



DDD workshop dag 1

Daniël Mertens

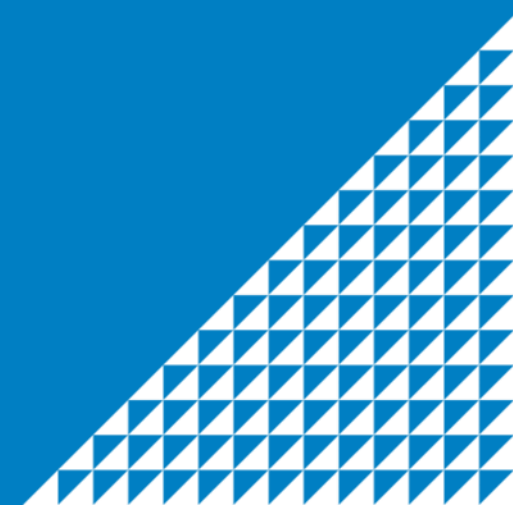
daniel.mertens@infosupport.com

▲ Slides en code

Go to: <https://bit.ly/PXL-DDD-2024>



Herhaling dag 1

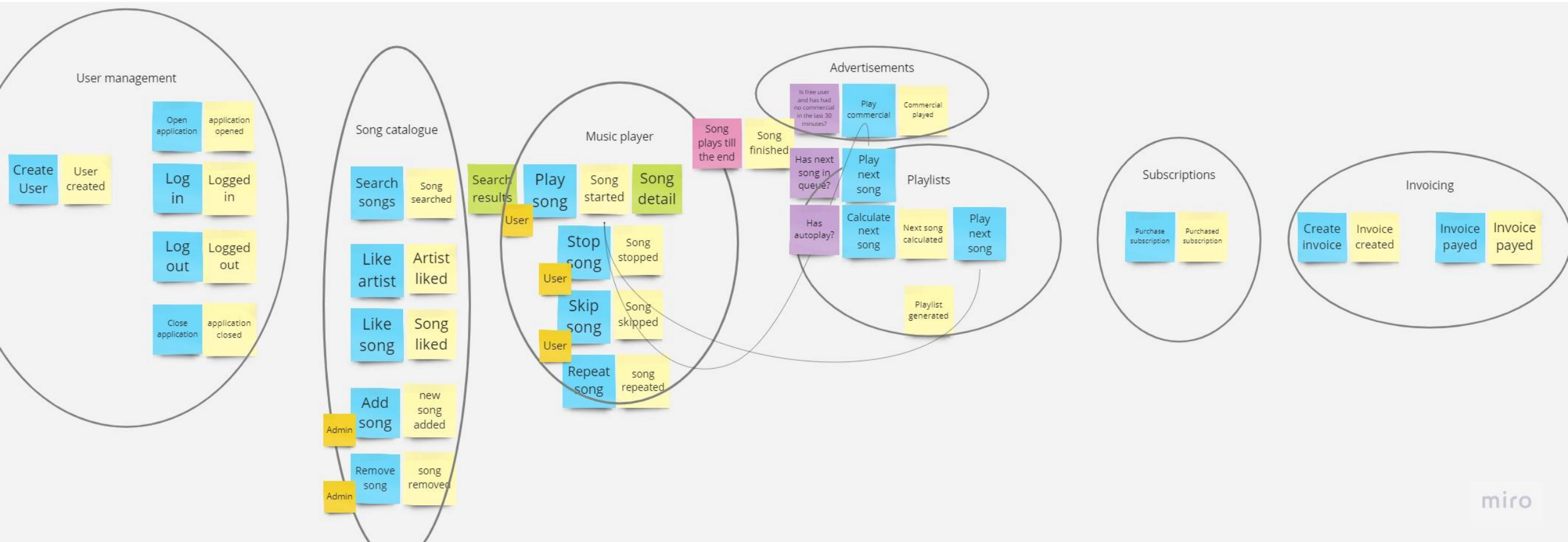


▲ Tactical vs Strategic

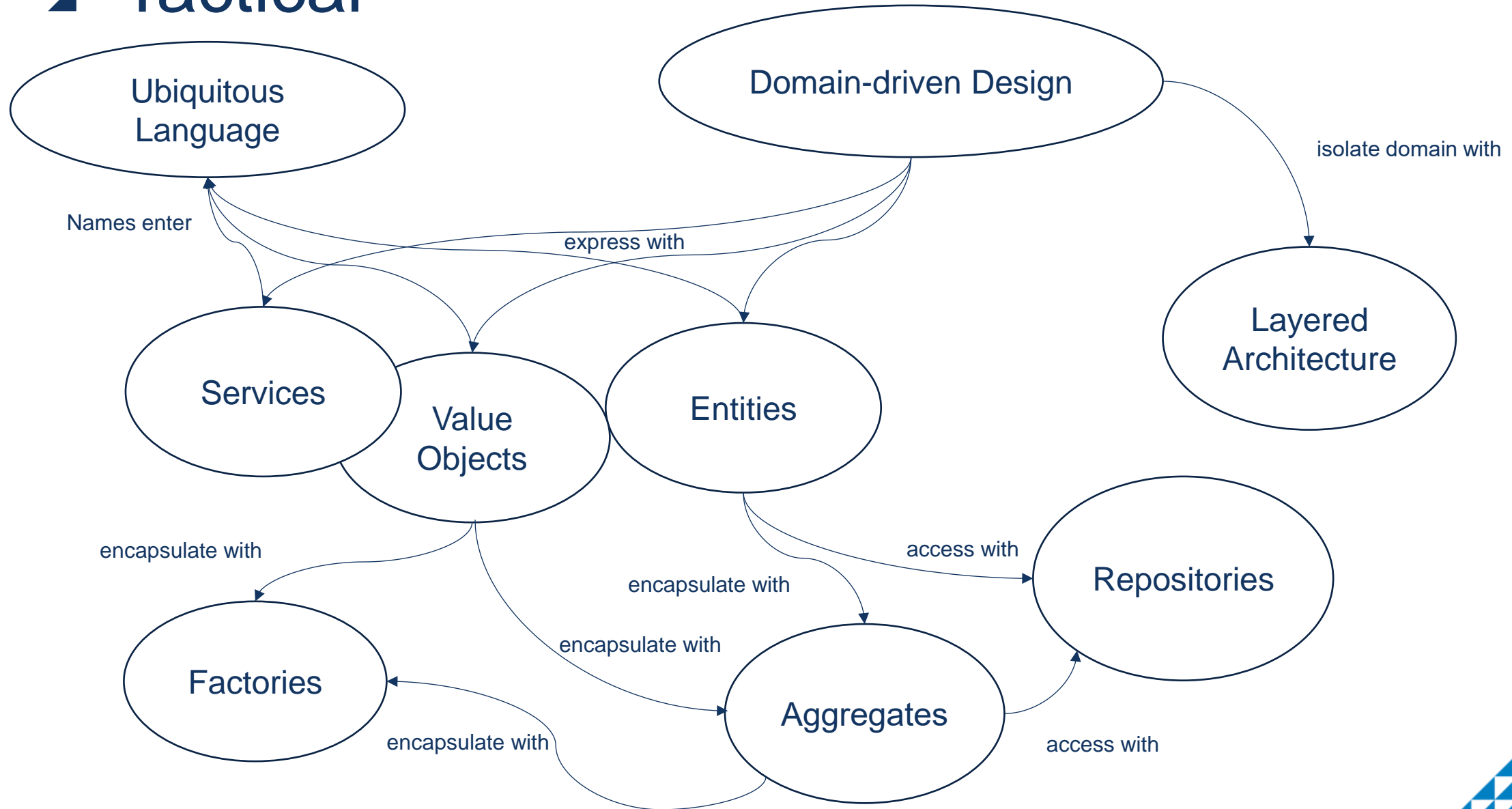
STRATEGIC

TACTICAL





▲ Tactical





▲ Value objects

- “Represent a descriptive aspect of the domain with no conceptual identity.”
- Immutable
- Distinguishable only on value
- Easy to use



▲ Entity

- “Objects which have continuity through a life cycle and are being distinct independent of its attribute values over time.”
- Mutable
- Has a unique id in the domain
- More complicated to use

▲ Aggregates

- “A cluster of associated objects that is treated as a unit for the purpose of data changes.”
- Zijn verantwoordelijk voor hun eigen consistency.
- Een verzameling van Entities en Value Objects.
- Er is exact 1 Aggregate Root (entity)
 - Verantwoordelijk voor de consistency
 - Enige toegang voor changes

▲ Factories

- “Aggregates can become complex object graphs of which the construction should be the responsibility of factories.”
- Factories worden gebruikt wanneer normale creatie van een value object of entity to complex wordt.
- Gebruikt vaak DI om services aan te roepen tijdens creatie.





▲ Repositories

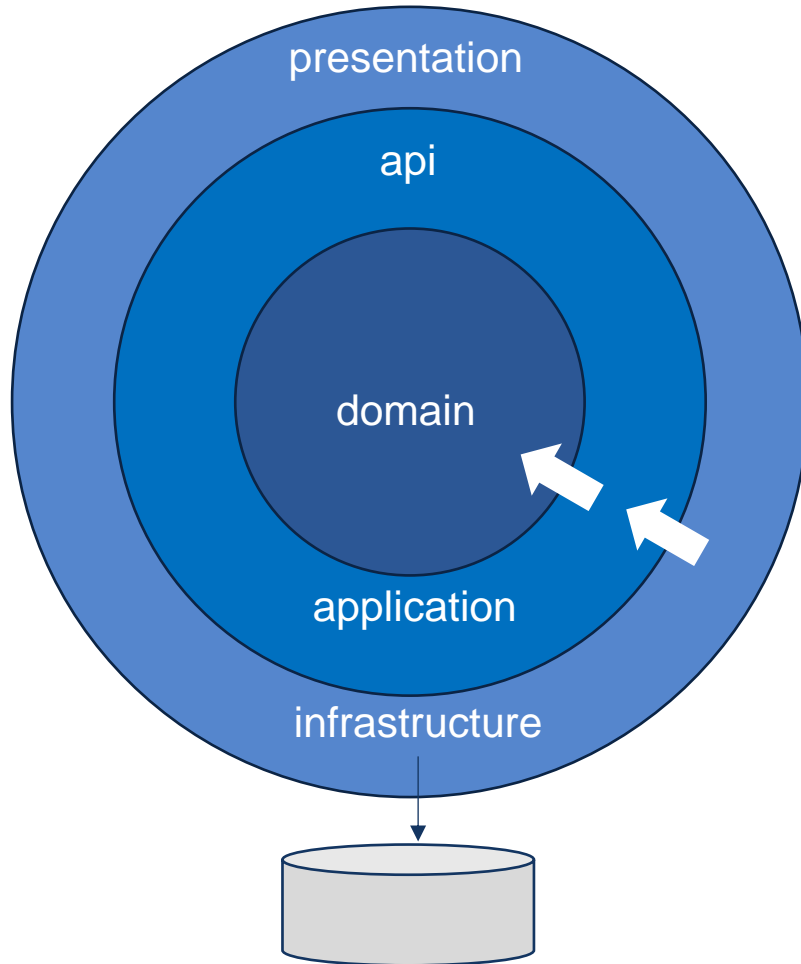
- “Represent all objects of a certain type as a collection with querying capabilities.”
- Houdt entities bij.
- Kan connecteren met
 - Database
 - Cache
 - Text file
 - ...



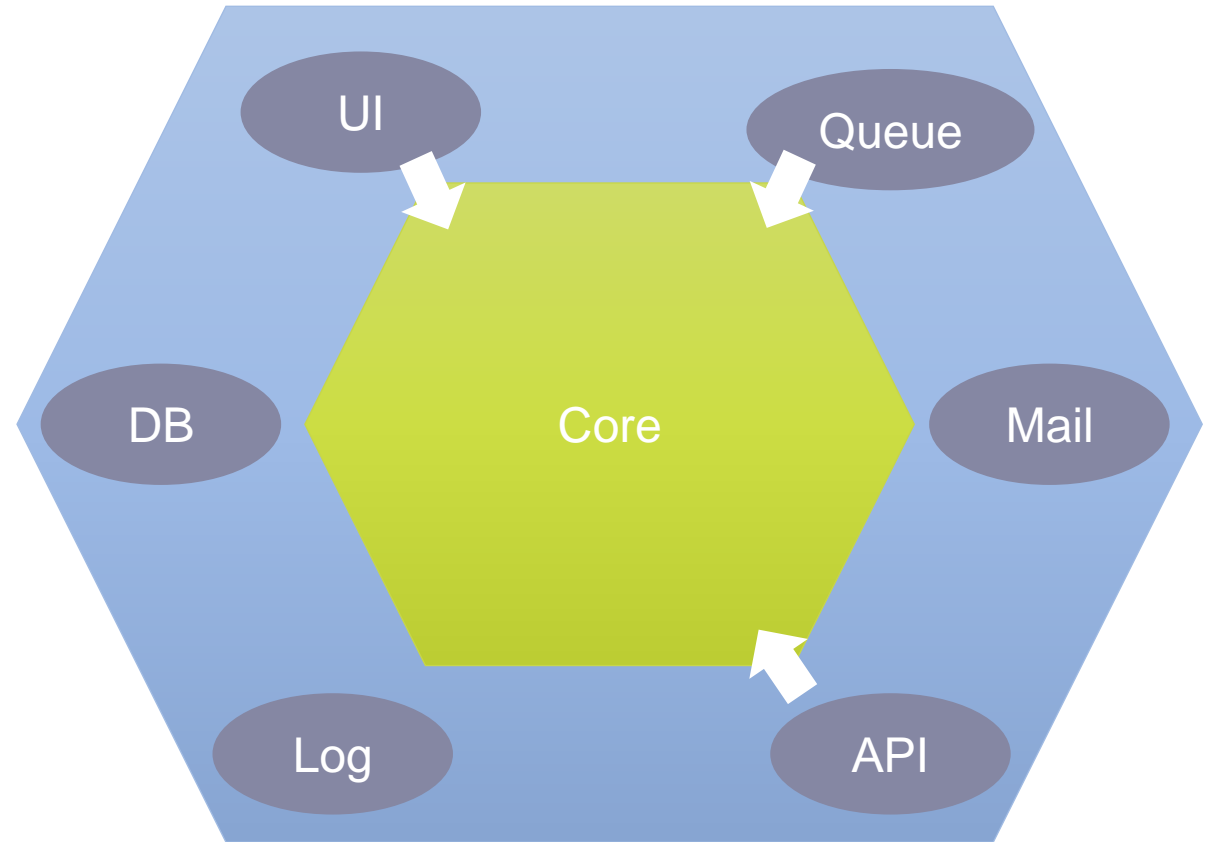
▲ Services

- “A process or transformation in the domain which is not a natural responsibility of an Entity or a Value Object.”
- Domain services helpen met
 - het encapsuleren van domain logica
 - het ophalen van informatie binnen of buiten het domein dat nodig is voor validatie
 - ...

Architecture



Onion architecture



Hexagonal architecture



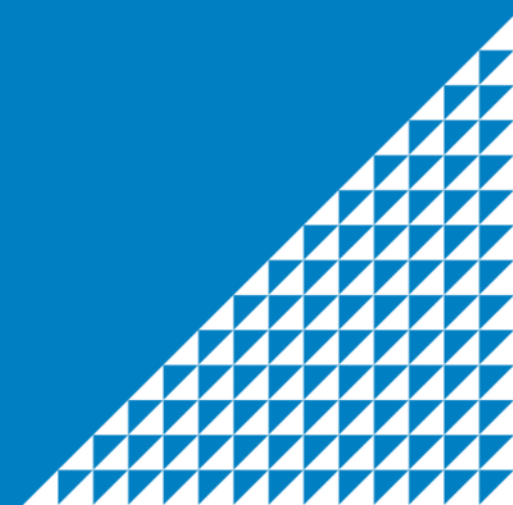
Oefening 1:

Implementeer een goeie structuur in je project

Implementeer de application layer

Implementeer de api/console

Zorg ervoor dat je van buiten het domain een aggregate kan creëren en updaten





▲ Persistence

- Data opslaan in een database
- Laten we starten van de code.



▲ State stored

- Sla de state op in de database
 - Voeg id toe
 - Voeg version toe
- Vervuil je domain state niet met een constructor

▲ Event sourcing

- Sla enkel de events op in de database
- Bouw de aggregate elke keer terug op
- ⚠️ Zorg dat je steeds backwards compatible blijft



▲ Wat kies je?

- Event sourcing wanneer:
 - Je historische data moet bijhouden
 - Je wil bijhouden waarom de state is verandert
 - Je instaat wil zijn om de state terug op te bouwen op eender welk moment in het verleden
- Anders hou je de state bij
- Hybrid

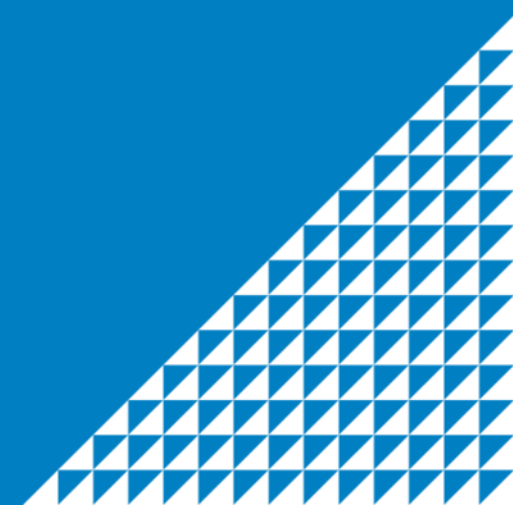


Oefening 2:

Implementeer de database

Sla je state op in de database

Kies zelf voor event sourced of state stored



▲ Messaging

- Niet essentieel voor DDD maar veel gebruikt
- Twee soorten events
 - Commands: Er moet een actie uitgevoerd worden
 - Notification: Er is iets gebeurt



▲ Command events

- Een domain beslist dat een actie moet gebeuren.
- Dit event heeft een specifieke ontvanger in gedachte.
- Dit event veroorzaakt typische iets in een ander domain.



▲ Notification event

- Een domain deelt mee dat iets gebeurt is.
- Dit event wordt breed uitgestuurd.
- Kan 0, 1 of meerdere ontvangers hebben.
- Bevat:
 - Enkel een id van de verzender;
 - Of alle informatie relevant aan de notificatie



▲ Synchroon vs asynchroon

- Synchroon
 - Alles in 1 process
 - Resultaat naar gebruiker als alles klaar is
 - Kan transactioneel
- Asynchroon
 - Message bus is vereist
 - Kan niet transactioneel. Vereist Saga patroon
 - Robust
 - Eventual consistency



▲ Hands on





Oefening 3:

Implementeer een side effect in je code

- Verstuur een command/notification
- Behandel het event

▲ Projections

- Handig wanneer je met event sourcing werkt
- Koppelt je aggregate los van je query modellen
 - Het splitsen van je gedrag en queries (CQRS)



▲ Voorbeeld in code





Oefening 4:

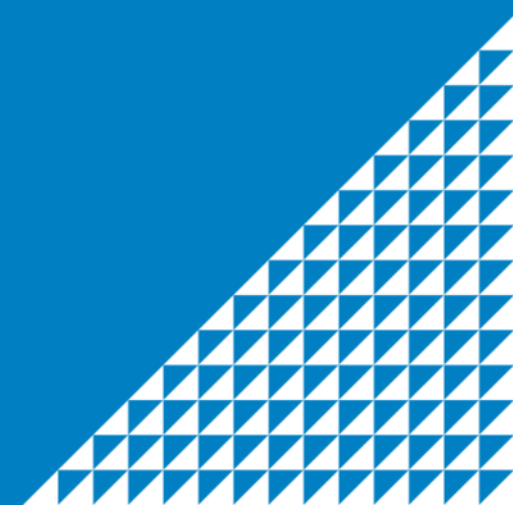
Implementeer een projection service

Query data via projections ipv aggregates



Andere use-cases voor projections

SAG: SQL augmented generation



▲ PXL handshake event



Bedankt en success met toekomstige DDD projecten

Daniel.mertens@infosupport.com

