



# How to Participate in the Future of Java!

*Heather VanCura*

*Jfokus 2013*

*<http://jcp.org>*

*Twitter @JCP\_org*

# For the *community*, by the *community*



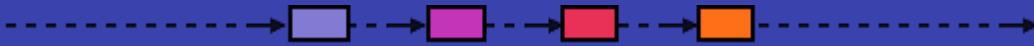
- The Java Community Process (JCP) is the open, inclusive process to develop and revise Java technology specifications (JSRs), reference implementations (RI), and technology compatibility kits (TCK).
  - The JCP program now has over 1,500 corporate and individual members.
  - More than 350 Java technology specifications are in development in the JCP program; ~two thirds have reached Final Release.

# Why participate?



- As a corporation:
  - To influence the technologies that drive your market.
  - Because it's cheaper than developing everything yourself.
  - To provide opportunities for your developers.
  - To increase the size of your market.
  - To gain a competitive advantage (ship new products faster.)
- As an individual:
  - Acquire knowledge: Learn from experts, early access.
  - Make better specifications based on real world experience.
  - Experience and skill development.
  - Professional Visibility: Curriculum, articles, presentations.
  - Fame and fortune – grow your reputation and network.

# Levels of participation



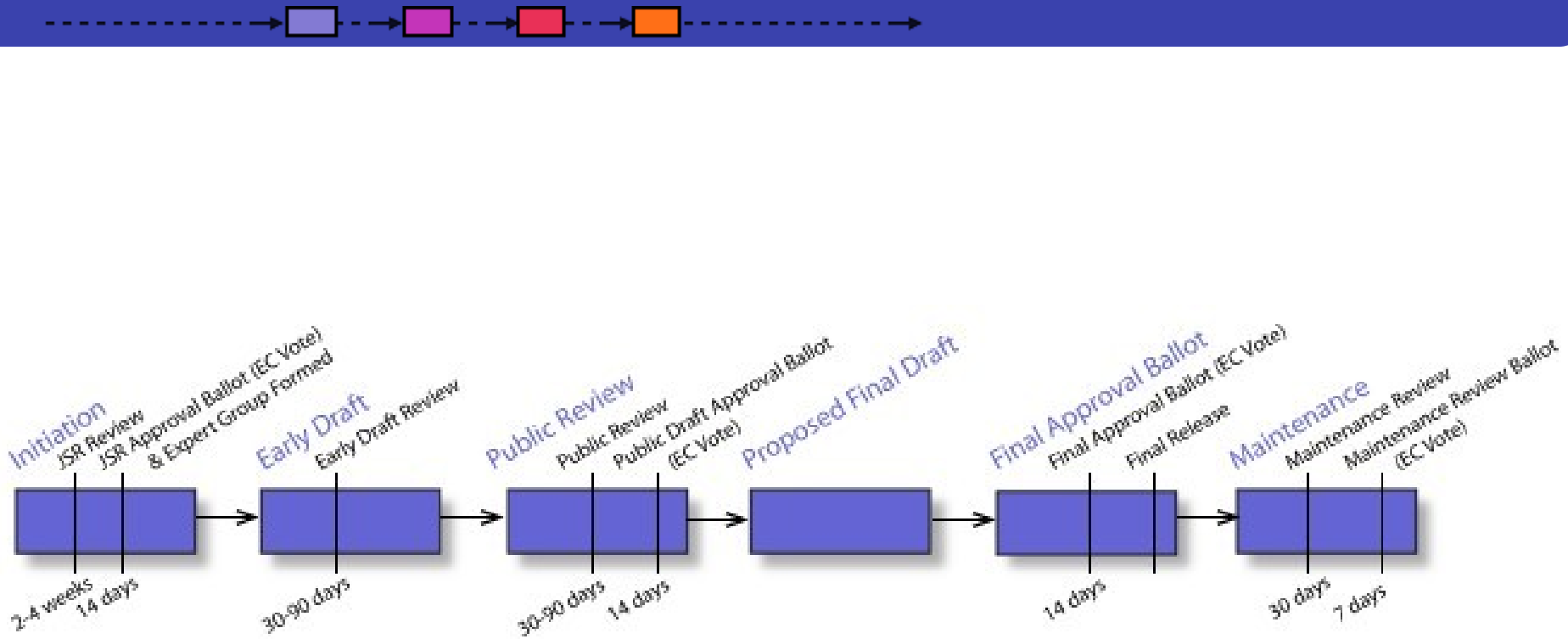
- Members.
  - Review specs, may participate in Expert Groups, vote in Executive Committee elections; flexible time commitment.
- Expert Group members.
  - Create JSRs (write the spec, develop the RI and TCK); expert in field, high time commitment, help drive the direction of the JSR; moderate time commitment.
- Specification Leads.
  - Responsible for leading the JSR & Expert Group (EG), delivery of spec, RI & TCK; a leading expert in area of JSR; considerable time commitment and recognition.
- Executive Committee members.
  - Define JCP governance, processes, and contractual terms of membership. Vote on JSRs at defined stages through the process. Moderate time commitment.

# Join the JCP



- Join the JCP program as a corporation, non-profit, or individual member.
  - Free for non-profits , Java User Groups, and individuals!
- See **<http://jcp.org/en/participation/membership2>**
- Register for an account on JCP.org.
- Download, fill out, and scan, fax or mail the JSPA form to the JCP Program Office.

# The JSR development cycle



# JCP.next



- A series of three JSRs:
- *JSR 348: Towards a new version of the Java Community Process* completed in October 2011.
- This JSR focused on relatively simple changes that we were able to implement within about six months.
- *JSR 355: Executive Committee Merge* completed in August 2012.
  - Implementation began in October 2012 and will be completed in October 2013.
- *JSR 358: A major revision of the Java Community Process* is in progress now.
  - This JSR will implement more complex changes and will modify the JSPA as well as the Process Document.

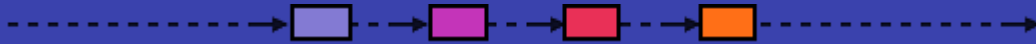
# JCP.next.1 (JSR 348)



- The JSR was completed within 6 months and implemented a number of important changes JCP 2.8, implemented in October 2011 (~14 months ago).
- Three themes aimed at:
  - Transparency
  - Participation
  - Agility



# JCP.next.2 (JSR 355)



# JSR 355 (the EC merge)



- Because *Java is One Platform*, and because we expect ME and SE to converge over time, JSR 355 was introduced to merge the two Executive Committees into one.
- This JSR reduces the number of EC members and keep the same ratio (2:1) of ratified and elected seats.
  - Oracle will gave up one of its permanent seats.
  - No other member may hold more than one seat (IBM gave up one of their seats).
- No other significant changes to our processes.

# JCP.next.3 (JSR 358)



- The last of the three JCP.next JSRs, JSR 358, is in process now.
- This will tackle more complex issues such as licensing, IP flow, and governance, and will modify the JSPA.
- This is **a big deal** (because all the lawyers will get involved.)
  - It will take a long time.

# Legal framework and governance



- Java Specification Participation Agreement (JSPA)
  - A legal contract between members and Oracle.
  - Addresses Intellectual Property (IP) grants and the terms under which the spec, RI, and TCK should be licensed.
  - <http://www.jcp.org/aboutJava/communityprocess/JSPA2.pdf>.
- The Process Document
  - Defines the governance of the organization.
  - Defines the processes that are used to submit define, develop, review, approve, and maintain specifications.
  - Defines the obligations to produce an RI and TCK.
  - <http://jcp.org/en/procedures/jcp2>.

# Modifying the JSPA



- The JSPA has not been significantly modified since 2002. (some minor changes were introduced in 2005.)
- Since then the organization and the environment in which we operate have changed significantly.
- The document is due for updating and clean up.
- However, it is extremely complex and difficult to understand.
- We must be very careful when making changes.
- Plus...

# All the lawyers will get involved

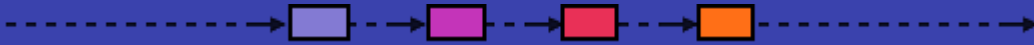


# Our shopping list



- Intellectual Property (IP) flow.
- Compatibility policy.
- Licensing and open-source.
- Independent implementations.
- The role of the RI.
- The role of individuals.
- Transparency.
- Refactoring the JSPA.
- Fee structure.
- And lots more (see the **Issue Tracker** for the full list.)

# Why it matters



- The JSPA is the foundation of the Java ecosystem.
- It defines the way in which IP is granted and the terms under which the spec, RI, and TCK must be licensed.
- The JSPA ensures:
  - that implementations of Java are compatible,
  - that the technologies we incorporate into Java are “safe” from an IP perspective (we know who owns them,)
  - that people can implement them and use them with confidence.



# Follow us on java.net



- As JSR 348 mandates, all our work is carried out in public.
- Start with our public java.net project.
- There you will find links to:
  - The Observer mailing list (all Expert Group mail is copied here.)
  - The Issue Tracker.
  - The Document Archive (meeting agendas and minutes, task lists, and working drafts are published here.)

# Results...



- JSR 348 introduced version 2.8 of the Process.
- JSR 355 merged the two ECs into one EC.
- It is now easier for JCP members and the public to observe and participate in the work of Expert Groups.
- Transparency is the default mode of operation.
- All new JSRs will be bound by the new rules.
  - Existing JSRs voluntarily adopt them—almost all Active JSRs have.
- Now we need your participation!

# JCP.Next JSR Progress Report



- JCP 2.8 went into effect in October 2011; JCP 2.9 went into effect November 2012.
- All JSRs started since then are run under this version of the Process.
- Spec-Leads of in-flight JSRs are encouraged to voluntarily adopt the new Process – almost all have.
  - On doing so they are required to operate transparently, and become subject to the new JSR deadlines.

# JCP.Next Scorecard



- There are 29 Active/In Progress JSRs.
  - <http://jcp.org/en/jsr/all?statusOnly=Active&activeMonths=12>
- 25 Active JSRs are operating under version 2.8 or above.
  - 1 Renewal Ballot (JSR 354, Money and Currency API).
- 10 JSRs have been initiated under JCP 2.8 (1 was rejected.)
- 19 JSRs have migrated to JCP 2.8.
- 1 JSR is in the process of migration to JCP 2.9.
  - Excludes those that have already posted a Final or Maintenance Release.
- The PMO encourages Spec Leads to migrate to current JCP version at JSR milestone postings.

# New JCP 2.8 JSRs



- JSR 352, Batch Applications for the Java Platform (IBM) – Proposed Final Draft, January 2013.
- JSR 353, Java API for JSON Processing (Oracle) – PR, January 2013.
- JSR 354, Money and Currency API (Credit Suisse) – JSR Review, February 2012.
- JSR 355, EC Merge (Oracle) – Final Release, August 2012.
- JSR 356, Java API for WebSocket (Oracle) – PR, January 2013.
- JSR 358, A major revision of the Java Community Process (Oracle) – JSR Review, July 2012.
- JSR 359, SIP Servlet 2.0 (Oracle) – JSR Review, July 2012.
- JSR 360, Connected Limited Device Configuration 8 (Oracle) – JSR Review, October 2012.
- JSR 361, Java ME Embedded Profile (Oracle) – JSR Review, October 2012.

# Migrated JCP 2.8/2.9 JSRs (1)



- JSR 236, Concurrency Utilities for Java EE (Oracle) – PR, February 2013; migrated March 2012.
- JSR 338, Java Persistence 2.1 (Oracle) – PR, January 2013.
- JSR 339, JAX-RS 2.0: The Java API for RESTful Web Services (Oracle) – PR, October 2012.
- JSR 340, Java Servlet 3.1 Specification (Oracle) – PR, February 2013.
- JSR 341, Expression Language 3.0 (Oracle) – PR, August 2012.
- JSR 343, Java Message Service 2.0 (Oracle) – PR, February 2013.
- JSR 344, JavaServer Faces 2.2 (Oracle) – PR, January 2013.
- JSR 345, Enterprise JavaBeans 3.2 (Oracle) – PR, February 2013.
- JSR 346, Contexts and Dependency Injection for Java EE 1.1 (RedHat) – PR, November 2012.
- JSR 349, Bean Validation 1.1 (RedHat) – PR, October 2012, JCP 2.9
- JSR 107, JCACHE - Java Temporary Caching API (G Luck, Oracle) – EDR, October 2012.

# Migrated JCP 2.8/2.9 JSRs (2)



- JSR 335, Lambda Expressions for the Java Programming Language (Oracle) – February, July 2013.
- JSR 337, Java SE 8 Release Contents (Oracle) – EG Formation, migrated September 2012.
- JSR 308, Annotations on Java Types (M. Ernst, Oracle) – EDR2, February 2012.
- JSR 310, Date and Time API (S. Colebourne, M. Santos, Oracle) – EDR, September 2012.
- JSR 347, Data Grids for the Java Platform (RedHat) – JSR Review, April 2011; migrated December 2012.
- JSR 350, Java State Management (Oracle) – JSR Review, Aug 2011; migrated December 2012.
- JSR 351, Java Identity API (Oracle) – JSR Review, September 2011; migrated January 2013.

# JSRs in the process of migrating



- JSR 342, Java Platform, Enterprise Edition 7 (Java EE 7) Specification (Oracle) – EDR2, November 2012.
- JSR 331, Constraint Programming API (J Feldman) – Maintenance Release, October 2012 (did not migrate/already Final).



# Transparency checklist for JSRs



- Is the schedule for the JSR publicly available, current, and updated regularly?
- Can the public read and/or write to a wiki for the JSR?
- Is there a publicly accessible discussion board for the JSR that you read and respond to regularly?
- Has the Spec Lead &/or Expert Group spoken at conferences and events about the JSR recently?
- Are open-source processes being used for the development of the RI and/or the TCK?
- What are the Terms of Use required to use the collaboration tools for the JSR development?
- Does the Community tab of the JSR have links to and information about all public communication mechanisms and sites for the development of the JSR?

# Get involved!



# How to participate?



- As an individual – OK.
- As part of a team – better.
- Work through your JUG.
  - Help each other.
  - Teach each other.
  - Work with each other.



# JUG members ~ 40 JUGs



**BeJUG (Belgium)**

**BreizhJUG (Brittany)**

**CeJUG (Brazil)**

**Central Ohio JUG (USA)**

**Connecticut JUG (USA)**

**Detroit JUG (USA)**

**Duchess (Women)**

**FASOJUG (Burkina Faso)**

**Green Tea JUG (China)**

**Houston JUG (USA)**

**iJUG e.V. (Germany)**

**IndiJava (India)**

**Japan JUG**

**Java Student User Group (Vienna)**

**Java Web User Group (London)**

**Jozi JUG (South Africa)**

**JUG-AFRICA**

**JUG Chennai (India)**

**JUG Cologne (Germany)**

**JUG-EG (Egypt)**

**JUGF (Germany)**

**JUG Indonesia**

**JUG-MK (Macedonia)**

**JUG-RU (Russia)**

**JUG-USA**

**London Java Community (UK)**

**Malaysia-JUG**

**MoroccoJUG**

**Nashville JUG (United States)**

**Oklahoma City JUG (USA)**

**Rio JUG (Brazil)**

**Riviera JUG (France)**

**Ruhrjug (Germany)**

**Silicon Valley JavaFX User Group (USA)**

**SouJava (Brazil)**

**Utah JUG (USA)**

**Vancouver Island JUG (Canada)**

# JUG representation on the EC



- **SouJava** (Bruno Souza, Fabio Velloso, Yara Senger):
  - Nominated by Oracle to a ratified seat on the SE/EE EC in the May 2011 Special Election.
- **London Java Community** (Ben Evans, Martijn Verburg, Trisha Gee):
  - Won an elected seat on the SE/EE EC in the May 2011 Special Election and re-elected in 2012.
- Both have already made significant contributions to the work of the JCP.

# Adopt a JSR!



Thank you **LJC** and **SOU Java**  
sociedade de usuários java

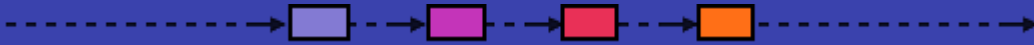
# What is it?



- A JUG-lead initiative to improve Java standards.
- Groups of JUG members work on JSRs.
- See <http://java.net/projects/adoptajsr> for full details.



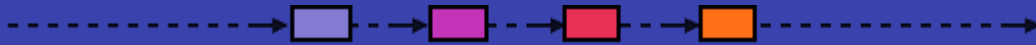
# Why should JUGs get involved?



- To move the entire Java ecosystem forward.
  - Help define the future of Java – best way to predict it.
- To speed up the standards-development process.
- To become part of the solution as opposed to staying silent.
- To be at the forefront of new technology.
- To ensure that Java standards are practical and relevant for the real world as opposed to the ivory tower.
  - We need expert input from end-users and developers.
- To ensure that Java standards are truly global.
  - We need expert input from non-English speakers.

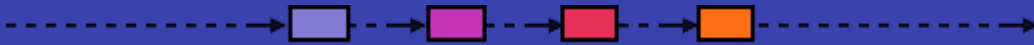


# How Will it Help your JUG?



- Receive attention from industry.
  - Big vendors, local companies see the value of all your efforts.
- Become more professional.
  - Participation requires organization and commitment—this helps promote focus in your group.
- Attract people.
  - Developers working together has a global impact, and reaps personal benefits.
- Collaborate with other JUGs.
  - Be part of something larger!

# How to get started on Adopt-a-JSR



- Join the JCP as a JUG.
  - Not required, but recommended. You need to do this in order to contribute code, and you get voting rights and other benefits with this membership!
- Join the Adopt-a-JSR project:
  - <http://java.net/projects/adoptajsr/pages/Home>
  - Subscribe to [members@adoptajsr.java.net](mailto:members@adoptajsr.java.net) (archive).
- Update the wiki to indicate your interest in a particular JSR and create a wiki page to provide updates on your plans/progress.
- Communicate directly with the Specification Lead/Expert Group (links on [jcp.org](http://jcp.org)) or the [java.net](http://java.net) users alias for the specification.
- Mail your JUG members to gather interest.
- Outline what the JSR is and why the JUG should get involved; identify areas that you want to work on; coordinate this with the Spec Lead.
- Use twitter hashtag [#adoptajsr](https://twitter.com/adoptajsr) to give updates on progress, [cc@jcp\\_org](mailto:cc@jcp_org).

# How JUGs can Join the JCP



- JUGs can join the JCP for free!
  - The \$2000 annual fee is waived for JUGs.
    - <http://jcp.org/en/participation/jug>
- You will get the benefits of JCP Membership, including support for your JUG meetings and gathering for JUGs.
  - Submit your own JSRs to the JCP, serve on JSR Expert Groups, participate in Executive Committee elections.
- JUGs should use the instructions for organizations:
  - <http://jcp.org/en/participation/membership2>

# Guidelines for Feedback



- The specification lead and the Expert Group has the final decision on incorporating the feedback.
- For specifications in Public Drafts, new feature requests may not be considered for this release. For specifications in Early Draft, new features may be accepted if they are within the existing scope. If new features are not accepted for this release, then they may be considered for the next release.
- Multiple JUGs can and should collaborate on a particular JSR. There is always plenty of work and going through the material multiple times only makes it that more better.
- When communicating with the EG on mailing lists, include "Adopt-a-JSR" in the subject line.
- When filing a JIRA issue, use "adoptajsr" tag.

# Suggestions for Feedback



- Download and review the latest version of the specification (available from the JSR page or the corresponding java.net project). You don't need to necessarily review the entire specification; can focus on specific sections.
- Download and review the javadocs for clarity and completeness.
- Build sample applications using the implementations.
- Explore integration of one/multiple technologies with your existing applications.
- File bugs on the corresponding spec or RI project.

# How can you help?



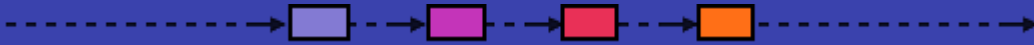
- Help to moderate the mailing lists; steer towards helpful conversations
- Evangelize the JSR through social media, blogging or lightning talks.
  - Give a talk at your user group or online.
- Help with documentation.
  - Translate into your native language.
- Help to maintain the FAQ/Wiki.
- Improve project infrastructure and JSR visibility.
  - Help set up JSR hosting (java.net, GitHub, etc.)
  - Help with Search Engine Optimization of website.
  - Ensure that downloads, mailing lists, issue trackers are easy to find.

# How else can you help?



- Organize Hack days.
- Test the early RI builds; use them to find pain points, report bugs, suggest feature enhancements.
- Help triage issues; reproduce issues, erase/merge duplicates, set priorities/categories etc.
- Give feedback on design; discuss issues and deliver feedback, think about how you would use the JSR as a developer.
- Help build the RI; get coding with the actual implementation of the spec.
- Help build the TCK; all implementations must pass the test suite – this is a great way to gain test experience.

# Adopt-a-JSR Transparency sub-project



- Transparency sub-project of the Adopt-a-JSR program hosted on java.net.
  - <http://www.java.net/form/jcp-2829-transparency?force=408>
- JUGs or individuals do transparency checks or audits on JSRs.
- Provide qualitative and quantitative data to the community, Spec Leads, and the Program Office on the JSR's compliance. with the JCP's transparency and participation requirements.
- A matrix of results will be provided for each JSR actively operating under JCP 2.8 and above.
  - <http://java.net/projects/jsr-transparency/pages/Home>
  - Once responses are received :-).

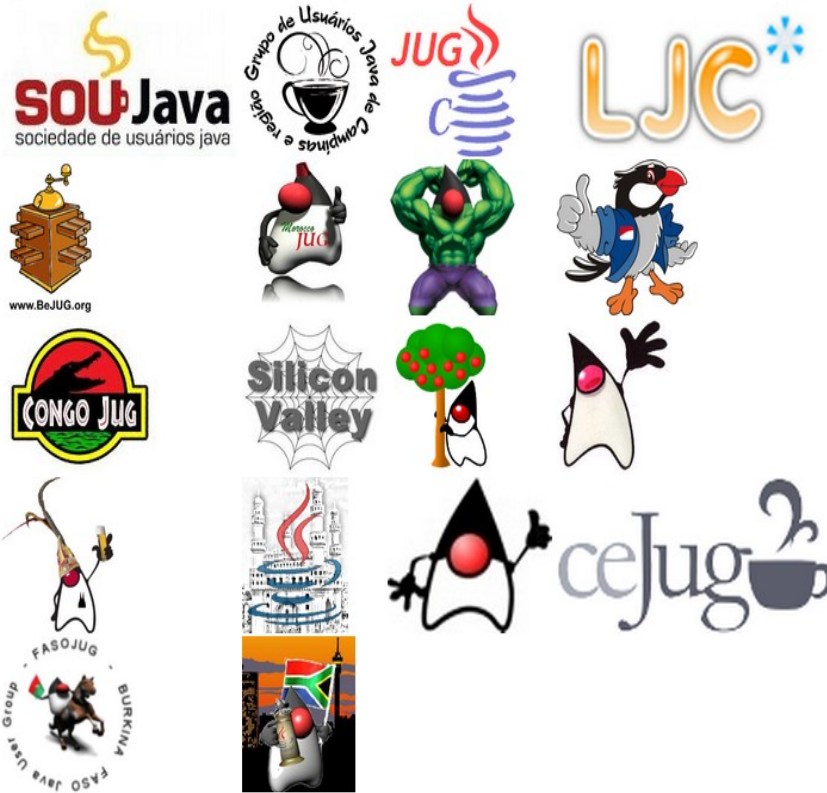


# Community-provided assessment



- The Transparency sub-project has online survey forms for your input to do a JSR check-up on your JSRs:
  - The quality and utility of the public issue tracker.
  - How the Spec Lead and Expert Group communicate and interact on the JSR's public mailing list. Are issues being resolved and addressed?
  - Whether the Expert Group represents all relevant sectors of the Java community,
  - Whether the Expert Group has published a draft of the JSR and whether this represents a convergence of competing implementations.

# JUGS Around the World!



# Adopt-a-JSR Adopters



- 20 JUGS participating: Campinas JUG (Brazil), ceJUG (Brazil), GOJava (Brazil), SouJava (Brazil), Peru JUG, SV JUG (USA), Houston JUG (USA), Toronto JUG (Canada), London Java Community (UK), BeJUG (Belgium), Madrid JUG (Spain), JUG Cologne (Germany), Indonesia JUG, Hyderabad JUG (India), JUG Chennai (India), Morocco JUG, Faso JUG (Africa), Jozi JUG (South Africa), Mbale (Uganda).
- 23 JSRs 'adopted': 107, 299, 308, 310, 321, 331, 335, 338, 339, 342, 343, 344, 345, 346, 347, 348, 349, 352, 353, 354, 355, 356, 358.
- JSR 339, Java API for RESTful Web Services 2.0 & JSR 353, Java API for JSON Processing, are the most popular JSRs, being adopted by at least 5 JUGs.

# Java EE 7 JSRs - specific requests



- Nine of the Java EE 7 JSRs have specific requests from Adopt-a-JSR participants.
  - <http://glassfish.java.net/adoptajsr/>
- JSR 236, Concurrency Utilities for Java EE.
- JSR 339, Java API for RESTful Web Services.
- JSR 343, Java Message Service 2.0.
- JSR 344, JavaServer Faces 2.2.
- JSR 345, Enterprise JavaBeans 3.2
- JSR 346, Contexts and Dependency Injection for Java EE 1.1.
- JSR 352, Batch Applications for the Java Platform.
- JSR 353, Java API for JSON Processing.
- JSR 356, Java API for WebSocket.

# Use Cases of Adopters



- Cologne JUG: JSR 344, JavaServer Faces (JSF) 2.2 - kick off event January 2013, more events planned.
- BeJUG: JSR 356, Java API for WebSocket - Hack day in March 2013/Tic-Tac-Toe game.
- Chennai JUG: JSR 338, Java Persistence 2.1, JSR 344, JSF 2.2, JSR 352, Batch, JSR 353, JSON, JSR 356, WebSocket - Urban Traveler application.
- Hyderabad JUG: JSR 338, JPA, and JSR 107, JCACHE, Food Court Management system.
- London Java Community: JSR 310, Date and Time API and JSR 335, Lambda Expressions (Martijn Verburg).
- SouJava: JSRs 310, 339, 344, and other JSRs:  
<http://soujava.org.br/servicos/adopt-a-jsr/>
- New! GitHub: <https://github.com/organizations/Adopt-a-JSR>

# Participate in OpenJDK



- Go to **<http://java.net/projects/adoptopenjdk/>**
- Join the mailing list.
- Join the IRC channel.
- Or contact your local JUG leader.

# OpenJDK

# Putting the *community* back into the JCP



- No more barriers to participation.
- All members of the Java community can participate:
  - In the implementation of the platform through OpenJDK and GlassFish.
  - In the evolution of the platform through the JCP.
- If you care about the future of Java...participate!

# Find out more...



- Adopt-a-JSR online meeting Wednesday, 27 February, 09:00 PST.
  - See details: <https://blogs.oracle.com/jcp/>
  - Recording from January 2013 meeting available.
    - <http://jcp.org/en/resources/multimedia>
- Follow the JCP online:
  - Twitter @jcp\_org
  - [blogs.oracle.com/jcp](https://blogs.oracle.com/jcp)
  - Facebook: Java Community Process



# Make the future Java!





*Questions?*

*Tack! Thank you!*

*Heather VanCura*

*<http://jcp.org>*

*[Blogs.oracle.com/jcp](http://blogs.oracle.com/jcp)*

*Twitter @JCP\_org*