How I Built an Access Management System Using Apache Directory Fortress

Shawn McKinney

Nov 18, 2016

ApacheCon EU, Seville

Session Objectives

- ✓ Learn about some access management specifications
- ✓ Take an *unflinching* look at an open source project named Apache Directory Fortress

Introductions

Shawn McKinney

- 

Software Architect
-  PMC Apache Directory Project
- 

Engineering Team

Agenda

We've got options:

1. What it does (specs & requirements)
2. How it works (design)
3. How it built (implementation)
4. What can it do (demos)

Pick any three

Demo Menu

1. Learn about some
   - Basic integration - RBAC0 - wicket-sample
   - Intermediate - RBAC1 - role-engineering-sample
   - Advanced - RBAC2 & 3 - apache-fortress-demo
2. Testing on
   - Fortress Web - manual or selenium
   - " " Rest - junit
   - " " Console - ad-hoc
   - " " Command Line Interface - sys-admin stuff
3. Have fun with
   - Multi-tenancy & / or Benchmarking - setting up, running, verifying

Cut to the Chase

The recipe for *any* successful technology project:

Mix well:

- Well defined set of functional specifications
- Understanding of the non-functional requirements
- Usage of common platform elements
- Practice accepted development methodologies

Specs & Requirements

What do we Build?

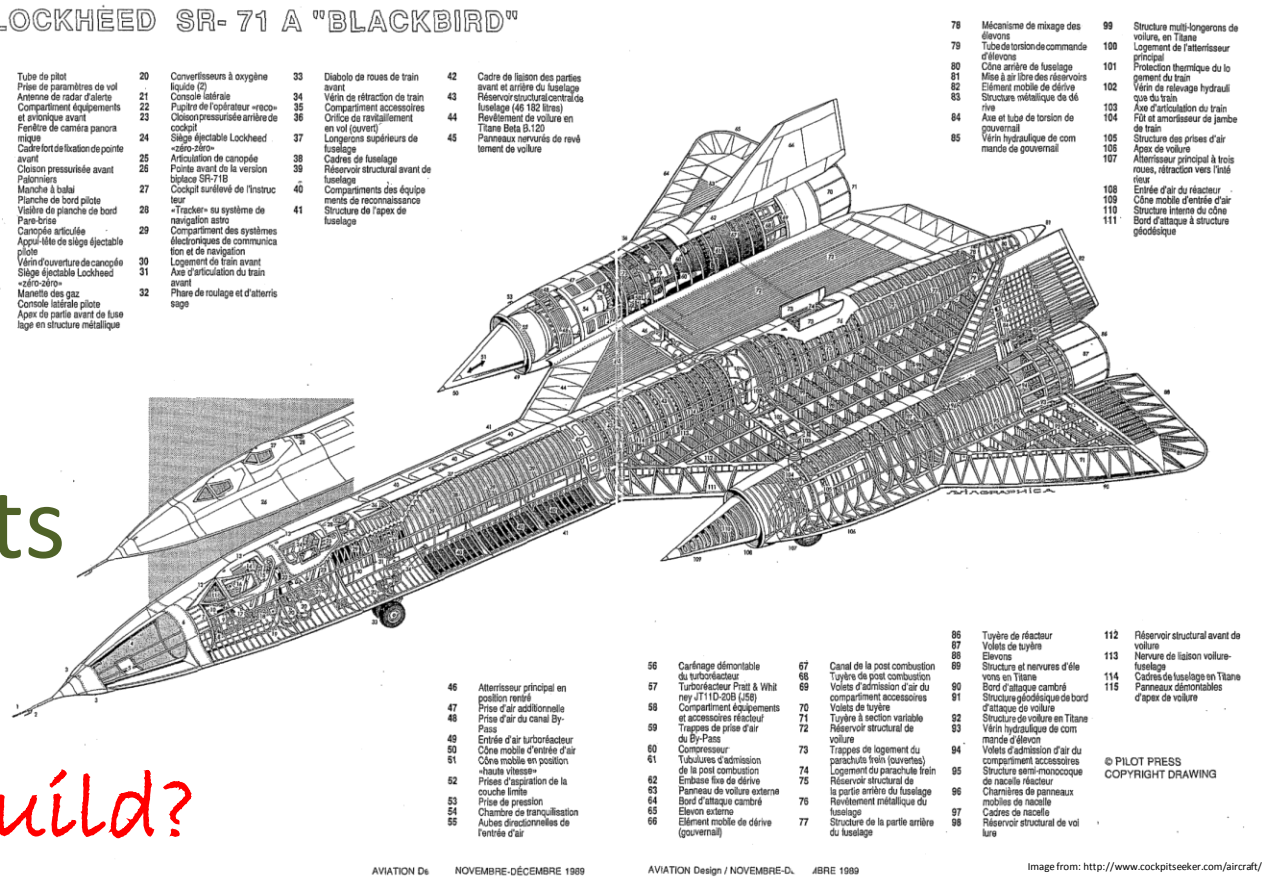


Image from: http://www.cockpitseeker.com/aircraft/

System Requirements

- **Policy Enforcement APIs** – Works on multiple platforms
- **Authentication** – Works within various protocols, i.e. SAML, OpenID Connect
- **Authorization** – Fine-grained and standards-based
- **Audit Trail** – Centralized and queryable
- **Administration** – Manage policy lifecycle
- **Service-based SLA** – Security, performance, and reliability

Why Use Functional Specifications?

- Don’t have to (repeatedly) explain yourself.
- Saves the trouble (and risk) of deciding *what*.
- Instead focus on *how*.
- Satisfies req’s didn’t know about (yet).

Which Functional Specifications

- Protocols Must Be Standards-Based:
  - Role-Based Access Control - ANSI INCITS 359
  - ~~Attribute-Based Access Control (ABAC)~~
  - IETF Password Policies (Draft)
  - ARBAC02 Delegated Administration Model

Role-Based Access Control (RBAC)

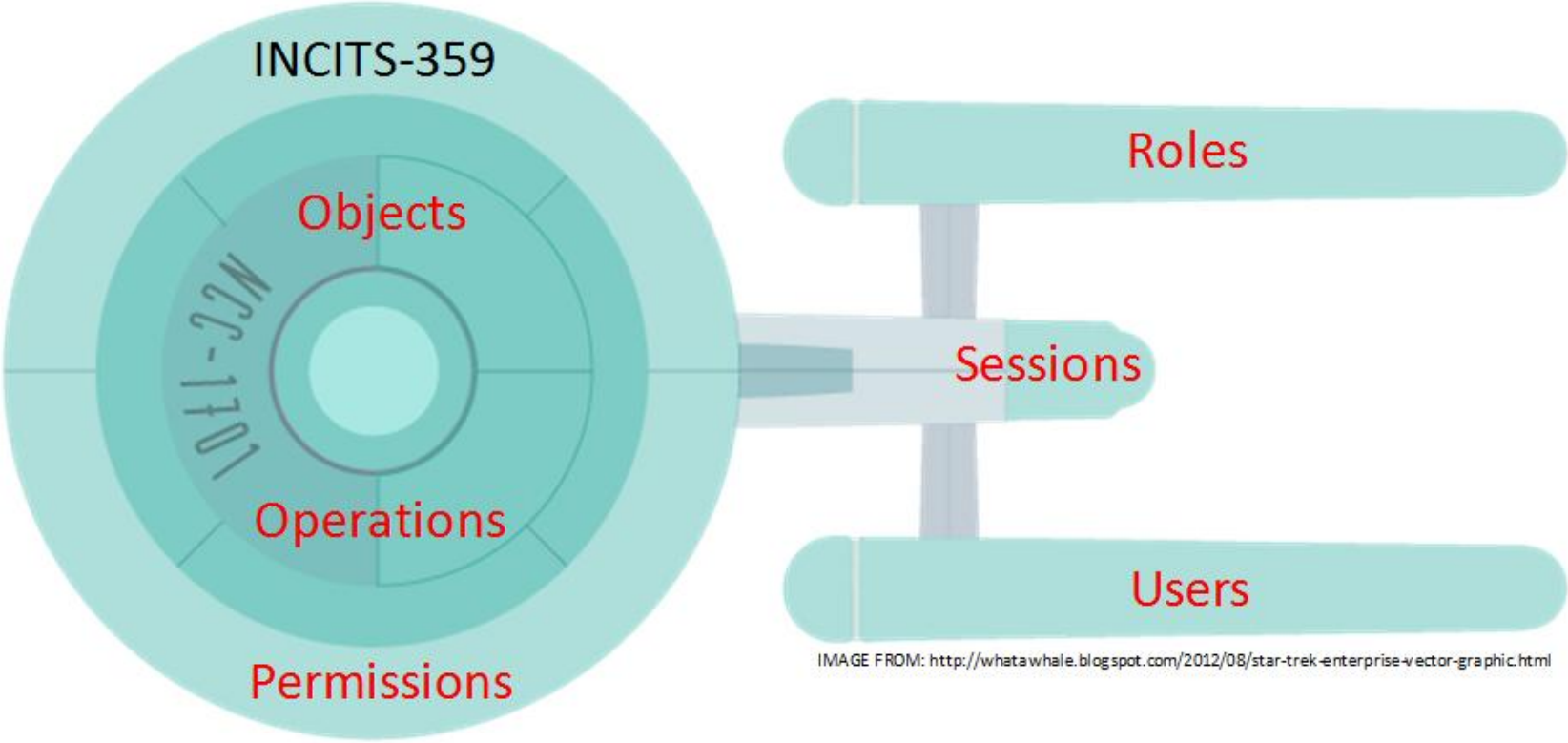
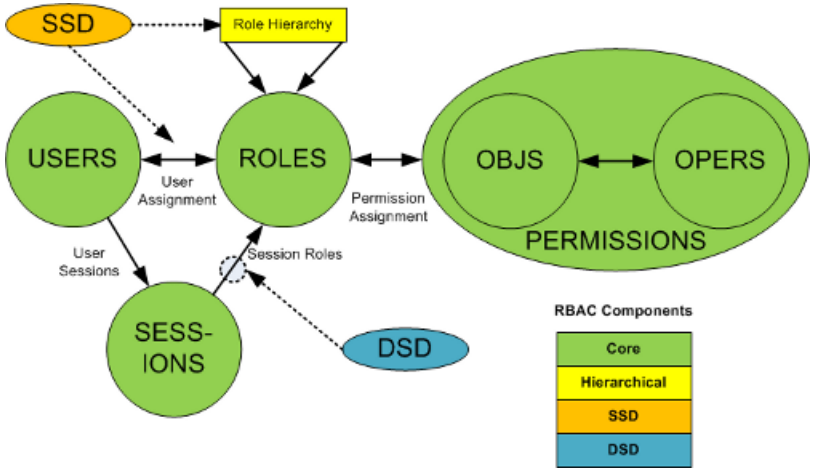


IMAGE FROM: http://whatawhale.blogspot.com/2012/08/star-trek-enterprise-vector-graphic.html

Role-Based Access Control (RBAC)

http://csrc.nist.gov/groups/SNS/rbac/

- RBAC0
  - Users, Roles, Perms, Sessions
- RBAC1
  - Hierarchical Roles
- RBAC2
  - Static Separation of Duties (SSD)
- RBAC3
  - Dynamic Separation of Duties (DSD)



ANSI INCITS 359

Static Separation of Duty Use Case

Set Name	Role Name	Type	Cardinality
Activities	Football	Static	3 (at most two)
	Band		
	Debate		

Dynamic Separation of Duty Use Case

Set Name	Role Name	Type	Cardinality
Sat Nite	Date	Dynamic	2 (at most one)
	Camping		
	Game		

Other SoD Use Cases

Many possibilities apply to financial, government, health care, education and business use cases.

RBAC Functional Model

CreateSession(*user*, *session*)
This function creates a new session with a given user as owner and an active role set. The function is valid if and only if:

- the user is a member of the *USERS* data set, and
- the active role set is a subset of the roles assigned to that user. In a RBAC implementation, the session's active roles might actually be the groups that represent those roles.

The following schema formally describes the function. The *session* parameter, which represents the session identifier, is actually generated by the underlying system.

$$
\begin{aligned}
&CreateSession(user: NAME; ars: 2^{NAMES}; session: NAME) \triangleleft \\
&\quad user \in USERS; ars \subseteq \{r: ROLES \mid (user \mapsto r) \in UA\}; session \notin SESSIONS \\
&\quad SESSIONS' = SESSIONS \cup \{session\} \\
&\quad user\_sessions' = user\_sessions \setminus \{user \mapsto user\_sessions(user)\} \cup \\
&\qquad \{user \mapsto (user\_sessions(user) \cup \{session\})\} \\
&\quad session\_roles' = session\_roles \cup \{session \mapsto ars\} \triangleright
\end{aligned}
$$

Z-notation

ANSI RBAC Functional Model

Three standard interfaces:

1. Administrative – CRUD
2. Review – policy interrogation
3. System – policy enforcement

Admin RBAC

[Link to AdminMgr javadoc]

Fortress Admin APIs map to the INCITS 359 specs

```
public interface AdminMgr {
  User addUser( User user );
  void deleteUser( User user );
  Role addRole( Role role );
  void deleteRole( Role role );
  void assignUser( UserRole uRole );
  void deassignUser( UserRole uRole );
  Permission addPermission( Permission perm );
  void deletePermission( Permission perm );
  void grantPermission( Permission perm, Role role );
  void addAscendant( Role childRole, Role parentRole);
  void addDescendant(Role parentRole, Role childRole);
  void addDsdRoleMember(SDSet dsdSet, Role role);
  void addInheritance(Role parentRole, Role childRole)
… http://git-wip-us.apache.org/repos/asf/directory-fortress-core.git
```

$$GrantPermission(object, operation, role{:}\ NAME) \triangleleft$$

$$(operation, object) \in PERMS; role \in ROLES$$

$$PA' = PA \cup \{(operation, object) \mapsto role\}$$

$$assigned\_permissions' = assigned\_permissions \setminus \{role \mapsto assigned\_permissions(roles)\} \cup \{role \mapsto (assigned\_permissions(role) \cup \{(operation, object)\})\} \triangleright$$

[Link to INCITS 359 spec]

Review RBAC

[Link to ReviewMgr javadoc]

Fortress Review APIs map to the INCITS 359 specs

```
public interface ReviewMgr {
    Permission readPermission( Permission permission );
    List<Permission> findPermissions( Permission permission );
    User readUser( User user );
    List<User> findUsers( OrgUnit ou );
    List<User> assignedUsers( Role role );
    Set<String> authorizedRoles( User user );
    List<Permission> rolePermissions( Role role );
    List<Permission> userPermissions( User user );
    Set<String> authorizedPermissionUsers(Permission perm);
    SDSet dsdRoleSet(SDSet set);
    Set<String> dsdRoleSetRoles(SDSet dsd);
    List<SDSet> dsdRoleSets(Role role);
    SDSet ssdRoleSet(SDSet set);
    Set<String> ssdRoleSetRoles(SDSet dsd);
    List<SDSet> ssdRoleSets(Role role);
    List<Role> findRoles(String searchVal);
…
```

http://git-wip-us.apache.org/repos/asf/directory-fortress-core.git

$$UserPermissions(user: NAME; result: 2^{PERMS}) \triangleleft$$

$$user \in USERS$$

$$result = \{r: ROLES; op: OPS; obj: OBJS \mid (user \mapsto r) \in UA \wedge ((op, obj) \mapsto r) \in PA \bullet (op, obj)\} \triangleright$$

[Link to INCITS 359 spec]

System RBAC

[Link to AccessMgr javadoc]

Fortress AccessMgr APIs map to the INCITS 359 specs

```
public interface AccessMgr {
  Session createSession( User user, boolean isTrusted );
  List<Permission> sessionPermissions( Session session );
  Set<String> authorizedRoles( Session session );
  void addActiveRole( Session session, UserRole role );
  void dropActiveRole( Session session, UserRole role );
  User getUser( Session session );
  boolean checkAccess( Session session, Permission perm);
}
```

http://git-wip-us.apache.org/repos/asf/directory-fortress-core.git

$$CheckAccess(session, operation, object: NAME; out\ result: BOOLEAN) \triangleleft$$

$$session \in SESSIONS; operation \in OPS; object \in OBJS$$

[Link to INCITS 359 spec]

$$result = (\exists r: ROLES \bullet r \in session\_roles(session) \wedge ((operation, object) \mapsto r) \in PA) \triangleright$$

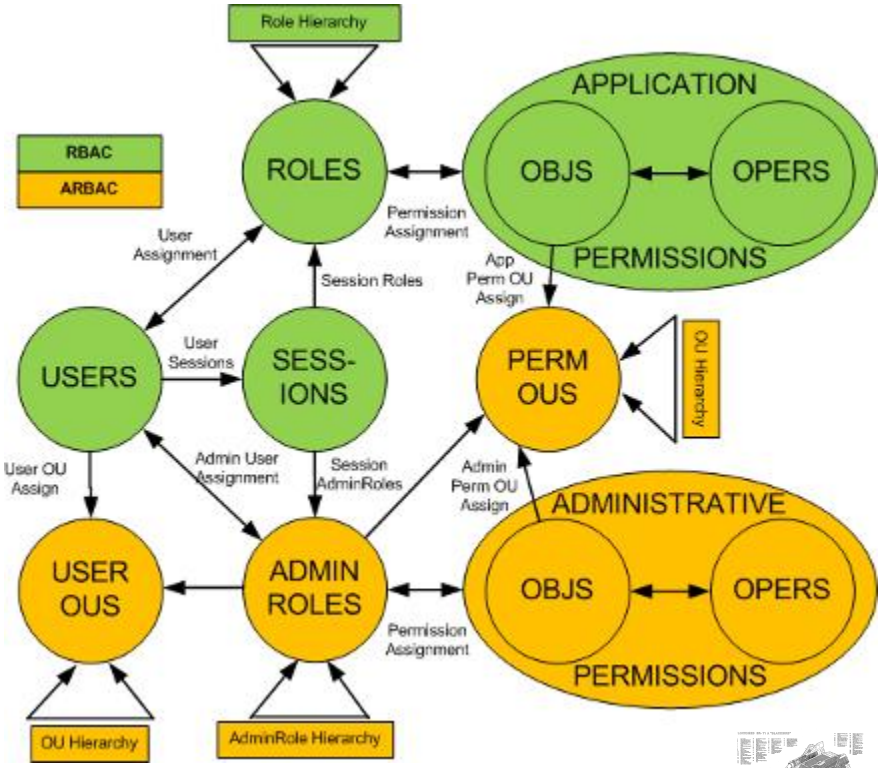
Administration Requirements

- Decentralize and distribute administrative capabilities widely
- Tight restrictions administrators
- RBAC system to control the RBAC system

Admin Role-Based Access Control (ARBAC)

http://profsandhu.com/journals/tissec/p113-oh.pdf

- Use ARBAC02 Model for administrative delegation
- Object Model: (Data)
  - AdminRoles, AdminPerms, User Orgs, Perm Orgs
- Functional Model: (APIs)
  - Delegated Administration
  - Delegated Review
  - Delegated System Mgr



Password Policies

1. A configurable limit on failed authentication attempts.
2. A counter to track the number of failed authentication attempts.
3. A time frame in which the limit of consecutive failed authentication attempts.
4. The action to be taken when the limit is reached.
5. An amount of time the account is locked (if it is to be locked)
6. Password expiration.
7. Expiration warning
8. Grace authentications
9. Password history
10. Password minimum age
11. Password minimum length
12. Password Change after Reset
13. Safe Modification of Password

https://tools.ietf.org/html/draft-behera-ldap-password-policy-10

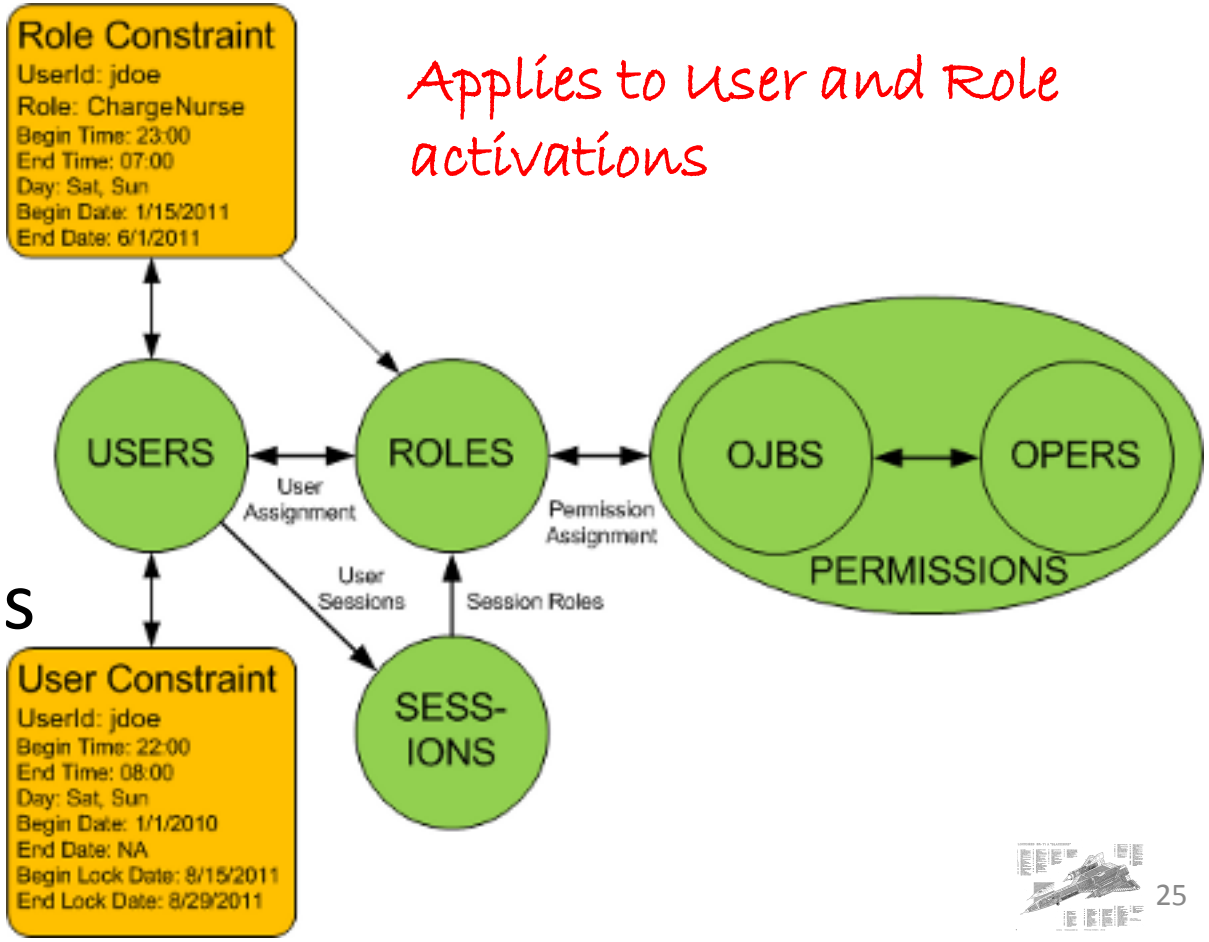
symas

Other Requirements

- Audit Trail
- Lockout Procedures based on Time & Date

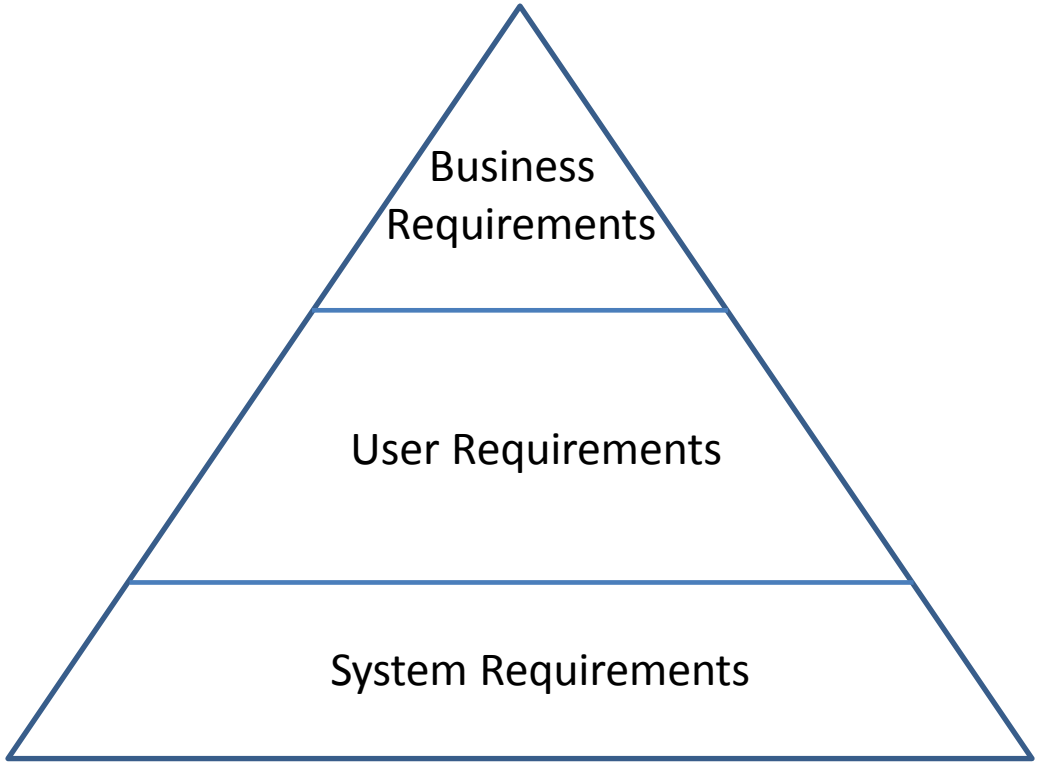
Temporal Constraints

- Time of Day
- Day of Week
- Begin and End Date
- Lockout Periods



Non-Functional Requirements

- Fault Tolerant
- Highly Available
- Multitenant
- Highly Performant



Design

Image from: http://flaviendachet.blogspot.com/2011/11/lockheed-sr-71-cutaways.html

How do we Build?

Design Considerations

- Many problems to solve:
  - Graphing, caching, configuration, persistence, logging, multitenancy, session storage, replication and performance.
  - Not to mention testing, packaging, documentation and integration.
  - But, Strive to Keep It Simple Stupid (KISS).
  - Reuse, don't reinvent.

Image from: http://wfps.k12.mt.us/teachers/carmichaelg/homework.htm

Data Persistence

- Choose between Database or LDAP for Physical Model
- Need Java framework for data access operations (DAO)

LDAP Persistence

Satisfies the SLAs:

- OpenLDAP
  - Reads/Search/Bind > 75K/second
  - Update/Delete > 10K/second
  - Replication/Highly-Available
  - Audit Trail
  - Runs on most platforms
  - Commercial support options available

Java LDAP SDK Options

- JNDI – many problems
- Netscape / Mozilla LDAP API - obsolete
- UnboundID Java LDAP API – license concerns
- Apache LDAP API – just right

Java LDAP SDK

Apache Directory LDAP API™

The modern Java LDAP API

The Apache Directory LDAP API is an ongoing effort to provide an enhanced LDAP API, as a replacement for JNDI and the existing LDAP API (jLdap and Mozilla LDAP API).

This is a "schema aware" API with some convenient ways to access all types of LDAP servers, not only ApacheDS but any LDAP server.

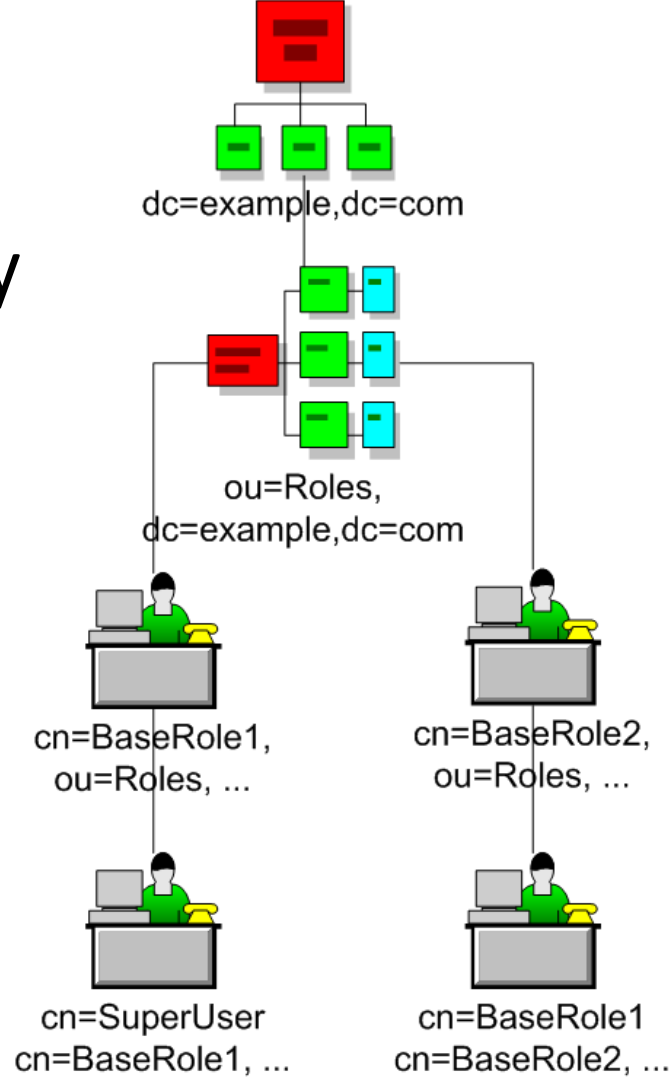
The API is OSGI ready and extensible. New controls, schema elements and network layer could be added or used in the near future.

http://directory.apache.org/api/

Data Structures

RBAC1 Limited Role Hierarchy

- Single Inheritance
- Less flexible (not useful)
- Maps onto the LDAP physical hier data model just fine

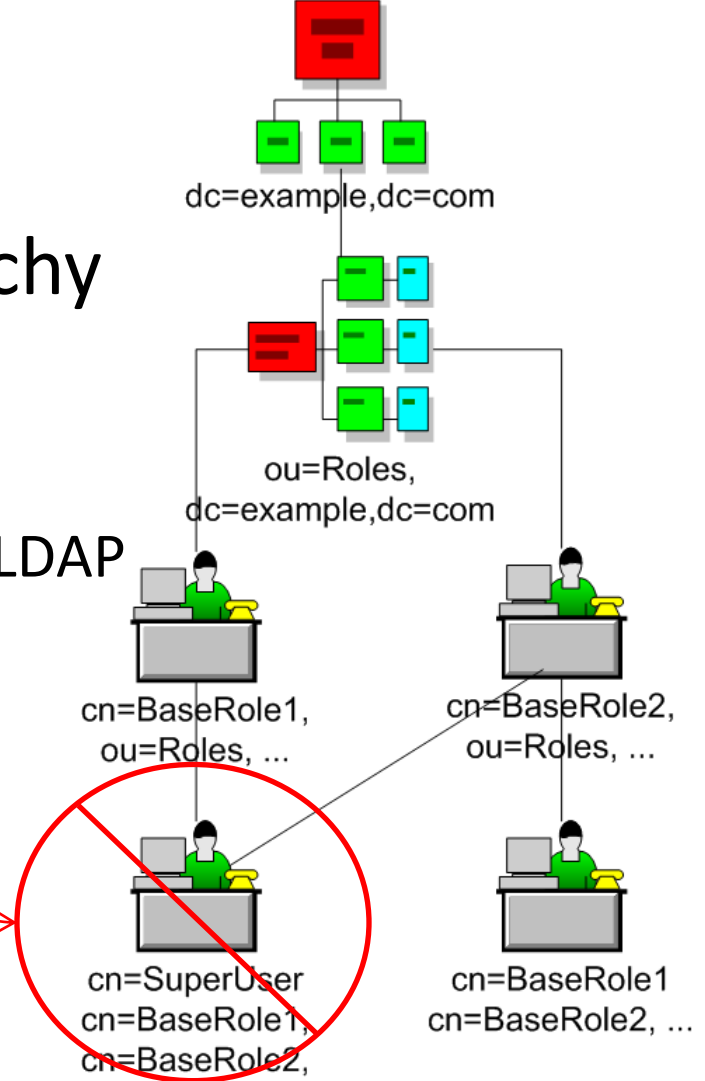


symas

Data Structures

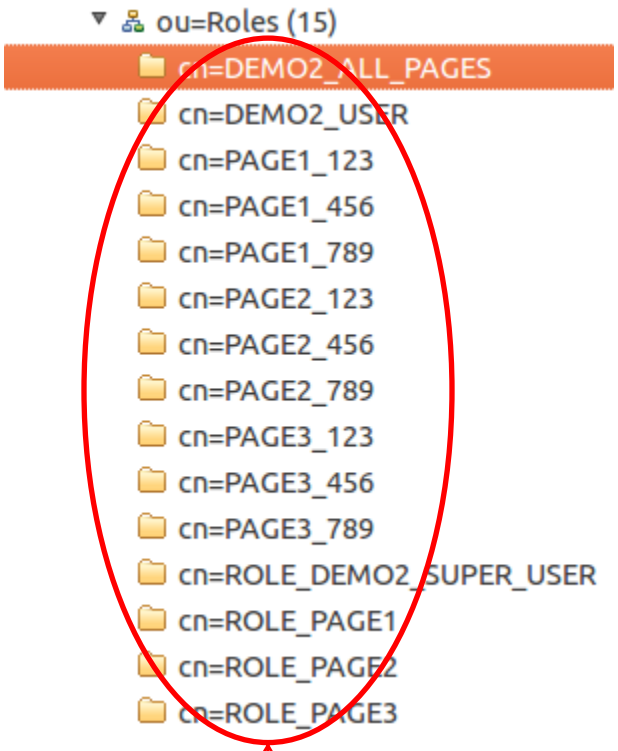
RBAC1 General Role Hierarchy

- Multiple Inheritance
- More flexible (very usable)
- A graph doesn't map onto LDAP physical model



Can't do this with LDAP

Graph Stored Flat in the Tree



cn=DEMO2_ALL_PAGES,ou=Roles,ou=RBAC,dc=example,dc=com

DN: cn=DEMO2_ALL_PAGES,ou=Roles,ou=RBAC,dc=example,dc=com

Attribute Description	Value
objectClass	***ftMods (auxiliary)***
objectClass	***ftProperties (auxiliary)***
objectClass	***ftRls (structural)***
objectClass	***top (abstract)***
cn	**DEMO2_ALL_PAGES**
ftId	**c413ef78-496a-43e9-b61b-a30c2a766618**
ftRoleName	**DEMO2_ALL_PAGES**
description	Grants User ability to switch between Pages
ftCstr	DEMO2_ALL_PAGES$0$$$$$$$
ftParents	ROLE_PAGE1
ftParents	ROLE_PAGE2
ftParents	ROLE_PAGE3
roleOccupant	uid=poweruser,ou=People,dc=example,dc=com

1. Roles all at same depth

2. Use a multi-occurring parent attribute

Use Simple Directed Graph

- http://jgrapht.org/
- A **simple directed graph**. A **simple directed graph** is a **directed graph** in which neither multiple edges between any two vertices nor loops are permitted.
- http://jgrapht.org/javadoc/org/jgrapht/graph/SimpleDirectedGraph.html

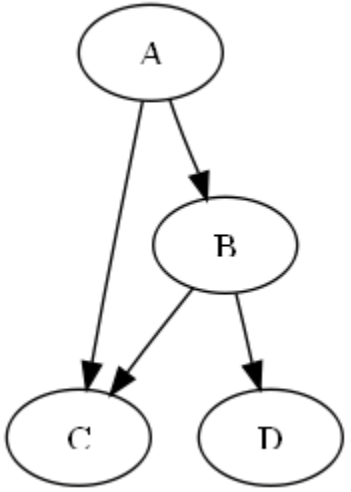
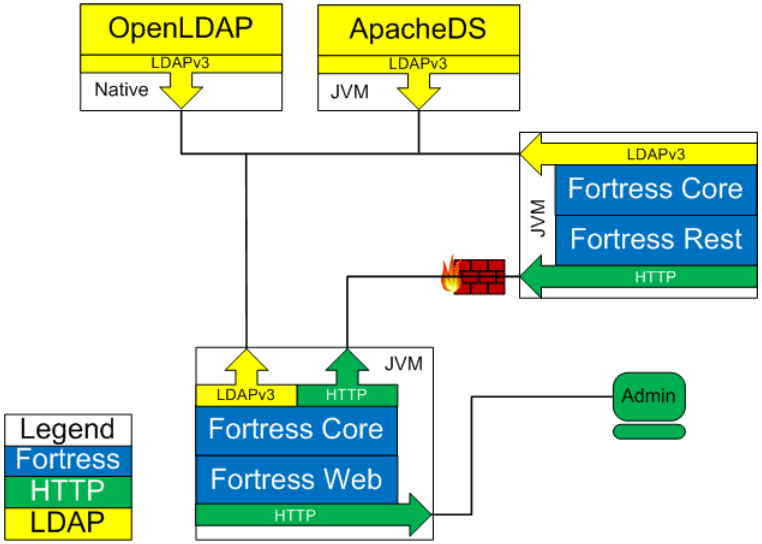


Image from: https://code.google.com/p/fluentdot/wiki/DemoSimpleDirectedGraph

What About Firewalls?

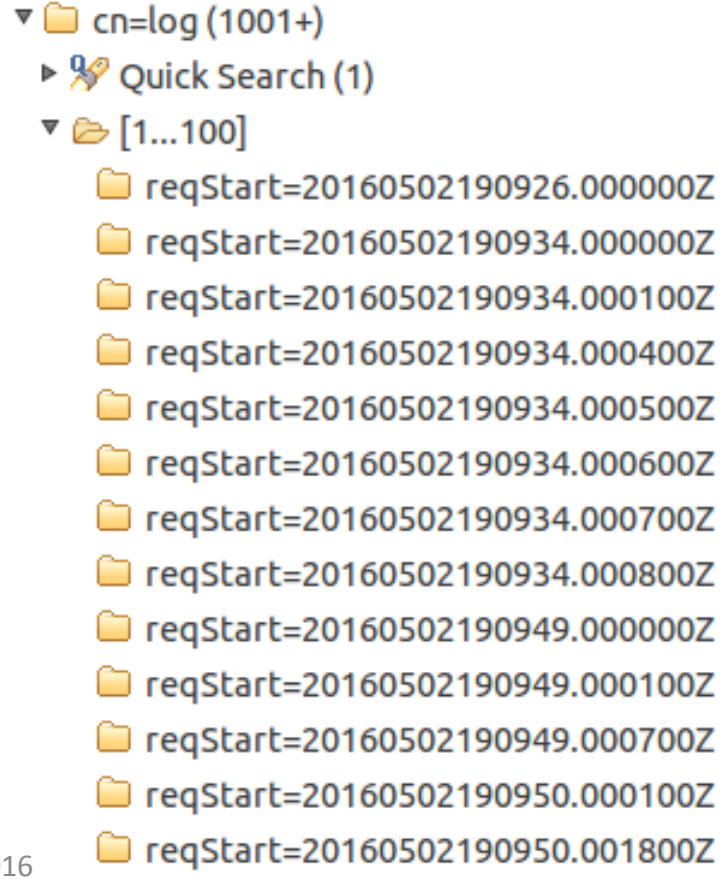
(LDAPv3 protocol isn't always allowed)

- Core API can transmit using either LDAPv3 or HTTP.



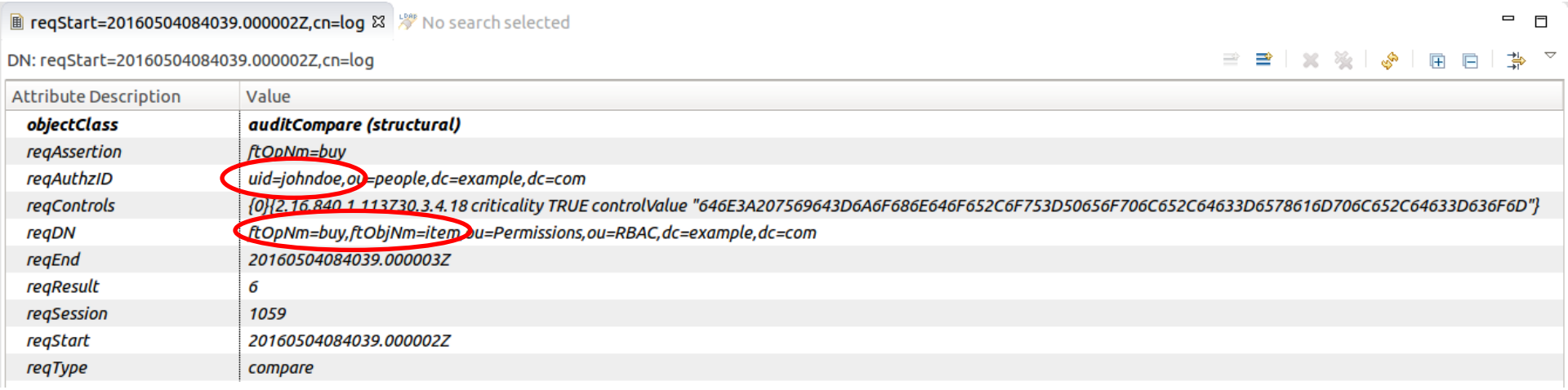
Audit

- Use OpenLDAP access log to record events:
  - Authentication
  - Check Access
  - Edits
  - Interrogations

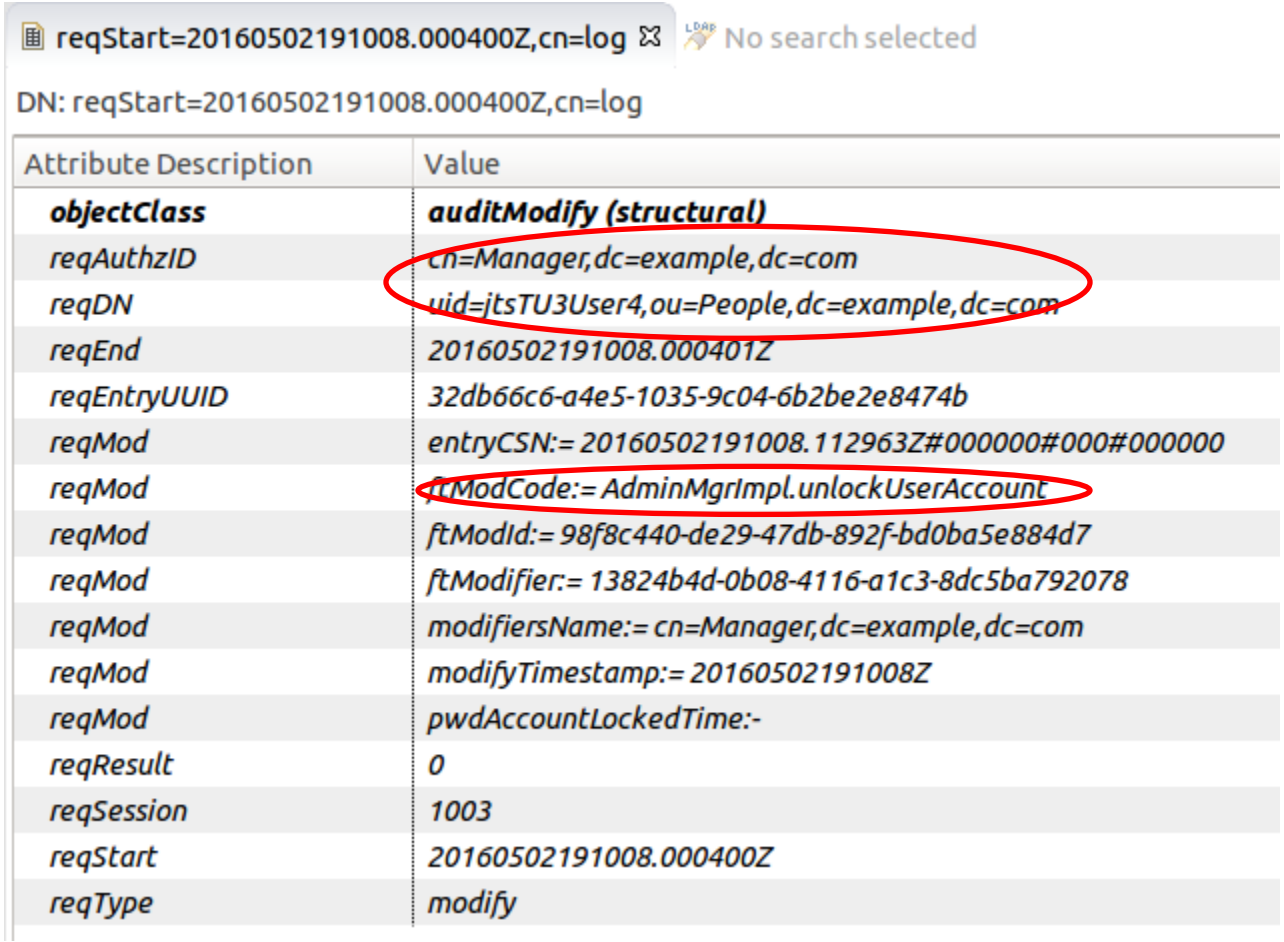


symas

Authorization Events



Administration Events



Authorization API

Method Summary

All Methods | **Instance Methods** | **Abstract Methods**

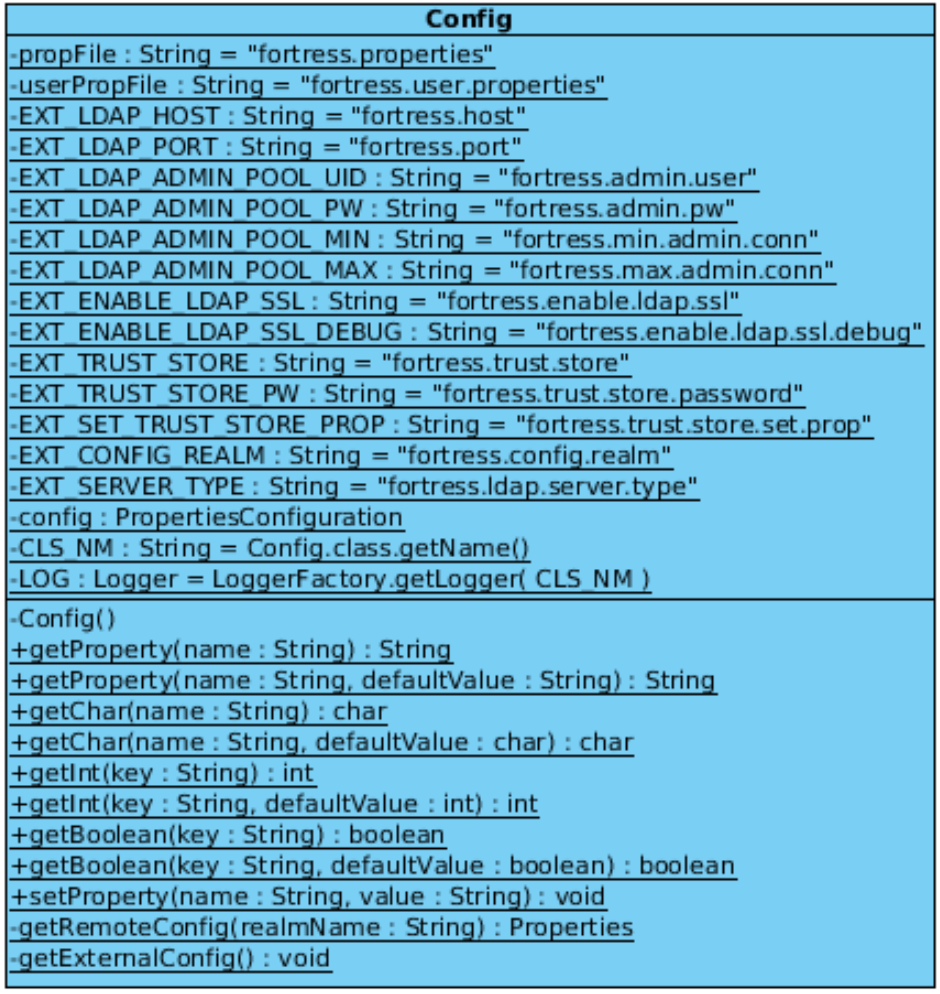
Modifier and Type	Method and Description
`List<AuthZ>`	**`getUserAuthZs(UserAudit uAudit)`** This method returns a list of authorization events for a particular user **`UserAudit.userId`** and given timestamp field **`UserAudit.beginDate`**. Method also can discriminate between all events or failed only by setting **`UserAudit.failedOnly`**.
`List<Mod>`	**`searchAdminMods(UserAudit uAudit)`** This method returns a list of admin operations events for a particular entity **`UserAudit.dn`**, object **`UserAudit.objName`** and timestamp **`UserAudit.beginDate`**.
`List<AuthZ>`	**`searchAuthZs(UserAudit uAudit)`** This method returns a list of authorization events for a particular user **`UserAudit.userId`**, object **`UserAudit.objName`**, and given timestamp field **`UserAudit.beginDate`**. Method also can discriminate between all events or failed only by setting flag **`UserAudit.failedOnly`**..
`List<Bind>`	**`searchBinds(UserAudit uAudit)`** This method returns a list of authentication audit events for a particular user **`UserAudit.userId`**, and given timestamp field **`UserAudit.beginDate`**.
`List<AuthZ>`	**`searchInvalidUsers(UserAudit uAudit)`** This method returns a list of failed authentication attempts on behalf of an invalid identity **`UserAudit.userId`**, and given timestamp **`UserAudit.beginDate`**.
`List<Mod>`	**`searchUserSessions(UserAudit uAudit)`** This method returns a list of sessions created for a given user **`UserAudit.userId`**, and timestamp **`UserAudit.beginDate`**.

Configuration

- Must be capable of retrieving properties from multiple data locations
  - File, directory, system properties, other
- Can be extended or replaced later if need be

Use Apache Commons Configuration

- Application uses façade
- Properties may be overwritten at runtime



symas

Local and Remote Config

```
# Host name and port of LDAP DIT:
host=localhost
port=389

# Options are openldap or apacheds (default):
ldap.server.type=openldap

# Audit only works if ldap.server.type == openldap:
disable.audit=false

# Used for SSL Connection to LDAP Server:
enable.ldap.ssl=${enable.ldap.ssl}
enable.ldap.ssl.debug=${enable.ldap.ssl.debug}
trust.store=${trust.store}
trust.store.password=${trust.store.password}
trust.store.set.prop=${trust.store.set.prop}

# Used to enable STARTTLS on Connection to LDAP Server
enable.ldap.starttls=false

# Used for SSL Demo with Tomcat:
key.store=${key.store}
key.store.password=${key.store.password}

# These credentials are used for read/write access to all nodes under suf
admin.user=cn=Manager,dc=example,dc=com
# LDAP admin root pass is encrypted using 'encrypt' target in build.xml:
admin.pw=secret

# This is min/max settings for LDAP administrator pool connections that h
min.admin.conn=1
max.admin.conn=10
```

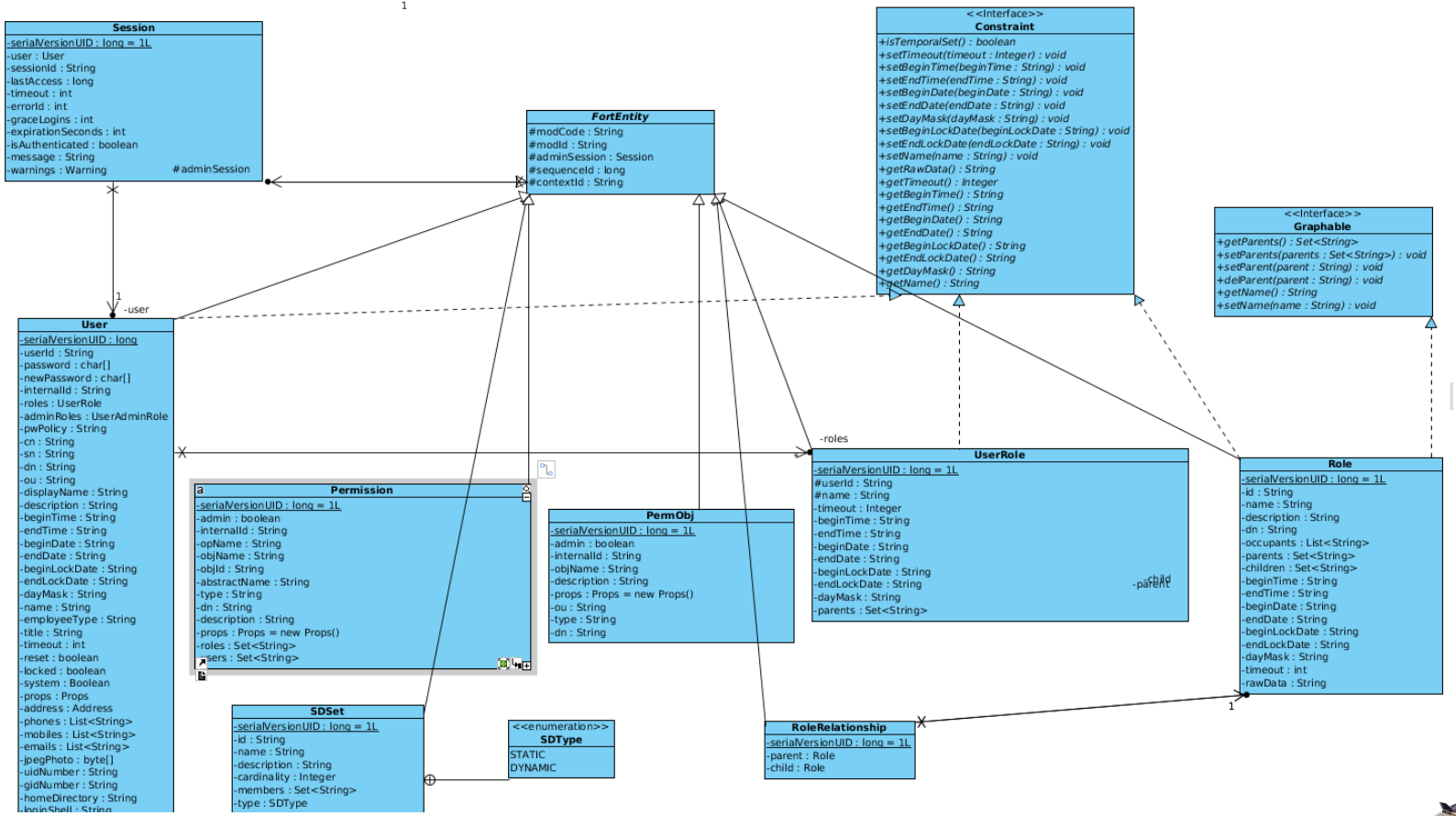
DN: cn=DEFAULT,ou=Config,dc=example,dc=com

Attribute Description	Value
objectClass	***device (structural)***
objectClass	***ftProperties (auxiliary)***
cn	**DEFAULT**
▼ ftProps (63 values)	
ftProps	adminperm.root:ou=AdminPerms,ou=ARBAC,dc=example,dc=com
ftProps	adminrole.root:ou=AdminRoles,ou=ARBAC,dc=example,dc=com
ftProps	attr.delimiter:$
ftProps	audit.root:cn=log
ftProps	authn.type:default
ftProps	clientside.sorting:true
ftProps	config.realm:DEFAULT
ftProps	example.root:ou=Examples,dc=example,dc=com
ftProps	field.length:130
ftProps	group.objectclass:configGroup
ftProps	group.properties:configParameter
ftProps	group.protocol:configProtocol
ftProps	group.root:ou=Groups,dc=example,dc=com
ftProps	ldap.filter.10:=

Data Model Questions

- How do I represent the physical data model?
- How do I represent the logical data model?
- How do I support multitenancy?

Logical RBAC Model



Logical Model

Constructor Summary

Constructors

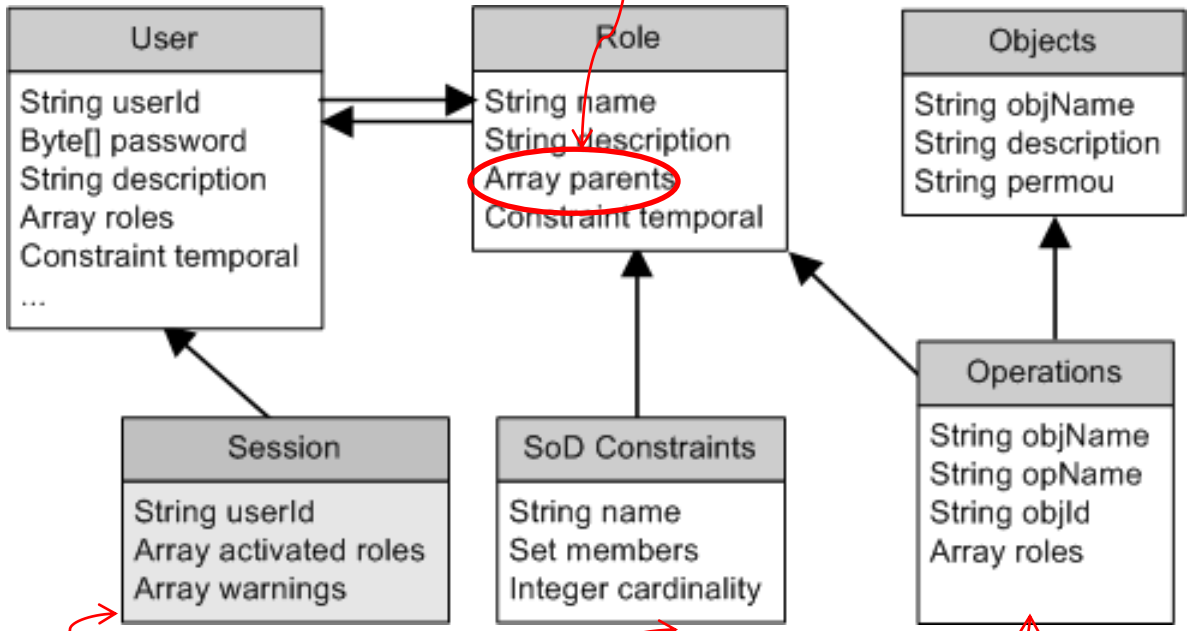
Constructor and Description
`Permission``()` Default constructor is used by internal Fortress classes and not intended for external use.
`Permission``(`**`String`** `objName)` Constructor is used for APIs that do not require opName for example ARBAC canGrant/canRevoke.
`Permission``(`**`String`** `objName, `**`String`** `opName)` This constructor is commonly used to create Permission that is a target for authorization API.
`Permission``(`**`String`** `objName, `**`String`** `opName, boolean admin)` This constructor adds the admin flag which is used to process as Administrative permission.
`Permission``(`**`String`** `objName, `**`String`** `opName, `**`String`** `objId)`

`void`	**`setObjId`**`(`**`String`** `objId)` Set optional objId which can be used to tag a Permission object with an identity, i.e.
`void`	**`setObjName`**`(`**`String`** `objName)` This attribute is required and sets the authorization target object name.
`void`	**`setOpName`**`(`**`String`** `opName)` Set the Permission operation name.

Physical RBAC Model

- Users
- Roles
- Permissions
- Constraints

[directory-fortress-core.git] / ldap / schema /



Physical Model - Permissions

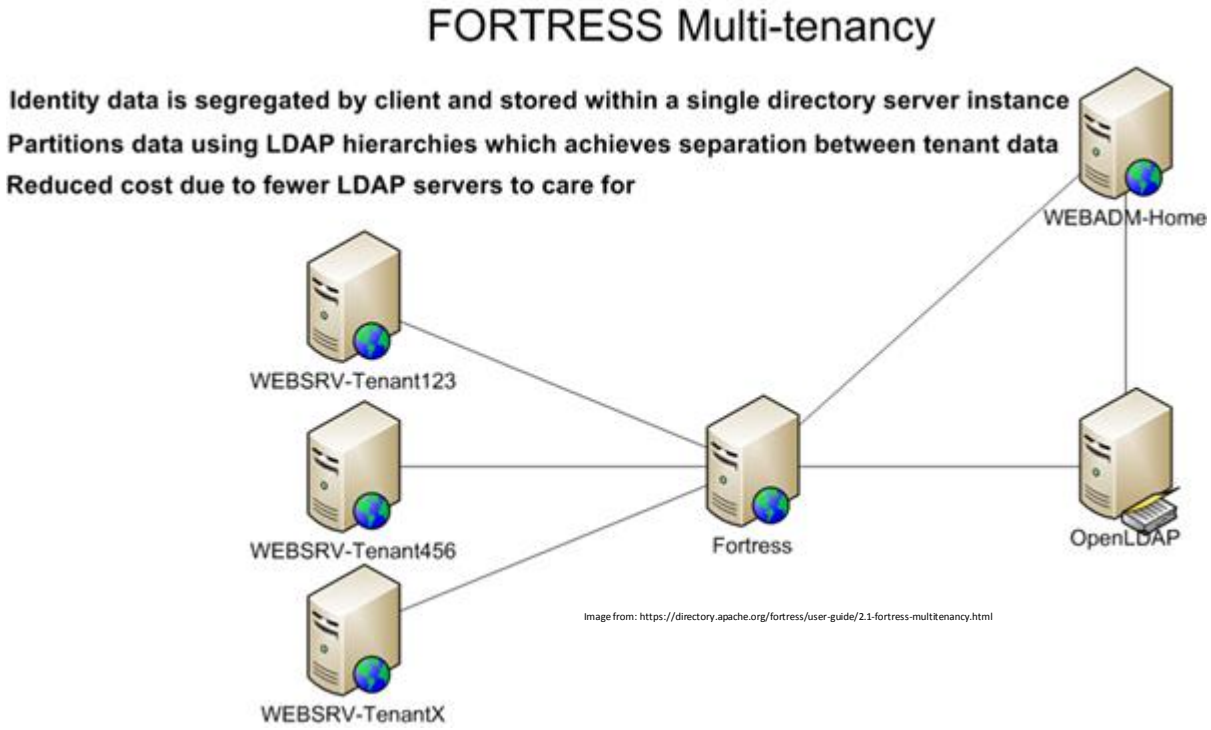
ftOpNm=ship,ftObjNm=Item,ou=Permissions,ou=RBAC,dc=example,dc=com

DN: ftOpNm=ship,ftObjNm=Item,ou=Permissions,ou=RBAC,dc=example,dc=com

Attribute Description	Value
objectClass	***ftMods (auxiliary)***
objectClass	***ftOperation (structural)***
objectClass	***ftProperties (auxiliary)***
objectClass	***organizationalRole (structural)***
objectClass	***top (abstract)***
cn	**Item.ship**
ftId	**4a586cd5-b633-4697-9001-7c98c9c4c477**
ftObjNm	**Item**
ftOpNm	**ship**
ftPermName	**Item.ship**
description	Place a product up for sale
ftRoles	Role_Sellers

Roles here is efficient at runtime

Multitenancy



Multitenancy Defined

Multitenancy

From Wikipedia, the free encyclopedia

Software Multitenancy refers to a software architecture in which a single instance of a software runs on a server and serves multiple tenants. A tenant is a group of users who share a common access with specific privileges to the software instance. With a multitenant architecture, a software application is designed to provide every tenant a dedicated share of the instance including its data, configuration, user management, tenant individual functionality and non-functional properties. Multitenancy contrasts with multi-instance architectures, where separate software instances operate on behalf of different tenants. [1]

Commentators regard multitenancy as an important feature of cloud computing.[2][3]

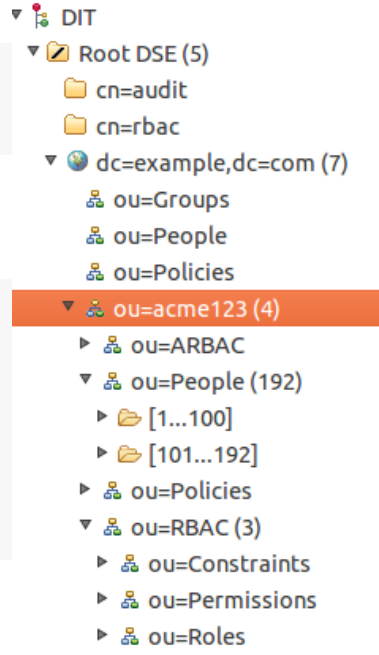
Multitenant DIT

1. Each tenant gets its own copy of the data. For example if a tenant's id is acme123, there will be a container underneath the suffix:

```
ou=acme123, dc=example, dc=com.
```

2. Beneath the acme123 *container* node will be that tenant's copy of data. For example:

```
ou=People,  ou=acme123, dc=example, dc=com
ou=Roles,   ou=acme123, dc=example, dc=com
ou=Perms,   ou=acme123, dc=example, dc=com
ou=Groups,  ou=acme123, dc=example, dc=com
...
```



Multitenant Object Model

- Client's id is passed in factory initialization
- Lifecycle of object processes data on behalf of the client id passed during initialization
  - AnyMgr:
    - createInstance(tenantId);

```
// Instantiate the AccessMgr implementation.
AccessMgr accessMgr =
    AccessMgrFactory.createInstance("Client123");
```

web-1	web-2	web-3
tenant: inbev	tenant: sabmiller	tenant: heineken
realm-1	realm-2	realm-3
tenant: default	tenant: default	tenant: heineken

Web & Realm run in separate contexts

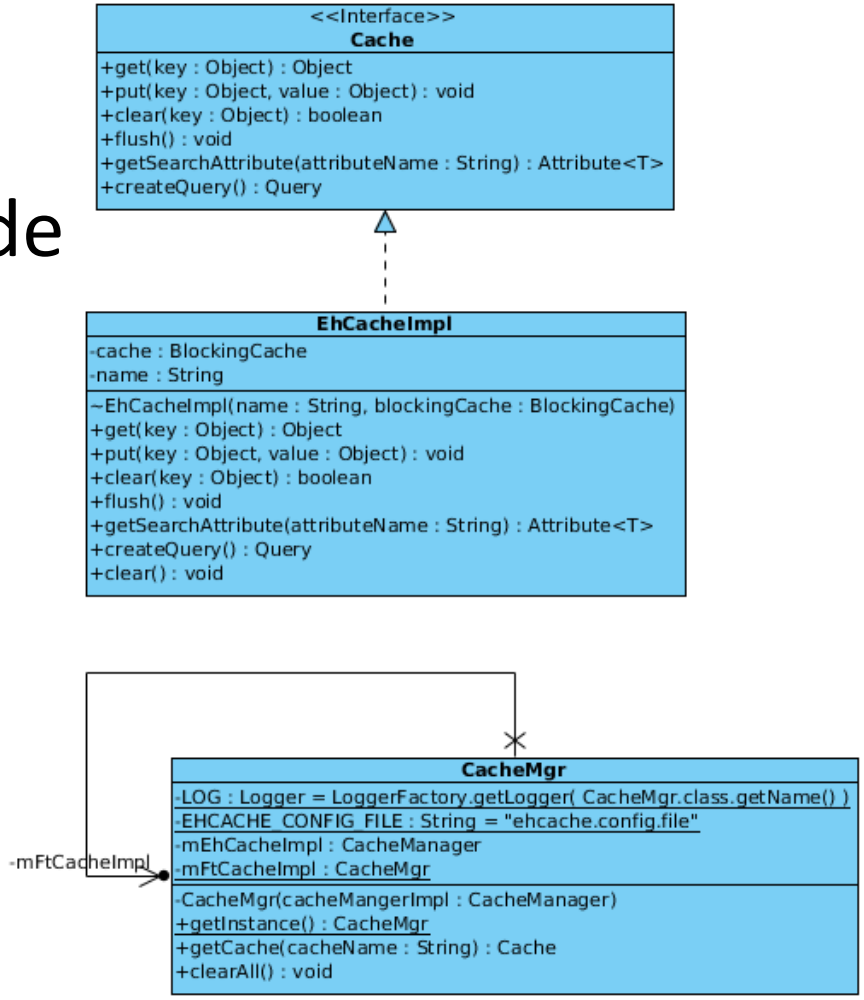
Caching

Need it for:

- Hierarchical Roles
- Static Separation of Duty datasets
- Dynamic Separation of Duty datasets
- Organizational Structures

Use Ehcache

Hide it behind a Facade



Implementation

Intro to Apache Fortress

Image from: http://sploid.gizmodo.com/fascinating-photos-reveal-how-they-built-the-sr-71-blac-1683754944

symas

Project Guidelines

- Open Source with permissive license
- High Quality and Well Maintained
- Diverse and Active Community
- Accepted and Transparent Dev Processes
- Extensible and Supportable for Many Years

Project Advantages

- Established Project Methodologies
- Well defined and understood specifications.
- Well understood technology base to build on.
- 3rd time implementing solution of this type.
  - *Practice makes perfect*

Project Dev Processes

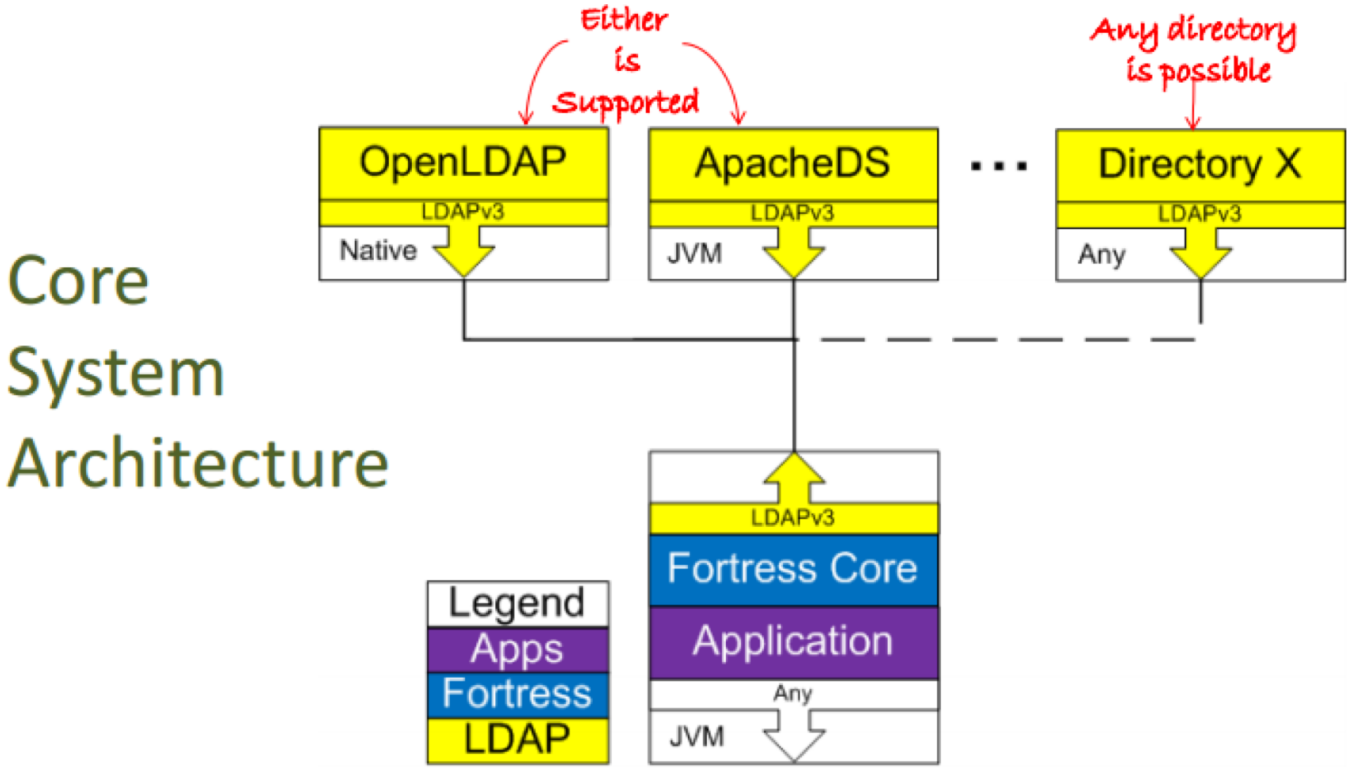
Need a sponsor that provides:

- Source Code Management
- Bug Tracking
- Mailing Lists
- Build Servers
- Binary Code Distribution
- Automated Testing

Apache Fortress™

Access Management SDK and Web Components

Apache Fortress™ is a standards-based access management system, written in Java, that provides role-based access control, delegated administration and password policy services with LDAP.



symas

Overview

- Sub-project of Apache Directory
- Written in Java
- Four Components:
  - Core – Java APIs + utilities
  - Realm – Java EE policy enforcement
  - Web – Administrative UI
  - Rest – APIs over HTTP interface

History [edit]

Fortress was first contributed in 2011 to the OpenLDAP Foundation[1] and moved to the Apache Directory project in 2014.[2]

Project History

Releases [edit]

- 2.0.0-RC1 on 2016-11-7 (current release)
- 1.0.1 on 2016-07-22
- 1.0.0 on 2016-04-12
- 1.0-RC42 on 2016-03-19
- 1.0-RC40 on 2015-04-10

http://en.wikipedia.org/wiki/Apache_Fortress

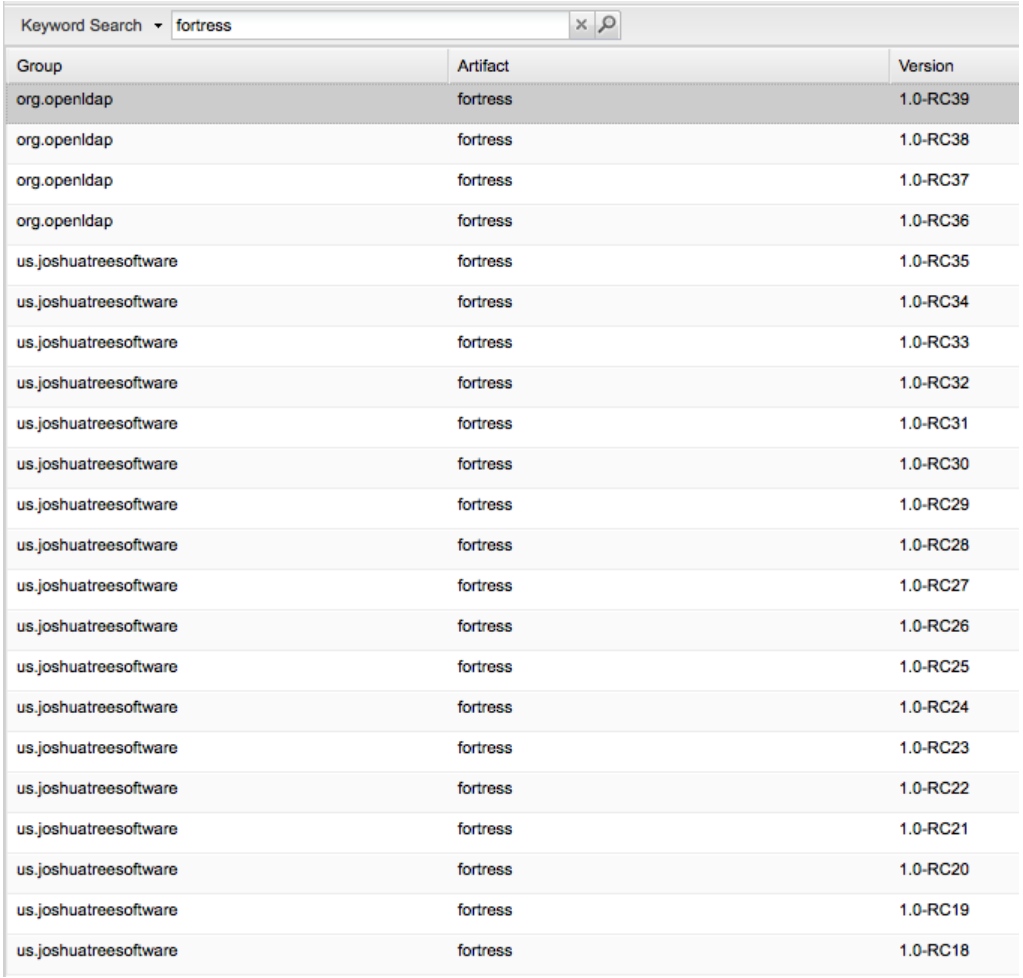
References [edit]

1. ^ "Request for Contribution of Identity Access Management Software to OpenLDAP Project". OpenLDAP Foundation. 15 July 2011. Retrieved 20 April 2016.
2. ^ "Notice to donate Fortress to the Apache Directory Project". Apache Software Foundation. 16 September 2014. Retrieved 20 April 2016.

History (cont)

25 Prior Releases

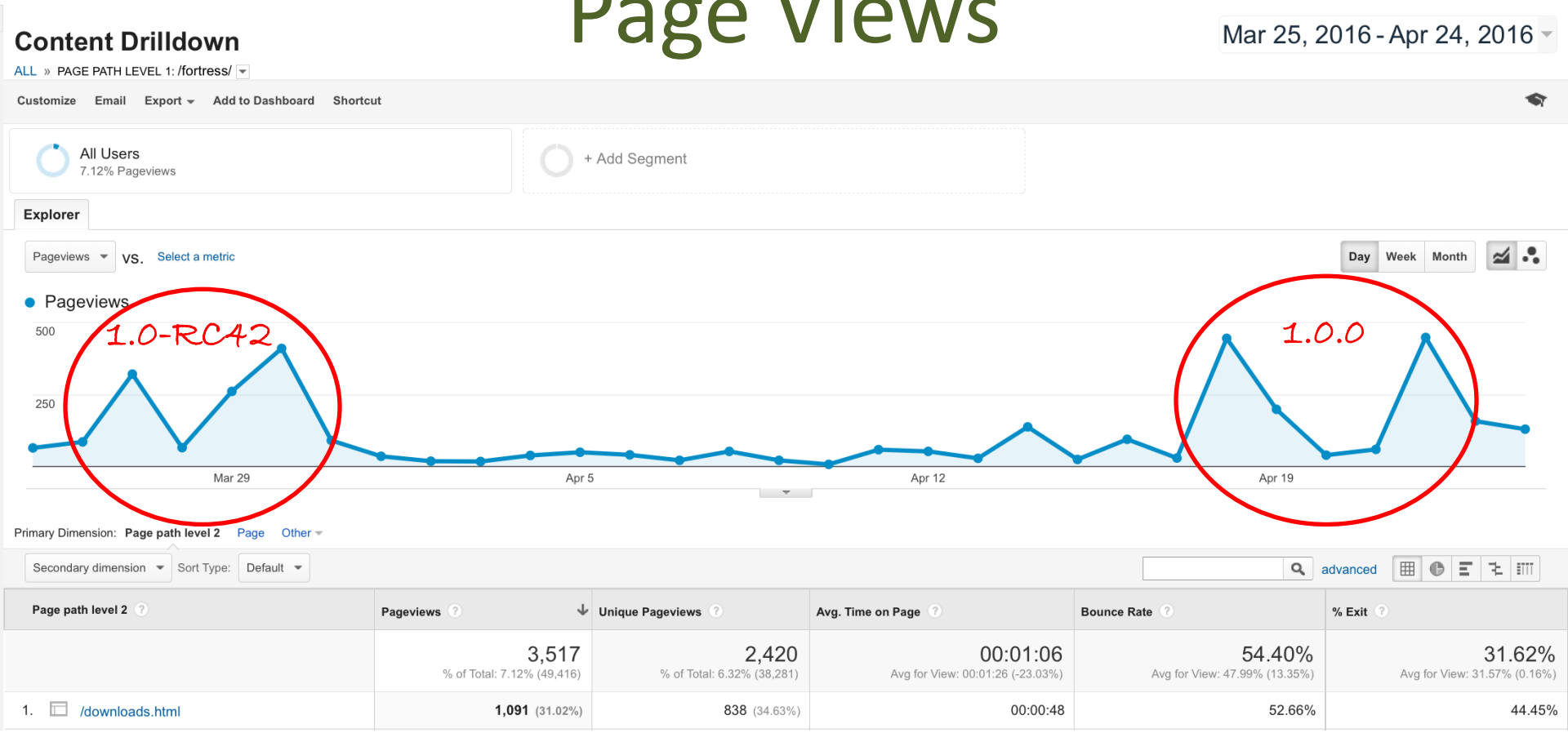
1. http://mvnrepository.com/artifact/us.joshuatreesoftware
2. http://mvnrepository.com/artifact/org.openldap
3. http://mvnrepository.com/artifact/org.apache.directory.fortress



Keyword Search ▾ fortress

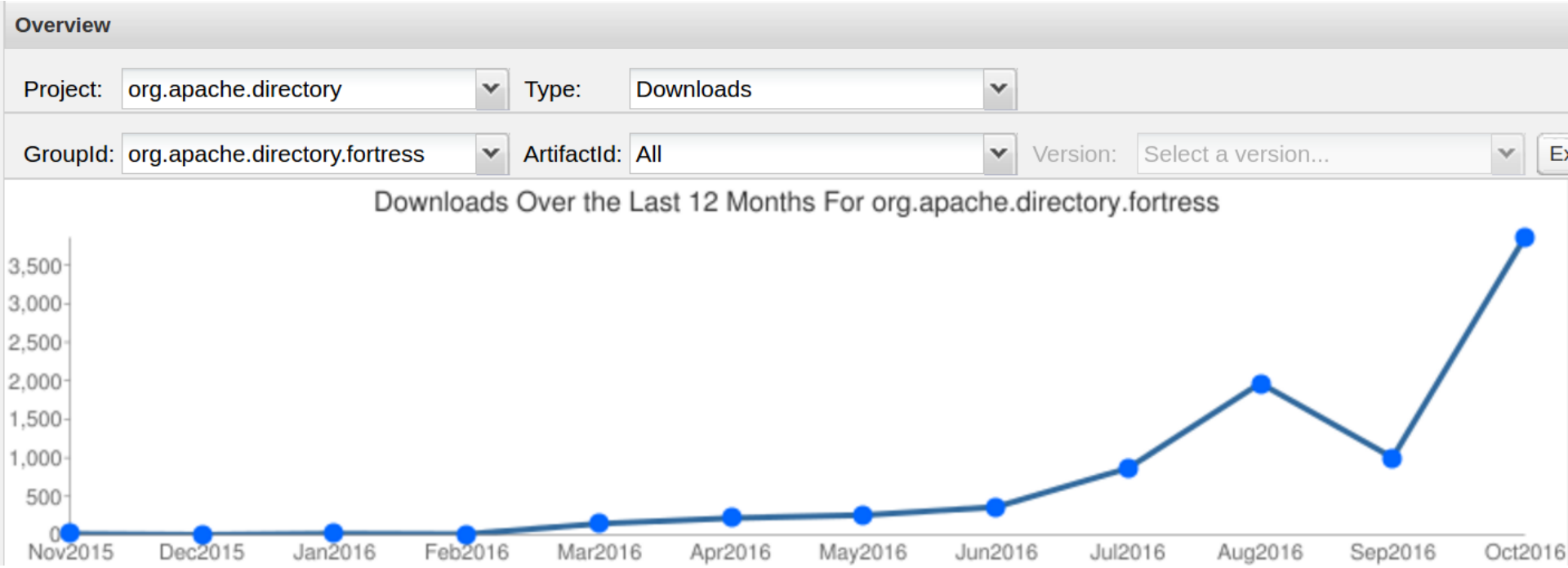
Group	Artifact	Version
org.openldap	fortress	1.0-RC39
org.openldap	fortress	1.0-RC38
org.openldap	fortress	1.0-RC37
org.openldap	fortress	1.0-RC36
us.joshuatreesoftware	fortress	1.0-RC35
us.joshuatreesoftware	fortress	1.0-RC34
us.joshuatreesoftware	fortress	1.0-RC33
us.joshuatreesoftware	fortress	1.0-RC32
us.joshuatreesoftware	fortress	1.0-RC31
us.joshuatreesoftware	fortress	1.0-RC30
us.joshuatreesoftware	fortress	1.0-RC29
us.joshuatreesoftware	fortress	1.0-RC28
us.joshuatreesoftware	fortress	1.0-RC27
us.joshuatreesoftware	fortress	1.0-RC26
us.joshuatreesoftware	fortress	1.0-RC25
us.joshuatreesoftware	fortress	1.0-RC24
us.joshuatreesoftware	fortress	1.0-RC23
us.joshuatreesoftware	fortress	1.0-RC22
us.joshuatreesoftware	fortress	1.0-RC21
us.joshuatreesoftware	fortress	1.0-RC20
us.joshuatreesoftware	fortress	1.0-RC19
us.joshuatreesoftware	fortress	1.0-RC18

Page Views



Page path level 2	Pageviews	Unique Pageviews	Avg. Time on Page	Bounce Rate	% Exit
	3,517 % of Total: 7.12% (49,416)	2,420 % of Total: 6.32% (38,281)	00:01:06 Avg for View: 00:01:26 (-23.03%)	54.40% Avg for View: 47.99% (13.35%)	31.62% Avg for View: 31.57% (0.16%)
1. /downloads.html	**1,091** (31.02%)	838 (34.63%)	00:00:48	52.66%	44.45%

Maven Downloads



Open HUB

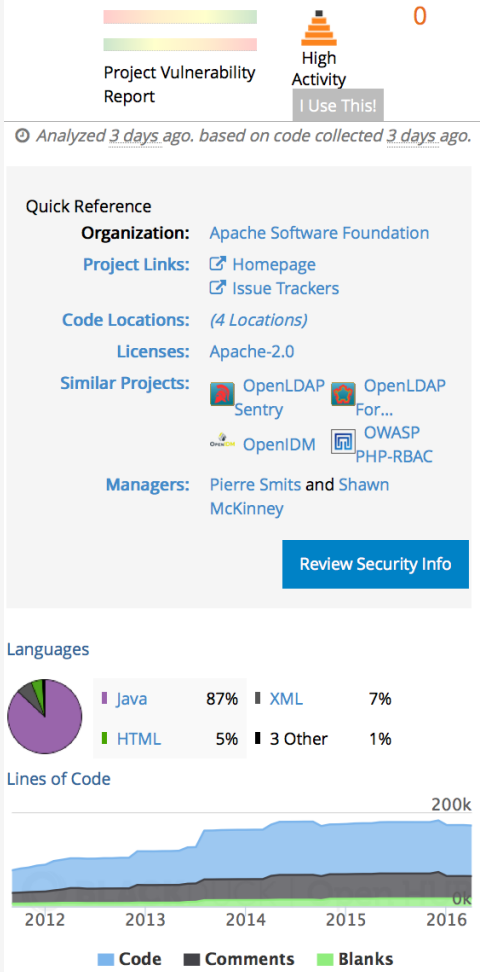
In a Nutshell, Apache Fortress...

... has had 2,001 commits made by 7 contributors
representing 112,077 lines of code

... is mostly written in Java
with an average number of source code comments

... has a well established, mature codebase
maintained by a average size development team
with increasing Y-O-Y commits

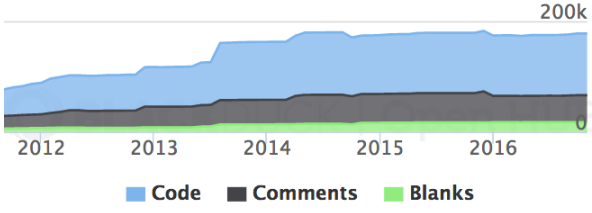
... took an estimated 28 years of effort
(COCOMO model)
starting with its first commit in September, 2011
ending with its most recent commit 10 days ago



Open HUB Details

Code

Lines of Code

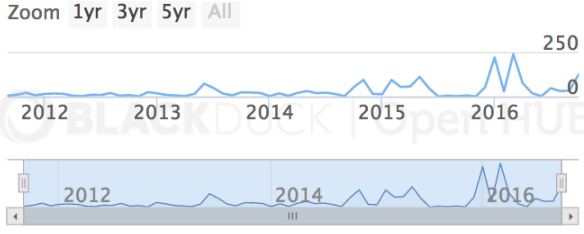


Languages

Java	88%
XML	7%
HTML	5%
3 Other	<1%

Activity

Commits per Month



30 Day Summary

Oct 5 2016 — Nov 4 2016

126 Commits

2 Contributors

12 Month Summary

Nov 4 2015 — Nov 4 2016

876 Commits

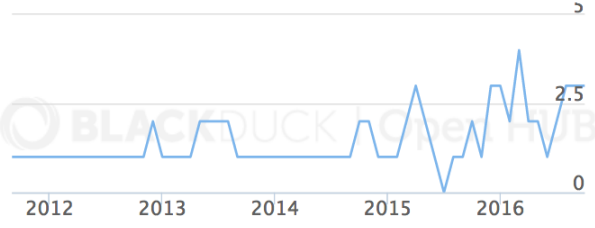
Up + 397 (82%) from previous 12 months

5 Contributors

Up + 1 (25%) from previous 12 months

Community

Contributors per Month



Most Recent Contributors

- Shawn McKinney
- clp207
- Stefan Seelmann
- ...slav Vakhlyuev
- Emmanuel Lécharny
- brainmaster

Ratings

Be the first to rate this project

Click to add your rating

Review this Project!

Project Releases

Download Fortress

Fortress Archive Downloads

To Download Apache Fortress Archives:

- Download Binaries
- Download Sources

Fortress Maven Dependencies

To embed Apache Fortress components into applications via Maven dependencies:

```
<!-- 1. Fortress Core API usage: -->
<dependency>
    <groupId>org.apache.directory.fortress</groupId>
    <artifactId>fortress-core</artifactId>
    <version>${fortress-version}</version>
</dependency>
```

Bug Tracking

FORTRESS
Key: FC · Lead: Emmanuel Lecharny · Category: Directory · URL: http://directory.apache.org/fortress

Overview Administration

- Summary
- **Issues**
- Road Map
- Change Log
- Reports
- Versions
- Source
- Reviews

Issues

All issues
Unresolved
Added recently
Resolved recently
Updated recently
Assigned to me
Reported by me
Unscheduled
Outstanding

Unresolved: By Priority

Priority	Issues	Percentage
Critical	1	3%
Major	32	89%
Minor	2	6%
Trivial	1	3%

View Issues

Unresolved: By Assignee

Assignee	Issues	Percentage
Chris Pike	3	8%
Emmanuel Lecharny	3	8%
Unassigned	30	83%

View Issues

Unresolved: By Version

Version	Issues
1.0.0-RC42	1
1.0.0	1
1.1.0	2
Unscheduled	32

View Issues

Status Summary

Status	Issues	Percentage
Open	36	18%
Resolved	150	76%
Closed	12	6%

View Issues

Unresolved: By Component

Component	Issues
No Component	36

View Issues

Unresolved: By Issue Type

Issue Type	Issues	Percentage
Bug	9	25%
Improvement	18	50%
New Feature	5	14%
Task	3	8%
Test	1	3%

View Issues

Static Code Analysis

NAME	VERSION	LOC	RCI
Apache Fortress Core	2.0.0-RC1-SNAPSHOT	39,895 ↗	99.1%
Apache Fortress Realm	2.0.0-RC1-SNAPSHOT	1,000	99.0%
Apache Fortress Rest	2.0.0-RC1-SNAPSHOT	4,940 ↗	99.1%
Apache Fortress Web	2.0.0-RC1-SNAPSHOT	16,759 ↘	99.7%

Excellent rule compliance

SonarQube code scans run nightly:

- Fortress Core: https://analysis.apache.org/dashboard/index/211987
- Fortress Realm: https://analysis.apache.org/dashboard/index/212344
- Fortress Web: https://analysis.apache.org/dashboard/index/212576
- Fortress Rest: https://analysis.apache.org/dashboard/index/212372

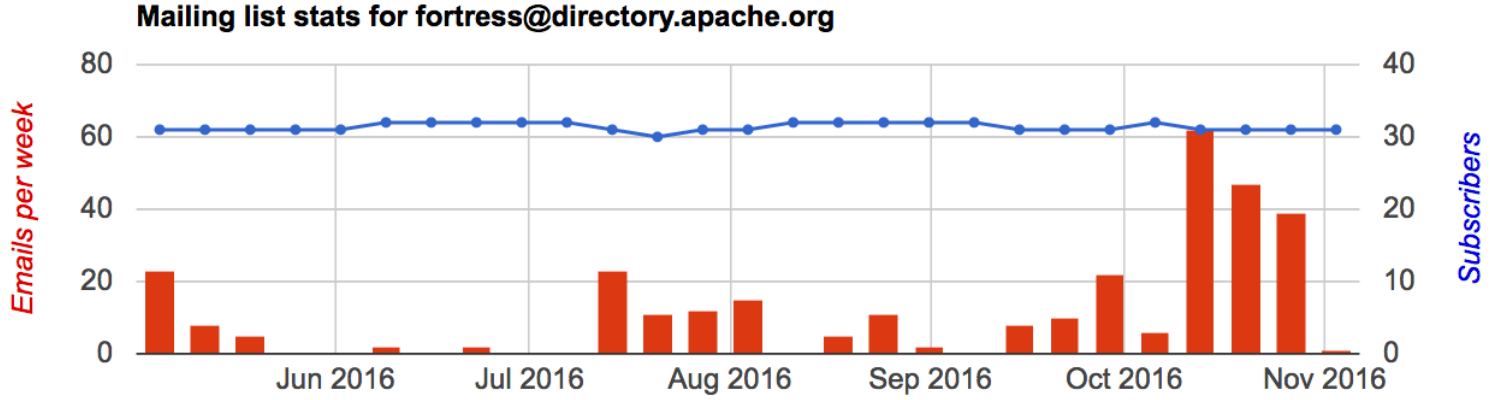
symas

Mailing List

http://mail-archives.apache.org/mod_mbox/directory-fortress/

- fortress@directory.apache.org:

Currently: 31 subscribers (up 0 in the last 3 months) (228 emails sent in the past 3 months, 86 in the previous cycle)



http://mail-archives.apache.org/mod_mbox/directory-fortress/

Year 2016			
Oct 2016	Browse		152
Sep 2016	Browse		43
Aug 2016	Browse		26
Jul 2016	Browse		53
Jun 2016	Browse	Med activity	4
May 2016	Browse		34
Apr 2016	Browse		60
Mar 2016	Browse		105
Feb 2016	Browse		74
Jan 2016	Browse		66

Year 2015			
Dec 2015	Browse		55
Nov 2015	Browse		27
Oct 2015	Browse		3
Sep 2015	Browse	Low activity	20
Aug 2015	Browse		25
Jul 2015	Browse		1
Jun 2015	Browse		36
May 2015	Browse		29
Apr 2015	Browse		64
Jan 2015	Browse		2

Year 2014			
Dec 2014	Browse	Crickets chirping	10
Nov 2014	Browse		1
Oct 2014	Browse		4

Mailing List

symas

Apache Fortress

From Wikipedia, the free encyclopedia

This article **relies too much on references to primary sources**. Please improve this by adding secondary or tertiary sources. *(April 2016)*

Apache Fortress is an open source project of the Apache Software Foundation and a subproject of the Apache Directory. It's a standards-based access management system, written in Java, that provides role-based access control, delegated administration and password policy services with LDAP.

Standards implemented:

- Role-Based Access Control (ANSI INCITS 359)
- Administrative Role-Based Access Control (ARBAC02)
- IETF Password Policy (draft)

Fortress has four separate components:

- Core - Java Access Management SDK
- Realm - Java EE security for Apache Tomcat
- Rest - HTTP protocol wrappers for the APIs
- Web - HTML pages for the APIs

Contents [hide]

Apache Fortress

Developer(s)	Apache Software Foundation
Stable release	1.0.0 / April 12, 2016
Development status	Active
Written in	Java
Operating system	Cross-platform
Type	Access Control
License	Apache License 2.0
Website	http://directory.apache.org/fortress/

Notability Concerns

Notability Concerns

@Smckin: wrt notability, merely being an Apache project does not make it notable, notability is not inherited. Significant coverage in multiple independent published sources is required, please see WP:GNG for the full criteria. AFAICT, the only source in the article that qualifies is the SD Times link. If sufficient sources cannot be found then that's usually a sign that it's WP:TOOSOON for the article.

And about primary sources, you're not supposed to just add a bunch of secondary source links to the article; articles should be written mainly based on secondary source coverage and using the sources as citations. See WP:PSTS, WP:V. -- intgr [talk] 07:29, 22 April 2016 (UTC)]

Notability Concerns (cont)

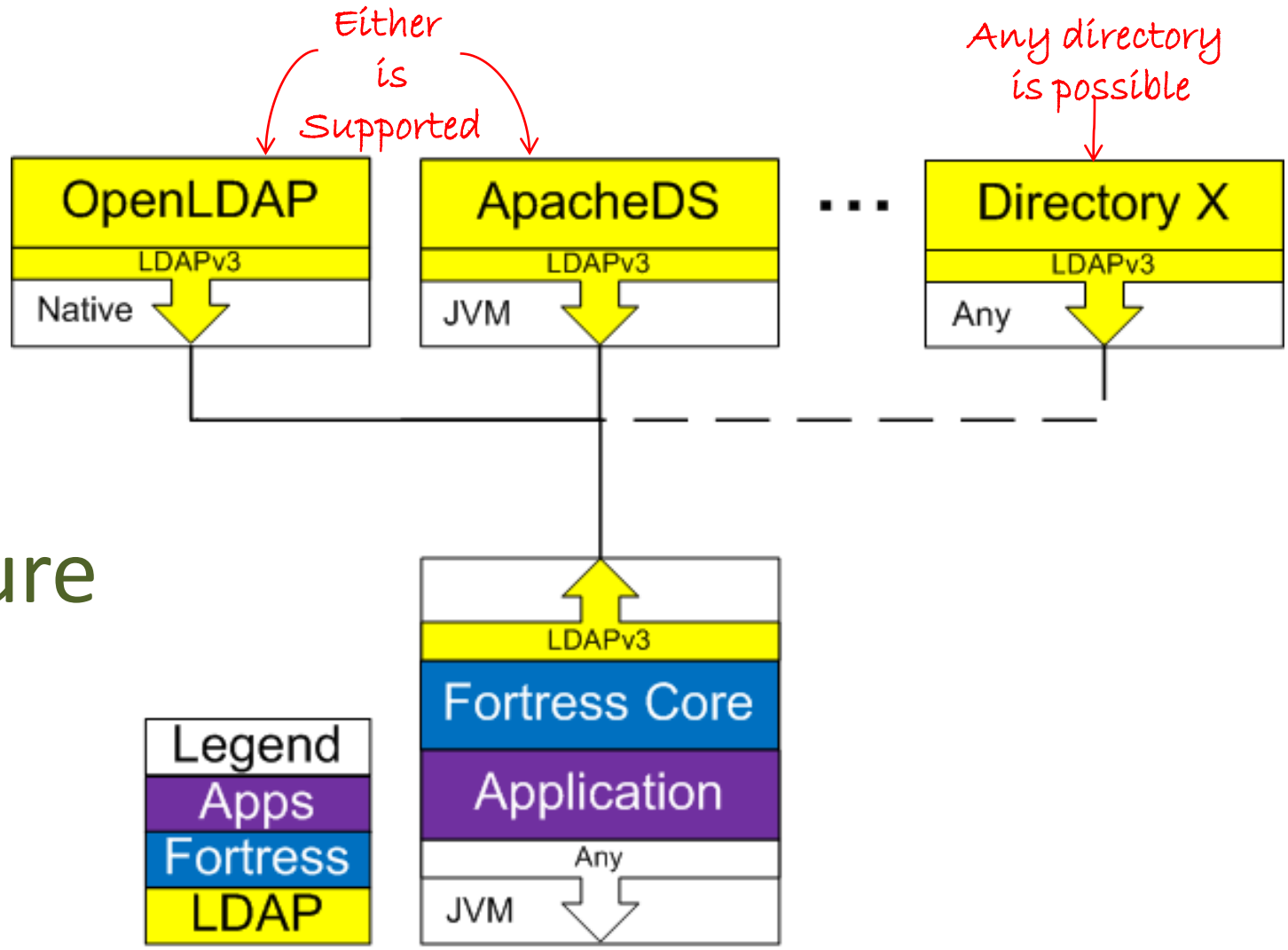
@Intgr: According to your reference: "Notability of one or more members of some group or class of subjects may or may not apply to other possible members of that group. Discuss based upon the individual subject, not the subject's overarching classification or type. If a subject under discussion is independently notable, provide the evidence to show that." Which leaves this discussion open to interpretation despite your assertion of such being a hard rule. So I must disagree with your statement wrt notability on this subject, and in particular Apache projects which are notable due to the long list of public activity and artifacts that are generated. There are mailing lists where people discuss the software, websites where their artifacts are published, and conferences where their ideas as discussed. Have you looked? These sources may not fulfill WP's traditional definition of notability but that does not lessen their significance. The Apache projects, and the events that surround them, are of historical importance. Even projects incubating are noteworthy due to their inherent historical and technological significance. Having references on wikipedia will provide a context for understanding what these communities built, when and how. They help those that follow understanding what has already been done and what work remains. Smckin (talk) 02:15, 24 April 2016 (UTC)

@Intgr: says "wrt notability, merely being an Apache project does not make it notable" It is necessary to grant notability to apache projects so that they can then become tertiary sources for other projects. Smckin (talk) 02:39, 24 April 2016 (UTC)

Components

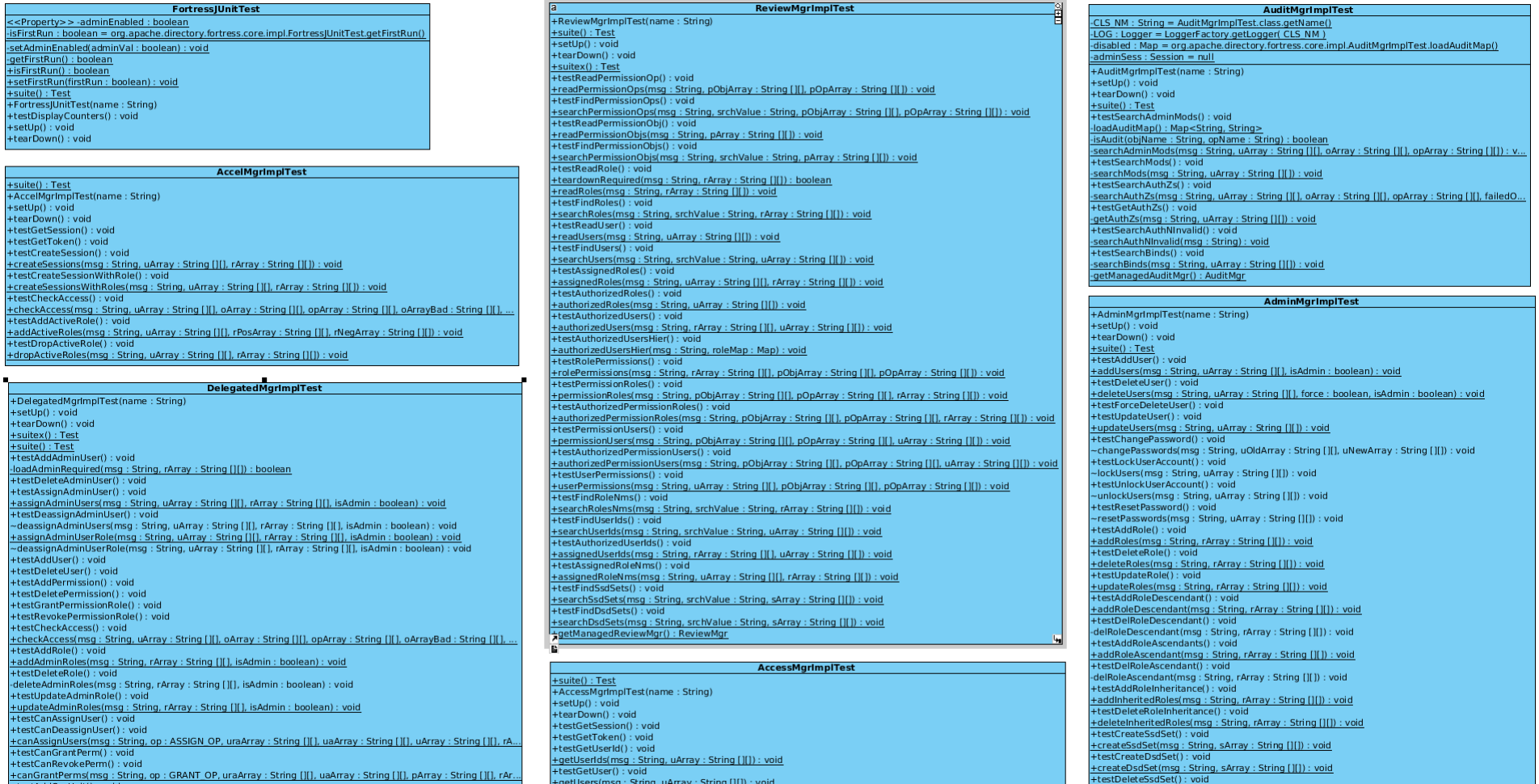
1. Core – Java SDK
2. Realm – Java EE Policy Enforcement
3. Rest – HTTP Interface
4. Web – HTML Interface

Core System Architecture



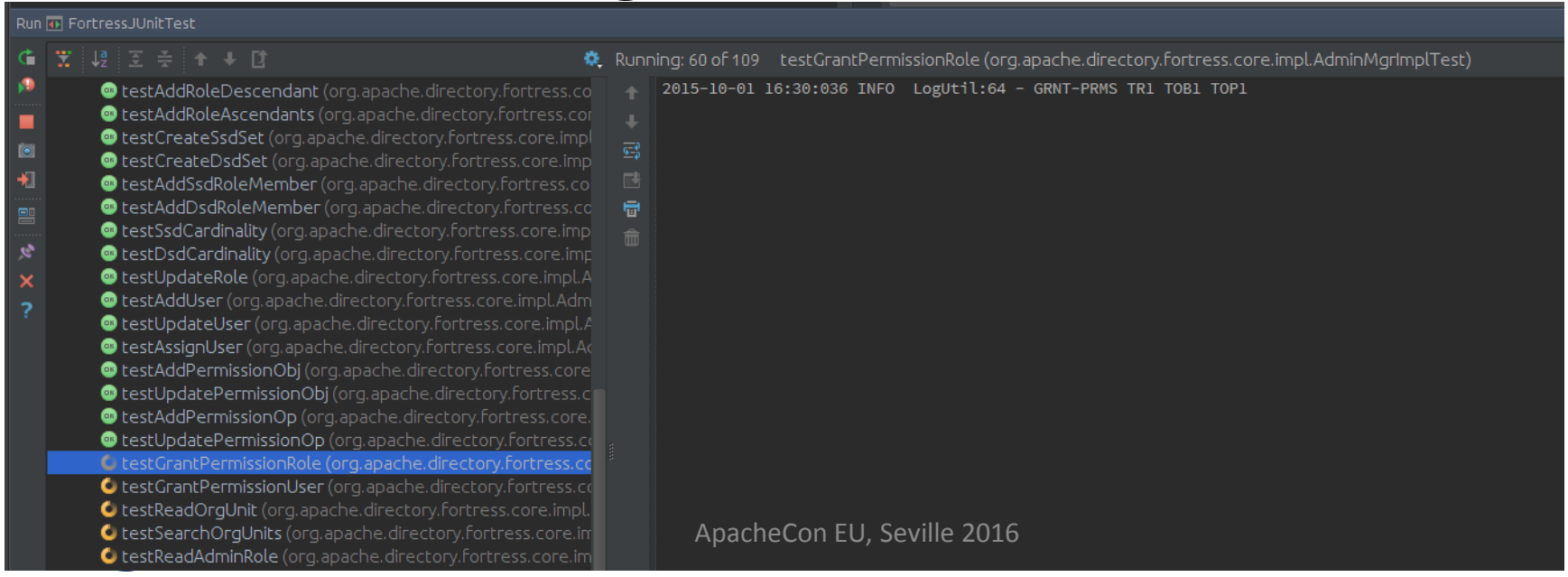
symas

Testing


FortressJUnitTest
<<Property>> -adminEnabled : boolean
-isFirstRun : boolean = org.apache.directory.fortress.core.impl.FortressJUnitTest.getFirstRun()
-setAdminEnabled(adminVal : boolean) : void
-getFirstRun() : boolean
+isFirstRun() : boolean
+setFirstRun(firstRun : boolean) : void
+suite() : Test
+FortressJUnitTest(name : String)
+testDisplayCounters() : void
+setUp() : void
+tearDown() : void
AccelMgrImplTest
+suite() : Test
+AccelMgrImplTest(name : String)
+setUp() : void
+tearDown() : void
+testGetSession() : void
+testGetToken() : void
+testCreateSession() : void
+createSessions(msg : String, uArray : String [][], rArray : String [][]) : void
+testCreateSessionWithRole() : void
+createSessionsWithRoles(msg : String, uArray : String [][], rArray : String [][]) : void
+testCheckAccess() : void
+checkAccess(msg : String, uArray : String [][], oArray : String [][], opArray : String [][], oArrayBad : String [][], ...
+testAddActiveRole() : void
+addActiveRoles(msg : String, uArray : String [][], rPosArray : String [][], rNegArray : String [][]) : void
+testDropActiveRole() : void
+dropActiveRoles(msg : String, uArray : String [][], rArray : String [][]) : void
DelegatedMgrImplTest
+DelegatedMgrImplTest(name : String)
+setUp() : void
+tearDown() : void
+suitex() : Test
+suite() : Test
+testAddAdminUser() : void
-loadAdminRequired(msg : String, rArray : String [][]) : boolean
+testDeleteAdminUser() : void
+testAssignAdminUser() : void
+assignAdminUsers(msg : String, uArray : String [][], rArray : String [][], isAdmin : boolean) : void
+testDeassignAdminUser() : void
~deassignAdminUsers(msg : String, uArray : String [][], rArray : String [][], isAdmin : boolean) : void
+assignAdminUserRole(msg : String, uArray : String [][], rArray : String [][], isAdmin : boolean) : void
~deassignAdminUserRole(msg : String, uArray : String [][], rArray : String [][], isAdmin : boolean) : void
+testAddUser() : void
+testDeleteUser() : void
+testAddPermission() : void
+testDeletePermission() : void
+testGrantPermissionRole() : void
+testRevokePermissionRole() : void
+testCheckAccess() : void
+checkAccess(msg : String, uArray : String [][], oArray : String [][], opArray : String [][], oArrayBad : String [][], ...
+testAddRole() : void
+addAdminRoles(msg : String, rArray : String [][], isAdmin : boolean) : void
+testDeleteRole() : void
-deleteAdminRoles(msg : String, rArray : String [][], isAdmin : boolean) : void
+testUpdateAdminRole() : void
+updateAdminRoles(msg : String, rArray : String [][], isAdmin : boolean) : void
+testCanAssignUser() : void
+testCanDeassignUser() : void
+canAssignUsers(msg : String, op : ASSIGN_OP, uraArray : String [][], uaArray : String [][], uArray : String [][], rA...
+testCanGrantPerm() : void
+testCanRevokePerm() : void
+canGrantPerms(msg : String, op : GRANT_OP, uraArray : String [][], uaArray : String [][], pArray : String [][], rAr...
ReviewMgrImplTest
+ReviewMgrImplTest(name : String)
+suite() : Test
+setUp() : void
+tearDown() : void
+suitex() : Test
+testReadPermissionOp() : void
+readPermissionOps(msg : String, pObjArray : String [][], pOpArray : String [][]) : void
+testFindPermissionOps() : void
+searchPermissionOps(msg : String, srchValue : String, pObjArray : String [][], pOpArray : String [][]) : void
+testReadPermissionObj() : void
+readPermissionObjs(msg : String, pArray : String [][]) : void
+testFindPermissionObjs() : void
+searchPermissionObjs(msg : String, srchValue : String, pArray : String [][]) : void
+testReadRole() : void
+teardownRequired(msg : String, rArray : String [][]) : boolean
+readRoles(msg : String, rArray : String [][]) : void
+testFindRoles() : void
+searchRoles(msg : String, srchValue : String, rArray : String [][]) : void
+testReadUser() : void
+readUsers(msg : String, uArray : String [][]) : void
+testFindUsers() : void
+searchUsers(msg : String, srchValue : String, uArray : String [][]) : void
+testAssignedRoles() : void
+assignedRoles(msg : String, uArray : String [][], rArray : String [][]) : void
+testAuthorizedRoles() : void
+authorizedRoles(msg : String, uArray : String [][]) : void
+testAuthorizedUsers() : void
+authorizedUsers(msg : String, rArray : String [][], uArray : String [][]) : void
+testAuthorizedUsersHier() : void
+authorizedUsersHier(msg : String, roleMap : Map) : void
+testRolePermissions() : void
+rolePermissions(msg : String, rArray : String [][], pObjArray : String [][], pOpArray : String [][]) : void
+testPermissionRoles() : void
+permissionRoles(msg : String, pObjArray : String [][], pOpArray : String [][], rArray : String [][]) : void
+testAuthorizedPermissionRoles() : void
+authorizedPermissionRoles(msg : String, pObjArray : String [][], pOpArray : String [][], rArray : String [][]) : void
+testPermissionUsers() : void
+permissionUsers(msg : String, pObjArray : String [][], pOpArray : String [][], uArray : String [][]) : void
+testAuthorizedPermissionUsers() : void
+authorizedPermissionUsers(msg : String, pObjArray : String [][], pOpArray : String [][], uArray : String [][]) : void
+testUserPermissions() : void
+userPermissions(msg : String, uArray : String [][], pObjArray : String [][], pOpArray : String [][]) : void
+testFindRoleNms() : void
+searchRolesNms(msg : String, srchValue : String, rArray : String [][]) : void
+testFindUserIds() : void
+searchUserIds(msg : String, srchValue : String, uArray : String [][]) : void
+testAuthorizedUserIds() : void
+assignedUserIds(msg : String, rArray : String [][], uArray : String [][]) : void
+testAssignedRoleNms() : void
+assignedRoleNms(msg : String, uArray : String [][], rArray : String [][]) : void
+testFindSsdSets() : void
+searchSsdSets(msg : String, srchValue : String, sArray : String [][]) : void
+testFindDsdSets() : void
+searchDsdSets(msg : String, srchValue : String, sArray : String [][]) : void
+getManagedReviewMgr() : ReviewMgr
AccessMgrImplTest
+suite() : Test
+AccessMgrImplTest(name : String)
+setUp() : void
+tearDown() : void
+testGetSession() : void
+testGetToken() : void
+testGetUserId() : void
+getUserIds(msg : String, uArray : String [][]) : void
+testGetUser() : void
+getUsers(msg : String, uArray : String [][]) : void
AuditMgrImplTest
-CLS_NM : String = AuditMgrImplTest.class.getName()
-LOG : Logger = LoggerFactory.getLogger(CLS_NM)
-disabled : Map = org.apache.directory.fortress.core.impl.AuditMgrImplTest.loadAuditMap()
-adminSess : Session = null
+AuditMgrImplTest(name : String)
+setUp() : void
+tearDown() : void
+suite() : Test
+testSearchAdminMods() : void
-loadAuditMap() : Map<String, String>
-isAudit(objName : String, opName : String) : boolean
-searchAdminMods(msg : String, uArray : String [][], oArray : String [][], opArray : String [][]) : v...
+testSearchMods() : void
-searchMods(msg : String, uArray : String [][]) : void
+testSearchAuthZs() : void
-searchAuthZs(msg : String, uArray : String [][], oArray : String [][], opArray : String [][], failedO...
+testGetAuthZs() : void
-getAuthZs(msg : String, uArray : String [][]) : void
+testSearchAuthNInvalid() : void
-searchAuthNInvalid(msg : String) : void
+testSearchBinds() : void
-searchBinds(msg : String, uArray : String [][]) : void
-getManagedAuditMgr() : AuditMgr
AdminMgrImplTest
+AdminMgrImplTest(name : String)
+setUp() : void
+tearDown() : void
+suite() : Test
+testAddUser() : void
+addUsers(msg : String, uArray : String [][], isAdmin : boolean) : void
+testDeleteUser() : void
+deleteUsers(msg : String, uArray : String [][], force : boolean, isAdmin : boolean) : void
+testForceDeleteUser() : void
+testUpdateUser() : void
+updateUsers(msg : String, uArray : String [][]) : void
+testChangePassword() : void
~changePasswords(msg : String, uOldArray : String [][], uNewArray : String [][]) : void
+testLockUserAccount() : void
~lockUsers(msg : String, uArray : String [][]) : void
+testUnlockUserAccount() : void
~unlockUsers(msg : String, uArray : String [][]) : void
+testResetPassword() : void
~resetPasswords(msg : String, uArray : String [][]) : void
+testAddRole() : void
+addRoles(msg : String, rArray : String [][]) : void
+testDeleteRole() : void
+deleteRoles(msg : String, rArray : String [][]) : void
+testUpdateRole() : void
+updateRoles(msg : String, rArray : String [][]) : void
+testAddRoleDescendant() : void
+addRoleDescendant(msg : String, rArray : String [][]) : void
+testDelRoleDescendant() : void
-delRoleDescendant(msg : String, rArray : String [][]) : void
+testAddRoleAscendants() : void
+addRoleAscendant(msg : String, rArray : String [][]) : void
+testDelRoleAscendant() : void
-delRoleAscendant(msg : String, rArray : String [][]) : void
+testAddRoleInheritance() : void
+addInheritedRoles(msg : String, rArray : String [][]) : void
+testDeleteRoleInheritance() : void
+deleteInheritedRoles(msg : String, rArray : String [][]) : void
+testCreateSsdSet() : void
+createSsdSet(msg : String, sArray : String [][]) : void
+testCreateDsdSet() : void
+createDsdSet(msg : String, sArray : String [][]) : void
+testDeleteSsdSet() : void

Integration Tests

- Full test coverage of the APIs
- Positive and Negative Use Cases
- No manual testing



Automated Testing

https://builds.apache.org/view/All/job/dir-fortress-core-docker-test/org.apache.directory.fortress$fortress-core/lastCompletedBuild/console

Jenkins

Jenkins › All › dir-fortress-core-docker-test › Apache Fortress Core › #131

Back to Project

Status

Console Output

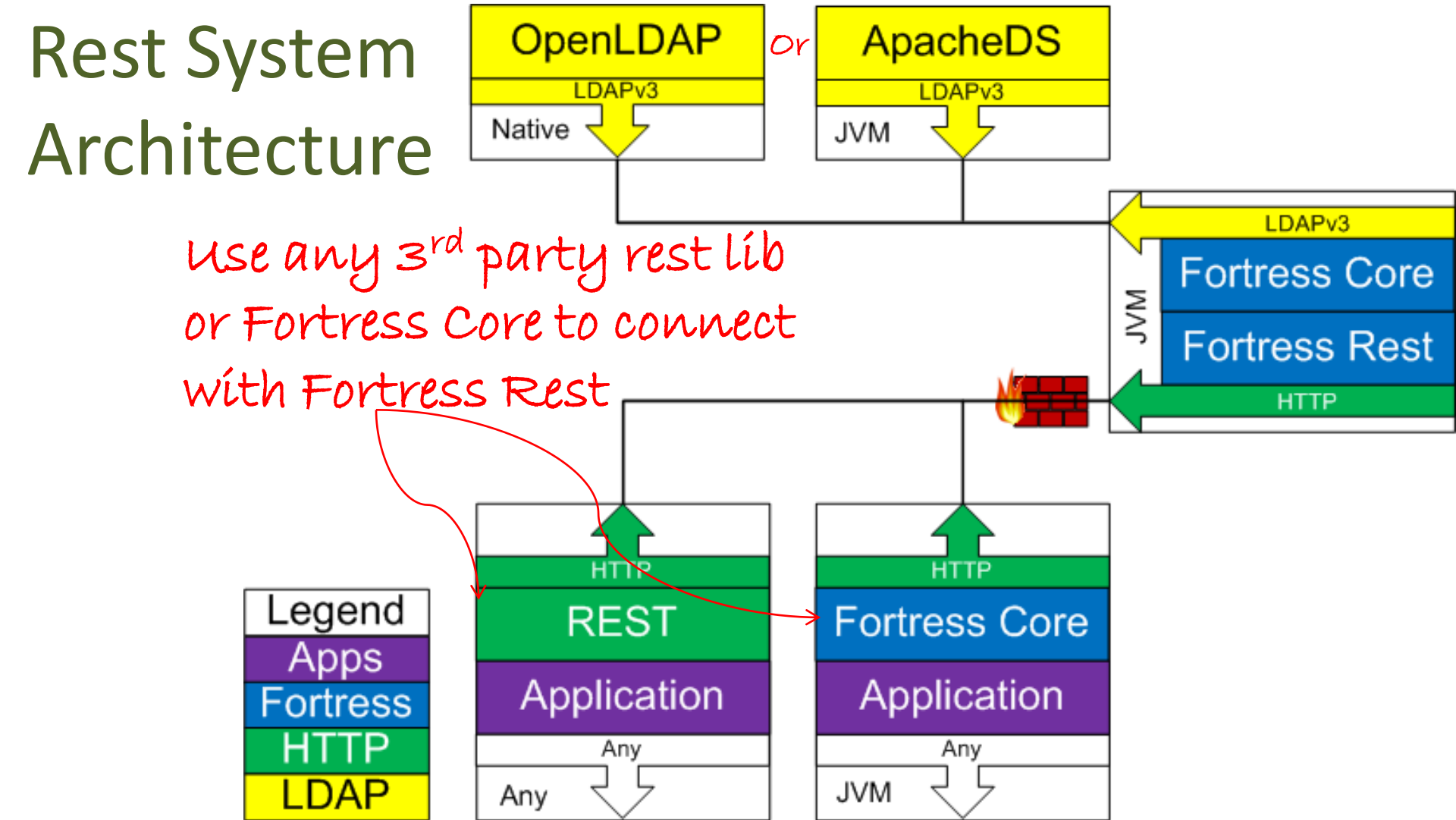
```
Tests run: 113, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 224.5 sec - in org.apache.directory.fortress.core.impl.FortressJUnitTest

Results :

Tests run: 113, Failures: 0, Errors: 0, Skipped: 0

[JENKINS] Recording test results
log4j:WARN No appenders could be found for logger (org.apache.commons.beanutils.converters.BooleanConverter).
log4j:WARN Please initialize the log4j system properly.
[INFO]
[INFO] --- maven-antrun-plugin:1.8:run (default) @ fortress-core ---
[INFO] Executing tasks

fortress-load:
[INFO] Executed tasks
[JENKINS] Archiving disabled
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 4:12.665s
[INFO] Finished at: Wed Sep 16 05:34:14 UTC 2015
[INFO] Final Memory: 25M/200M
[INFO] ------------------------------------------------------------------------
[JENKINS] Archiving disabled
Waiting for Jenkins to finish collecting data
```

https://builds.apache.org/view/All/job/dir-fortress-core-docker-test/org.apache.directory.fortress$fortress-core/

Core Benchmarks

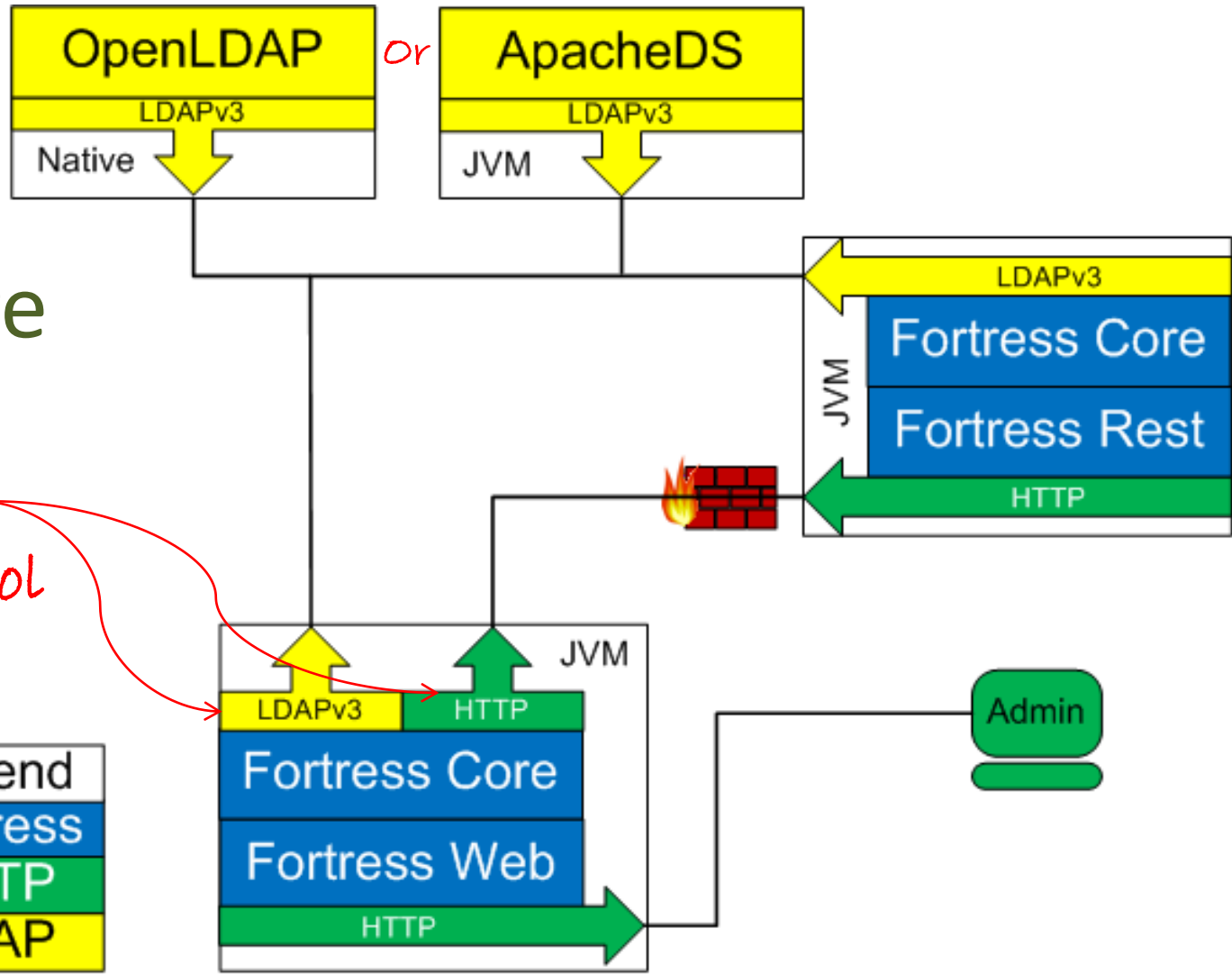
- Jmeter tests for various scenarios
  - Fortress createSession, checkAccess
  - Accelerator createSession, checkAccess

Rest System Architecture



Web System Architecture

Option to use either HTTP or LDAPV3 protocol



symas

Demo

Menu

1. Learn about some
   - Basic integration - RBAC0 - wicket-sample
   - Intermediate - RBAC1 - role-engineering-sample
   - Advanced - RBAC2 & 3 - apache-fortress-demo
2. Testing on
   - Fortress Web - manual or selenium
   - " " Rest - junit
   - " " Console - ad-hoc
   - " " Command Line Interface - sys-admin stuff
3. Have fun with
   - Multi-tenancy & / or Benchmarking - setting up, running, verifying

Apache Fortress Demo

- Three Pages and Three Customers
- One role for every page to customer combo
- Users may be assigned to one or more roles
- At most one role may be activated

Pages	Customer 123	Customer 456	Customer 789
Page One	PAGE1_123	PAGE1_456	PAGE1_789
Page Two	PAGE2_123	PAGE2_456	PAGE2_789
Page Three	PAGE3_123	PAGE3_456	PAGE3_789

Demo 1 Usage Policy

- Both super and power users may access everything.
- But power users are limited to one role activation at a time.
- Super users are not restricted.

Super & Power Users	Customer 123	Customer 456	Customer 789
Page1	True	True	True
Page2	True	True	True
Page3	True	True	True

User123	Customer 123	Customer 456	Customer 789
Page1	True	False	False
Page2	True	False	False
Page3	True	False	False

User1	Customer 123	Customer 456	Customer 789
Page1	True	True	True
Page2	False	False	False
Page3	False	False	False

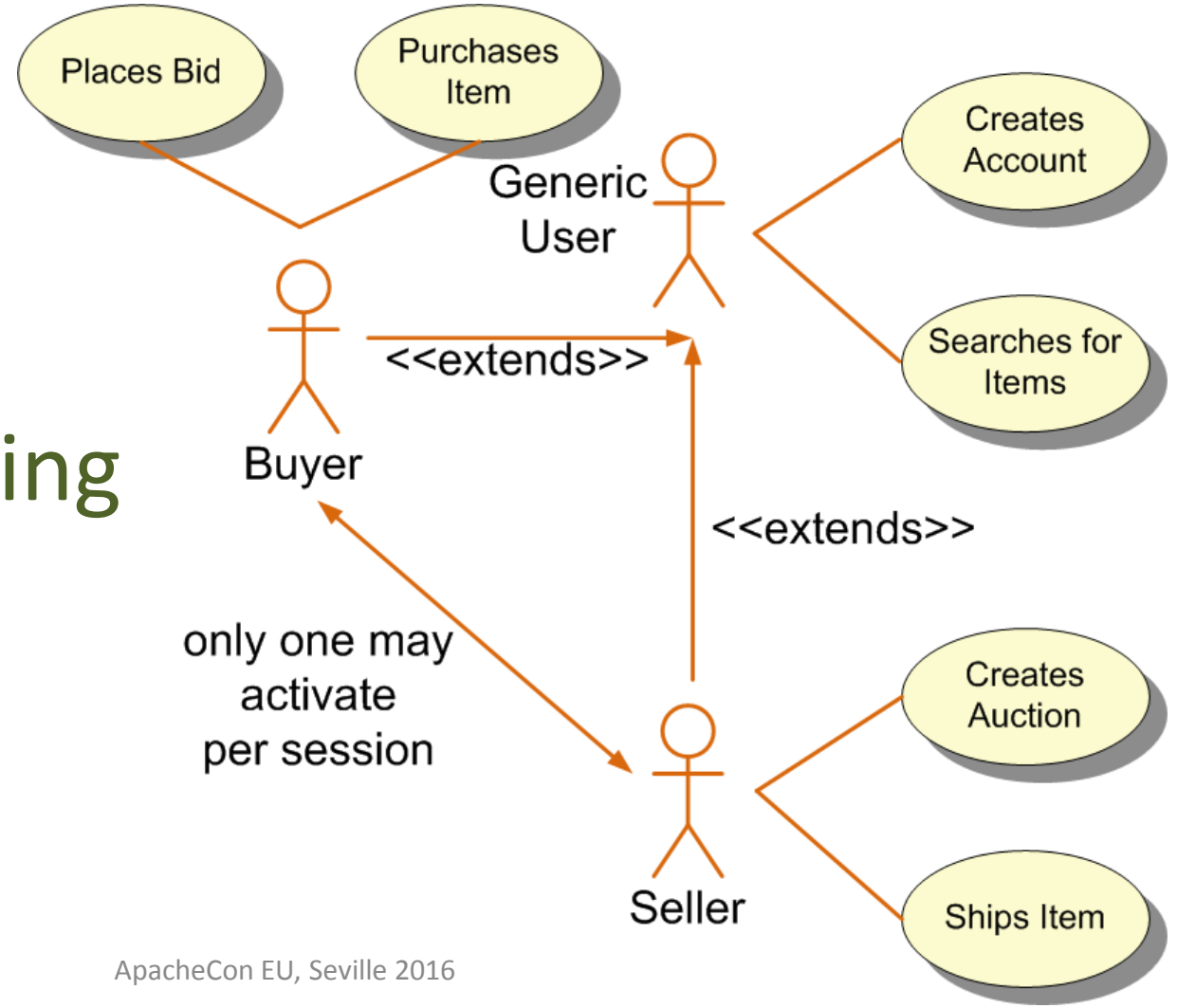
User1_123	Customer 123	Customer 456	Customer 789
Page1	True	False	False
Page2	False	False	False
Page3	False	False	False

Apache Fortress Demo

- https://github.com/shawnmckinney/apache-fortress-demo

User-tic-tac-toe	Customer 123	Customer 456	Customer 789
Page1	False	True	True
Page2	True	False	False
Page3	True	False	False

Demo 2
Role Engineering Sample

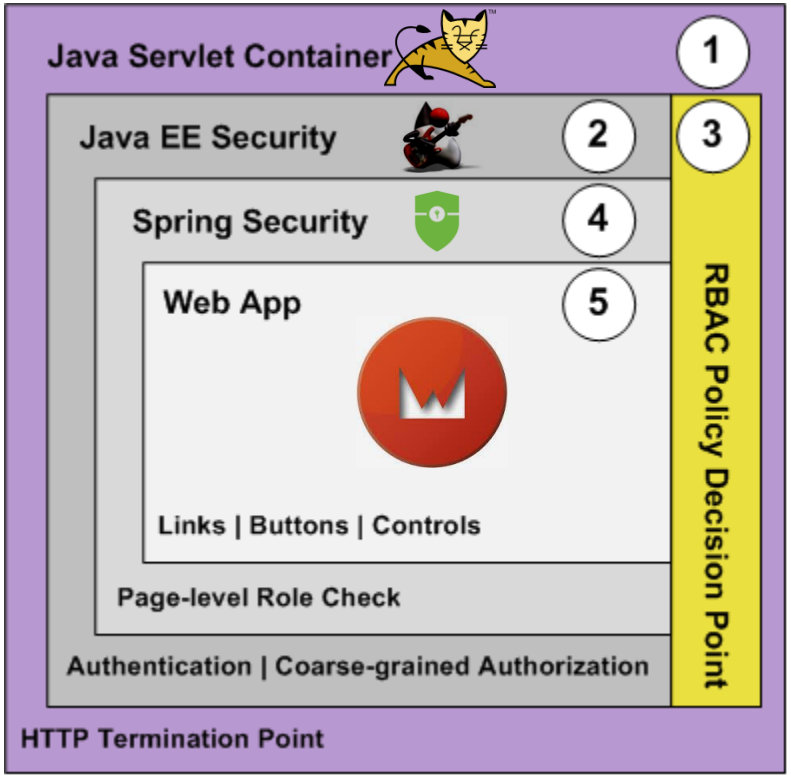


Demo 2 Role Engineering Sample

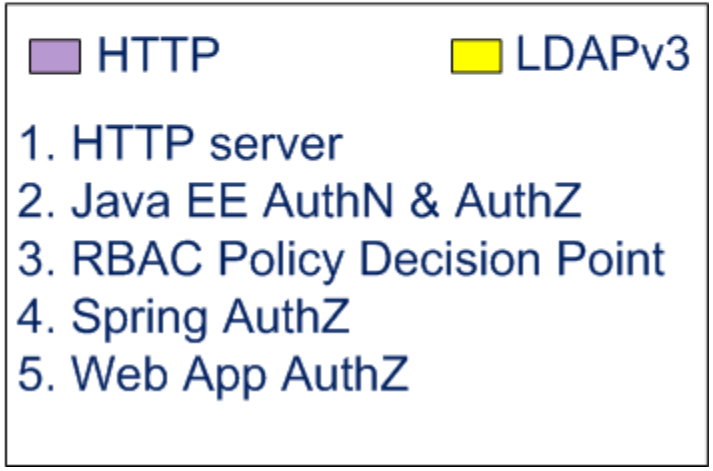
1. Java EE Authentication and Authorization
2. Spring Page-level Authorization
3. RBAC Permission Checks
   - Links
   - Buttons
4. Other RBAC Controls
   - Dynamic Separation of Duty
   - Role Switcher

Declarative

Demo 2 Role Engineering Sample



https://github.com/shawnmckinney/role-engineering-sample



Demo 2 Role Engineering Sample

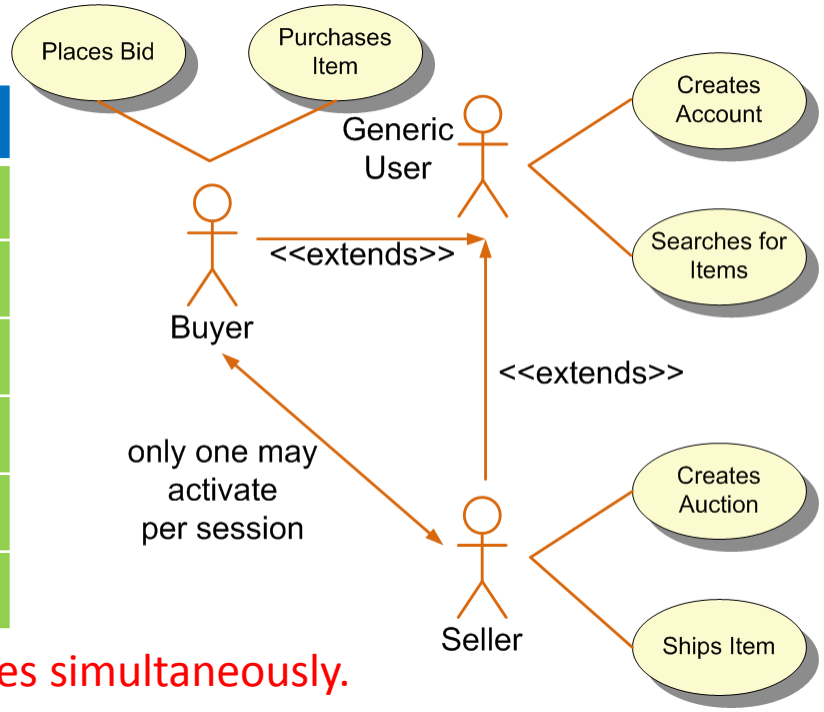
- Two pages
- Each has buttons controlled by RBAC Permissions.
- One Role per page.

User to Role	Buyers Page	Sellers Page
ssmith	True	False
jtaylor	False	True
Johndoe*	True	True

* DSD constraint limits user from activating both roles simultaneously.

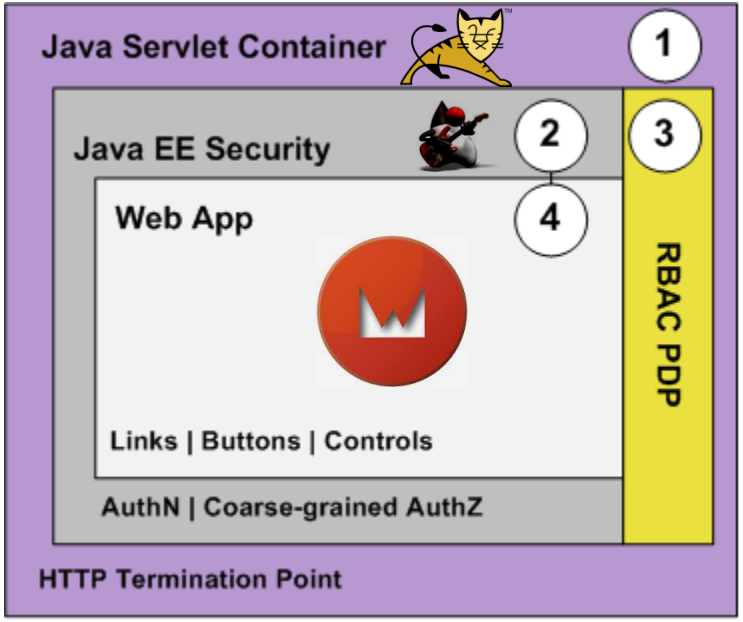
Demo 2 Role Engineering Sample

		Buyer	Seller	Both
	Permission	**ssmith**	**rtaylor**	**johndoe***
1	Item.bid	True	False	True
2	Item.purchase	True	False	True
3	Item.ship	False	True	True
4	Item.search	True	True	True
5	Account.create	True	True	True
6	Auction.create	False	True	True



* DSD constraint limits user from activating both roles simultaneously.

Demo 3 Web Sample

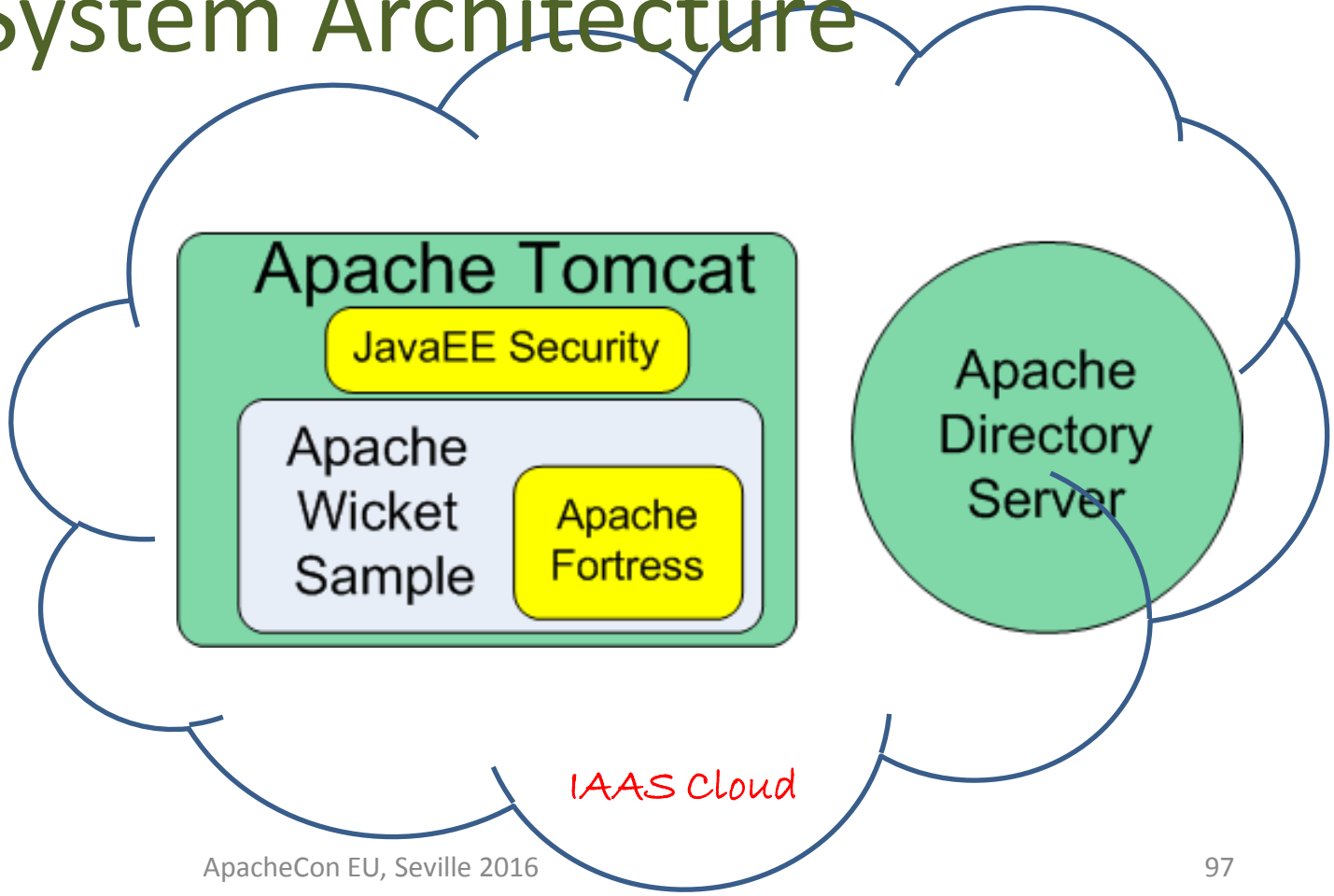


https://github.com/shawnmckinney/wicket-sample

HTTP
LDAPv3

1. HTTP server
2. Java EE AuthN & AuthZ
3. RBAC Policy Decision Point
4. Web App AuthZ

Demo 3 System Architecture



Security Layers with Wicket Sample

1. JSSE ← Confidentiality and Itegrity
2. Java EE Security ← authN and coarse-grained authZ
3. Web App Framework ← fine-grained authZ

Add Web Framework Security

```
add(
   new SecureIndicatingAjaxButton( "Page1", "Add" )
   @Override
   protected void onSubmit( ... )
   {
      if( checkAccess( customerNumber ) )
      {
        // do something here:
      }
      else
      {
         target.appendJavaScript( ";alert('Unauthorized');" );
      }
   }
   });
```

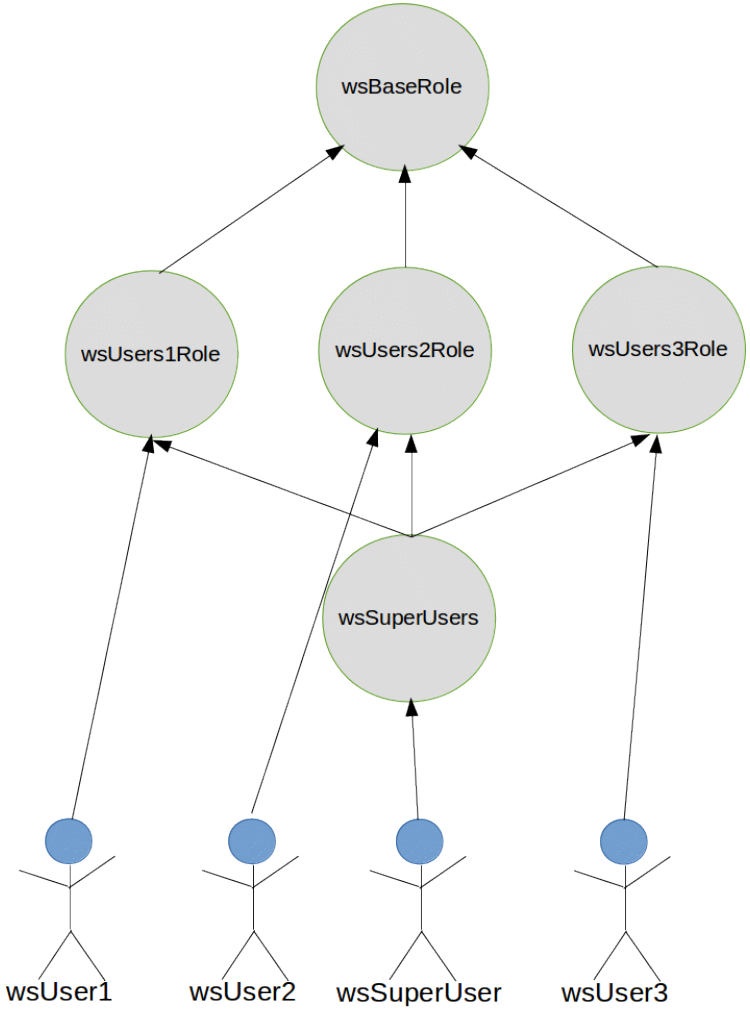
fine-grained authorization (programmatic)

Demo 3 Web Sample

Github link to

[Wicket Sample Policy File]

User	Page1	Page2	Page3
wsUser1	True	False	False
wsUser2	False	True	False
wsUser3	False	False	True
wsSuperUser	True	True	True

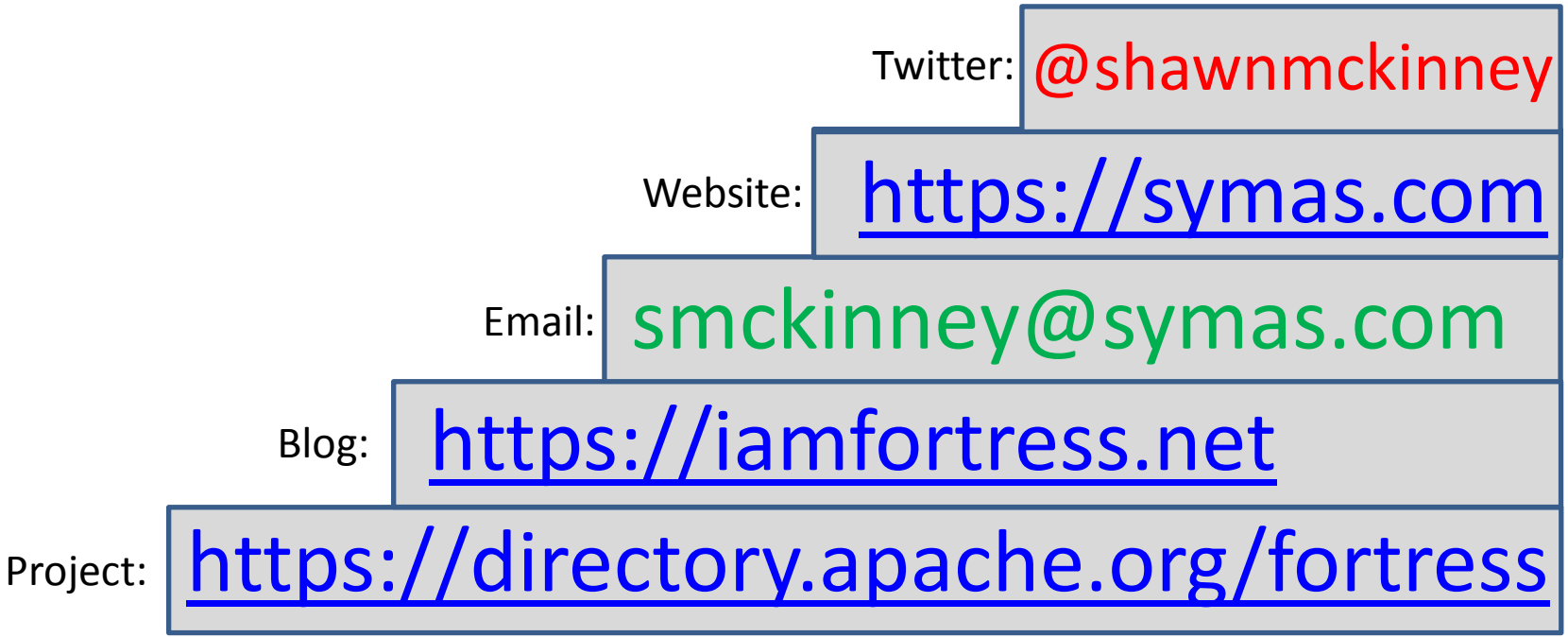


Tutorial Links

In Gitub:

1. Wicket Sample:
   - https://github.com/shawnmckinney/wicket-sample

2. End-to-End Security Demo:
   - https://github.com/shawnmckinney/apache-fortress-demo


Twitter: @shawnmckinney
Website: https://symas.com
Email: smckinney@symas.com
Blog: https://iamfortress.net
Project: https://directory.apache.org/fortress