FreeIPA

www.freeipa.org

Identity Management in the FOSS World

Simo Sorce
Principal Software Engineer
Red Hat, Inc.

What is FreeIPA ?

- Acronym: Free Identity, Policy, Audit
- Purpose: Make it simpler to manage a complex problem
- Means: Use standard protocols and components
- Target: System Administrators form 7 to 100 years old :-)

freeIPA

identity | policy | audit

Why should I care ?

- Organizations and companies need to manage their users and resources.
- So far IdM has been the realm of proprietary vendors
  - That means the keys of our organizations are in their hands
- We can't have a fully free environment if the Identity space can't be managed through Free Software
- Security + Freedom

freeIPA

identity | policy | audit

The Identity Management Problem

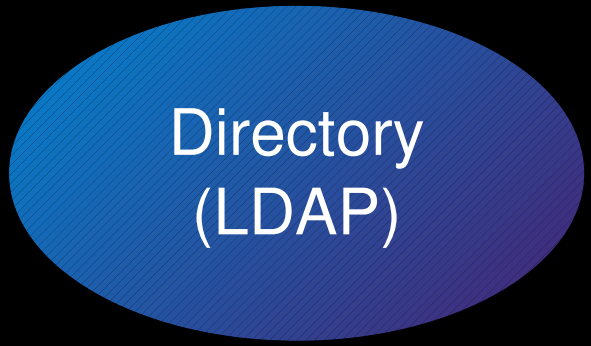
- Needs:
  - Single source for Identities (duplication = confusion)
  - Single-Sign-On / Single-Password
  - Single data store for auditing/reporting (compliance)
  - Single point of Management (comprehensive view)

- Implementation problems:
  - Synchronization and/or Integration
  - Distribution of data/credentials
  - Single points of failure
  - Integrated Management Interfaces

freeIPA

identity | policy | audit

FreeIPA Components



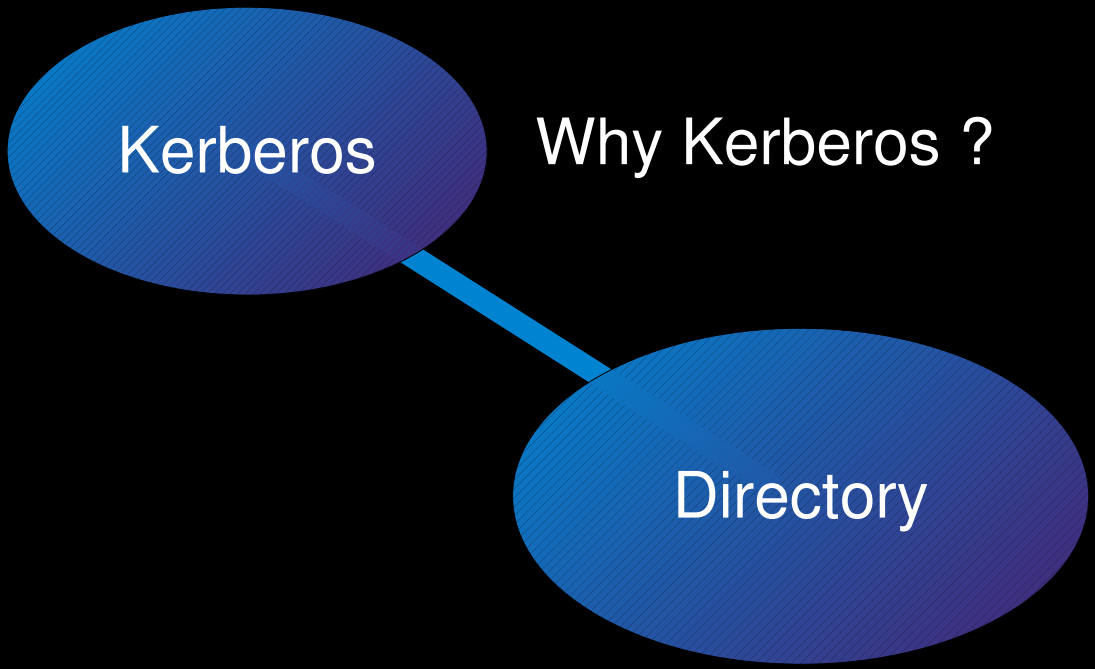
Why a Directory ?

Why a Directory ?

- We need a storage mechanism to:
  - store identity information
  - perform fine grained access control
  - organize Identities and allow group relationships
  - distribute Information across all clients
  - replicate Information on multiple servers

- Yes, but why LDAP ?
  - Standard
  - Extensible
  - Flexible

FreeIPA Components



Why Kerberos ?

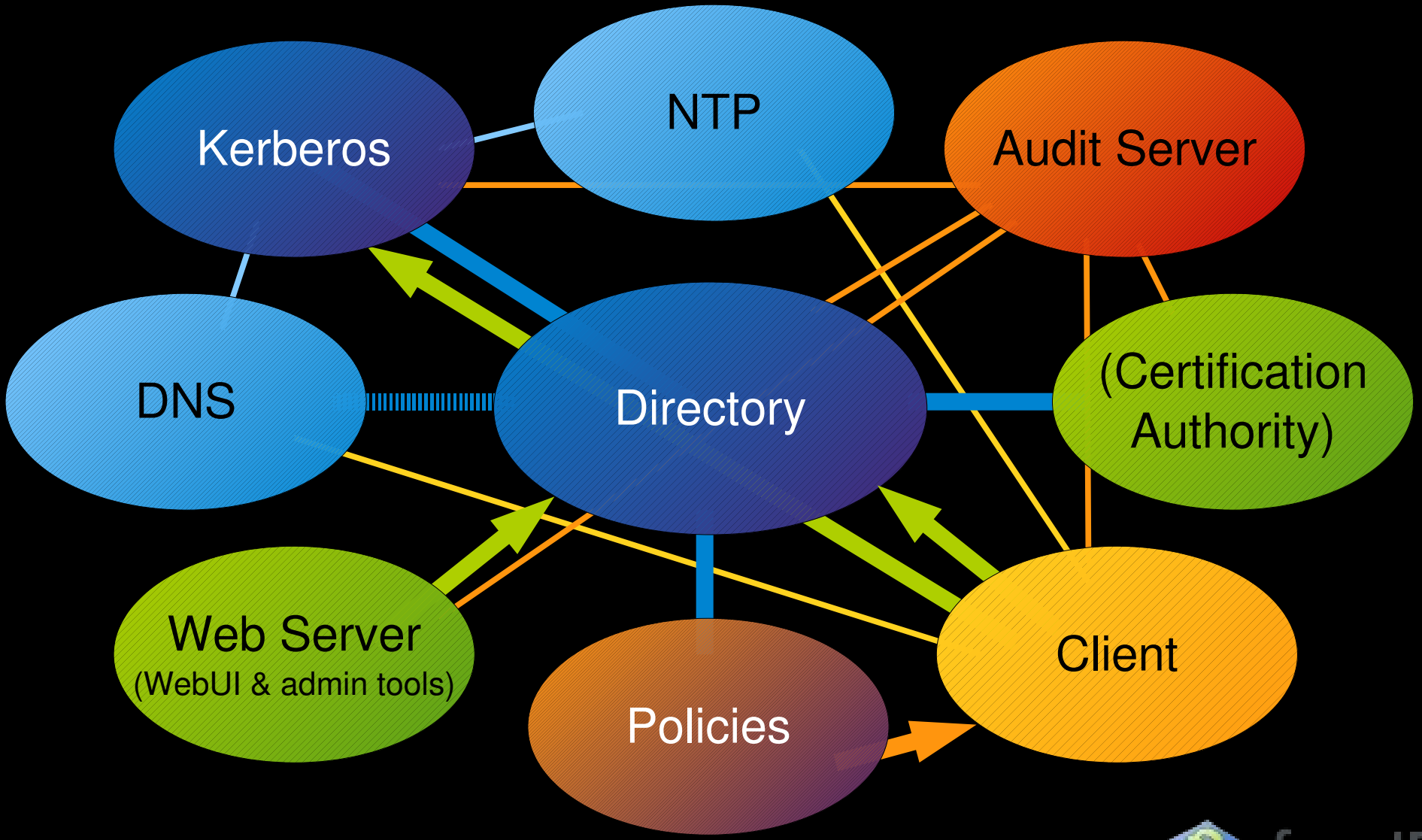
- We need an authentication system that:
  - provides Single Sing On authentication
  - allows administrators and users alike to carry on their identity while they access various services
  - is a tested standard and is a validated secure solution
  - is extensible/extended to use new authentication technologies like Smart Cards and new encryption algorithms as need arises.

- Is kerberos the only way within FreeIPA?
  - Predominant
  - Ldap binds as an alternative for some services

freeIPA

identity | policy | audit

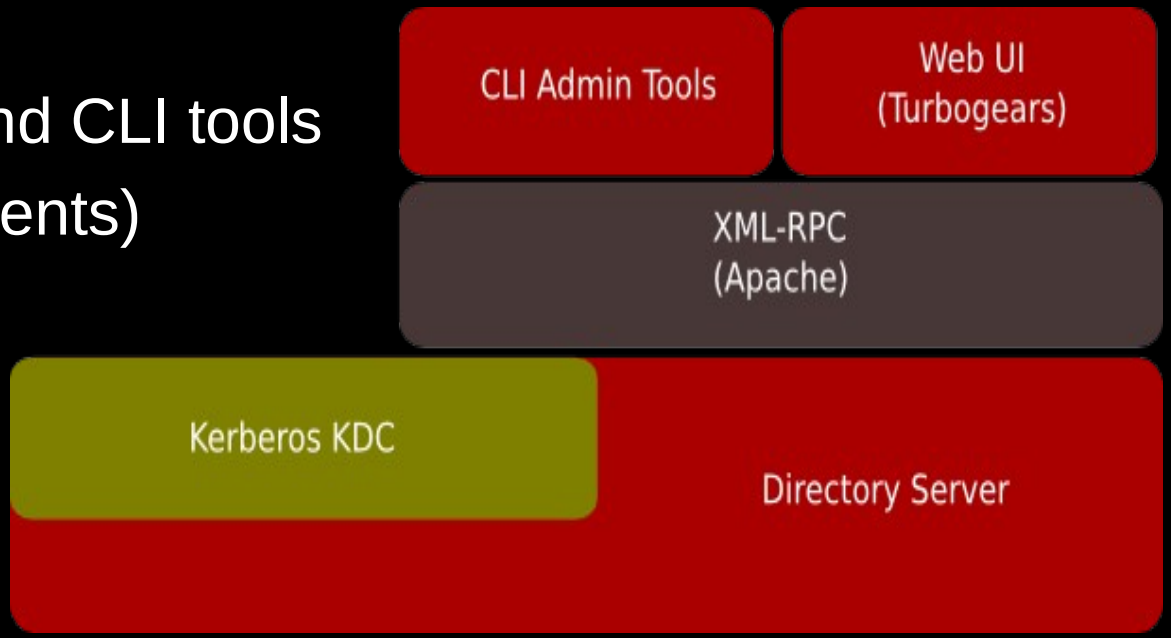
FreeIPA components



freeIPA
identity | policy | audit

FreeIPA (v1) components

- Fedora Directory Server
- MIT Kerberos
- Apache (+ mod_nss, mod_auth_krb, mod_proxy)
- Python, Turbogears
- Custom FDS plugins and CLI tools
- nss_ldap,pam_krb5 (clients)
- Self Signed CA
- NO policies
- NO Audit

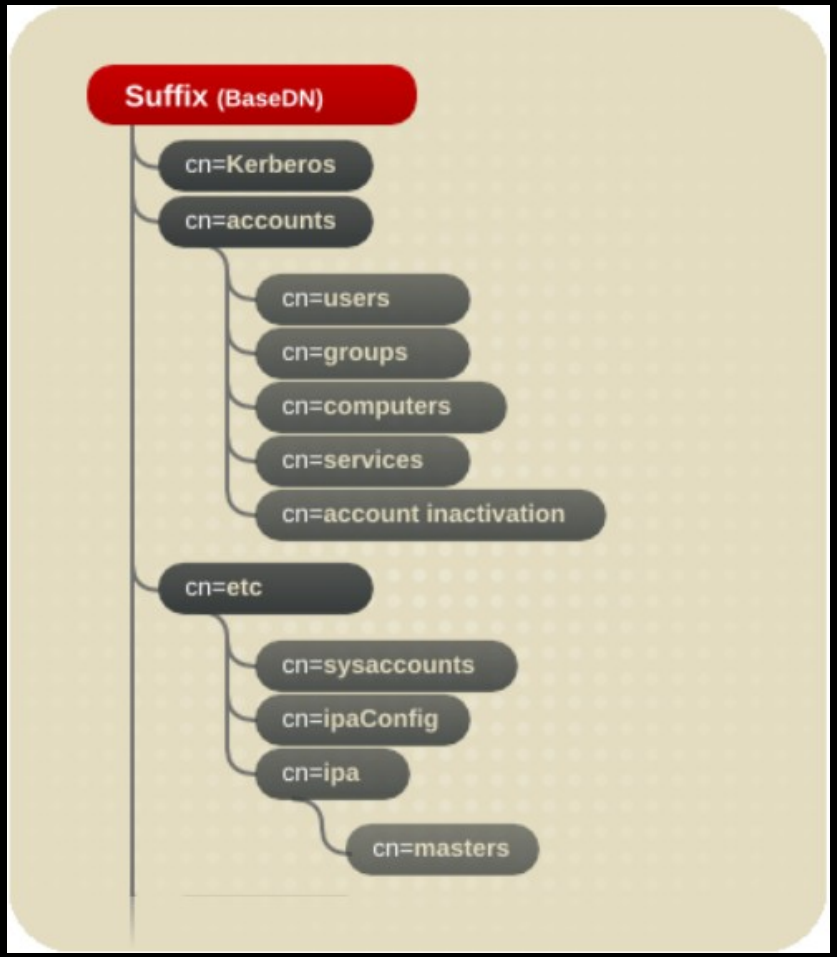


freeIPA
identity | policy | audit

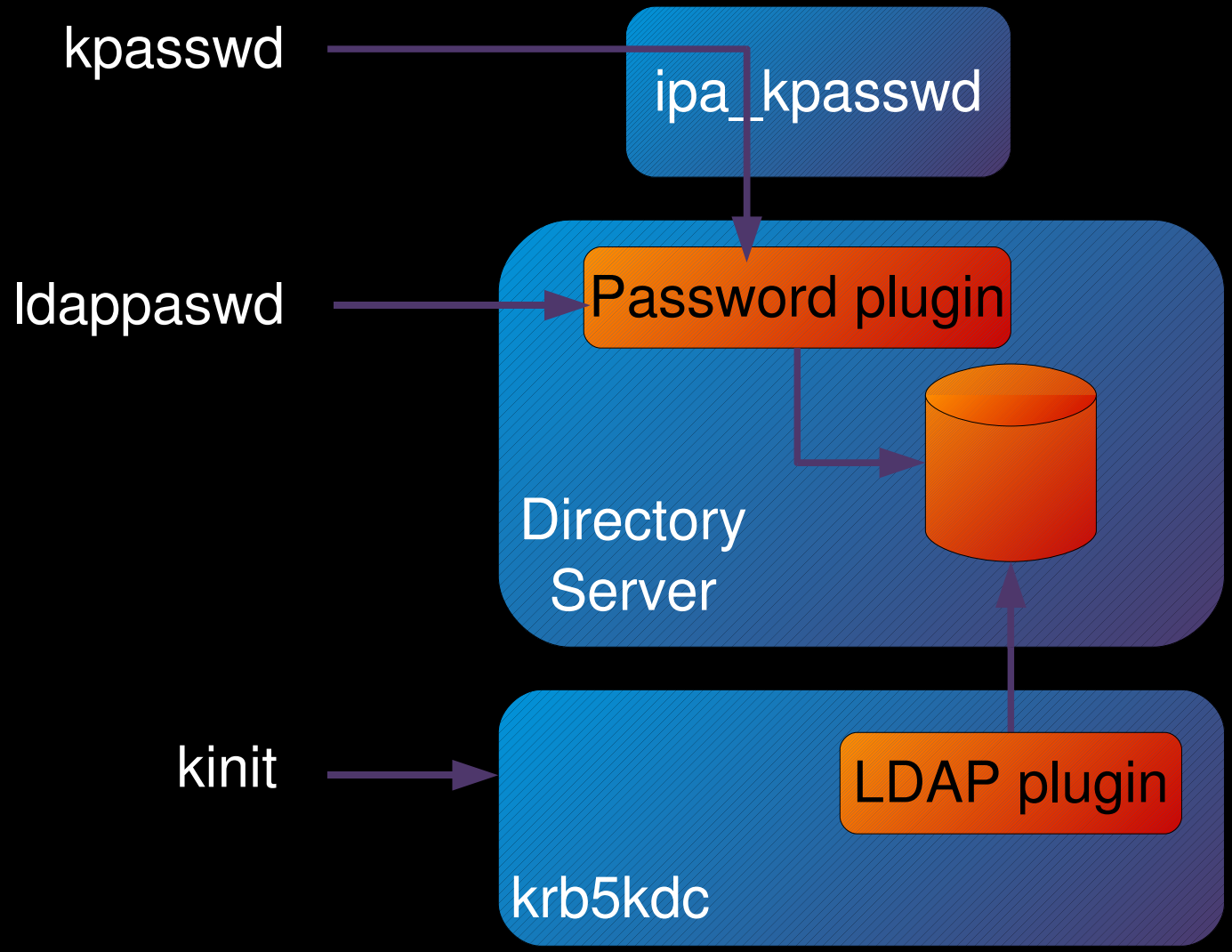
Directory structure

- Accounts, configuration and Kerberos data are kept in separate containers. This allows simpler ACIs and makes it simpler to add more subtrees later without having to reconfigure clients.

  In v1.2 a subtree called cn=compat was added to help legacy clients (Solaris) that do not yet support rfc2307bis

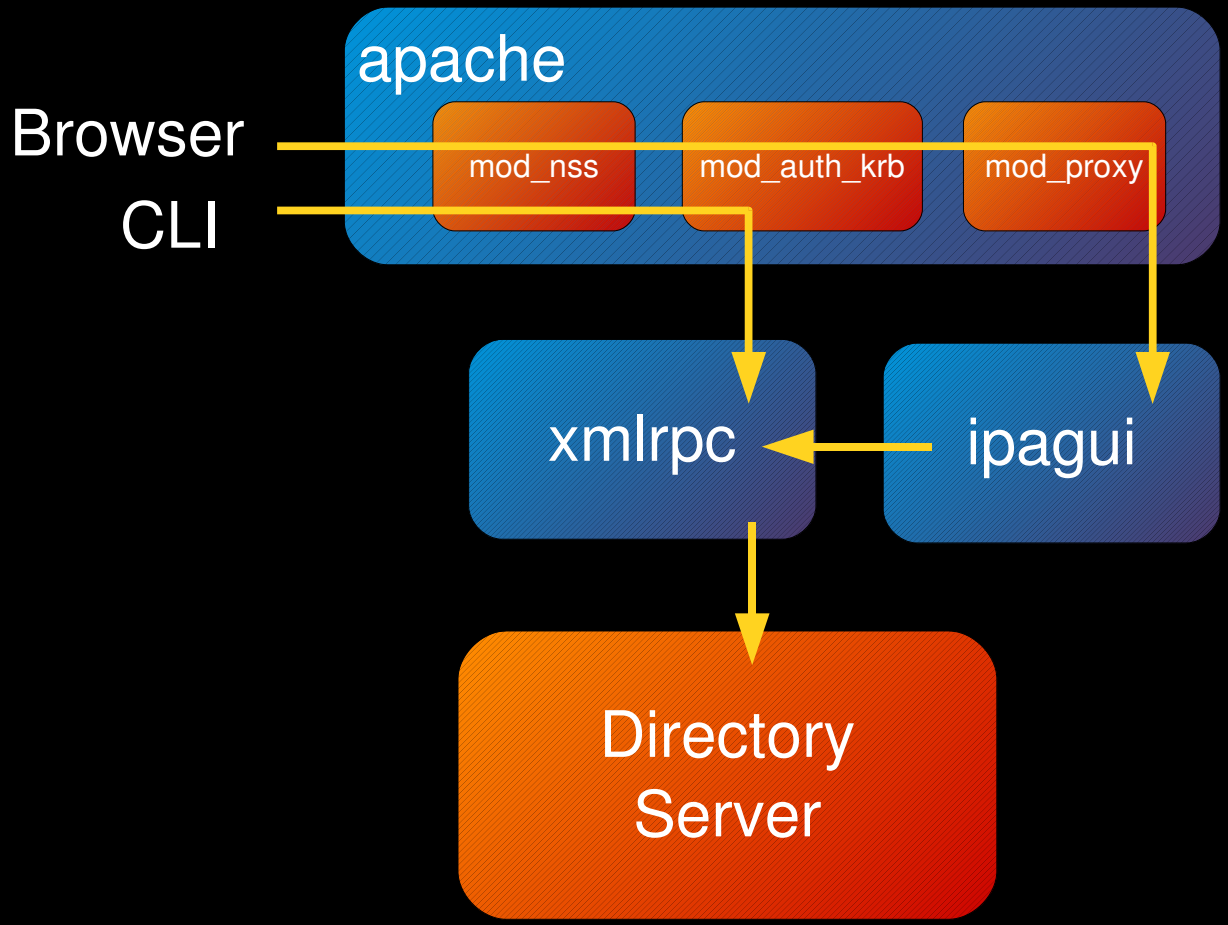


The Kerberos/directory integration

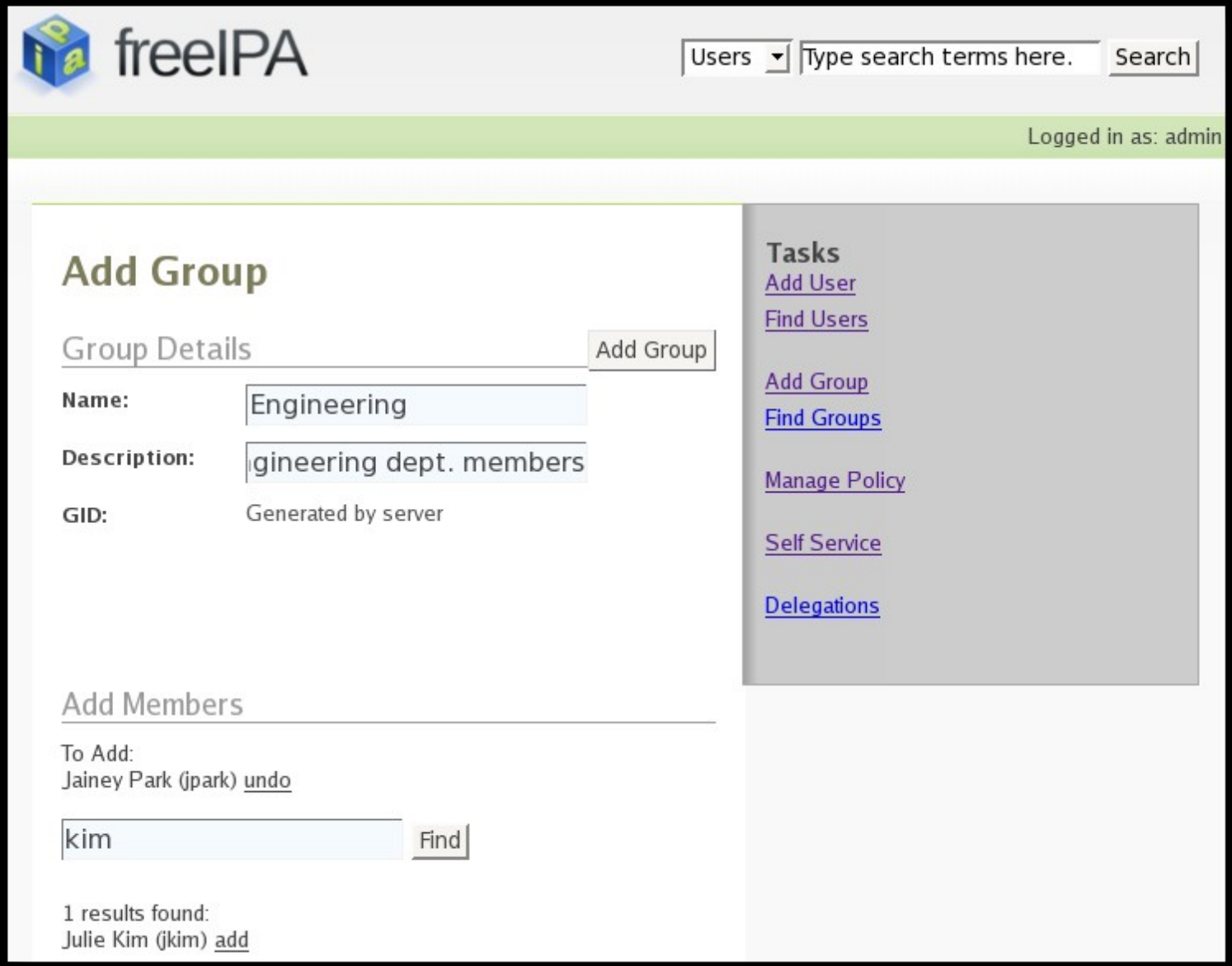


Management Interfaces in v.1

- Everything revolves around the Directory



Web Interface



freeIPA
identity | policy | audit

Command Line Interface

- More than 20 distinct command line tools
- Examples:
  - Ipa-adduser[group/service/delegation]
  - ipa-deluser[group/service/delegation]
  - ipa-finduser[group/service/delegation]
  - ipa-moduser[group/service/delegation]
  - ipa-passwd
  - ipa-pwpolicy
  - ipa-defaultoptions
  - Ipa-change-master-key
  - ...

Not enough low level for you ?

- ldapadd
- ldapmodify
- ldapdelete
- ldappasswd
- … and the joy of manually writing ldif files and horribly breaking your own installation :-)

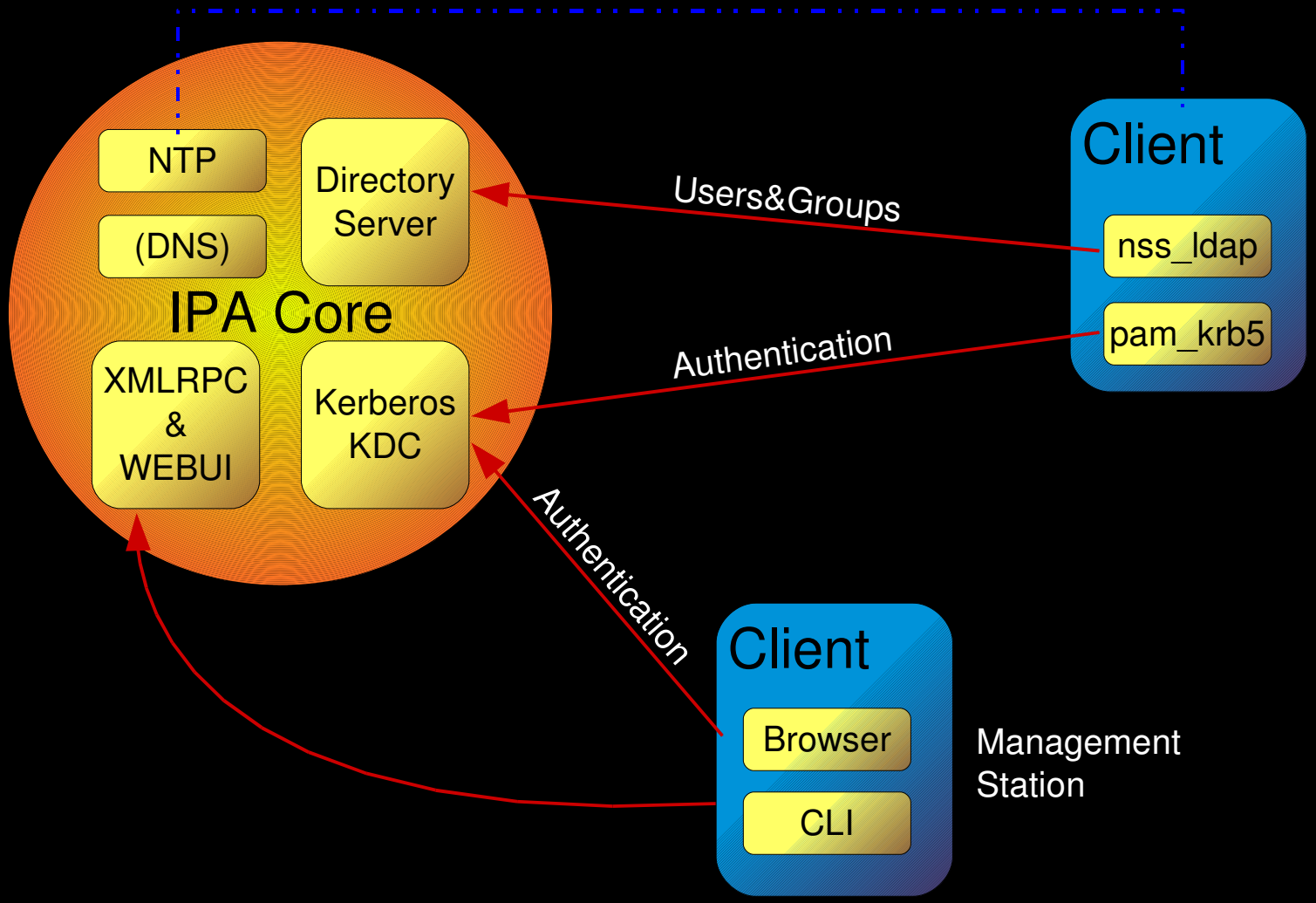
Hey, wait a moment!
Didn't we say we want to make it SIMPLE ?

freeIPA
identity | policy | audit

Making it simpler ...

- Example: initial configuration made very simple
  - Install packages
  - Run ipa-server-install
  - Answer a few questions:
    - DNS Domain and Realm name (defaults suggested)
    - Directory Manager password (required)
    - Admin User Password (required)
    - Done!
- The installation program configures all necessary components: NTP, Directory Server, Kerberos, apache, ipa-kpasswd, ipa-gui, client side bits

Basic IPA v1 network diagram



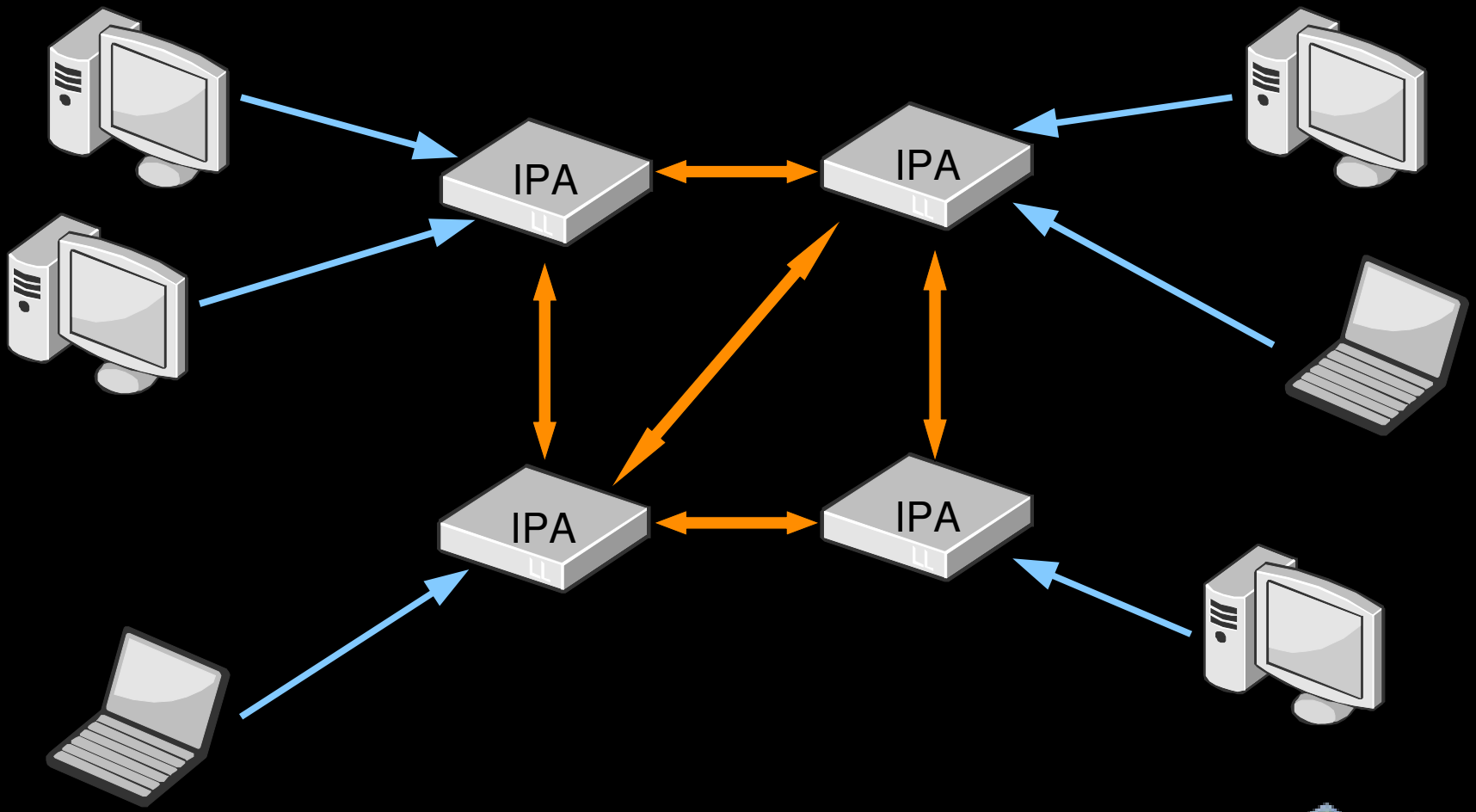
freeIPA
identity | policy | audit

A little more complex: multiple servers.

- Directory server supports Multi Master Replication
  - All information including Kerberos keys is replicated se
    - no need for kpropd
  - Replication is performed at the attribute level
  - DS does automatic conflict resolution
- Setting up replication is done with just 2 commands
  - ipa-replica-prepare on one master
  - Ipa-replica-install on the new server
- Replicas are managed with one command
  - ipa-replica-manage

IPA v1 network topology

- We fully tested up to 4 masters so far, but there is no inherent limitation in the replication protocols



freeIPA
identity | policy | audit

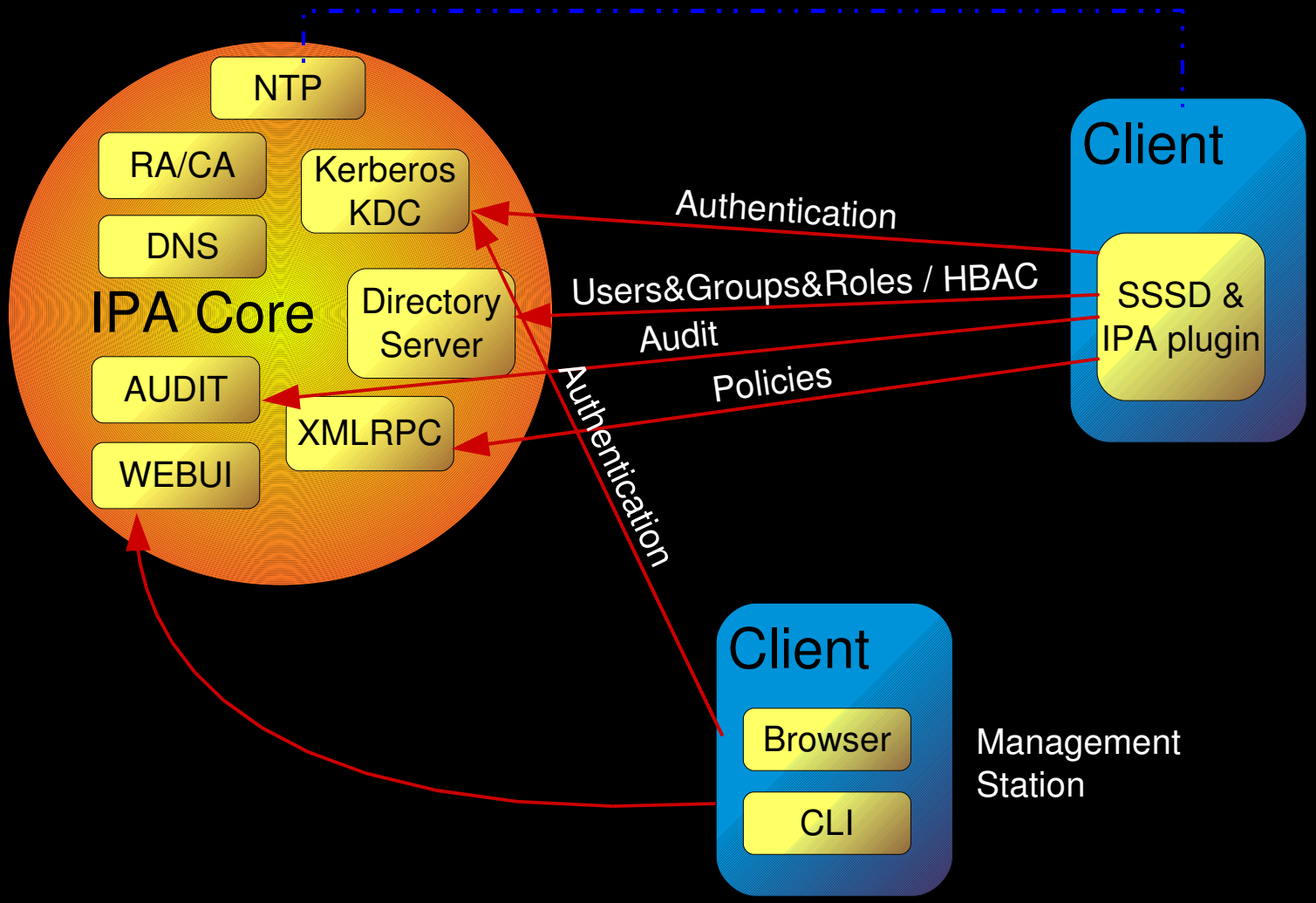
Version 2: new components

- Client agent
  - SSSD: System Security Services Daemon + IPA plugin
  - Manages all connections, caches, support offline ops.
- Policy infrastructure
  - Policy processor + Management interfaces
- Host Based Access Control
  - Centrally managed, rules stored in LDAP
- Roles
  - Centrally defined in LDAP
- Audit Daemon
  - Audit API and client daemon + collecting server daemon

Version2: new components (continued)

- New Web UI
  - Better User Interface
  - Extensible through a plugin system
- DNS Integration
  - LDAP BIND Plugin + GSS-TSIG for Dynamic Updates
- Registration Authority
  - This component will simplify using a Certification Authority and installing certificates on client machines
- Legacy LDAP services
  - Automount maps
  - Translation plugin to present legacy netgroups to clients

Simplified IPA v2 network diagram



freeIPA
identity | policy | audit

Clients and Machine Identities

- In version 1 creation of kerberos keytabs for hosts is a manual operation (except for the ipa server)
  - ipa-addservice/ipa-getkeytab
- In version 2 we will finally have an agent that is run on client machines.
  - The client installation process will automatically retrieve credentials for the client (host/xyz.foo.bar@FOO.BAR)
  - Agent can be trusted by the server + sign&seal of connections to the server is possible using GSSAPI.
  - Increases security of logins and perform validation by default
  - Allows clients to perform operations like requesting certificates form the Registration Authority

freeIPA

identity | policy | audit

Policies

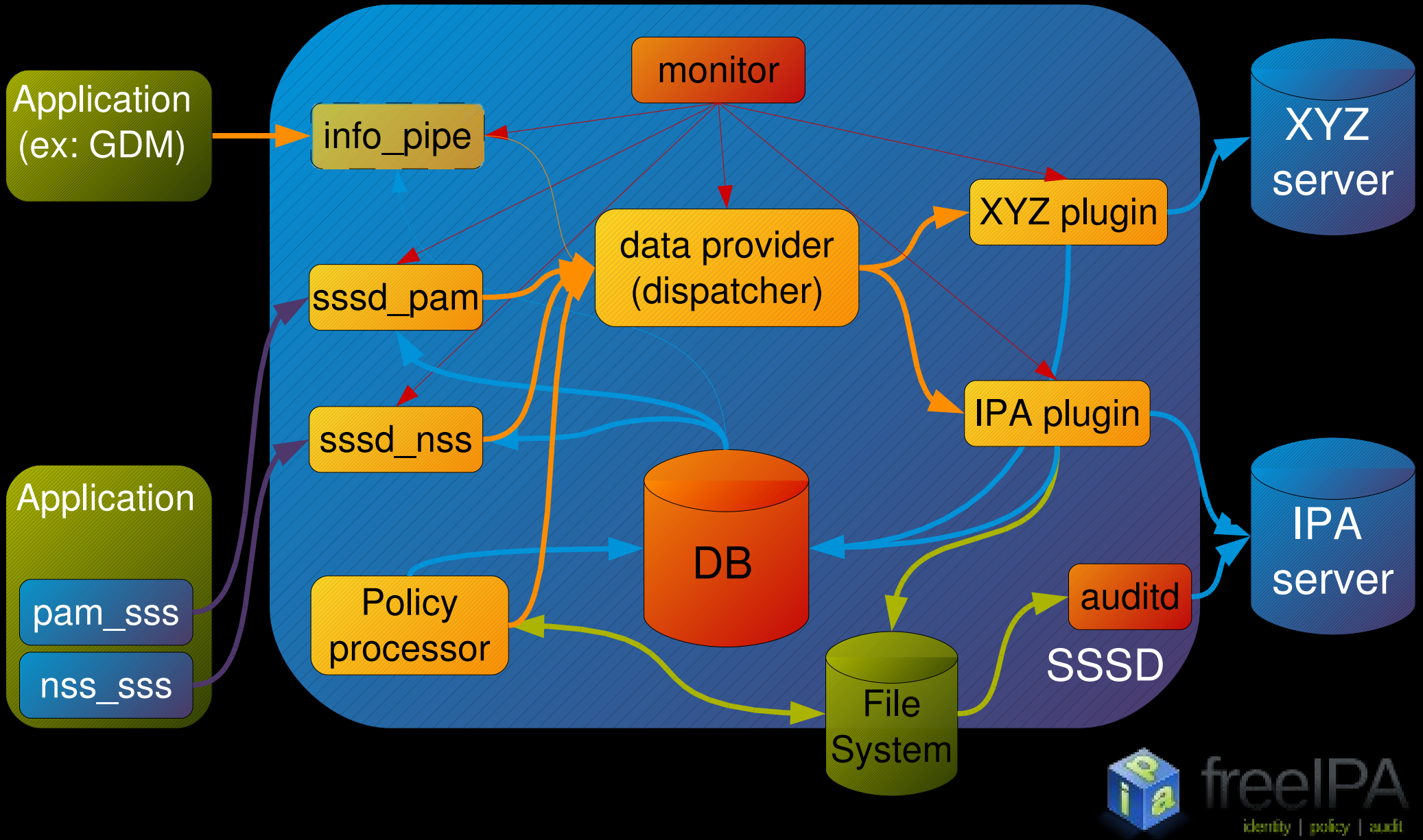
- Policies use XML and RelaxNG based templates
  - Interpreted and merged with local configuration files on the client by the policy processor
  - Also used to build the UI used to manage them
- Policies can be grouped in Policy Groups
- The association between policies and machines is stored in the directory
  - Group of Machines associated to Group of Policies
  - Delegation to junior admins possible through ACLs
  - Roles are also distributed together with policies
    - (SELinux Users, PolicyKit roles, etc...)

Auditing

- Log collection on clients
  - Audit logs from the kernel
  - Syslog files collection / rsyslog
  - API to send audit events
  - Store and forward client based on AMQP

- Log collection on the server
  - AMQP queues
  - Potential for routing audit events to different servers depending on the queue
  - Storage of audit events to allow analysis through common reporting tools

Client diagram



Thank You!

Questions?

http://freeipa.org

freeIPA

identity | policy | audit