

# Обзор сервера приложений GlassFish

**Преподаватель:**  
**Денис Сергеевич Фокин**

**E-mail: [denis.fokin@gmail.com](mailto:denis.fokin@gmail.com)**

*В презентации использованы слайды:*  
*Alexis Moussine-Pouchkine*  
*[alexis.mp@sun.com](mailto:alexis.mp@sun.com)*

# J2EE 1.4

- J2EE — чрезвычайно мощная платформа
  - Промышленный стандарт для стабильных приложений
- Но эта мощь имеет некоторые недостатки
  - Слишком сложно для новичка
  - Простейшее приложение требует нетривиальной работы
- Возможно ли сохранить мощь...  
но облегчить разработку простейших задач?
- Да... и это одна из целей Java EE 5!

# Java EE 5.0 = (J2EE 1.4).next

- Java EE 5.0 = (J2EE 1.4).next
- Идея Java EE 5: Простота разработки
- POJO
  - Больше свободы, меньше требований
- Широкое использование аннотаций
  - Использование дескрипторов минимально
- DRY принцип
- Вставка ресурсов
- Новые API и фреймворки

# Java EE 5

Изменение спецификаций

- JAX-WS 2.0 & JSR 181
- JAXB 2.0
- EJB 3.0
- Java Persistence (JPA)
- JavaServer Faces 1.2 – new to Platform
- JSP 2.1 – Unification w/ JSF 1.2
- StAX – Pull Parser – **новое для платформы!**

# 5.0 vs 1.4

```
package endpoint;  
import java.rmi.*;
```

```
public class HelloServiceImpl  
    implements HelloServiceSEI {  
  
    public String sayHello(String param)  
        throws java.rmi.RemoteException {  
        return "Hello " + param;  
    }  
}
```

```
package endpoint;  
import java.rmi.*;
```

```
public interface HelloServiceSEI  
    extends java.rmi.Remote {  
    public String sayHello(String param)  
        throws java.rmi.RemoteException;  
}
```

```
<?xml version='1.0' encoding='UTF-8' ?>  
<webservices xmlns='http://java.sun.com/xml/ns/j2ee' version='1.1'>  
  <webservice-description>  
    <webservice-description-name>  
      HelloService</webservice-description-name>  
    <wsdl-file>  
      WEB-INF/wsdl/HelloService.wsdl</wsdl-file>  
    <jaxrpc-mapping-file>  
      WEB-INF/HelloService-mapping.xml  
    </jaxrpc-mapping-file>  
    <port-component xmlns:wsdl-port_ns='urn:HelloService/wsdl'>  
      <port-component-name>HelloService</port-component-name>  
      <wsdl-port>wsdl-port_ns:HelloServiceSEIPort</wsdl-port>  
      <service-endpoint-interface>  
        endpoint.HelloServiceSEI  
      </service-endpoint-interface>  
      <service-impl-bean>  
        <servlet-link>WSServlet_HelloService</servlet-link>  
      </service-impl-bean>  
    </port-component>  
  </webservice-description>  
</webservices>  
<?xml version='1.0' encoding='UTF-8' ?>  
<configuration  
  xmlns='http://java.sun.com/xml/ns/jax-rpc/ri/config'>  
  <service name='HelloService'  
    targetNamespace='urn:HelloService/wsdl'  
    typeNamespace='urn:HelloService/types'  
    packageName='endpoint'>  
    <interface name='endpoint.HelloServiceSEI'  
      servantName='endpoint.HelloServiceImpl'>  
    </interface>  
    </service>  
  </configuration>
```

# Использование JAX-WS 2.x

```
import javax.jws.WebService;  
@WebService  
public class MySimpleWS {  
    public String sayHello(String s) {  
        return "Hello " + s;  
    }  
}
```

# **EJB 3.0**

- Single POJO
- Optional XML descriptor, many defaults
- Easy reference to remote EJB
- Tight integration with JPA
- Compatible with EJB 2.x

# Stateless Session Bean – EJB 2.1

// Extend Session Bean

```
public class CartBean implements javax.ejb.SessionBean
{
    // implement life cycle method
    public void ejbCreate() {...}
    // implement container callback methods
    public void ejbActivate() { ... }
    public void ejbPassivate() {... }
    public void ejbRemove() { ... }
    public void setSessionContext(SessionContext ctx) {
        //...
    }
    // implement business method
    public void addBook(String title) {...}
}
```



# Stateless Session Bean – EJB 3.0

```
import javax.ejb.Stateless;
@Stateless
public class CartBean implements Cart {
    // implement business method
    public void addBook(String title) {...}
}

public interface Cart {
    public void addBook(String title);
}
```

# Client View – EJB Reference

## // OLD – EJB 2.1

```
Context initial = new InitialContext();  
Context myEnv = (Context)initial.lookup("java:comp/env");  
Object objref = myEnv.lookup("ejb/Cart");  
CartHome home=(Cart)  
    PortableRemoteObject.narrow(objref.Cart.class);  
CartBean cart = home.create();
```

## // NEW – EJB 3.0

```
@EJB public Cart cart; // field injection
```

## // setter injection

```
@EJB public void setCart(Cart cart) {...}
```

# Java Persistence API (JPA) 1.0

- JPA — стандарт отображения О на ДБ
  - JSR 220 (EJB 3.0), теперь в отдельном JSR
- Разработан на базе существующих технологий
  - TopLink, Hibernate, Kodo, Java Data Objects (JDO),
- CMP, и т.д.
  - Hibernate, TopLink, Kodo/OpenJPA all implement this API
  - Может использоваться в Java SE, web или основанных на EJB приложениях
- Все реализации с открытым исходным кодом и Pluggable

# Java Persistence API

@Entity

public class Customer { // maps to

@Id

// maps to

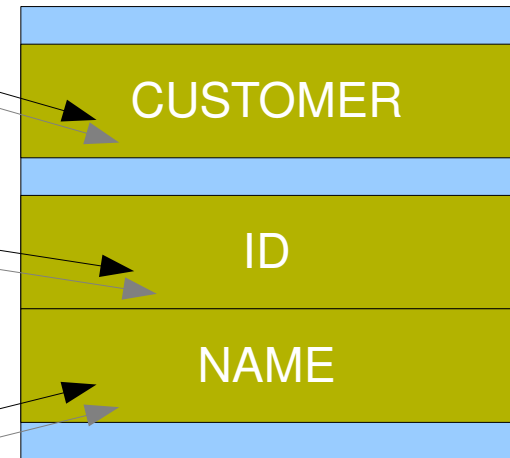
public int getId() { ... }

public void setId(int id) {...}

public String getName() { ... }

public void setName() { ... }

}



# Что же такое GlassFish?

- Java EE 5 Reference Implementation
  - В составе Java EE 5 SDK
- Промышленное качество
  - Sun Java System Application Server 9.x
- Открытый исходный код
  - OSI лицензия - CDDL (такая же как OpenSolaris, NetBeans)
- Сообщество на <http://glassfish.java.net>
  - Исходные коды, bug DB, обсуждения на Java.Net
  - Документация

# Развитие проекта GF

Tomcat  
Jasper  
Catalina  
JSTL  
Struts

Crimson  
XSLTC  
Xalan  
Xerces

JAXB  
JAX-RPC  
JSF

Запуск  
GlassFish



V1 final

V1UR1

V2 final

J1'04  
June 2004

J1'05  
June 2005

J1'06  
May 2006

August 2007



# Версии проекта GlassFish

- GlassFish v1
  - Выпущен!
  - Декабрь 2006 : UR1 — исправление ошибок
- GlassFish v2
  - Новый стек WS, производительность, скорость запуска
  - Балансировка нагрузки, управление кластерами, защита от сбоев
  - Некоторая поддержка скриптовых языков
  - Создание сообщества пользователей и разработчиков
- GlassFish v3
  - Большие архитектурные изменения
  - Модульность, поддержка скриптовых языков

# GlassFish

## Немного о реализации

- WS/XML Stack: JAXB, JAX-WS, StAX
- Web Tier: Grizzly, JSP, Servlets
- Java Persistence: TopLink Essentials
- Rich Clients: AJAX and Java Web Start
- Enterprise Quality: Management, Clustering
- Tools, and more...



# GlassFish v2

- SJS Application Server 9.1
- Beta 1 март 2007 и...
  - Sun Web Developer Pack: jMaki, Phobos, WADL, ...
  - jRubyOnRails: Соединение скорости разработки RoR и мощи Java EE
- Дополнения и улучшения
  - JAX-WS 2.1, Кластеризация, HA, поддержка JBI
  - Лучший User Experience
  - Несколько пользовательских профайлов
  - Центр обновлений

# **AJAX и поддержка скриптинга**

- jMaki - <http://ajax.dev.java.net>
  - Encapsulates very easily AJAX widgets
- DynaFaces - <http://jsf-extensions.dev.java.net>
  - AJAX and JSF
- WoodStock - <http://woodstock.dev.java.net>
  - Библиотека JSF компонентов с AJAX
- JSF Templating  
<https://jsftemplating.dev.java.net/>
  - Шаблоны для страниц и компонентов
- Phobos - <http://phobos.dev.java.net>
  - Скриптовый язык на сервере

# Реализация JAX-WS и JAXB

- Динамическая среда выполнения
- Лучшая реализация
  - JAX-WS быстрее чем Axis
- Разделение транспорта и представления
  - HTTP, JMS, SMTP, TCP/IP
  - Fast Infoset (binary), Textual, Others
- Project Tango (WSIT)
- Взаимодействие с Microsoft
- Превосходная производительность

# GF v2: Grizzly и JSP контейнеры

- JSP контейнер
  - Can use JSR-199 (javac APIs in Mustang)
  - 10x performance improvement
- Grizzly улучшен по сравнению с GlassFish v1
  - Очень гибкий и настраиваемый
  - Производительный и расширяемый
  - Поддерживает Comet
- Используется в Jetty, etc...

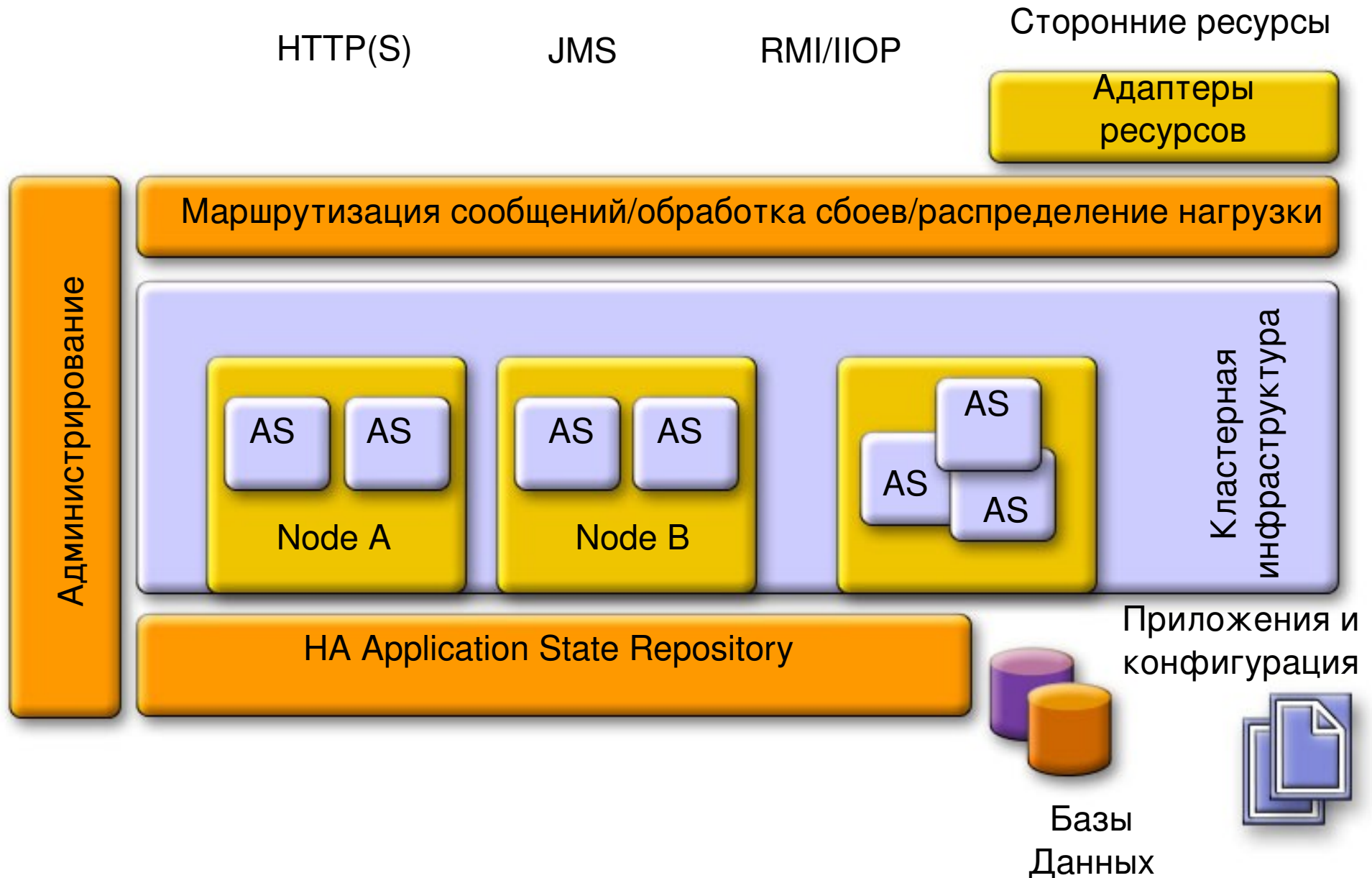
# Top Link Essentials / JPA

- Вклад Oracle
- Очень активное сообщество
  - Oracle, Sun, TmaxSoft, independents
  - Mail: [persistence@glassfish.dev.java.net](mailto:persistence@glassfish.dev.java.net)
- Подключаемо
  - В GlassFish, JEUS, JOnAS, Tomcat, Geronimo, JBoss, Oracle, ...
- Верно и обратное
  - Hibernate & Kodo / OpenJPA run on GlassFish

# Динамическая кластеризация и репликация в памяти

- Project Shoal
  - <http://shoal.dev.java.net>
  - Dynamic clusters
  - Uses JXTA by default
- Репликация в памяти (In-Memory Replication)
- Использование HADB позволяет достигнуть 99.999% доступности

# Кластеры



# Особенности управления

- Централизованное
- Доступно через GUI, CLI, IDEs, Java-программы
- Off-line конфигурация
- Инфраструктура отвечающая требованиям
  - Java Management Extensions, JMX™ API
- Конфигурируемые ANT задачи
- API для управления и мониторинга
  - Application Server Management eXtensions, AMX
  - Можно следит с помощью jConsole и других средств



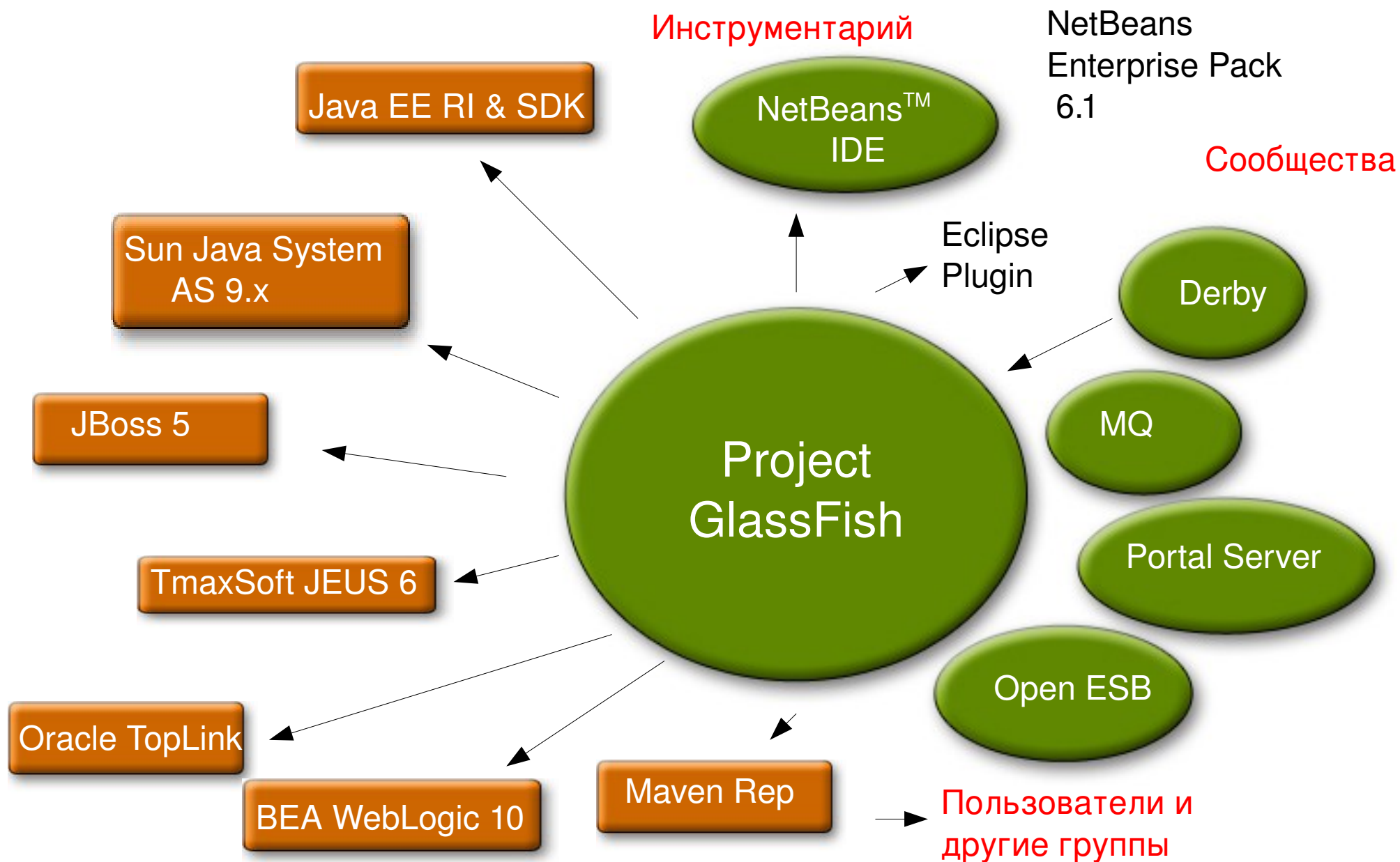
# **JBI – Java Business Integration**

- Реализация Open-ESB 2.0
  - Включено в GlassFish v2
- BPEL engine
- Связывание компонентов
  - HTTP, File, FTP, JMS, TCP, ...
- Поддержка инструментария
  - NetBeans Enterprise Pack 5.5.x
- <http://www.glassfishwiki.org/jbiwiki/Wiki.jsp?page=Jbicomps>

# Производительность

- Скорость запуска
  - Быстрый старт(в GF v3 лучше)
- JSP, JSF, Grizzly
- Web Services Stack benchmark
- Оптимизация Corba
- EJB контейнер

# (Некоторые) Реализации и Вклады



# Фреймворки и приложения



# Переход на GlassFish

- <http://blogs.sun.com/stories>
- PeerFlix
  - GlassFish v1 on Solaris 10 / x2100
  - MySQL, Kodo JDO
- Wotif.com
  - Large # hits, ehCache / Horizontal Scaling
  - Spring and Hibernate
- Harvard University
  - On-line archive for sharing data within and across universities
  - Lucene, PostgreSQL, Apache Shale
- More (small and large)
  - Help us collect these
- > [stories@sun.com](mailto:stories@sun.com)



# GlassFish v3

- Небольшой размер, быстрый, модульный
  - Текущая версия (запуск ~1.5 секунды)
- Ядро <100K
- Контейнер для Web 2.0
  - Web приложения Java, Phobos, jRuby/RoR, ...
- Скорее не Java EE 5 контейнер
  - ... а и Java EE контейнер
- Видеопримеры и исходный код

[http://blogs.sun.com/dochez/entry/first\\_glassfish\\_v3\\_screencast1](http://blogs.sun.com/dochez/entry/first_glassfish_v3_screencast1)

# ИСТОЧНИКИ

- *JSR 220: Enterprise JavaBeans™, Version 3.0.* EJB Core Contracts and Requirements
- *JSR 220: Enterprise JavaBeans™, Version 3.0.* Java Persistence API
- *PRO EJB 3 Java Persistence API*
- [\*http://ru.sun.com/java/j2ee/\*](http://ru.sun.com/java/j2ee/)

# **Вопросы и ответы**