MIT Kerberos & Red Hat

Past, Present and Future

Dmitri Pal

Sr. Engineering Manager, Red Hat Inc.

October 2011

Agenda

- Setting up context
- This year accomplishments
- Plans for next year and beyond

Context

- Red Hat has been sponsoring FreeIPA community for several years
- This year we have seen major upstream releases of FreeIPA: 2.0 and 2.1
  - FreeIPA is a MIT Kerberos based domain controller for Linux/UNIX environments
  - FreeIPA provides centralized authentication, identity and policy management
- Kerberos related enhancements are in large driven by FreeIPA project goals

This year accomplishments

- Cross Realm Kerberos Trusts (ongoing effort)
- NSS Crypto
- Multiple identities per user
- Automatic selection of the identity based on the target
- Authhub project

Cross Realm Kerberos Trusts

- Last year on the conference we talked about FreeIPA and Cross Realm Kerberos Trusts.
- Since then:
  - PAD specification created and submitted
  - KDB re-factored
  - Built DAL extensions that allow:
    - generating MS-PAC authorization data
    - attaching authorization data to tickets.
  - FreeIPA is switching back to use kpasswd instead of the homegrown solution (ipa_kpasswd)

NSS Crypto

- NSS is a FIPS certified crypto module
- Main crypto was updated to work with NSS a year ago
- PKINIT was added this year
- Now all crypto can be switched to use NSS at the build time
- Will be available in MIT Kerberos 1.10
- Plan is to make it available in Fedora 17
- Will be available in a RHEL release next year

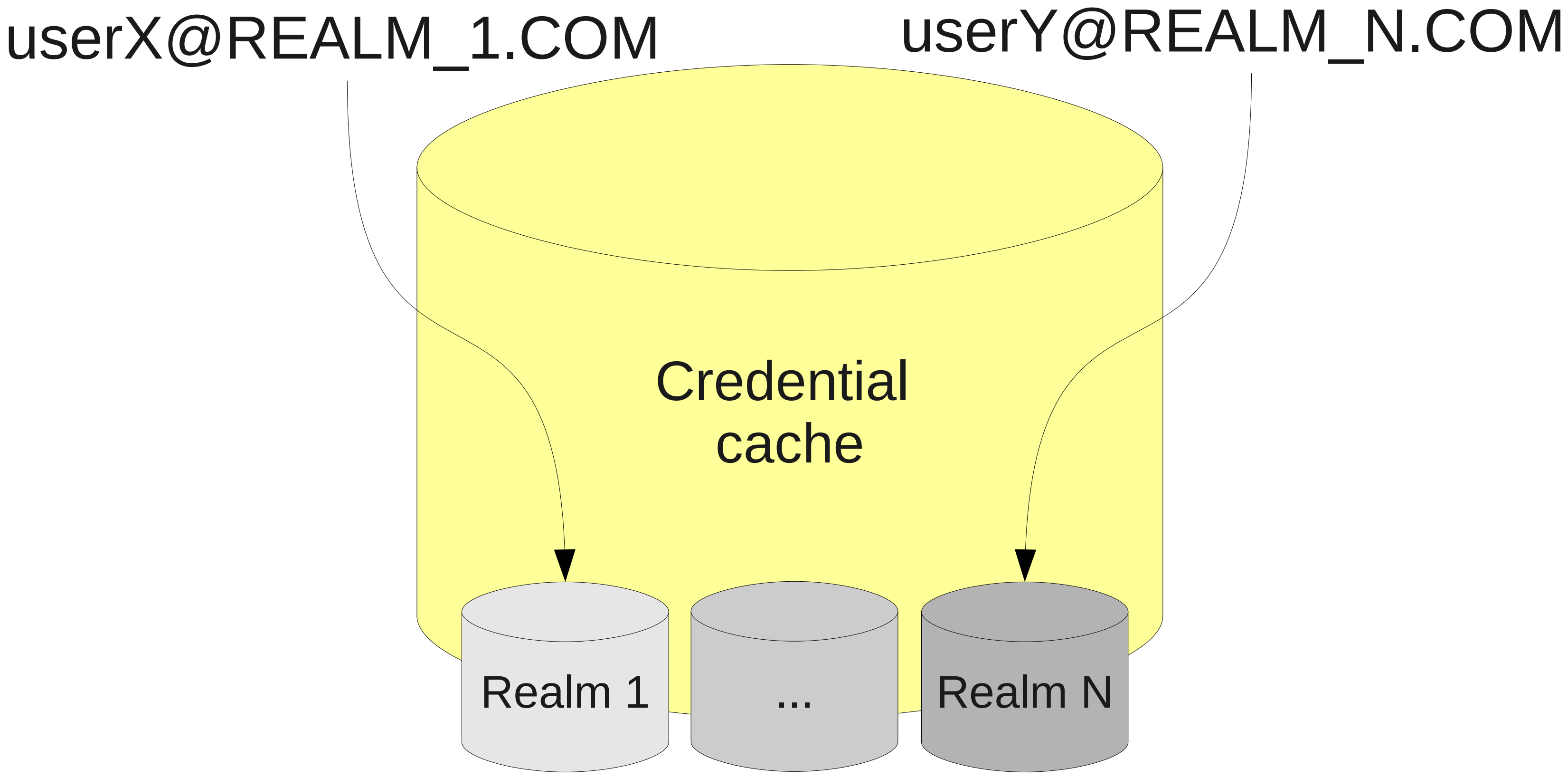
Multiple Identities per User

- Use case:
  - More than one Kerberos environment needs to be accessed at a time from a machine
  - Examples:
    - Corporate and Community (Red Hat and Fedora)
    - Community and Home (Hope office and Fedora)
  - Not frequent yet but was clearly indicated as barrier to Kerberos adoption in community infrastructures like Fedorahosted

Multiple Identities per User (continued)

- This feature allows user to keep credentials for principals in multiple realms working at the same time, without needed to constantly kdestroy/kinit to switch from one realm to another.
- The implementation was done by the MIT team

Multiple Identities per User - Illustration

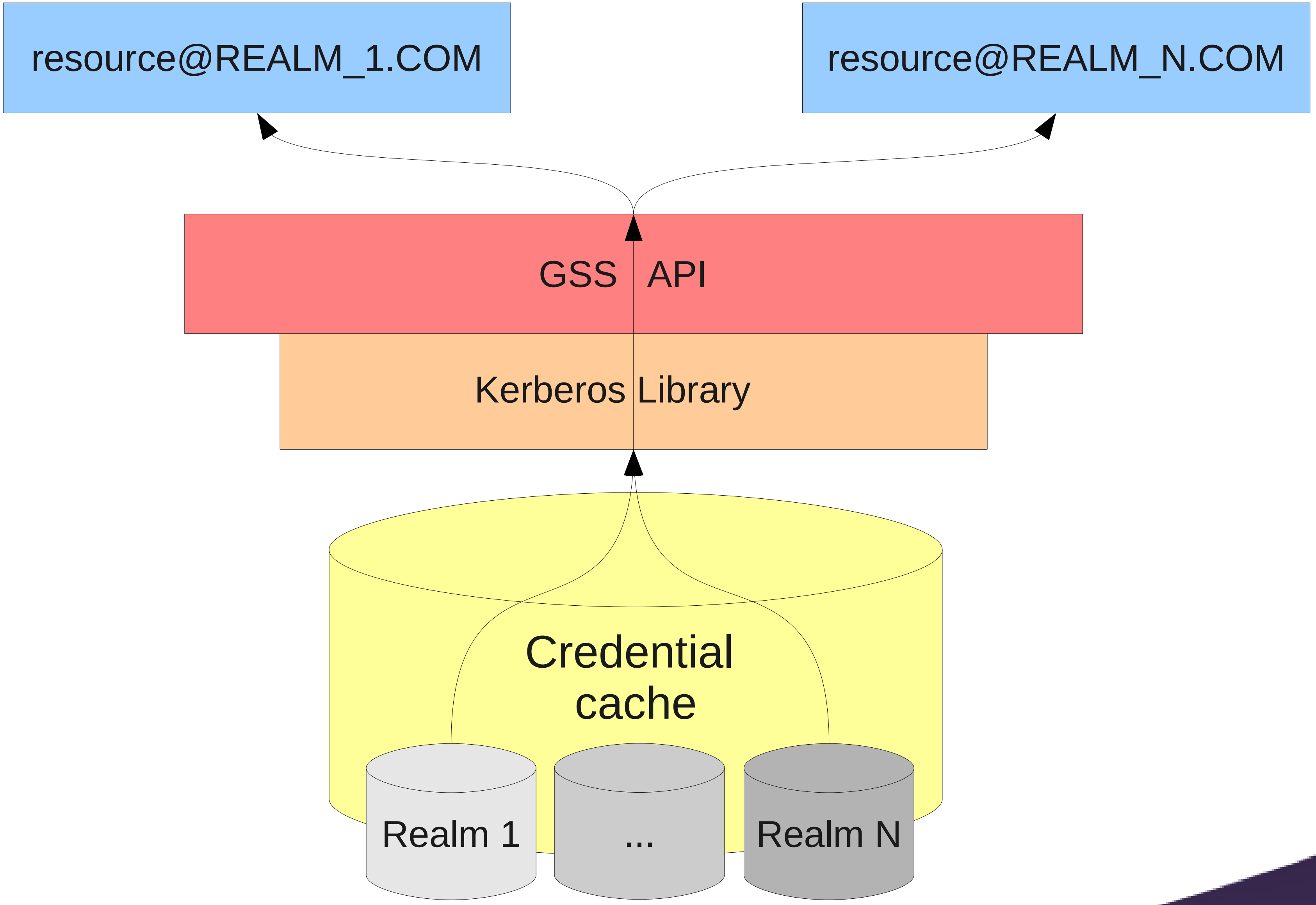


Implemented using directory structure

Automatic Identity Selection

- Use case
  - Once you can have multiple credential caches you should be able to access resources in the different Kerberos realms from one host using the right user identity.
  - The right credential cache should be selected based on the service the user is accessing
- The implementation was done by the MIT team

Automatic Identity Selection



AuthHub

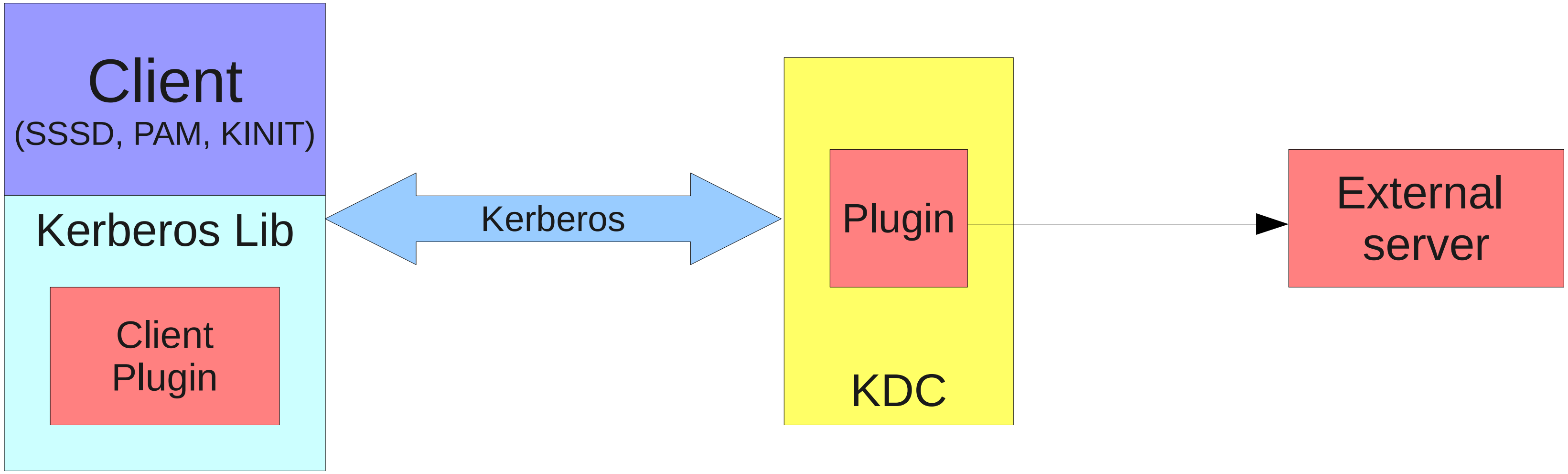
- https://fedorahosted.org/AuthHub/
- Goal:
  - Make Kerberos KDC pluggable for external authentication methods
- Based on the OTP FAST spec from Gareth Richards
  - https://datatracker.ietf.org/doc/draft-ietf-krb-wg-otp-preauth/
  - In active review

AuthHub – project phases

- Phase 1 – proof of concept
  - Demonstrated in May 2011 that it is possible to get a TGT as a result of the 2FA against external servers
  - Reached out to vendors...
    - No interest
    - No involvement
  - Project slowed down as we had to regroup.
- Phase 2 – was supposed to be vendor focused solutions
- Phase 3 – was supposed to be vendor neutral solution

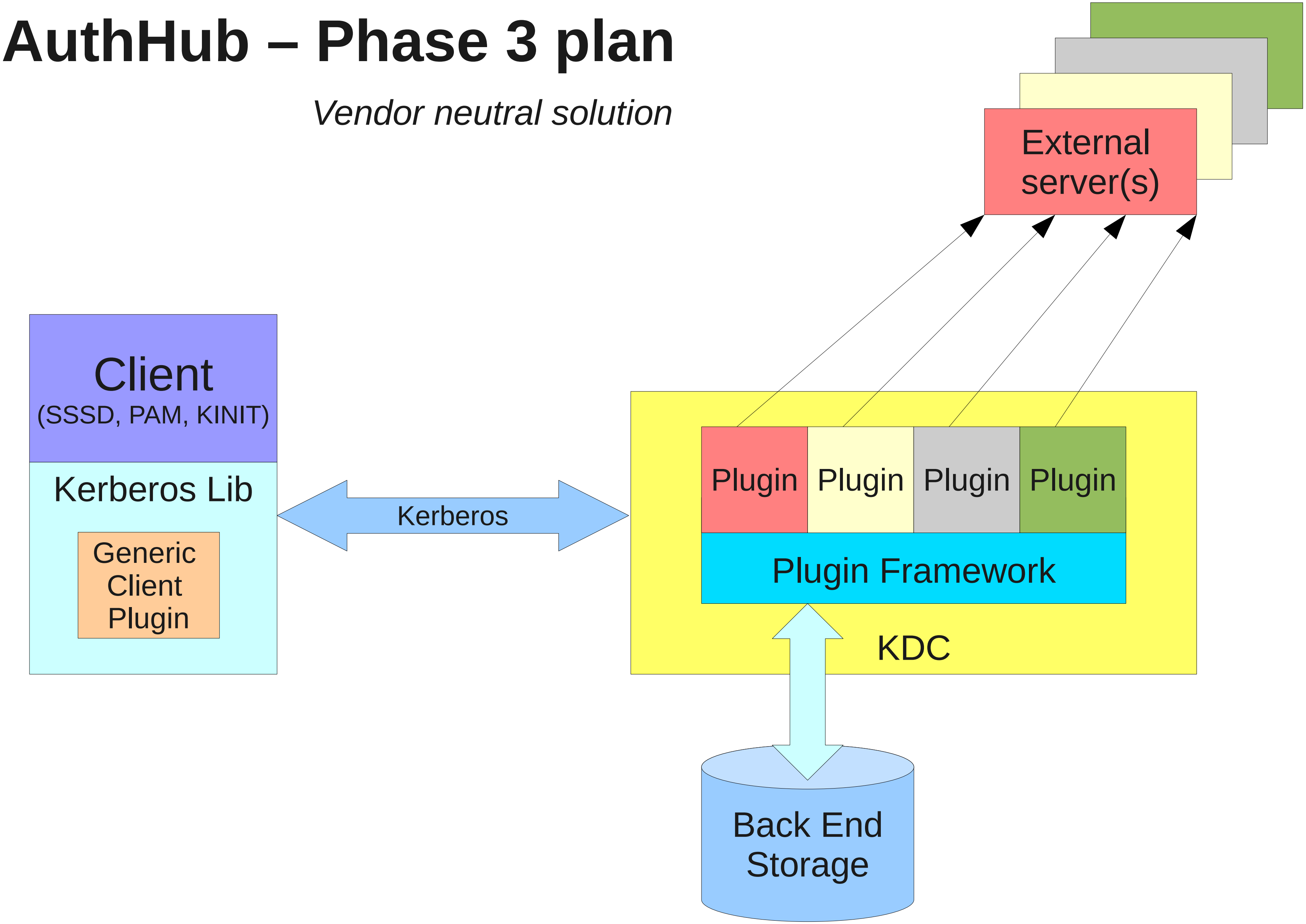
AuthHub – Phase 2 plan

Vendor specific solution



AuthHub – Phase 3 plan

Vendor neutral solution



AuthHub – Current Situation & Plan

- Focusing on KDC improvements to support external authentication
  - Libvirto – main async loop abstraction
  - Async processing inside KDC process was checked in
- Reach out to vendors again
- Next steps
  - Develop plugin framework
  - Implement prototypes using public interfaces:
    - Yubikey
    - Google authenticator

Plans for 2012 and Beyond

- Cross Realm Kerberos Trusts
  - Build Samba4 components against MIT Kerberos libraries
  - Deliver upstream functionality in spring
  - Release a supported version later in 2012
- Continue AuthHub project
- Automatic ticket rotation service
- Key rotation functionality
- Daemon to shield access to keytab when GSSAPI connection is established
- End-to-end Smart Card support
- Desktop integration

Automatic Ticket Rotation Service

- k5start – is not a part of the MIT tree
- Has AFS integration (not a part of Fedora or RHEL)
- A separate service – independent service
- Intent:
  - Create a service that would be a part of the MIT tree
  - Tickets will be renewed automatically during the GSSAPI exchange via this service
  - Design details yet to be discussed
  - Red Hat plans to contribute this functionality
  - Co-sponsors welcome!

Key Rotation

- Some customers would like to implement automatic key rotation functionality, mimicking policies implemented in AD.
- Intent:
  - Create a service that would be able fetch a new key
  - Design details yet to be discussed
  - Red Hat plans implement this functionality
  - Unclear how generic and independent from other components and technologies like SSSD it can be
  - Co-sponsors welcome!

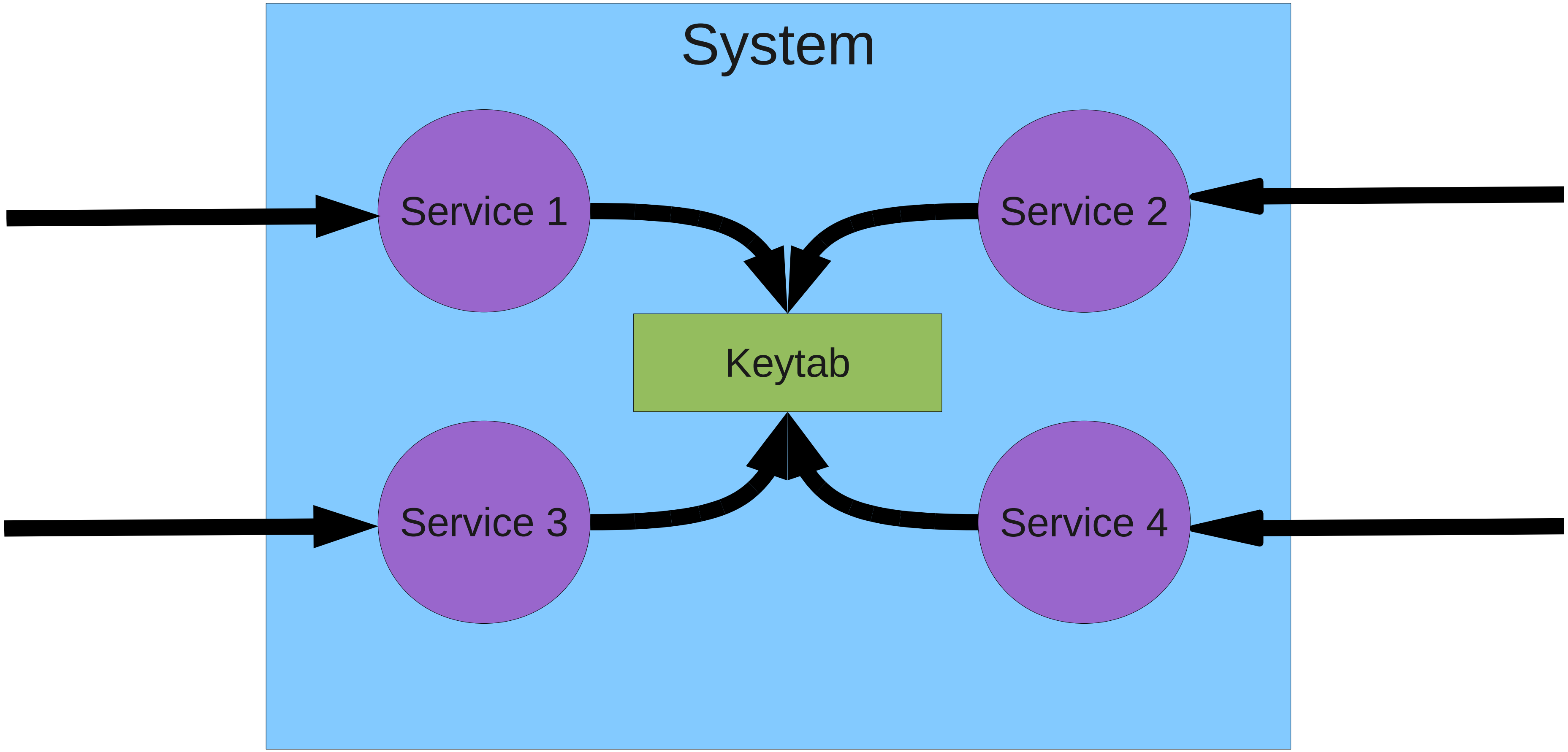
Daemon for GSSAPI Connections

- Problem:
  - Services need access to their keytabs.
  - Services are usually exposed to the network and some times easier to compromise.
- Problem in the context of FreeIPA:
  - Authorization data is passed around in the ticket (PAC)
  - Validating PACs against KDC is a costly operation
  - Cannot trust service signature on PAC if the key is available to a service that can be compromised

Daemon for GSSAPI Connections (continued)

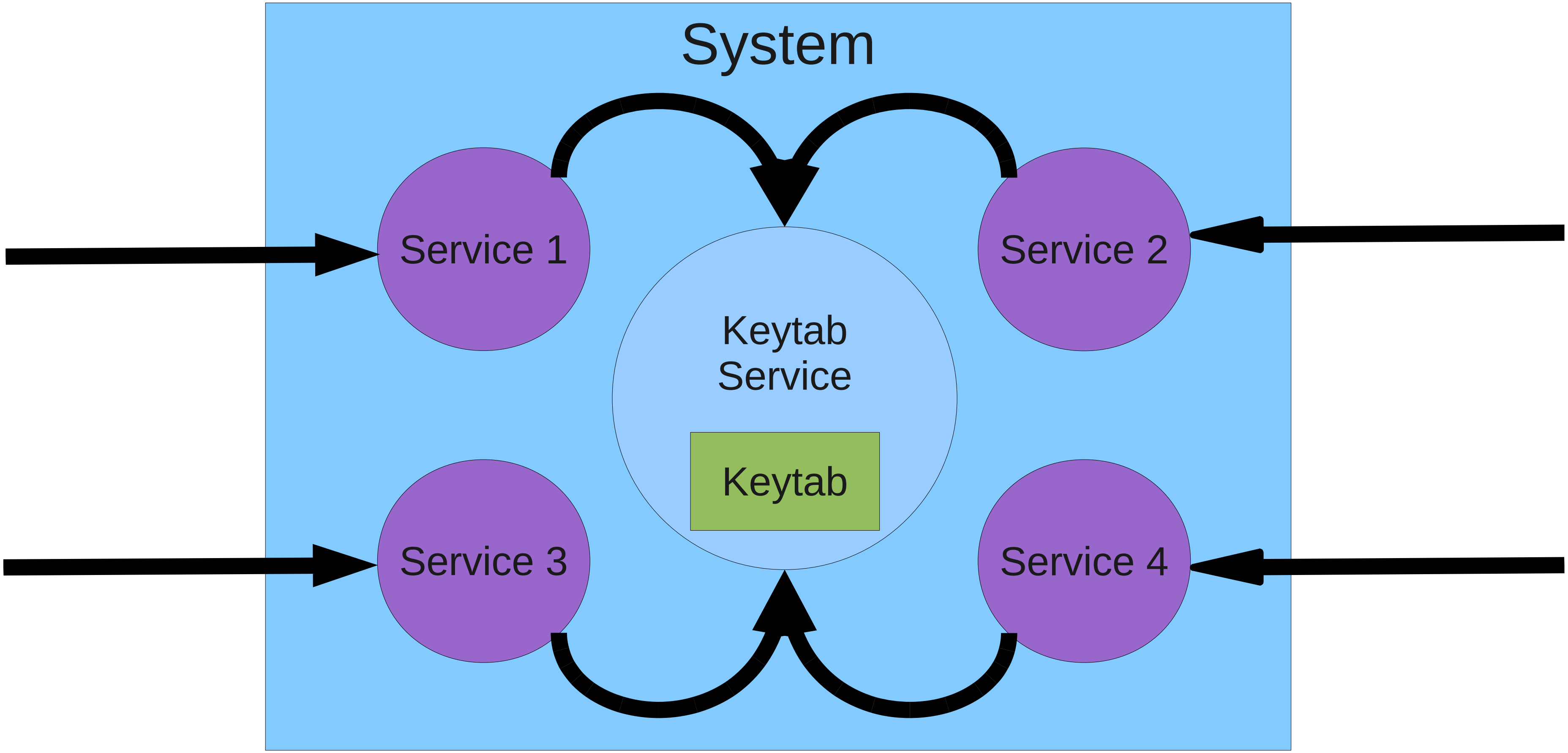
- Solution:
  - Create a special service that will have access to keytab
  - Other services will talk to that service during GSSAPI exchange
  - The new service will do the GSSAPI exchange on behalf of other services
  - libgssapi will proxy communication to the service transparently to applications. No change in APIs

Current situation with GSSAPI



If any service is compromised keytab is compromised and thus can’t be trusted as a PAC signature validation mean.

Proposed Solution



If a service is compromised, keytab is not compromised and still can be trusted to do PAC validation.

End-to-end SC support

- System Security Services Daemon (SSSD) improvements:
  - Add SC support
  - Add PKINIT support
- FreeIPA improvements:
  - Add automatic support for PKINIT
- KDC improvements:
  - Pass extensions from the certificate to the ticket
  - Level of assurance

Level of Assurance

- Tickets can be acquired in different ways:
  - Password
  - 2FA with external server
  - Smart Card
- Ticket should carry information about how it was acquired
- Would allow services to check and differentiate
- We want to start design discussion and come up with a specification within reasonable timeframe

Desktop Integration

- Support of the secondary identities
  - Nice UI around kinit for secondary identities
- Tighter integration of the ticket renewal UI
  - Support of the secondary identities
  - Better notification mechanisms
  - Simpler configuration
  - Prompting for the right credential

Questions?