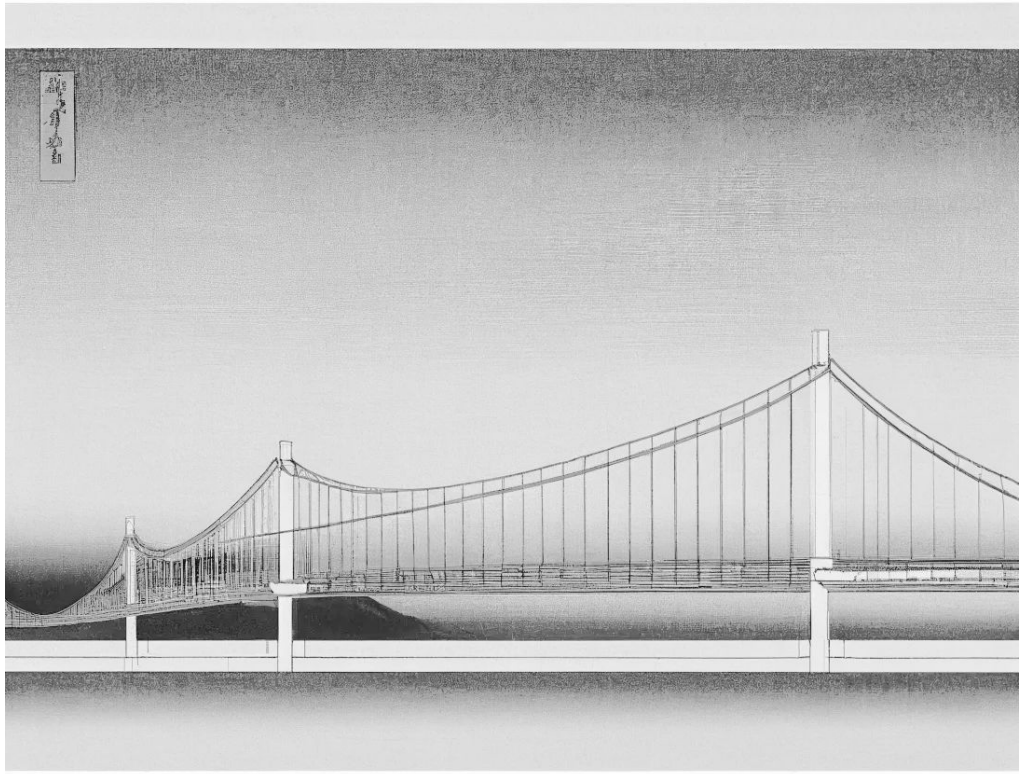
FreeIPA

Open Source Identity Management Solution

High level architecture

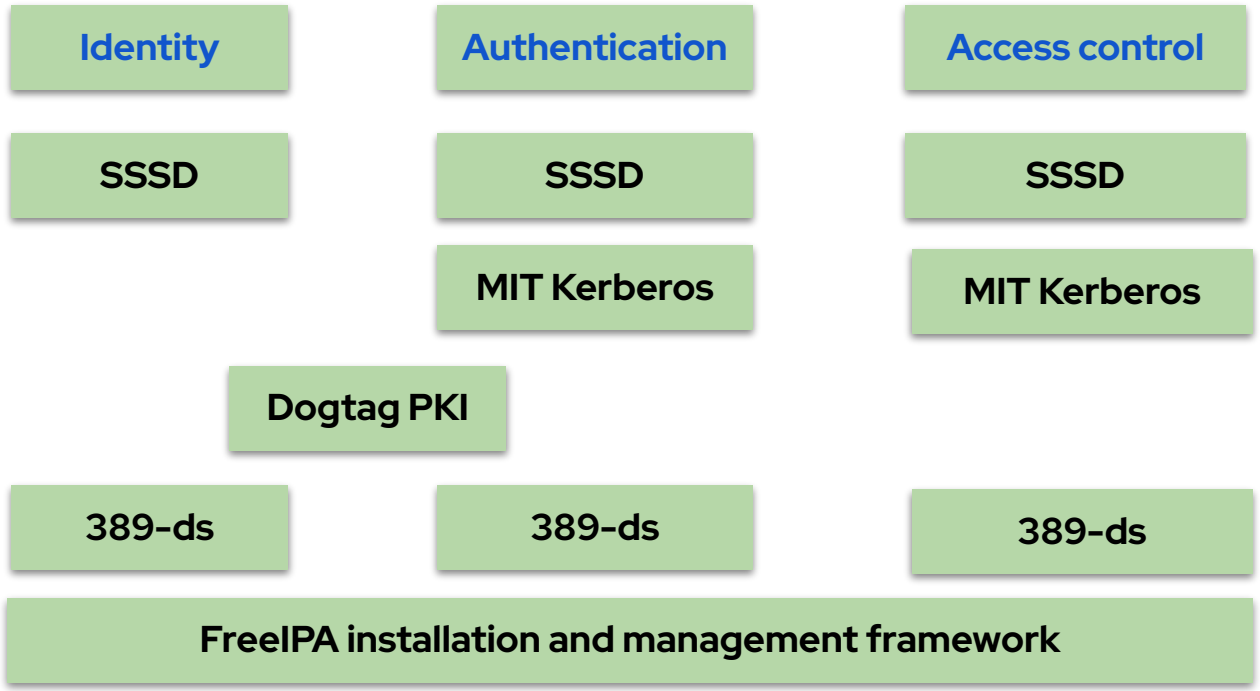
A reconstruction view



Alexander Bokovoy

Sr. Principal Software Engineer / Red Hat

FreeIPA pillars

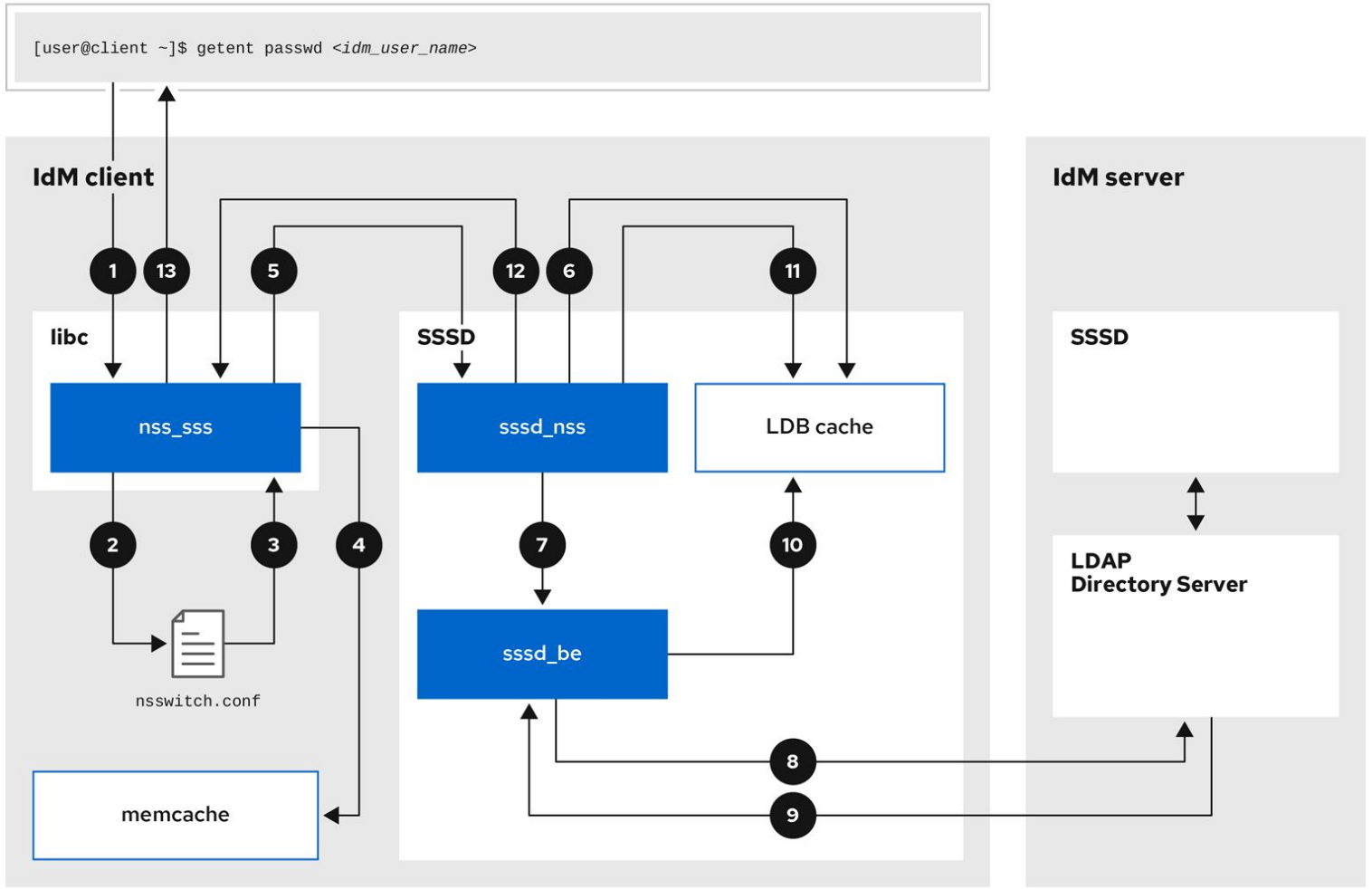


FreeIPA pillars in Active Directory trust

Identity	Authentication	Access control
Samba	Samba	Samba
SSSD	SSSD	SSSD
	MIT Kerberos	MIT Kerberos
Dogtag PKI		
389-ds	389-ds	389-ds
FreeIPA installation and management framework		

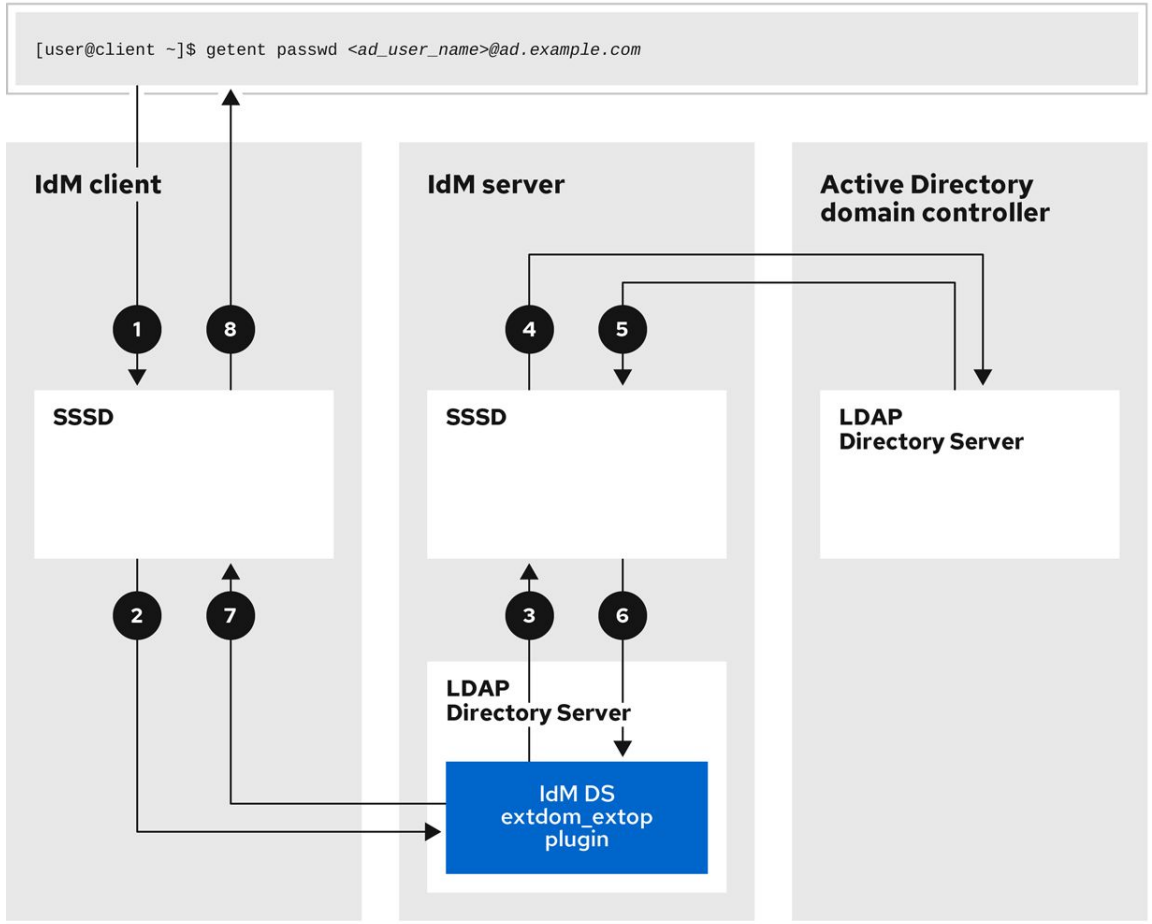
FreeIPA
Open Source Identity Management Solution

Identity resolution



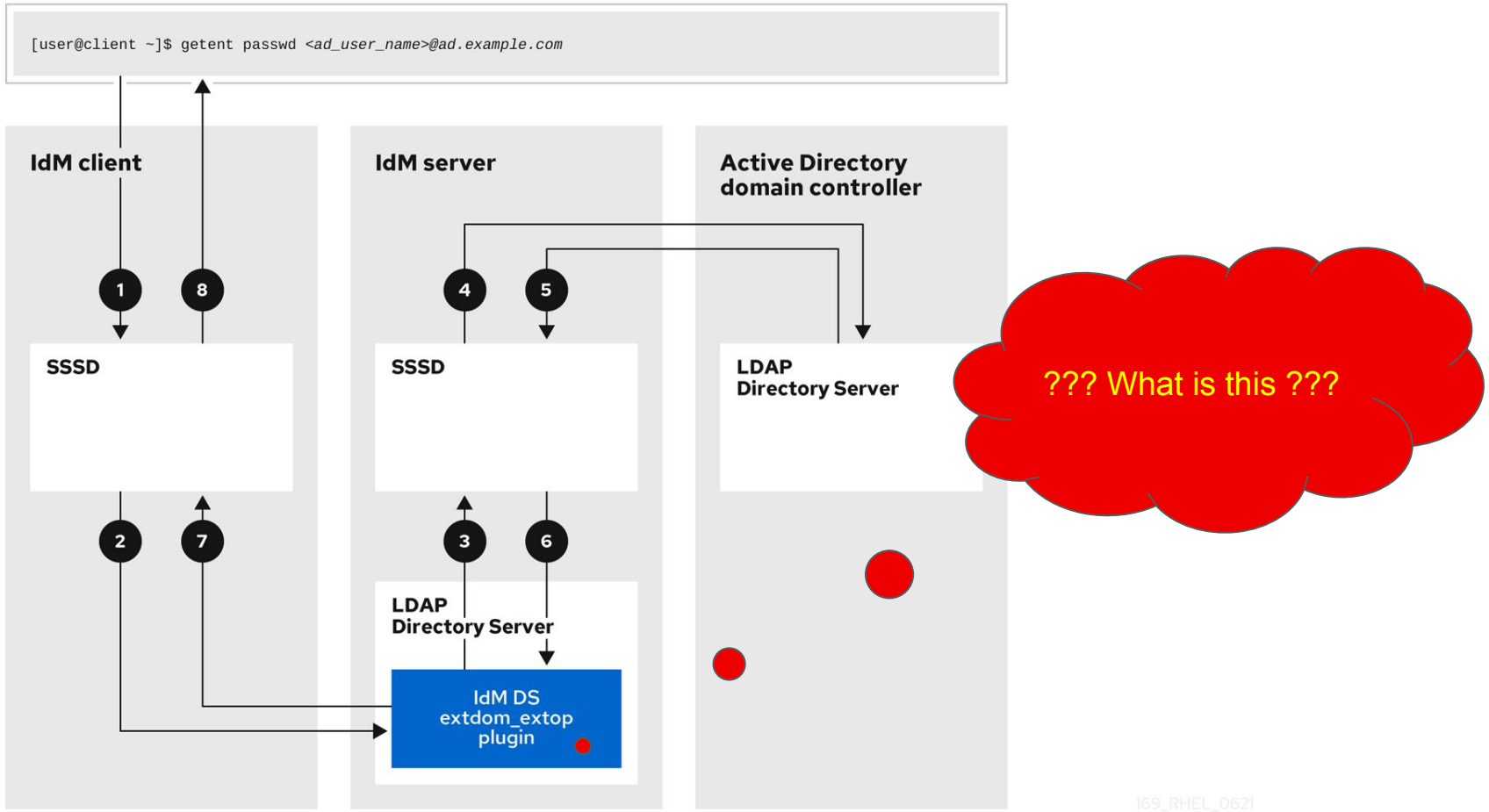
Detailed description is in RHEL IdM guide 'Configuring and managing Identity Management': 8.1. Data flow when retrieving IdM user information with SSSD

Identity resolution in Active Directory trust



Detailed description is in RHEL IdM guide 'Configuring and managing Identity Management': 8.2. Data flow when retrieving AD user information with SSSD

Identity resolution in Active Directory trust

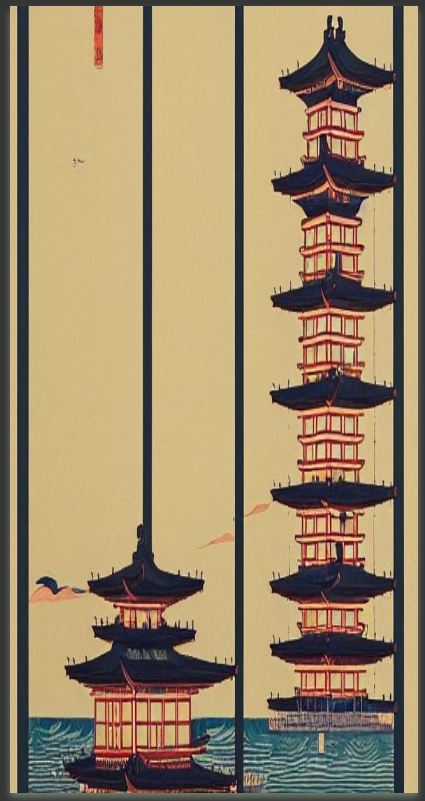


Detailed description is in RHEL IdM guide 'Configuring and managing Identity Management': 8.2. Data flow when retrieving AD user information with SSSD

FreeIPA and 389-ds LDAP server

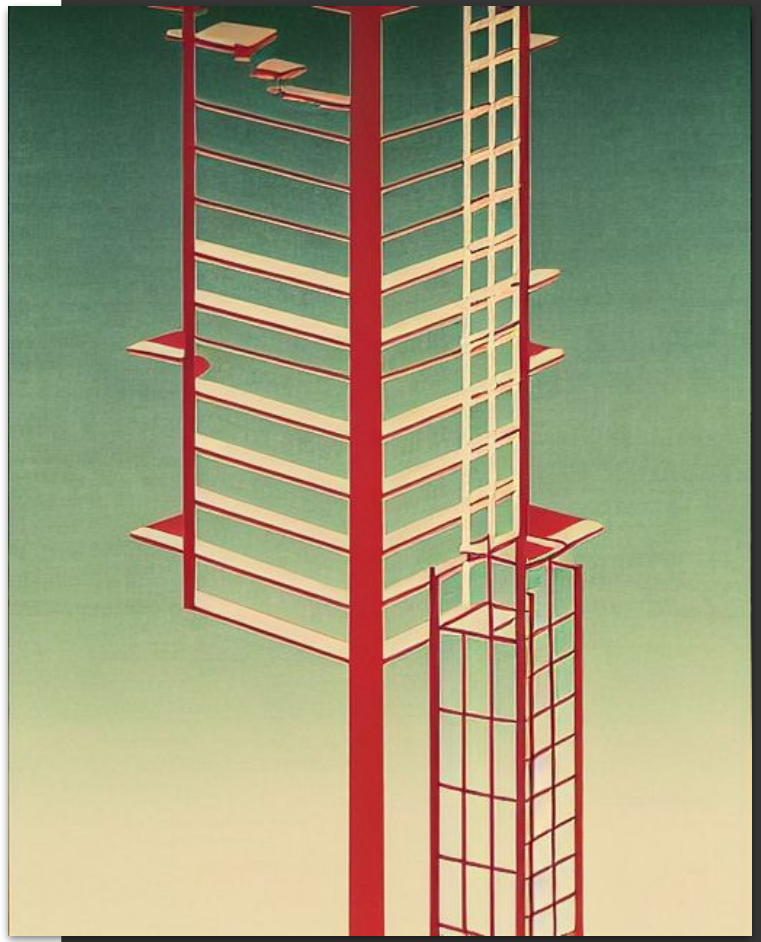
FreeIPA utilizes 389-ds Directory Server as its core component:

- LDAP database with a flat directory tree
- All similar objects placed in the same container
  - cn=users,cn=accounts,$SUFFIX
  - cn=groups,cn=accounts,$SUFFIX
  - cn=computers,cn=accounts,$SUFFIX
  - cn=services,cn=accounts,$SUFFIX
  - ...
  - ipa env | grep container_ | cut -d_ -f2-
    - 56 object types in FreeIPA 4.10
- Access to the objects is defined in terms of 389-DS ACIs
  - ACI: a rule that says how an LDAP bind might access certain sub-trees or individual records. Can be specified down to an individual attribute.
  - All objects with a password-like (users, computers, services) can be configured to access data with a permission-privilege-role in FreeIPA, it translates to a set of ACIs internally.



FreeIPA and 389-ds LDAP server

- 389-ds has more than 80 points that can be overridden in the processing of LDAP operation
- FreeIPA overrides some of them where it makes sense to extend the behavior
- Most common operations:
  - ADD (pre- and post-)
  - MODIFY (pre- and post-)
  - BIND (pre- and post-)
  - DELETE (pre- and post-)
  - MODRDN (pre- and post-)
- Most visible identity plugins
  - Compatibility tree (slapi-nis plugins)
  - Extended identity operations (ipa-extdom-extop)



FreeIPA and 389-ds LDAP server

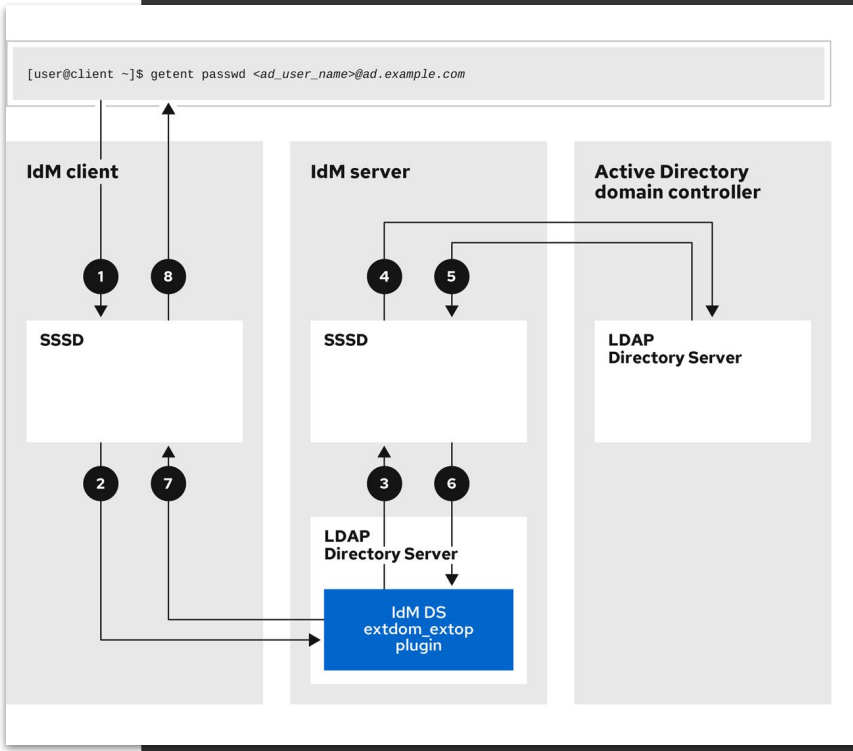
ipa-extdom-extop: extended LDAP operation to request identity information about users and groups from trusted Active Directory forests

Core of the Trust to Active Directory feature

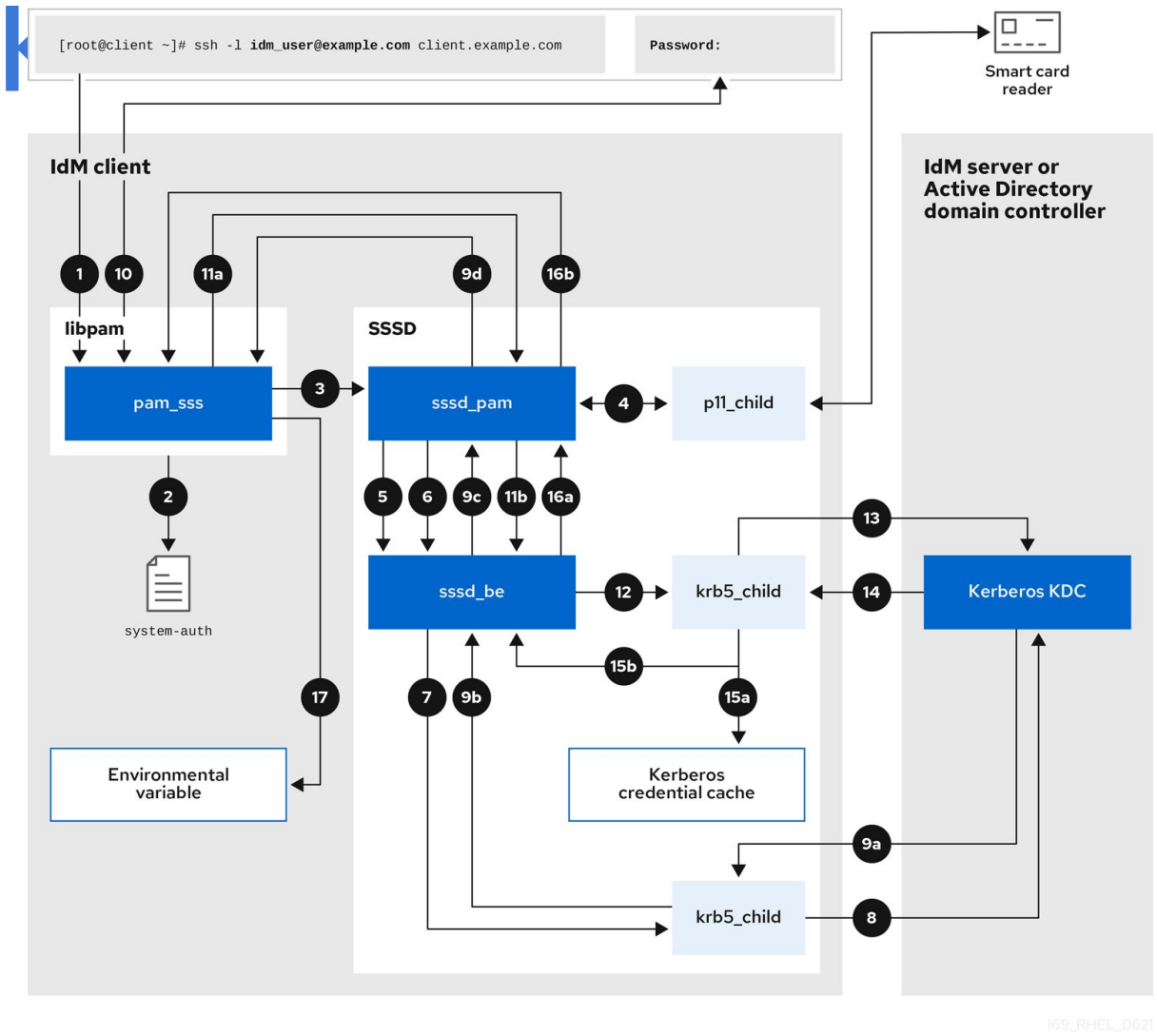
- SSSD on IPA clients uses LDAP extended operation to query information about AD users/groups or to convert SIDs to UID/GIDs
- At LDAP server the request is intercepted by the ipa-extdom-extop plugin
- Requests passed to the SSSD running on the IPA server
- SSSD on the IPA server will look up information against the trusted forest's domain controllers
- Responses will be returned back to LDAP client (SSSD on IPA clients)

Benefits:

- Only IPA servers need to know trusted object credentials when talking to AD DCs
- Multiple IPA clients may ask the same questions, cache reuse increases performance



Authentication with



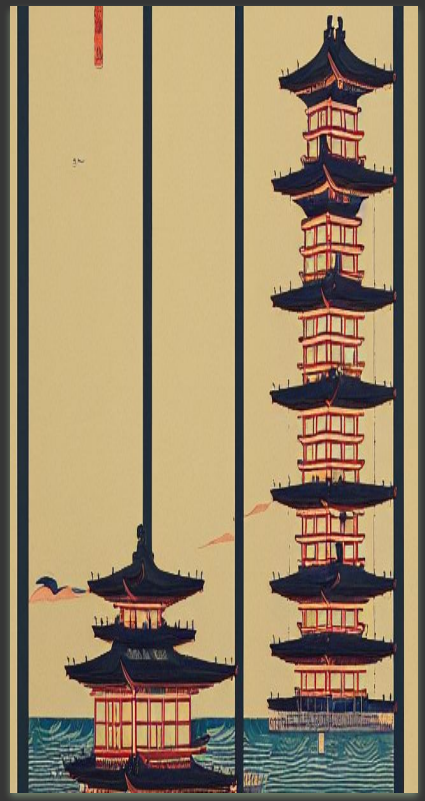
Detailed description is in RHEL IdM guide 'Configuring and managing Identity Management': 8.3. Data flow when authenticating as a user with SSSD in IdM

FreeIPA and 389-ds LDAP server

Kerberos authentication to LDAP server

- LDAP server utilizes SASL to abstract authentication
  - SASL GSSAPI or SASL SPNEGO mechanisms allow using Kerberos

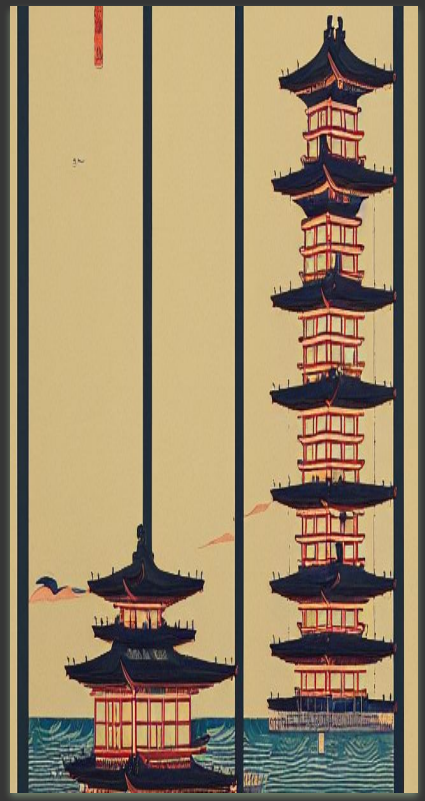
- LDAP server does not check password for Kerberos authentication
  - 3-way negotiation is performed in Kerberos
    - A client authenticates itself against KDC
    - Client uses a service ticket obtained from KDC to talk to 389-ds
    - 389-ds decodes the provided ticket using its own key
  - At no point LDAP server has any idea how client authenticated to KDC
- if SASL GSSAPI or SASL SPNEGO authentication is successful, LDAP server maps authenticated identity to LDAP object
- LDAP object's DN becomes the 'bind DN' for the rest of access control checks



FreeIPA and 389-ds LDAP server

LDAP objects with a password

- LDAP schema defines attributes and object classes
  - userPassword attribute is used to represent a password. It is a multivalued attribute, may represent multiple password forms (different hashes, clear text)
- In the standard LDAP schemas many object classes can have passwords:
  - organization, person, organizationalUnit, domain, simpleSecurityObject, posixAccount, shadowAccount, posixGroup, ipHost
- LDAP bind process
  - Done by hashing a clear text bind password and comparing the result with hashed values of the userPassword attribute in the bind DN entry
  - There are multiple 'password storage scheme' methods for hash generation in 389-ds
  - 389-ds does not know about multi-factor authentication, so 2FA does not work directly



FreeIPA and 389-ds LDAP server

LDAP password change process

- Traditionally LDAP client either modifies userPassword attribute values directly or provides a clear text
  - LDAP modify accepts 'whatever' is given and performs modification of just that specific value
  - LDAP password change operation accepts a clear text

- What to do if a clear text password is provided to modify userPassword but there is no clear text version in the userPassword attribute?
  - 389-ds needs to hash the clear text password to find out which hash would match

- Password policy is applied to clear text password on a change
  - Policies are flexible but they assume access to clear text passwords at the change time

- 389-ds password change process cannot create multiple hashes out of the same clear text password
  - Only default password hash type is used

FreeIPA and 389-ds LDAP server

Kerberos objects in LDAP

- LDAP schema defines attributes and object classes
  - FreeIPA derived its Kerberos schema from the one originally defined by Novell Inc. in 2006
  - Canonical principal name is matched exactly
  - Principal aliases are searched by ignoring the case
  - Kerberos key attribute is encrypted with the master key of the Kerberos realm and may contain multiple keys with different encryption types
  - Kerberos policies apply at key (or password) change

FreeIPA and 389-ds LDAP server

Kerberos principal password change via LDAP

- LDAP object for kerberos principal has a separate attribute for its Kerberos key
- 389-ds does not know how to generate it from a clear text password
- 389-ds does not know how to apply Kerberos password policy

Password change from KDC side

- KDC uses a database driver that knows FreeIPA LDAP schema
- KDC directly stores computed Kerberos keys in LDAP entries
- KDC applies Kerberos password policy

389-ds LDAP server is extended with plugins to blend Kerberos and LDAP password operations

- **ipa-pwd-extop** plugin handles password-related updates
- **ipa-enrollment** plugin converts computer entry to Kerberos service on enrollment
- **ipa-winsync** plugin makes Windows users IPA Kerberos principals on replication
- **ipa-lockout** plugin blends LDAP authentication and Kerberos policies together

FreeIPA and 389-ds LDAP server

ipa-pwd-extop plugin is the main engine behind LDAP authentication and keytab retrieval

Password-related operations:

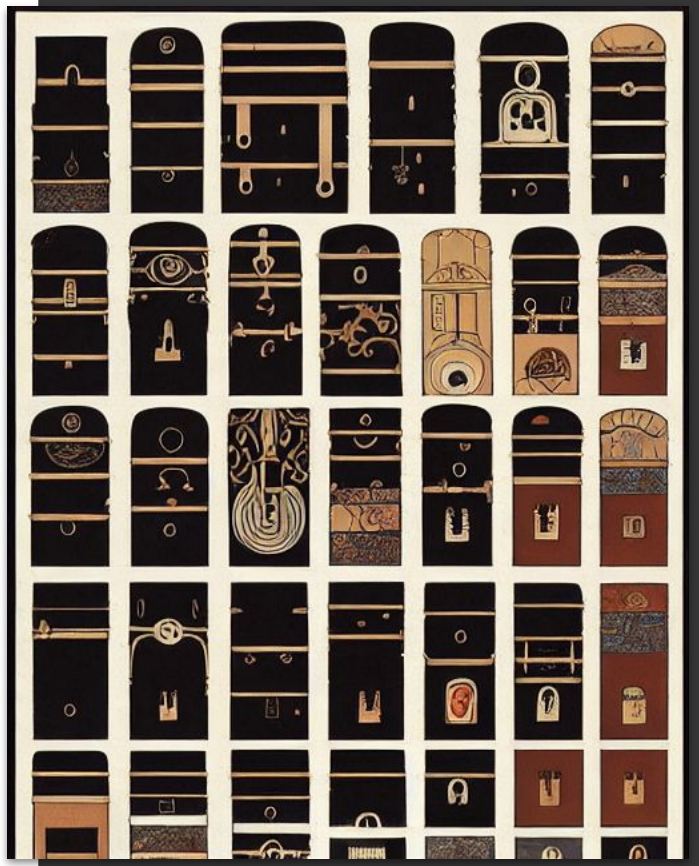
- pre-BIND operations:
  - Password migration
  - 2FA token validation and synchronization
- pre-MODIFY and pre-ADD operations:
  - password policy check
  - (re)generate password keys for Kerberos
  - (re)generate password hashes for Samba
- post-MODIFY and post-ADD operations:
  - password history update
  - password expiration update for the Kerberos keys
    - krbPasswordExpiration is set for hosts
    - krbLastPwdChange is set for all other entries
    - 2FA token configuration update

FreeIPA and 389-ds LDAP server

ipa-pwd-extop plugin is the main engine behind LDAP authentication and keytab retrieval

Kerberos key-related operations:

- Keytab generation via special LDAP extended operation:
  - v1: set Kerberos key for a principal based on what client has sent
    - used by Samba integration to handle RC4 hashes on a password change through the SMB protocol
  - v2: extended to avoid accepting keys from the client
    - generate Kerberos key for a principal at the server side and return to the client
    - allow retrieval of the existing key subject to access control
    - used by Samba integration to produce RC4 password hashes out of Kerberos keys



FreeIPA and 389-ds LDAP server

ipa-lockout plugin blends LDAP authentication and Kerberos policies together

pre-BIND operations:

- Applies lockout policy
  - denies BIND for a specified time interval if krbLoginFailedCount is above krbPwdMaxFailure attribute value from the policy associated with the bind DN

post-BIND operations:

- update Kerberos password policy attributes after successful or unsuccessful LDAP BIND operation
  - on unsuccessful BIND the following attributes updated:
    - krbLoginFailedCount and krbLastFailedAuth
  - on successful BIND
    - resets krbLoginFailedCount
    - updates krbLastSuccessfulAuth if IPA config allows (replication storms)

FreeIPA and MIT Kerberos KDC server

KDC is a Key Distribution Center in Kerberos realm

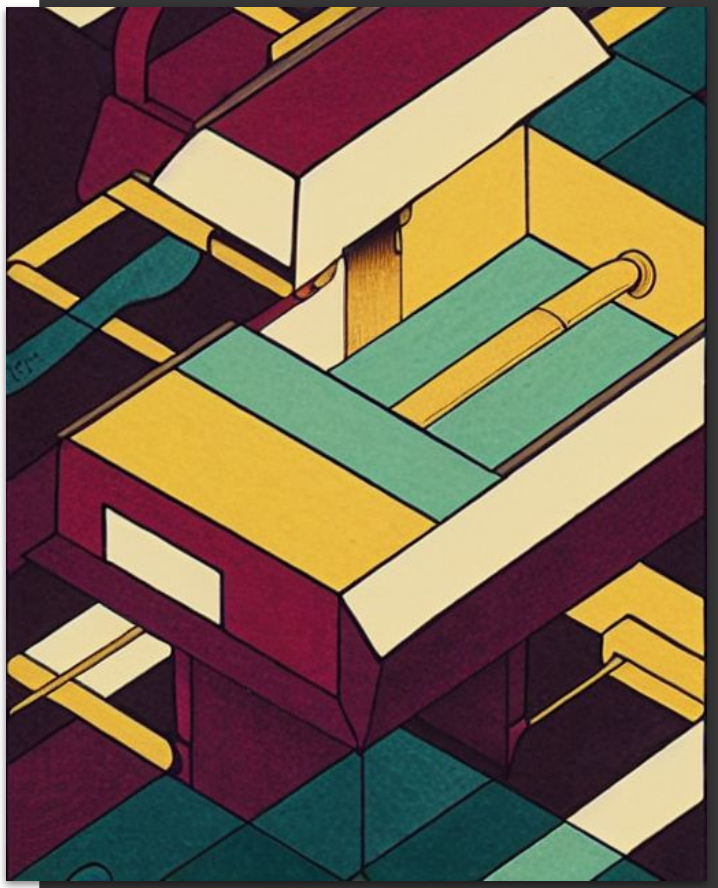
- KDC knows about Kerberos principals and their keys
- KDC authenticates Kerberos principals and issues tickets for a limited time
- Kerberos clients ask KDC for service tickets to other Kerberos principals (services)

MIT Kerberos is one of the two opensource Kerberos implementations

- MIT Kerberos: https://web.mit.edu/Kerberos/
  - Used in RHEL and Fedora, required by FreeIPA
- Heimdal Kerberos: https://www.heimdal.software/
  - Used by Samba AD by default and by Apple products

MIT Kerberos allows to tune the behavior of KDC with a database driver, KDB

- KDB abstracts out storage of Kerberos principals and authorization policies
- KDB may implement additional features like MS-PAC support and audit operations



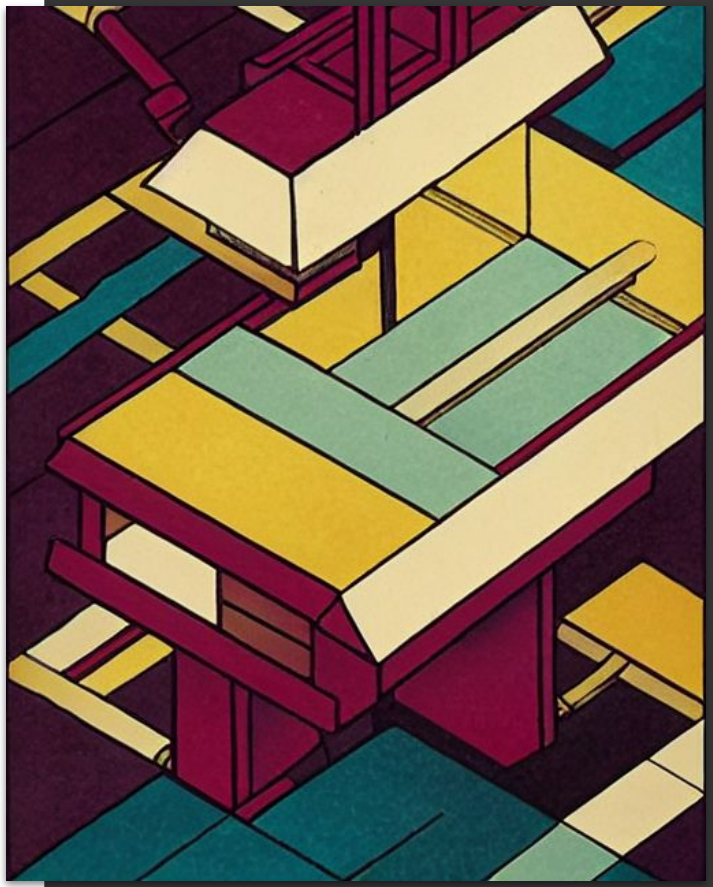
FreeIPA and MIT Kerberos KDC server

ipa-kdb is a KDB driver FreeIPA provides:

- stores Kerberos principals information in LDAP
- stores encrypted Kerberos keys in LDAP
- stores Kerberos password policies in LDAP
- implements identity and policy part of new Kerberos pre-authentication methods

Configuration is deployed during the initial server provisioning

- IPA server installer defines LDAP container to store Kerberos information
- IPA replica installer reuses already existing Kerberos keys after an initial LDAP replication is completed
- KDC configuration files define minimal bootstrap configuration, everything else is sourced from LDAP



FreeIPA and MIT Kerberos KDC server

Kerberos pre-authentication methods

- actual method Kerberos client uses to prove possession of the principal's credentials
- the method to pre-authenticate because authentication means ticket issuance

Pre-authentication methods supported by FreeIPA

- Password-oriented methods
  - pre-authentication with encrypted timestamp and SPAKE exchange
- Public key certificate-based pre-authentication: PKINIT
- RADIUS-based authentication: OTP pre-authentication
- External IdP-based authentication: SSSD IdP pre-authentication
- Wrappers for other methods: FAST channel pre-authentication required for OTP and IdP methods

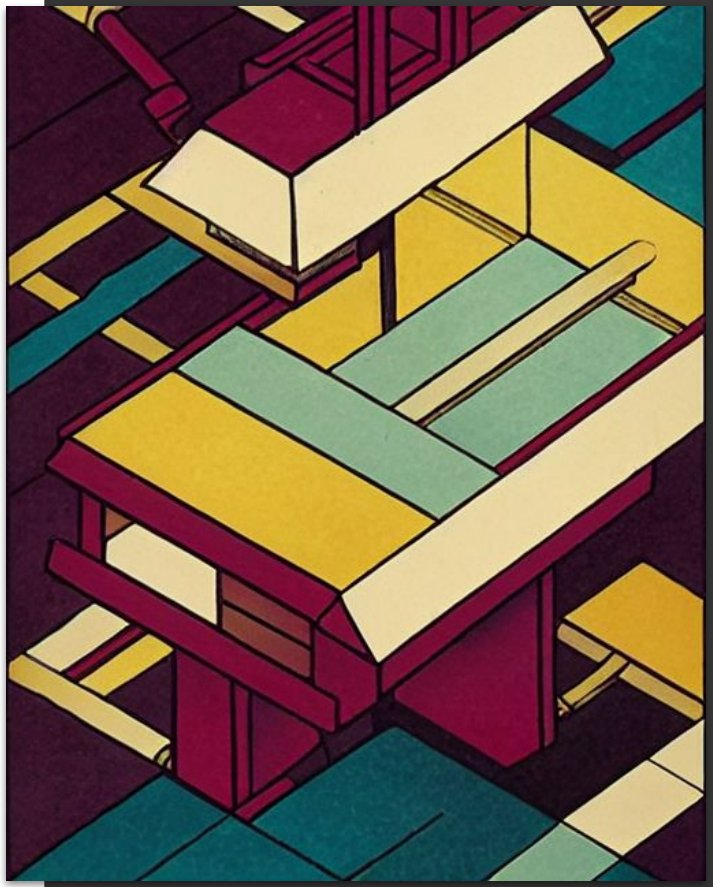
```
[47700] 1530630765.756232: Received answer from stream 17.10.20.22:88
[47700] 1530630765.756302: Response was from master KDC
[47700] 1530630765.756321: Received error from KDC: -1765328360/Preauthentication failed
[47700] 1530630765.756325: Decoding FAST response
[47700] 1530630765.756376: Preauth tryagain input types: 136, 19, 138, 133, 137
kpasswd: Preauthentication failed getting initial ticket
```

FreeIPA and MIT Kerberos KDC server

Kerberos pre-authentication methods requires coordinated operation of Kerberos client, KDC, and KDB.

OTP pre-authentication:

- KDB driver needs to inform the KDC that this Kerberos principal supports OTP pre-authentication
- KDC needs to tell the Kerberos client which pre-authentication methods supported
- Kerberos client needs to send OTP details to KDC over a secure channel provided with FAST pre-authentication method
- KDC needs to talk to a RADIUS backend to authenticate and authorize the principal using OTP details
- RADIUS backend (ipa-otpd) will need to do its magic to grant ticket issuance by KDC

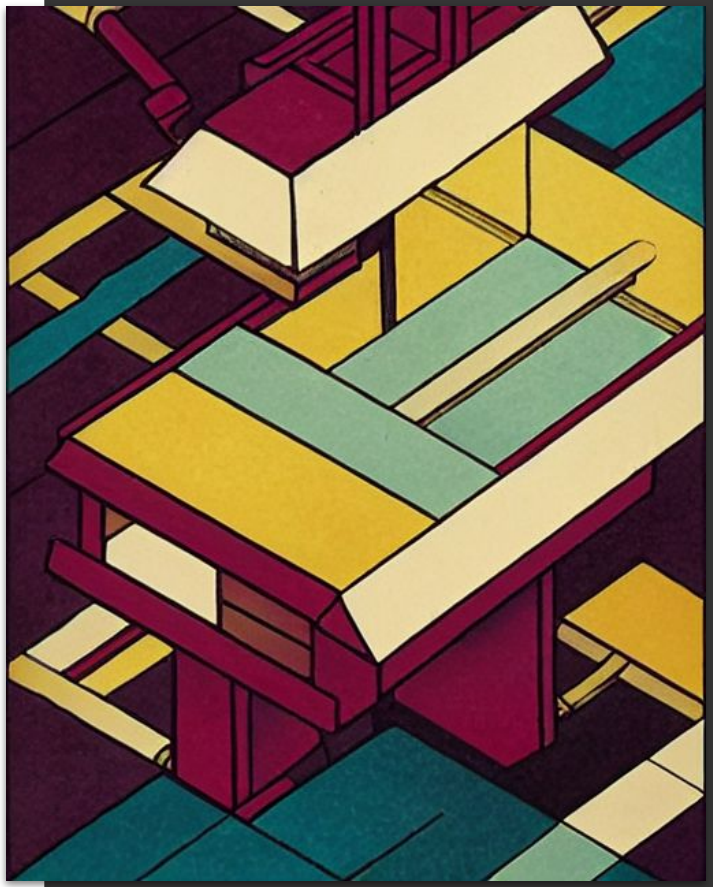


FreeIPA and MIT Kerberos KDC server

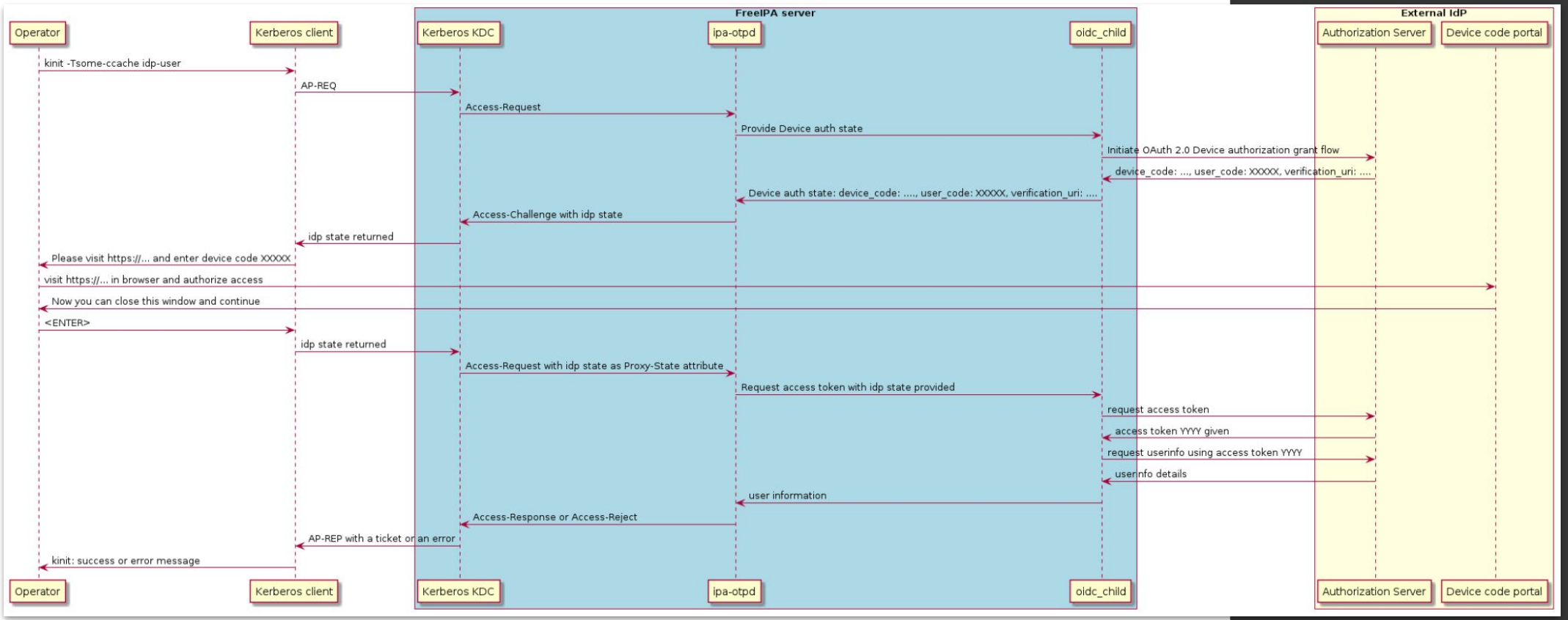
Kerberos pre-authentication methods requires coordinated operation of Kerberos client, KDC, and KDB.

SSSD idp pre-authentication:

- KDB driver needs to inform the KDC that this Kerberos principal supports external IdP pre-authentication
- KDC needs to request an IdP prompt from a RADIUS backend (ipa-otpd) and tell the Kerberos client which pre-authentication methods supported
- Kerberos client will need to show IdP prompt to the user
- KDC needs to talk to a RADIUS backend (ipa-otpd) to wait until external IdP authorizes IPA OAuth2 client access to user's details
- RADIUS backend (ipa-otpd) will need to do its magic to grant ticket issuance by KDC



Authentication with external IdP in Kerberos



FreeIPA access control

Centralized access control in FreeIPA is not uniform

- Host-based access control rules (HBAC)
  - Rules defined as (host, service, subjects) triplets
  - Host is a set of hosts or host groups
  - HBAC service is a PAM service name in Linux
  - Subjects are users or groups of users the rule applies to
- HBAC rules enforced by SSSD
  - On the target host
  - Evaluated against the POSIX user attempting to access a PAM service

HBAC rules do not apply to LDAP access

- 389-ds LDAP server does not use PAM for authorization
- 389-ds has its own access controls: ACIs

HBAC rules do not apply to Kerberos access

- KDC does not limit access to initial ticket granting ticket: either you possess credentials or you are not (except for consecutive failures and lockouts)
- KDB driver may apply Kerberos authentication indicators to decide whether a service ticket can be granted or not

FreeIPA access control

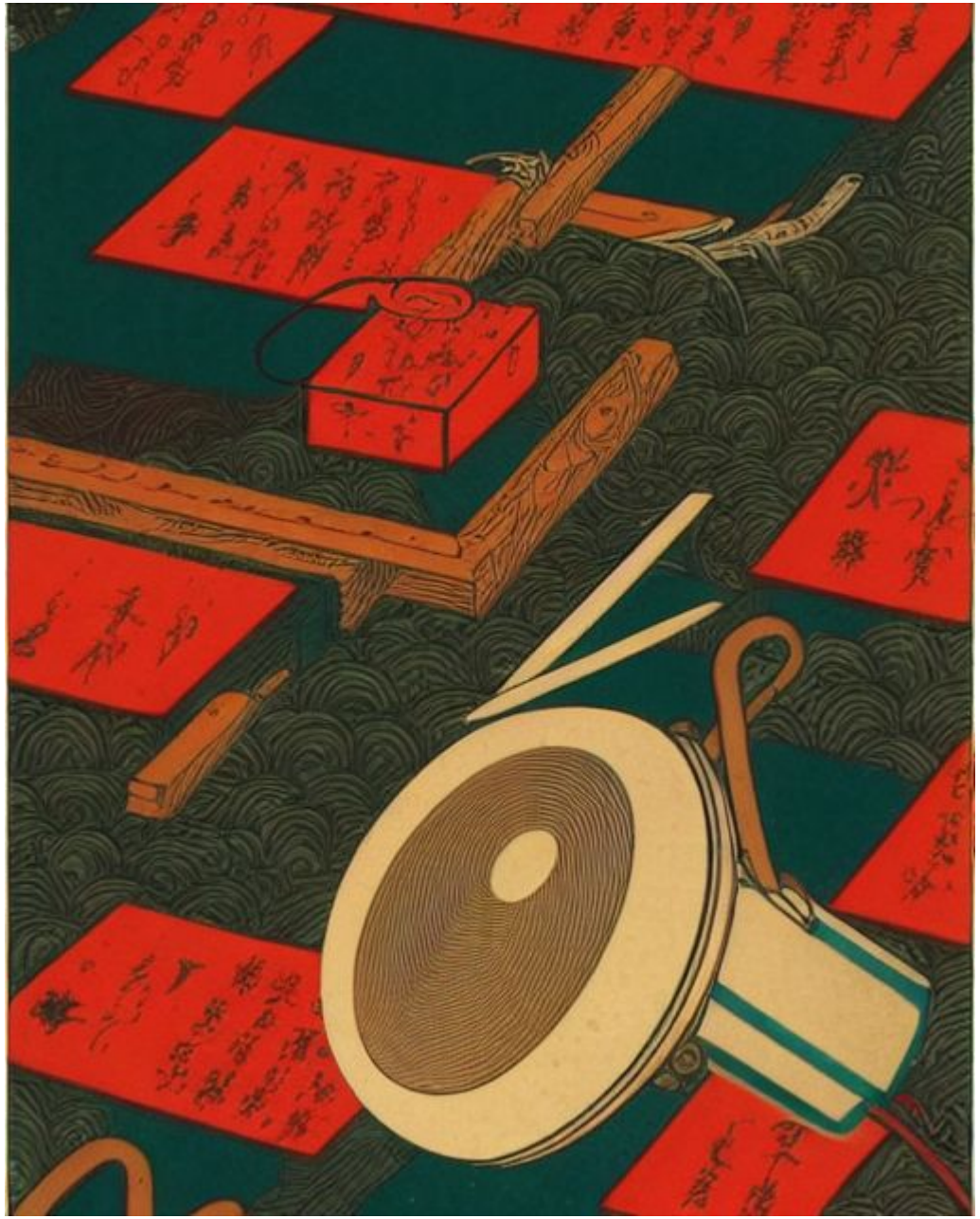
Kerberos access and authentication indicators

- Authentication indicator
  - How the ticket granting ticket was obtained
    - Which pre-authentication method was used in that process
  - Copied from the TGT to a service ticket

- At KDC side
  - KDB driver may reject request to issue a service ticket if presented TGT lacks a particular authentication indicator
  - Kerberos service ticket will never be issued to ‘wrongly’ authenticated principals
- At the application level
  - Kerberos application may check the service ticket to see if it contains an authentication indicator of a specific value
  - SSSD’s pam_sss_gss implements this approach
  - Apache’s mod_gssapi implements this approach

FreeIPA
Open Source Identity Management Solution

Questions?



Images generated with the help of a Stable Diffusion driver using ukiyo-e style prompts