CUSTIS®

Domain Driven Design в условиях разработки распределенных приложений

Николай Гребнев

CUSTIS®

Содержание

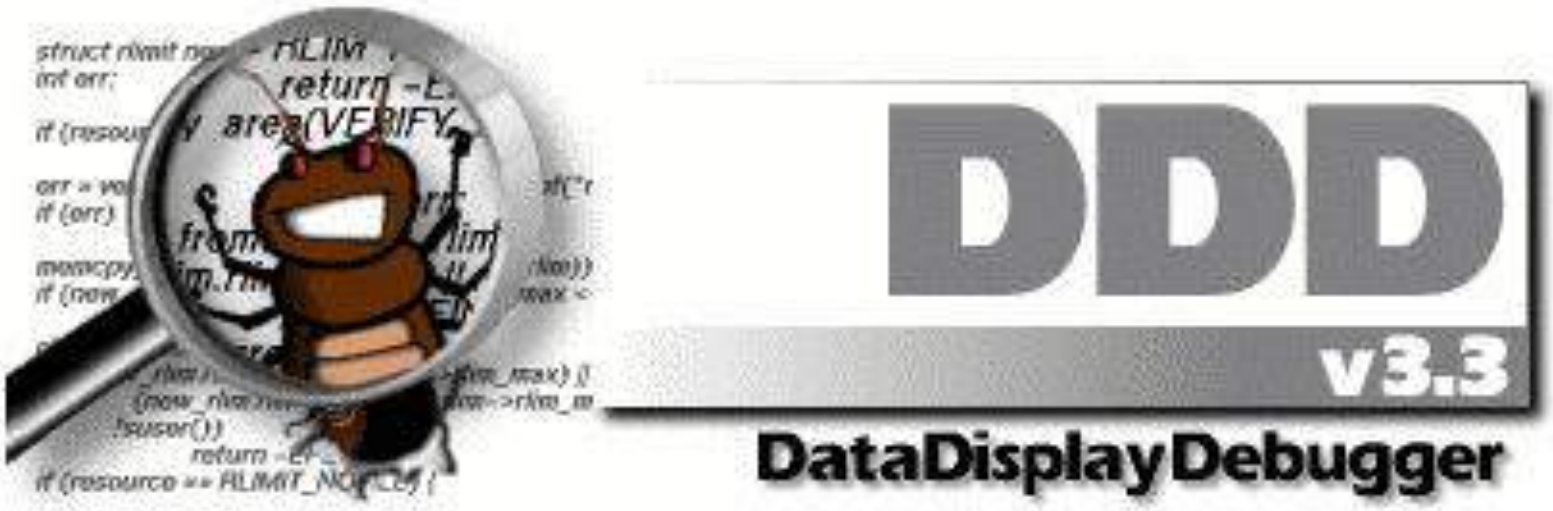
- Что такое DDD?
- Распределенные приложения
- Проблемы DDD в распределенных приложениях
- Как быть?
- Принимаем решение
- Подводим итоги

CUSTIS®

ЧТО ТАКОЕ DDD?

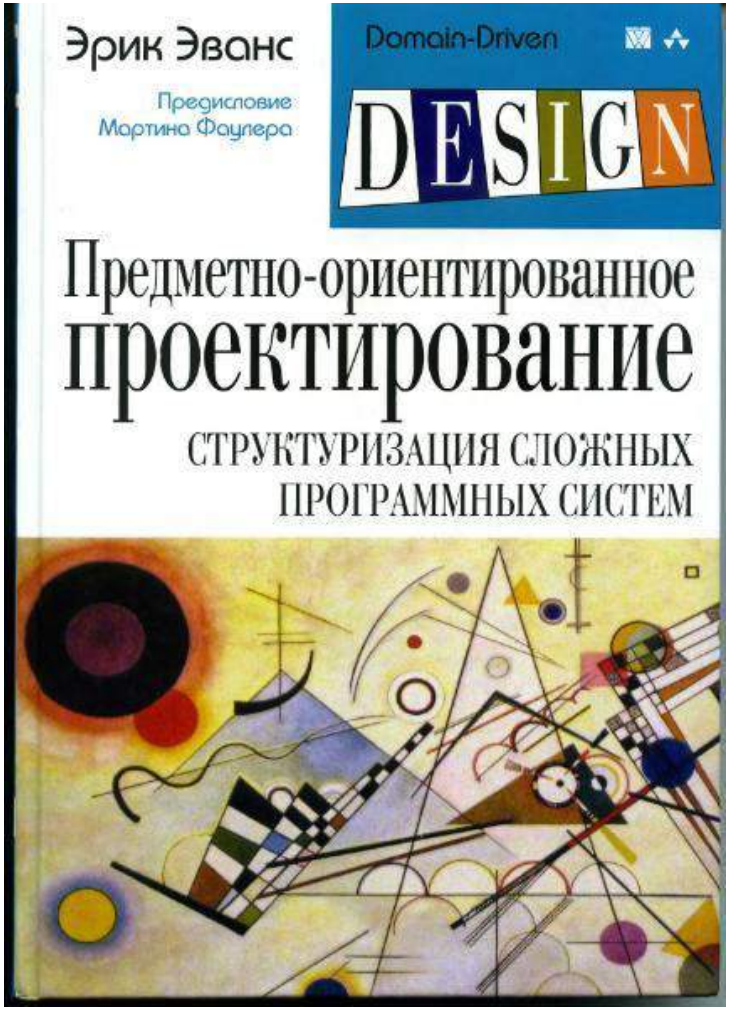
CUSTIS®

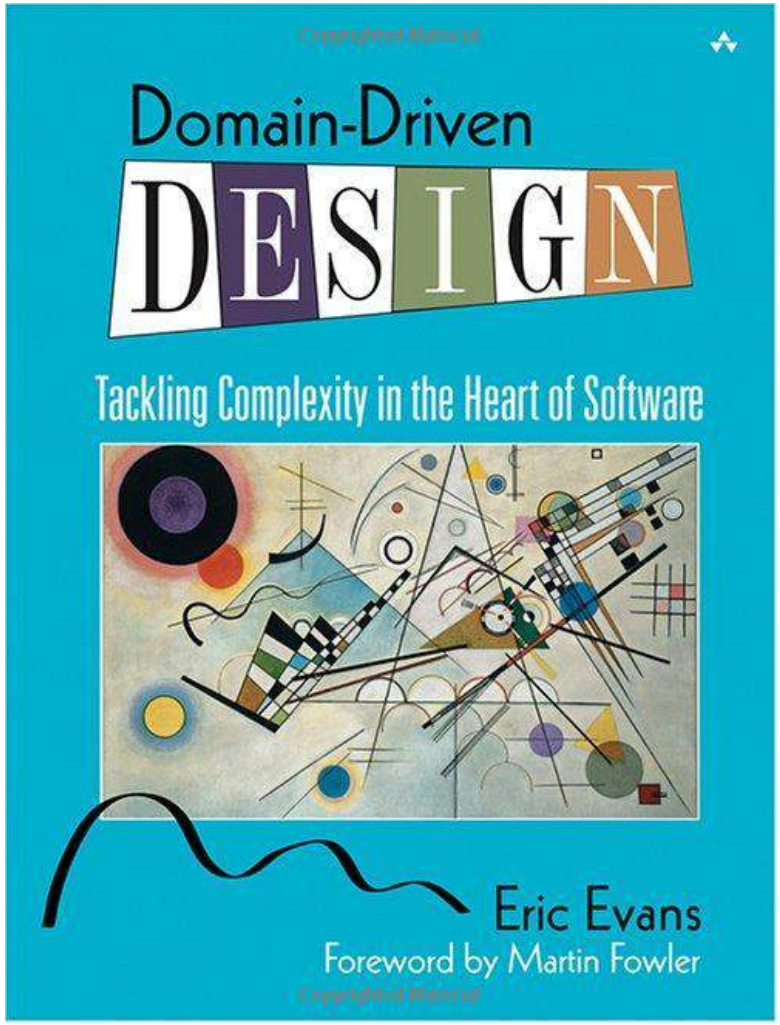
DDD – DataDisplayDebugger?



CUSTIS®

DDD – Domain Driven Design!





CUSTIS®

Domain Driven Design

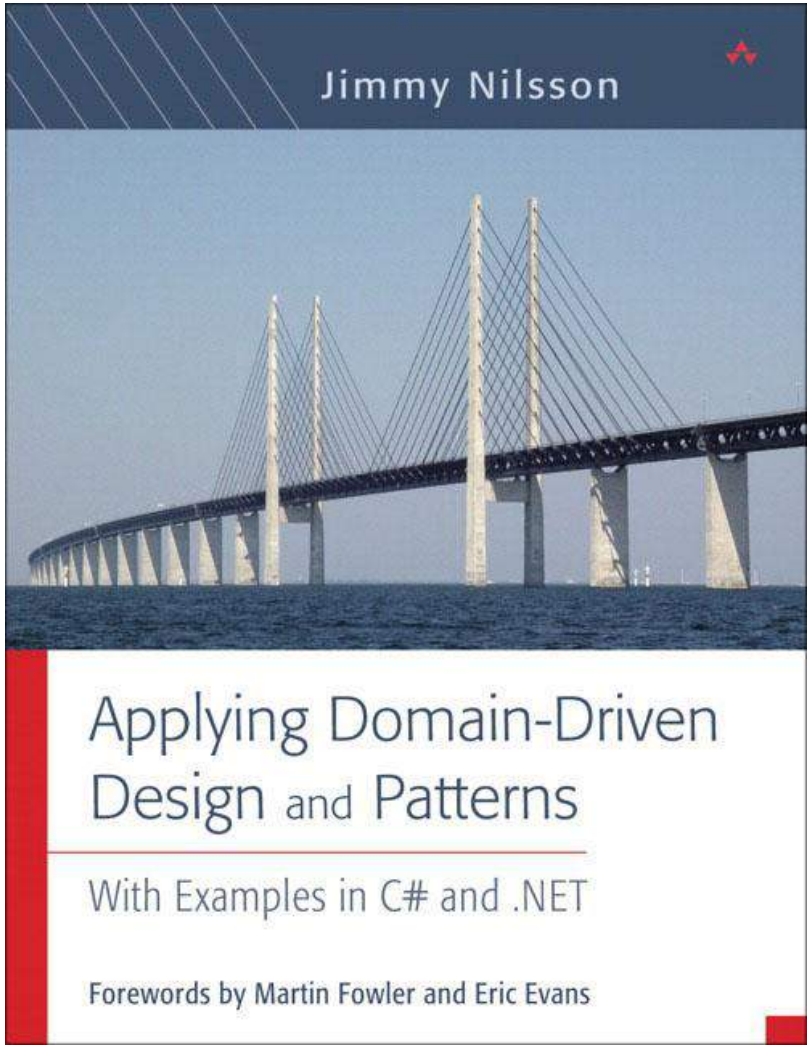
CUSTIS®

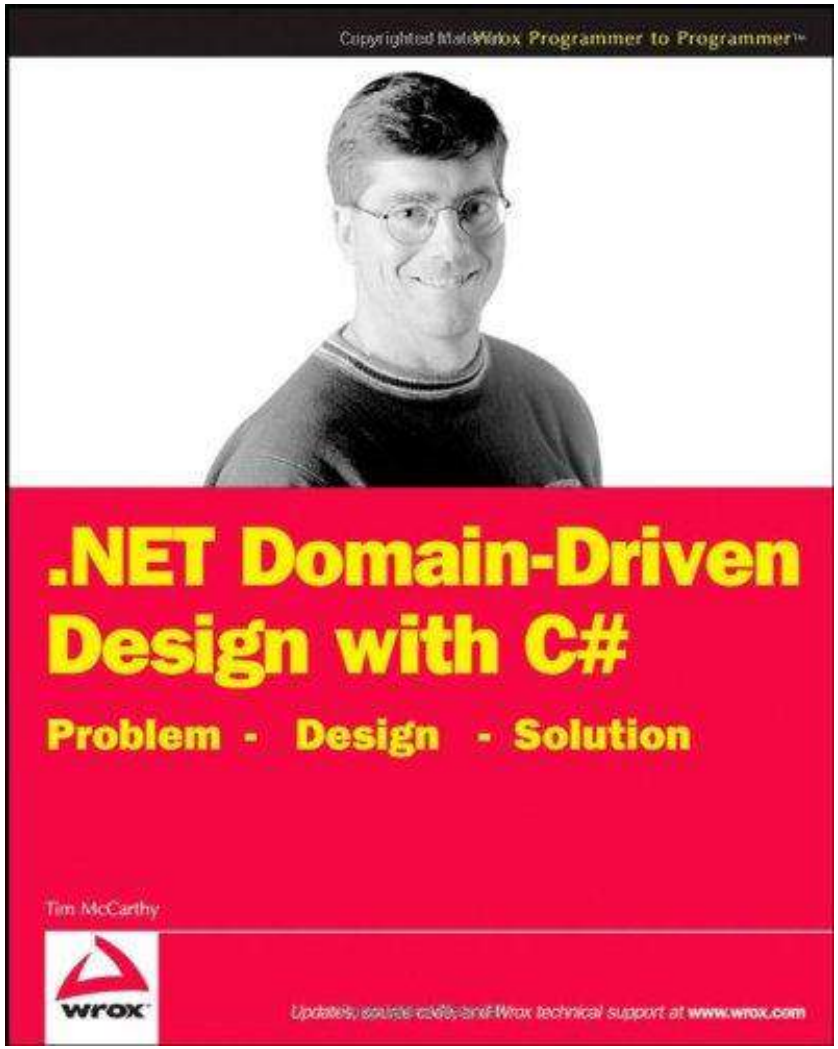
Domain Driven Design

Проектирование по модели – проектирование архитектуры, при котором соблюдается максимально точное соответствие между некоторым подмножеством элементов программы и элементами модели (Эрик Эванс)

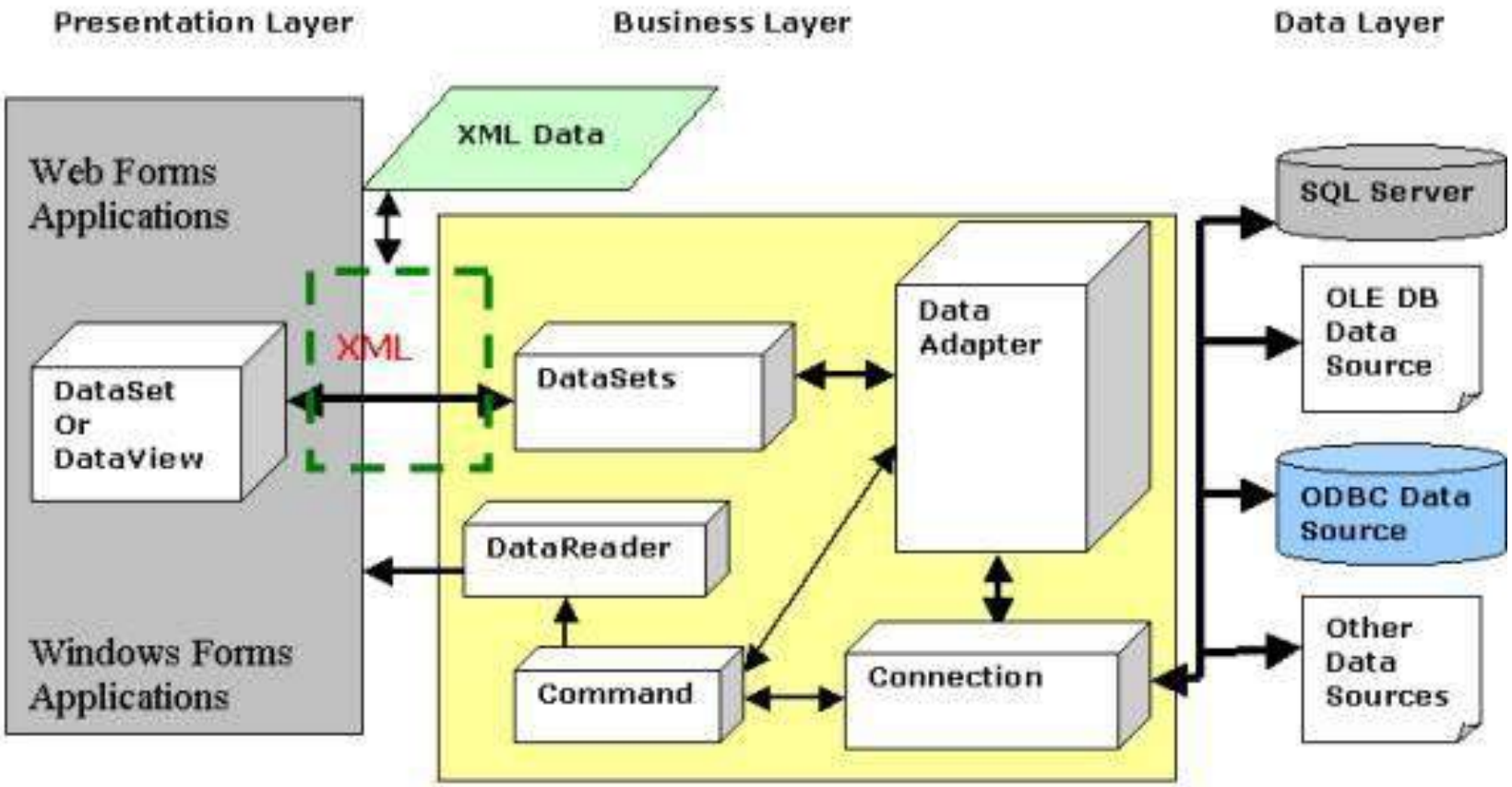
CUSTIS®

Domain Driven Design





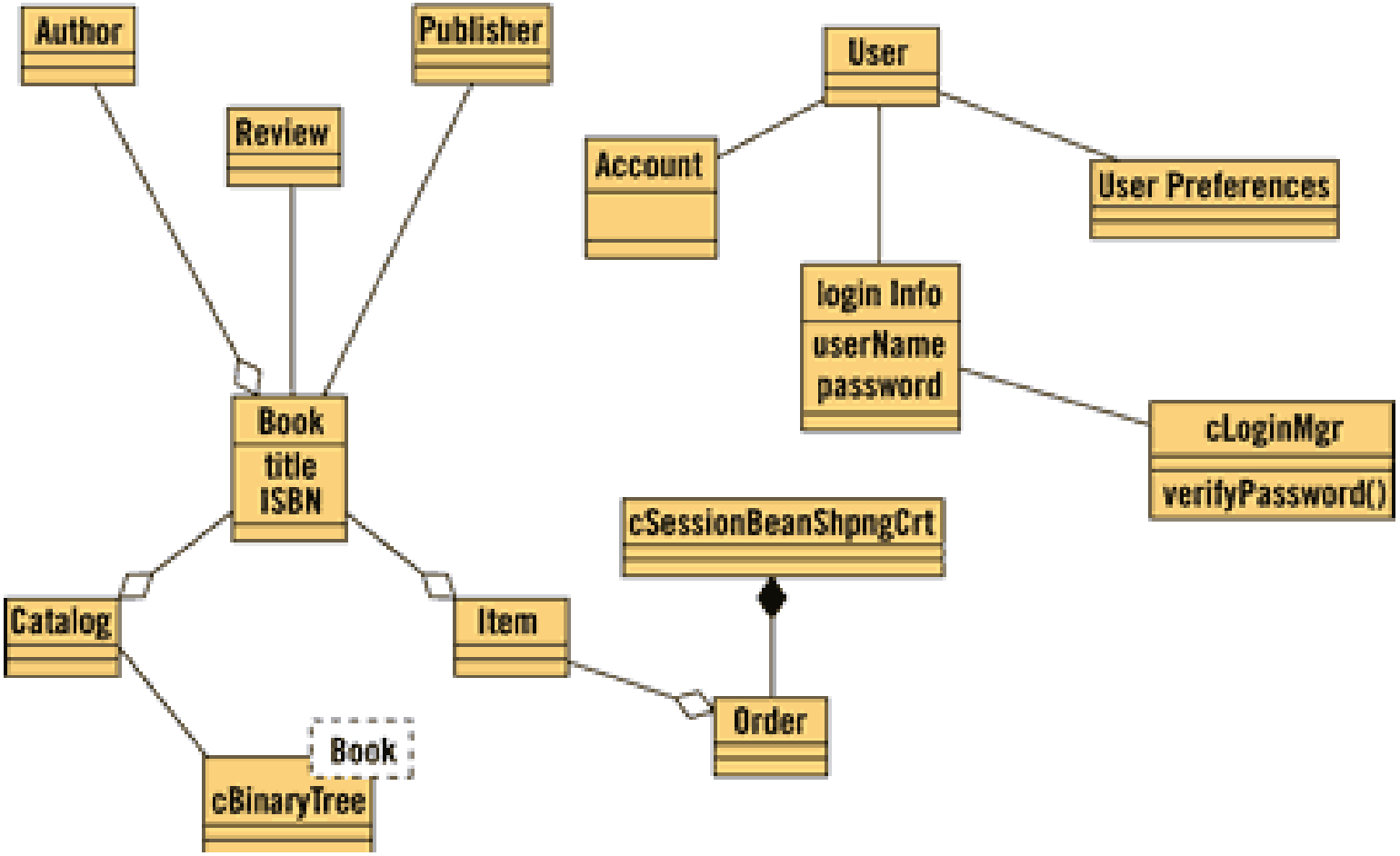
Модель программы



ADO.NET Components in Windows DNA Architecture

CUSTIS®

Модель предметной области



CUSTIS®

Модель программы

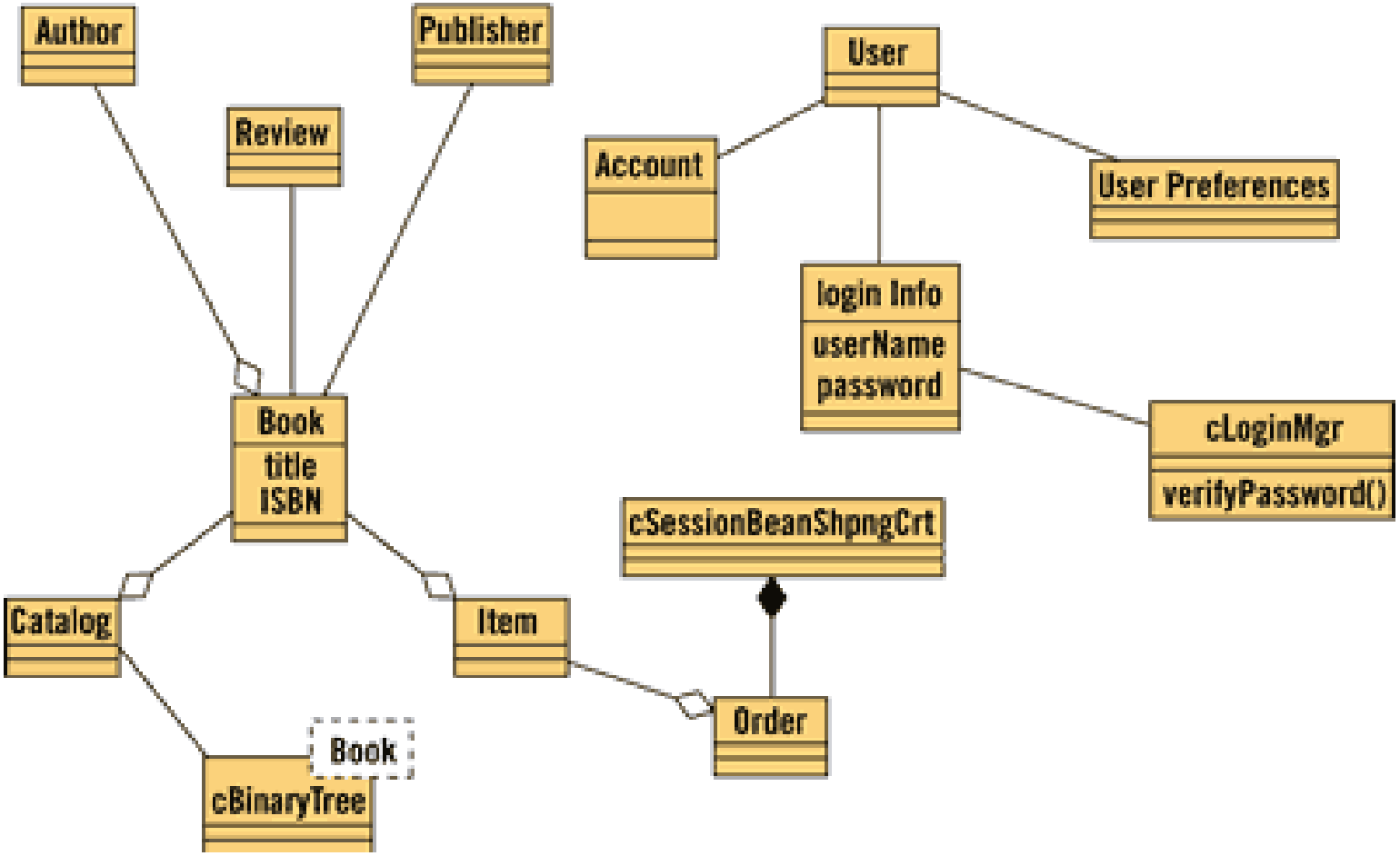
- DataSet
- DataReader
- Command
- DataAdapter
- Connection
- И т. д.

Модель предметной области

- Книга
- Автор
- Издатель
- Читатель
- И т. д.

CUSTIS®

Модель предметной области



CUSTIS®

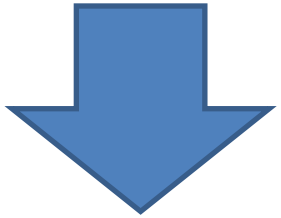
Domain Driven Design

DDD – проекция языка предметной области на объектно ориентированный язык программирования

CUSTIS®

Почему DDD?

- Эффективный способ борьбы со сложностью
- Единый язык



- Низкая стоимость разработки и сопровождения

CUSTIS®

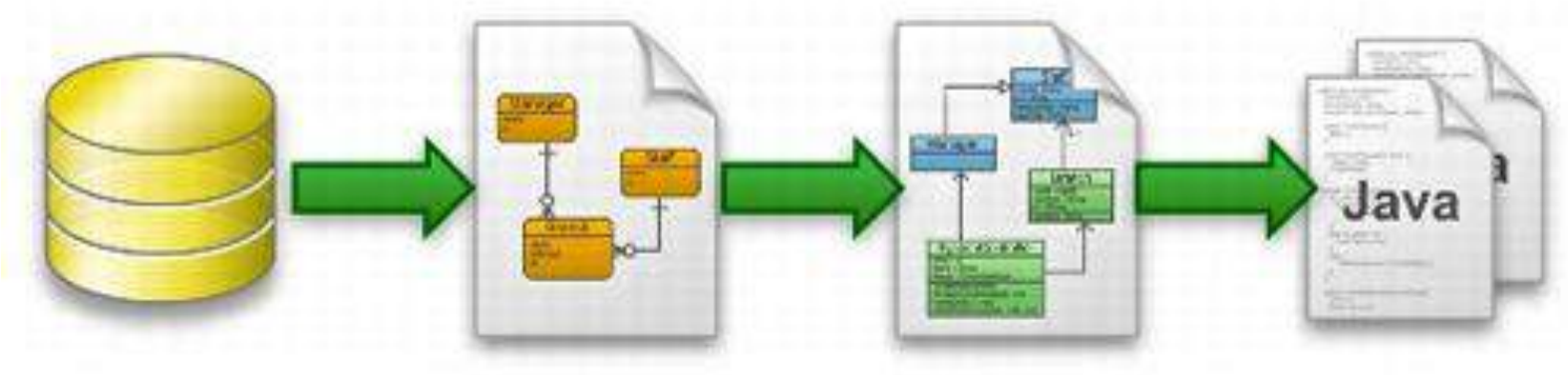
Почему DDD в Agile?

- Расстояние между заказчиком и разработчиком невелико
- Заказчик и разработчик разговаривают на одном языке
- Итеративная разработка – постепенное изменение модели

CUSTIS®

ORM (Object-relational mapping)

ORM – технология программирования для конвертации данных между несовместимыми типами систем в объектно-ориентированные языки



CUSTIS®

ORM

CUSTIS®

ORM ≠ DDD

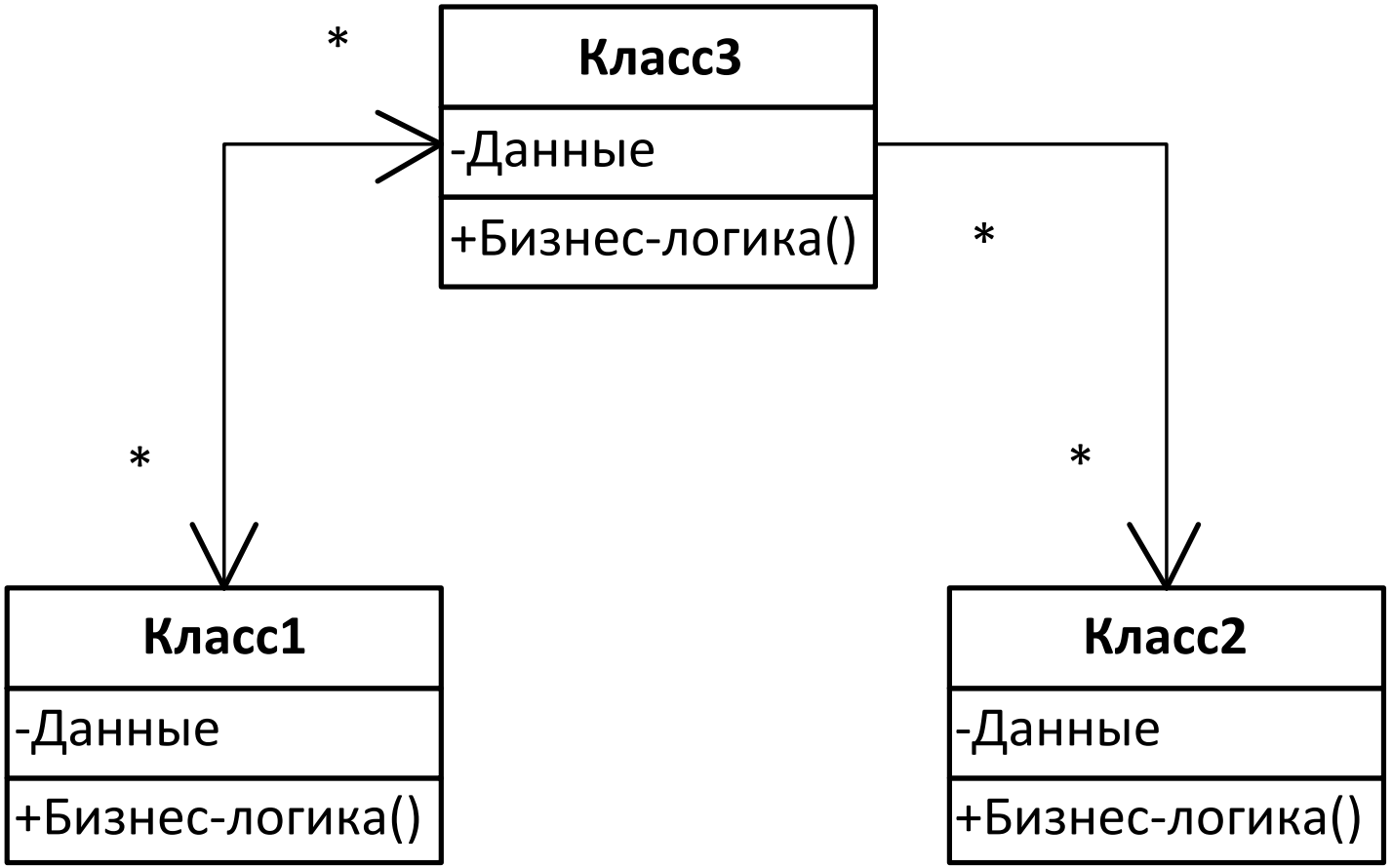
CUSTIS®

Архитектурные стили при работе с ORM

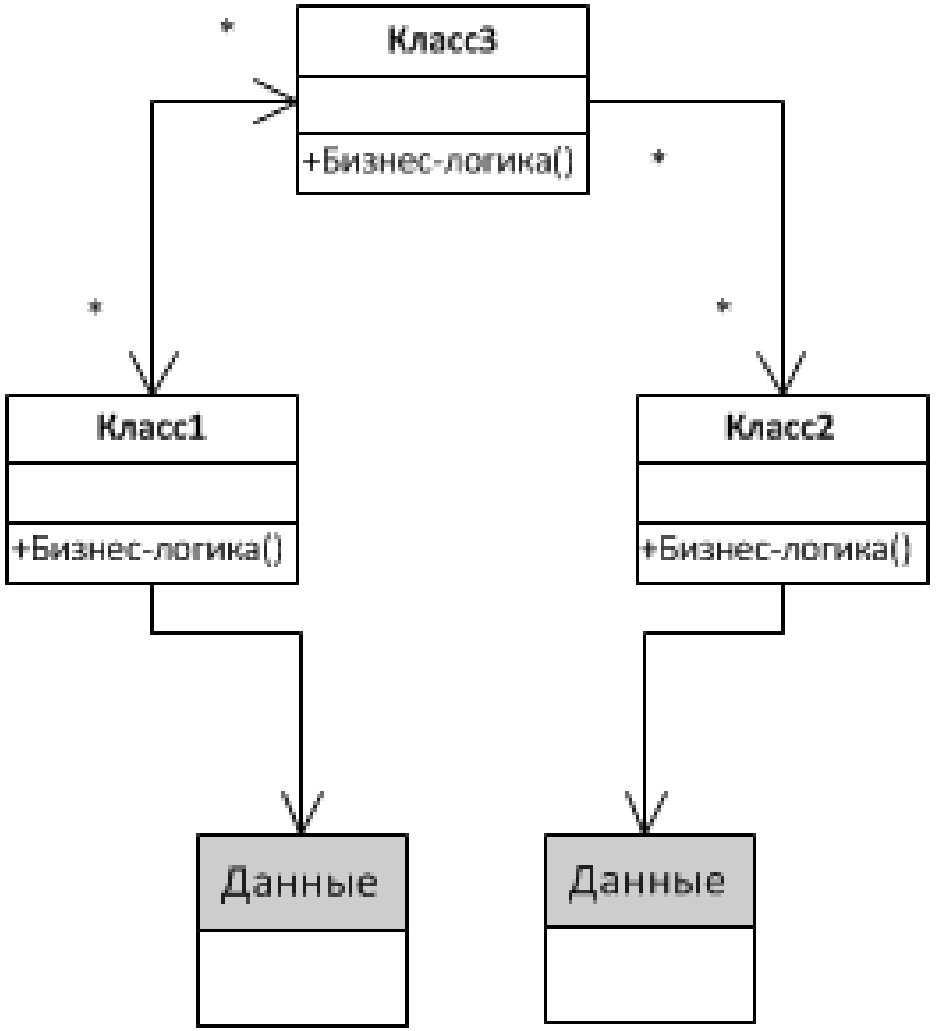
CUSTIS®

Rich VS Anemic

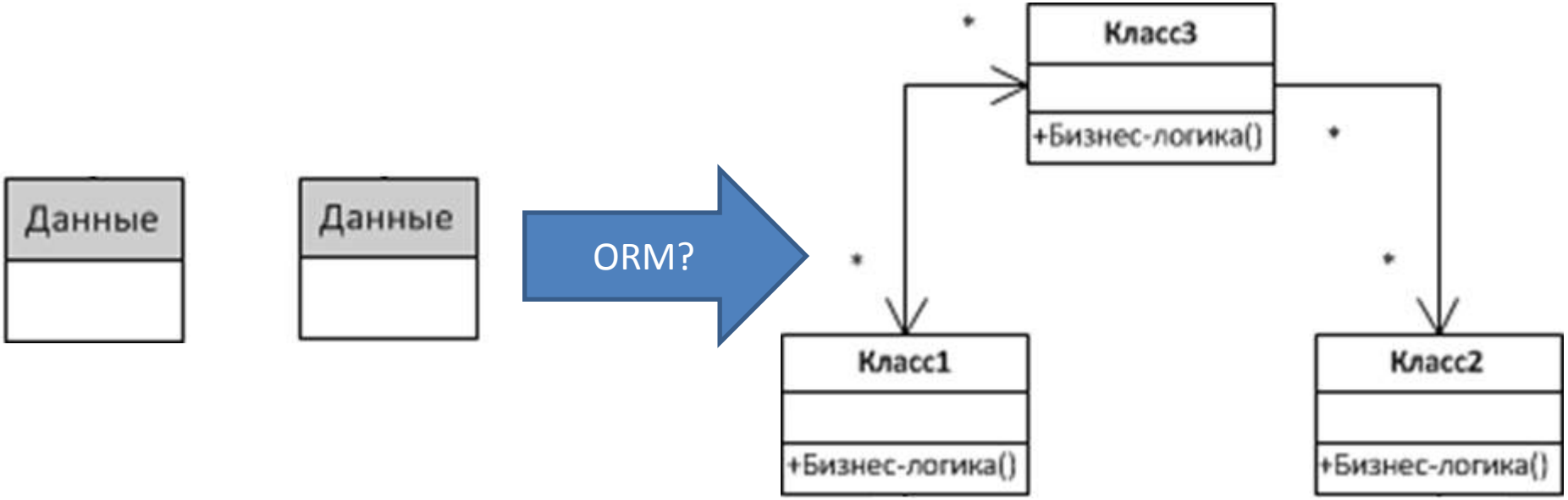
Rich


*
Класс3
-Данные
+Бизнес-логика()
*
*
*
Класс1
-Данные
+Бизнес-логика()
Класс2
-Данные
+Бизнес-логика()

Anemic


Класс3
+Бизнес-логика()
*
*
*
*
Класс1
+Бизнес-логика()
Класс2
+Бизнес-логика()
Данные
Данные

Anemic?


Данные
Данные
ORM?
Класс3
+Бизнес-логика()
Класс1
+Бизнес-логика()
Класс2
+Бизнес-логика()

CUSTIS®

Rich!

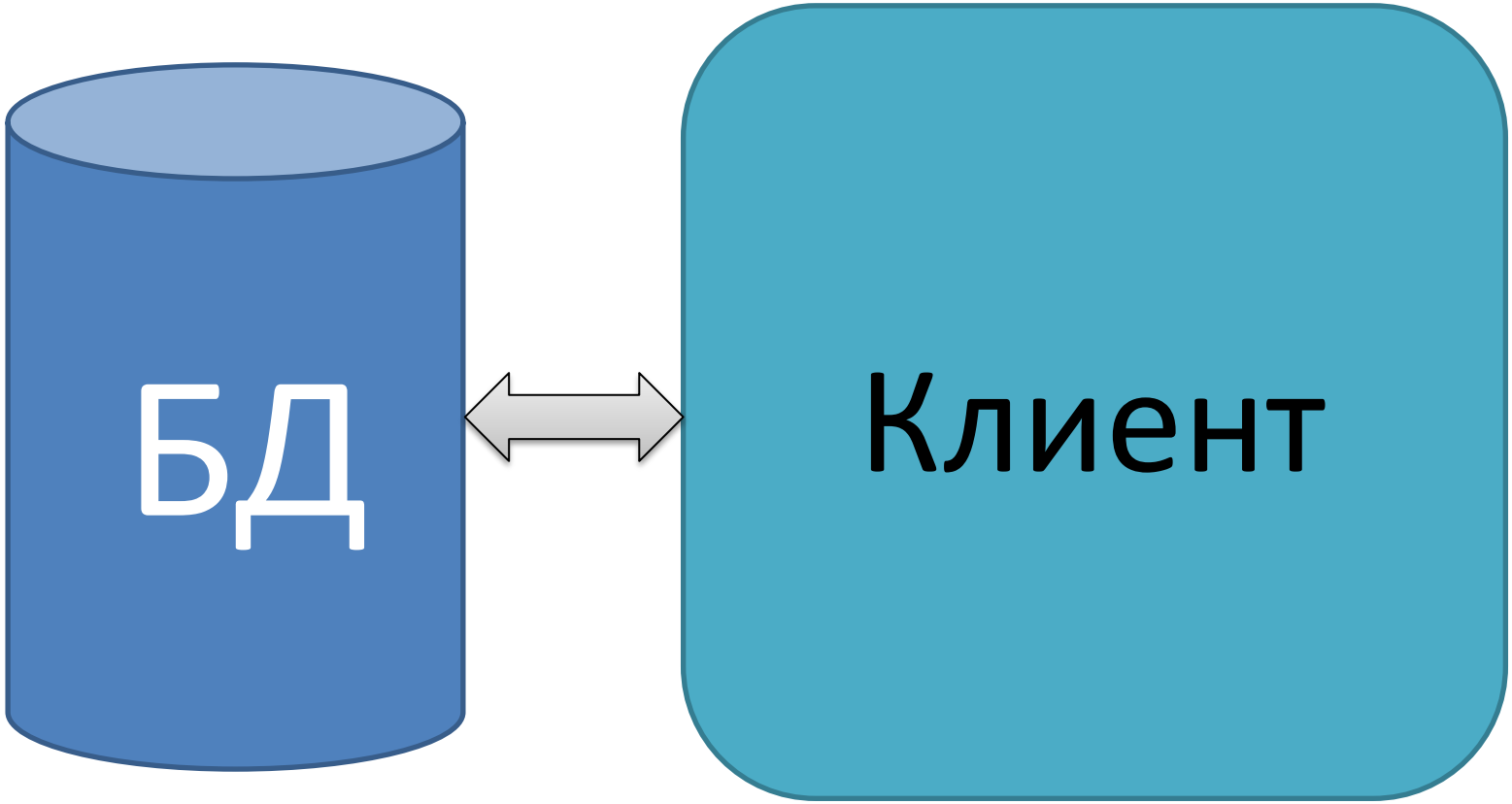
CUSTIS®

РАСПРЕДЕЛЕННЫЕ ПРИЛОЖЕНИЯ

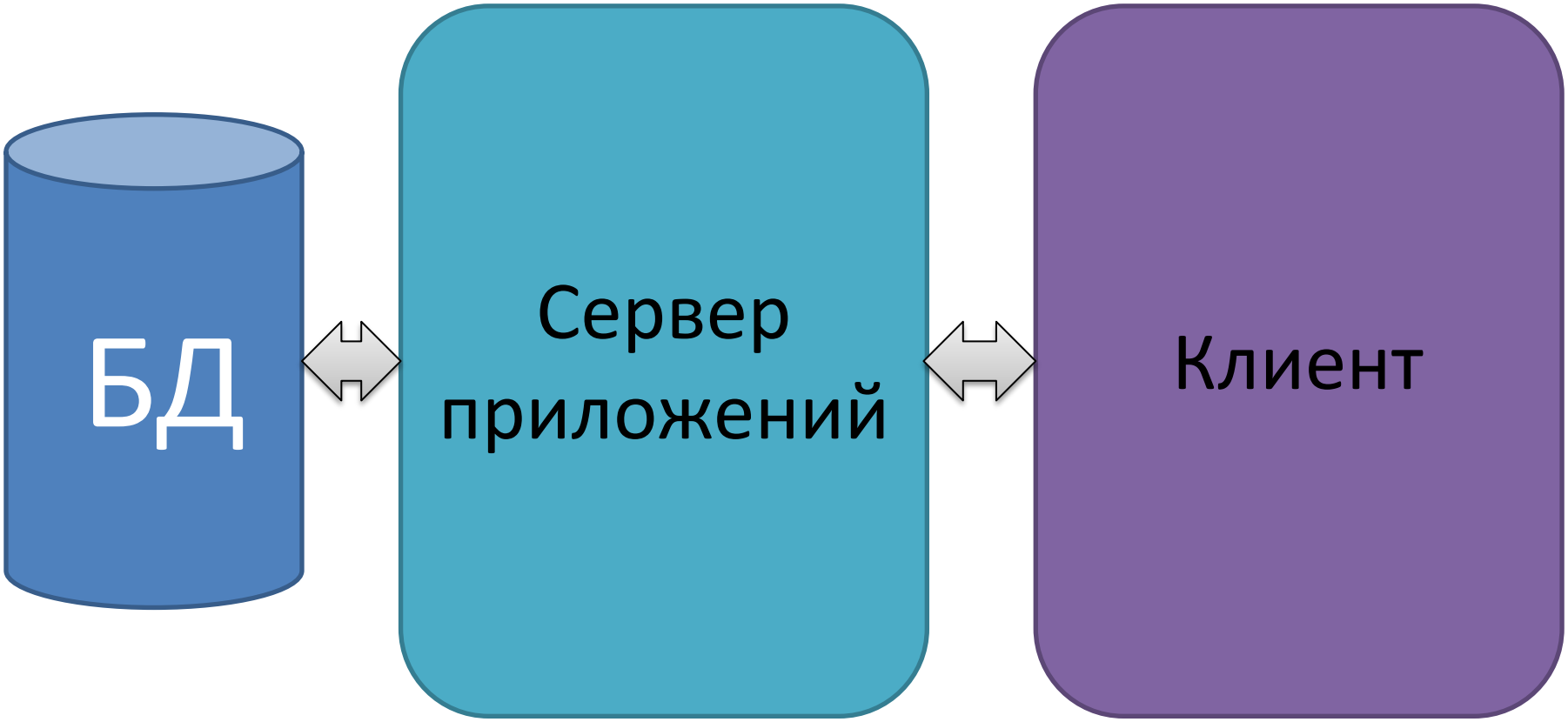
Шаблоны распределенных архитектур

- Клиент-Сервер
- 3-х звенная (многозвенная) архитектура
- SOA
- Enterprise Service Bus

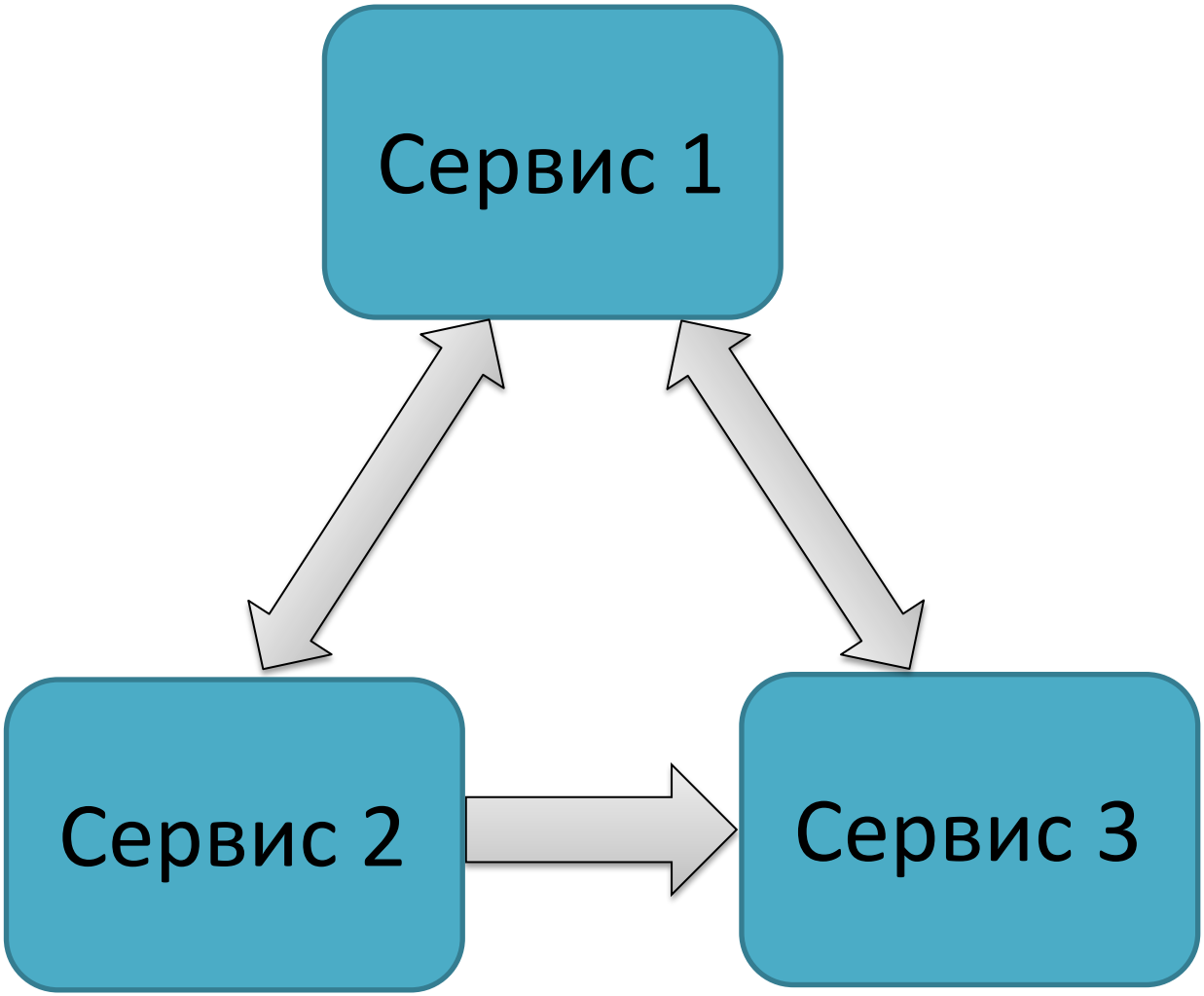
Клиент-Сервер



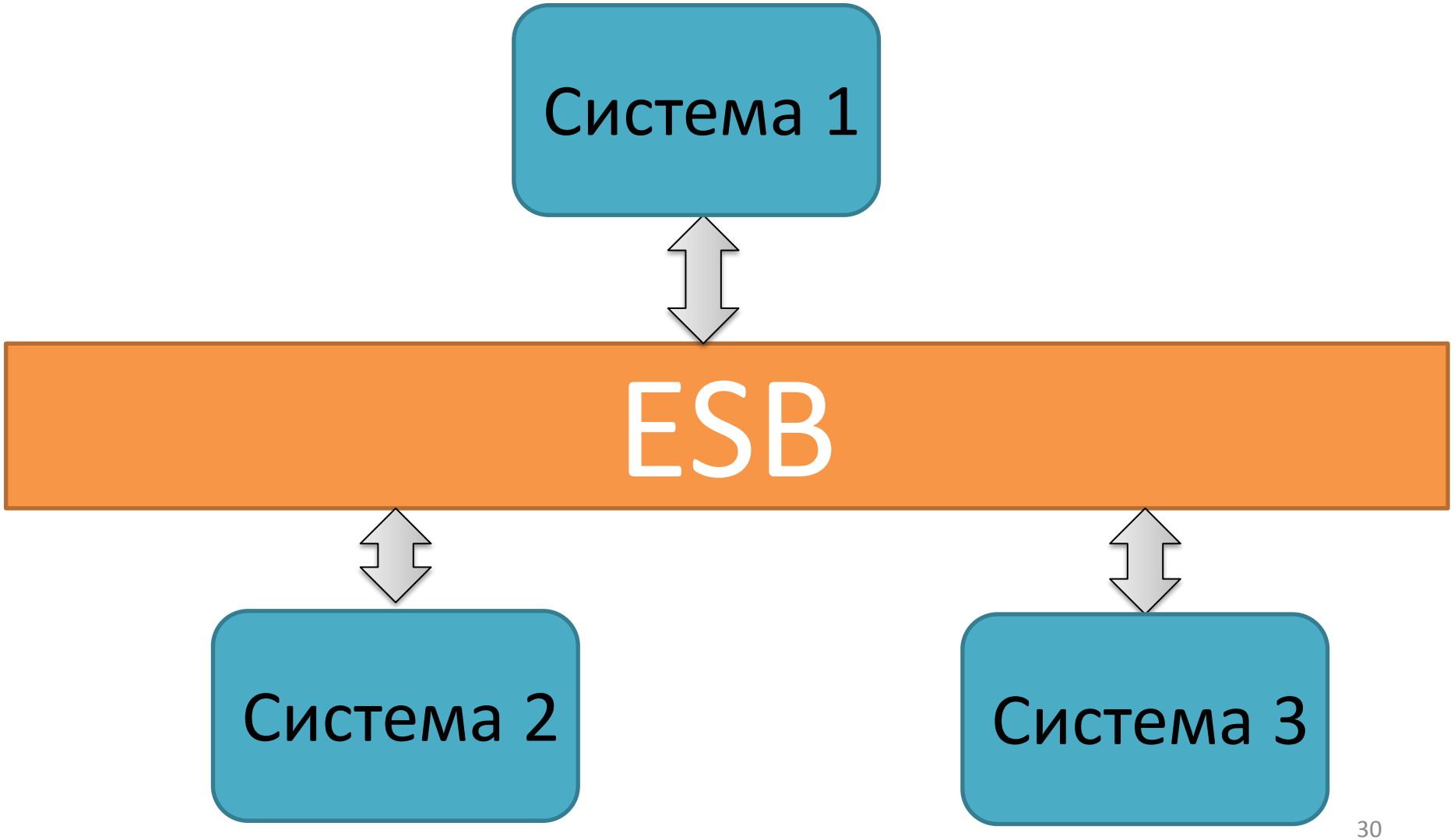
3-х звенная архитектура



SOA



Enterprise Service Bus



CUSTIS®

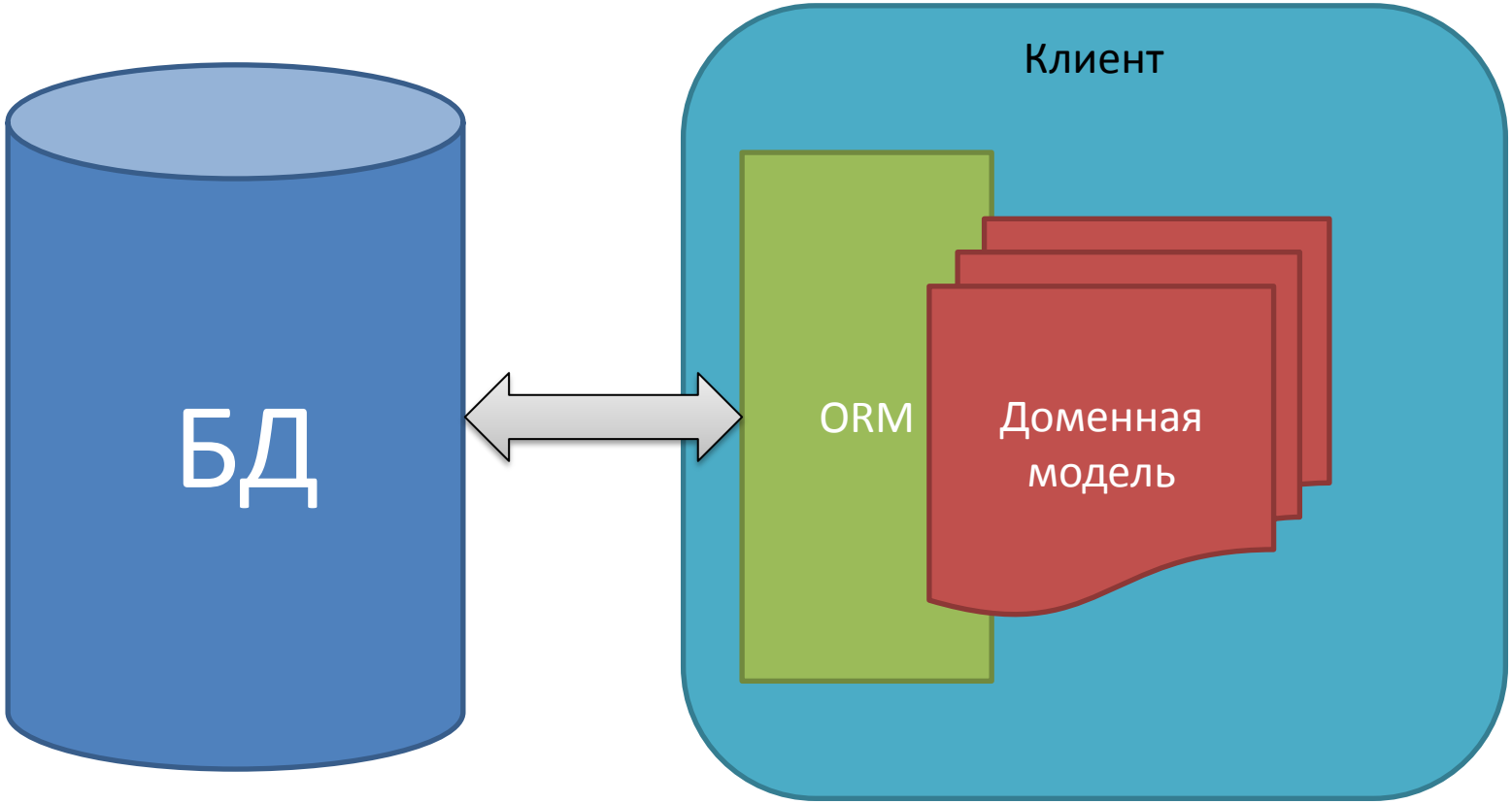
Почему распределенная архитектура?

- Безопасность
- Эффективность
- Совместная работа

CUSTIS®

DDD В РАСПРЕДЕЛЕННЫХ ПРИЛОЖЕНИЯХ

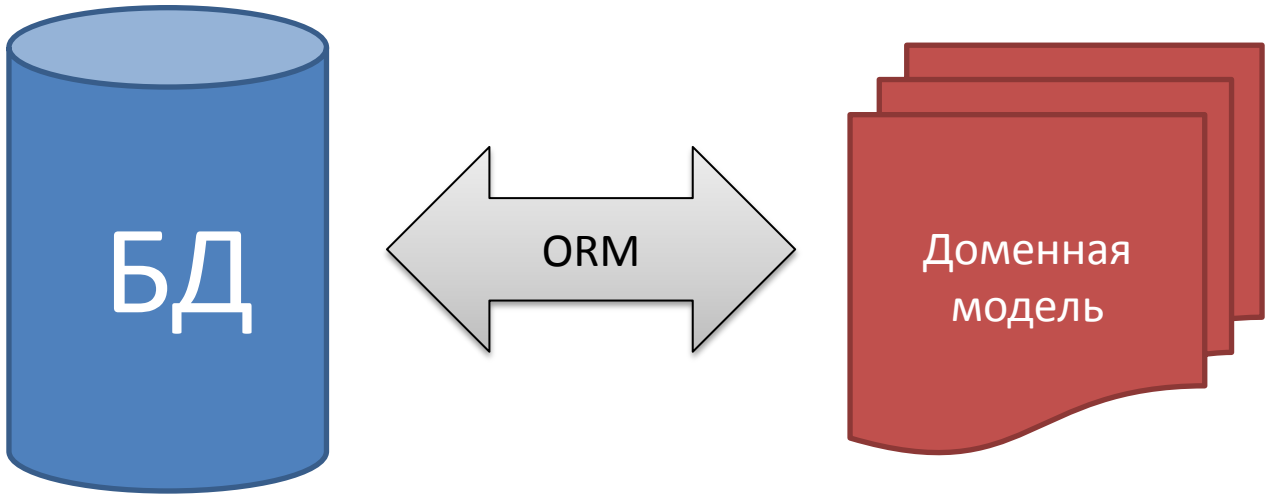
Клиент-Сервер



Проблемы?

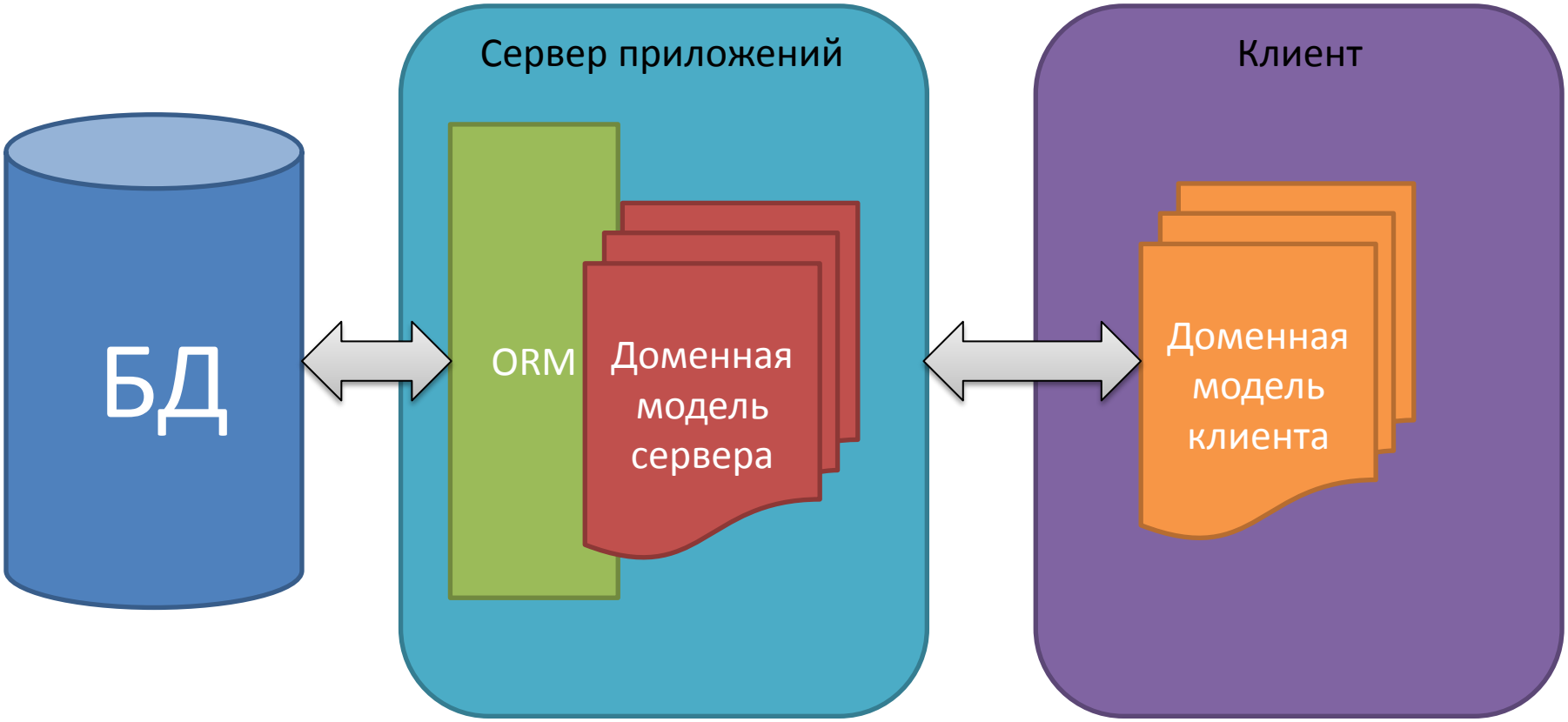
CUSTIS®

Нет проблем!



CUSTIS®

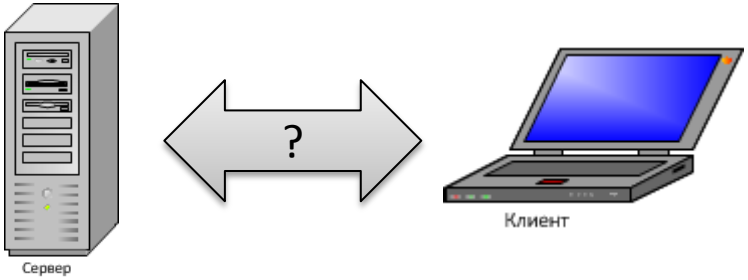
3-х звенная архитектура



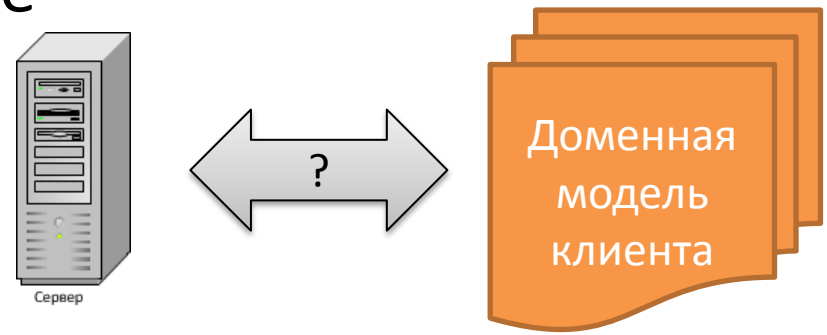
Проблемы?

Проблемы

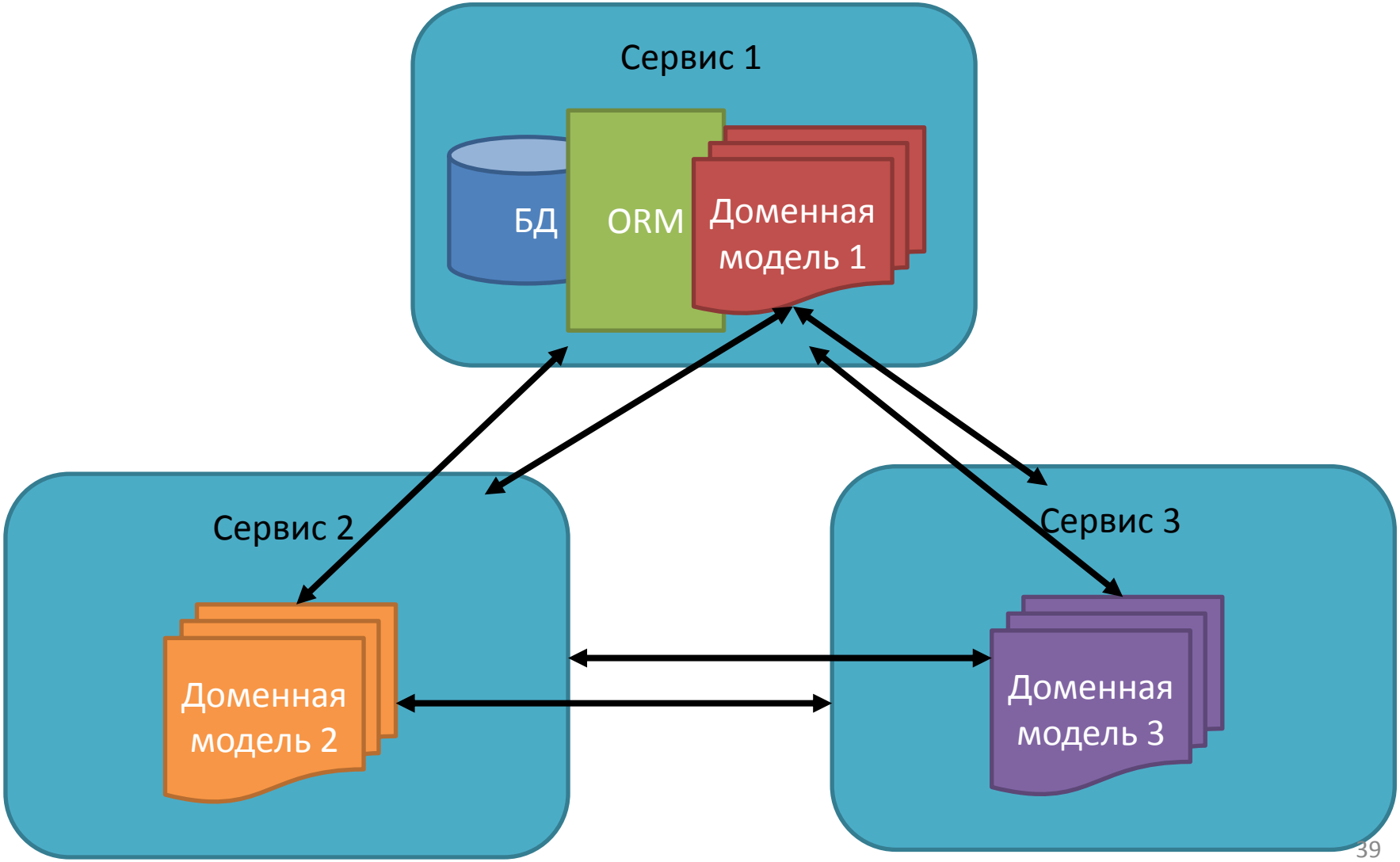
- Как?
  - Построить взаимодействие с сервером



  - Преобразовать данные в доменную модель на клиенте



SOA

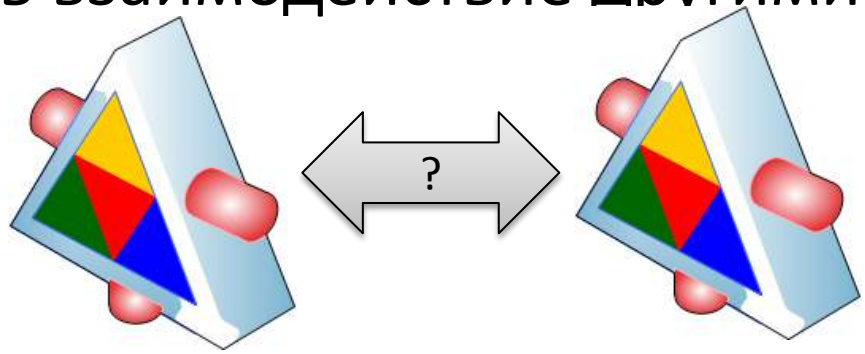

Сервис 1
БД
ORM
Доменная модель 1
Сервис 2
Доменная модель 2
Сервис 3
Доменная модель 3

Проблемы?

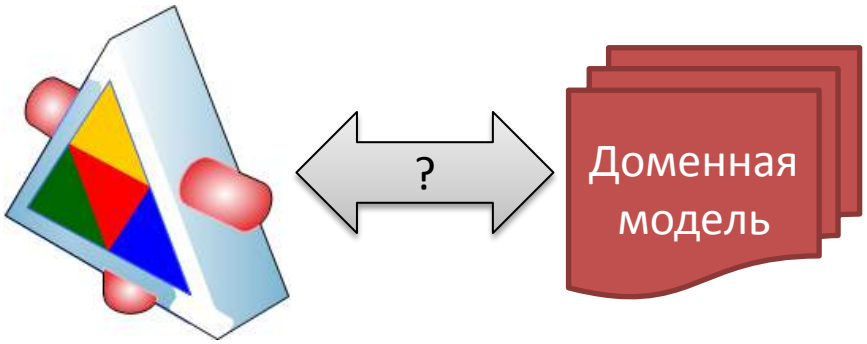
CUSTIS®

Проблемы

- Как?
  - Построить взаимодействие другими сервисами

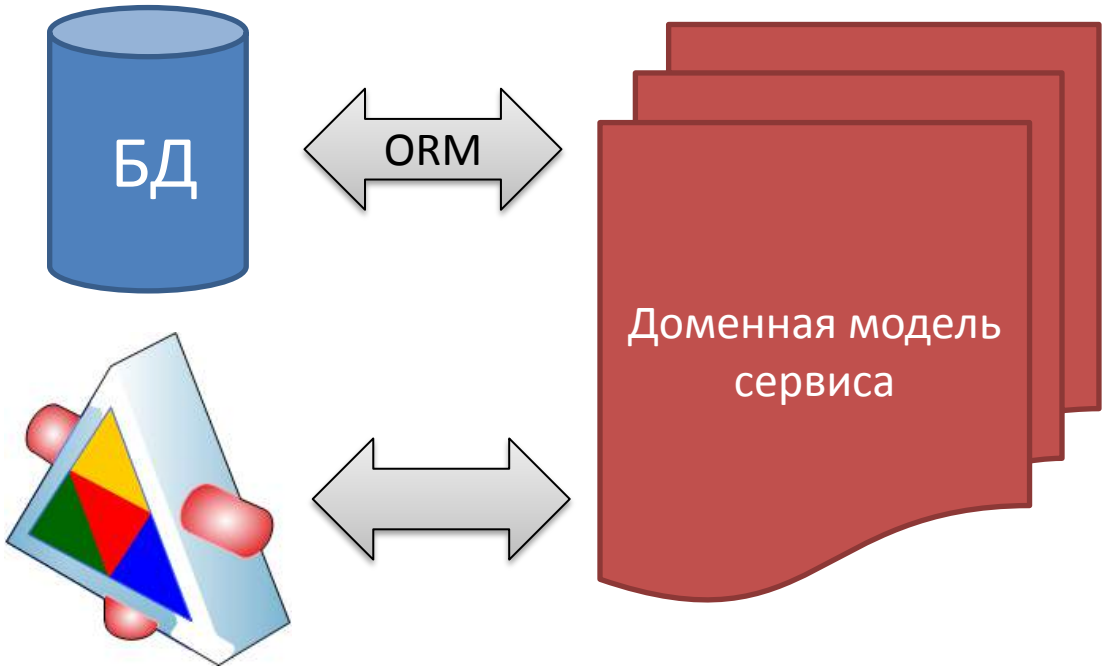


  - Преобразовать данные в локальную доменную модель



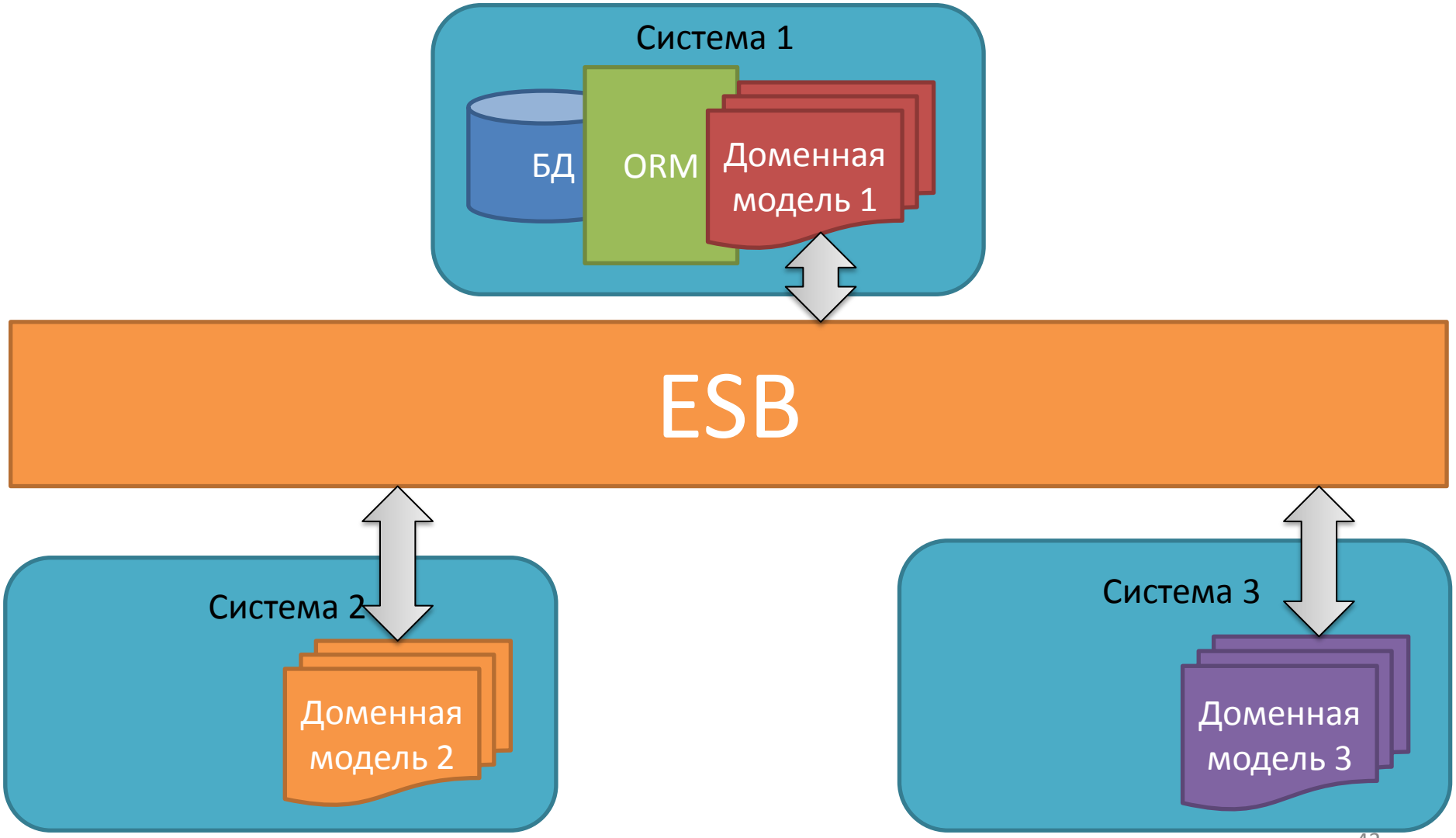
Проблемы

- Как?
  - Сочетать использование ORM и работу с другими сервисами?



CUSTIS®

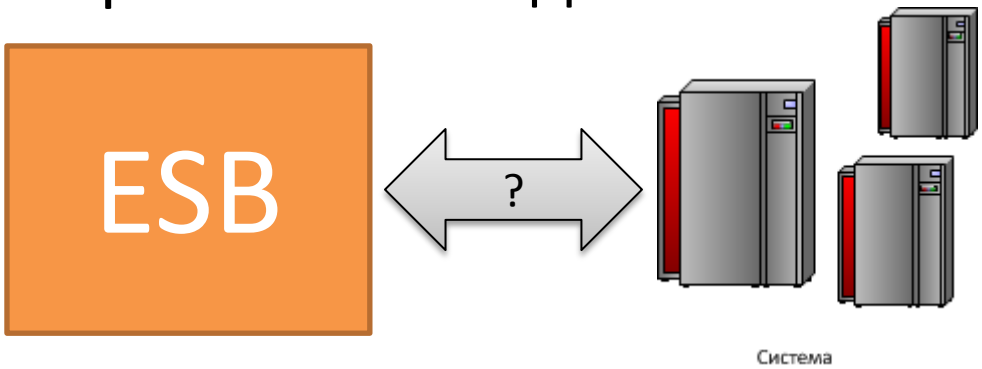
Enterprise Service Bus



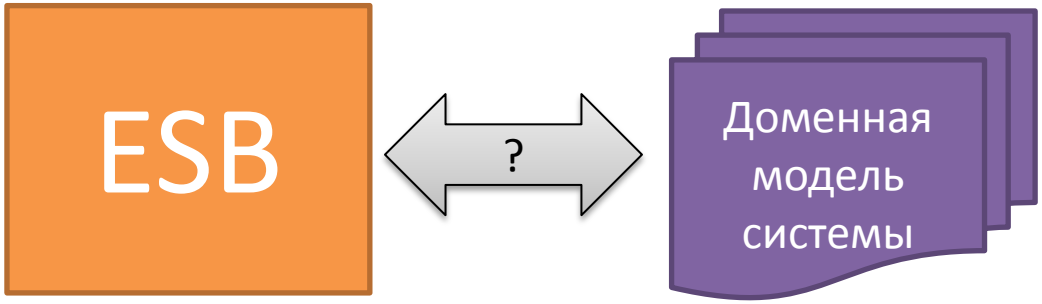
Проблемы?

Проблемы

- Как?
  - Построить взаимодействие с шиной



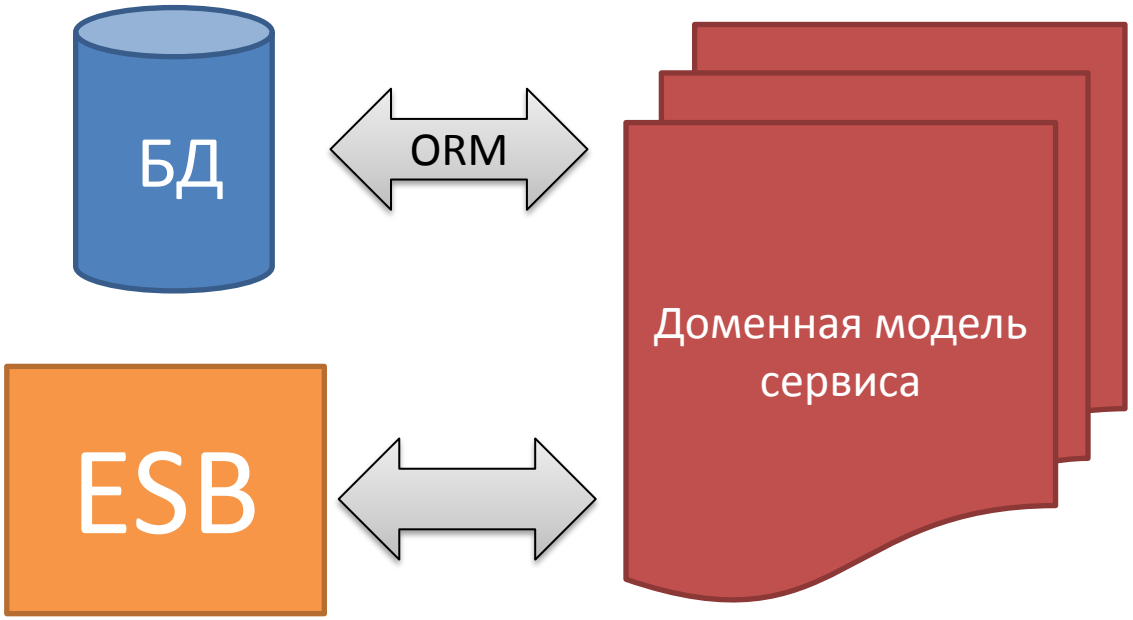
  - Преобразовать данные из шины в доменную модель



CUSTIS®

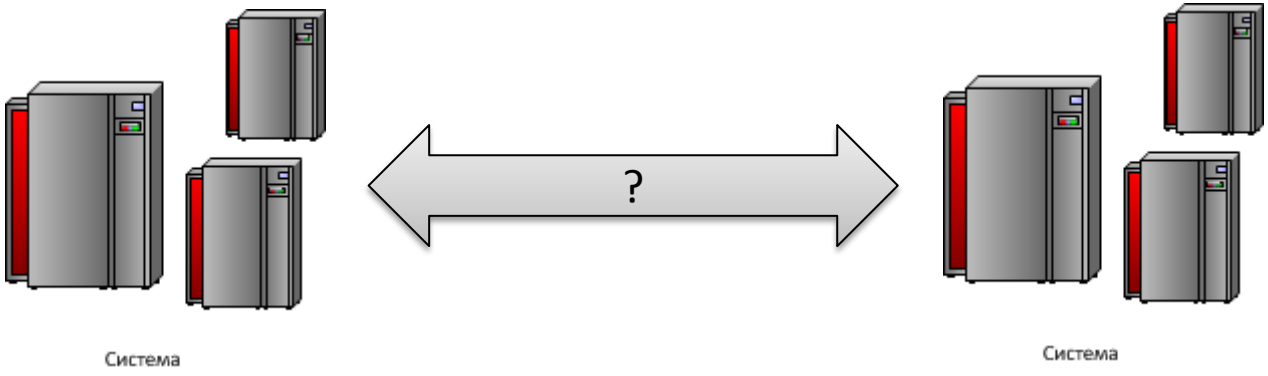
Проблемы

- Как?
  - Сочетать использование ORM и работу с шиной



Проблемы в DDD в распределенных приложениях

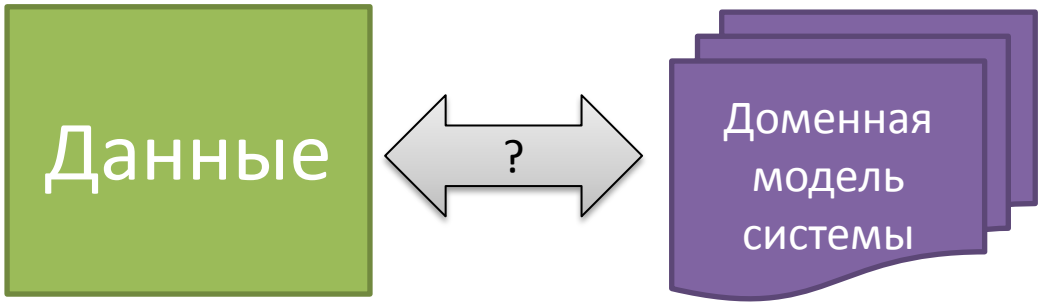
- Взаимодействие с удаленными источниками данных



CUSTIS®

Проблемы в DDD в распределенных приложения

- Преобразование данных из удаленных источников в доменную модель



CUSTIS®

Соединение

Сервис временно недоступен

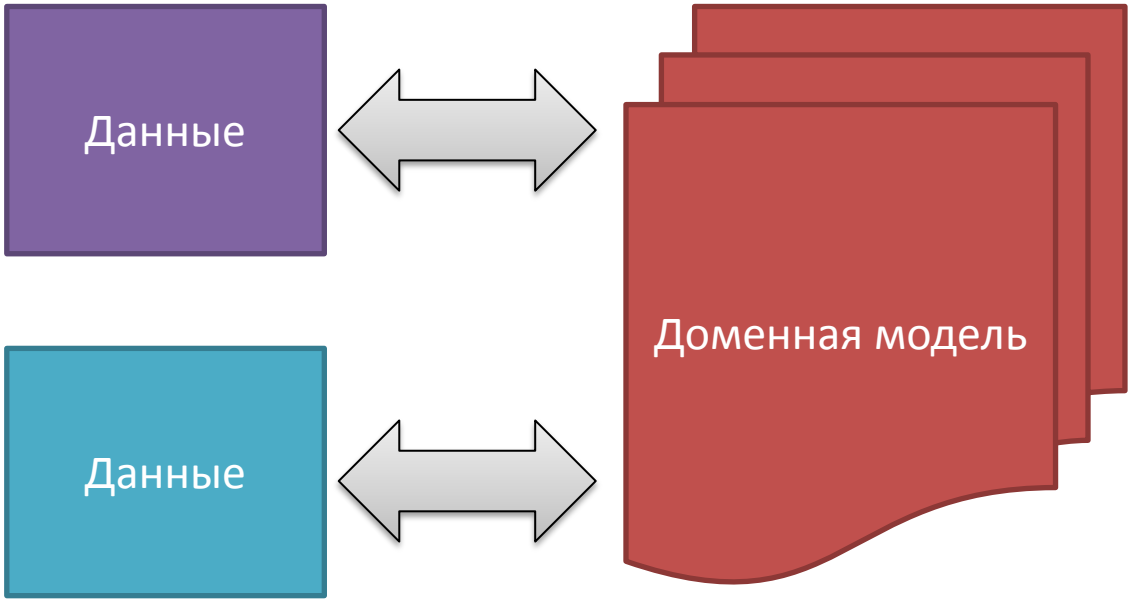
Распределенная архитектура вносит в модель объекты отсутствующие в предметной области

Данные по запросу

CUSTIS®

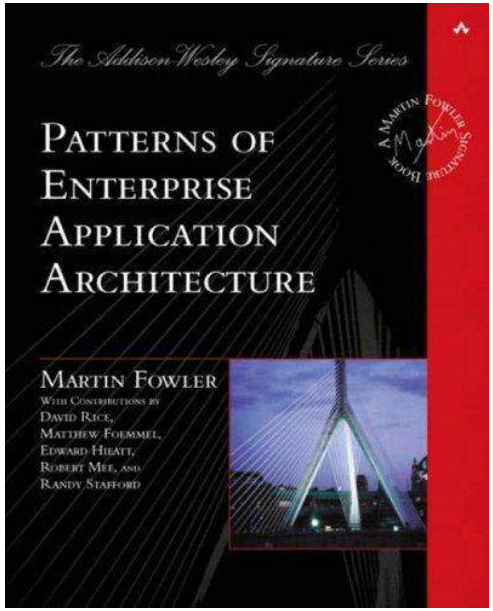
Проблемы в DDD в распределенных приложения

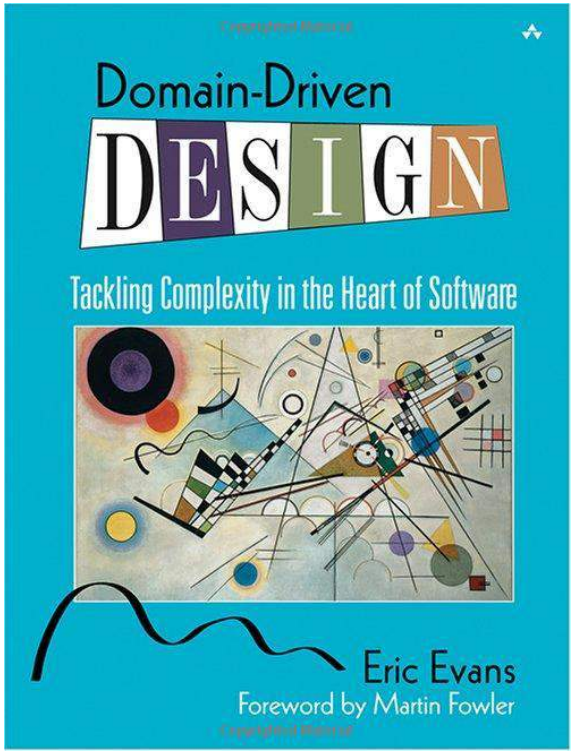
- Сочетание различных источников данных



CUSTIS®

Развитие DDD





Microsoft .NET Entity Framework

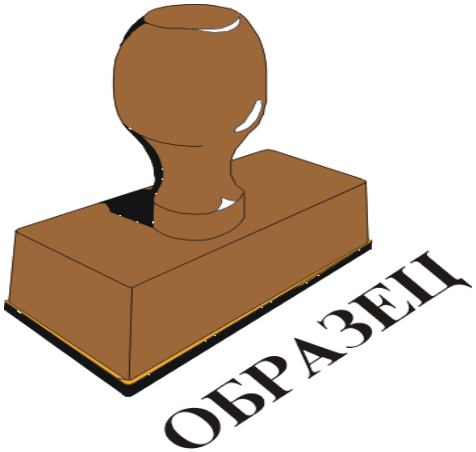
2002 год

2003 год

2008 год

Причины

- Отсутствие шаблонов для удаленного взаимодействия



- Отсутствие готовых инструментальных средств

КАК БЫТЬ?

CUSTIS®

Шаблоны удаленного взаимодействия

RPC (Remote Procedure Call)

- .NET Remoting
- CORBA
- WCF
- и т. д.

CUSTIS®

Почему нет?

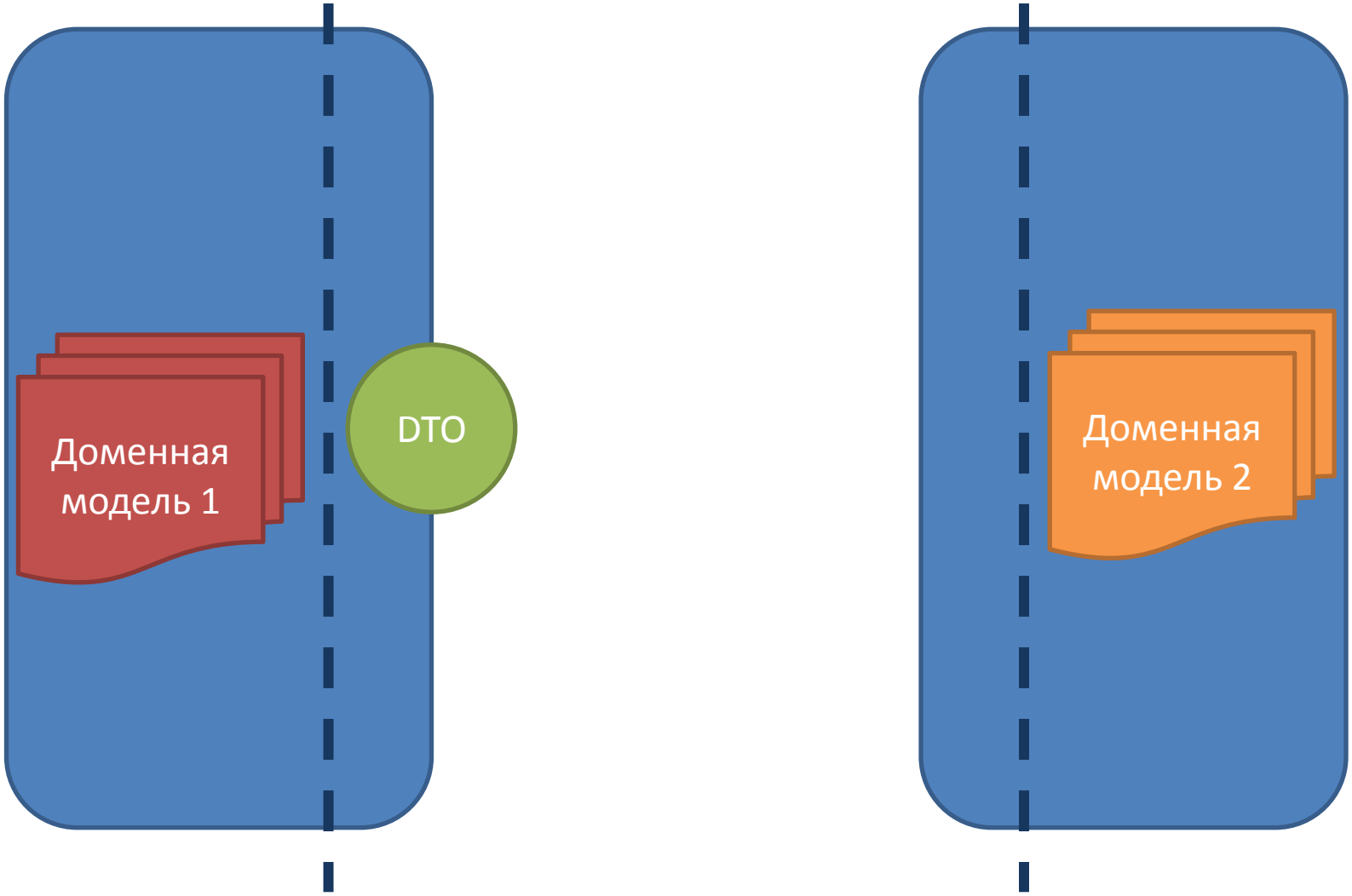
CUSTIS®

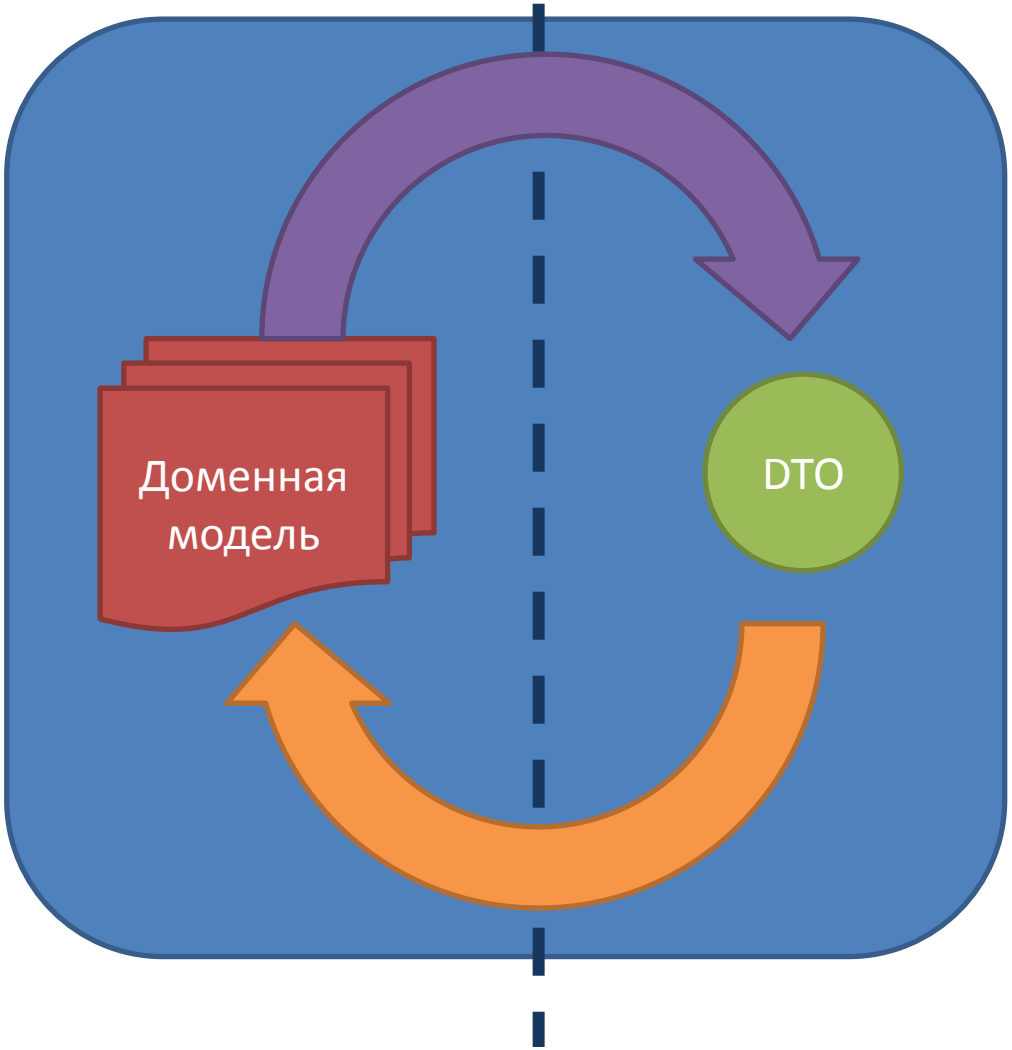
Почему нет?

- Доменные модели разные
- Стоимость удаленного вызова на порядки выше стоимости локалького

CUSTIS®

Data Transfer Object


Доменная модель 1
DTO
Доменная модель 2


Доменная модель
DTO

Инструменты

CUSTIS®

Что делать?

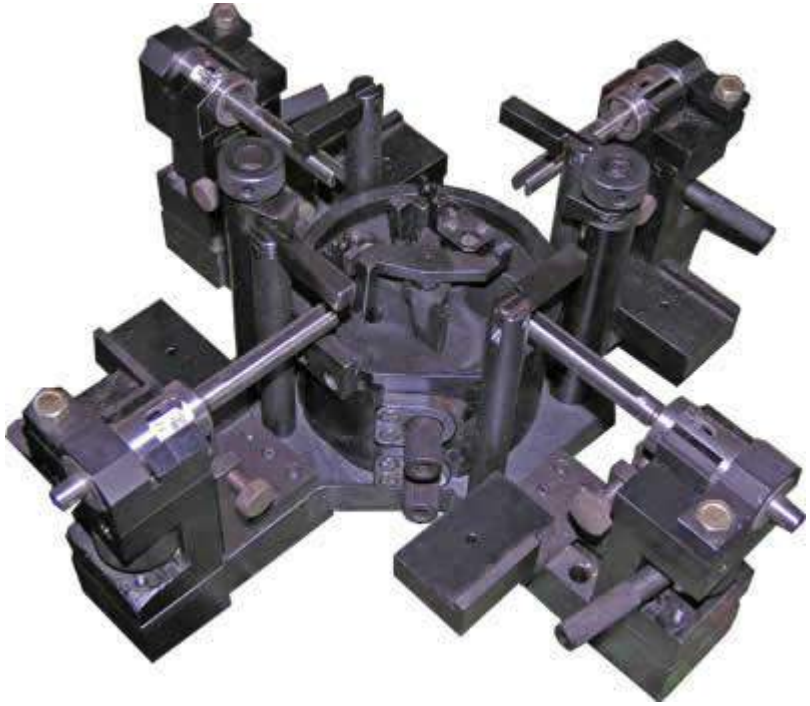
Логика преобразования данных и удаленного взаимодействия

Модель предметной области

CUSTIS®

Что делать?

Разработать инструментарий

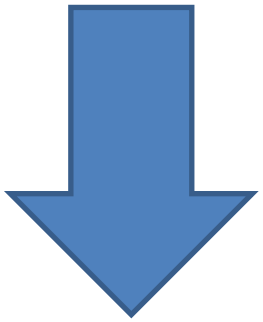


ПРИНИМАЕМ РЕШЕНИЕ

CUSTIS®

Клиент-сервер

- Множество инструментов и платформ
- Готовые архитектурные шаблоны



- DDD – выгодно

CUSTIS®

3-х звенная архитектура

- Тонкий клиент – см. клиент-сервер
- Толстый клиент:
  - Мало логики – не использовать DDD на клиенте
  - Средне логики – помещать инфраструктурный код в доменную модель
  - Много логики – разработать свой слой преобразования данных

CUSTIS®

SOA и ESB

- Удаленного взаимодействия немного – поместить логику преобразования данных в доменную модель

- Иначе – разработать свой слой преобразования данных

ПОДВОДИМ ИТОГИ

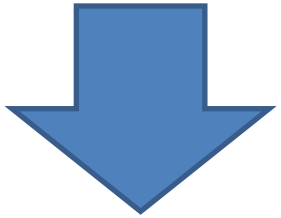
CUSTIS®

Распределенная архитектура – данность

CUSTIS®

Что нам дает DDD

- Эффективный способ борьбы со сложностью
- Единый язык



- Низкая стоимость разработки и сопровождения

CUSTIS®

Но есть проблемы!

CUSTIS®

Так что же делать?

- Оценить:
  - Сложность доменной модели
  - Необходимость сильно распределенной архитектуры
  - Стоимость разработки собственных инструментов

CUSTIS®

Вопросы?

Докладчик: Николай Гребнев

e-mail: ngrebnev@gmail.com

http://www.slideshare.net/ngrebnev/domain-driven-design-6988494