OSS4B
(OPEN SOURCE SOFTWARE FOR BUSINESS)

# Building Open Source Identity Management with FreeIPA

**Martin Kosek**

mkosek@redhat.com

http://www.oss4b.it/

**OSS4B 2013 - Open Source Software for Business**

19-20 September 2013, Monash University Prato Centre

Prato, Tuscany, Italy

# Agenda

- What is Identity Management
- Meet Active Directory
- Introduce FreeIPA - Features and examples
- Addressing interoperability with Active Directory

# Getting a Context

- What is identity management?
  - “Identity management (IdM) describes the management of individual principals, their authentication, authorization, and privileges within or across system and enterprise boundaries with the goal of increasing security and productivity while decreasing cost, downtime and repetitive tasks.”

Wikipedia

- This is the theory, but what does it mean?
  - Identities (principals): users, machines, services/applications
  - Authentication: /etc/passwd, LDAP, NIS, Kerberos
  - Authorization: Policies, ACLs, DAC, MAC
  - Within or across systems: all this can be configured locally
    - May become a synchronization nightmare on network

# IdM Related Technologies

- Active Directory
  - Main identity management solution deployed in more than 90% of the enterprises
- LDAP (389 DS, OpenLDAP, eDirectory, SunDS, ...)
  - Often used for custom IdM solution
- Kerberos
  - Authentication
- Samba (Samba 4 DC, Samba FS, Winbind)
- NIS (NIS+) - obsoleted

# Active Directory vs. Open Source

- Why is Active Directory so popular?
  - **Integrated** solution
  - It is relatively **easy to use**
  - **Simple configuration** for clients
  - All the **complexity** is **hidden** from users and admins
  - Has comprehensive interfaces

# Active Directory vs. Open Source (2)

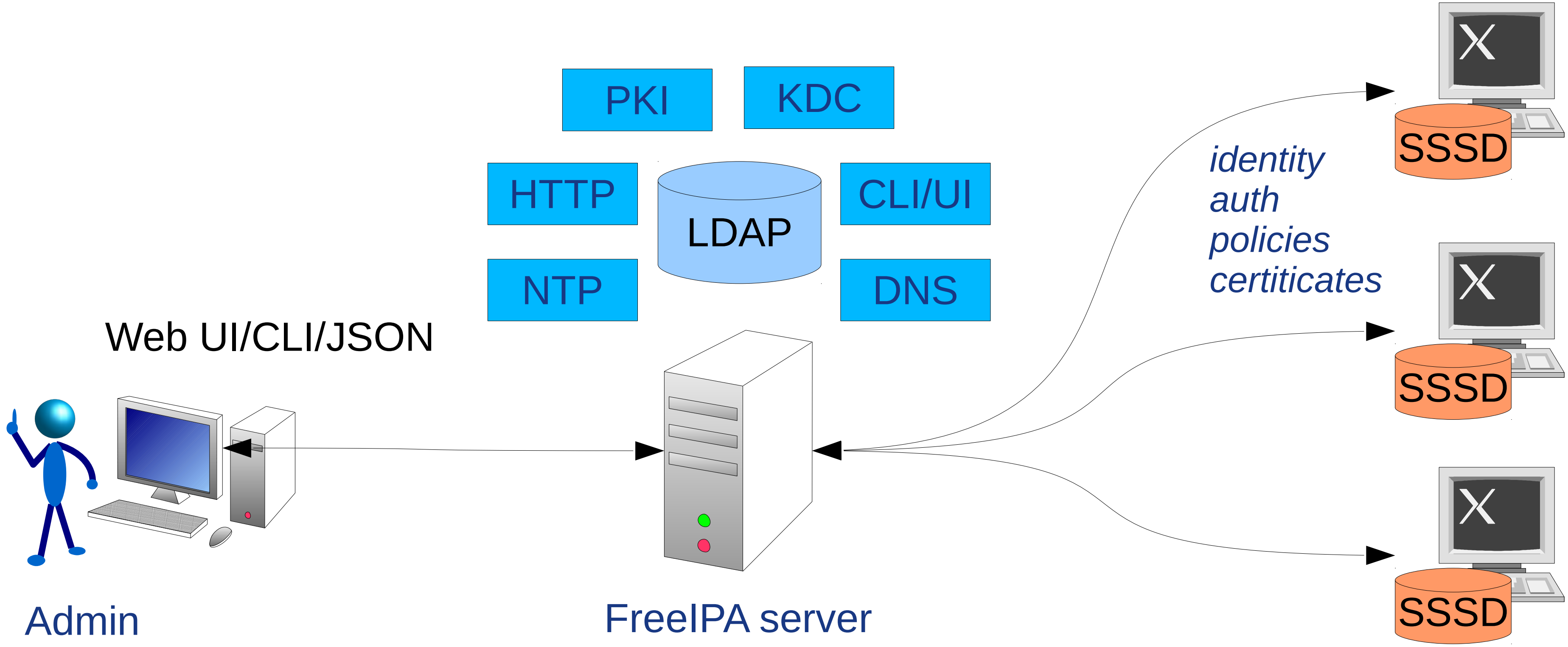
- What about Open Source tools?
  - Solve **individual** problems
    - *“do one thing and do it well”*
  - Bag of technologies lacking integration
  - Hard to install and configure
    - Have you ever tried manual LDAP+Kerberos configuration?
  - Too many options exposed
    - Which to choose? Prevent shooting myself in the leg
  - Lack of good user interfaces

**Is the situation really that bad?**

# Introducing FreeIPA

- IPA stands for Identity, Policy, Audit
  - So far we have focused on **identities** and related **policies**
- Main problems FreeIPA solves:
  - **Central management** of authentication and identities for Linux clients better than stand-alone LDAP/Kerberos/NIS - based solutions
  - Lets IdM available to regular admins, hides complexity
    - Install with one command, in several minutes
  - Acts as a **gateway** between the Linux infrastructure and AD environment making infrastructure more manageable and more cost effective
    - This is a requirement, as we said earlier, Active Directory is often the main Identity Management source
    - More about that topic later

# High-level architecture



# Example - Using FreeIPA CLI

```
$ kinit admin
Password for admin@EXAMPLE.COM:

$ klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: admin@EXAMPLE.COM

Valid starting       Expires              Service principal
10/15/12 10:47:35  10/16/12 10:47:34  krbtgt/EXAMPLE.COM@...
```

# Example - Using FreeIPA CLI (2)

```
$ ipa user-add --first=John --last=Doe jdoe --random
-----------------
Added user "jdoe"
-----------------
  User login: jdoe
  First name: John
  Last name: Doe
  Full name: John Doe
  Display name: John Doe
  Initials: JD
  Home directory: /home/jdoe
  GECOS field: John Doe
  Login shell: /bin/sh
  Kerberos principal: jdoe@EXAMPLE.COM
  Email address: jdoe@example.com
  Random password: xMc2XkI=ivVM
  UID: 1998400002
  GID: 1998400002
  Password: True
  Kerberos keys available: True
```

# Features: Deployment

- Hide complexity of LDAP+Kerberos+CA+... deployment
- We have few requirements
  - Sane DNS environment, reverse records
    - DNS is crucial to **identify machines**
    - Service **principals,** X509 **Certificates** use DNS names
    - **SSH** identify targets via DNS names
  - Static FQDN hostnames - forms principals
- Configuration with one command
  - ipa-server-install, ipa-client-install
- Supports replicas
  - Essential for redundancy and fault protection
  - ipa-replica-install

# Features: Identity Management

- Users, groups:
  - Automatic and unique **UID**s, **across replicas**
  - Manage users’ **SSH** public keys
  - Role-based access control, self-service
- Hosts, host groups, netgroups:
  - Manage host life-cycle, enrollment
- Services/applications
  - Manage keytab, certificate
- Automatic group membership based on rules
  - Just add a user/host and all matching group/host group membership is added

# Features: DNS

- Optional feature
- DNS data stored in LDAP
- Plugin for BIND9 name server to serve the data
  - bind-dyndb-ldap
- Allows integration of DNS records with rest of the framework

# Features: Policy Management

- **HBAC**
  - Control who can do what and where
  - Enforced by SSSD for authentication requests going through PAM
  - Useful when automember is in use

```
$ ipa hbacrule-show labmachines_login
  Rule name: labmachines_login
  Enabled: TRUE
  User Groups: labusers, labadmins
  Host Groups: labmachines
  Services: sshd, login
```

# Features: Policy Management (2)

- **SUDO:**

```
$ ipa sudorule-show test
  Rule name: test
  Enabled: TRUE
  User Groups: labadmins
  Host Groups: labmachines
  Sudo Allow Commands: /usr/sbin/service
```

- **Automount:**

```
$ ipa automountkey-find prato auto.direct
-----------------------
1 automount key matched
-----------------------
  Key: /nfs/apps
  Mount information: export.example.com:/apps
```

# Features: Policy Management (3)

- **SELinux** user roles
  - Centrally assign SELinux user roles to users
  - Avoid configuring roles per-server using “semanage user” command

```
$ ipa selinuxusermap-show labguests
  Rule name: labguests
  SELinux User: guest_u:s0
  Enabled: TRUE
  User Groups: labusers
  Host Groups: labmachines
```

# Use case: Kerberize a web service

- Enable SSO for a web application with few commands

```
# ipa-client-install -p admin -w PAsSw0rd --unattended
Discovery was successful!
Hostname: web.example.com
Realm: EXAMPLE.COM
DNS Domain: example.com
IPA Server: ipa.example.com
BaseDN: dc=example,dc=com
Synchronizing time with KDC...
Enrolled in IPA realm EXAMPLE.COM
...
DNS server record set to: web.example.com -> 10.0.0.10
...
Client configuration complete.
```

# Use case: Kerberize a web service (2)

```
# kinit admin

# ipa service-add HTTP/web.example.com

# ipa-getkeytab -p HTTP/web.example.com -s ipa.example.com \
   -k /etc/httpd/conf/httpd.keytab

# chown apache:apache /etc/httpd/conf/http.keytab
# chmod 0400 /etc/httpd/conf/http.keytab
```

# Use case: Kerberize a web service (3)

```
# yum install mod_auth_kerb            # Kerberos auth for Apache
# cat /etc/httpd/conf.d/webapp.conf
<Location "/secure">
        AuthType Kerberos
        AuthName "Web app Kerberos authentization"
        KrbMethodNegotiate on
        KrbMethodK5Passwd on
        KrbServiceName HTTP
        KrbAuthRealms EXAMPLE.COM
        Krb5Keytab /etc/httpd/conf/http.keytab
        KrbSaveCredentials off
        Require valid-user
</Location>

# service httpd restart
```

# Introducing SSSD

- SSSD is a service/daemon used to retrieve information from a central identity management system.
- SSSD connects a Linux system to a central identity store like:
  - Active Directory
  - FreeIPA
  - Any other directory server
- Provides identity, authentication and access control

# Introducing SSSD (2)

- **Multiple parallel sources** of identity and authentication – domains
- All information is **cached** locally for offline use
  - Remote data center use case
  - Laptop or branch office system use case
- Advanced features for
  - **FreeIPA** integration
  - **AD** integration - even without FreeIPA

# FreeIPA and Active Directory

- Active Directory is present in most of the businesses
- IdM in Linux and Windows cannot be 2 separate isles
  - Doubles the identity and policy management work
- Need to address some form of cooperation
- 3rd party solutions for AD Integration
  - Enables machine to join AD as Windows machines
  - Linux machines are 2nd class citizens
  - Increases costs for the solution +Windows CLA
  - Does not offer centralization for **Linux native** services
    - SELinux, Automount, ...

# FreeIPA and Active Directory (2)

- FreeIPA v2 - **winsync**
  - User and password synchronization
  - Easier management, but still 2 **separate** identities
  - One-way, name collisions, no SSO from AD
- FreeIPA v3+ - **Cross-realm Kerberos trusts**
  - Users in AD domain can access resources in a FreeIPA domain and vice verse
  - One Identity, no name collisions, SSO with AD credentials

# FreeIPA and Active Directory (3)

- Stage 1: allow AD users to connect to FreeIPA services. For example:
  - PuTTY from Windows machine connecting to FreeIPA-managed Linux machine via SSH - **with SSO!**
  - Mounting Kerberos-protected NFS share
- Stage 2: allow FreeIPA users to interactively log in into AD machines
  - Requires support for *Global Catalog* on FreeIPA server side
  - Work in progress, planned for FreeIPA 3.4 (Q4/2013)

# Cross-Realm Kerberos Trust

- FreeIPA deployment is a fully managed Kerberos realm
- Can be integrated with Windows as RFC4120 compliant Kerberos realm
- Traditional Kerberos trust management applies:
  - Manual mapping of Identities in both Active Directory and Linux (*~/.k5login*)
  - Does not scale with thousands of users and computers
- Better approach - **native cross forest trusts**
  - AD DC thinks considers FreeIPA server as another AD DC
  - MS-specific extensions to standard protocols need to be supported

# Cross-Realm Kerberos Trust (2)

- FreeIPA Samba passdb backend:
  - Expansion of traditional Samba LDAP passdb backend
  - New schema objects and attributes to support trusted domain information
  - Creates the actual Trust using LSA pipe via SMB protocol
  - Exposes the LSA pipe to FreeIPA framework - trust handling

- FreeIPA KDC backend:
  - Verifies and sign MS-PAC coming from a trusted cross forest realm
  - Accepts principals and tickets from a trusted realm
  - Generates MS-PAC information out of LDAP

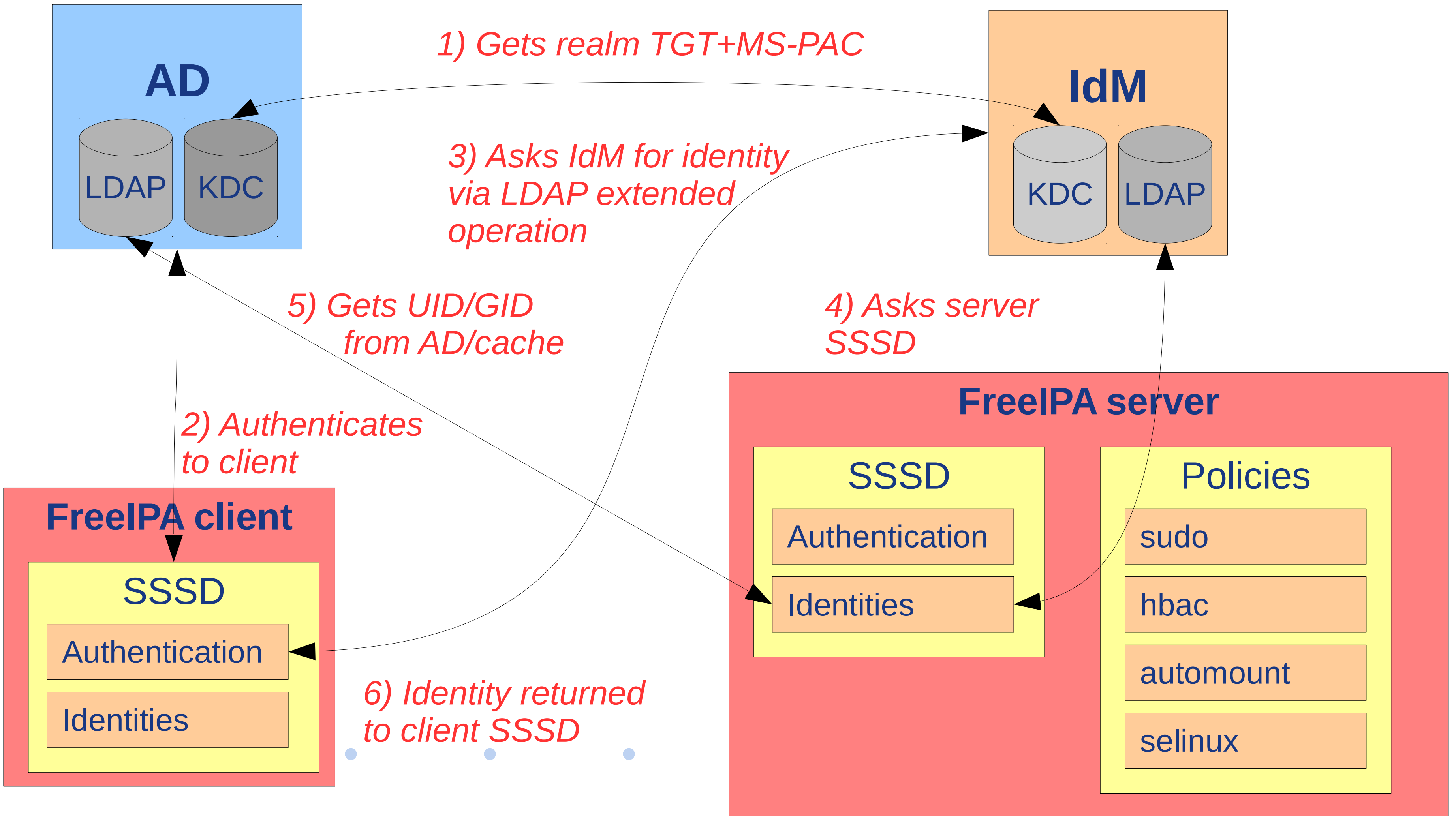
# Is it enough? What is the catch?

- We can manage **Linux** machines with FreeIPA
- We can manage **Windows** machines with AD
- We can establish a **trust** between them - good!
- Works great for green field deployments
- **BUT!**
  - What about users **already using Linux-AD integration**?
    - Identity Management for Unix AD LDAP extension
    - Third party plugins
  - What about users in **legacy machines**?
    - Older Linuxes, UNIXes...
    - They cannot use the modern SSSD with AD support
  - Address before moving forward

# Existing Linux-AD integration

- Main problem is with UID/GID generation
  - FreeIPA 3.0-3.2 generates them from SID
    - Maps Windows style SID (e.g. *S-1-5-21-16904141-148189700-2149043814-1234*) to UNIX-style UID/GID based on user ranges (e.g. UID *9870001234*, GID *9870001234*)
- AD users may already contain defined UID/GID attributes
  - Identity Management for Unix AD LDAP extension
  - UID/GID are already used on Linux machines
  - If changed, file ownership breaks
- Allow reading these attributes!
  - New setting for AD Trust
  - SSSD reads the POSIX attributes from AD and uses them

# Existing Linux-AD integration (2)



1) Gets realm TGT+MS-PAC

2) Authenticates to client

3) Asks IdM for identity via LDAP extended operation

4) Asks server SSSD

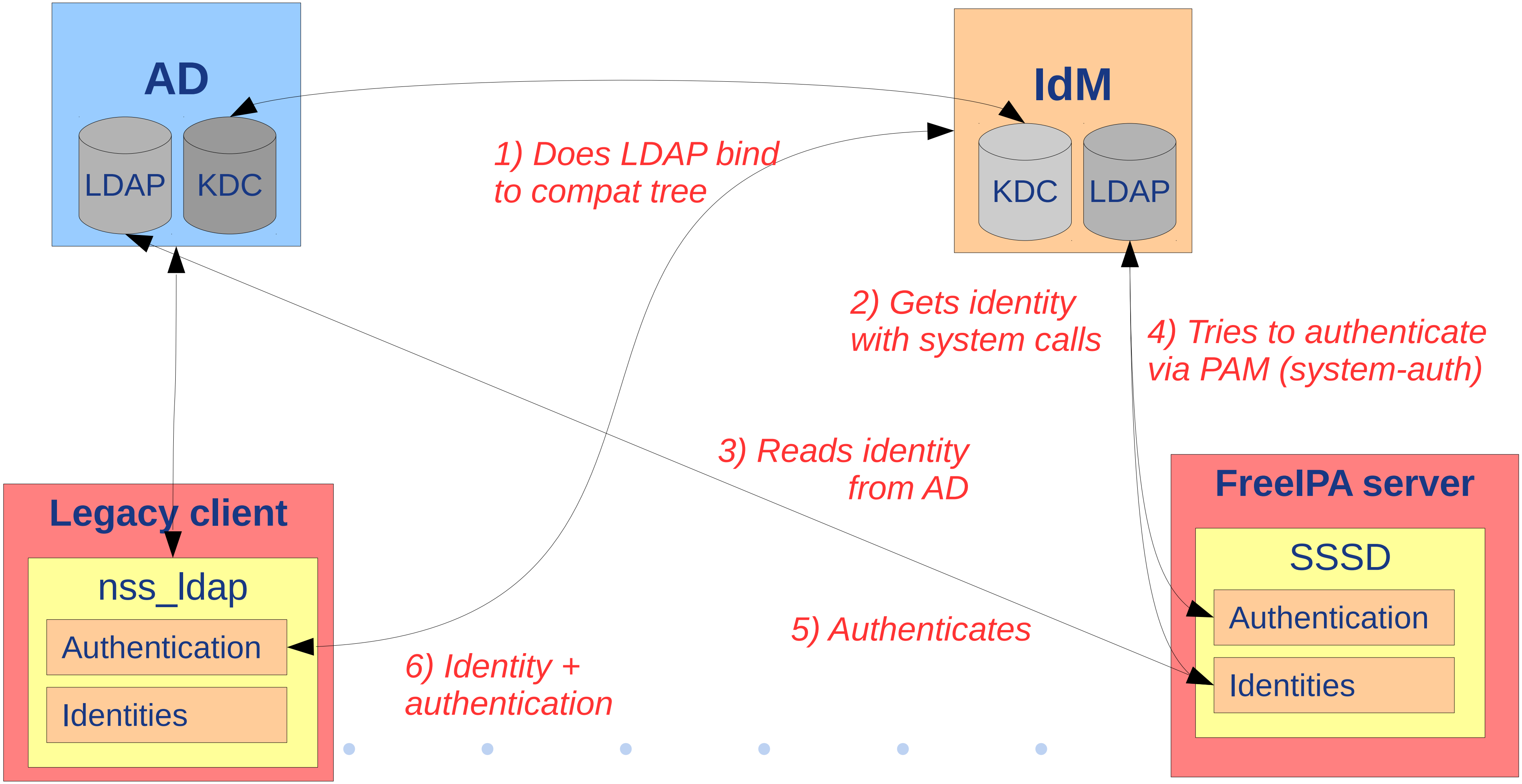
5) Gets UID/GID from AD/cache

6) Identity returned to client SSSD

# Legacy clients using AD Trust

- Administrator may want both AD and Linux users authenticate in older systems
  - SSSD with AD support may not available
- Solved by compatibility LDAP tree in FreeIPA server
  - Exposes a compatibility tree managed by slapi-nis DS plugin
  - Provides both identity and authentication standard via LDAP operations
  - Intercepts LDAP bind operations
    - For FreeIPA user, it just does LDAP bind to FreeIPA LDAP tree
    - For external user:
      - Asks SSSD for user/group (getpwnam_/getgrnam_r), it asks AD
      - Does PAM system-auth command, also via SSSD

# Legacy clients using AD Trust (2)


AD
LDAP
KDC
IdM
KDC
LDAP
1) Does LDAP bind to compat tree
2) Gets identity with system calls
3) Reads identity from AD
4) Tries to authenticate via PAM (system-auth)
5) Authenticates
6) Identity + authentication
Legacy client
nss_ldap
Authentication
Identities
FreeIPA server
SSSD
Authentication
Identities

# Other resources, contact

- Web: www.freeipa.org
- Code: www.fedorahosted.org/freeipa/
- IRC: #freeipa on freenode
- Mailing lists:
  - freeipa-interest@redhat.com
  - freeipa-users@redhat.com
  - freeipa-devel@redhat.com

**QUESTIONS?**