Next Generation Directory-Based User Management for Cloud Infrastructure

Nov 16, 2016

ApacheCon EU, Seville

symas

Introductions

- Katarina Valalikova
  - @KValalikova
  - k.valalikova@evolveum.com
- Shawn McKinney
  - @shawnmckinney
  - smckinney@symas.com

“I had to keep guessing at the channel;
I had to discern, mostly by inspiration,
the signs of hidden banks;
I watched for sunken stones;
When you have to attend to things of that sort,
to the mere incidents of the surface,
the reality—the reality, I tell you—fades.
The inner truth is hidden.”

Joseph Conrad, *Heart of Darkness*

https://en.wikipedia.org/wiki/File:VingtAnnees_286.jpg

Session Objective

Uncover a hidden navigation channel for users and machines through ‘the cloud’.

Session Agenda

- History
- Building Blocks
- Security Model
- Solution
- Use Cases
- Demo
- Questions

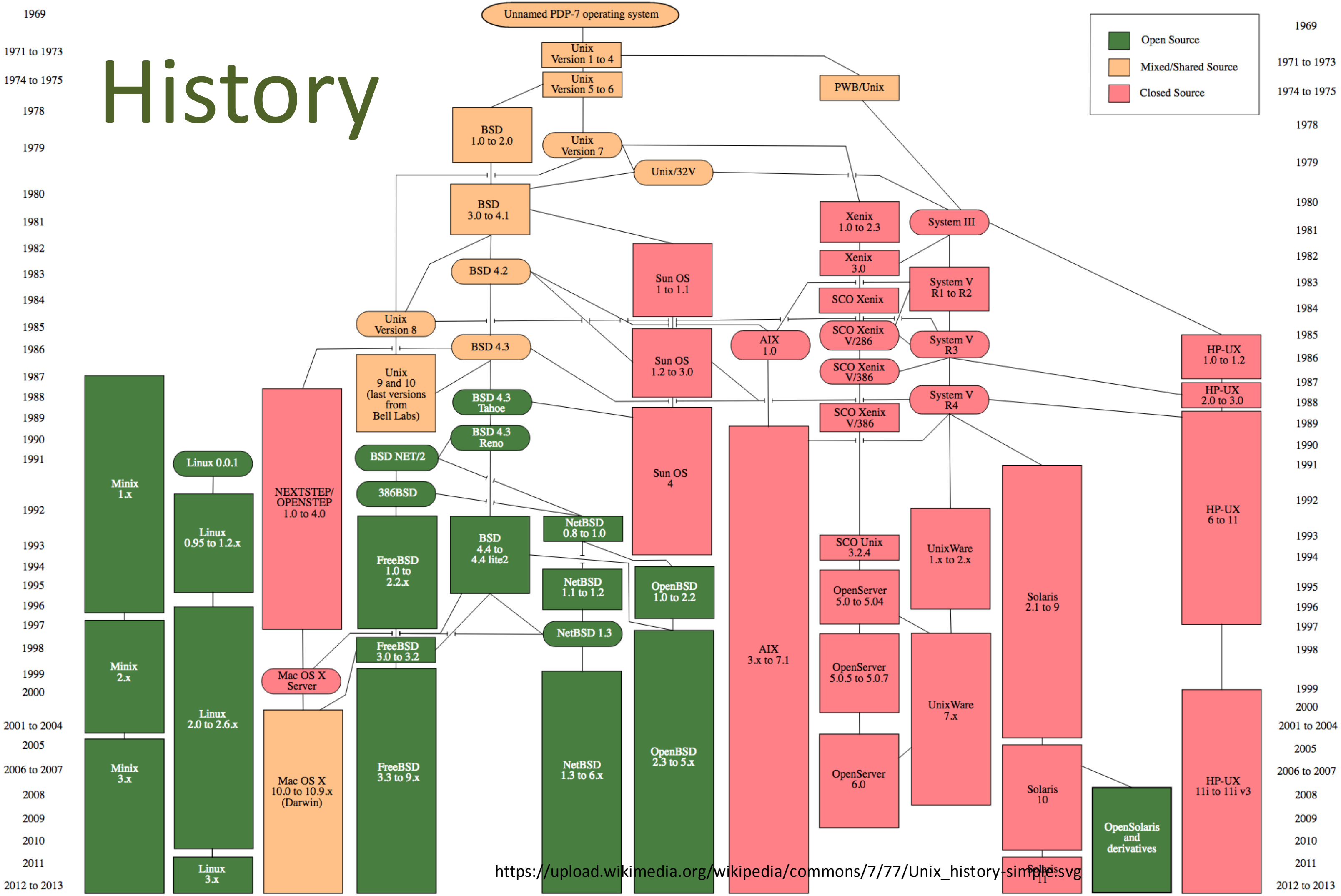
Image from: HTTP://EVENTS.LINUXFOUNDATION.ORG/EVENTS/APACHECON-NORTH-AMERICA

symas

History

Knowing the path forward necessarily means we understand where we’ve been.

History



Open Source

Mixed/Shared Source

Closed Source

1969

1971 to 1973

1974 to 1975

1978

1979

1980

1981

1982

1983

1984

1985

1986

1987

1988

1989

1990

1991

1992

1993

1994

1995

1996

1997

1998

1999

2000

2001 to 2004

2005

2006 to 2007

2008

2009

2010

2011

2012 to 2013

Unnamed PDP-7 operating system

Unix Version 1 to 4

Unix Version 5 to 6

PWB/Unix

BSD 1.0 to 2.0

Unix Version 7

Unix/32V

BSD 3.0 to 4.1

Xenix 1.0 to 2.3

System III

BSD 4.2

Sun OS 1 to 1.1

Xenix 3.0

System V R1 to R2

SCO Xenix

Unix Version 8

AIX 1.0

SCO Xenix V/286

System V R3

HP-UX 1.0 to 1.2

BSD 4.3

Sun OS 1.2 to 3.0

SCO Xenix V/386

Unix 9 and 10 (last versions from Bell Labs)

BSD 4.3 Tahoe

System V R4

HP-UX 2.0 to 3.0

NEXTSTEP/ OPENSTEP 1.0 to 4.0

SCO Xenix V/386

BSD 4.3 Reno

Minix 1.x

Linux 0.0.1

BSD NET/2

Sun OS 4

386BSD

HP-UX 6 to 11

Linux 0.95 to 1.2.x

NetBSD 0.8 to 1.0

FreeBSD 1.0 to 2.2.x

BSD 4.4 to 4.4 lite2

SCO Unix 3.2.4

UnixWare 1.x to 2.x

Solaris 2.1 to 9

NetBSD 1.1 to 1.2

OpenBSD 1.0 to 2.2

OpenServer 5.0 to 5.04

NetBSD 1.3

FreeBSD 3.0 to 3.2

AIX 3.x to 7.1

Minix 2.x

Mac OS X Server

OpenServer 5.0.5 to 5.0.7

UnixWare 7.x

Linux 2.0 to 2.6.x

Mac OS X 10.0 to 10.9.x (Darwin)

FreeBSD 3.3 to 9.x

NetBSD 1.3 to 6.x

OpenBSD 2.3 to 5.x

Minix 3.x

OpenServer 6.0

Solaris 10

HP-UX 11i to 11i v3

OpenSolaris and derivatives

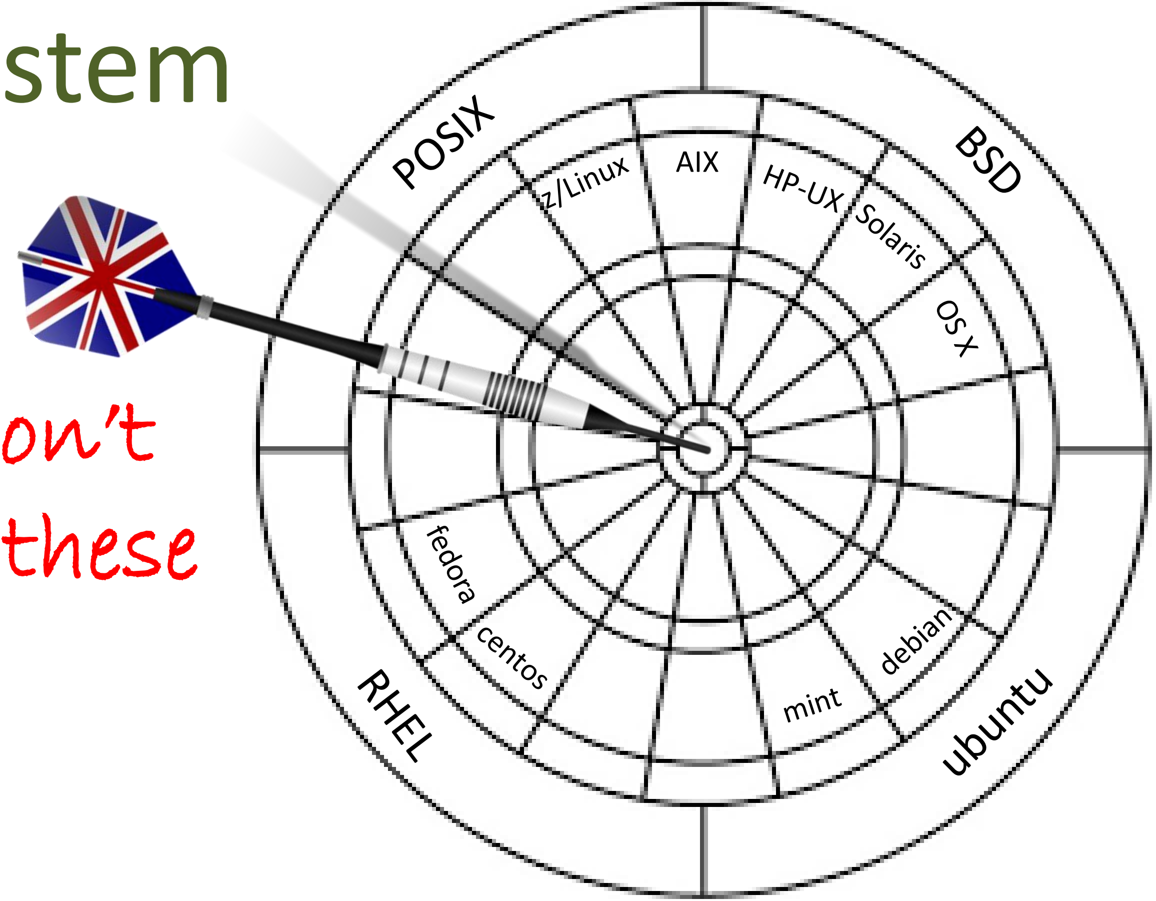
Linux 3.x

Solaris 11

https://upload.wikimedia.org/wikipedia/commons/7/77/Unix_history-simple.svg

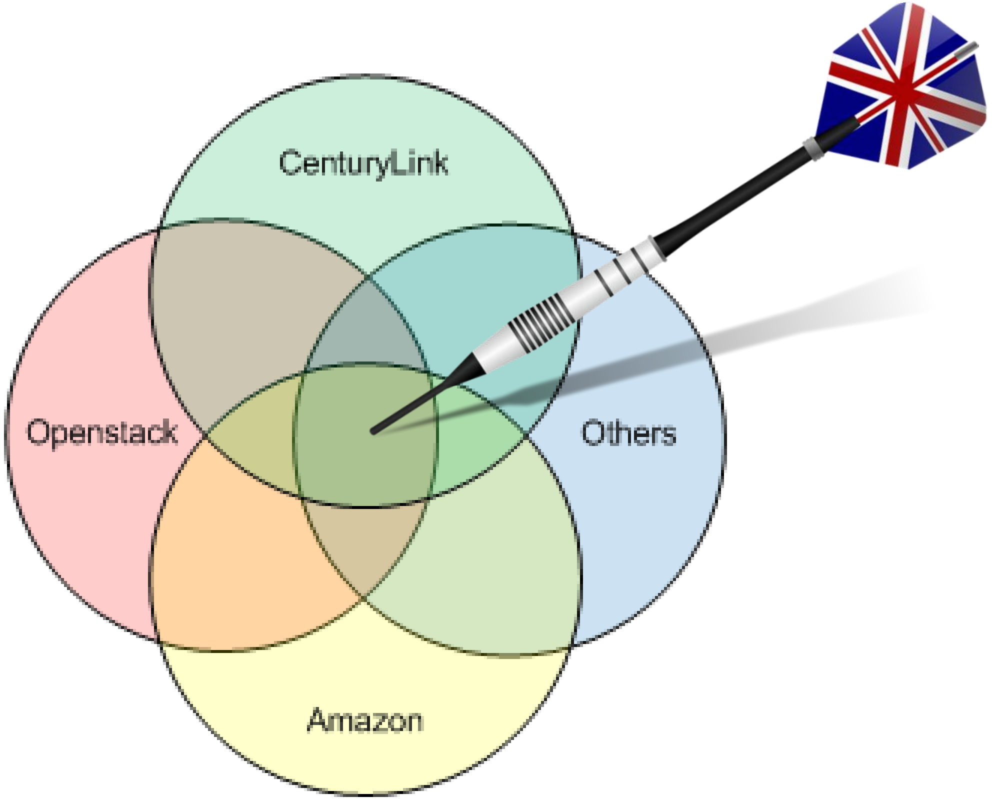
Operating System

A new wheel won't work on all of these

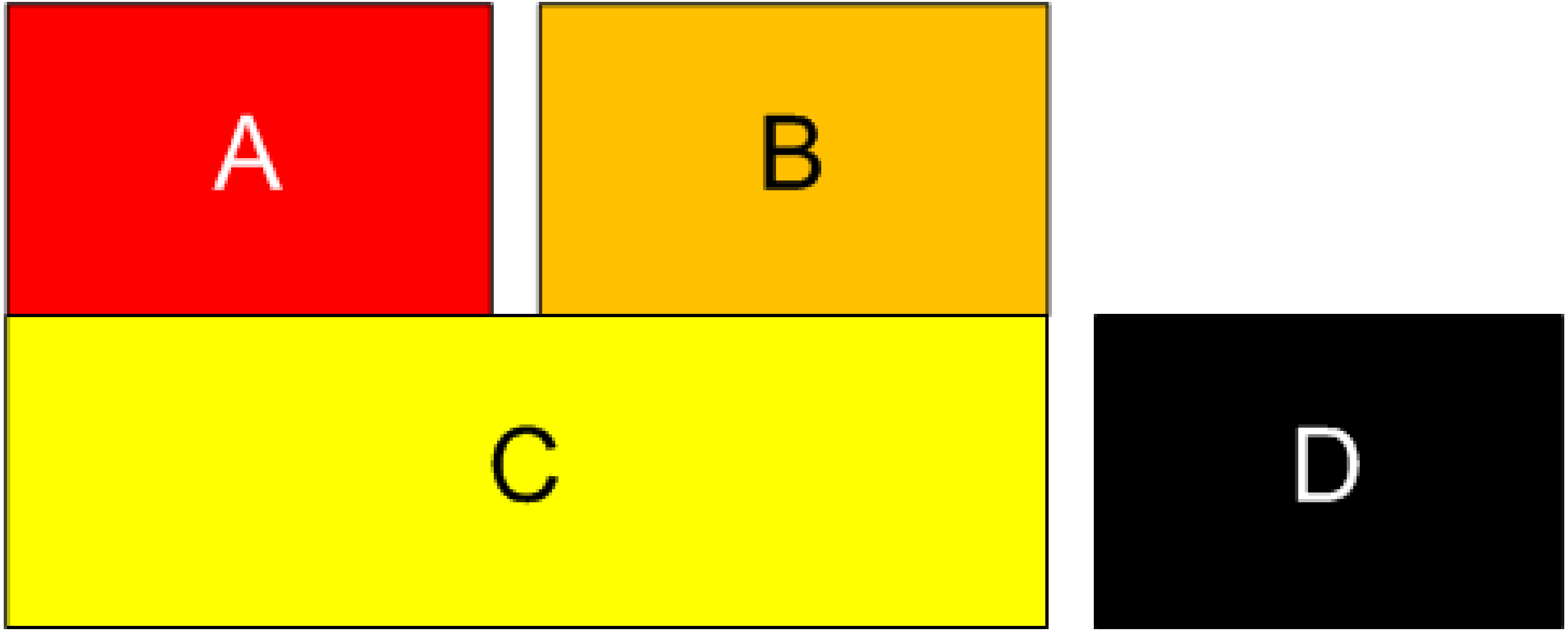


Cloud Infrastructure

Nor across all of these



Building Blocks



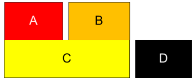
Basic Building Blocks

1. POSIX security controls
2. Directory services

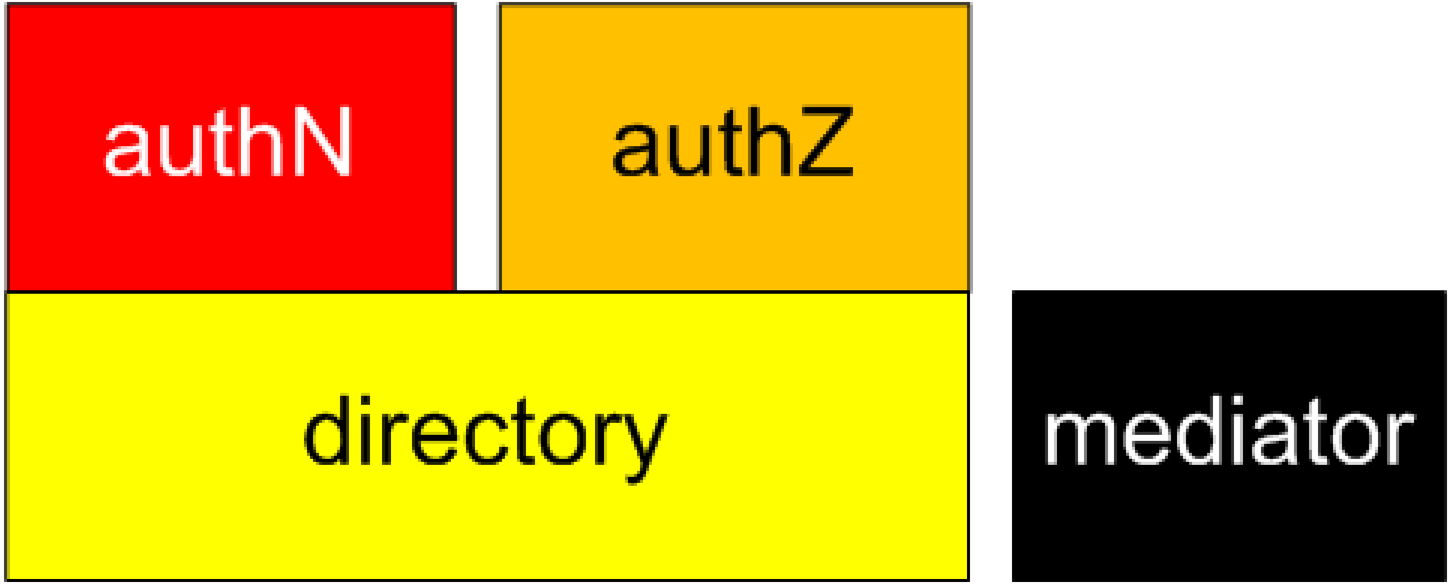
Best practices

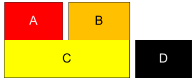
Advanced Building Blocks

3. Mediation relatively new practice

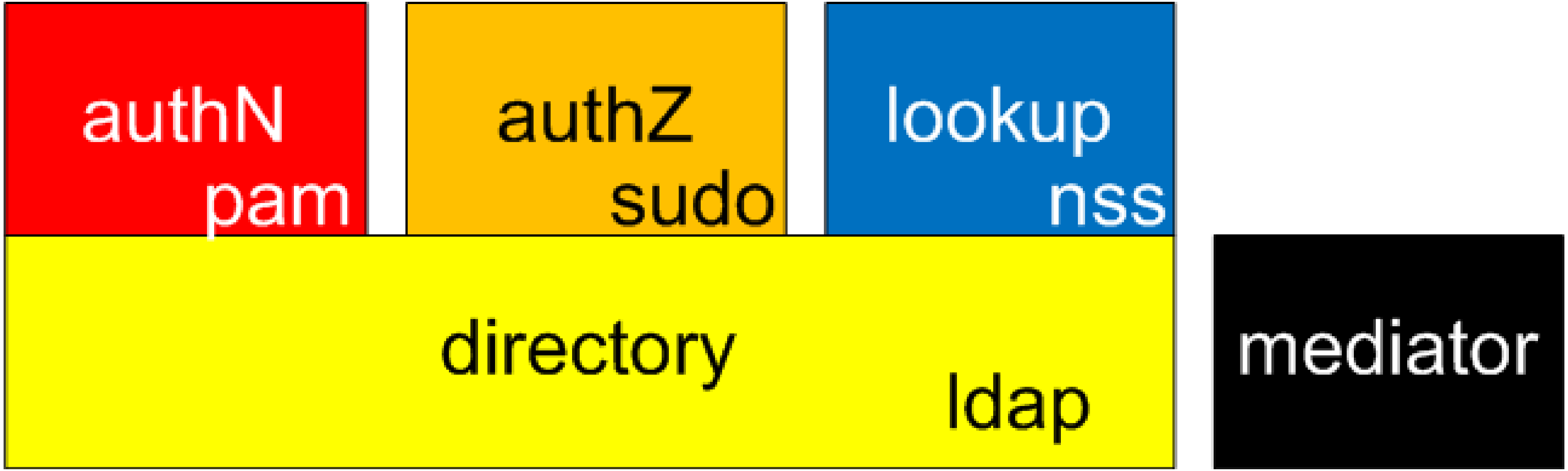


Building Blocks Conceptual





Building Blocks Actual



authN
pam

authZ
sudo

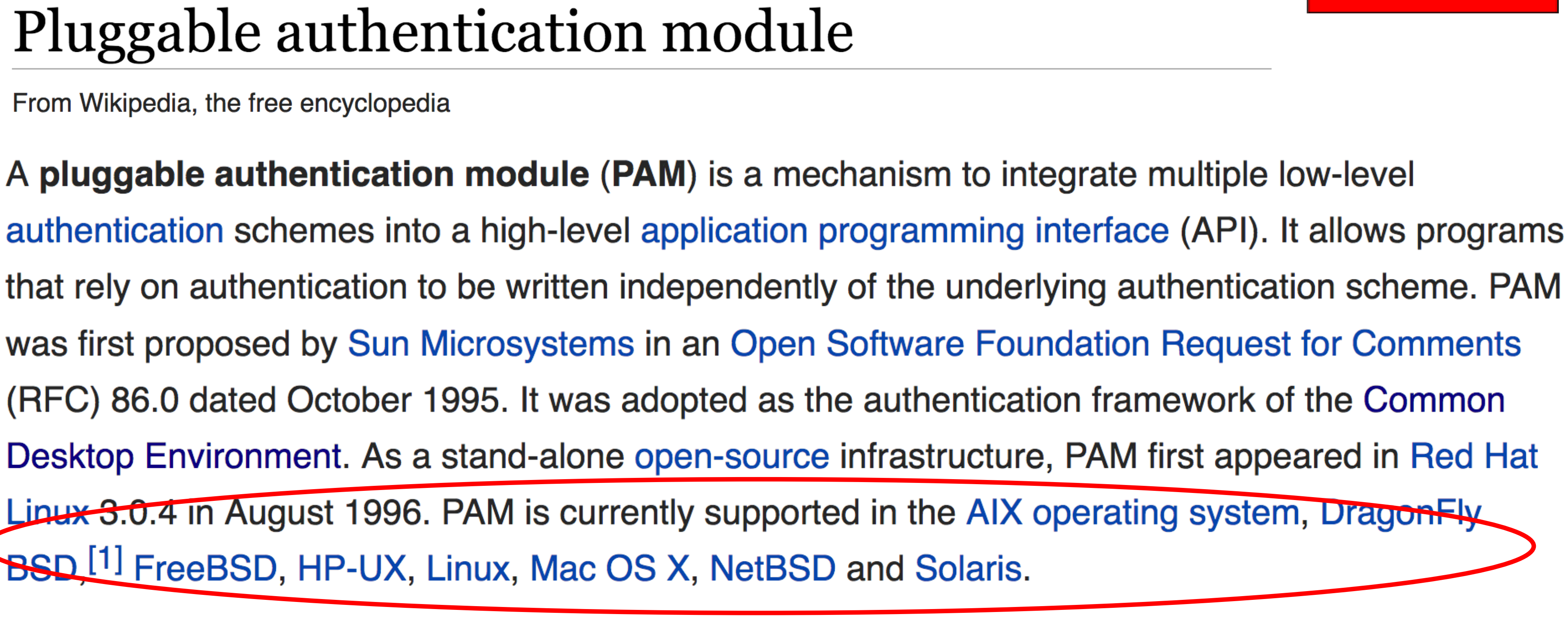
lookup
nss

directory
ldap

mediator

Building Blocks - AuthN

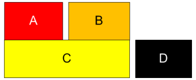
pam



Pluggable authentication module

From Wikipedia, the free encyclopedia

A **pluggable authentication module** (**PAM**) is a mechanism to integrate multiple low-level authentication schemes into a high-level application programming interface (API). It allows programs that rely on authentication to be written independently of the underlying authentication scheme. PAM was first proposed by Sun Microsystems in an Open Software Foundation Request for Comments (RFC) 86.0 dated October 1995. It was adopted as the authentication framework of the Common Desktop Environment. As a stand-alone open-source infrastructure, PAM first appeared in Red Hat Linux 3.0.4 in August 1996. PAM is currently supported in the AIX operating system, DragonFly BSD,[1] FreeBSD, HP-UX, Linux, Mac OS X, NetBSD and Solaris.



Pluggable Authentication Module

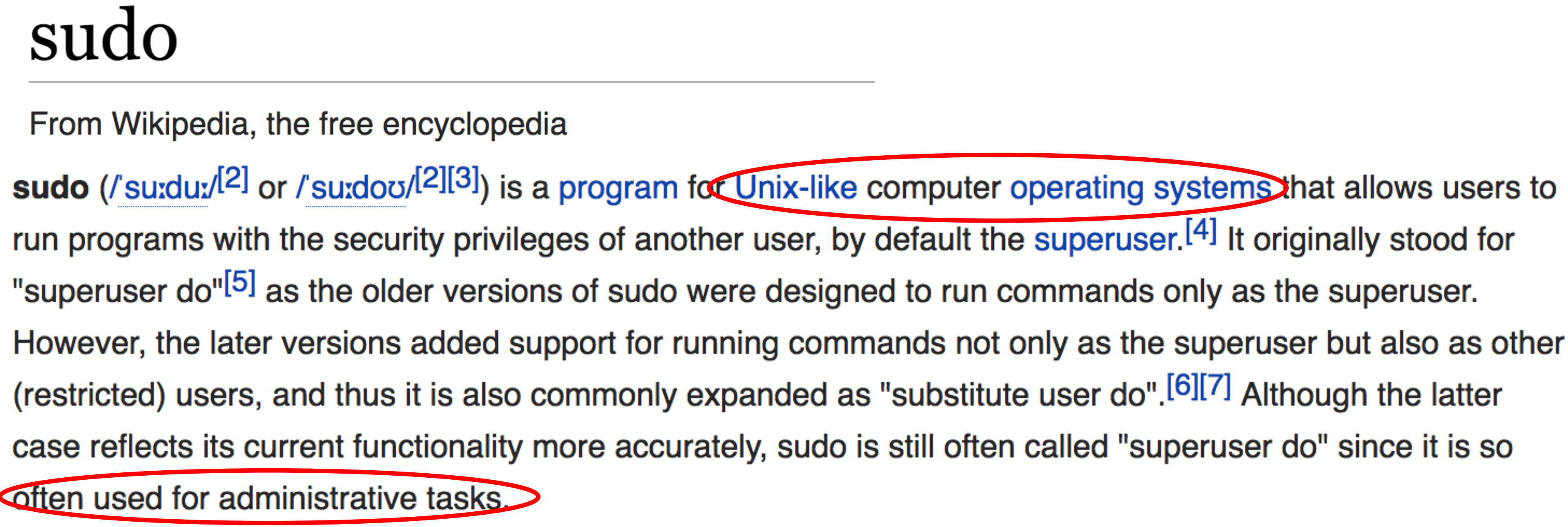
pam

- Authentication
- Coarse-grained Authorization

Just an authN service

Building Blocks - AuthZ

sudo



sudo

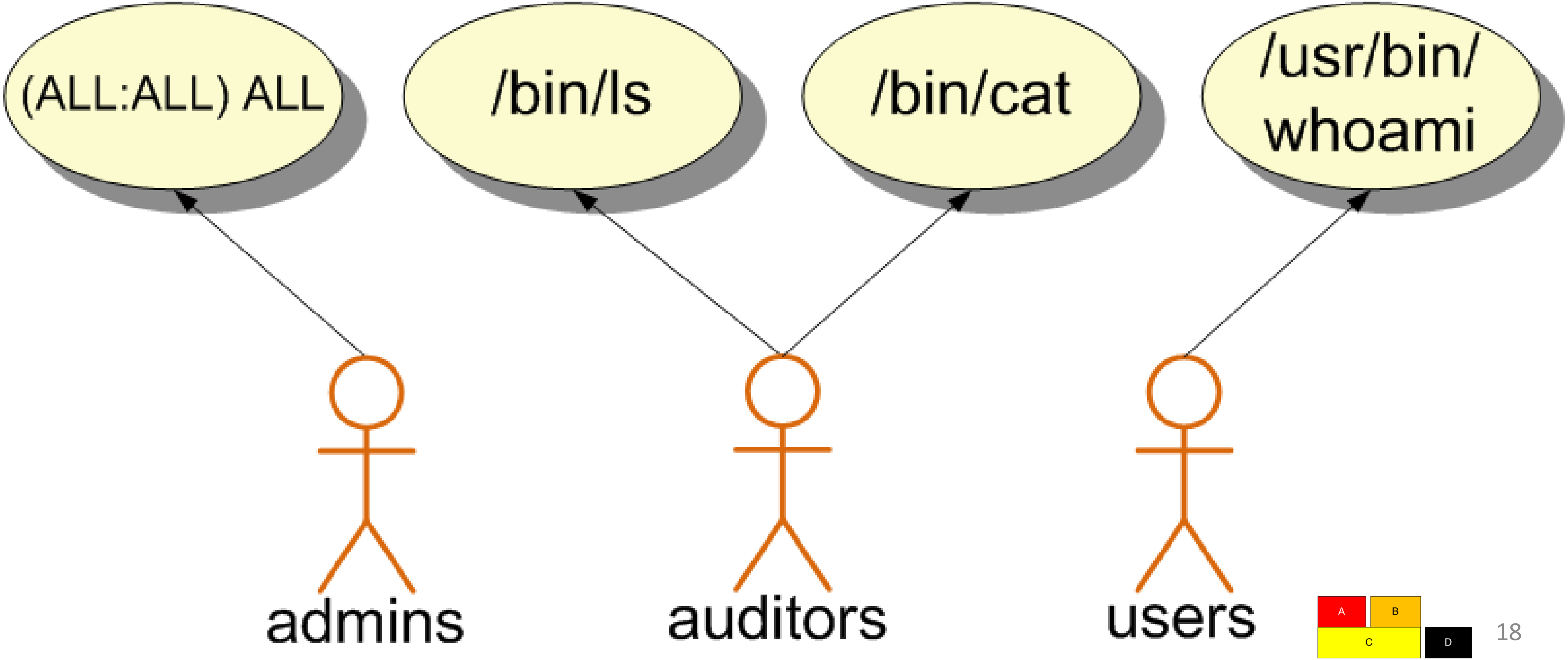
From Wikipedia, the free encyclopedia

sudo (/ˈsuːduː/[2] or /ˈsuːdoʊ/[2][3]) is a program for Unix-like computer operating systems that allows users to run programs with the security privileges of another user, by default the superuser.[4] It originally stood for "superuser do"[5] as the older versions of sudo were designed to run commands only as the superuser. However, the later versions added support for running commands not only as the superuser but also as other (restricted) users, and thus it is also commonly expanded as "substitute user do".[6][7] Although the latter case reflects its current functionality more accurately, sudo is still often called "superuser do" since it is so often used for administrative tasks.

sudo

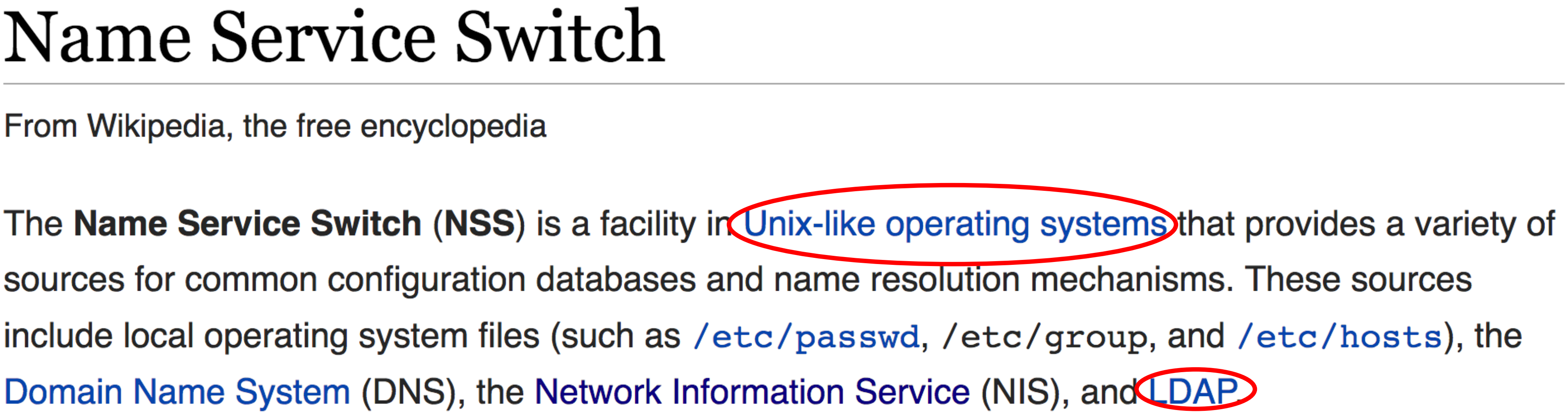
Just an authZ service

sudo



Building Blocks – Reporting

nss



Name Service Switch

From Wikipedia, the free encyclopedia

The **Name Service Switch** (**NSS**) is a facility in Unix-like operating systems that provides a variety of sources for common configuration databases and name resolution mechanisms. These sources include local operating system files (such as `/etc/passwd`, `/etc/group`, and `/etc/hosts`), the Domain Name System (DNS), the Network Information Service (NIS), and LDAP.

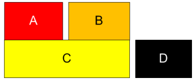
A B
C D

Name Service Switch

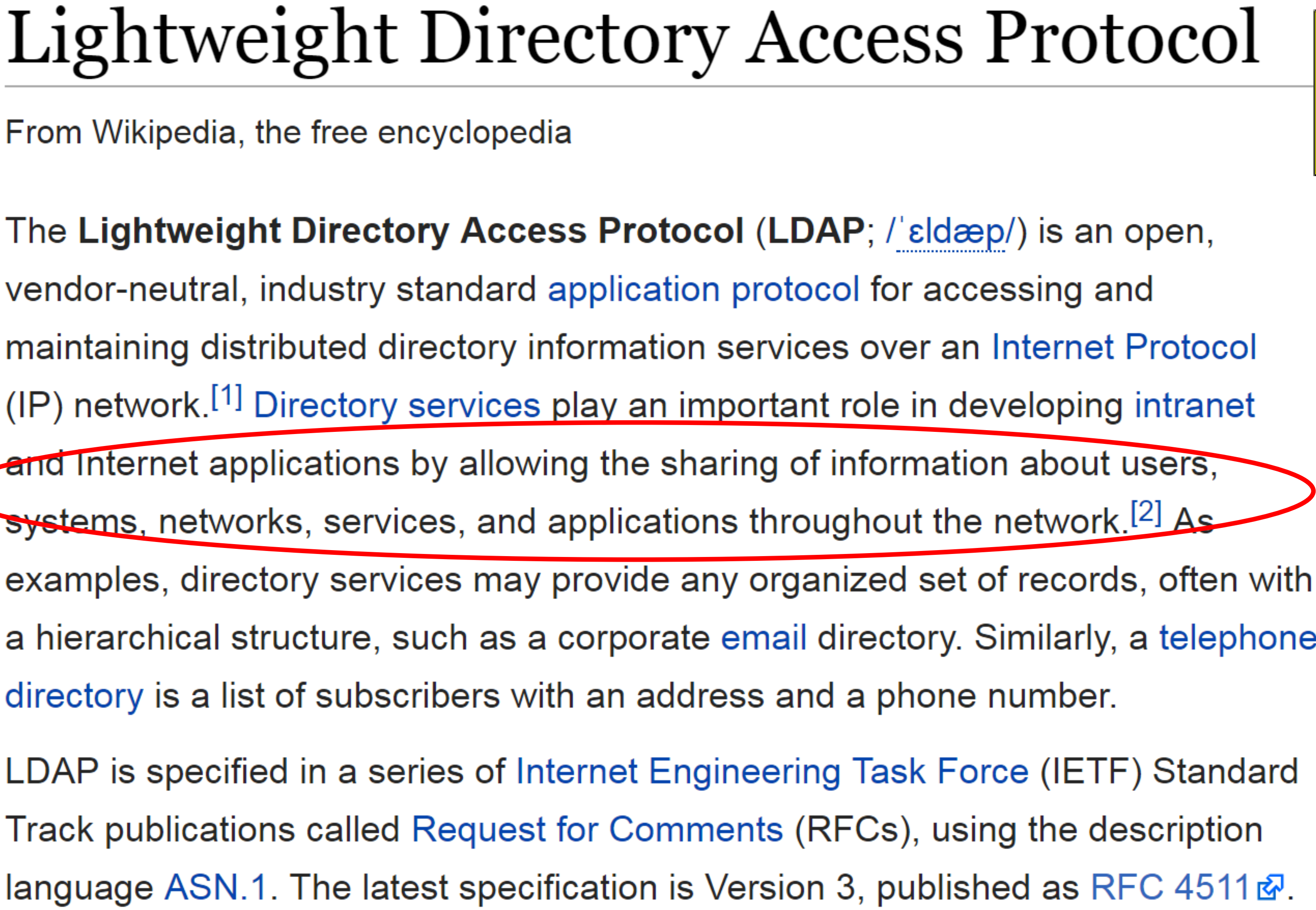
nss

- Used by unix processes to lookup user and group info

Just a lookup service



ldap



Lightweight Directory Access Protocol

From Wikipedia, the free encyclopedia

The **Lightweight Directory Access Protocol** (**LDAP**; /ˈɛldæp/) is an open, vendor-neutral, industry standard application protocol for accessing and maintaining distributed directory information services over an Internet Protocol (IP) network.[1] Directory services play an important role in developing intranet and Internet applications by allowing the sharing of information about users, systems, networks, services, and applications throughout the network.[2] As examples, directory services may provide any organized set of records, often with a hierarchical structure, such as a corporate email directory. Similarly, a telephone directory is a list of subscribers with an address and a phone number.

LDAP is specified in a series of Internet Engineering Task Force (IETF) Standard Track publications called Request for Comments (RFCs), using the description language ASN.1. The latest specification is Version 3, published as RFC 4511.

A B C D

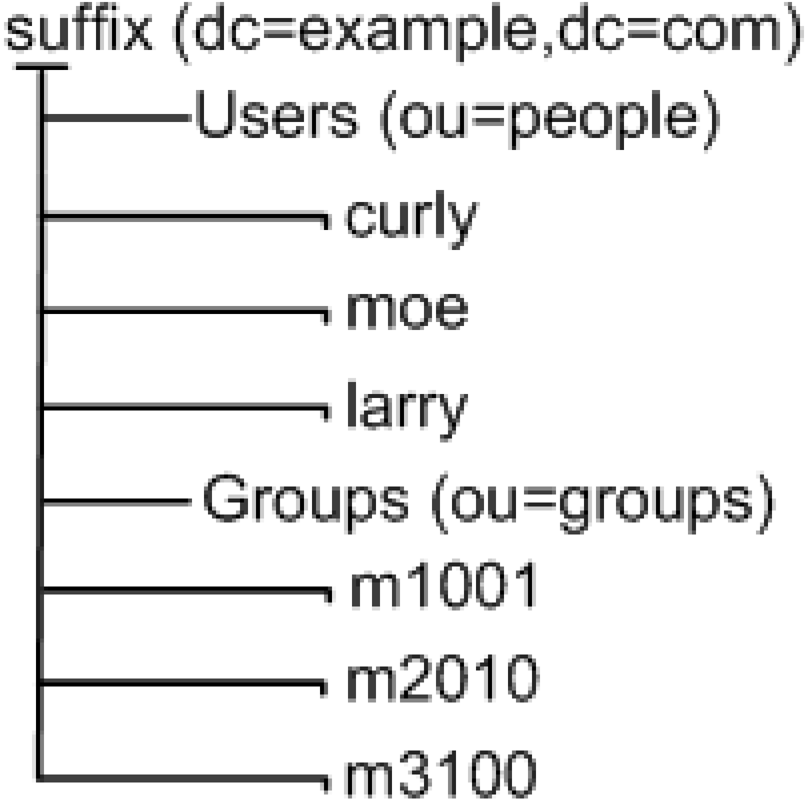
Building Blocks - LDAP

ldap

Just a

System of record

- Users
- Passwords
- Groups



suffix (dc=example,dc=com)

Users (ou=people)

curly

moe

larry

Groups (ou=groups)

m1001

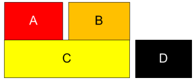
m2010

m3100

Building Blocks - Mediator

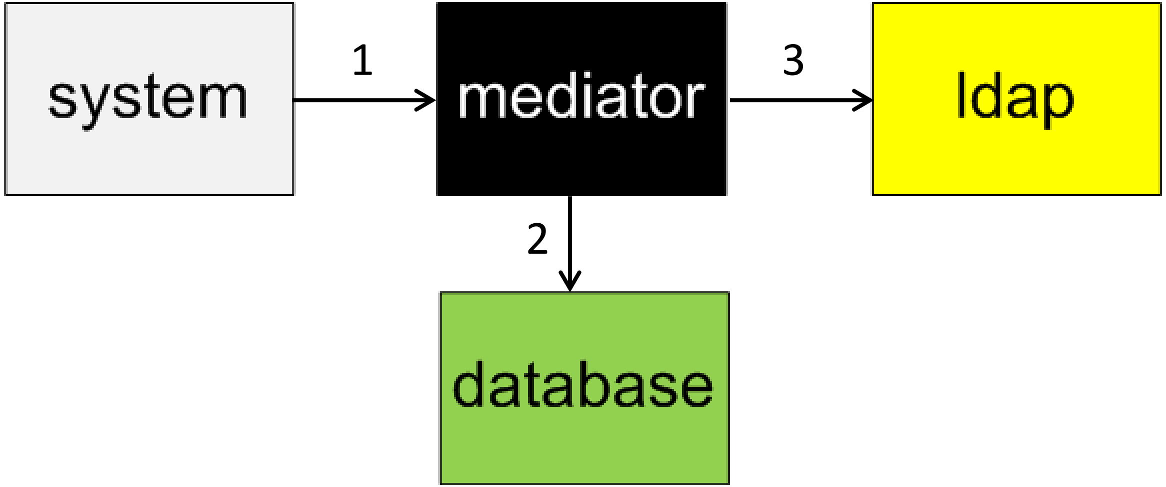
mediator

- Keeps things in synch between the machines and LDAP as things change.

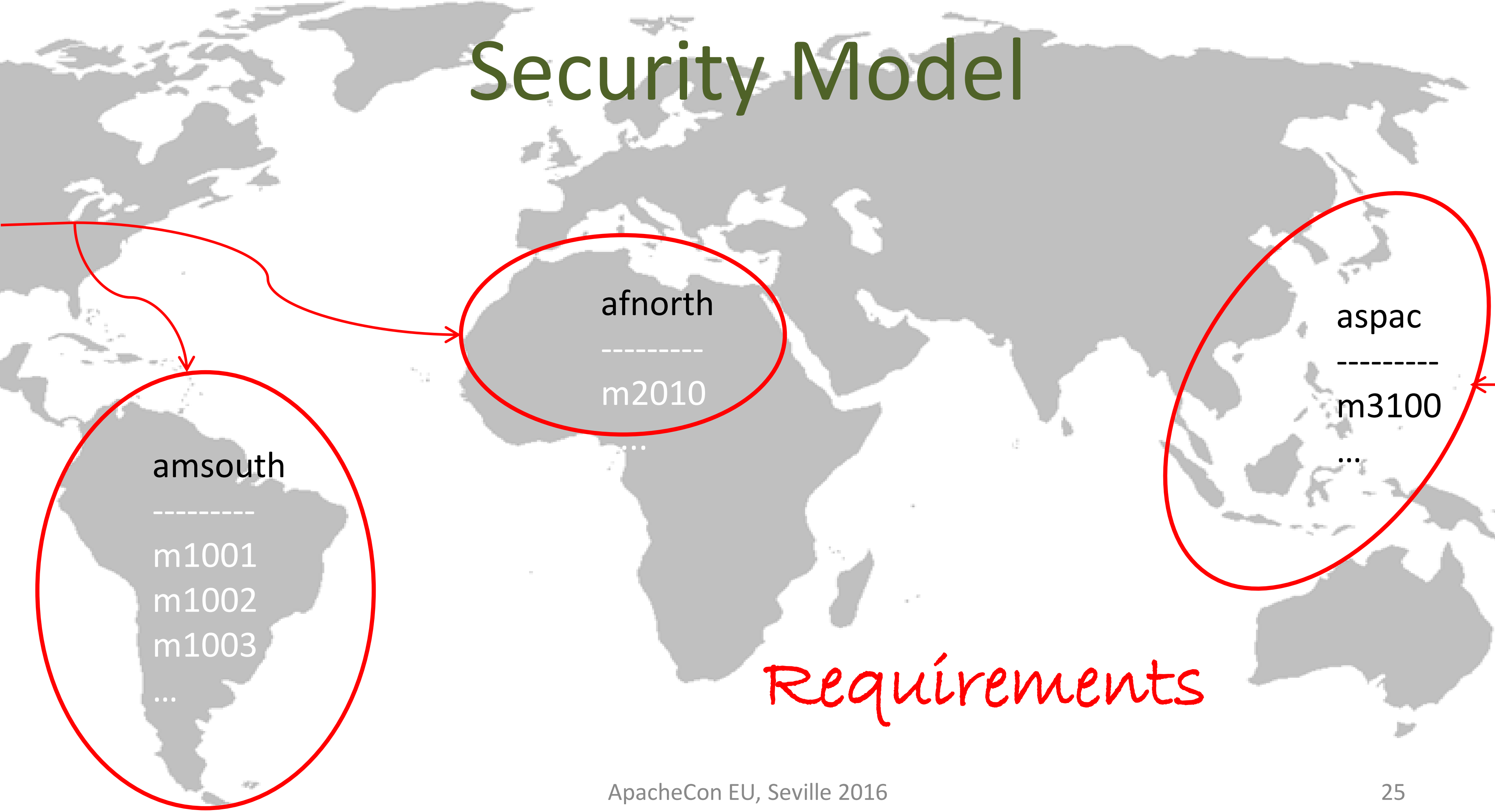


Mediator

1. Machine added to network, notifies mediator
2. Based on policies stored in DB
3. Updates ldap accordingly



Security Model



amsouth

----

m1001

m1002

m1003

…

afnorth

----

m2010

…

aspac

----

m3100

…

Requirements

Three Kinds of Security Checks

PAM

1. Authentication with LDAP
2. Coarse-grained authZ - memberOf target machine
   - (i.e. LDAP group name == hostname)

sudo

3. Medium-grained authZ. memberOf at least one:
   - Admin - root access
   - User - typical user access
   - Auditor - read-only access to entire machine.

symas

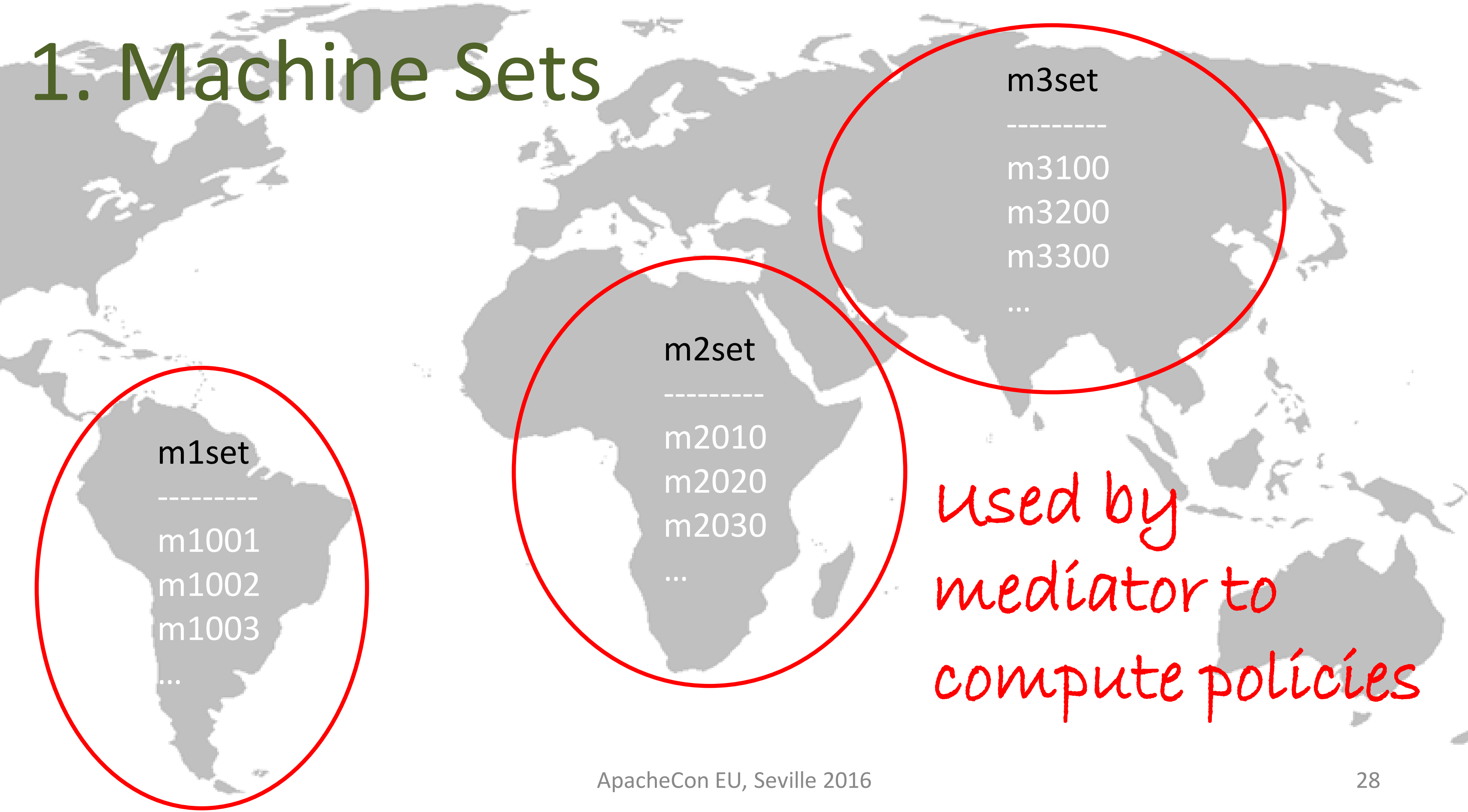
Three Types of Control Groups

mediator 1. Machine Sets

PAM 2. Machines

sudo 3. Security Roles

1. Machine Sets



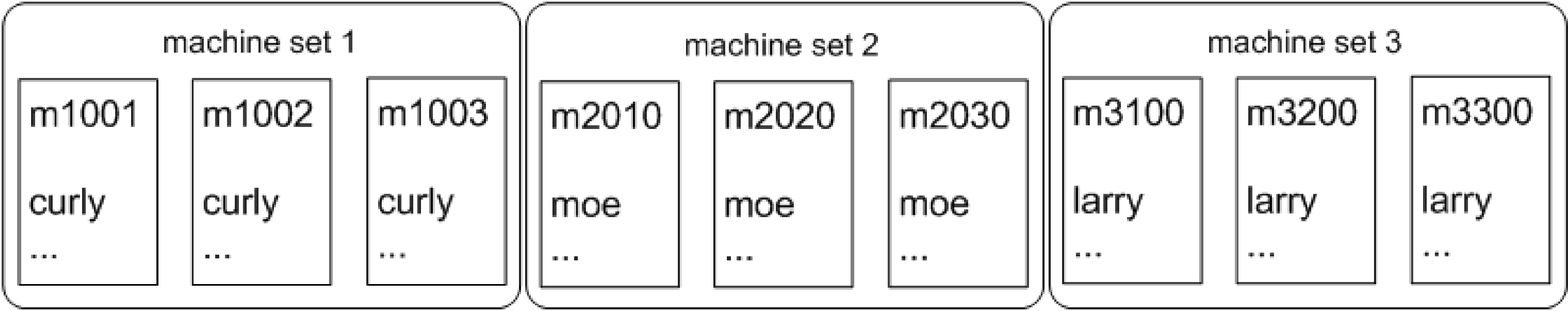
m1set
----
m1001
m1002
m1003
…

m2set
----
m2010
m2020
m2030
…

m3set
----
m3100
m3200
m3300
…

used by mediator to compute policies

2. Machines



machine set 1

m1001
curly
...

m1002
curly
...

m1003
curly
...

machine set 2

m2010
moe
...

m2020
moe
...

m2030
moe
...

machine set 3

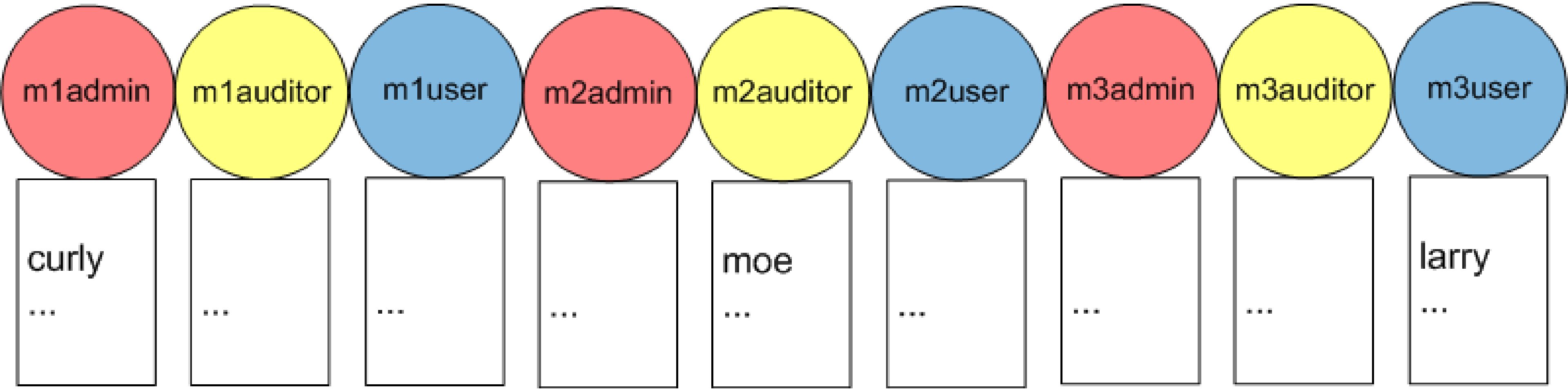
m3100
larry
...

m3200
larry
...

m3300
larry
...

used by PAM

3. Security Roles

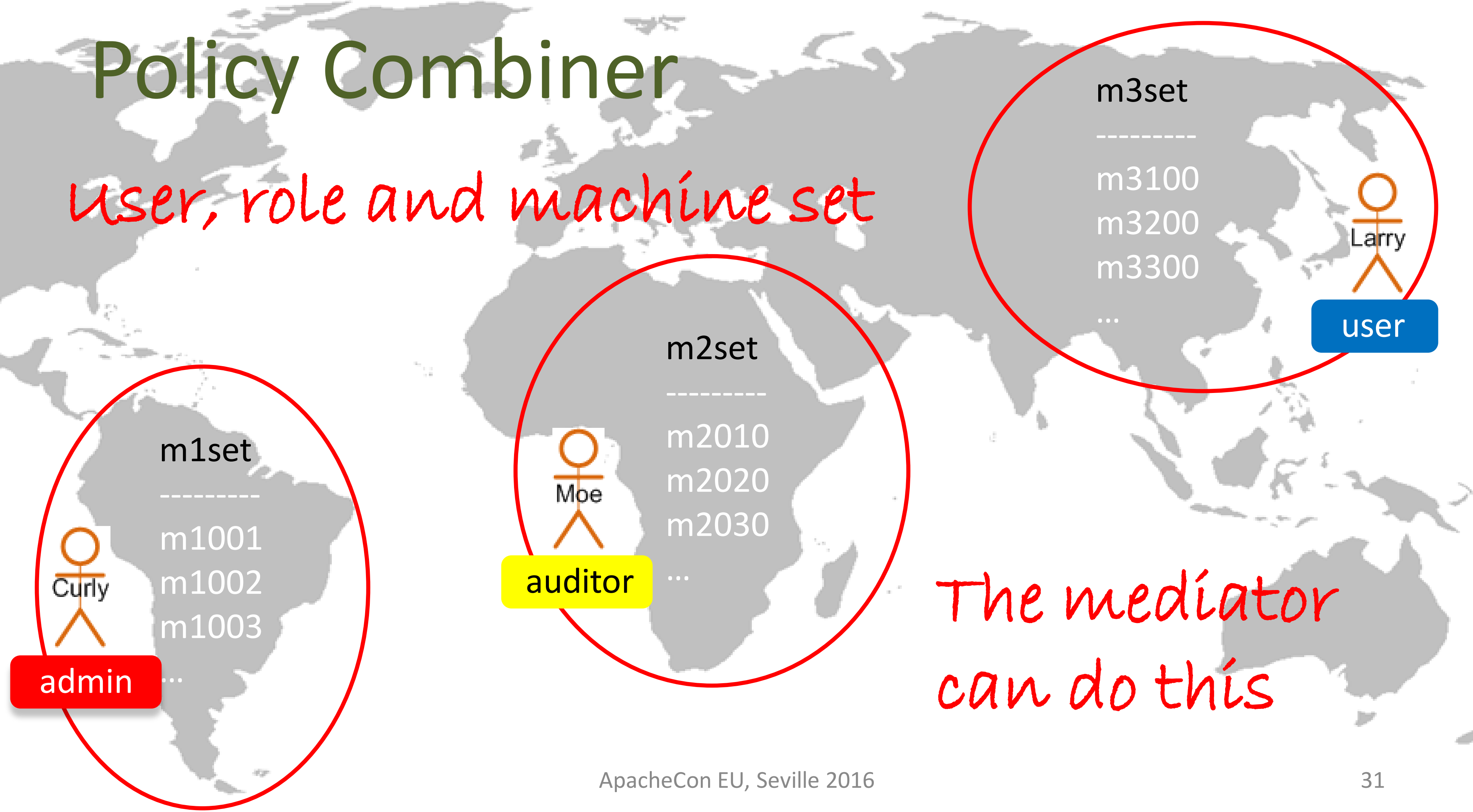


m1admin	m1auditor	m1user	m2admin	m2auditor	m2user	m3admin	m3auditor	m3user
curly				moe				larry
…	…	…	…	…	…	…	…	…

used by sudo

Policy Combiner

user, role and machine set



m3set
----
m3100
m3200
m3300
…

Larry
user

m2set
----
m2010
m2020
m2030
…

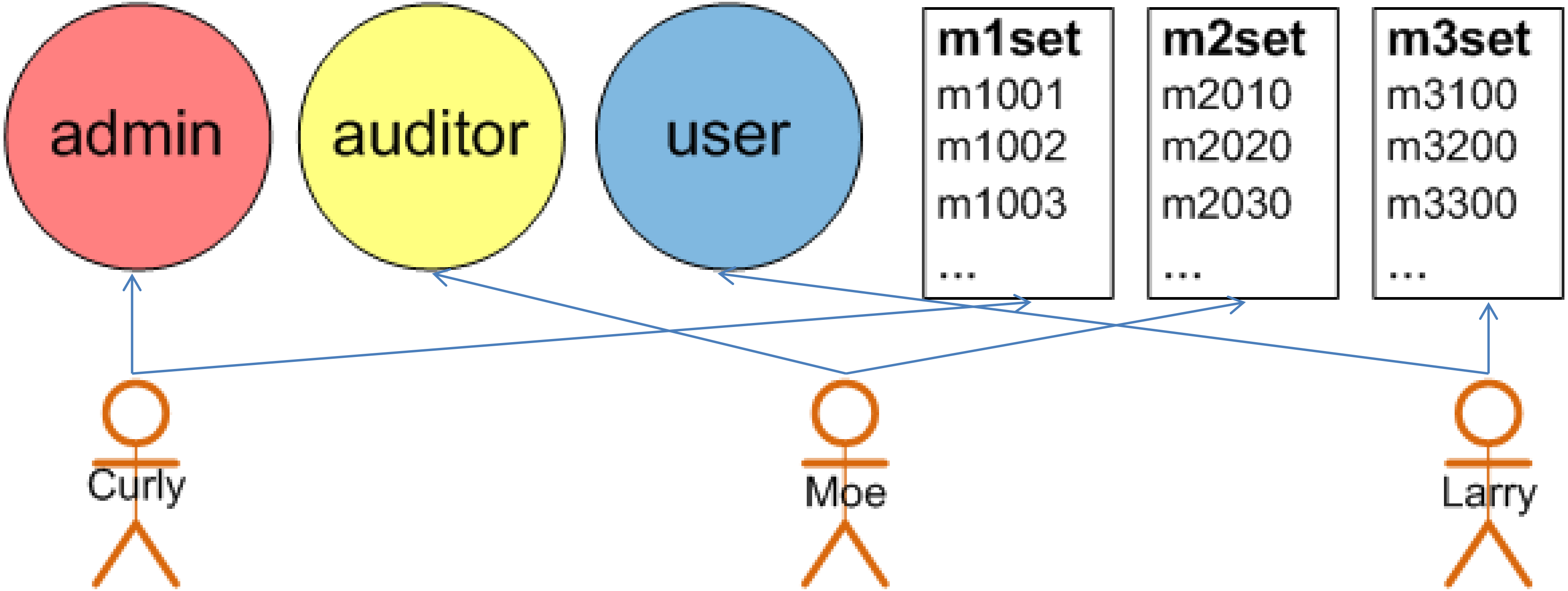
Moe
auditor

m1set
----
m1001
m1002
m1003
…

Curly
admin

The mediator can do this

Pick Two


admin
auditor
user
m1set
m1001
m1002
m1003
...
m2set
m2010
m2020
m2030
...
m3set
m3100
m3200
m3300
...
Curly
Moe
Larry

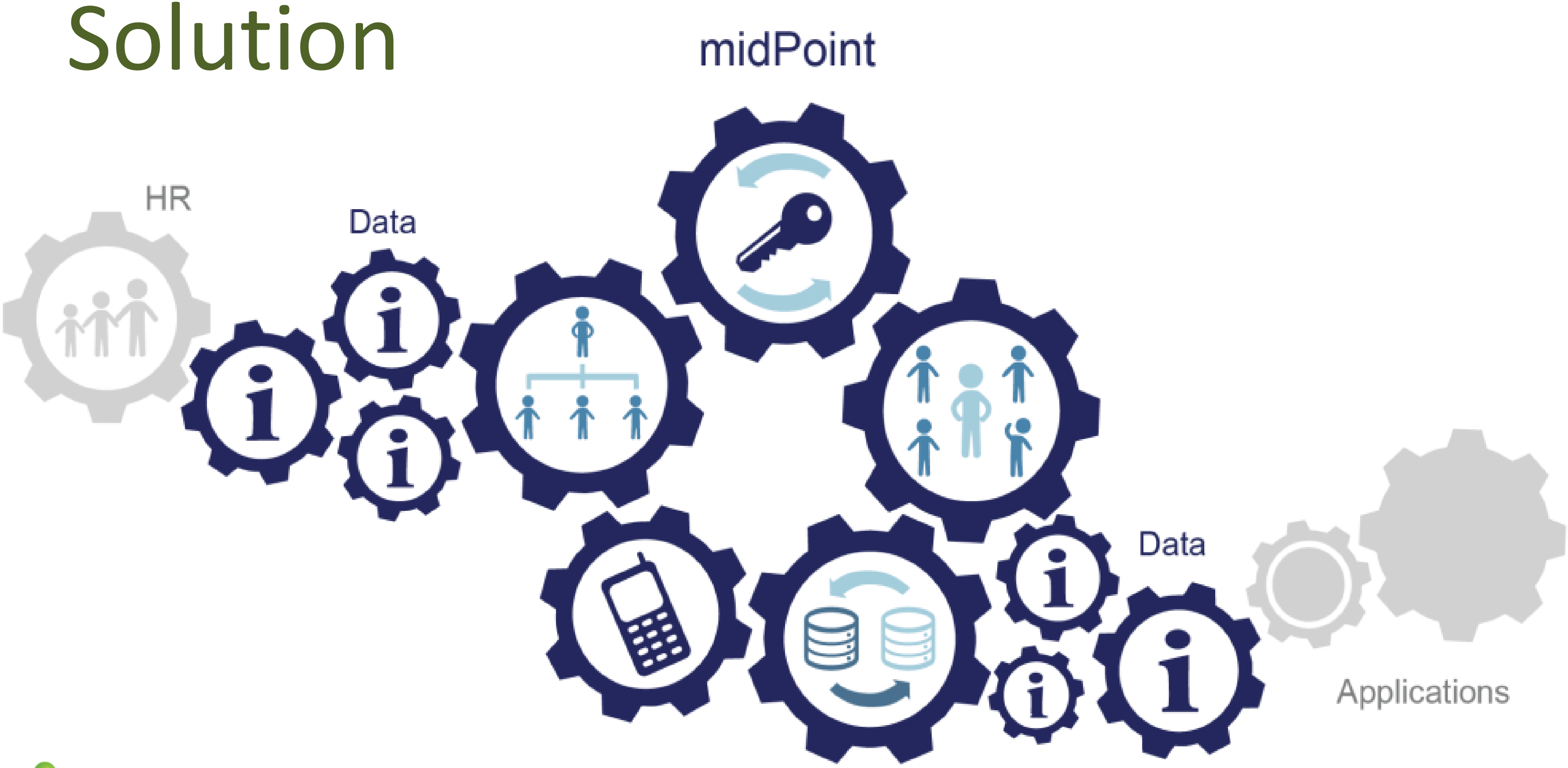
Solution

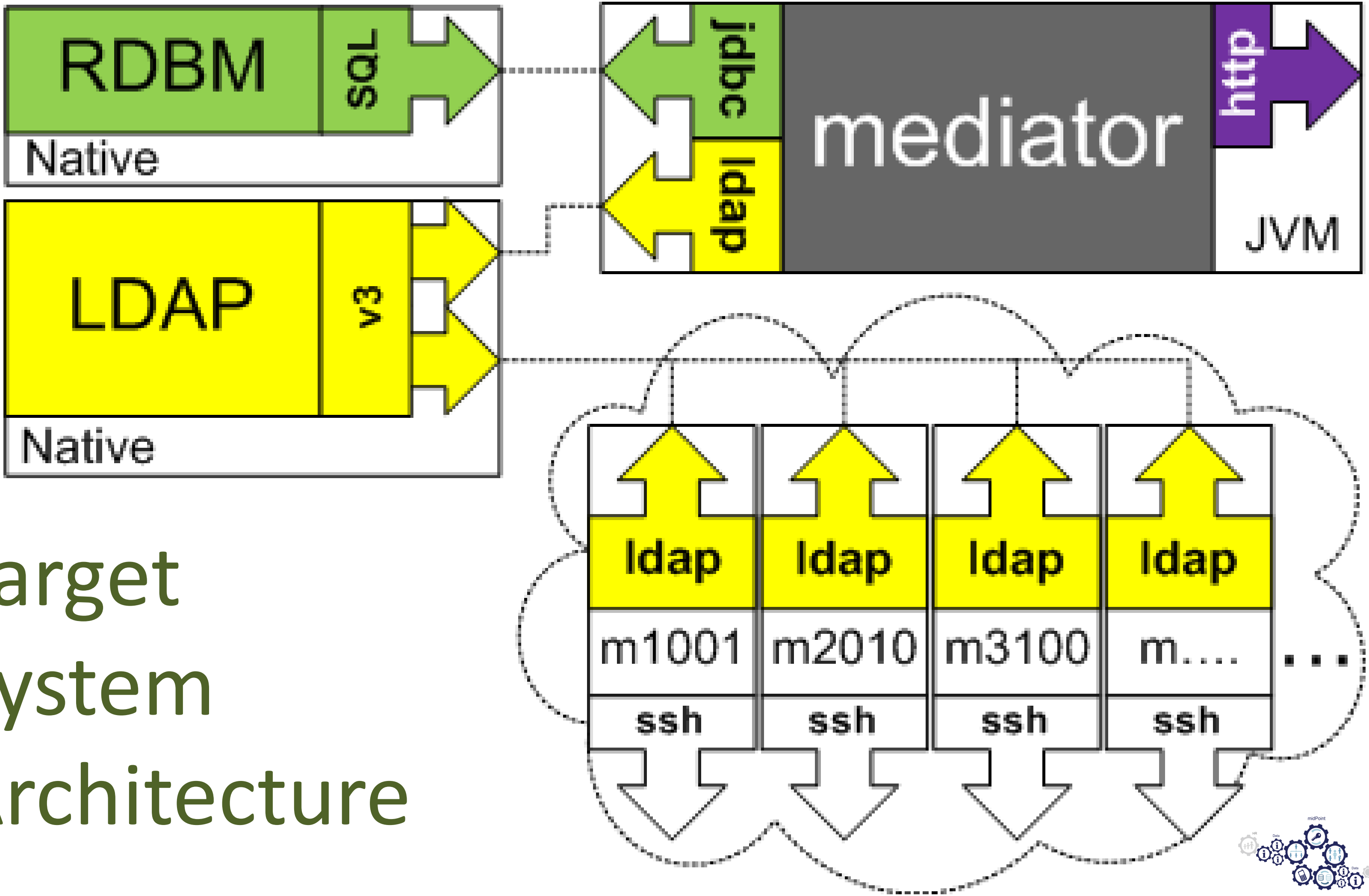
- How to manage users, machines and their access rights in the most flexible and dynamic way as possible?
- Chef? Puppet? Ansible?
- IaaS?
- Identity management?

Solution

- Blueprints for machines
  - Start new machine, set up new machine,…
- Identity management product for managing
  - users,
  - security and access groups for machines,
  - access rights for users

Solution



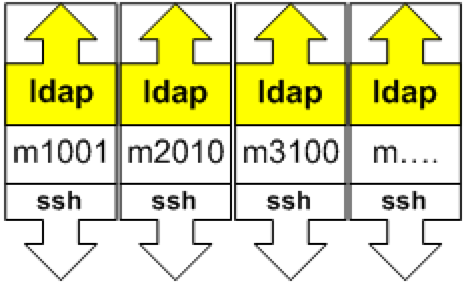


Target System Architecture

Client-side Solution

Script during machine instantiation:

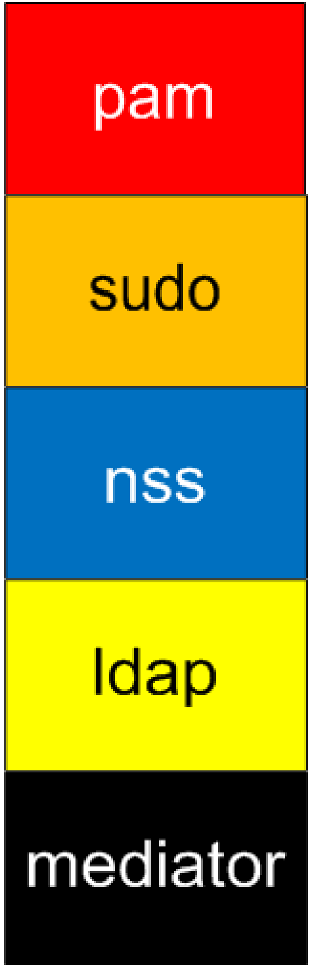
1. Configures pam, sudo & nss to LDAP
2. Call mediator to add LDAP machine group
3. Call mediator to recompute LDAP groups



ldap | ldap | ldap | ldap

m1001 | m2010 | m3100 | m....

ssh | ssh | ssh | ssh



pam

sudo

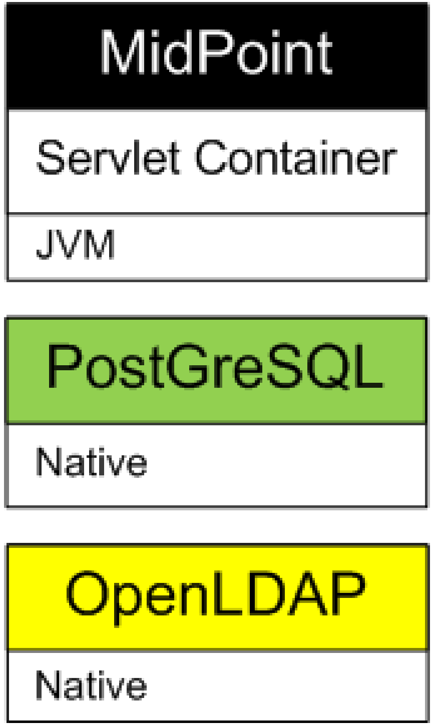
nss

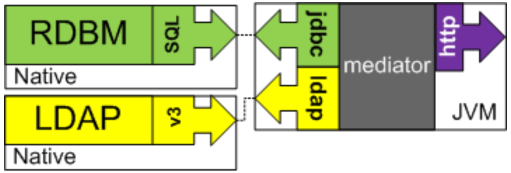
ldap

mediator

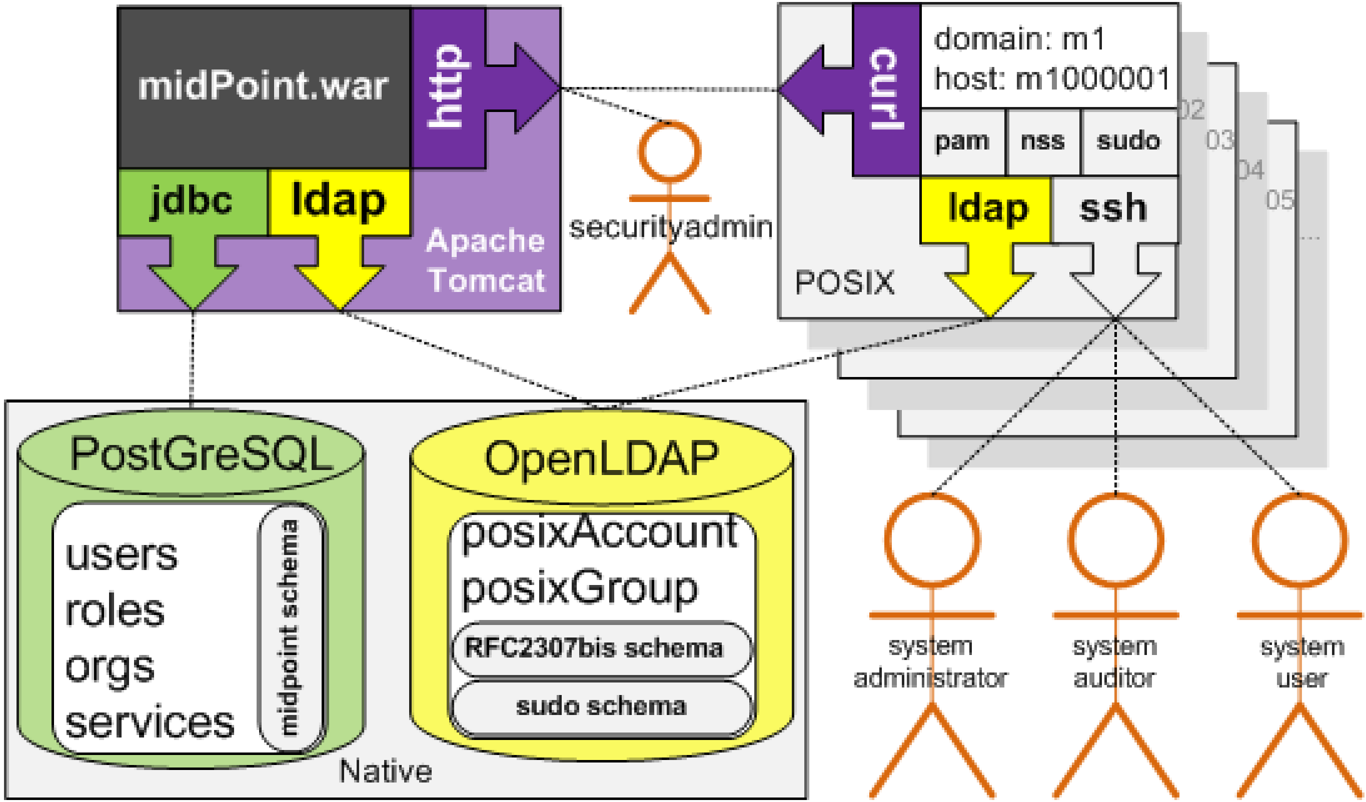
Server-side Solution

1. MidPoint - mediator
   - delegated admin, approvals, audit
   - html & http admin services
2. PostGreSQL – master database
   - users, roles, orgs, svcs
3. OpenLDAP – security database
   - users, groups
   - posixAccount, posixGroup

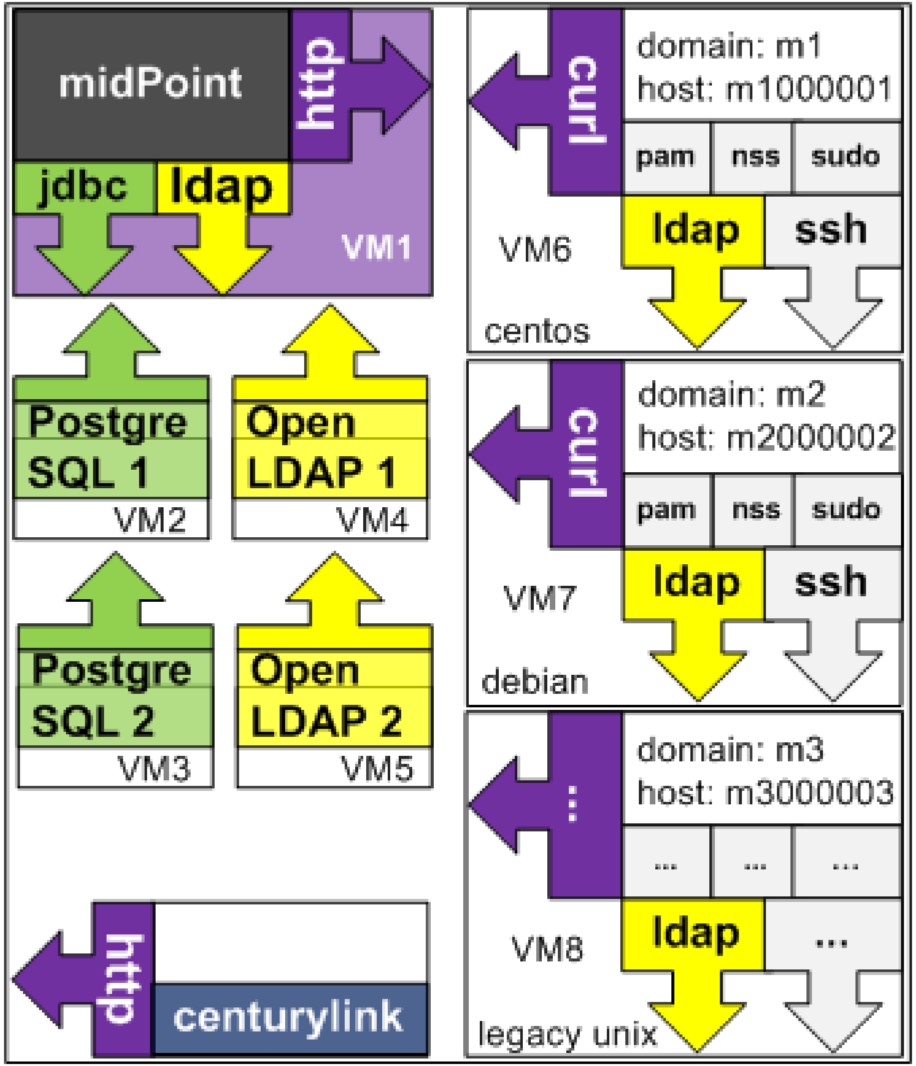




High-level Solution Design



Detail Design



symas

Data Models

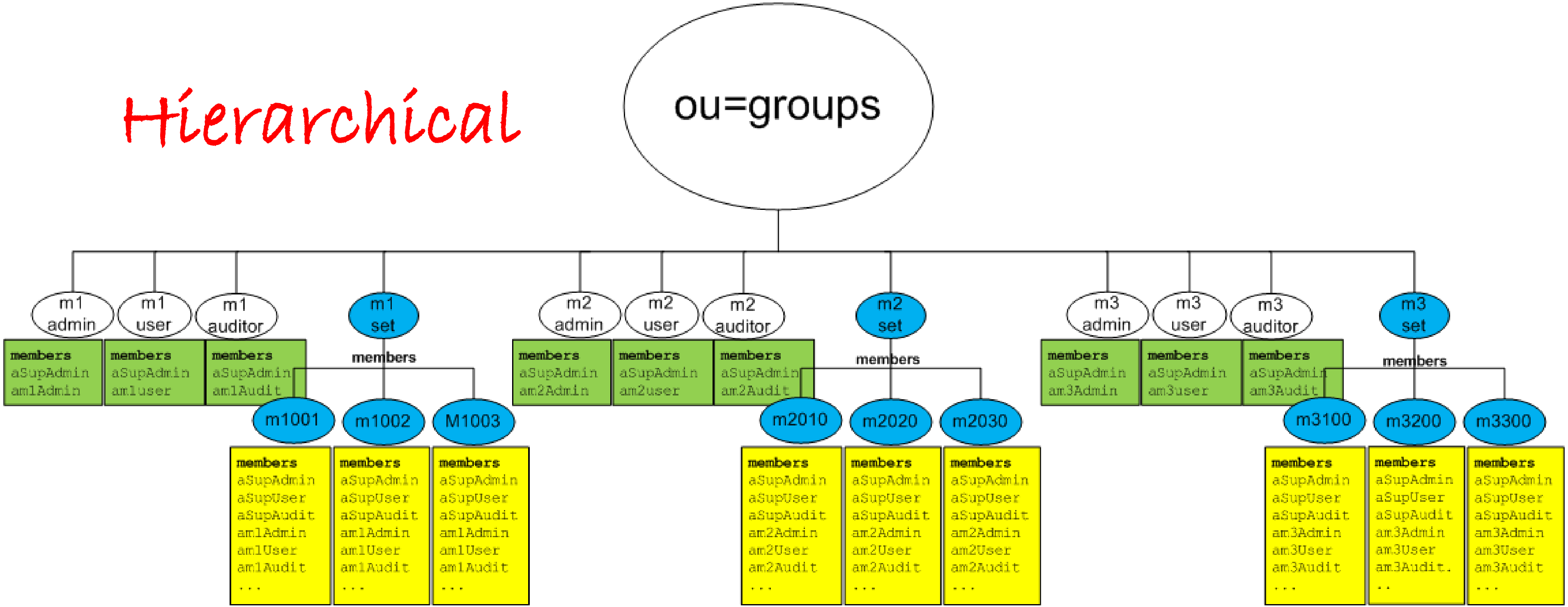
LDAP Data Model

Standard object schemas:

1. RFC2307bis
   - posixAccount
   - posixGroup
2. SudoRole

LDAP Data Model

Hierarchical



ou=groups

m1 admin
members
aSupAdmin
am1Admin

m1 user
members
aSupAdmin
am1user

m1 auditor
members
aSupAdmin
am1Audit

m1 set
members

m1001
members
aSupAdmin
aSupUser
aSupAudit
am1Admin
am1User
am1Audit
...

m1002
members
aSupAdmin
aSupUser
aSupAudit
am1Admin
am1User
am1Audit
...

M1003
members
aSupAdmin
aSupUser
aSupAudit
am1Admin
am1User
am1Audit
...

m2 admin
members
aSupAdmin
am2Admin

m2 user
members
aSupAdmin
am2user

m2 auditor
members
aSupAdmin
am2Audit

m2 set
members

m2010
members
aSupAdmin
aSupUser
aSupAudit
am2Admin
am2User
am2Audit
...

m2020
members
aSupAdmin
aSupUser
aSupAudit
am2Admin
am2User
am2Audit
...

m2030
members
aSupAdmin
aSupUser
aSupAudit
am2Admin
am2User
am2Audit
...

m3 admin
members
aSupAdmin
am3Admin

m3 user
members
aSupAdmin
am3user

m3 auditor
members
aSupAdmin
am3Audit

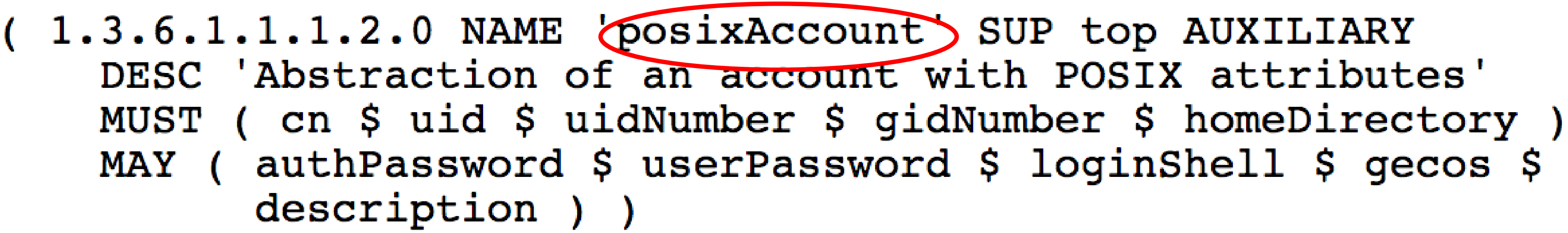
m3 set
members

m3100
members
aSupAdmin
aSupUser
aSupAudit
am3Admin
am3User
am3Audit
...

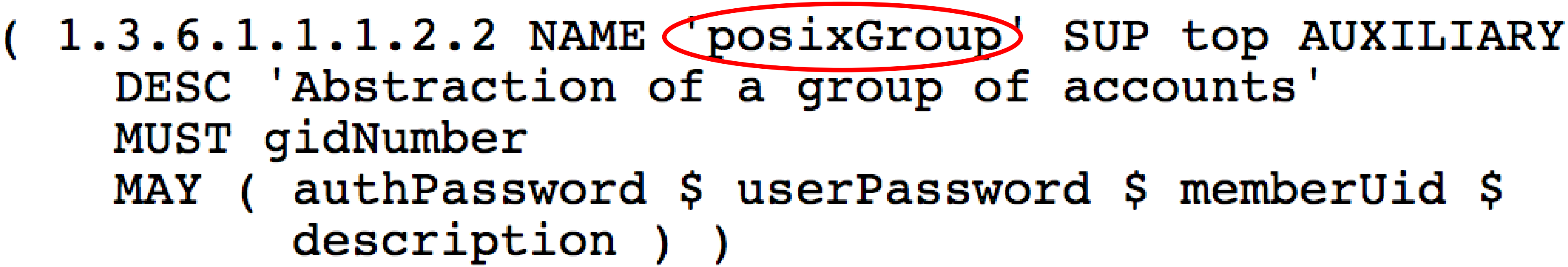
m3200
members
aSupAdmin
aSupUser
aSupAudit
am3Admin
am3User
am3Audit.
..

m3300
members
aSupAdmin
aSupUser
aSupAudit
am3Admin
am3User
am3Audit
...

Use RFC2307bis LDAP Schema



```
( 1.3.6.1.1.1.2.0 NAME 'posixAccount' SUP top AUXILIARY
    DESC 'Abstraction of an account with POSIX attributes'
    MUST ( cn $ uid $ uidNumber $ gidNumber $ homeDirectory )
    MAY ( authPassword $ userPassword $ loginShell $ gecos $
          description ) )
```



```
( 1.3.6.1.1.1.2.2 NAME 'posixGroup' SUP top AUXILIARY
    DESC 'Abstraction of a group of accounts'
    MUST gidNumber
    MAY ( authPassword $ userPassword $ memberUid $
          description ) )
```

Machine Set M1

```
dn: cn=m1set, ou=Groups, ...
description: Machine Set 1
member: cn=m1001,...
member: cn=m1002,...
member: cn=m1003,...
```

…

Machine M1001

```
dn: cn=m1001, ou=Groups,…
objectClass: posixGroup
description: Machine Group M1001
member: uid=curly,ou=People,…
member: uid=frank,ou=People,…
member: uid=marla,ou=People,…
…
```

Security Role M1Admin

```
dn: cn=m1admin, ou=Groups, ...
objectClass: posixGroup
description: Admin Machine Set 1
cn: m1admin
member: uid=curly,ou=People,...
member: uid=frank,ou=People,...
member: uid=marla,ou=People,...
…
```

sudo LDAP Schema

```
objectclass ( 1.3.6.1.4.1.15953.9.2.1
   NAME 'sudoRole' SUP top STRUCTURAL
   DESC 'Sudoer Entries'
   MUST ( cn )
   MAY ( sudoUser $ sudoHost $ sudoCommand
  $ sudoRunAs $ sudoRunAsUser
  $ sudoRunAsGroup $ sudoOption
  $ sudoNotBefore $ sudoNotAfter
  $ sudoOrder $ description )
)
```

sudo M1Admin

```
dn: cn=admin access to
  m1,ou=sudo,dc=example,dc=com
objectClass: sudoRole
cn: admin access to m1
sudoUser: %m1admin
sudoHost: m1001
sudoHost: m1002
sudoHost: m1003
sudoHost: m1004
```

Provisioning Overview

- Why use IDM when it only complicates monitoring, req’s additional resources, upgrades, etc…
- How to adapt to the elastic cloud environment
- Chef, Puppet, Ansible – not enough

Basic Provisioning

1. Adding a new User synchs downstream
2. Adding a new Machine happens automatically
3. Scoping a Role to a Domain

Advanced Provisioning

- Security reports

Advanced Provisioning

- Security reports
- Governance and compliance

Advanced Provisioning

- Security reports
- Governance and compliance
- Temporal assignments

Advanced Provisioning

- Security reports
- Governance and compliance
- Temporal assignments
- Auditing

Advanced Provisioning

- Security reports
- Governance and compliance
- Temporal assignments
- Auditing
- Additional systems

More Advanced Provisioning

- How do you know which permissions the user has?

More Advanced Provisioning

- How do you know which permissions the user has?
- Which of these permissions are obsolete and should be denied?

More Advanced Provisioning

- How do you know which permissions the user has?
- Which of these permissions are obsolete and should be denied?
- Who and why were these permissions assigned to the user?

And Still More

- Additional target systems, applications, monitoring

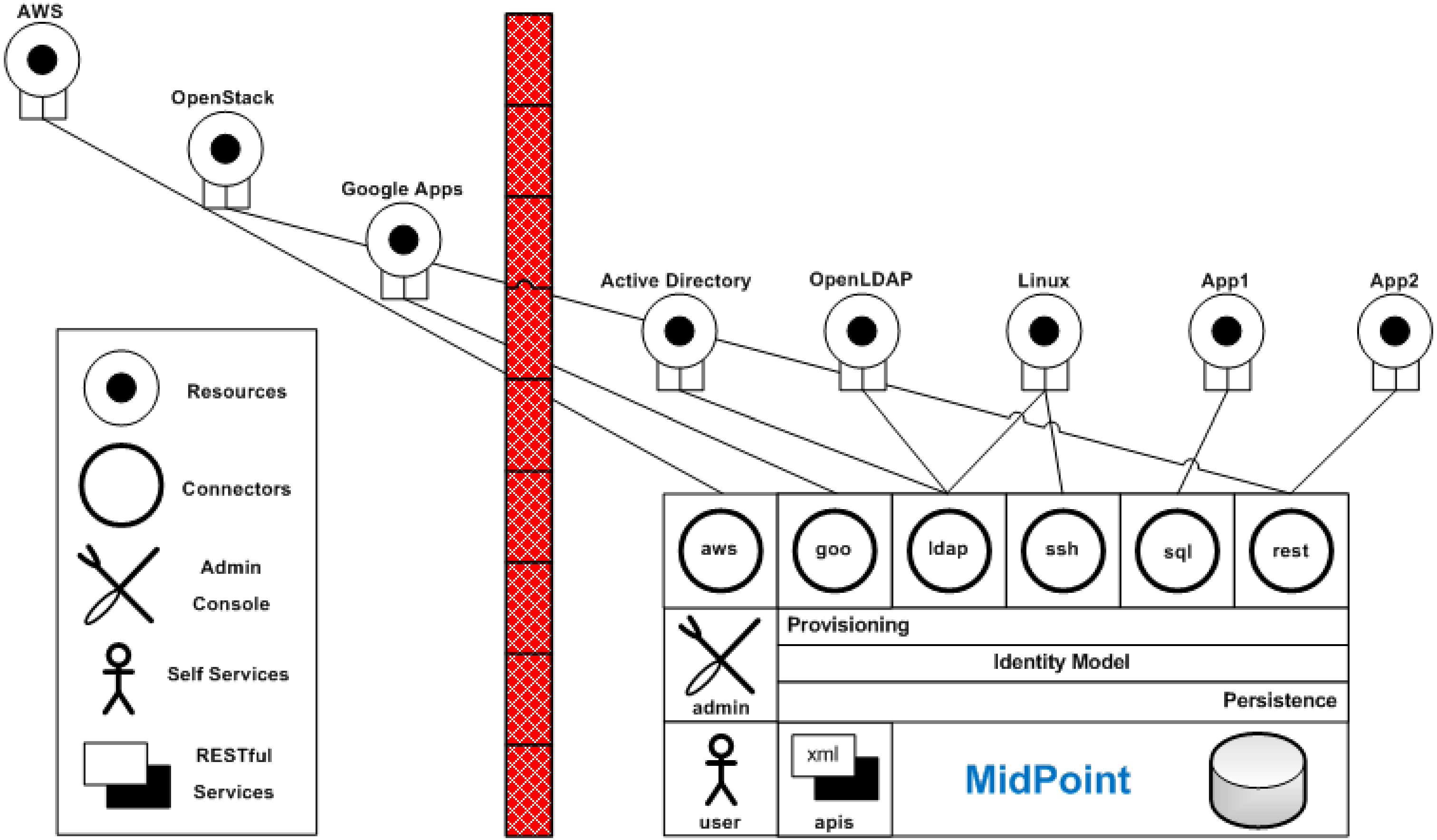
And Still More

- Additional target systems, applications, monitoring
- Handling changes made outside mediator

And Still More

- Additional target systems, applications, monitoring
- Handling changes made outside mediator
- Network crashes, ...

Midpoint (mediator)



Use Cases

Manage a large cluster of machines for a technology company with 100 employees and 100,000 customers.

Overview

- Using Debian and Redhat systems
- Deploys into the cloud
- Maintain strict control

Demo User to Role to Machine

	<----- Set 1------>			<-------Set 2 ------>			<----- Set 3 ----->		
User-Role-Machine	**m1001**	**m1002**	**m1003**	**m2010**	**m2020**	**m2030**	**m3100**	**m3200**	**m3300**
Curly	Admin	Admin	Admin						
Moe				Auditor	Auditor	Auditor			
Larry							User	User	User

Demo Intro

- Create new machine
- Nothing up my sleeve

Use Case 1

Create a New Machine

- Assigns Users to Machine and Security Groups
- Log onto new machine

Curly

Use Case 2

Assign User to Role

- Add to Security Role
- Add to Machine Groups
- Delegated Admin
- Self service

Moe

Use Case 3

Deassign User Role

- Remove User from Machine and Security Groups

Larry

Use Case 4

Remove a Machine

- Deletes the Machine Group from LDAP

Use Case N

- Approvals
- Temporal based assignments
- Audit trail

Wrap-up

- Built on Open Source Solutions
- Cookbooks published soon
- There is no security without identity management. -- Radovan Semancik

Contact Us

- Katarina Valalikova
  - @KValalikova
  - k.valalikova@evolveum.com
- Shawn McKinney
  - @shawnmckinney
  - smckinney@symas.com