Hexagonale Architektur

in der Praxis

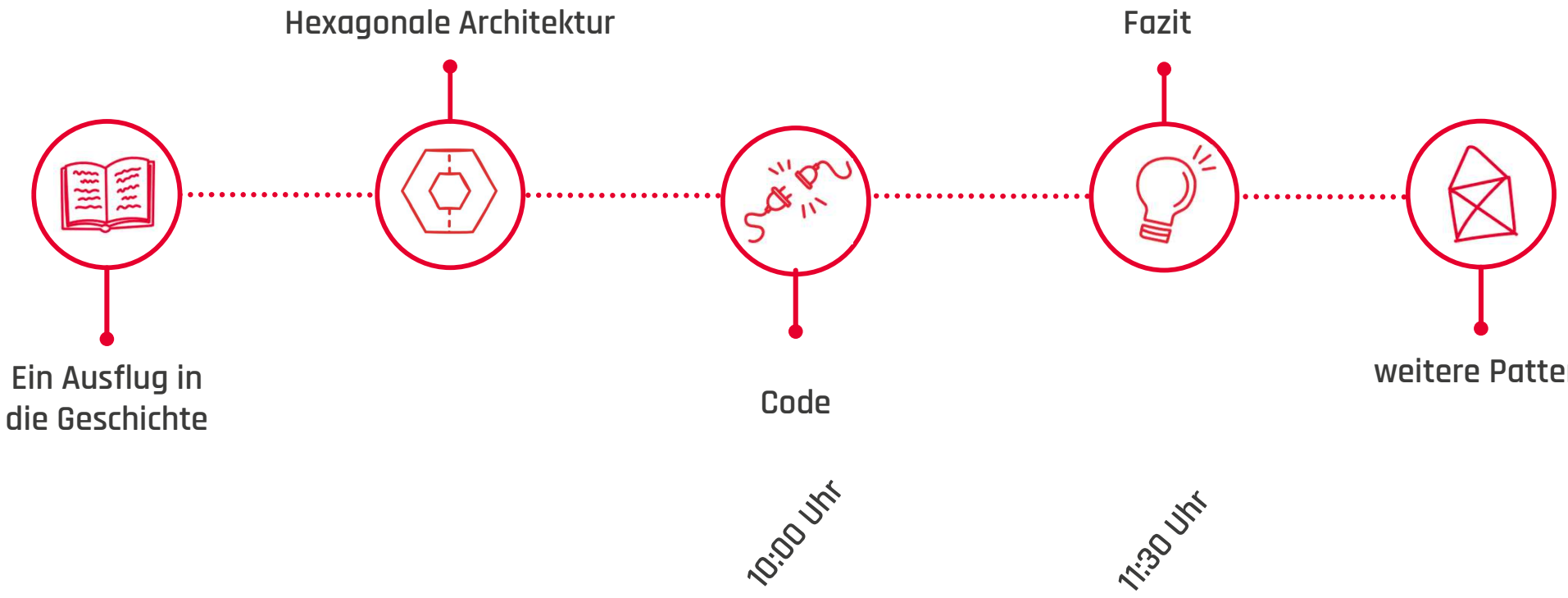
ARCHITECTING
YOUR
BUSINESS


IT ARCHITECTURE


CLOUD-NATIVE SOFTWARE


APIS & DATA PRODUCTS

Agenda


Hexagonale Architektur
Fazit
Ein Ausflug in die Geschichte
Code
weitere Pattern
10:00 Uhr
11:30 Uhr

Alexander Wessel

Software Engineer & IT Architect

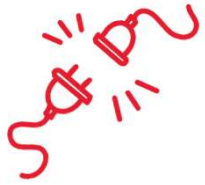
alexander.wessel@pentacor.de

linkedin.com/in/a-wessel

Wer sind wir

IT Architecture

Cloud-Native
Software

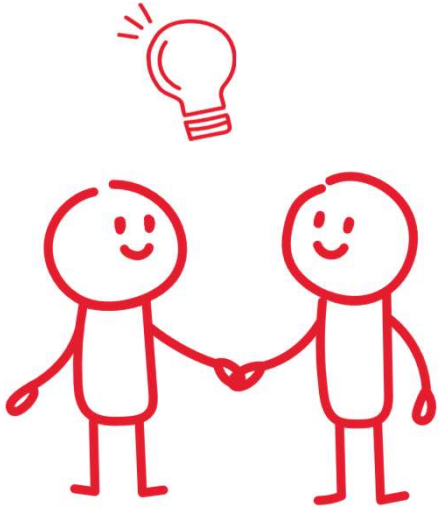


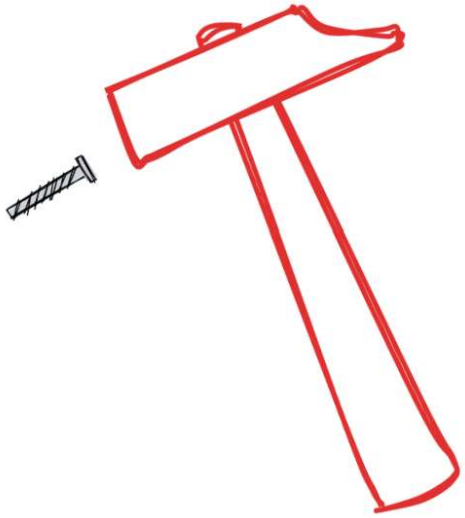
APIs and Data
Products

Was tun wir

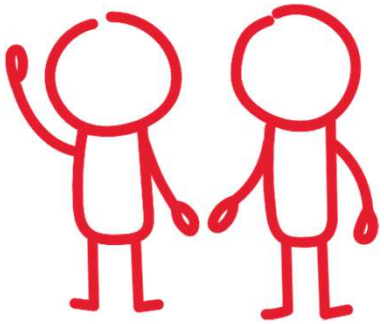
menti.com/alrrmcfh4c4g

Wer seid ihr?

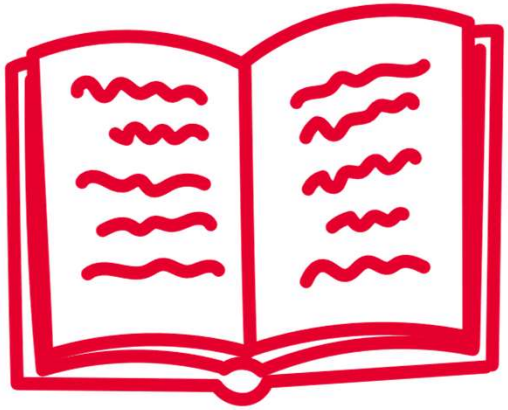




Disclaimer



Ein Ausflug in die Geschichte

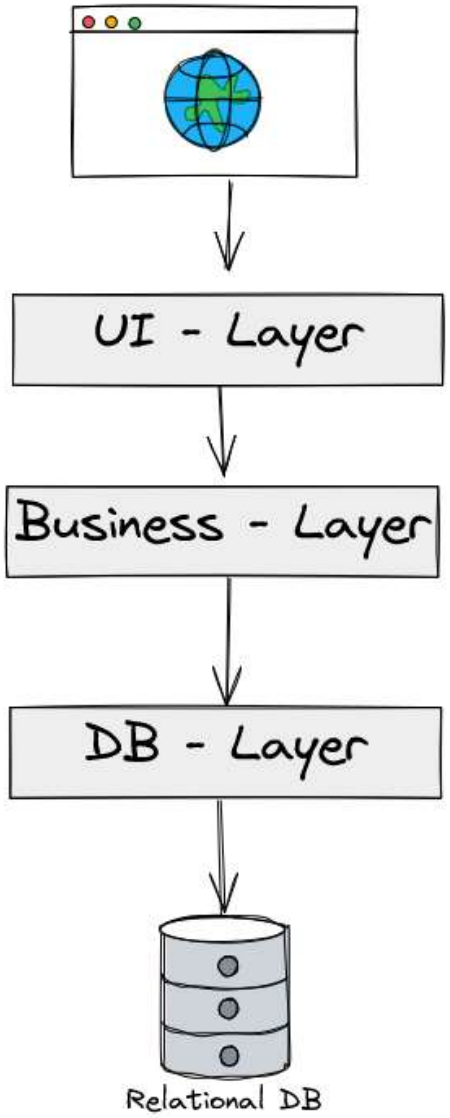



NO ANGELS
Imagine it's April...
2000

Schichten Architektur

Eigenschaften

- obere Schicht (Layer) hängt nur von der darunter liegenden Schicht ab
- erlaubt eine Abstraktion von der konkreten Implementierung eines Layers



Alistair Cockburn

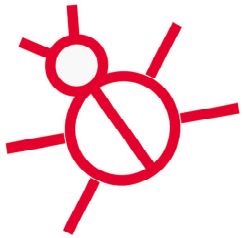
Imagine it's April...

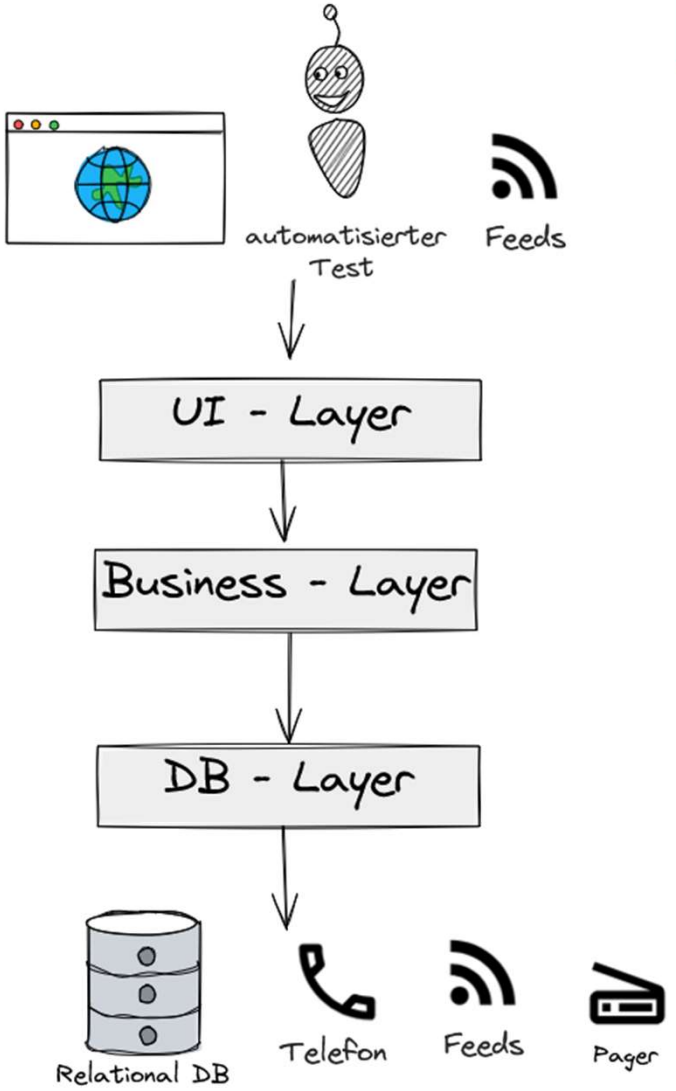
2000

Schichten Architektur

Wetterwarn-System

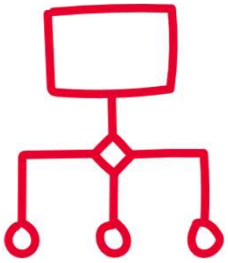
- sehr viele Datenquellen
- zahlreiche unterschiedliche Kanäle



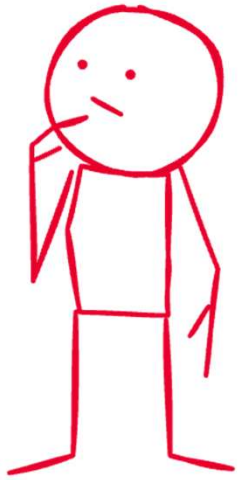


Wartbarkeit

Wartbarkeit: analysierbar, änderbar, erweiterbar, testbar



- Komplexität erfassbar
- Änderungen möglichst lokal
- Dinge leicht hinzufügen können
- Tests möglichst schnell und einfach



Wartbarkeit der Geschäftslogik

Geschäftslogik bildete den Wert einer Applikation

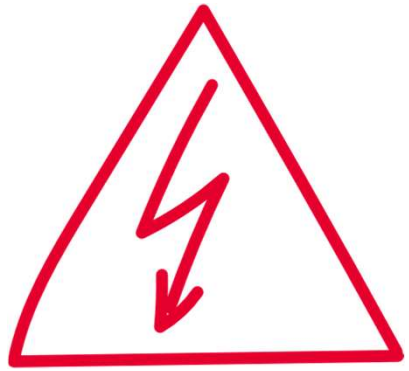
- Wettbewerbsvorteil
- inhärente Komplexität
- Lebenszyklus von Fachlichkeit & Technik sind unterschiedlich


Coding < 2010

IT-Landschaften der 2010er

Dinge die noch nicht erfunden bzw. kein Mainstream waren

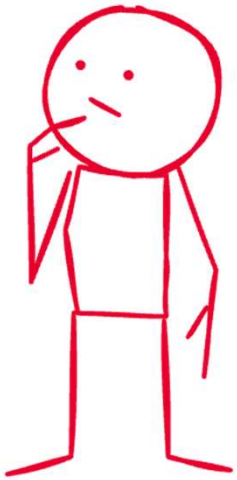
- DevOps
- Docker / Container
- Cloud / XaaS
- Statische Code-Analyse

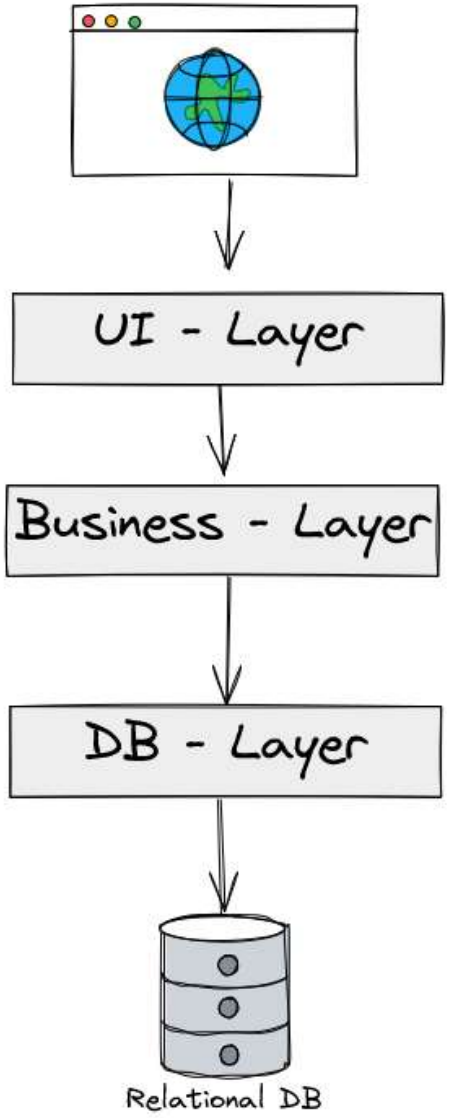


Auswirkungen

Wartbarkeit

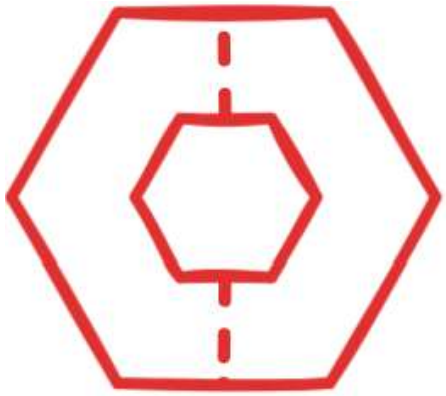
- Änderungen von Technik hat Auswirkungen auf Geschäftslogik
- Tests möglichst schnell < Integrationstests gegen zentrale DB



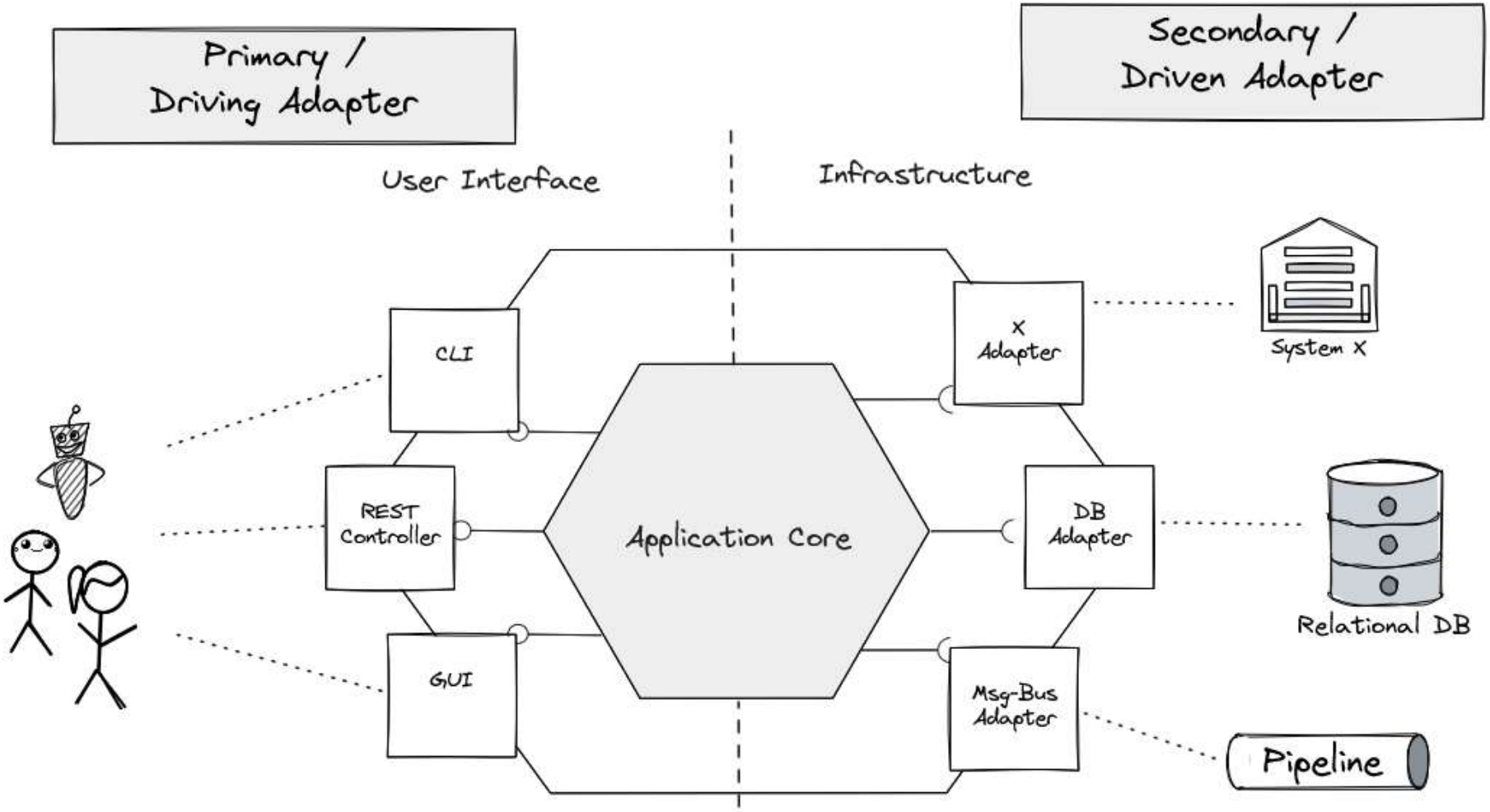


Ports und Adapters Pattern

Hexagonale Architektur



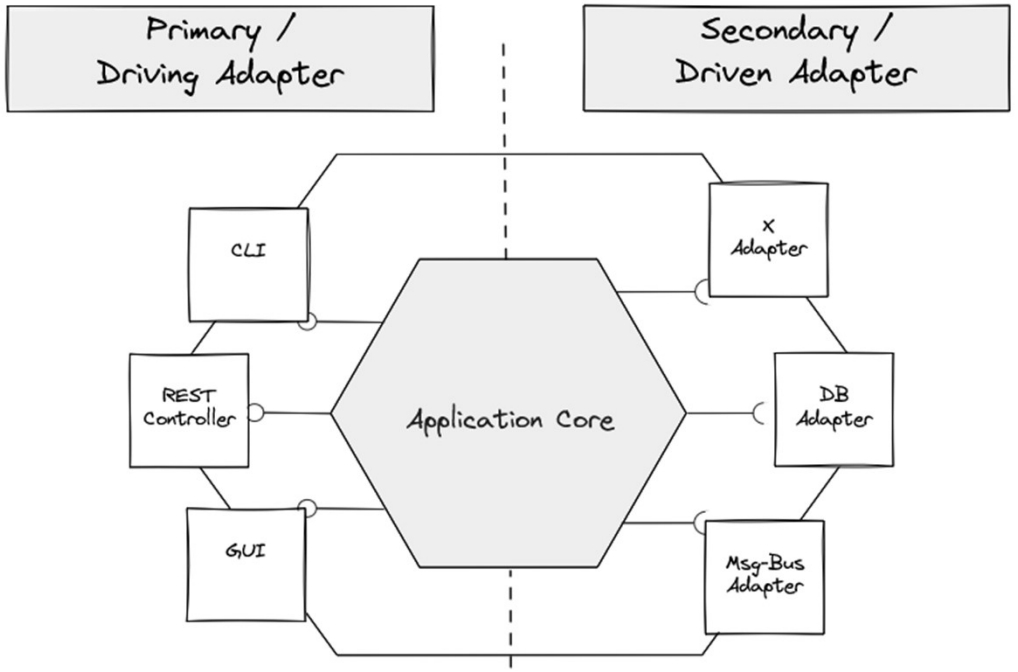
Hexagonale Architektur



Hexagonale Architektur

Eigenschaften

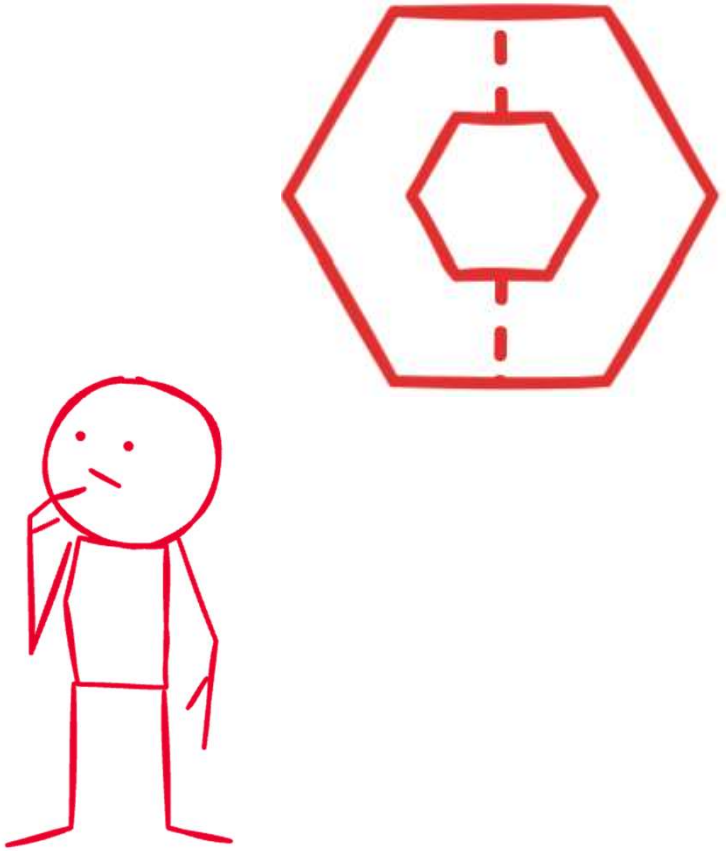
- alle Abhängigkeiten zeigen zur Fachlichkeit
- technische Adapter implementieren die Driving & Driven Ports



Auswirkungen

Wartbarkeit

- Änderungen von Technik durch Implementierung weiterer Adapter
- Tests können schnell sein, durch Nutzung z.B: eines DB-Mock-Adapters

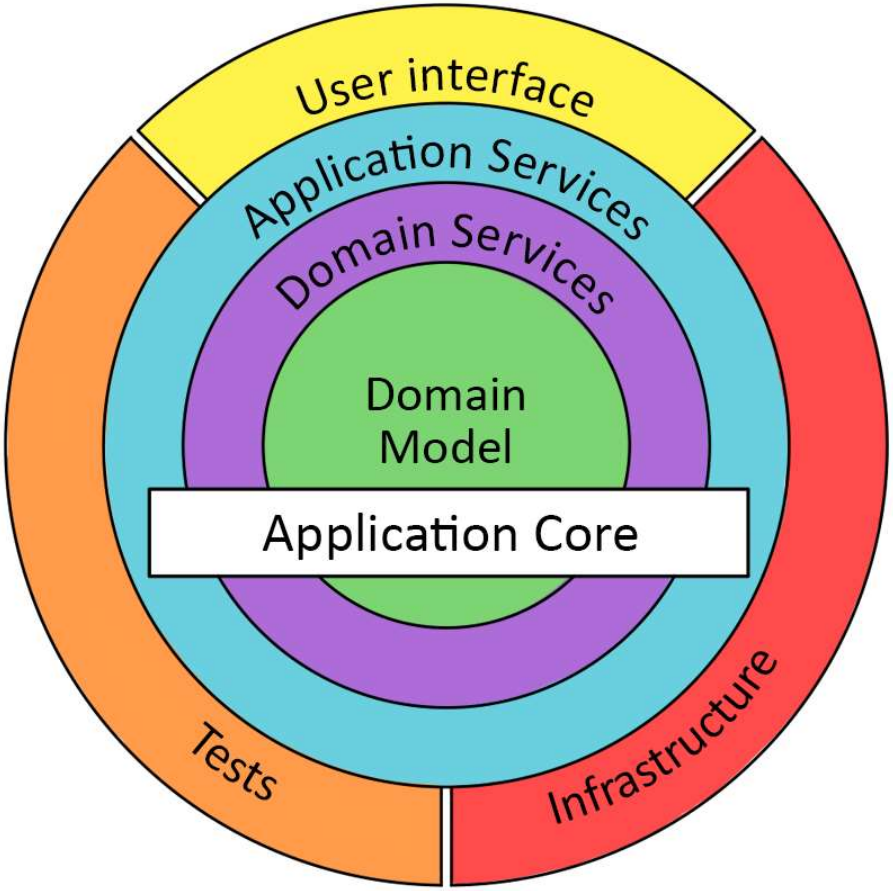


Geschmacks-Variationen

Jeffrey Palermo

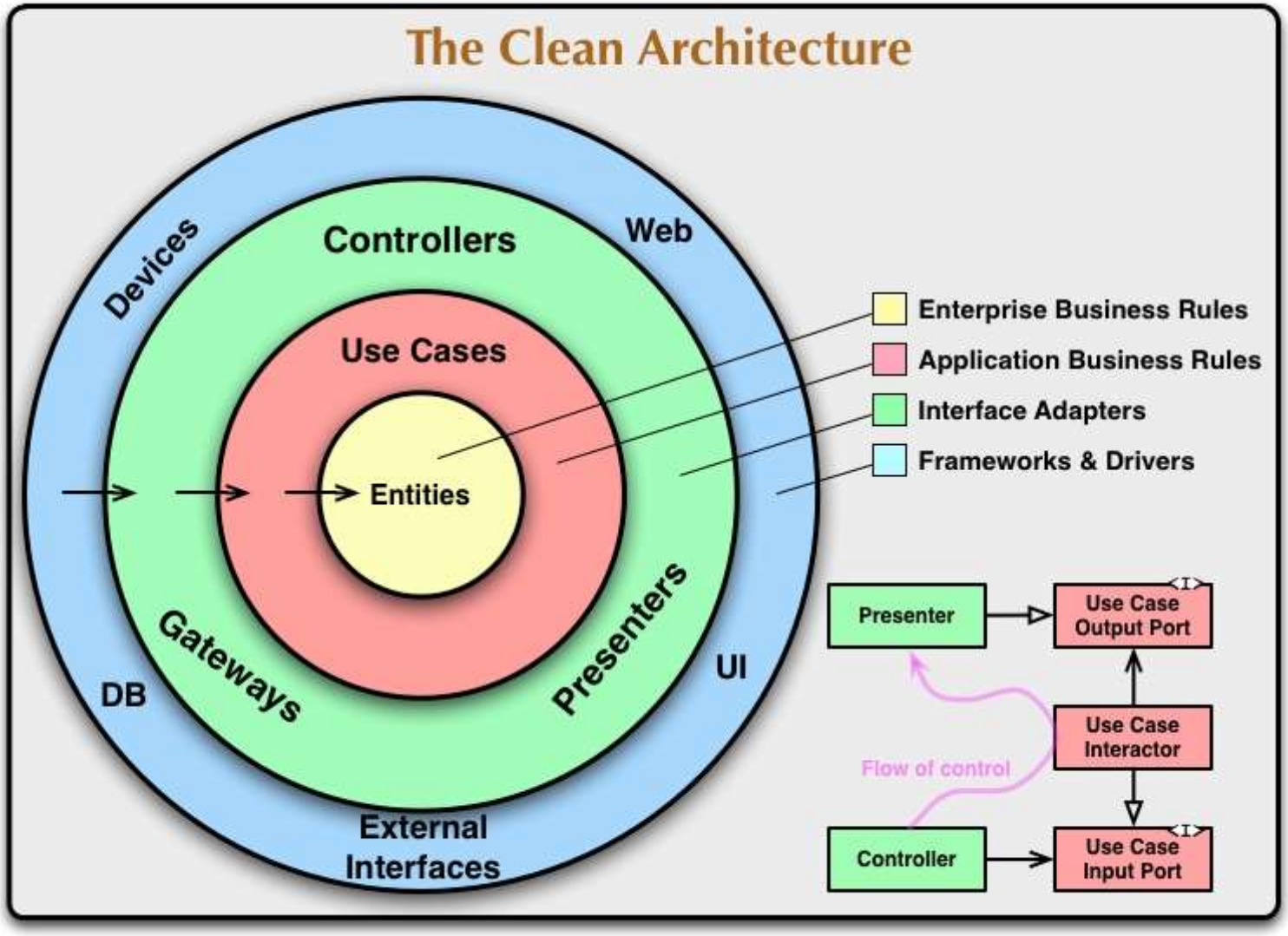
Onion Architecture

Onion Architecture



Robert C. Martin

Clean Architecture


The Clean Architecture
Devices
Controllers
Web
Use Cases
Entities
Gateways
Presenters
DB
UI
External
Interfaces
Enterprise Business Rules
Application Business Rules
Interface Adapters
Frameworks & Drivers
Presenter
Use Case
Output Port
Use Case
Interactor
Flow of control
Controller
Use Case
Input Port

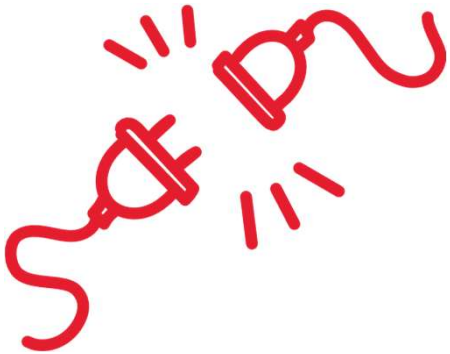
Problem solved? KI macht den REST...

github.com/alwessel/hexagon-workshop/

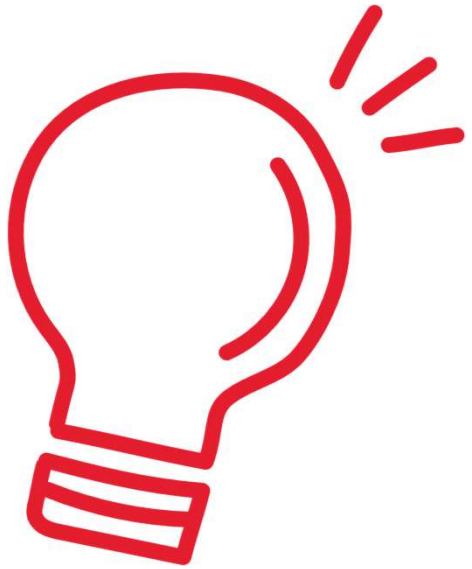
Schritt 1 – 5
bis 11:00 Uhr

Code

Hexagon im Code - Übung



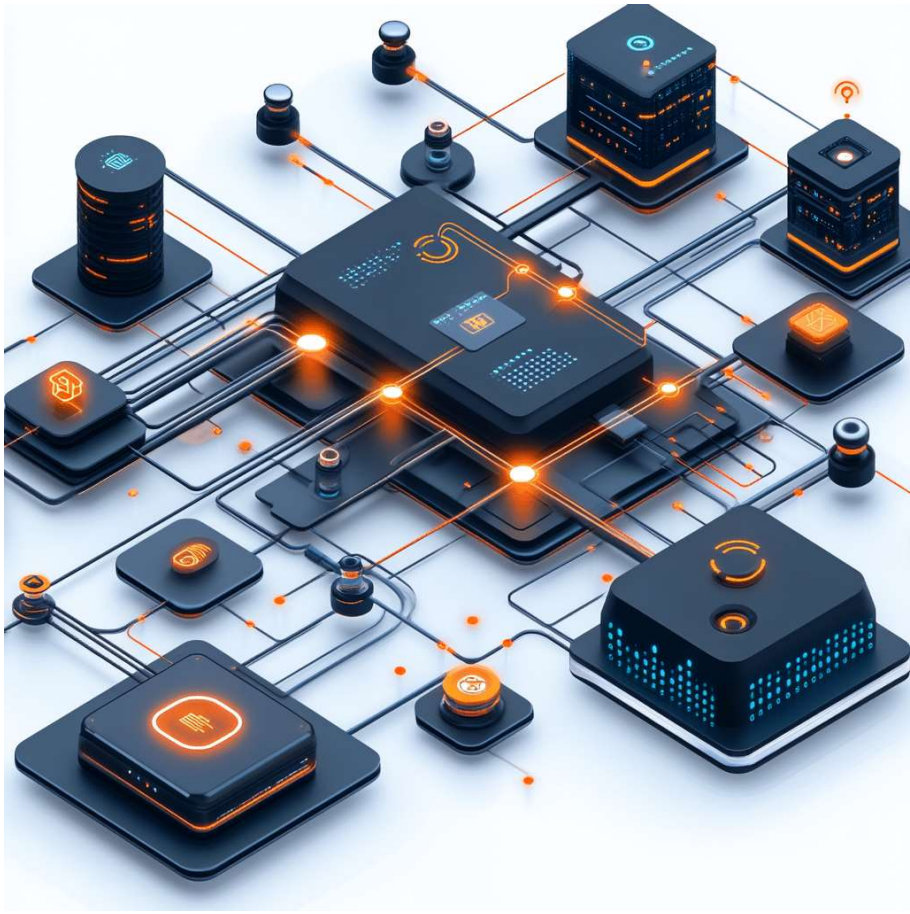
Fazit



Vorteile

Verbesserte Wartbarkeit durch

- Code der Geschäftslogik sauber / kleiner
- Entkopplung Technik / bessere Austauschbarkeit
- bessere Testbarkeit / Erweiterbarkeit



Nachteile

Mehraufwand durch Indirektion

- Viele Interfaces mit meist einer Implementierung
- Wie häufig ändern sich Adapter?

Alternativen

Hexagon Light?

- Trennung Fachlichkeit & Technik durch Frameworks

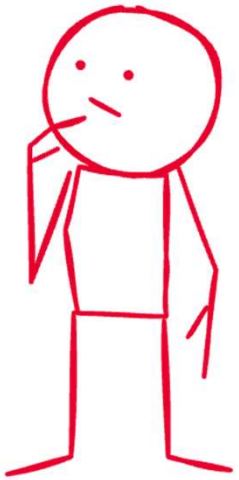
→ JPA

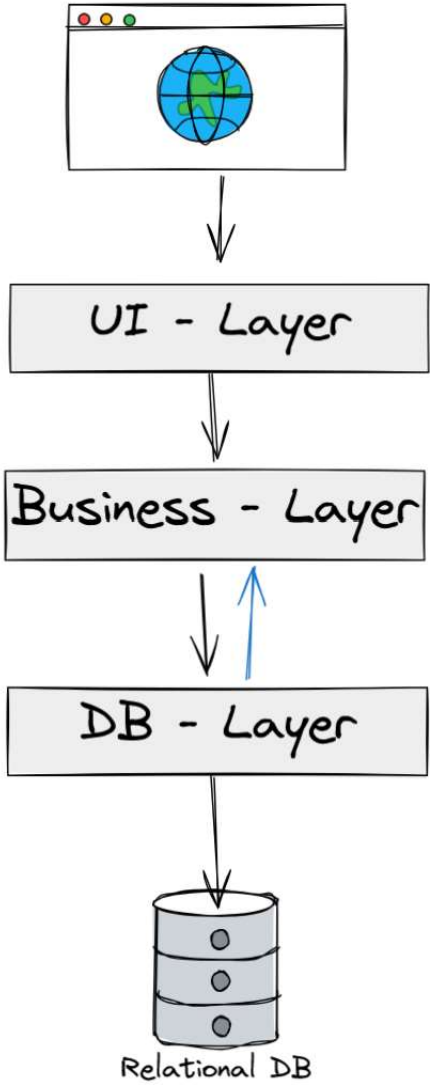
→ Spring Data

→ ArchUnit

- bessere Testbarkeit

→ Mockito & (Test-)Container

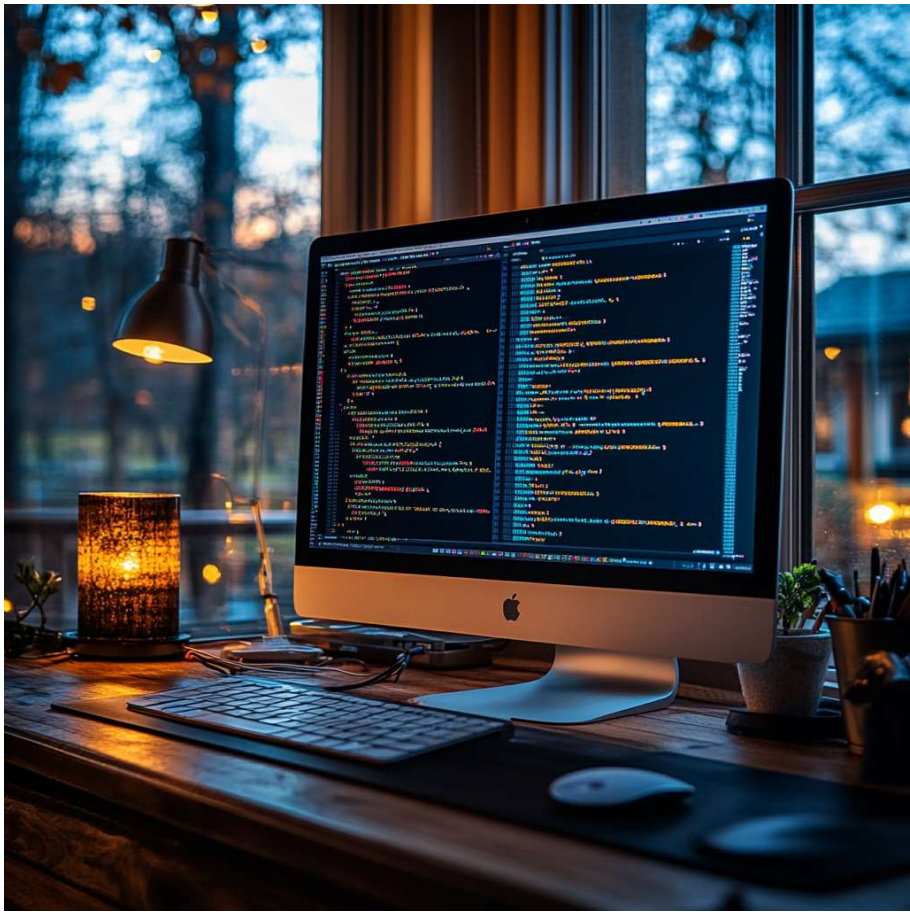




Eure Meinung

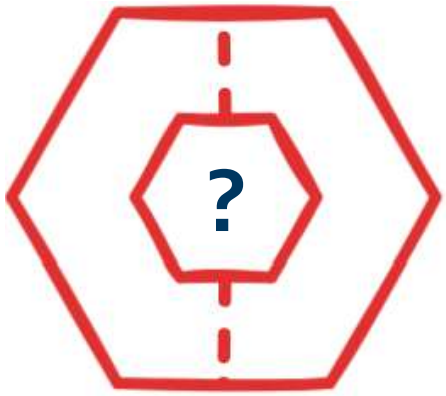
Vorteile / Nachteile

- Würdet ihr es nutzen und wenn ja wann?



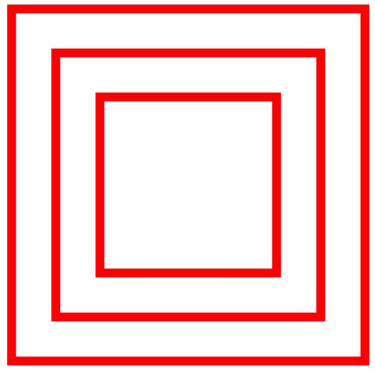
Weitere Konzepte

Pattern - Application Core



Eric Evans

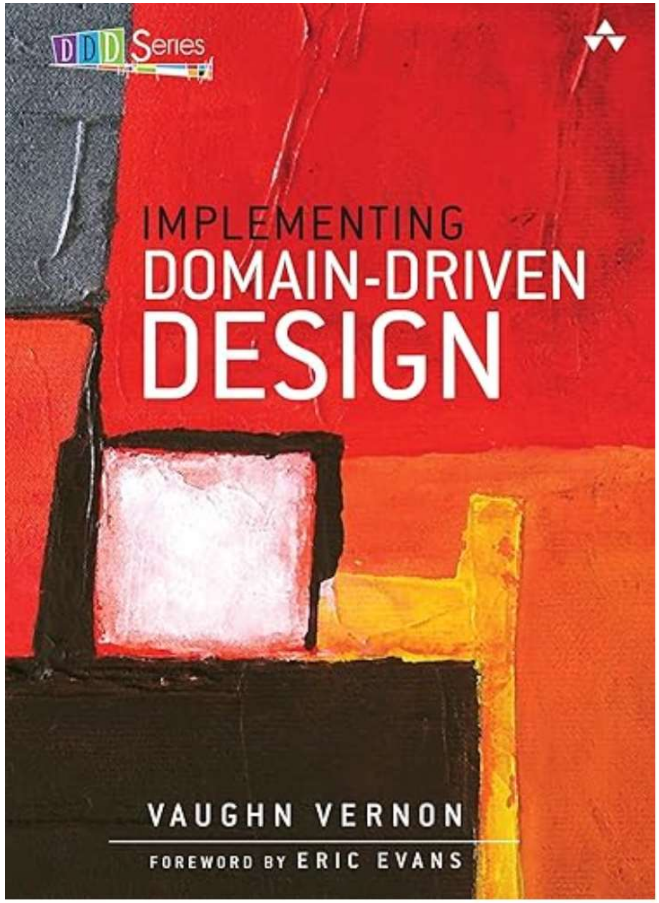
DDD – Domain Driven Design



Domain Driven Design

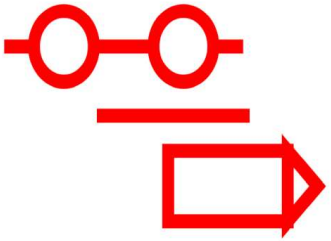
Fokus auf der Fachdomäne

- Fachlichkeit treibt die Architektur bis in den Code
- Ubiquitous Language - Einheitliche Begriffe und Konzepte sorgen für Verstädnnis von Fachexperten bis zum Entwickler



Bertrand Meyer

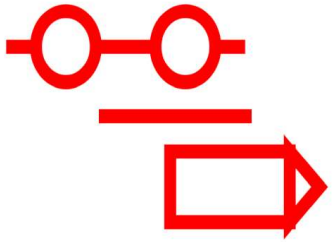
CQS - Command-Query-Separation



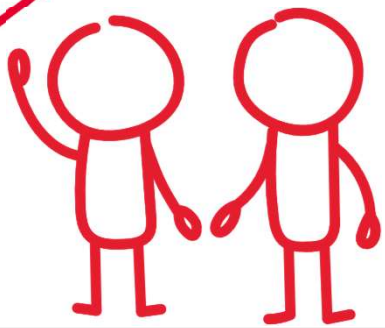
Command-Query-Separation

Trennt Lesen & Schreiben

- Separation of Concerns / Single Responsibility
- Insbesondere hilfreich bei Asymmetrie zwischen Lesen- & Schreibe-Last



Danke!
Fragen?



Chemnitz
European Capital
of Culture
2025

Feedback →

Weitere Informationen

- https://www.amazon.de/Hexagonal-Architecture-Explained-Alistair-Cockburn/dp/173751978X
- https://herbertograca.com/2017/11/16/explicit-architecture-01-ddd-hexagonal-onion-clean-cqrs-how-i-put-it-all-together/
- https://www.pentacor.de/post/was-ist-das-command-query-separation-pattern