RBAC Authorization With Apache Directory Server And Fortress

Shawn McKinney
Emmanuel Lécharny

Introducing Fortress

- Accepted as a Directory sub-project in October 2014.
- IP clearance on its way
- Code injection in Git repository last week
- ~90 000 SLOCs

Introducing Fortress

RBAC Access Control + Management Apps

- **Core** - Software Development Kit
- **Realm** - Policy Enforcement Plug-in for Java EE Apps + Servers
- **Web** - Administrative GUI for Management of Policies
- **REST** - XML/HTTP Policy Server

Fortress Features

- ANSI Role-Based Access Control
- ARBAC02 Model Delegated Administration
- IETF Password Policies
- LDAPv3 Interoperability
- Multi-tenant Data and Object Model
- Audit Trail (OpenLDAP only)
- Works with any LDAP v3 compliant server

Introducing ApacheDS

LDAP + Kerberos Server, and more...

- ApacheDS - LDAP Server
- Studio - Ldap Browser
- LDAP API - A new Java LDAP API
- Mavibot - MVCC btree
- Escimo - SCIM 2.0 implementation

Introducing ApacheDS

- Get accepted in Apache Incubator in 2003
- Became a TLP in 2005
- 37 releases since then
- Around 700 000 SLOCs

ApacheDS Features

- LDAP server in Java
- Kerberos server built-in
- Embeddable
- Password Policy support
- Multi-master replication (RFC 4533)
- X500 authorization
- Multi-platform (Windows, Linux, Mac OSX)

Role-Based Access Model

Early History

- Introduced in 1992 (David Ferraiolo and Richard Kuhn)
- Meant to address critical shortcomings of DAC
- Integrity was lacking as the requirement for data and process to be modified only in authorized ways by authorized users.

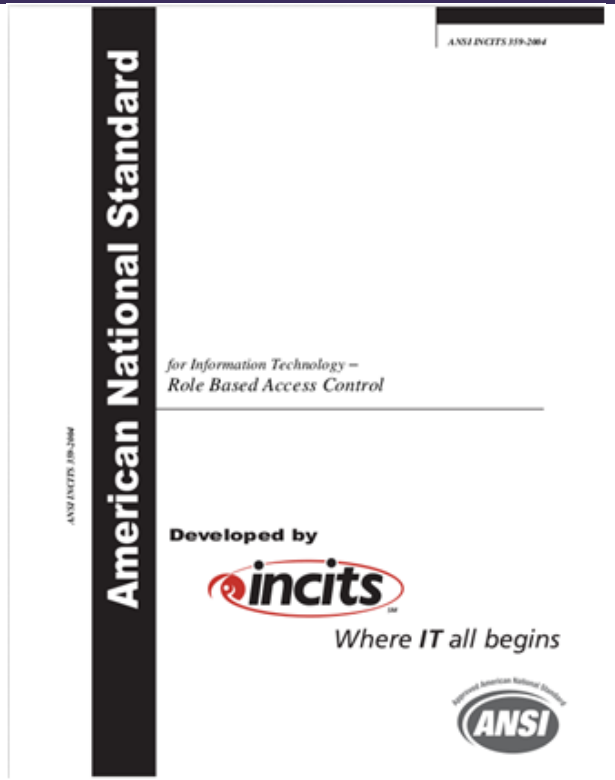
Role-Based Access Control Standard

Middle Years – 'Towards a Unified Standard'

- 2000, 'The NIST Model for a Role-Based Access Control: Towards a Unified Standard' (Sandhu, Ferraiolo, Kuhn).
- RBAC formal model
- Basis for the standard to follow.
- Functional Specs written in Z-notation

ANSI INCITS 359

IN 2004 ANSI FORMALIZED RBAC INTO A STANDARD



ANSI RBAC INCITS 359

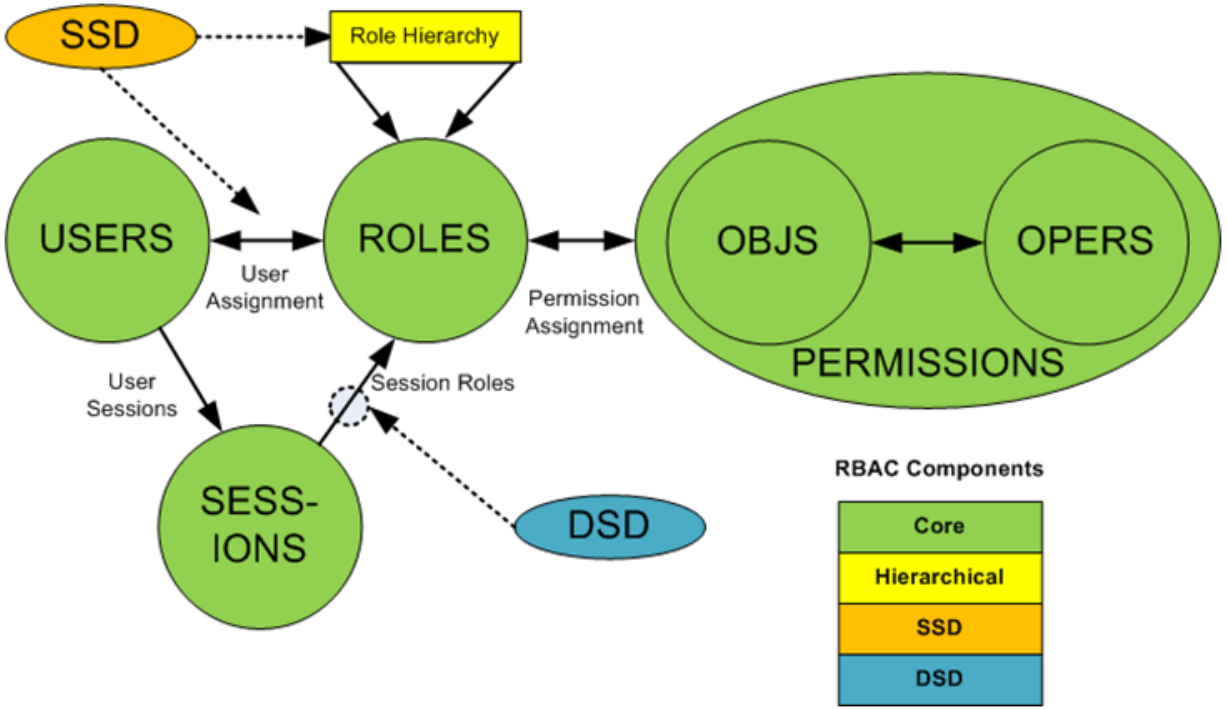
RBAC 0 – Users, Roles, Perms Objects, Operations

RBAC 1 – Hierarchical Roles

RBAC 2 – Static Separation of Duties

RBAC 3 – Dynamic Separation of Duties

ANSI RBAC INCITS 359



ANSI RBAC Functional Model

Three standard interfaces:

- **Administrative** - CRUD
- **Review** - policy interrogation
- **System** - policy enforcement

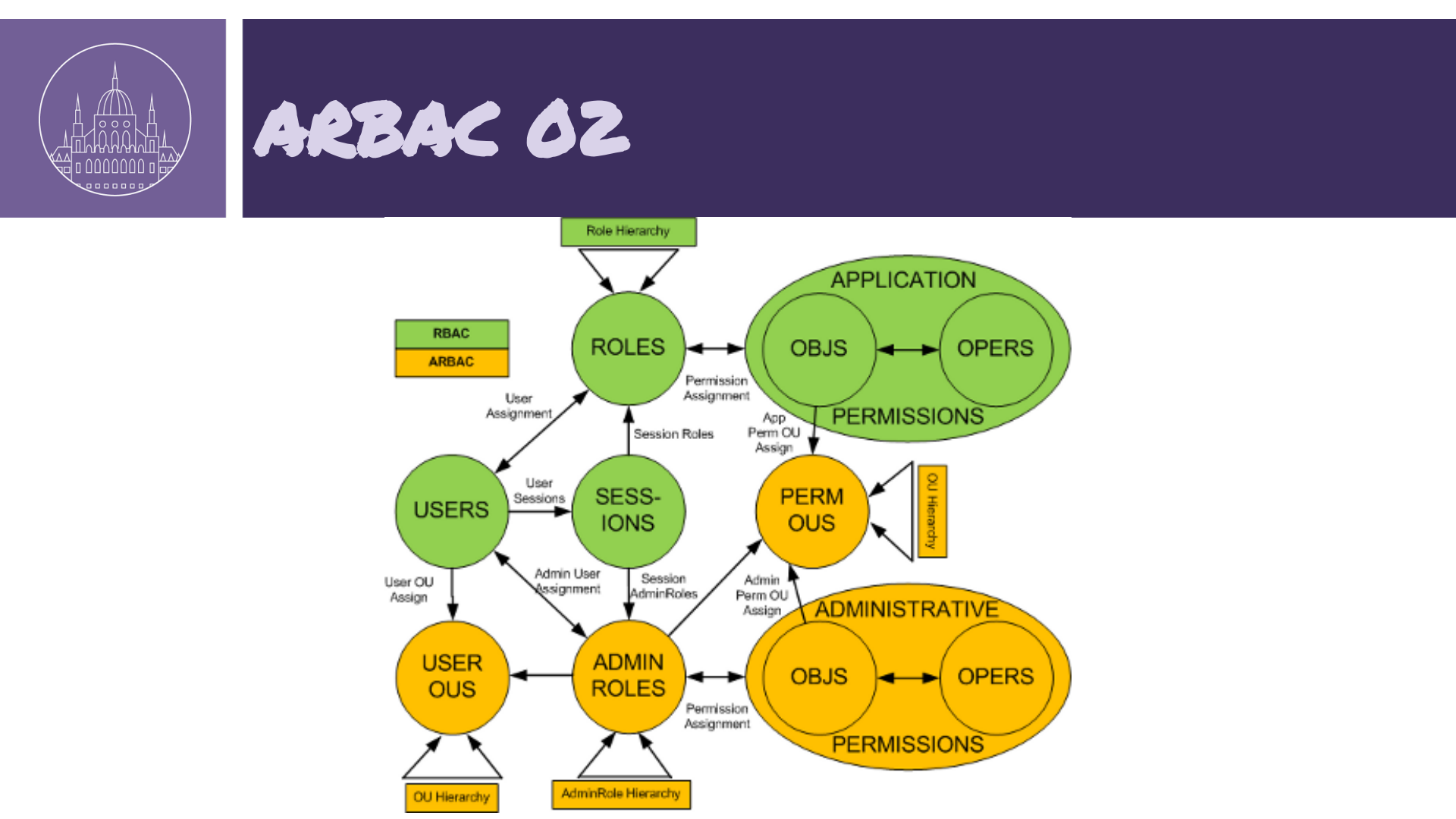
ANSI RBAC Object Model

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

ARBAC02

Delegated Admin :

- Manage the RBAC system
- Used for controlling what the administrators can do



ARBAC 02

Role Hierarchy
RBAC
ARBAC
ROLES
APPLICATION
OBJS
OPERS
PERMISSIONS
Permission Assignment
User Assignment
Session Roles
App Perm OU Assign
USERS
User Sessions
SESS-IONS
PERM OUS
OU Hierarchy
User OU Assign
Admin User Assignment
Session AdminRoles
Admin Perm OU Assign
ADMINISTRATIVE
USER OUS
ADMIN ROLES
OBJS
OPERS
PERMISSIONS
Permission Assignment
OU Hierarchy
AdminRole Hierarchy

ARBAC Functional Model

Three standard interfaces:

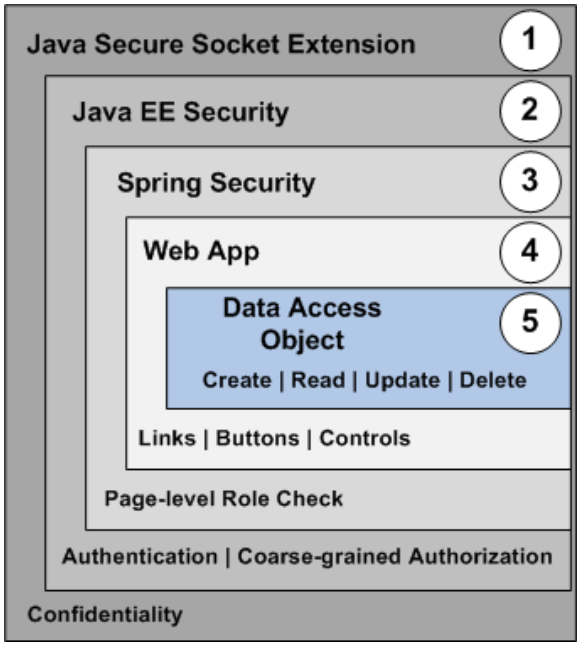
- **Administrative** - CRUD
- **Review** - policy interrogation
- **System** - policy enforcement

ARBAC Object Model

Four basic elements:

- **Admin Role** – an administrator
- **Admin Permission** – approval to perform an RBAC administrative Operation
- **User Org Unit** – maps to an org chart of people
- **Perm Org Unit** – maps to a graph of IT applications and systems

Demo Fortress



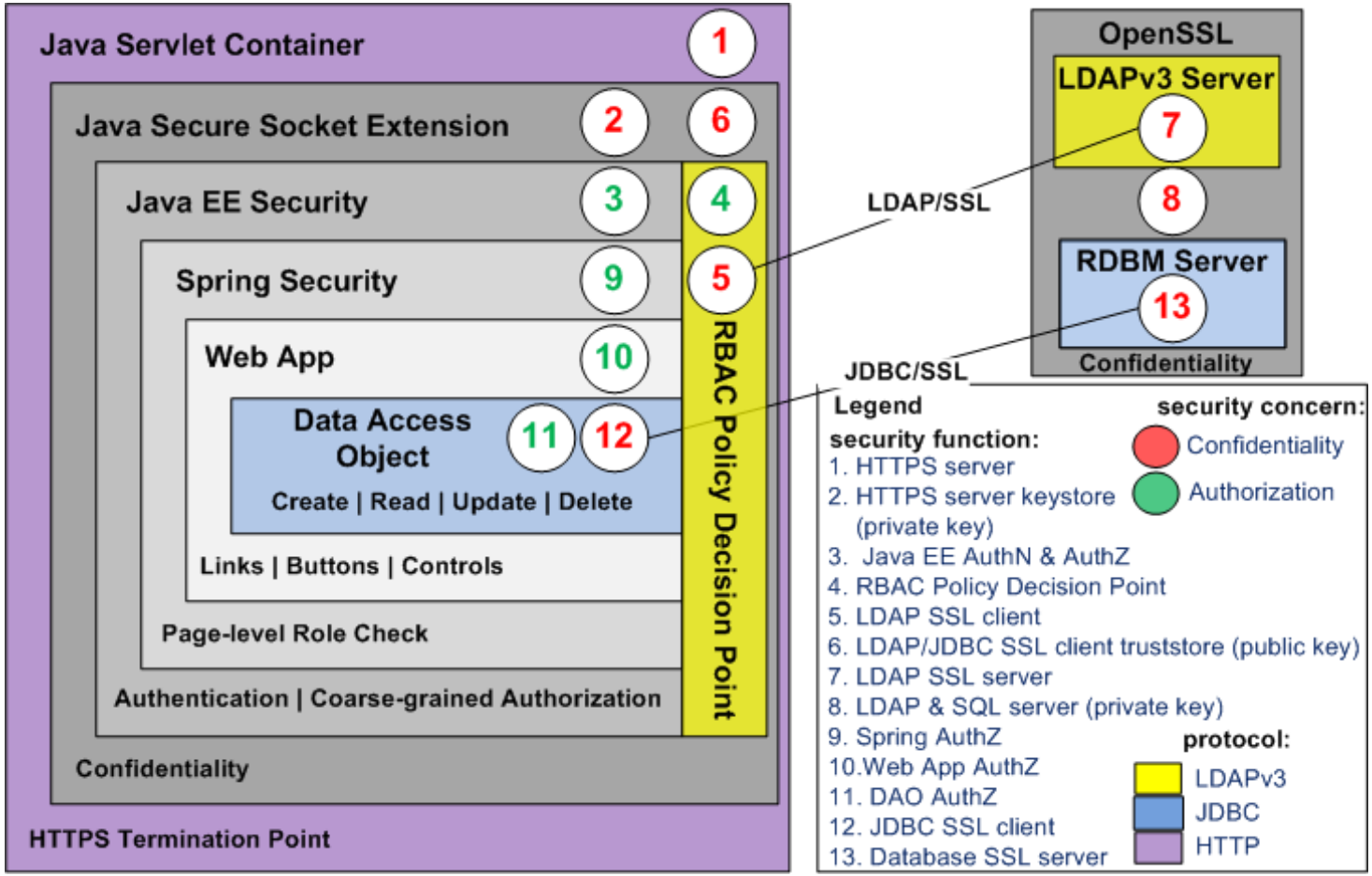
----------> JSSE

----------> Java EE Security

----------> Spring Security

----------> Web App Framework

----------> Database Functions


Java Servlet Container
1
Java Secure Socket Extension
2
6
Java EE Security
3
4
Spring Security
9
5
Web App
10
Data Access Object
11
12
Create | Read | Update | Delete
Links | Buttons | Controls
Page-level Role Check
Authentication | Coarse-grained Authorization
RBAC Policy Decision Point
Confidentiality
HTTPS Termination Point
LDAP/SSL
JDBC/SSL
OpenSSL
LDAPv3 Server
7
8
RDBM Server
13
Confidentiality
Legend
security function:
1. HTTPS server
2. HTTPS server keystore (private key)
3. Java EE AuthN & AuthZ
4. RBAC Policy Decision Point
5. LDAP SSL client
6. LDAP/JDBC SSL client truststore (public key)
7. LDAP SSL server
8. LDAP & SQL server (private key)
9. Spring AuthZ
10.Web App AuthZ
11. DAO AuthZ
12. JDBC SSL client
13. Database SSL server
security concern:
Confidentiality
Authorization
protocol:
LDAPv3
JDBC
HTTP

Demo Security Policy

	P1 C123	P1 C456	P1 C789	P2 C123	P2 C456	P2 C789	P3 C123	P3 C456	P3 C789
SUPER USER *	T	T	T	T	T	T	T	T	T
POWER USER	T	T	T	T	T	T	T	T	T
USER1	T	T	T	F	F	F	F	F	F
USER 123	T	F	F	T	F	F	T	F	F
USER1 123	T	F	F	F	F	F	F	F	F

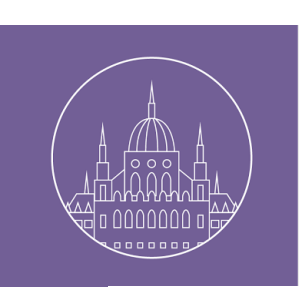
* denotes Dynamic Separation of Duties is NOT enforced

WHERE TO GET DEMO CODE

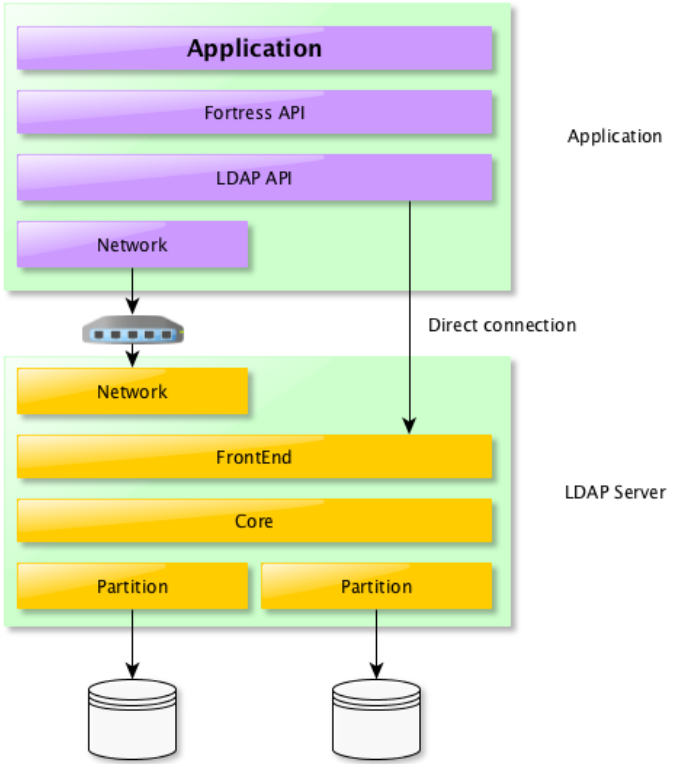
DOWNLOAD FROM HERE:

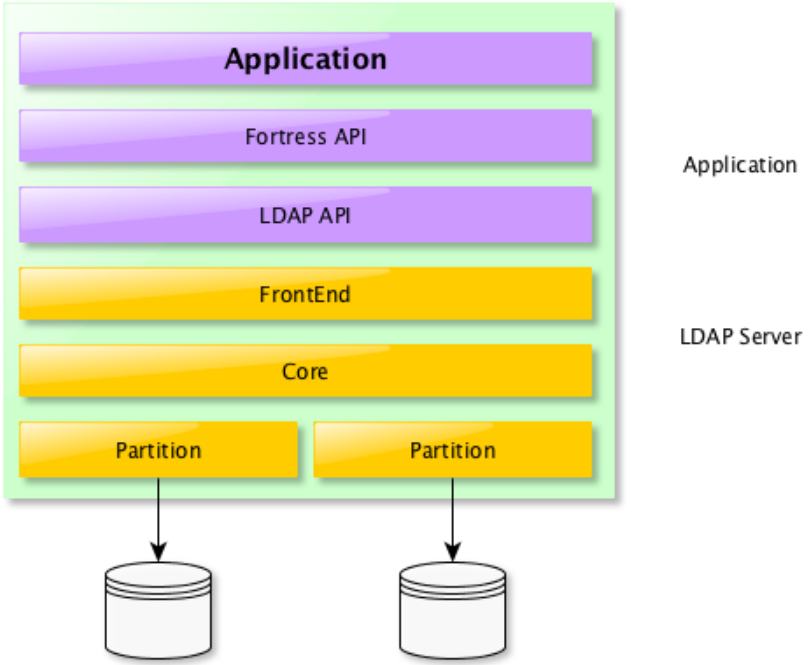
HTTPS://GITHUB.COM/SHAWNMCKINNEY/APACHE-FORTRESS-DEMO

FOLLOW INSTRUCTIONS IN THE README TO GENERATE TUTORIAL DOCUMENT



Fortress + ApacheDS





Future...

- ApacheDS in Fortress
  - in-memory
  - All in one
  - Fast
  - SyncreplL

Future...

- Fortress in ApacheDS
  - All in one
  - Fast
  - Lightweight

OpenLDAP already as it : https://symas.com/products/symas-enforcement-foundry/suite/accelerator/

Contacts

http://directory.apache.org

http://www.symas.com

Shawn McKinney,

twitter@shawnmckinney

smckinney@apache.org

Emmanuel Lécharny

twitter@elecharny

elecharny@apache.org