An Introduction to the FAS Replacement

Aurelien Bompard
Developer for Fedora @ Red Hat
abompard

Christian Heimes
Developer for Fedora @ Red Hat
tiran

Stephen Coady
Developer for Fedora @ Red Hat
scoady

A bit about us

Aurelien – Fedora contributor since its creation, in the infra team since 2012, tech lead on the AAA project.

Christian – FreeIPA engineer since 2015, liaison between IPA and CPE team.

Stephen – Joined the CPE team at Red Hat about 1 year ago. I have a node.js background but don’t hold that against me.

Why does FAS need to be replaced?

- Based on the TurboGears framework
- Python 2 only
- Supported on RHEL6 only, which goes EOL this fall
- It’s hard to work on it, which causes higher maintenance and complicates improvements

Why FreeIPA?

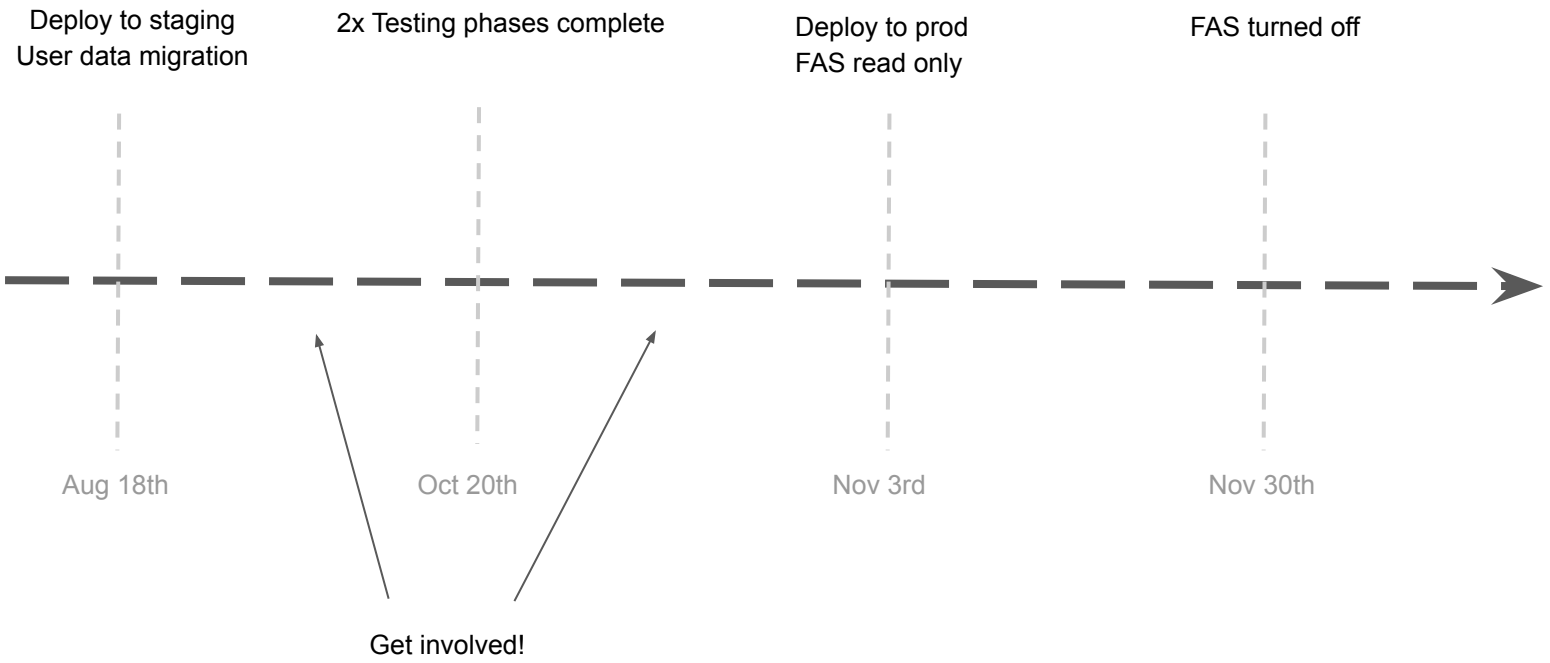
In 2019 CPE team approached IPA team regarding new FAS solution.
FreeIPA vs. $COMMERCIAL_SOLUTION

- Pros
  - Open Source
  - FreeIPA is extensible and flexible
  - FAS would be a great showcase for FreeIPA
  - FreeIPA already part of old FAS and supported by Ipsilon IdP
  - "Eat your own dog food"
- Cons
  - self-hosting
  - long-term maintenance cost

How this will affect...

- FAS users
  - New UI to register, login and edit settings
  - Group membership requests are manual
  - Check your settings after the migration is done (fullnames…)
- Application developers
  - Your applications should migrate to the new API (REST/JSON)
  - API is authenticated via Kerberos, no admin accounts in the conf files
- System admins
  - Easier management of users/groups
  - More powerful feature set
  - Better CLI, API and scripting capabilities

Roadmap


Deploy to staging
User data migration
2x Testing phases complete
Deploy to prod
FAS read only
FAS turned off
Aug 18th
Oct 20th
Nov 3rd
Nov 30th
Get involved!

Design Goals

- Self service
  - Users can register, edit their settings, change their password, enroll OTP tokens…
- Some operations are on-demand
  - Group creation & editing
  - Group join requests
- Much more power for admins thanks to FreeIPA
  - Web UI and CLI
  - Red Hat supported product
- Less dev maintenance work on the IdM side

Technical Challenges

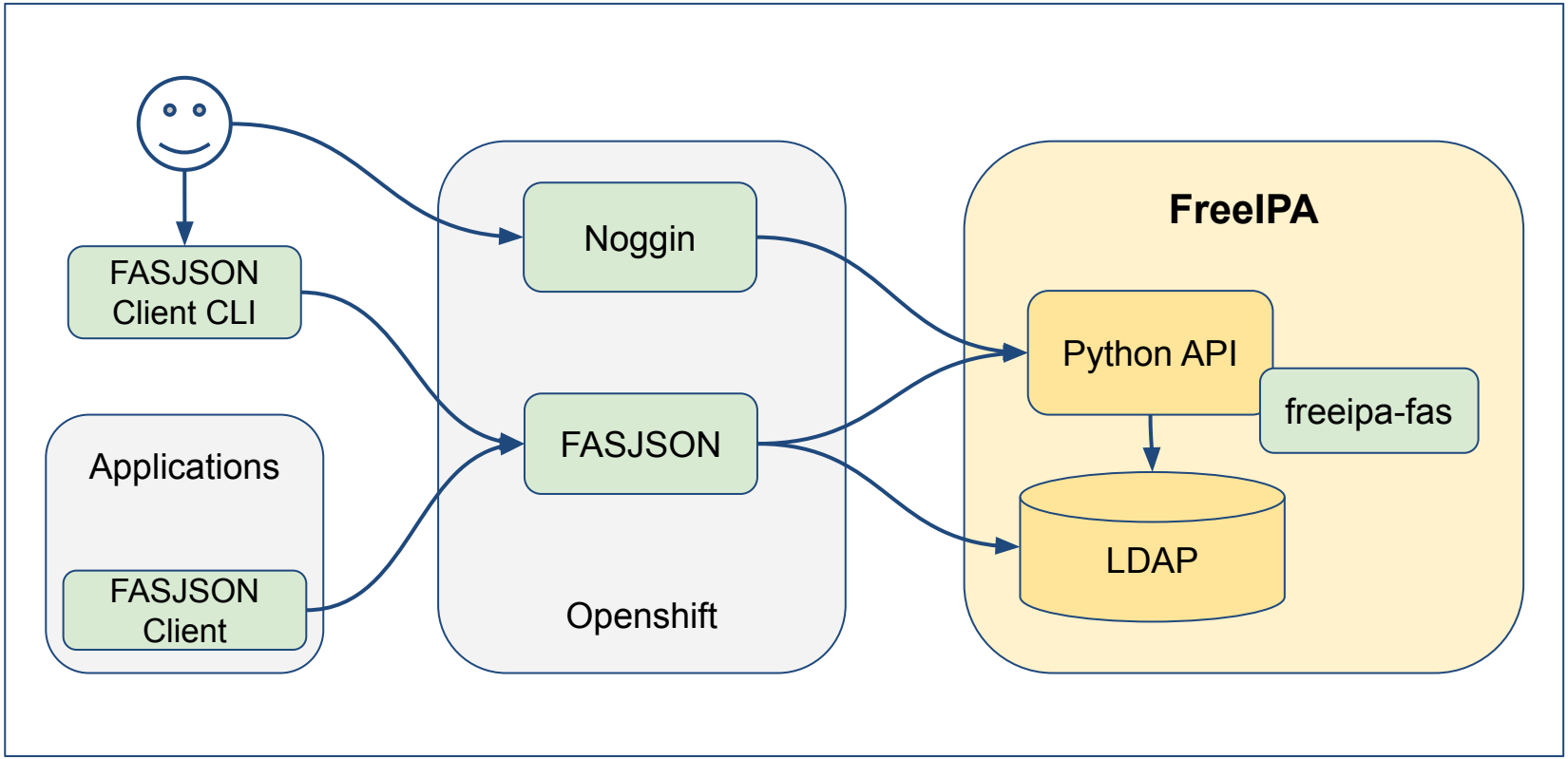
- The data migration from FAS to FreeIPA
- CentOS, OpenSUSE
- Duplicate accounts
- Incompatible properties between FAS and FreeIPA
- Applications currently using FAS

Architecture

Applications

- **FreeIPA**: the data store
- **FreeIPA-FAS**: a FreeIPA plugin for Fedora-specific data
- **Ipsilon**: the authentication portal (OIDC)
- **Noggin** (and noggin-messages): the self-service user portal
- **FASJSON** (and fasjson-client): the REST/JSON API
- **FAS2IPA**: the migration script

Architecture



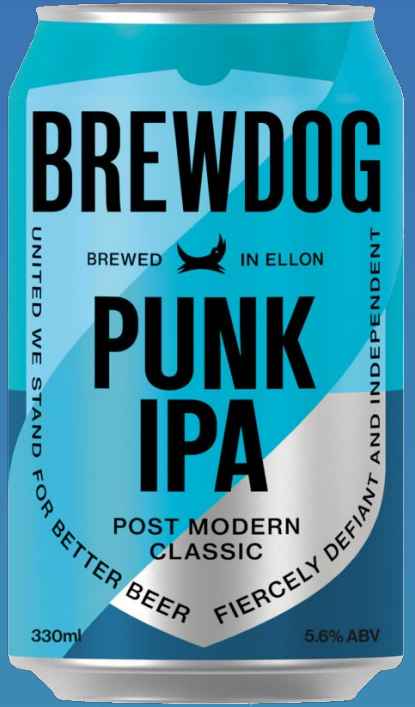
FASJSON

- REST API for applications to get data from FreeIPA
- Authentication via Kerberos (keytabs for apps)
- JSON responses
- Endpoints:
  - user, users
  - group, groups
  - search users
  - certificate signing
- Pagination
- OpenAPI spec & Swagger UI

FASJSON Client

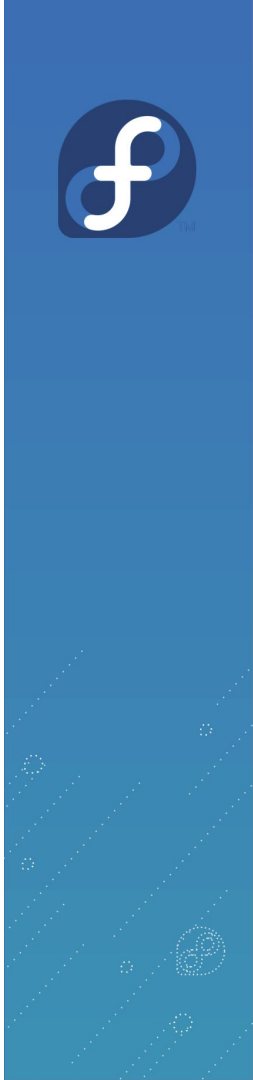
- An easier way to query FASJSON from Python
- Will check the kerberos authentication
- Call methods and get dictionaries or lists
- Pagination support
- Convenience method to get all users in one call
- CLI to generate CSRs and get them signed

What is FreeIPA?



What is FreeIPA?

- Open Source Identity Management solution
- LDAP backend, embeds PKI, Kerberos, OTP server, ...
- Web UI
- CLI tool
- JSON RPC API
- extensible via plugins (Python, JavaScript, LDAP schema, LDIF)
- Documentation:
  - https://www.freeipa.org/
  - Demo: https://ipa.demo1.freeipa.org/
  - dev blogs
  - online training course




sssd


389
directory server
port389.org

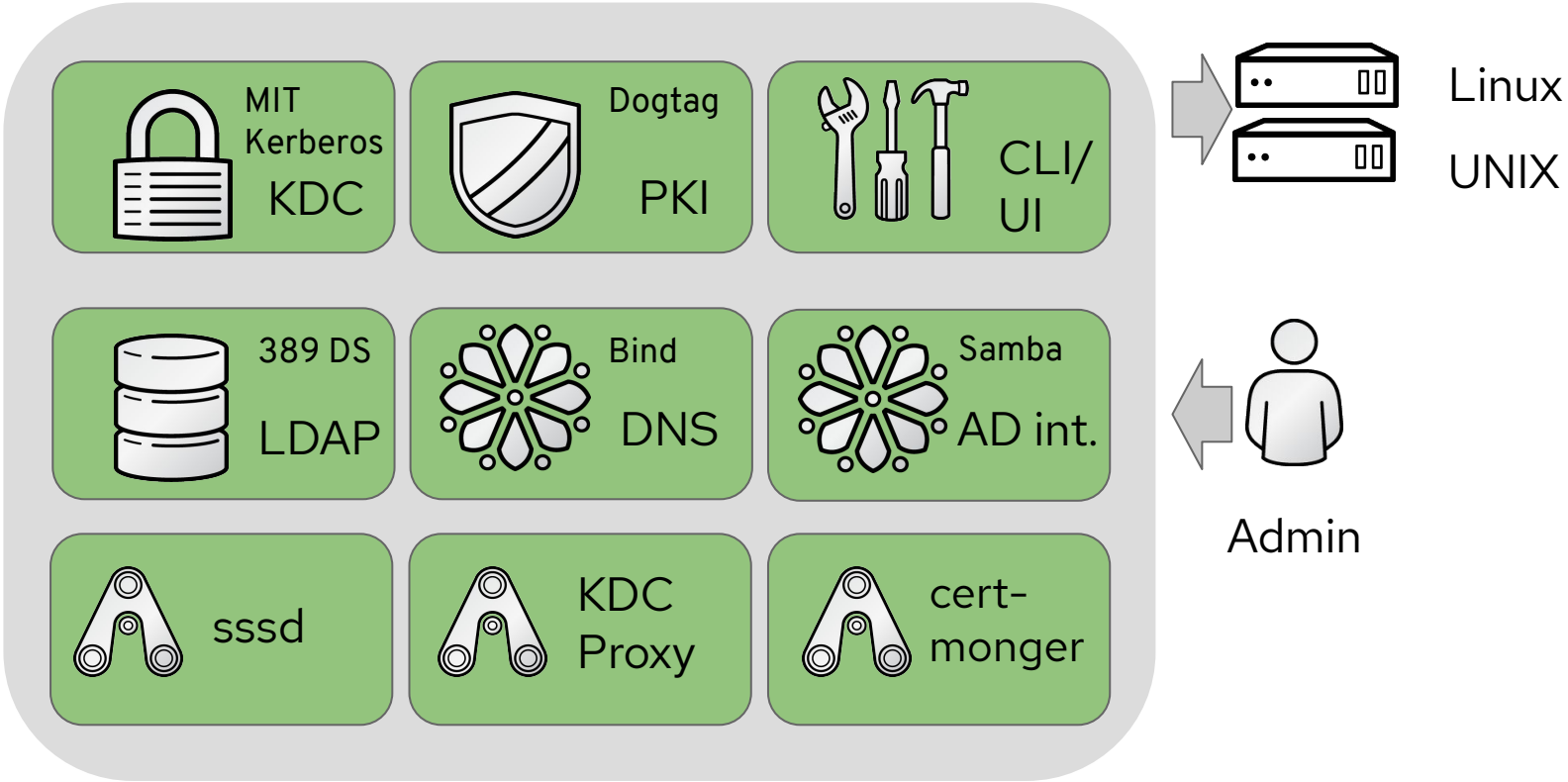

FreeIPA
Open Source Identity Management Solution


APACHE
HTTP SERVER PROJECT

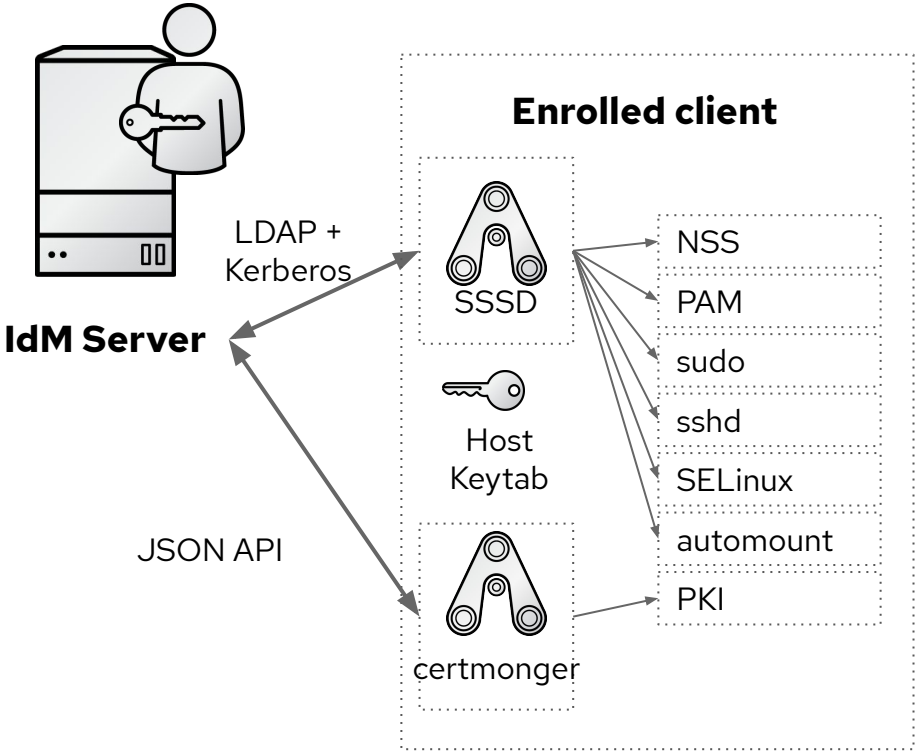

Dogtag
Certificate System


SAMBA

FreeIPA core components



Client OS integration

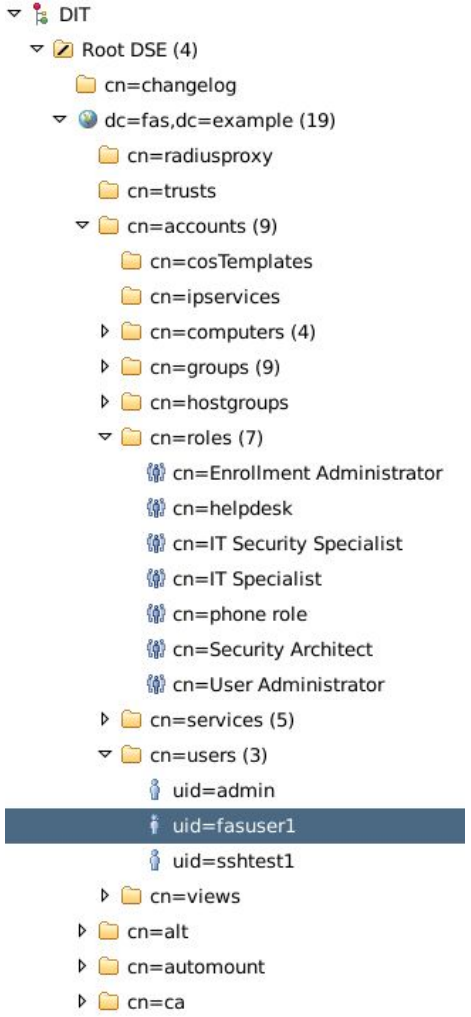


- identity
- authentication
- authorization (HBAC)
- 2FA authentication
- SSH public keys

zero-config Kerberos over HTTPS (DNS URI record)

What is LDAP?

- Not a “regular” (SQL) database
- hierarchical “address book” database
- standardized schema and protocol (RFCs)
- optimized for reading and replication
- extensible
- rich server-side access control (ACI)

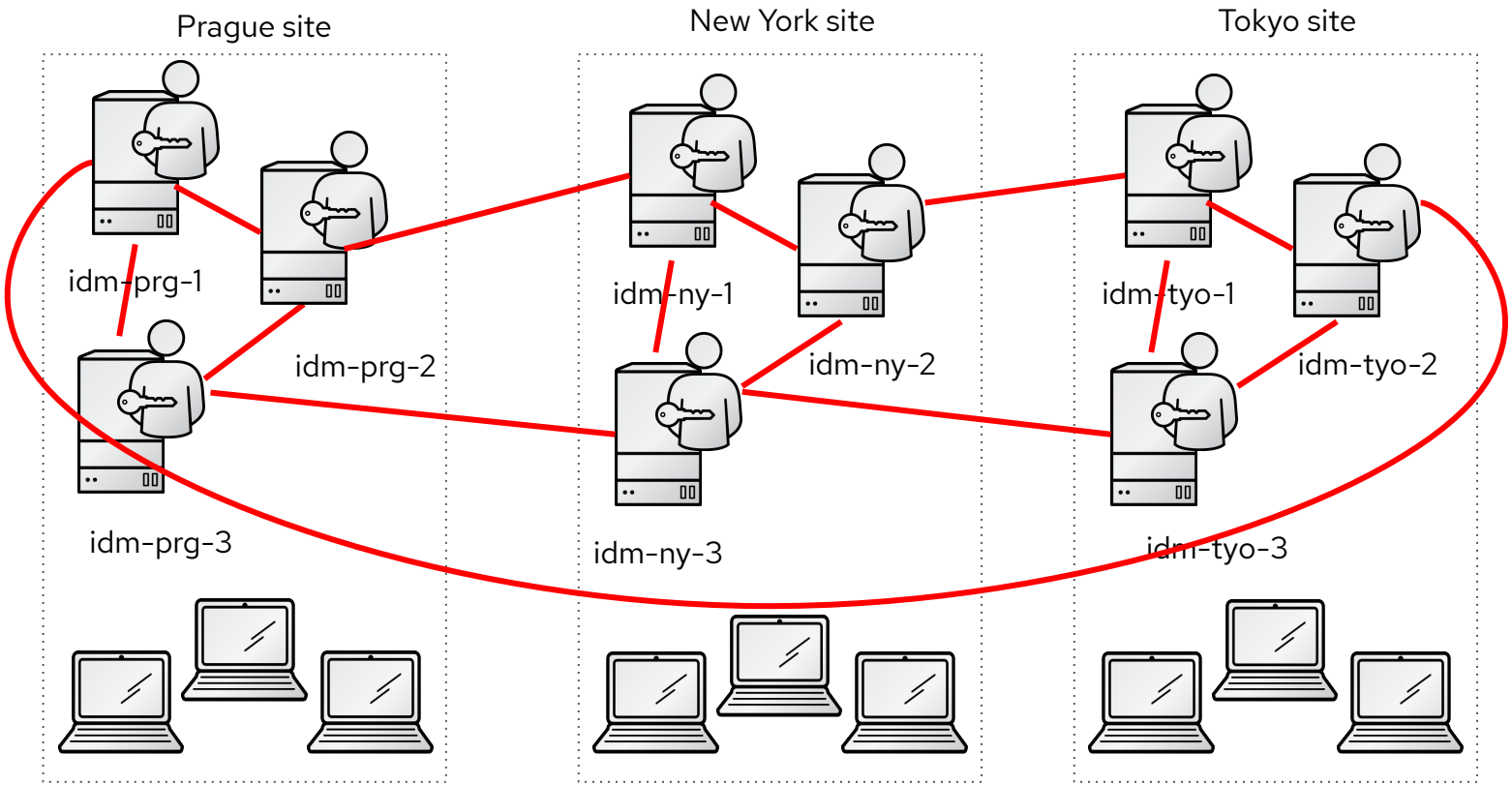


LDAP schema

- entries have DN, object classes, and attributes
- DN (unique identifier / path)

  `uid=fasuser1,cn=users,cn=accounts,dc=fas,dc=example`
- objectClasses (mandatory and optional attributes)
- user attributes
- operational / auto-generated attributes
- attribute types
  - single/multi-valued
  - text, int, date, binary, bool, DN, ...
  - DN (member “foreign key” reference, memberOf back reference)

Multi-primary replication topology (master/master)



Role Based Access Control (RBAC)

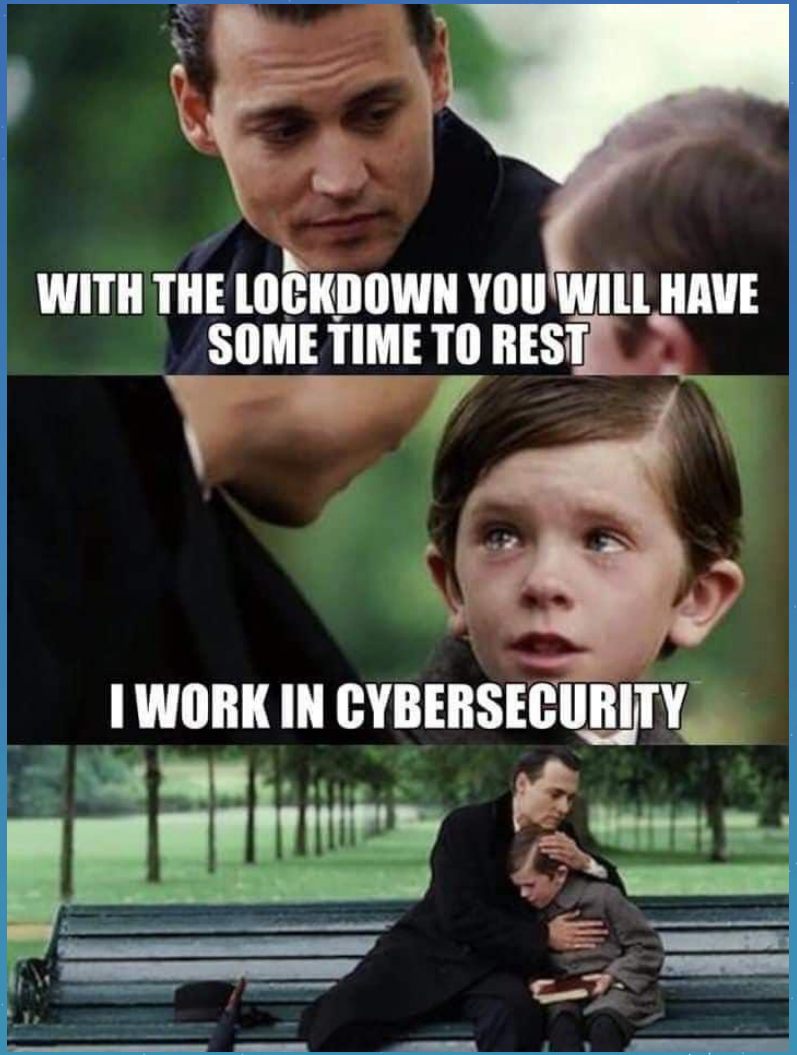
- LDAP server performs access control
  - services impersonate user via credential delegation (Kerberos S4U2Proxy)
- RBAC
  - Permissions: "`Add FAS Agreement`", "`Delete FAS Agreement`"
  - Privilege: "`FAS Agreement Administrators`"
  - Role: "`FAS Agreement Administrator`"
  - User, groups, services, hosts, host groups
- self-service
  - user can modify own IRC nick name
- delegation
  - membership managers of a group can add/remove members

FreeIPA-FAS plugin

- Extended the user object with IRCNick, locale, GPG keys etc.
- Additional group attributes such as url, IRC channel, mailing list
- Group: User agreements
- Access Control Information (ACIs)
  - Users can modify their own attributes
  - Users can self-manage group membership
  - Read access to FAS user and group information
- New permissions, privileges, roles, indexes, unique values
- Web UI extension

Code: https://github.com/fedora-infra/freeipa-fas/

Demonstration



Contributing

- Repos are on Github, with a similar structure
- Unit tests and linting with tox
- Vagrant for local development
- Developer documentation in Noggin’s docs
- Github Project (Kanban) to track progress

When my project has 100% code coverage...

Thanks for listening!

https://github.com/fedora-infra/noggin

https://github.com/fedora-infra/fasjson

https://github.com/freeipa/freeipa

Freenode @ #fedora-aaa