

The Organization and Management of Enterprise Java Related Projects at Eclipse Foundation

Suren Konathala

- Contributor

- Contributor

- Contributor

August 2018

Contents

[Introduction](#)

[Purpose of this document](#)

[What is not covered in this](#)

[Eclipse Enterprise for Java \(EE4J\)](#)

[Eclipse Jakarta EE Platform](#)

[Summary of Projects governed by Jakarta EE Working Group](#)

[High level overview](#)

[Summary of the process in 100 words](#)

[Stakeholders/Players](#)

[EE4J PMC](#)

[The Jakarta EE Working Group](#)

[The Jakarta EE Platform](#)

[Other specifications](#)

[JakartaEE Working Group Governance Structure / Committees](#)

[Project lifecycle](#)

[Notes \(Remove before publishing\)](#)

Introduction

TBD

Purpose of this document

- To document organizational structure at Eclipse Foundation that manages enterprise Java related projects. All the projects are tracked under EE4J top level project and Java EE related projects contributed by Oracle are part of it..
- To document stakeholders involved and their roles and responsibilities.
- To point out the process on how new features are added (Including proposal, submission, review, test, acceptance).

- To point out (define) how bug fixes are managed (Including proposal, submission, review, test, acceptance).

What is not covered

- Status of individual projects
- Details of what each project is about

Eclipse Enterprise for Java (EE4J)

Eclipse Enterprise for Java (EE4J) is an open source initiative to create standard APIs, implementations of those APIs, and technology compatibility kits for Java runtimes that enable development, deployment, and management of server-side and cloud-native applications. EE4J is based on the Java™ Platform, Enterprise Edition (Java EE) standards, and uses Java EE 8 as the baseline for creating new standards.

The mission of Eclipse EE4J is to create standard APIs, implementations of those APIs, and technology compatibility kits for Java runtimes that enable development, deployment, and management of server-side and cloud-native applications. EE4J is based on the Java EE standards, and uses Java EE 8 as the baseline for creating new standards.

EE4J enables the use of nimble processes, flexible licensing, and an open governance process for evolution of the platform. An open process, that is not dependent on a single vendor or lead, encourages participation and innovation, and serves the collective interests of the entire community.

EE4J establishes commonality between its constituent projects by defining an integrated set of standards using common processes and common compatibility requirements. EE4J provides compatibility for existing and new users by providing compatibility across Java EE 8 and EE4J versions.

The success of EE4J depends upon:

- Successful transfer of Java EE specifications, APIs, TCKs, and reference implementations from Oracle to the EE4J project.
- A nimble, flexible and open process for evolving JakartaEE standard APIs, implementations of those APIs, and technology compatibility kits.
- A strong community of developers, vendors, and end users who support and evolve the EE4J projects.

- Adapting and evolving the Jakarta EE platform as a standard and an open source implementation of that standard, and delivering innovations that address new requirements of existing users, and that attract new users.
- Meeting well-defined compatibility standards across JakartaEE implementations, and across Java EE 8 and EE4J versions.
- Enabling competing vendors and complementary technology providers to provide innovations that add value to EE4J technologies.

Eclipse Jakarta EE Platform

The Eclipse Jakarta EE Platform project produces the Jakarta EE platform specification, which is an umbrella specification that aggregates all other Jakarta EE specifications.

Summary of Projects governed by Jakarta EE Working Group

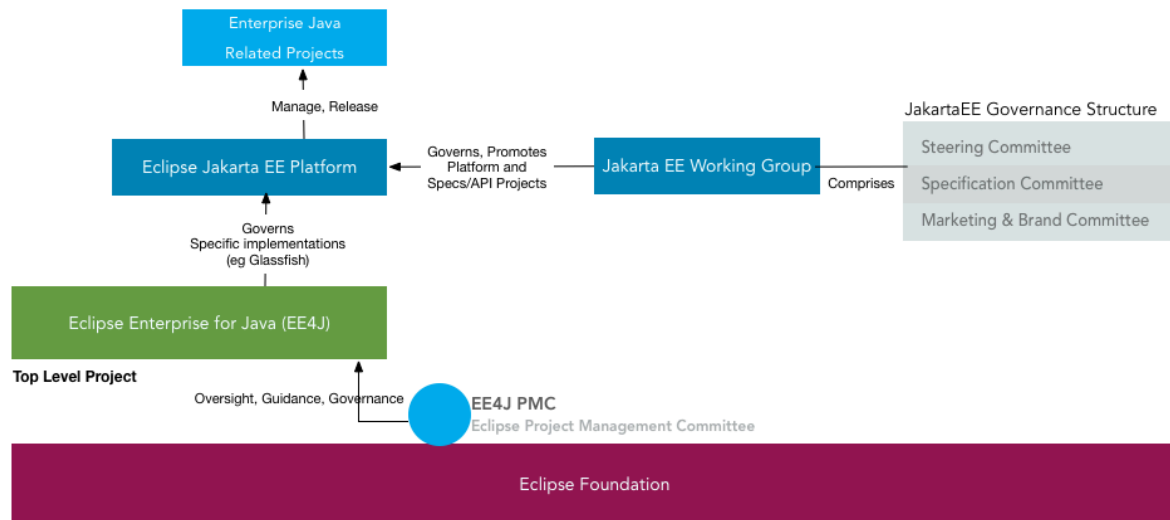
No	Eclipse Foundation Project Name	Scope	Related Oracle Java EE Project	JSR
1	Eclipse Project for JAX-WS	Eclipse Project for JAX-WS provides the API and TCK for 1. Java API for XML-Based Web Services (JAX-WS) 2.0 (starting from the specification defined by JSR-224) 2. Java APIs for XML Messaging (starting from the specification defined by JSR-67) 3. Web Services Metadata for the Java Platform (starting from the specification defined by JSR-181)	Java API for XML-Based Web Services (JAX-WS) 2.2	JSR 224
2	Eclipse Project for Interceptors	Eclipse Project for Interceptors provides the API and TCK for Interceptors, starting from the specification defined by JSR-318.	Interceptors 1.2	JSR 318
3	Eclipse Project for JASPIC	Eclipse Project for JASPIC provides the API and TCK for Java Authentication Service Provider Interface for Containers, starting from the specification defined by JSR-196.	Java Authentication Service Provider Interface for Containers 1.1	JSR 196
4	Eclipse Project for JSP	Eclipse Project for JSP provides the API and TCK for JavaServer Pages, starting from the specification defined by JSR-245.	JavaServer Pages 2.3	JSR 245

5	Eclipse Project for Concurrency Utilities	Eclipse Concurrency provides the API, reference implementation and TCK for Concurrency Utilities for Java™ EE, starting from the specification defined by JSR-236.	Concurrency Utilities for Java EE 1.0	JSR 236
6	Eclipse Project for Common Annotations	Eclipse Project for Common Annotations provides the API and TCK for Common Annotations for the Java™ Platform, starting from the specification defined by JSR-250.	Common Annotations for the Java Platform 1.3	JSR 250
7	Eclipse Project for JTA	Eclipse Project for JTA provides provides the API and TCK for Java™ Transaction API, starting from the specification defined by JSR-907.	Java Transaction API (JTA) 1.2	JSR 907
8	Eclipse Project for Enterprise Security	Eclipse Enterprise Security provides the API and TCK for Java™ EE Security API, starting from the specification defined by JSR-375.	Java EE Security API 1.0	JSR 375
9	Eclipse Project for JSTL	Eclipse Project for JSTL provides the API and TCK for JavaServer Pages Standard Tag Library, starting from the specification defined by JSR-52.	Standard Tag Library for JavaServer Pages (JSTL) 1.2	JSR 52
10	Eclipse Project for JavaMail	Eclipse Project for JavaMail provides the API, reference implamentation and TCK for JavaMail, starting from the specification defined by JSR-919.	JavaMail 1.6	JSR 919
11	Eclipse Project for JMS	Eclipse Proeject for JMS provides the API and TCK for Java™ Message Service (JMS) API, starting from the specification defined by JSR-343.	Java Message Service API 2.0	JSR 343
12	Eclipse Project for WebSocket	Eclipse Project for WebSocket provides the API and TCK for Java API for WebSocket, starting from the specification defined by JSR-356.	Java API for WebSocket 1.1	JSR 356
13	Eclipse Project for JAX-RS	Eclipse Project for JAX-RS provides the API and TCK for Java API for RESTful Web Services (JAX-RS), starting from the specification defined by JSR-370.	Java API for RESTful Web Services (JAX-RS) 2.1	JSR 370
14	Eclipse Project for JSON Processing	Eclipse Project for JSON-P provides the API, TCK, and an implementation for Java API for JSON Processing, starting from the specification defined by JSR-374.	Java API for JSON Processing 1.1	JSR 374
15	Eclipse Implementation of JAXB	Eclipse Implementation of JAXB contains the source code, documentation, and tests for JAXB reference implementation.	NA	NA

15	Eclipse GlassFish Tools	Eclipse GlassFish Tools is intended for integration of GlassFish with Eclipse Web Tools Platform. This includes, but not limited to, starting/stopping the server, publishing to the server and creating/editing server specific project artifacts.	NA	NA
17	Eclipse Project for Stable Jakarta EE APIs	Eclipse Project for Stable Jakarta EE APIs provides a home for stable (legacy) Jakarta APIs, RIs and TCKs which are no longer actively developed.	NA	NA
18	Eclipse Metro	Eclipse Metro contains the source code, documentation, and tests for JAX-WS, SAAJ, Web Services Interoperability Technologies (WSIT) and StAX extensions implementations.	NA	NA
19	Eclipse ORB	This project contains the source code, documentation, and tests for the CORBA ORB that is used in the GlassFish application server project, including the RMI-IIOP support that allows use of the CORBA IIOP protocol under the Java RMI API.	NA	NA
20	Eclipse Jakarta EE Platform	Eclipse Jakarta EE Platform provides the Jakarta EE platform specification and related materials.	NA	NA

High level overview

Organization and Management of Enterprise Java Related Projects at Eclipse Foundation



Summary of the process in 100 words

<TBD>

Stakeholders/Players

EE4J PMC

Provides oversight, guidance, best practices, governance, and several other support functions for all of the open source projects that are part the EE4J Top Level Project. That is the Top Level Project is part of the organizational structure, and the PMC is the actual group that provides the services.

The PMC provides governance for all projects, regardless of what the projects actually do. So this includes both specification and implementation projects. It's also the PMC that oversees the release process (with help from the EMO).

The Jakarta EE Working Group

Though the Specification Committee, is responsible for taking the specifications produced by the open source projects and designating them as "official". They are further responsible for managing the process by which implementations of the specifications are certified.

The Jakarta EE Platform

Is one of the specifications. It happens to be what is referred to as a "profile", so it's a special specification, but there are other specifications that are also special.

Other specifications

<TBD>

JakartaEE Working Group Governance Structure / Committees

There are three committees in the Jakarta EE Working Group viz.,

1. Steering Committee
2. Specification Committee
3. Marketing and Brand Committee

Project lifecycle

<TBD>

Notes (Remove before publishing)

- Focus is on enterprise Java (Java EE) related projects
- "Other Projects" block - projects like Config, Bean Validation, CDI, Batch or maybe JCache. How and where do we add these?