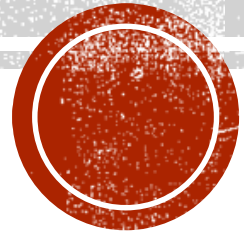
РАЗВОЈ СОФТВЕРА 2

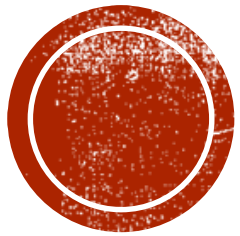
Концепти дизајна управљаног доменом



УВОД

- Да ли се ауто налази на левој или на десној страни?

Не можемо само да куцамо програм…
Треба да имамо визију…



ДИЗАЈН УПРАВЉАН ДОМЕНОМ

ШТА ЈЕ ТО ДИЗАЈН УПРАВЉАН ДОМЕНОМ (DDD)?

DDD је приступ у развоју софтвера
где се у центру налази развој програмског **модела домена**,
који садржи темељно разумевање процеса и правила домена

-Мартин Фаулер-

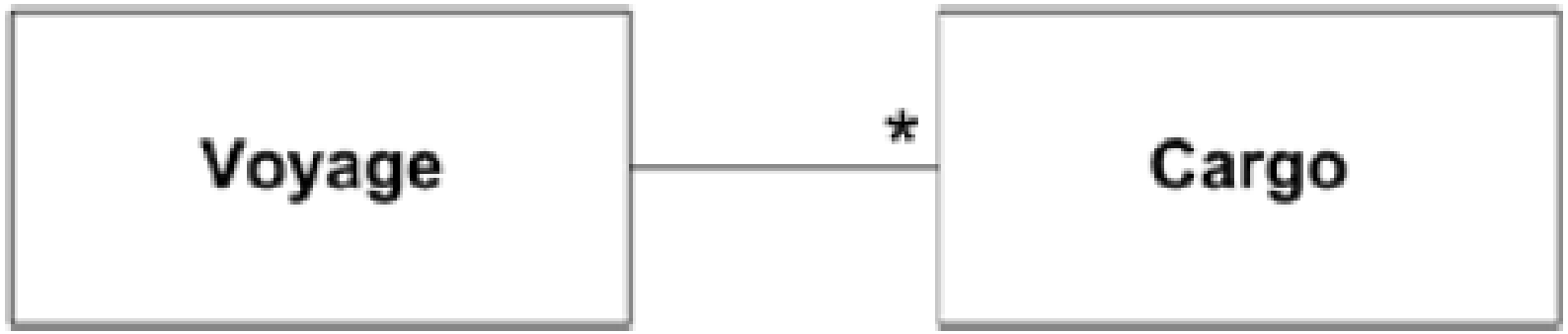
ШТА ЈЕ ТО ДИЗАЈН УПРАВЉАН ДОМЕНОМ (DDD)?

- То је приступ дизајну система који има следеће особине:
  - Модел је у „центру“ DDD
    - Модел и дизајн обликују једно друго
    - Модел је језик за чланове тима
    - Модел је знање
  - Потребно је држати фокус се на најважнијем

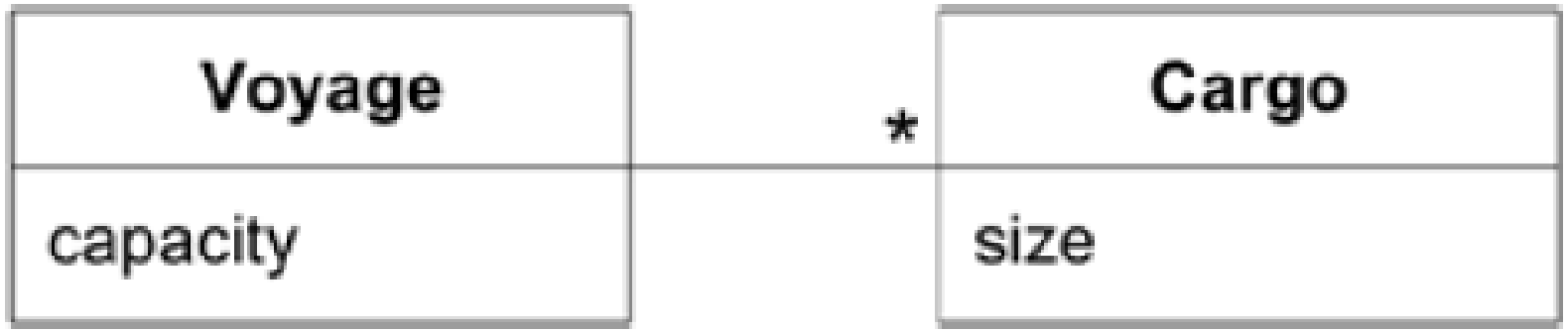
СВЕОБУХВАТНИ ЈЕЗИК

- Потреба за неким заједничким језиком
- Језик и модел
- Како то изгледа?
  - Говор (заједнички)
  - Дијаграми
  - Документи (који нису превише дуги)
  - UML (који није превише детаљан)

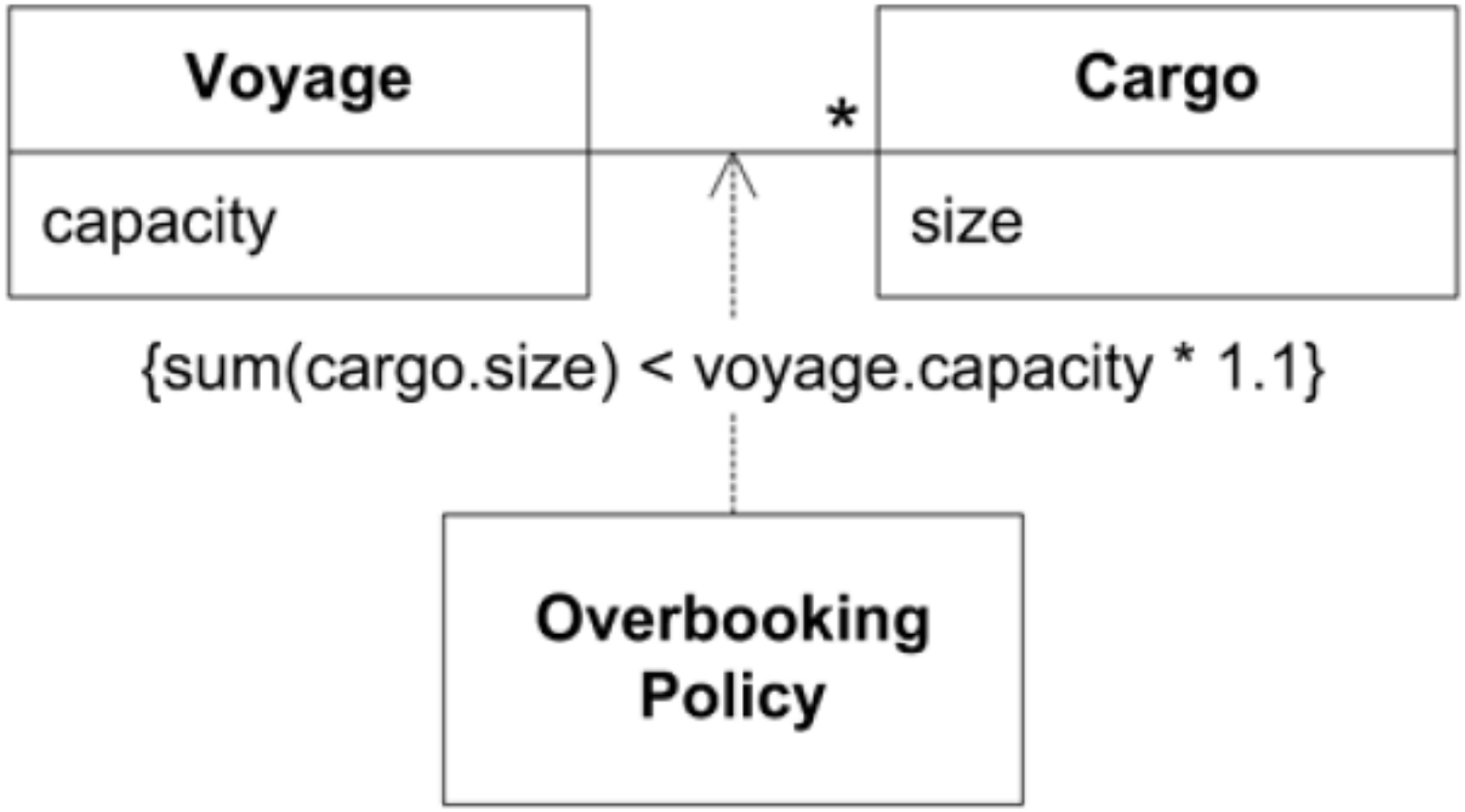
ПРИМЕР- VOYAGE CARGO

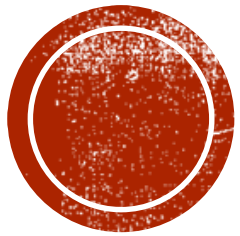


ПРИМЕР - VOYAGE CARGO



ПРИМЕР - VOYAGE CARGO

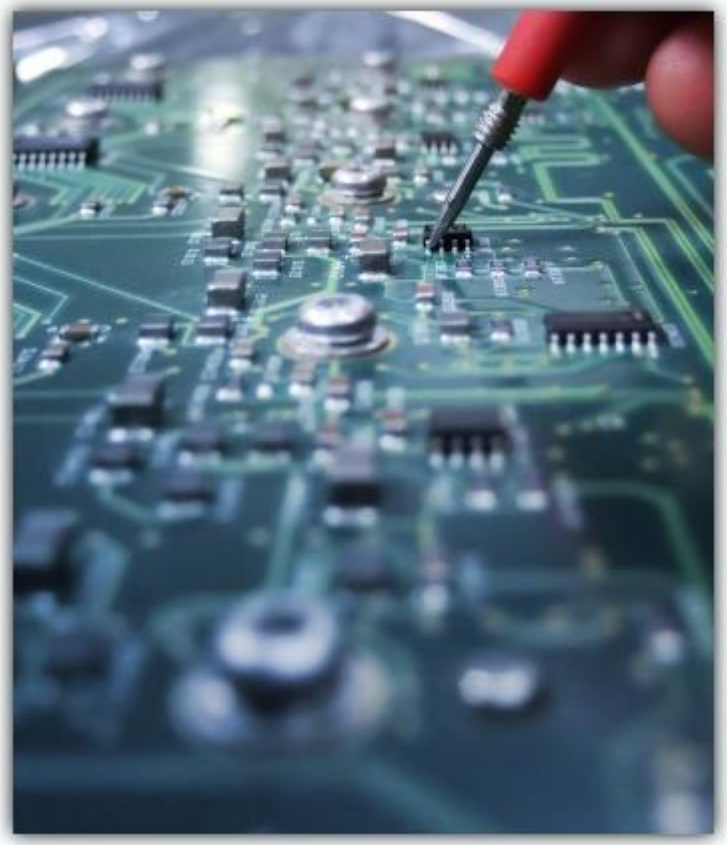




АРХИТЕКТУРА СОФТВЕРСКОГ СИСТЕМА

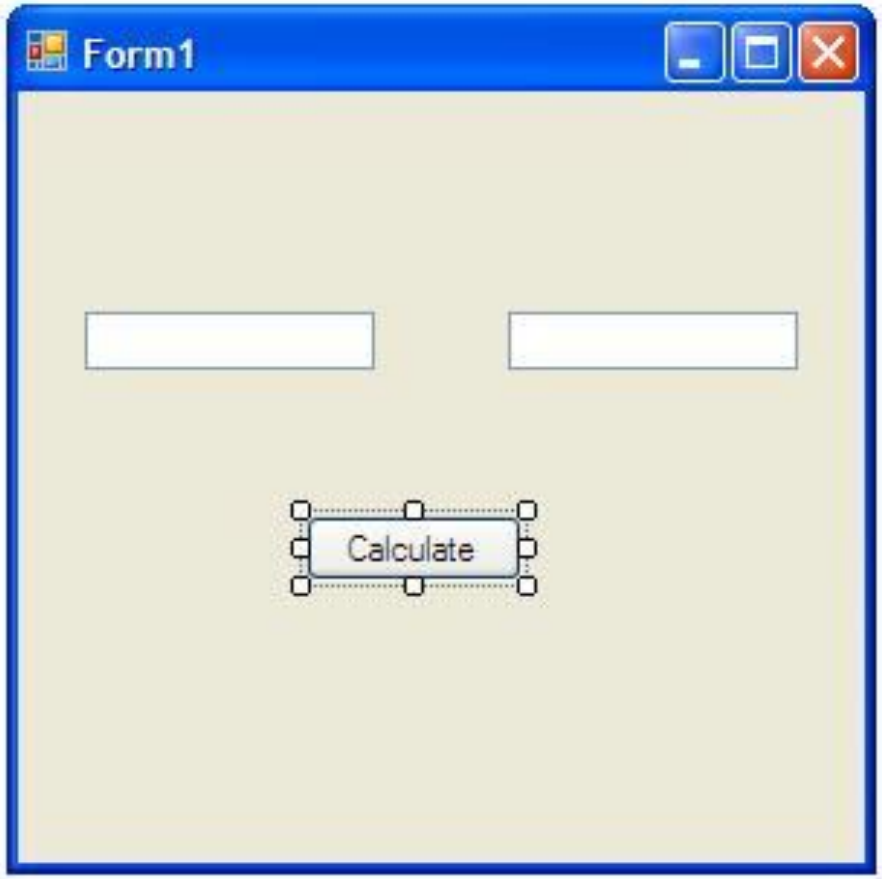
НОВА АРХИТЕКТУРА

- Како изградити нову архитектуру уз помоћ DDD?

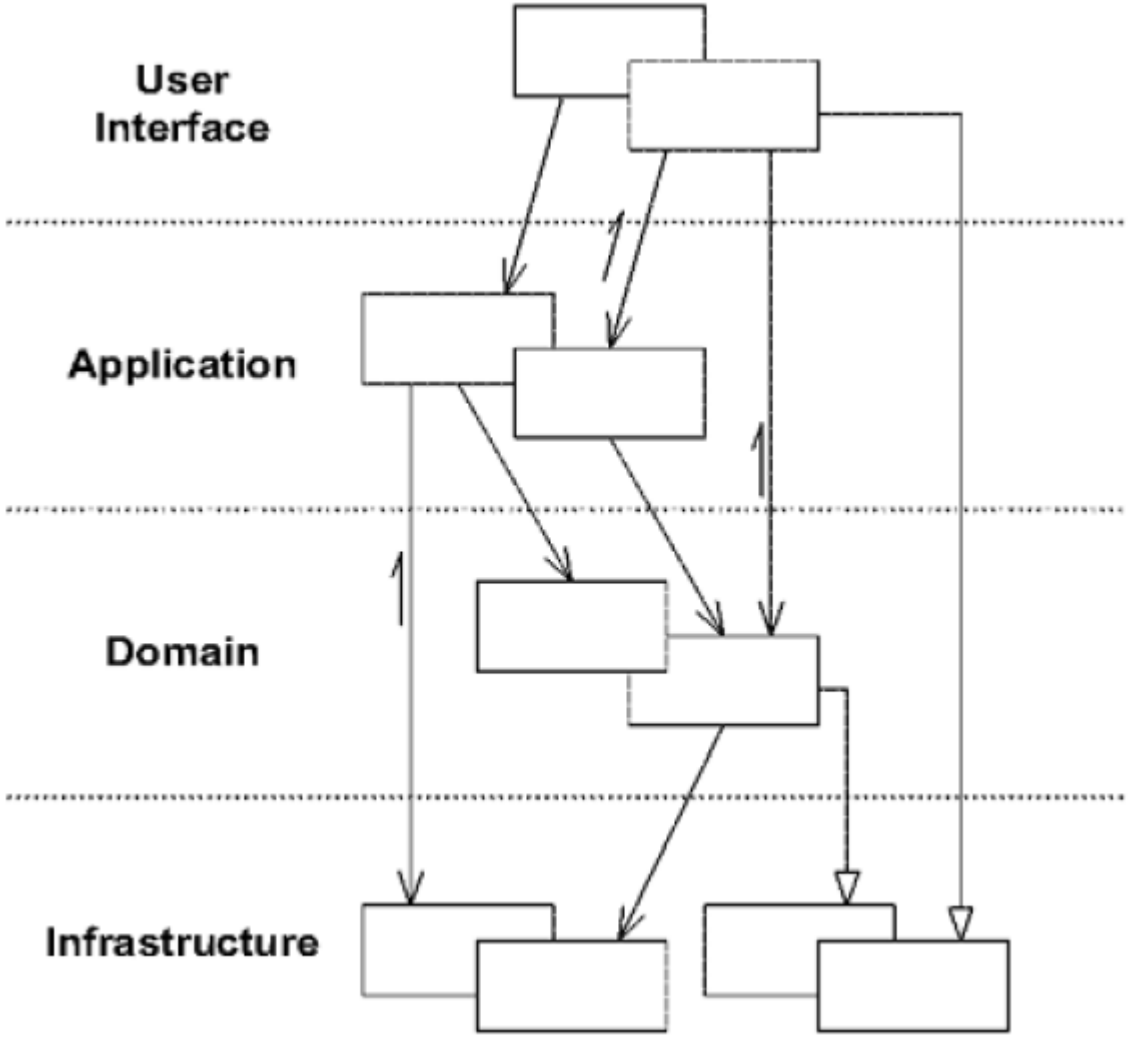


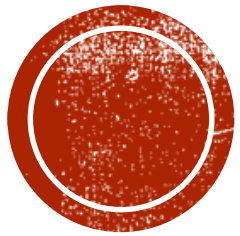
“ANTI-PATTERN” SMART UI

- Предности
- Мане
- Закључак



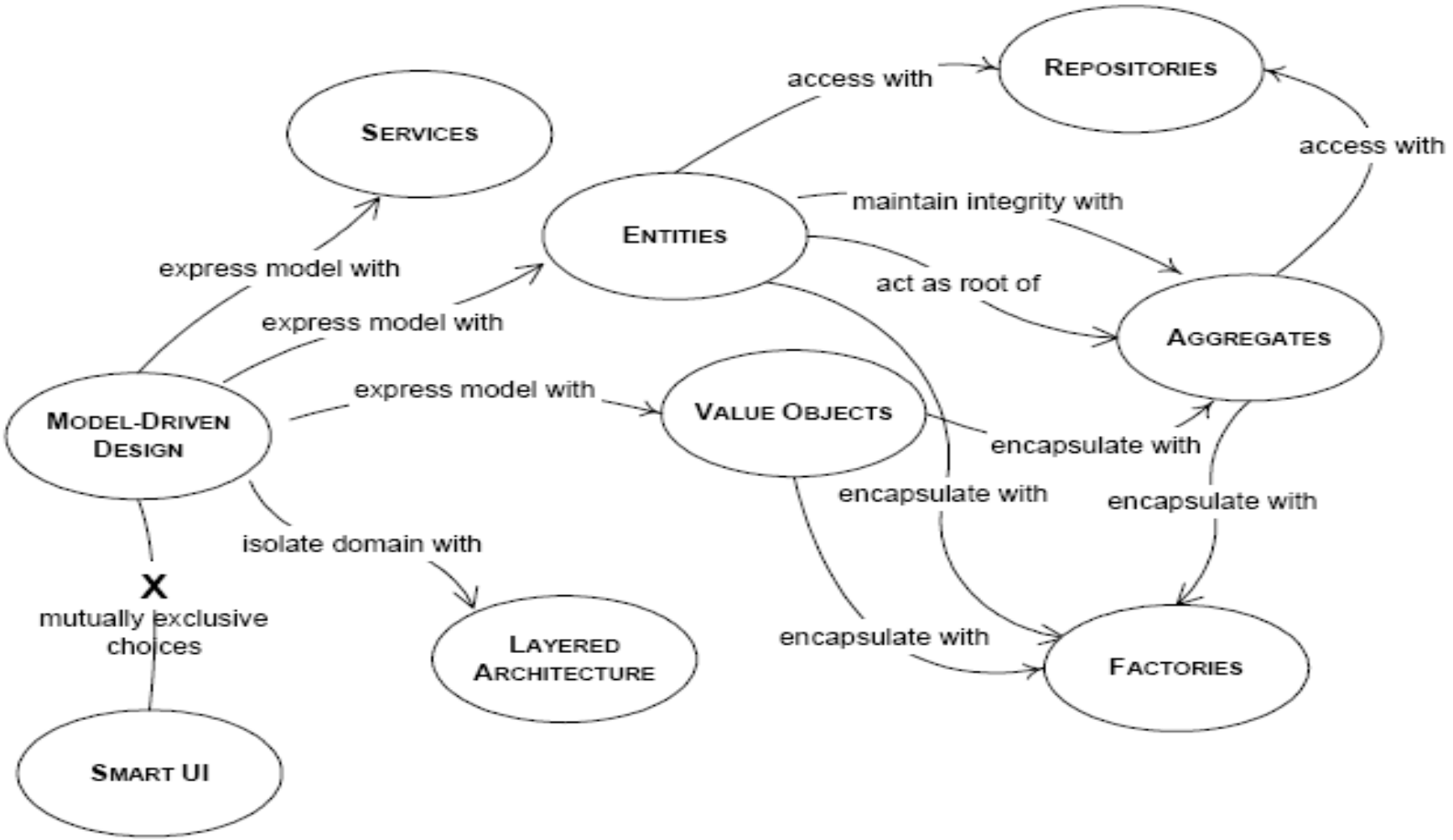
СЛОЈЕВИТА АРХИТЕКТУРА



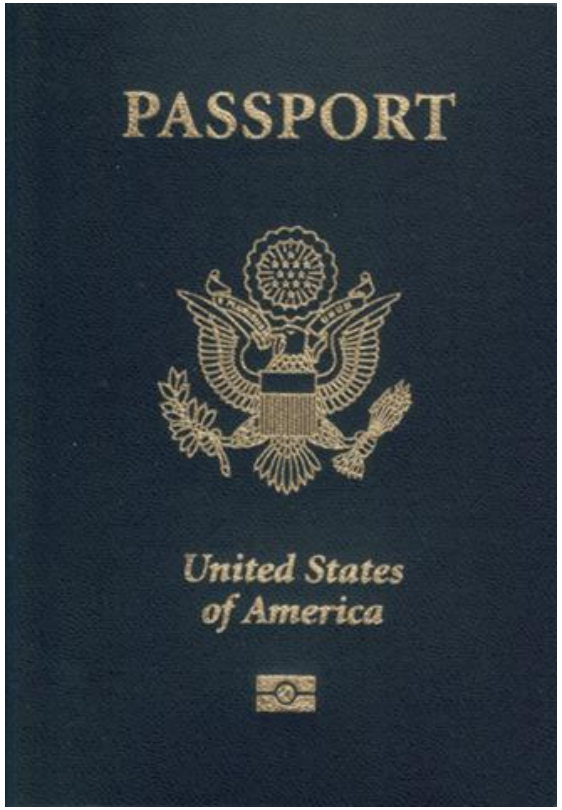


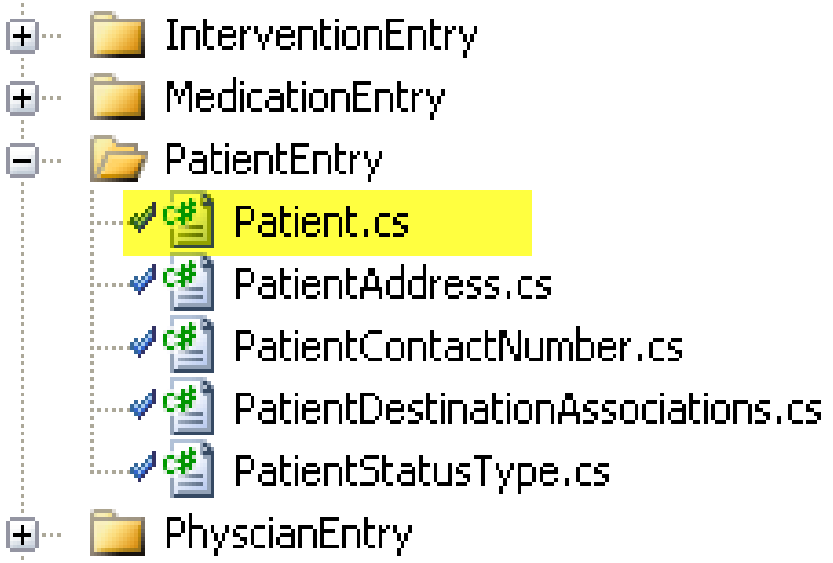
ГРАДИВНИ БЛОКОВИ ЗА ДИЗАЈН УПРАВЉАН ДОМЕНОМ

ГРАДИВНИ БЛОКОВИ ЗА DDD



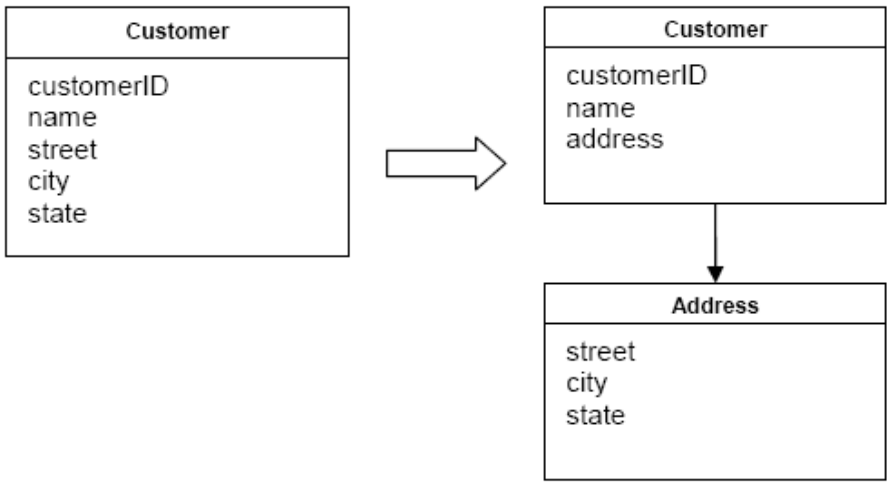
ЕНТИТЕТ

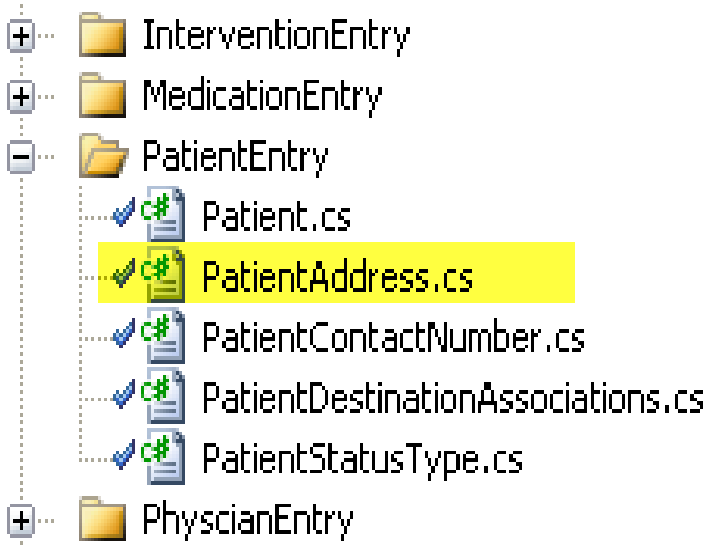

PASSPORT
United States
of America


InterventionEntry
MedicationEntry
PatientEntry
Patient.cs
PatientAddress.cs
PatientContactNumber.cs
PatientDestinationAssociations.cs
PatientStatusType.cs
PhyscianEntry

ВРЕДНОСНИ ОБЈЕКАТ

- Скоро као ентитет, само без идентитета





ВРЕДНОСНИ ОБЈЕКАТ ИЛИ ЕНТИТЕТ? КОГА ТО ЗАНИМА?

- Адреса пацијента може да буде *Вредносни објекат*
  …зато што тај податак не игра важну улогу у третману пацијента током његовог лечења

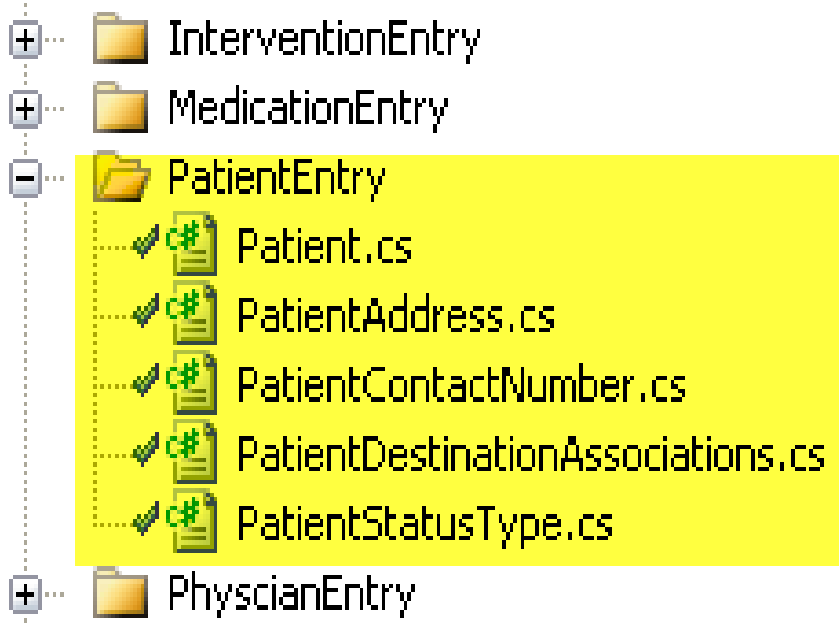
- Поштанска адреса може бити *Ентитет*
  …зато што је од криичног значаја да се зна где се пошиљка треба испоручити

СЕРВИС

Добар *Сервис* има три карактеристике:

- Операција није природна ни за *Ентитет* ни за *Вредносни објекат*
- Интерфејс је јасно дефинисан
- Операција не манипулише стањима (енг. stateless)

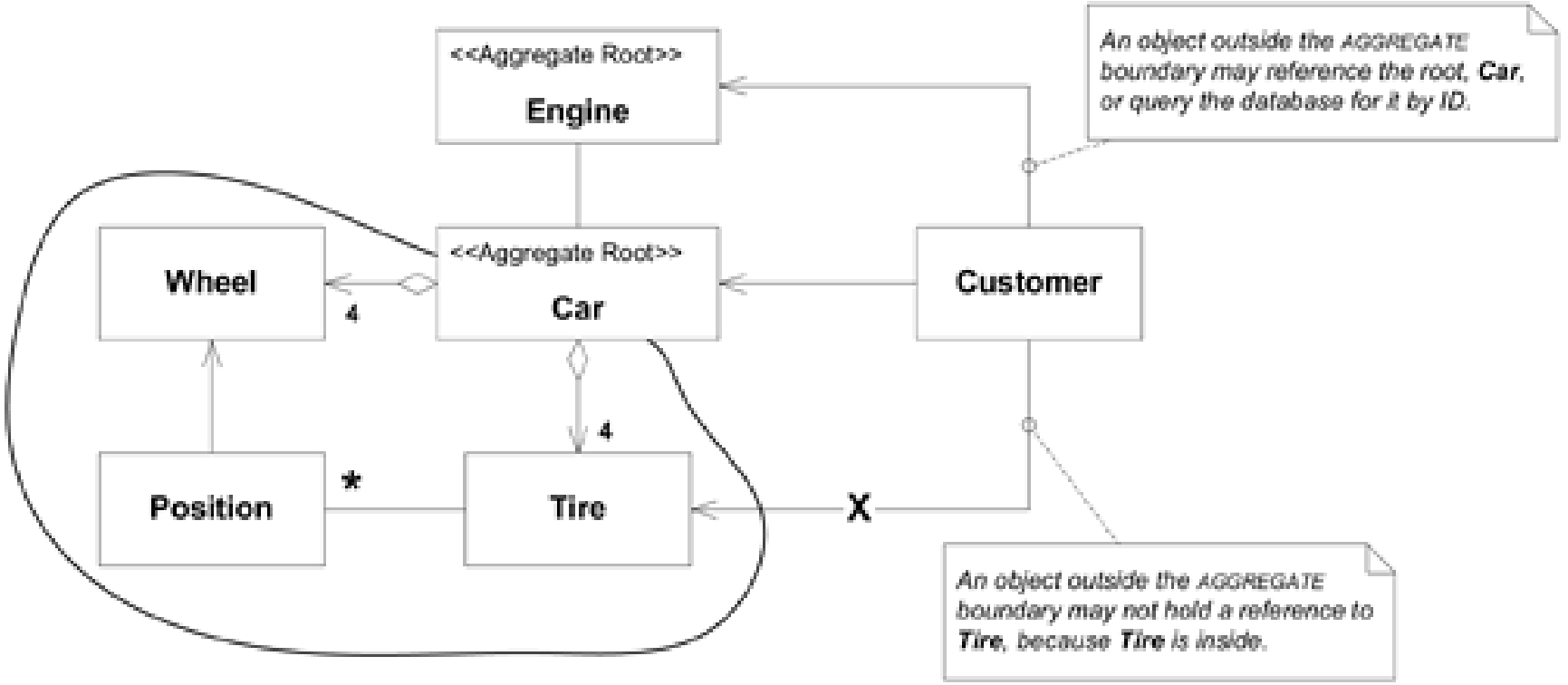
МОДУЛ



Када се неки елементи сместе заједно у *Модул*, тиме је другим програмерима који проучавају дати дизајн поручено да те елементе и надаље треба разматрати заједно

АГРЕГАТ

- `Car` представља агрегат за `Tire`
- Ентитет `Car` је *Корен агрегата*



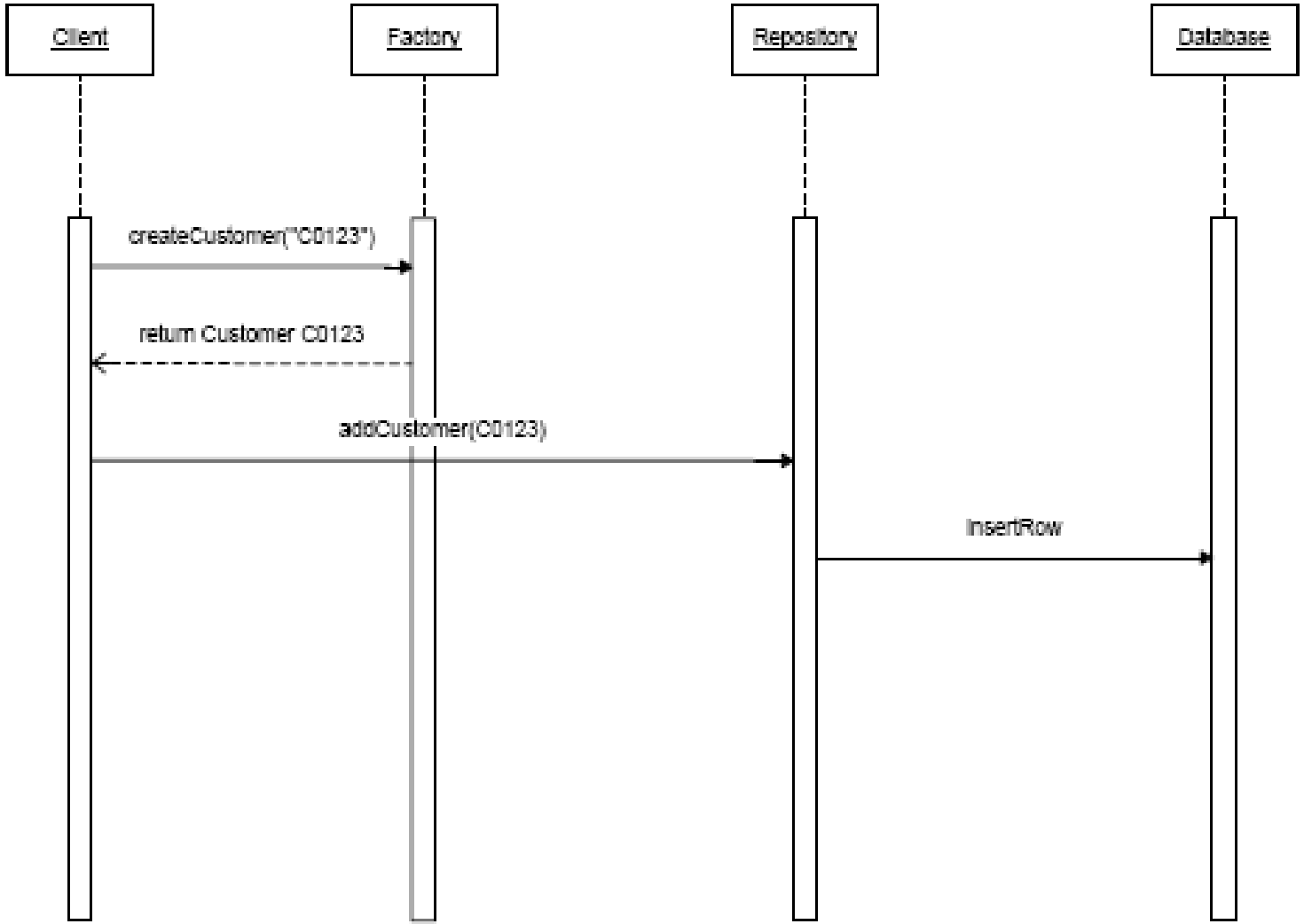
ФАБРИКА

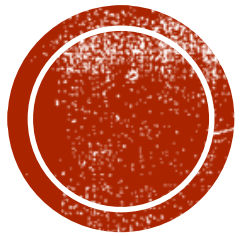
- Одговорности за *Фабрику*
- Захтеви за *Фабрику*
  - Атомичност
  - Жељени тип је апстрактан
- Реконструкција смештених објеката

РЕПОЗИТОРИЈУМ

- Упити над *Репозиторијумом*
- Имплементација *Репозиторијума*
  - Апстрактност типа
  - Предност добијена раскидањем веза са клијентом
  - Контрола трансакције је одговорност клијента

ФАБРИКЕ И РЕПОЗИТОРИЈУМИ





РЕФАКТОРИСАЊЕ РАДИ ДУБЉЕГ УВИДА

РЕФАКТОРИСАЊЕ РАДИ ДУБЉЕГ УВИДА

- Непрекидно рефакторисање

  Рефакторисање ради дубљег увида није исто као техничко рефакторисање, па за то не постоје готови обрасци/шаблони

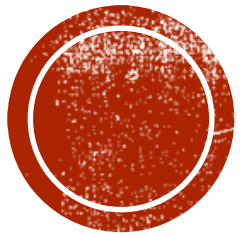
- Извлачње кључних појмова „на светлост дана“

  Постоје моменти када много малих промена дода врло мало на постојећи дизајн, а постоје и моменти када неколико промена направи велику разлику

ИЗВЛАЧЊЕ КЉУЧНИХ ПОЈМОВА „НА СВЕТЛОСТ ДАНА“

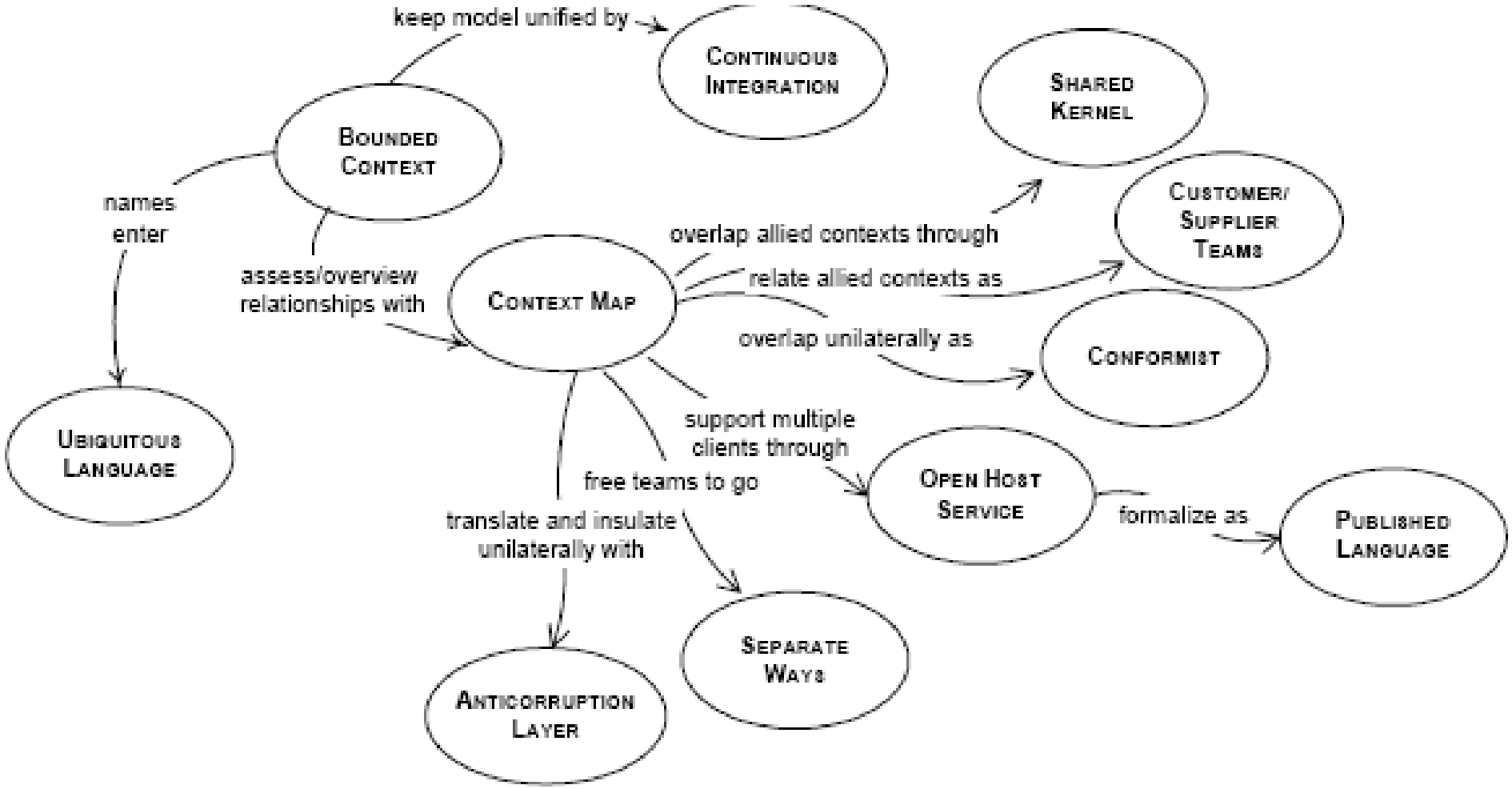
- Ограничење
- Процес
- Спецификација

```
Customer customer =
customerRepository.findCustomer(customerIdentiy);

//…

Specification customerEligibleForRefund = new Specification(

        new CustomerPaidHisDebtsInThePast(),

        new CustomerHasNoOutstandingBalances() );

if (customerEligibleForRefund.isSatisfiedBy(customer))

{

        refundService.issueRefundTo(customer);

}
```



ЗАШТИТА ИНТЕГРИТЕТА МОДЕЛА

ЗАШТИТА ИНТЕГРИТЕТА МОДЕЛА

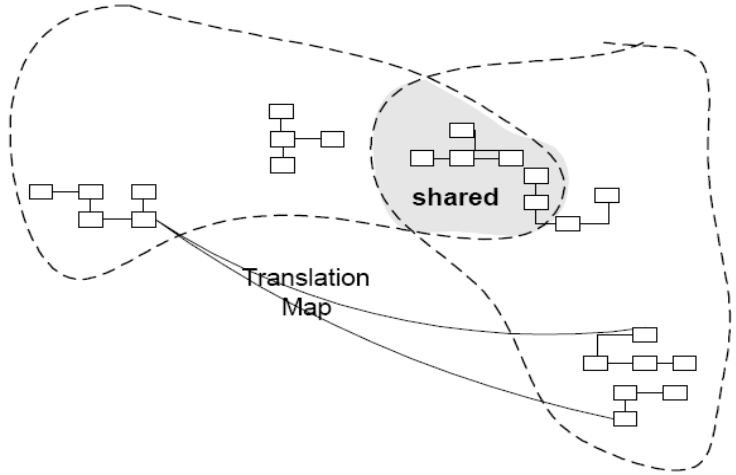


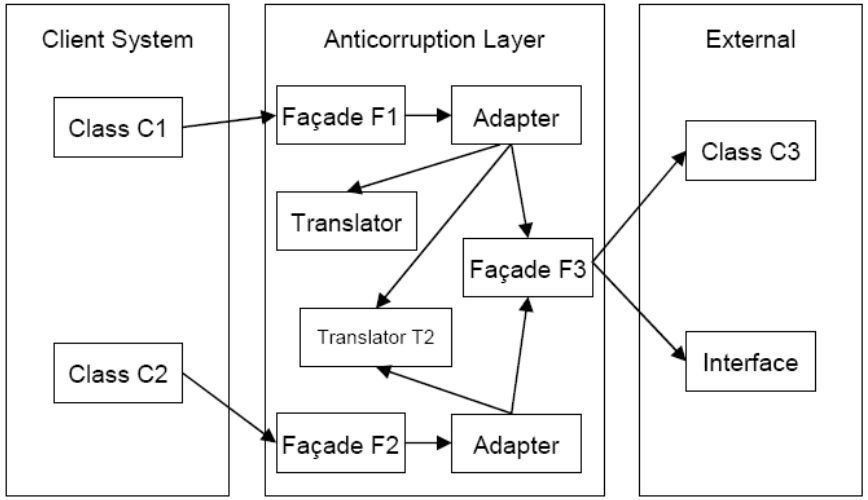
ОГРАНИЧЕНИ КОНТЕКСТ

- Сваки модел има контекст
- Када се ради са старим системом, креира се нови модел и нови контекст
- Модел треба да буде довољно мали како би се могао доделити једном тиму
- Непрекидна интеграција
- Мапа контекста (документ)

ИНТЕРАКЦИЈА ИЗМЕЂУ РАЗЛИЧИТИХ КОНТЕКСТА

- Дељено језгро
- Муштерија-Снадбевач
- Конформиста
- Антикорупцијски слој
- Раздвојени путеви
- Сервис „Отворени домаћин“
- Дестилација





ИНФРАСТРУКТУРА

ПРИПРЕМА ИНФРАСТРУКТУРЕ

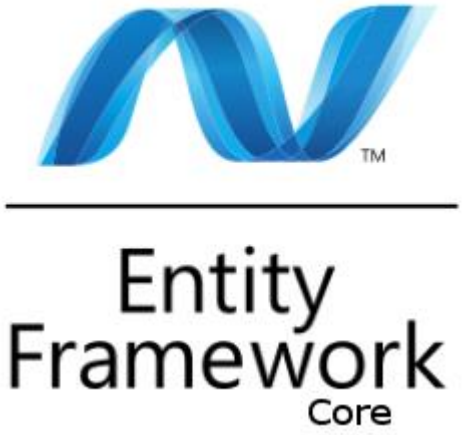
- Игнорисање перзистентности

ИНФРАСТРУКТУРА ORM

- Класификација ORM-ова
- Како ORM решава проблеме
- Захтеви за ORM
- Мапа идентитета
- Јединица рада (енг. Unit Of Work)
- Лењо учитавање (енг. Lazy load)

ENTITY FRAMEWORK, DAPPER, NHIBERNATE

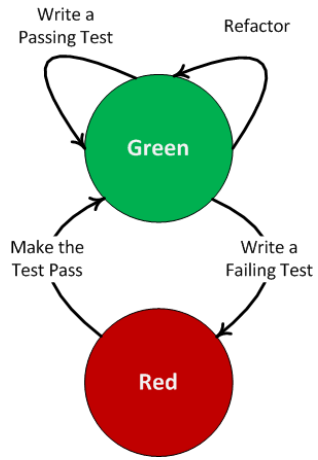
- Those ORM tools provides features required to quickly build an advanced persistence layer in code
- You work only with objects, so you understand Business model



ШАБЛОНИ И ПРАКСЕ КОЈЕ СЕ УКЛАПАЈУ СА DDD

- Агилне технике се добро уклапају са DDD

…таква је нпр. TDD…



- Нови начини пројектовања

…као што су SOA, AOP, IoC…