JAAS

Java Authentication and Authorization Service

AboutCoté

- Developer for 10+ years: FundsXpress, Coral, Cobalt Group, BMC Software, Inc.
- Now an Industry Analyst at RedMonk
- http://www.PeopleOverProcess.com
- http://www.DrunkAndRetired.com
- cote@redmonk.com

JAAS!

- Makes you hate Java, but realize why you love it.
- Authentication is great, authorization too weird for my tastes.

Meta-Bullets

> I say "jazz," some say "J-A-A-S."

> Please interrupt me all you want. Linear thinking is boring.

> Took me a long time to connect all the dots. Good luck with 45 minutes! ;)

> This presentation at:

> http://www.jaasbook.com/presentation/

JAAS in Action

- > Wrote for Manning in 2004-2005.
- > With help from Chip Holden (mrchippy.blogspot.com).
- > No one would pay $50 for it.
- > Never published.
- > Entire Book, for free: www.JaasBook.com.
- > If you'd like to work on it, have at it!

Definitions

Two words that begin with “auth”

Authentication

- Verifying the identity of users.
- “Who is this person accessing my system?”
- Subjects.
- Principals.
- Interaction with 3rd party identity stores.

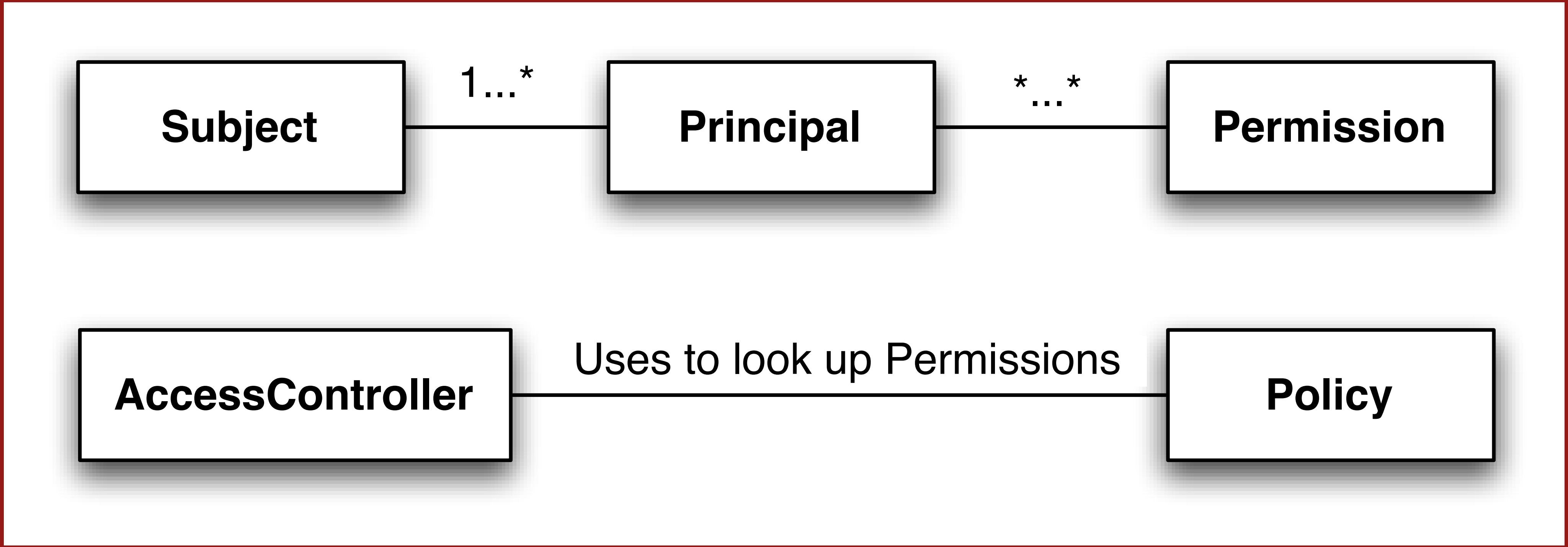
Authorization

> The process of enforcing access restrictions on authenticated users.

> "Can the Subject access the document?"

> JAAS has Principals, like groups.

Classes

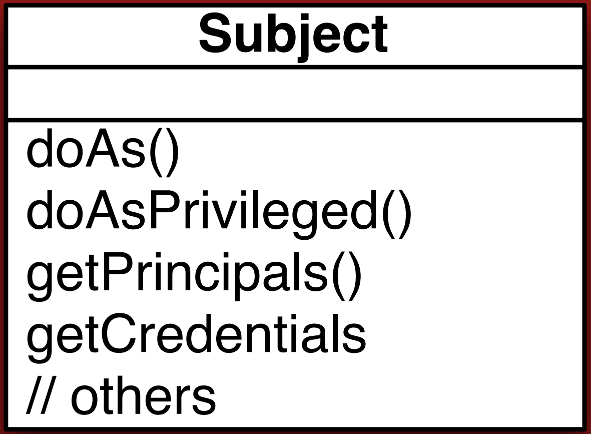

Subject
1...*
Principal
...
Permission
AccessController
Uses to look up Permissions
Policy

Authentication

"Who are you?"

Authentication: Subject

- Much like a user.
- But more like a wallet.
- Doesn't have to be a human.
- Has Principals and Credentials.
- javax.security.auth.Subject



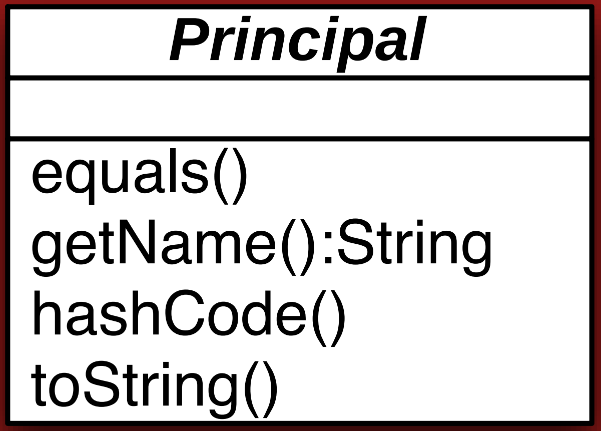
Authentication: Principal

> Different identities or groups that Subject is a member of.

> Examples: Admin, Employee, Manager, SSN, Preferred Customer.

> Any grouping or identifier.

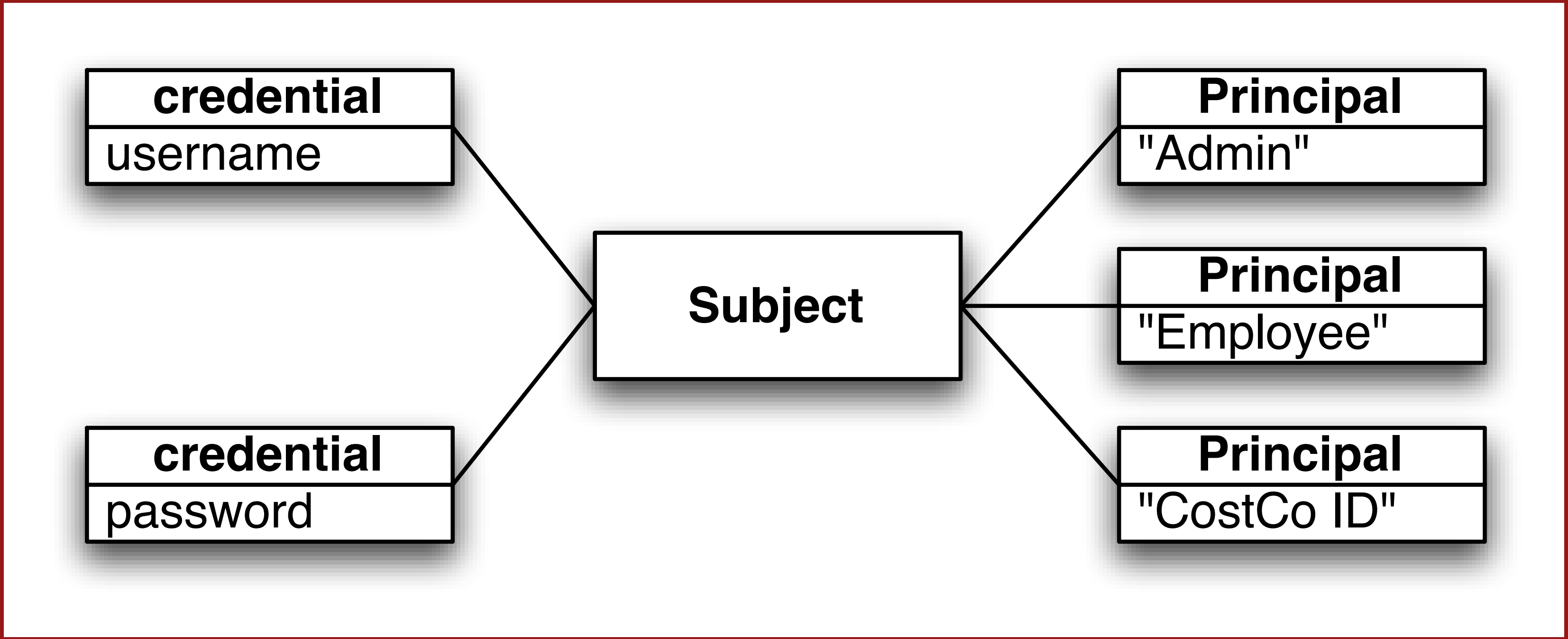
> java.security.Principal.



Authentication: Credential

> Opaque objects. Can be anything.
> Examples: username, password, certificates.
> Destroyable & Refreshable interfaces.

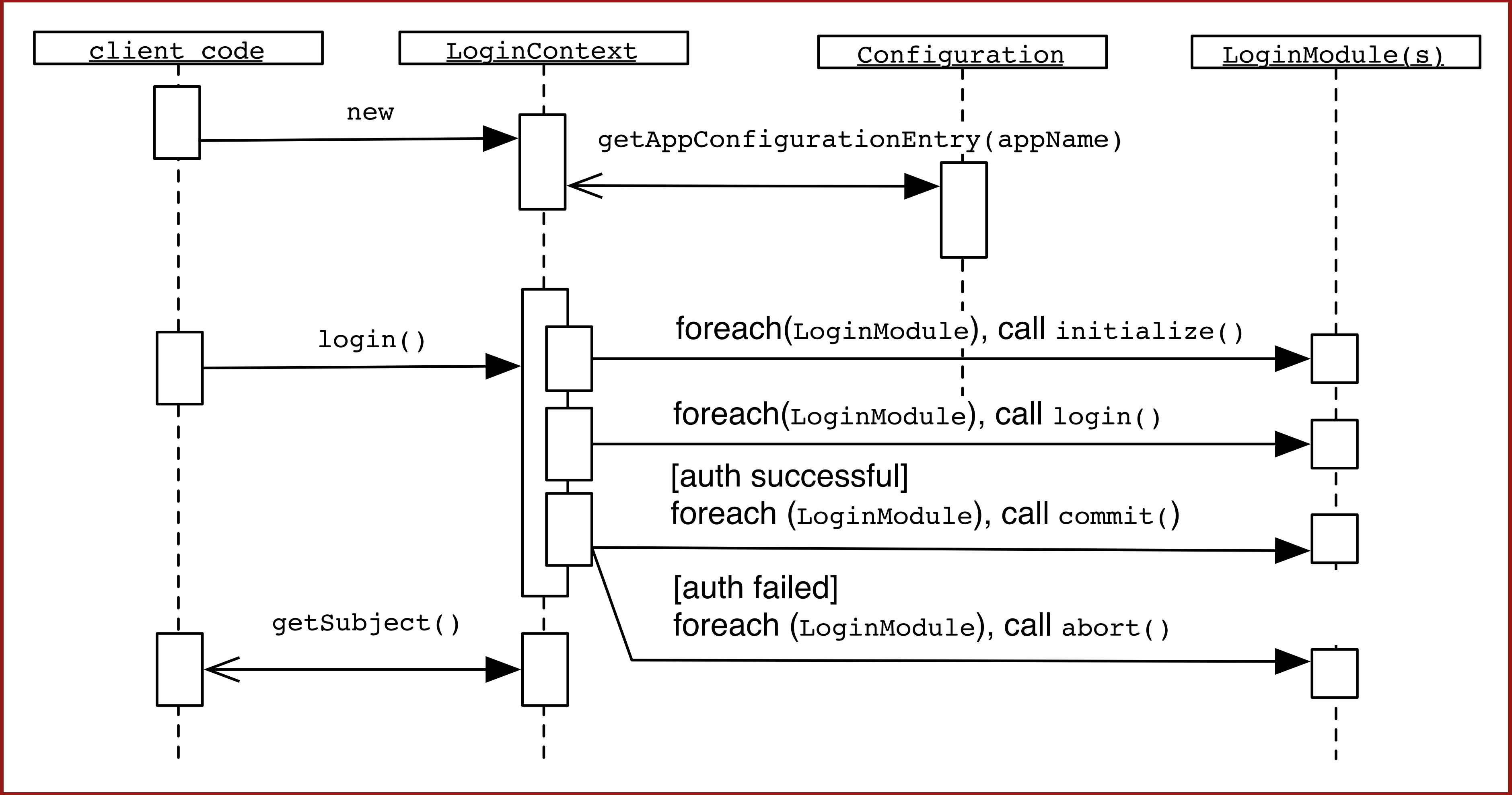
Authentication: Classes



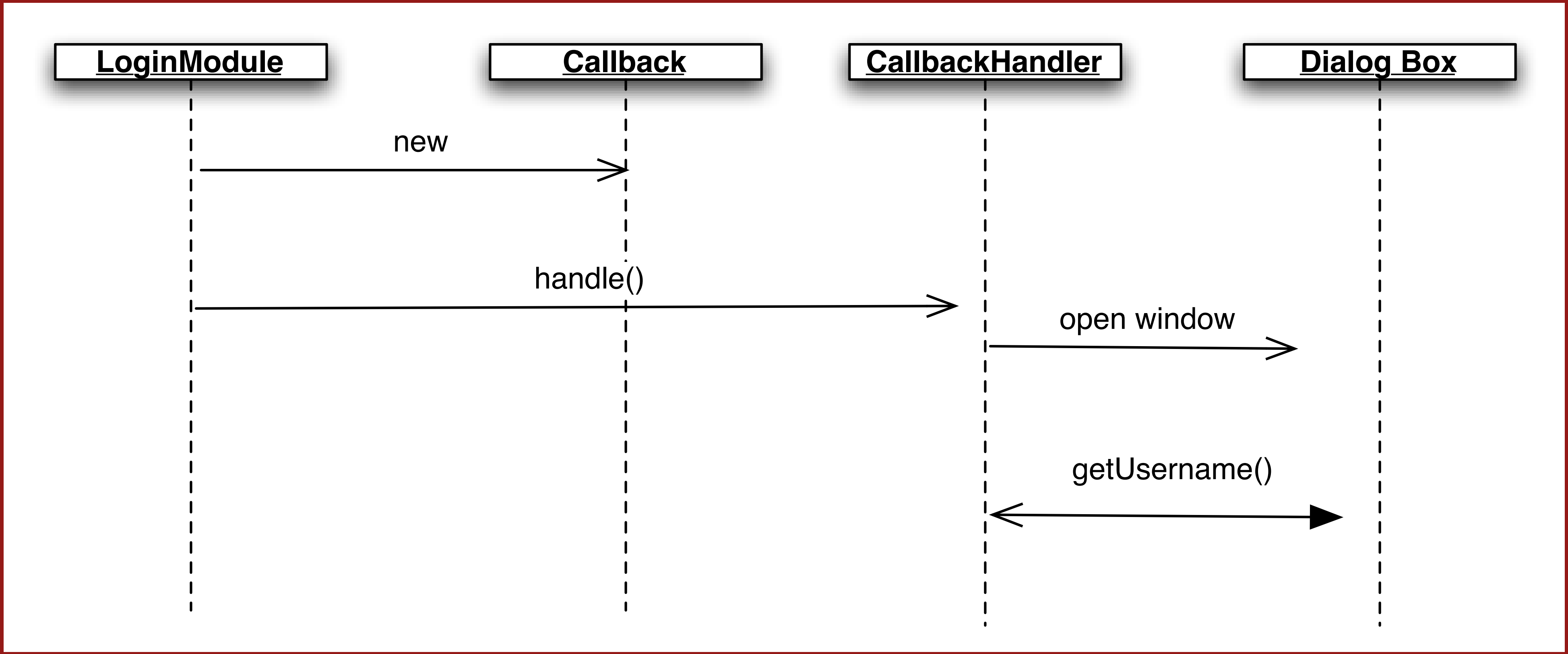
Authentication: LoginModules

> Strength of JAAS.
> Easier to customize than Authorization.
> Best documented.
> Callbacks and CallbackHandler kind of weird and confusing.

LoginModule Life-cycle


client code
LoginContext
Configuration
LoginModule(s)
new
getAppConfigurationEntry(appName)
login()
foreach(LoginModule), call initialize()
foreach(LoginModule), call login()
[auth successful]
foreach (LoginModule), call commit()
[auth failed]
foreach (LoginModule), call abort()
getSubject()

Authentication: Callback[Handler]

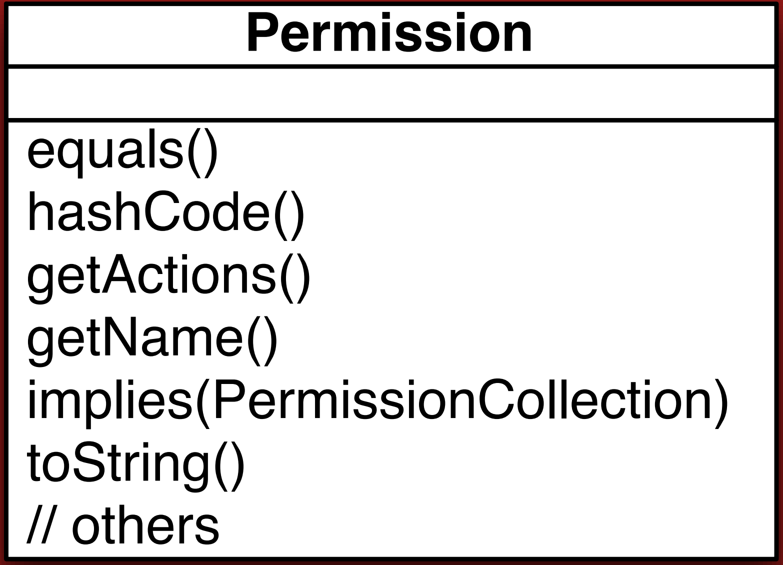


Authorization

“What can you do?”

Authorization: Permission

- The ability to perform some action,
- Usually to some target.
- Type
- Name
- Targets/Actions



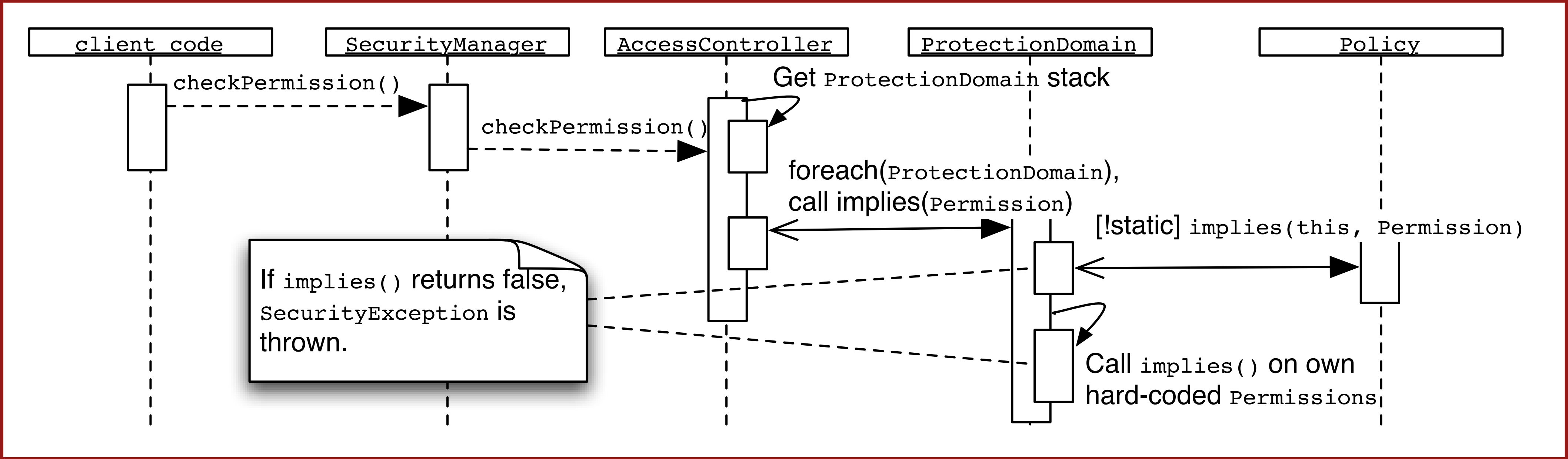
Authorization: Policy

- Service that answers all queries about which Principals have which Permissions.
- “Can this Principal perform this action?”
- An abstract class.
- Where majority of customization occurs.

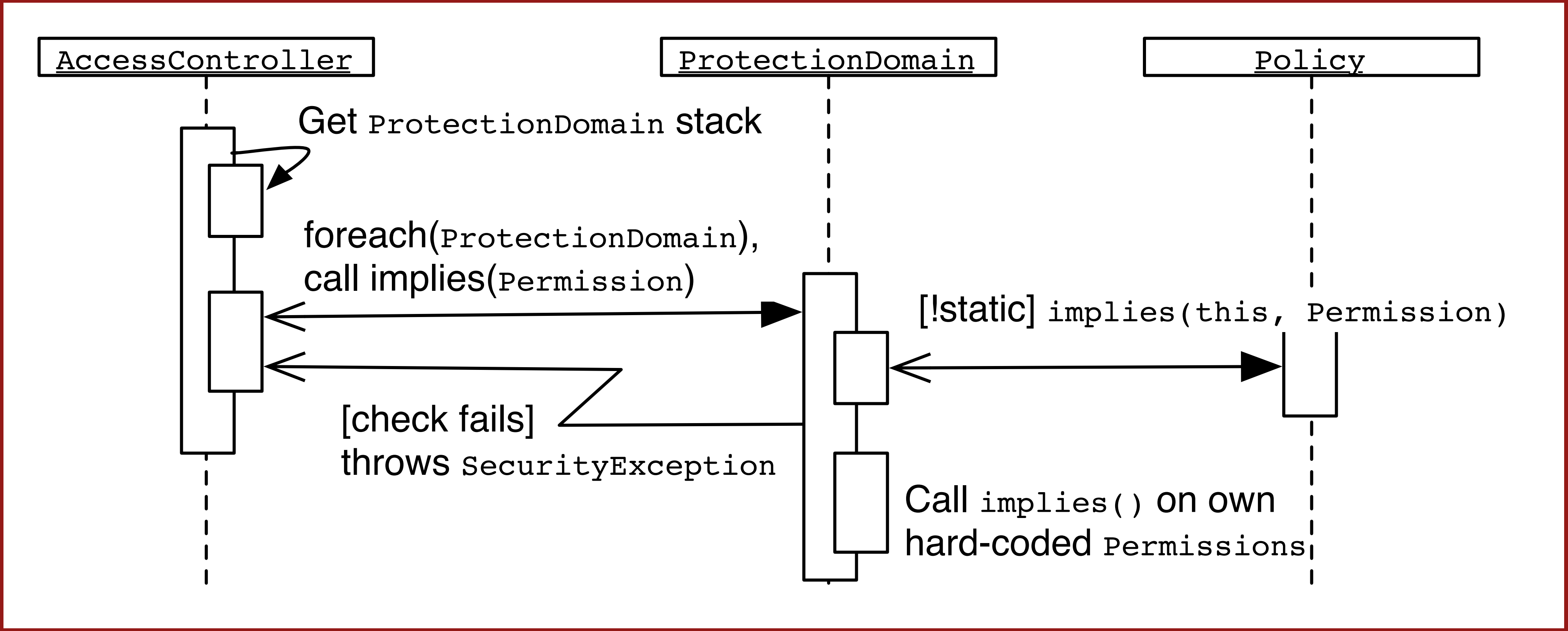
Authorization: Support Classes

- ProtectionDomain - Pricipals and CodeSource
- The Enforcers:
- SecurityManager - original
- AccessController - new for JAAS.

Authentication: Checking Permissions


client code
SecurityManager
AccessController
ProtectionDomain
Policy
checkPermission()
checkPermission()
Get ProtectionDomain stack
foreach(ProtectionDomain), call implies(Permission)
[!static] implies(this, Permission)
If implies() returns false, SecurityException is thrown.
Call implies() on own hard-coded Permissions

Authorization: ProtectionDomains


AccessController
ProtectionDomain
Policy
Get ProtectionDomain stack
foreach(ProtectionDomain), call implies(Permission)
[!static] implies(this, Permission)
[check fails] throws SecurityException
Call implies() on own hard-coded Permissions

Code: Custom Implementations

More than just flat-files.

Code

> DbLoginModule, DbConfiguration

> DbPolicy

> Base classes for Principal, Permission, LoginModule, ServletFilters.

Code: DbLoginModule

- chp04.Main.java
- DbConfiguration - cofigures authentication sources, LoginModules
- DbLoginModule - uses Callback to collect credentials.

Code: DbPolicy

> Flat-files don't fit in a multi-user application.

> chp06.Main.java

> CompositePolicy

> DbPolicy: implies() calls getPermissions(), then calls implies() on returned PermissionCollection

Helpers

Glue and Base Classes

Helpers: Extensions

- CompositePolicy
- BundledCallbackHandler
- BaseLoginModule
- ActionsPermission

Web-app Help

> Built in auth vs. filters.
> RolesTag
> PermissionTag

Built in Auth vs. Filters

- Each container has it’s own way of wiring up the Subject and configuring J2EE authentication.
- It’s messy and crappy. Book covers crapulance at length.
- Just do your own logging in and store the Subject in the session.
- User filters for URL blocking.

Tag libs

- Book uses crappy built-in approach.
- Easy to convert to retrieving Subject from session.

Role Tag

```
<auth:roles roles="customer">
<p>Only the <b>customer</b> role sees this.</p>
</auth:roles>

<auth:roles roles="admin,superadmin">
<p>Only the <b>admin</b> role sees this.</p>
</auth:roles>
```

Permission Tag

```
<perm:granted type="java.io.FilePermission"
              name="/tmp/test.txt"
              actions="read,write">
Granted FilePermission to read and write to /tmp/test.txt
</perm:granted>
<perm:notGranted type="java.io.FilePermission"
                 name="/tmp/test.txt"
                 actions="read,write">
Not granted FilePermission to read and write to /tmp/test.txt
</perm:notGranted>
```

Other Books

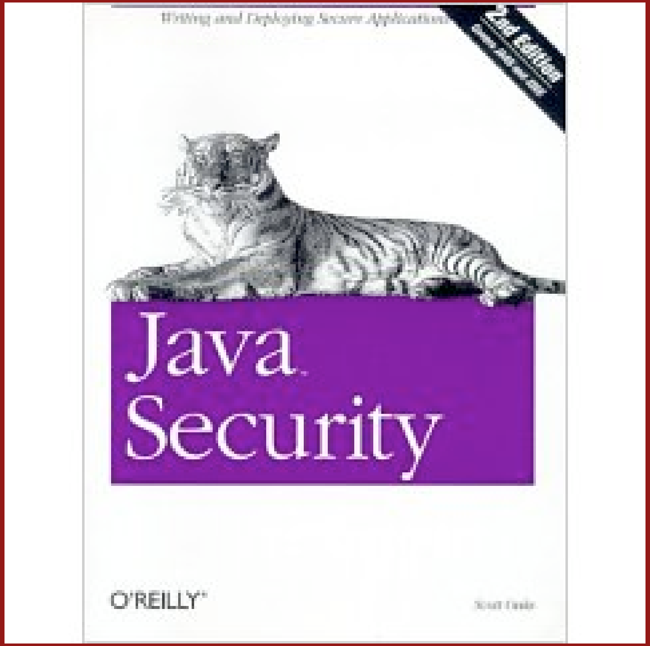
Li Gong · Gary Ellison · Mary Dageforde

Inside Java 2 Platform Security, Second Edition

Architecture, API Design, and Implementation

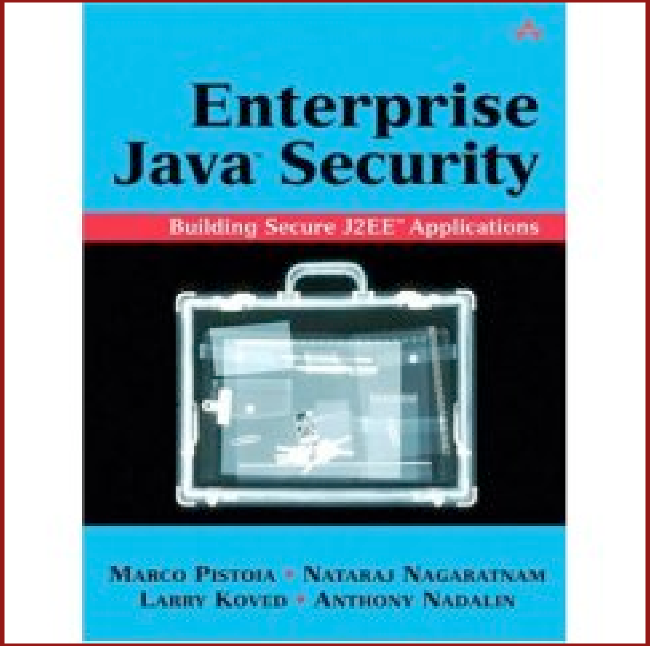
The Java Series

...from the Source


2nd Edition

Java Security

O'REILLY


Enterprise Java Security

Building Secure J2EE Applications

Marco Pistoia · Nataraj Nagaratnam

Larry Koved · Anthony Nadalin

> JAASBook.com

> http://del.icio.us/bushwald/jaas/

Thanks & Good luck!

> Use the Authentication.

> Maybe use the authorization.

Q&A

What else do you want to know?