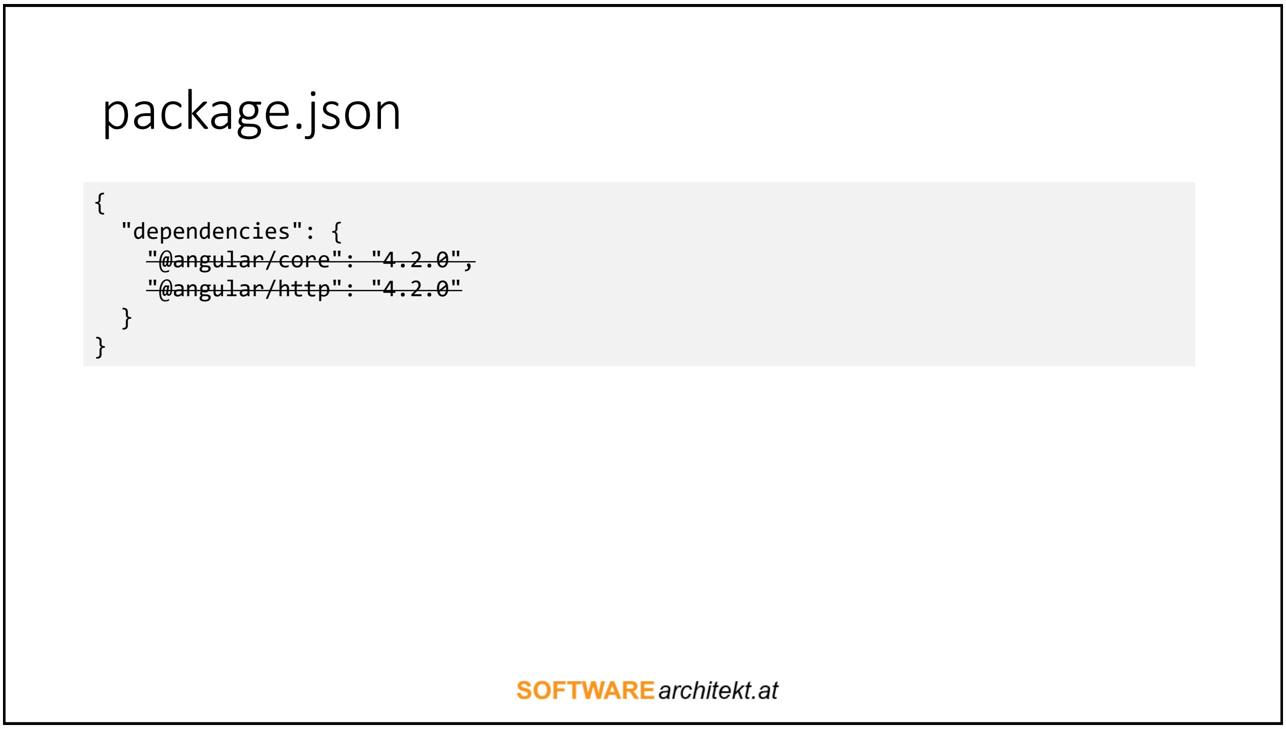
SOFTWARE*architekt.at*



package.json

```
{
  "dependencies": {
    "@angular/core": "4.2.0",
    "@angular/http": "4.2.0"
  }
}
```

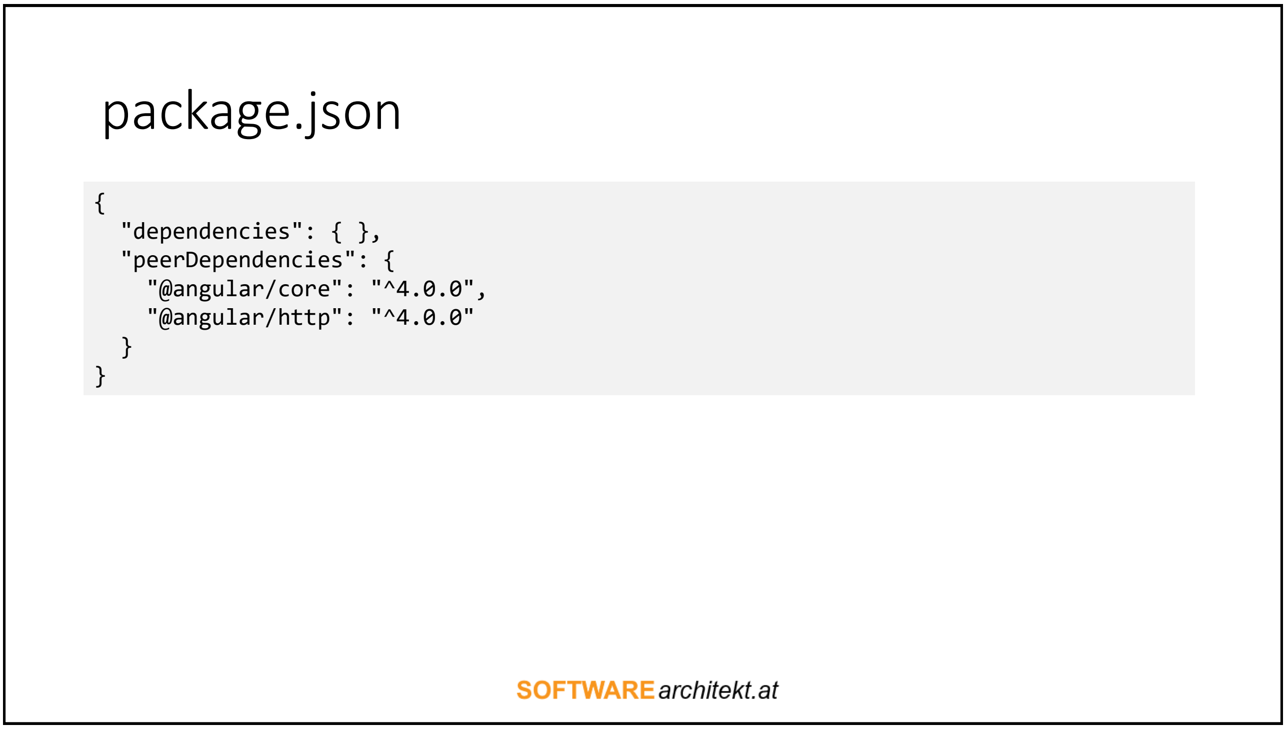
SOFTWARE*architekt.at*



package.json

```
{
  "dependencies": {
    ~~"@angular/core": "4.2.0",~~
    ~~"@angular/http": "4.2.0"~~
  }
}
```

SOFTWARE architekt.at



package.json

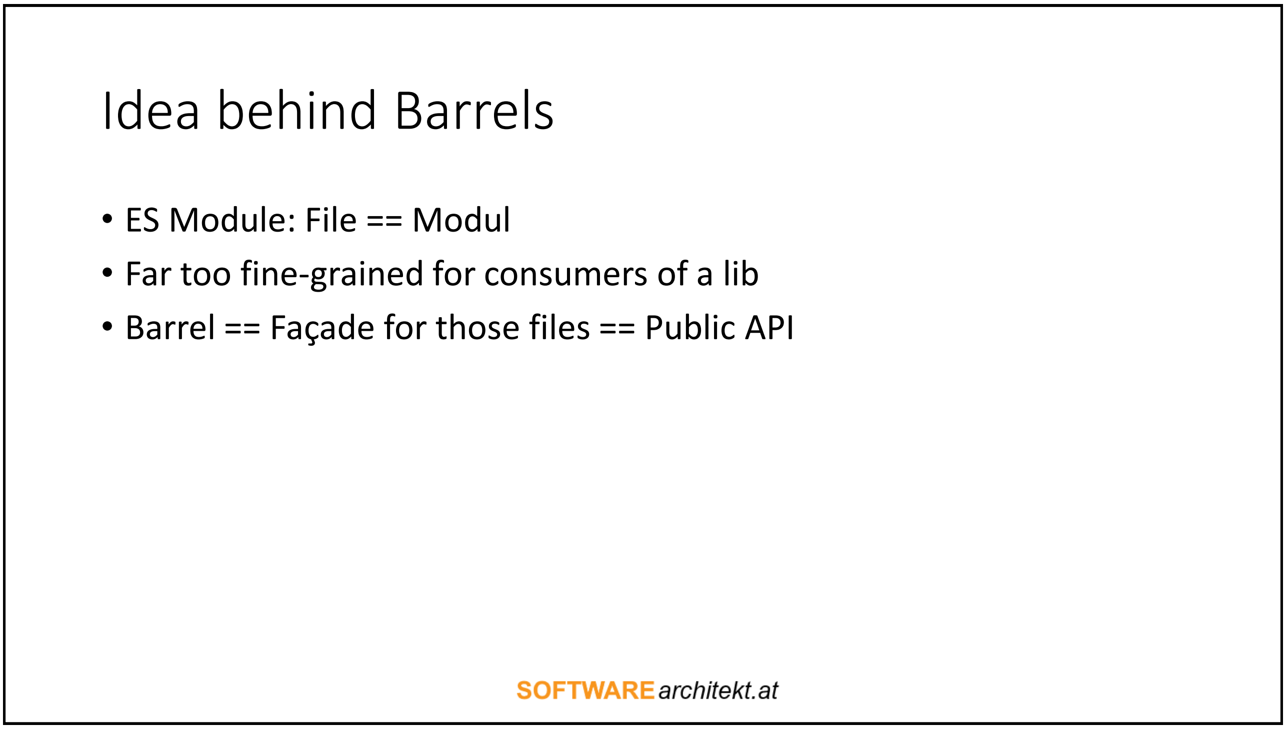
```
{
  "dependencies": { },
  "peerDependencies": {
    "@angular/core": "^4.0.0",
    "@angular/http": "^4.0.0"
  }
}
```

SOFTWARE architekt.at

package.json

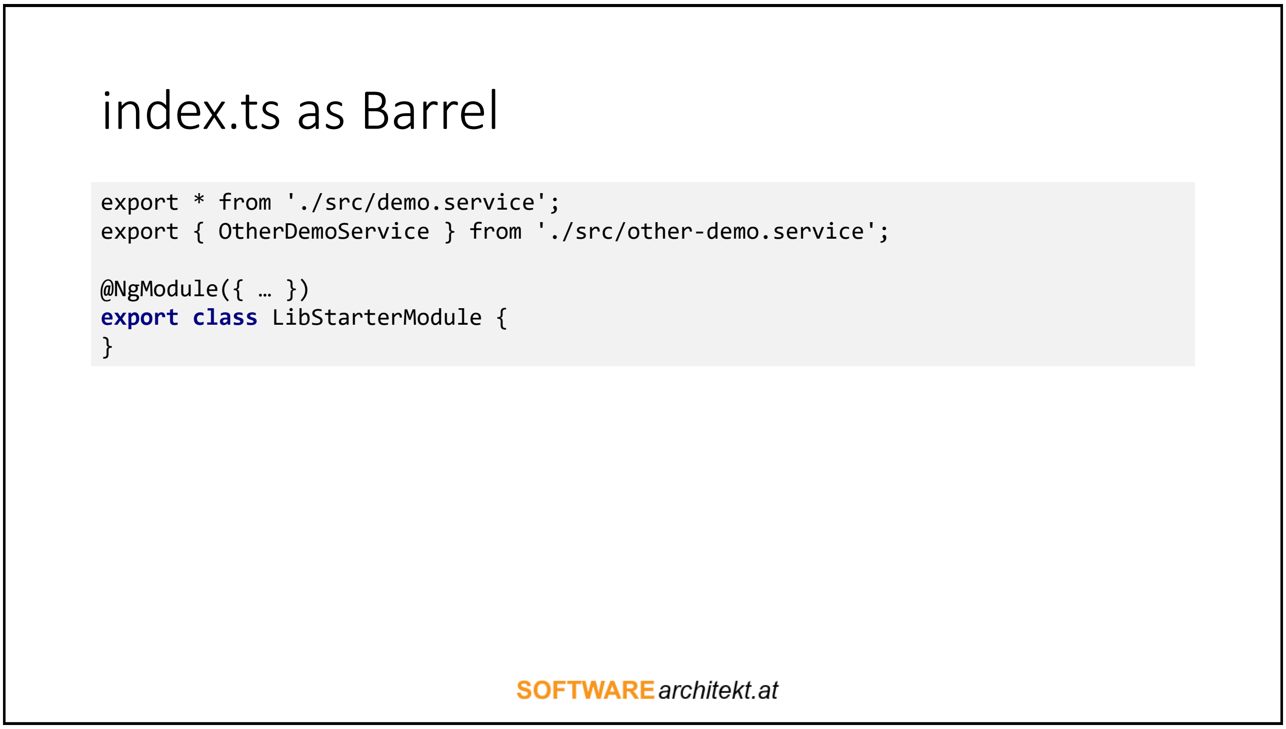
```
{
  "dependencies": { },
  "peerDependencies": {
    "@angular/core": "^4.0.0",
    "@angular/http": "^4.0.0"
  },
  "devDependencies": {
     [...]
  }
}
```

Barrels



Idea behind Barrels

- ES Module: File == Modul
- Far too fine-grained for consumers of a lib
- Barrel == Façade for those files == Public API



index.ts as Barrel

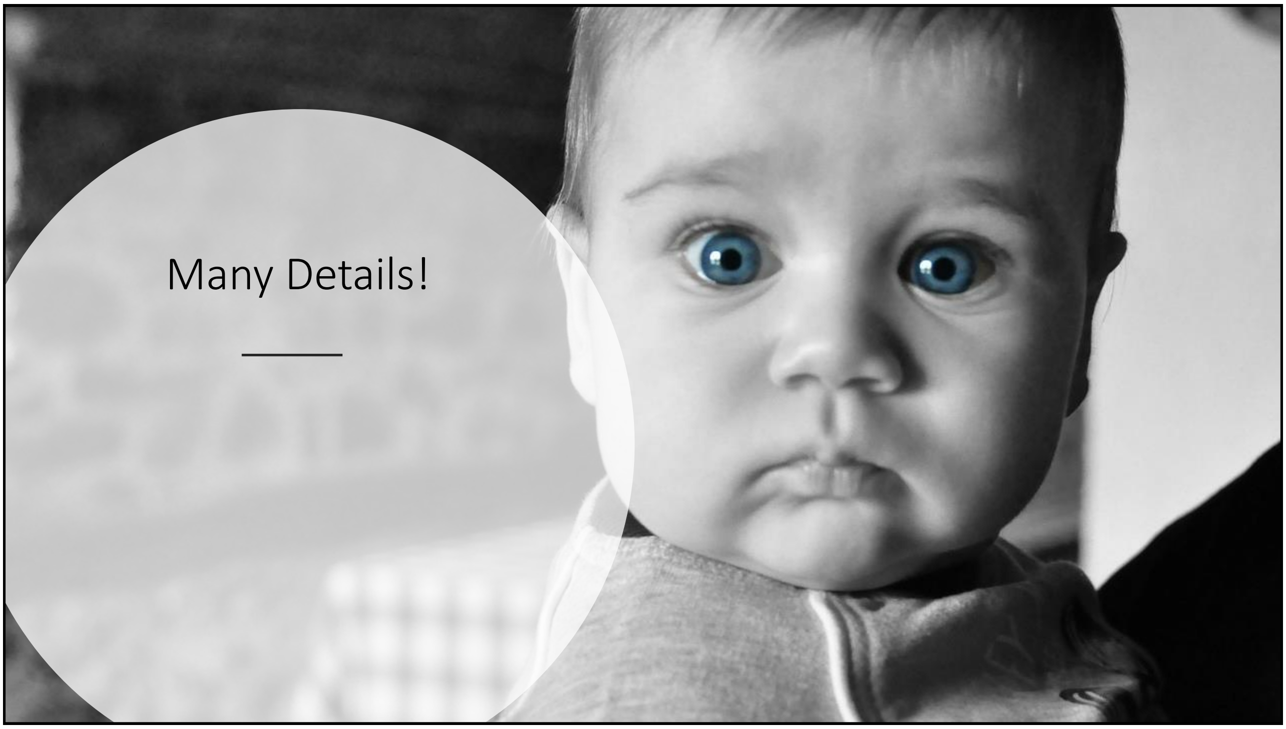
```
export * from './src/demo.service';
export { OtherDemoService } from './src/other-demo.service';

@NgModule({ … })
export class LibStarterModule {
}
```

Project Start

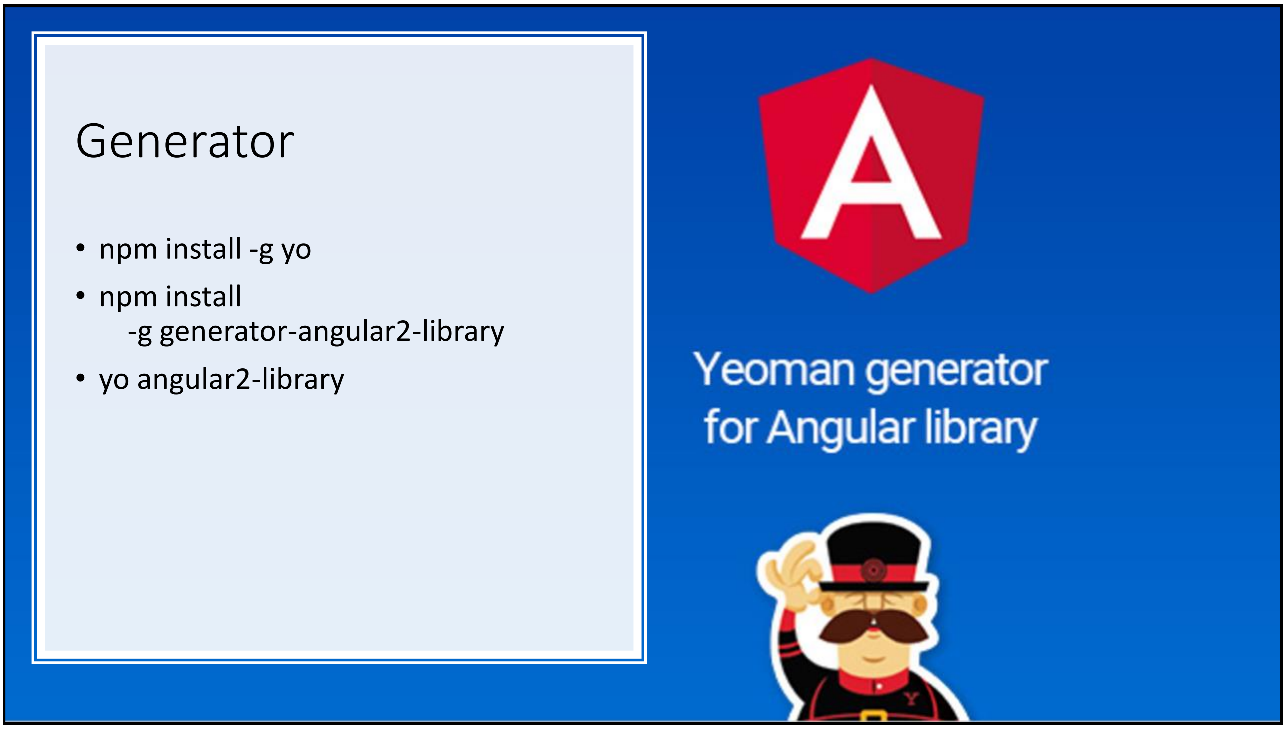
ENGINE
START
STOP
OFF

https://goo.gl/hjt7G3

Angular Package Format



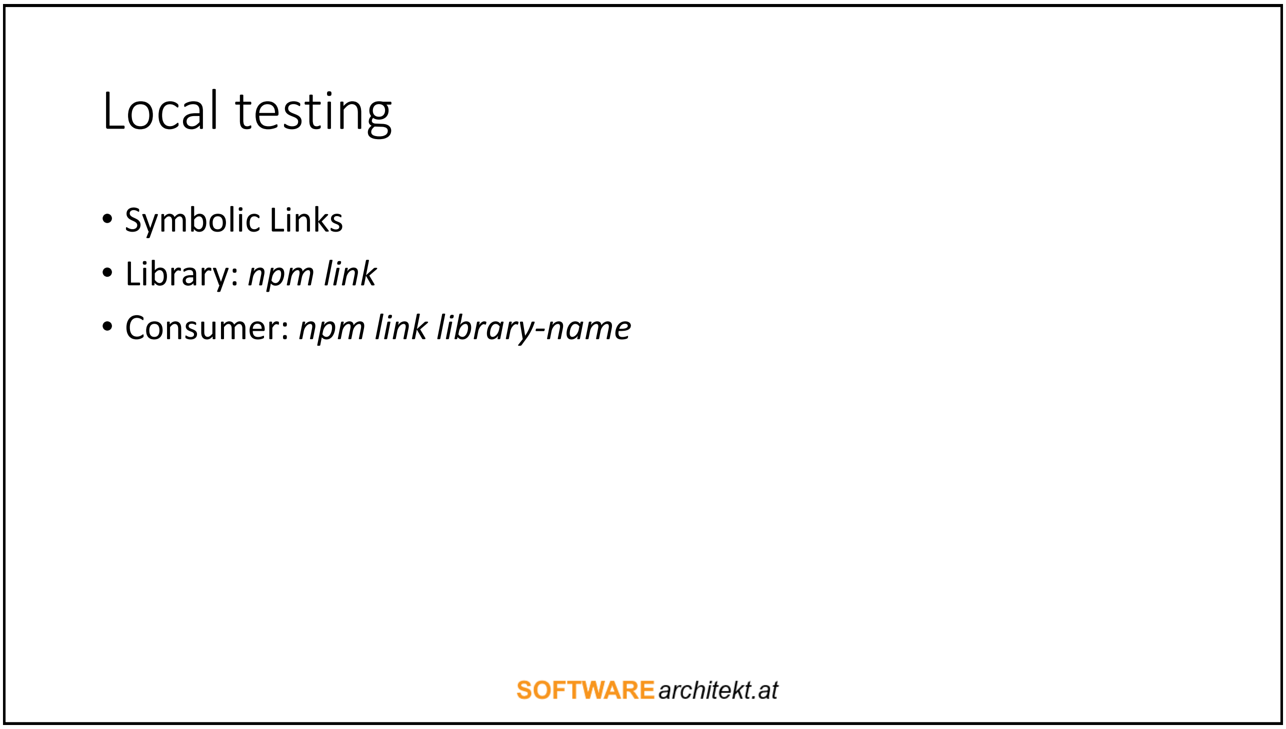
Many Details!



Generator

- npm install -g yo
- npm install
  -g generator-angular2-library
- yo angular2-library

Test and Deployment



Local testing

- Symbolic Links
- Library: *npm link*
- Consumer: *npm link library-name*

SOFTWARE *architekt.at*

Lessons Learned

- Use "playground app" within lib for debugging (besides unit tests)
- Npm link: ng serve --preserve-symlinks
- Testing before production: link dist folder and test with AOT*

* will be default soon

SOFTWARE architekt.at

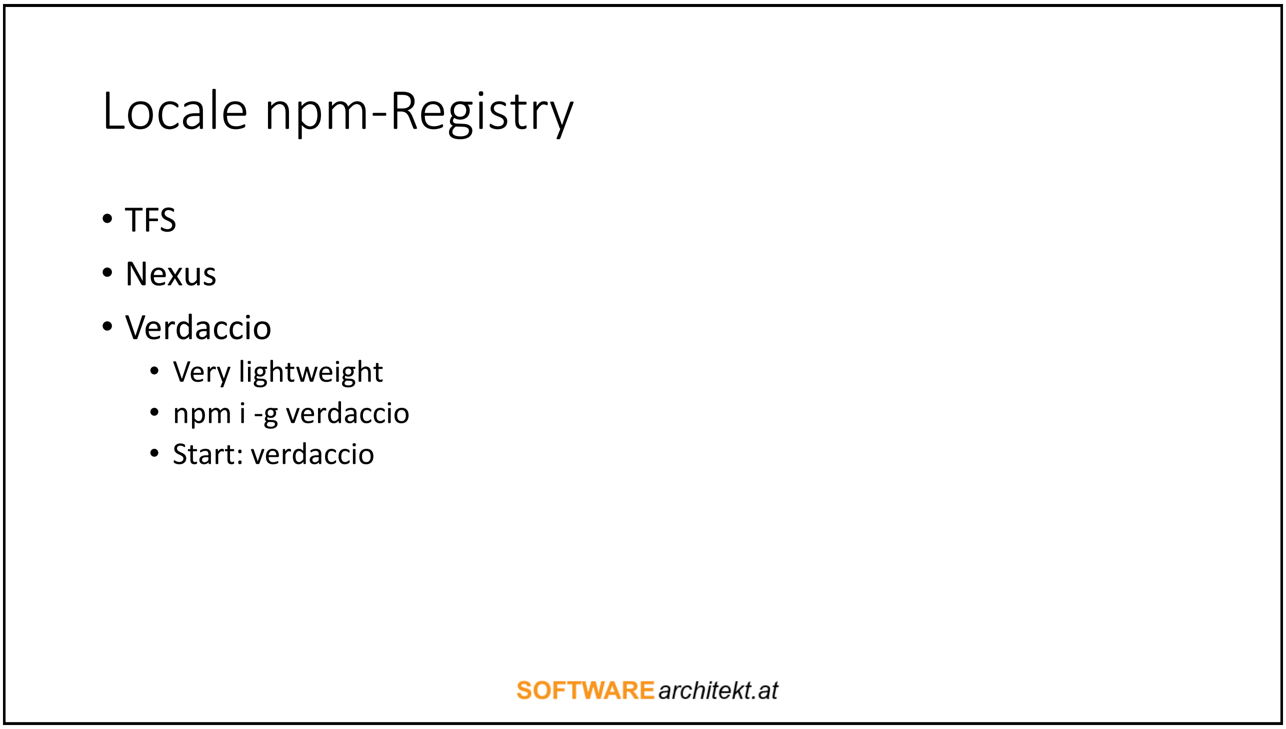
Publishing to npm Registry

- Increment version in package.json
  - npm version [patch | minor | major | version]
- npm run build

- cd ../dist
- npm publish --registry http://localhost:4873

- npm install --registry http://localhost:4873

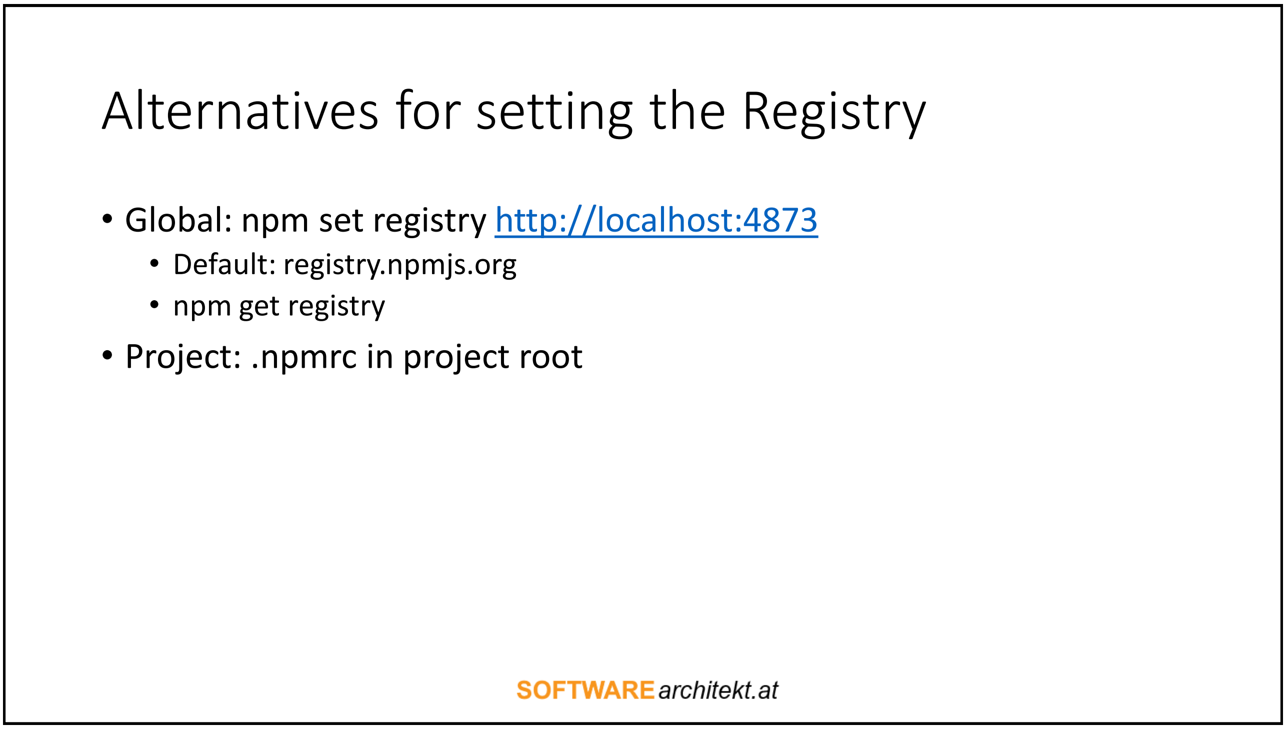
SOFTWARE architekt.at



Locale npm-Registry

- TFS
- Nexus
- Verdaccio
  - Very lightweight
  - npm i -g verdaccio
  - Start: verdaccio

SOFTWARE architekt.at



Alternatives for setting the Registry

- Global: npm set registry http://localhost:4873
  - Default: registry.npmjs.org
  - npm get registry
- Project: .npmrc in project root

SOFTWARE architekt.at

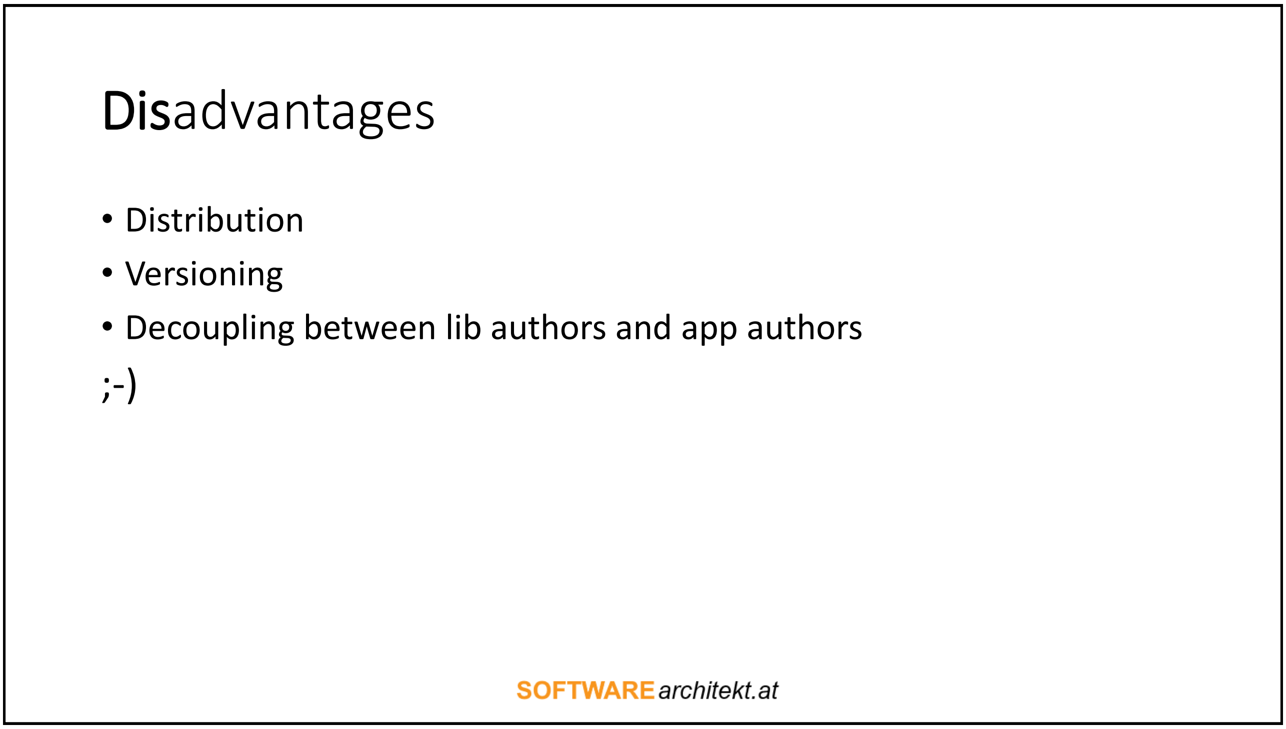
DEMO

SOFTWAREarchitekt.at

Advantages

- Distribution
- Versioning
- Decoupling between lib authors and app authors

SOFTWAREarchitekt.at

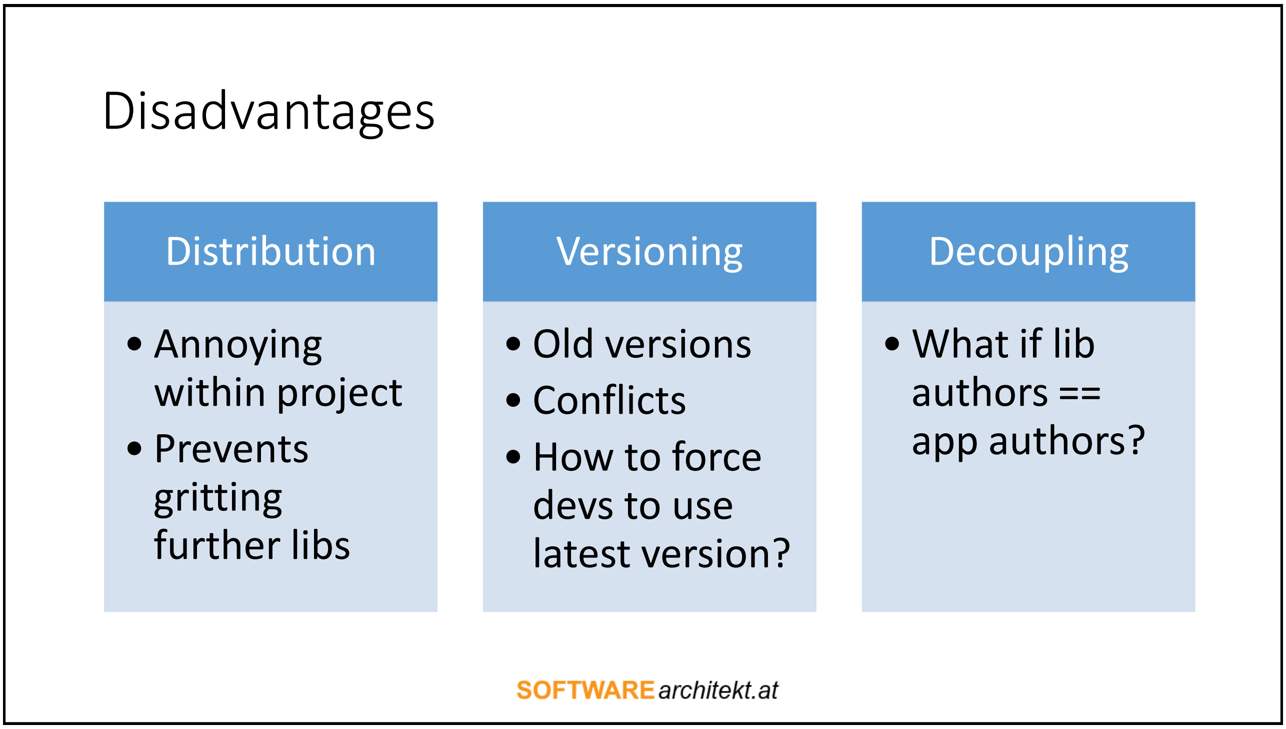


Disadvantages

- Distribution
- Versioning
- Decoupling between lib authors and app authors

;-)

SOFTWARE architekt.at



Disadvantages

Distribution

- Annoying within project
- Prevents gritting further libs

Versioning

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

Decoupling

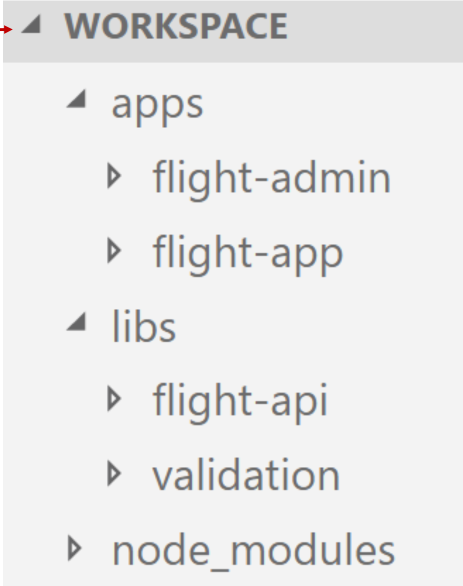
- What if lib authors == app authors?

SOFTWARE architekt.at

Monorepos

Monorepo Structure

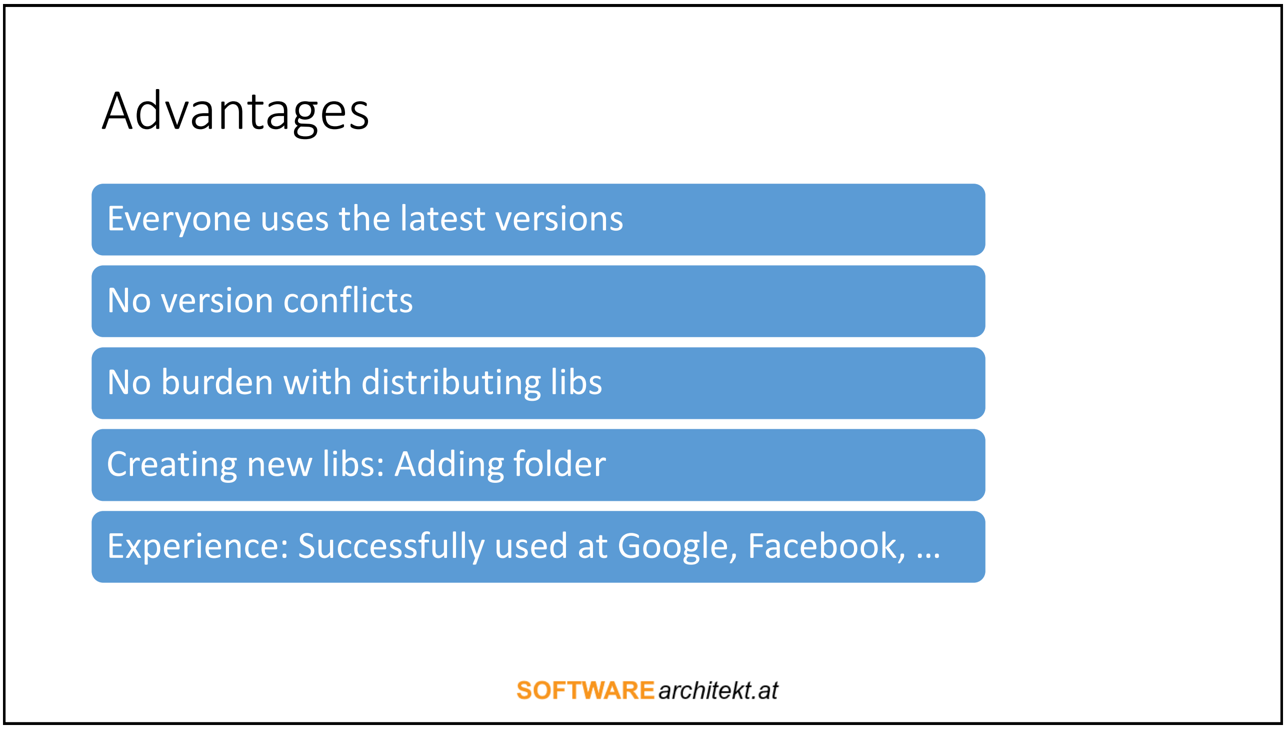
Project Workspace **or** global Workspace



```
◢ WORKSPACE
   ◢ apps
      ▷ flight-admin
      ▷ flight-app
   ◢ libs
      ▷ flight-api
      ▷ validation
   ▷ node_modules
```

Define Code Owners for different folders

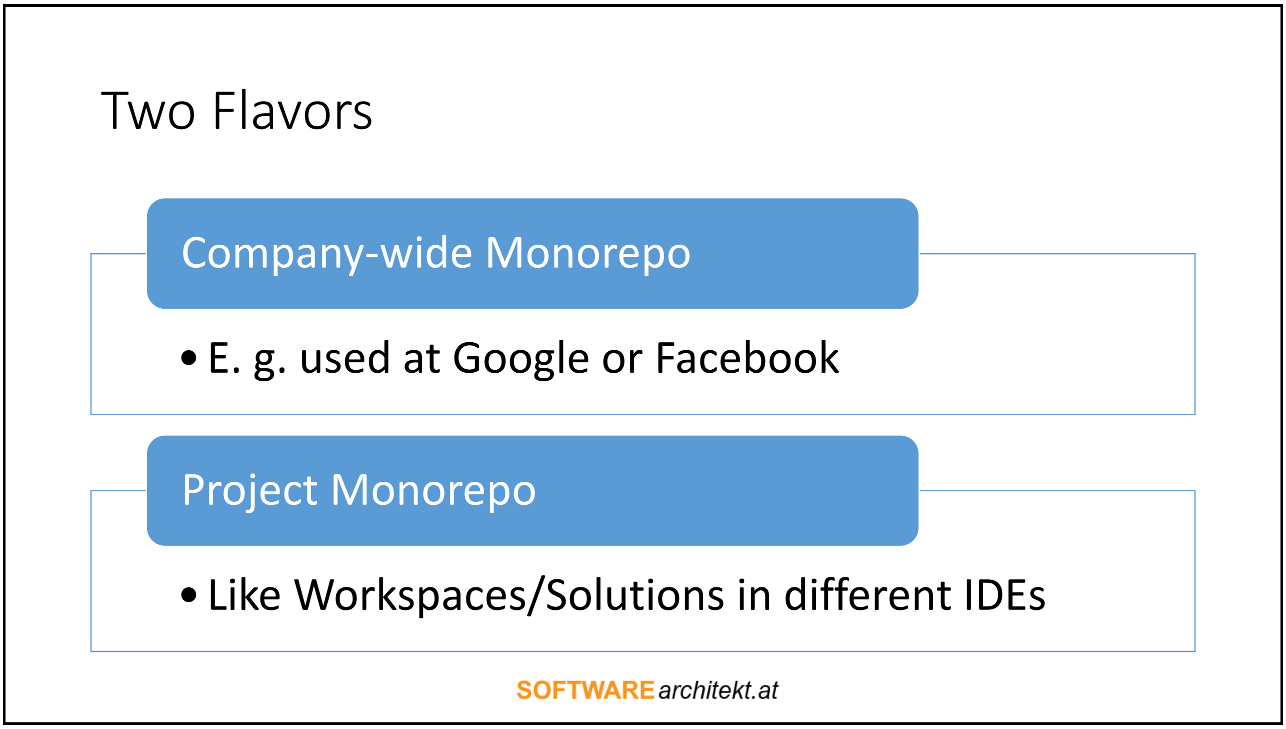
SOFTWARE*architekt.at*



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, …

SOFTWARE architekt.at



Two Flavors

Company-wide Monorepo

- E. g. used at Google or Facebook

Project Monorepo

- Like Workspaces/Solutions in different IDEs

SOFTWARE architekt.at

Tooling & Generator

https://nrwl.io/nx

Nrwl Extensions for Angular

An open source toolkit for enterprise Angular applications.

SOFTWARE *architekt.at*

Usage

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

create-nx-workspace myworkspace
ng generate app myapp
ng generate lib mymodule

ng serve --app=myapp
ng generate component myButton --app=mymodule
ng build --app=myapp
```

SOFTWARE *architekt.at*

DEMO

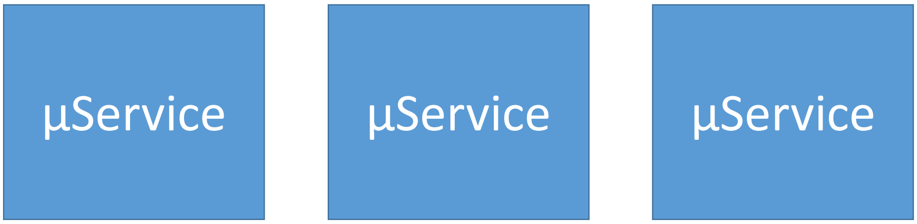
SOFTWARE architekt.at

Microservices

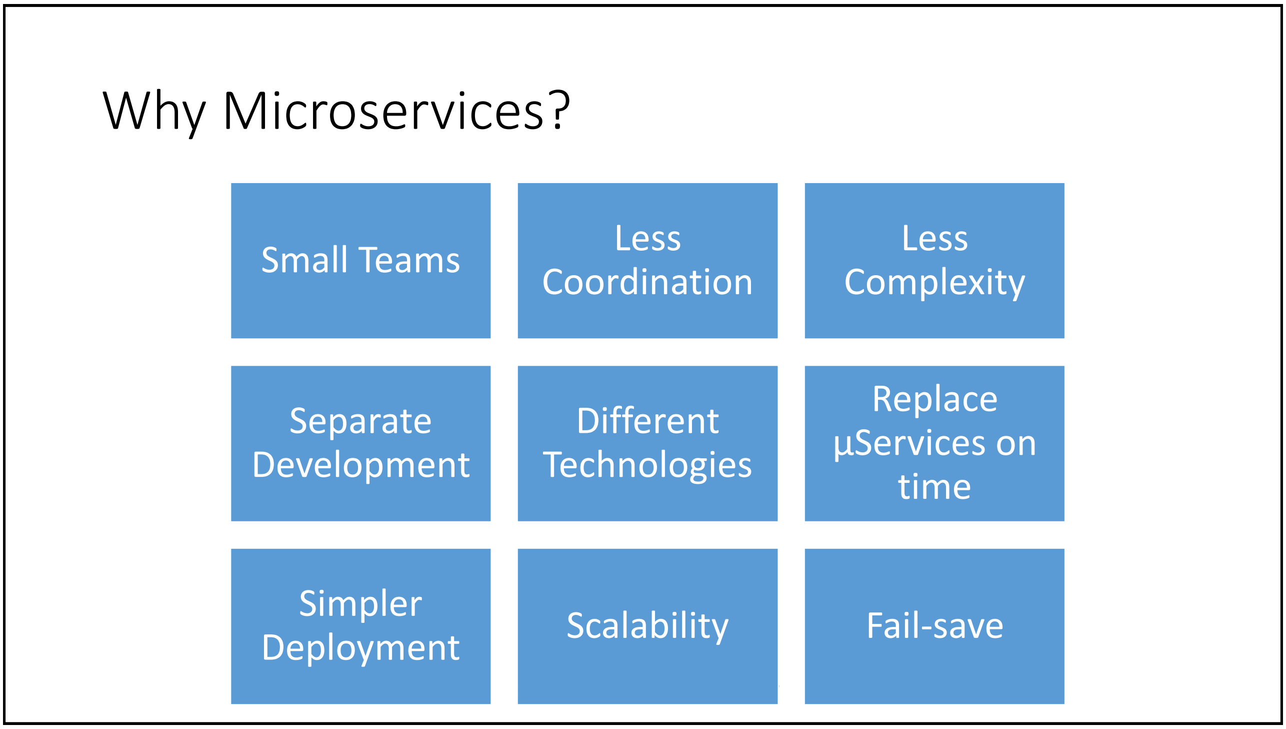
Idea

SOFTWARE architekt.at

Idea

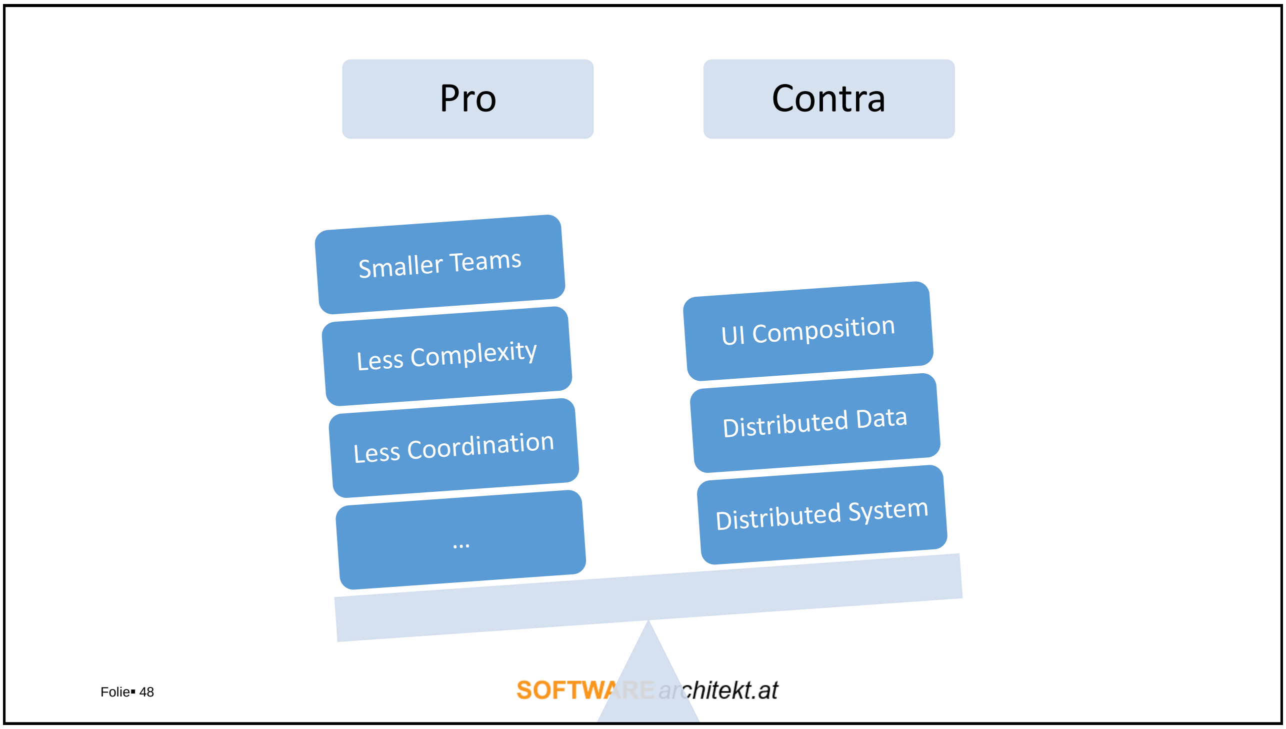


SOFTWARE architekt.at



Why Microservices?

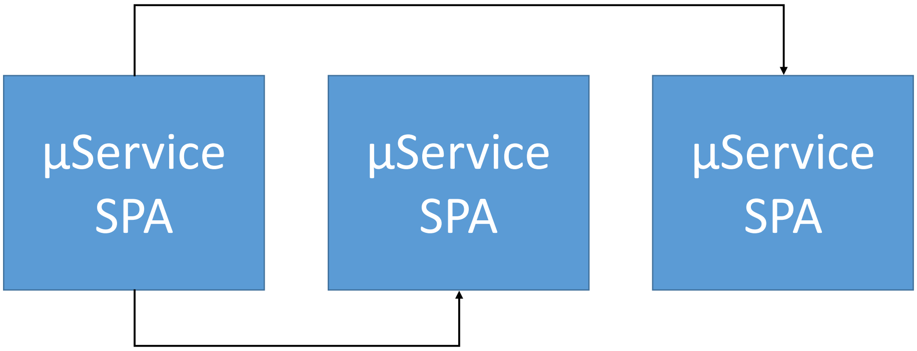
Small Teams	Less Coordination	Less Complexity
Separate Development	Different Technologies	Replace µServices on time
Simpler Deployment	Scalability	Fail-save



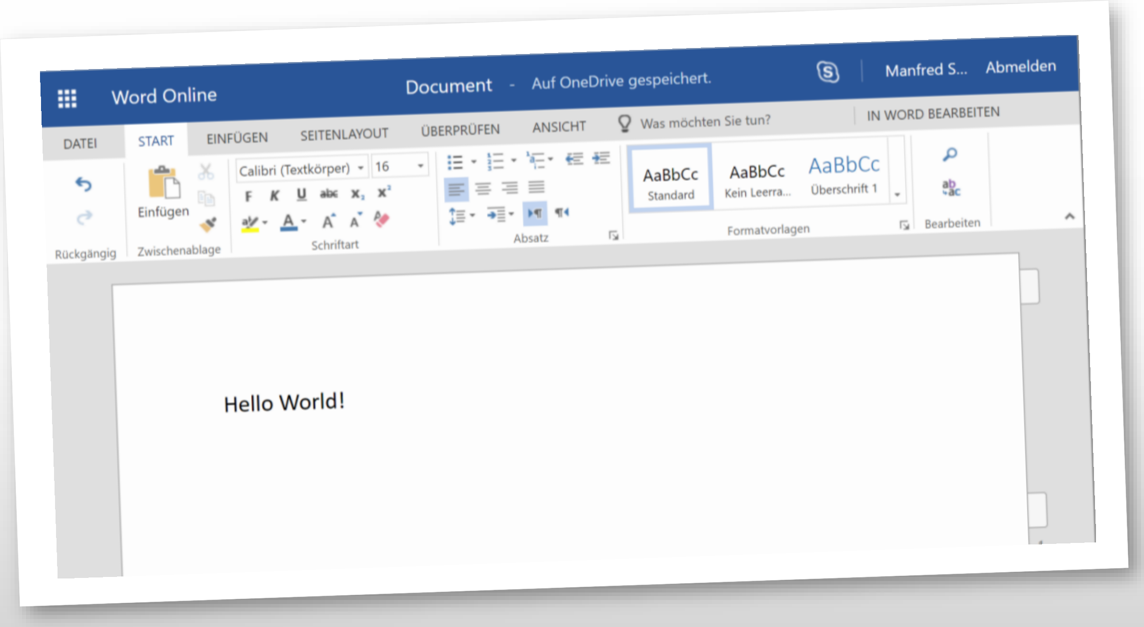
UI Composition w/ Hyperlinks

+ Easy
+ Isolation
+ Separate Deployment
+ Separate Development

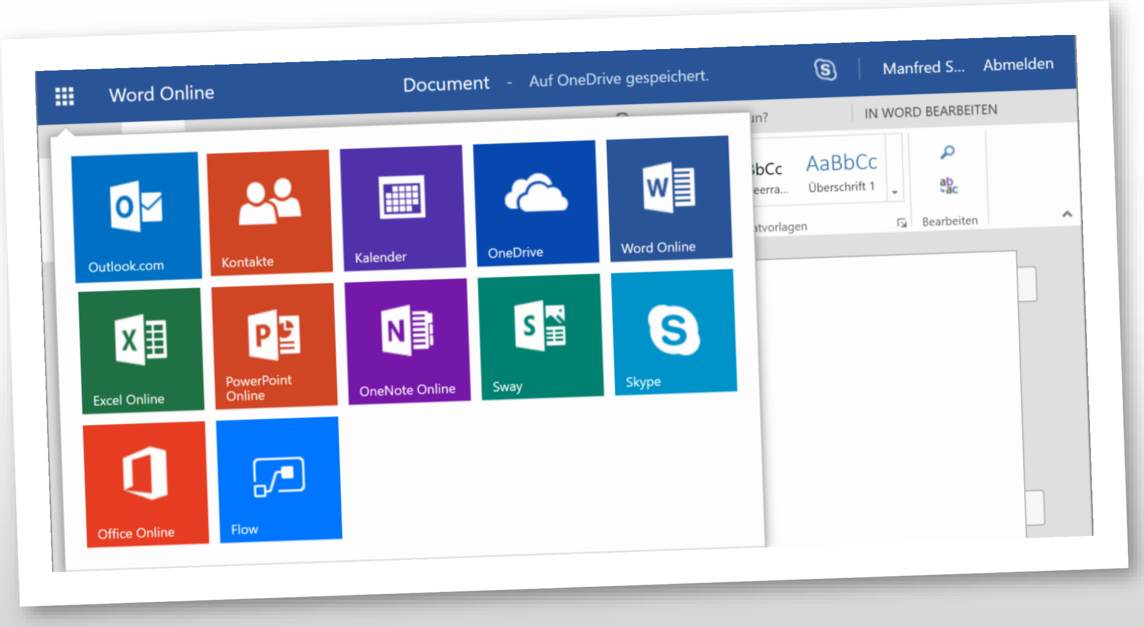
- Nesting
- No SPA
- Loosing state

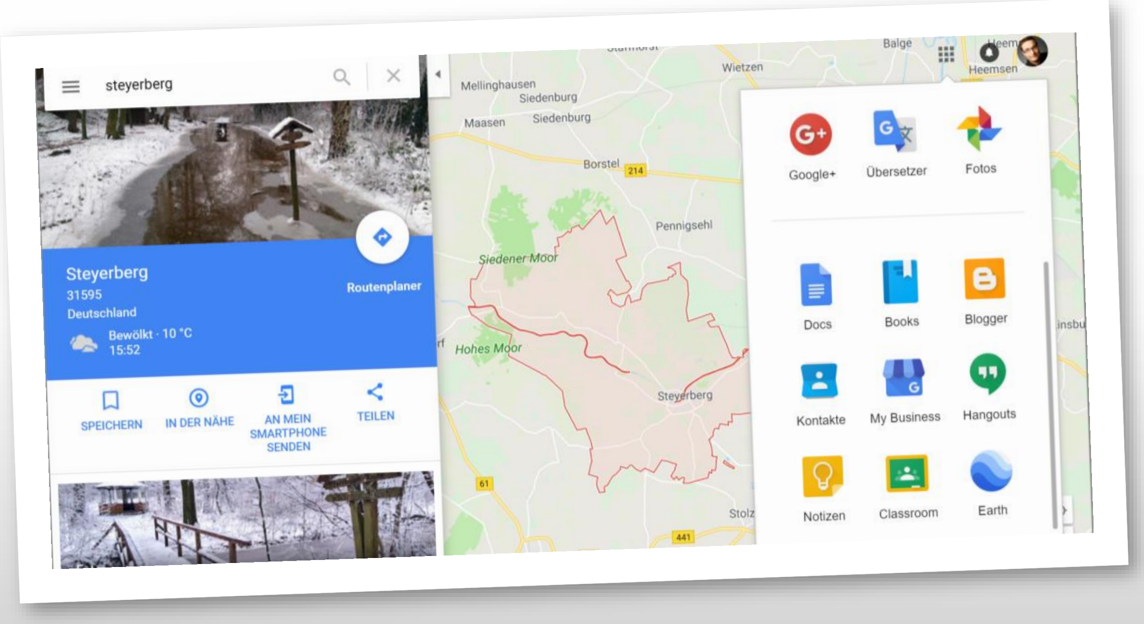


SOFTWARE_architekt.at_



Hello World!


Word Online
Document - Auf OneDrive gespeichert.
Manfred S...
Abmelden
IN WORD BEARBEITEN
AaBbCc
Überschrift 1
Bearbeiten
Outlook.com
Kontakte
Kalender
OneDrive
Word Online
Excel Online
PowerPoint Online
OneNote Online
Sway
Skype
Office Online
Flow


steyerberg
Steyerberg
31595
Deutschland
Bewölkt · 10 °C
15:52
Routenplaner
SPEICHERN
IN DER NÄHE
AN MEIN SMARTPHONE SENDEN
TEILEN
Mellinghausen
Siedenburg
Maasen
Siedenburg
Wietzen
Balge
Heemsen
Borstel
214
Pennigsehl
Siedener Moor
Hohes Moor
Steyerberg
61
441
Stolz
Google+
Übersetzer
Fotos
Docs
Books
Blogger
Kontakte
My Business
Hangouts
Notizen
Classroom
Earth

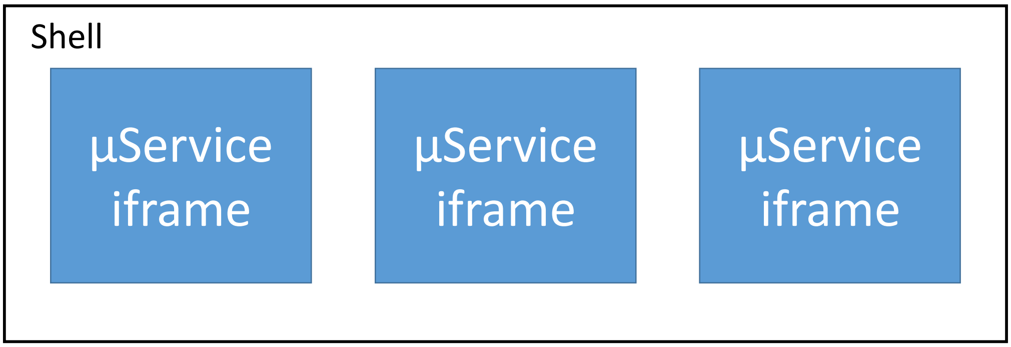
UI Composition w/ iframes

Wait! WTF? iframes?

UI Composition w/ iframes

+ Isolation
+ SPA
+ Can hold state
+ Messaging API

+ Separate Deployment
+ Separate Development
- iframes ;-)
- Nesting
- Overlapping elements



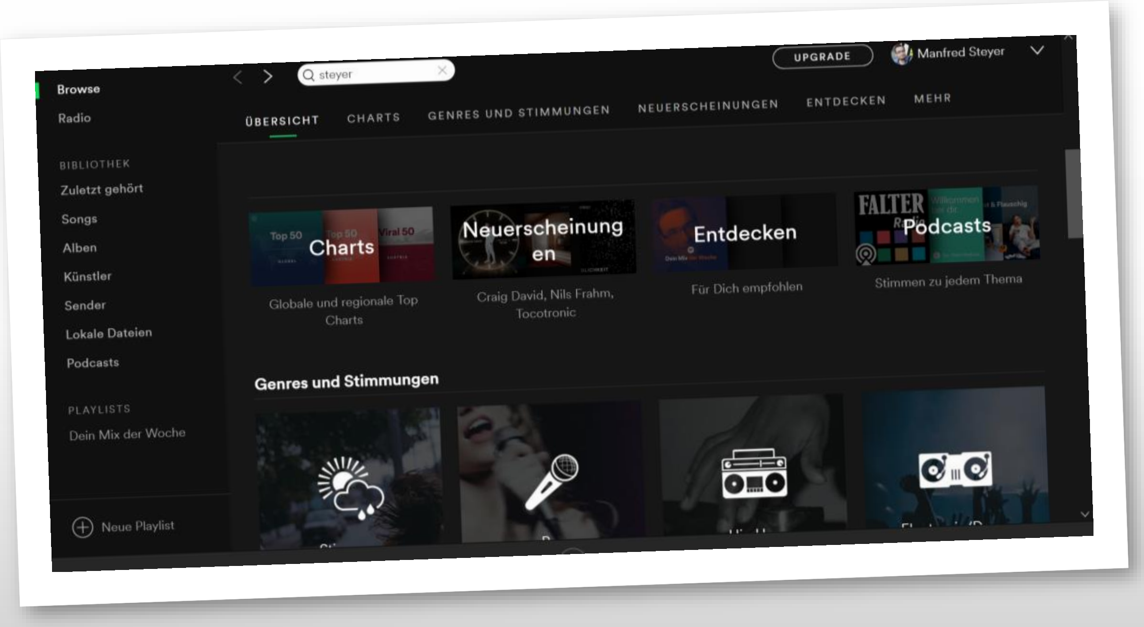
Shell

µService iframe

µService iframe

µService iframe

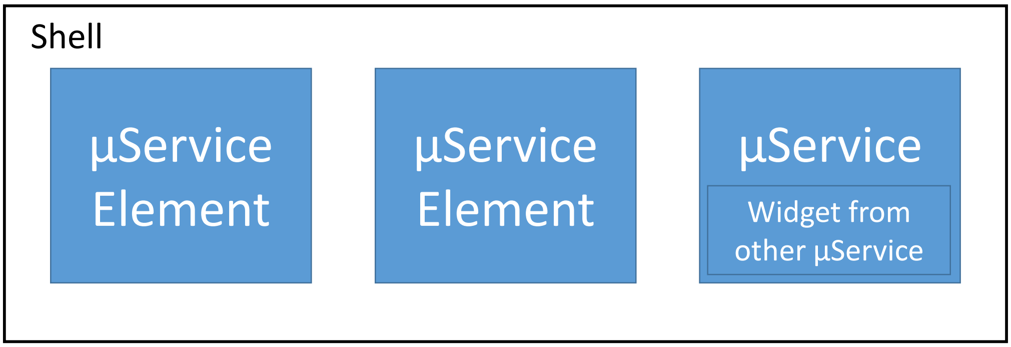
SOFTWAREarchitekt.at



UI Composition w/ Web Components

+ SPA
+ Nesting: Easy
+ Separate Deployment
+ Seperate Development

+ Communication: DOM
~ Isolation
~ Framework Adoption

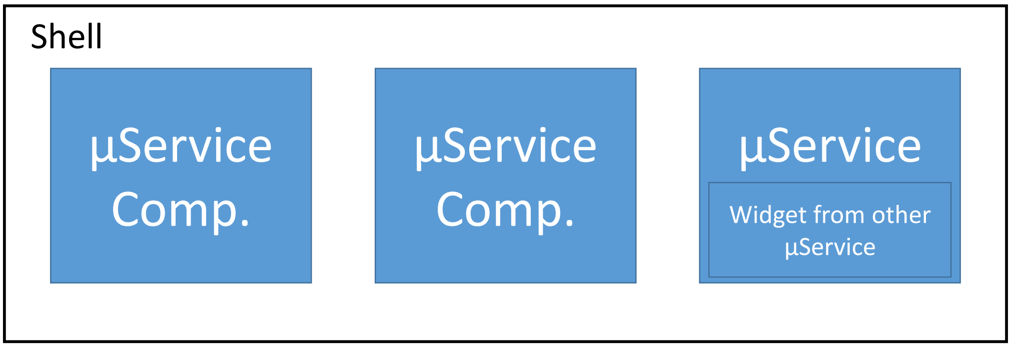


Experimental in Angular 6

UI Composition w/ Libraries

+ SPA
+ Can hold state
+ Nesting: Easy
+ Straight and well-known
+ Separate Development

- Bad Isolation
- Version Conflicts
- Mixing frameworks difficult
- No separate Deployment


Shell

µService Comp.

µService Comp.

µService

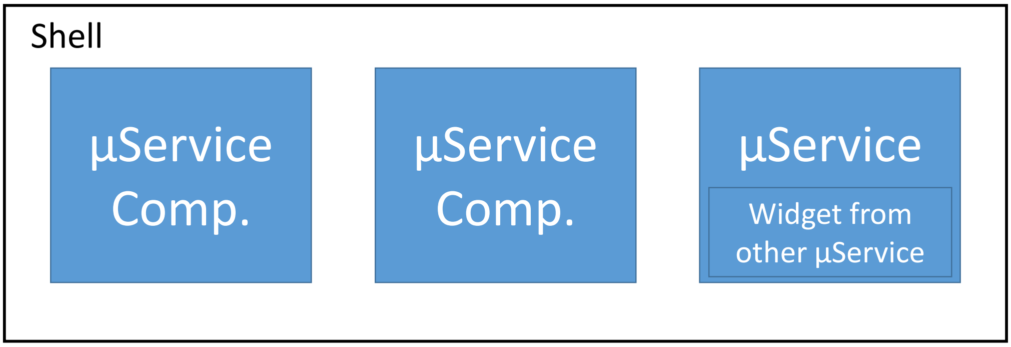
Widget from other µService

SOFTWARE architekt.at

UI Composition w/ Monorepo

+ SPA
+ Can hold state
+ Nesting: Easy
+ No Version Conflicts

- Bad Isolation
- Mixing frameworks difficult
- Separate Dev difficult
- No Separate Deployment



Shell

µService Comp.

µService Comp.

µService

Widget from other µService

Not an either/or-thing!

UI Composition

- Hyperlinks
- iFrames
- Web Components

Reusable Logic

- Libs
- Open Source

Structure for one µService

- Project Monorepo

Just one possible example!

DEMO

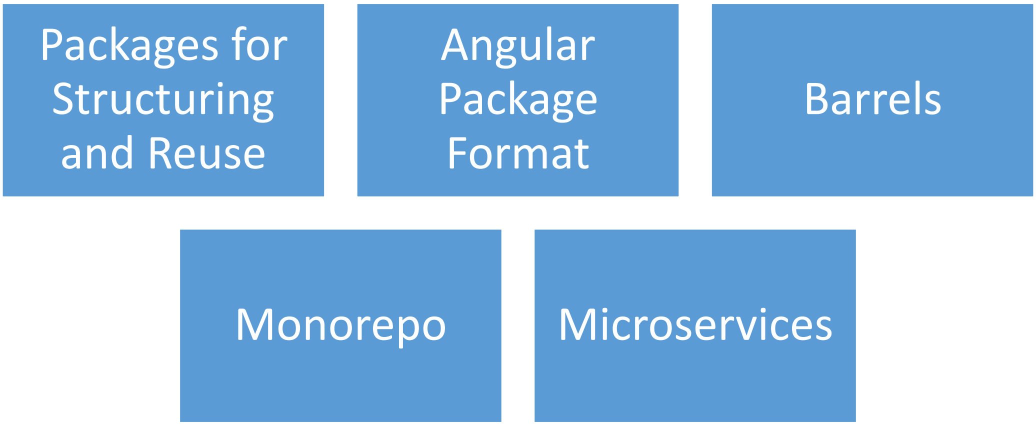
SOFTWAREarchitekt.at

Blog

- 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
- Microservice Clients With Web Components Using Angular Elements: Dreams Of The (Near) Future?

SOFTWAREarchitekt.at

Conclusion



Packages for Structuring and Reuse

Angular Package Format

Barrels

Monorepo

Microservices

SOFTWARE *architekt.at*