JSR 375 (EE Security API) Review

October 23 2015

Alex Kosowski

Agenda

- Goals
- Information to be gathered
- Next steps
- Q & A

Goals

Goals

- Provide an intermediate update to the EC for JSR 375, EE Security API
- Focus more on how the EG is following the process
  - Meeting transparency and participation obligations

Information to be gathered

About this JSR

- What is the scope of this JSR?
  - Improve the Java EE Security API

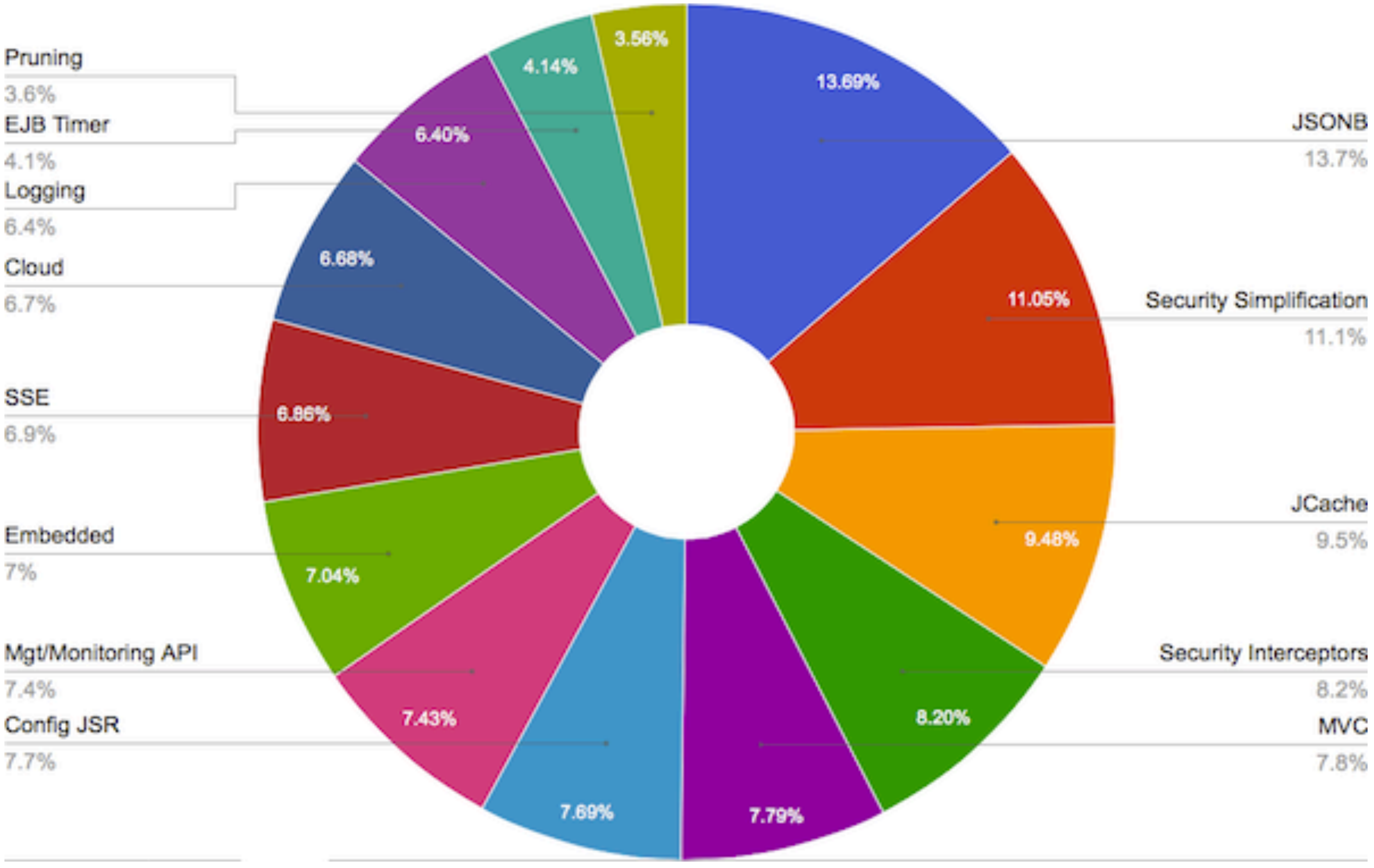
- What the JSR plans to achieve?
  - Improve portability by minimizing server-specific API usage
  - Modernize by using
    - Contexts and Dependency Injection (CDI)
    - Expression Language (EL)
    - Lambda Expressions
  - App Developer Friendly
    - Annotation defaults not requiring XML
    - Security configuration independent of server

Introduction

- First platform-wide EE Security API JSR ever
- Previous EE Security JSRs more subject-area focused
  - JSR 351, Java Identity API: renewal, May 2015:
  - JSR 196, Java Authentication Service Provider Interface for Containers (JASPIC): MR2 Start, June 2013
  - JSR 115, Java Authorization Contract for Containers (JACC): MR3 Start, June 2013
- Target platform is Java EE 8, included in platform

Business/marketing/ecosystem justification

- Why do this JSR? EE 8 survey, 4500 total responses



Business/marketing/ecosystem justification

- What's the need?
  - – Java EE Security viewed as not portable, abstract/confusing, antiquated
  - – Doesn’t fit cloud app developer paradigm: requires app server configuration
  - – Losing value to non-standard 3rd Party Frameworks…less likely to move back to Java EE
- How does it fit in to the Java ecosystem?
  - – Cross-cutting, affecting all EE containers
- Is the idea ready for standardization?
  - – Based on de facto standards widely used in 3rd party libraries

History

- List the significant dates in the history of the JSR.
  - August 2014: First proposed to Oracle Java EE Architects
  - December 2014: Approved by JCP
  - March 2015: Expert Group started discussions
  - Currently preparing for Early Draft Review

Technical scope and features

- Wide and thin approach: Low hanging fruit
- Incorporate CDI, Expression Language, Lambda Expressions
- Terminology
  - Clarify terms: caller/user, realm/id store, groups/roles
- API for Authentication Mechanism
  - Helper classes and annotations for simplifying JSR 196 JASPIC usage by app developers
  - CDI Events: PreAuthenticate, PostAuthenticate, PreLogout, etc
- API for Identity Store
  - Read-only identity store API, standard credential and persistence support, CDI based
  - JAAS adapter for potentially leveraging existing LoginModules

Technical scope and features

- API for Password Aliasing
  - Standard syntax/rules for password alias usage in annotations and deployment descriptors
- API for Role/Permission Assignment
  - Application portable group to role mapping
  - One-to-one group to role mapping
  - Application changeable role mapping
- API for Security Context
  - CDI bean for accessing Security Context
  - login(), logout(), runAs(), isAuthenticated(), isUserInRole()
  - For all managed beans: CDI, Servlet, EJB, JAX-RS, etc

Technical scope and features

- API for Authorization Interceptors
  - Expression Language Authorization Rules
  - AccessDecisionVoter

The Expert Group

- Adam Bien
- David Blevins (Tomitribe)
- Rudy De Busscher
- Ivar Grimstad
- Les Hazlewood (Stormpath)
- Will Hopkins (Oracle)
- Werner Keil
- Matt Konda (Jemurai)
- Alex Kosowski (Oracle)*
- Darran Lofthouse (RedHat)
- Jean-Louis Monteiro (Tomitribe)
- Ajay Reddy (IBM)
- Pedro Igor Silva (RedHat)
- Arjan Tijms (ZEEF)

* Spec lead

The Expert Group

- How does the EG operate?
  - On going conversations on expert mailing list
  - Comments in JIRA issues
  - Proposals and examples in GitHub and Google Docs
- What collaboration tools are used to facilitate EG communications?
  - java.net:
    - https://java.net/projects/javaee-security-spec
  - Google Doc Folder:
    - https://drive.google.com/drive/folders/0B6fBL__7IToLaXRyRnUzTXJPeEk
  - GitHub Organization:
    - https://github.com/javaee-security-spec

Other deliverables

- Other than Spec, RI, and TCK are you delivering, for example:
  - Additional documentation?
  - User's guide?
  - Sample code?
  - FAQ?
  - Other artifacts?
- No firm commitments
- Active EG of bloggers and contributors

Publicity

- Aquarium Blog
  - https://blogs.oracle.com/theaquarium/entry/java_ee_security_api_jsr
- Devoxx France 2015 – Spec lead presentation
  - https://www.parleys.com/tutorial/finally-security-api-jsr-375
- Devoxx UK 2015 – Spec lead presentation
  - https://www.parleys.com/tutorial/finally-security-api-jsr-375-1
- Java One 2015 Conference Session [CON3659]
  - "Finally, the Java EE Security API (JSR 375)"
- Java One 2015 BOF [BOF3666]
  - "How Would You Improve the Java EE Security API?"

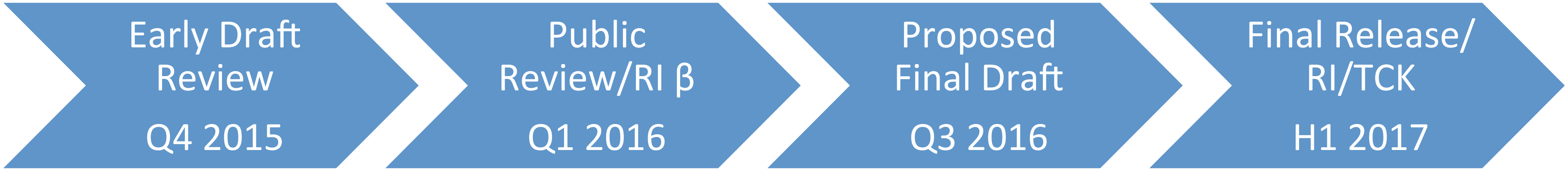
Collaboration with other community groups

- Are you working with other community groups or organizations?
  - No

Implementations

- How many implementations (apart from the RI) exist?
  - Git Hub EG Contributed Proposals
    - https://github.com/javaee-security-spec/javaee-security-proposals
  - Git Hub EG Contributed Examples
    - https://github.com/javaee-security-spec/javaee-security-examples

Schedule



Early Draft Review

Q4 2015

Public Review/RI β

Q1 2016

Proposed Final Draft

Q3 2016

Final Release/ RI/TCK

H1 2017

IP flow

- Provide pointers to the licenses for the the Spec, RI, and TCK.
  - https://jcp.org/en/jsr/detail?id=375
- How are you handling contributions from non JCP members?
  - Accepting contributions in our GitHub repos
  - Considered suggestions, to be developed by EG

IP flow

- What Terms of Use apply to your collaboration tools?
  - java.net
    - https://www.java.net/javanet-web-site-terms-use
  - GitHub
    - https://help.github.com/articles/github-terms-of-service
  - Google Docs
    - https://www.google.com/intl/en/policies/terms/
- Do you have a Contributor Agreement?
  - Yes, for EG members
  - No non-EG contributors
- Any legal issues or concerns?
  - No

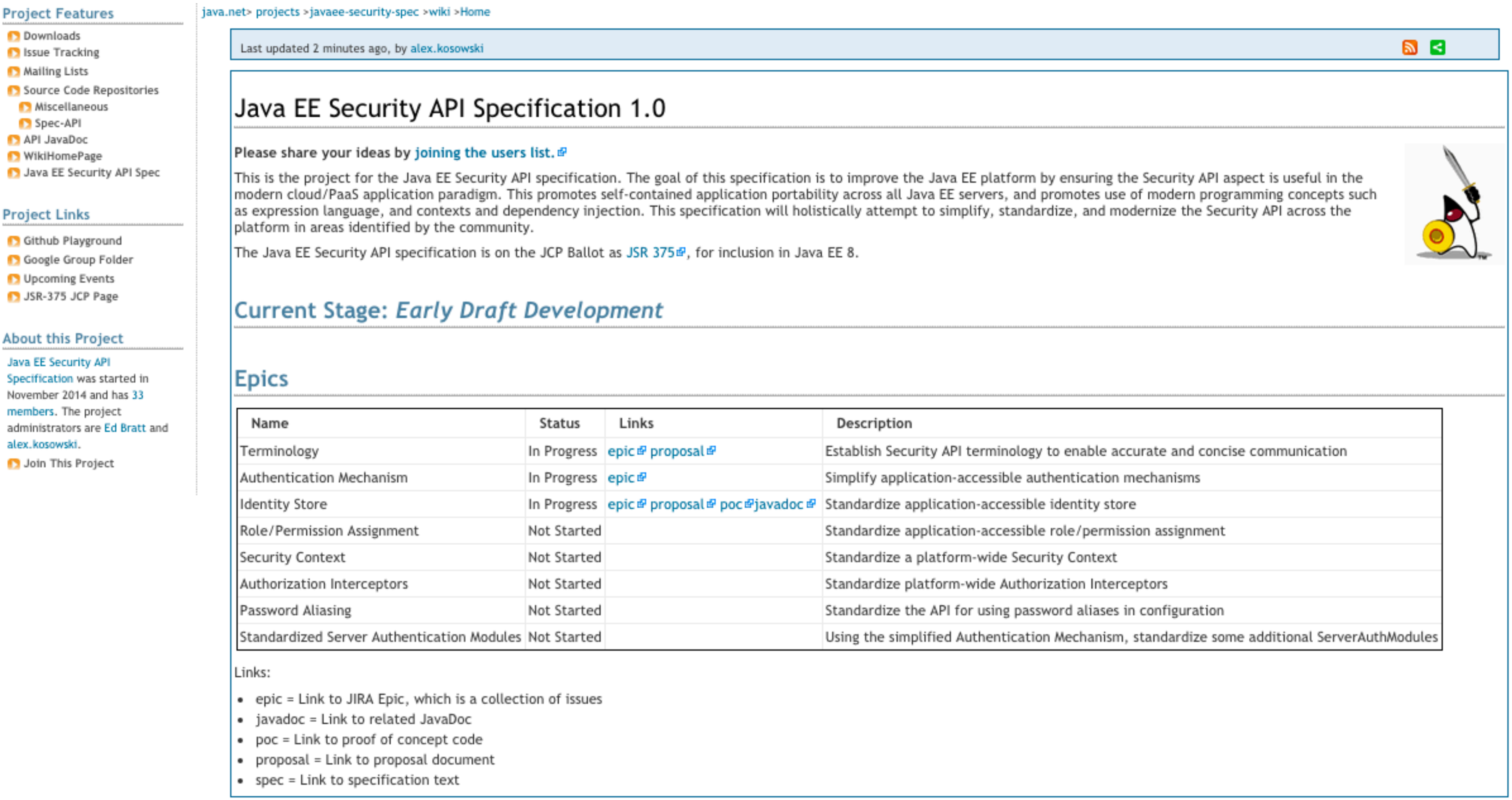
RI and TCK development

- How are you developing the RI and TCK?
  - RI: Glassfish
    - Spec Lead + 1 Oracle resource
    - EG committers
  - TCK
    - Spec Lead + 1 Oracle resource
- Do you have a source-code repository?
  - https://java.net/projects/javaee-security-spec/sources

Participation and transparency

- Provide a pointer to the JSR page on JCP.org
  - https://jcp.org/en/jsr/detail?id=375
- Provide a pointer to the “JSR project website"
  - https://java.net/projects/javaee-security-spec

Participation and transparency



Project Features

- Downloads
- Issue Tracking
- Mailing Lists
- Source Code Repositories
  - Miscellaneous
  - Spec-API
- API JavaDoc
- WikiHomePage
- Java EE Security API Spec

Project Links

- Github Playground
- Google Group Folder
- Upcoming Events
- JSR-375 JCP Page

About this Project

Java EE Security API Specification was started in November 2014 and has 33 members. The project administrators are Ed Bratt and alex.kosowski.

- Join This Project

java.net> projects >javaee-security-spec >wiki >Home

Last updated 2 minutes ago, by alex.kosowski

Java EE Security API Specification 1.0

Please share your ideas by joining the users list.

This is the project for the Java EE Security API specification. The goal of this specification is to improve the Java EE platform by ensuring the Security API aspect is useful in the modern cloud/PaaS application paradigm. This promotes self-contained application portability across all Java EE servers, and promotes use of modern programming concepts such as expression language, and contexts and dependency injection. This specification will holistically attempt to simplify, standardize, and modernize the Security API across the platform in areas identified by the community.

The Java EE Security API specification is on the JCP Ballot as JSR 375, for inclusion in Java EE 8.

Current Stage: *Early Draft Development*

Epics

Name	Status	Links	Description
Terminology	In Progress	epic proposal	Establish Security API terminology to enable accurate and concise communication
Authentication Mechanism	In Progress	epic	Simplify application-accessible authentication mechanisms
Identity Store	In Progress	epic proposal poc javadoc	Standardize application-accessible identity store
Role/Permission Assignment	Not Started		Standardize application-accessible role/permission assignment
Security Context	Not Started		Standardize a platform-wide Security Context
Authorization Interceptors	Not Started		Standardize platform-wide Authorization Interceptors
Password Aliasing	Not Started		Standardize the API for using password aliases in configuration
Standardized Server Authentication Modules	Not Started		Using the simplified Authentication Mechanism, standardize some additional ServerAuthModules

Links:

- epic = Link to JIRA Epic, which is a collection of issues
- javadoc = Link to related JavaDoc
- poc = Link to proof of concept code
- proposal = Link to proposal document
- spec = Link to specification text

Java Community Process

Adopt-a-JSR

- Are you participating in the Adopt-a-JSR program?
  - Not presently
  - Plan to present to Princeton JUG, Philadelphia JUG
  - Promote adopting JSR 375

Mailing lists or forums

- How are you communicating with the public and how can they communicate with you?
  - To the public: Expert and Users mailing list, JIRA issues
  - From the public: Users mailing list, JIRA issues
- Provide pointers to public mailing list(s) and/or forum(s)
  - https://java.net/projects/javaee-security-spec/lists
- Total number of messages?
  - [10/19/2015] Expert=544, User=547
- Total number of participants?
  - [10/19/2015] Expert=16, User=37
- How many messages per month?
  - [10/19/2015] Average 68

Issue tracker

- Total number of issues?
  - 28 Issues + 3 Epics
- How many in each state?
  - 22 Open, 4 In Progress, 2 Resolved
- Average number of issues logged per month? 3.5
- How many different people logged them? 11
- How does this break down between Spec Lead, EG members, and non-EG members?
  - Spec Lead: 6 (including Epics)
  - EG member: 16
  - Non EG member: 9

Document archive

- Provide a pointer to your document archive.
  - https://java.net/projects/javaee-security-spec/downloads
- Are meeting minutes and materials published?
  - Automatically via mailing lists
- What other materials are available for download?
  - Presentation slides, proposals, code samples
- "Working documents" in a shared Google folder:
  - https://drive.google.com/drive/folders/0B6fBL__7IToLaXRyRnUzTXJPeEk
- Code proposals/examples in GitHub Organization:
  - https://github.com/javaee-security-spec

Next steps

Next steps

- Complete the Early Draft Review
- Meet with local JUGs to see if they want to adopt JSR 375
- Continue to promote and socialize the JSR

Q & A

Thank you!

http://jcp.org