



**OSGi**<sup>™</sup>  
**Alliance**

the dynamic module system for Java

# eRCP Standardization Proposal

June 5, 2008

# What is eRCP?

- eRCP is an embedded version of the Eclipse Rich Client Platform
- Reduces RCP size/function to fit on devices
- Utilizes the same OSGi Service Platform as RCP
- Rich widget capabilities
- Mobile device extensions
- eRCP Applications are binary compatible across a range of devices
- Has been available for over a year and is being used for many company internal projects

# Why is eRCP important?

- There are a **HUGE** number of devices out there that are becoming quite capable Java application platforms
- Brings OSGi to devices
- Has extensive rich UI capabilities
- Higher level of device abstraction and integration
- eRCP can be the next mobile Java standard
- Great platform for mobile mash-ups

# Why standardize eRCP?

- Given the broad range of phones, service providers do not typically implement platforms themselves. They spec requirements and look for partners to implement their spec
  - They want to be able specify a standard
  - They want to be able to verify the implementation is compliant
  - They do not want competitors introducing non-compliant implementations that devalue the eRCP benefit of being able to run the same binaries across multiple platforms/devices
  - They are fearful that open source projects are too “fluid”
- JavaDoc by itself is not sufficient to re-implement eRCP
- eRCP does not have a consolidated test suite for all function
- Fills gap in current OSGi spec by providing a UI model

# Proposal

- OSGi Alliance will standardize eRCP
- eRCP would continue to be a subset of RCP so compatibility with RCP is maintained
- Provide API spec with additional detail not in current JavaDoc to enable independent implementations of eRCP
- Collect test suites from platform components, fill in what's missing, and consolidate into Compliance Tests (TCK)
- Implementations with additional function could still be compliant
- Supported by Sprint, ProSyst, Nokia, IBM

# What API is to be standardized?

- OSGi – No, reference OSGi Service Platform Standard
- Eclipse infrastructure – Yes
  - Core runtime, registry, expressions, commands, jobs
- GUI – Yes
  - eSWT, Mobile Extensions, eJFace, ui.workbench
- XML – No, reference W3C standards
- eUpdate – No, it is only one possible deployment model

# Licensing

- Spec would need new license model
  - Spec would contain Eclipse materials under EPL that would remain under EPL
- Compliance tests (TCK) could use existing OSGi TCK license model
  - Internal materials may be EPL or AL2 with terms passed through in notices file
  - OSGi Alliance would grant Eclipse rights to use TCK (normal licensing policy for qualified non-profits)
  - 3rd Party implementer needs to become an Alliance member for TCK access

# What does this mean to OSGi Alliance?

## ➤ Work that would be done in MEG

- Consolidate existing eRCP JavaDoc
- Fill in spec gaps
- Consolidate existing API testcases
- Fill in testcases to produce complete TCK
- Handle spec drafts, feedback, and approvals

## ➤ Give feedback to Eclipse community

- Recommend new API and changes

# What does this mean to Eclipse?

## ➤ For eRCP team

- Work with OSGi eRCP standardization group (some people will likely be in both groups)
- Consider OSGi input (like any other community input)

## ➤ For RCP team

- Provide existing functional documentation
- Answer questions for not well documented areas
- Consider OSGi input (like any other community input)

# Standard Maintenance Plan

- OSGi Alliance provides ongoing input through Eclipse community process
- Update the eRCP standard with each major revision of eRCP (probably yearly)
- Add testcases to TCK as required

# Action Required

- Board vote to agree to Eclipse participation in this process