~~RBAC-Enable Your Java Web Applications with~~

How I Built an IAM System using Java and Apache Directory Fortress

Shawn McKinney

October 29, 2015

JavaOne

San Francisco

symas

Session Objectives

Learn about:

- ✓ Identity and Access Management Specifications (What)
- ✓ System Design Considerations (How)
- ✓ Apache Directory Fortress (Why)

symas

Introductions

Shawn McKinney

- symas Systems Architect
- PMC Apache Directory Project
- OpenLDAP™ PROJECT Engineering Team

Session Agenda

- Examine Specs & Requirements for IAM
- Examine Designs for an IAM System
- Intro to Apache Fortress
  - Project Details
  - Components
  - Future


Image from: HTTP://EVENTS.LINUXFOUNDATION.ORG/EVENTS/APACHECON-NORTH-AMERICA

symas

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
- Common elements from the platform
- Accepted development practices

Specs & Requirements

What do we Build?

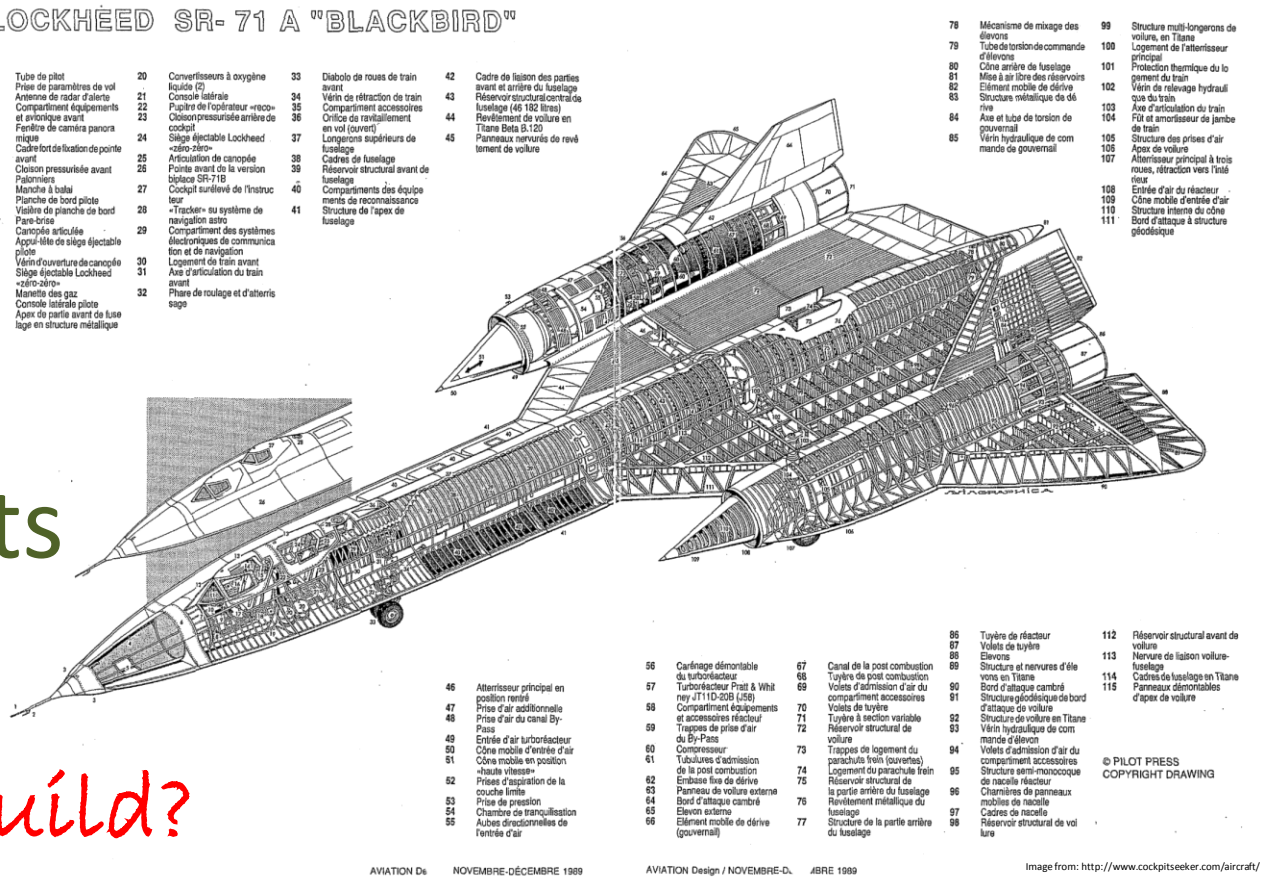


Image from: http://www.cockpitseeker.com/aircraft/

High-Level System Requirements

- **Security** - Access control checking inside common containers like JavaEE and Spring.
- **Authentication** - May use both simple and complex protocols.
- **Authorization** - Use standards-based access control methodologies.
- **Administration** - Policy usage is managed by delegation policies.
- **Audit** - Record of operations inside persistent data store.
- **Service-based SLA** - Maintain service level agreements for security, performance, and reliability.

Why Use Functional Specifications?

- Saves the trouble (and risk) of deciding ‘what’ to do.
- Instead we get to focus on ‘how’ to do it.
- Difference between being handed a blank sheet of paper or a coloring book.

Which Functional Specifications

- Protocols Must Be Standards-Based:
  - Role-Based Access Control (RBAC) - ANSI INCITS 359
  - ~~Attribute-Based Access Control (ABAC)~~ 
  - IETF Password Policies (Draft)
  - ARBAC02 Delegated Administration Model
  - Must cooperate with JavaEE and Spring Security, OAuth2, SAML 2.0, OpenID Connect, UMA, etc.

Use INCITS 494?

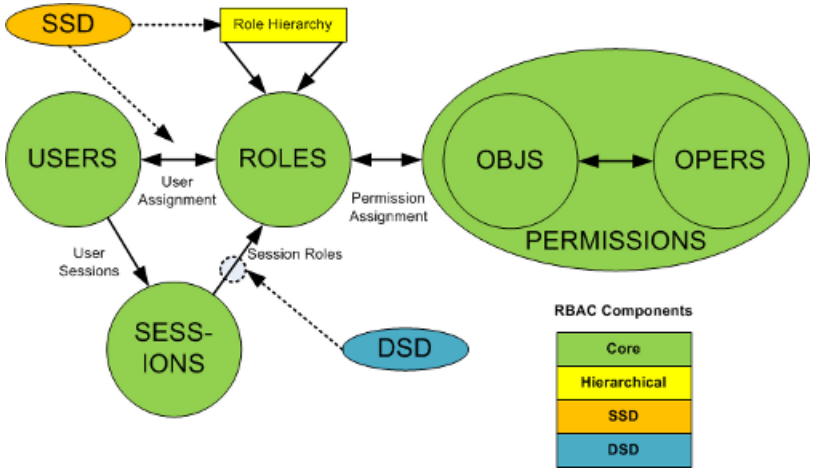
Access Control Requirements

- Policy Database that can be centralized and federated
- Fine-grained permissions
- Common functional and object models

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

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.

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

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]

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

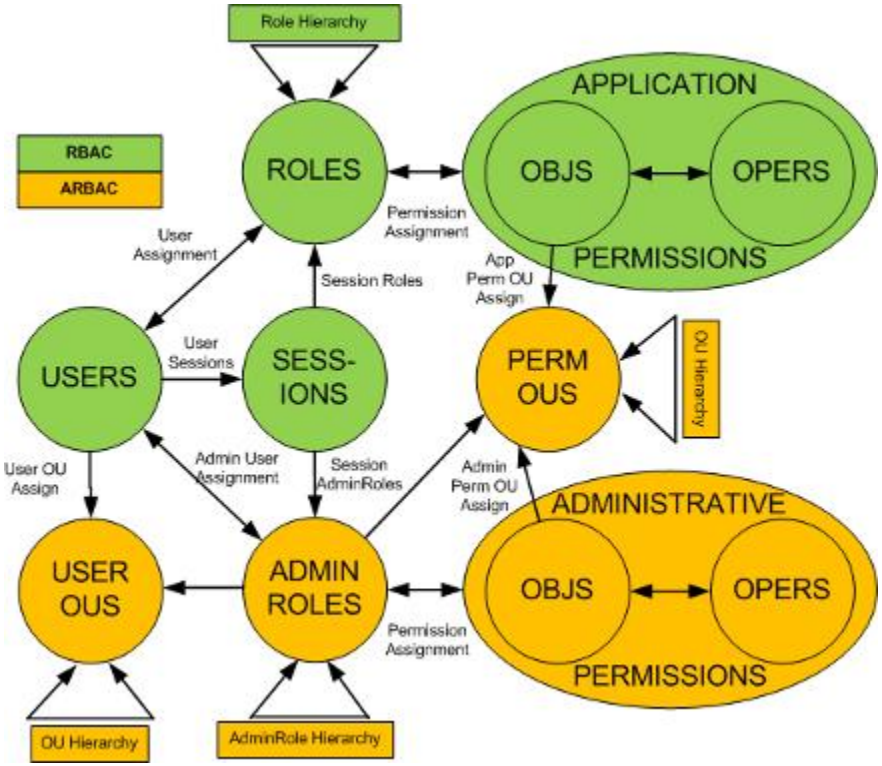
Administration Requirements

- 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



ARBAC02 Functional Model

Three standard interfaces:

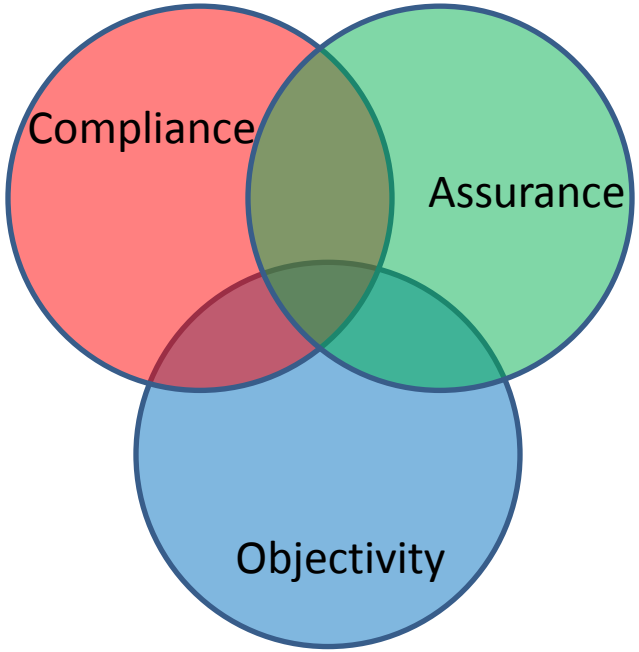
1. Delegated Administrative – CRUD
2. Delegated Review – policy interrogation
3. Delegated System – policy enforcement

Other Key Requirements

- Centralized Audit Trail and Reporting API
- Password Policy Control
- Lockout Procedures based on Time & Date

Audit

- System
- Principal Identity
- Date
- Resource
- Resource Identity
- Operation
- Result



Password Policies

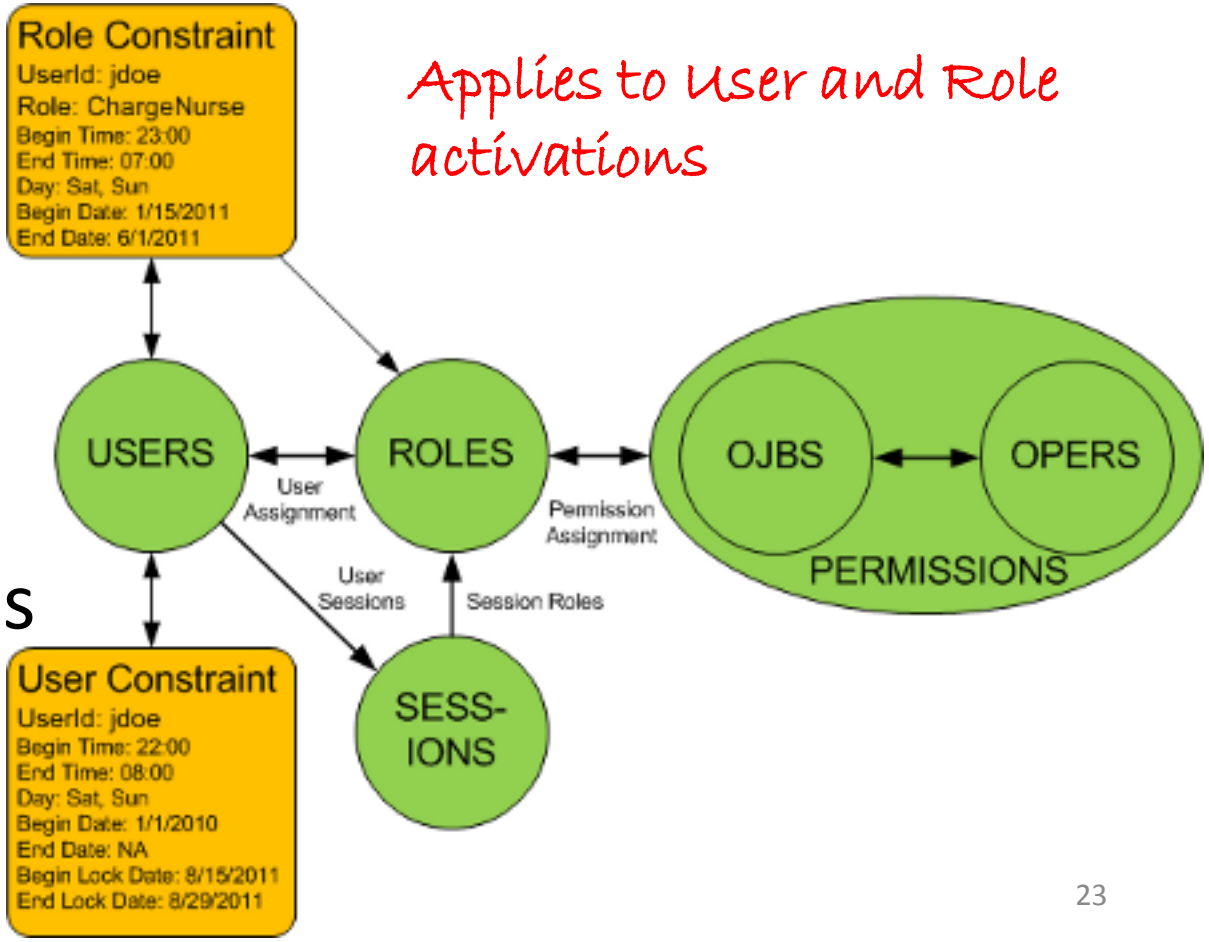
1. A configurable limit on failed authentication attempts.
2. A counter to track the number of failed authentication attempts.
3. A time frame in which the limit of consecutive failed authentication attempts.
4. The action to be taken when the limit is reached.
5. An amount of time the account is locked (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

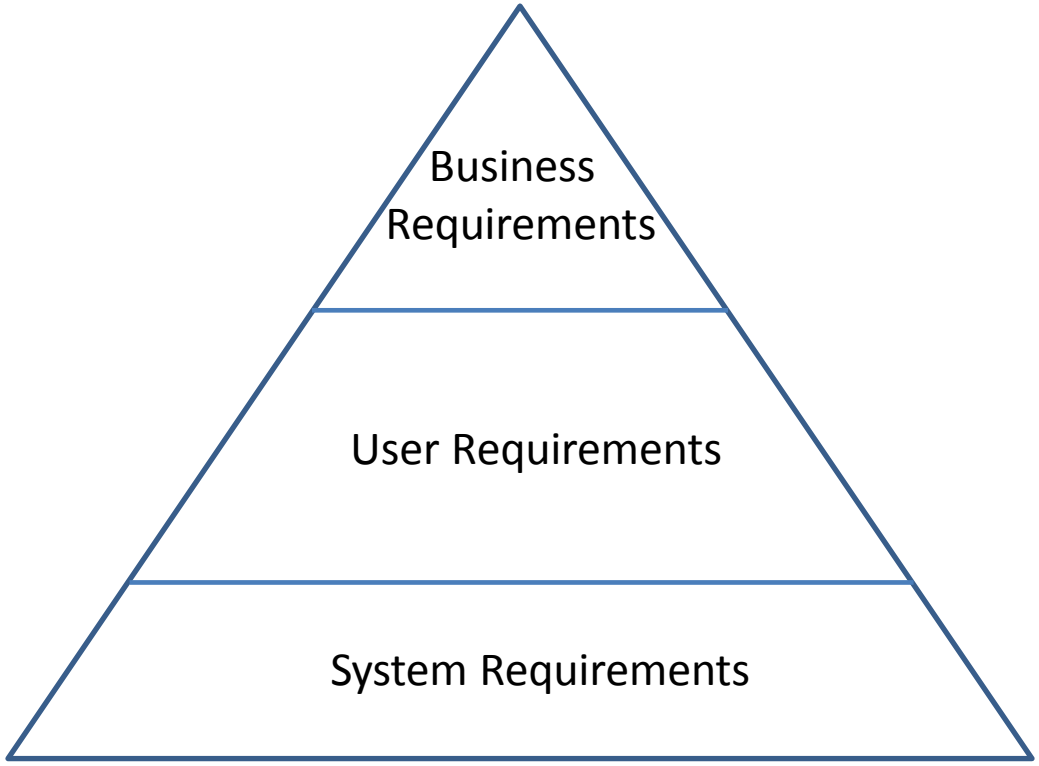
Temporal Constraints

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



Non-Functional Requirements

- Fault Tolerant
- Highly Available
- Multitenant
- Full Audit Trail
- Highly Performant



Non-Functional Requirements

- Optimized for Performance
- Updates
  - > 10,000 TPS
- Searches/Binds/ Permission Checks
  - > 100,000 TPS
  - Latency < 1ms

Design Overview

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, data access, logging, multitenancy, session storage and replication and performance.
  - Not to mention testing, packaging, documentation, integration.
  - But, Strive to Keep It Simple Stupid (KISS).
  - Reuse, don't reinvent.

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

How to Store and Retrieve the Data?

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

LDAP Directory for Physical Storage

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

LDAP Data Access Options

- JNDI – Not Comprehensive Enough
- Netscape LDAP API – Obsolete
- UnboundID Java LDAP API – License Concerns
- Apache LDAP API - Perfect

Apache Directory LDAP API

LDAP API 1.0
- Home
- News

Downloads
- Version 1.0.0-M31 New
- Older versions

Getting Started
- Vision
- Java API
- Groovy API

Documentation
- Five minutes tutorial
- User Guide
- JavaDocs
- Cross-Reference
- Developer Guide

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.

Download Apache LDAP API 1.0.0-M31

Free for Linux, Mac OS X & Windows

News

Apache Directory LDAP API 1.0.0-M31 released — *posted on July 5th, 2015*

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

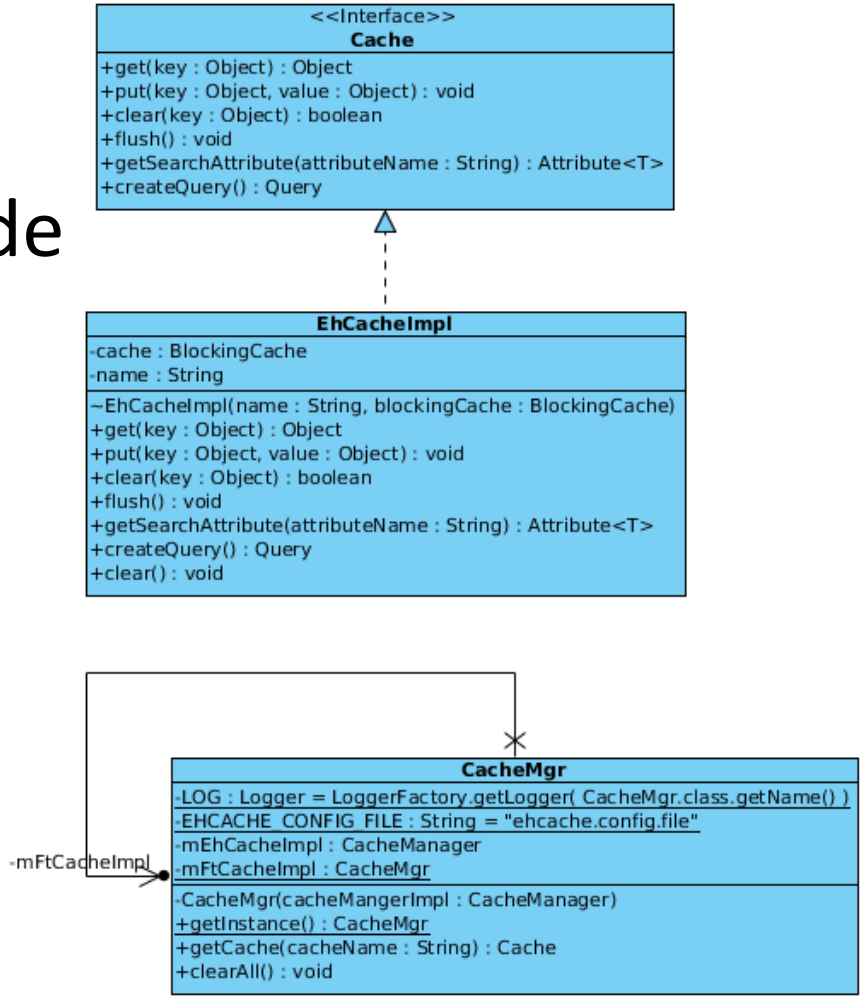
To Cache or Not

Need it for:

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

Use Ehcache

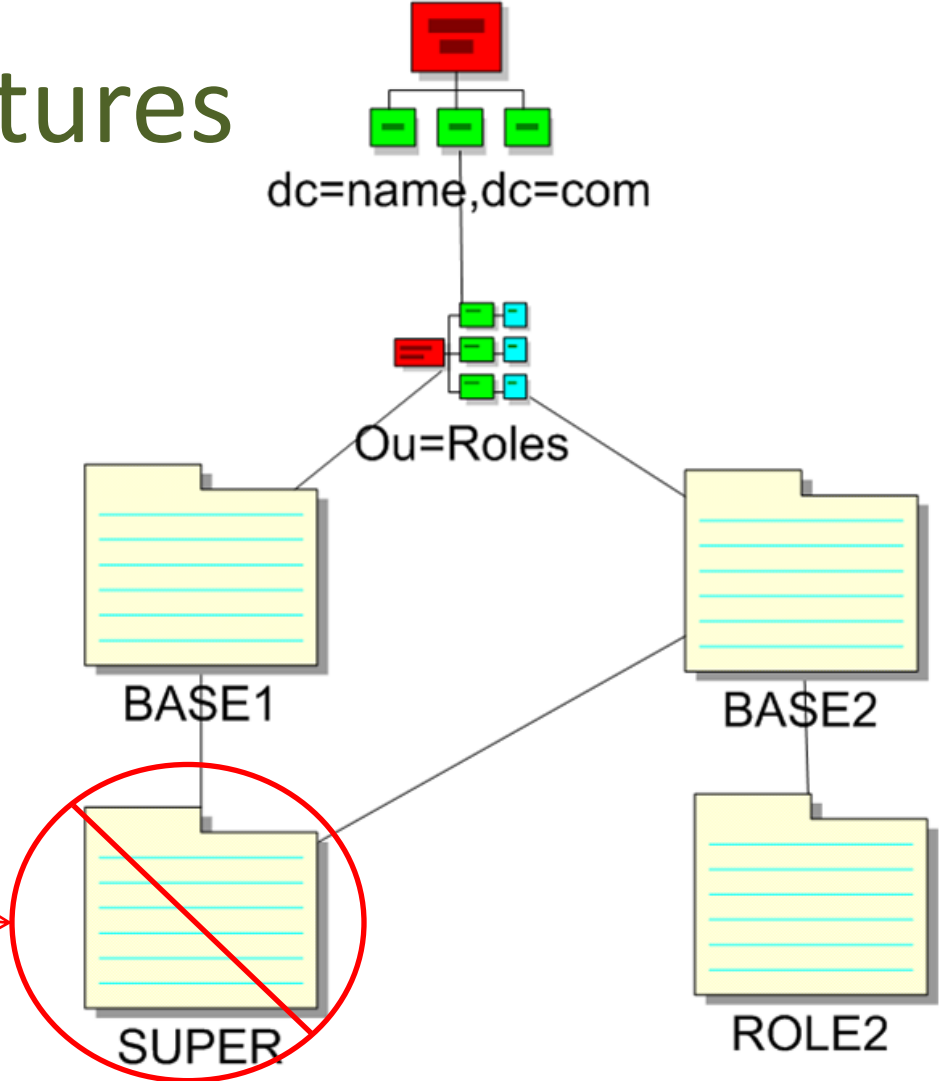
Hide it behind a Facade



Nested Data Structures

General Role Hierarchy

- Multiple Inheritance
- More flexible
- Graph doesn't map onto LDAP hierarchical structural capabilities



Can't do this with LDAP

symas

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

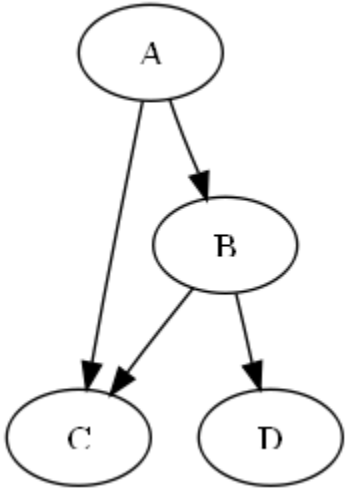


Image from: https://code.google.com/p/fluentdot/wiki/DemoSimpleDirectedGraph

Persistent or Transient Session?

Each has its own benefits…

Transient

1. Less processing on server
2. Less data stored
3. More flexibility in terms of attributes managed

Persistent

1. Less data to transfer over wire
2. Less processing on client
3. Supports session timeout and concurrency controls

Use Either

Core

- Transient Sessions

Accelerator

- Persistent Sessions

What About Firewalls?

(LDAPv3 protocol isn't always allowed)

- Core API can transmit using either LDAPv3 or HTTP.

Audit

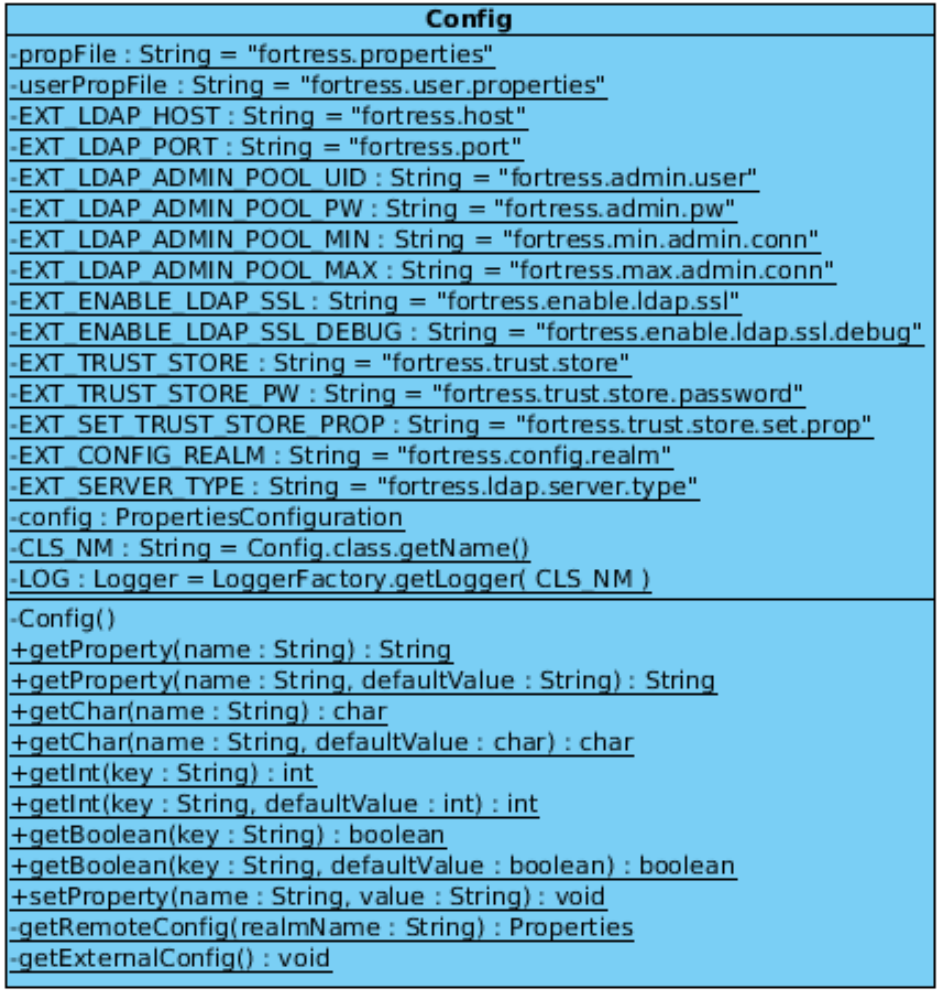
- Use OpenLDAP slapo access log to record
  - Authentication
  - Check Access
  - Admin and Review calls
  - APIs to interrogate (AuditMgr)

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

Object Model Questions

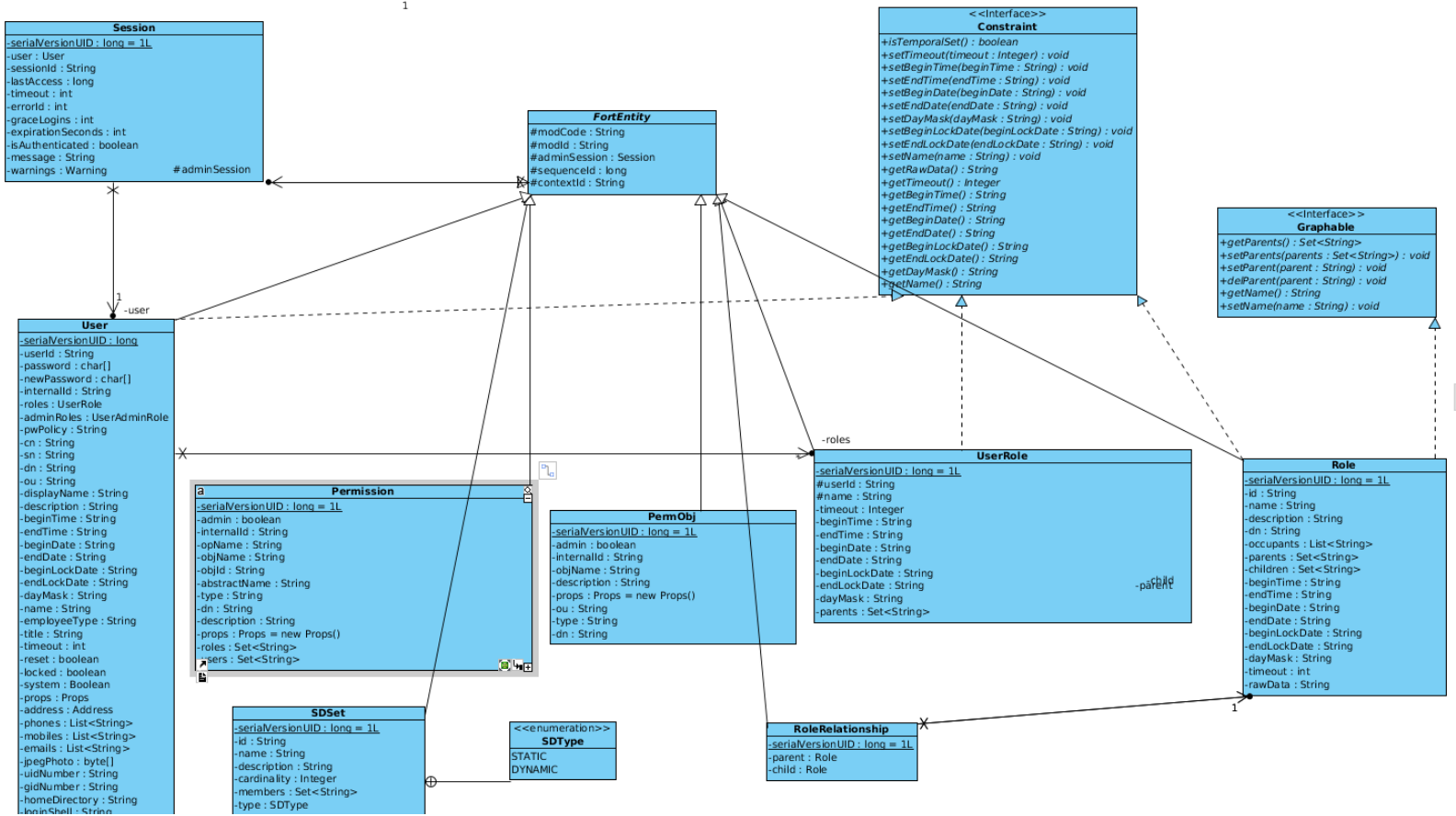
- What specific object model do I use?
- How do I represent the physical data model?
- How do I represent the logical data model?
- How do I support multitenancy?

RBAC Object Model

Six basic elements:

1. User – human or machine entity
2. Role – a job function within an organization
3. Object – maps to system resources
4. Operation – executable image of program
5. Permission – approval to perform an Operation on one or more Objects
6. Session – contains set of activated roles for User

Logical RBAC Model

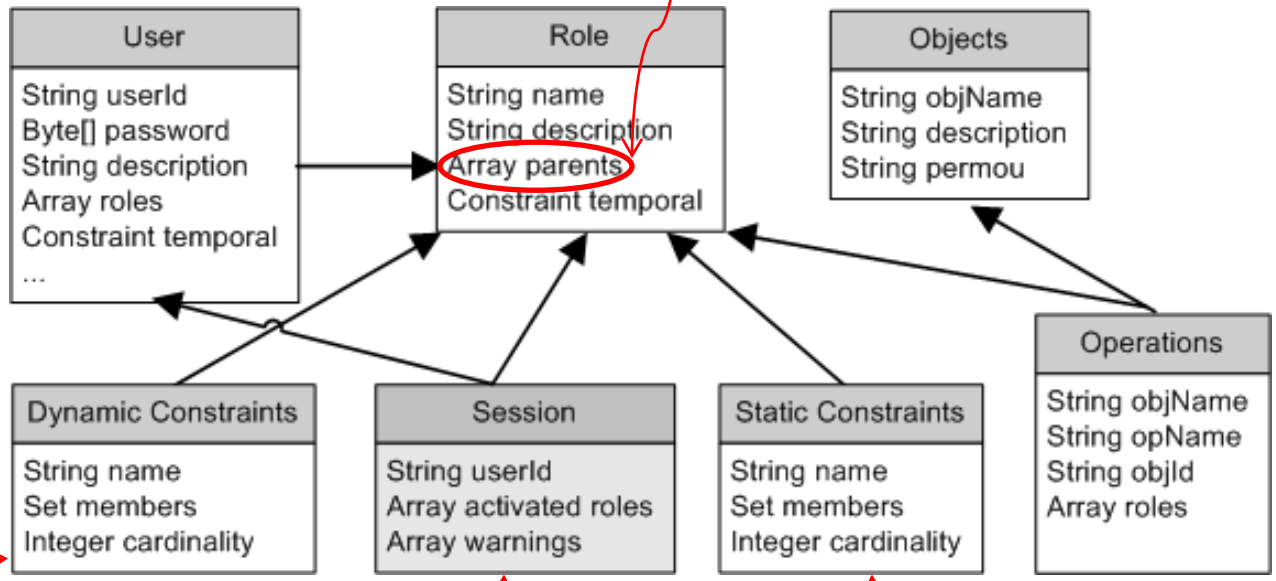


[directory-fortress-core.git] / src / main / java / org / apache / directory / fortress / core / model /

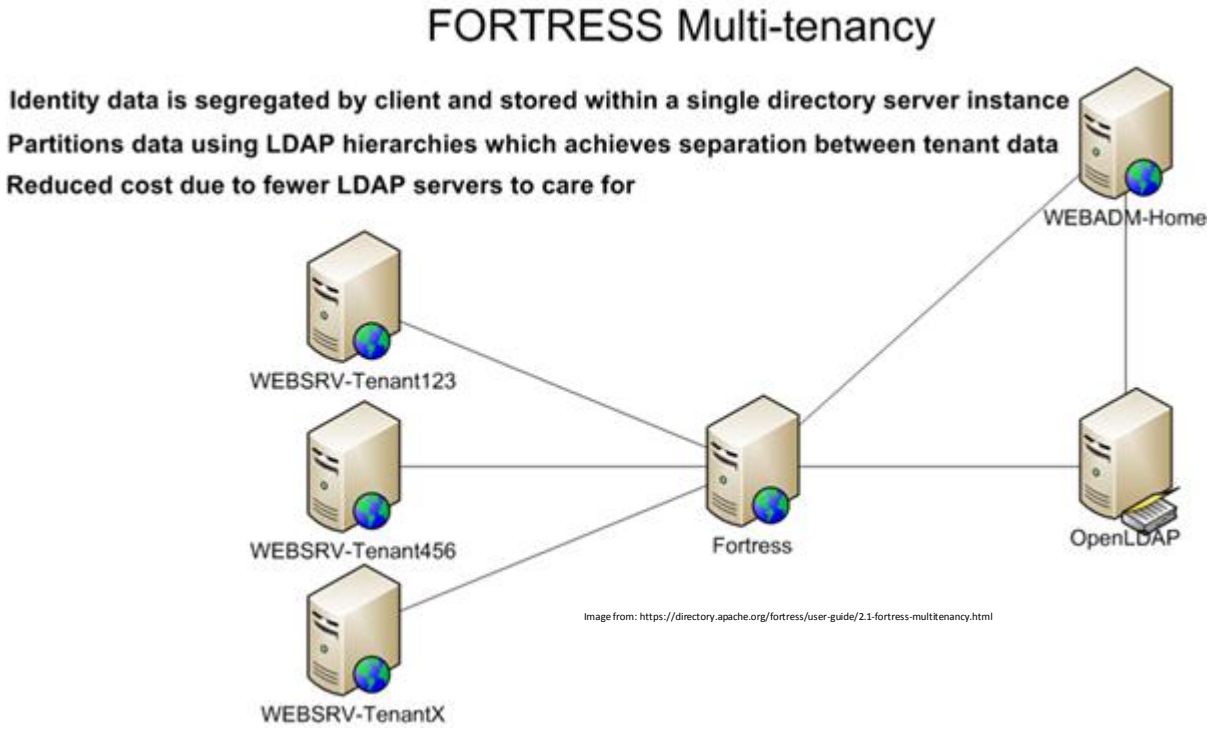
Physical RBAC Model

- Users
- Roles
- Permissions
- Constraints

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



Multitenancy



Multitenancy Defined

Multitenancy

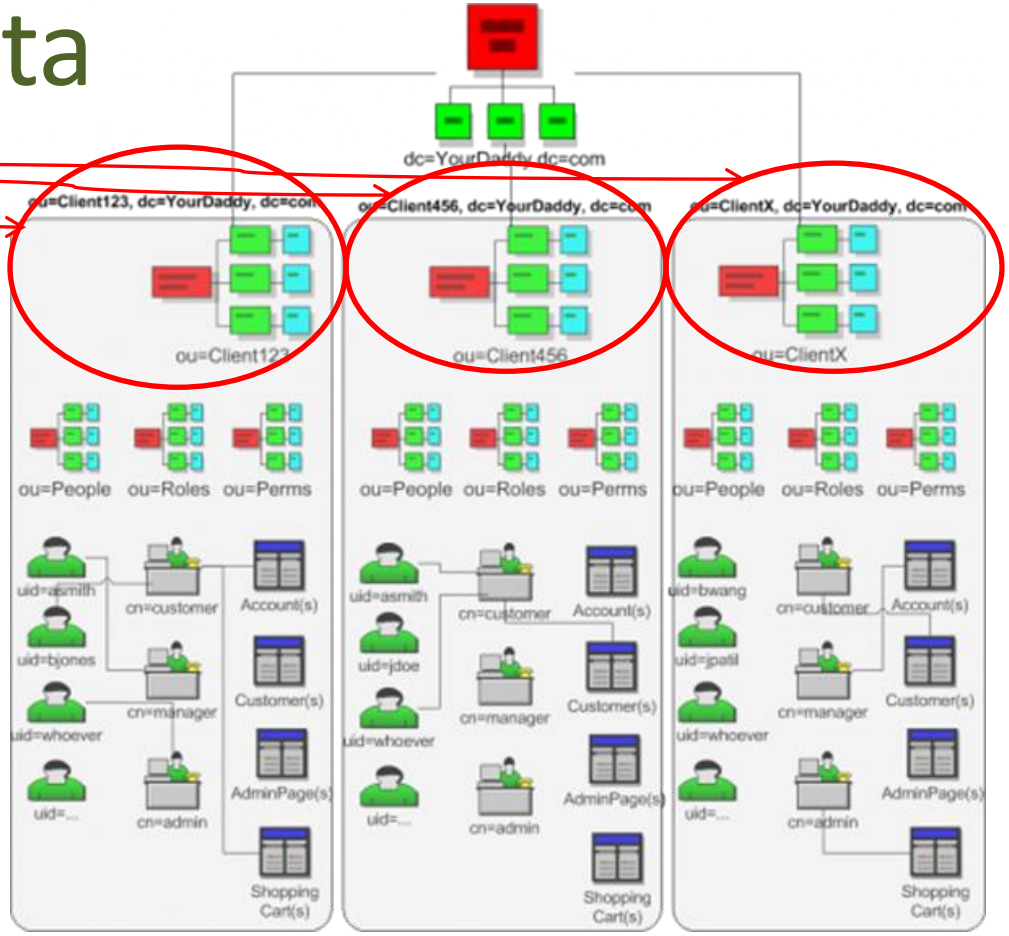
From Wikipedia, the free encyclopedia

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 Data

- Leverage LDAP's natural affinity to partition data by client organization.
- Each tenant has its own complete copy of DIT segregated by organizational unit
- Reduced cost due to fewer servers to maintain



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");
```

Multitenant Object Model

- Client's id of tenant is referenced inside transient contextId field.
- Every entity in system will extend this class.
- Clients are read-only.

```
public abstract class SecurityEntity
{
      …
      @XmlTransient protected String contextId;
}
```

Security Model Questions

- How to secure apps running in container in a vendor neutral way?
- How not to impede normal Web app flows with security?
- How to propagate identity across multiple layers in the system and applications?

Project Implementation

Intro to Apache Fortress

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

symas

What Gets Built

1. Java APIs ← Software Dev Kit
2. Web Tier (HTML) ← Admin GUI
3. Service Tier (HTTP) ← RESTful Interface
4. Policy Enforcement Component for Java EE

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 IAM solution of this type.
  - *Practice makes perfect*

Project Dev Processes

Need a sponsor that provides:

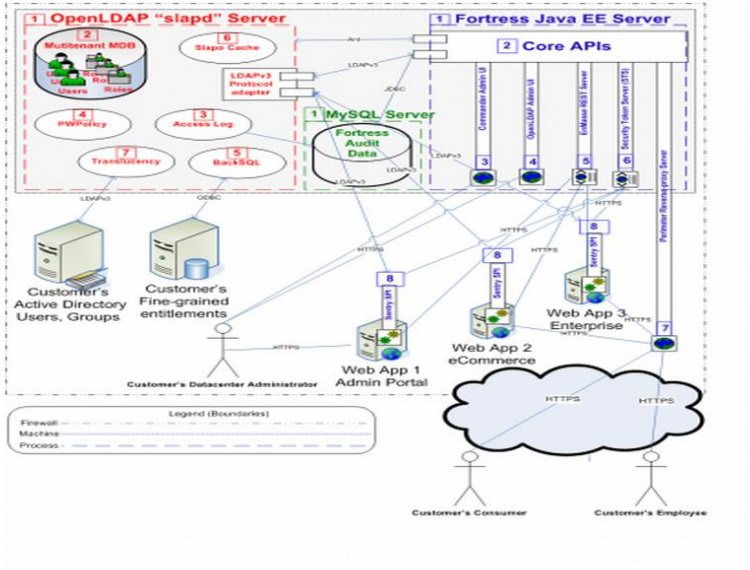
- Source Code Management
- Bug Tracking
- Mailing Lists
- Build Servers
- Binary Code Distribution
- Automated Testing

Apache Directory Fortress

Apache Fortress™

OSS IAM Java SDK

Apache Fortress™, or simply *Fortress*, is a standards based and Open Source Identity Access Management Java SDK for LDAP v3 compliant systems. Fortress is 100% compliant with RBAC, ARBAC02 and IETF's password policy draft



News

Apache Fortress Core 1.0-RC40 released ! *posted on April 15th, 2015*

symas

https://directory.apache.org/fortress/

Apache Fortress Overview

- Sub-project of Apache Directory
- Java Based Identity and Access Management
- Permission-based Access Control Model (RBAC)
- Four Components:
  - Core – Java APIs + utilities
  - Realm – Java EE policy enforcement
  - Web – Administrative UI
  - Rest – APIs over HTTP interface

symas

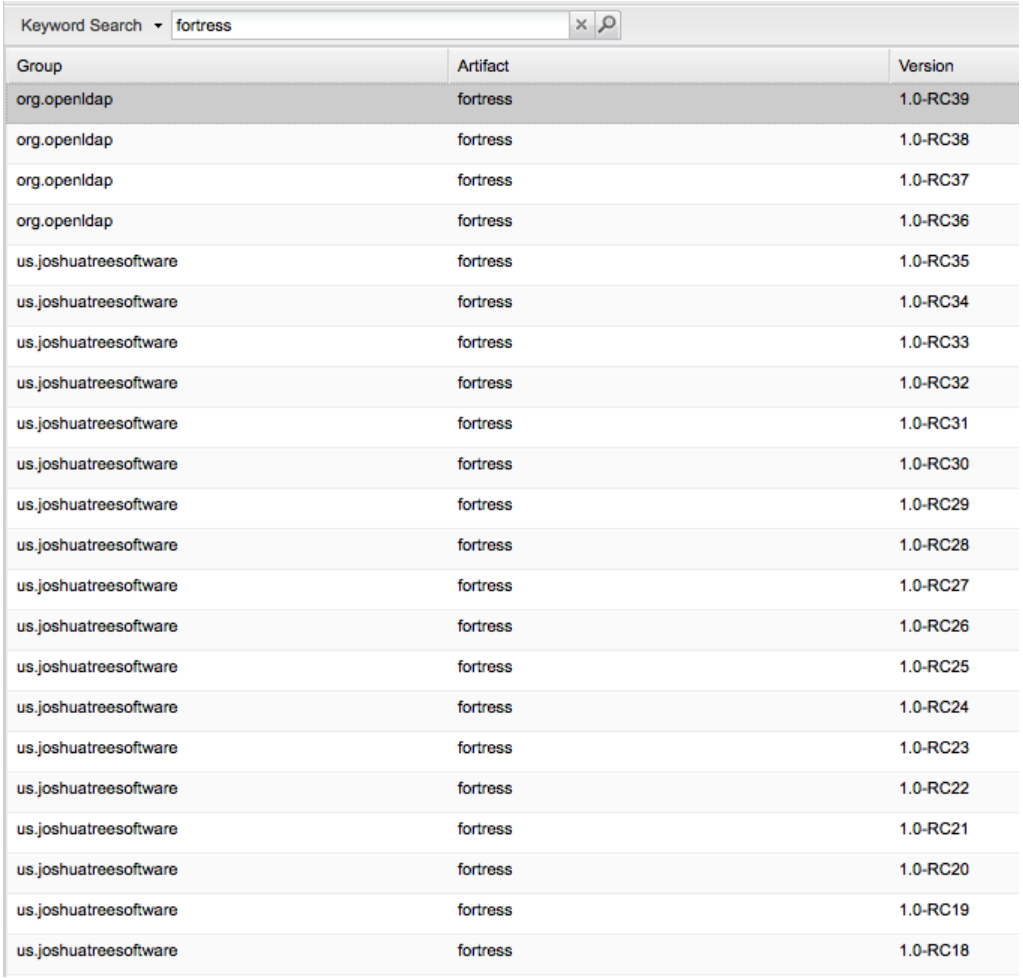
Apache Fortress Project History

- Core & Realm released in '11 to OpenLDAP Project
- Rest component in '12 to OpenLDAP
- Web component in '13 to OpenLDAP
- Moved to Apache Directory project in '14

History (cont)

23 Releases

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

https://www.openhub.net/p/apache-fortress

In a Nutshell, Apache Fortress...

... has had 1,112 commits made by 6 contributors
representing 109,491 lines of code

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

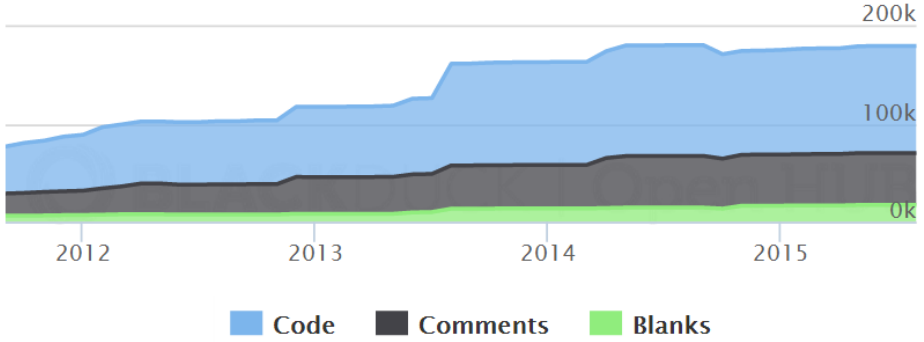
... has a codebase with a long source history
maintained by a average size development team
with increasing Y-O-Y commits

... took an estimated 28 years of effort (COCOMO model)
starting with its first commit in September, 2011
ending with its most recent commit 4 months ago

Languages

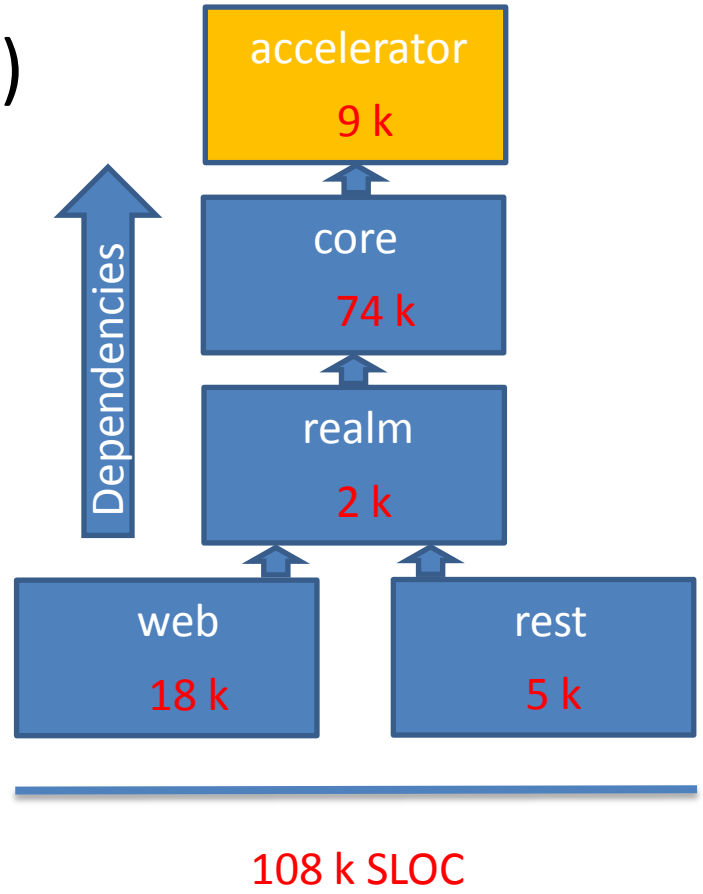
Java	86%	XML	7%
HTML	6%	4 Other	1%

Lines of Code



Source Lines of Code

- Accelerator – 9 k (client-side)
- Core – 74 k (36 k is for test)
- Realm – 2 k
- Web – 18 k
- Rest – 5 k

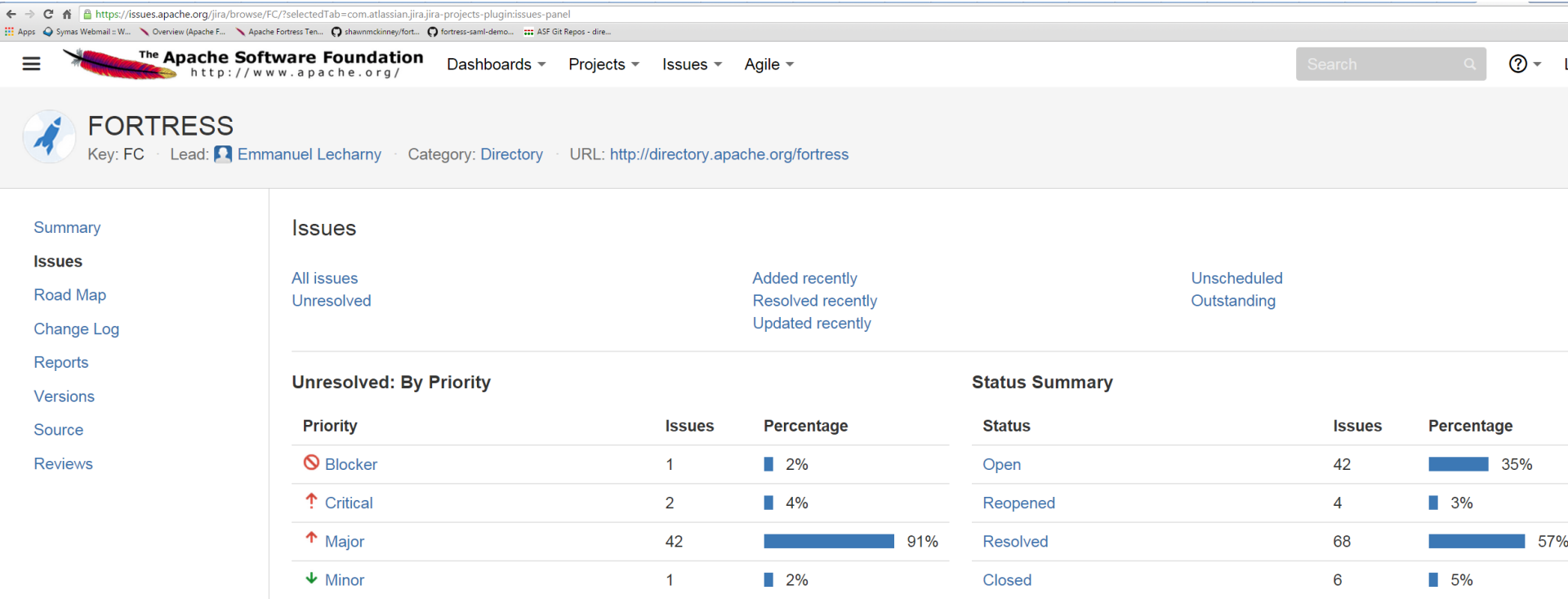


Project Releases

https://directory.apache.org/fortress/downloads.html

symas

JIRA Bug Tracking



https://issues.apache.org/jira/browse/FC

Jenkins 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

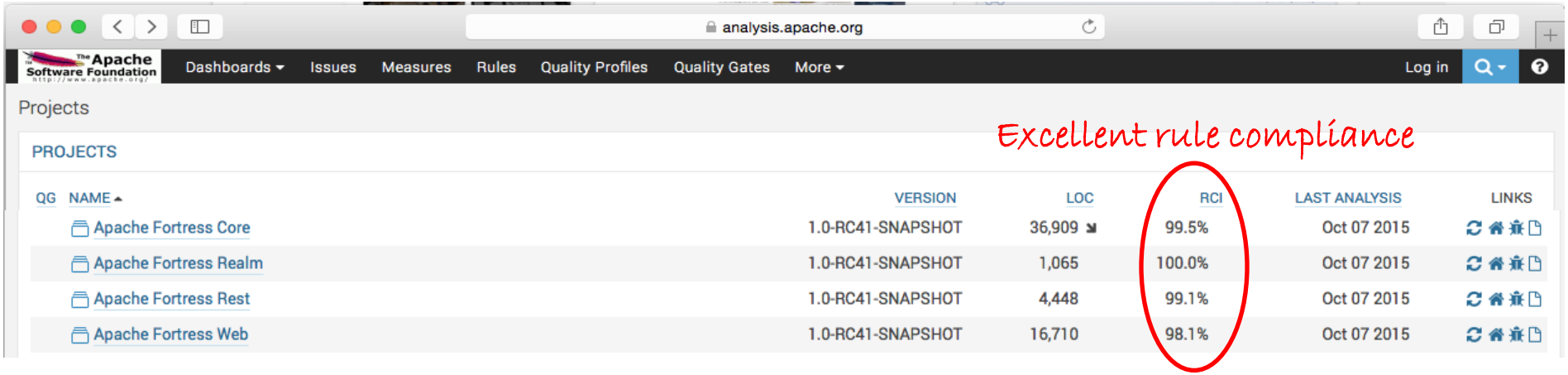
Status

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/

Static Code Analysis

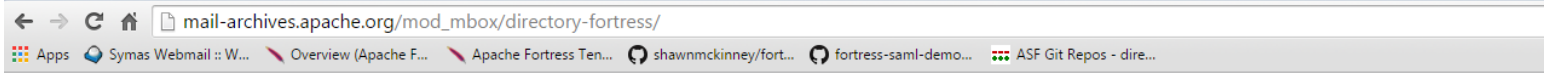


SonarQube code scans run nightly:

- 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



Mailing list archives: fortress@directory.apache.org

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

List information	
Writing to the list	fortress@directory.apache.org
Subscription address	fortress-subscribe@directory.apache.org
Digest subscription address	fortress-digest-subscribe@directory.apache.org
Unsubscription addresses	fortress-unsubscribe@directory.apache.org
Getting help with the list	fortress-help@directory.apache.org
Feeds:	Atom 1.0

Year 2015		
Sep 2015	Browse	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

Low activity

symas

Components

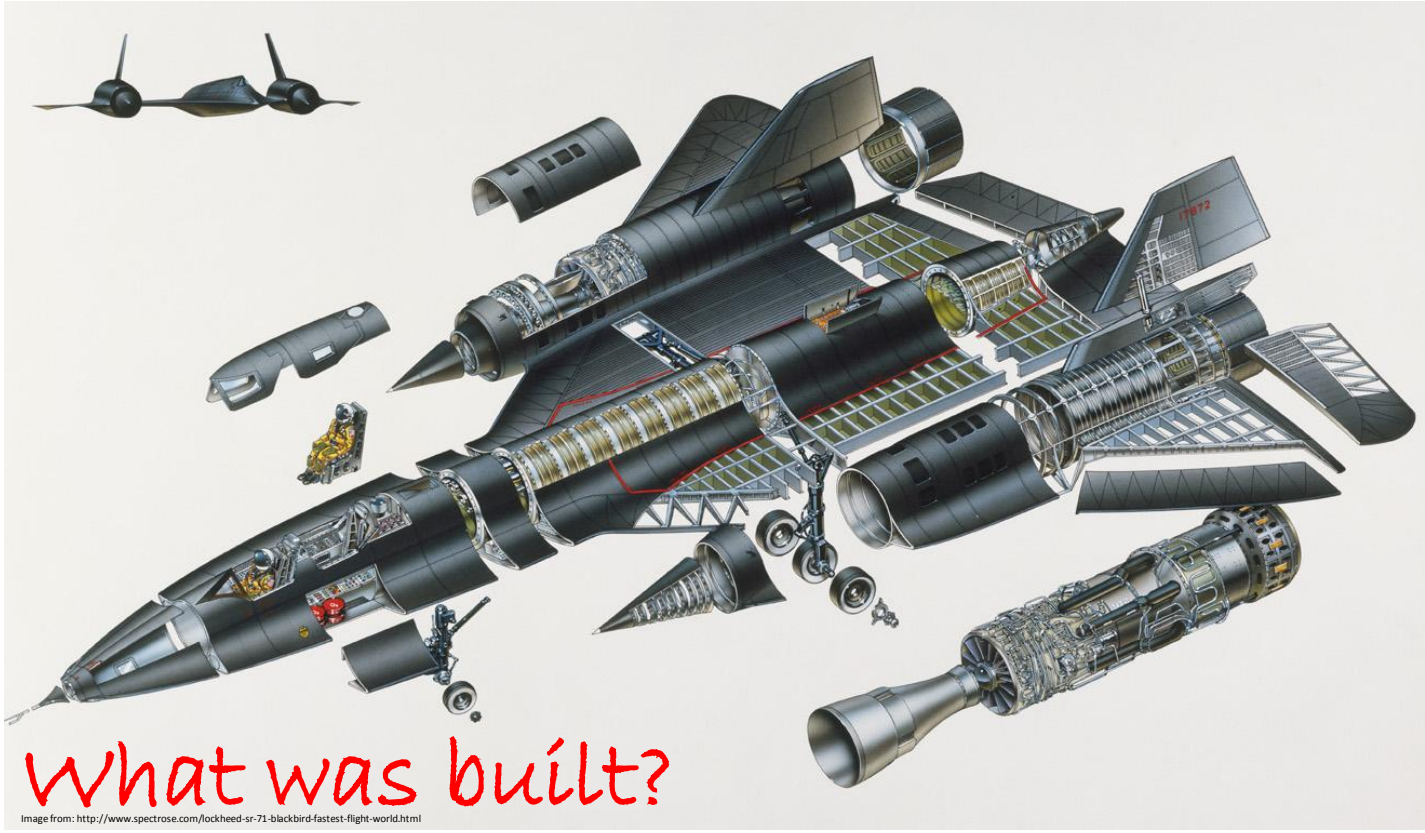


Image from: http://www.spectrose.com/lockheed-sr-71-blackbird-fastest-flight-world.html

Components

1. Core – Java SDK
2. Realm – Java EE Policy Enforcement
3. Rest – HTTP Interface
4. Web – HTML Interface
5. Accelerator – Extended LDAPv3 Protocol Interface

1. Fortress Core

Project Page:

- https://directory.apache.org/fortress/

Project Source Git Repo:

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

Fortress Core

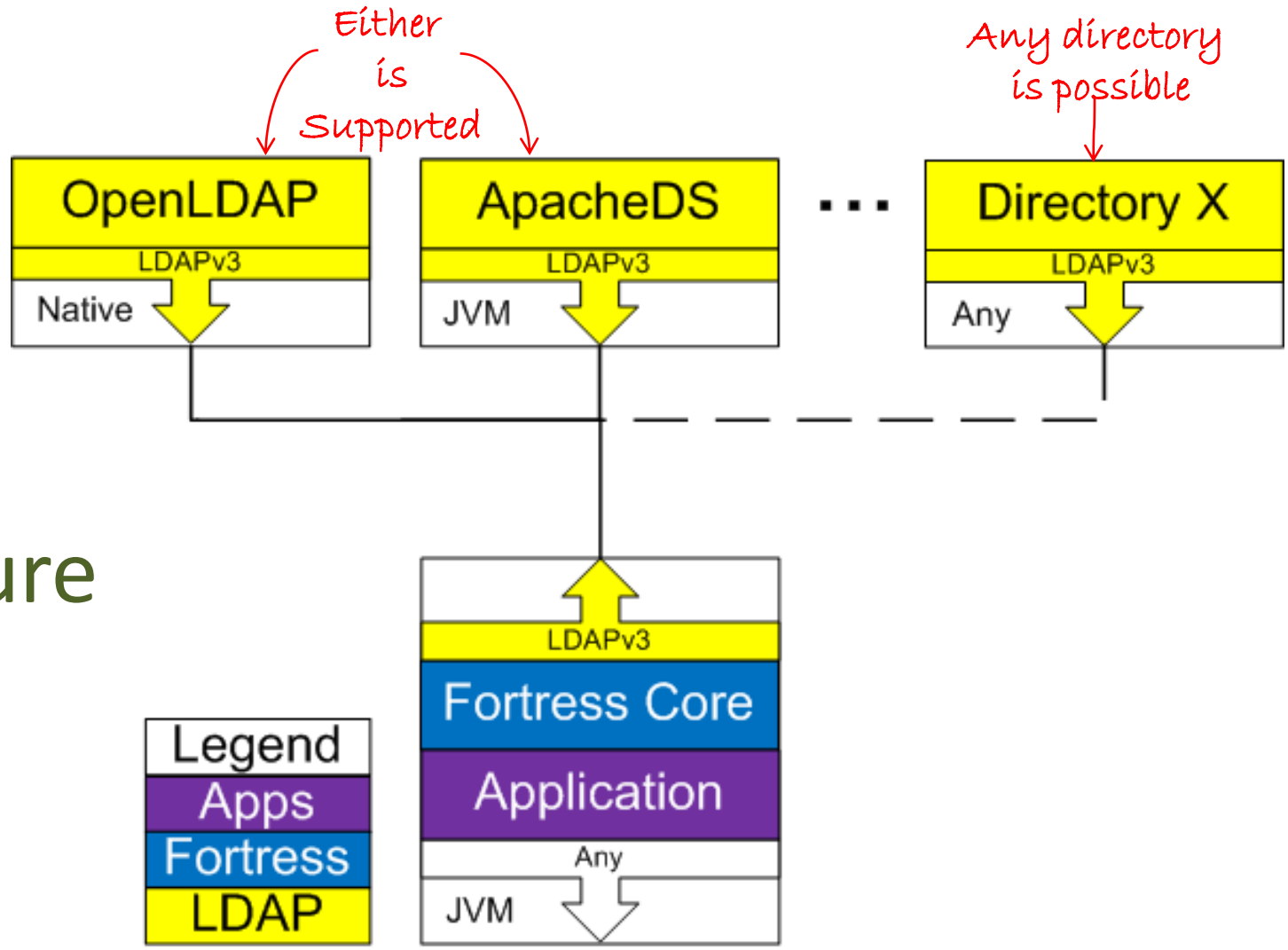
- Identity and Access Management SDK
- Communicates LDAPv3 protocol
- Communicates with HTTP protocol (REST)
- Extensive Regression Tests

Fortress Core Depends On

- Mostly other Apache components like
  - Commons
  - CXF
  - Directory

- With some help from
  - Javax
  - Jgrapht
  - ehcache

Core System Architecture



symas

Core Administrative Utilities

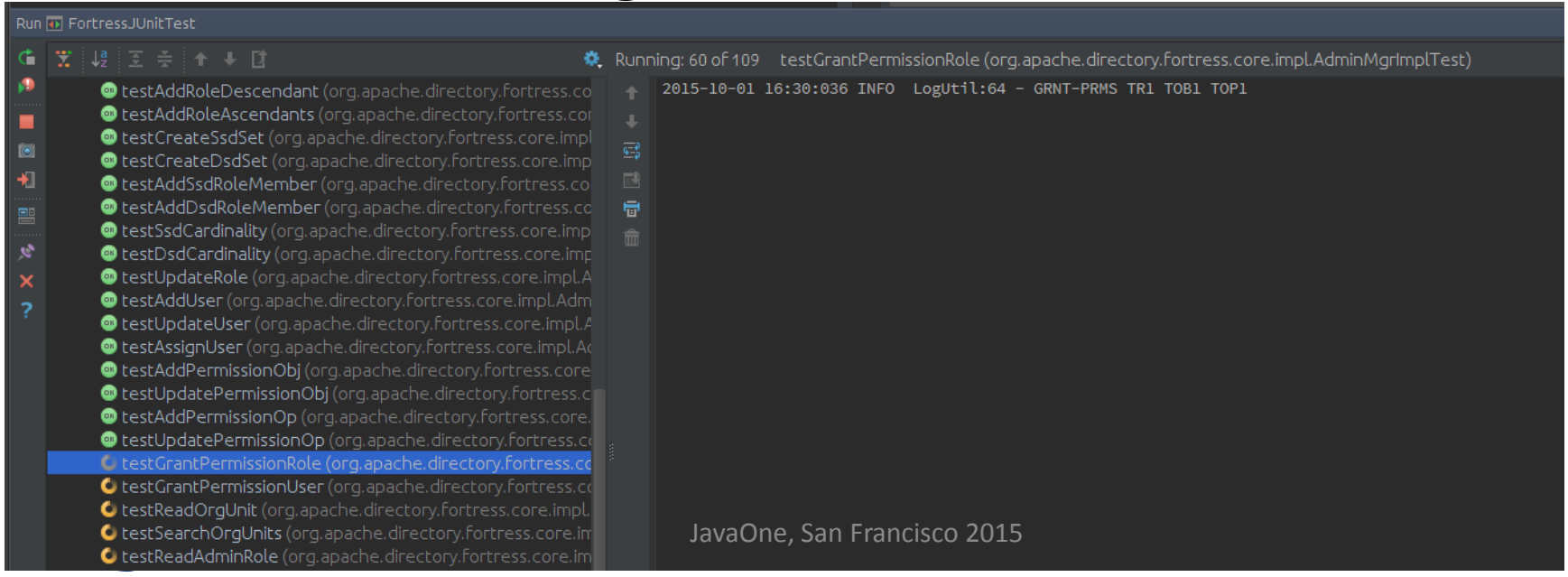
Drive the APIs using:

1. Fortress Ant Admin – automate policy loads using XML files
2. Command Line Interface – for sys admins
3. System Console – interactive, ad hoc testing

Core Unit Tests

Core Unit Tests

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



Core Benchmarks

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

symas

Core Javadoc

- Useful when learning the interfaces:
  - https://directory.apache.org/fortress/gen-docs/latest/apidocs/

2. Fortress Realm

Project Source Git Repo:

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

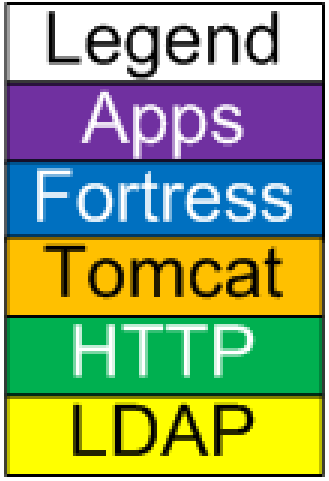
Fortress Realm

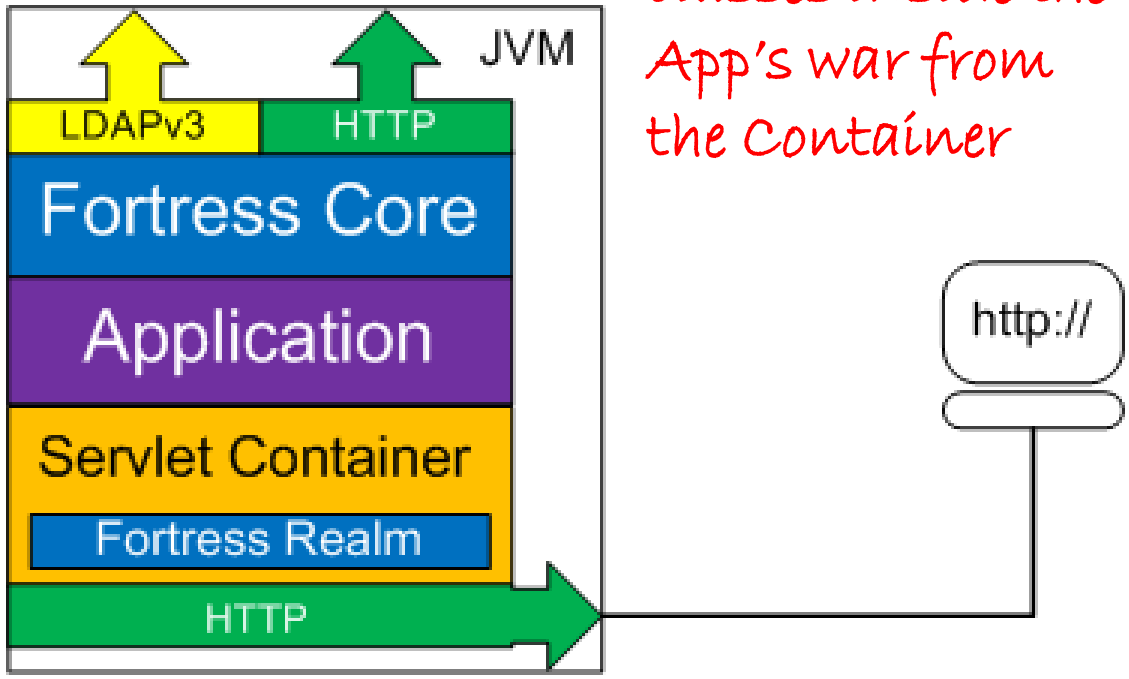
- Standard Java EE Security for Apps running inside containers
- Declarative Policy Enforcement
  - Authentication
  - Authorization
  - Audit
- Uses Fortress Core APIs
- https://git-wip-us.apache.org/repos/asf?p=directory-fortress-realm.git

Realm System Architecture

1. Shares the RBAC session (activated roles) created inside the Realm with the App.

2. Isolates Fortress classes inside the App's war from the Container





Realm Classloader Isolation

Ensure a deterministic deployment process:

1. URLClassloader in Realm Proxy gets its implementation classes from the App's war.
2. Tomcat uses context.xml to connect App's war with a particular Security Realm.

Realm Identity Propagation

1. RBAC session, serializes and stores inside the principal object.

2. Container hands reference to serialized principal to app caller: `servletRequest.getUserPrincipal().toString();`

3. Web app deserializes principal.toString() into RBAC session `j2eePolicyMgr.deserialize( szPrincipal )`

4. Web app pushes RBAC session into HTTP session.

3. Fortress Rest

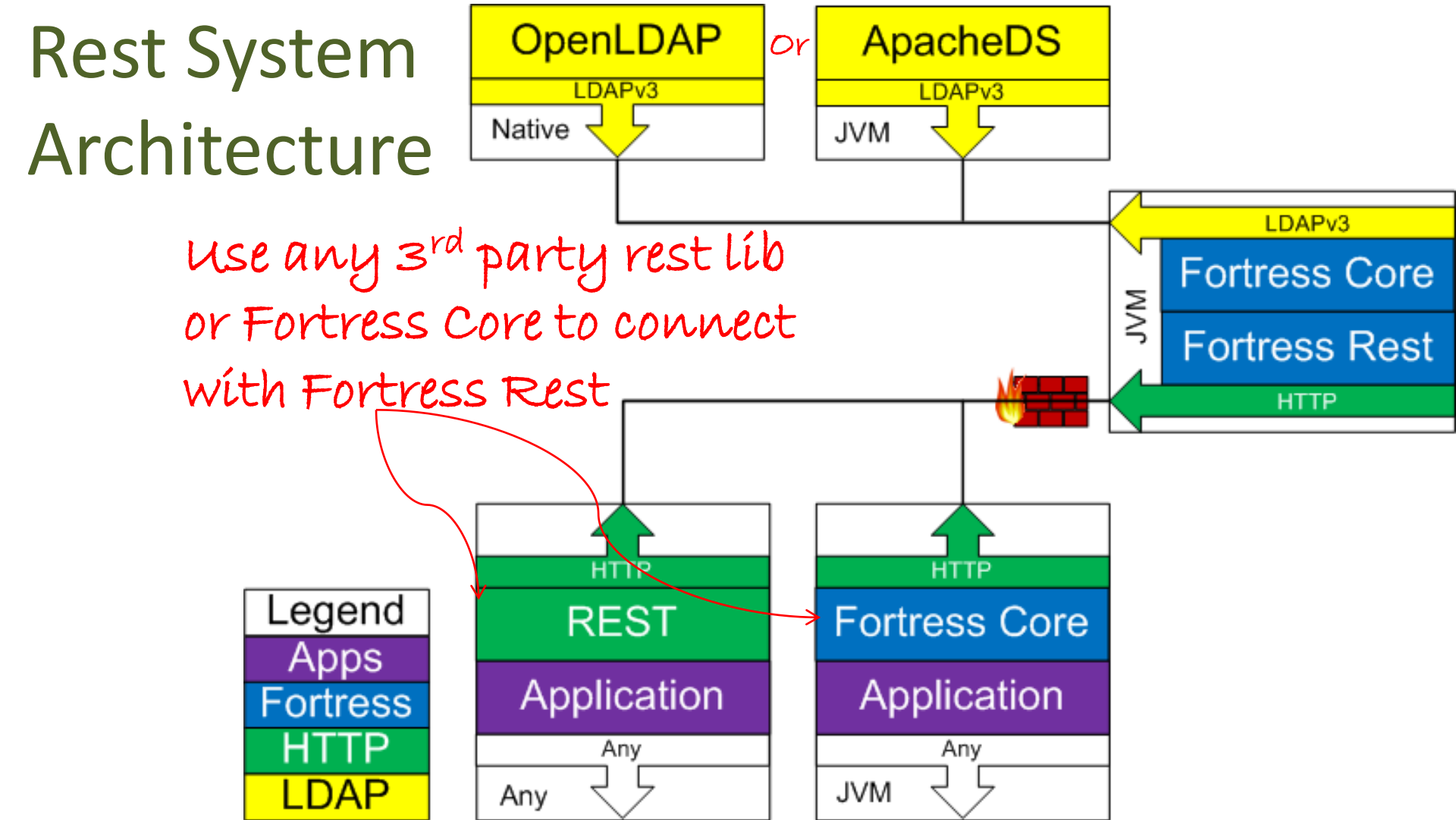
Project Source Git Repo:

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

Fortress Rest

- > 100 HTTP Rest Services
- Wraps Fortress Core APIs
- Secured using Fortress Realm
  - Multi-layer authorization
- Uses Apache CXF
- https://git-wip-us.apache.org/repos/asf?p=directory-fortress-enmasse.git

Rest System Architecture



Fortress Rest Security Model

- Java EE Authentication, Authorization (coarse-grain)
- Apache CXF Service Level Authorization (medium grain)
- ARBAC02 style fine-grained permission checking
- Audit trail (if using OpenLDAP)
- TLS

4. Fortress Web

Project Source Git Repo:

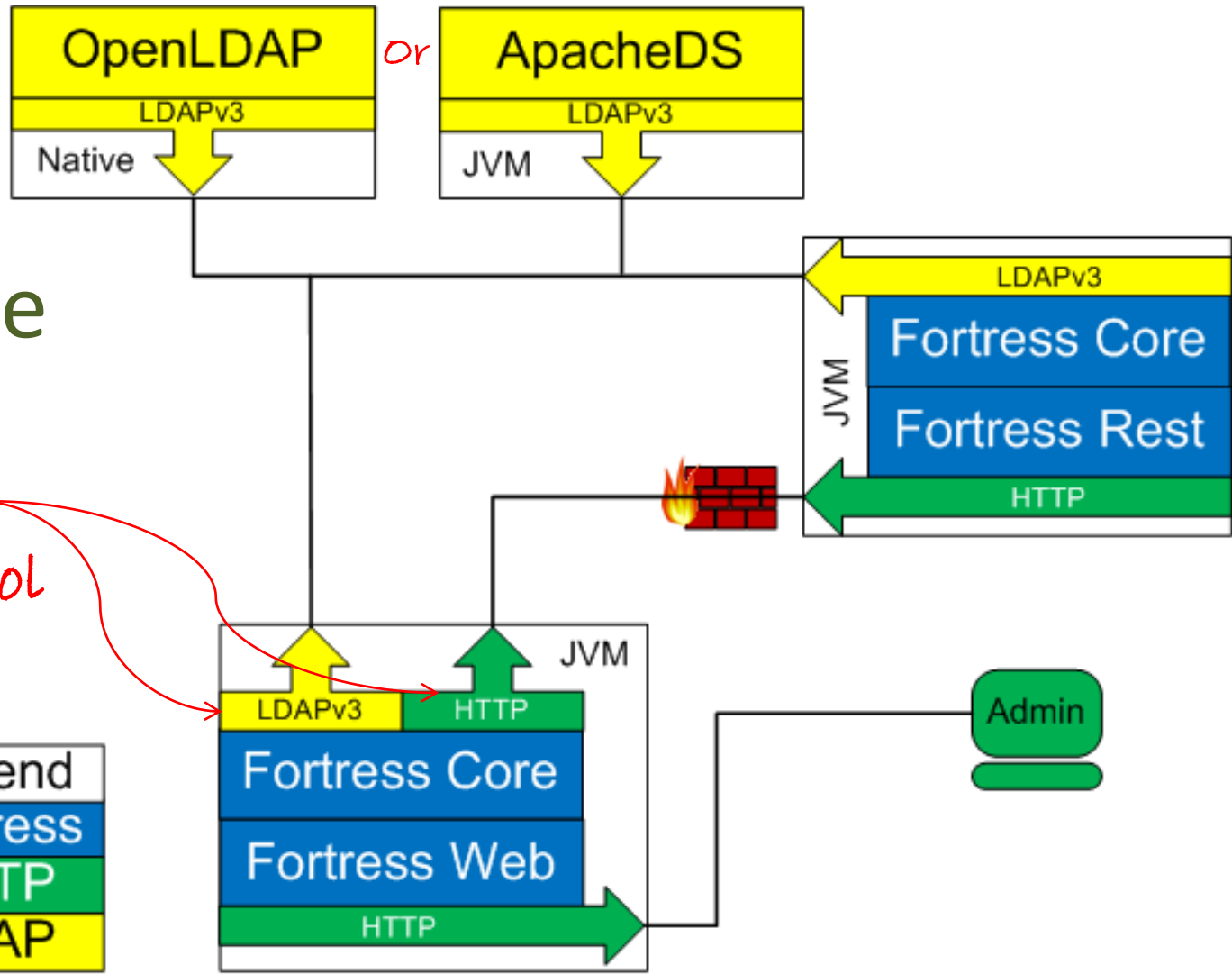
- http://git-wip-us.apache.org/repos/asf/directory-fortress-commander.git

Fortress Web

- Administrative User Interface
- Uses Fortress Core APIs
- Secured with Fortress Realm
  - Multi-layer authorization
- Apache Selenium automated tests
- Apache Wicket Web framework
- https://git-wip-us.apache.org/repos/asf?p=directory-fortress-commander.git

Web System Architecture

Option to use either HTTP or LDAPV3 protocol



symas

Fortress Web Security Model

- Java EE Authentication and Authorization (coarse-grain)
- Spring Page Authorization (medium grain)
- ARBAC02 style fine-grained permission checking
- Audit trail (if using OpenLDAP)
- TLS

5. Fortress Accelerator

- Server-side – OpenLDAP Overlay

  source not yet released

- Client-side

  ssh://git-master.openldap.org/~git/git/openldap-fortress-accelerator.git

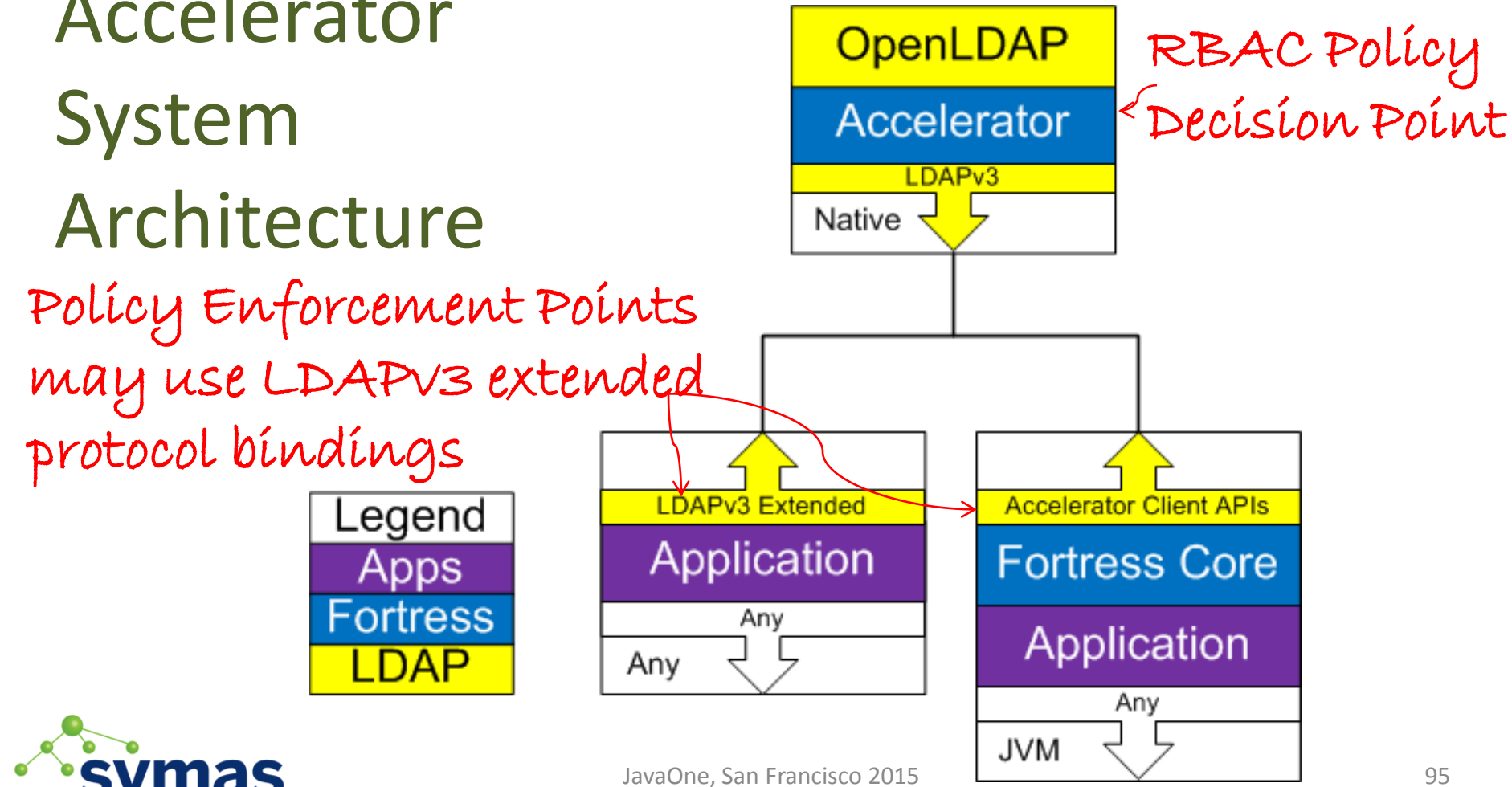
OpenLDAP Fortress Accelerator

- Implements RBAC System Manager APIs
- Server-side runs natively inside OpenLDAP daemon (slapd)
- Client-side runs on various platforms
  - C, Java, Python, …
  - Anything capable of LDAPv3 extended operations
- **ssh://git-master.openldap.org/~git/git/openldap-fortress-accelerator.git**

Accelerator Benefits

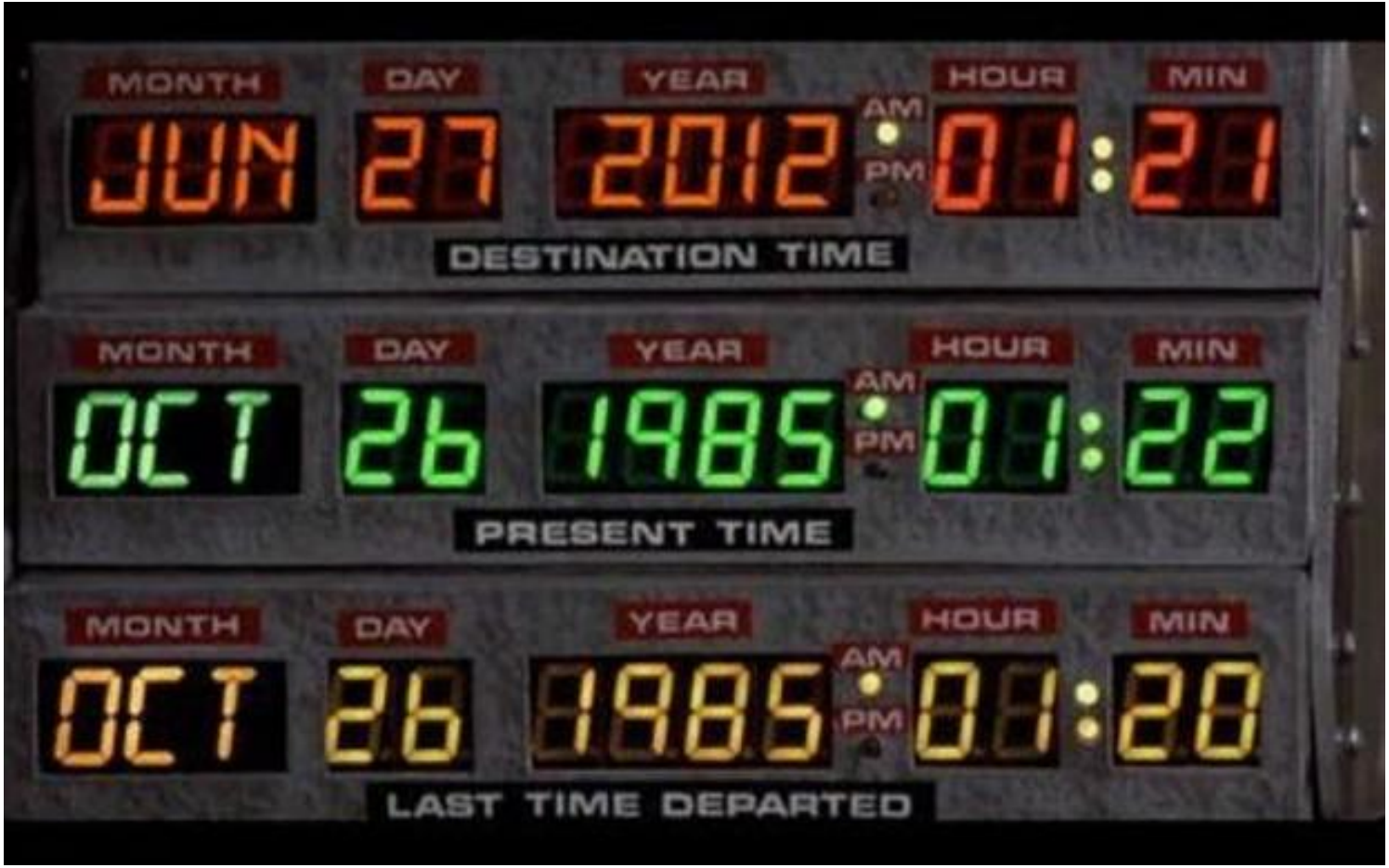
- Optimized for performance (< 1ms roundtrip)
- Session State Persistent inside server's database
- Less processing overhead on client machine
- Richer audit trail stored than with the Core APIs
- Client-side independence – Java, C, Python, …

Accelerator System Architecture



Future

What's next?



Roadmap

- IETF RBAC Standardization for Directories
- Accelerator and Audit for Apache Directory Server
- Web Access Management / SSO
- Make the REST services really restful
- Policy Enforcement Modules for:
  - common linux distros
  - common web framework
  - other languages like C, Python, Ruby, …

More on IETF Standardization

- Encourage interoperability across directories
- Standard RBAC Object Model (LDAP Schema)
- Standard RBAC Functional Model (LDAPv3 operations)

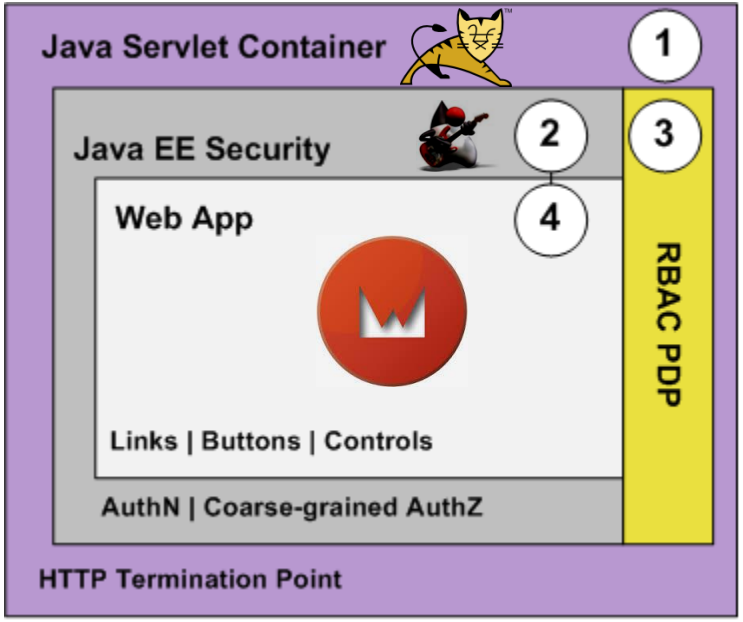
Future Think

- ANSI INCITS 494-2012 RBAC Policy Enhanced
- Attribute-Based Access Control
  - But where are the functional specs?
- XACML
- OAuth2 & UMA

Usage

How do I use what was built?

Demo Wicket Sample

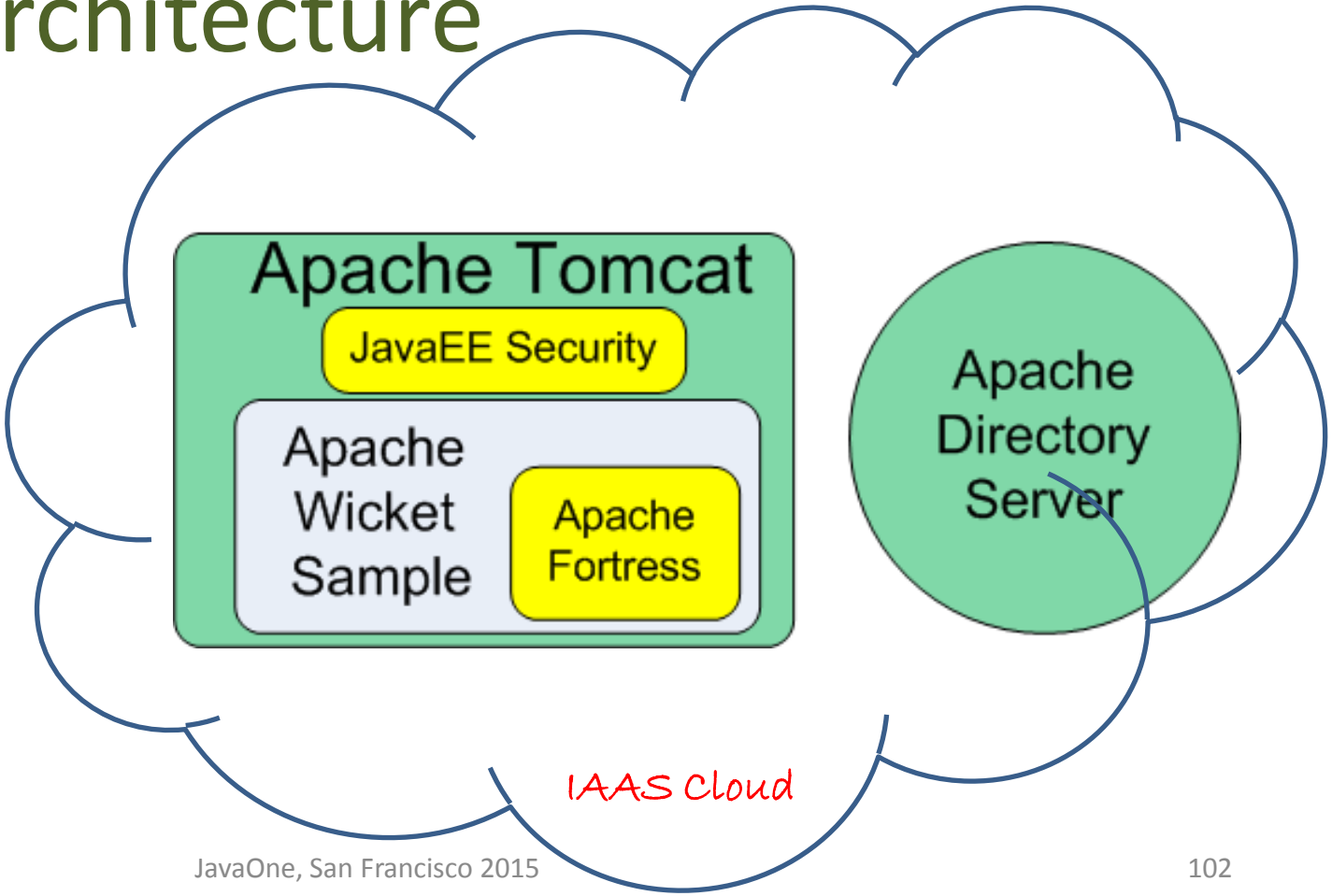


https://github.com/shawnmckinney/wicket-sample

HTTP LDAPv3

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

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');" );
      }
   }
   });
```

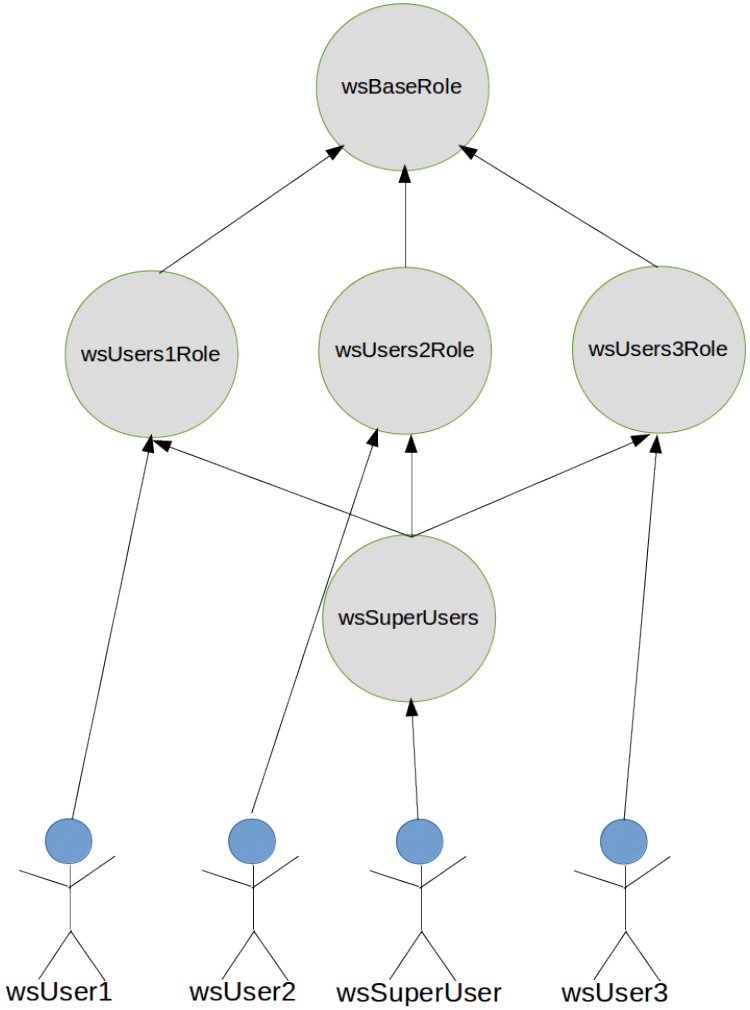
fine-grained authorization (programmatic)

Demo Wicket 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



symas

Tutorial Links

In Gitub:

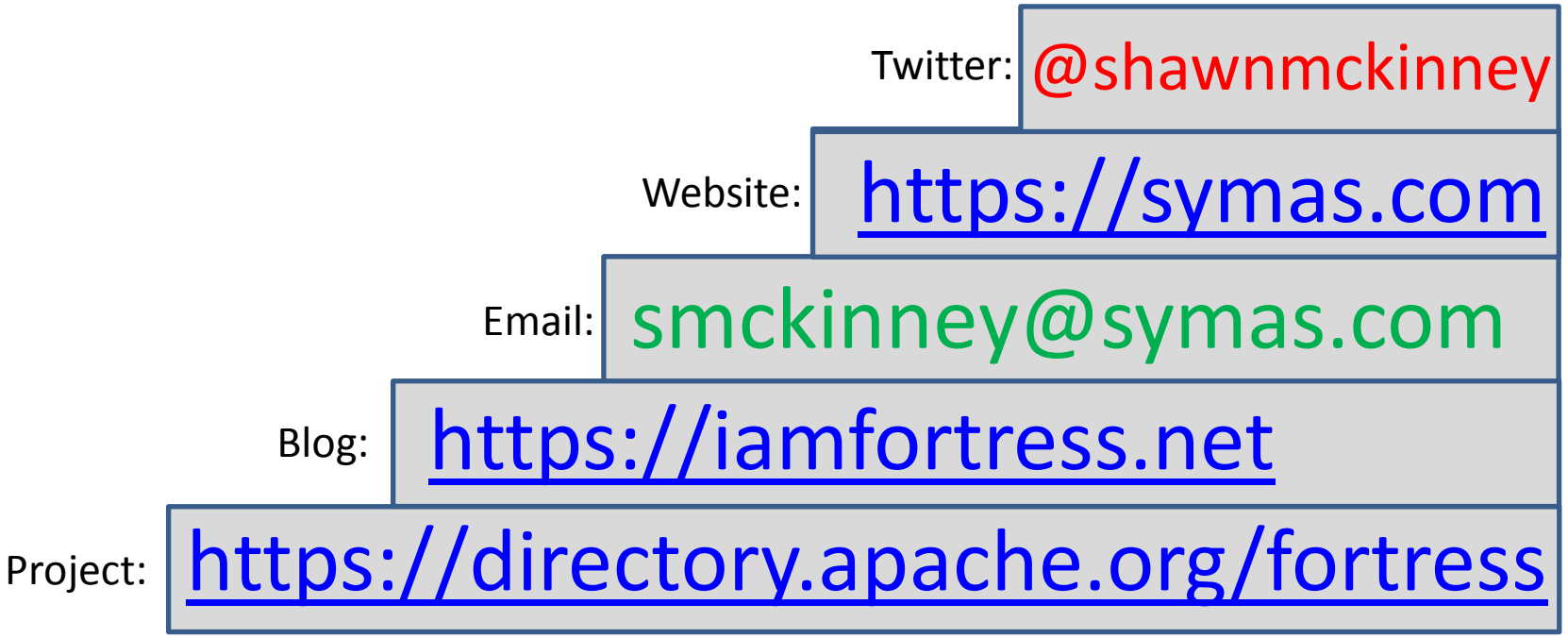
1. Wicket Sample:
   - https://github.com/shawnmckinney/wicket-sample

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

Related Sessions

- CON3568 - Federated RBAC: Fortress, OAuth2 (Oltu), JWT, Java EE, and JASPIC
  - October 27, 11:00 am - 12:00 pm | Hilton—Plaza Room B
- CON2324 – A Practical Guide to Role Engineering
  - October 27, 2:30 p.m. | Hilton—Plaza Room B
- CON2323 - The Anatomy of a Secure Web Application Using Java Redux
  - October 28, 3:00 pm - 4:00 pm | Hilton—Plaza Room A

Contact Me


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