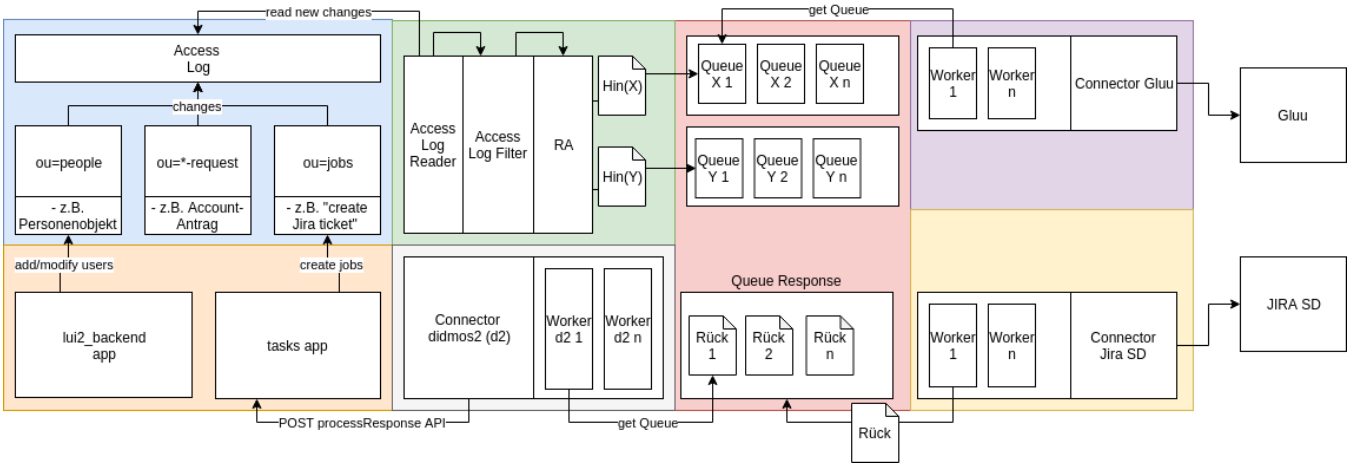
didmos2 Provisioner

Overview

didmos2 Provisioner is a software package that allows the provisioning of an OpenLDAP database to target systems. It transfers new created user objects as well as every modification to the configured target systems. The following picture shows the flow of didmos2 Provisioner:



On the one hand, didmos2 Provisioner is based on the OpenLDAP access log (blue box) to learn about data changes and to generate individual documents from it. On the other hand, Rabbit MQ is used to forward these documents to the systems responsible for the operation of the target systems, called workers . The seamless and secure transfer of documents is guaranteed by RabbitMQ even in the event of network failures.

RA (Requesting Authority, green box) is responsible for collecting data changes in individual documents. OpenLDAP Access Log is queried at regular intervals with a search filter to be configured for data changes. For each entry found in the access log, a SCIM document is generated (configurable) for each target system concerned, and stored in a temporary queue in RabbitMQ that is specific to the object concerned (red box). Change documents for the same object are stored in the same queue.

The associated worker is notified by a permanent queue. There are one or more workers (scalable as required) per target system. The worker reads the job from his permanent queue and thus knows from which temporary queue the SCIM documents can be obtained. The attribute mapping SCIMTarget System is configurable. Each document is transmitted to the target system via the ICF interface (Identity Connector Framework) and the status is stored in a temporary queue as a SCIM document, which is also processed by another worker (response worker, gray box). The response worker receives the document stating whether and which actions should be carried out on the didmos2 backend. Communication with didmos2 backend again takes place via a corresponding ICF connector.

Existing connectors

As already mentioned, the communication between didmos2 Provisioner and the target systems takes place via so called ICF connectors. ICF is a standard interface designed for the transfer of identity data.

The following ICF connectors are currently supported:

- SCIM connector (provided by DAASI International and used for various worker implementations for communication with e.g. didmos Core and other systems)
- LDAP connector (provided by Evolveum)
- Active Directory LDAP connector (provided by Evolveum)
- Gluu Connector (provided by DAASI to synchronize and provision data to GLUU)

Connectors for other systems can be developed and integrated.

Overview of configuration parameters

General parameters

Parameter name	Description
LOG_LEVEL	Logging level
RECEIVE_QUEUE	The name of the RabbitMQ queue from where the worker gets the requests
RESPONSE_QUEUE	The name of the RabbitMQ queue to which the worker puts the responses
RETRY_TIME	The time in seconds to wait before retrying an action

RabbitMQ parameters

Parameter name	Description
RABBITMQ_ADDRESSES	RabbitMQ server URL
RABBITMQ_PORT	RabbitMQ server port
RABBITMQ_USERNAME	RabbitMQ user name
RABBITMQ_PASSWORD	RabbitMQ user password

didmos2 backend connector parameters

Parameter name	Description
RESPONSE_API_URL	The backend REST URL
RESPONSE_USER_NAME	didmos2 user name for basic authentication
RESPONSE_USER_PASSWORD	didmos2 user password for basic authentication
RESPONSE_RESOURCE_TYPES	different resourcetypes sent by the RA
RESPONSE_OBJECT_CLASSES	matching Objectclasses to resourcetypes (order of entries ind the list =^ mapping)
RESPONSE_ENDPOINTS	matching endpoints at didmos2 BE to objectclasses / resourcetypes (order of entries ind the list =^ mapping)

LDAP/AD LDAP connector parameters (see also https://wiki.evolveum.com)

Parameter name	Description
LDAP_SERVER	LDAP server name
LDAP_PORT	LDAP server port
ALLOW_UNTRUSTED_SSL	Whether connector skips certificate validity check against its default truststore (e.g. Java cacerts) When set to false, connector checks server certificate validity in SSL/TLS mode (recommended) When set to true, connector does not check server certificate validity. Do not use this option in the production
ENABELED_SECURITY_PROTOCOLS	Set of security protocols that are acceptable for protocol negotiation Possible values: SSL, SSLv2, SSLv3, TLS, TLSv1, TLSv1.1, TLSv1.2
CONNECT_TIMEOUT	Timeout to connect (in milliseconds)
MAX_NUM_ATTEMPTS	Maximum number of attempts to retrieve the entry or to re-try the operation This number is applicable in replicated topology when handling connection failures and re-trying on another server, when following referrals and in similar situations
AUTHENTICATION_TYPE	The authentication mechanism to use Possible values: simple, SASL-GSSAPI Default value: simple
BASE_CONTEXT	The base DN that the connector will use if the base DN is not specified explicitly
BIND_DN	The DN of the object to bind to
BIND_PASSWORD	Bind password
USE_PERMISSIVE_MODIFY	Use permissive modify LDAP control for modify operations Possible values: never, auto, always Default value: auto
PAGING_STRATEGY	Specifies strategy of using paging mechanisms such as VLV or Simple Paged Results Possible values: none, auto, spr, vlv Default value: auto
PW_HASH_ALGORITHM	Hash the passwords with a specified algorithm before they are sent to the server
UID_ATTRIBUTE	Name of the attribute which will be used as ICF UID
OPERATIONAL_ATTRIBUTES	Operational attributes that apply to all object classes

STRUCTURAL_OBJECT_CLASS	If set to true, adds all additional structural object classes without children to the auxiliary object classes list on the connector

Additional AD and LDAP connector parameters (see also https://wiki.evolveum.com)

Parameter name	Description
USER_OBJECT_CLASS	Object class to use for user accounts. Default: user
GROUP_OBJECT_CLASS	Object class to use for user accounts. Default: group
MEMBER_ATTRIBUTE	Group member attribute name. Default: member
GLOBAL_CATALOG_STRATEGY	Strategy of global catalog usage Do not use global catalog explicitly. The global catalog will only be used when following the referrals
ALLOW_BRUTE_FORCE_SEARCH	If set to true then the connector will try to search all defined servers for an entry if all other attempts fail
RAW_USER_ACCOUNT_CONTROL_ATTRIBUTE	If set to false then the connector will interpret the content of userAccountControl attribute and will decompose it to pseudo-attributes for enabled state, lockout, etc. If set to true then the connector will NOT do any interpretation and the userAccountControl will be exposed as a simple attribute.
NATIVE_AD_SCHEMA	If set to true, then the connector will use native AD schema definition. If set to false, connector will use LDAP-like schema definition exposed by the AD server. Default value: false EXPERIMENTAL. There may be subtle differences between LDAP schema and AD schema. Not completely tested yet.
TWEAK_SCHEMA	Extend the declared AD schema with tweaks that allow practical usage of the schema. AD will generally allow any attribute to be set to any object regardless for the schema. This is often used is practice. E.g. declared AD schema for users and groups does not include samAccountName attribute. But that attribute is needed for users and groups to work correctly. If this configuration property is set to true (which is the default) then the connector will artificially add these attributes to the schema.
INCLUDE_OBJECT_CATEGORY_FILTER	Enables inclusion of explicit object category filter in all searches. Normally the connector would derive search filter only based on the attributes specified in the query. E.g. (&(uid=foo)(cn=bar)). If includeObjectClassFilter is set to true, then also explicit filter for objectClass and objectCategory will be included. E.g (&(objectClass=inetOrgPerson)(objectCategory=CN=Person,CN=Schema,CN=Configuration,DC=example,DC=com)(uid=foo)(cn=bar)) Only works if includeObjectClassFilter is enabled and native AD schema is used. Default value: false. EXPERIMENTAL. Not completely tested yet.
ADD_DEFAULT_OBJECT_CATEGORY	If set to true then the connector will automatically add default object category to all created objects. Object category is automatically determined from schema. Only works if native AD schema is enabled. Default value: false. EXPERIMENTAL. Not completely tested yet.
FORCE_PASSWORD_CHANGE_AT_NEXT_LOGON	If set to true then the connector will force password change at next log-on every time when the password is changed. If set to false (default) the password change at next log-on will not be forced.
SCRIPT_EXECUTION_MECHANISM	The mechanism that will be used to execute scripts on resource. The default WinRM mechanism will execute the script by using WinRM client built into the connector. Local strategy means execution on the local machine where the connector is deployed. Possible values: winrm, local. Default value: winrm