

Architekturschemen

N-Tier Architektur:

- 3 Schichten: User Interface, Business Logic, Data Layer
- User Interface: Interaktionsebene
 - Controller
 - Views
- Business Logic: Anwendungsfälle
 - Application Logic
 - interaktion mit Datenebene
- Data Layer: Datenrepräsentation
 - Datenmodelle
 - Datenabfragen
 - Datenbankzugriff

User Interface

Business Logic

Data Access

Clean Architektur:

- Fokus auf Use Cases
- Applikation baut auf unabhängigen Datenmodellen auf
- innere Schichten (lila) definieren Schnittstellen, die von äußeren Schichten implementiert werden
- Infrastruktur unabhängig

Domain: - Datenmodelle

Applikation: - Implementierung der Use Cases unter Verwendung der Datenmodelle

- Aus Applikationssicht ist nur die Domain sichtbar
- Definition von Schnittstellen, die von Präsentation und Infrastruktur genutzt werden

Infrastruktur: - Datenbankzugriff, Services, 3rd Party Applikationen

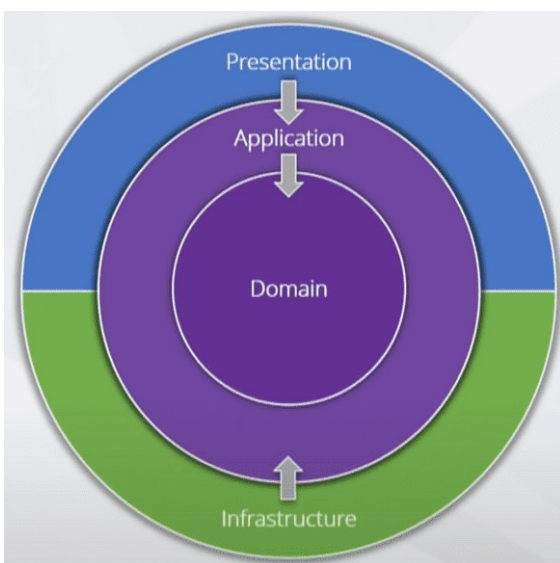
Präsentation: - entry point für user, Bearbeitung/weiterleitung von Useranfragen an die Applikation

Vorteile:

- einzelne Schichten sind voneinander losgelöst
 - > einfaches, vom Rest der Applikation unabhängiges Testen

Nachteile:

- selbst für kleine Features können viele Entitäten nötig sein
 - > over engineering



🍷 Architektur:

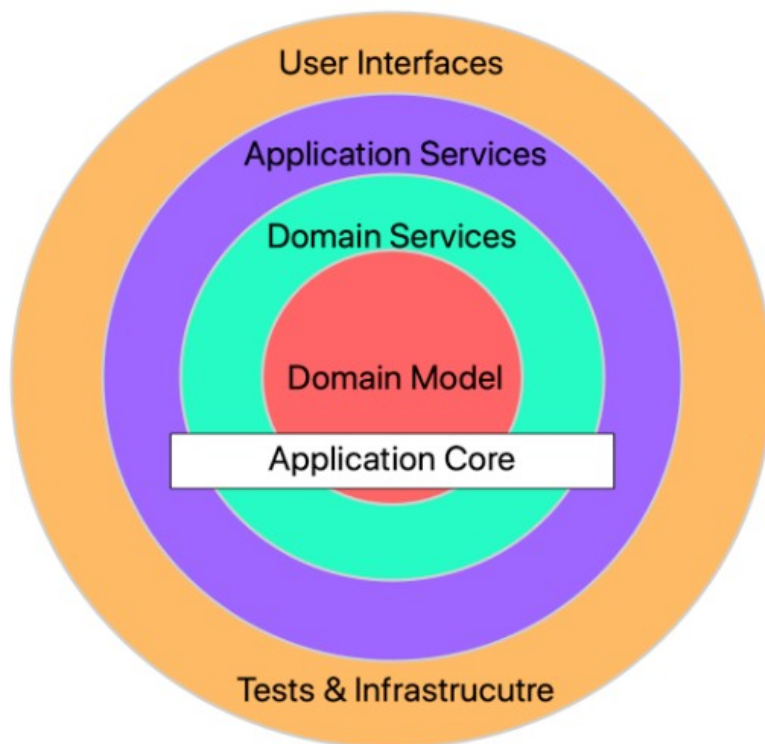
- Schichten sind von den darunter liegenden Schichten abhängig
 - > jedoch unabhängig von darüber liegenden Schichten
- äußere Schichten können ohne großen aufwand verändert werden

Vorteile:

- Flexibilität
- eindeutige Abhängigkeiten

Nachteile

- Viele Interfaces zur Kommunikation zwischen den Schichten notwendig
- Logik ist gebunden an den Core
 - Änderungen am Core können das Applikationsverhalten nach außen beeinträchtigen



Quellen:

<https://www.dandoescode.com/blog/clean-architecture-an-introduction/>
<https://tech.ovoenergy.com/onion-architecture/>