Hexagon Architecture
In the real world

DDD Cologne / Mar 2, 2020

Christoph Baudson / @sustainablepace

REWE digital

Christoph Baudson

- **Software dev** at **REWE Digital**
- Currently **Fulfillment**
- **Organizer** of several meetups
- @sustainablepace
- sustainablepace.net

DOMAIN DRIVEN DESIGN

Bonn Agile Meetup

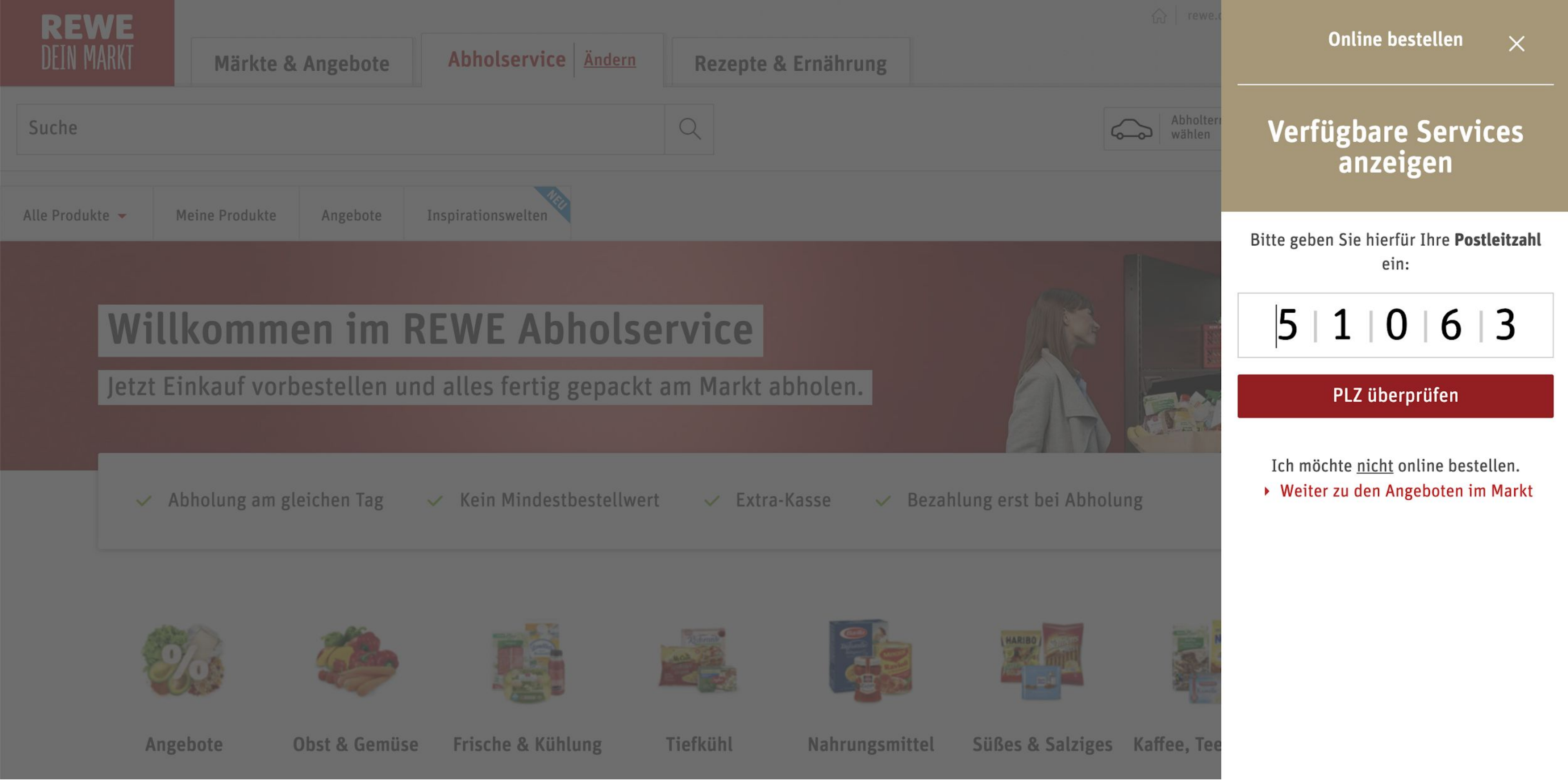
Hexagon Architecture in the real world

1. **Domain “Pickup”**
2. Prototype Pickup App
3. Problems
4. Enter Hexagon Architecture
5. Results

Pickup domain:

Customers **order online** and

pick up their order in a **local store**



Willkommen im REWE Abholservice

Jetzt Einkauf vorbestellen und alles fertig gepackt am Markt abholen.

✓ Abholung am gleichen Tag ✓ Kein Mindestbestellwert ✓ Extra-Kasse ✓ Bezahlung erst bei Abholung

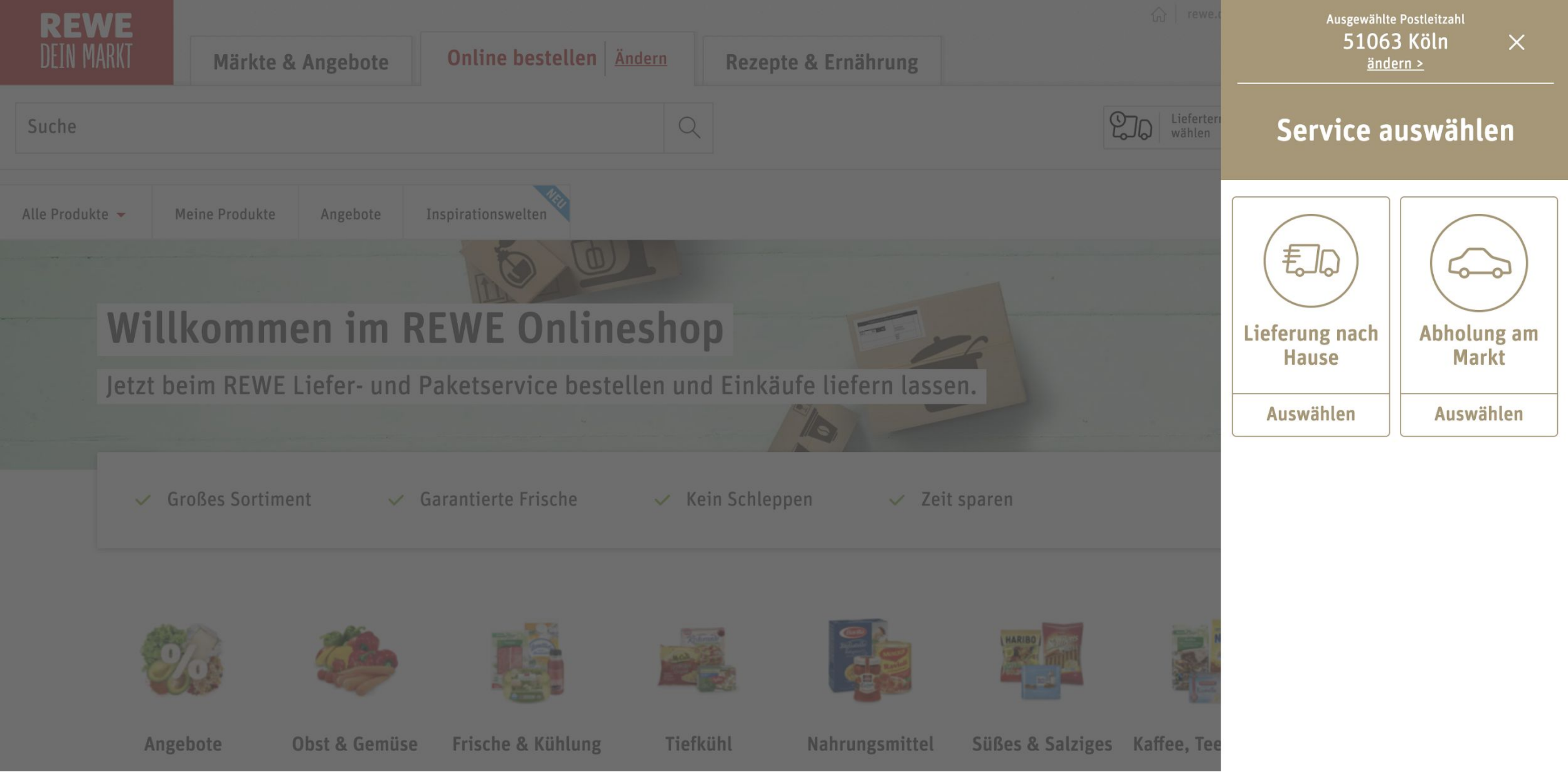
Verfügbare Services anzeigen

Bitte geben Sie hierfür Ihre **Postleitzahl** ein:

5 | 1 | 0 | 6 | 3

Ich möchte nicht online bestellen.

Weiter zu den Angeboten im Markt

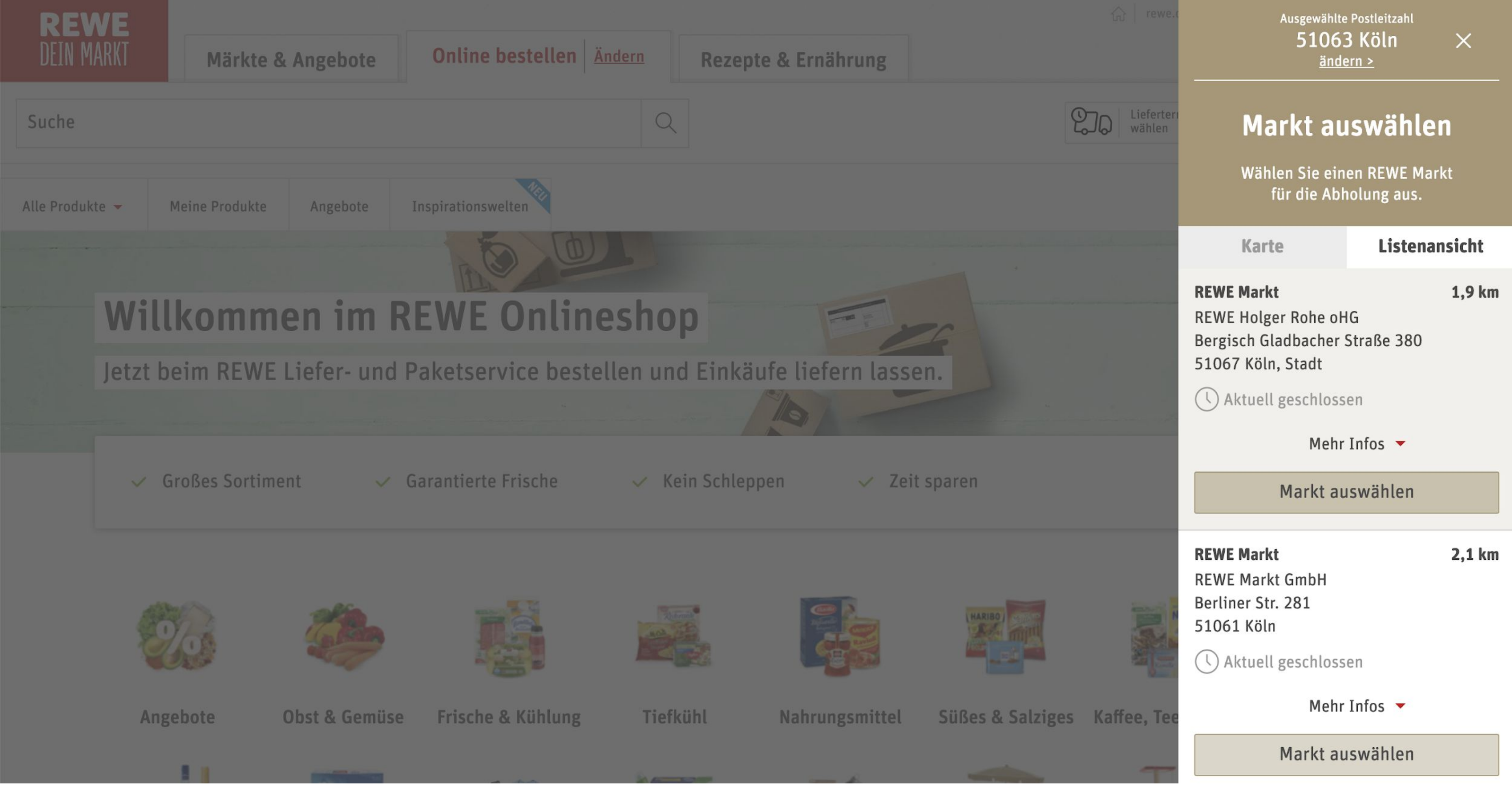


Willkommen im REWE Onlineshop

Jetzt beim REWE Liefer- und Paketservice bestellen und Einkäufe liefern lassen.

Ausgewählte Postleitzahl
51063 Köln
ändern >

Service auswählen



Willkommen im REWE Onlineshop

Jetzt beim REWE Liefer- und Paketservice bestellen und Einkäufe liefern lassen.

✓ Großes Sortiment ✓ Garantierte Frische ✓ Kein Schleppen ✓ Zeit sparen

Ausgewählte Postleitzahl
51063 Köln
ändern >

Markt auswählen

Wählen Sie einen REWE Markt für die Abholung aus.

Karte | Listenansicht

REWE Markt **1,9 km**
REWE Holger Rohe oHG
Bergisch Gladbacher Straße 380
51067 Köln, Stadt

Aktuell geschlossen

Mehr Infos

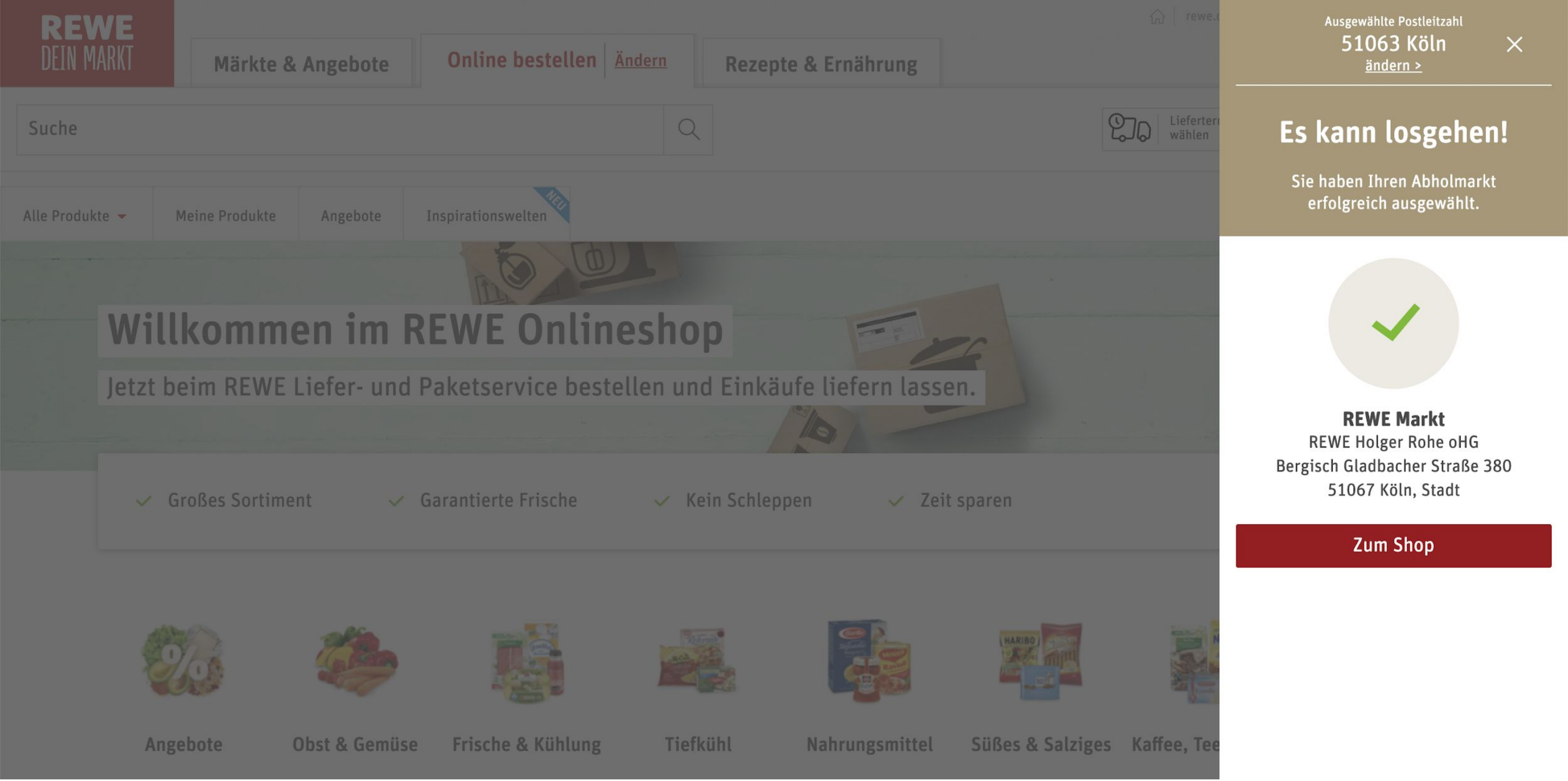
Markt auswählen

REWE Markt **2,1 km**
REWE Markt GmbH
Berliner Str. 281
51061 Köln

Aktuell geschlossen

Mehr Infos

Markt auswählen



Willkommen im REWE Onlineshop

Jetzt beim REWE Liefer- und Paketservice bestellen und Einkäufe liefern lassen.

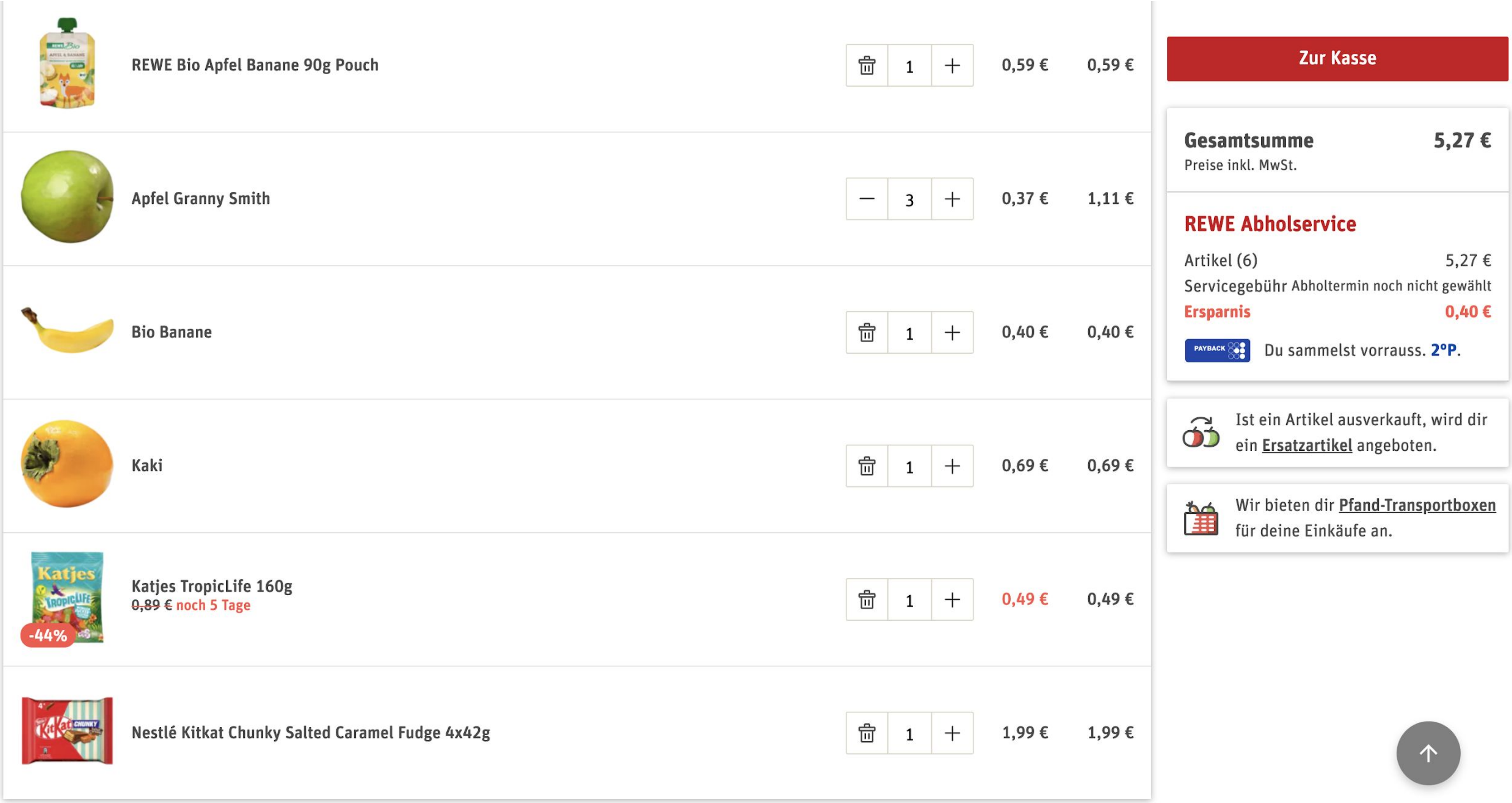
✓ Großes Sortiment ✓ Garantierte Frische ✓ Kein Schleppen ✓ Zeit sparen

Ausgewählte Postleitzahl
51063 Köln

Es kann losgehen!

Sie haben Ihren Abholmarkt erfolgreich ausgewählt.

REWE Markt
REWE Holger Rohe oHG
Bergisch Gladbacher Straße 380
51067 Köln, Stadt



	REWE Bio Apfel Banane 90g Pouch	1	0,59 €	0,59 €
	Apfel Granny Smith	3	0,37 €	1,11 €
	Bio Banane	1	0,40 €	0,40 €
	Kaki	1	0,69 €	0,69 €
-44%	Katjes TropicLife 160g ~~0,89 €~~ noch 5 Tage	1	0,49 €	0,49 €
	Nestlé Kitkat Chunky Salted Caramel Fudge 4x42g	1	1,99 €	1,99 €

Gesamtsumme **5,27 €**
Preise inkl. MwSt.

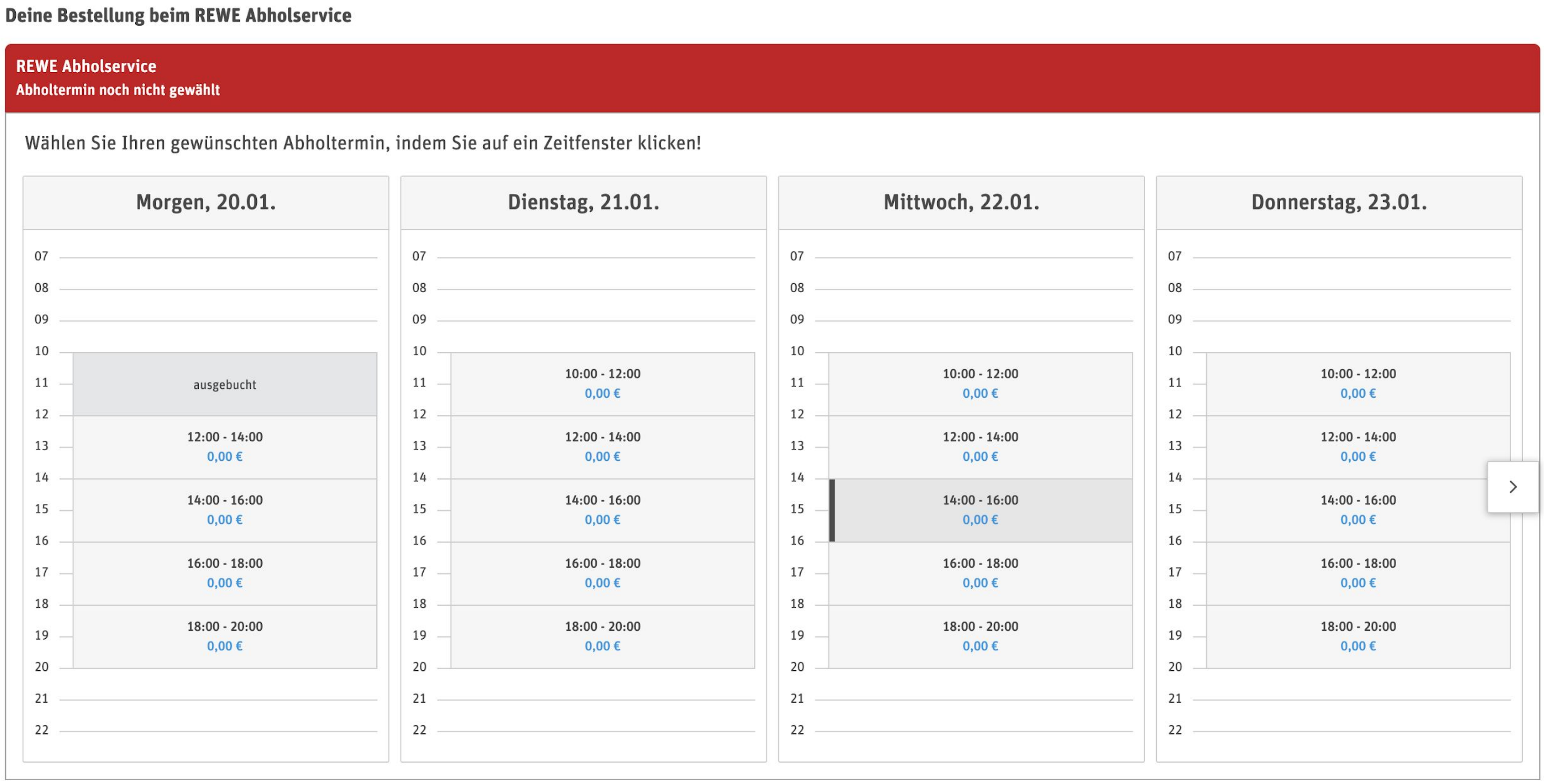
REWE Abholservice

Artikel (6)	5,27 €
Servicegebühr Abholtermin noch nicht gewählt	
Ersparnis	0,40 €

PAYBACK Du sammelst vorrauss. **2°P**.

Ist ein Artikel ausverkauft, wird dir ein Ersatzartikel angeboten.

Wir bieten dir Pfand-Transportboxen für deine Einkäufe an.



Deine Bestellung beim REWE Abholservice

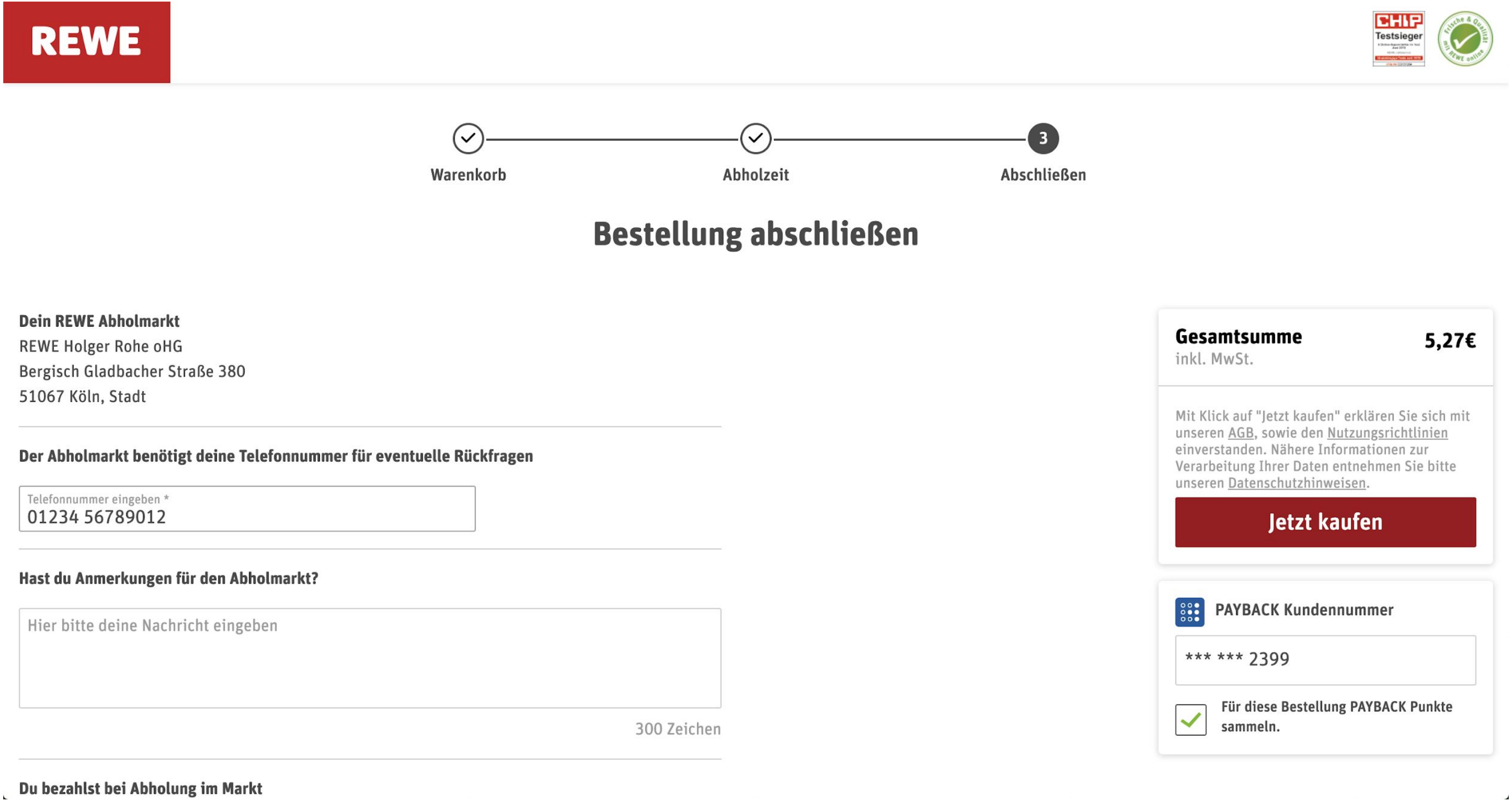
REWE Abholservice
Abholtermin noch nicht gewählt

Wählen Sie Ihren gewünschten Abholtermin, indem Sie auf ein Zeitfenster klicken!

07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22

Morgen, 20.01.	Dienstag, 21.01.	Mittwoch, 22.01.	Donnerstag, 23.01.
ausgebucht	10:00 - 12:00 0,00 €	10:00 - 12:00 0,00 €	10:00 - 12:00 0,00 €
12:00 - 14:00 0,00 €	12:00 - 14:00 0,00 €	12:00 - 14:00 0,00 €	12:00 - 14:00 0,00 €
14:00 - 16:00 0,00 €	14:00 - 16:00 0,00 €	14:00 - 16:00 0,00 €	14:00 - 16:00 0,00 €
16:00 - 18:00 0,00 €	16:00 - 18:00 0,00 €	16:00 - 18:00 0,00 €	16:00 - 18:00 0,00 €
18:00 - 20:00 0,00 €	18:00 - 20:00 0,00 €	18:00 - 20:00 0,00 €	18:00 - 20:00 0,00 €

REWE digital



Bestellung abschließen

Dein REWE Abholmarkt
REWE Holger Rohe oHG
Bergisch Gladbacher Straße 380
51067 Köln, Stadt

Der Abholmarkt benötigt deine Telefonnummer für eventuelle Rückfragen

Telefonnummer eingeben *
01234 56789012

Hast du Anmerkungen für den Abholmarkt?

Hier bitte deine Nachricht eingeben

300 Zeichen

Du bezahlst bei Abholung im Markt

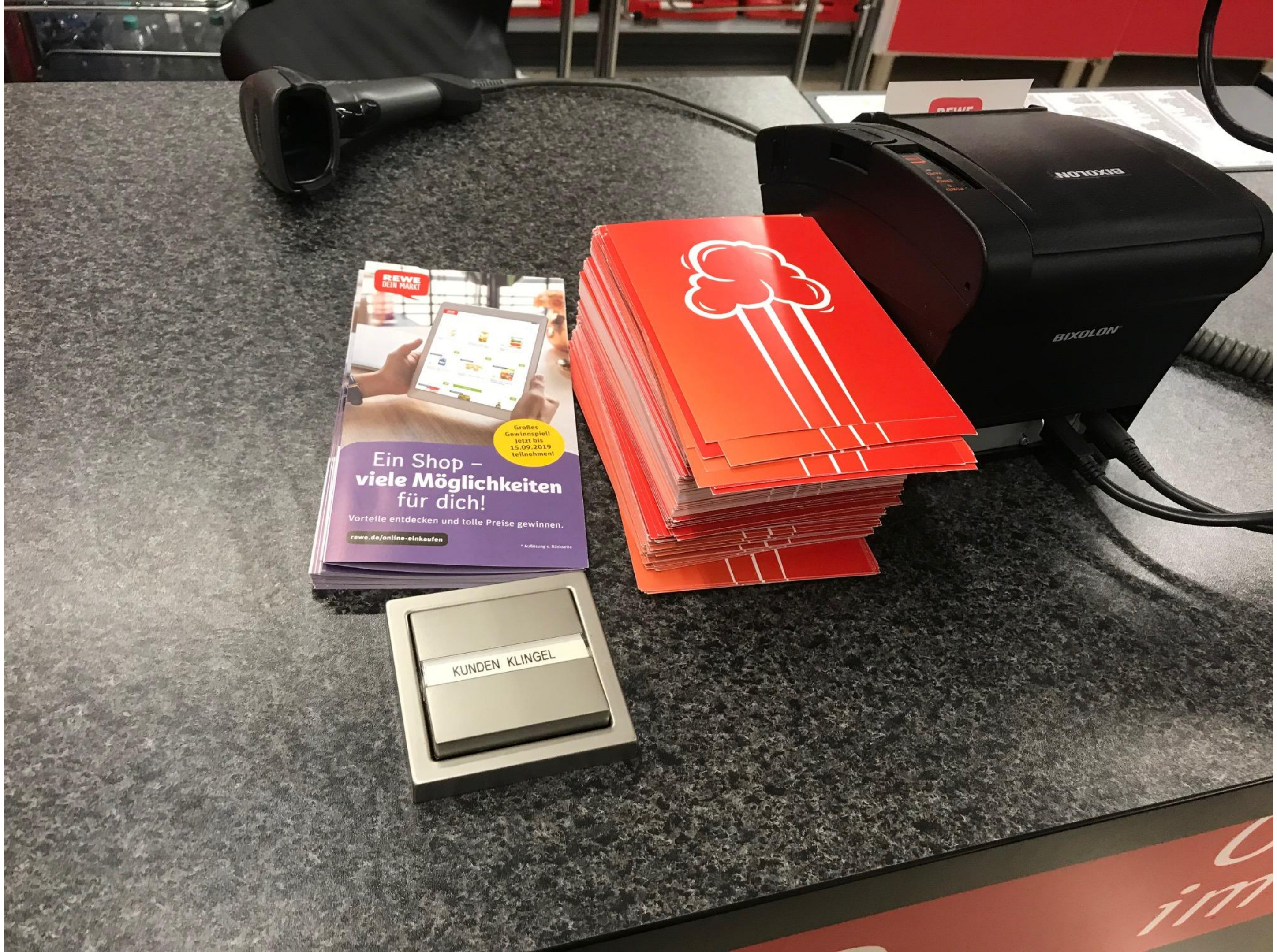
Gesamtsumme **5,27€**
inkl. MwSt.

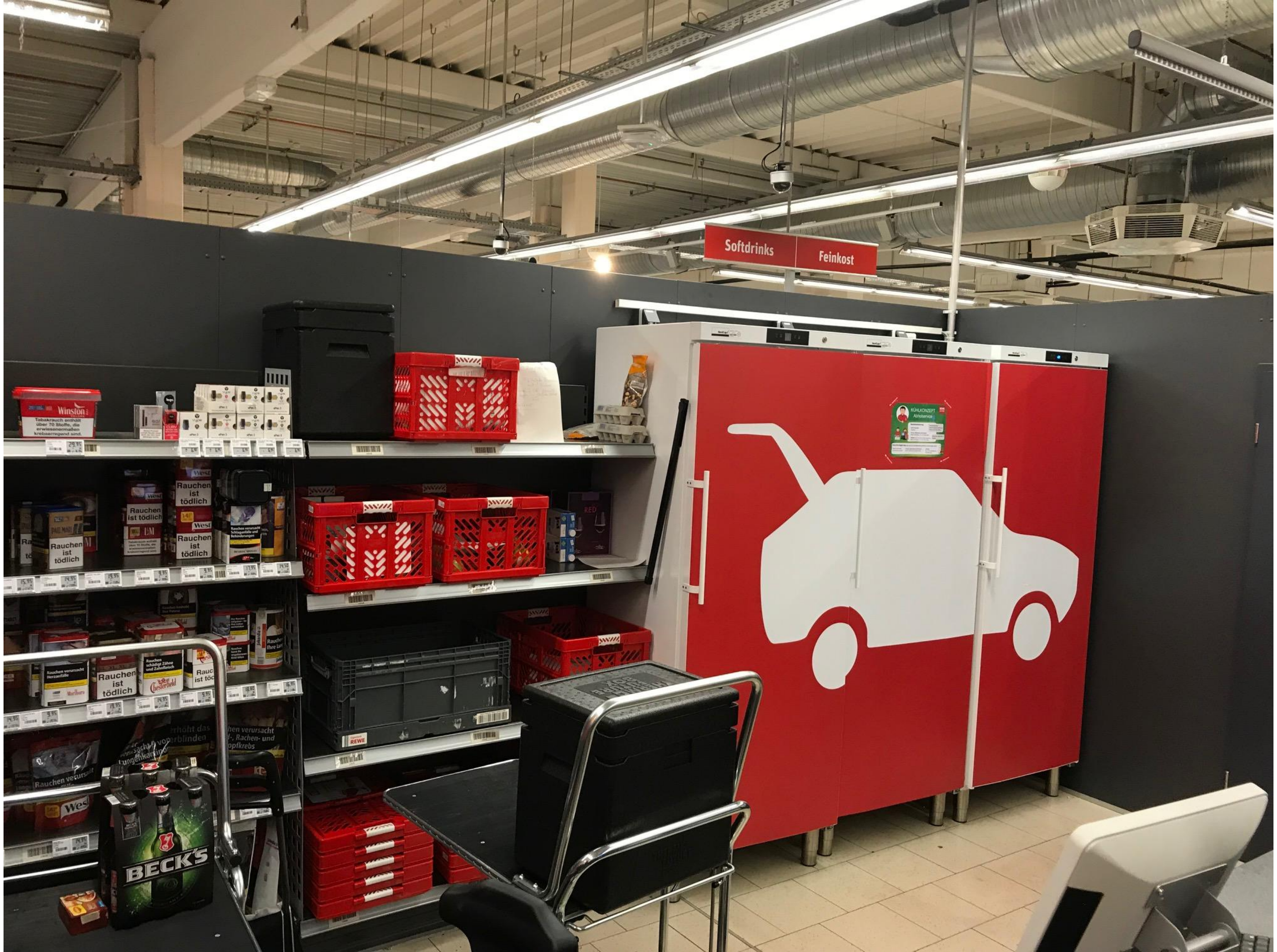
Mit Klick auf "Jetzt kaufen" erklären Sie sich mit unseren AGB, sowie den Nutzungsrichtlinien einverstanden. Nähere Informationen zur Verarbeitung Ihrer Daten entnehmen Sie bitte unseren Datenschutzhinweisen.

PAYBACK Kundennummer

*** *** 2399

☑ **Für diese Bestellung PAYBACK Punkte sammeln.**

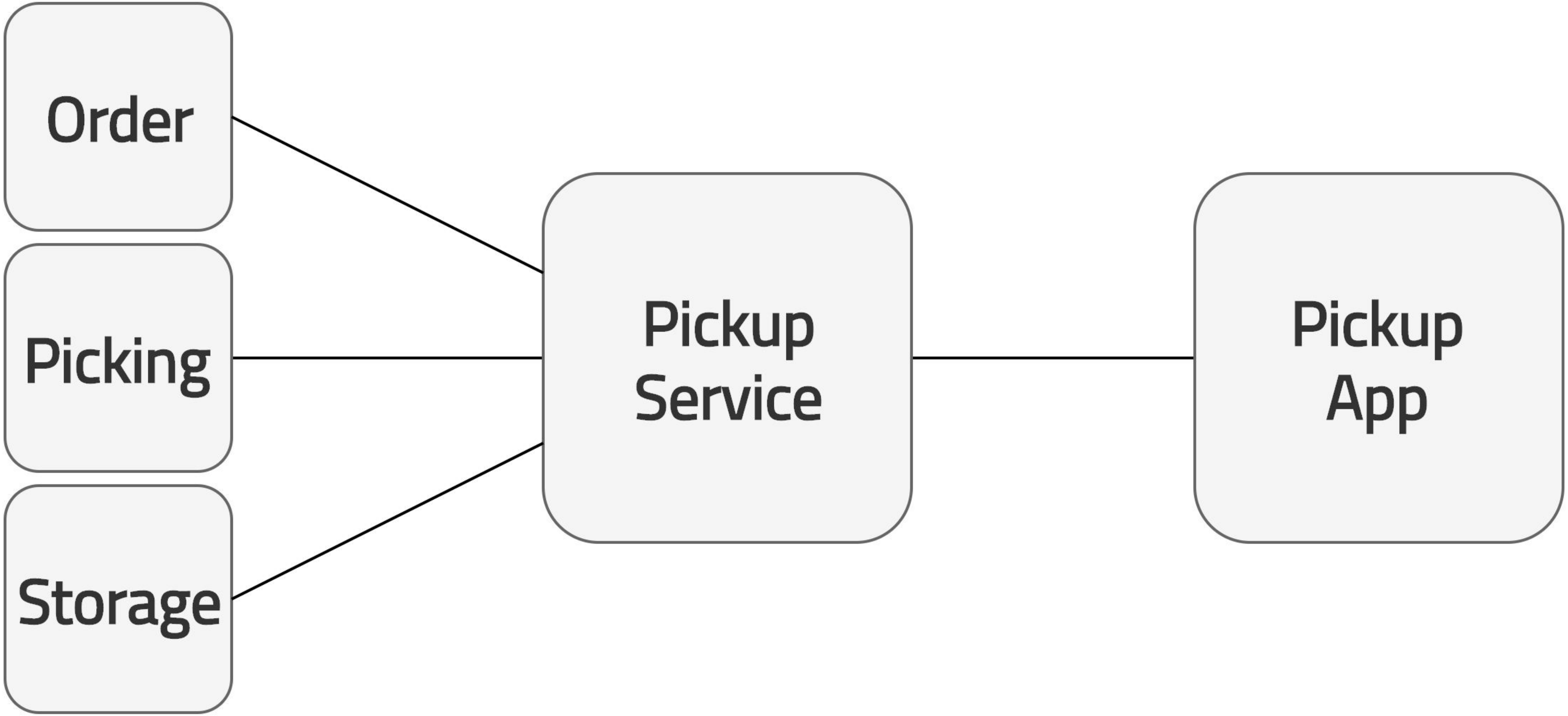

REWE DEIN MARKT
Großes Gewinnspiel! Jetzt bis 15.09.2019 teilnehmen!
Ein Shop – viele Möglichkeiten für dich!
Vorteile entdecken und tolle Preise gewinnen.
rewe.de/online-einkaufen
* Auflösung s. Rückseite
KUNDEN KLINGEL
BIXOLON

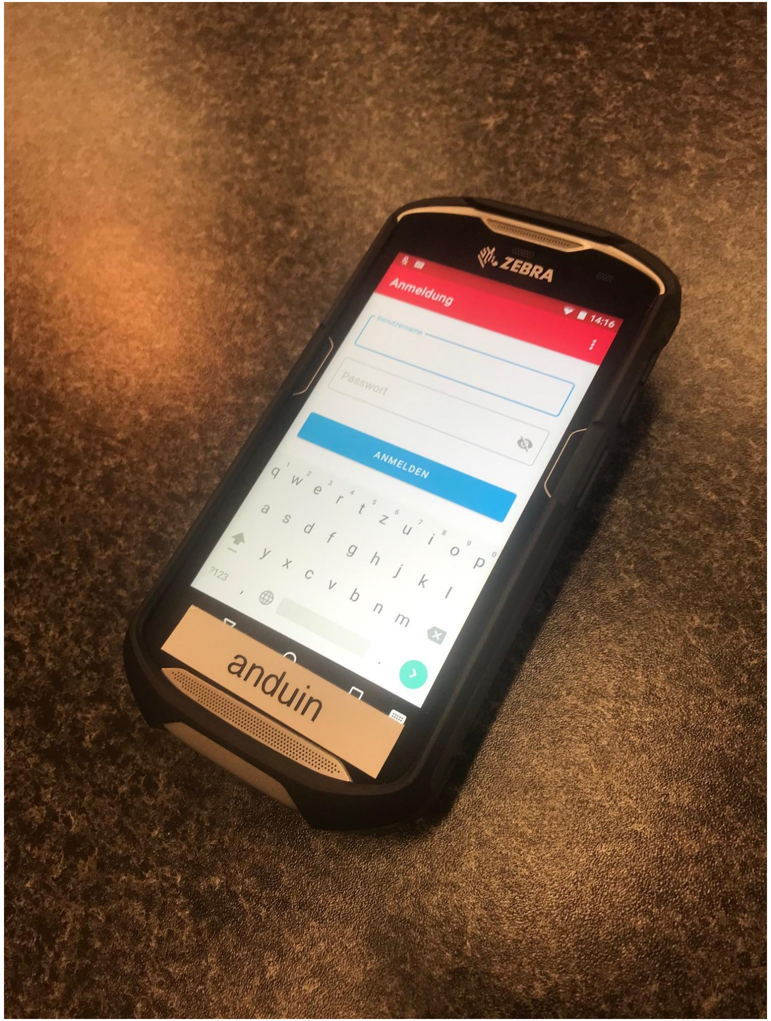

Softdrinks
Feinkost
KÜHLKONZEPT
Abholservice
Winston
Rauchen ist tödlich
Chesterfield
BECK'S
REWE

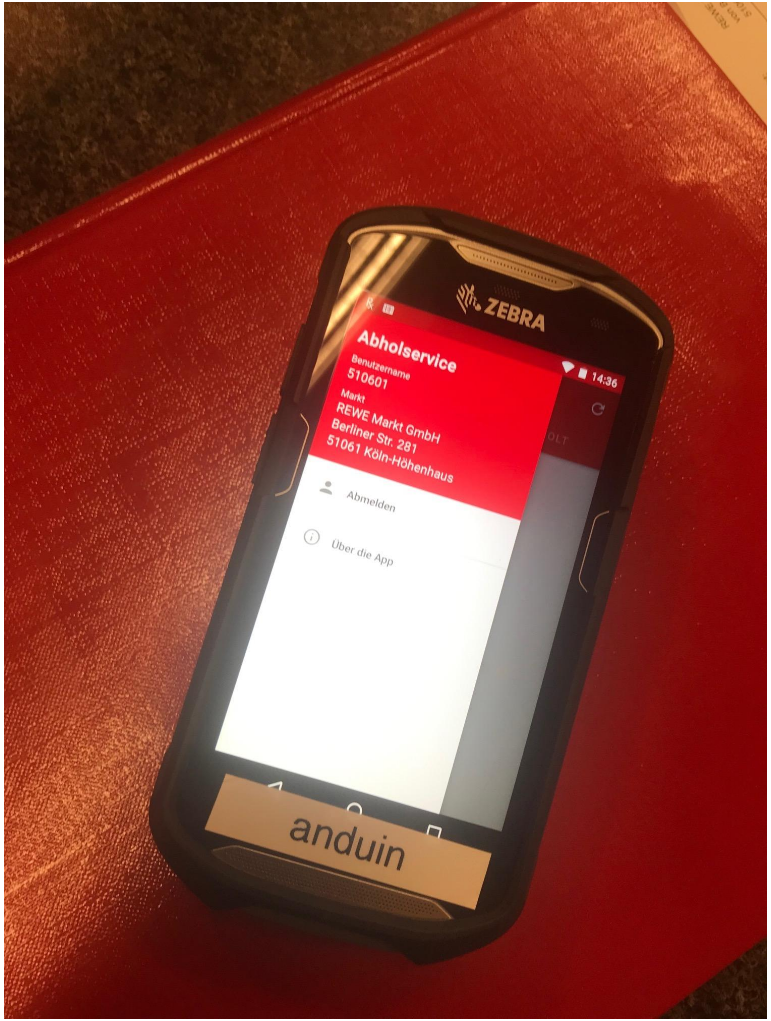
1. No more **manual** printing
2. Printing is **wasteful**
3. Unable to do **tracking**

Hexagon Architecture in the real world

1. Domain “Pickup”
2. **Prototype Pickup App**
3. Problems
4. Enter Hexagon Architecture
5. Results

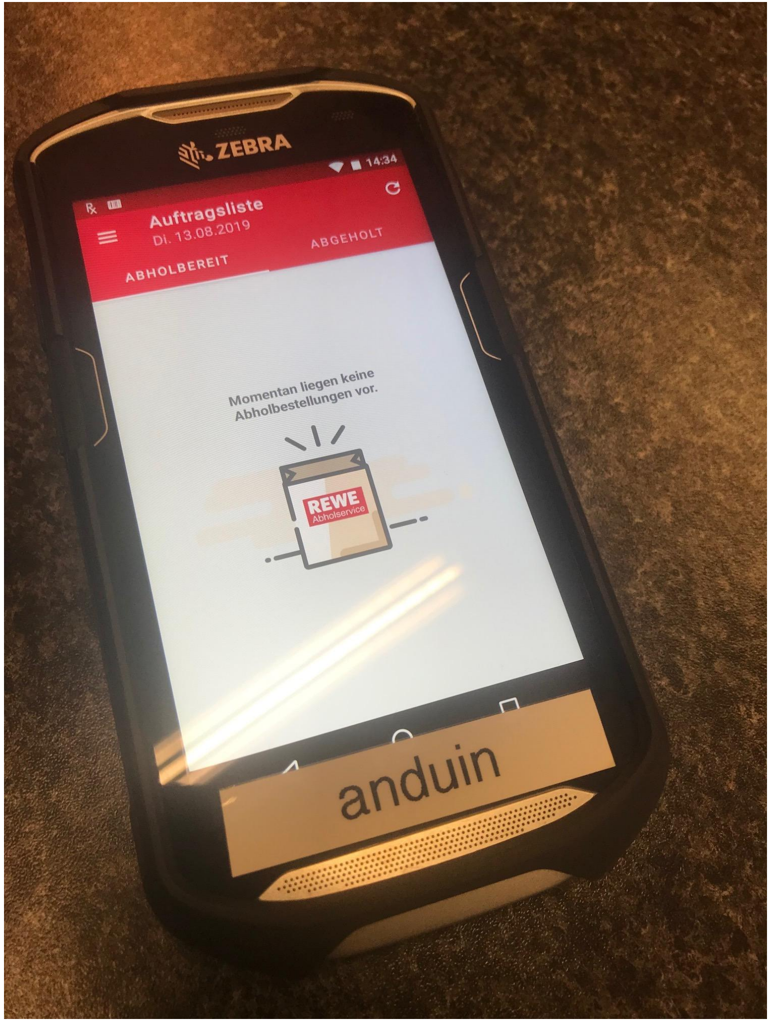

Order
Picking
Storage
Pickup Service
Pickup App

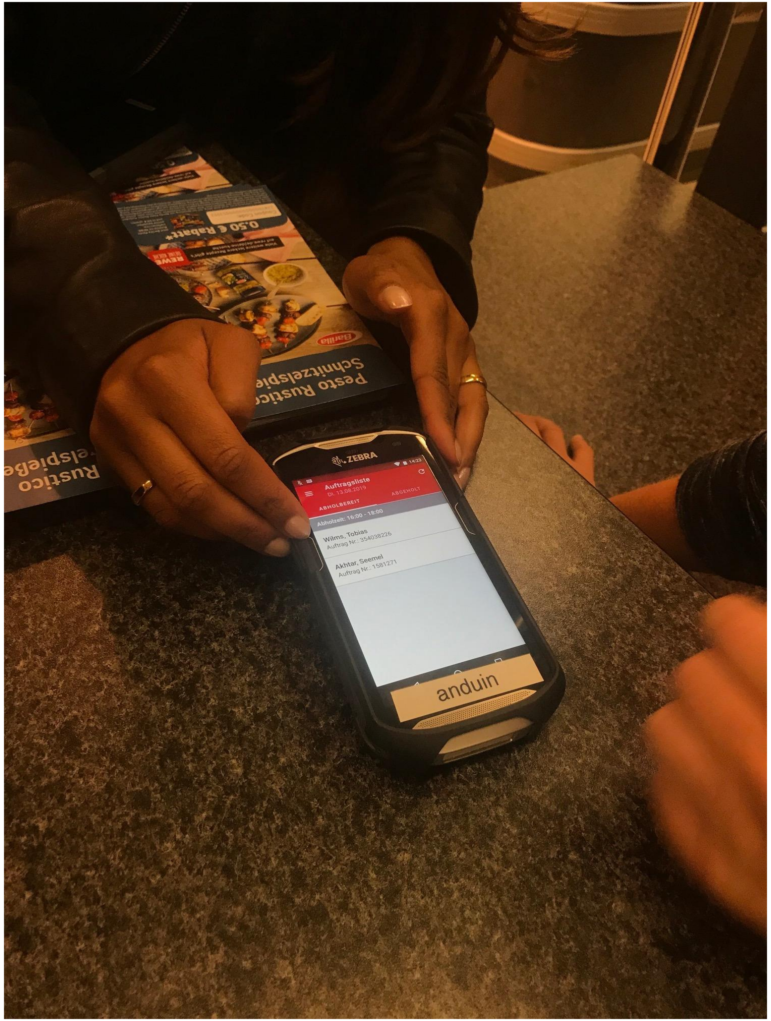


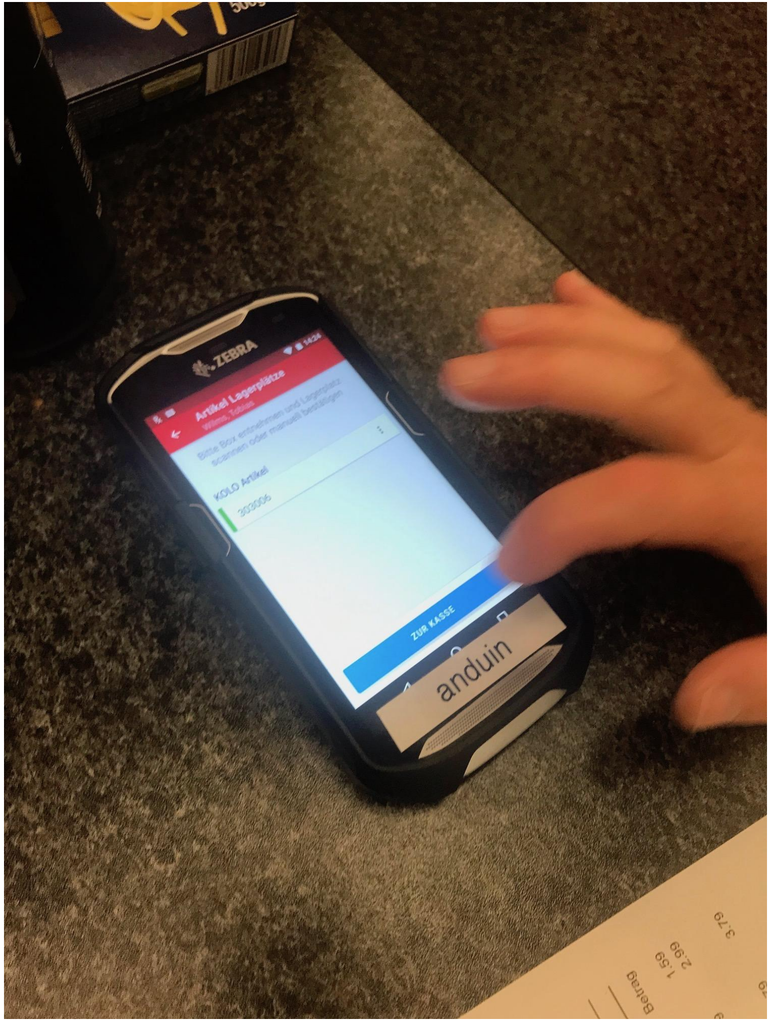


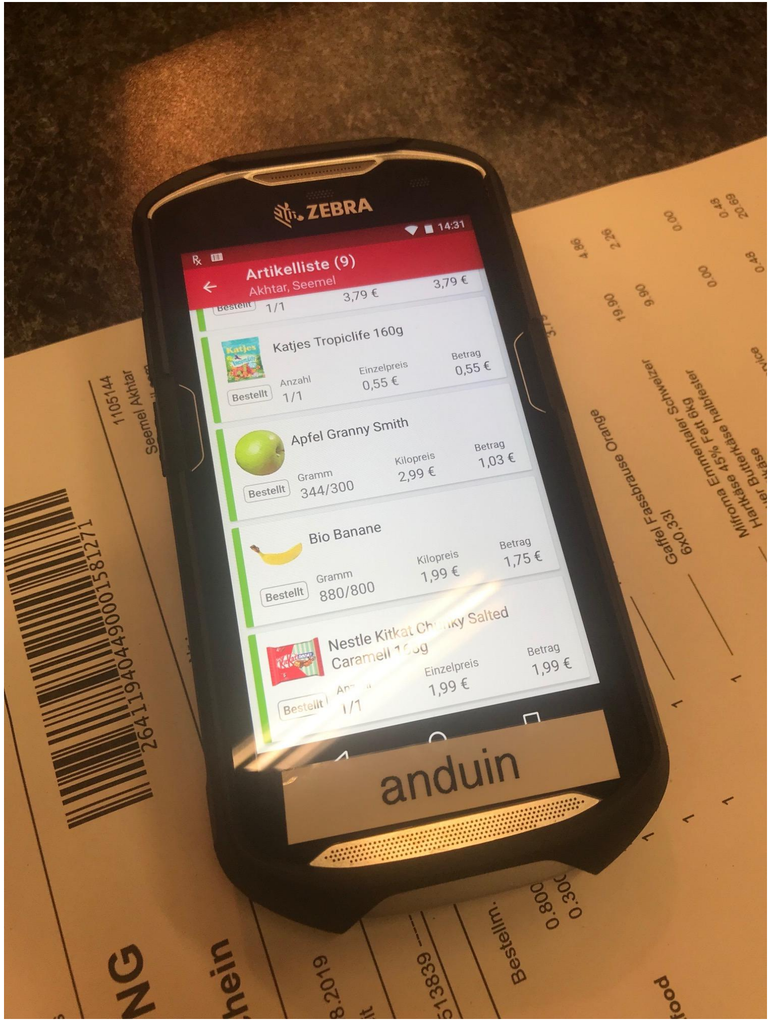
Benutzername
510601

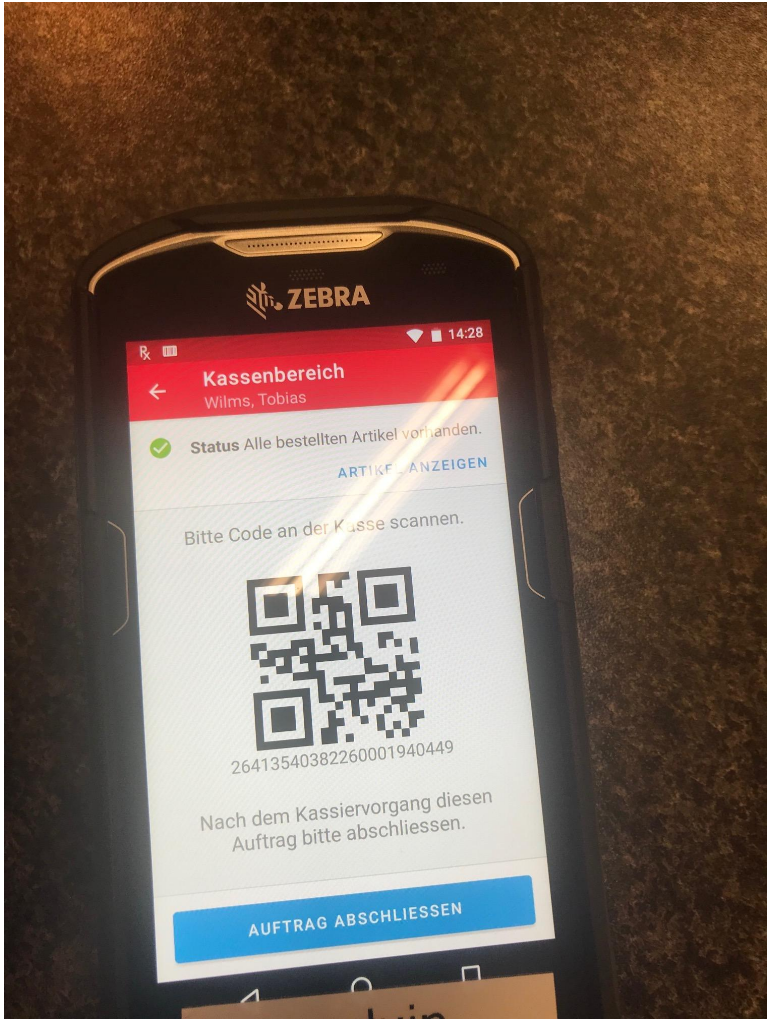
Markt
REWE Markt GmbH
Berliner Str. 281
51061 Köln-Höhenhaus


ZEBRA
14:34
Auftragsliste
Di. 13.08.2019
ABHOLBEREIT
ABGEHOLT
Momentan liegen keine Abholbestellungen vor.
REWE
Abholservice
anduin


0,50 € Rabatt
Pesto Rustico Schnitzelspieße
Barilla
ZEBRA
14:23
Auftragsliste
Di. 13.08.2019
ABHOLBEREIT
ABGEHOLT
Abholzeit: 16:00 - 18:00
Wilms, Tobias
Auftrag Nr.: 354038226
Akhtar, Seemel
Auftrag Nr.: 1581271
anduin


ZEBRA
Artikel Lagerplätze
KOLO Artikel
ZUR KASSE
anduin


ZEBRA
14:31
Artikelliste (9)
Akhtar, Seemel
Katjes Tropiclife 160g
Anzahl 1/1
Einzelpreis 0,55 €
Betrag 0,55 €
Apfel Granny Smith
Gramm 344/300
Kilopreis 2,99 €
Betrag 1,03 €
Bio Banane
Gramm 880/800
Kilopreis 1,99 €
Betrag 1,75 €
Nestle Kitkat Chunky Salted Caramell
Einzelpreis 1,99 €
Betrag 1,99 €
Bestellt
anduin



Kassenbereich
Wilms, Tobias

Status Alle bestellten Artikel vorhanden.

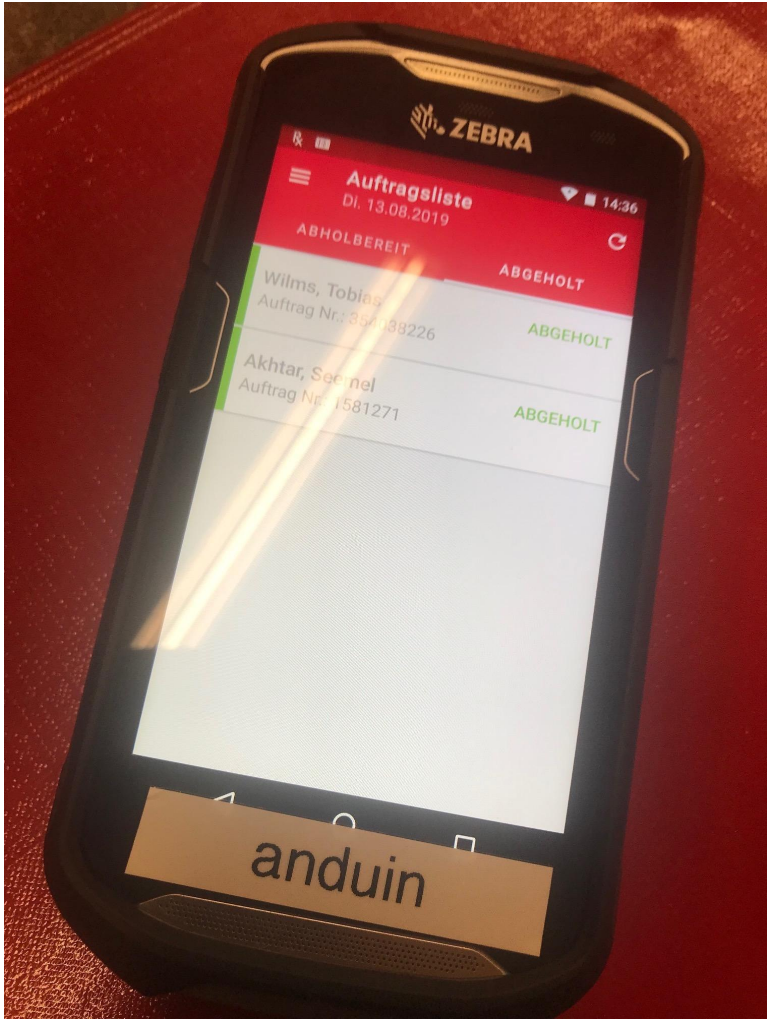
ARTIKEL ANZEIGEN

Bitte Code an der Kasse scannen.

26413540382260001940449

Nach dem Kassiervorgang diesen Auftrag bitte abschliessen.

AUFTRAG ABSCHLIESSEN



Auftragsliste
Di. 13.08.2019

ABHOLBEREIT | ABGEHOLT

Wilms, Tobias
Auftrag Nr.: 35[illegible]38226
ABGEHOLT

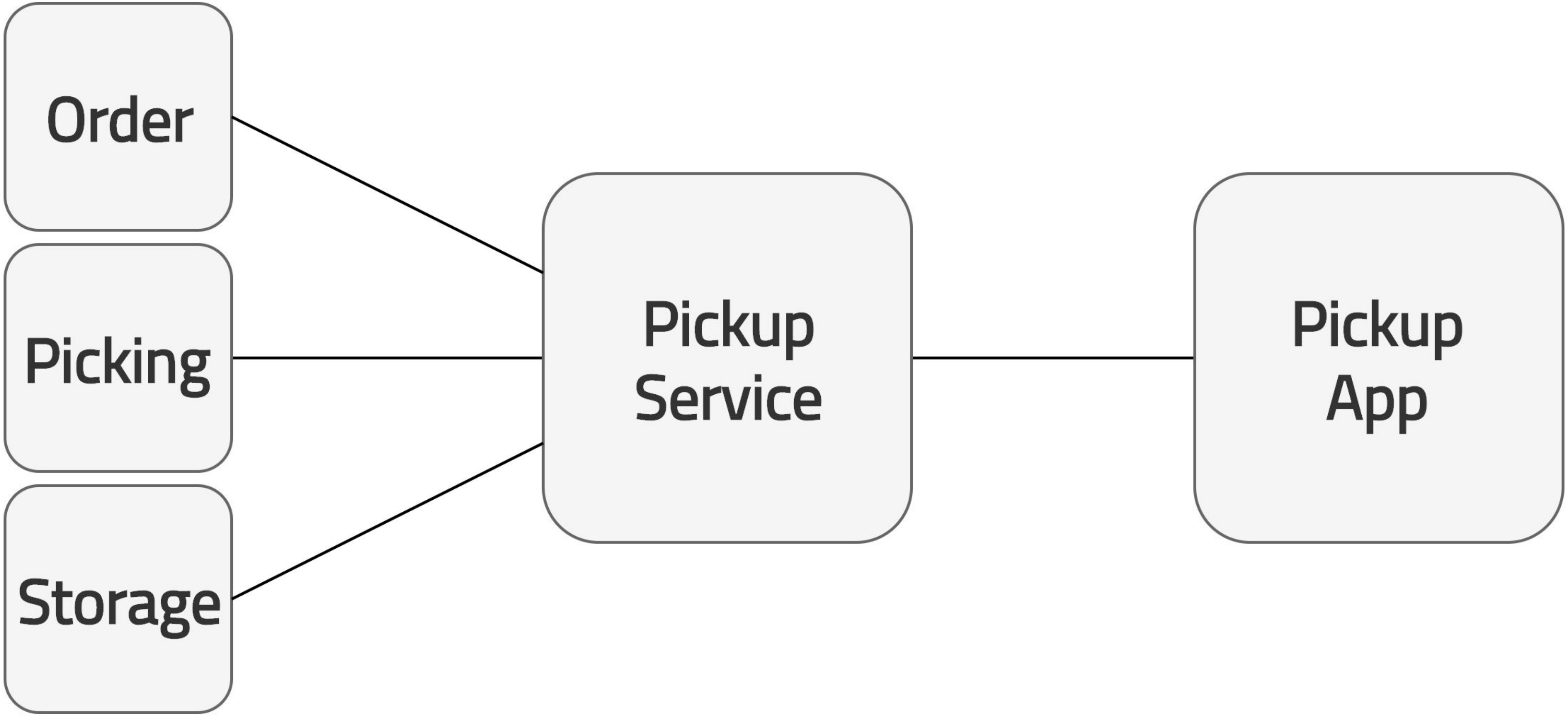
Akhtar, Se[illegible]mel
Auftrag Nr.: [illegible]1581271
ABGEHOLT

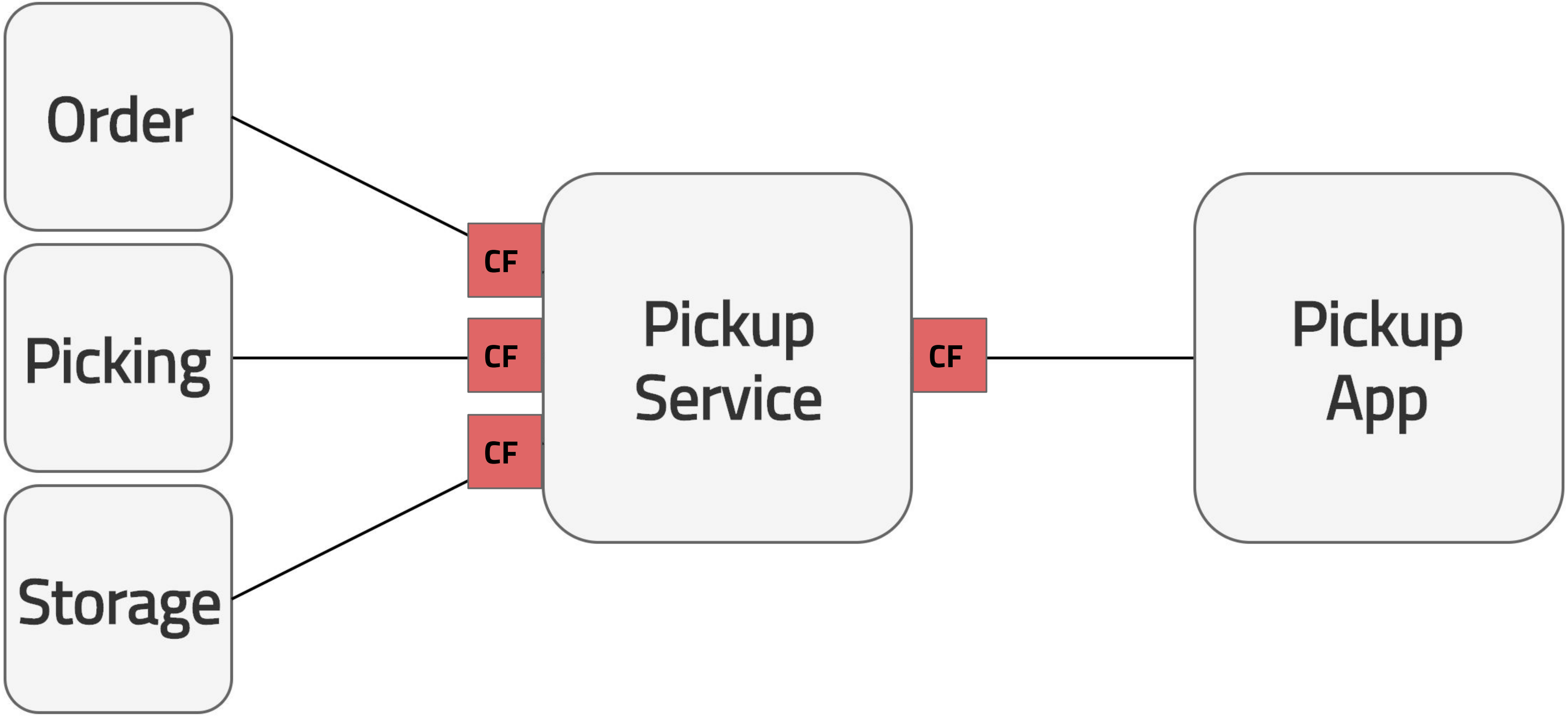
Hexagon Architecture in the real world

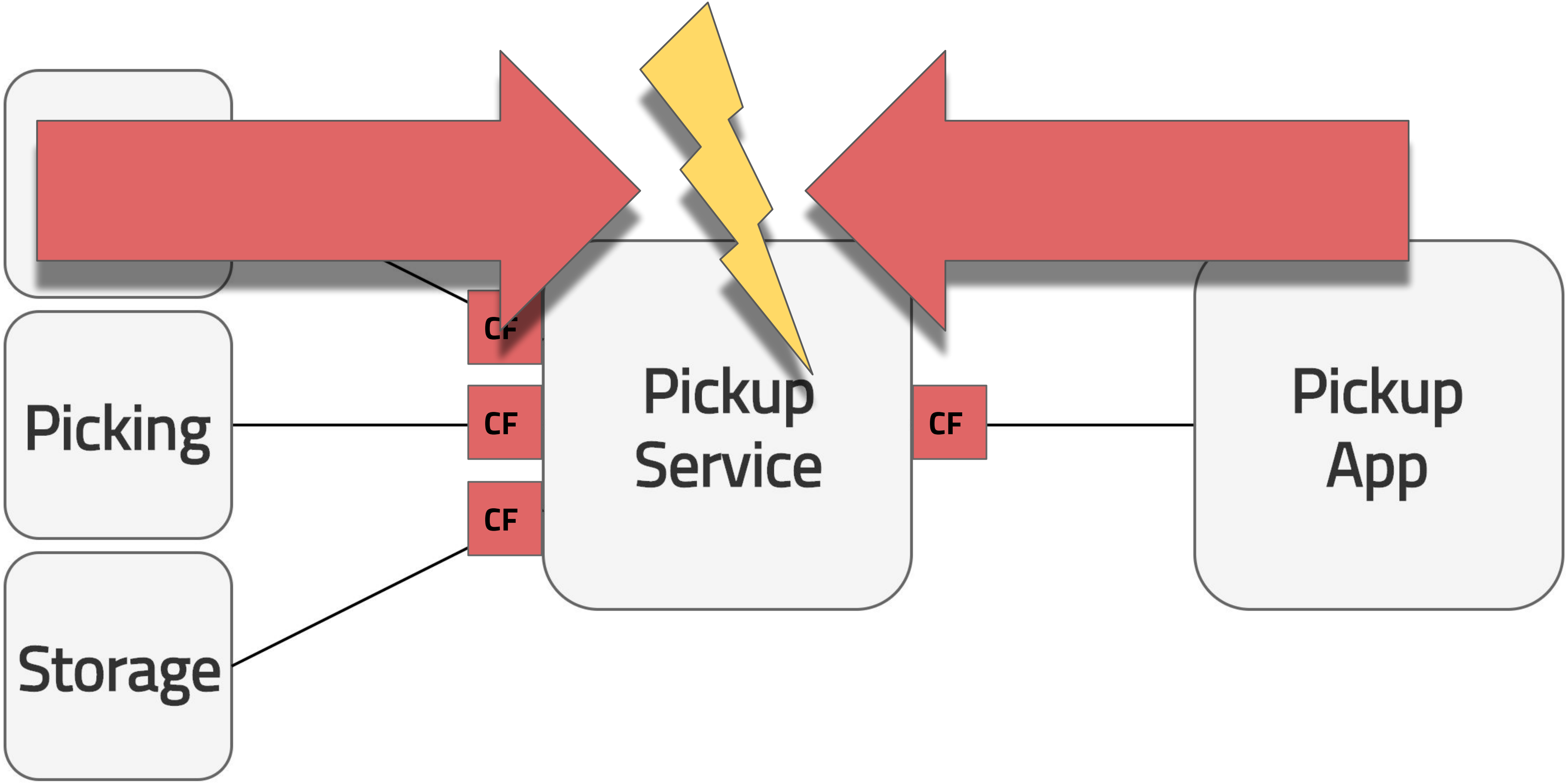
1. Domain “Pickup”
2. Prototype Pickup App
3. **Problems**
4. Enter Hexagon Architecture
5. Results

Problem #1

No domain model


Order
Picking
Storage
Pickup Service
Pickup App

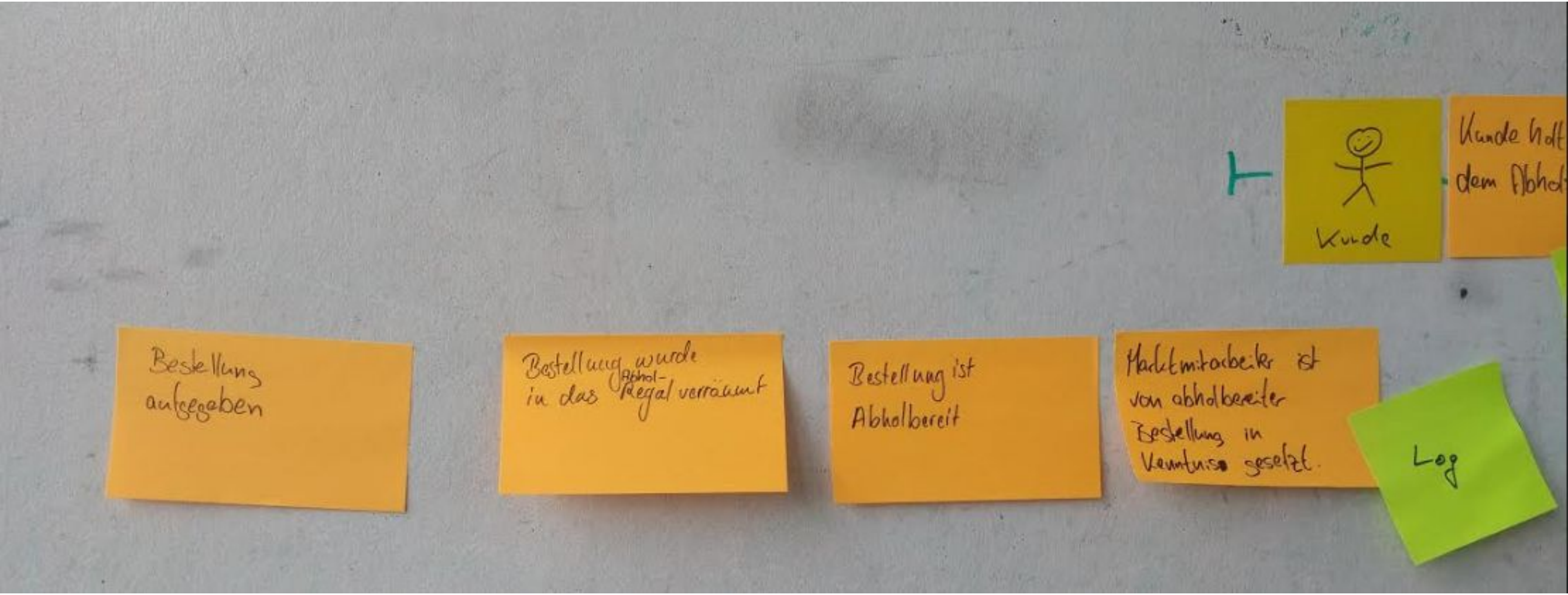

Order
Picking
Storage
CF
CF
CF
Pickup Service
CF
Pickup App

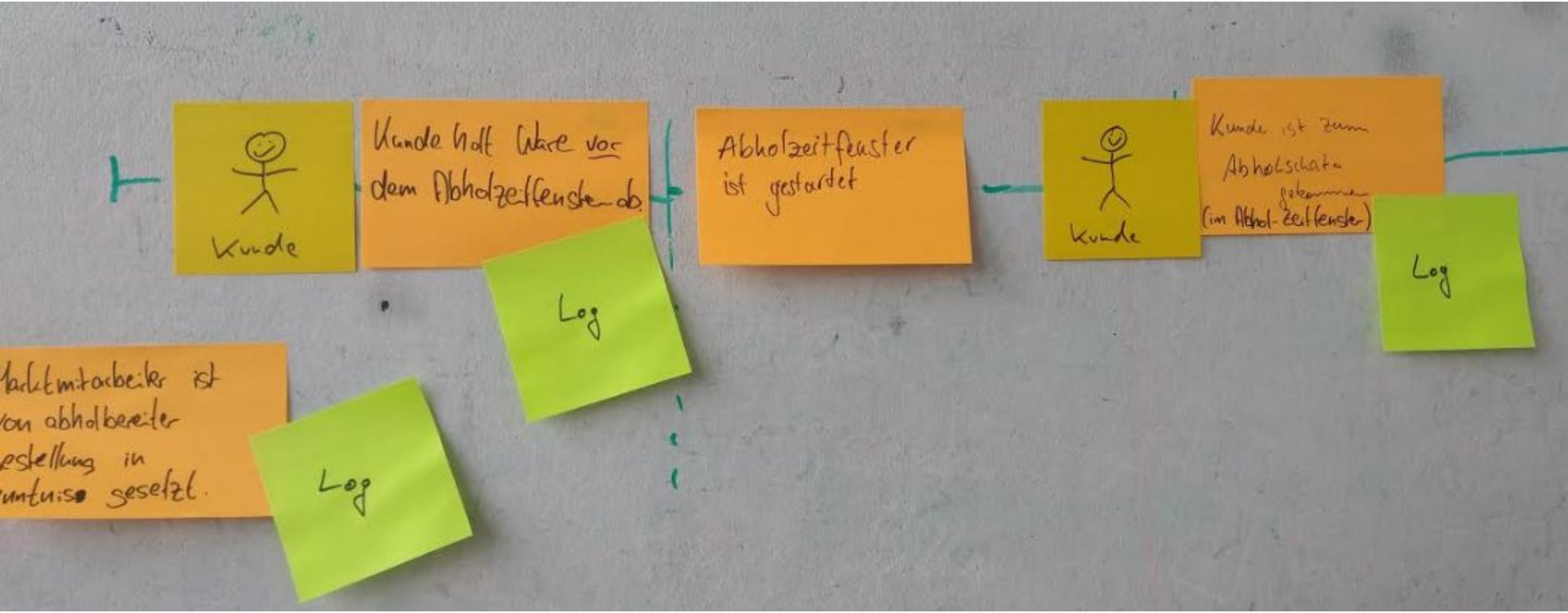


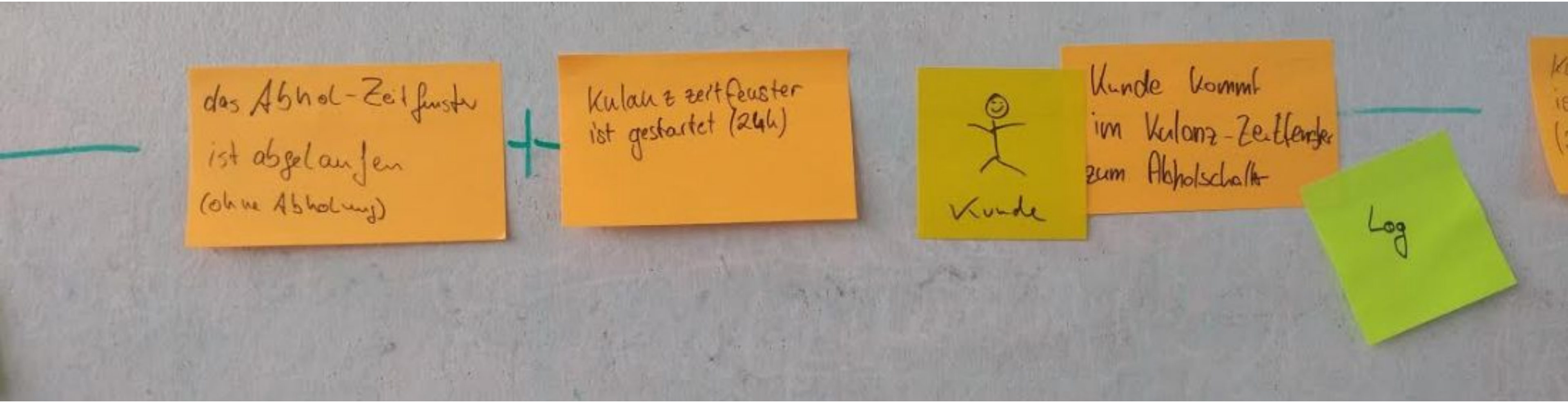
REWE digital

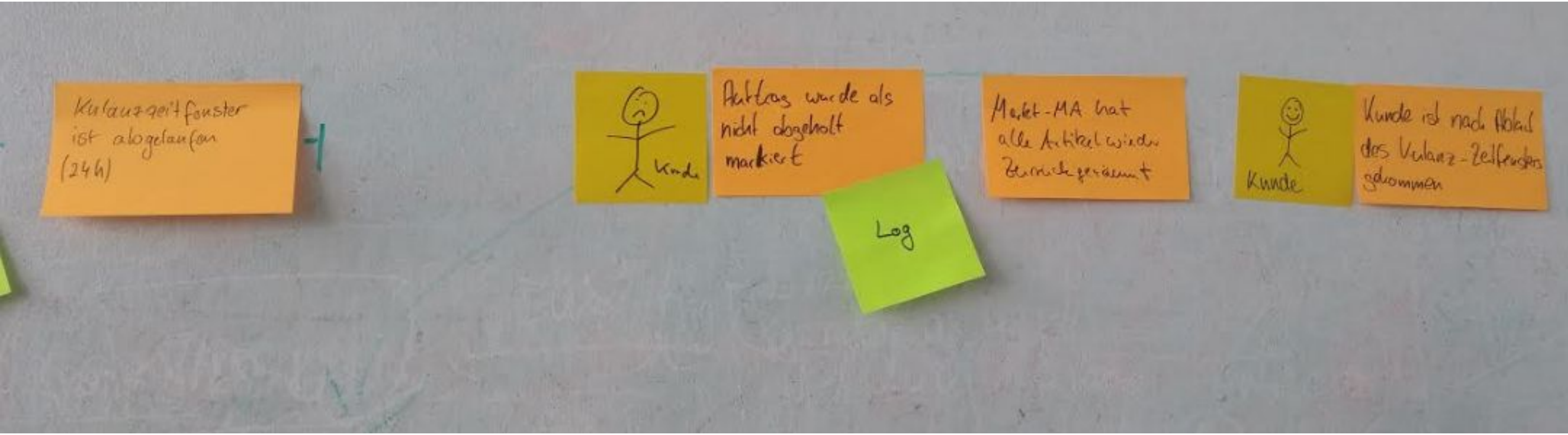
Problem #2

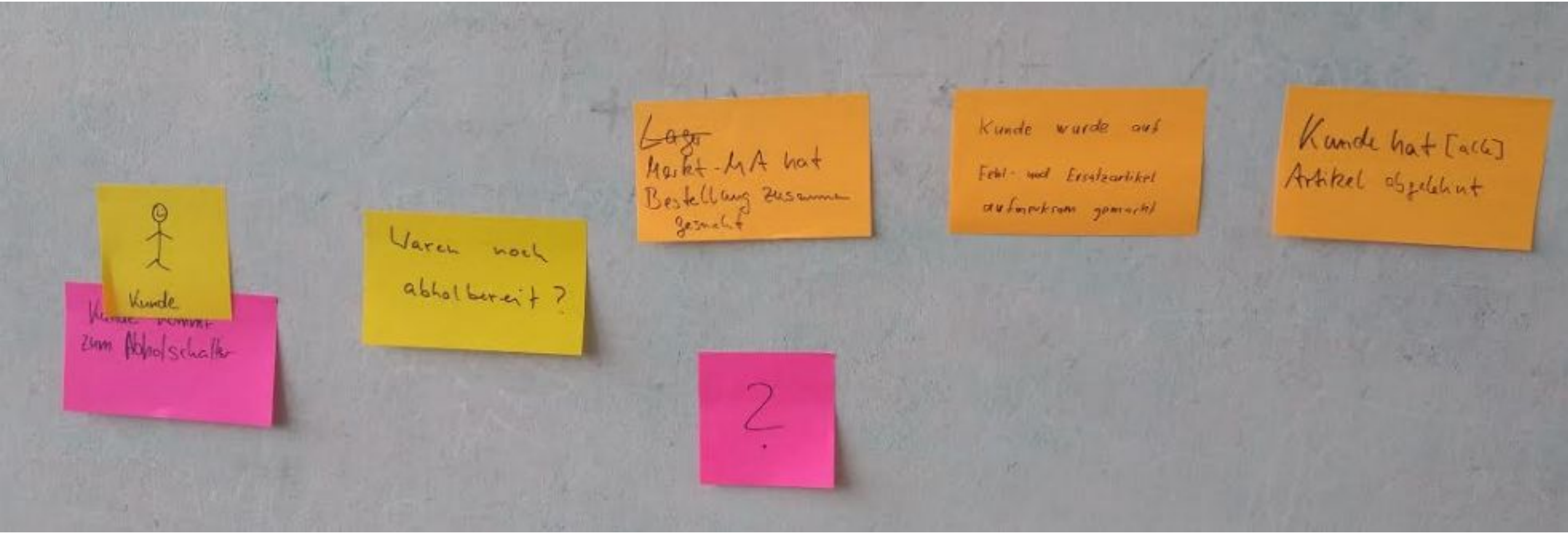
No business metrics

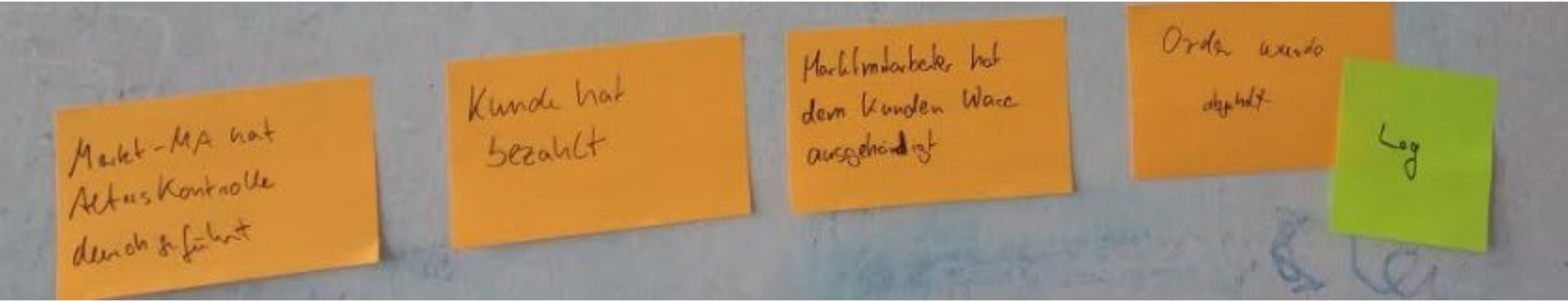

Bestellung aufgegeben
Bestellung wurde in das Abhol-Regal verräumt
Bestellung ist Abholbereit
Marktmitarbeiter ist von abholbereiter Bestellung in Kenntnis gesetzt.
Kunde
Kunde holt
dem Abhol
Log


Kunde
Kunde holt Ware vor dem Abholzeitfenster ab
Log
Abholzeitfenster ist gestartet
Kunde
Kunde ist zum Abholschalter gekommen (im Abhol-Zeitfenster)
Log
arktmitarbeiter ist
on abholbereiter
estellung in
enntnis gesetzt.
Log


das Abhol-Zeitfenster ist abgelaufen (ohne Abholung)
Kulanzzeitfenster ist gestartet (24h)
Kunde
Kunde kommt im Kulanz-Zeitfenster zum Abholschalter
Log

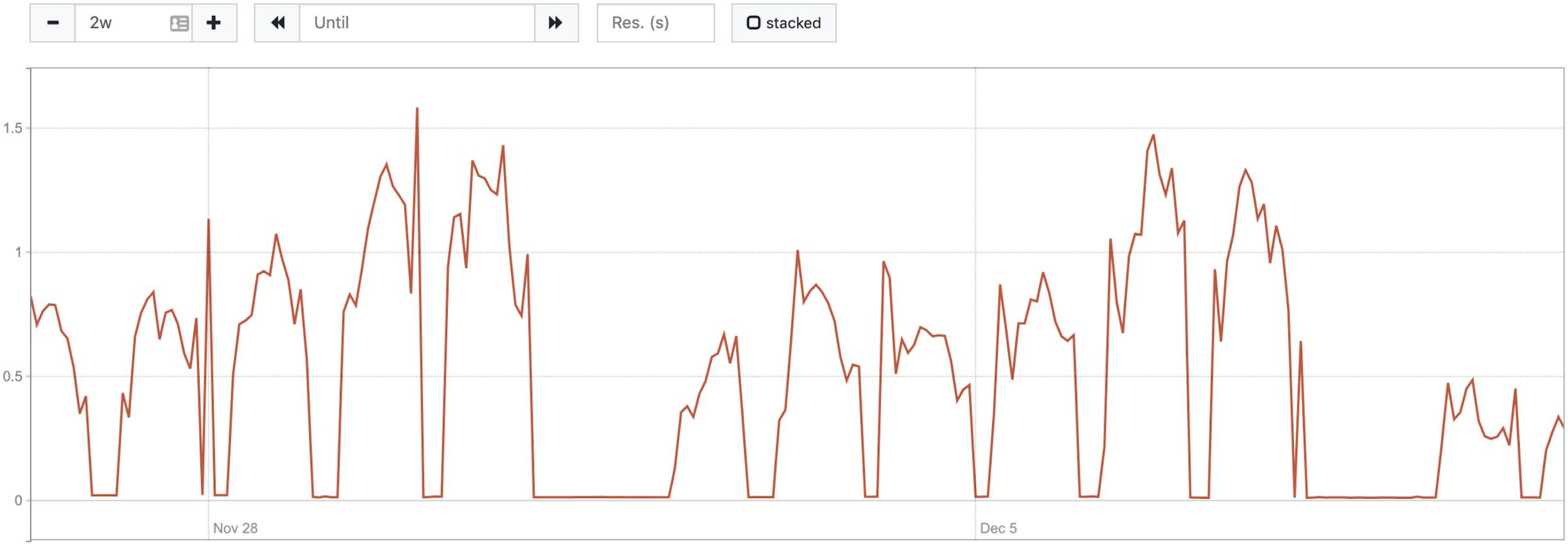

Kulanzzeitfenster ist abgelaufen (24h)
Kunde
Auftrag wurde als nicht abgeholt markiert
Log
Markt-MA hat alle Artikel wieder zurückgeräumt
Kunde
Kunde ist nach Ablauf des Kulanz-Zeitfensters gekommen


Kunde
Kunde kommt zum Abholschalter
Waren noch abholbereit?
~~Lager~~ Markt-MA hat Bestellung zusammengesucht
?
Kunde wurde auf Fehl- und Ersatzartikel aufmerksam gemacht
Kunde hat [alle] Artikel abgelehnt

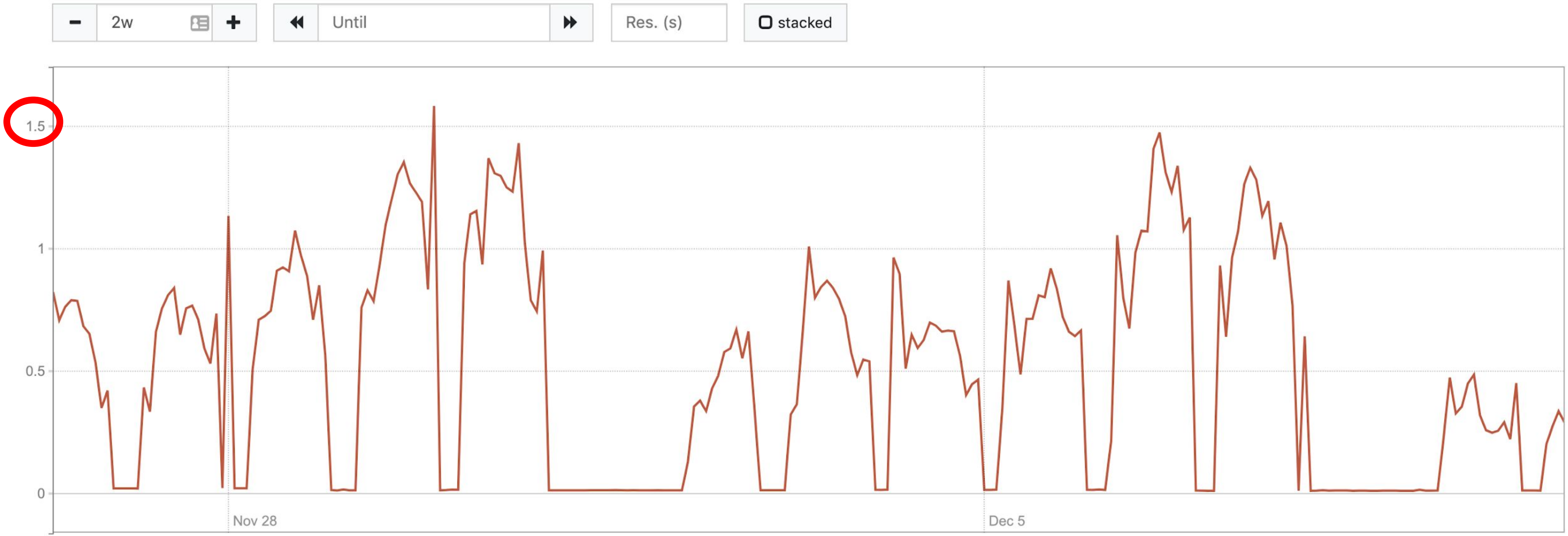

Markt-MA hat Alterskontrolle durchgeführt
Kunde hat bezahlt
Marktmitarbeiter hat dem Kunden Ware ausgehändigt
Order wurde abgeholt
Log

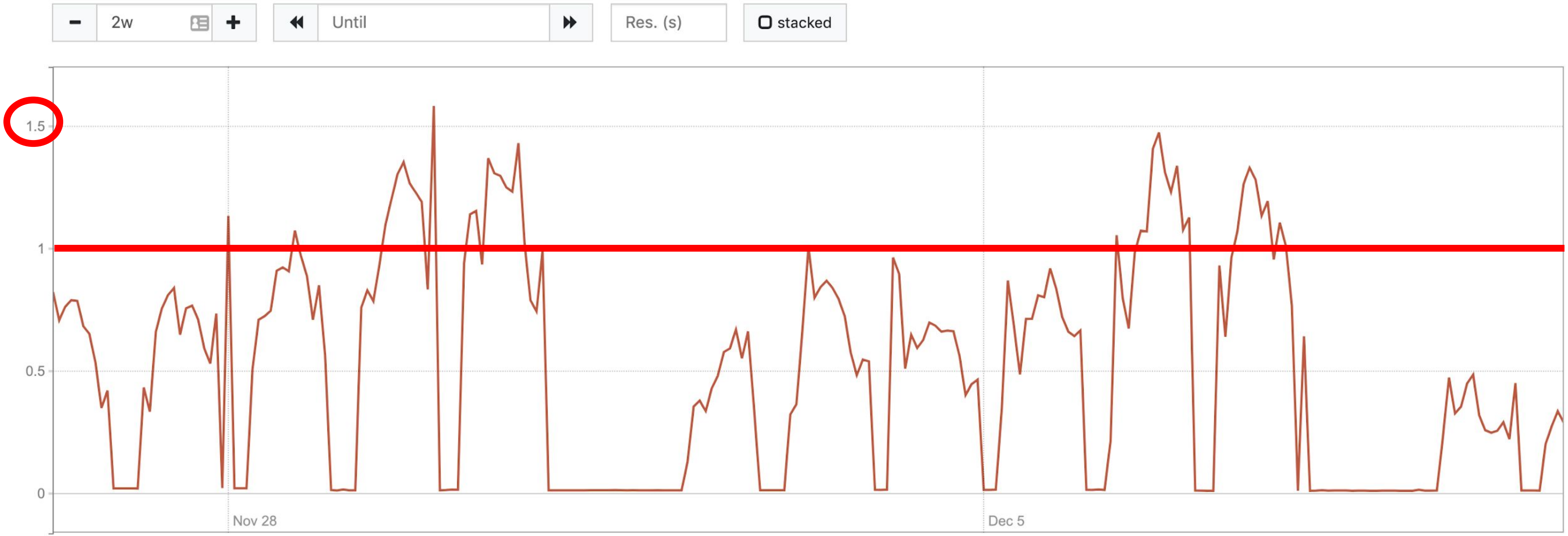
Problem #3

Performance



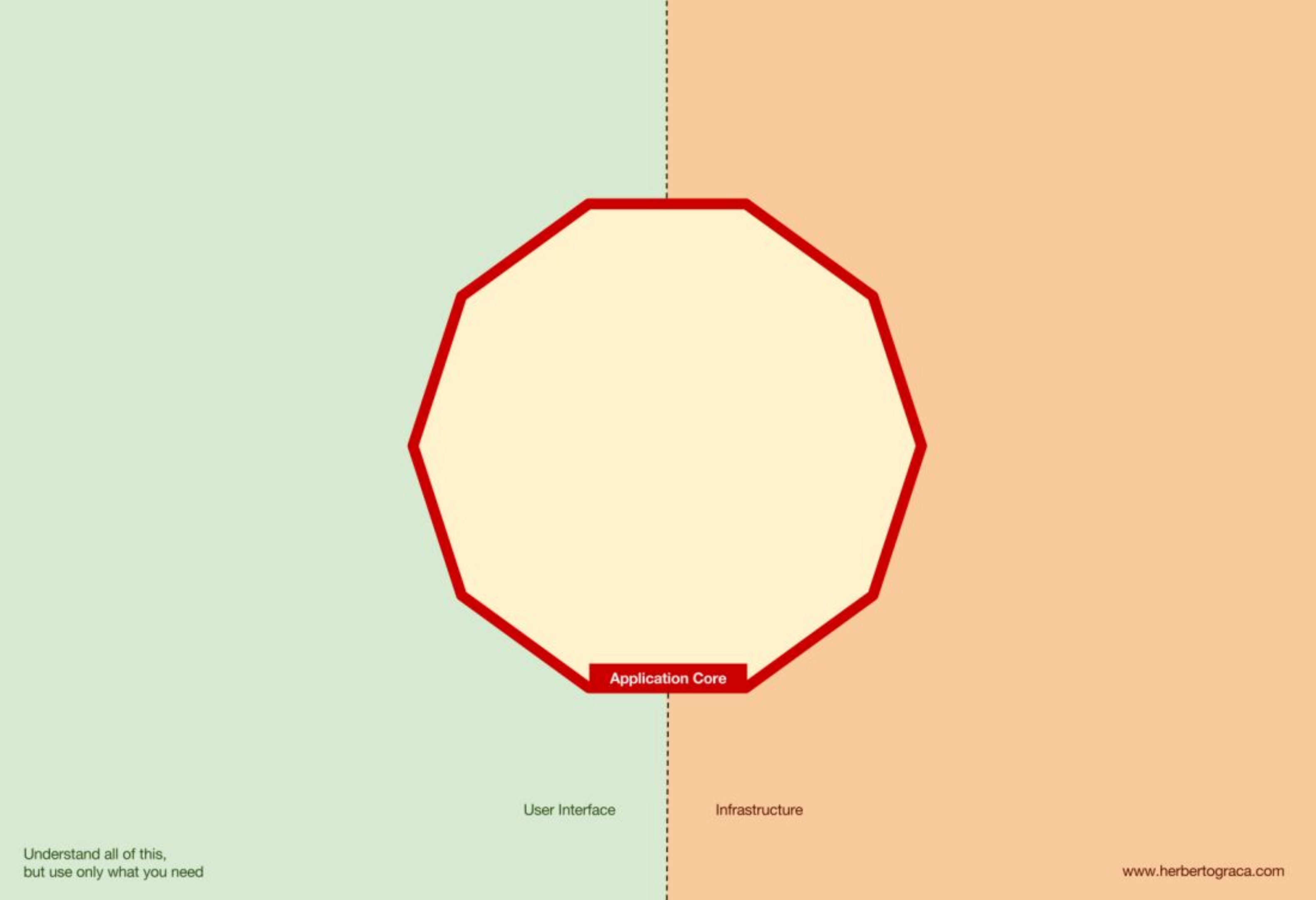
−
2w
+
Until
Res. (s)
□ stacked
1.5
1
0.5
0
Nov 28
Dec 5


−
2w
+
Until
Res. (s)
stacked
1.5
1
0.5
0
Nov 28
Dec 5


2w
Until
Res. (s)
□ stacked
1.5
1
0.5
0
Nov 28
Dec 5

Hexagon Architecture in the real world

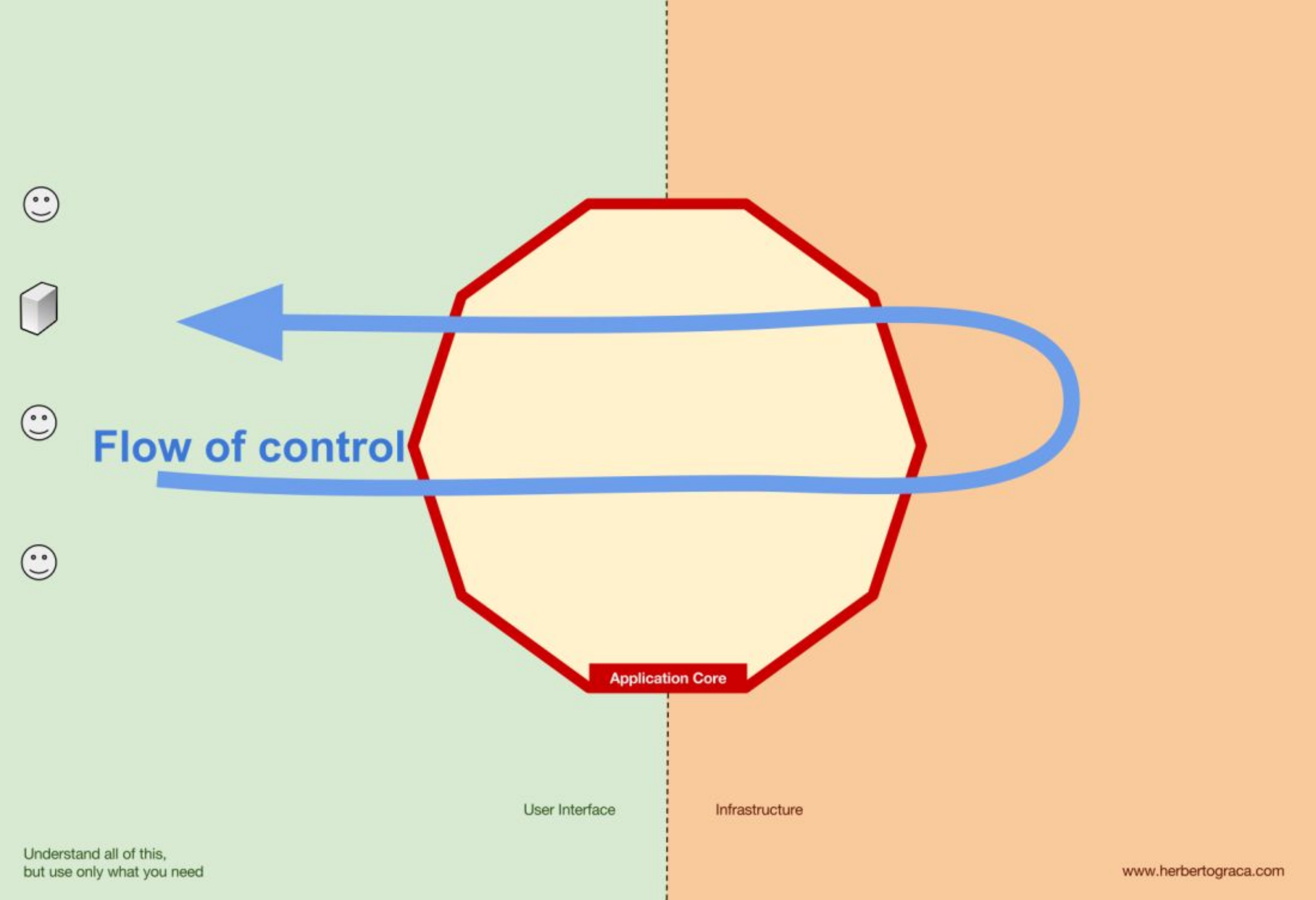
1. Domain “Pickup”
2. Prototype Pickup App
3. Problems
4. **Enter Hexagon Architecture**
5. Results



Understand all of this,
but use only what you need

www.herbertograca.com

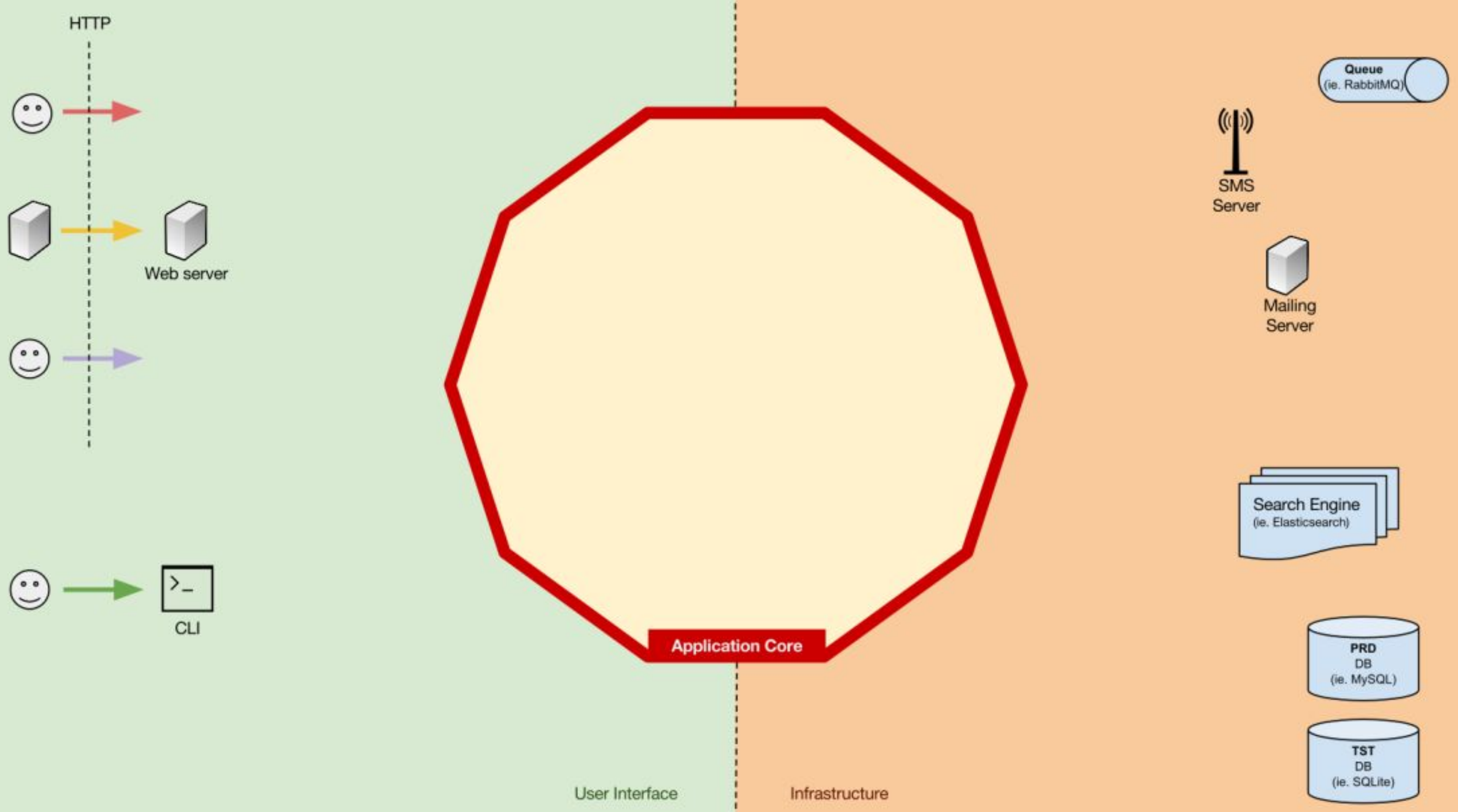
REWE digital



Understand all of this,
but use only what you need

www.herbertograca.com

REWE digital

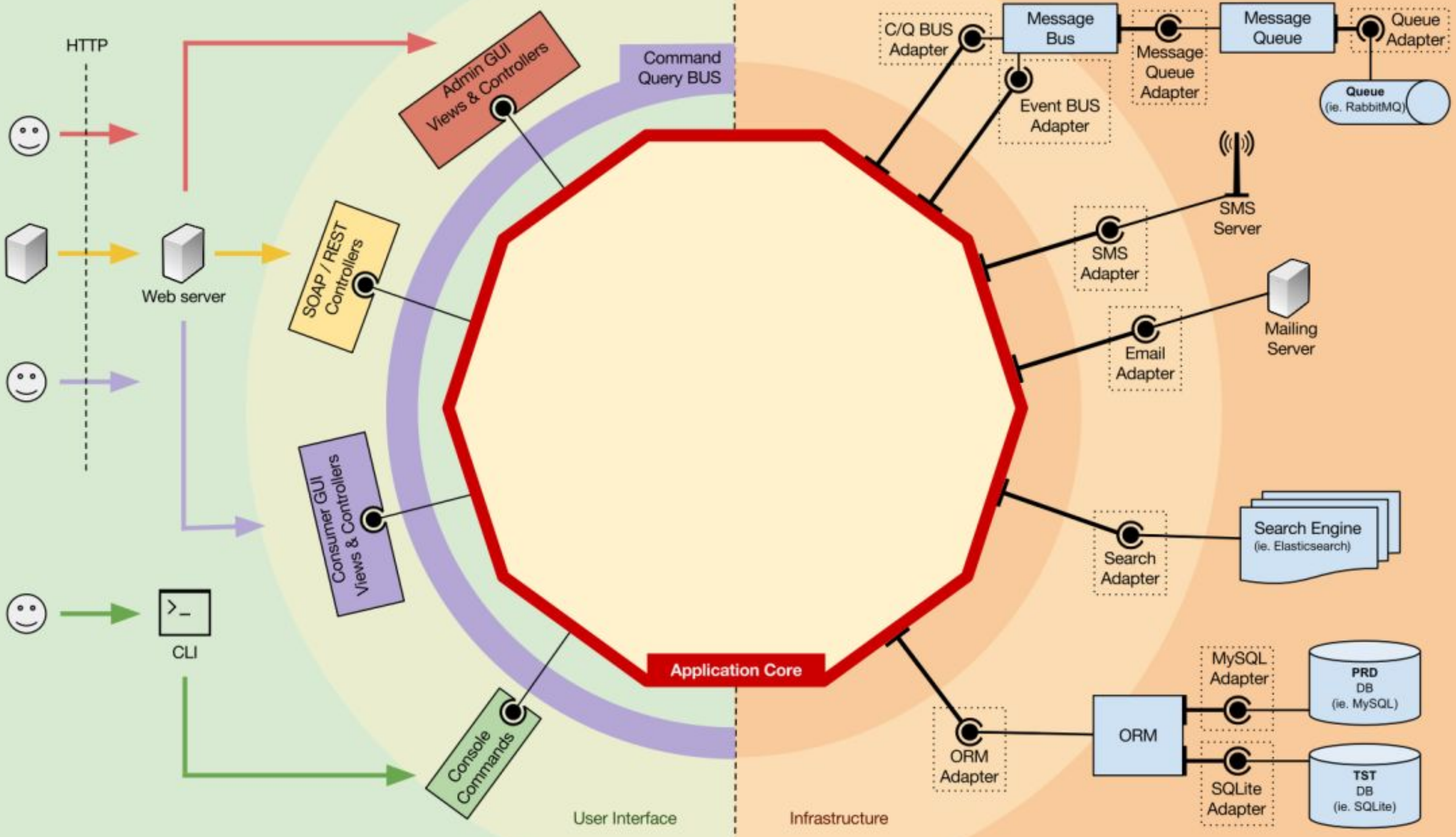


Understand all of this,
but use only what you need

www.herbertograca.com

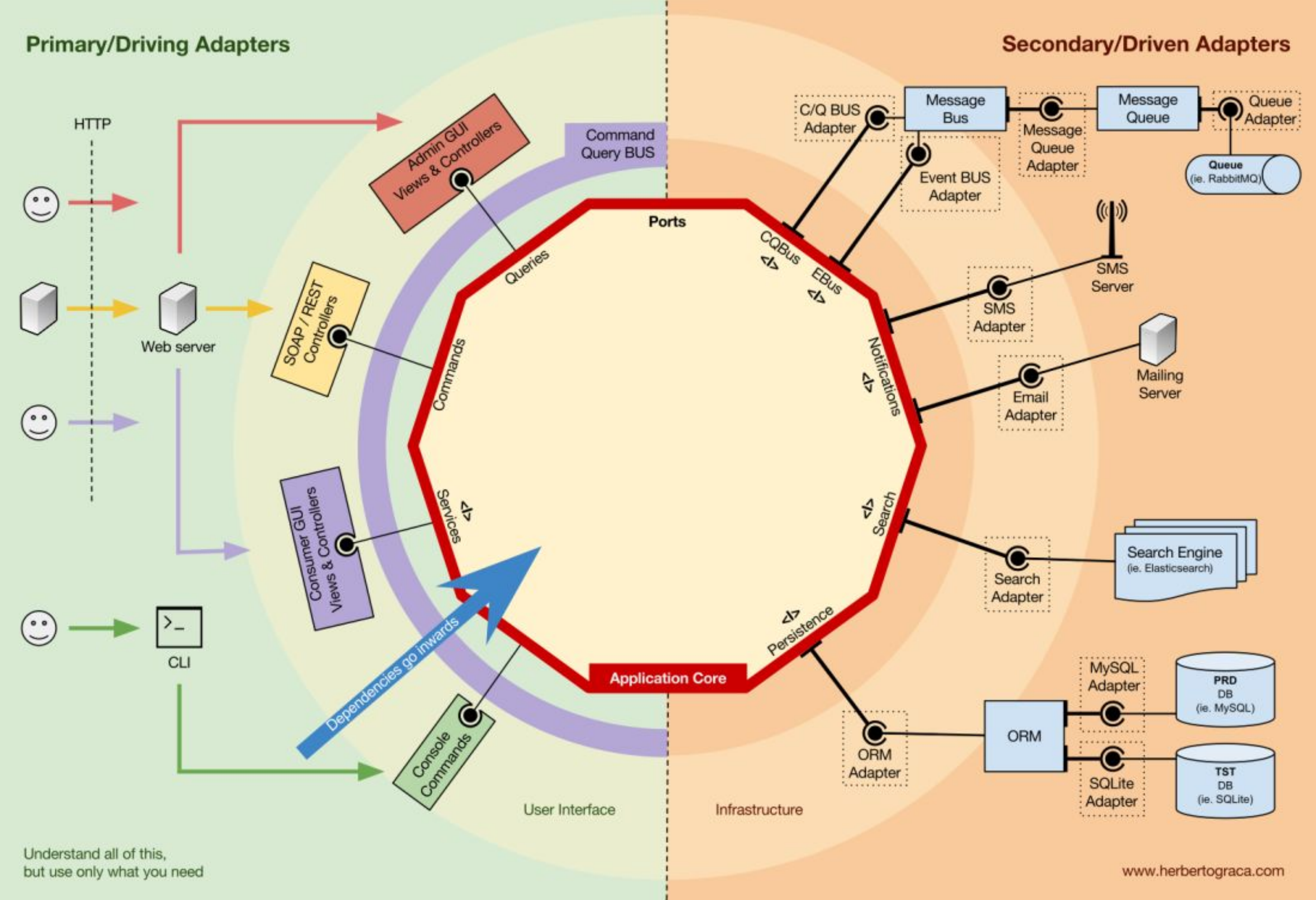
Primary/Driving Adapters

Secondary/Driven Adapters



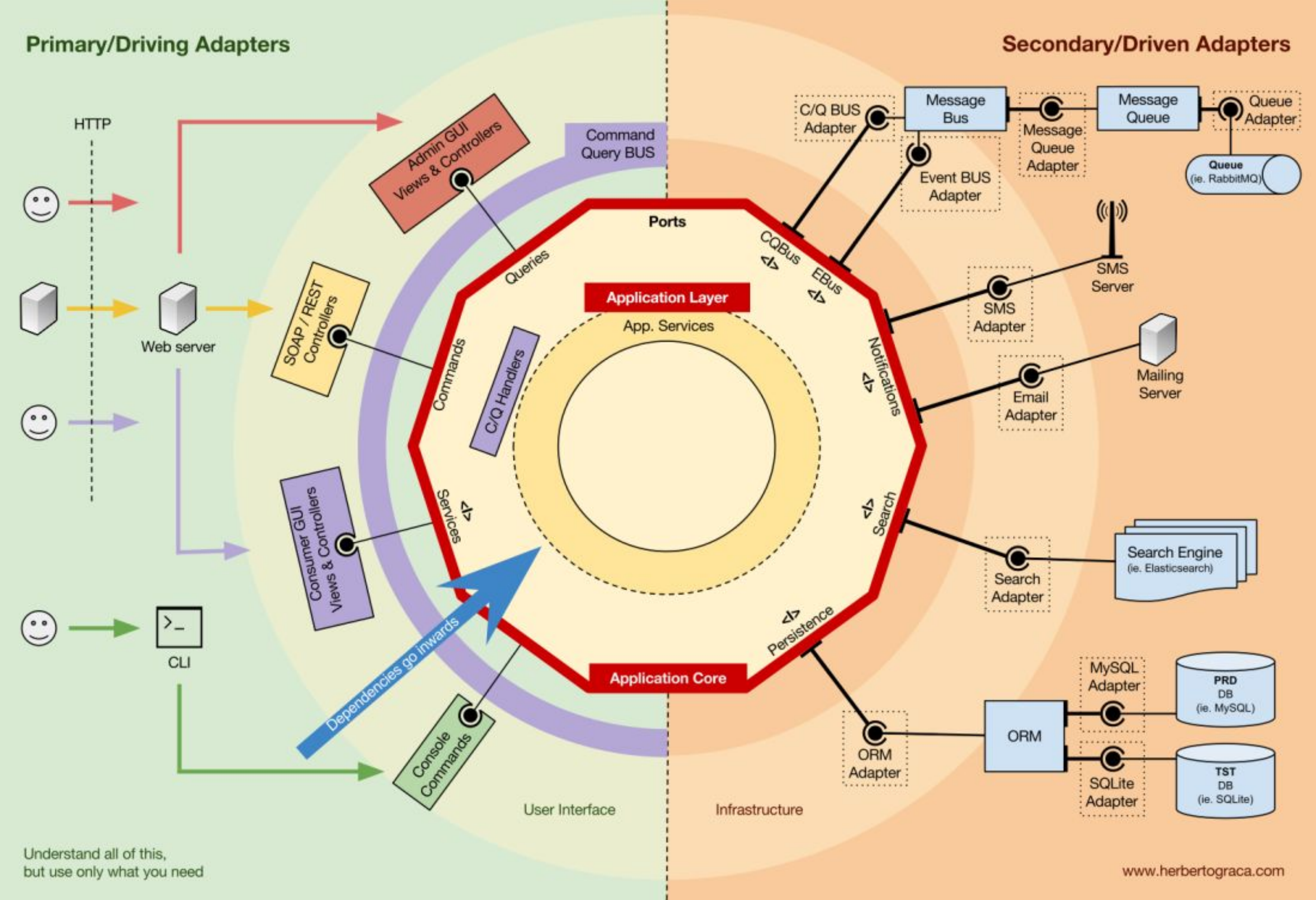
Understand all of this,
but use only what you need

www.herbertograca.com



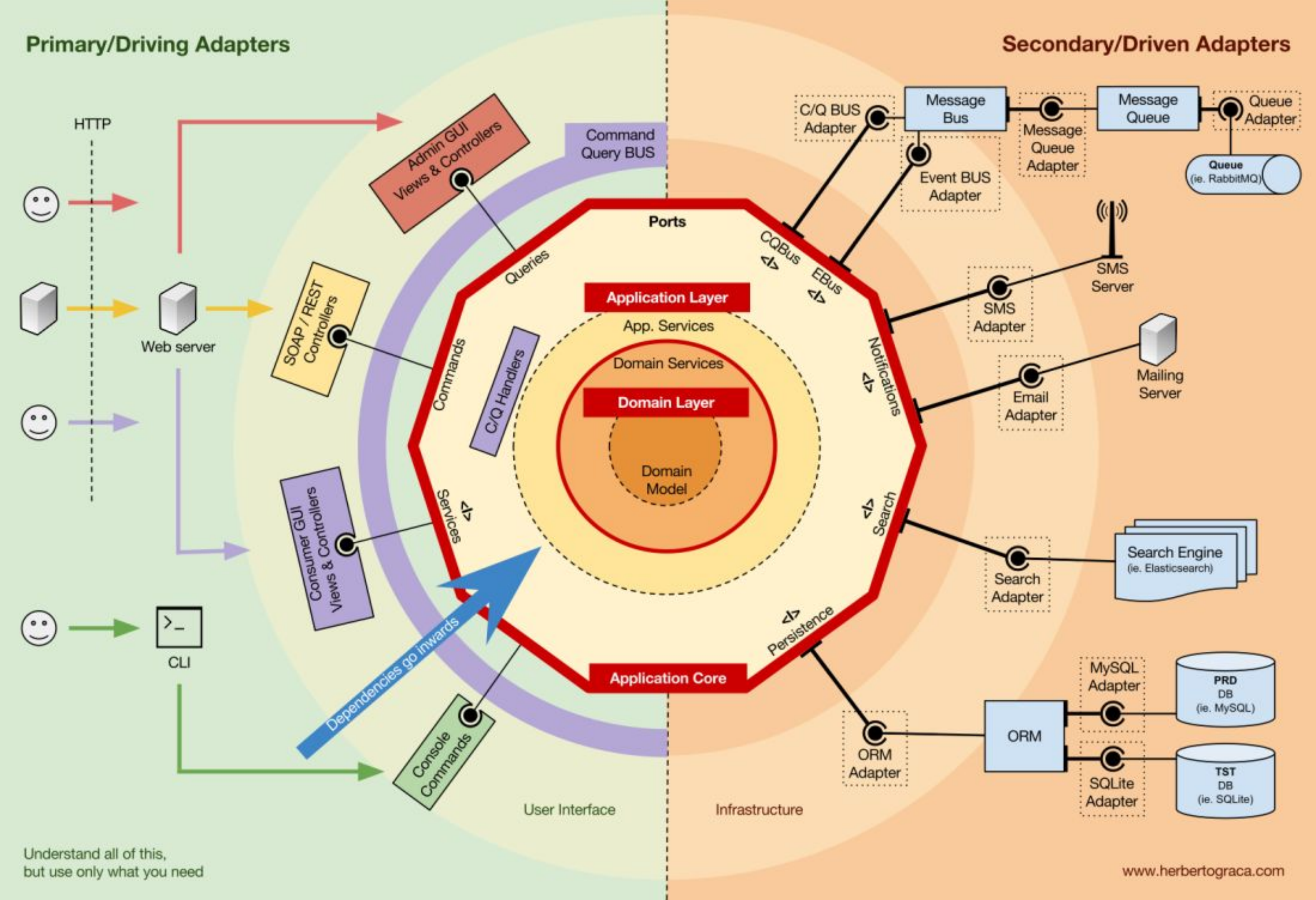
Understand all of this,
but use only what you need

www.herbertograca.com



Understand all of this,
but use only what you need

www.herbertograca.com



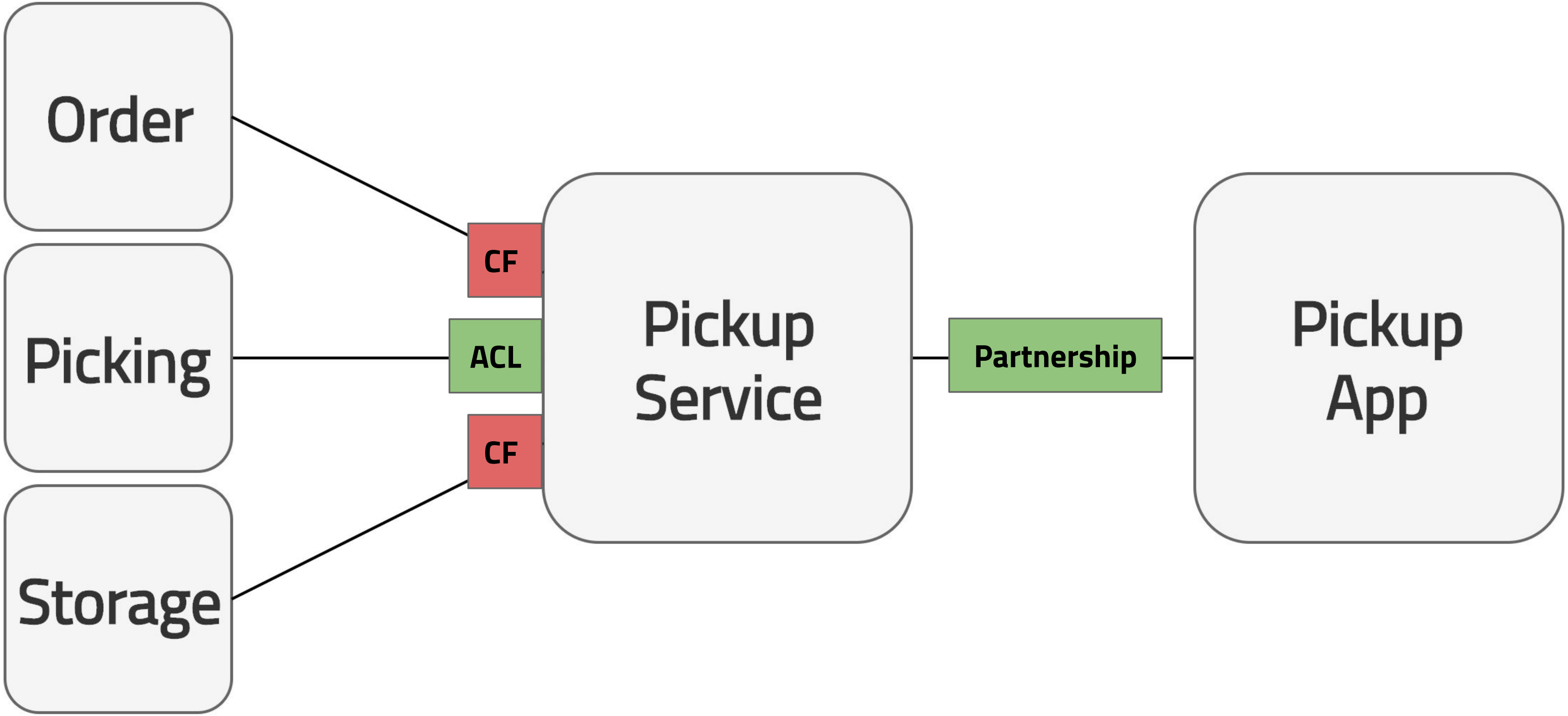
REWE digital

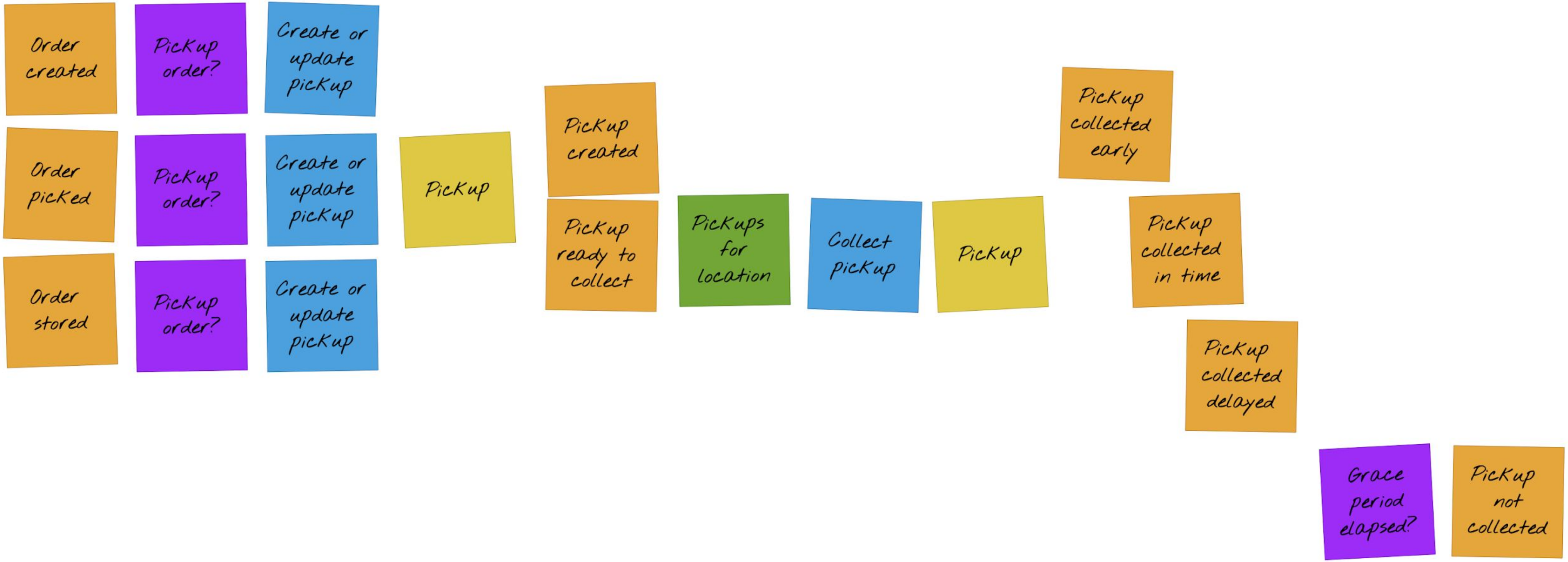
Hexagon Architecture in the real world

1. Domain “Pickup”
2. Prototype Pickup App
3. Problems
4. Enter Hexagon Architecture
5. **Results**

Problem #1

No domain model


Order
Picking
Storage
CF
ACL
CF
Pickup Service
Partnership
Pickup App



Order created

Pickup order?

Create or update pickup

Order picked

Pickup order?

Create or update pickup

Order stored

Pickup order?

Create or update pickup

Pickup

Pickup created

Pickup ready to collect

Pickups for location

Collect pickup

Pickup

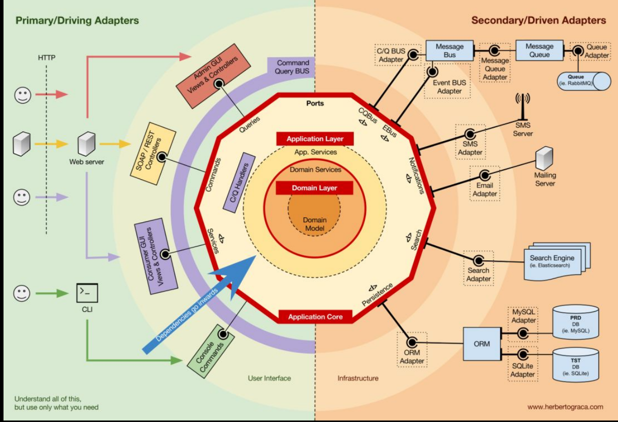
Pickup collected early

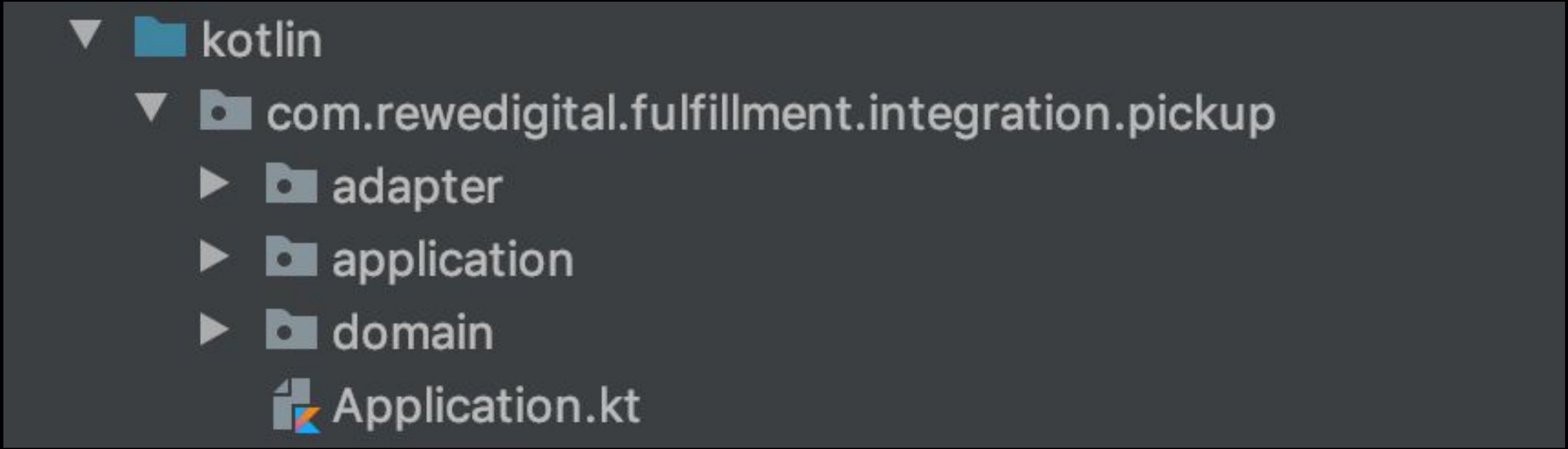
Pickup collected in time

Pickup collected delayed

Grace period elapsed?

Pickup not collected





▼ kotlin

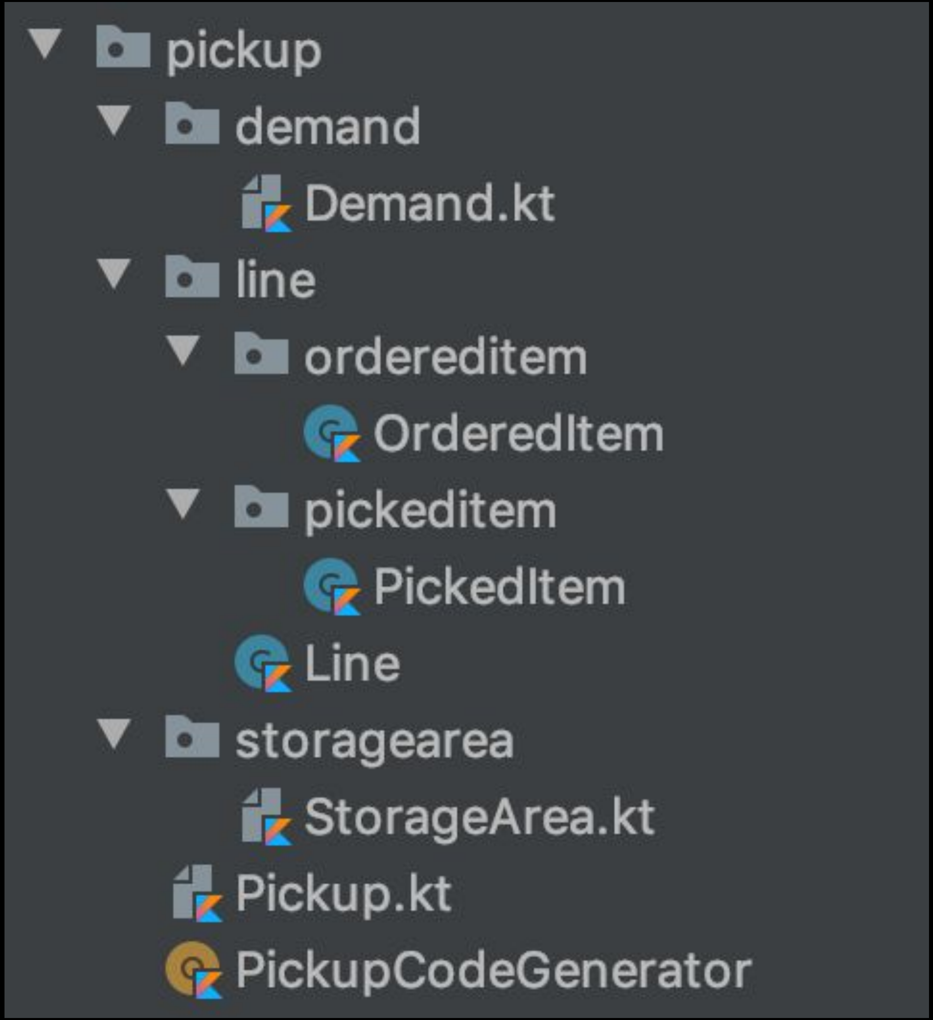
▼ com.rewedigital.fulfillment.integration.pickup

▶ adapter

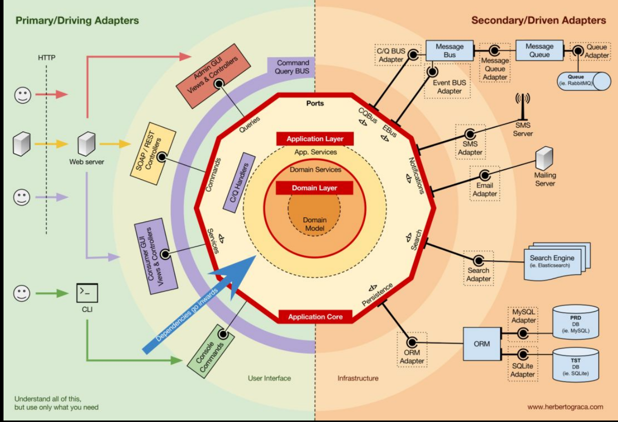
▶ application

▶ domain

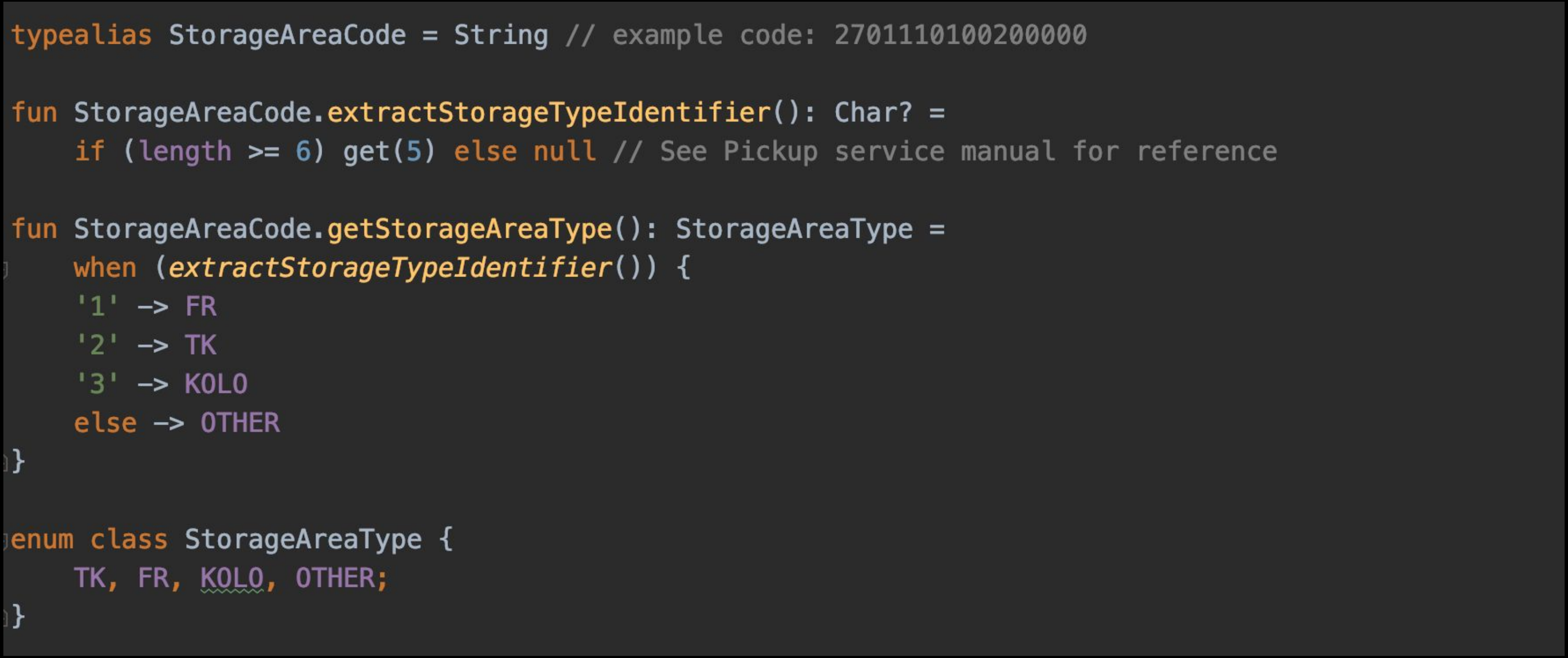
Application.kt



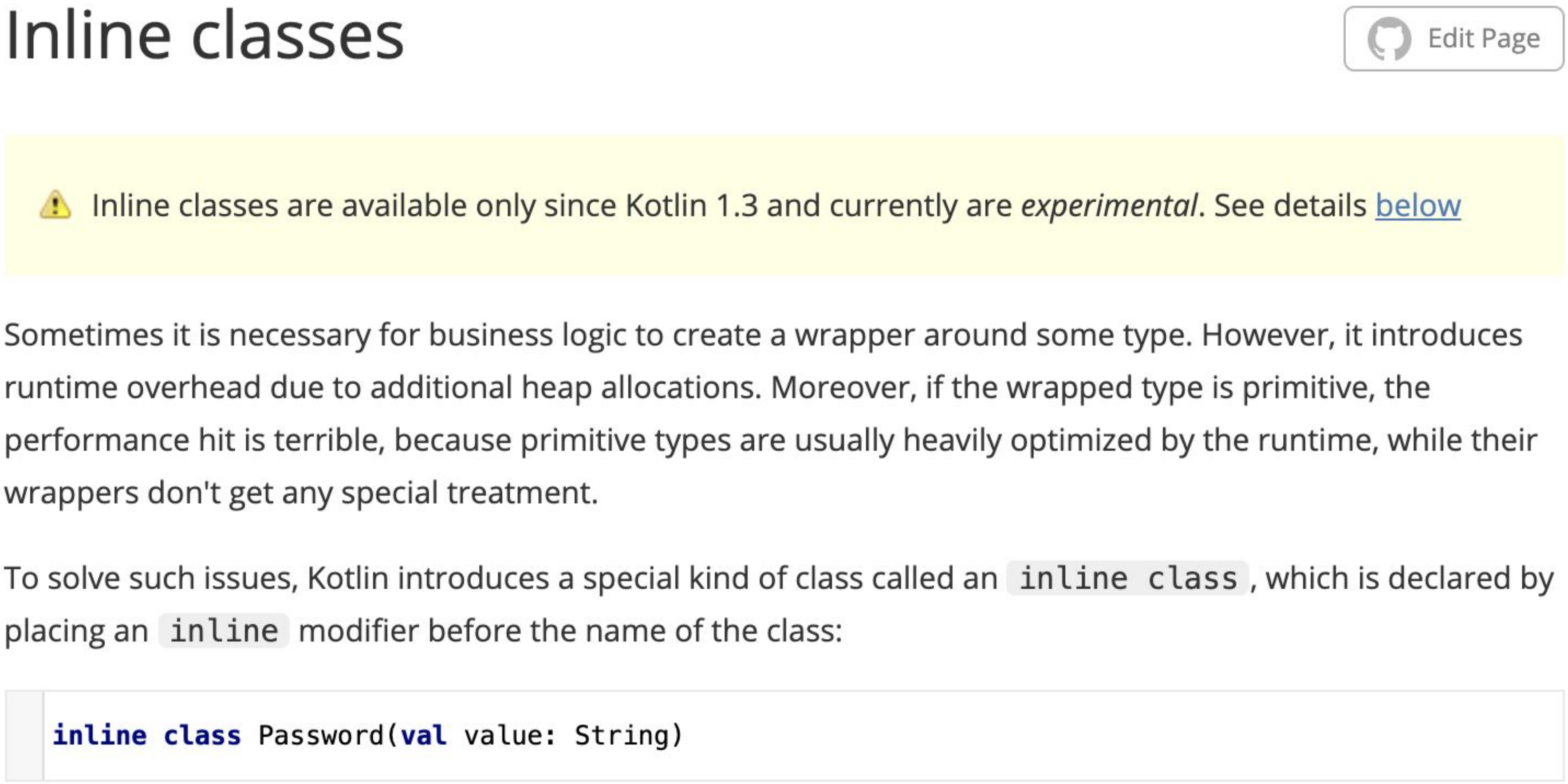
```
pickup
  demand
    Demand.kt
  line
    ordereditem
      OrderedItem
    pickeditem
      PickedItem
    Line
  storagearea
    StorageArea.kt
  Pickup.kt
  PickupCodeGenerator
```



REWE digital



```kotlin
typealias StorageAreaCode = String // example code: 2701110100200000

fun StorageAreaCode.extractStorageTypeIdentifier(): Char? =
    if (length >= 6) get(5) else null // See Pickup service manual for reference

fun StorageAreaCode.getStorageAreaType(): StorageAreaType =
    when (extractStorageTypeIdentifier()) {
    '1' -> FR
    '2' -> TK
    '3' -> KOLO
    else -> OTHER
}

enum class StorageAreaType {
    TK, FR, KOLO, OTHER;
}
```



Inline classes

⚠ Inline classes are available only since Kotlin 1.3 and currently are *experimental*. See details below

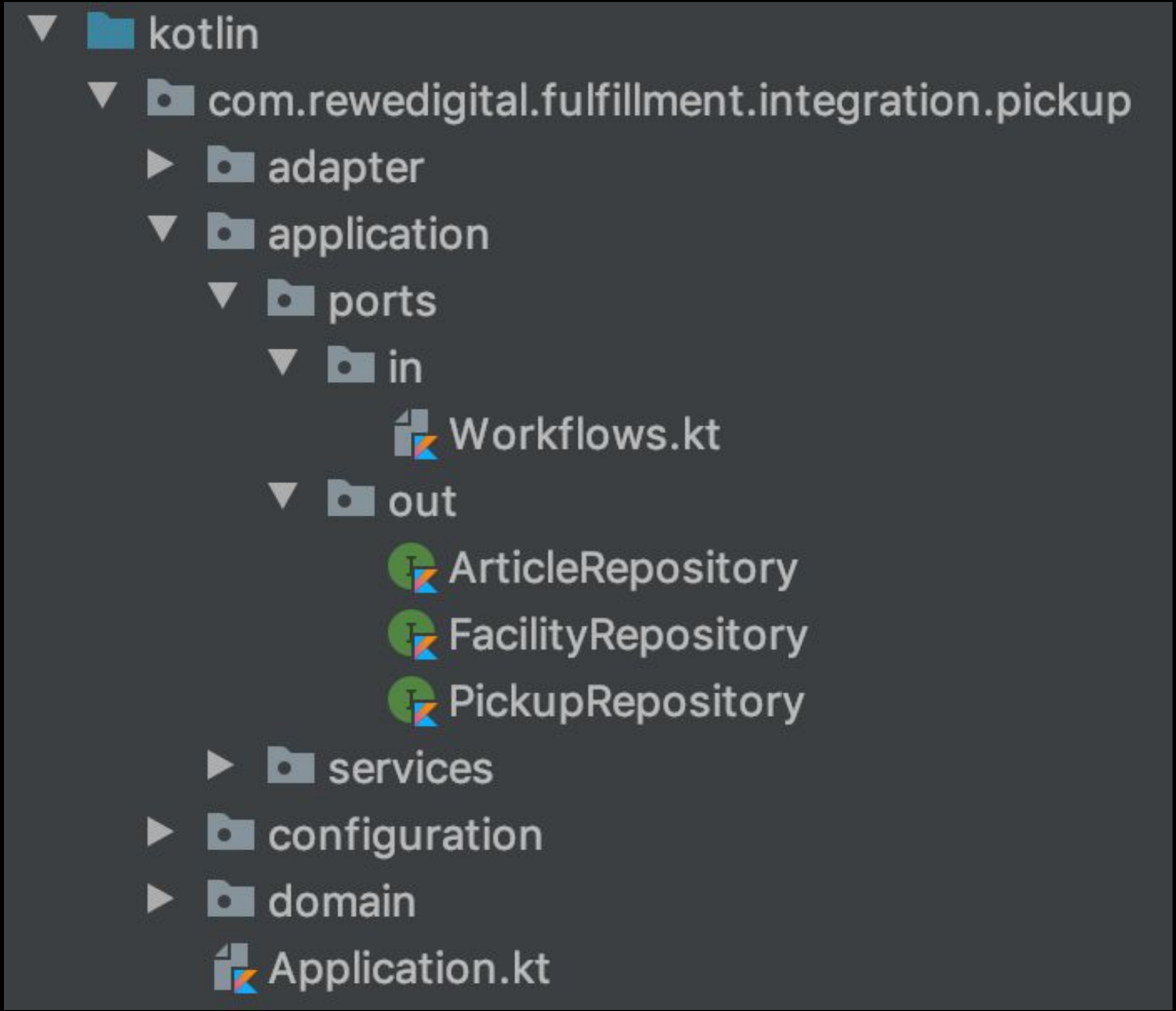
Sometimes it is necessary for business logic to create a wrapper around some type. However, it introduces runtime overhead due to additional heap allocations. Moreover, if the wrapped type is primitive, the performance hit is terrible, because primitive types are usually heavily optimized by the runtime, while their wrappers don't get any special treatment.

To solve such issues, Kotlin introduces a special kind of class called an `inline class`, which is declared by placing an `inline` modifier before the name of the class:

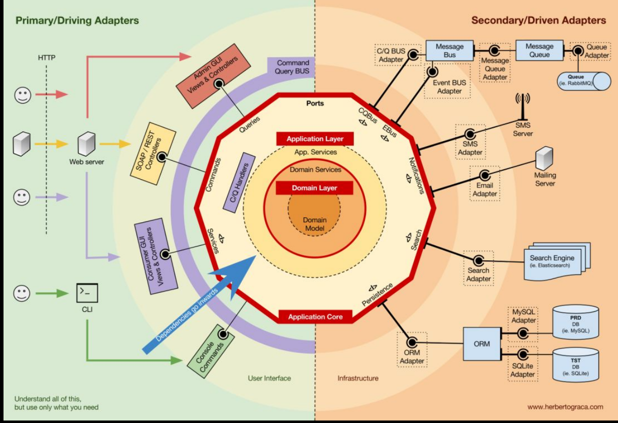
```kotlin
inline class Password(val value: String)
```

Problem #2

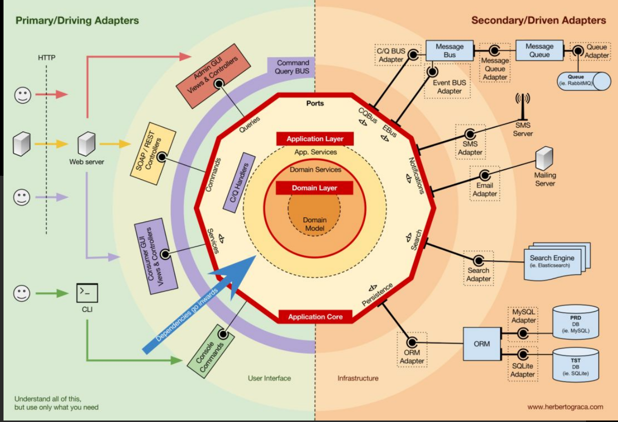
No business metrics

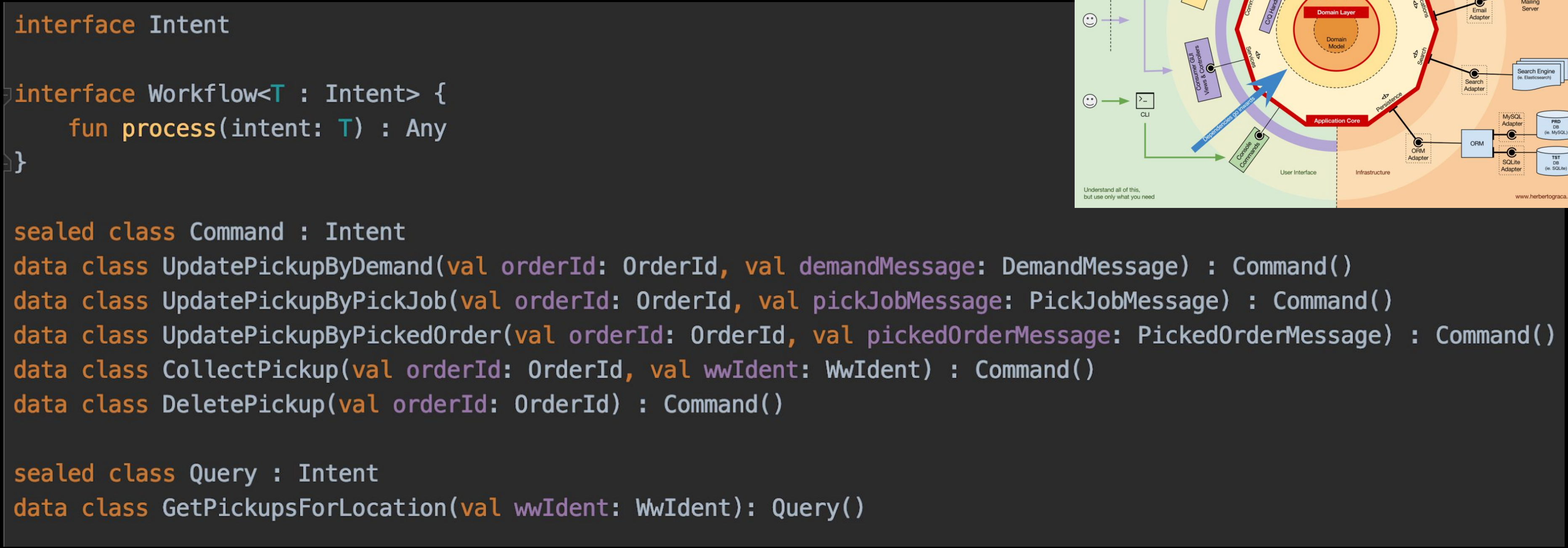


```
▼ kotlin
  ▼ com.rewedigital.fulfillment.integration.pickup
    ▶ adapter
    ▼ application
      ▼ ports
        ▼ in
            Workflows.kt
        ▼ out
            ArticleRepository
            FacilityRepository
            PickupRepository
      ▶ services
    ▶ configuration
    ▶ domain
      Application.kt
```



REWE digital

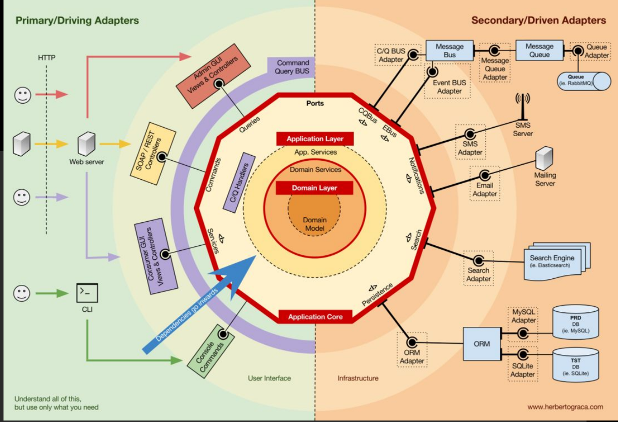




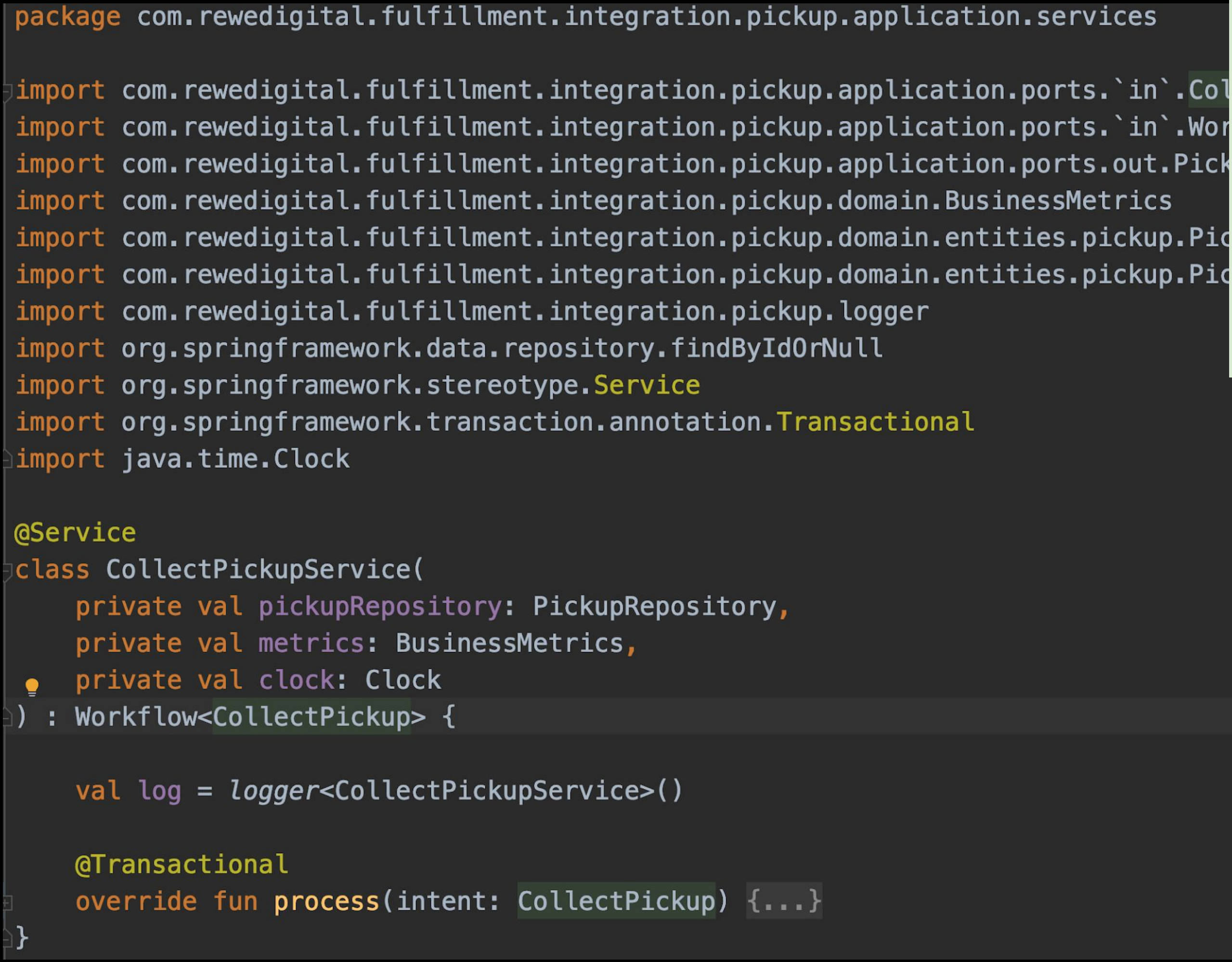
```kotlin
interface Intent

interface Workflow<T : Intent> {
    fun process(intent: T) : Any
}

sealed class Command : Intent
data class UpdatePickupByDemand(val orderId: OrderId, val demandMessage: DemandMessage) : Command()
data class UpdatePickupByPickJob(val orderId: OrderId, val pickJobMessage: PickJobMessage) : Command()
data class UpdatePickupByPickedOrder(val orderId: OrderId, val pickedOrderMessage: PickedOrderMessage) : Command()
data class CollectPickup(val orderId: OrderId, val wwIdent: WwIdent) : Command()
data class DeletePickup(val orderId: OrderId) : Command()

sealed class Query : Intent
data class GetPickupsForLocation(val wwIdent: WwIdent): Query()
```

REWE digital

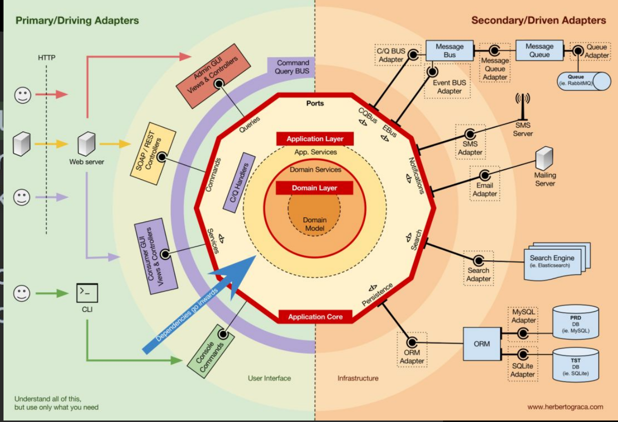
```
@Component
class PickJobMessagePolicy(
    val workflow: Workflow<UpdatePickupByPickJob>
) {

    val log = logger<PickJobMessagePolicy>()

    fun processMessage(pickJobMessage: PickJobMessage) {
        if (pickJobMessage.deliveryType != DELIVERY_TYPE_PICKUP || pickJobMessage.status != STATUS_COMMISSIONED) {
            return
        }
        workflow.process(UpdatePickupByPickJob(pickJobMessage.identifier, pickJobMessage))
    }

    companion object {
        const val DELIVERY_TYPE_PICKUP = "PICKUP"
        const val STATUS_COMMISSIONED = "COMMISSIONED"
    }
}
```



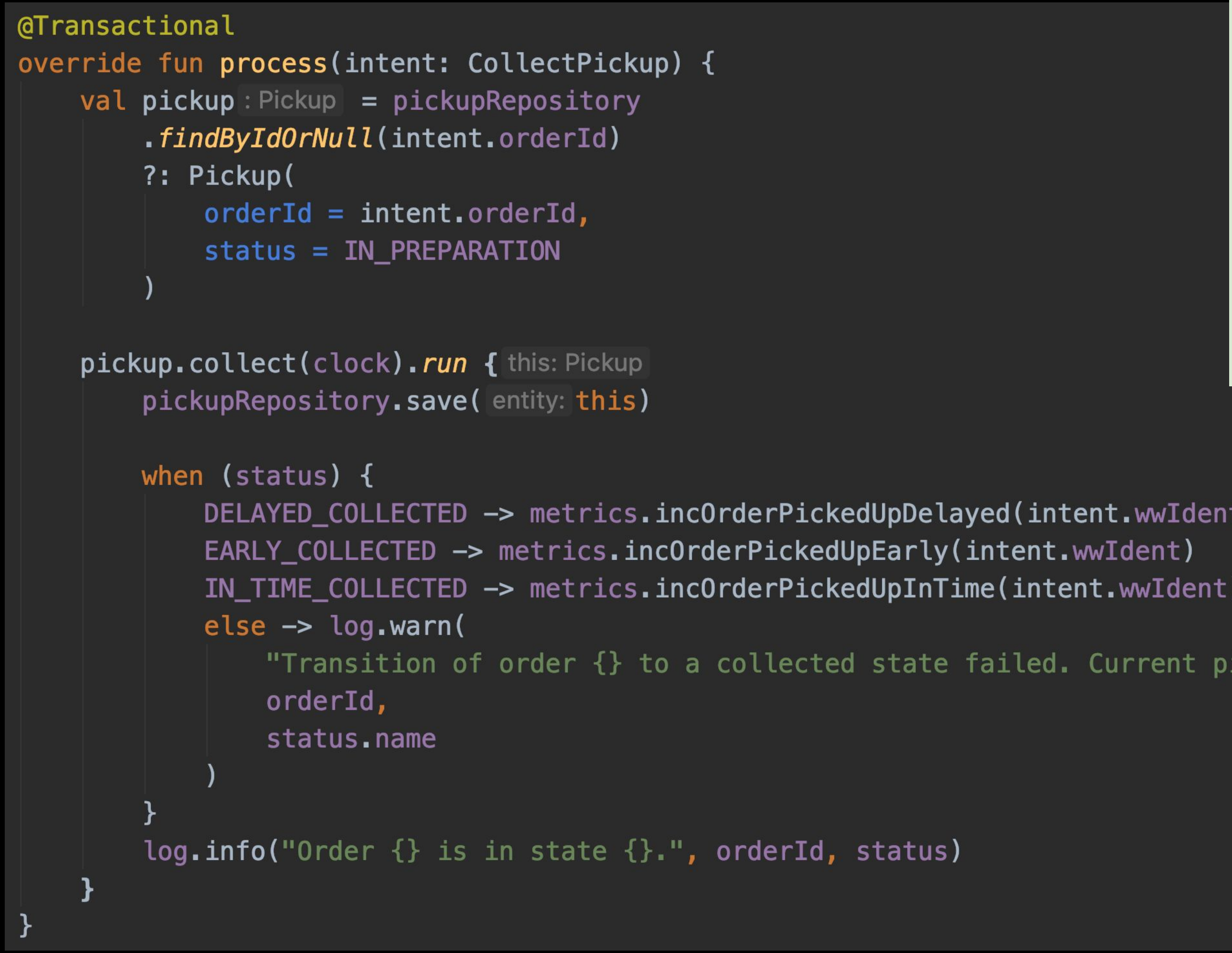
REWE digital



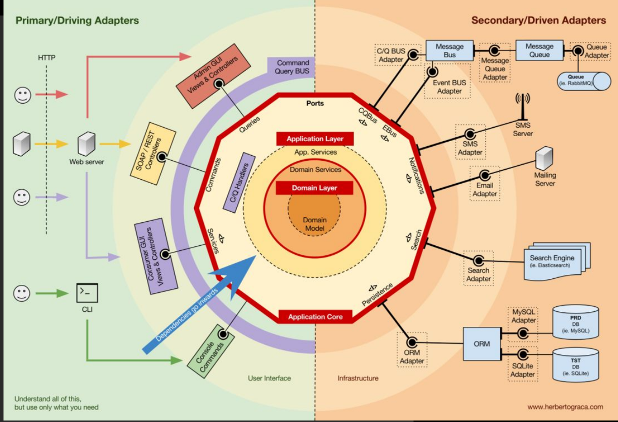
```
package com.rewedigital.fulfillment.integration.pickup.application.services

import com.rewedigital.fulfillment.integration.pickup.application.ports.`in`.Col
import com.rewedigital.fulfillment.integration.pickup.application.ports.`in`.Wor
import com.rewedigital.fulfillment.integration.pickup.application.ports.out.Pick
import com.rewedigital.fulfillment.integration.pickup.domain.BusinessMetrics
import com.rewedigital.fulfillment.integration.pickup.domain.entities.pickup.Pic
import com.rewedigital.fulfillment.integration.pickup.domain.entities.pickup.Pic
import com.rewedigital.fulfillment.integration.pickup.logger
import org.springframework.data.repository.findByIdOrNull
import org.springframework.stereotype.Service
import org.springframework.transaction.annotation.Transactional
import java.time.Clock

@Service
class CollectPickupService(
    private val pickupRepository: PickupRepository,
    private val metrics: BusinessMetrics,
    private val clock: Clock
) : Workflow<CollectPickup> {

    val log = logger<CollectPickupService>()

    @Transactional
    override fun process(intent: CollectPickup) {...}
}
```



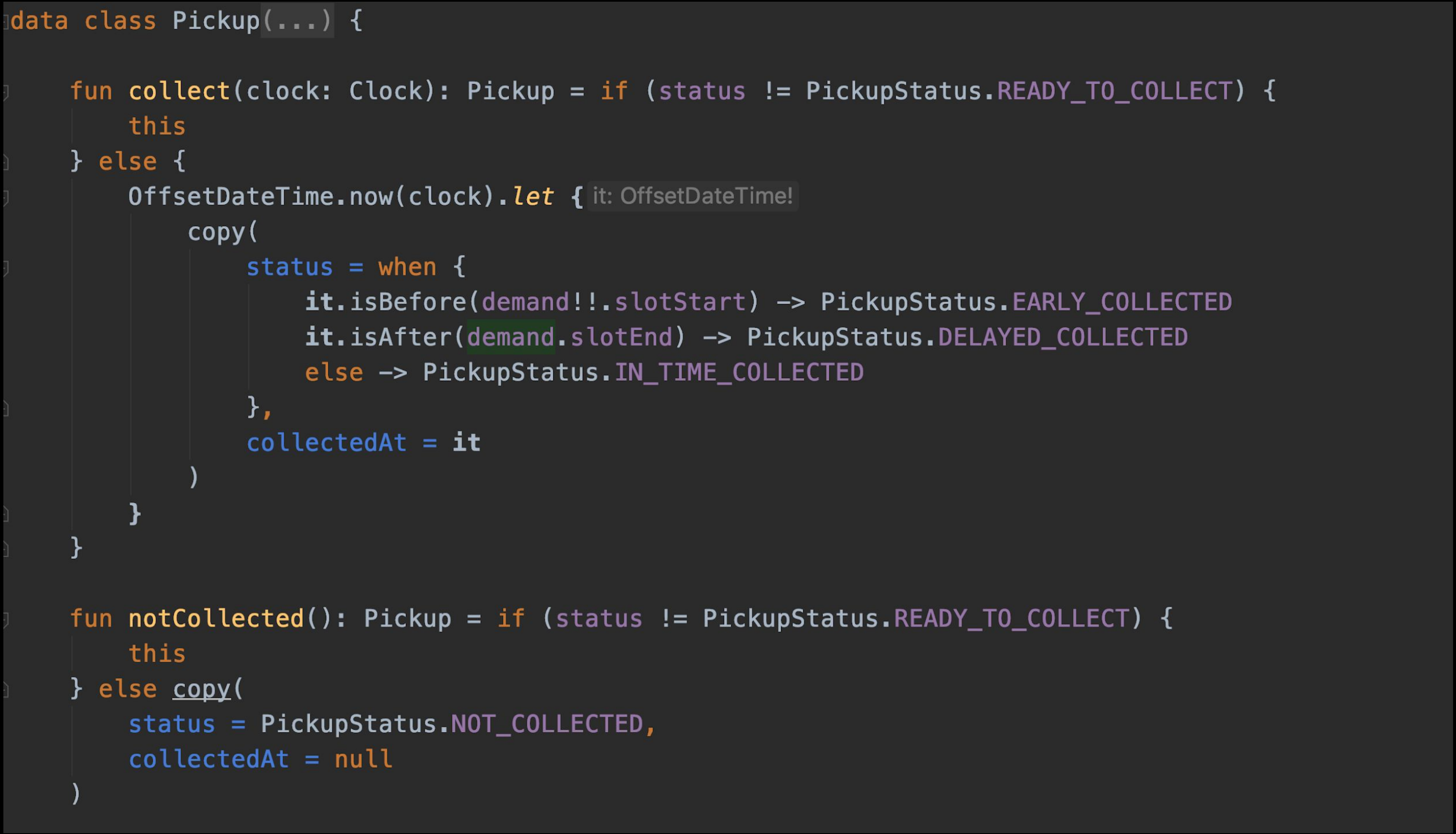
REWE digital



```
@Transactional
override fun process(intent: CollectPickup) {
    val pickup : Pickup = pickupRepository
        .findByIdOrNull(intent.orderId)
        ?: Pickup(
            orderId = intent.orderId,
            status = IN_PREPARATION
        )

    pickup.collect(clock).run { this: Pickup
        pickupRepository.save( entity: this)

        when (status) {
            DELAYED_COLLECTED -> metrics.incOrderPickedUpDelayed(intent.wwIdent)
            EARLY_COLLECTED -> metrics.incOrderPickedUpEarly(intent.wwIdent)
            IN_TIME_COLLECTED -> metrics.incOrderPickedUpInTime(intent.wwIdent)
            else -> log.warn(
                "Transition of order {} to a collected state failed. Current pickup state is {}",
                orderId,
                status.name
            )
        }
        log.info("Order {} is in state {}.", orderId, status)
    }
}
```



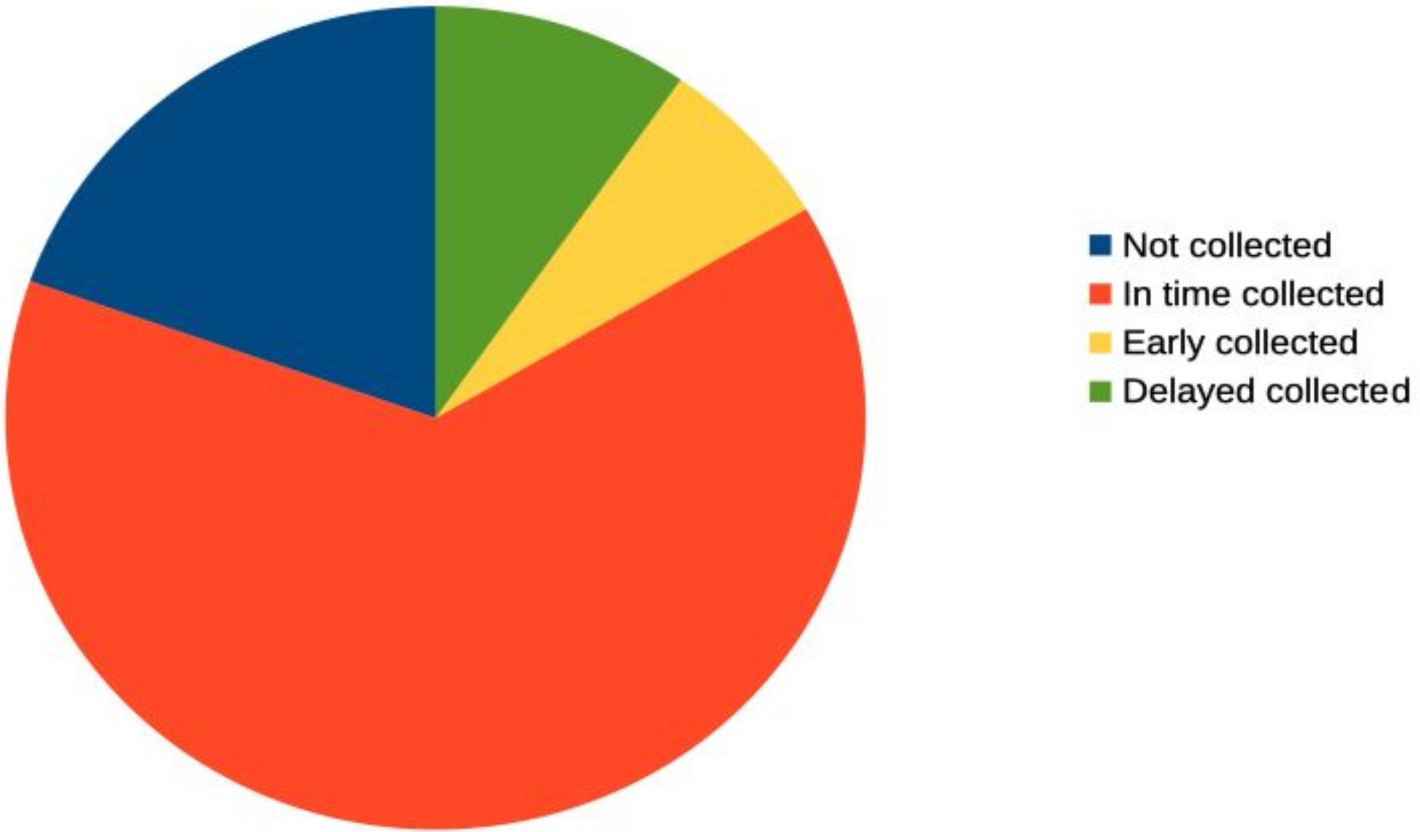
REWE digital



```
data class Pickup(...) {

    fun collect(clock: Clock): Pickup = if (status != PickupStatus.READY_TO_COLLECT) {
        this
    } else {
        OffsetDateTime.now(clock).let { it: OffsetDateTime!
            copy(
                status = when {
                    it.isBefore(demand!!.slotStart) -> PickupStatus.EARLY_COLLECTED
                    it.isAfter(demand.slotEnd) -> PickupStatus.DELAYED_COLLECTED
                    else -> PickupStatus.IN_TIME_COLLECTED
                },
                collectedAt = it
            )
        }
    }

    fun notCollected(): Pickup = if (status != PickupStatus.READY_TO_COLLECT) {
        this
    } else copy(
        status = PickupStatus.NOT_COLLECTED,
        collectedAt = null
    )
```

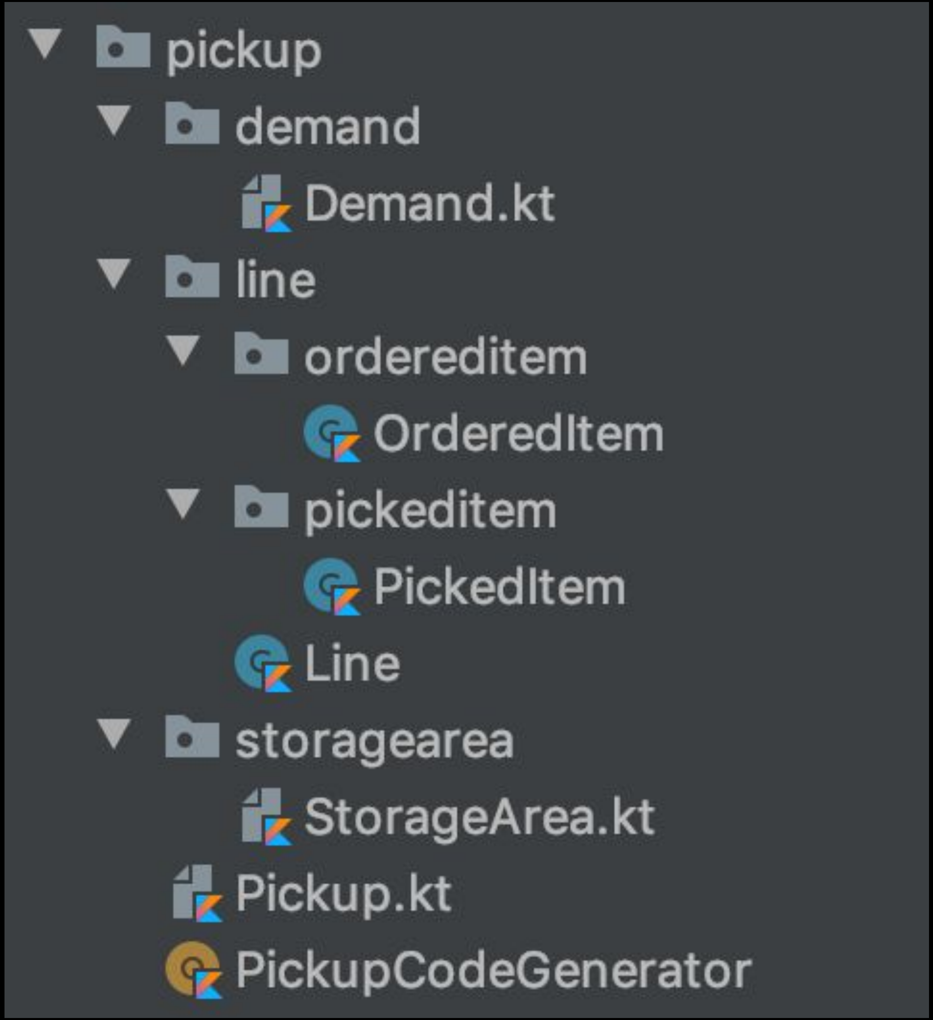
Pickup collected status

Since 12/2019

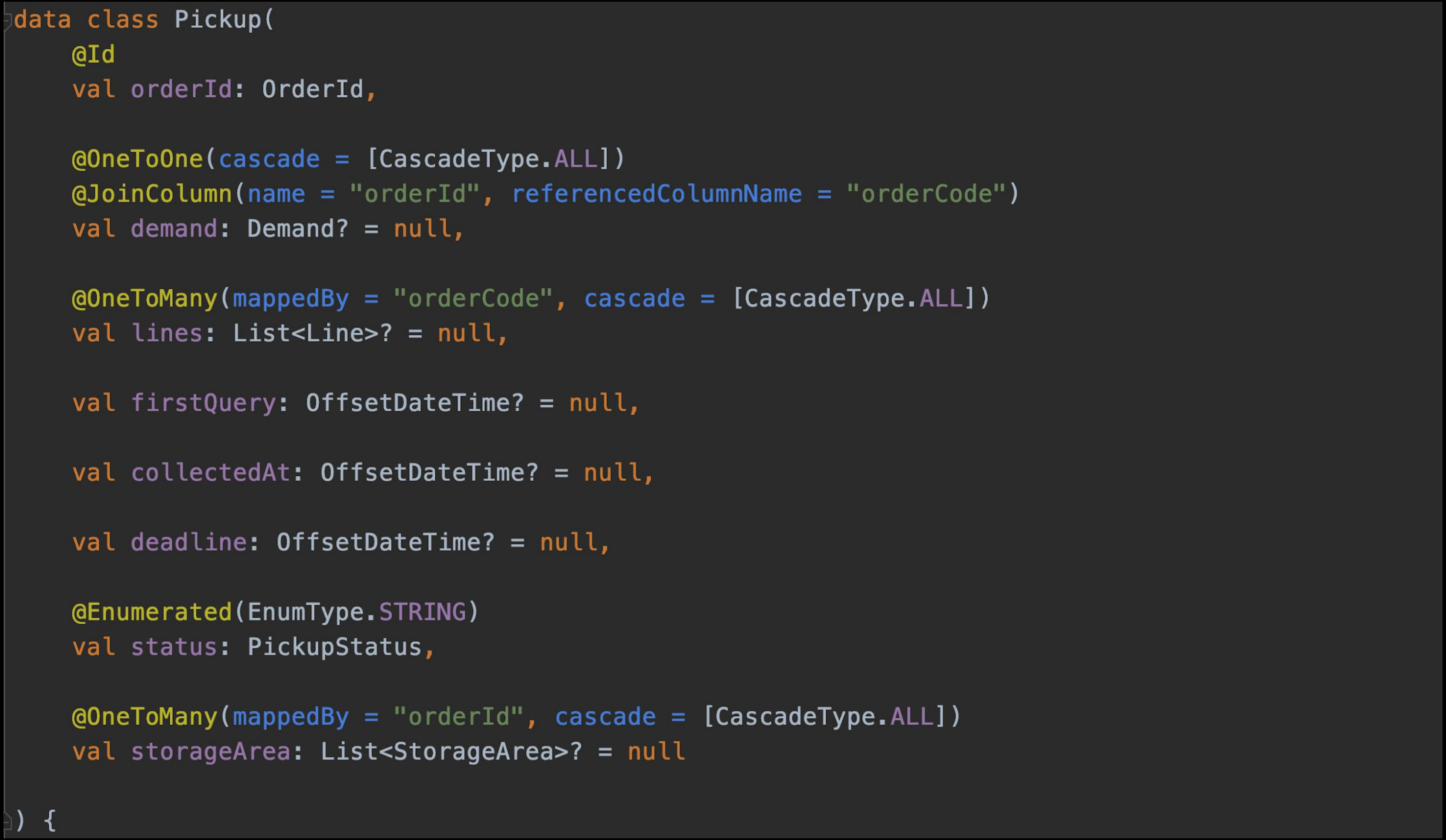


Problem #3

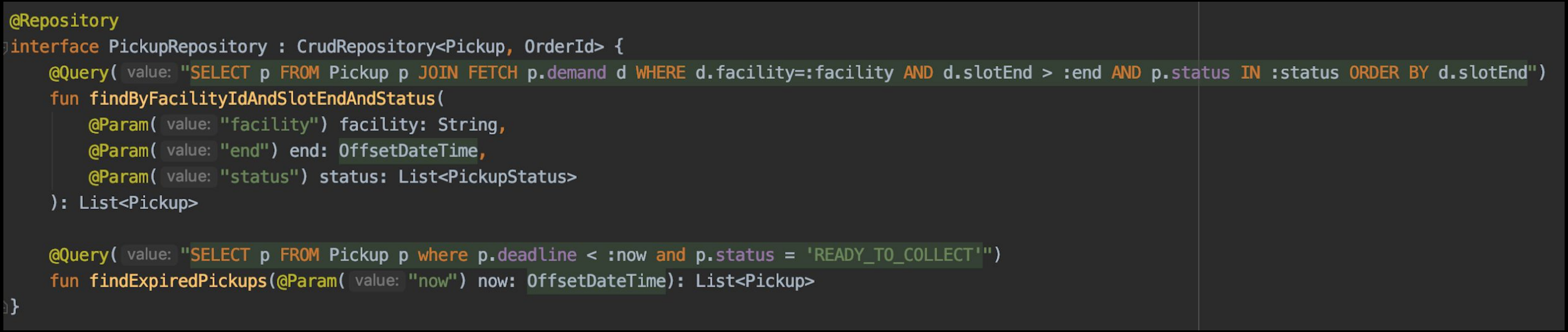
Performance



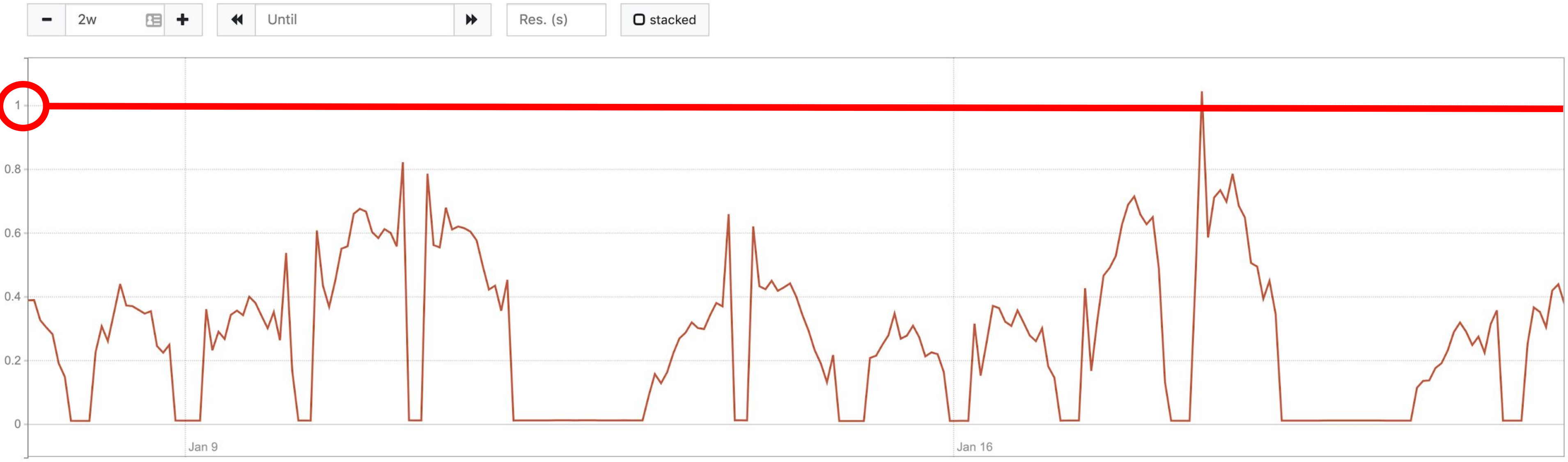
```
▼ pickup
  ▼ demand
      Demand.kt
  ▼ line
    ▼ ordereditem
        OrderedItem
    ▼ pickeditem
        PickedItem
      Line
  ▼ storagearea
      StorageArea.kt
    Pickup.kt
    PickupCodeGenerator
```

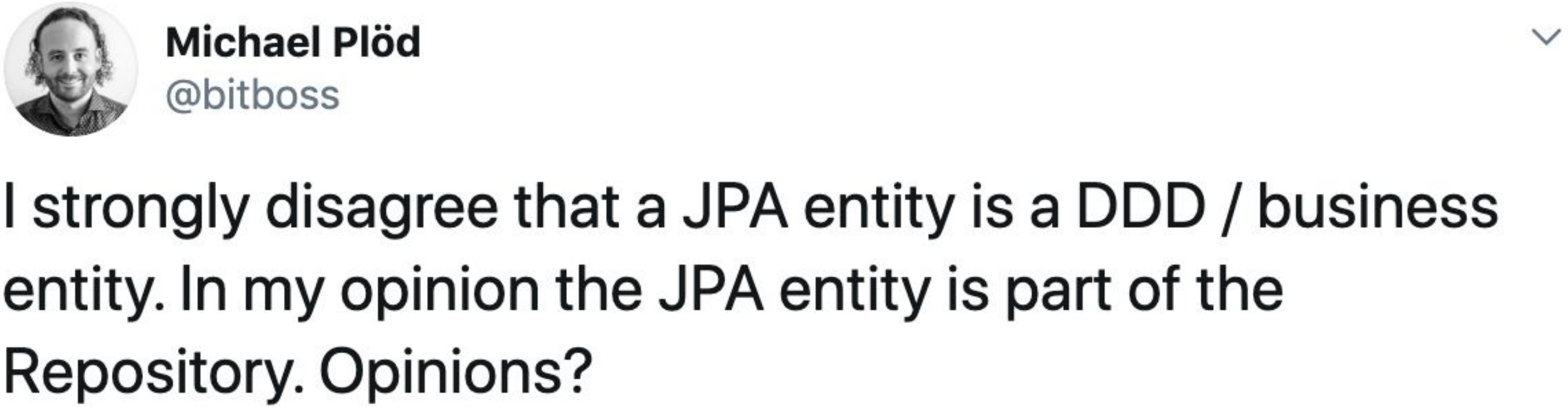


```kotlin
data class Pickup(
    @Id
    val orderId: OrderId,

    @OneToOne(cascade = [CascadeType.ALL])
    @JoinColumn(name = "orderId", referencedColumnName = "orderCode")
    val demand: Demand? = null,

    @OneToMany(mappedBy = "orderCode", cascade = [CascadeType.ALL])
    val lines: List<Line>? = null,

    val firstQuery: OffsetDateTime? = null,

    val collectedAt: OffsetDateTime? = null,

    val deadline: OffsetDateTime? = null,

    @Enumerated(EnumType.STRING)
    val status: PickupStatus,

    @OneToMany(mappedBy = "orderId", cascade = [CascadeType.ALL])
    val storageArea: List<StorageArea>? = null

) {
```



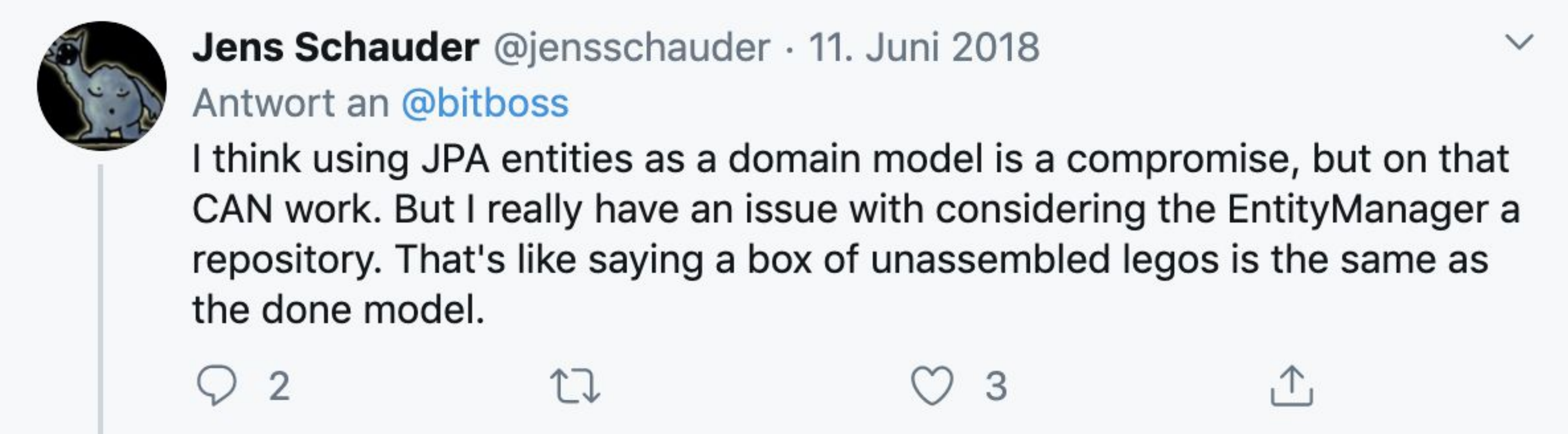
```
@Repository
interface PickupRepository : CrudRepository<Pickup, OrderId> {
    @Query( value: "SELECT p FROM Pickup p JOIN FETCH p.demand d WHERE d.facility=:facility AND d.slotEnd > :end AND p.status IN :status ORDER BY d.slotEnd")
    fun findByFacilityIdAndSlotEndAndStatus(
        @Param( value: "facility") facility: String,
        @Param( value: "end") end: OffsetDateTime,
        @Param( value: "status") status: List<PickupStatus>
    ): List<Pickup>

    @Query( value: "SELECT p FROM Pickup p where p.deadline < :now and p.status = 'READY_TO_COLLECT'")
    fun findExpiredPickups(@Param( value: "now") now: OffsetDateTime): List<Pickup>
}
```


−
2w
+
Until
Res. (s)
stacked
1
0.8
0.6
0.4
0.2
0
Jan 9
Jan 16



Michael Plöd
@bitboss

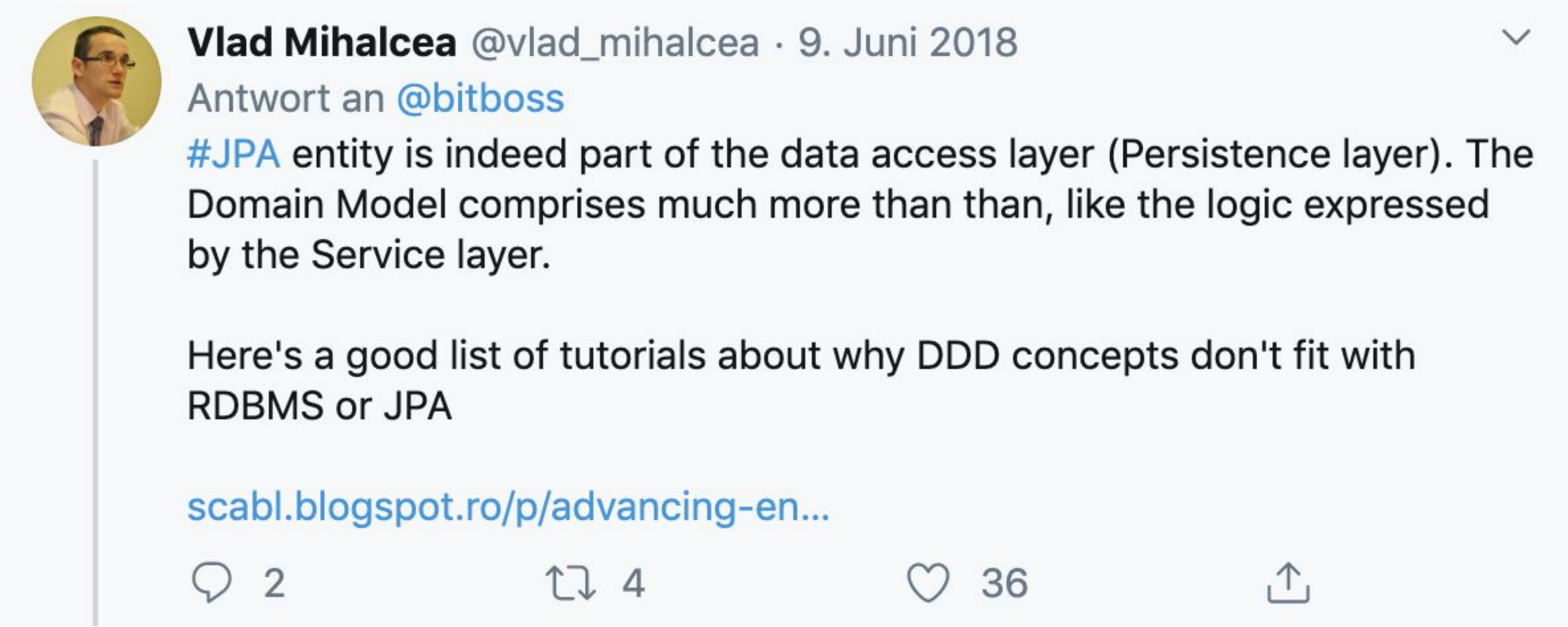
I strongly disagree that a JPA entity is a DDD / business entity. In my opinion the JPA entity is part of the Repository. Opinions?



Jens Schauder @jensschauder · 11. Juni 2018

Antwort an @bitboss

I think using JPA entities as a domain model is a compromise, but on that CAN work. But I really have an issue with considering the EntityManager a repository. That's like saying a box of unassembled legos is the same as the done model.



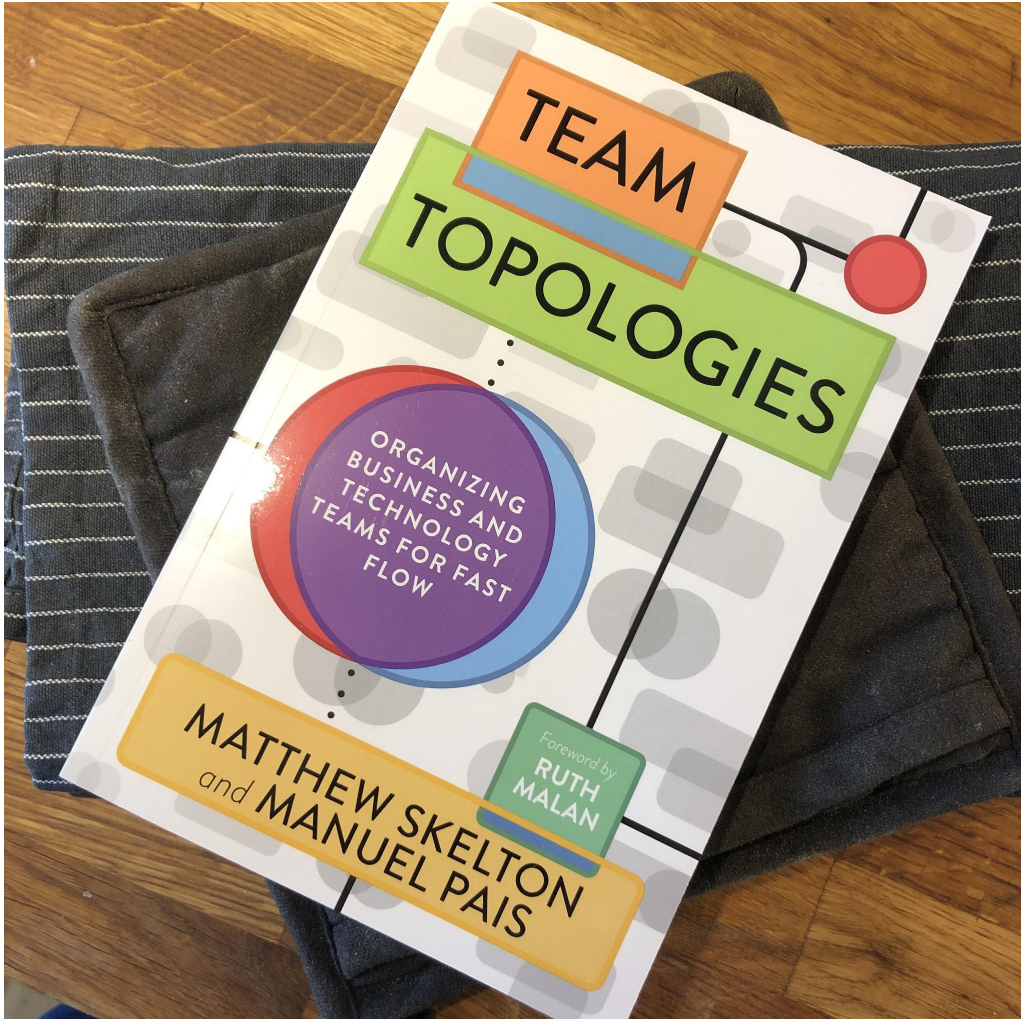
Vlad Mihalcea @vlad_mihalcea · 9. Juni 2018

Antwort an @bitboss

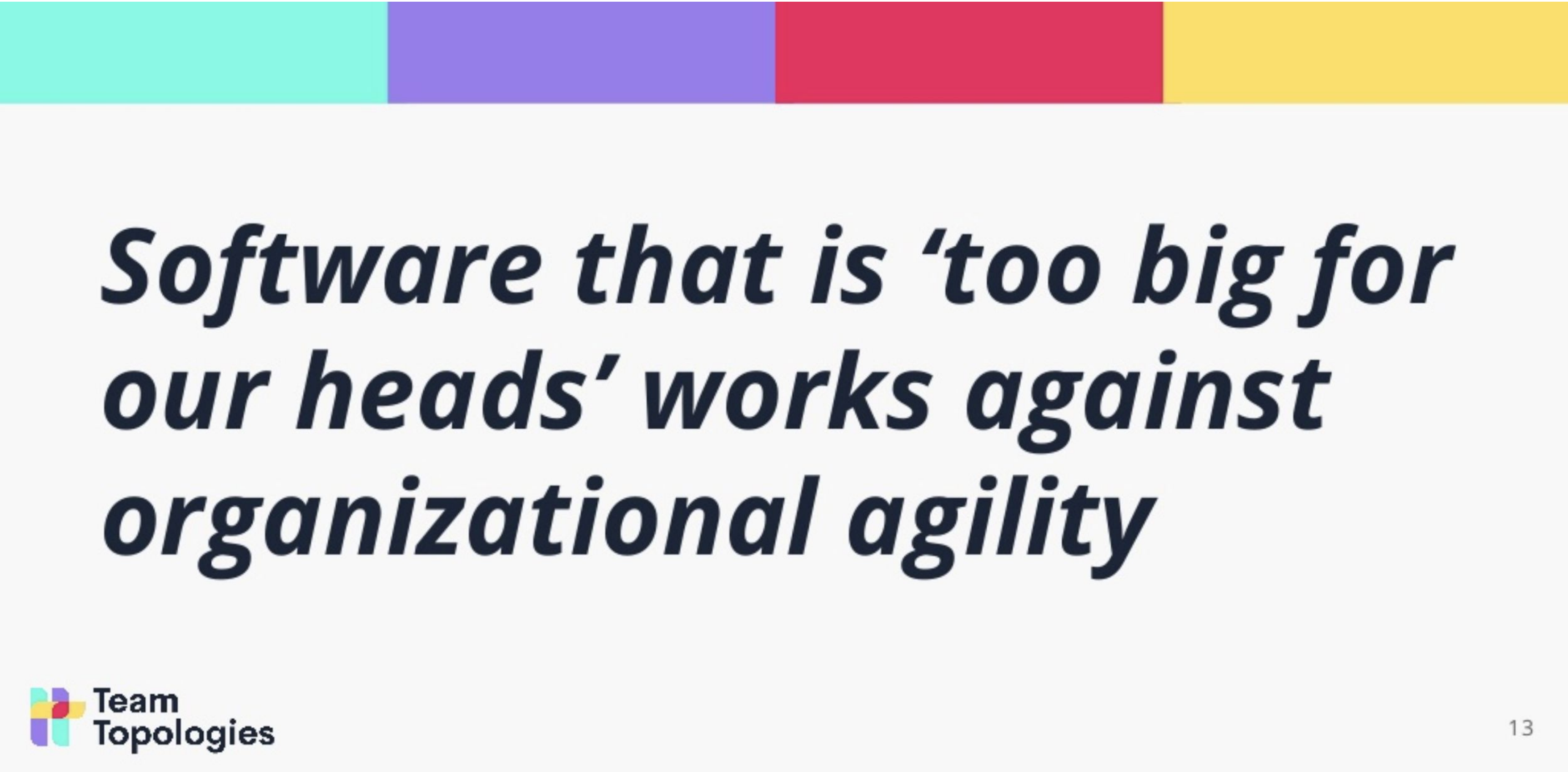
#JPA entity is indeed part of the data access layer (Persistence layer). The Domain Model comprises much more than than, like the logic expressed by the Service layer.

Here's a good list of tutorials about why DDD concepts don't fit with RDBMS or JPA

scabl.blogspot.ro/p/advancing-en...

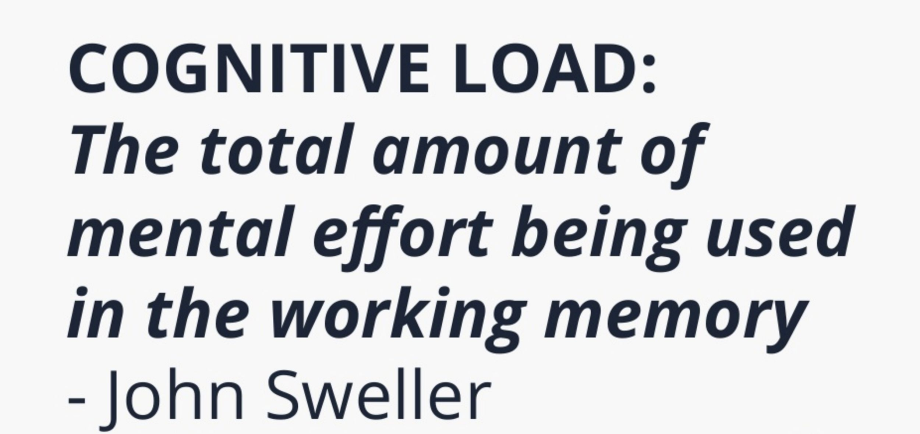

TEAM
TOPOLOGIES
ORGANIZING BUSINESS AND TECHNOLOGY TEAMS FOR FAST FLOW
Foreword by RUTH MALAN
MATTHEW SKELTON and MANUEL PAIS

https://medium.com/@pia.fak.sunnanbo/going-to-jfokus-2020-522fbf141dcb

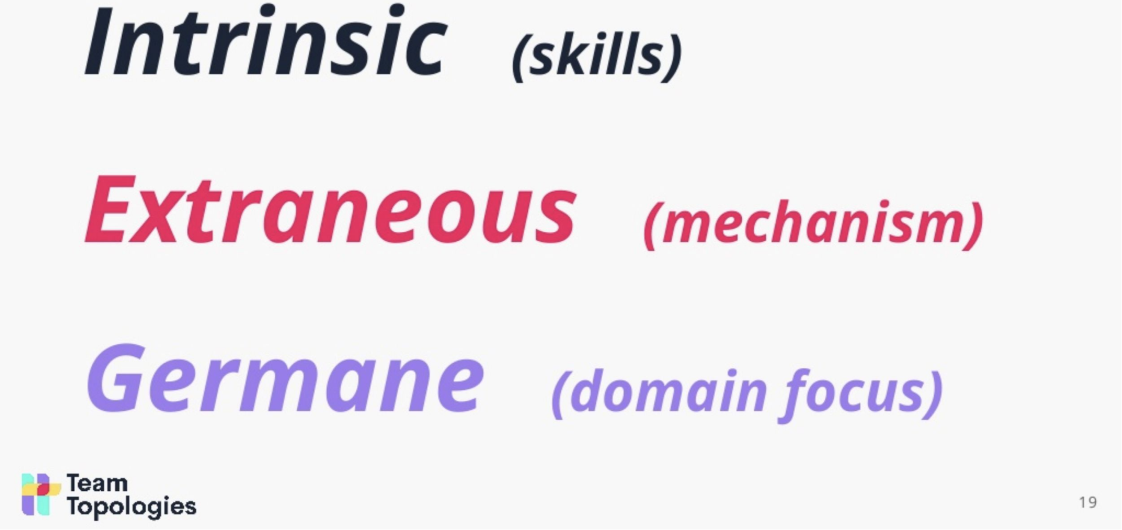


Software that is 'too big for our heads' works against organizational agility

Team Topologies



COGNITIVE LOAD:
The total amount of mental effort being used in the working memory
- John Sweller

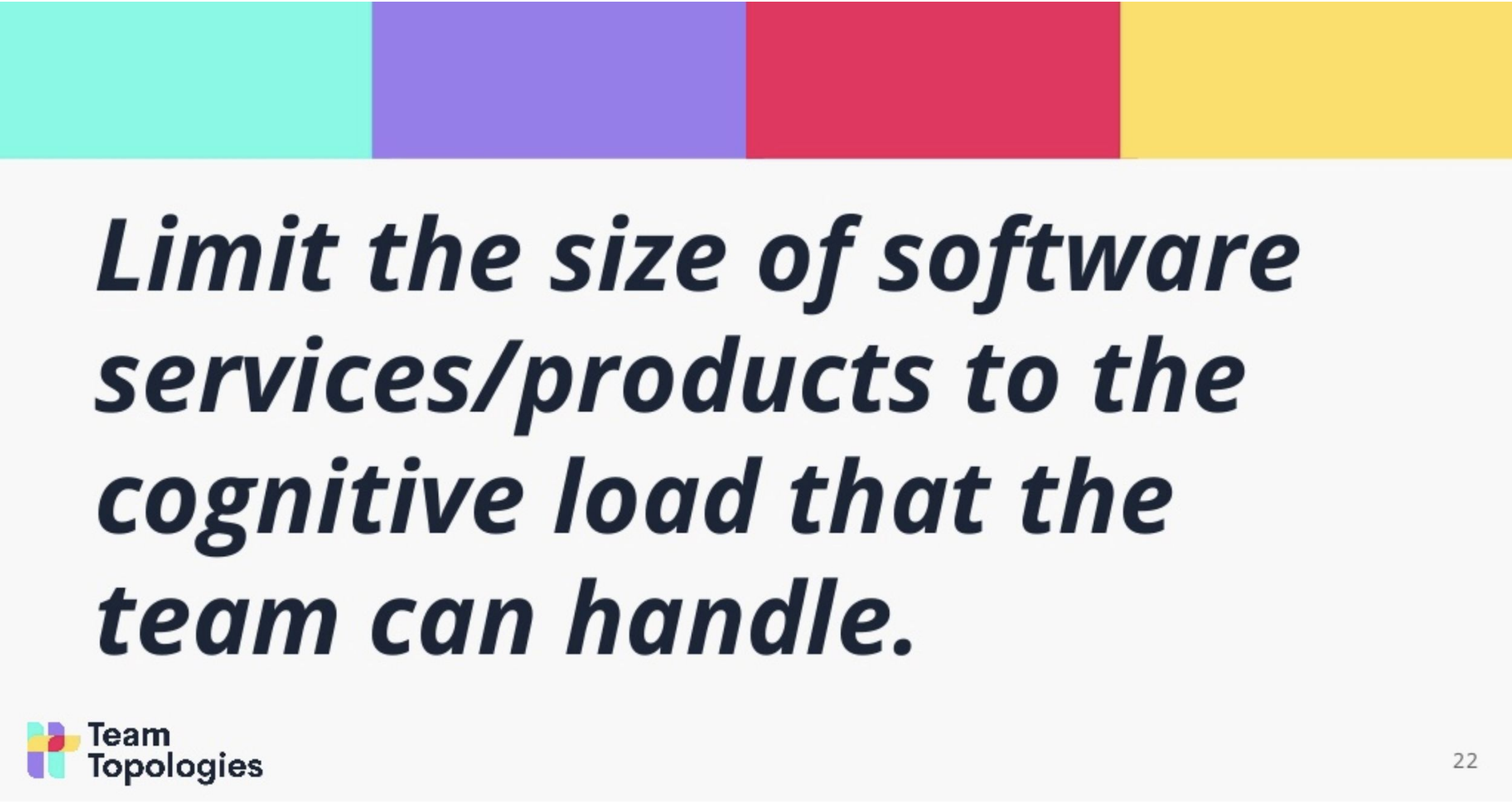


Intrinsic ***(skills)***

Extraneous ***(mechanism)***

Germane ***(domain focus)***

Team Topologies



Limit the size of software services/products to the cognitive load that the team can handle.

Team Topologies

Thank you :)

Christoph Baudson / @sustainablepace