



International
JavaScript Conference



ManfredSteyer



Maintainable Architectures with Angular, Monorepos and Strategic Design

Manfred Steyer

SOFTWARE*architekt.at*



Experience



What do we need for a good architecture ?



BEST PRACTICES



PATTERNS



METHODOLOGY

Sharing Experience for Architectures

Domain-Driven

DESIGN

Tackling Complexity in the Heart of Software

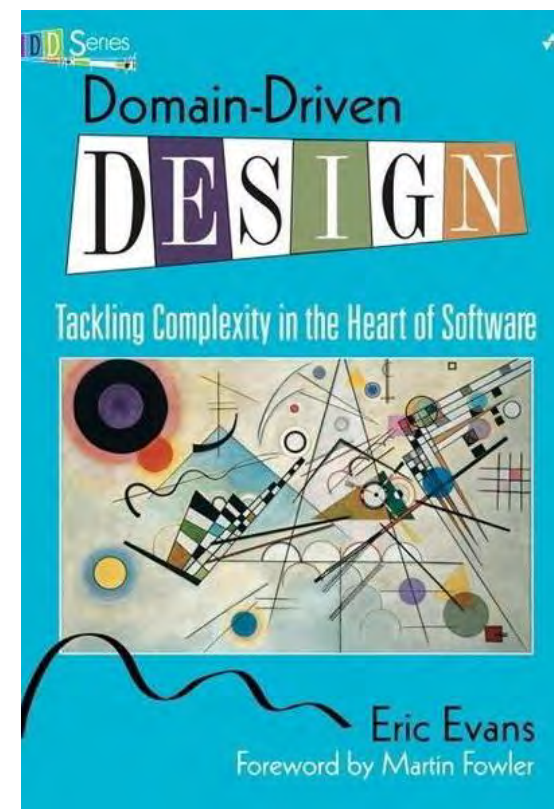


Eric Evans
Foreword by Martin Fowler

Domain Driven Design

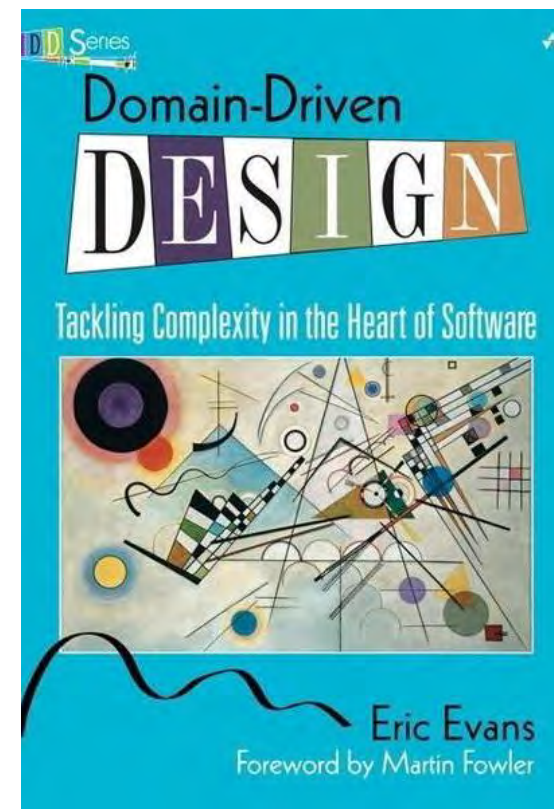


How can we improve our
frontend architectures
with ideas from DDD?





How can we improve our
frontend architectures
with ideas from DDD?



Contents

1)
Strategic
Design 101

2)
Angular
Monorepos

3)
Nx Monorepos
and DDD

4)
DEMO

About me...

- Manfred Steyer
SOFTWAREarchitekt.at
- Angular Trainings and Consultancy
- Google Developer Expert (GDE)
- Trusted Collaborator in the Angular Team



Public: Frankfurt, Vienna, Munich,
Düsseldorf, Stuttgart
In-House: everywhere

<http://softwarearchitekt.at/workshops>



Strategic Design

in a nutshell

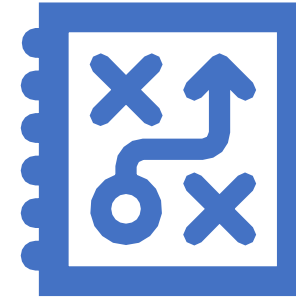
Domain Driven Design

Decomposing a System



Strategic Design

Design Patterns
& Practices



Tactical Design

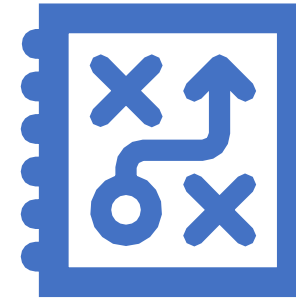
Domain Driven Design

Decomposing a System



Strategic Design

Design Patterns
& Practices



Tactical Design



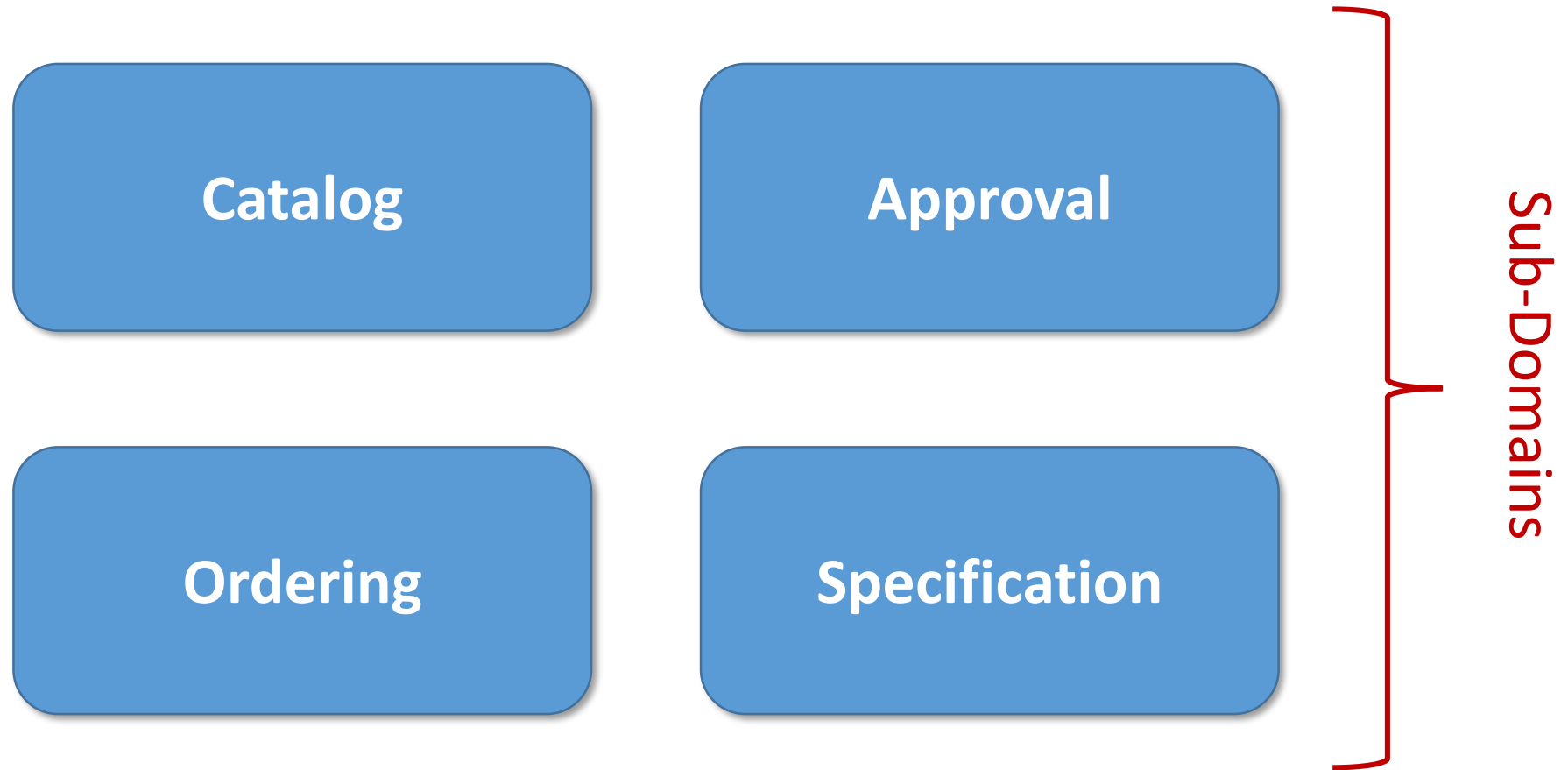


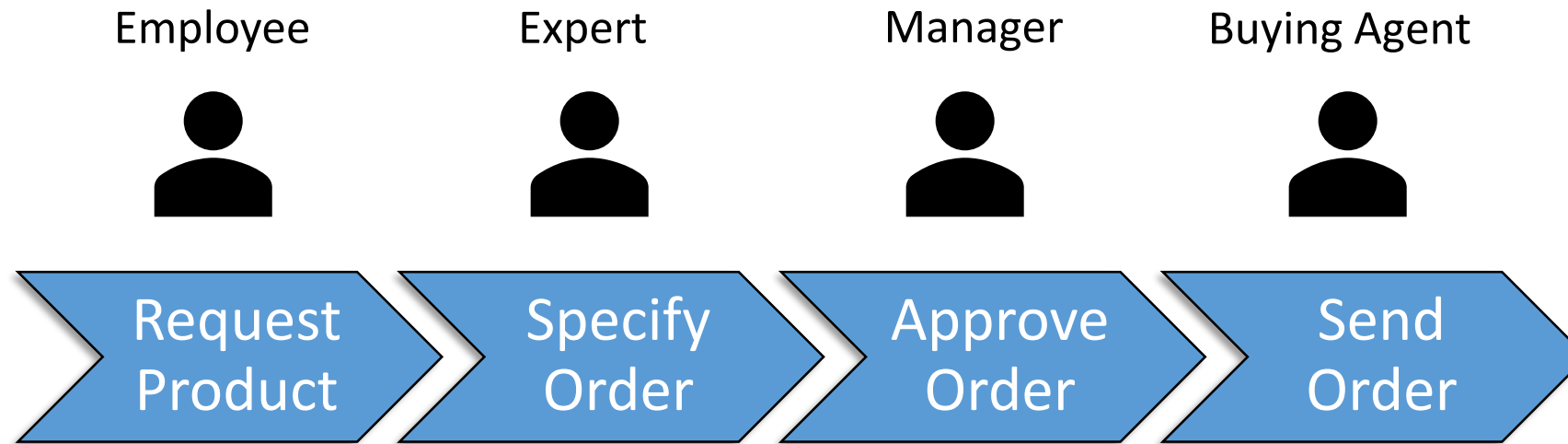
This is what Strategic DDD prevents

Example

e-Procurement System

Example

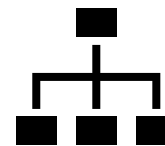




Product



Budget



Hierarchy

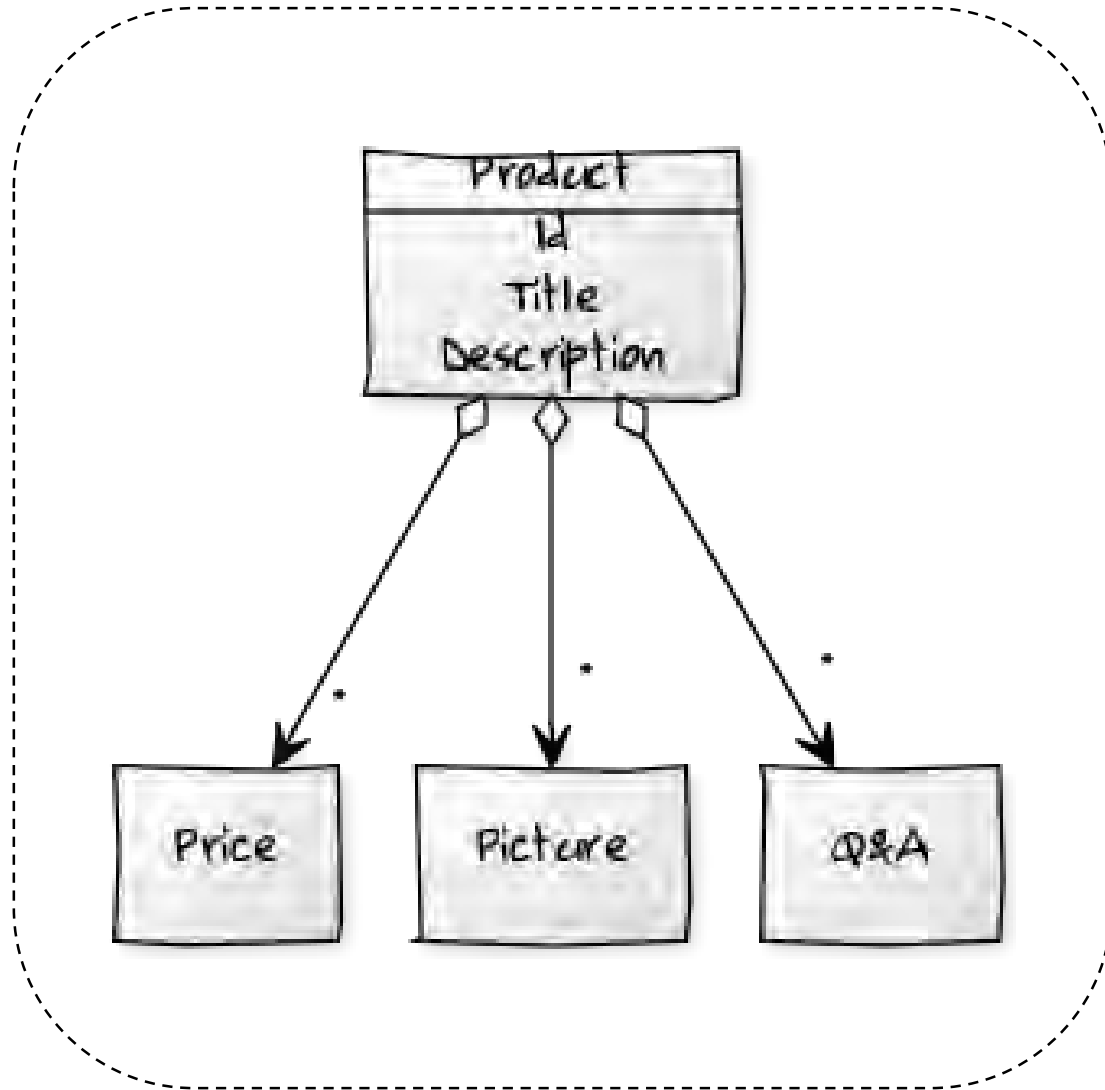


Manager

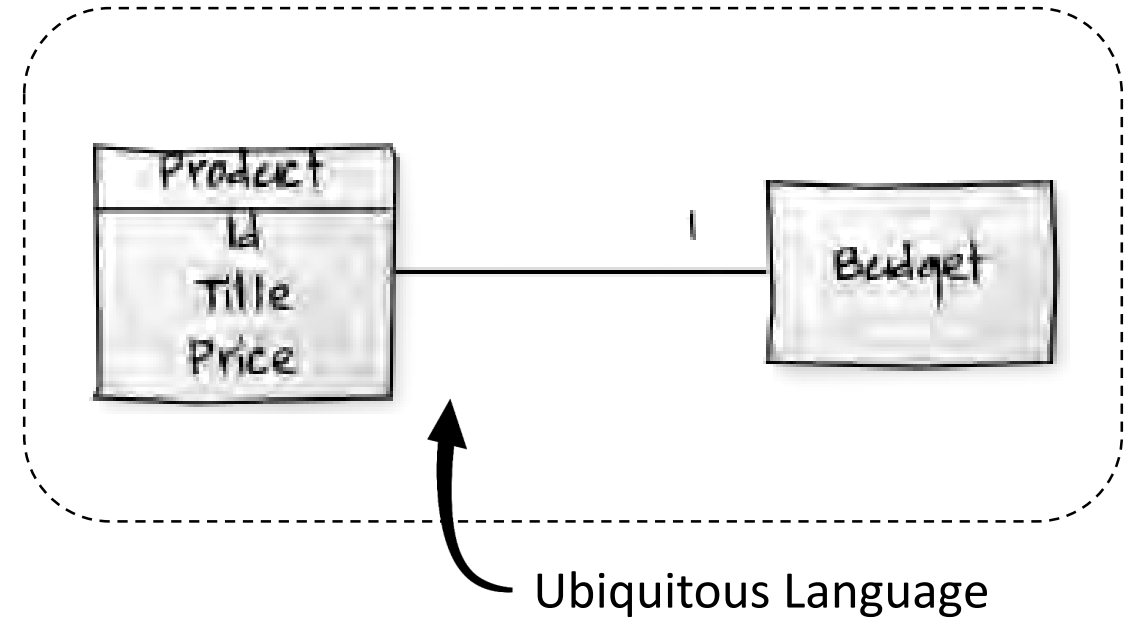


Manager

Catalog



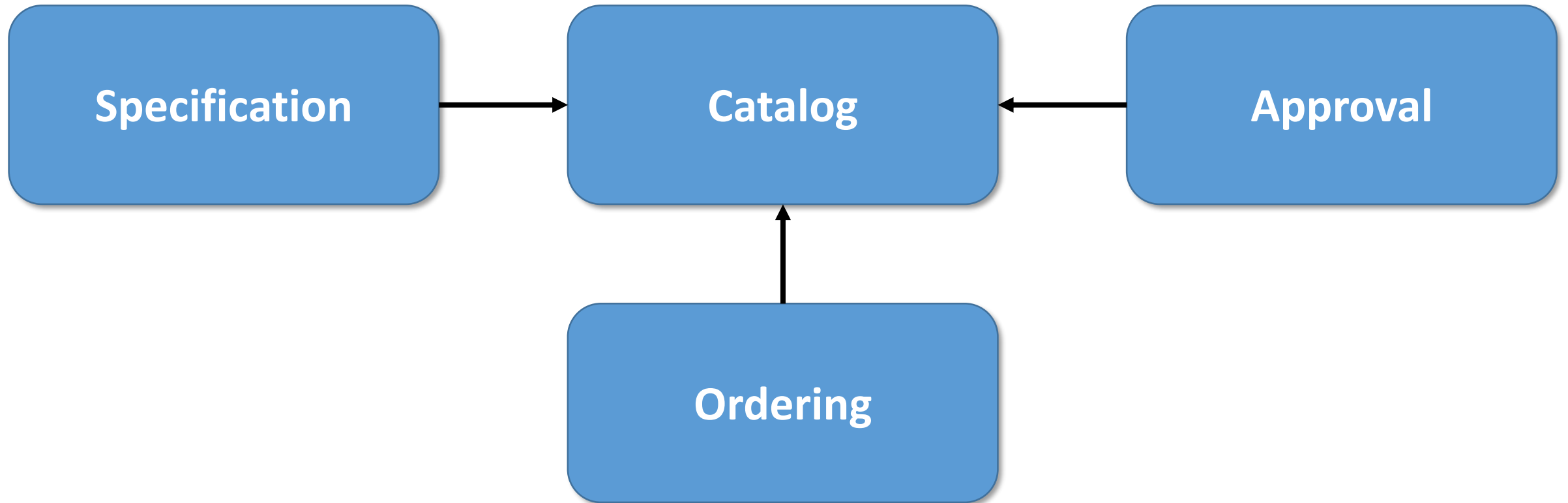
Approval



Bounded Context

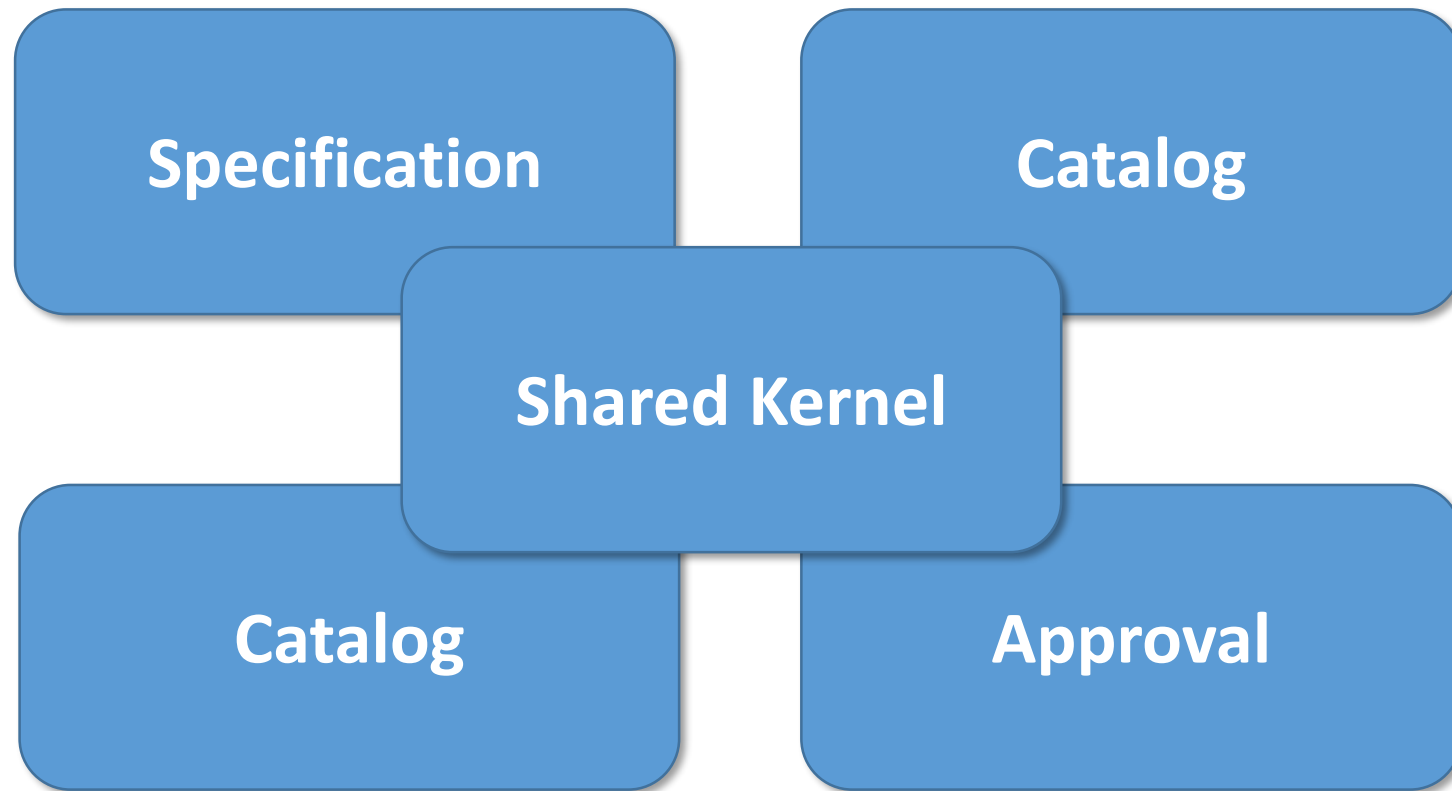
Ubiquitous Language

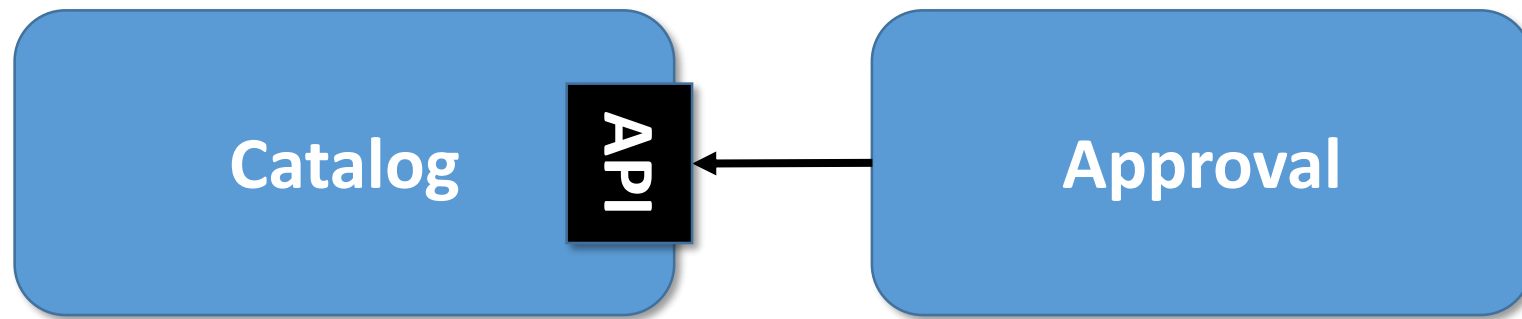
Context Map



Context Map

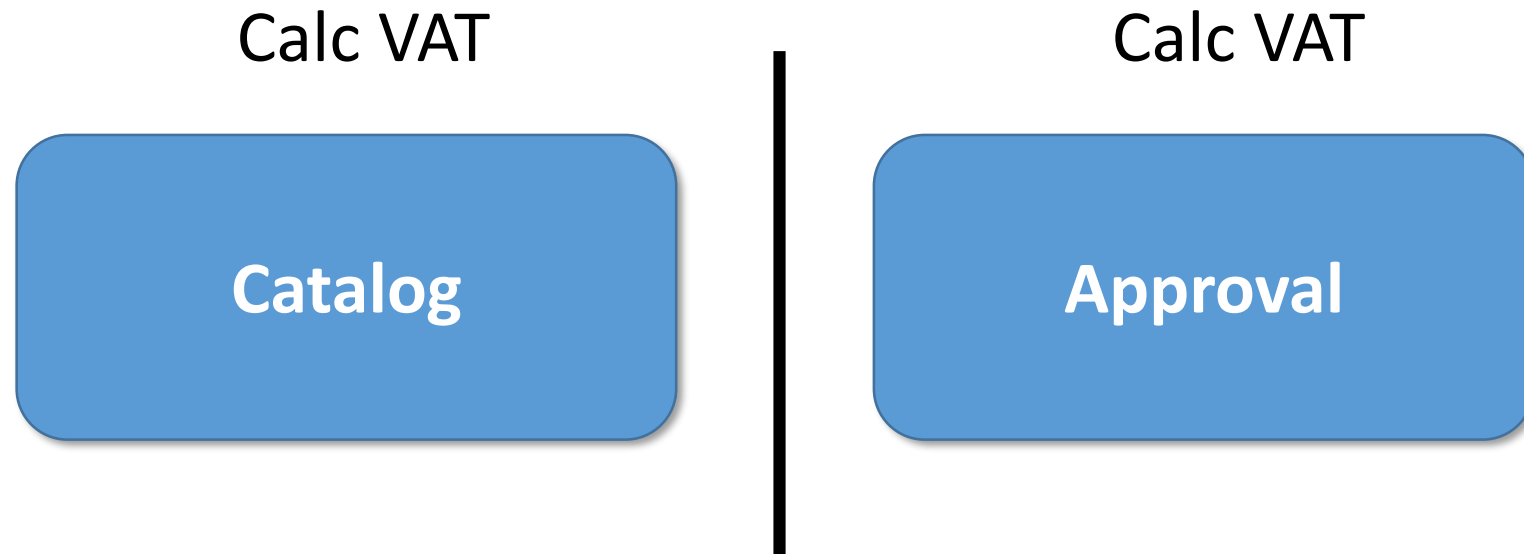
Responsibilities?
Breaking Changes?





Open-/Host-Service

Seperate Ways



Domain-Driven

DESIGN

Tackling Complexity in the Heart of Software



Eric Evans
Foreword by Martin Fowler

Lots of additional approaches for cross-domain communication ...



2) Monorepos

Workspace

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

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
```


Advantages

Everyone uses the latest versions

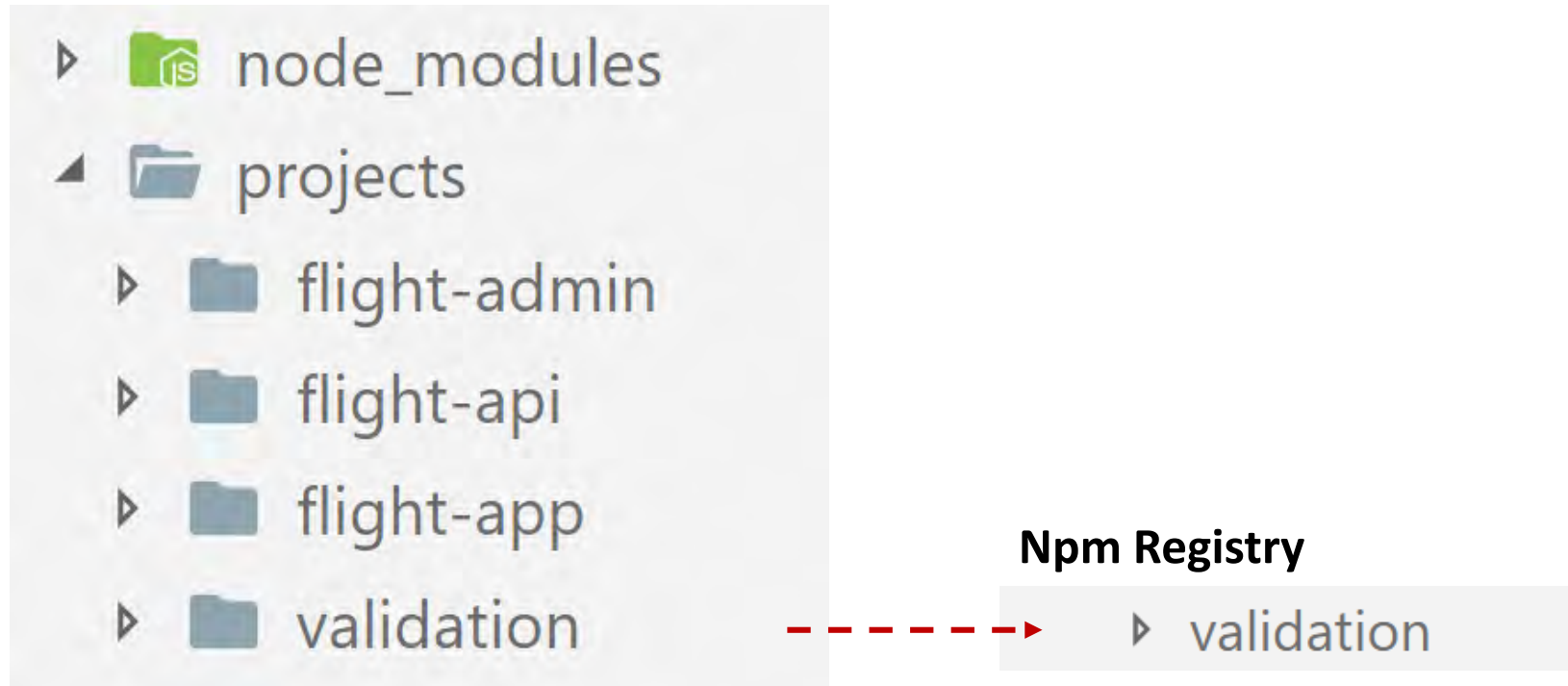
No version conflicts

No burden with distributing libs

Creating new libs: Adding folder

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

Moving back and forth



Tooling & Generator

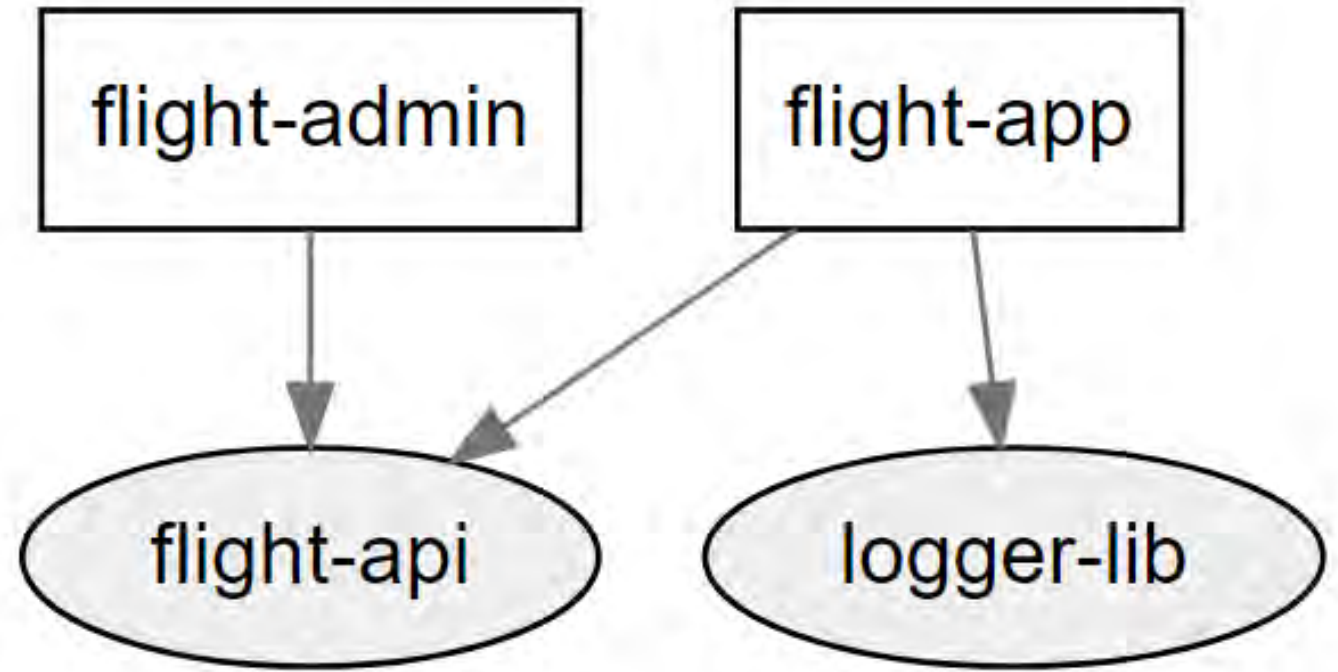
<https://nrwl.io/nx>



Nrwl Extensions for Angular

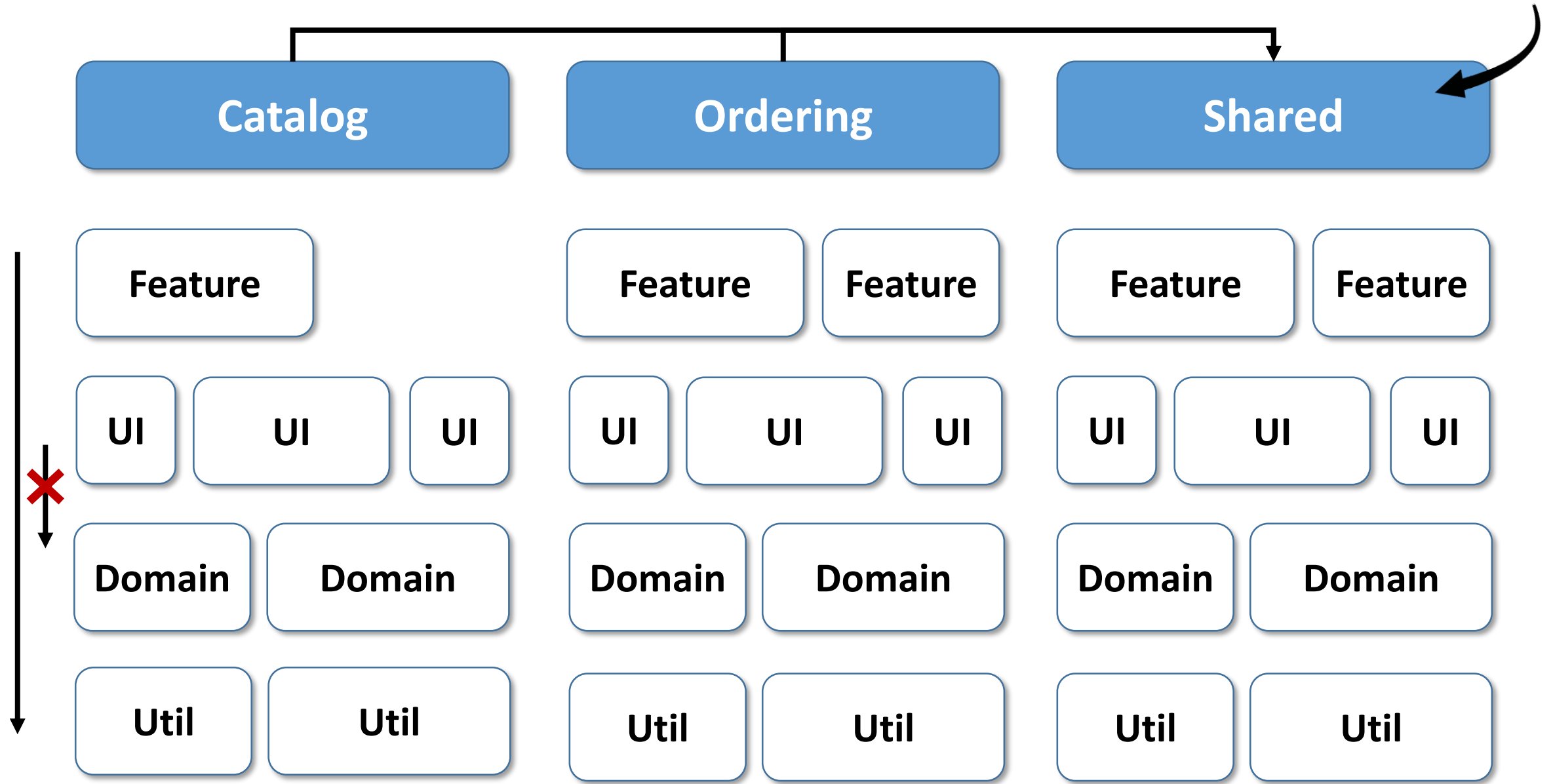
An open source toolkit for enterprise Angular applications.

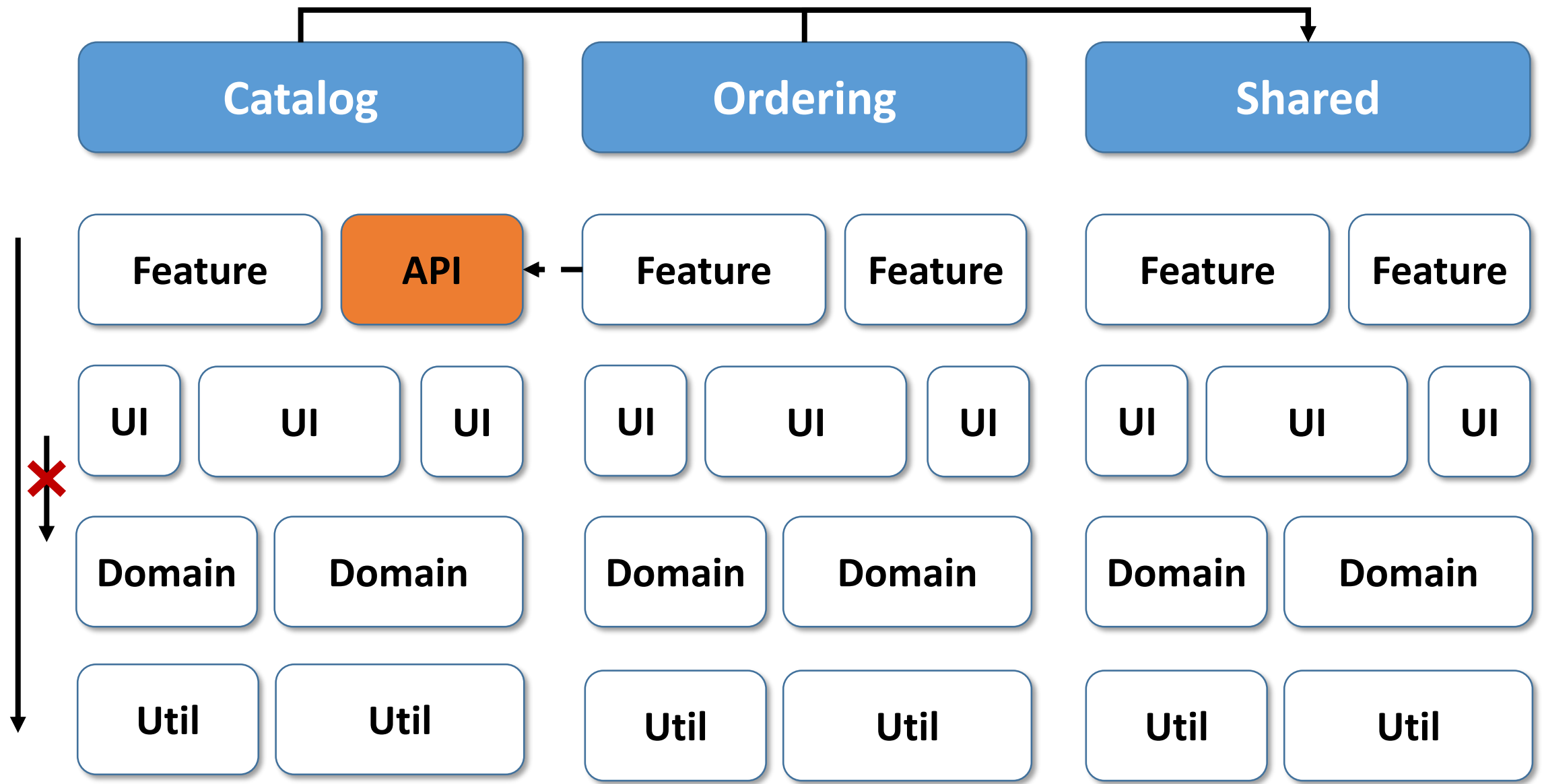
Visualize Module Structure



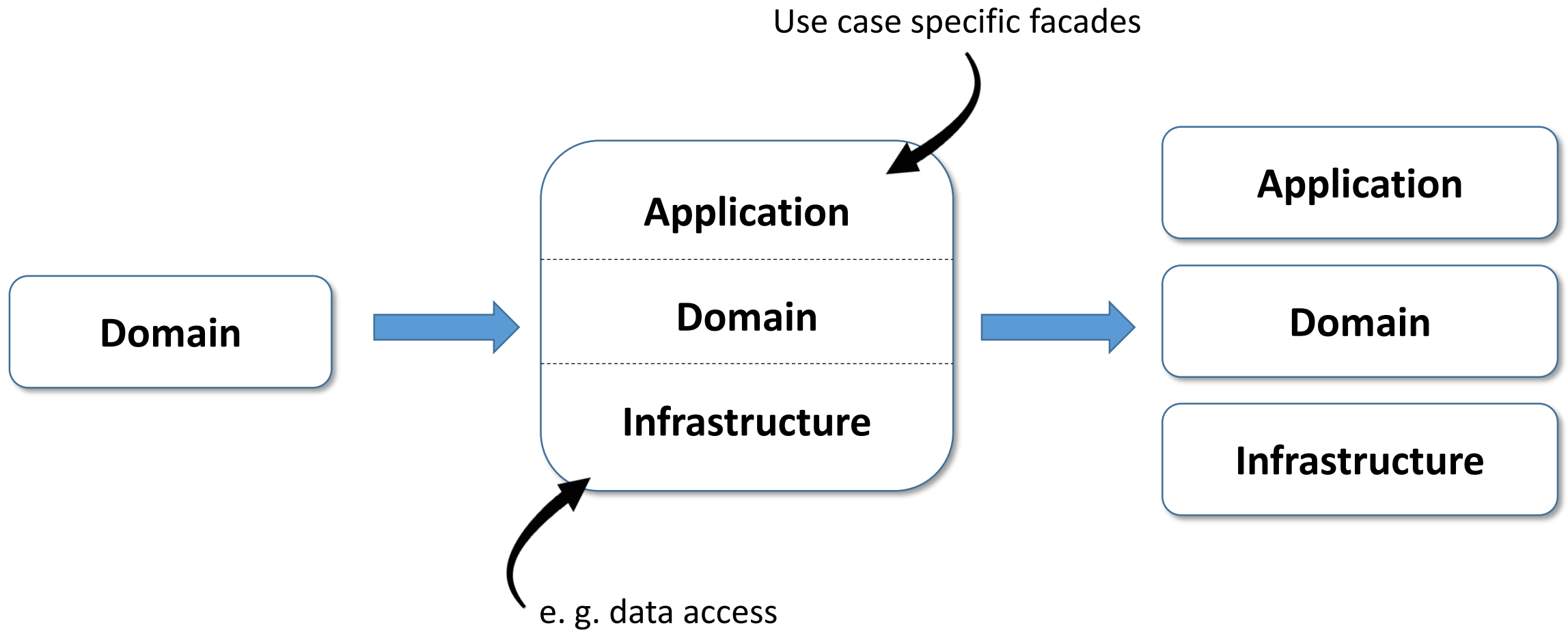


Nx Monorepos and Strategic Design





Isolate your domain!





Choice

Choice

Alternatives to
layering

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

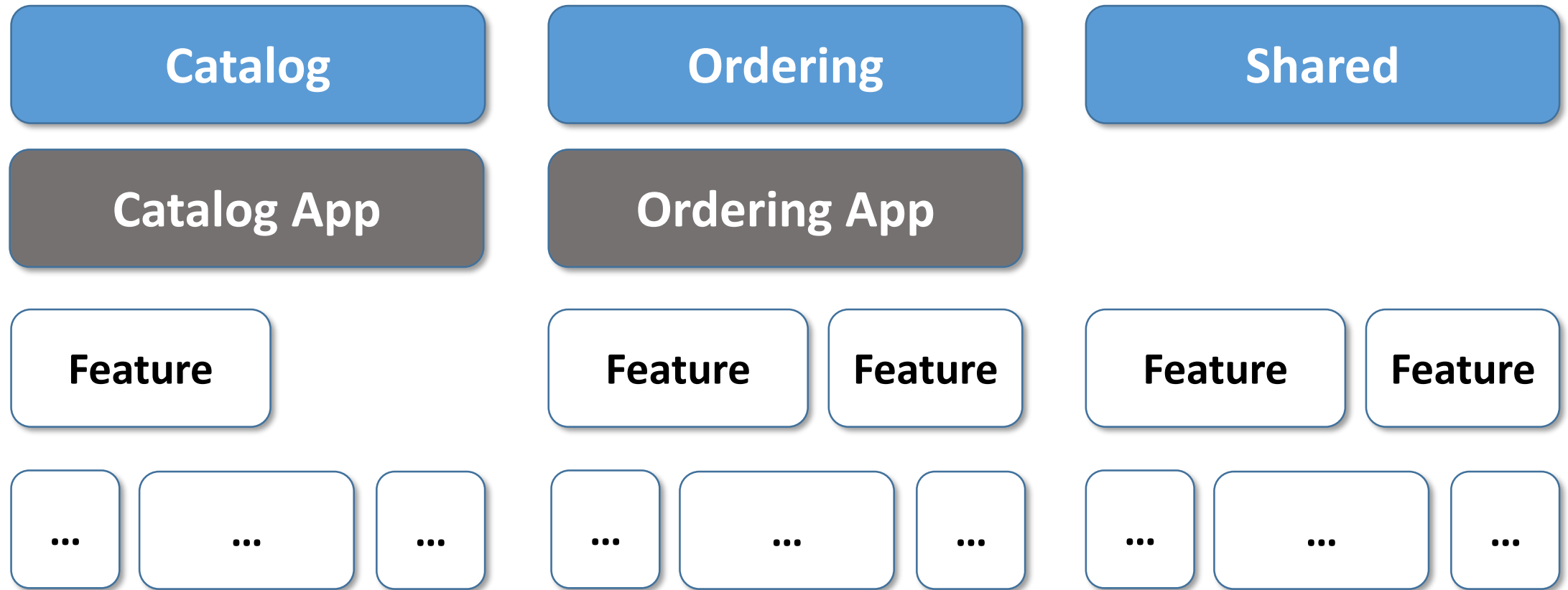
DEMO



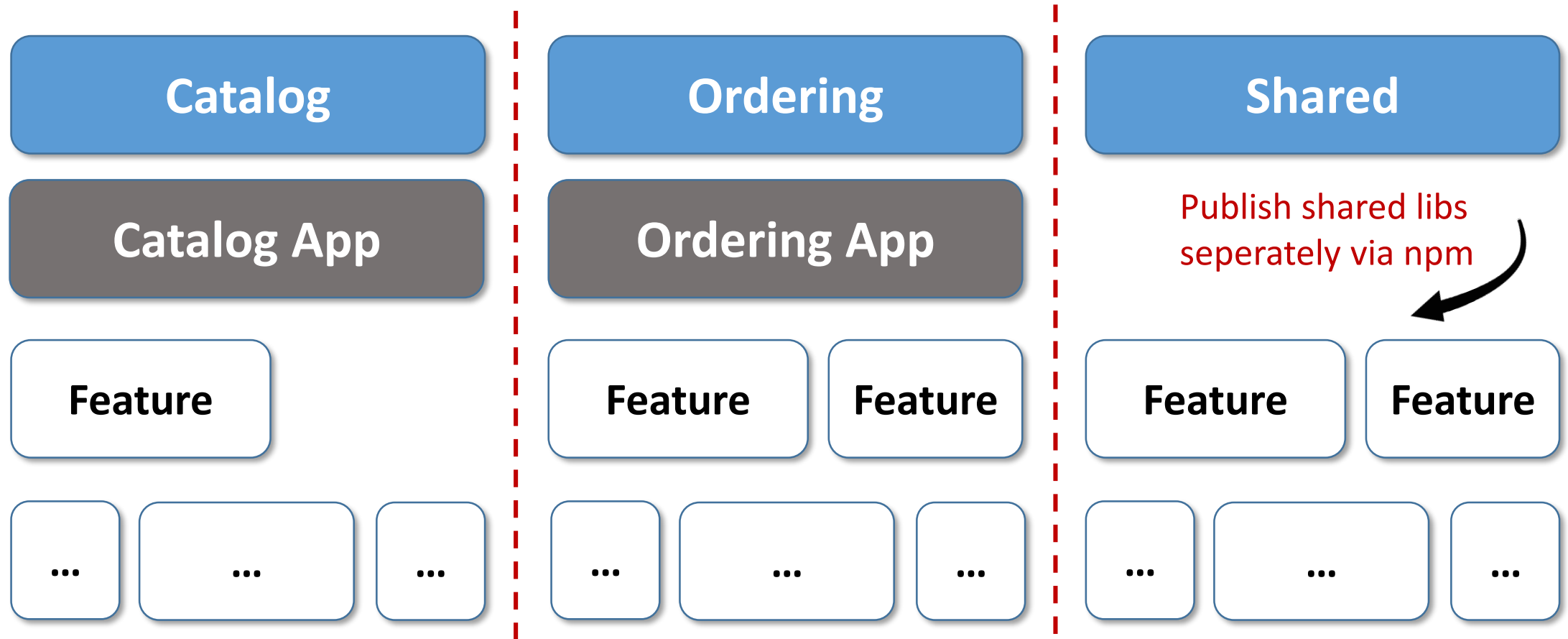
Micro Apps?

Short outlook

Option 1: One App per Domain



Option 2: One Monorepo per Domain



Summary

Slicing into sub-domains

Slicing into layers

Non-strict layering

Finegrained libraries

Enforce restrictions



Contact and Downloads

[mail] manfred.steyer@SOFTWAREarchitekt.at

[web] SOFTWAREarchitekt.at

[twitter] ManfredSteyer

Slides & Examples



Public: Frankfurt, Vienna, Munich, Düsseldorf, Stuttgart
In-House: everywhere

<http://softwarearchitekt.at/workshops>