CloudVision agni

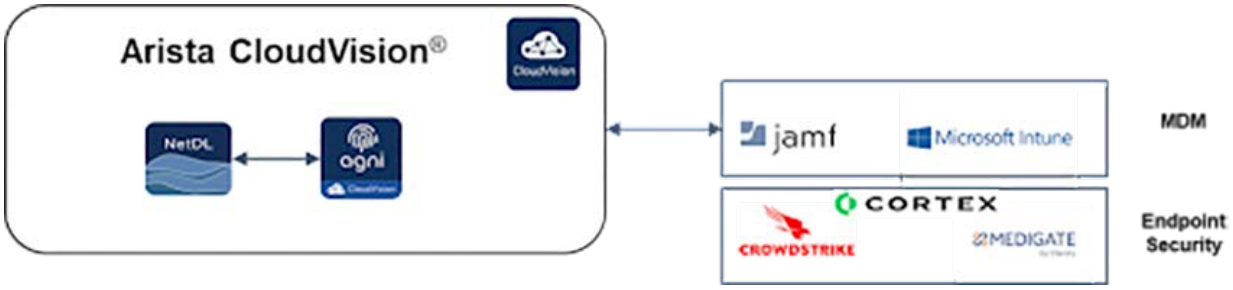
2025.2.0 API Guide

June, 2025 (version: 1.1)

Introduction

This document captures the REST API details of Arista Guardian for Network Identity (AGNI) and provides a detailed understanding of various API options present in AGNI. The URLs, credential information, and user objects mentioned in this document are for illustration purposes only. AGNI follows the RPC API model to integrate with native and third-party applications. Administrators can configure AGNI using the API elements.

AGNI integrates with the application APIs for its interaction with Arista CVaaS, CV-CUE, NDR, and several other third-party applications such as Medigate, CrowdStrike, Palo Alto Cortex XDR, JAMF, Microsoft Intune, and many more.



REST API Standards Followed by Arista Services

The RESTful APIs in AGNI adhere to the following standards:

- Session-based - The API user must initiate a session with the service to make any API calls.
- Supports API token-based authentication - AGNI supports API token-based authentication. Each token can be assigned access privileges from the AGNI user interface.
- Supports the HTTP verbs
    - POST - to create/modify/delete a resource
- Supports application/JSON as response format.

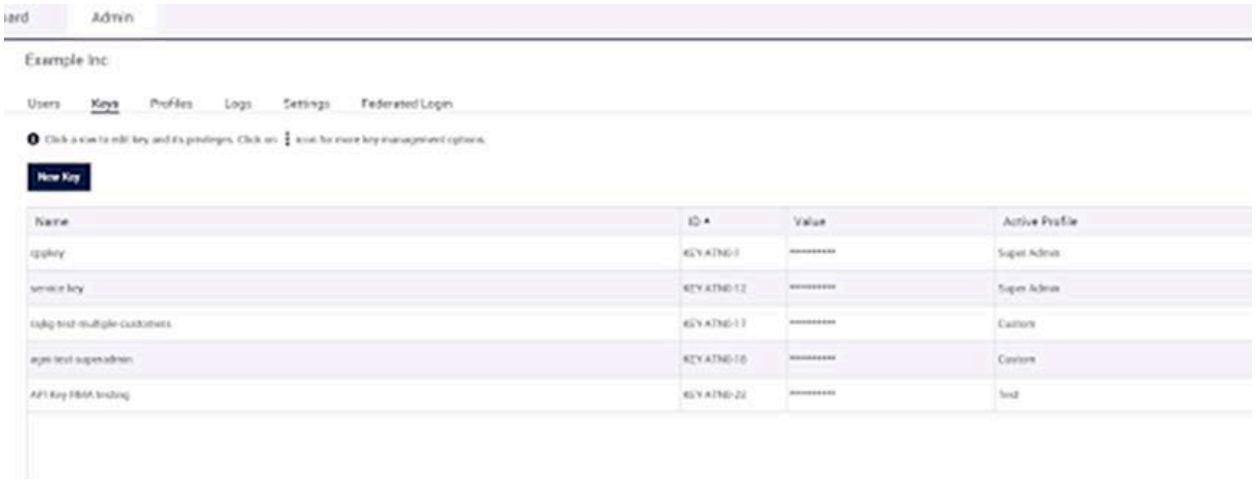
API Token Authentication

A unique API key must be generated in the wireless SSO login system. This API key is used to execute the AGNI APIs from utilities such as Postman and browsers.

Accessing AGNI APIs

To access AGNI APIs, follow one of the following methods:

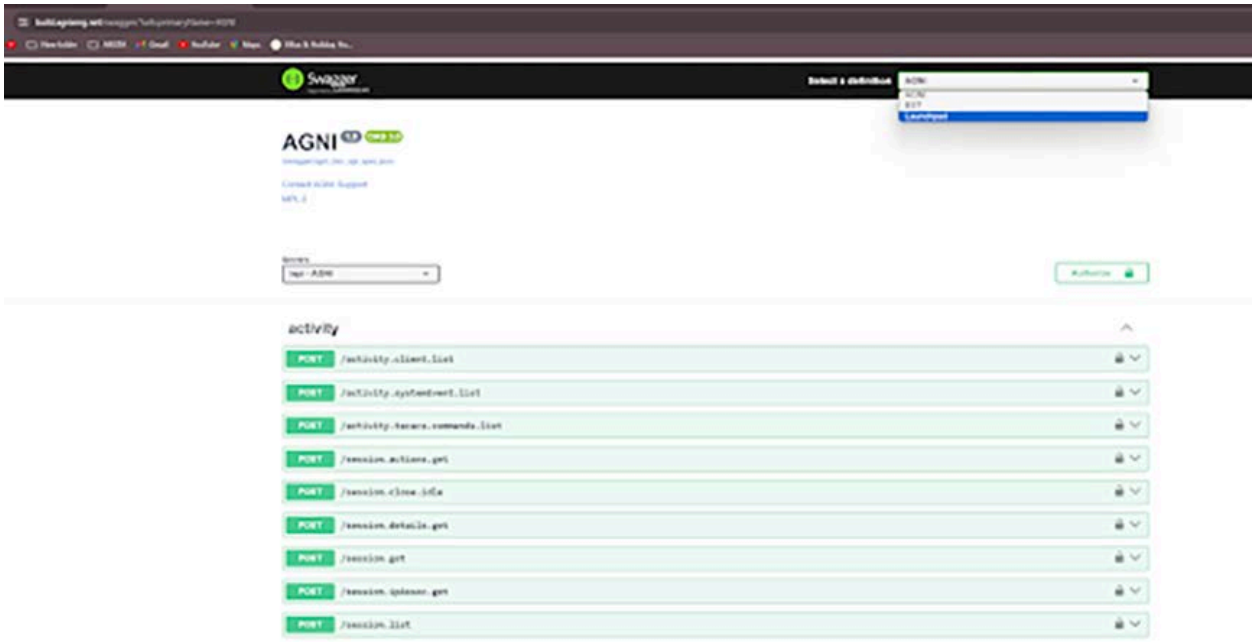
a. Wireless SSO Login
    i. Access Arista AGNI APIs by logging in to CV-CUE with Super-admin & Admin rights. OR
    ii. Access AGNI API portal by using the URL: https://<agni cluster name>/swagger/
b. Launchpad API Keys: Access Wireless launchpad **UI**-> **Admin**-> **Keys**



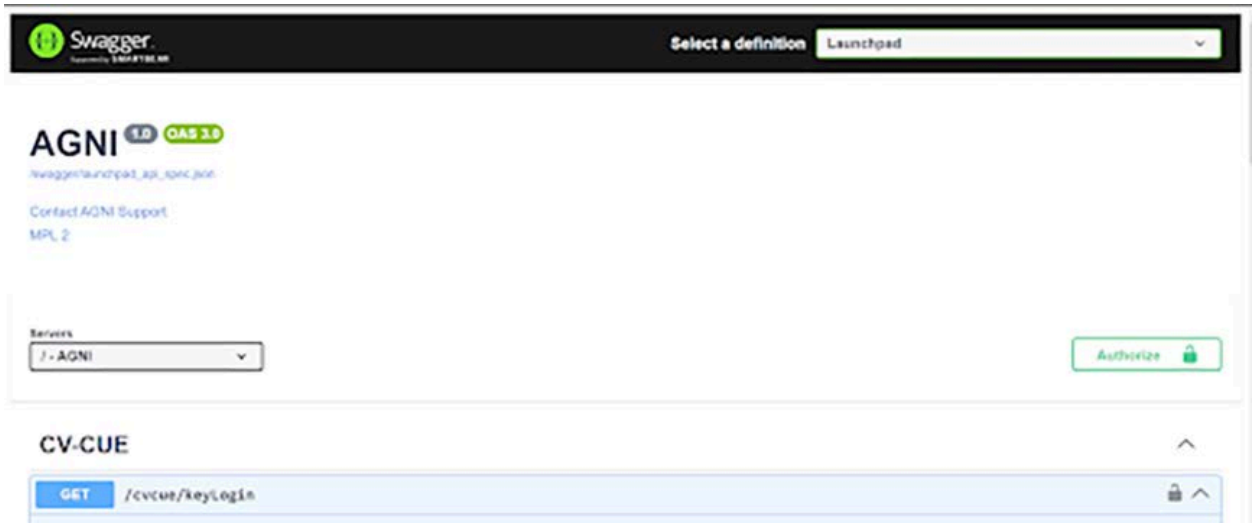
Using these APIs and keys, generate API cookies. These API cookies will be used for system integrations.

Generating the API cookies

1. Login to AGNI using wireless SSO Super-Admin/Admin privileges
2. Access https://<agnifqdn>/swagger/
3. Select Launchpad APIs



4. Select CV_CUE to get the API method.

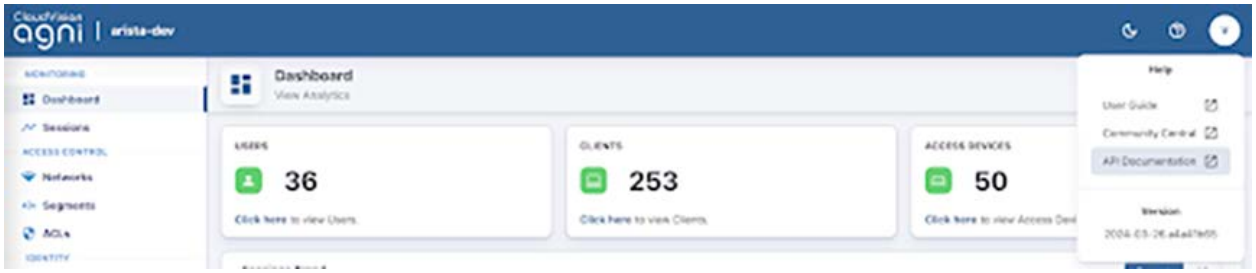


5. Use API Key, Value, and CAS URL (optional) as the input to this API.
6. The AGNI API cookies are generated, which can be used to run API calls using an automated tool or script to access AGNI.

API Modules

All the resources served by AGNI are logically categorised in functionality-based Modules. Each module comprises resources that are functionally related to each other. A resource is part of only one module.

Starting with the release 2024.1.0, the API documentation is available in a new tab from the help section on AGNI portal (see image below).



AGNI Modules

- Dashboard:
  AGNI dashboard uses the following APIs to populate the Users, Clients, APs, and other statistics
  - Stats.get
- Sessions:
  All sessions are enlisted in the sessions section. Admin can get session details by using the following API:
  - Session.list - Used to get the complete list of sessions
  - Session.get: Used to get the auth request of a session from the list
  - Session.details.get: Used to get the session details based on the auth request
  - session.ipAssoc.get: Used to get the session details with IP address as an input
  - Session.close.idle: Used to get the closed sessions from an organization
  - Session.action.get: Used to get the actions of a particular session
  - Activity.client.list: Used to get the client details involved in that session
  - activity.systemEvents.list: Used to get the list of system events
  - Activity.tacacs.command.list: Used to get the list of TACACS+ commands executed for a session
- Network:
  This module covers the APIs related to AGNI networks
  - Config.network.add: Used to add a network in AGNI
  - Config.network.delete: Used to delete the network in AGNI
  - Config.network.get: Used to get the network config
  - Config.network.list: used to get the list of networks configured in an organization

- config.network.onboard.clientGroup.list: Used to get the client groups associated with the network
- config.network.onboard.userGroup.list: Used to get the user groups associated with the network
- config.network.preLoginURL.list: Used to get the list of domains to be configured in walled garden list of Ap in case of captive portal networks
- Config.network.settings.get: Used to get the network settings
- Config.network.settings.update: Used to update the UPSK network parameters
- config.network.sharedClients.add: Used to add shared clients in UPSK networks
- config.network.sharedClients.delete: Used to delete the shared clients from a UPSK network
- config.network.sharedClients.list: Used to get a list of shared clients from a UPSK network
- config.network.update: Used to update the networks in AGNI
- config_upsk_rebuild: Used to regenerate the UPSK

- Segments:
  This module adds, updates, and deletes the policy segments in AGNI. These APIs are also used to get the configured segment policy rules.
  - config.segment.list: Used to get the list of segments configured in the network.
  - config.segment.get: Used to get the segment configurations.
  - config.segment.add: Used to add a new segment rule in AGNI
  - config.segment.update: Used to update segment rule in AGNI
  - config.segment.delete: used to delete the segment rule in AGNI
  - Config.segment.reorder:
  - config.segment.action.attribute.list: Used to get the action attribute list configured in a segment
  - config.segment.action.list: Used to get the actions list from the segment
  - segment.condition.attribute.list: Used to get the conditions list from the segment
  - segment.condition.attribute.values: Used to get the attribute values from segments
  - config.segment.condition.operators.list: used to get the conditions operators list from the segment
  - segment.policy.eval: Use to evaluate the segment policies against a client
  - Segment_policy_reeval: Use to re-evaluate the segment policies against a client
- Access Control List:
  This module enlists the APIs to add, modify, and delete ACLs in AGNI.
  - config.acl.add: Use to add the ACLs in AGNI
  - config.acl.update: use to update the existing ACL in AGNI
  - config.acl.delete: use to delete the existing ACL in AGNI
  - config_acl_list: Use to get the list of ACLs configured in AGNI
  - config.acl.get: Use to get the ACL profile in AGNI
  - config.acl.validate: Use to validate the configured ACL

- Identity Provider
  This module captures the APIs to get the configurations, add, update and delete the identity provider in AGNI
  - config.idp.add: Use to add identity provider in AGNI.
  - config_idp_creds_validate: Use to validate the configured identity provider credentials
  - config_idp_delete: Use to delete the identity provider from AGNI
  - config_idp_get: Use to get the IDP details configured in AGNI
  - config_idp_groups_list: Use to get the user group list from identity provider
  - config.idp.list: Use to get the identity provider list configured in AGNI
  - config_idp_sync: Use to sync user groups and other attributes from the identity provider to AGNI
  - config_idp_syncSettings_attrs_list: Use to get the list of settings of attributes which are synced between an identity provider and AGNI
  - config.idp.syncSettings.attrs.update: Use to update the attributes' settings synced between an identity provider and AGNI
  - config.idp.syncSettings.group.addBulk: Use to add IDP synced user groups in AGNI
  - idp.syncSettings.group.deleteBulk: Use to delete IDP synced user groups in AGNI
  - idp.syncSettings.group.list: Use to get the list of IDP user groups configured in AGNI
  - idp.syncSettings.syncInterval.get: Use to get the sync interval configured in AGNI
  - idp.syncSettings.syncInterval.update: Use to update the sync interval of sync between AGNI and IDP
  - idp.syncStatus.get: Use to get the status sync status between AGNI and IDP
  - config.idp.update: Use to update the identity provider configuration in AGNI
  - config.idp.user.attributes.get: Use to get the idp user attributes
  - config.idp.user.attributes.list: Use to get the list of idp user attributes
  - config.oidc.login.redirectURL.get: Use to get the oidc login redirect url
- User:
  This module lists the APIs to add, update, and delete the users in AGNI
  - identity.user.add: Use to Add a user in AGNI
  - identity.user.update: Use to update the added users in AGNI
  - identity.user.delete: Use to delete the users in AGNI
  - identity.user.list: Use to get the list of users in AGNI
  - identity.user.get: Use to get the user details from AGNI
  - identity.user.clients.delete: Use to delete the client entries associated with that user
  - identity.user.password.update: Use to update the user password
  - identity.users.localUserGroup.addBulk: Use to add user groups to user
  - identity.users.localUserGroup.deleteBulk: Use to delete the user group from the user

- User Group:
  This module gives the APIs that are used to manage user groups in AGNI
  - config.userGroup.add: Use to add a new user group in AGNI
  - config.userGroup.delete: Use to delete a user group from AGNI
  - config.userGroup.get: Use to get the details of a user group from AGNI
  - config.userGroup.list: use to get the list of user groups configured in AGNI
  - config.userGroup.update: Use to update the usergroup in AGNI
- Client:
  This API Module gives the list of APIs that are used to manage the clients.
  - identity.client.add: Use to add new clients in AGNI
  - identity.client.update: Use to update the existing client
  - identity.client.delete: Use to delete the existing client
  - identity.client.list: Use to get the client list from AGNI
  - identity.client.get: Use to get the client details
  - identity.client.attributes.update: Use to update the client attributes
  - identity.client.attributes.delete: Use to delete the client attributes
  - identity.client.coa: Use to send triggered CoA to a particular client
  - identity.client.count: Use to get the client count at a location or assigned to a particular user
  - identity.client.deleteBulk: Use to bulk delete the clients from client listing
  - identity.client.disconnect: Use to disconnect a client using CoA
  - identity.client.profile.get: Use to get the client profile information
  - identity.clients.clientGroup.import: Use to import the client groups
  - identity.client.macVendor.list: Use to get the mac vendor list for client profiling
  - identity.client.profile: Use to profile the clients using given fingerprints
  - identity.client.profile.update: Use to update the client profiles
- Access Devices
  This module gives the APIs to configure the access devices in AGNI
  - config.nad.add: Use to add access devices in AGNI
  - config.nad.count: use to get the access devices count configured
  - config.nad.delete: Use to delete the access device from Access Devices listing
  - config.nad.deleteBulk: Use to perform bulk delete access devices operation
  - config.nad.get: Use to get the access device details
  - config.nad.import: Use to import access devices in AGNI using CSV format file as input
  - config.nad.list: Use to list all access devices from access devices listing
  - config.nad.update: Use to update the access device details
  - config.nadGroup.add: Use to add access devices group
  - config.nadGroup.delete: Use to delete the access device group
  - config.nadGroup.get: Use to get the access device details
  - config.nadGroup.list: Use to get the list of access device groups
  - config.nadGroup.nad.add: Use to add access device in the access device group
  - config.nadGroup.nad.delete: Use to delete the access devices from access devcie groups

- config.nadGroup.nad.list: Use to get the access device list under an access device group
- config.nadGroup.update: Use to update the access device group
- config.acgDevice.add: Use to add arista cloud gateway (ACG) in AGNI
- config.acgDevice.delete: Use to delete the ACG in AGNI
- config.acgDevice.get: Use to get the details of ACG
- config.acgDevice.list: Use to get the list of ACGs
- config.acgDevice.token.generate: Use to generate the ACG token
- config.acgDevice.update: Use to update the ACG details

Radsec Certificates: These APIs to get the RadSec ca certificate

- config.cert.radSec.ca.get: Use to get the RadSec ca certificate
- config.radSec.hostName.get: Use to get the AGNI RadSec hostname

- TACACS+
  These APIs are used to add, delete, and update the TACACS+ directory. These are also used to get the list. These APIs are used to add, delete, and update the TACACS+ profiles. These are also used to get the list
  - config.tacacs.dict.add: Use to add a new TACACS+ dictionary in AGNI
  - config.tacacs.dict.update: Use to update the TACACS+ dictionary
  - config.tacacs.dict.delete: Use to delete the TACACS+ dictionary
  - config.tacacs.dict.get: Use to get the TACACS+ dictionary details
  - config.tacacs.dict.list: Use to get the list of TACACS+ dictionary
  - config.tacacs.profile.add: Use to add a new TACACS+ profile
  - config.tacacs.profile.delete: Use to delete the existing TACACS+ profile
  - config.tacacs.profile.get: Use to get the details of a TACACS+ profile
  - config.tacacs.profile.list: Use to get the list of TACACS+ profiles
  - config.tacacs.profile.update: Use to update the TACACS+ profile
  - config.deviceAdmin.settings.get: Use to get the device administration configurations
  - config.deviceAdmin.settings.update: Use to update the device admin policies
  - config.deviceAdmin.settings.userGroup.list: Use to get the user group list associated with the device admin policy
  - config.deviceadmin.policy.action.attribute.list: Use to get the list of action attributes configured in the device admin policy
  - config.deviceadmin.policy.action.list: Use to get the list of device admin policy actions
  - config.deviceadmin.policy.add: Use to add the device admin policy
  - config.deviceadmin.policy.condition.attribute.list: Use to get the list of condition attributes in the device admin policy
  - config.deviceadmin.policy.condition.attribute.values: Use to get the device admin policy condition attribute value
  - config.deviceadmin.policy.condition.operators.list: Use to get the device admin policy condition operator list
  - config.deviceadmin.policy.delete: Use to delete the existing device admin policy

    - config.deviceadmin.policy.get: Use to get the details of device admin policy
    - config.deviceadmin.policy.list: Use to get the device admin policies list
    - config.deviceadmin.policy.reorder: Use to reorder the device admin policy
    - config.deviceadmin.policy.update: Use to update the device admin policy
- Certificates: These APIs are used to get the server, ca, issuer, RadSec ca-cert, and other certificate-related configurations. Also, used to add and delete third-party CA certificates
    - config.cert.ca.crl.sync: Use to sync the CRL
    - config.cert.ca.crl.upload: Use to sync the CRLfile in AGNI
    - config.cert.ca.crlConfig.update: Use to update the CA cert CRLconfig
    - config.cert.ca.delete: Use to delete the CA cert
    - config.cert.ca.get: Use to get the CA cert
    - config.cert.ca.list: Use to get the CA certs list
    - config.cert.ca.upload: Use to upload the ca-cert
    - config.cert.eap.client.enroll: Use to get EAP client certificate
    - config.cert.radSec.server.get: Use to get the RadSec server certificate
    - config.cert.radsec.client.enroll: Use to get RadSec certificate for clients
    - config.issuerCACert.get: Use to get the issuer certificate
    - config.radSecCert.get: Use to get the RadSec certificate
    - config.rootCACert.get: Use to get the root certificate
- Captive Portal
  The portal APIs are used to add, modify and delete the portals in AGNI
    - config.portal.add: Use to add a new captive portal
    - config.portal.delete: Use to delete the AGNI captive portal
    - config.portal.get: Use to get the portal details
    - config.portal.list: use to get the list of captive portals
- Guest Client
  The guest client APIs are used to add the guest user
    - Identity.guest.user.add: Used to add a new guest user
    - Identity.guest.user.addBulk: Used to add a batch of guest users
    - Identity.guest.user.count: Used to get the guest user count
    - Identity.guest.user.clients.list: Used to get the list of clients assigned to a guest user
    - Identity.guest.user.delete: Used to delete a guest user
    - identity.guest.user.deleteBulk: Used to delete a batch of guest users
    - Identity.guest.user.export: Used to export the users from AGNI
    - Identity.guest.user.list: Used to get the guest user list
    - Identity.guest.user.get: Used to get the details of a particular guest user
    - Identity.guest.users.import: Used to import the guest users
    - Identity.guest.user.update: Used to update the guest user parameters
    - identity.guest.user.updateBulk: Used to update the batch of guest users
- Guest User Batch
  This API allows the admin to add, modify, and delete the guest user batch
    - config.guestBatch.add: Used to add the guest user batch

    - config.guestBatch.delete: Used to delete the existing guest user batch
    - config.guestBatch.get: Used to get the guest batch details
    - config.guestBatch.list: Used to get the list of guest user batches
    - config.guestBatch.update: Used to update existing guest batches
- Audit Logs
    - config.audit.list: Use to get the audit logs from AGNI
- AGNI Service:
  These APIs are used to get the AGNI system logs, session logs, and email sent the status
    - email.send: Use to send the emails to user
    - service.log: Use to get the service logs
    - session.log: Use to get the session logs
- Organisation Details:
  This module is used to get the IDP organisation details
    - org.info: Use to get the IDP details configured in AGNI
    - org.update: Use to update the IDP details
- Concourse Apps: All third-party plugins get, verify & update APIs. The concourse API includes
    - Arista NDR
      These APIs are used to get the Arista NDR notifications, NDR settings, and generate NDR tokens.
    - Crowdstrike
      These APIs are used to generate Crowdstrike secrets, get the settings using these secrets, and get the updated settings from Crowdstrike
    - Arista CV-CUE
      These API Calls are used to get the guest portal list in Guest Manager, CV_CUE settings, and get the updated settings from CV-CUE
    - Intune
      These APIs are used to get the Intune settings, updates, and verify the settings from Intune (this is used in the compliance check feature)
    - Jamf
      These APIs are used to generate the Jamf Scep URL, get settings, and updates
    - Workspace
      These APIs are used to get the SCEP profile URL, get settings, and updates
    - other plugins

- License Usage
  These APIs are used to get the existing license usage

    - org.licenseUsage.get: Used to get the license usage information
    - org.licenseUsage.list: Used to get the past license usage information list
    - org.licenseUsage.compute: Used to compute the license usage