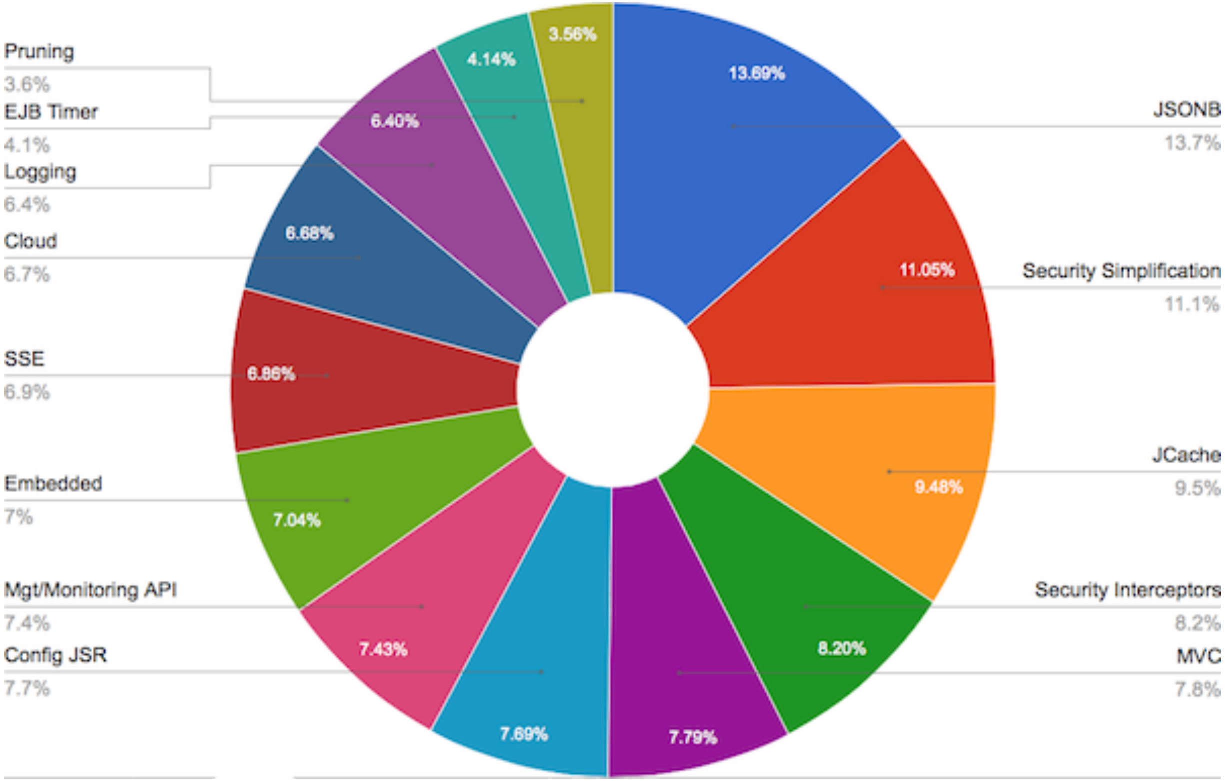
JSR 375 update

June 15, 2015

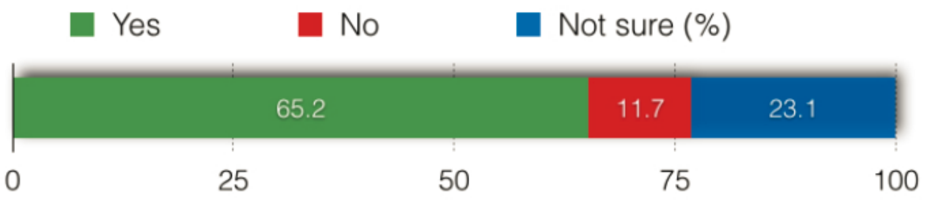
Werner Keil, Jean-Louis Monteiro

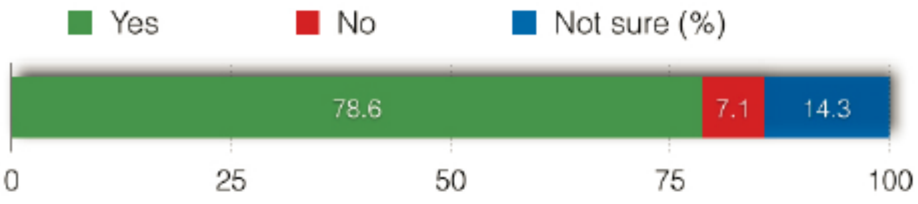
Why a Java EE Security JSR?

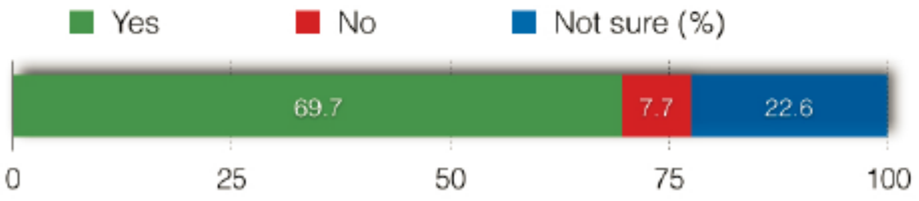
- Java EE 8 survey results

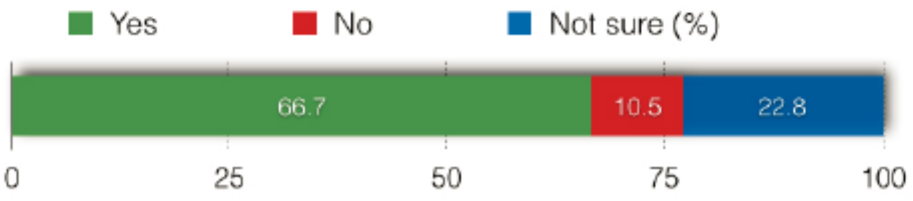


Survey Results


Should we standardize on requirements for simple security providers and their configuration?

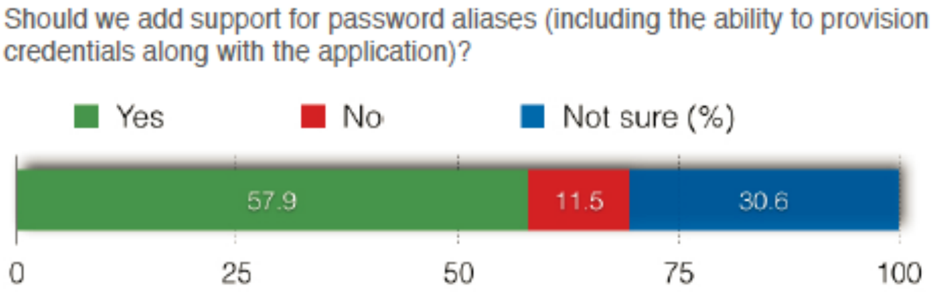

Should we consider adding Security Interceptors in Java EE 8?


Should we standardize group-to-role mapping?


Should we simplify authorization by introducing an EL-enabled authorization annotation?

Survey Results – Password Aliasing

API for Password Aliasing



Should we add support for password aliases (including the ability to provision credentials along with the application)?

- Deferred from Java EE 7

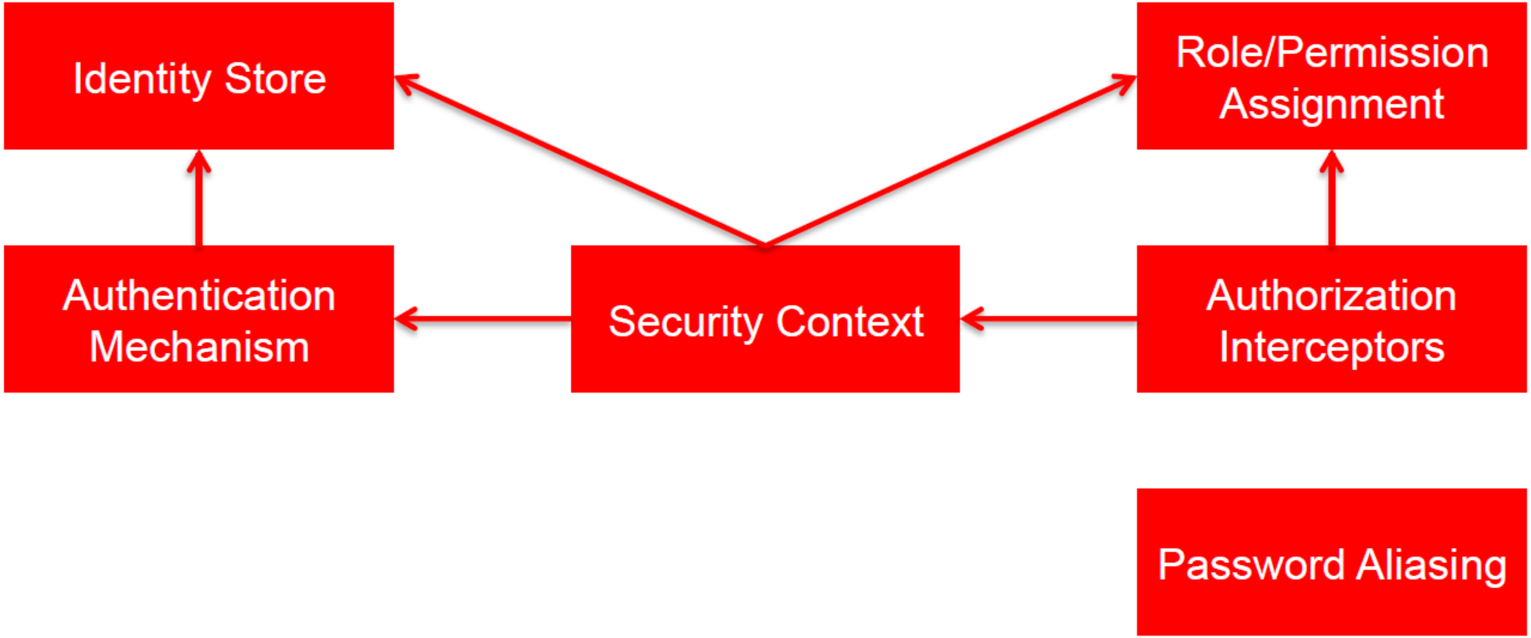
Planning – Epics

Name	Status	Description
Terminology	In Progress	Establish Security API terminology to enable accurate and concise communication
Authentication Mechanism	In Progress	Simplify application-accessible authentication mechanisms
Identity Store	In Progress	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

See: https://java.net/projects/javaee-security-spec/pages/Home

Ideas

To modernize, standardize, simplify



Identity Store

Role/Permission Assignment

Authentication Mechanism

Security Context

Authorization Interceptors

Password Aliasing

Ideas – Standardized Authenticators

- OpenID Connect ServerAuthModule

```
@Authenticator("javax.security.authenticator.OpenIDConnect")
@WebServlet("/SimpleServlet")
@ServletSecurity(@HttpConstraint(rolesAllowed = {"manager"}))
public class SimpleServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request,
      HttpServletResponse response)
         throws ServletException, IOException {
             response.getWriter().print("my GET");
    }
}
```

Ideas – Authentication Events

- Throw standardized CDI events at important moments
  - PreAuthenticate Event
  - PostAuthenticate Event
  - PreLogout Event
  - PostLogout Event

Ideas – Authentication Event Usage

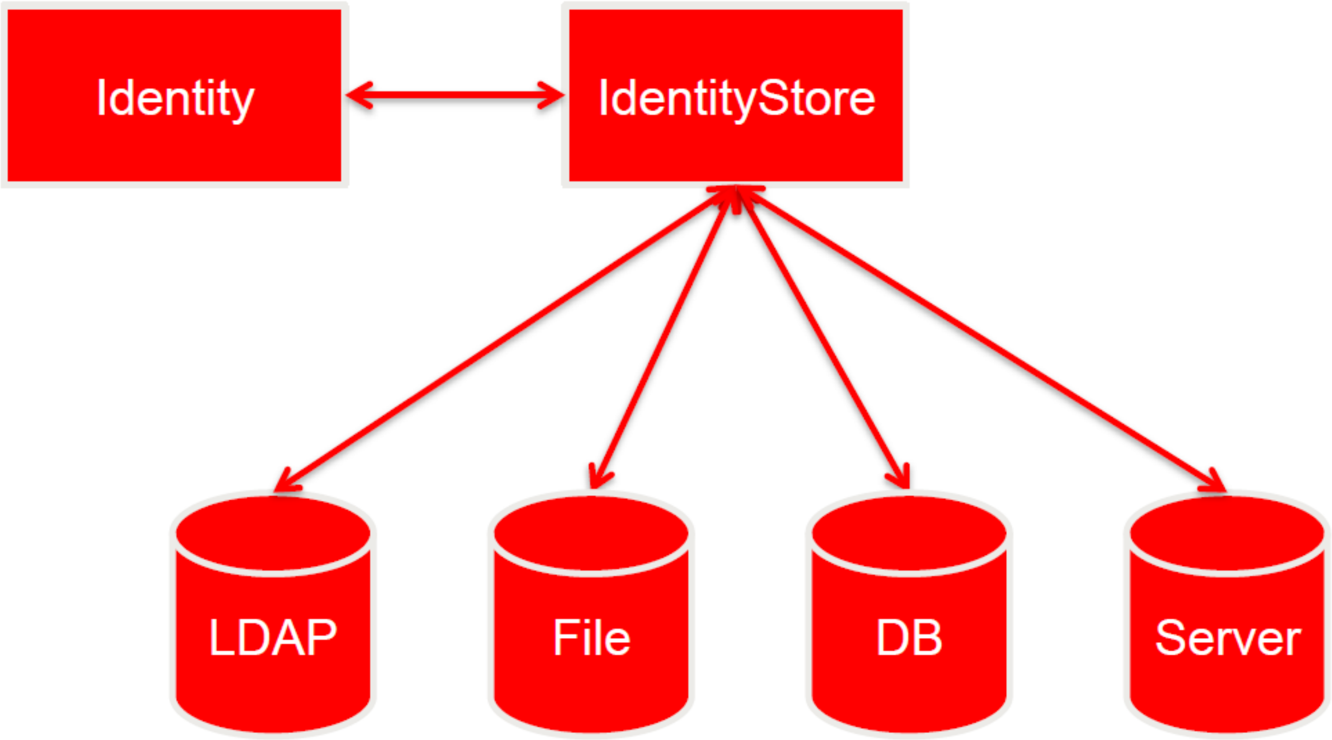
- Possible Use Cases
  - Tracking number of logged-in users
  - Tracking failed login attempts per account
  - Side effects, like creating a new local user after initial successful authentication via a remote authentication provider
  - Loading application-specific user preferences

    ...

Ideas – Identity Management

- Application manages its own users and groups
- Need to access a repository of identities, like users
- Users may be stored in app-specified repository (e.g. LDAP)
- Users are managed without access to server configuration

Ideas – Identity Management

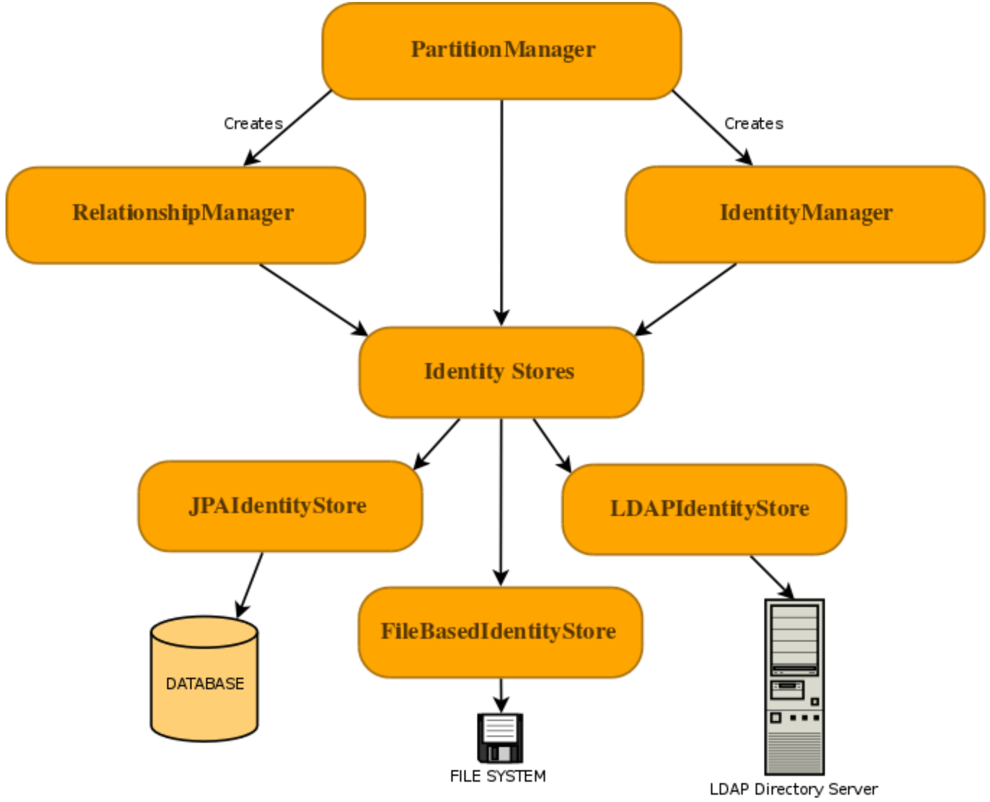


Identity Management

Current Solutions

- No Java EE support
- Only proprietary server support
- 3rd party security frameworks provide user/group APIs

IDM – Components

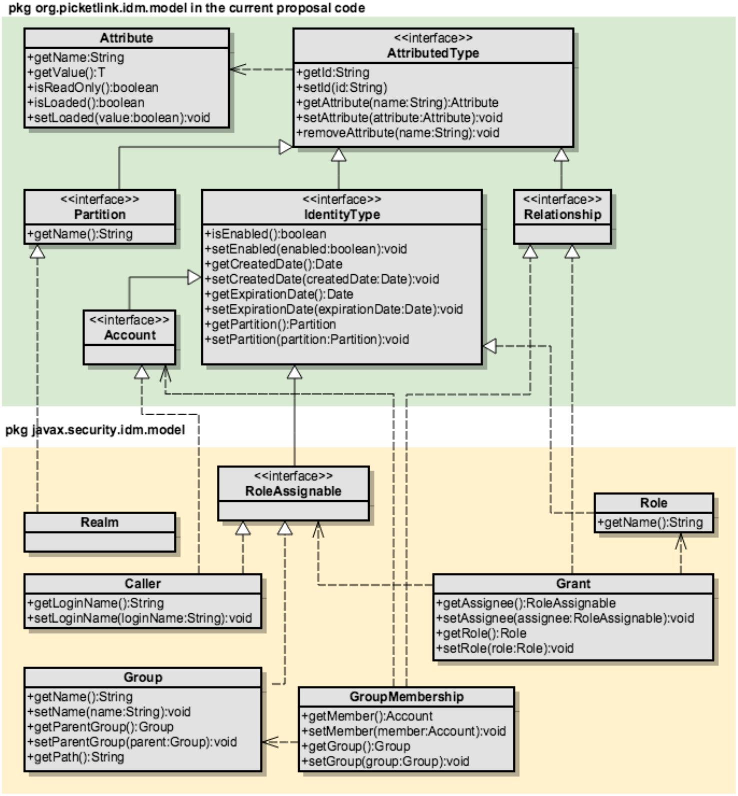


IDM – CDI Injection

Identity Management components should be injectable via CDI.

Component	**Scope**
Partition Manager	@ApplicationScoped
Identity Manager	@RequestScoped
Relationship Manager	@RequestScoped

IDM – Java EE Identity Model



pkg org.picketlink.idm.model in the current proposal code

Attribute
+getName:String
+getValue():T
+isReadOnly():boolean
+isLoaded():boolean
+setLoaded(value:boolean):void

<<interface>>
AttributedType
+getId:String
+setId(id:String)
+getAttribute(name:String):Attribute
+setAttribute(attribute:Attribute):void
+removeAttribute(name:String):void

<<interface>>
Partition
+getName():String

<<interface>>
IdentityType
+isEnabled():boolean
+setEnabled(enabled:boolean):void
+getCreatedDate():Date
+setCreatedDate(createdDate:Date):void
+getExpirationDate():Date
+setExpirationDate(expirationDate:Date):void
+getPartition():Partition
+setPartition(partition:Partition):void

<<interface>>
Relationship

<<interface>>
Account

pkg javax.security.idm.model

<<interface>>
RoleAssignable

Realm

Role
+getName():String

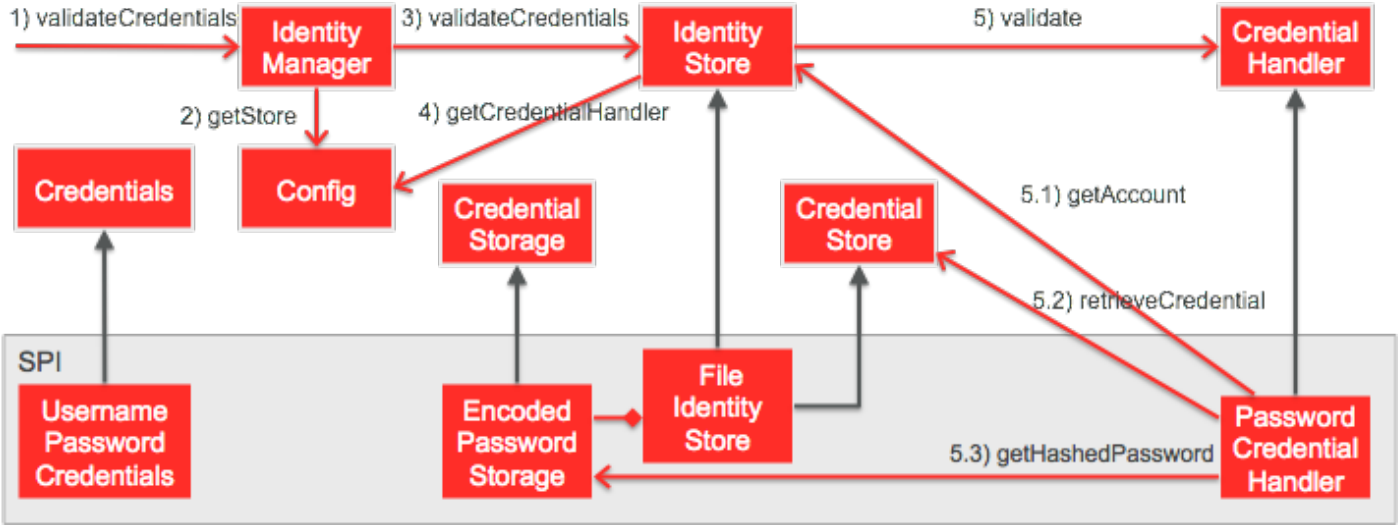
Caller
+getLoginName():String
+setLoginName(loginName:String):void

Grant
+getAssignee():RoleAssignable
+setAssignee(assignee:RoleAssignable):void
+getRole():Role
+setRole(role:Role):void

Group
+getName():String
+setName(name:String):void
+getParentGroup():Group
+setParentGroup(parent:Group):void
+getPath():String

GroupMembership
+getMember():Account
+setMember(member:Account):void
+getGroup():Group
+setGroup(group:Group):void

IDM – Credential Validation



Links

- JSR Page: https://jcp.org/en/jsr/detail?id=375
- Java.net Project Page: https://java.net/projects/javaee-security-spec
- Users Mailing List: users@javaee-security-spec.java.net
- Github Playground: https://github.com/javaee-security-