



JavaOne™



**MOVING JAVA
FORWARD**

ORACLE

The Java Identity API (JSR 351)

Ron Monzillo

THE FOLLOWING IS INTENDED TO OUTLINE OUR GENERAL PRODUCT DIRECTION. IT IS INTENDED FOR INFORMATION PURPOSES ONLY, AND MAY NOT BE INCORPORATED INTO ANY CONTRACT. IT IS NOT A COMMITMENT TO DELIVER ANY MATERIAL, CODE, OR FUNCTIONALITY, AND SHOULD NOT BE RELIED UPON IN MAKING PURCHASING DECISION. THE DEVELOPMENT, RELEASE, AND TIMING OF ANY FEATURES OR FUNCTIONALITY DESCRIBED FOR ORACLE'S PRODUCTS REMAINS AT THE SOLE DISCRETION OF ORACLE.

Ron Monzillo: ron.monzillo@oracle.com

Consulting Member of Technical Staff, Oracle Identity Management

Joined Sun Microsystems in March 1999

Java EE Platform and Servlet Security Architect

Specification Lead:

JSR 351 The Java Identity API

JSR 196 The Java Authentication SPI for Containers

JSR 115 The Java Authorization Contract for Containers

OASIS WS-Security contributor and editor of SAML Token profile

Agenda

- Charter, Supporters, Transparency, Open Source, and Next Steps
- Problem Statement
- Proposal
- Sample Use Cases
- Summary
- Q & A

Charter

To define application programming interfaces and identity interaction models that facilitate and control the use of identity by applications and in access control decisions.












JSR #351

Java™ Identity API JSR Review Ballot

These are the final results of the JSR Review Ballot for JSR #351. The Executive Committee for SE/EE has approved this ballot.

Votes

SE/EE

Credit Suisse 	Eclipse Foundation, Inc 	Ericsson AB 	Fujitsu Limited <input type="checkbox"/>
Goldman Sachs & Co. <input type="checkbox"/>	Google Inc. <input type="checkbox"/>	Hewlett-Packard 	IBM 
Intel Corp. 	Keil, Werner 	London Java Community 	Oracle 
RedHat 	SAP AG 	SouJava <input type="checkbox"/>	VMWare <input type="checkbox"/>

Icon Legend

Yes 

No 

Abstain

Not voted

Initial Expert Group and Supporters

Initial Expert Group*:

IBM

Oracle

RedHat

SAP AG

Please volunteer if you think you can help!

<http://jcp.org/en/jsr/egnom?id=351>

Supporting this JSR

American Express

Aspect Security

Boeing

Ericsson AB

OWASP

Transparency

- The expert group members will be identified on the JSR project page and in every draft of the specification.
- The Expert Group email list will be publicly readable via an email archive, and via observer subscription.
- A public email list will be available for comments, as will an issue tracker.
- The Reference Implementation will be hosted at java.net as an open source project.
- The schedule will be kept current on the JSR project page

Licensing

Reference Implementation will be developed as Open Source project within java.net, under Apache License, version 2.0
<http://www.apache.org/licenses/LICENSE-2.0>

License for Final Release of Specification

<http://jcp.org/aboutJava/communityprocess/licenses/jsr351/351SpecLicenseIdentityJSR.pdf>

The TCK will be licensed at no charge

<http://jcp.org/aboutJava/communityprocess/licenses/jsr351/351SATCKJSRIdentity1.pdf>

Schedule and Applicability

Proposed Schedule:

Expert Group formed: October 2011

Early Draft: March 2012

Public Review: July 2012

Final Release: January 2013

Target platform:

This specification is targeted for compatibility with Java SE and Java EE platforms beginning with Version 6.0. We also expect the output of this project to be relevant to Java applications running in consumer appliances.

Agenda

- Charter, Supporters, Transparency, Open Source, and Next Steps
- Problem Statement
- Proposal
- Sample Use Cases
- Summary
- Q & A

The advent of social networking sites, the adoption of single-sign-on and identity federation services, and more generally, the increased use of the internet in conducting business, have combined to amplify the need for Java developers to be able to appropriately consume, produce, and safeguard the disclosure of network identity.

Terminology – What is Identity?

Identity, Privacy, and Trust

- *Identity attributes* are properties of a digital subject
- A *digital subject* is a digital representation of an *entity* that is an actor or target of a digital operation.
- *Privacy* is the degree to which the availability of an entity's identity attributes can be controlled
- *Trust* is an evaluation of the reliability of a representation of a digital subject
- An *identity* is a set of identity attributes that distinguish an entity

How is Identity Used?

- Rapidly expanding class of applications and of information
- Business objects that reference or represent “Entities”
 - Patient Record, HR, Frequent Flyer, etc.
- May be used to “personalize” a provided service
- Use of Identity in Security systems
 - consumed in access control decisions and in authentication
 - Authentication systems may serve as source of identity attributes (of active entity) as by-product of authentication or federation interactions

Where does Identity Come From?

- Diverse sources and consumers impose new requirements on representation and handling of identity
- Identity data created by many distinct authorities
 - Self-asserted, governments, employer, social networks, online portals, corporations,...
 - No single-source completely describes a subjects identity
- Moved beyond (IT) directory-centric model
- regulatory and privacy requirements

How is Identity Represented in Java?

The lack of adequate interfaces in the Java platform is forcing application, system, and identity framework developers to rely on non-standard interfaces, which is resulting in indirect, inconsistent, poorly integrated, and inferior support for network identity.

impeding both interoperability and portability

Agenda

- Charter, Supporters, Transparency, Open Source, and Next Steps
- Problem Statement
- Proposal
- Summary
- Q&A
- References and Pointers

The next sections detail a proposed architecture and design for the API to be delivered. They are offered as an initial proposal. The output of JSR 351 will be an architecture and API that embodies the decisions and contributions of the Expert group.

What's the Proposal?

- Standardize the Representation
- Promote Attribute Service
- Standardize a Declarative Programming Style

Standardize Representation of Identity in Java

- Standard Attribute Interfaces
 - Attributes named, multi-valued, and meta-data qualified
 - meta-data: Issuer*, time-of-issue, validity period, usage-constraints
- Uniform, domain model independent, framework for representing Identity Attributes
 - Including content from mechanism specific security credentials
- Represents Identity in form that is compatible with its use within the interfaces of the Java Security Model
- Represents Identity such that it can be propagated between Java systems

Promote Attribute Service

As Point of Interaction, Governance, and Virtualization

- Local point of reference for applications
 - Encapsulation of diverse repository protocols and locations
 - Able to optimize interactions with remote repositories
- Authoritative representation of source, validity, and related meta-data
- Authorization and Audit of application use of identity attributes
 - in support of compliance with Identity governance model

Standardize Declarative Programming Style

for Identity in Java

- Client-side Java framework for consumption, generation, propagation, and governance of identity attributes
- Declaration of use, virtualization of source
- familiar to Java developers
- Dependency Injection replaces lookup
- Annotations declare dependencies
- `AccessControlContext` represents actors

What's NOT in the proposal

- Standardization of a fixed set of identity attributes (i.e., a specific domain model) that Java developers should use!
- That remains the responsibility of specific communities (e.g., citizens, finance, education, medicine) or application architects (e.g., CRM, HR,...)
- This JSR will:
 - provide a domain model independent Java framework for representing and interacting with identity attributes.

Interface architecture

- Layer 1: Representation and JRE Integration
- Layer 2: Services
- Layer 3: Application Development

Layer 1: Representation and JRE integration

- Attribute interfaces to represent identity attributes and associated meta-data.
 - Common representation of existing Java Object types as attributes
 - Reference and represent content acquired from the attribute service
 - Represent mechanism security credentials
 - Represent relationships among attributes and collections of attributes
 - convey identity attributes within the Java access control context
- Extensible meta-data attachment mechanism
 - Issuer, validity period, usage constraints, ...

Identity Attribute Interface

```
public interface IdentityAttribute {  
    Collection getAttributeNames();  
    Collection getAttributeValues();  
    Collection<String> getPropertyNames();  
    Collection getPropertyValues(String propertyName);  
}
```

named, multi-valued, and meta-data qualified

Layer 2: Services

- Attribute Service
 - Agent that affords application access to attributes in one or more potentially distributed attribute repositories
 - supports the integration of applications as attribute providers accessible within this service framework.
- Attribute-based Policy subsystem
 - Able to process policies that are contingent on identity attributes, acquired during processing of policy
- multi-tenancy will factor in definition of services

Service Provider Architecture

- Attribute Service composed of attribute providers
 - specification will define contracts to facilitate repository integrations by third parties.
 - Reference Implementation will provide some specific integrations including those provided by contributors to Open Source project
 - FaceBook, Twitter, and LinkedIn will be among the identity repositories considered for integration within the reference integration, as will their associated programming interfaces and protocols, including FaceBook Connect, OpenID Connect, and OAUTH 2.0.

Layer 3: Application Development

- define annotations to:
 - Inject identity attributes and or references to identity attributes
 - Export application attributes to the attribute service.
 - Insert access control enforcement points within applications.
- reuse or extend existing standard annotations
- propose and advocate for improvements in the existing Java access control interfaces to facilitate more efficient enforcement of user-centric access control decisions.

Attribute Injected

```
import javax.security.identity.IdentityAttribute;  
import javax.security.identity.annotation.Attribute;  
public class InjectAttribute {  
    @Attribute(name ="attributeName")  
    IdentityAttribute theAttribute;  
}
```

Attribute Exported

```
import javax.security.identity.IdentityAttribute;  
import javax.security.identity.annotation.AttributeProvider;  
public class ExportAttribute {  
    @AttributeProvider(name = "attributeName")  
    IdentityAttribute theAttribute;  
}
```


Attribute Exported

for Method Allowed Check

```
import javax.security.identity.IdentityAttribute;
import javax.security.identity.annotation.AttributeProvider;
import javax.security.identity.annotation.MethodAllowed;
public class TheClass {
    @AttributeProvider(name = "attributeName")
    IdentityAttribute theAttribute;
    @MethodAllowed
    public String theMethod(String theArgument) {...}
}
```

Able to provide contextual attributes to authorization system (on demand)

Agenda

- Charter, Supporters, Transparency, Open Source, and Next Steps
- Problem Statement
- Proposal
- Sample Use Cases
- Summary
- Q & A

Sample Use Cases

1. Application is Client of Attribute Service
2. Application is Attribute Provider
3. Identity Propagation
4. Authentication System Binds Attributes to Java Authentication State
5. Authorization System consumes Attributes

Sample use Cases

Comments on Intent

- Were used to develop the JSR proposal.
- Described in more detail and by example in the proposal
- Not to preclude the consideration of additional use cases
- Not to represent intent to standardize or limit applicability to a specific domain model

Sample Use Cases

Some Conclusions

- Means to identify the entity or entities who are the target of an attribute interaction
 - Implicit (e.g., the caller principal) or by specific identification
- Need to establish injection and attribute provider contexts.
 - E.g., application, request, session, acc, thread, etc.
- Need ability to represent attributes by reference
 - To sustain attribute protection during identity propagation
- Need policy constructs to indicate attribute dependencies
 - To be fulfilled outside of Access Control Context



Agenda

- Charter, Supporters, Transparency, Open Source, and Next Steps
- Problem Statement
- Proposal
- Sample Use Cases
- Summary
- Q & A

Summary (JCP)

- JSR 351 Approval Ballot completed midnight Oct 3, PST/PSD
- JSR Chartered
- Expert Group forming
- Project being bootstrapped
- Reference Implementation will be hosted at java.net as open source project

Summary (Proposal)

- Client-side Java framework for consumption, generation, propagation, and governance of identity attributes
 - Distributed identity repositories and disparate domain models
- Declaration of use, virtualization of source
 - Annotations declare dependencies
 - Dependency Injection replaces lookup
- Intended for use in Java environments that support CDI
 - Including Java EE
 - Intent to support applications running in consumer appliances.

More Information

- <http://jcp.org/en/jsr/summary?id=351>
- How to volunteer for the Expert Group
 - <http://jcp.org/en/jsr/egnom?id=351>
- (coming soon) <http://java.net/projects/>
- Ron.monzillo@oracle.com

Q&A

