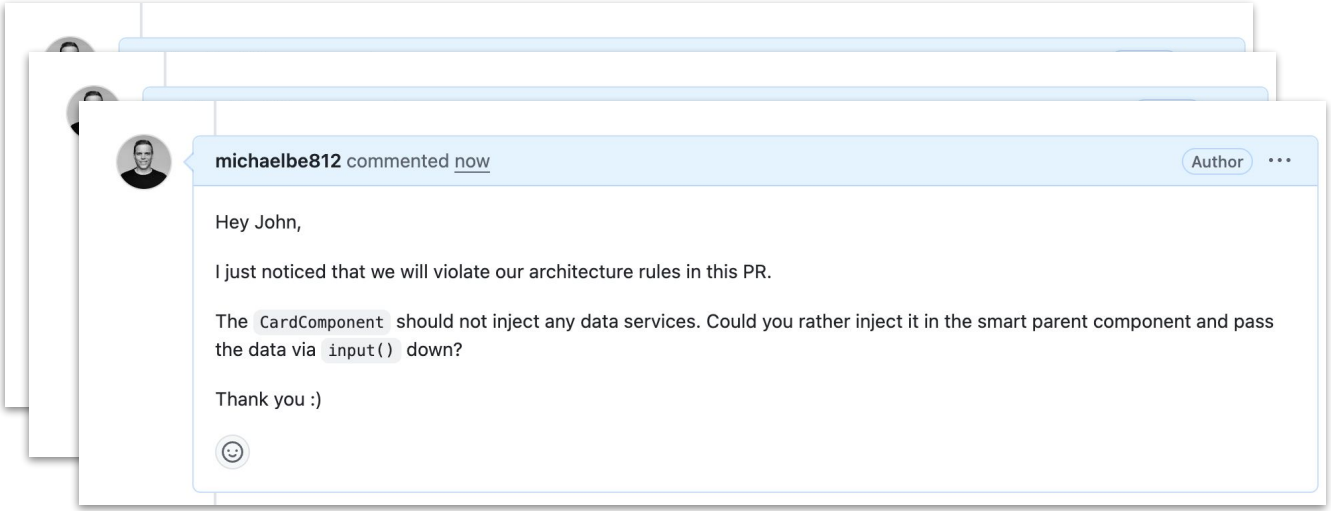
Architecture enforcement done right

Familiar PR comment?



Agenda

Agenda

- The problem
- Intro about architecture
- Architecture Validation
- Case Study / Demo
- Conclusion

Michael Berger

Freelancer | Trainer | Consultant

michael.berger@berger-engineering.io

@michaelbe812

www.berger-engineering.io

michaelbe812

https://www.linkedin.com/in/michael-berger-angular-expert/

Berger Software Engineering

Architecture

Architecture |What is it?

With architecture we define a ***ruleset*** how we want to ***structure things*** and how these things should ***work together***.

Code organization /

Modularization

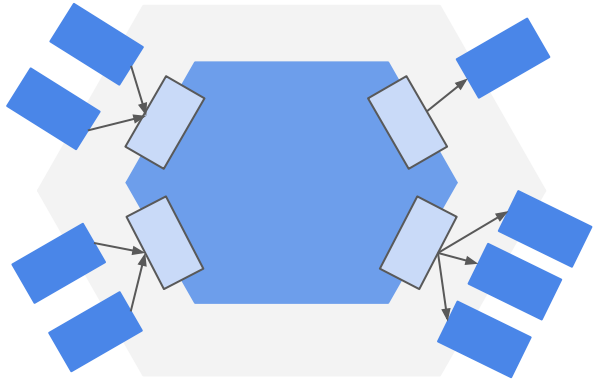
Dependencies

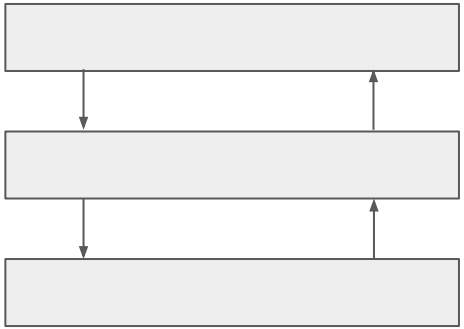
Encapsulation

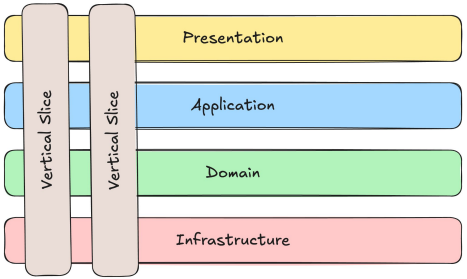
API's

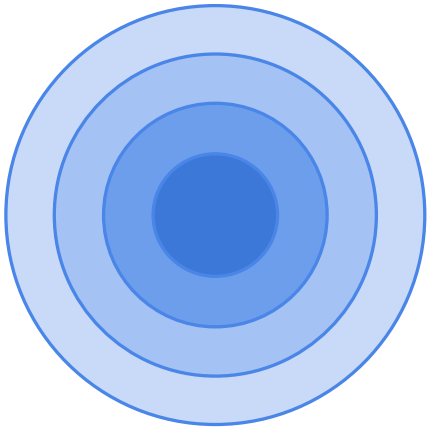
Architecture | Patterns

- Layered Architecture
- Vertical Slice Architecture / Verticals
- Clean Architecture
- Hexagonal Architecture







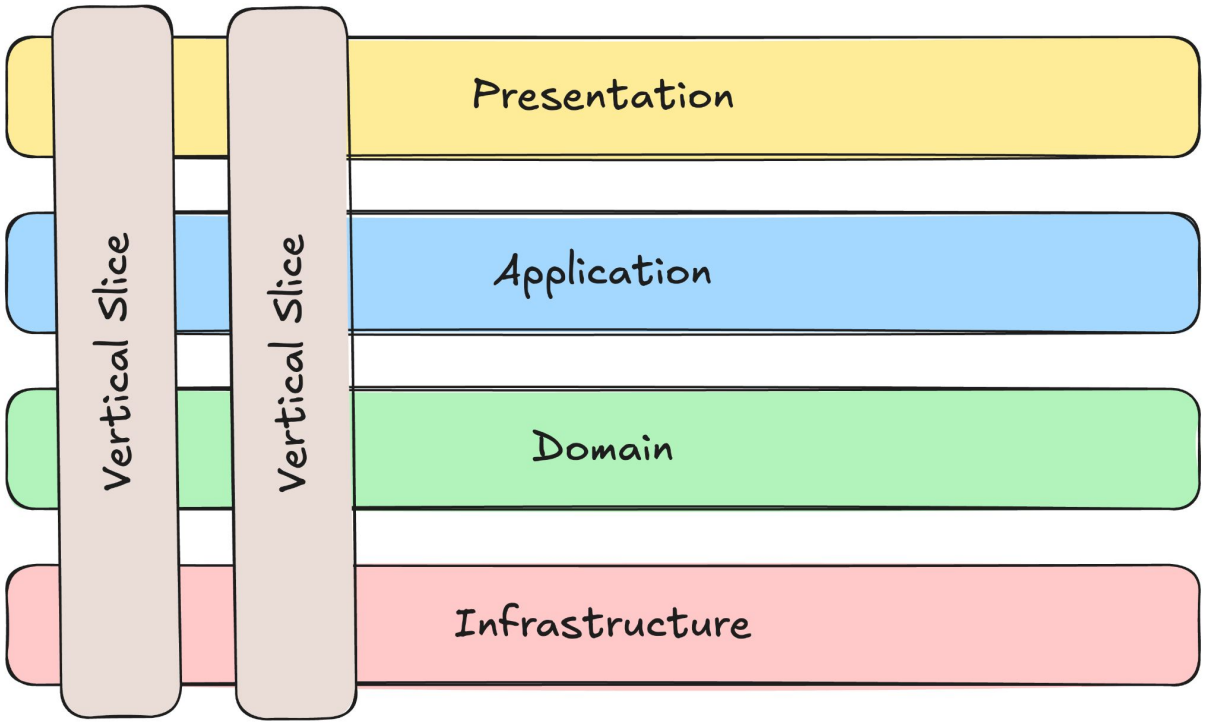


Verticals | What

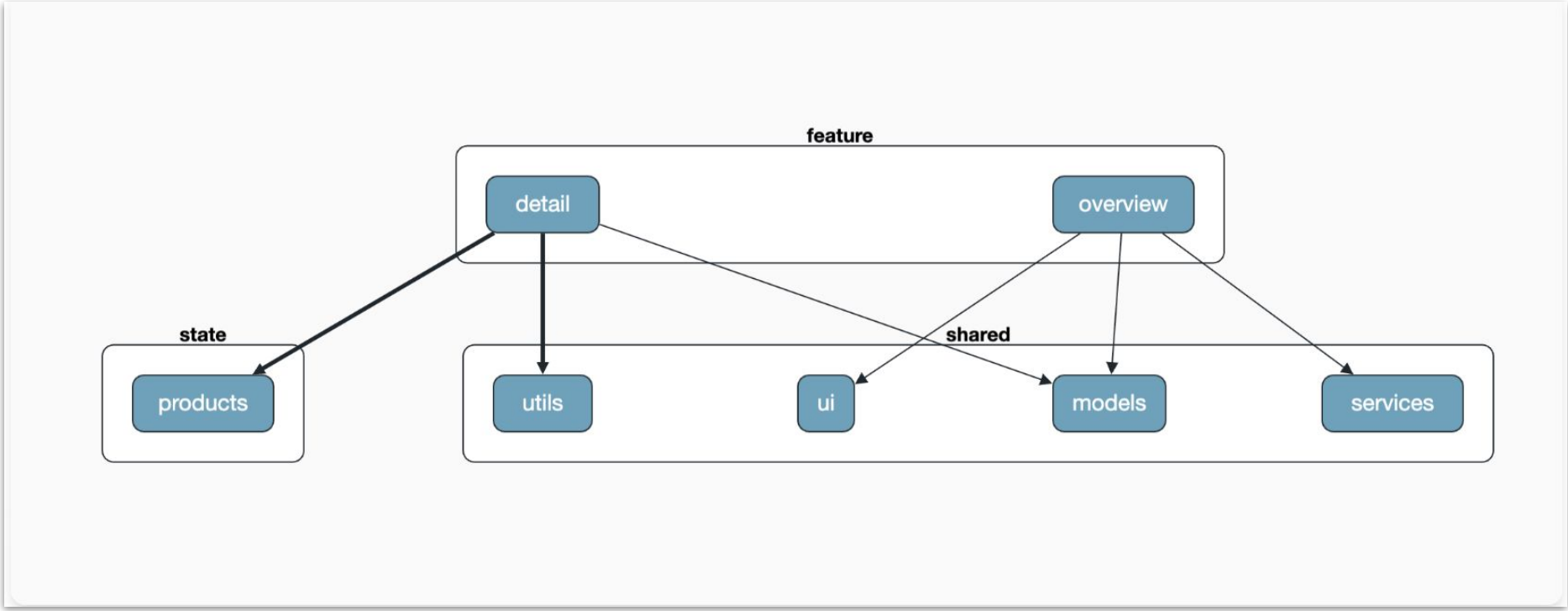
Verticals, more precisely ***Vertical Slice Architecture (VSA)***, is a pattern that aims to organize the application by vertical slices.

Each Slice encapsulates a specific functionality or feature. This means that each vertical layer contains all necessary components to provide a specific functionality.

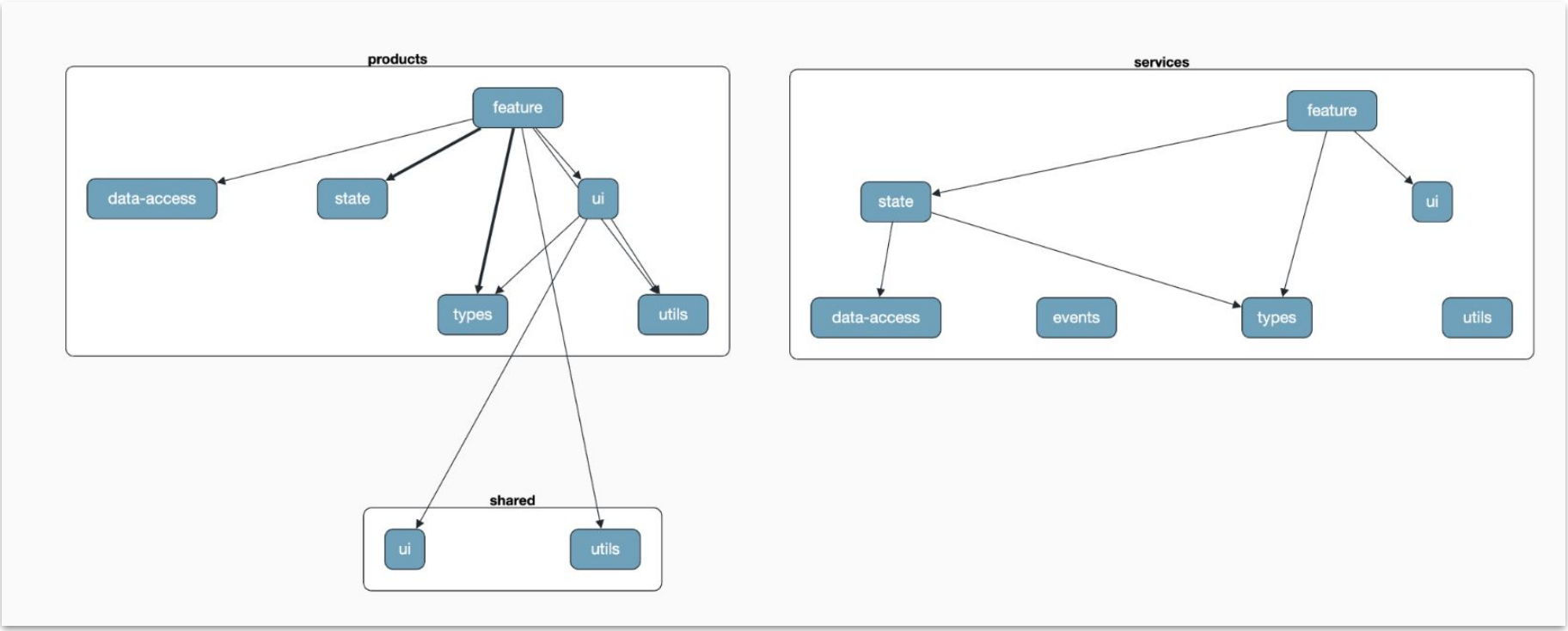
Verticals | Characteristics



Verticals | Impact visualized



Verticals | Impact visualized



Read more: https://www.berger-engineering.io/articles/visual-impact-of-verticals-en

Verticals | Pros and Cons

Pros

- Natural module encapsulation
- Improved Code co-location
- Better DX
- Rigid Structure
- Good natural decoupling
- Scales with complexity

Cons

- Not always DRY

Architecture Validation

Prerequisites for Architecture Validation

- Rules must be deterministic
- Therefore architecture must be predictable by patterns → e.g. always the same folder structure

Overview about the tooling landscape

- Tools for Architecture Validation which work together with ESLint:
  - Sheriff
  - nx module boundaries
  - eslint-plugin-boundaries

Tools | nx/module-boundaries

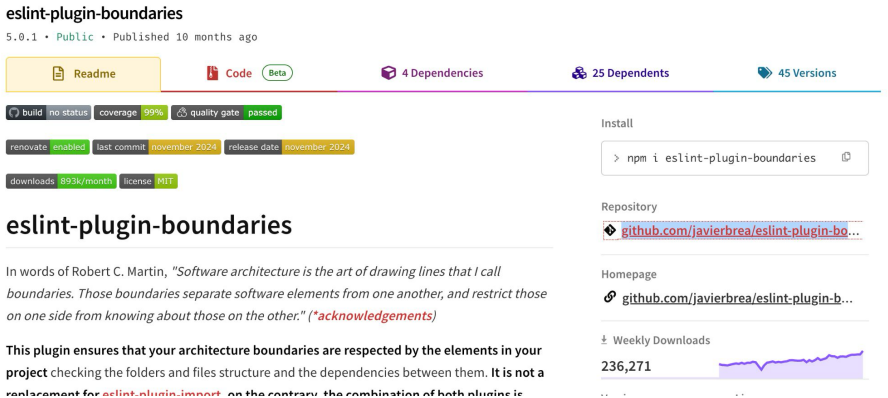
Core Principle:

- Tagging nx projects and implementing access rules based on tags
- multi-dimensional, e.g. 2 dimensions:
  - type: feature, ui, data-access, utility
  - scope: e.g. admin, shopping-cart → e.g. a sub-domain/bounded context

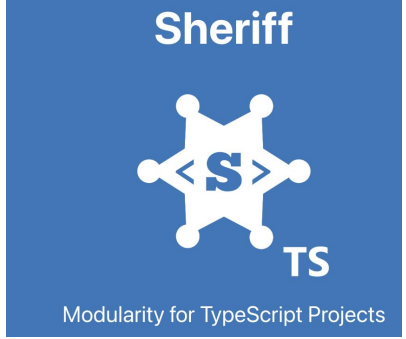
Tools | eslint-plugin-boundaries

Core principles

- Tagging of modules based on path structure
- Defining dependency rules and access rules between modules



Tools | sheriff



Core Principles

- Enforces module boundaries for encapsulation. Works with and without barrel files (index.ts)
- Very flexible dependency rule configuration to model access-rules based on tagging
- Zero external dependencies
- Can be integrated with ESLint and/or used via a standalone CLI

Credits to
Rainer Hahnekamp

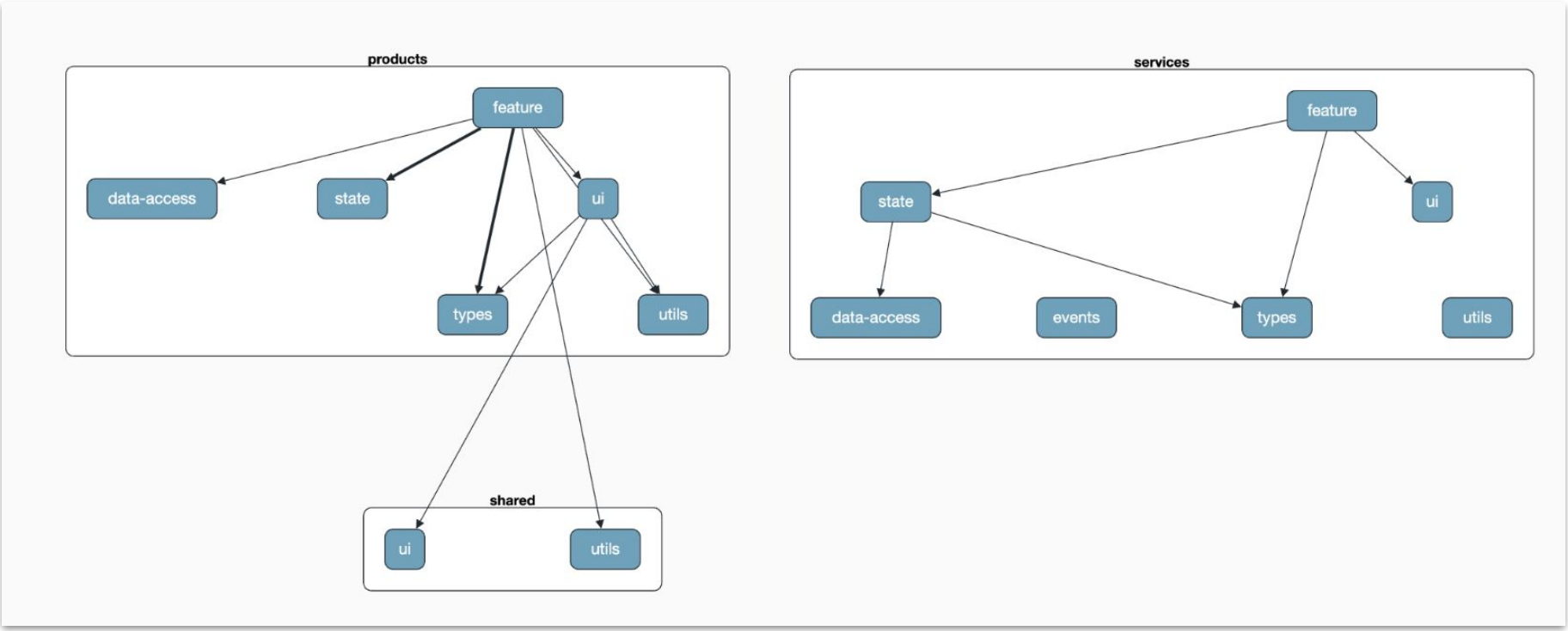
Tool Comparison

	Nx	Sheriff	eslint-plugin boundaries
Flexibility	-	↑	↑
Setup	↑	-	-
Boilerplate	↓	↑	-

↑ Good - OK ↓ Not good

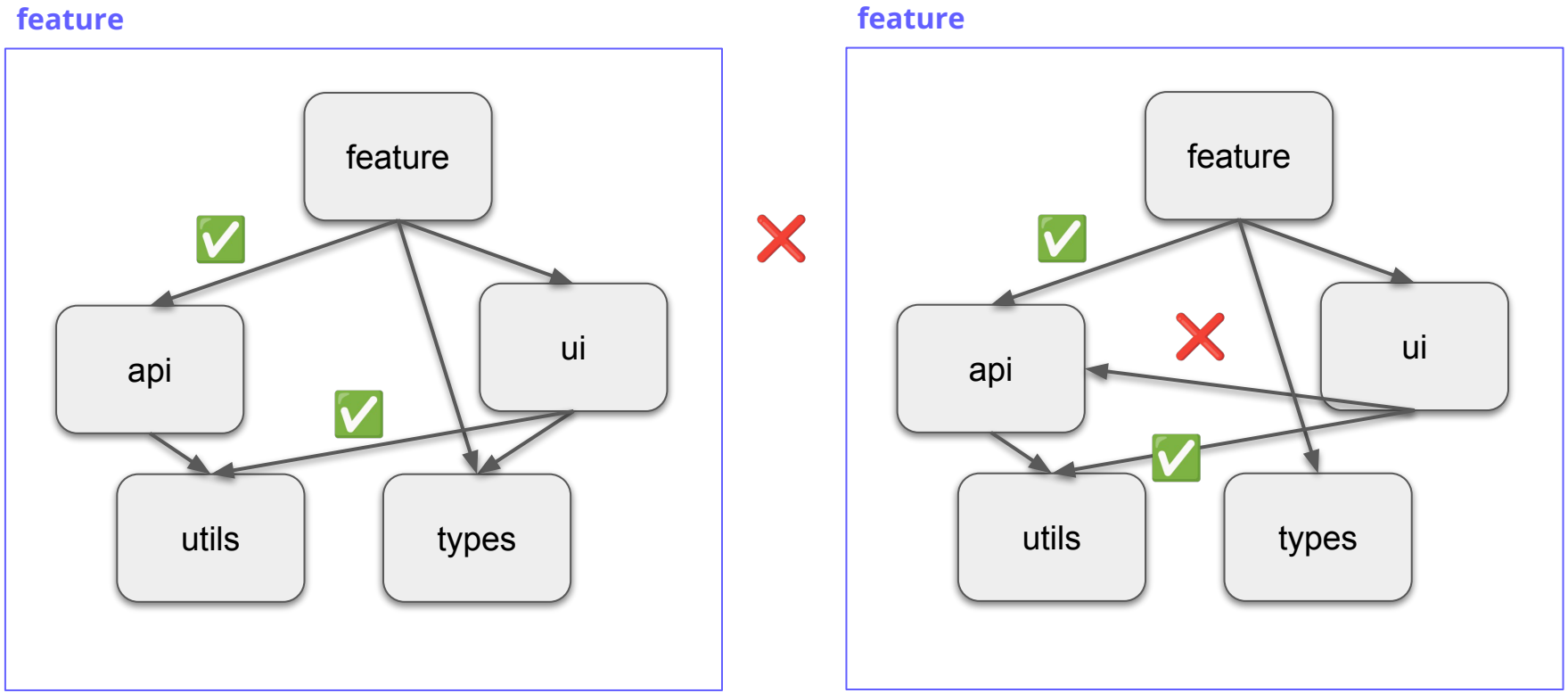
Case Study / Demo

Recap

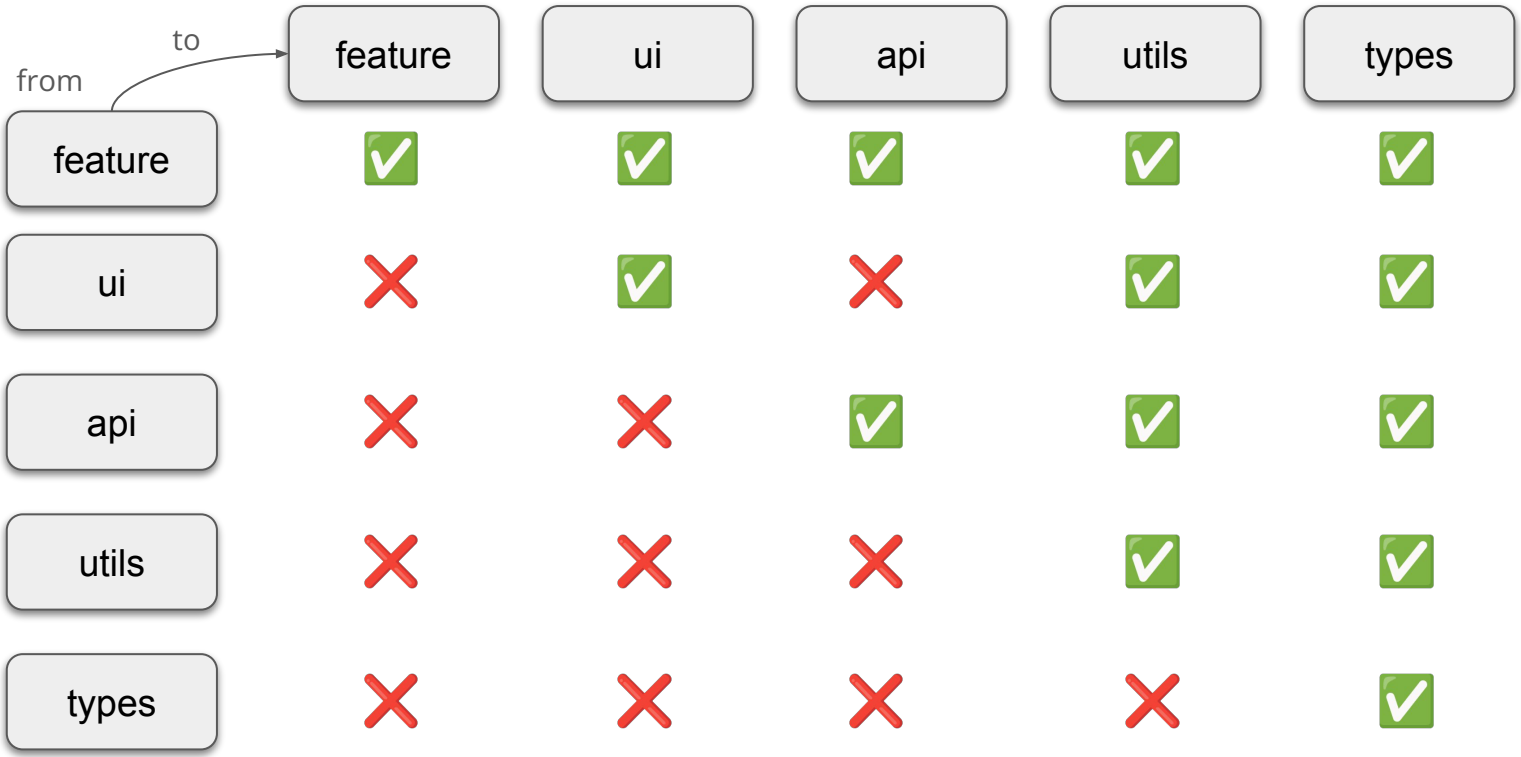


Read more: https://www.berger-engineering.io/articles/visual-impact-of-verticals-en

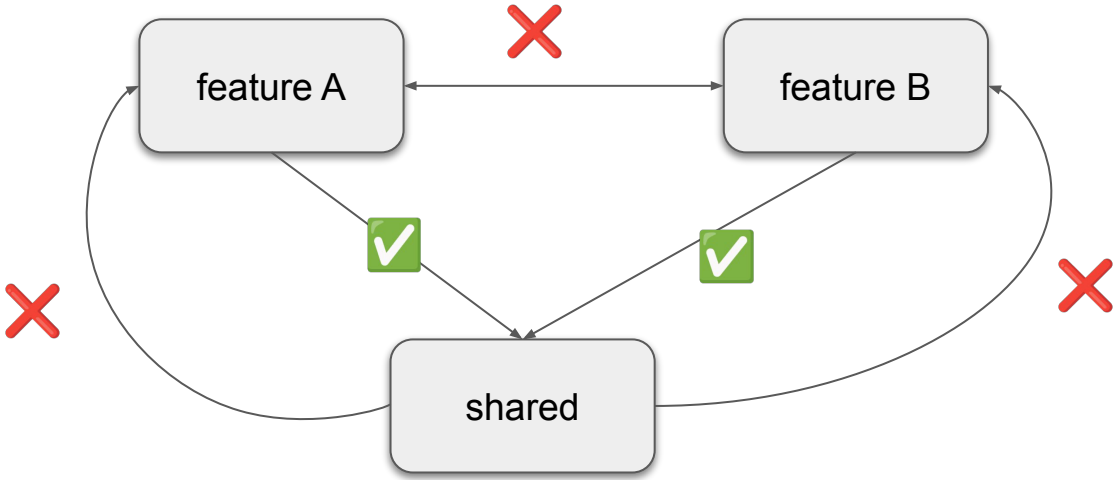
Isolated features



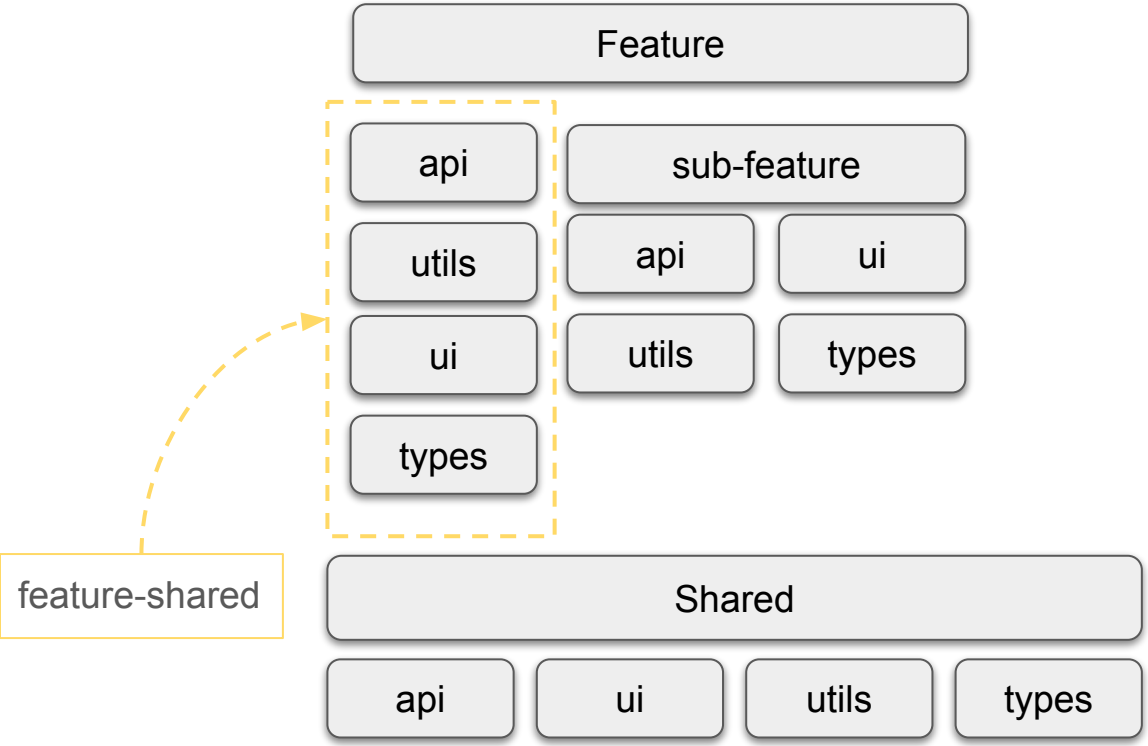
Access Rules within a feature



Access rules between features



Complex Modularization



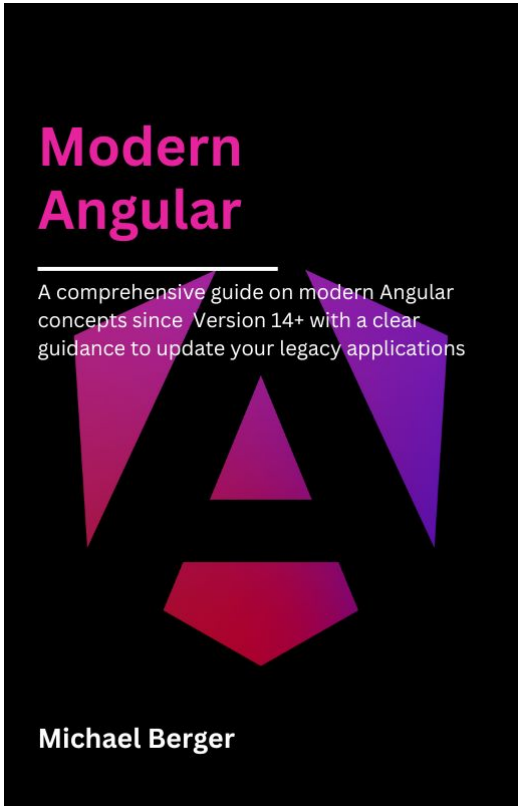
Demo-Time

Conclusion

Takeaways

- ✅ Enforcing architecture patterns is relatively easy with the right tool
- ✅ Verticals are a great fit for evolving architectures
- ✅ Architecture validation is a key measure to preserve a healthy project

THANK YOU



THE guide for modern Angular

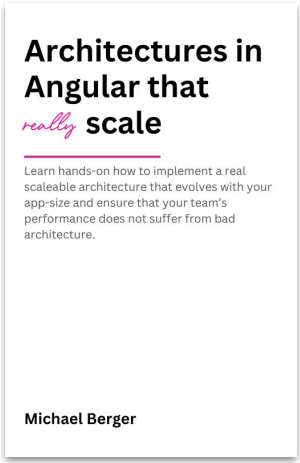
This book will cover:

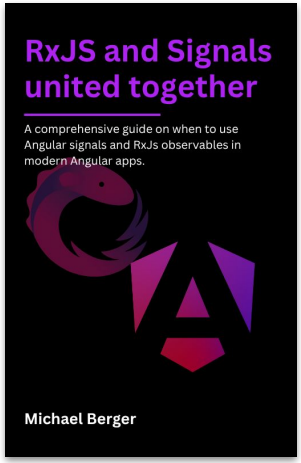
- All modern Angular concepts
- A clear decision guidance for each concept when and how to update “legacy” apps
- + free Agents.md for a modern Angular AI Agent

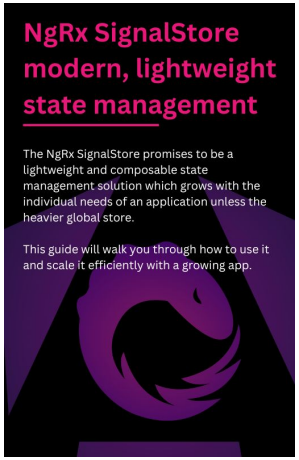
Exclusive Pre-Order discount until 15.11.2025

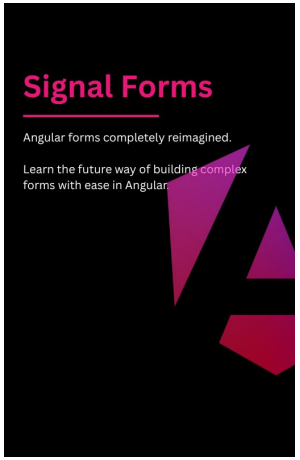
https://book.modern-angular.com/

Guides for modern Angular Developers









https://guides.modern-angular.com/

scan to visit

Michael Berger

Freelancer | Trainer | Consultant

michael.berger@berger-engineering.io

call

https://www.berger-engineering.io/appointments

@michaelbe812

michaelbe812

Slides & Demo Repo

Repo

https://github.com/michaelbe812/talk-automated-architecture-validation

Slides

https://www.berger-engineering.io/talks#ng-stuttgart-2025-10-22