# Domain Driven Design

@OliverMilke
@cloudogu

# AGENDA

1. **meta**
2. Motivation
3. Taktisches Design
4. Strategisches Design
5. Event Storming

cloudogu

# About me

## Oliver Milke

Software Craftsman

- > 10 Jahre Softwareentwicklung
- Software Craftsman @Cloudogu EcoSystem
- JUG Ostfalen
- Fitness / Freeletics

http://oliver-milke.de/

https://twitter.com/OliverMilke

https://github.com/omilke

https://stackoverflow.com/users/2108919/omilke

# Wer sind Sie?

? objektorientierte Programmierung

? Branche

# AGENDA

1. meta
2. **Motivation**
3. Taktisches Design
4. Strategisches Design
5. Event Storming

# Herausforderungen

! Problemwelt verstehen

! Korrekte Lösung

✓
- In time
- In budget

! Software ist kein Selbstzweck

# Domain Driven Design

! Herangehensweise

✓ Abbildung von abgrenzbarem Problemfeld in Software

✓ Zusammenspiel
- Agile Software-Entwicklung
- Clean Code


THERE'S NO SILVER BULLET
@bryanMMathers

# Wissen über Domäne

! In den Köpfen der Experten

✓ Team aus Experten und Software-Entwicklern

✓ Wissen extrahieren

- Interviews
- Event Storming
- Domain Storytelling

# Fachliches Modell

! Abstraktion des Expertenwissen

✓ Software als Reflektion der Domäne

✓
- Dinge werden zu Klassen
- Umgangsformen zu Methoden

✓ Begriffe bleiben,
Software als Werkzeug

# Modell kommunizieren

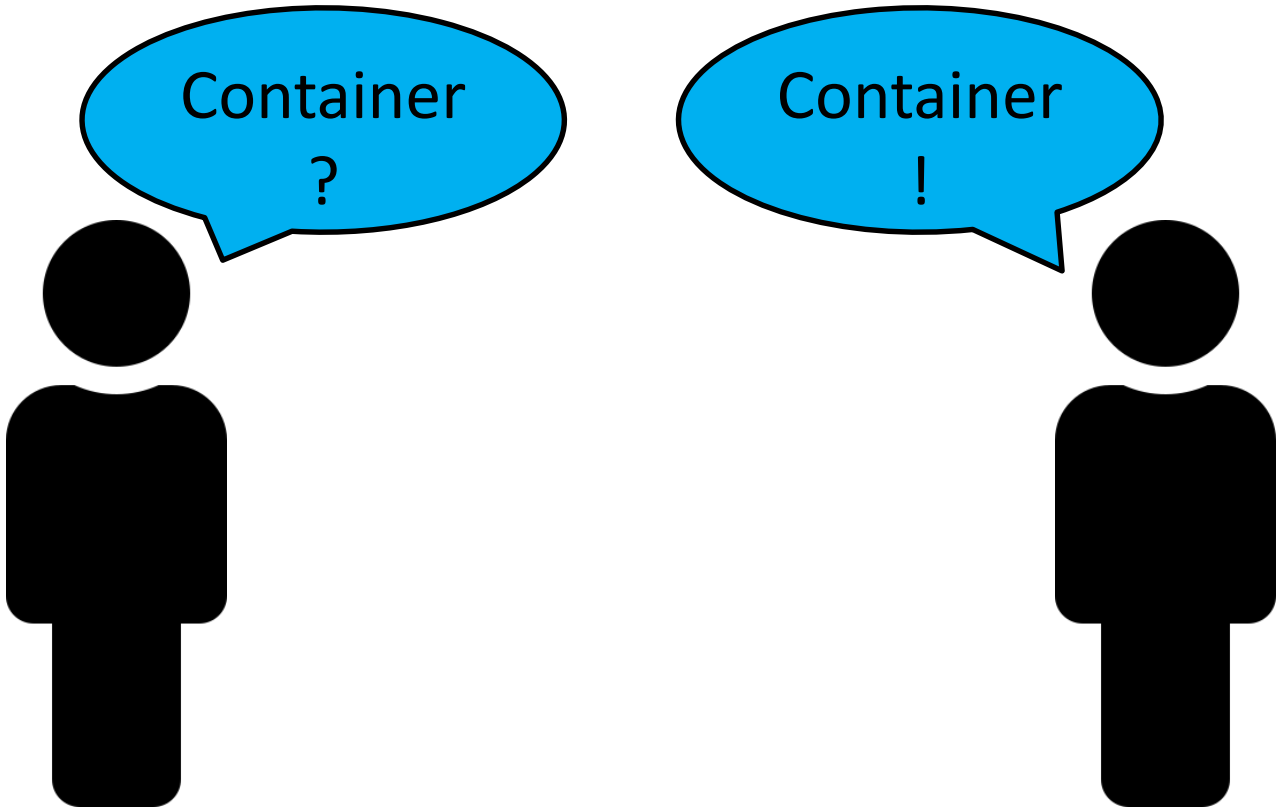
! Wissen muss verteilt werden

✓
- Grafisch
- Mündlich
- schriftlich

✓ Gemeinsame Sprache notwendig

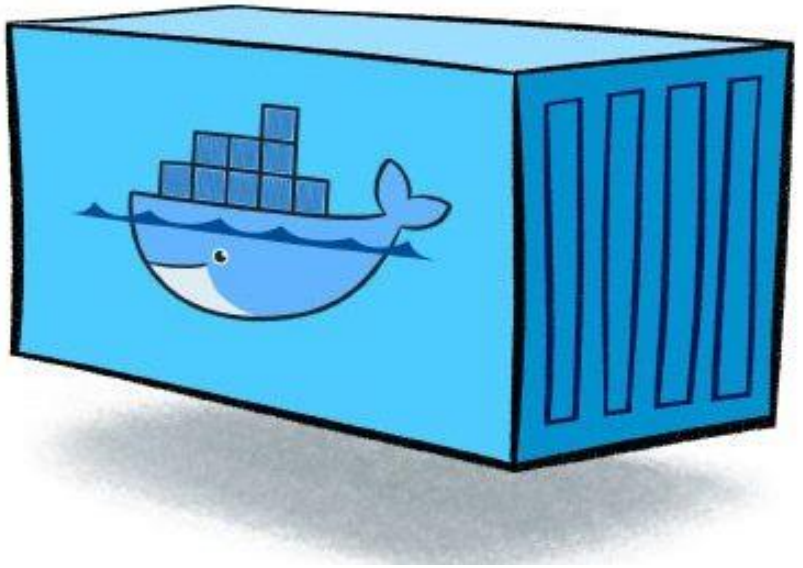
! Ubiquitous Language

# Ubiquitous Language

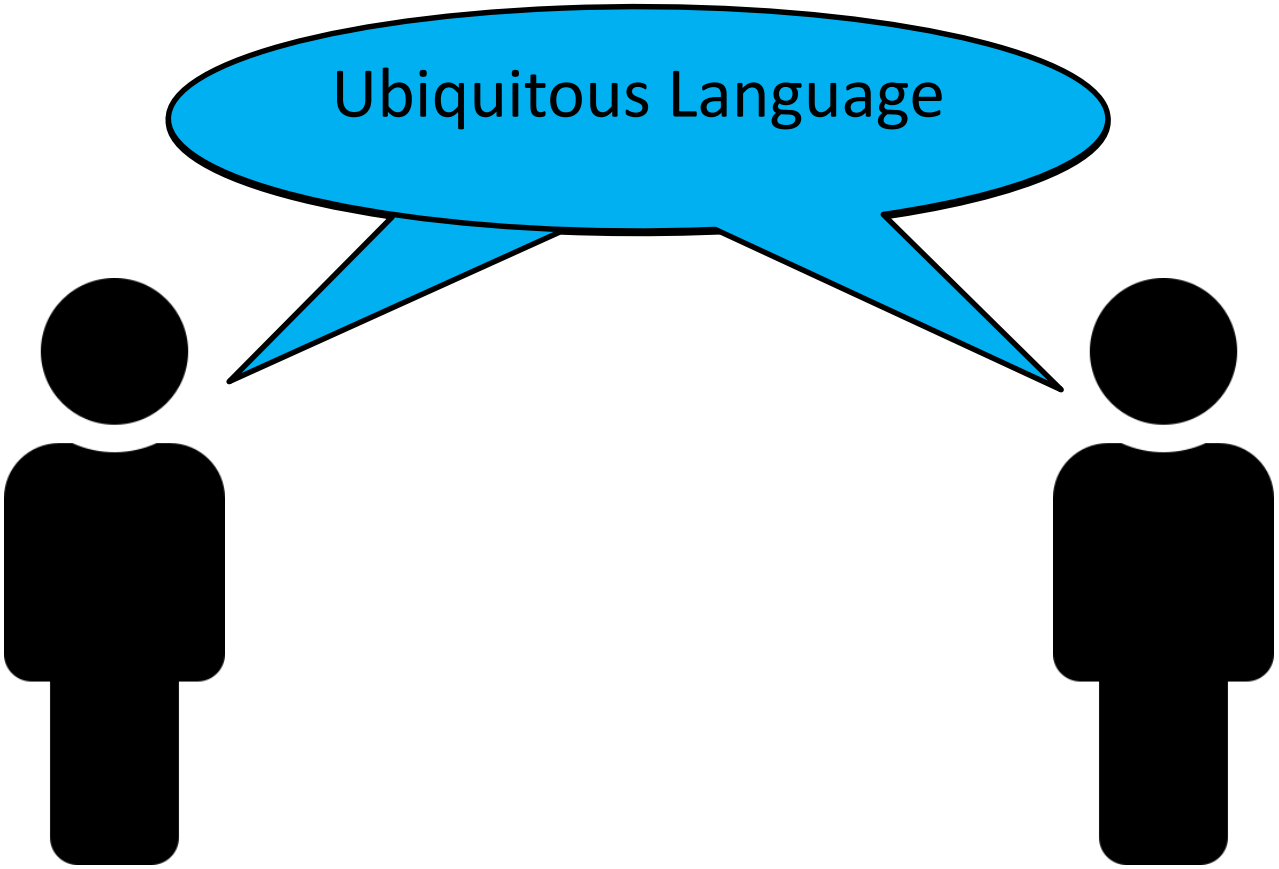


# Ubiquitous Language


https://commons.wikimedia.org/wiki/Container#/media/File:DTTX_724681_20050529_IL_Rochelle.jpg



# Ubiquitous Language



# AGENDA

1. meta
2. Motivation
3. Taktisches Design
4. Strategisches Design
5. **Event Storming**

# Event Storming

! Fachexperten + Softwareentwickler

✓ Mit
- lange Wand
- Stickies

Ohne
- Stühle
- Tische
- Diagramme

# Event Storming


https://en.wikipedia.org/wiki/Event_storming#/media/File:Event_Storming_example_process.jpg

# AGENDA

1. meta
2. Motivation
3. Taktisches Design
4. **Strategisches Design**
5. Event Storming

# Große Projekte

! Die Regel, nicht die Ausnahme

✓
- Mehrere Teams
- Parallele Entwicklung

? Aufteilung?

# Großes Modell

! Vollständige Abbildung

✓
- Koordination
- Komplexität

! Schwer erreichbares Ideal

# Separate Modelle

! Bewusste Aufteilung in Teilmodelle

✓
- Klare Abgrenzung
- Separate Entwicklung

! Ein Modell für ein Team

# Strategisches Design

! Realisierung im “Großen”

✓ Aufteilung der Domänen in verschiedene Sub-Domänen

✓
- Kern-Domäne
- Unterstützende Domänen
- Allgemeine Domänen

# Kern-Domäne

- ! Kritisch für den Unternehmenserfolg
- ✓ Exzellente Realisierung ist Wettbewerbsvorteil
- ✓ Planungsfeature für Kino

# Sekundäre Domains

! Unterstützende Domäne

entscheidend, jedoch nicht Teil des Kerns

✓ Marketing-Feature für Kino

! Allgemeine Domäne

nicht spezailisiert, notwendig für Gesamtlösung

✓ Buchhaltung für Kino

# Problem- und Lösungs-raum

- ! Realität eines Unternehmens ist Problemraum
- ✓ Aufteilung in Domänen
- ! Software ist Lösungsraum
- ✓ Sub-Domänen haben 1- n  Bounded Contexts

# Bounded Contexts

! Modelle gültig in Bounded Contexts

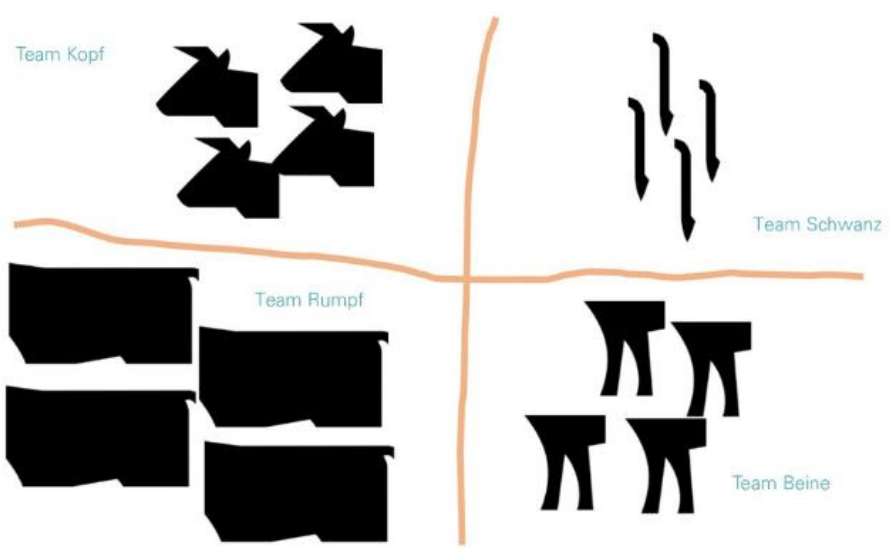
✓ Klare Grenzen in
- Team-Organisation
- Benutzung der Software
- DB-Schemata etc.
- Technologie

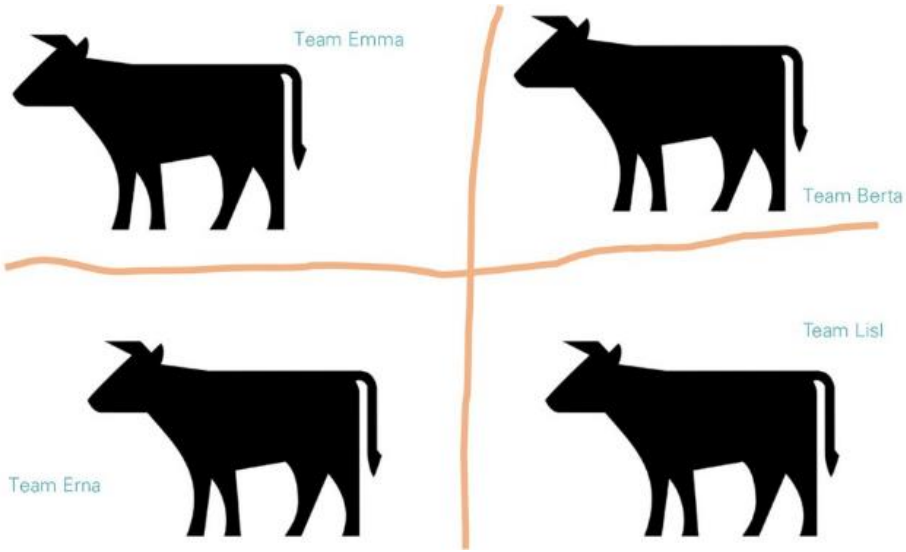
!
- Konsistenz innerhalb
- keine Ablenkung von außen

# Aufteilung der Domäne

! Geschäftsgetrieben – nicht Lösung getrieben

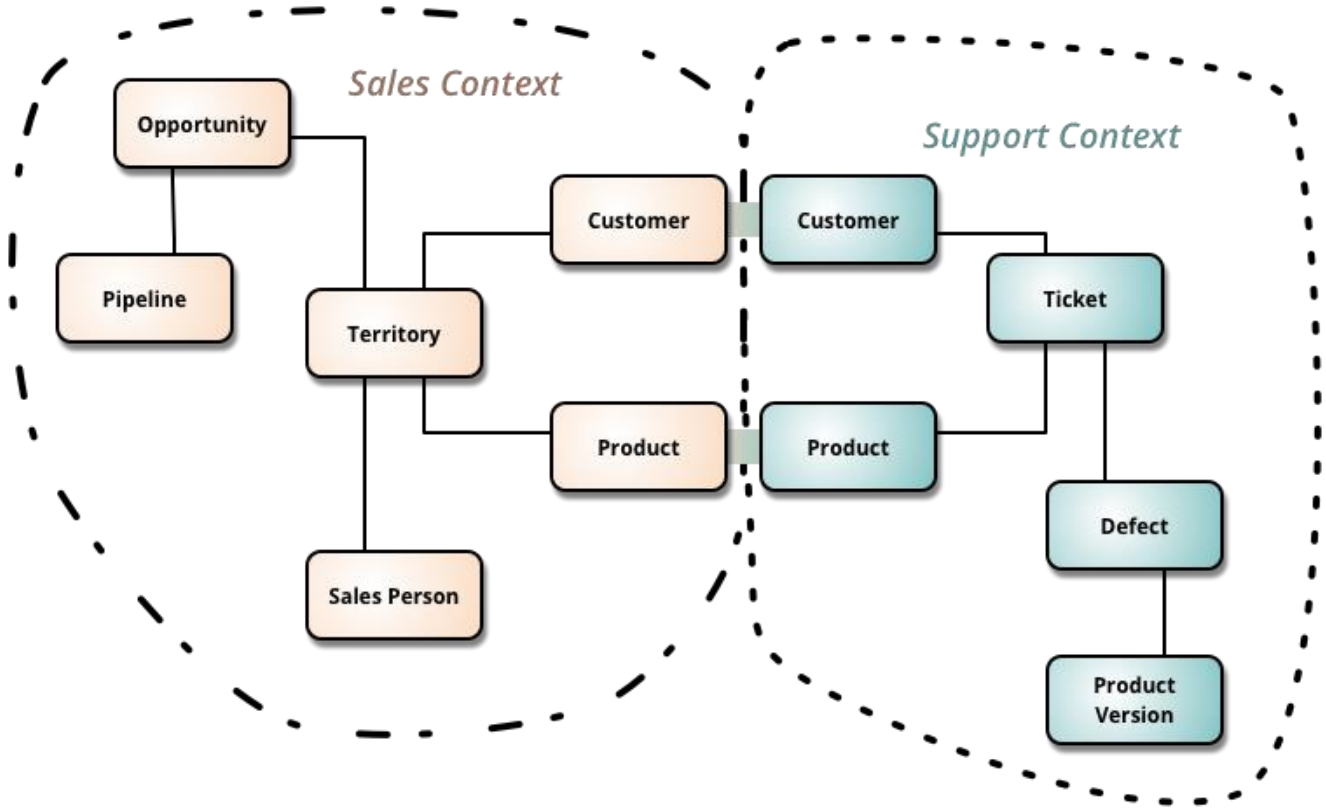
✓
- Grenzen in Prozessen
- Grenzen in Verantwortlichkeiten



# Context Mapping

! Zusammenhang zwischen Dingen aus verschiedenen Bounded Contexts




https://martinfowler.com/bliki/BoundedContext.html

# Preis der Freiheit

! Erhaltung der Grenzen

✓
- Definition der Modellgrenzen
- Definition der Beziehungen
- Technische Trennung

! Integrationsstrategien
- Shared Kernell
- Open-Host-Service
- Separate Ways

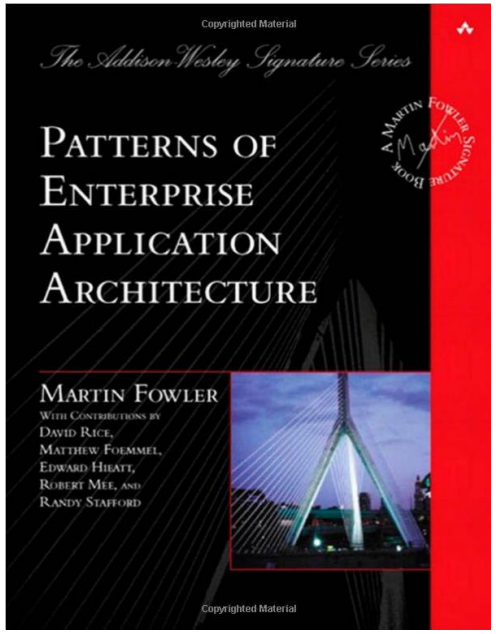
# AGENDA

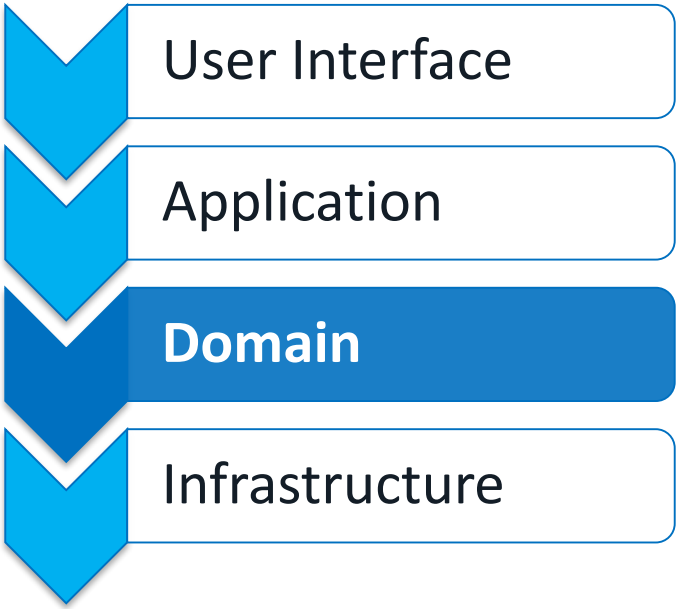
1. meta
2. Motivation
3. **Taktisches Design**
4. Strategisches Design
5. Event Storming

# Taktisches Design

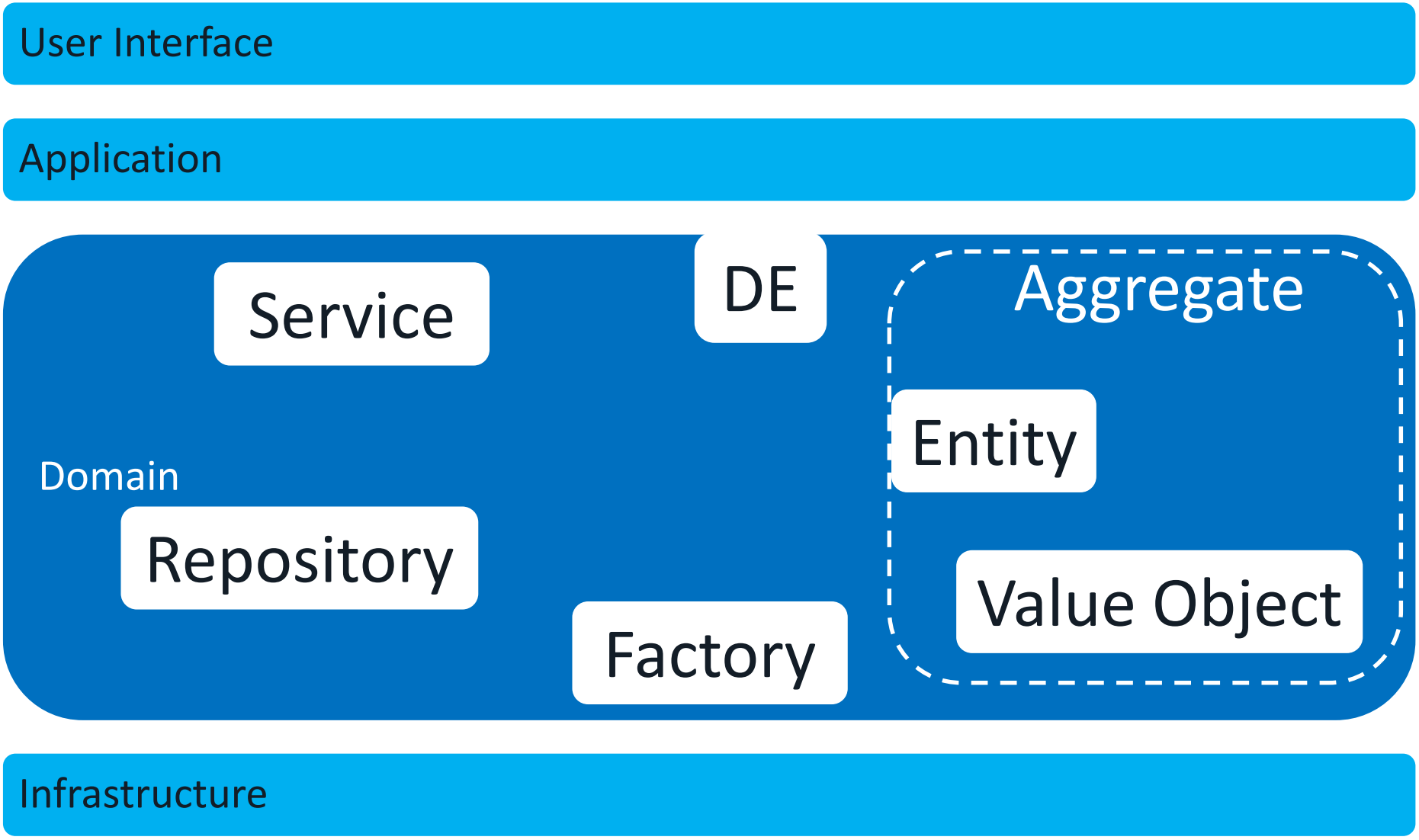
! Realisierung im “Kleinen”

✓ Fachliches Modell mit OOP
Gegenstände und Umgangsformen



# Elemente der Domain



# Value Objects

! Werte einer Fachdomäne

✓
- Ohne Identität
- Unveränderlich
- Operationen

✓ Kombination aus anderen Value Objects

✓
- 2,5
- 5/2
- 2 ½
- zwei-einhalb

# Entities

! Kernobjekte einer Fachdomäne

✓
- Identität
- Zustand
- Lebenszyklus

✓ Aufgebaut aus Value Objects

✓
- Bestellung
- Urlaubsantrag
- Behandlungsfall

# Aggregate

! konsistente Einheit aus (mehreren) Entities

✓
- Wurzel-Entity als zentrales Element
- Operationen

✓ Referenzierung über Identität

✓
- Bestellung mit Artikeln

# Services

! Übergreifende Komponente für Abläufe / Prozesse

✓
- Optional
- Zustandslos

✓ Transformieren / Produzieren
- Value Objects
- Entities

# Repositories

! Lagerverwaltung für Aggregate

✓
- Speichern
- Laden

✓
- Interface ist fachlich
- Implementierung kennt Infrastruktur

# Factories

Hilfselemente

- Erhalten Kapselung
- Rein fachlich

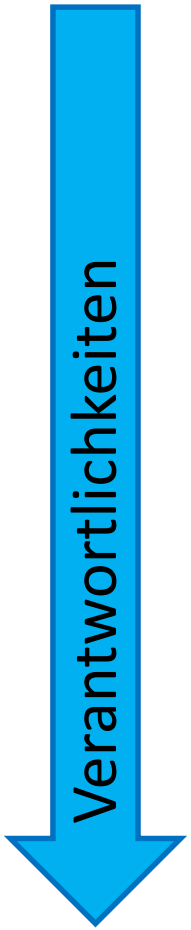
# Domain Events

Fachliche Ereignisse

- Bereits vollzogene Operationen
- Kommunikation zwischen Bounded Contexts

- Bestellung abgegeben
- Rechnung bezahlt

# Elemente der Domain

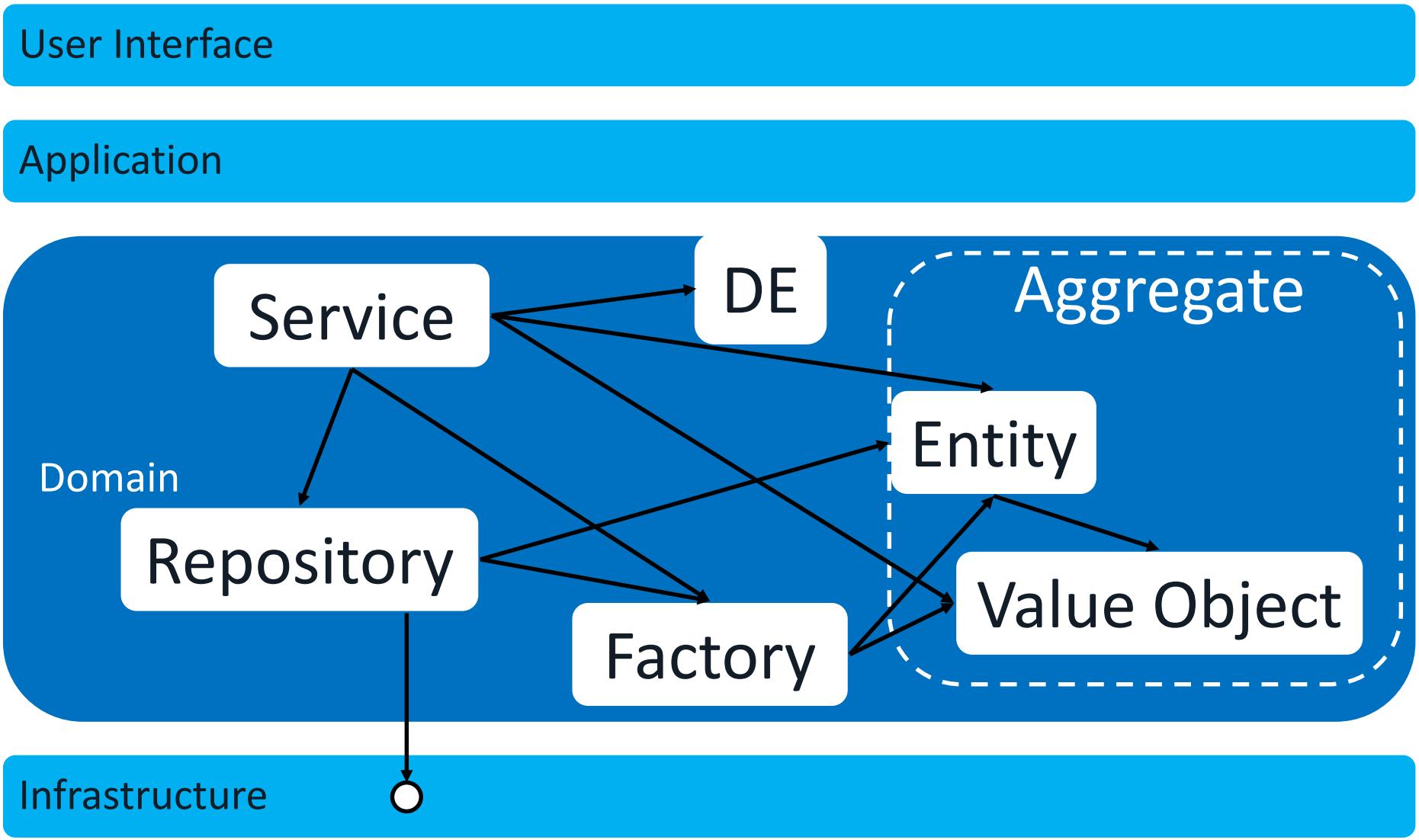



**Services** — zustandslos

**Entities** — Veränderlicher Zustand

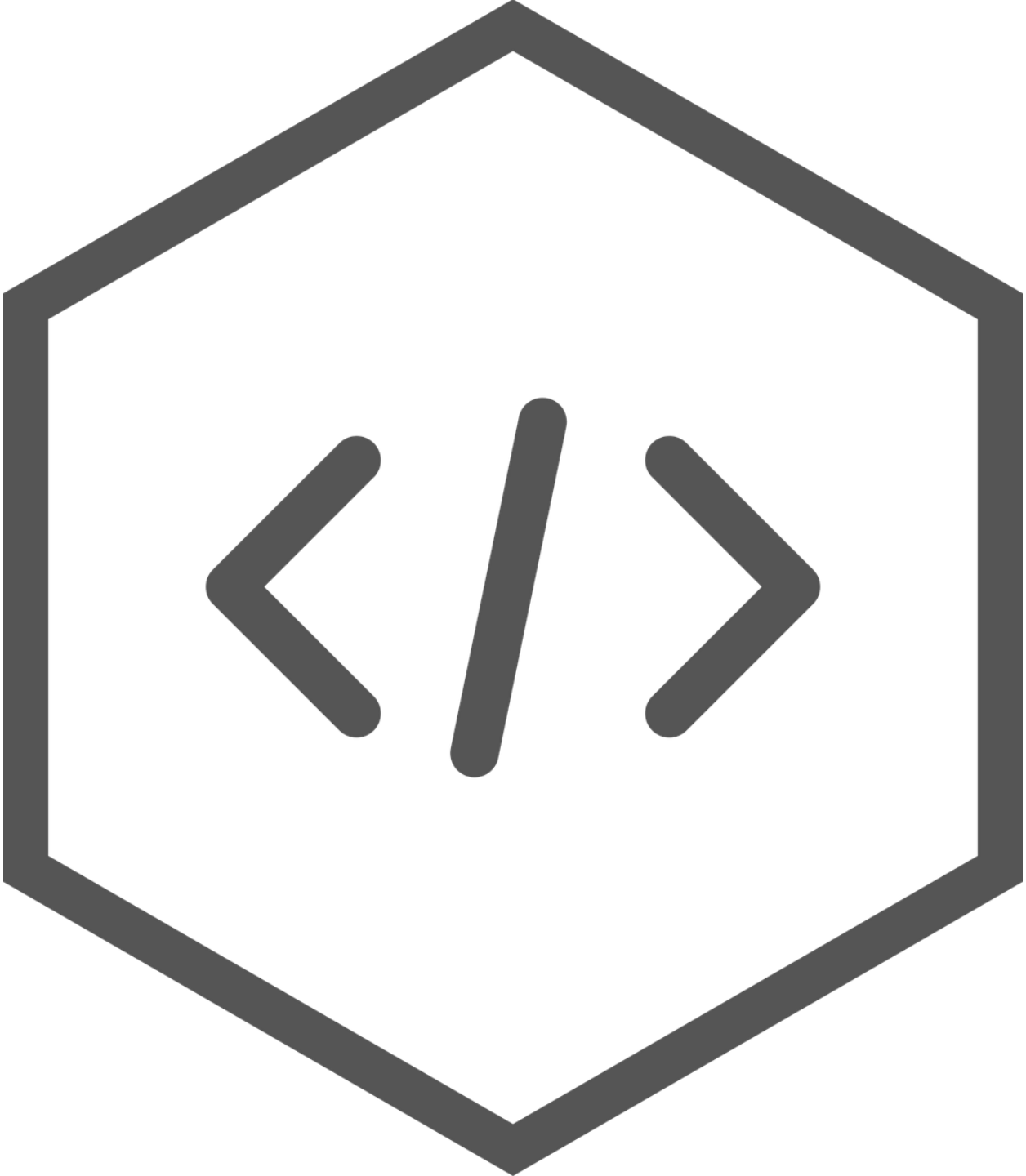
**Value Objects** — Unveränderlicher Zustand

# Elemente der Domain



# Anticorruption Layer

- ! Schutz des eigenen Modells
- ✓ Typischerweise in Repositories realisiert



# Schlussfolgerung

# Zusammenfassung

! Ganzheitliche Herangehensweise

✓
- Korrekte Lösung
- In time
- In budget

✓
- Taktisches Design

# Literatur

- Blaues Buch: Domain Driven Design, Eric Evans
- Rotes Buch: Implementing DDD, Vaughn Vernon

- Domain Driven Design kompakt
  - Vaughn Vernon
  - Carola Lilienthal, Henning Schwentner
- DDD Reference
  - Eric Evans

Oliver Milke

## Get in touch

- https://twitter.com/OliverMilke
- http://oliver-milke.de/
- dev@oliver-milke.de

Thank you

feedback plz