Introducing the Eclipse Foundation Specification Process
Agenda

• Background
• Creating the EFSP
• What is a Specification?
• Eclipse Foundation Specification Process
• EFSP and the JCP
• Certification
Background
Why are we doing this?

• Opportunity meets necessity
• Java EE migration to Eclipse Foundation requires a spec process to replace the JCP
• We expect that this process will be used elsewhere
What’s the Big Deal?

Specifications

• Guides you to implement collectively developed idea
• Support multiple implementations
• Allow for interoperability
Guiding Principles

- “Code First”
- No more “Spec Lead”
- Specifications run as open source projects
- “Compatible” implementations, rather than one “Reference” implementation
- Self-certification
- Branding for compatible implementations of Profiles
Jakarta EE Spec Process: 2018 Key deliverables

- Establish spec process for existing (JCP) and new specs
- Compatibility process
- Brand licensing
Creating the EFSP
Start with The Eclipse Development Process

- Open source rules of engagement
- Governance, structure, definitions, reviews
- General framework for projects
- Day-by-day development rules/process is defined by the project
Open Source Rules of Engagement

- Transparency
- Openness
- Meritocracy
- Vendor neutrality
Development Process
EDP 2018

• “Progress Review”
  • Release Review becomes a kind of Progress Review

• Projects may release within one year of engaging in a successful Progress Review
  • IP Policy must be followed at all times

• Formalize the list of terms
Eclipse Foundation Specification Process

• Spec. development in open source
  • Extends the EDP
  • “Just enough” process
• “Specification Project”
  • Formal alignment of Specification Projects with Working Groups
  • Specification Committee approval
• Participants and Participant Representatives
What is a Specification?
A Specification is...

...a collection of APIs, descriptions of semantic behavior, data formats, and/or protocols intended to enable the development of independent compatible implementations.

![Diagram of Specification components]

- Specification
  - Specification Document
  - Technical Artifacts
  - TCK
  - 1..n Compatible Implementations
Specification Document

- Textual description of the obligations and rules
- May specify optional parts
  - Must be possible to implement all optional parts
- Must not override rules in referenced specifications
TCK

• Specification must designate a TCK
  • May be different for different versions
• Must be developed under an approved Open Source License
Compatible Implementations

• A Specification must reference at least one Compatible Implementation
• Must be developed under an approved Open Source License
• At least one Compatible Implementation must implement all optional features
Open Source License

A “Compatible Implementation” must exist under an “Open Source License”, which is one of:

• Eclipse Public License - v 2.0 (+ Secondary Licenses)
• Eclipse Distribution License - v 1.0 (BSD-3-Clause)
• Apache License - v 2.0.

This list may be augmented by a Working Group with the approval of the EF Board.
Profiles and Platforms

• A specification may be designated as a “Profile”
  • Profiles aggregate other specifications
• A Profile may be designated a “Platform”
The Eclipse Foundation Specification Process
Eclipse Specification Process

• “Just enough” process
• Based on/extends the EDP
• IP flows are similar to the EDP process
• Patent rights need to be addressed
• Customizable
Specification Project

An Eclipse Project that...

• Is designated as a “Specification Project”
• Is “owned” by a Working Group
• Requires Specification Committee approvals
• Has a special class of committer
Who’s Who?

PMC

Governance, oversight

Approval, IP integrity

Specification Committee

Specification Project

Git

Issues

Resolve

Manage, plan, schedule, ...

Employs

Contributors

Pull requests

Push, accept pull requests

Committers

Has member

JAKARTA EE

Employs

Copyright (c) 2018, Eclipse Foundation, Inc. Made available under the Eclipse Public License 2.0 (EPL-2.0)
Specification Committer

• Must be a Member
  • Employed by an Eclipse Foundation Member Company, or
  • Committer Member

• Must be covered by a Working Group Participation Agreement
Participant/Participant Representative

- A Participant Representative (committer) represents the interests of a Participant
  - Appointed to the project (no election)
- Participant
  - Individual Participant
  - Member Participant
Specification Process
Specification Committee Approvals

• Specification Project creation
• Release Plan
• Revision to the Scope
• Progress and Release Reviews
• Designation of a Profile or Platform
• Service Releases.
Revised Committer/Contributor Agreements

• Grant a license that says that if you contribute to an Eclipse project we can use your contributions to create a specification
  • Eclipse Contributor Agreement (ECA)
  • Individual Committer Agreement (ICA)
  • Member Committer Agreement (MCA)
  • Terms of Use

• State that if you post an idea on a Spec Project mailing list that you won’t sue us later if we use your idea
  • Terms of Use
EFSP and JCP
EFSP compared to the JCP...

Code first
Specification Committee
Specification Project
IP Flows through Participants
Collaborative Public Communication
Customizable by Working Group

Specification First
Executive Committee
Expert Group
IP Flows to and through Specification Lead
Led by Spec Lead
Public Communication
... EFSP compared to the JCP...

Document is open source
Eclipse Foundation Specification License
TCK is open source
Eclipse Foundation TCK License
One or more “Compatible Implementations”

Document is closed source
Chosen by Spec Lead
TCK is closed source
Proprietary License & NDA
One normative “Reference Implementation”
... EFSP compared to the JCP

Profiles  Profiles
Platforms  Platforms
Compatibility Claims
Final Specification

Release Candidate

Specification Version
- Specification Document (source and distribution)
- Technical Artifacts (source and distribution)
- TCK (source and distribution)
- Compatible Implementation

Release Review

Final Specification
- Specification Document (read-only text)
- Technical Artifacts (distribution)
- TCK (distribution)
- Compatible Implementation (1..n)

Ratified
Specification Licenses

- Eclipse Foundation Specification License
  - Allows implementers to create implementations of the spec under whatever license they would like
- Eclipse Foundation TCK License
  - Allows implementers to verify that they are compatible with a specific version of a ratified final specification
- Eclipse Foundation Trademark License
  - Allows compatible implementation to use the logo/trademark (e.g. Jakarta EE)
Self Certification

“Compatible Implementation”
• Implements a Final Specification
• Fulfills all of the requirements of the Ratified TCK
• Must publicly post TCK results
Brand

• Compatible Implementations of Profiles
• Eclipse Foundation Trademark License
• Must be an Eclipse Foundation Member
  • .....but no license fees or royalties
The Take Away
The EFSP is...

- Code first
- Open and Transparent
- Community-oriented
- Light(er) weight
- Rigorous IP management
- Built on the success of the EDP
Questions?