

Выбор технологий для корпоративного приложения

Константин Кривопустов
krivopustov@haulmont.com

О докладчике

2

- 15 лет в индустрии ИТ
- 10 лет опыта в Java
- технический директор **Haulmont**
 - компания-разработчик корпоративных систем
 - 100+ сотрудников в Самаре и Лондоне
- архитектор технологической платформы



- Введение: обзор задач и архитектуры приложений
- Часть 1: почему мы используем OpenJPA, а не Hibernate
- Часть 2: применение Vaadin для реализации пользовательского интерфейса

- Тиражируемые продукты
- Заказные решения

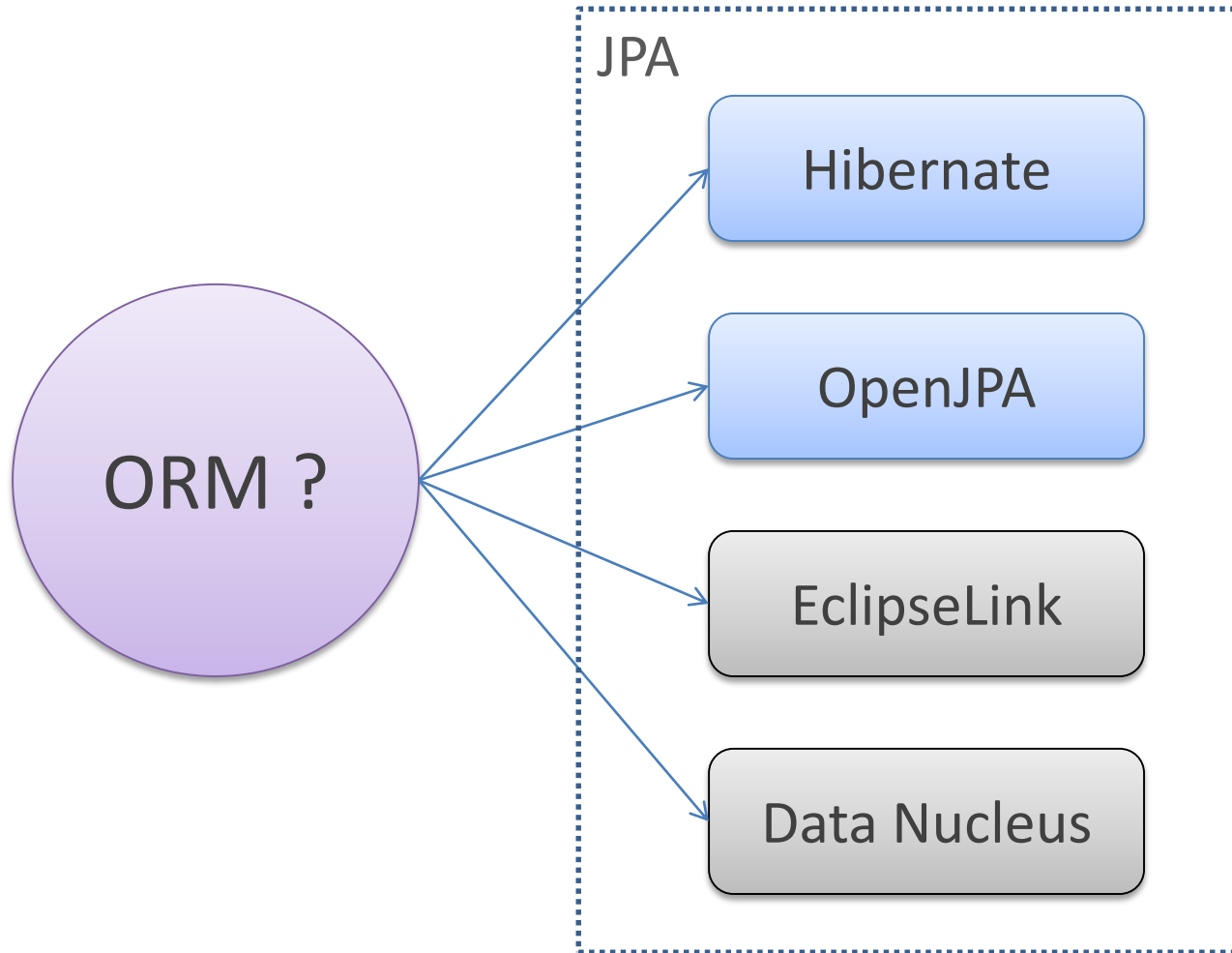
Корпоративные информационные системы

- Сложная модель данных
- Реляционная СУБД, динамические запросы
- Насыщенный UI, типовые компоненты

- Платформа СУВА

Часть 1: почему OpenJPA?

5



Критерии оценки ORM

6

- База данных – самое узкое место в системе
 - Плохо масштабируется
 - Капризна к качеству запросов
- ORM должен оптимально генерировать SQL запросы
 - Минимум запросов
 - Минимум данных возвращаемых или записываемых каждым запросом

OpenJPA: модификация байткода

7

```
public class Area implements Serializable,
    PersistenceCapable, Externalizable {

    private String name;

    public String getName() {
        return pcGetname(this);
    }

    public void setName(String name) {
        pcSetname(this, name);
    }

    private static final String pcGetname(Area area)
    {...}

    private static final void pcSetname(Area area, String s)
    {...}

    protected transient StateManager pcStateManager;
    private transient Object pcDetachedState;
}
```

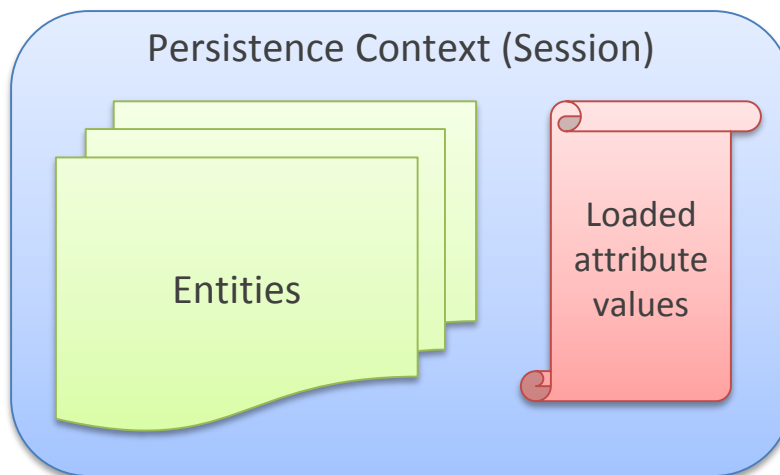
Отслеживание
изменений

Загрузка
по требованию
(lazy loading)

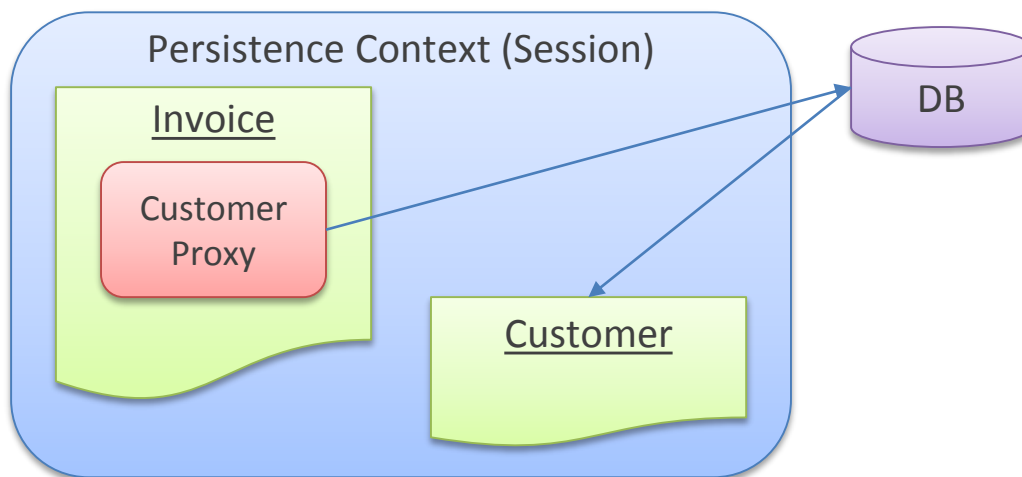
Hibernate: как он это делает?

8

Отслеживание
изменений



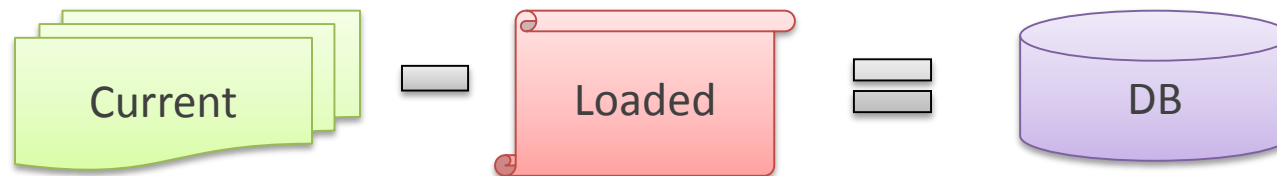
Загрузка
по требованию
(lazy loading)



Hibernate: проблемы

9

- Много объектов в контексте – долгий поиск изменений при коммите



- Отсоединили объект от контекста – потеряли информацию об изменениях



- Использование прокси ведет к невозможности создания Optional Lazy OneToOne ассоциации

Пример сущности

10

```
@Entity(name = "dn$Case")
@Table(name = "DN_CASE")
public class Case extends BaseEntity {
    private static final long serialVersionUID = 254;

    @Column(name = "ONE_STEP_ID")
    protected Integer oneStepId;

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "DEBTOR_ID")
    private Debtor debtor;

    @Column(name = "SECOND_DEBTOR_NAME")
    private String secondDebtorName;

    @OneToOne(fetch = FetchType.LAZY, cascade = Cascade)
    @JoinColumn(name = "WARRANT_ID")
    private Warrant warrant;

    @Column(name = "CASE_STATUS")
    private String status;

    @Column(name = "ONE_STEP_SECONDARY_STATUS_ID")
    private Integer osSecondaryStatusId;
```

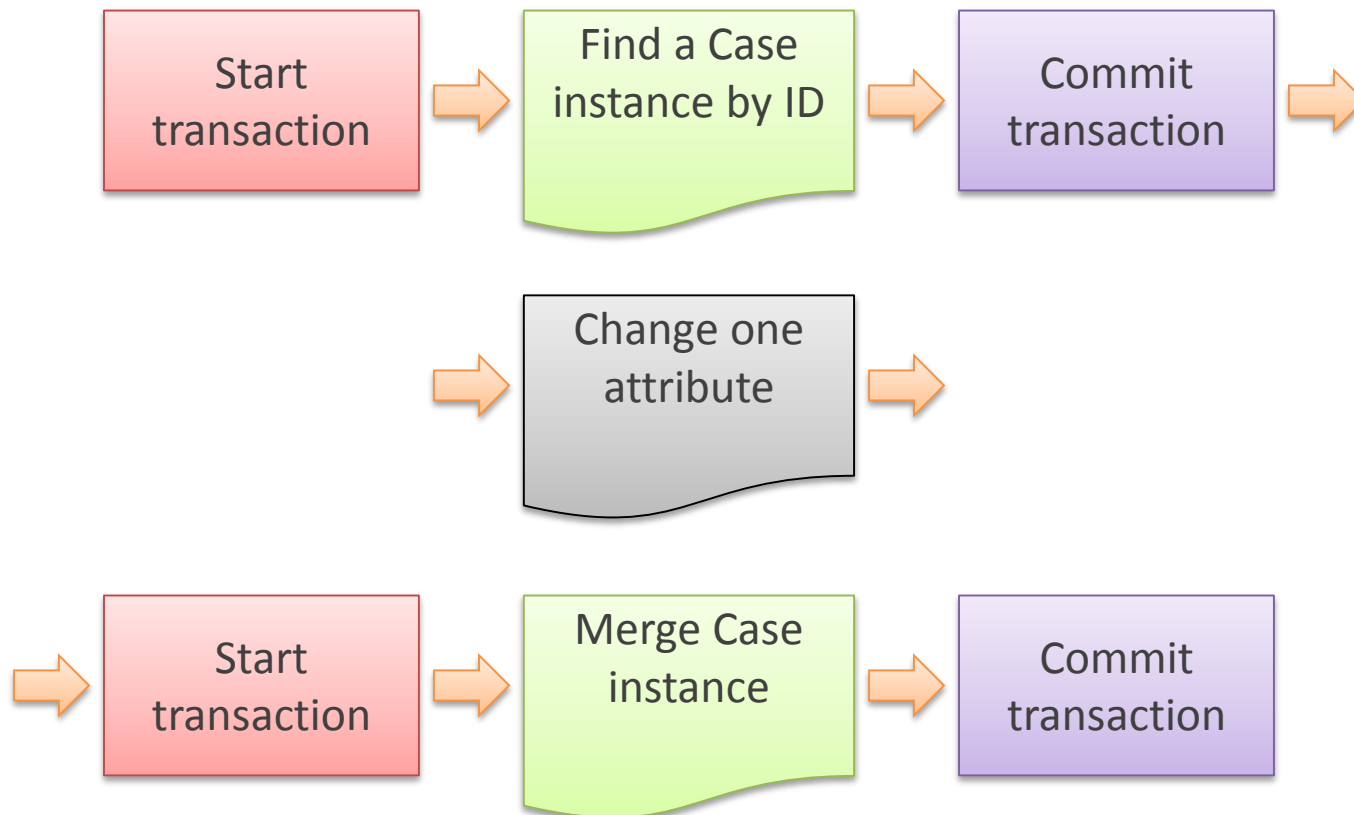
```
create table DN_CASE (
    ID uuid,
    CREATE_TS timestamp,
    CREATED_BY varchar(50),
    VERSION integer,
    UPDATE_TS timestamp,
    UPDATED_BY varchar(50),
    DELETE_TS timestamp,
    DELETED_BY varchar(50),
    ONE_STEP_ID integer,
    DEBTOR_ID uuid,
    WARRANT_ID uuid,
    CASE_STATUS varchar(100),
    CASE_SECONDARY_STATUS_ID uuid,
```

60 полей

1 200 000 записей

Изменение одного атрибута

11



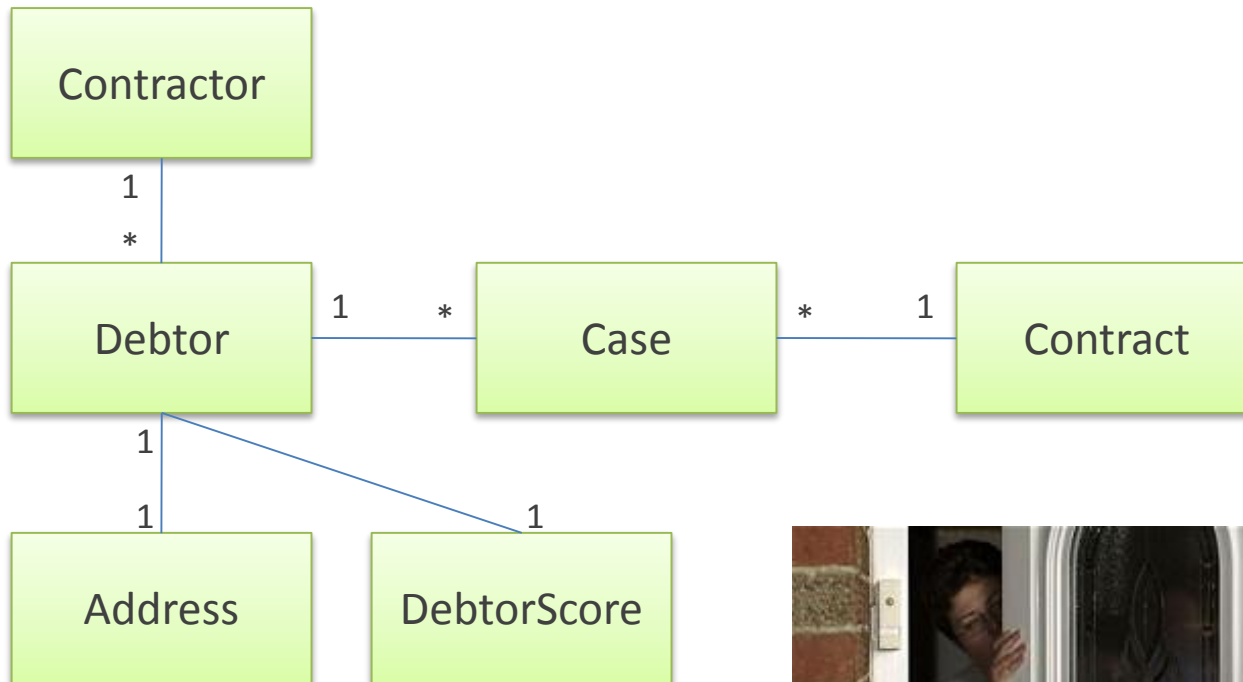
Изменение одного атрибута: результаты

12

		OpenJPA	Hibernate
Find	<pre>select CREATE_TS, -- 63 columns from DN_CASE where ID = ?</pre>	2.1 ms	2.1 ms
Merge	<pre>select LAST_VISIT_DATE, -- OpenJPA: 4 col, Hibernate: 63 col from DN_CASE where ID = ?</pre>	1.1 ms	2.1 ms
	<pre>update DN_CASE set LAST_VISIT_DATE = ? where ID = ?</pre>	1.3 ms	1.3 ms
	Merge JDBC time	2.4 ms	3.4 ms

Тестовая модель данных

13



Экран списка Debtor

14

Search





Contract Show rows

Contract Score above Score below

Score level Client Case status

[Show Reference Panel](#)

1-100 rows of [2969](#)

Reference	Debtor name	Company	Score	Error	Score %	Cases count	Lead case ref	Case age	Cases refs	Work types	Clients
10416277	ADRIAN JOURNAF	<input type="checkbox"/>	 Blocked		-1	1	1580113	583	1580113	Council Tax	
10443160	ADRIAN JOURNAF	<input type="checkbox"/>	 Poor		24.71	2	1671951	436	1671951, 1671	Council Tax	
10463137	ADRIAN JOURNAF	<input type="checkbox"/>									
10404825	ADRIAN JOURNAF	<input type="checkbox"/>	 Poor		22.75	1	1549637	675	1549637	Council Tax	
10239901	ADRIAN JOURNAF	<input type="checkbox"/>									
10423853	ADRIAN JOURNAF	<input type="checkbox"/>	 Fair		34.381	1	1598417	534	1598417	Council Tax	

Запрос для таблицы: OpenJPA

15

JPQL

```
select d from dn$Debtor d
  join d.cases cas
where cas.contract.id = ?1
order by d.contractor.name
```

FetchPlan

```
Debtor.isCompany
Debtor.lastVisitDate
Debtor.contractor
Debtor.cases
Contractor.name
Case.status
Case.contract
...
```

SQL

```
select ID, ... from DN_DEBTOR join ... join ... where ... ;
select ID, ... from DN_CASE where DEBTOR_ID in (?, ?, ...);
```

Запрос для таблицы: Hibernate

16

JPQL

```
select d from dn$Debtor d
  join d.cases cas
  left join fetch d.contractor contr
  left join fetch d.debtorScore ds
  left join fetch d.mainAddress
  left join fetch d.debtorActionsSettings
  left join fetch d.callcentrePrioritySettings
where cas.contract.id = ?1
order by contr.name
```

Hint

```
@OneToMany(
    mappedBy = "debtor",
    fetch = FetchType.LAZY)
@OrderBy("oneStepId")
@BatchSize(size = 200)
private List<Case> cases;
```

SQL

```
select ID, ... from DN_DEBTOR join ... join ... where ... ;
select ID, ... from DN_CASE where DEBTOR_ID in (?, ?, ...);
```


Запрос для таблицы: результаты

17

OpenJPA Hibernate

```
select ID, ...  
-- OpenJPA: 61 columns,  
-- Hibernate: 156 columns  
from DN_DEBTOR join ... join ...  
where ...
```

89 ms

164 ms

```
select ID, ...  
-- OpenJPA: 7 columns,  
-- Hibernate: 65 columns  
from DN_CASE  
where DEBTOR_ID in (?, ?, ...)
```

20 ms

47 ms

JDBC time

109 ms

211 ms

OpenJPA

- **FetchPlan**
 - Полноценные «недозагруженные» сущности со связями
 - Простой единообразный подход к описанию

Hibernate

- **Projections + ResultTransformer**
 - Read-only плоские DTO или maps
- **Lazy properties**
 - Статичное описание в модели, «все или ничего»

Что нам не нравится в OpenJPA

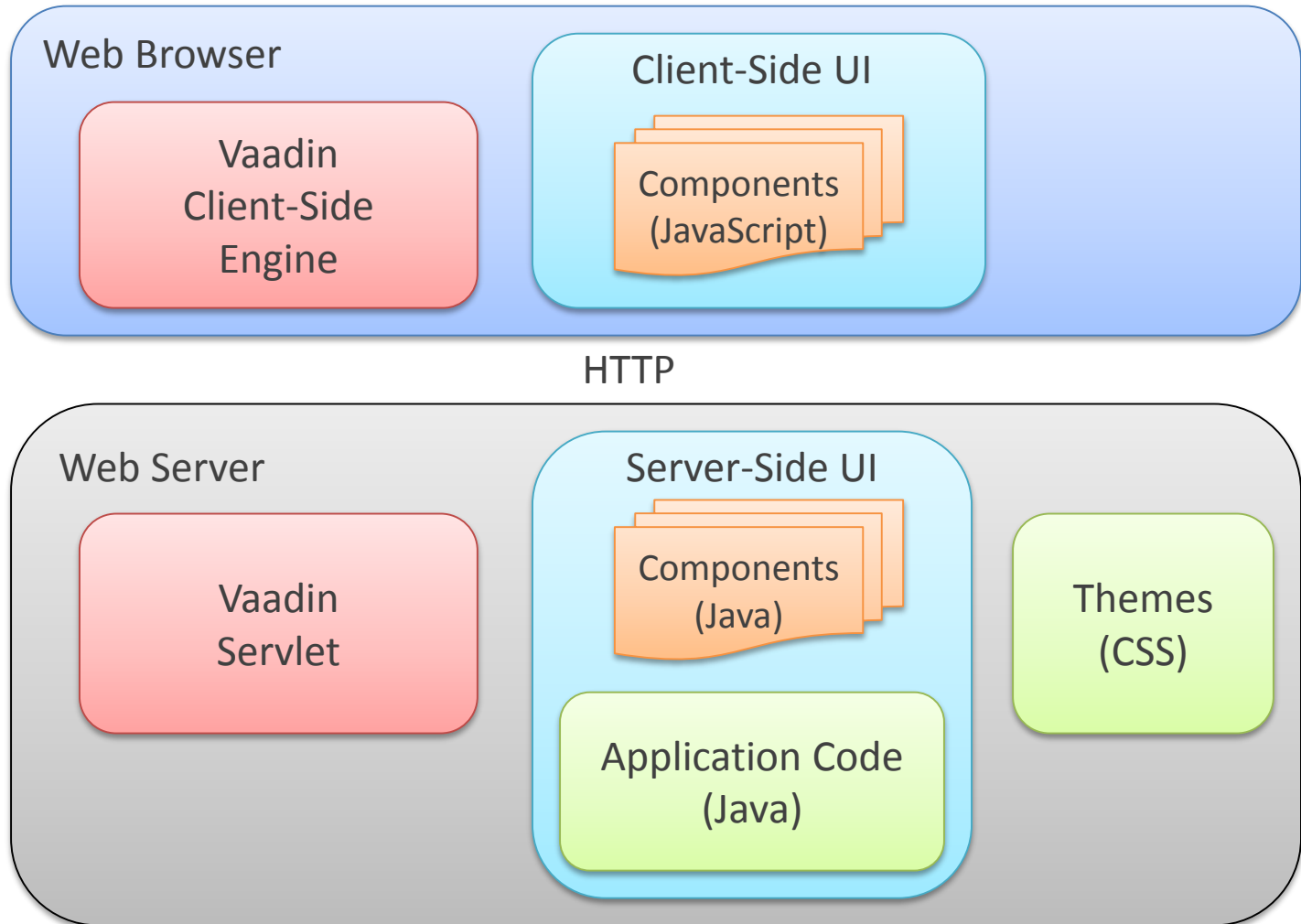
19

- **Ограниченный набор функций**
 - Только определенные стандартом JPQL
 - Иногда нужны функции SQL текущей СУБД
- **Нет возможности выполнить LEFT JOIN ... ON**
 - Иногда нужно задать условия внешнего соединения, не помещая их в WHERE
 - Hibernate: left join ... with
- **В последнее время не развивается**
 - Обратите внимание на EclipseLink и Data Nucleus

Часть 2: применение Vaadin

20

Архитектура Vaadin



- Rich Web UI
 - Функциональность сравнима с Swing
 - Широкий набор компонентов
 - Возможность создания своих компонентов
- Эффективность создания и сопровождения кода
 - Прост в изучении
 - Только Java
 - Весь прикладной код работает на сервере

Минусы Vaadin

22

- Память сервера
 - СЭД «ТЕЗИС»: 10...20 МВ на пользователя
 - 500 пользователей – 10GB heap
 - 1500 пользователей – 2x16GB heap
 - Репликация HTTP-сессий ?
- Подходит для приложений, но не для сайтов
- Создание собственных компонентов на GWT
- Громоздкий код инициализации UI

Пример кода инициализации

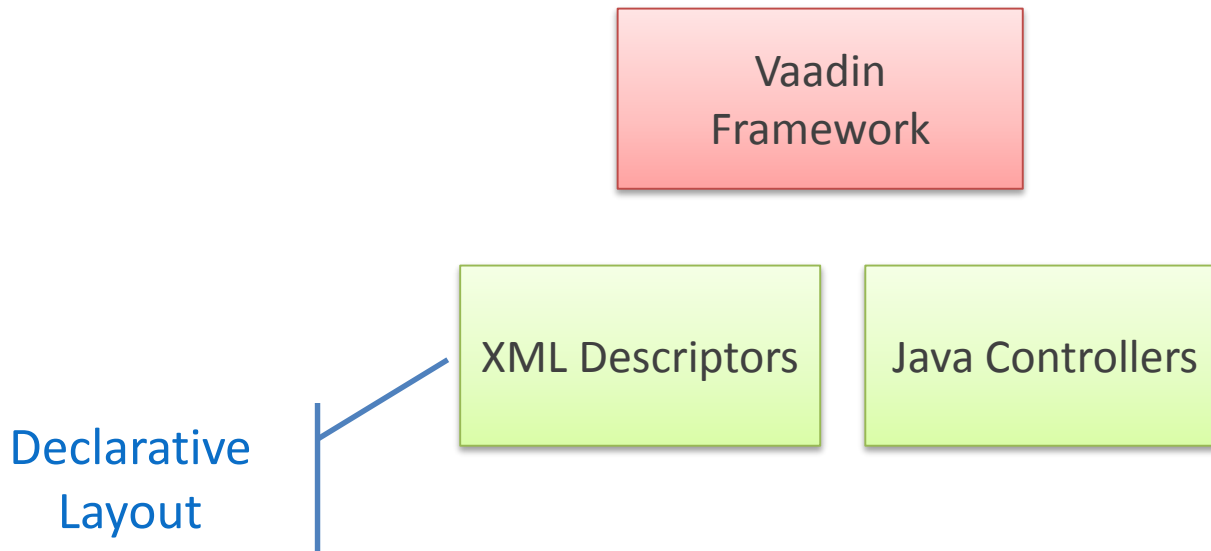
23

```
protected void init(VaadinRequest request) {
    initLayout();
    initContactList();
    initEditor();
    initSearch();
    initAddRemoveButtons();
}

private void initLayout() {
    HorizontalSplitPanel splitPanel = new HorizontalSplitPanel();
    setContent(splitPanel);
    VerticalLayout leftLayout = new VerticalLayout();
    splitPanel.addComponent(leftLayout);
    splitPanel.addComponent(editorLayout);
    leftLayout.addComponent(contactList);
    HorizontalLayout bottomLeftLayout = new HorizontalLayout();
    leftLayout.addComponent(bottomLeftLayout);
    bottomLeftLayout.addComponent(searchField);
    bottomLeftLayout.addComponent(addNewContactButton);
    leftLayout.setSizeFull();
    contactList.setSizeFull();
    bottomLeftLayout.setWidth("100%");
    searchField.setWidth("100%");
    editorLayout.setVisible(false);
}
```

Generic UI

24



Пример компоновки экрана в XML

25

Название *

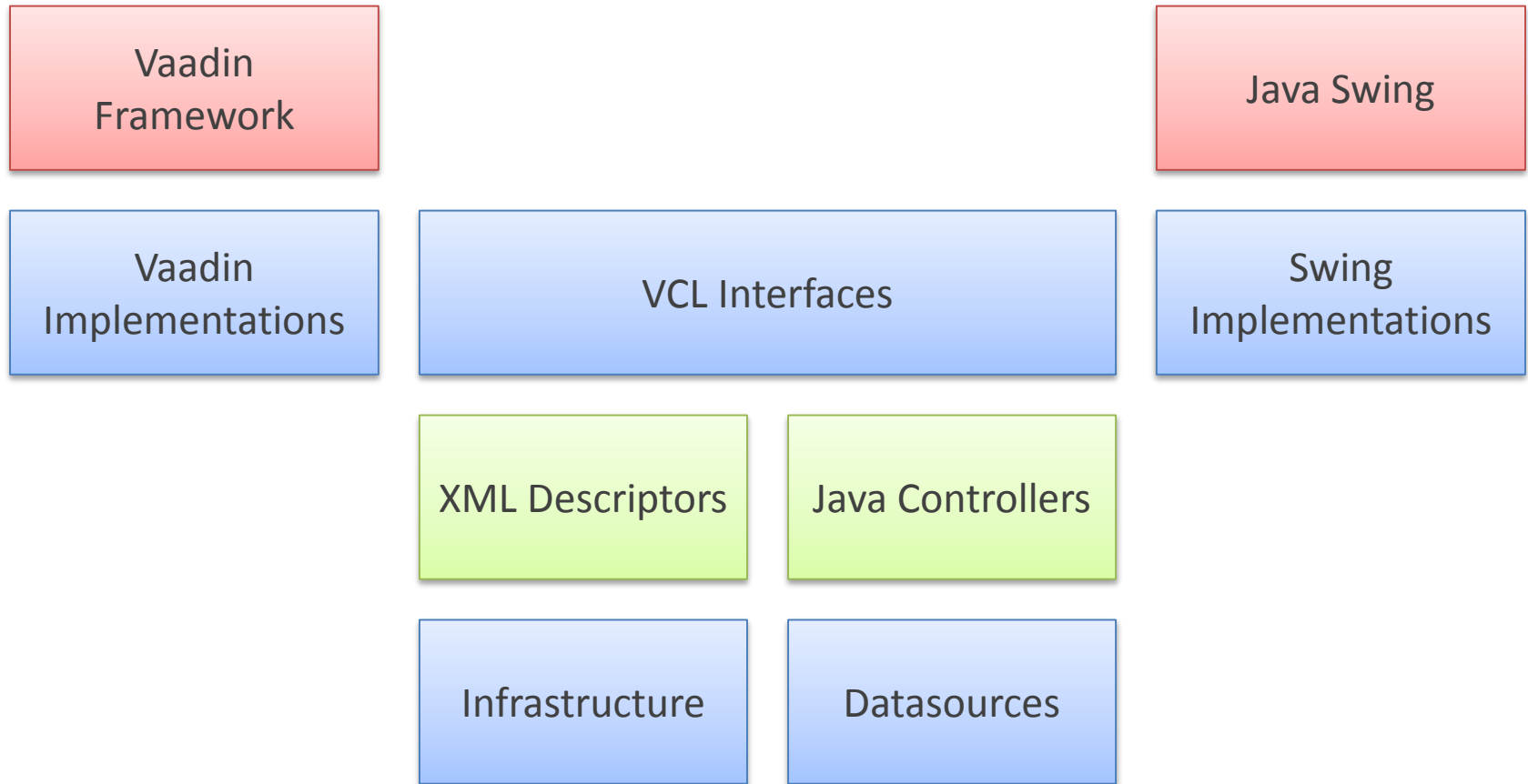
Тип литературы ▼ *

Фамилия	Имя	Отчество
---------	-----	----------

```
<layout>
  <fieldGroup id="fieldGroup"
    datasource="bookDs">
    <field id="name"
      width="250px"/>
    <field id="literatureType"
      optionsDatasource="literatureTypesDs"
      required="true"
      width="250px"/>
  </fieldGroup>
  <groupBox spacing="true"
    width="100%">
    <table id="authorsTable"
      height="200px"
      width="100%">
      <rows datasource="authorsDs"/>
      <columns>
        <column id="author.lastName"/>
        <column id="author.firstName"/>
        <column id="author.patronymic"/>
      </columns>
    </table>
  </groupBox>
</layout>
```

Generic UI

26



Web and Desktop implementation

27

Web Application

Waiting To Allocate (41) | Work In Progress (2) | Prebooks (2) | Issues (0)

Job Amend/Info

Amend | Create Note ▼ | Confirm ETA

Cancel ▼ | Export ▼ | Unlock Job

Search Filter

Spec.Instr [] Journey Type []

Not AA	State	Priority	Countdown ▲	Number	Booking Type	Ser
<input type="checkbox"/>	✈	0	-1d 21h 49m	002765	Prebook	HS
<input type="checkbox"/>		0	-2d 1h 9m	002764	Contract	HS
<input type="checkbox"/>		0	-2d 23h 1m	002794	Asap	HS
<input type="checkbox"/>	✈	0	-3d 1h 0m	002795	Asap	HS
<input type="checkbox"/>	✈	0	-3d 1h 39m	002793	Asap	HS
<input type="checkbox"/>		0	-3d 1h 9m	002732	Contract	HS
<input type="checkbox"/>		0	-3d 21h 27m	002781	Asap	HS
<input type="checkbox"/>		0	-3d 2h 46m	002788	Asap	HS

Waiting To Allocate (41) | Work In Progress (2) | Prebooks (2) | Issues (0)

Job Amend/Info

Amend | Create Note ▼ | Confirm ETA

Cancel ▼ | Export ▼ | Unlock Job

Search Filter

Spec.Instr [] Journey Type []

Not AA	State	Priority	Countdown ▲	Number	Booking Type
<input type="checkbox"/>	✈	0	-1d 21h 50m	002765	Prebook
<input type="checkbox"/>		0	-2d 1h 10m	002764	Contract
<input type="checkbox"/>		0	-2d 23h 2m	002794	Asap
<input type="checkbox"/>		0	-3d 1h 10m	002732	Contract
<input type="checkbox"/>	✈	0	-3d 1h 1m	002795	Asap
<input type="checkbox"/>	✈	0	-3d 1h 40m	002793	Asap
<input type="checkbox"/>		0	-3d 21h 28m	002781	Asap
<input type="checkbox"/>		0	-3d 2h 47m	002788	Asap
<input type="checkbox"/>		0	-3d 3h 13m	002790	Asap
<input type="checkbox"/>		0	-3d 4h 1m	002789	Asap

Desktop Application

Наши проблемы с Vaadin

28

- Быстродействие client-side
 - Определяется производительностью JavaScript
- Изменение внешнего вида, темизация
 - Vaadin 6: CSS, Vaadin 7: SCSS
- Отладка client-side кода
 - Консоль, Пошаговая отладка средствами GWT
- Интеграция с JavaScript компонентами
 - Vaadin 6: GWT-обертки, Vaadin 7: JavaScript API
- Расширение стандартных компонентов
 - Собственный fork библиотеки
- Vaadin 7 при сложной компоновке экрана тормозит в IE8
 - <http://dev.vaadin.com/ticket/12797>
 - Мы в платформе поддерживаем Vaadin и 6 и 7 версии

Ваши вопросы?

29

Спасибо за внимание!