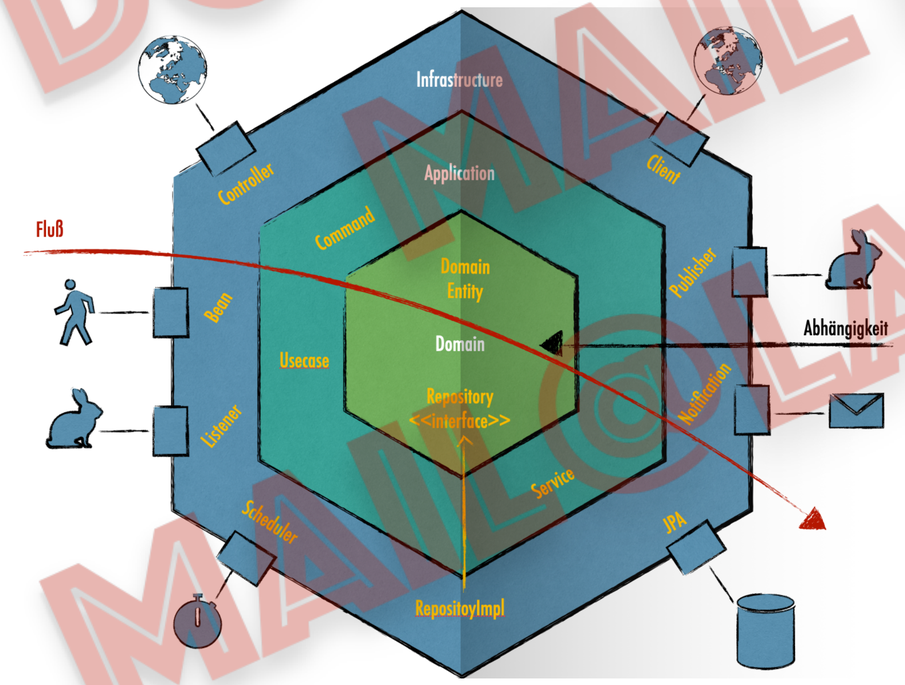
[Workshop]

Implementing DDD with hexagonal Architecture



mail@larmic.de / l.michaelis@neusta.de

@larmicDE

larmic

LarsMichaelis

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

Hinweis: Alle Bilder ohne Nachweis sind selbst erstellt oder von pixabay.com und zur freien kommerziellen Benutzung ohne Bildnachweis nutzbar

Tag 1

Willkommen und Motivation

Theorie DDD & hexagonale Architektur

Mittag

Vorbereitung des Projektes

Vorstellung Aufgabenstellung

Coding Challenge I

Räume

Tag 2

Coding Challenge II

Personen

Coding Challenge III

We are going deeper

~~Event Storming~~

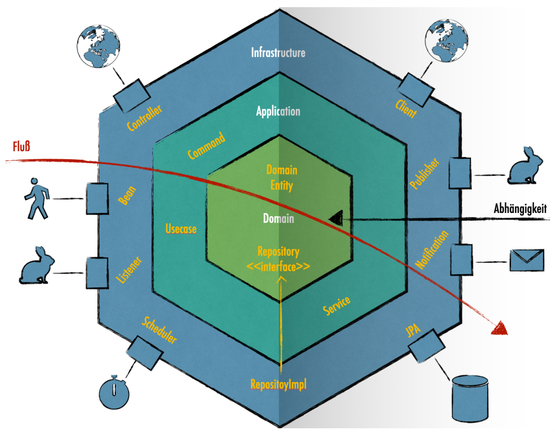
Abschluss und Feedback

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

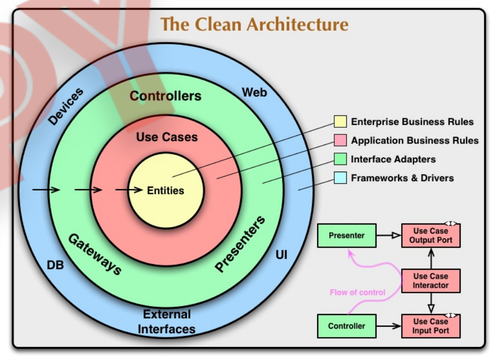
Akt 1

MOTIVATION

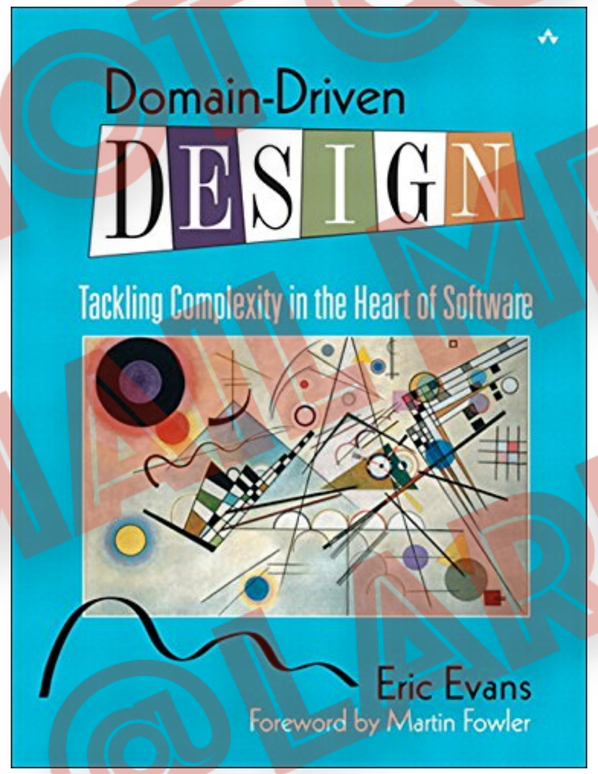
DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

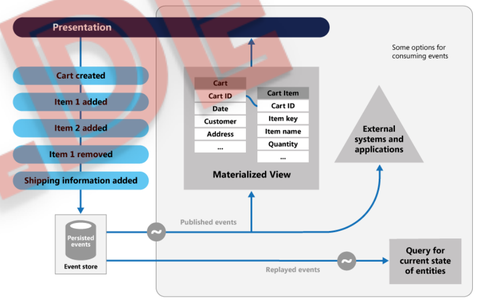


Hexagonale Architektur



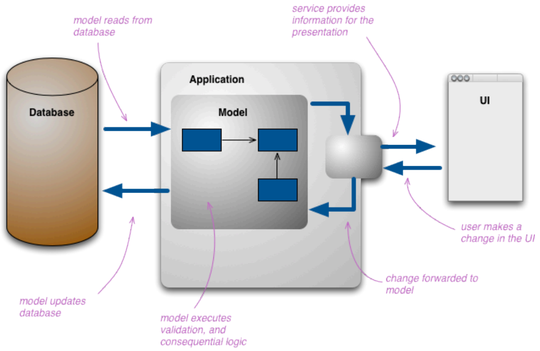
Clean Architecture





Event Sourcing

Event Storming



CQRS

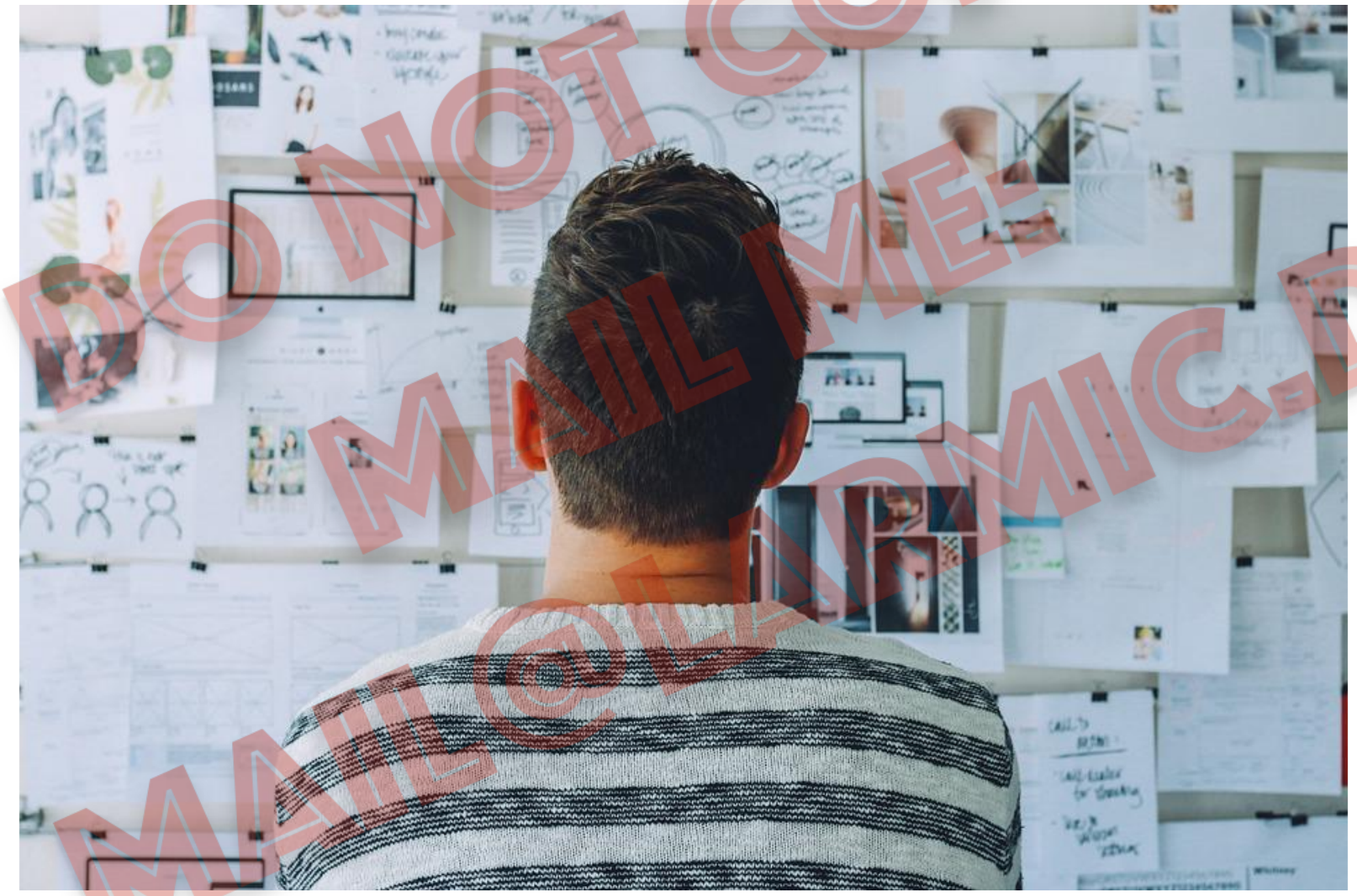
DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

Quellen: https://martinfowler.com/bliki/CQRS.html
https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html
https://docs.microsoft.com/de-de/azure/architecture/patterns/event-sourcing

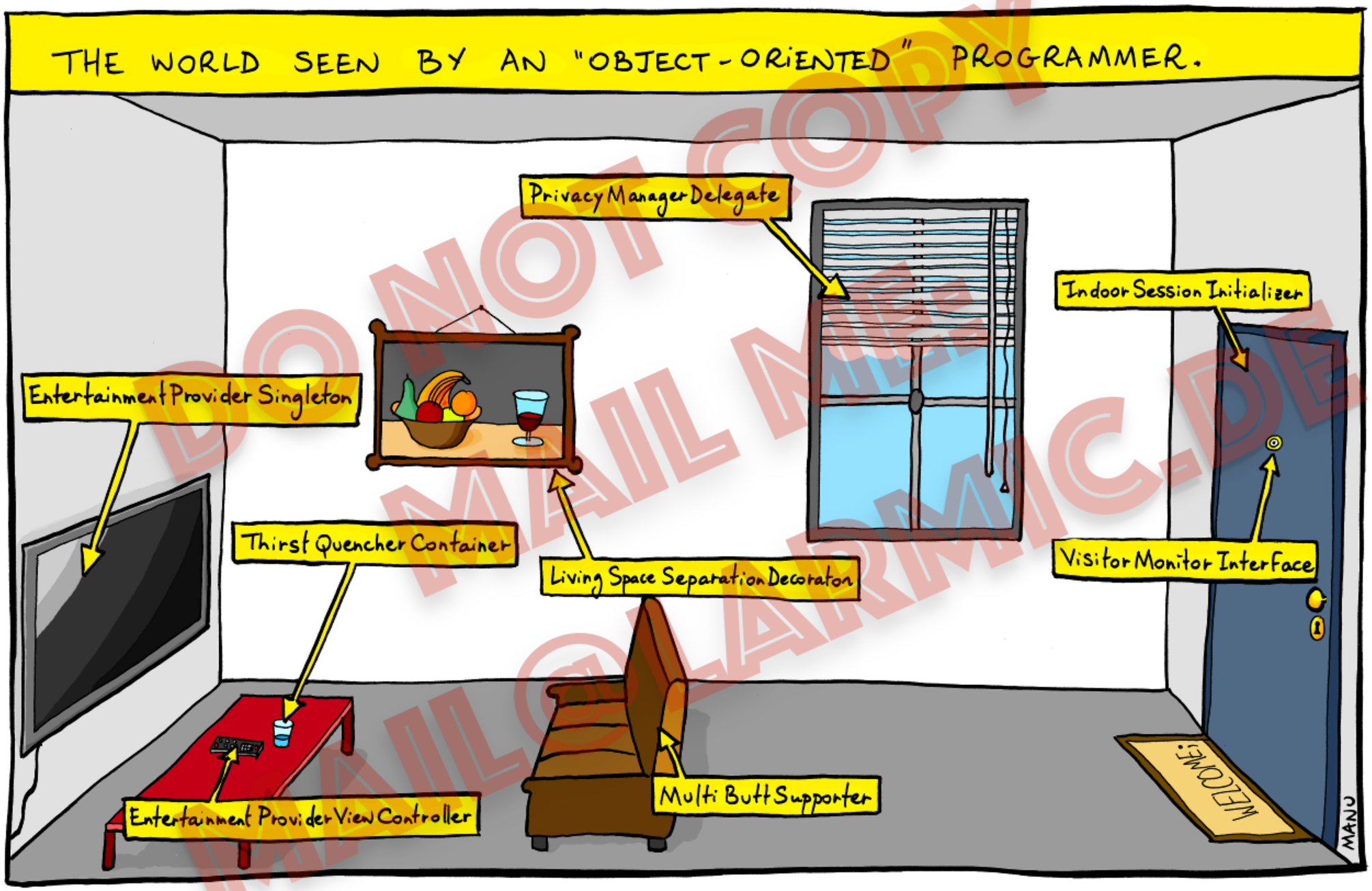
Kunde!

DO NOT COPY
MAIL ME:
MAIL@LARMIG.DE

Ich!



DO NOT COPY
MAIL ME
MAIL@LADMIC.DE



DO NOT COPY
MAIL MIR
MAIL@LARMIC.DE

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

I HOPE THE GUY WHO INVENTED AUTOCORRECT BURNS IN HELLO!

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

Was macht der Code?

```
let produktA = new Produkt(new Guid(Guid.create()), "Smartphone", new Preis(Waehrung.Euro, 420.00));
let produkte: Produkt[] = [];
produkte.push(produktA);

let rabatt = new Rabatt(produktA.id, 20, 1);

let warenkorb = new Warenkorb();
warenkorb.rabatteHinzufuegen([rabatt]);
warenkorb.hinzufuegen(produkte);

expect(warenkorb.gesamtkosten.betrag).toBe(336.00);
```

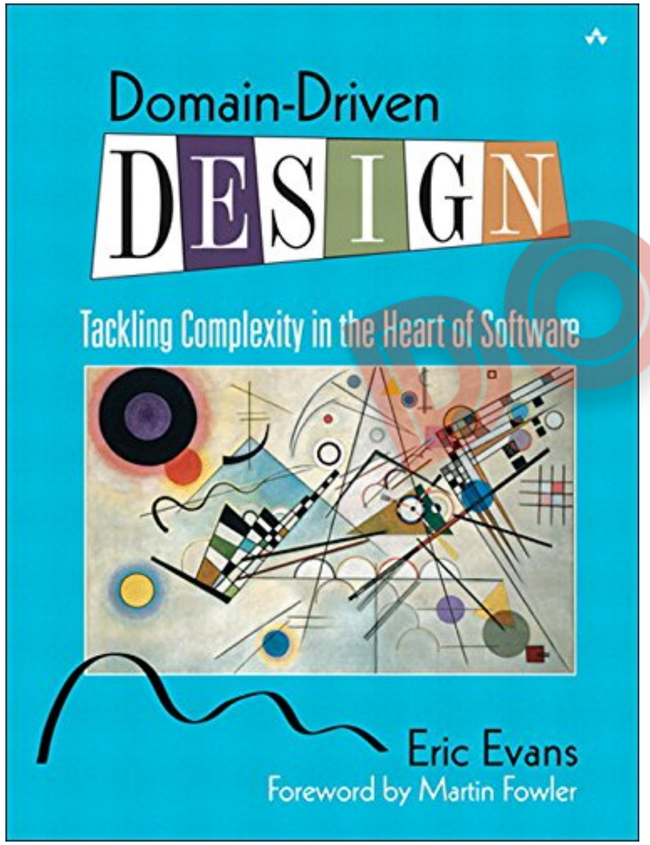
DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

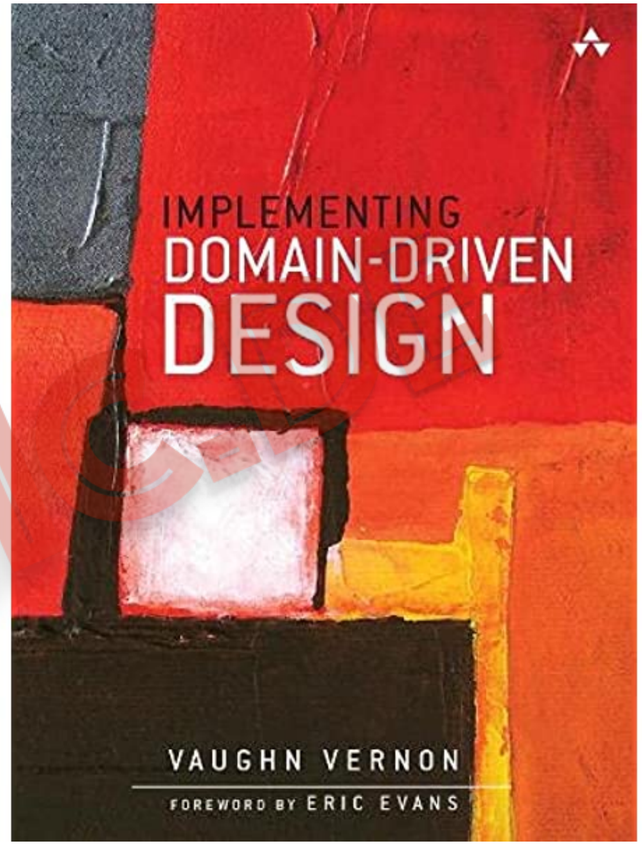
https://www.youtube.com/watch?v=tXXa4HAepZA&t=1964s

Akt 2

THEORIE DDD

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE





DO NOT COPY
MAIL ME:
MAIL@LARMIC.SK

Ziele / Hoffnungen

Es sind weniger Änderungen notwendig, weil die Fachlichkeit bekannt ist

Technische Änderungen haben keinen Impact auf die Fachlichkeit!

Weniger Stille-Post-Syndrom

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

Ziele / Hoffnungen

~~*Es sind weniger Änderungen notwendig, weil die Fachlichkeit bekannt ist*~~

Technische Änderungen haben keinen Impact auf die Fachlichkeit!

Weniger Stille-Post-Syndrom

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

„Der entscheidende Punkt, um den es mir geht, ist auch der Grund, warum das Konzept „Domain Driven Design“ heißt:

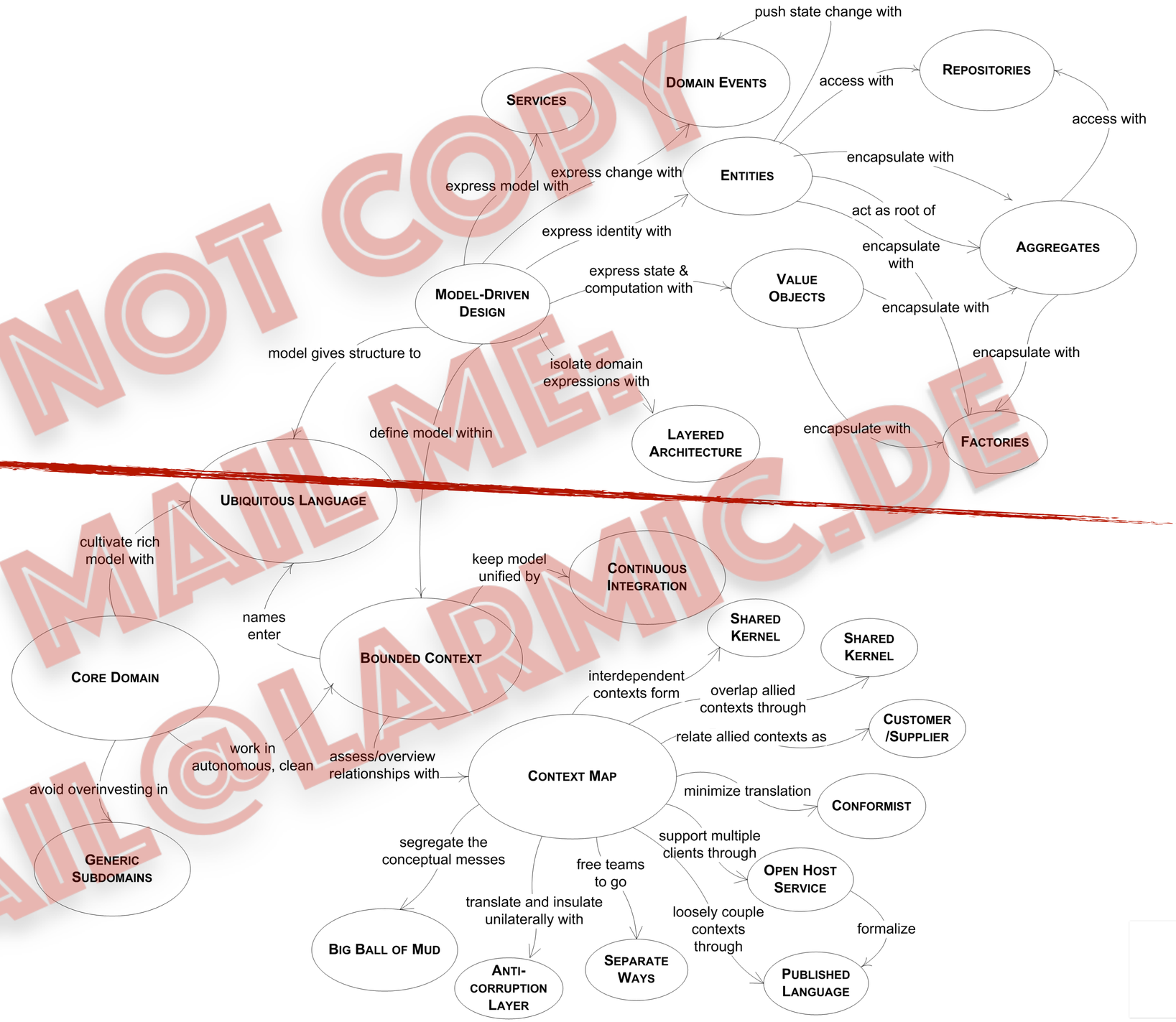
Wenn wir Software entwickeln, sollte unser Fokus nicht primär auf den Technologien liegen, die wir verwenden.

Stattdessen sollte sich unser Hauptaugenmerk auf die **geschäftliche Seite** richten, also auf den **fachlichen Bereich**, den wir mit unserer Software unterstützen wollen - **kurz: die Domäne**.“

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

Tactical Design

[Feiner Pinselstrich
Tiefe Flughöhe]



Strategic Design

[Grober Pinselstrich
Hohe Flughöhe]

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

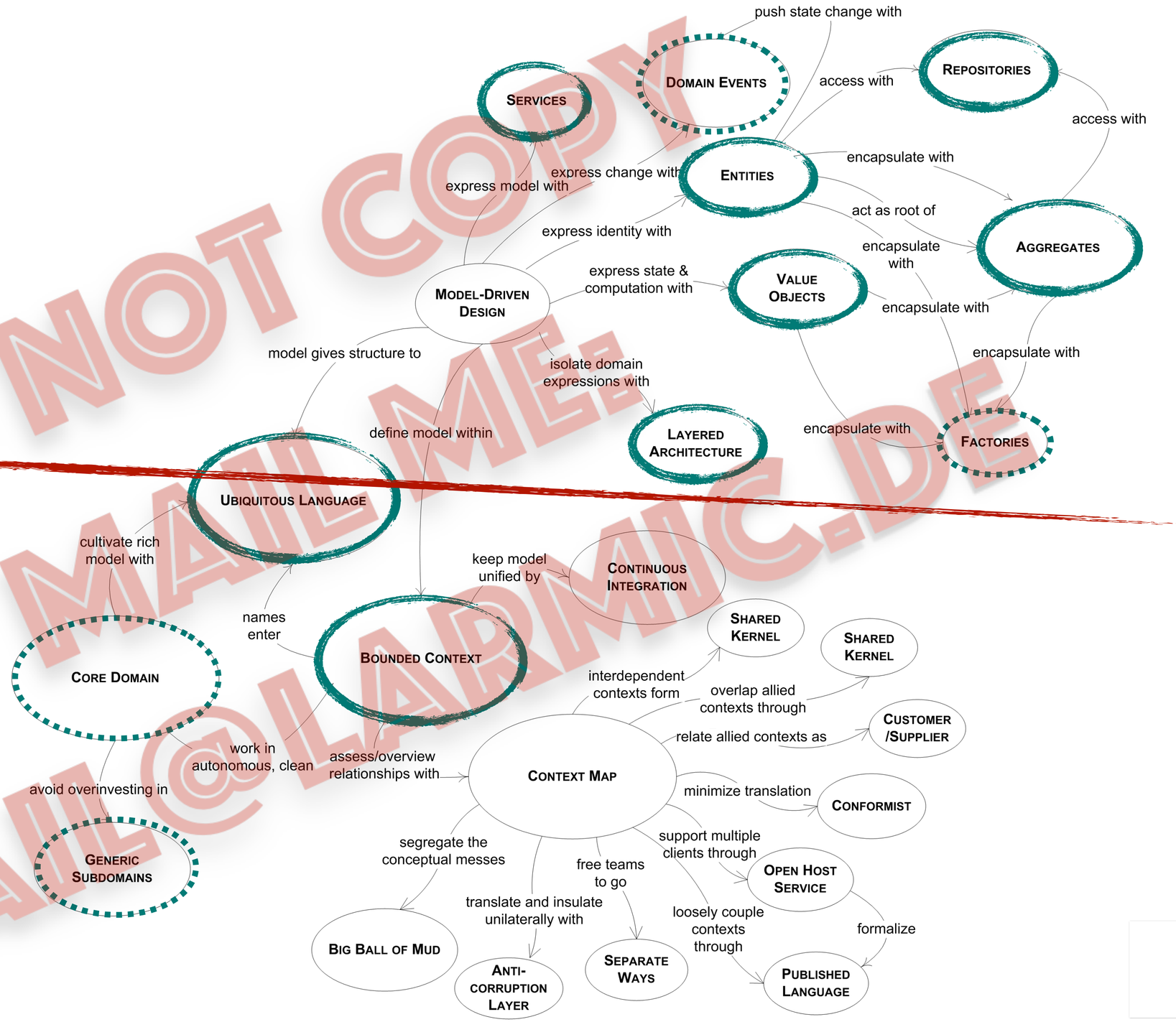
Quelle: https://www.domainlanguage.com/ddd/reference/

Tactical Design

[Feiner Pinselstrich
Tiefe Flughöhe]

Strategic Design

[Grober Pinselstrich
Hohe Flughöhe]



DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

Quelle: https://www.domainlanguage.com/ddd/reference/

Ubiquitous Language

[Allgegenwärtige Sprache]

„A language structured around the domain model
and used by **all team members within a bounded context**
to connect all the activities of the team with the software.“

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

Quelle: https://www.domainlanguage.com/ddd/reference/

Ubiquitous Language

[Allgegenwärtige Sprache]

„You can find the documents at the lager!“

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

Ubiquitous Language

[Allgegenwärtige Sprache]

„ clup_pin_honorary_office_silver "

=

Silberne Ehrennadel

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

Ubiquitous Language

[Allgegenwärtige Sprache]

Race != Breed

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

Ubiquitous Language

[Allgegenwärtige Sprache]

- Übersetzung bedeutet Genauigkeitsverlust & Begriffe müssen jedes mal übersetzt werden
- Missverständnisse werden häufiger.
- Code und Realität weichen eher und stärker voneinander ab und erschwert die Orientierung im Code
- Die Sprache wird weniger weiterentwickelt.

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

Bounded Context

[Beispiel: Shop]

Warenkorb

Bestellung

Artikel

Rabatt-Code

Produkt

Adresse

Bestellgewicht

Versandadresse

Kontodaten

Paypal

Kreditkarte

Giro

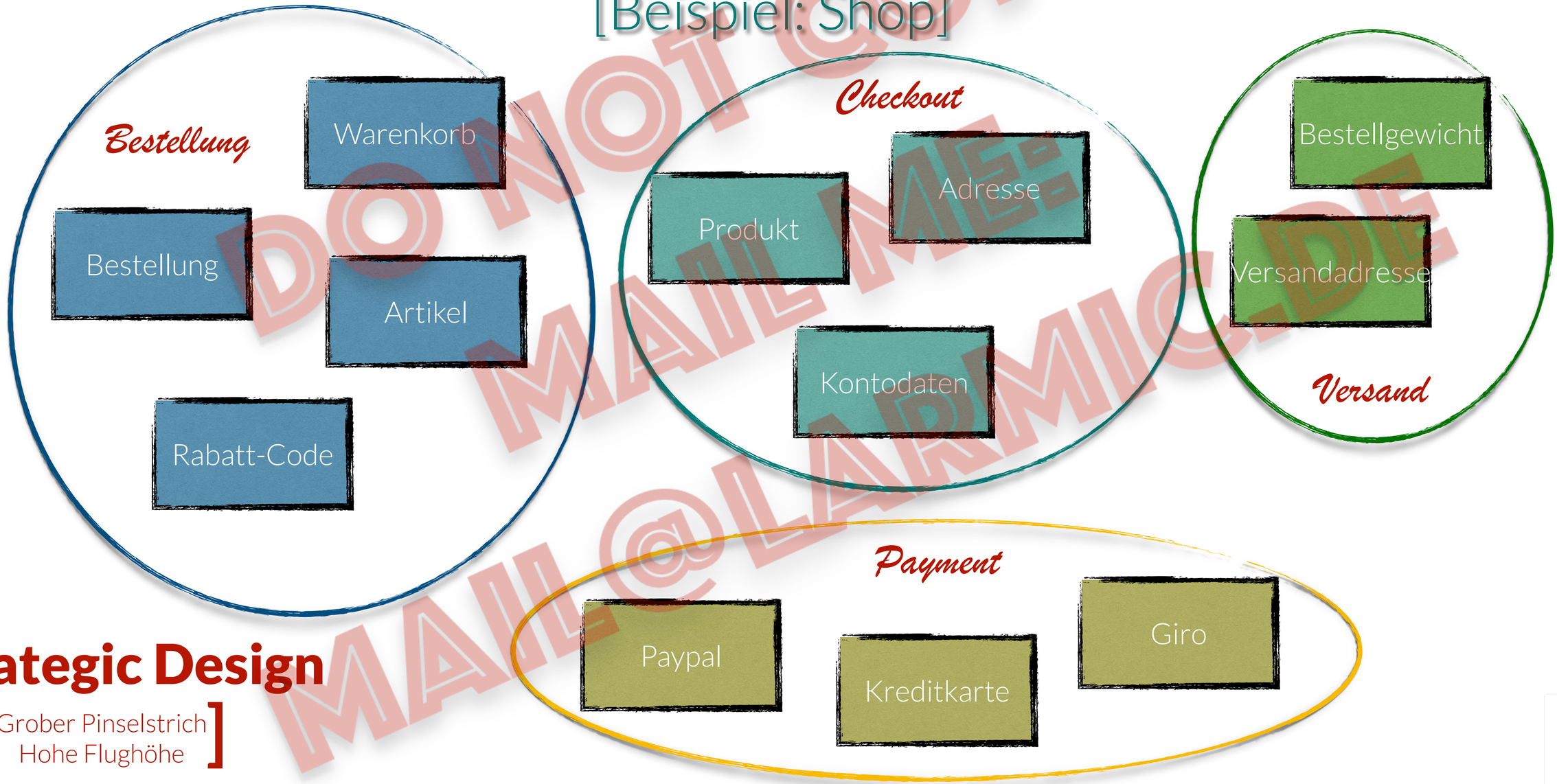
Strategic Design

[Grober Pinselstrich
Hohe Flughöhe]

DO NOT COPY
MAIL ME
MAIL@LARMICIDE

Bounded Context

[Beispiel: Shop]



Strategic Design

[Grober Pinselstrich
Hohe Flughöhe]

DO NOT COPY
MAIL ME
MAIL@LARMIC.DE

Bounded Context

[Beispiel: Shop]

- **Hohe Kopplung innerhalb**
- **Wenige Abhängigkeiten dazwischen**

Strategic Design

[Grober Pinselstrich
Hohe Flughöhe]

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

Core Domain	Supporting Subdomain	Generic Subdomain
Herzstück	Gibt keine Standardlösung	Kann man kaufen
Profit	Wichtig, aber nicht so wichtig	Open Source
Einzigartig	Kann man outsourcen	
Nur die Besten!	Kann man weniger investieren	

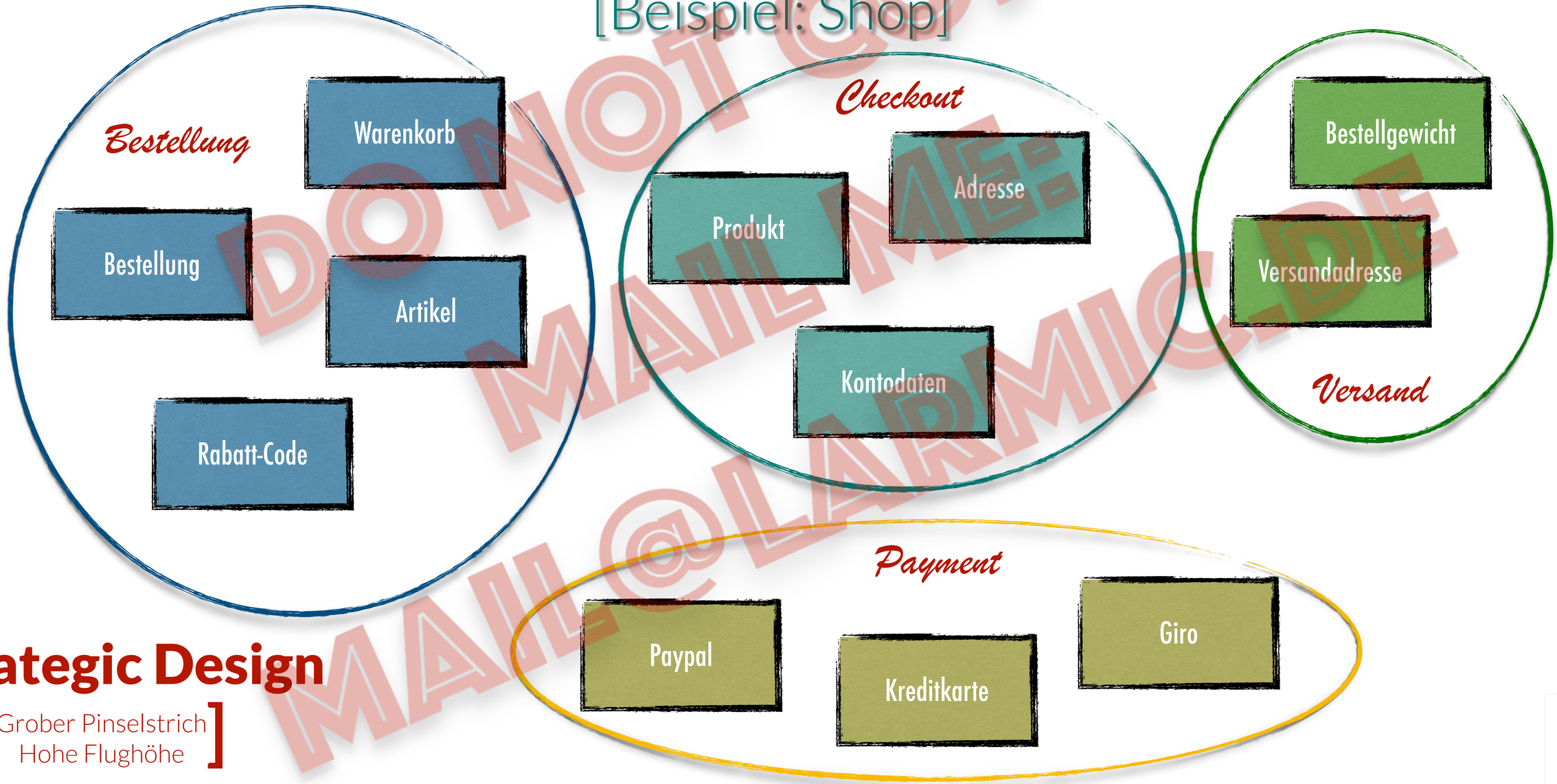
Strategic Design

[Grober Pinselstrich
Hohe Flughöhe]

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

Bounded Context

[Beispiel: Shop]



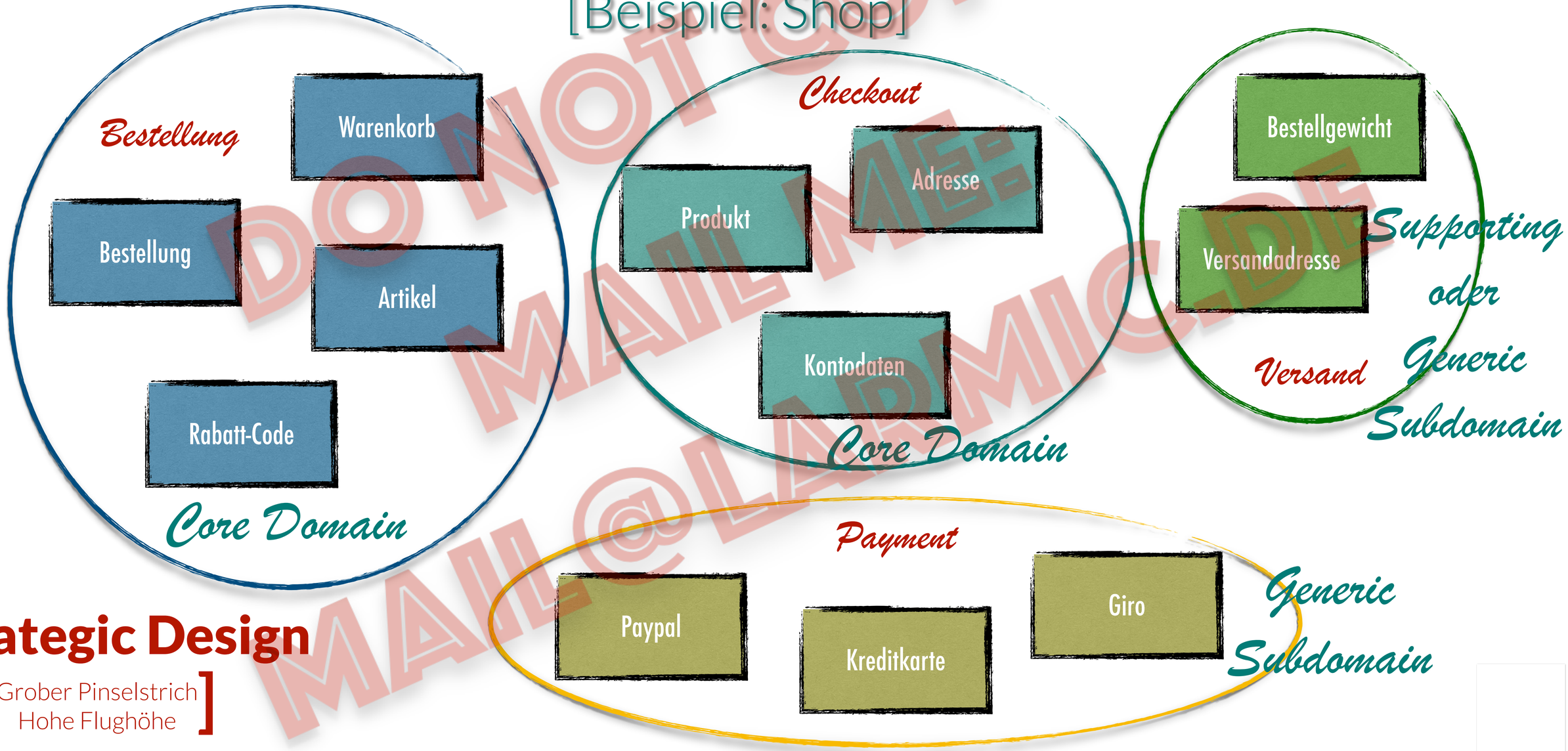
Strategic Design

[Grober Pinselstrich
Hohe Flughöhe]

DO NOT COPY
MAIL ME
MAIL@LARMICADE

Bounded Context

[Beispiel: Shop]



Strategic Design

[Grober Pinselstrich
Hohe Flughöhe]

DO NOT COPY
MAIL ME
MAIL@LARMICADE

Strategic Design

[Grober Pinselstrich
Hohe Flughöhe]

Event Storming

DO NOT COPY
MAIL-ME-
MAIL@LARMC.DE

https://www.youtube.com/watch?v=9MGaB7GhE4g

Strategic Design

[Grober Pinselstrich
Hohe Flughöhe]

Name:

V4
github.com/ddd-crew/bounded-context-canvas

Description

What benefits does this context provide, and how does it provide them?

Strategic Classification

Domain	Business Model	Evolution
- core	- revenue	- genesis
- supporting	- engagement	- custom built
- generic	- compliance	- product
- other?	- cost reduction	- commodity

Domain Roles

Role Types
- draft context
- execution context
- analysis context
- gateway context
- other

Inbound Communication

Collaborator | Messages

<Query> <Command> <Event>

Ubiquitous Language

Context-specific domain terminology

<Domain Term>
<definition>

Business Decisions

Key business rules, policies, and decisions

<Decision>

Outbound Communication

Messages | Collaborator

<Query> <Command> <Event>

DO NOT COPY
MAIL ME
MAIL@LARMIC.DE

Quelle: https://github.com/ddd-crew/bounded-context-canvas

Fazit

- Bounded-Context-Bestimmung ist nicht immer einfach
- Es gibt Tools (z.B. Event Storting), die dabei unterstützen können
- Es geht in erster Linie **nicht** um Daten, sondern um den Prozess

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

DDD Building Blocks

Aggregate Root

Value Object

Domain Event

Repository

Entity

Service

Factory

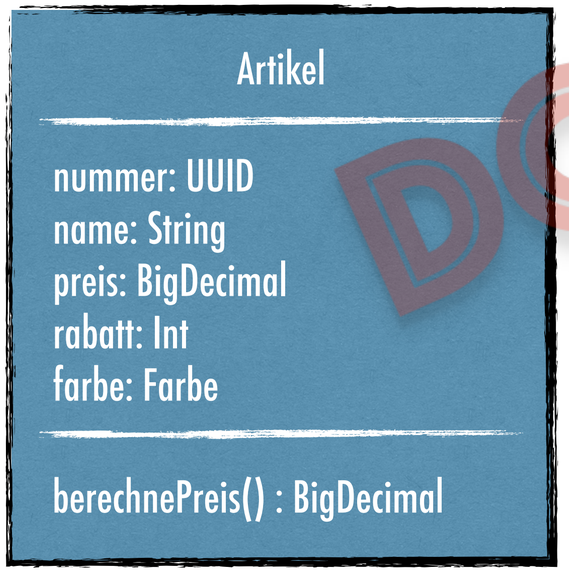
Tactical Design

[Feiner Pinselstrich
Tiefe Flughöhe]

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

DDD Building Blocks

[Entity]



- Attribute können sich ändern
- Durchläuft einen Lebenszyklus
- Ist über eine eine eindeutige ID identifizierbar
- Hat nichts mit einer Datenbank-Entity zu tun

Tactical Design

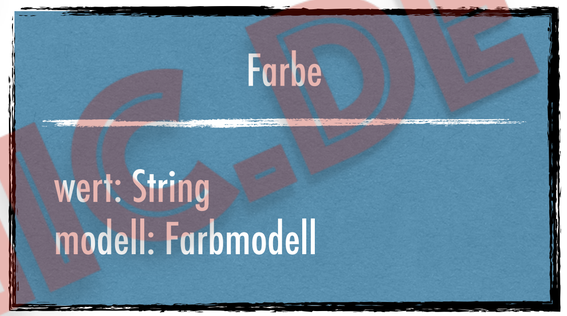
[Feiner Pinselstrich
Tiefe Flughöhe]

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

DDD Building Blocks

[Value Object]

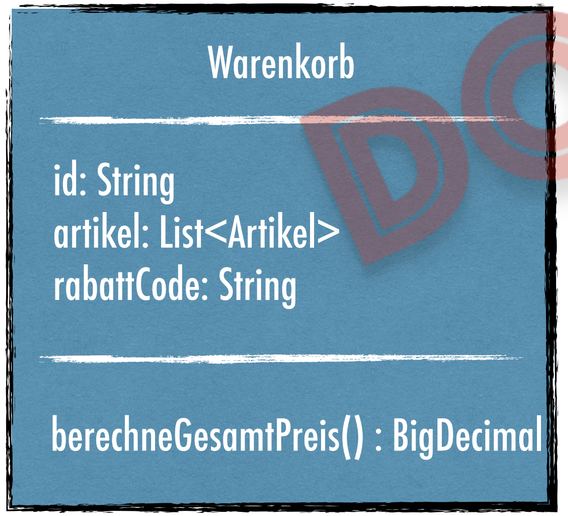
- Attribute sind nicht änderbar
- Wird durch die Attribute identifiziert



DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

DDD Building Blocks

[Aggregate Root]



- Gruppiert Entities und Value Objects
- Wird durch eine ID identifiziert
- Folgt dem „Information Hiding Principle“
- Entities werden nicht zwischen Aggregate Roots geteilt

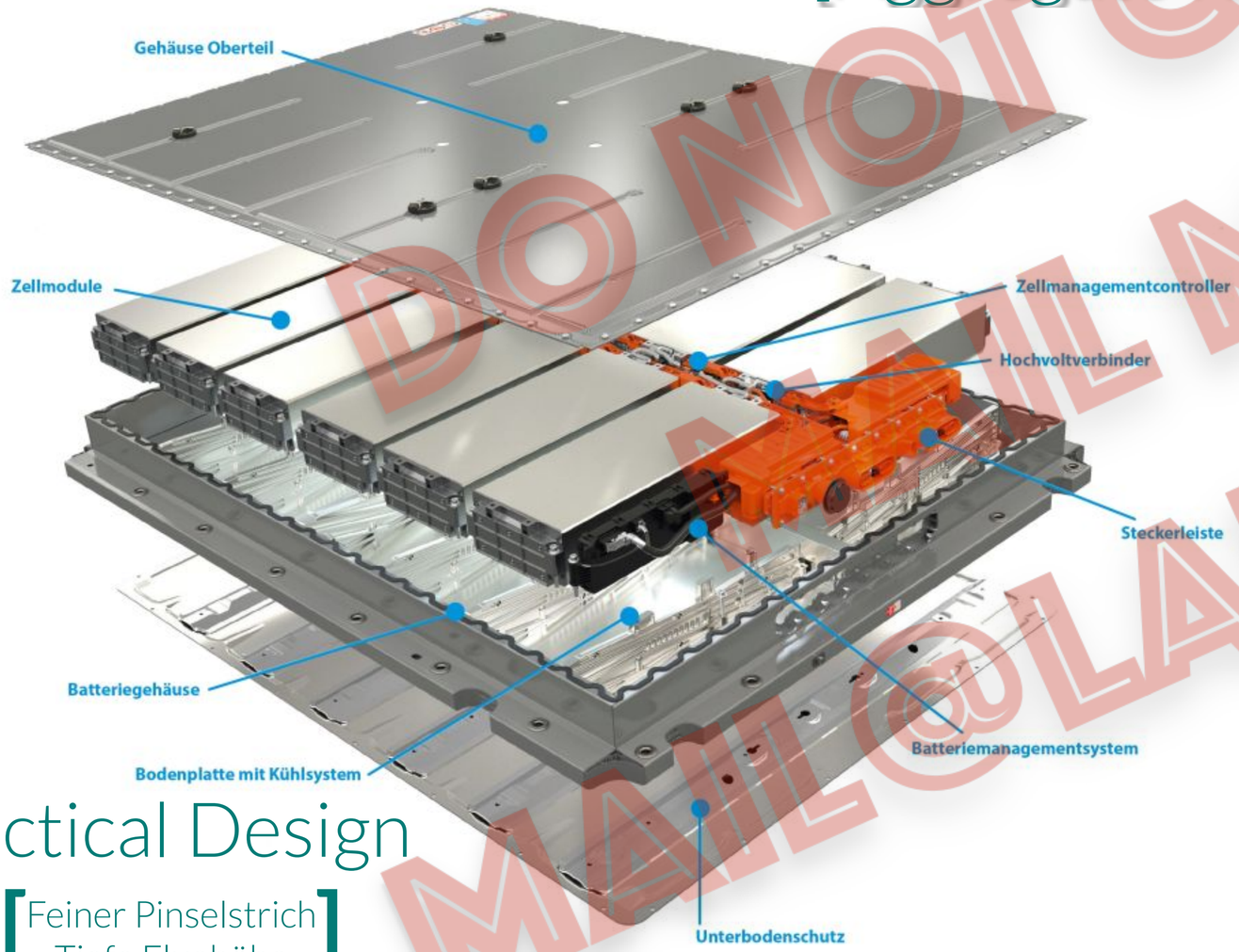
Tactical Design

[Feiner Pinselstrich
Tiefe Flughöhe]

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

DDD Building Blocks

[Aggregate Root]



- Ladestatus
- Starten des Ladevorgangs
- Wie geht es der Batterie?

Benötigen wir als Nutzer
Zugriff auf die einzelnen Zellen?

Tactical Design

[Feiner Pinselstrich
Tiefe Flughöhe]

Quelle: https://www.incoming-mobility.com/die-hochvolt-batterie-des-vw-id-3/

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

DDD Building Blocks

[Repository]

- Dient in der Regel für die Speicherung
- Nimmt nur Aggregate und Value Objects an
- Gibt nur Aggregate heraus

Tactical Design

[Feiner Pinselstrich
Tiefe Flughöhe]

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

DDD Building Blocks

[Factory]

- Dient zur Erstellung komplexer Aggregate
- Dient zur Erstellung komplexer Value Objects
- Es muss nicht unbedingt eine Factory geben!

Tactical Design

[Feiner Pinselstrich
Tiefe Flughöhe]

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

DDD Building Blocks

[Service]

- Nicht alles kann als einzelnes Aggregat modelliert werden
- Aggregat-übergreifende Prozesse

Tactical Design

[Feiner Pinselstrich
Tiefe Flughöhe]

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

DDD Building Blocks

[Domain Event]

BestellungAbgeschicktEvent

artikel: List<BestellArtikel>
bestellDatum: DateTime
preis: BigDecimal

- Häufig zur Kommunikation über Kontextgrenzen hinweg
- Kann auch in einer Domäne verwendet werden
- Event Sourcing
- Event Driven Architecture

Tactical Design

[Feiner Pinselstrich
Tiefe Flughöhe]

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

DDD Building Blocks

[Value Object oder Entity]

Tactical Design

[Feiner Pinselstrich
Tiefe Flughöhe]

DO NOT COPY
MAIL ME:
MAIL@EMIC.DE

DDD Building Blocks

[Value Object oder Entity]

Adresse

Tactical Design

[Feiner Pinselstrich
Tiefe Flughöhe]

DO NOT COPY
MICHAEL MAIC
MICHAELMAIC.DE

Tactical Design

[Feiner Pinselstrich
Tiefe Flughöhe]

UNIFIED
MODELING
LANGUAGE™

DO NOT COPY
MAIL@LARMC.DE

Tactical Design

[Feiner Pinselstrich
Tiefe Flughöhe]

Aggregate Design Canvas

1. Name

2. Description

3. State Transitions

8. Throughput

	Average	Maximum
Command handling rate		
Total number of clients		
Concurrency conflict chance		

4. Enforced Invariants

5. Corrective Policies

9. Size

	Average	Maximum
Event growth rate		
Lifetime of a single instance		
Number of events persisted		

6. Handled Commands

7. Created Events

Aggregate Design Canvas v1 designed by Kacper Gunia & Domain-Driven Design Crew.
For more information visit https://github.com/ddd-crew/aggregate-design-canvas.

This work is licensed under a Creative Commons Attribution 4.0 International License, CC BY 4.0.
To view a copy of the license, visit https://creativecommons.org/licenses/by/4.0/.

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

Quelle: https://github.com/ddd-crew/aggregate-design-canvas

Fazit

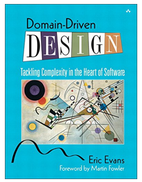
- Es gibt viele unterschiedliche DDD Begriffe (Building Blocks)
- Taktisches Design ist nicht einfach
- Building Blocks schaffen ein gemeinsames technisches Verständnis
- Fachsprache findet sich im Model wieder

Building Blocks wollen wir im Workshop nutzen und implementieren!

Tactical Design

[Feiner Pinselstrich
Tiefe Flughöhe]

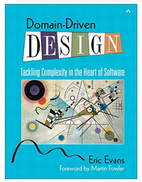
DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE



Fazit

- Fachsprache ist wesentlich
- Prozesse (und nicht Daten) stehen im Mittelpunkt
- Building Blocks sorgen für ein gemeinsames Verständnis zwischen Technikern und Nicht-Technikern
- DDD ist viel Arbeit (taktisches und strategisches Design, Anforderungsmanagement, …)

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE



Fazit

- Fachsprache ist wesentlich
- Prozesse (und nicht Daten) stehen im Mittelpunkt
- Building Blocks sorgen für ein gemeinsames Verständnis zwischen Technikern und Nicht-Technikern
- DDD ist viel Arbeit (taktisches und strategisches Design, Anforderungsmanagement, ...)

Fragen?

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

Akt 3

THEORIE HEX. ARCHITEKTUR

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

https://www.youtube.com/watch?v=-9S_rgrabTA

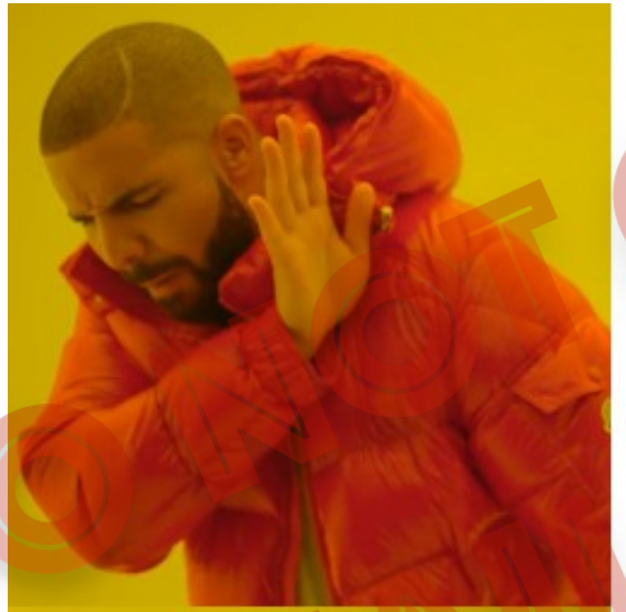
Was habt ihr für Architekturen?

DO NOT COPY
MAIL@LARMIG.DE

Wir haben Schichten

DO NOT COPY
MAIL ME:
MAN@LARMIG.DE

DO NOT COPY
MAIL ME:
MATT@LARMMC.DE

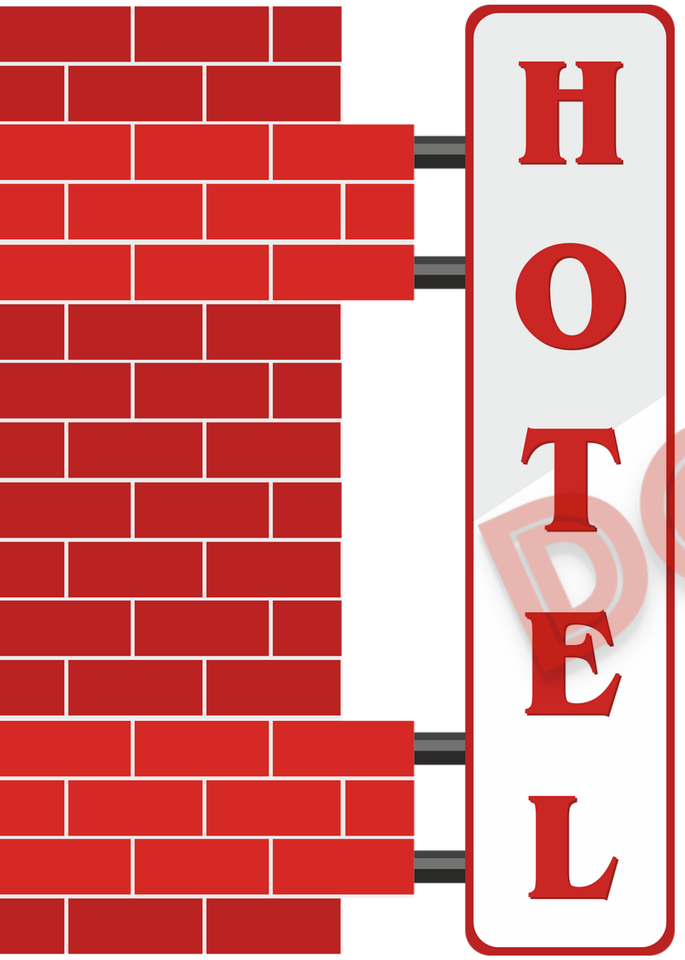


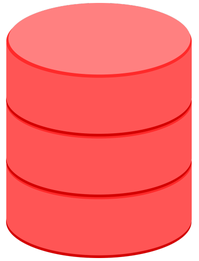
Du musst
die
Architektur
nicht verstehen

Du must
die Architektur
nur
umsetzen können

DO NOT COPY
MAIL ME:
MICHAEL.ARMIC.DE

Schichtmodell 1





Erzeuge DB-Entities und

Presentation

Controller

Model

DB-Entity

Business Logic

Mapper

Service

Manager

Helper

DB-Entity

Data Access

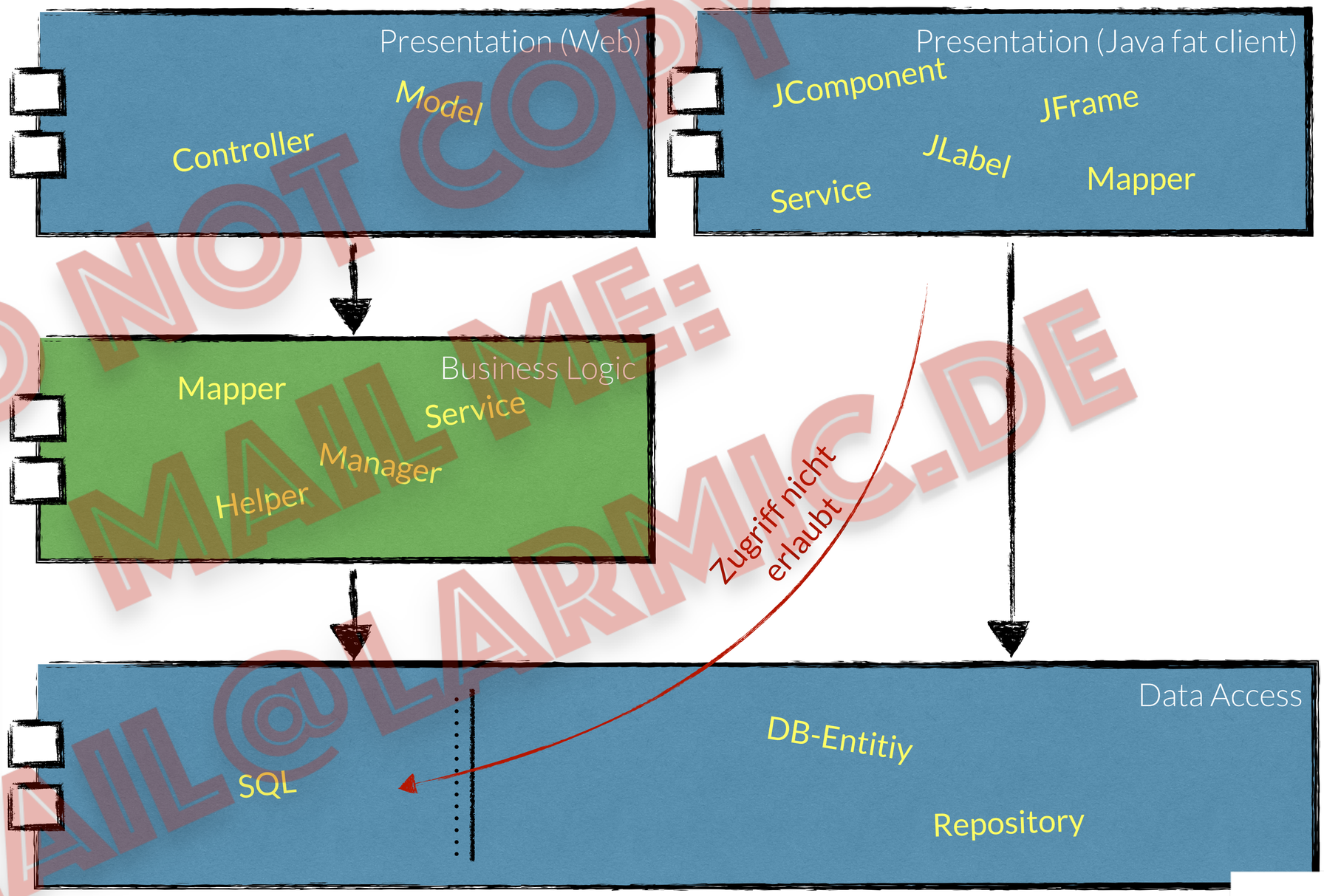
SQL

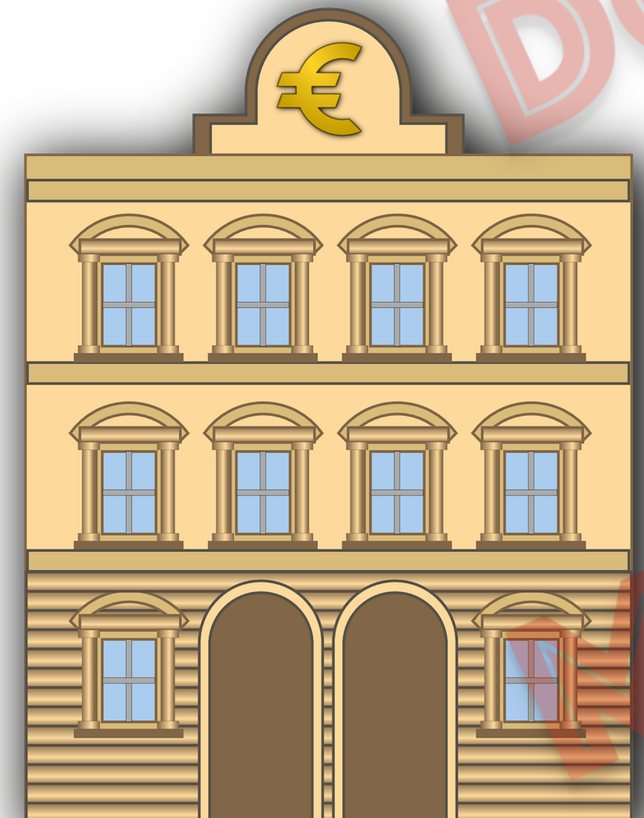
DB-Entitiy

DAO

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

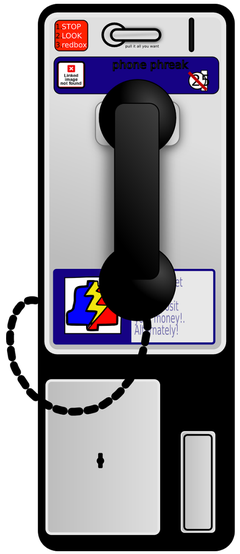
Schicht-modell 2

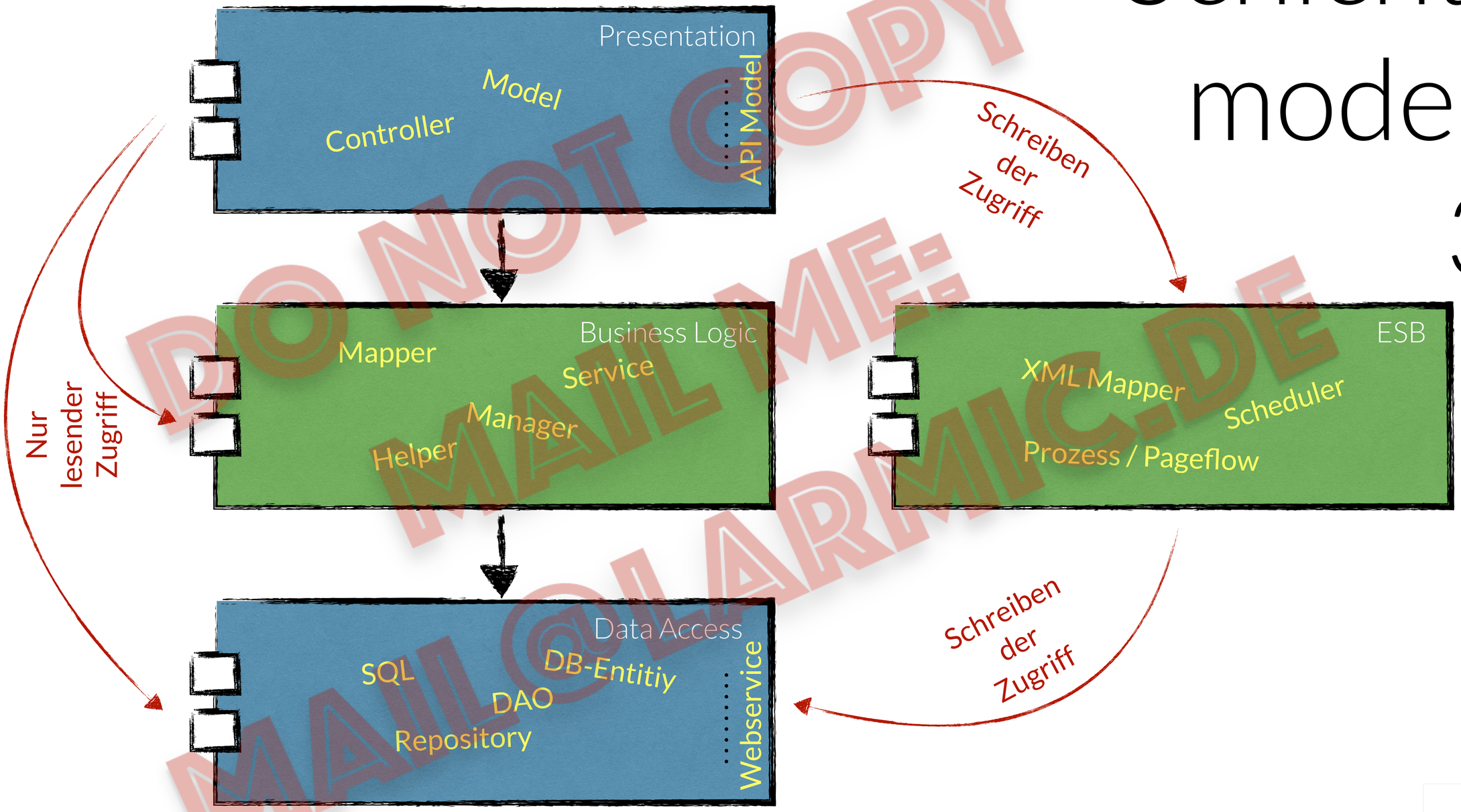




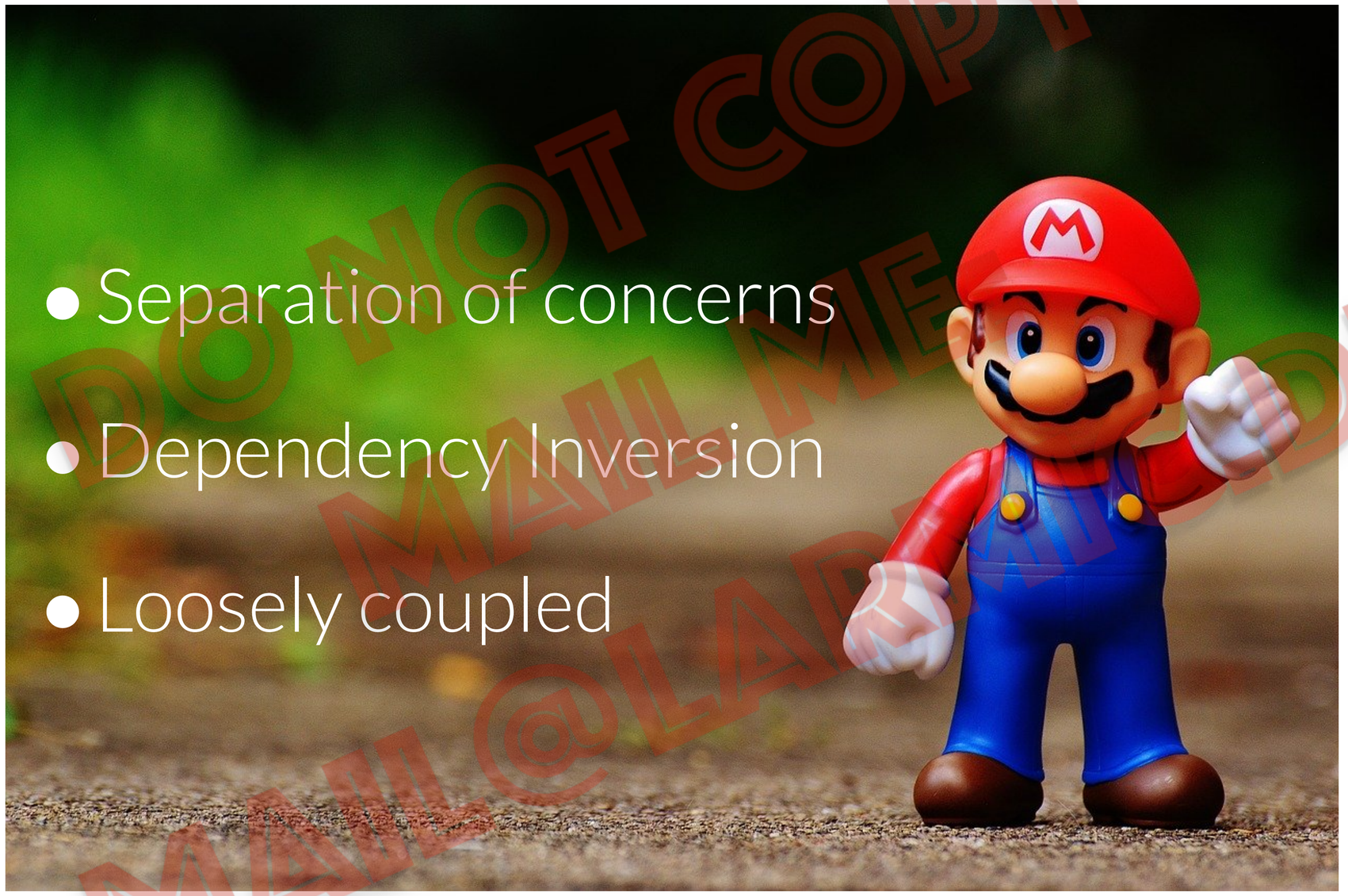
DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

Schicht-modell 3

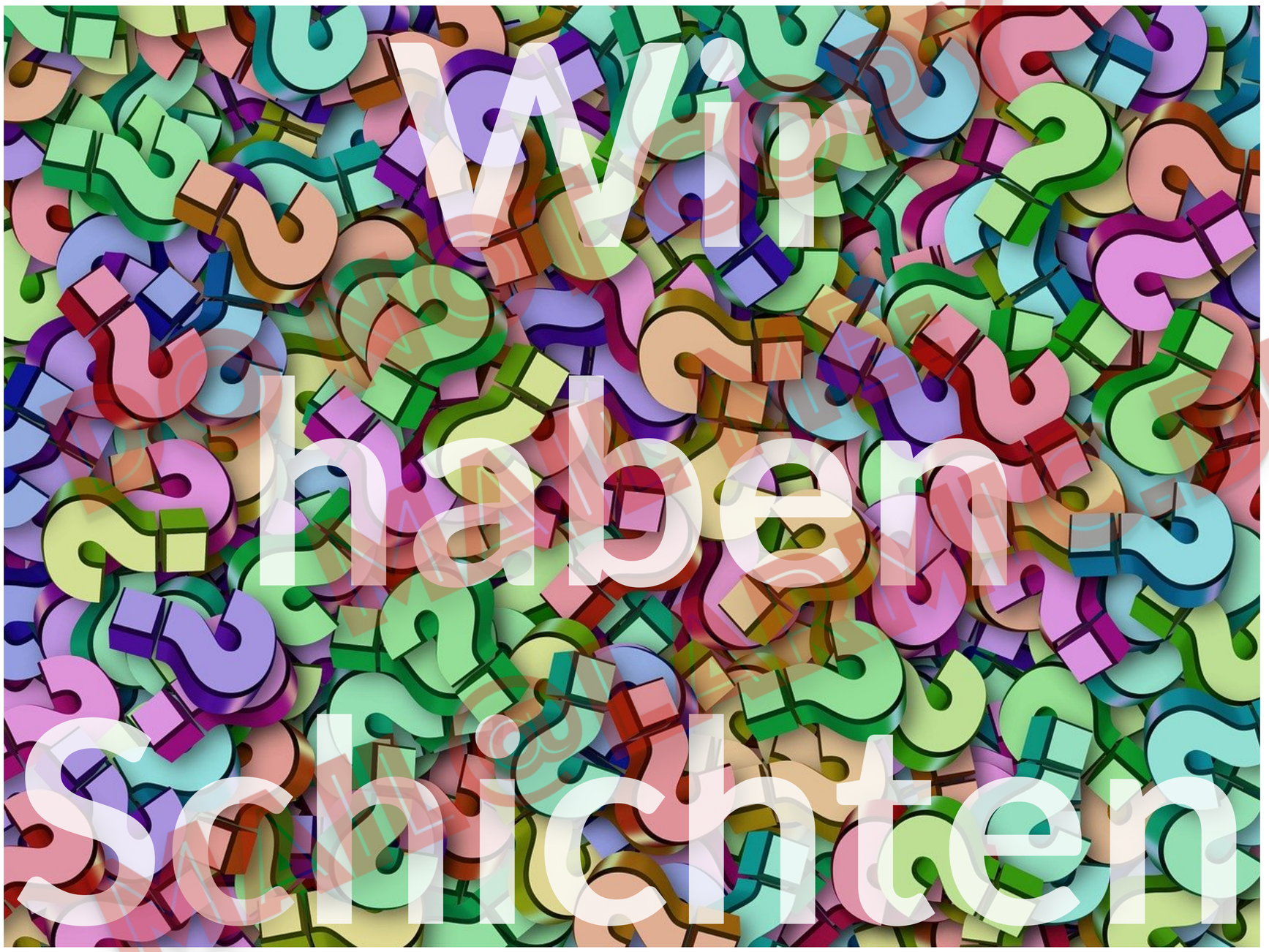


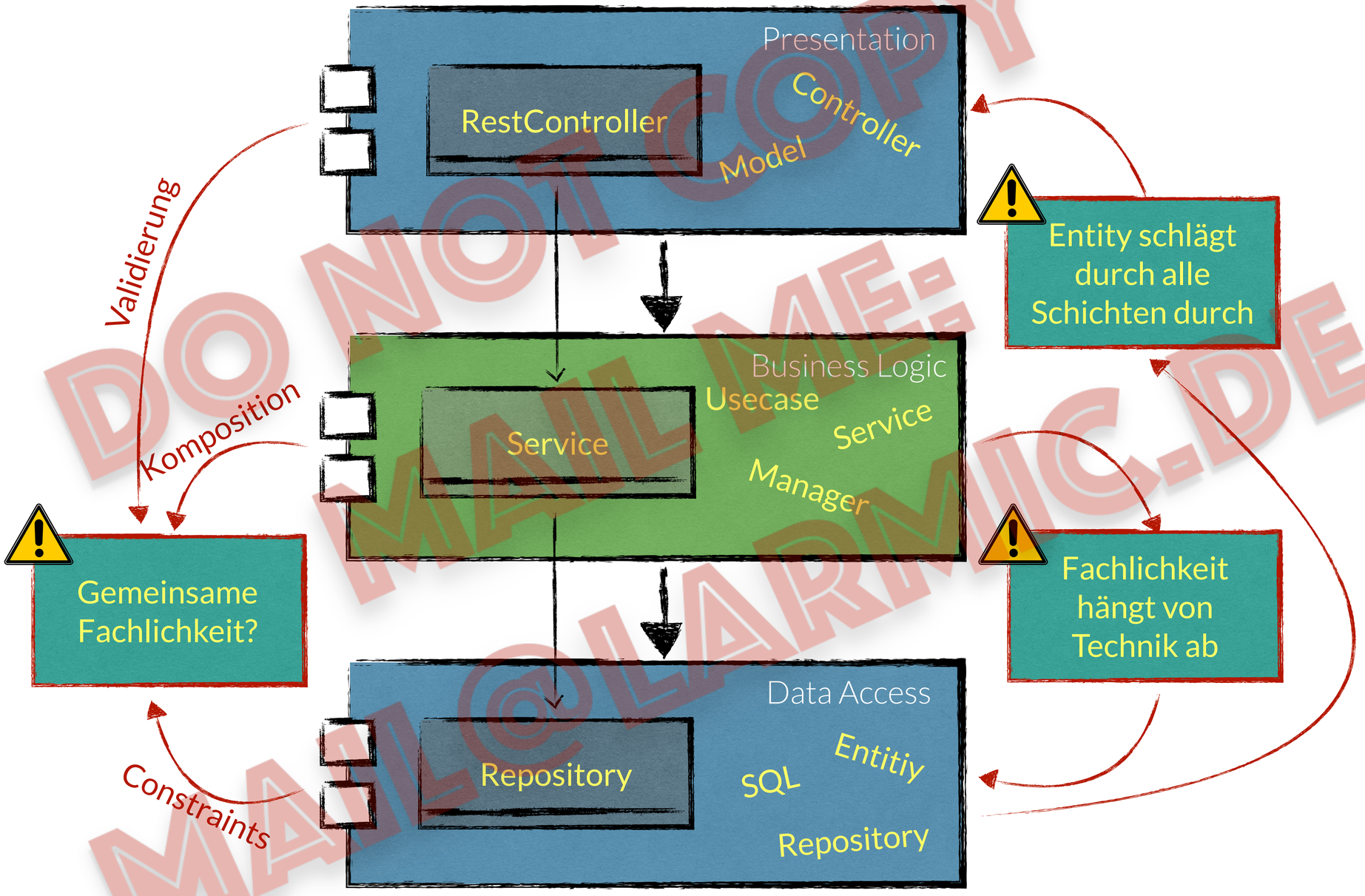


DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE



DO NOT COPY
MAIL ME
MAIL@LADEMIDE

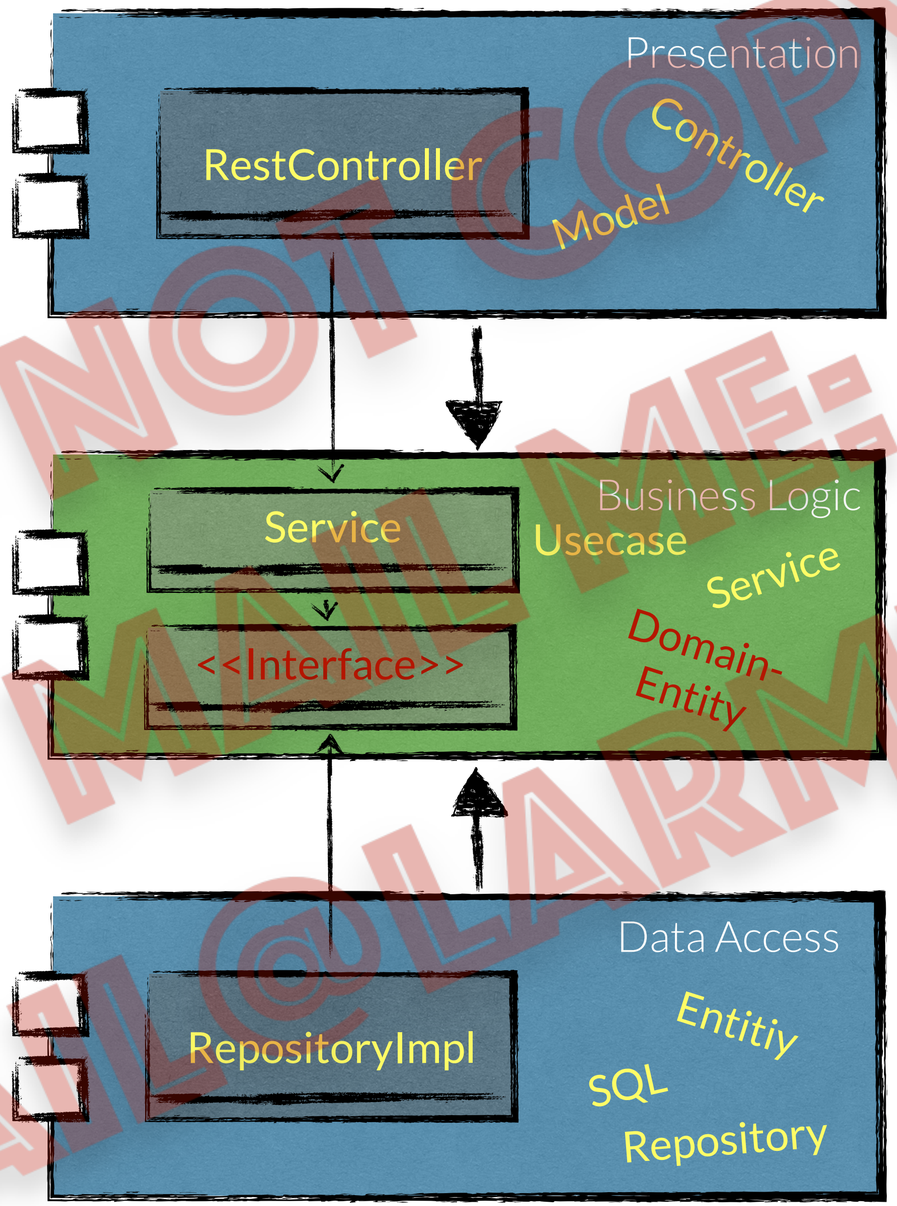

Wir
haben
Schichten



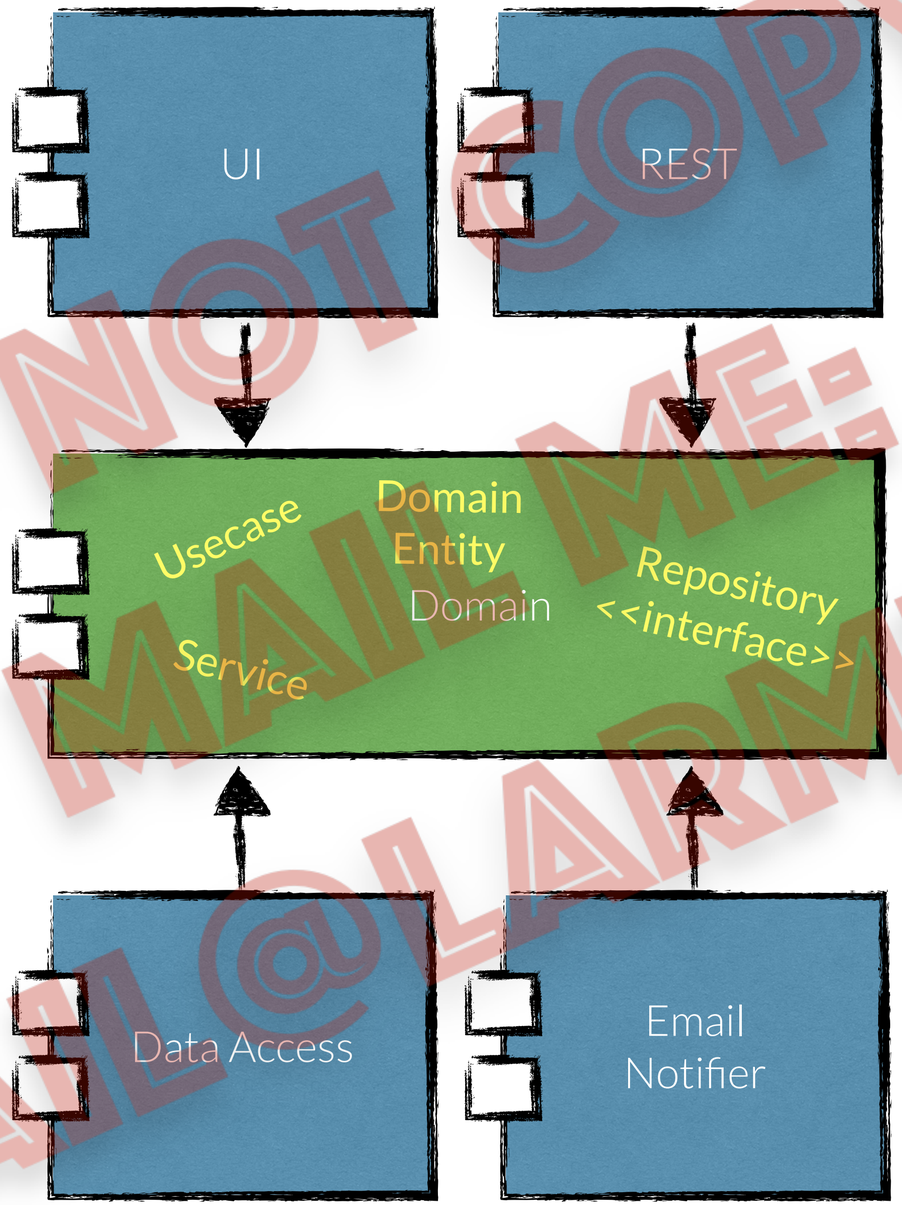
DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

How to Fix?

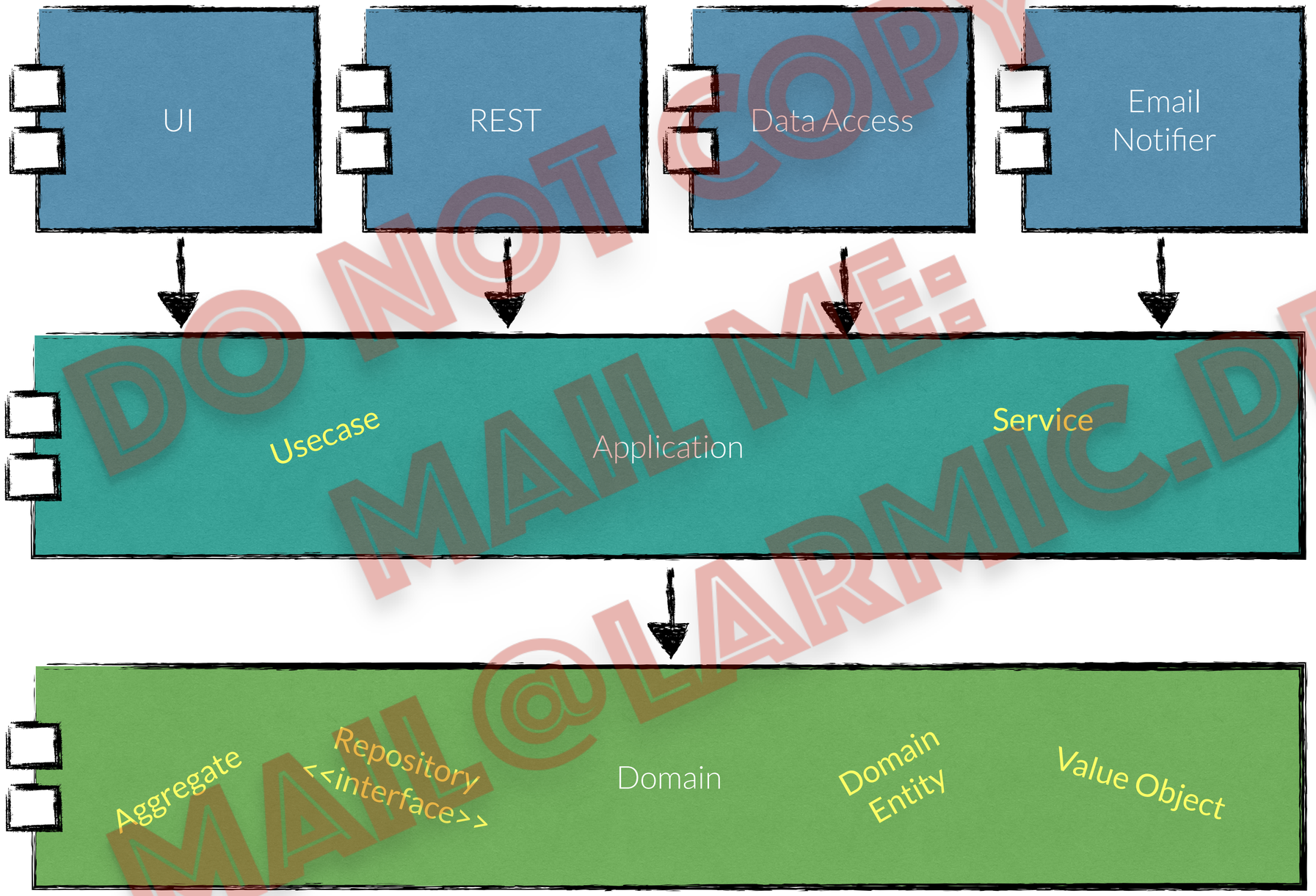
DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE



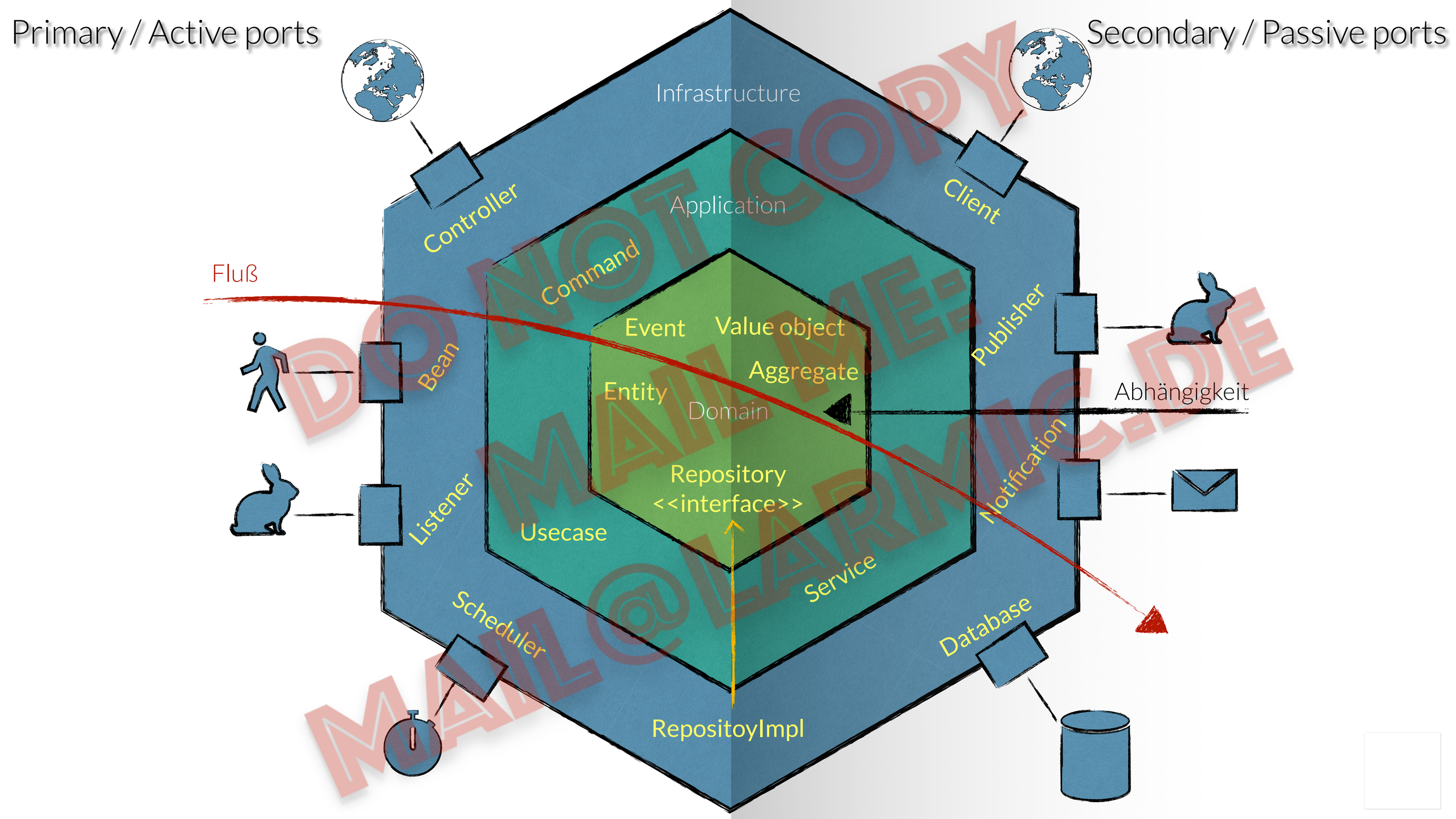
DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE



DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE



DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE



DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

Was habt ihr für Architekturen?

Wir haben Schichten!

Aber die Fachlichkeit steht im Zentrum!

DO NOT COPY
MATTHIAS MEY
MATTHIAS-MEY.DE

Was habt ihr für Architekturen?

Wir haben 3 Schichten!

Aber die Fachlichkeit steht im Zentrum!

DO NOT COPY

Akt 4

CODING CHALLENGE

VORBEREITUNG DES PROJEKTES

DO NOT COPY
MAIL ME:
MAIL@LARMIG.DE

User Story

[Vorbereitung des Projektes]

Anforderung

Erstelle eine Applikation, die unter */api/hello* ein *Hello World!* liefert. Einige Dich im Team (oder alleine) auf die passenden Tools und die Programmiersprache.

Akzeptanzkriterien

* **GIT-Repository ist erstellt**
* **Generelle Applikation ist erstellt**
* **Der Endpoint */api/hello* liefert *Hello World!***
* **Eine rudimentäre *readme.md* beschreibt, wie das Projekt gebaut und gestartet werden kann.**

Zeit

45 min

Empfehlung

Kein TDD, keine Tests

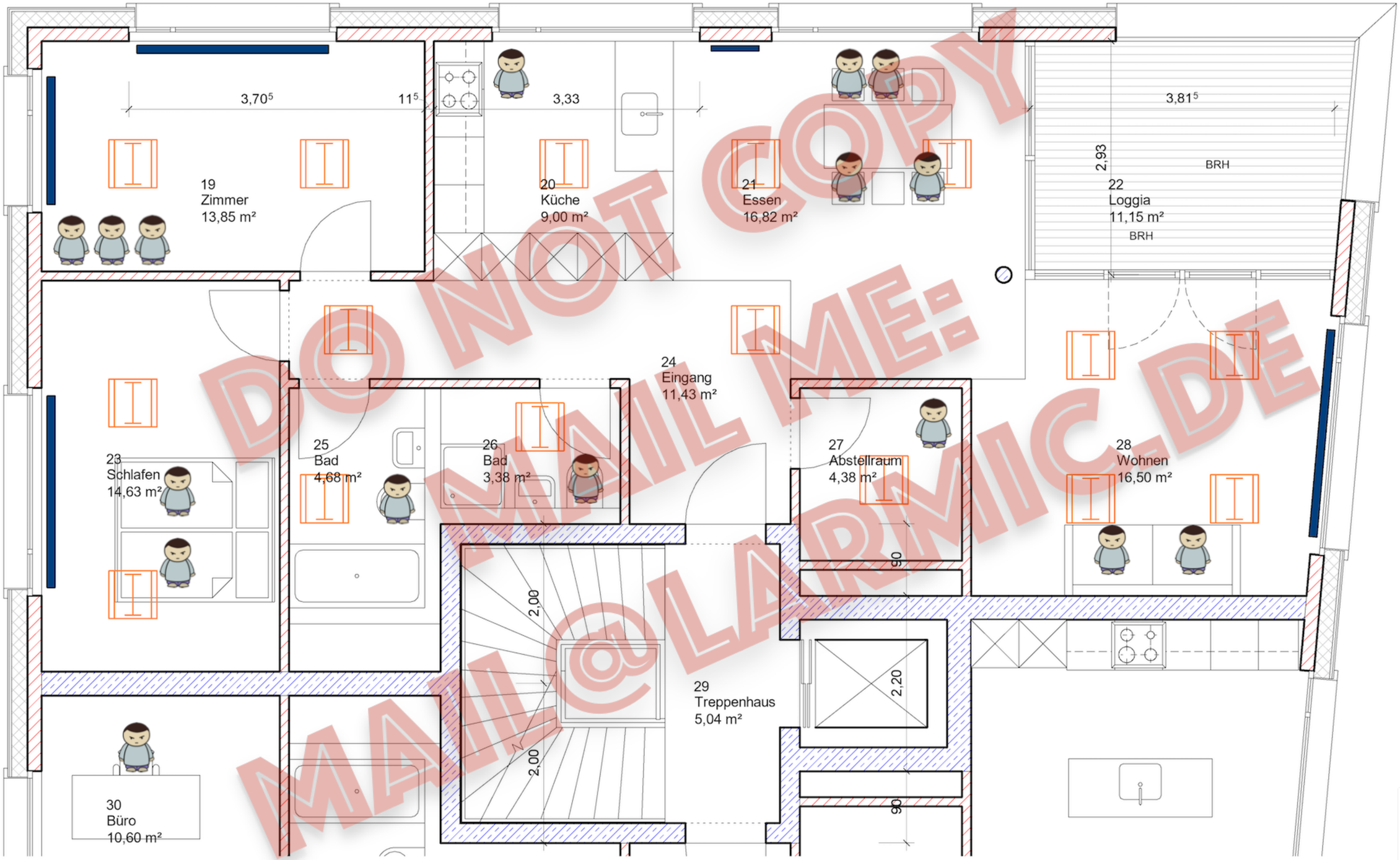
DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

Akt 5

CODING CHALLENGE

RÄUME UND PERSONEN

DO NOT COPY
MAIL ME:
MAIL@LARMIG.DE



DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

User Story

[Erfassen von Räumen]

Anforderung

Erweitere die Applikation um das Erfassen von Räumen.
Ein Raum enthält immer eine vierstellige, eindeutige Raumnummer und einen Namen.

Akzeptanzkriterien

* **Mit** *POST /api/room* **kann ein Raum angelegt werden**
* **Mit** *GET /api/room/{ID}* **kann ein Raum abgefragt werden.**
* **Es wird geprüft, dass die Raumnummer eine Zahl und 4-stellig ist. Vorangestellte Nullen sind dabei erlaubt.**
* **Es wird geprüft, dass die Raumnummer eindeutig ist.**

Zeit

60 min

Vereinfachung

Im Fehlerfall wird
ein Status 400 geworfen,
auch wenn manchmal z.B. ein
404 sinnvoll wäre.

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

User Story

[Hinzufügen von Personen]

Anforderung

Erweitere die Applikation um das Hinzufügen von Personen.
Eine Person hat einen Vor- und Nachnamen, sowie einen LDAP-Benutzernamen.
Optional sind die Namenszusätze *von*, *van* **und** *de*.

Akzeptanzkriterien

* **Mit** *PUT /api/room/{ID}/person* **kann eine Person zu einem Raum hinzugefügt werden**
* **Mit** *GET /api/room/{ID}* **kann ein Raum abgefragt werden. Die Antwort enthält auch die entsprechenden Personen (in Kurzschreibweise*).**
* **Es wird geprüft, dass der optionale Namenszusatz nur** *von*, *van* **oder** *de* **ist.**
* **Es wird geprüft, dass der LDAP-Benutzername eindeutig ist.**
* **Es wird geprüft, dass eine Person nur einem Raum zugeordnet ist.**

Zeit

60 min

*** Vorname [Namenszusatz] Nachname (LDAP-Benutzername)**

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

User Story

[Personen in mehreren Räume]

Anforderung

Erweitere die Applikation um das Hinzufügen von Personen in mehreren Räumen.

Akzeptanzkriterien

* Eine Person kann mehreren Räumen zugewiesen werden.

Zeit

75 min

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

User Story

[Domain-Events]

Anforderung

Erweitere die Applikation das Versenden von Events, wenn eine Person einem Raum hinzugefügt wird.

Akzeptanzkriterien

* **Bei einer Raumzuweisung wird ein** *PersonWurdeRaumZugeordnetEvent* **geworfen**

Zeit

30 min

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE

Best Practices

- Starte immer mit den **Anforderungen**
- Starte beim Modellieren immer mit den **Aggregaten**, **Entities** und **Value Objects**
- Nutze **kleine Aggregate** und so flach wie möglich
- Natürliche Schlüssel in Aggregaten sind selten sinnvoll
- Nutze zunächst immer nur die **Building Blocks**
- Verhindere (softwareseitig) die Nutzung von Technik im Domain- und Application-Layer
- Stelle die „Aufruf-Richtung“ (softwareseitig) sicher
- Mappen ist nicht böse
- Bei Massendaten bedingt einsetzbar

DO NOT COPY
MAIL ME:
MAIL@LARMIC.DE