

Java Platform Enterprise Edition (Java EE)

1. **Java Community Process (JCP)**: организация, созданная Sun Microsystems в 1998 г.; состоит из Java-разработчиков и владельцев патентов; занимается разработкой стандартов (JSR) технологий Java; два исполнительных комитета: SE/EE и ME.



2. **Java Specification Request (JSR)**: запрос на спецификацию Java; формальный документ описывающий спецификацию, которую предлагается добавить к Java-платформе;
3. **Стадии JSR**:

JSRs

- » JSRs by Platform
- » JSRs by Technology
- » JSRs by Stage
- » JSRs by Committee
- » List of All JSRs

My JCP

Use of JCP site is subject to the JCP Terms of Use and the Oracle Privacy Policy

JCP Info

- » About JCP
- » Get Involved
- » Community Resources
- » Community News
- » FAQ
- » Contact Us

JSRs: Java Specification Requests

List by JCP Stage

JSR Review (0 JSRs)
A specification is initiated by community members and approved for development by the Executive Committee.

Early Draft Review (4 JSRs)
Once a JSR is approved, a group of experts is formed to develop a first draft of the specification that anyone with an Internet connection can read and comment on the draft.

Public Review (2 JSRs)
The JSR draft goes out for review by the public where anyone with an Internet connection can read and comment on the draft.

Proposed Final Draft (3 JSRs)
The version of the draft specification that will be used as the basis for the RI and TCK.

Final Release (233 JSRs)
The leader of the Expert Group then sees that the reference implementation and its associated Technology Compatibility Kit are updated in response to ongoing issues.

Maintenance Review (3 JSRs)
The completed specification, reference implementation, and Technology Compatibility Kit are updated in response to ongoing issues.

Active (23 JSRs)
JSRs that have posted a JCP milestone draft within the last 12 months.

Inactive (0 JSRs)
JSRs that have both not posted a Final Release and have not posted a JCP milestone draft for the last 12 months.

Dormant (25 JSRs)
JSRs are listed as dormant when there is no Specification/Maintenance Lead, when a JSR has had an extended period of inactivity.

Withdrawn (98 JSRs)
JSRs that are withdrawn from the JCP by their Spec Leads before going to Final Release.

Rejected (14 JSRs)
JSRs that are voted down by the Executive Committee during the JSR Review, Draft Approval, or Final Approval Ballots.

4. Пример JSR 315, 369

JSR

Community

Expert Group

[Summary](#) | [Proposal](#) | [Detail \(Summary & Proposal\)](#)

JSRs: Java Specification Requests

JSR 315: Java™ Servlet 3.0 Specification

Stage	Access	Start	Finish
Maintenance Release	Download page	09 Mar, 2011	
Maintenance Draft Review	Download page	07 Jan, 2011	06 Feb, 2011
Final Release	Download page	10 Dec, 2009	
Final Approval Ballot	View results	17 Nov, 2009	30 Nov, 2009
Proposed Final Draft	Download page	07 May, 2009	
Public Review Ballot	View results	06 Jan, 2009	12 Jan, 2009
Public Review	Download page	12 Dec, 2008	12 Jan, 2009
Early Draft Review	Download page	05 May, 2008	04 Jun, 2008
Expert Group Formation		03 Jul, 2007	
JSR Review Ballot	View results	19 Jun, 2007	02 Jul, 2007

JSR

Community

Expert Group

[Summary](#) | [Proposal](#) | [Detail \(Summary & Proposal\)](#) | [Nominations](#)

JSRs: Java Specification Requests

JSR 369: Java™ Servlet 4.0 Specification

Stage	Access	Start	Finish
JSR Renewal Ballot	View results	17 Jan, 2017	30 Jan, 2017
Early Draft Review	Download page	15 Oct, 2015	14 Dec, 2015
Expert Group Formation		23 Sep, 2014	07 May, 2015
JSR Review Ballot	View results	09 Sep, 2014	22 Sep, 2014
JSR Review		26 Aug, 2014	08 Sep, 2014

Status: Active
JCP version in use: 2.10
Java Specification Participation Agreement version in use: 2.0

JSR-000369 Java™ Servlet 4.0 Early Draft Review

You must accept the [Software License Agreement](#) to download this software.

☒ Accept License Agreement | ☐ Decline License Agreement

Download Instruction: Click the product name or the file name to start the download.

Required Files

File Description and Name	Size
JSR-000369 Java Servlet 4.0 Early Draft Review Specification ✚ servlet-4_0_EDR.pdf	1,566 KB
JSR-000369 Java Servlet 4.0 Specification Early Draft Review API ✚ javax.servlet-api-4.0.0-b01.jar	99 KB
JSR-000369 Java Servlet 4.0 Specification Early Draft Review Javadocs ✚ javax.servlet-api-4.0.0-b01-javadoc.jar	646 KB
JSR-000369 Java Servlet 4.0 Specification Early Draft Review Sources ✚ javax.servlet-api-4.0.0-b01-sources.jar	232 KB

OVERVIEW PACKAGE CLASS USE TREE DEPRECATED INDEX HELP

PREV NEXT FRAMES NO FRAMES

Java Servlet 4.0 API Specification

Servlet API Documentation

Package	Description
javax.servlet	The javax.servlet package contains a number of classes and interfaces that describe and define the contracts between a servlet class and the runtime environment provided for an instance of such a class by a conforming servlet container.

Other Packages

Package	Description
javax.servlet.annotation	The javax.servlet.annotation package contains a number of annotations that allow users to use annotations to declare servlets, filters, listeners and specify the metadata for the declared component.
javax.servlet.descriptor	Provides programmatic access to a web application's configuration information that was aggregated from the web.xml and web-fragment.xml descriptors.
javax.servlet.http	The javax.servlet.http package contains a number of classes and interfaces that describe and define the contracts between a servlet class running under the HTTP protocol and the runtime environment provided for an instance of such a class by a conforming servlet container.

5. JSR: все на 22.03.2017

JSRs: Java Specification Requests

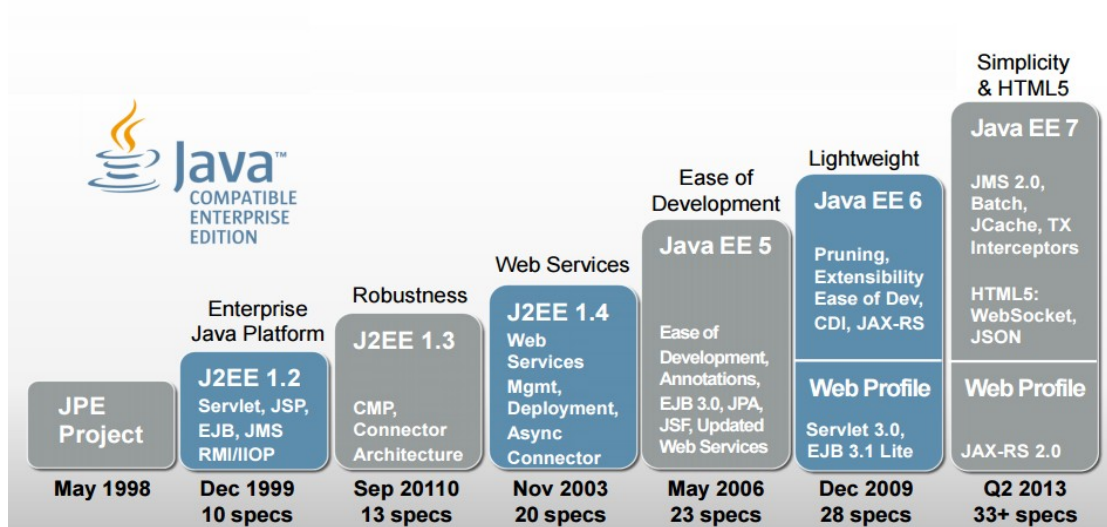
List of all JSRs

By Number By Date

1	Real-time Specification for Java Description: The Real-Time Specification for Java extends the Java™ platform to support both current practice and advanced real-time systems application programming. Status: Final Latest Stage: Final Release 3 Download page Start: 2006-07-12 Spec Lead: James J. Hunt, aicas GmbH
2	Boundary-Scan API Description: This API targeted the JavaCard™ platform so as to allow it to be usable on all available Java™ virtual machines from the very smallest to the most sophisticated. Status: Withdrawn Reason: Withdrawn at the request of the submitter. Latest Stage: Withdrawn Effective: 1999-01-09 Spec Lead: Neil Jacobson, Xilinx Spec Lead: Harold Ogle, Oracle

926	Java 3D™
Description:	Maintenance of the Java 3D specification.
Status:	Maintenance
Latest Stage:	Maintenance Release 2 Download page Start: 2007-01-18
Spec Lead:	Kevin Rushforth, Sun Microsystems, Inc.
<hr/>	
927	Java™ TV API 1.1
Description:	The maintenance of the Java TV specification.
Status:	Maintenance
Latest Stage:	Maintenance Release 2 Download page Start: 2008-10-20
Spec Lead:	Michael Lagally, Oracle
Spec Lead:	Jens Paetzold, Oracle
<hr/>	

6. Java EE 7



Java EE 7 – текущая имплементация



download.java.net/glassfish/4.0/promoted/

7. JRS Java EE (53 JSRs) :

JSRs by Platform

Certain JSRs are part of one or more of the platforms: collections of standards that comprise the three Java editions: Standard, Enterprise and Micro. Many JSRs are not part of the platform, but are not listed on this page. This page lists the JSRs that comprise the various versions of the three platform editions.

Java EE (53 JSRs) | Java SE (60 JSRs) | Java ME (85 JSRs)

Java EE

The Java Enterprise Edition offers APIs and tools for developing multitier enterprise applications. The JSRs in this list are part of the Java EE platform as component or umbrella JSRs.

5 XML Parsing Specification

Description: The Java™ API for XML Parsing (JAXP) allows developers to easily use XML Parsers in their applications via the industry standard SAX and DOM APIs.

Status: **Final**

Latest Stage: [Final Release](#) [Download page](#) [Start: 2000-03-21](#)

Spec Lead: [Rajiv Mordani, Oracle](#)

16 J2EE™ Connector Architecture

Description: The J2EE™ Connector architecture defines a standard architecture for integrating Java™ applications with existing back-end Enterprise Information systems.

Status: **Final**

Latest Stage: [Final Release](#) [Download page](#) [Start: 2001-09-24](#)

Spec Lead: [Binod PG, Sun Microsystems, Inc.](#)

19 Enterprise JavaBeans™ 2.0

Description: The Enterprise JavaBeans™ 2.0 specification extends the architecture with integration with JMS, improved support for entity bean persistence, a portable query language for interoperability.

Status: **Maintenance**

Latest Stage: [Maintenance Draft Review](#) [Download page](#) [Start: 2002-02-20](#) [End: 2002-03-25](#)

Spec Lead: [Linda Demichiel, Oracle](#)

52 A Standard Tag Library for JavaServer Pages™

Description: A standard tag library for JSP™ pages.

Status: **Maintenance**

Latest Stage: [Maintenance Draft Review 3](#) [Download page](#) [Start: 2006-10-04](#) [End: 2006-11-06](#)

Spec Lead: [Jan Luehe, Sun Microsystems, Inc.](#)

8. Java EE версии:

Версия	Полное имя	Дата публикации
1.0	Java 2 Platform Enterprise Edition, v 1.0	декабрь 1999
1.2	Java 2 Platform Enterprise Edition, v 1.2	2000
1.2.1	Java 2 Platform Enterprise Edition, v 1.2.1	23 мая 2000
1.3	Java 2 Platform Enterprise Edition, v 1.3	24 сентября 2001
1.4	Java 2 Platform Enterprise Edition, v 1.4	24 ноября 2003
5.0	Java Platform, Enterprise Edition, v 5	11 мая 2006
6.0	Java Platform, Enterprise Edition, v 6	6 декабря 2009
7.0	Java Platform, Enterprise Edition, v 7	12 июня 2013
8.0	Java Platform, Enterprise Edition, v 8	~ 2017 ^[1]
9.0	Java Platform, Enterprise Edition, v 9	~ 2018

9. Java EE спецификация:



Java™ Platform, Enterprise Edition (Java EE) Specification, v7

Please post comments to: users@javaee-spec.java.net

10. Java EE состав технологий:

Java Technology	App Client	Web	EJB	Status
EJB 3.2	Y ^a	Y	Y	REQ, OPT ^b
Servlet 3.1	N	Y	N	REQ
JSP 2.3	N	Y	N	REQ
EL 3.0	N	Y	N	REQ
JMS 2.0	Y	Y	Y	REQ
JTA 1.2	N	Y	Y	REQ
JavaMail 1.5	Y	Y	Y	REQ
Connector 1.7	N	Y	Y	REQ
Web Services 1.4	Y	Y	Y	REQ
JAX-RPC 1.1	Y	Y	Y	OPT
JAX-WS 2.2	Y	Y	Y	REQ
JAX-RS 2.0	N	Y	N	REQ
WebSocket 1.0	N	Y	N	REQ
JSON-P 1.0	Y	Y	Y	REQ

Java Technology	App Client	Web	EJB	Status
Concurrency Utilities for Java EE 1.0	N	Y	Y	REQ
Batch 1.0	N	Y	Y	REQ
JAXB 2.2	Y	Y	Y	REQ
JAXR 1.0	Y	Y	Y	OPT
Java EE Management 1.1	Y	Y	Y	REQ
Java EE Deployment 1.2 ^c	N	N	N	OPT
JACC 1.5	N	Y	Y	REQ
JASPIC 1.1	N	Y	Y	REQ

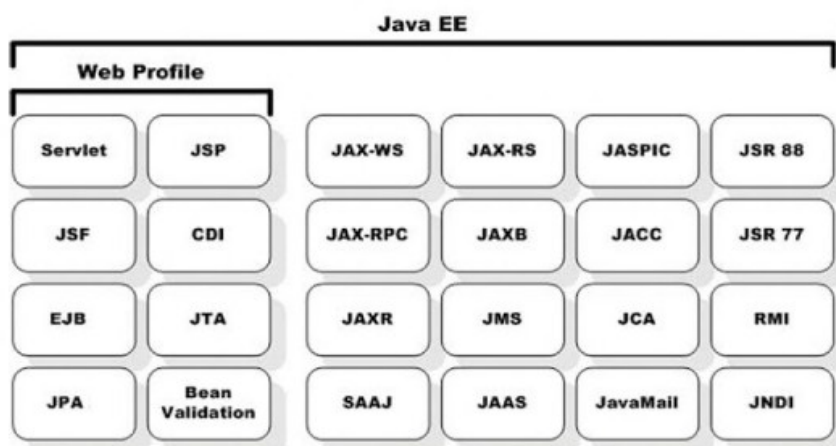
Java Technology	App Client	Web	EJB	Status
JSP Debugging 1.0	N	Y	N	REQ
JSTL 1.2	N	Y	N	REQ
Web Services Metadata 2.1	Y	Y	Y	REQ
JSF 2.2	N	Y	N	REQ
Common Annotations 1.2	Y	Y	Y	REQ
Java Persistence 2.1	Y	Y	Y	REQ
Bean Validation 1.1	Y	Y	Y	REQ
Managed Beans 1.0	Y	Y	Y	REQ
Interceptors 1.2	Y	Y	Y	REQ
Contexts and Dependency Injection for Java EE 1.1	Y	Y	Y	REQ
Dependency Injection for Java 1.0	Y	Y	Y	REQ

a. Client APIs only.

b. EJB entity beans and associated EJB QL only.

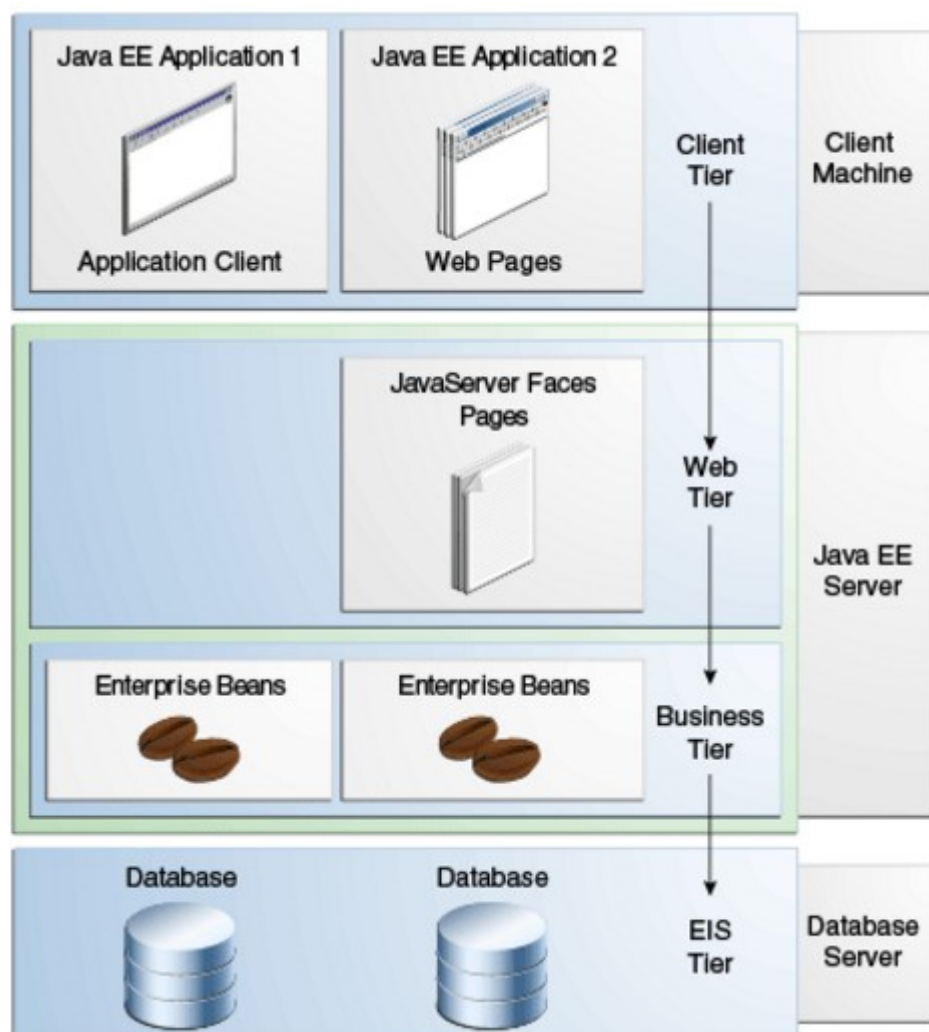
c. See section EE.6.22 on page 183 for details.

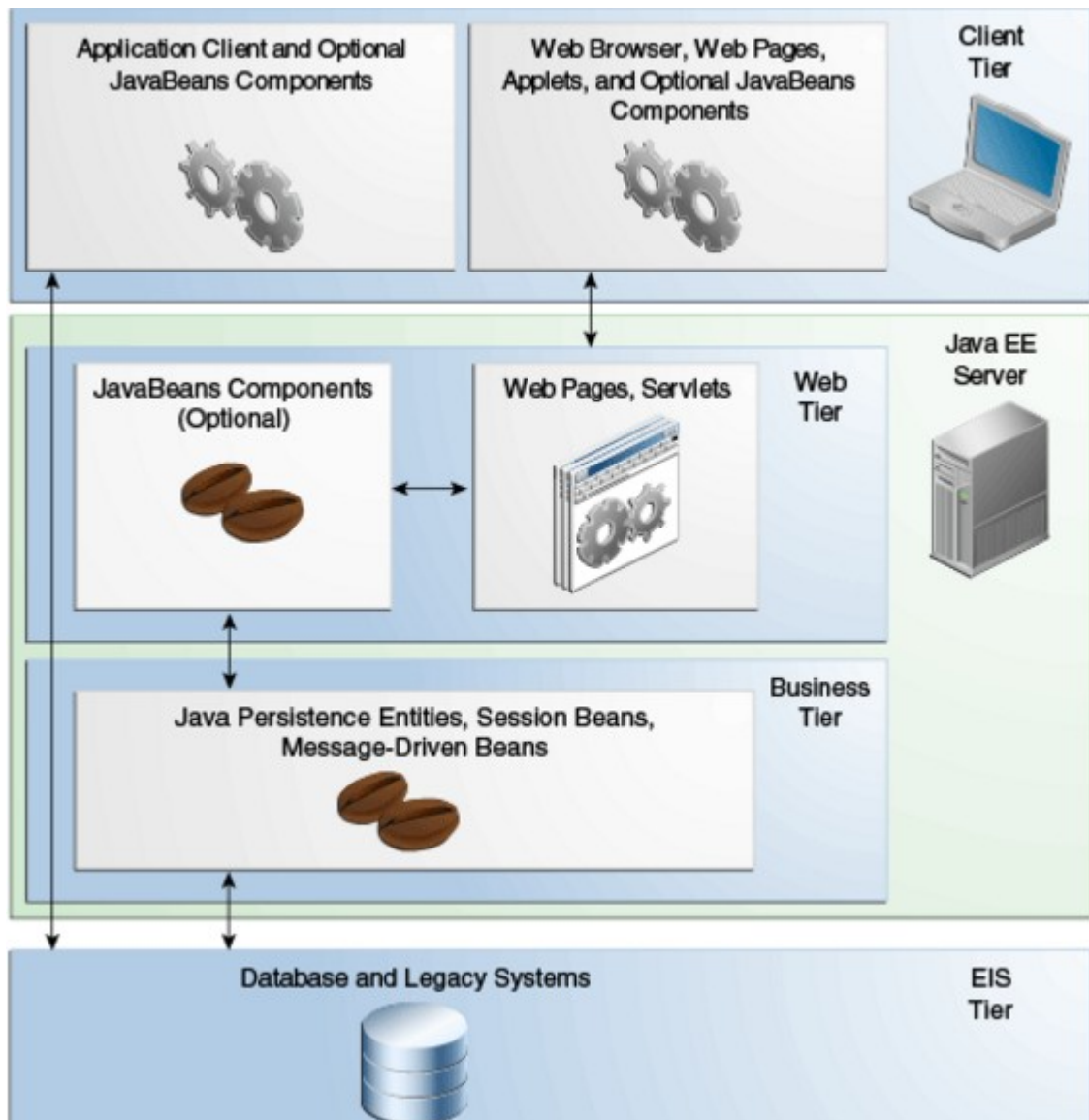
11. **Java EE профиль:** поименованное множество технологий Java EE; в Java EE 7.0 определен один профиль – web-профиль (Servlet, JSP, JSTL, JSF, WebSocket, EL).



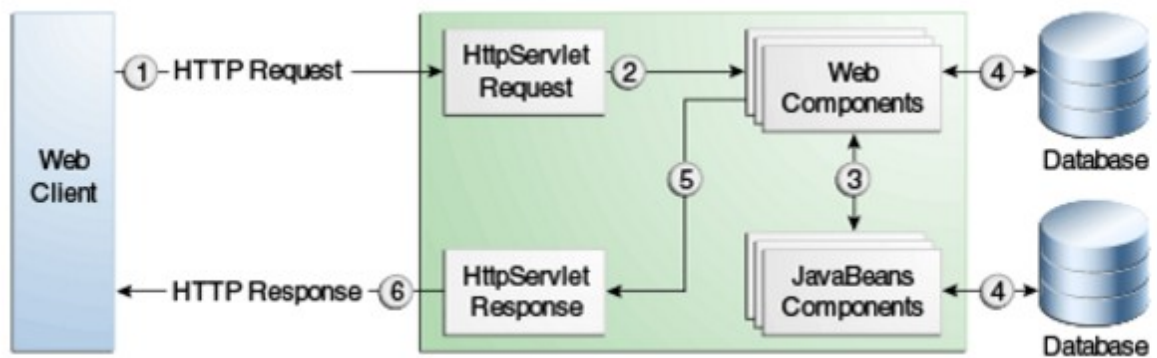
12. **Архитектура Java EE:** 4 типа контейнеров, контейнер – логическое объединение технологий с общим жизненным циклом, предназначен для сокрытия технических сложностей; каждый контейнер поддерживает свой набор интерфейсов; любое приложение разворачивается в один или несколько контейнеров.

16. **EJB Container:** службы управления компонентами EJB, все к чему можно обратиться по RMI-IIOP.
17. **Java EE SDK: Java EE Software Developer Kit** – комплект средств разработки, позволяющий разрабатывать приложения на платформе Java EE; в состав Java EE входит GlassFish Application Server.
18. **Java EE сетевые протоколы:** HTTP, HTTPS, RMI (JMRP), RMI-IIOP, IIOP (CORBA).
19. **Java EE контейнеры:** логические компоненты инфраструктуры: Applet Container, Web Container, Application Client Container, EJB Container.
20. **Java EE: четырехуровневая модель приложения**



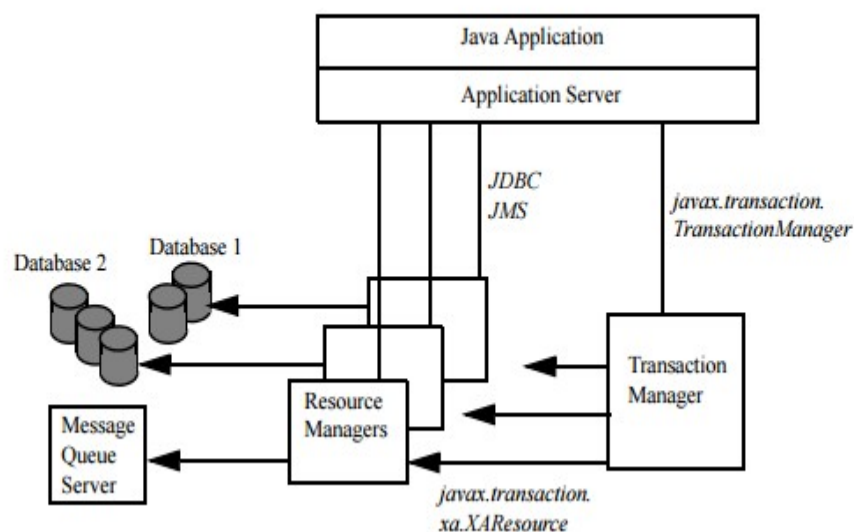


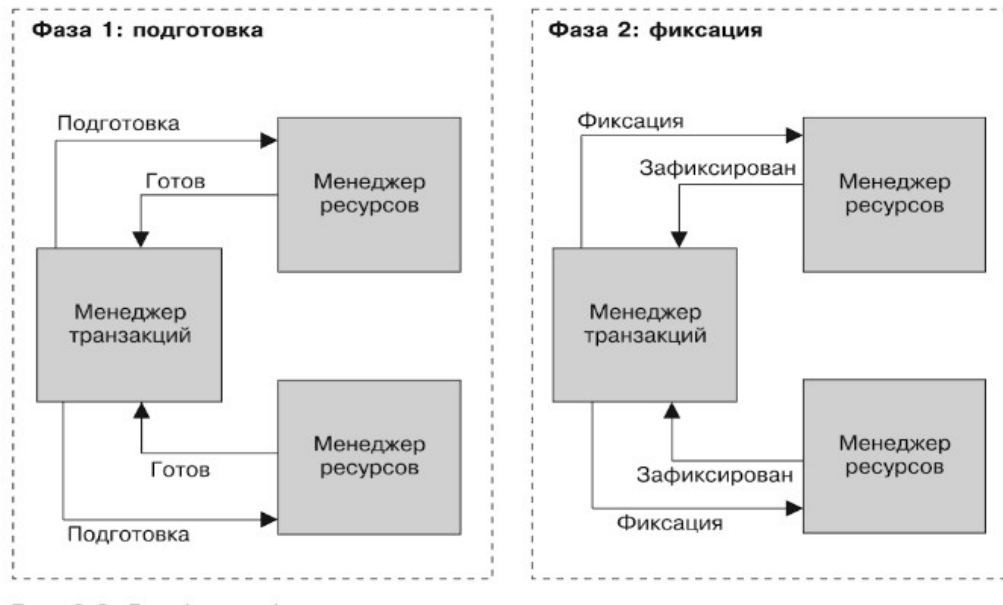
21. Java EE: web-приложение



Description of "Figure 6-1 Java Web Application Request Handling"

22. **Java EE: Java EE Server (Application Server)** – сервер обеспечивающий работу web и ejb –контейнеров.
23. **Java EE технологии:** Общий принцип: настройка на среду работы, применение модели провайдеров.
24. **Java EE: Базовые службы (сервисы):**
25. **Java EE CDI: Context and Dependency Injection** – внедрение зависимостей (DI – Мартин Фаулер); система аннотаций (@), с помощью которых могут описаны применяемые технологии; применяется, как правило, к POJO (Plain Old Java Object – простой Java-объект в старом стиле).
26. **Java EE JTA: Java Transaction API, JTA/XA: JTA Extended Architecture. Модели:** локальная (менеджер ресурсов), программная (возлагается на программиста), декларативная (выполняется контейнером). ACID.





27. Java EE JPA: Java Persistence API

JSR

Community

Expert Group

Summary

Proposal

Detail (Summary & Proposal)

Nominations

JSRs: Java Specification Requests

JSR 338: Java™ Persistence 2.1

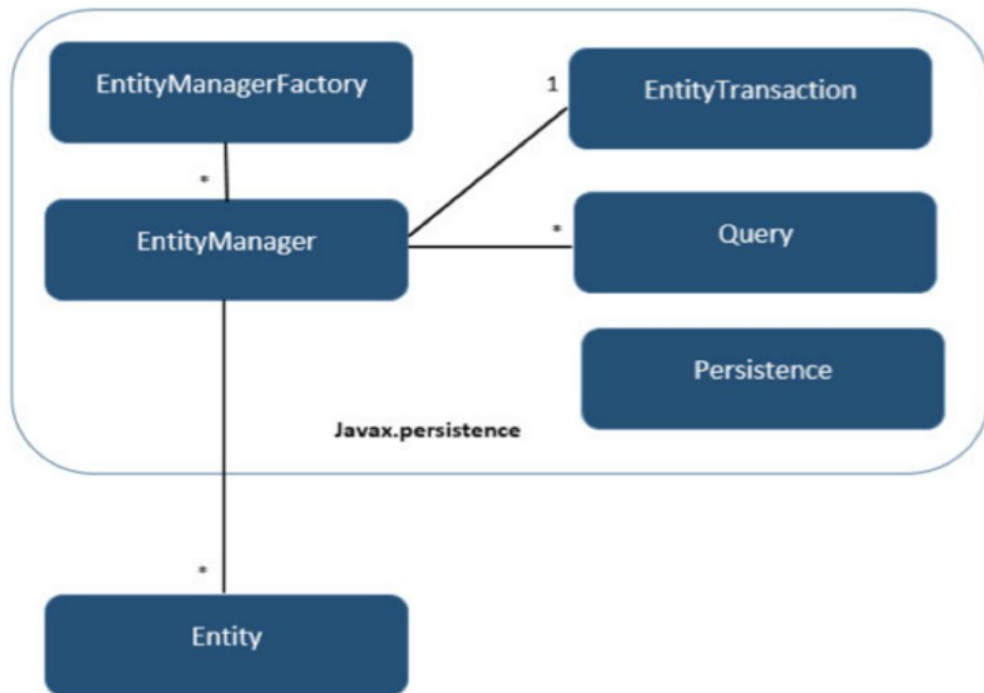
Stage

Final Release

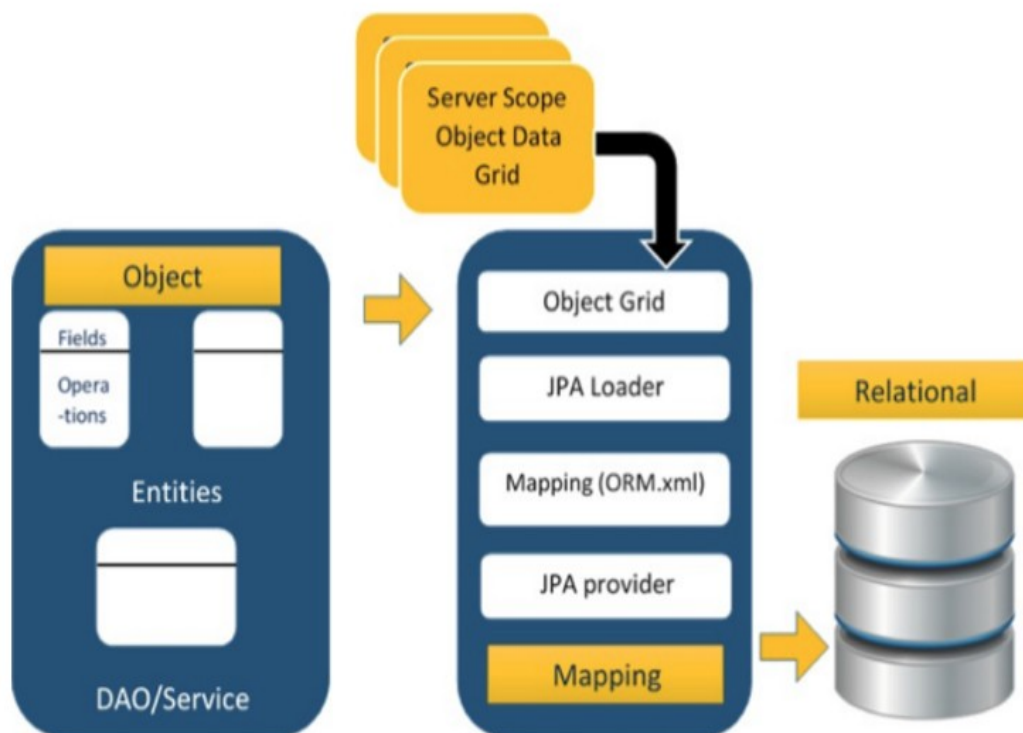
Final Approval Ballot

Proposed Final Draft

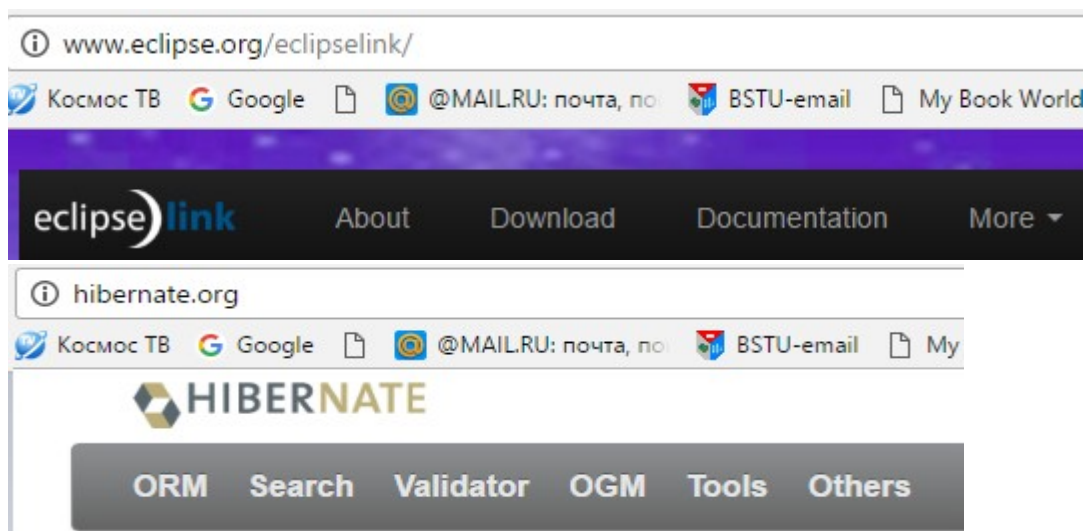
28. **Java EE JPA:** Entity, POJO (Plain Old Java Object – простой Java-объект в старом стиле).
29. **Java EE JPA:** `java.persistence`, EntityManagerFactory – фабрика классов для EntityManager; EntityManager – интерфейс для управления операциями сохранения; Entity-сущность, которая может быть сохранена как база данных, EntityTransaction – класс для поддержки транзакций, Persistence – класс со статическими методами для создания экземпляров EntityManagerFactory, Query – интерфейс реализуется вендором для доступа к реляционным объектам.



30. **Java EE JPA:** **ORM** - Object Relation Mapping, **DAO** - Data Access Object (объект для доступа к данным), **JPA Provider:** например: EclipseLink 2.5 (образцовая реализация JPA 2.1), Hibernate, TopLink; Mapping file (OXM - Object XML Mapping); **JPA Loader:** механизм кэш памяти (Object Grid) хранящей POJO-объекты, соответствующие реляционным объектам; **Object Grid:** данные в памяти

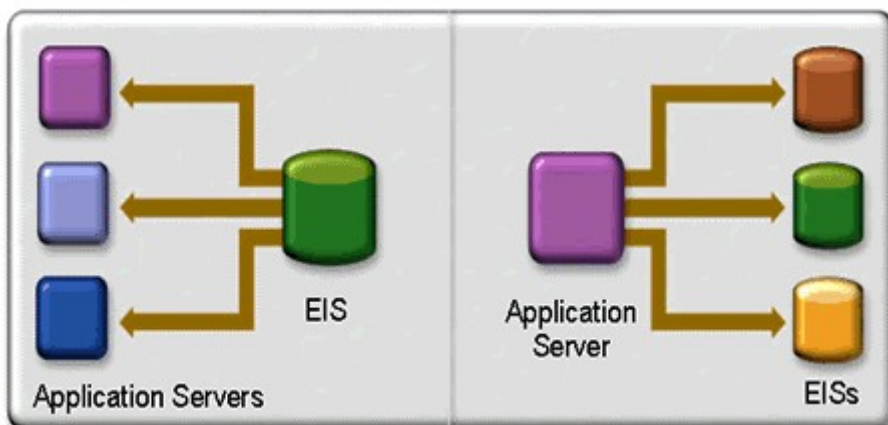


31. **Java EE JPA: JPQL** – Java Persistence Query Language, создан на основе HQL (Hibernate Query Language).
32. **Java EE JPA:** реализации EclipseLink 2.5 (образцовая реализация JPA 2.1), Hibernate, TopLink.

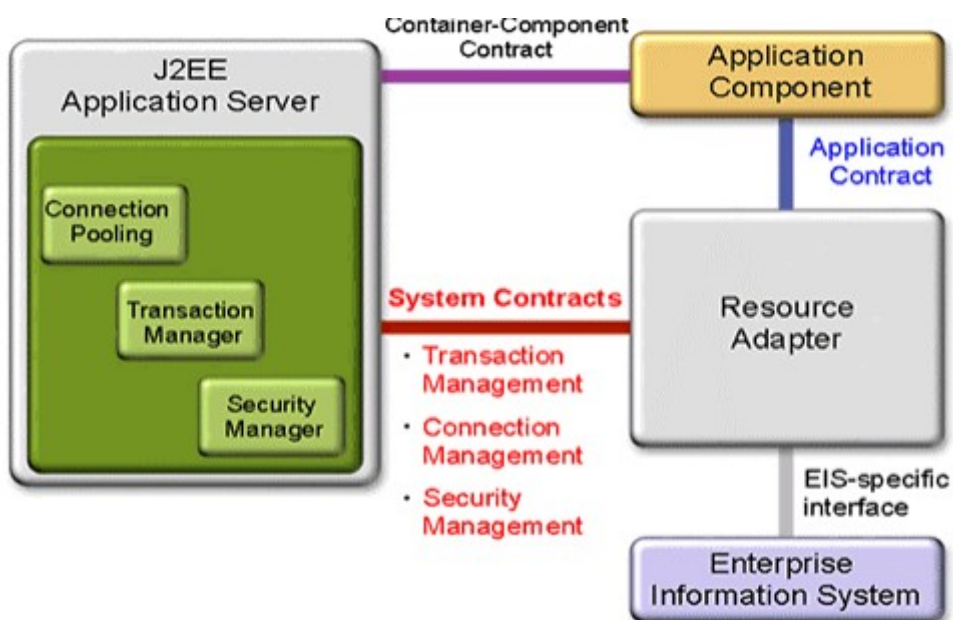


33. **Java EE JCA: Java Connector Architecture** – стандартная архитектура для доступа к ресурсам EIS

(Enterprise Information System – информационная система предприятия)



34. **Java EE JCA:** архитектура коннектора.



35. **Java EE JAAS: Java Authentication and Authorization Service.**

Реализация в Tomcat

1. **JAAS (Java Authentication and Authorization Service)** – Java-сервис по аутентификации и авторизации.
2. Для хранения информации о ролях пользователей, об именах пользователей и их паролях в сервере Apache Tomcat используется специальный xml-файл: **tomcat-users.xml**
3. Каждый пользователь характеризуется: именем, паролем и списком ролей, приписываемых ему.
4. Защищенный ресурс (html –страница, jsp-страница, сервлет) привязывается к ролям. Все пользователи, содержащие в своем списке роль,

обеспечивающую доступ (привязанную роль) к некоторому ресурсу, имеют возможность доступа к данному ресурсу.

5. Технология JAAS предполагает две формы аутентификации пользователей: **BASIC** и **FORM**.
6. BASIC-аутентификация подразумевает использование стандартной формы ввода данных для аутентификации (имени и пароля).
7. FORM-аутентификация подразумевает использование собственной формы (разработанной пользователем, но по определенным правилам jsp-страницы) для аутентификации.

Применение JAAS-технологии сводится к xml-описанию необходимой информации в дескрипторе приложения web.xml и к разработке форм для ввода имени и пароля пользователя и сообщения об ошибке (для FORM-аутентификации).

```
<!-- -----начало web.xml ----- -->
    <security-constraint>
        <web-resource-collection>
            <web-resource-name>JSP-Servlet Sss</web-resource-name>
            <url-pattern>/index.jsp</url-pattern>
            <http-method>GET</http-method>
            <http-method>POST</http-method>
        </web-resource-collection>
        <auth-constraint>
            <role-name>Rolexy</role-name>
        </auth-constraint>
    </security-constraint>
    <login-config>
        <auth-method>BASIC</auth-method>
    </login-config>
    <security-role>
        <role-name>Rolexy</role-name>
    </security-role>
<!-- -----продолжение web.xml ----- -->
```

36. **Java EE WebSocket: WebSocket** – протокол дуплексной связи (RFC 6455, Спецификация W3C). Первоначально соединение по HTTP, а потом переход websocket-протокол.
37. **Java EE WebSocket: соединение**

```
GET /chat HTTP/1.1
Host: server.example.com
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==
Sec-WebSocket-Origin: http://example.com
Sec-WebSocket-Protocol: chat, superchat
Sec-WebSocket-Version: 7
```

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: s3pPLMBiTxaQ9kYGzzhZRbK+xOo=
Sec-WebSocket-Protocol: chat
```

38. Java EE WebSocket: клиент

```
function exe_start() {
    if (ws == null){
        ws = new WebSocket('ws://localhost:1927/Websockets.websocket');
        ws.onopen = function () { ws.send('Соединение');}
        ws.onclose = function (s) {console.log('onclose ', s); }
        ws.onmessage = function (evt) {ta.innerHTML+='\n'+evt.data;}
        bstart.disabled = true;
        bstop.disabled = false;
    }
}

function exe_stop() {
    ws.close(3001, ' stopapplication');
    ws = null;
    bstart.disabled = false;
    bstop.disabled = true;
}
```

39. Java EE WebSocket: сервер

```

import javax.websocket.OnClose;
import javax.websocket.OnError;
import javax.websocket.OnMessage;
import javax.websocket.OnOpen;
import javax.websocket.Session;
import javax.websocket.server.ServerEndpoint;

@ServerEndpoint("/actions")
public class DeviceWebSocketServer {

    @OnOpen
    public void open(Session session) {
    }

    @OnClose
    public void close(Session session) {
    }

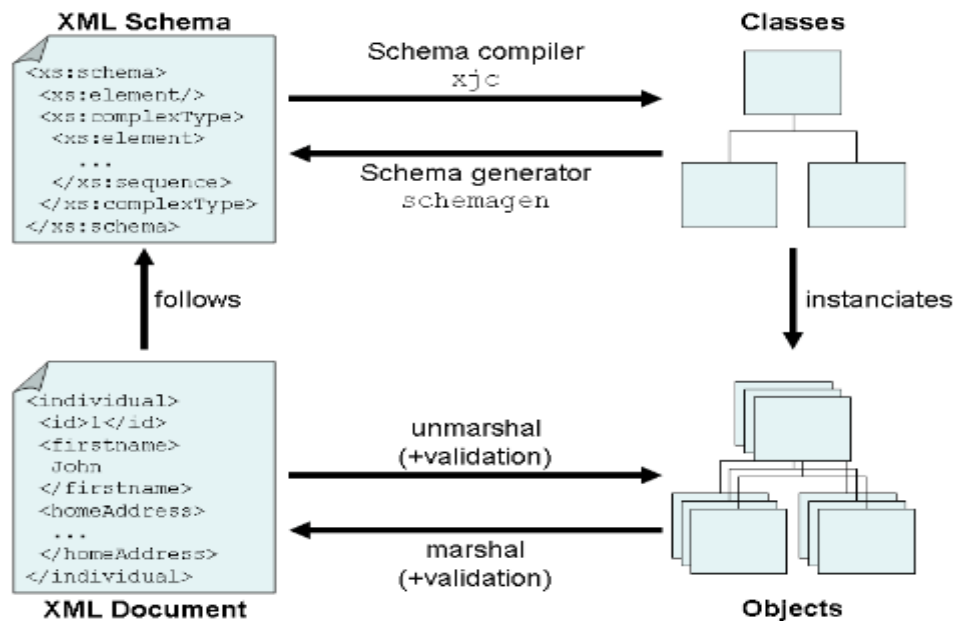
    @OnError
    public void onError(Throwable error) {
    }

    @OnMessage
    public void handleMessage(String message, Session session) {
    }
}

```

Java EE JAX-RPC: версия 1.1, SOAP 1.1

40. **Java EE JAX-WS:** JSR 224, версия 2.1, SOAP 1.1, SOAP 1.2, XML/HTTP – пересылка XML-сообщений (без SOAP), использует JAXB
41. **Java EE JAX-RS:** JSR 311, REST
42. **SAAJ:** API низкого уровня для работы с SOAP Web Service.
43. **Java EE JAXR: Java API for XML Registries.** UDDI – частный случай. Регистрация в хранилище (базе данных) XML-информации о Web-сервисах. Полное API для провайдеров: Registry provider, JAXR-провайдер.
44. **Java EE JAXB: Java Architecture for XML Binding.** JSR 222. Специальный инструмент для пересылки (сериализация/десериализация в XML) объектов.



45. **Java EE JSON-P: JSON with padding** – способ запросить данные, которые находятся в другом домене (страница полученная с сервера, не может связаться с другим сервером – политика ограничения домена), `<script>` – исключение; можно запросить js-скрипт, начиненный json-данными, только GET-метод

```
<script type="text/javascript"
  src="http://server2.example.com/Users/1234?jsonp=parseResponse">
</script>
```

```
parseResponse({"Name": "Foo", "Id": 1234, "Rank": 7});
```

46. **Java EE JSON-P**: альтернатива CORS, должен поддерживать браузер, применение заголовка (от сервера) `Access-Control-Allow-Origin` с указанием доверенных серверов.

47. **Java EE JASPIC: Java Authentication Service Provider Interface for Container** – контракт между сервером и разработчиком (для).

48. oo

49. oo

50. oo

51. oo

52. o

53. oo

54. oo

55.