



ОНЛАЙН-ОБРАЗОВАНИЕ


Онлайн-образование

Проверить, идет ли запись!





Меня хорошо видно && слышно?

Ставьте  , если все хорошо
Напишите в чат, если есть проблемы

The background of the slide is a high-angle, blue-tinted aerial photograph of a dense urban skyline, likely New York City. Overlaid on this image is a semi-transparent network diagram consisting of numerous small blue dots connected by thin, light-blue lines, creating a web-like pattern across the center of the slide.

Domain Driven Design ОСНОВЫ

Правила вебинара



Активно участвуем



Задаем вопрос в чат или голосом



Off-topic обсуждаем в Slack #канал группы или #general



Вопросы вижу в чате, могу ответить не сразу

Цели вебинара | После занятия вы сможете

1

Объяснить что такое DDD и зачем придумано

2

Сформулировать стратегические и тактические шаблоны DDD

3

Реализовать свою модель предметной области на основе практического примера

Смысл | Зачем вам это уметь

1

Знание и понимание DDD позволяет перейти на качественно другой уровень разработки систем

2

Основа DDD - ориентация на бизнес и то зачем приложение разрабатывается, это важно для Senior-разработчика

3

DDD позволяет строить качественные ООП модели предметной области приложения

Маршрут вебинара

Введение



Стратегические шаблоны



Тактические шаблоны



Рефлексия



Введение



Определение

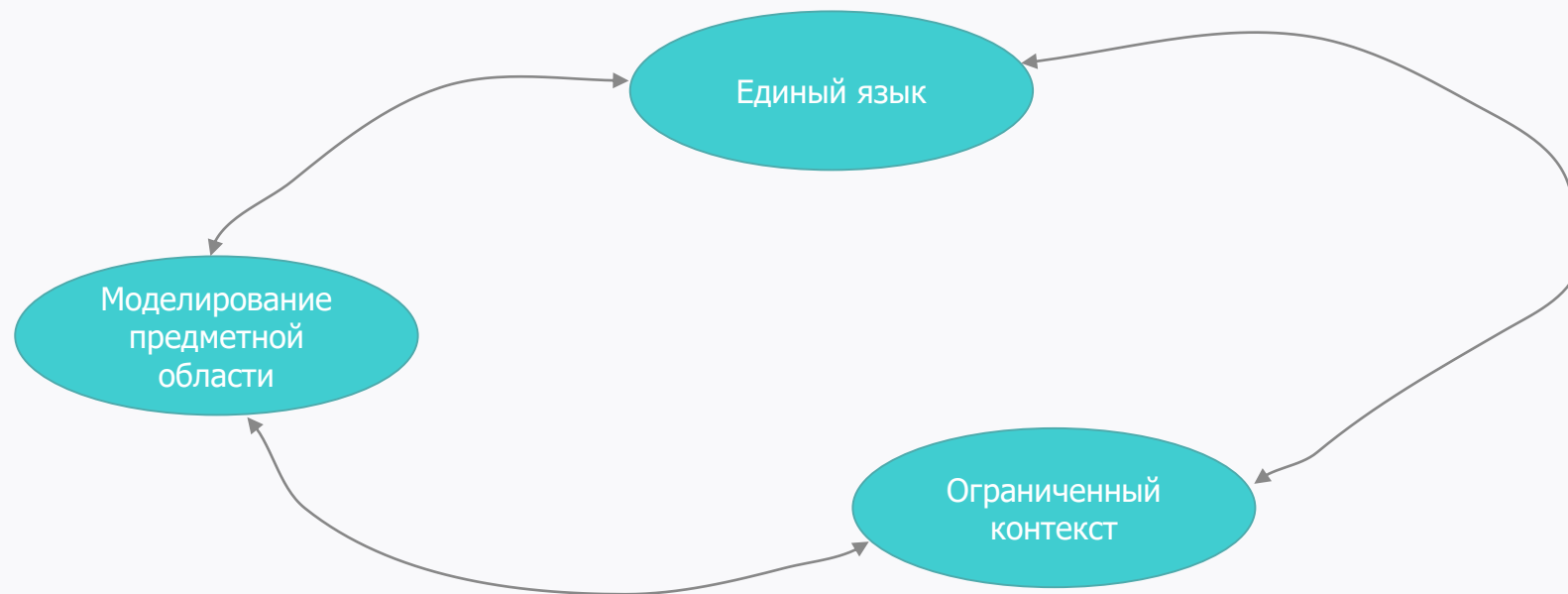
Domain Driven Design (Предметно-ориентированное проектирование, DDD) - это набор принципов, направленных на стирание границ между бизнесом и фактической реализацией в программном коде.

Основная цель - построение модели предметной области

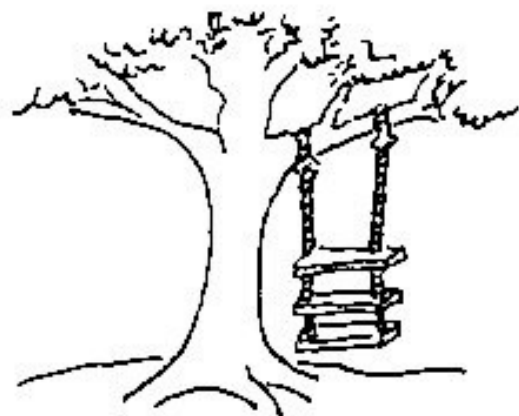
1. Одинаково понятна, как бизнес-экспертам, так и команде разработки
2. Явно выражена в программном коде, что позволяет разрабатывать продукт, который будет развиваться вместе с требованиями бизнеса

Основные принципы

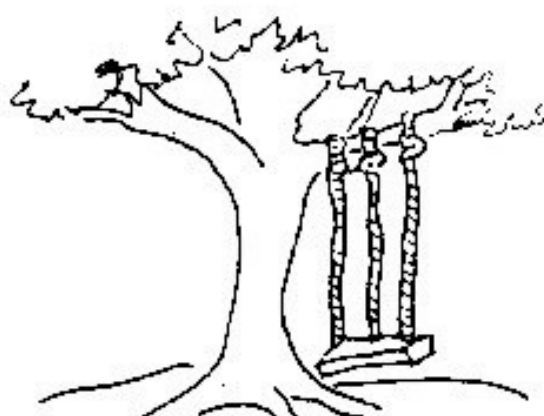
- ❑ Проектирование системы по модели
- ❑ Единый язык для проектирования модели и коммуникации между всеми участниками проекта, разработчиками, пользователями и специалистами предметной области
- ❑ Выделение ограниченного контекста предметной области для отражения в модели наиболее важных элементов для данной части предметной области



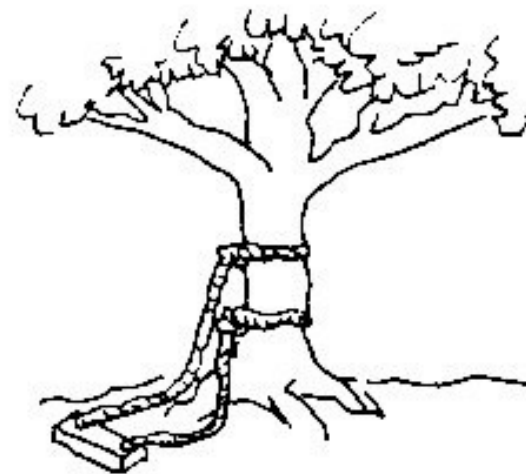
Мотивация



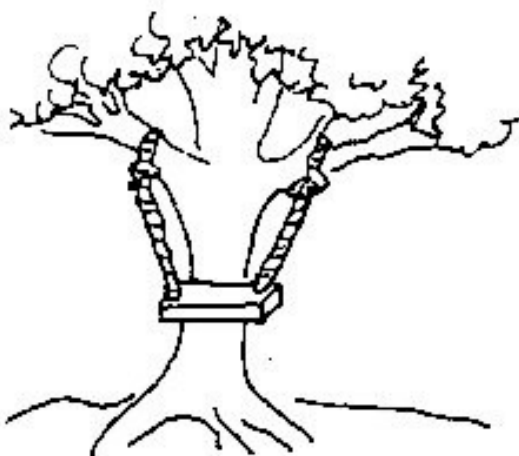
1. Как предложено
инициатором проекта



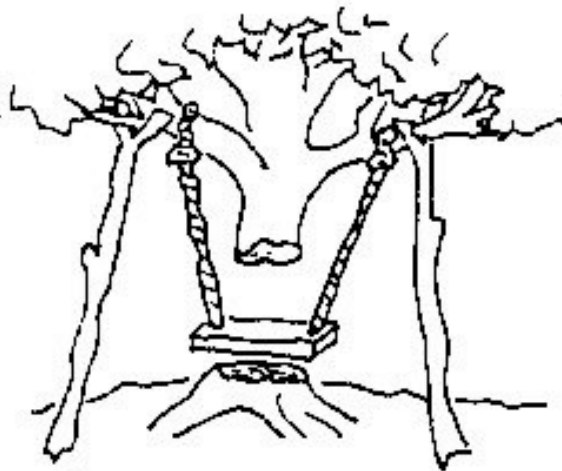
2. Как определено
в техническом задании



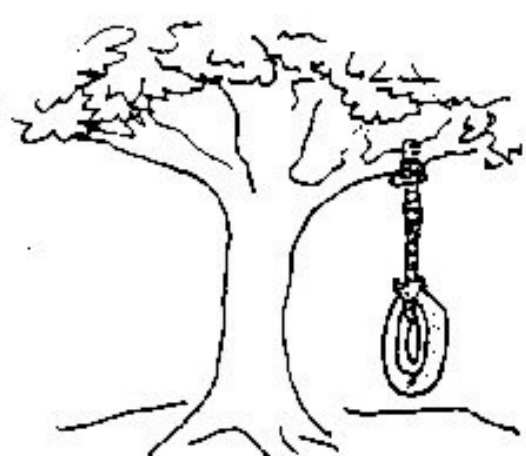
3. Как спроектировано



4. Как запрограммировано



5. Как установлено



6. Что, собственно, хотел заказчик

Плюсы и минусы



- ☐ Возможность реализации программы без качественного технического задания
- ☐ Уменьшение сложности
- ☐ Быстрая реализация и ввод в эксплуатацию базовой версии
- ☐ Гибкость архитектуры и возможность расширения

Плюсы и минусы



- ❑ Возможность реализации программы без качественного технического задания
- ❑ Уменьшение сложности
- ❑ Быстрая реализация и ввод в эксплуатацию базовой версии
- ❑ Гибкость архитектуры и возможность расширения



- ❑ Проблемы определения контекста предметной области
- ❑ Сложность формализации модели
- ❑ Технические проблемы при реализации архитектуры, ориентированной на модель

The image features a high-angle, aerial view of a city skyline, likely New York City, with numerous skyscrapers and buildings. The entire image is tinted with a blue and green color scheme. A network of white lines and dots is overlaid on the image, creating a digital or technological feel. The text "Стратегические шаблоны" is centered in the middle of the image, overlaid on a dark blue horizontal band.

Стратегические шаблоны

Единый язык

Единый язык (Ubiquitous Language) - это терминология, которая используется при разработке продукта и применяется всеми лицами, которые участвуют в разработке. Она одинаково понятна всем лицам, участвующим в разработке, бизнесу, аналитикам, разработчикам

Единый язык - пример

Единый язык (Ubiquitous Language) - это терминология, которая используется при разработке продукта и применяется всеми лицами, которые участвуют в разработке. Она одинаково понятна всем лицам, участвующим в разработке, бизнесу, аналитикам, разработчикам

```
employeeService.Create(employee);
```

VS

```
employees.Hire(employee);
```

Единый язык - пример 2

В команде, обсуждая модель применения вакцины от гриппа в виде кода, произносят фразу наподобие:
«Медсестры назначают вакцины от гриппа в стандартных дозах».

Единый язык - пример 3

В команде, обсуждая модель применения вакцины от гриппа в виде кода, произносят фразу наподобие:
«Медсестры назначают вакцины от гриппа в стандартных дозах».

Просто пишем код



```
patient.SetShotType(ShotTypes.Flu);  
patient.SetDose(does);  
parient.SetNurse(nurse);
```

Единый язык - пример 4

В команде, обсуждая модель применения вакцины от гриппа в виде кода, произносят фразу наподобие: «Медсестры назначают вакцины от гриппа в стандартных дозах».

Просто пишем код



```
patient.SetShotType(ShotTypes.Flu);  
patient.SetDose(does);  
parient.SetNurse(nurse);
```

Делаем прививку от гриппа



```
patient.giveFluShot();
```

Единый язык - пример 5

В команде, обсуждая модель применения вакцины от гриппа в виде кода, произносят фразу наподобие: «Медсестры назначают вакцины от гриппа в стандартных дозах».

Просто пишем код



```
patient.SetShotType(ShotTypes.Flu);  
patient.SetDose(does);  
parient.SetNurse(nurse);
```

Делаем прививку от гриппа



```
patient.giveFluShot();
```

Медсестры назначают вакцины от
гриппа в стандартных дозах



```
var vaccine = vaccines.standartFluDose();  
nurse.AppointVaccine(patient, vaccine);
```


Ограниченный контекст

Ограниченный контекст - это явная граница, внутри которой существует единый язык бизнес-процессов домена

Имеется в виду некоторое отображение домена на программу, которая его реализует.

Пример

Продажа бытовой техники - домен

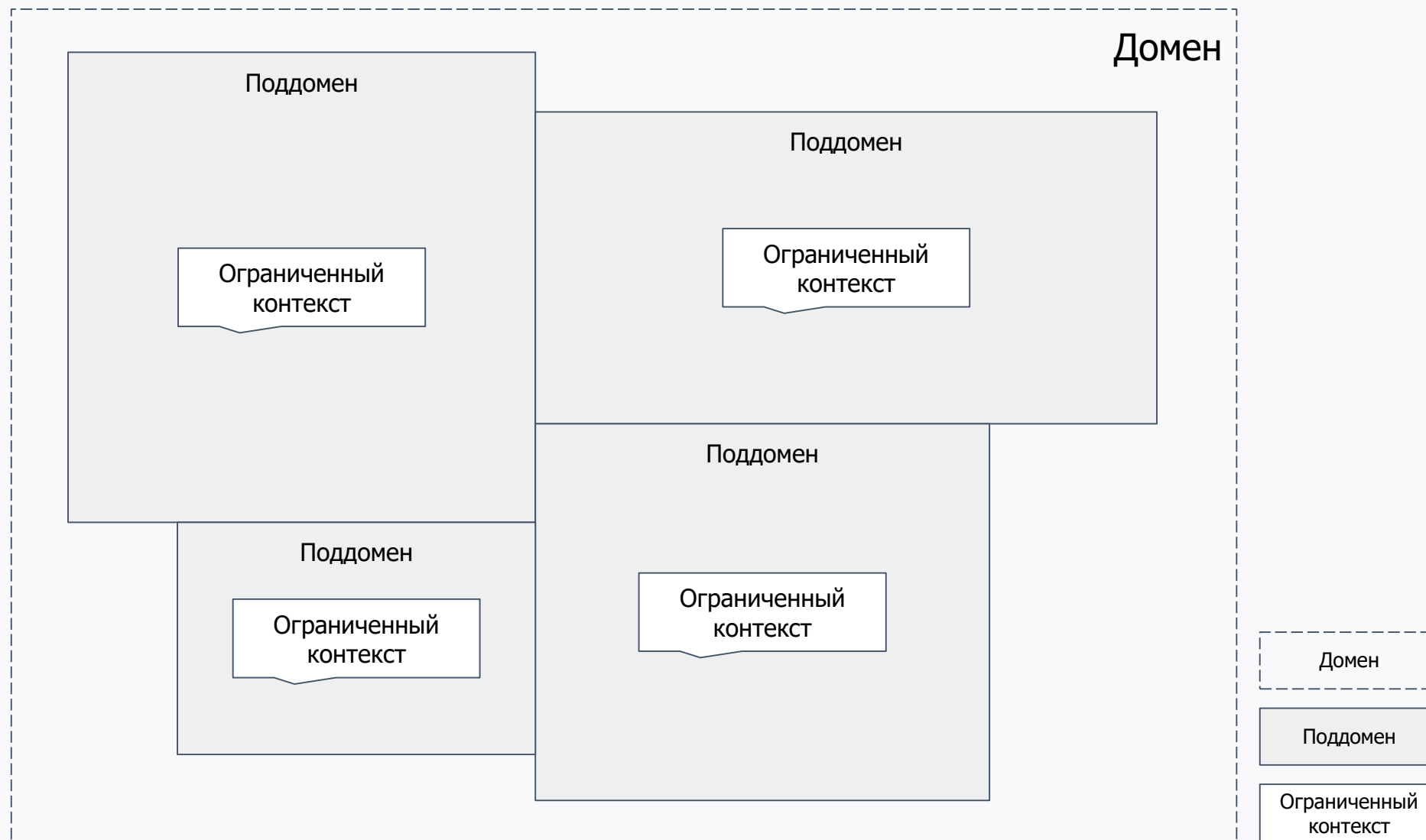
Создание заказа на продажу - поддомен

Интернет-магазин - ограниченный контекст

Доставка заказа - поддомен

Приложение курьера для доставки - ограниченный контекст

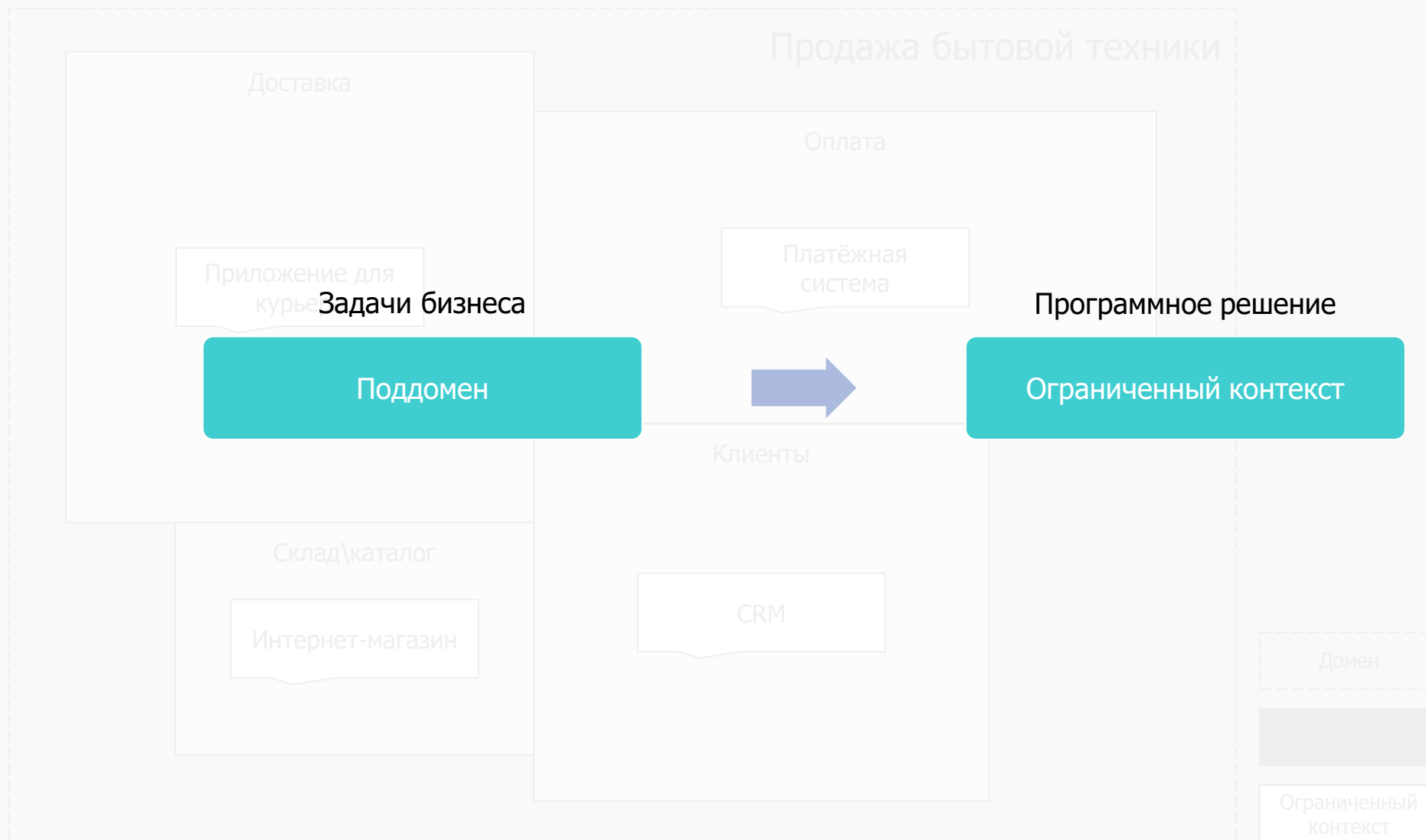
Ограниченный контекст - пример



Ограниченный контекст - пример 2



Ограниченный контекст - пример 3



Модель предметной области

Модель предметной области - это наши классы, написанные на C#, выраженные в терминах единого языка и работающие внутри приложения-ограниченного контекста.

С точки зрения бизнеса это понятия, которыми он оперирует, но важно, что это все одно и то же.

Анемичная модель и богатая модель

Anemic Domain Model (Анемичная модель) - модель без бизнес-логики, просто структура данных. Такой формат возникает, когда мы ставим в центр базу данных, это просто сценарий транзакции и бизнес-логика вне данных.

Rich Domain Model (Богатая доменная модель) - рассматривается в контексте DDD, структура данных и бизнес-логика внутри модели, в базе данных не должно быть логики, такой подход возможен, когда БД не стоит в центре логики.

Домен клиентов

Часть информационной системы для работы с клиентами

Клиент имеет некоторый набор данных, несколько мест работы и контакты.

Сценарий использования:

1. Находим нужного клиента по ФИО или создаем нового
2. Переходим в его карточку
 - а. Добавляем, удаляем контакты, места работы и редактируем общие данные
3. При сохранении происходит сохранение всех данных

Нужно описать эту структуру данных и реализовать CRUD



Демо





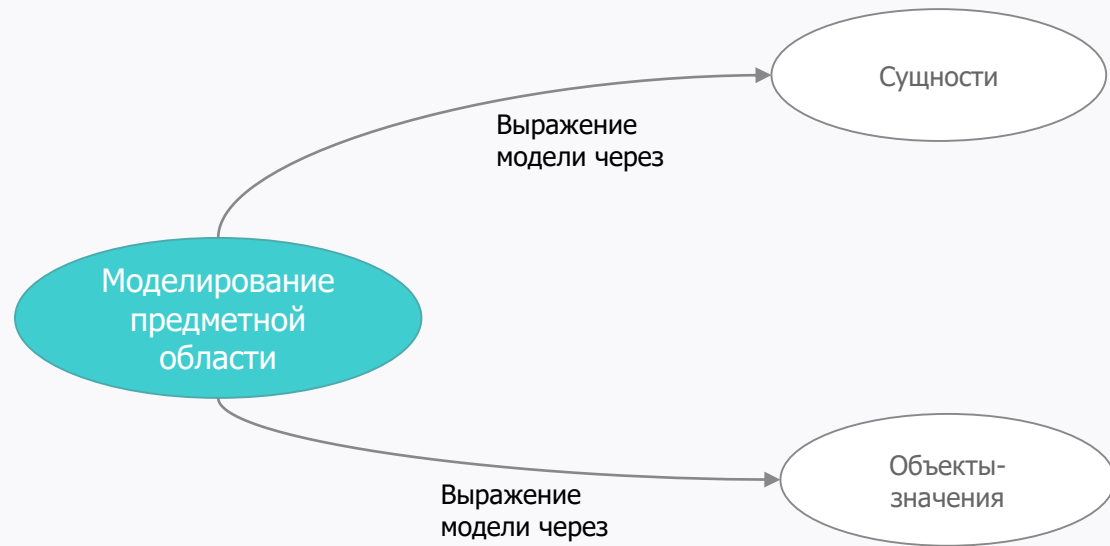
Тактические шаблоны



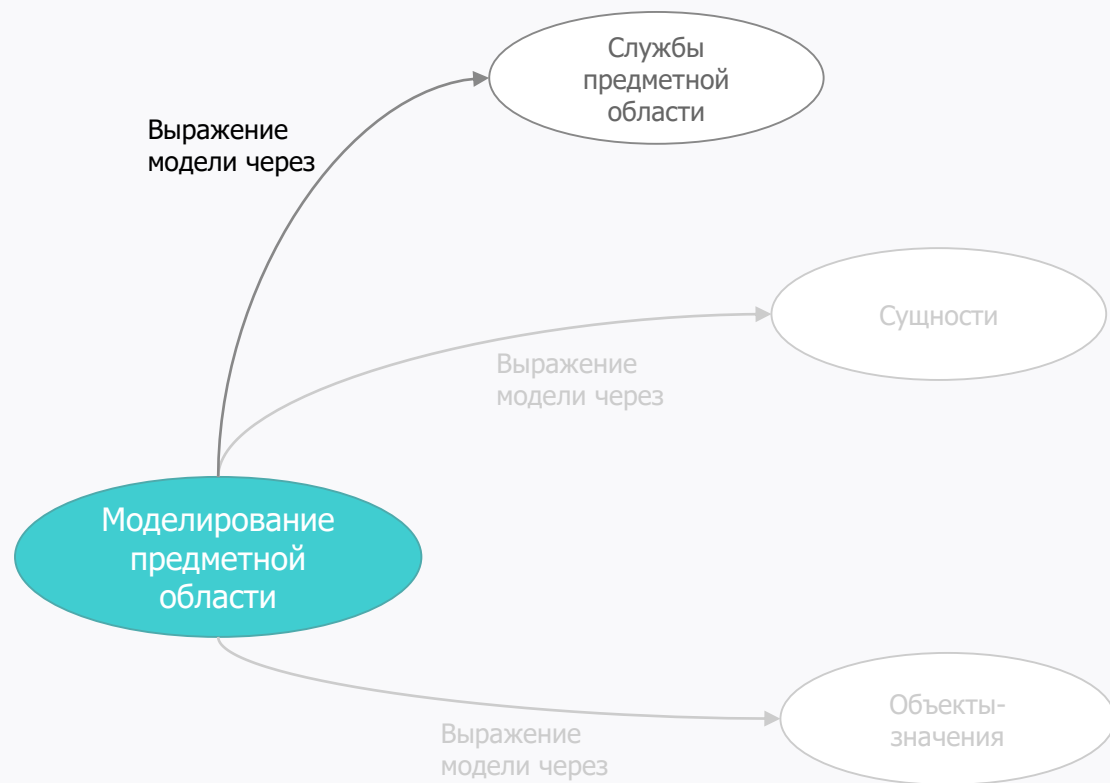
Структурные элементы

Моделирование
предметной
области

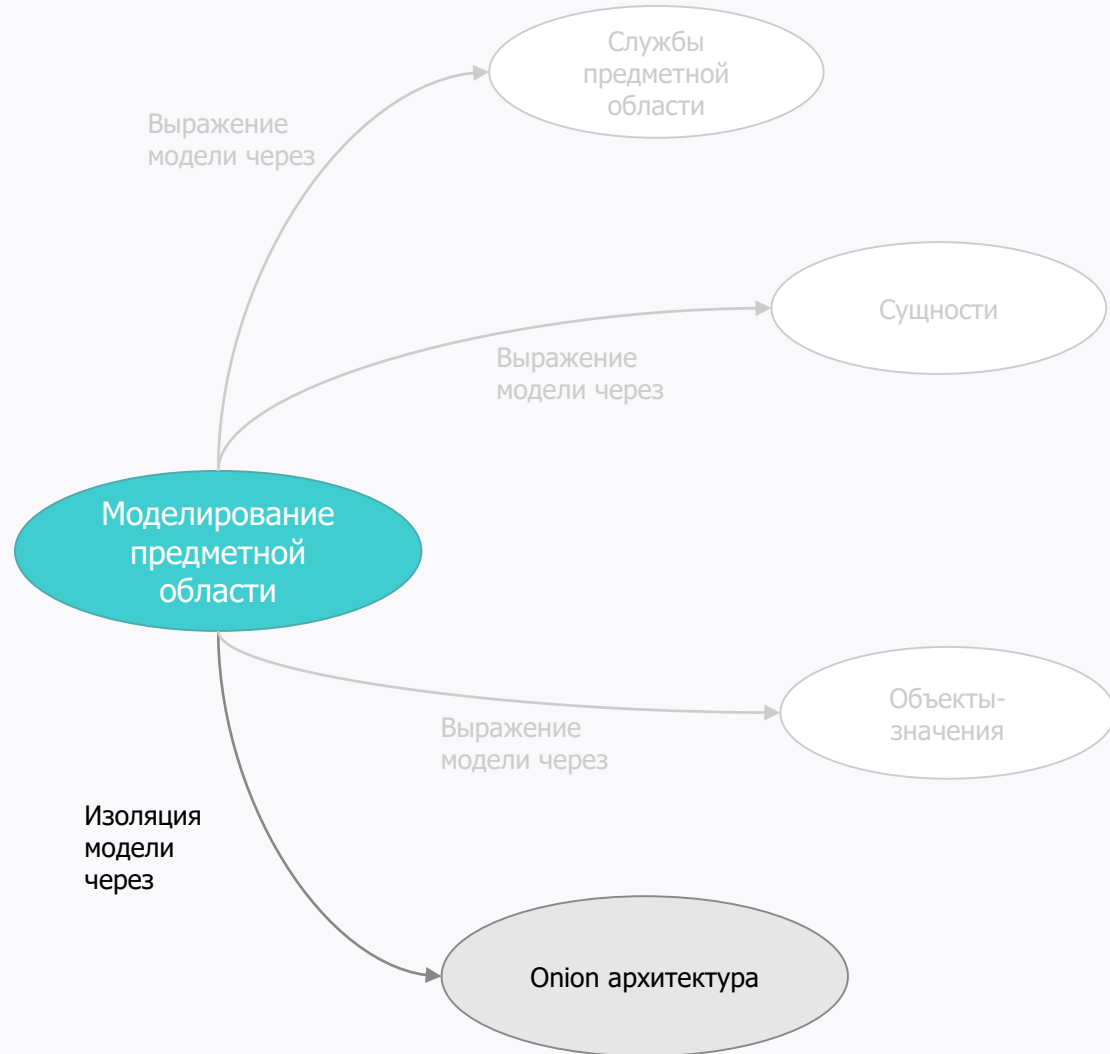
Структурные элементы - сущности



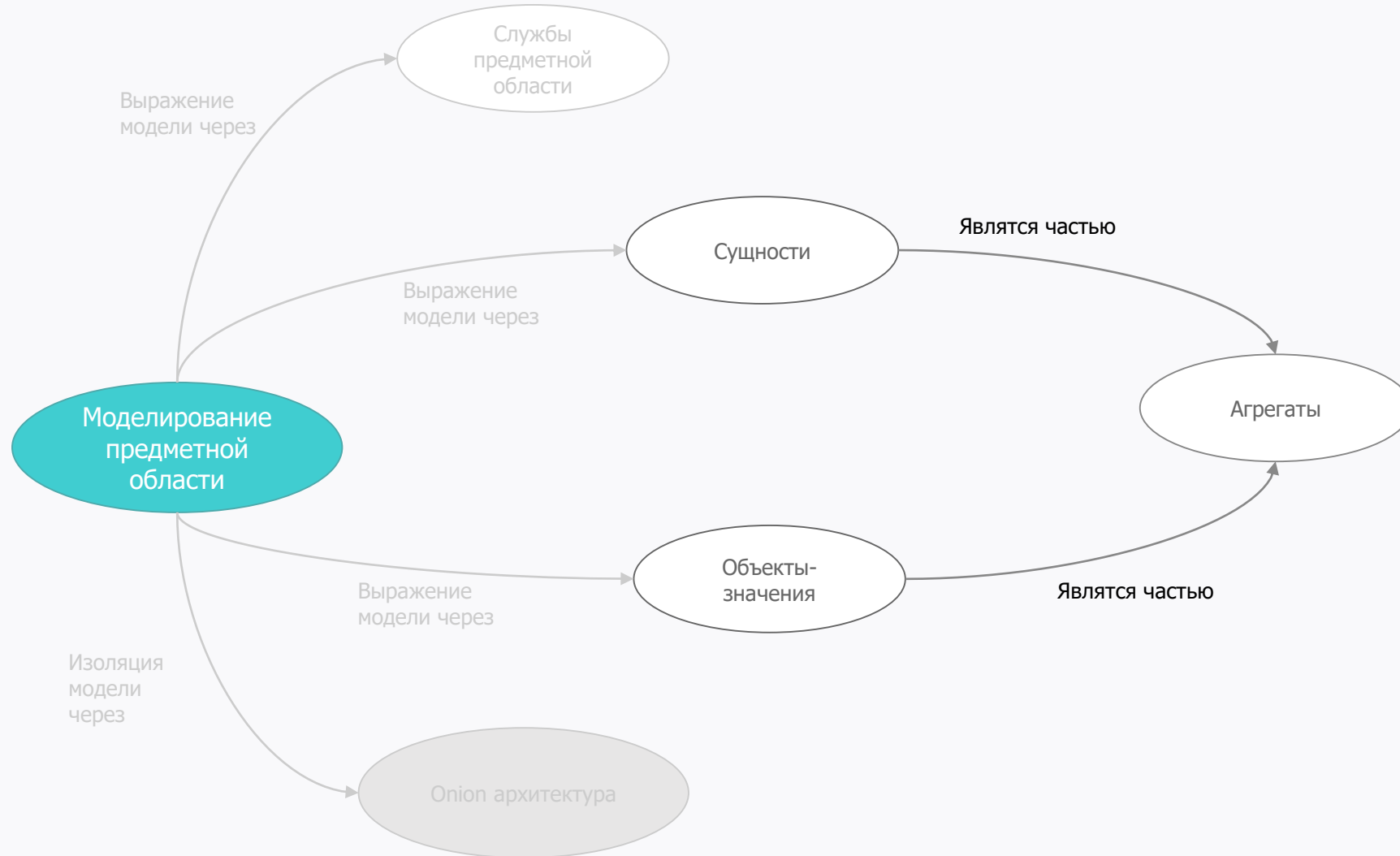
Структурные элементы - службы



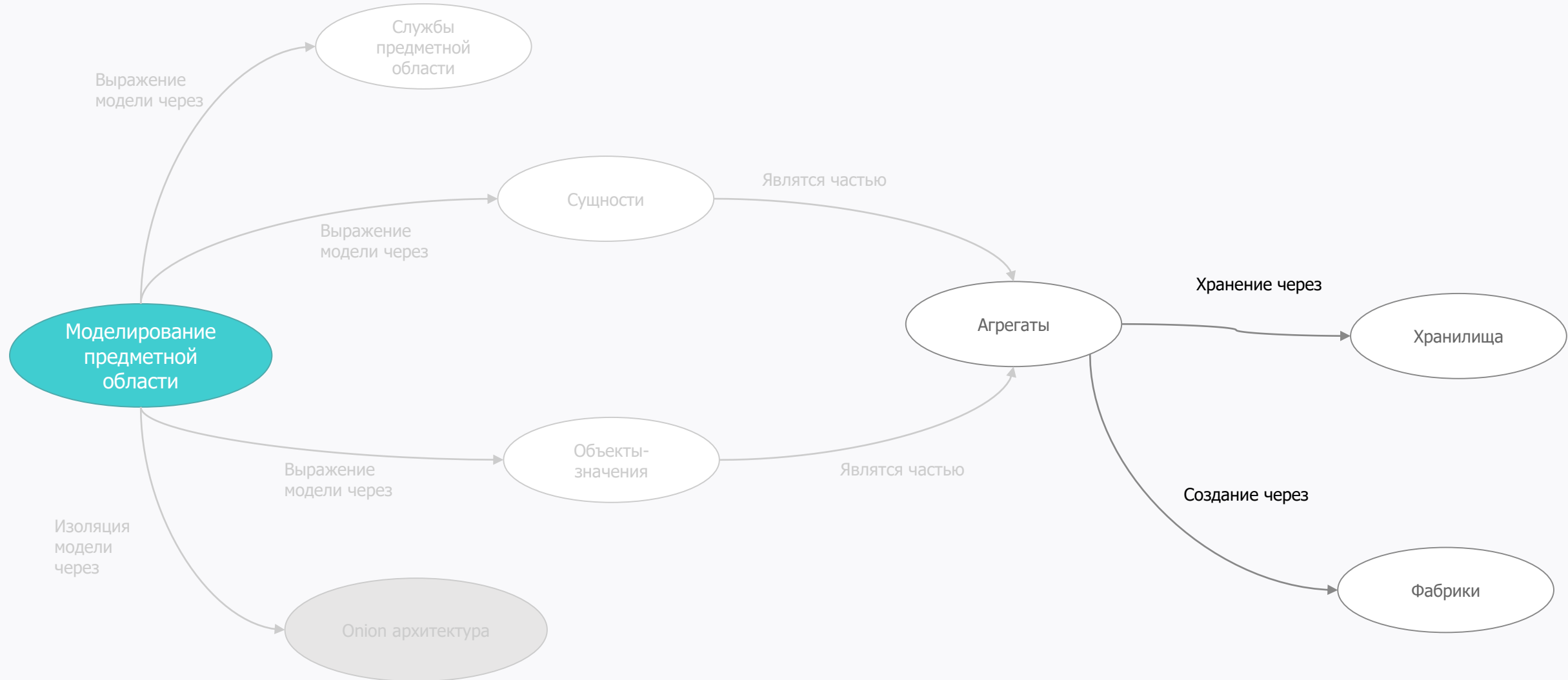
Структурные элементы - изоляция



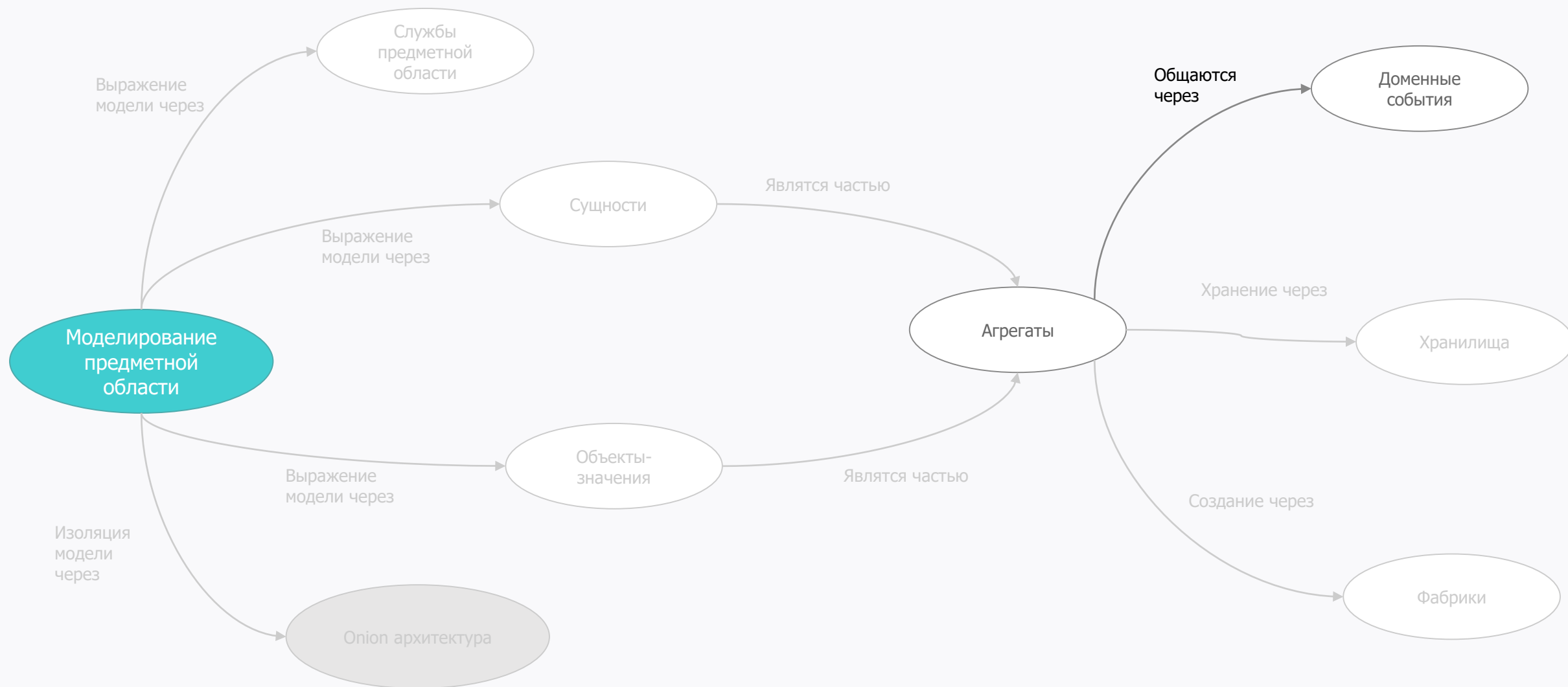
Структурные элементы - агрегаты



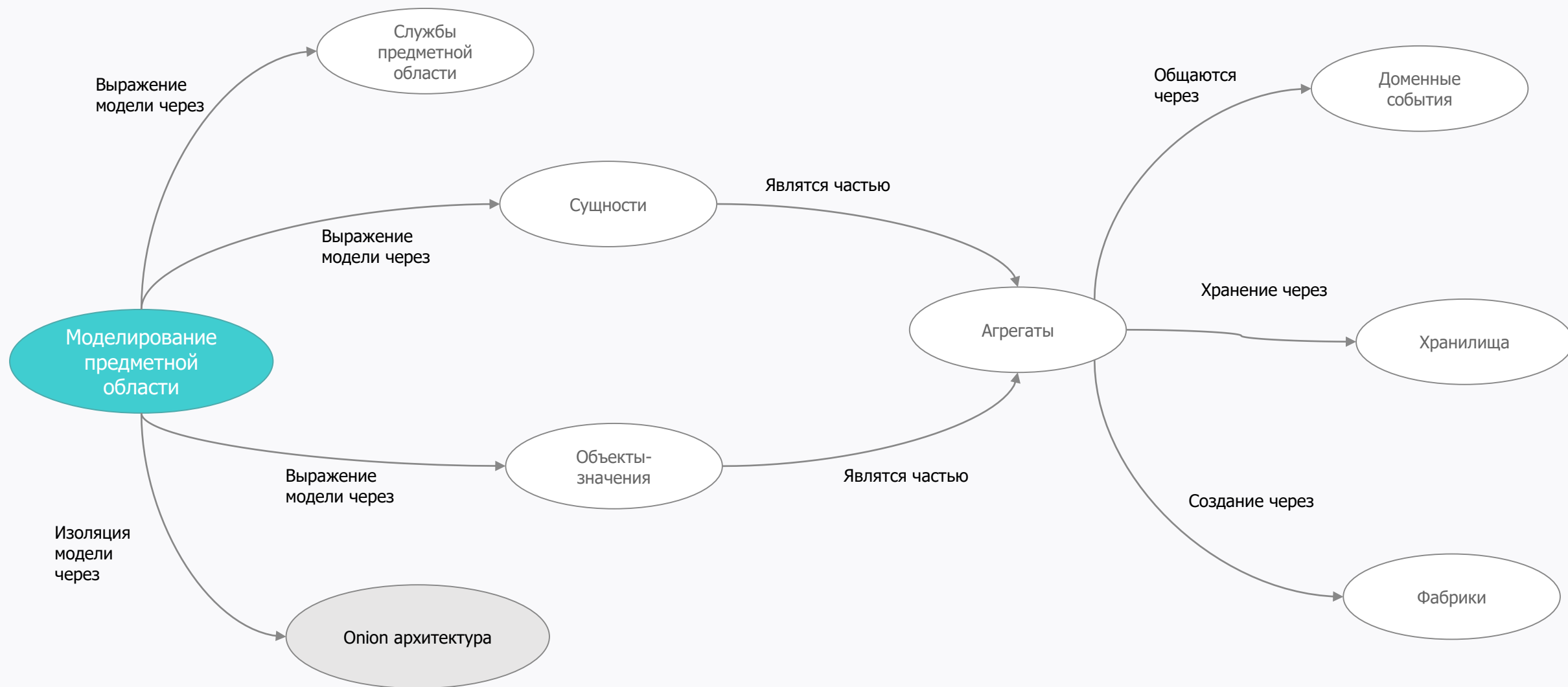
Структурные элементы - хранение агрегатов



Структурные элементы - общение агрегатов



Структурные элементы - обзор





Демо



Выводы

1

Посмотрели что такое DDD

2


Разобрали стратегические и тактические шаблоны

3

Посмотрели примеры кода

Основные выводы

- ❑ DDD - это не сложно.
- ❑ DDD решает проблемы, которые есть в большинстве приложений и рекомендует писать код в ООП стиле, оперируя понятными терминами, а не процедуры на C# вокруг таблиц в БД.
- ❑ DDD помогает сделать ваши проекты лучше.
- ❑ Не внедряйте DDD - просто используйте его принципы
- ❑ Если вы начнете вводить единый язык, делать ваши классы богаче и думать ограниченными контекстами - любой проект станет лучше, особенно, в микросервисах

The background of the entire image is an aerial photograph of a city, likely New York City, showing numerous skyscrapers and buildings. The image is overlaid with a semi-transparent blue layer. In the center of this blue layer, there is a network of white lines connecting various points, creating a geometric pattern. The text is centered within this blue area.

Заполните, пожалуйста,
опрос о занятии по ссылке в чате
<https://otus.ru/polls/21443/>

The background of the slide is a high-angle, aerial photograph of a city skyline, likely New York City, featuring numerous skyscrapers and dense urban development. The image is overlaid with a semi-transparent blue and green gradient. A network of thin, light blue lines connects various points across the gradient, creating a digital or network-like pattern. The word "Вопросы" is centered in the middle of the slide in a white, sans-serif font.

Вопросы



Спасибо за внимание!





Васкуп слайды



Сущность

Уникальное понятие предметной области, отличное от других объектов.

Такие объекты-сущности могут сильно отличаться своей формой за весь цикл существования, тем не менее их всегда можно однозначно идентифицировать. Для этого используются уникальные идентификаторы, создание которых необходимо продумать в первую очередь при проектировании сущности

Объект-значение

Сущность, полностью определяемая своими атрибутами, неизменяемая и которая может сравниваться по содержимому

Вопрос - интернет-магазин

Является ли адрес объектом-значением?

Служба предметной области

- ❑ Операция, выполняемая службой, относится к концепции предметной области, которая не принадлежит ни одной из существующих сущностей;
- ❑ Операция выполняется над различными объектами модели предметной области;
- ❑ Операция не имеет состояния

Событие

- ❑ Что то уже случилось
- ❑ Концепция масштаба предметной области, а не отдельного ограниченного контекста

Агрегат

- ❑ Кластер из объектов сущностей или значений
- ❑ Обращения - через корень агрегации
 - ❑ Сущность с глобальн уникальным идентификатором
 - ❑ Хранит инварианты
- ❑ Общаются событиями с другими агрегатами

Фабрика

- ❑ Фабрики – элементы программы, обязанности которого создавать другие объекты.
- ❑ Используются для создания сложных агрегатов или сущностей
- ❑ Фабричный метод в корне агрегата. Фабричный метод еще выгоден тем, что с его помощью можно выразить единый язык (конструктор же не выражает это).

Репозиторий \ хранилище

- ❑ Используется для сохранения\получения агрегатов