CDI 1.1 JSR

1. Identification

Submitting Member: Red Hat, Inc.

Name of Contact Person: Pete Muir

E-Mail Address: pmuir@redhat.com

Telephone Number: +44 1252 362709

Fax Number: +44 1252 548410

Specification Lead: Pete Muir

E-Mail Address: pmuir@redhat.com

Telephone Number: +44 1252 362709

Fax Number: +441252 548410

Initial Expert Group Membership:

TBD

Supporting this JSR:

TBD

2: Request

2.1 Please describe the proposed Specification:

Contexts and Dependency Injection for Java EE (CDI) 1.0 was introduced as part of the Java EE 6 platform, and has quickly become one of the most important and popular components of the platform.

CDI defines a powerful set of complementary services that help improve the structure of application code.

- A well-defined lifecycle for stateful objects bound to *lifecycle contexts*, where the set of contexts is extensible
- A sophisticated, typesafe *dependency injection* mechanism, including the ability to select dependencies at either development or deployment time, without verbose configuration
- Support for Java EE modularity and the Java EE component architecture—the modular structure of a Java EE application is taken into account when resolving dependencies between Java EE components
- Integration with the Unified Expression Language (EL), allowing any contextual object to be used directly within a JSF or JSP page
- The ability to *decorate* injected objects
- The ability to associate interceptors to objects via typesafe *interceptor bindings*
- An *event notification* model
- A web *conversation context* in addition to the three standard web contexts defined by the Java Servlets specification
- An SPI allowing *portable extensions* to integrate cleanly with the container

Since the final release of the CDI 1.0 specification a number of issues have been identified by the community and a update to the specification will allow these to be addressed. A list of proposed updates is provided here, however the EG will consider other issues raised as the JSR progresses.

- Global ordering of interceptors and decorators, as well as global enablement of alternatives [https://issues.jboss.org/browse/CDI-48]
- An API for managing built in contexts, allowing the built in implementation of the conversation context to be used outside of JSF [https://issues.jboss.org/browse/CDI-30]
- An embedded mode allowing startup outside of a Java EE container [https://issues.jboss.org/browse/CDI-26]
- Bean declaration at constructor level [https://issues.jboss.org/browse/CDI-55]
- Static injection [https://issues.jboss.org/browse/CDI-51]
- Inclusion of @Unwraps from Seam Solder [https://issues.jboss.org/browse/CDI-89]
- Numerous minor enhancements to the Portable Extensions SPI
- Client controlled contexts allowing for SaaS style multi-tennancy [https://issues.jboss.org/browse/CDI-103]
- Better support for CDI in libraries when used in the Java EE platform [https://issues.jboss.org/browse/CDI-84]
- Send CDI events for Servlet events [https://issues.jboss.org/browse/CDI-38]
- Application lifecycle events [https://issues.jboss.org/browse/CDI-86]

A full list of proposed features is available in the issue tracker http://bit.ly/cdi11features.

Additionally, a number of inconsistencies have been noticed and clarifications requested and the JSR intends to address all of these. A full list is available in the issue tracker http://bit.ly/cdi10bugs.

2.2 What is the target Java platform? (i.e., desktop, server, personal, embedded, card, etc.)

The target platform is Java EE.

2.3 The Executive Committees would like to ensure JSR submitters think about how their proposed technology relates to all of the Java platform editions. Please provide details here for which platform editions are being targeted by this JSR, and how this JSR has considered the relationship with the other platform editions.

The work will track Java EE 7, to be released as part of the Java EE 7 platform.

Should this JSR be voted on by both Executive Committees?

No. It should be voted on by the Java SE / EE Executive Committee only.

2.5 What need of the Java community will be addressed by the proposed specification?

The goal of the proposed specification is to address the needs of the Java community by augmenting the functionality of Contexts and Dependency Injection to address requests received from the Java community for additional features.

2.6 Why isn't this need met by existing specifications?

See 2.5 above. These new features are specific to the Contexts and Dependency Injection architecture, which does not yet provide them.

2.7 Please give a short description of the underlying technology or technologies:

See above.

2.8 Is there a proposed package name for the API Specification? (i.e., javapi.something, org.something, etc.)

The API specification will continue to use the javax.enterprise package.

2.9 Does the proposed specification have any dependencies on specific operating systems, CPUs, or I/O devices that you know of?

No.

2.10 Are there any security issues that cannot be addressed by the current security model?

No.

2.11 Are there any internationalization or localization issues?

No.

2.12 Are there any existing specifications that might be rendered obsolete, deprecated, or in need of revision as a result of this work?

No.

2.13 Please describe the anticipated schedule for the development of this specification.

February 2011	Expert group formed
Q3 2011	Early Draft Review
Q1 2012	Public Review
Q3 2012	Final release.

Jan 2011 Expert Group formed
Q3 2011 Early Draft
Q1 2012 Public Review
Q3 2012 Final Release

2.14 Please describe the anticipated working model for the Expert Group working on developing this specification.

The primary means of communication will be issue tracker, email, specification drafts and wiki. Phone conferences and face-to-face meetings will be scheduled if needed.

2.15 It is important to the success of the community and each JSR that the work of the Expert Group be handled in a manner which provides the community and the public with insight into the work the Expert Group is doing, and the decisions that the Expert Group has made. The Executive Committees would like to ensure Spec Leads understand the value of

this transparency and ask that each JSR have an operating plan in place for how their JSR will address the involvement of the community and the public. Please provide your plan here, and refer to the Spec Lead Guide for a more detailed description and a set of example questions you may wish to answer in your plan.

We will leverage the collaborative tools provided by the JBoss Community infrastructure. We have a public issue tracker for tracking most issues at https://issues.jboss.org/browse/CDI. We have a public EG discussion list at https://lists.jboss.org/mailman/listinfo/cdi-dev which is free for anyone to join and post to. We will have an EG-private mailing list, and we will a public wiki. The reference implementation will be developed entirely in the public Weld project in the JBoss Community. The TCK will be developed entirely in the public CDI TCK in the JBoss Community. We will leverage the Early Draft feature of JCP 2.6 to allow the public to see the spec in progress.

2.16 Please describe how the RI and TCK will de delivered, i.e. as part of a profile or platform edition, or stand-alone, or both. Include version information for the profile or platform in your answer.

Red Hat, Inc. will deliver a Reference Implementation (RI) and Technology Compatibility Kit (TCK) compatible with Java EE 7 and license compatible with Java EE licensing as part of the final specification.

2.17 Please state the rationale if previous versions are available stand-alone and you are now proposing in 2.13 to only deliver RI and TCK as part of a profile or platform edition (See sections 1.1.5 and 1.1.6 of the JCP 2 document).

N/A

2.18 Please provide a description of the business terms for the Specification, RI and TCK that will apply when this JSR is final.

The Specification will be licensed via the ASL 2.0 license.

The RI and TCK will be licensed via the ASL 2.0 license.

3: Contributions

3.1 Please list any existing documents, specifications, or implementations that describe the technology. Please include links to the documents if they are publicly available.

The CDI 1.0 Specification (JSR 299)

http://jcp.org/eng/jsr/detail?id=220

Java Platform, Enterprise Edition Specification Version 6

http://jcp.org/en/jsr/detail?id=316

3.2 Explanation of how these items might be used as a starting point for the work.

The specification and javadoc from JSR 299 will be the starting point for this work.