
Implementation DDD
with Service Layers +
Data Mapper

Кто этот парень?!

Артем Антоненко

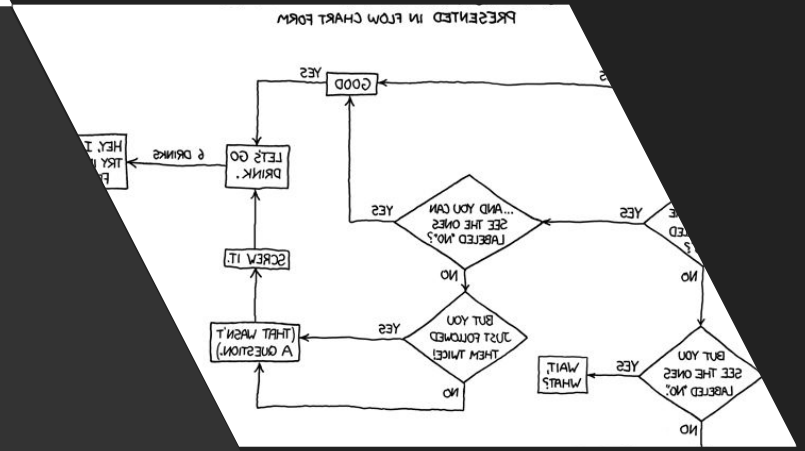
Senior Web Developer in Oracle corp.

Опыт в менеджменте более 9 лет.

Консультант в сфере реинжиниринга бизнес-процессов.

Более 5 лет в девелопменте.

Черный пояс Айкидо Есинкан.



Чем занимаюсь в IT?

- RESTful Web services
- Distributed Web Services
- Extend Testing Frameworks
- ATDD driver
- Setup SDLC

Ассистент

Евгений Кузнецов

yevhenii.kuznetsov@gmail.com

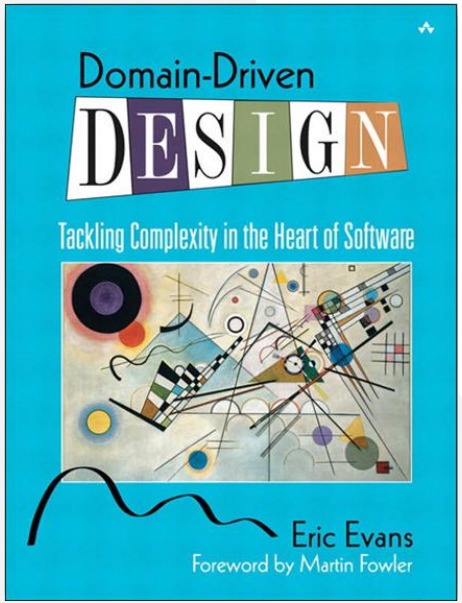
Agenda

… проектирование не детерминировано, методы проектирования чаще всего являются эвристическими методами, т.е. “практическими правилами” или “способами, которые могут сработать”, а не воспроизводимыми процессами, которые всегда приводят к предсказуемым результатам.

Стив Макконнелл

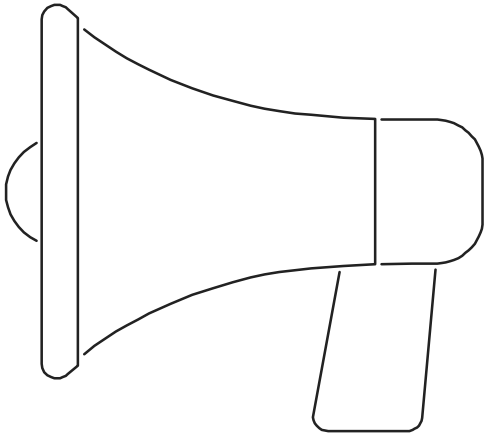
DDD

Предметно-ориентированное проектирование



Модель предметной области приносит наибольшую пользу только тогда, когда она предоставляет специалисту в этой предметной области и инженеру разработчику единый язык

Мартин Фаулер, апрель 2003г



QUESTION

Как вы строите общение с заказчиком?
Как вы понимаете чего он хочет?

Как продать DDD?

- Выделение и совершенствование УТП
- Точное описание БП
- Низкий шанс появления “тайных знаний”
- Снижение затрат на обучение пользователей
- Улучшение структуры предприятия
- Избавление от лишней документации

DDD. Основные понятия

- Единый язык
- Ограниченный контекст
- Предметная область
- Смысловое ядро
- Карта контекстов

DDD. Основные понятия

Единый язык

Создание глоссария с простыми определениями, а также альтернативными терминами с оценкой их перспективности.

Контекст

Граница, в пределах которой понятия

единого языка имеют вполне конкретное

значение.

Предметная область

это то, что делает организация, и среда, в которой она это делает. Часто состоит из предметных подобластей.

DDD. Основные понятия

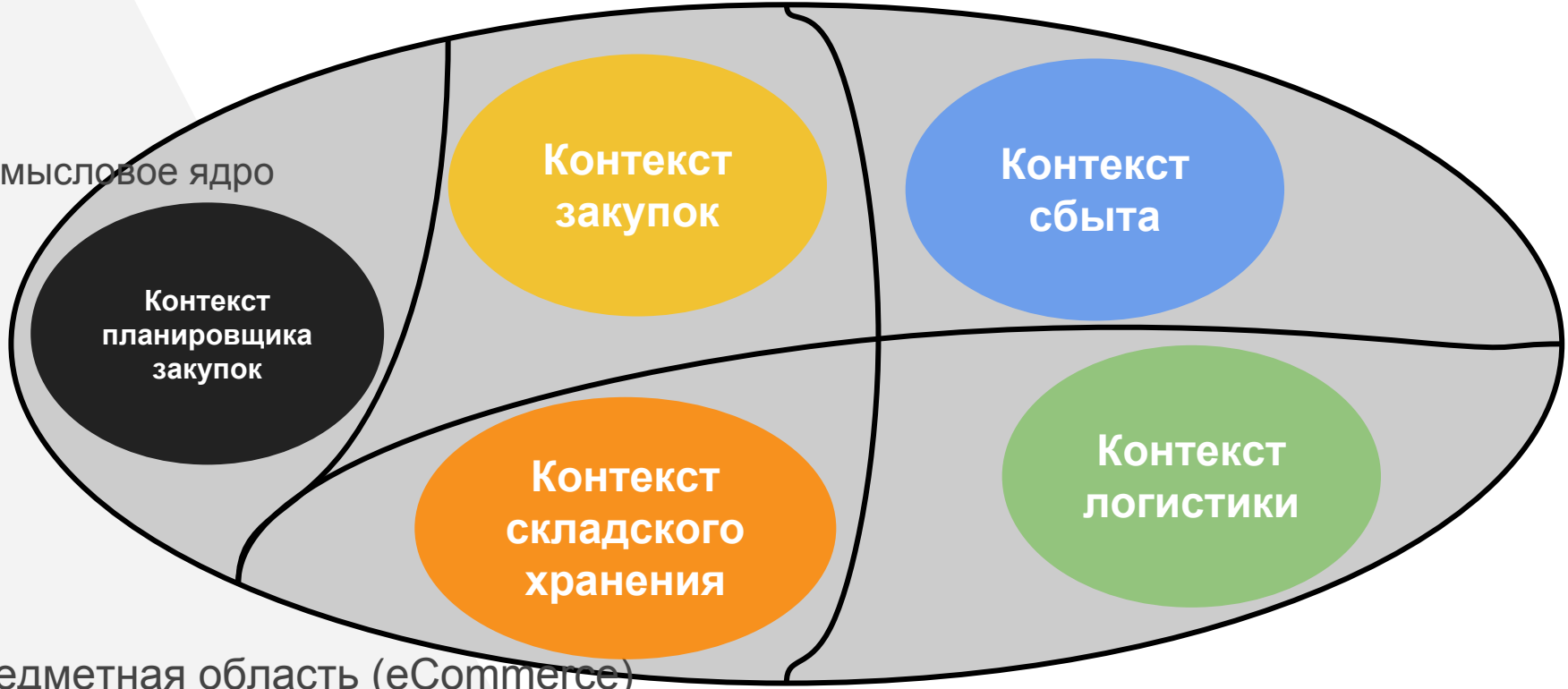
Смысловое ядро

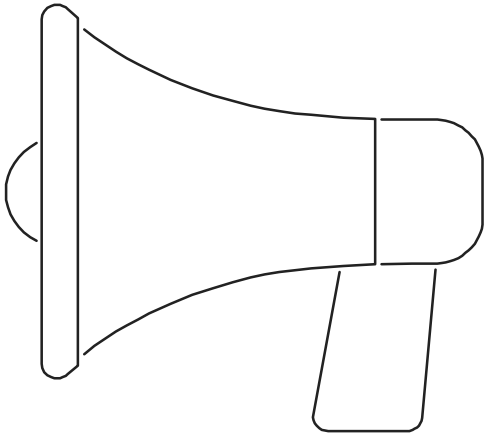
Подобласть, имеющая первостепенное значение для организации. Стратегия бизнеса строится вокруг смыслового ядра.

Карта контекстов

Схема, которая описывает все контексты, входящие в предметную область и их взаимосвязь

DDD. Пример


Смысловое ядро
Контекст планировщика закупок
Контекст закупок
Контекст сбыта
Контекст складского хранения
Контекст логистики
Предметная область (eCommerce)



PRACTICE

1. Выберите предметную область
2. Определите смысловое ядро
3. Создайте контексты
4. Создайте единый язык

DDD. Архитектура

- Layered Architecture
- LA + Dependency Inversion
- Hexagonal or Ports and Adapters
- Service Oriented
- CQRS
- Event Driven

SOLID

Principles

SOLID - это мнемонический акроним, введённый Майклом Фэзерсом (*Michael Feathers*) для первых пяти принципов, названных Робертом Мартином в начале 2000-х, которые означали пять основных принципов объектно-ориентированного программирования и проектирования.

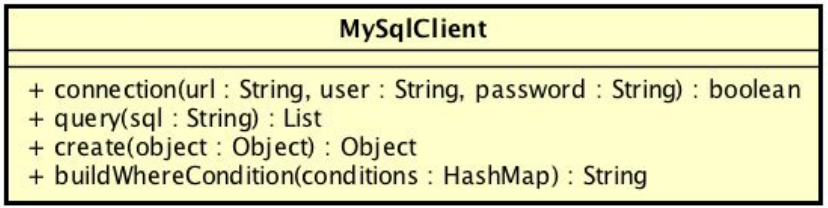
SOLID

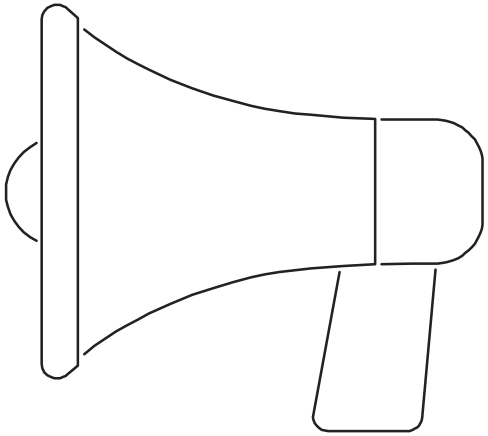
- Single responsibility
- Open-closed
- Liskov substitution
- Interface segregation
- Dependency inversion

SOLID. Single responsibility

"A class should have only one reason to change."

Robert C. Martin





QUESTION

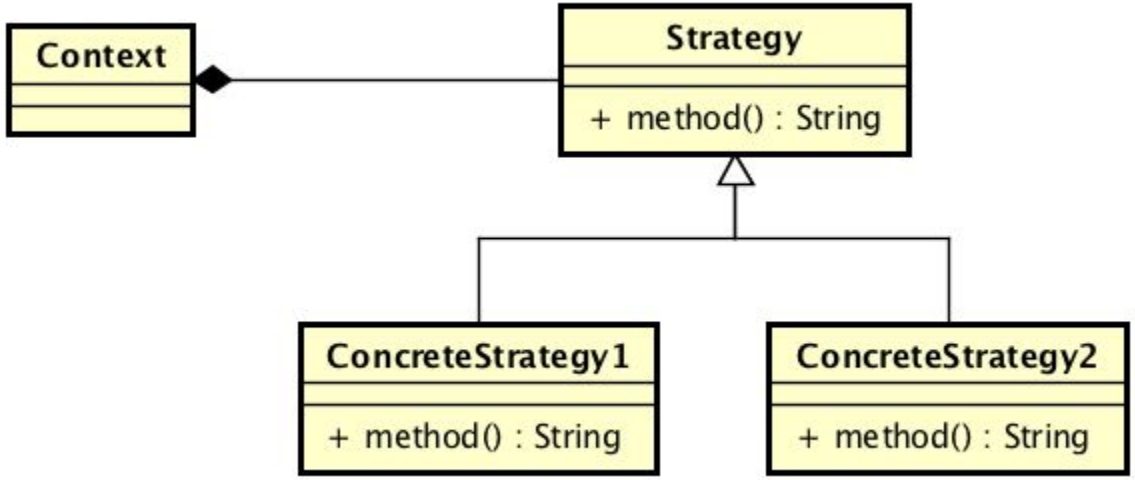
Какие причины для изменений класса?

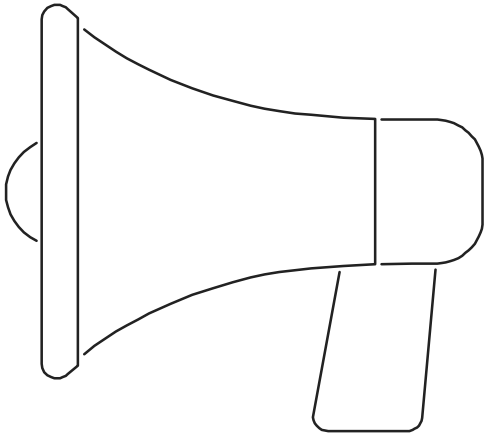
Как разделить ответственность?

SOLID. Open-closed

"Software entities (classes, modules, functions, etc.) should be open for extension, but closed for modification"

Bertrand Meyer





PRACTICE

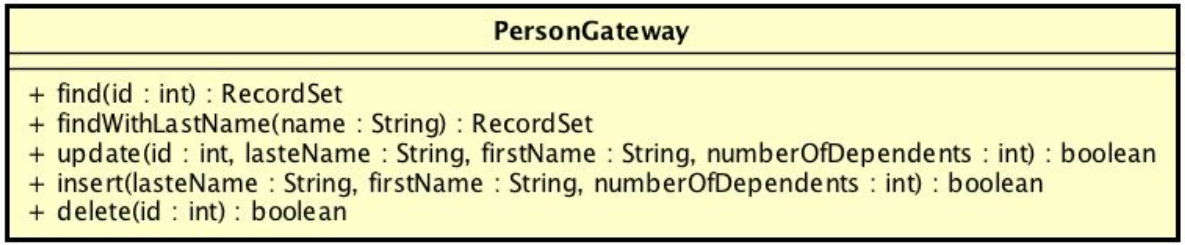
Реализуйте принцип открытости - закрытости на примере программы для обслуживания болидов формулы-1 при заезде на пит-стоп.

Data Source

Architectural Patterns

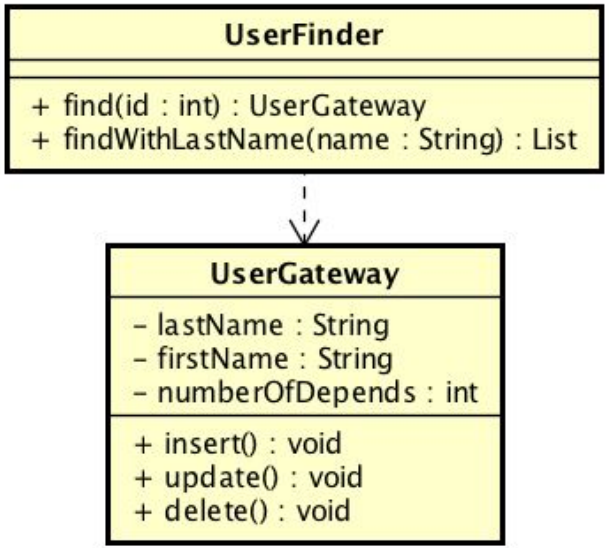
Data Source. Table Data Gateway

Объект, который выполняет роль шлюза к базе данных



Data Source. Table Row Gateway

Объект, который выполняет роль шлюза к отдельной записи источника данных. Каждой строке таблицы базы данных соответствует свой экземпляр шлюза записи данных



Data Source. Table & Row Data Gateway

Назначение

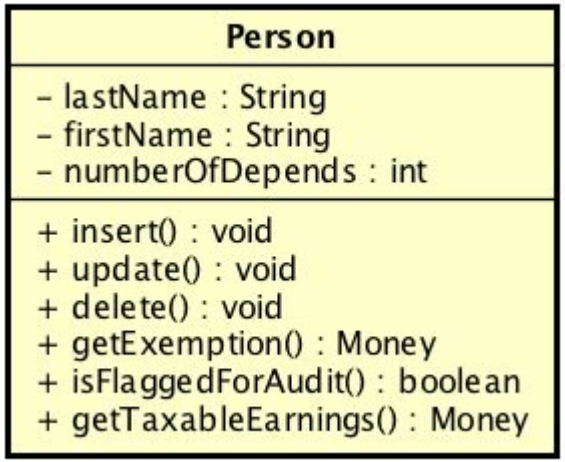
Инкапсулирование логики доступа к базе данных. Прекрасно подходит для применения в сценариях транзакций. Может сочетаться с Data Mapper.

Недостатки

Нет возможности реализовать сложную доменную логику из-за прямого маппинга таблиц/записей на объект.

Data Source. Active Record

Объект, выполняющий роль оболочки для строки таблицы или представления базы данных. Он инкапсулирует доступ к базе данных и добавляет данным логику домена



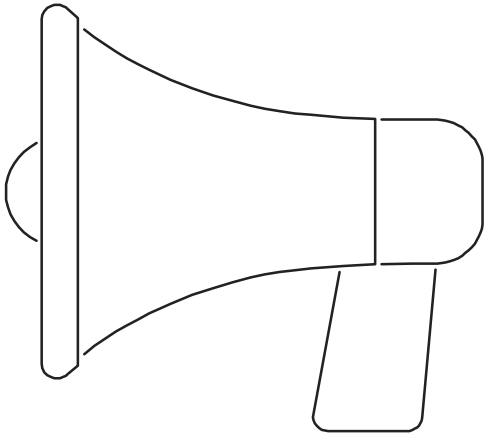
Data Source. Active Record

Назначение

Реализация простых вариантов доменной логики.

Недостатки

Применение только в изоморфных схемах. Высокая сложность реализации композиций, наследования, коллекций и т.д.



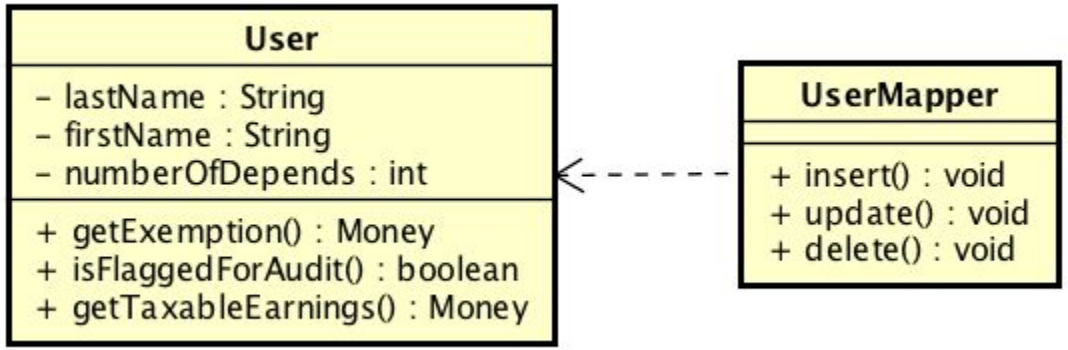
QUESTION

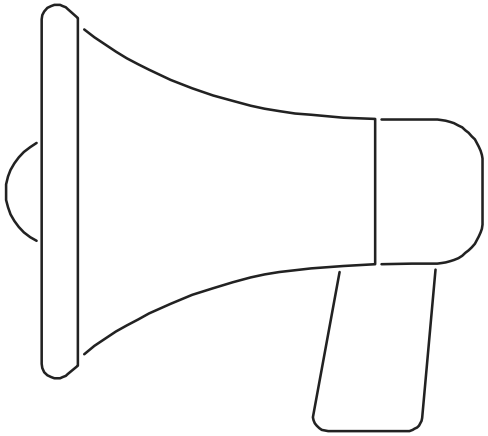
Какой принцип SOLID нарушает Active Record?

Что можете сказать о написании тестов при данном подходе?

Data Source. Data Mapper

Слой преобразователей, который осуществляет передачу данных между объектами и базой данных, сохраняя последние независимыми друг от друга и от самого преобразователя





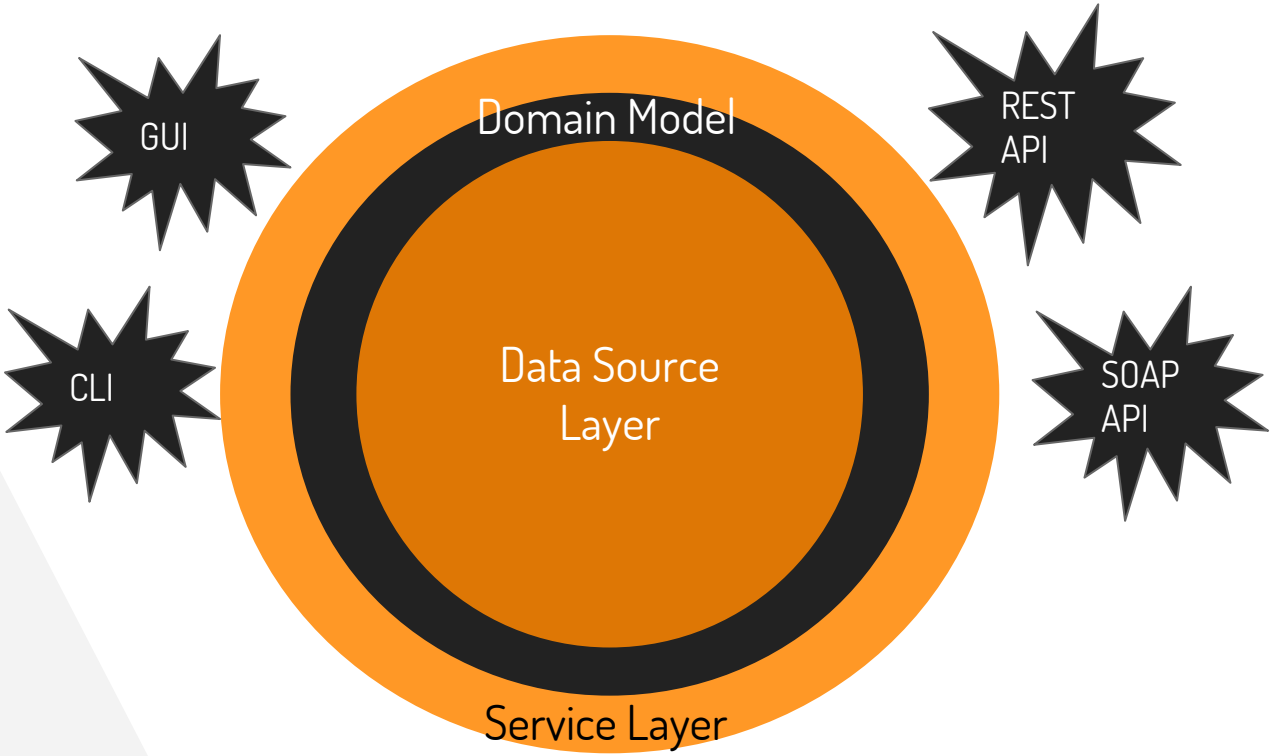
PRACTICE

Реализуйте все 4 типа архитектур источников данных для задачи создания разных типов депозитов с разным поведением

Service Layer

Defines an application boundary

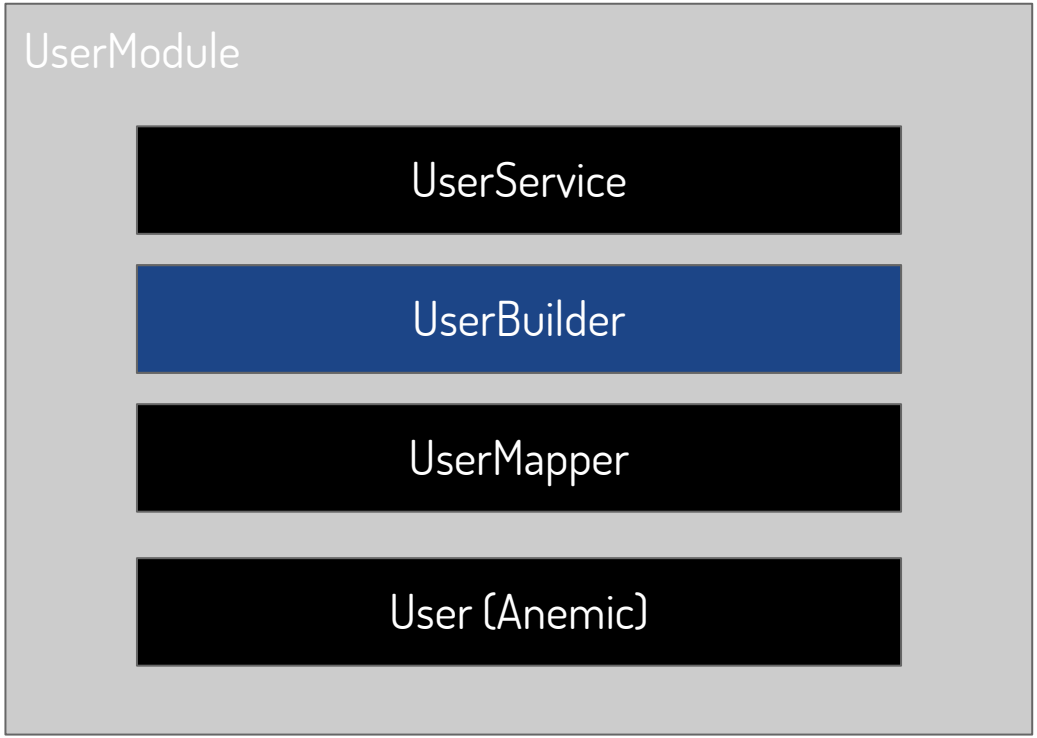
Service Layer



Module

Distributed architecture

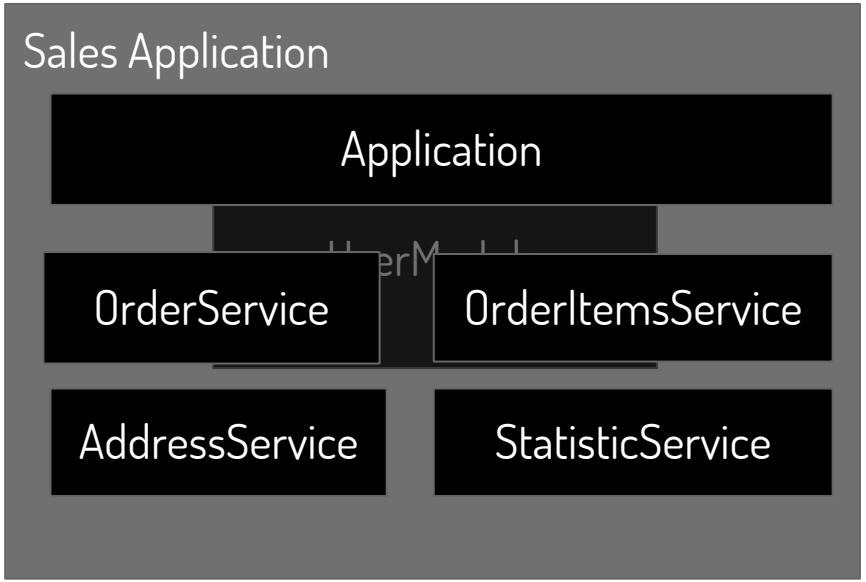
Module


UserModule
UserService
UserBuilder
UserMapper
User (Anemic)

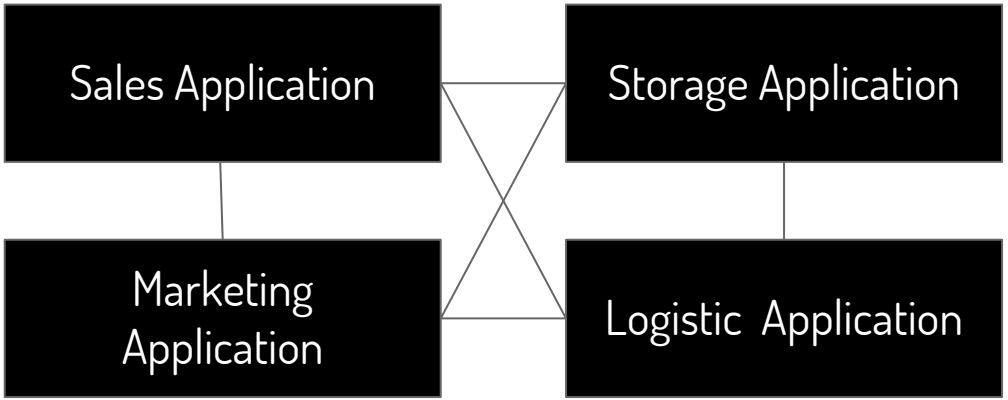
Module. Rules

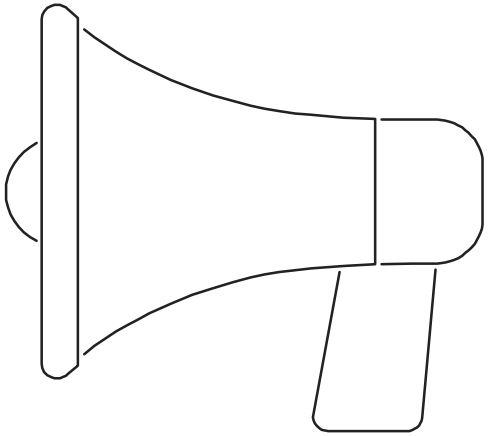
- Сервис не хранит состояния
- Сервис может взаимодействовать с другими сервисами или с мапперами
- Состояние хранит только модель
- Маппер знает как создать и сохранить модель
- Маппер отделяет источник данных от доменной модели
- Источников данных может быть больше одного
- Этим подходом реализовывается только data-centric архитектура

Module


Sales Application
Application
OrderService
OrderItemsService
AddressService
StatisticService

Module


Sales Application
Storage Application
Marketing Application
Logistic Application



PRACTICE

Составьте модульную структуру для одного из контекстов, определенных ранее.

Спасибо за внимание!

Задать дополнительные вопросы по презентации вы можете по адресу: antonenko.artem@gmail.com