Direct Connect API Documentation

Introduction

The Juniper Licensing APIs should be a well-defined set of REST APIs that enable Clients to integrate their Network Applications and Cloud Platforms with Juniper Entitlement Management system. As part of this B2B integration Clients should be able to:

- Systematically validate an entitlement by providing an activation code or software support reference number (SSRN).
- Query for associated license/voucher entitlements by providing a valid customerID and groupID.
- Systematically activate an Entitlement and generate a License Key; by supplying a device serial number/software support reference number (SSRN) from their entitlement group.
- Systematically generate a voucher; by providing a device serial number/software support reference number (SSRN), expiration date and a pinned domain certificate.
- Systematically revoke the license key or vouchers from a device.

Clients integrating their Orchestration Applications and Cloud Platforms with Juniper APIs should be able to automate the license/voucher generation.

Required headers in each API request sent

The following are the required headers to be sent in each API call.

Header	**Value**
Authorization	OAuth2 Bearer Token
Accept	Application/json
Content-Type	Application/json

Client OAuth2.0-based Onboarding Process

This table lists the high-level steps for setting up OAuth2.0 authentication and invoking the Juniper Service APIs.

Action	Who?
Review the Juniper Service APIs license/Terms of Use.	Customer/Partner
Open the firewall ports	Customer/Partner and Juniper

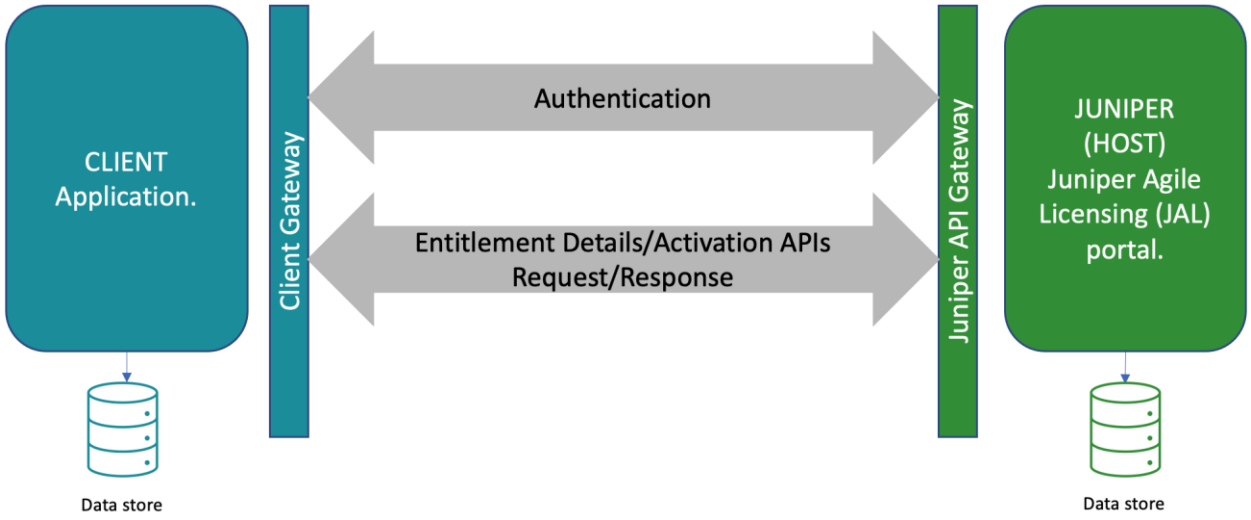
Generate OAuth2.0 credentials (clientID, secret) and share with Customer/Partner.	Juniper
Create a Customer/Partner profile in the Juniper API gateway with OAuth2.0 as the authentication mechanism.	Juniper
Create APIs and assign the Customer/Partner profile.	Juniper
Share the API endpoints along with token endpoint with the Customer/Partner.	Juniper
Validate that the Customer/Partner can invoke the APIs by passing the Access token (which is obtained by request to token endpoint) in the Authorization header.	Customer/Partner
Establish the Customer Source Identifier.	Juniper and Customer/Partner collaborate
Specify to Juniper an email aliases that can be used to send an email when Juniper is NOT able to connect to the endpoint.	Customer/Partner
Specify to Juniper an email aliases that is to be used for providing Juniper to provide new secret that would need to be used when the time to rotate the secret arrives (once a year).	Customer/Partner

High-level API Model

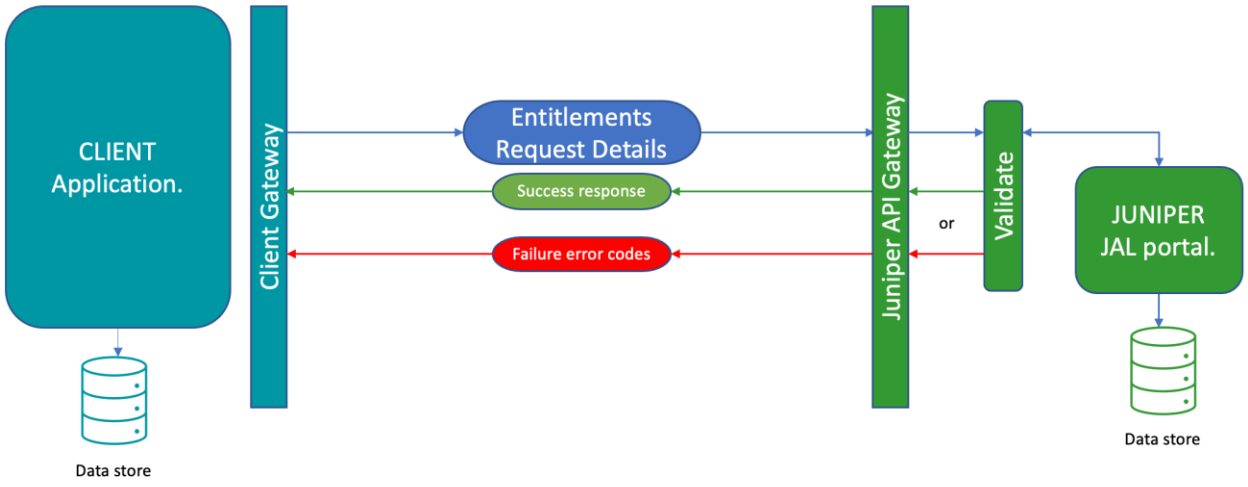
The following illustrates a high-level view of the Licensing API framework. This framework allows you to integrate your network or cloud solution system with Juniper's Support CRM system. This is a B2B integration. Two authentication mechanisms are supported. An authentication phase followed by a request/response is the typical flow.

The APIs are RESTful, the data format is JSON, and the transport is HTTPS (TLS 1.2+).

LICENSING API MODEL



ENTITLEMENT DETAILS API MODEL

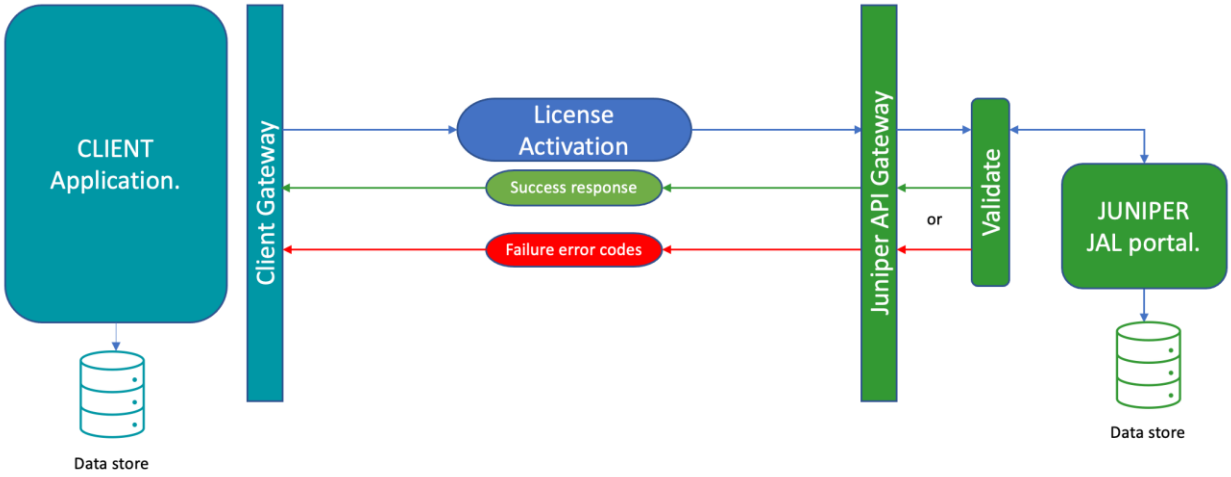


Rate Limit

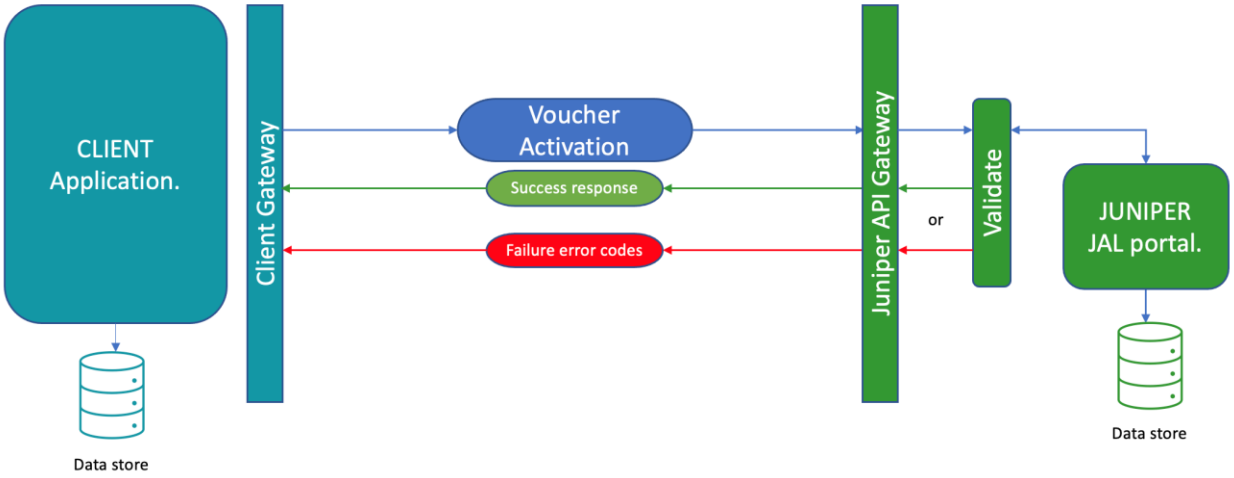
- Rate Limit is defined as the maximum number of API requests that can be sent per hour.
- Each client will be allowed a maximum of 300 requests/hour.

- Upon receipt of a request the rate of requests is calculated and if it exceeds 300 requests/hour the request will not be processed, and a rate limit exceeded error is sent in response.

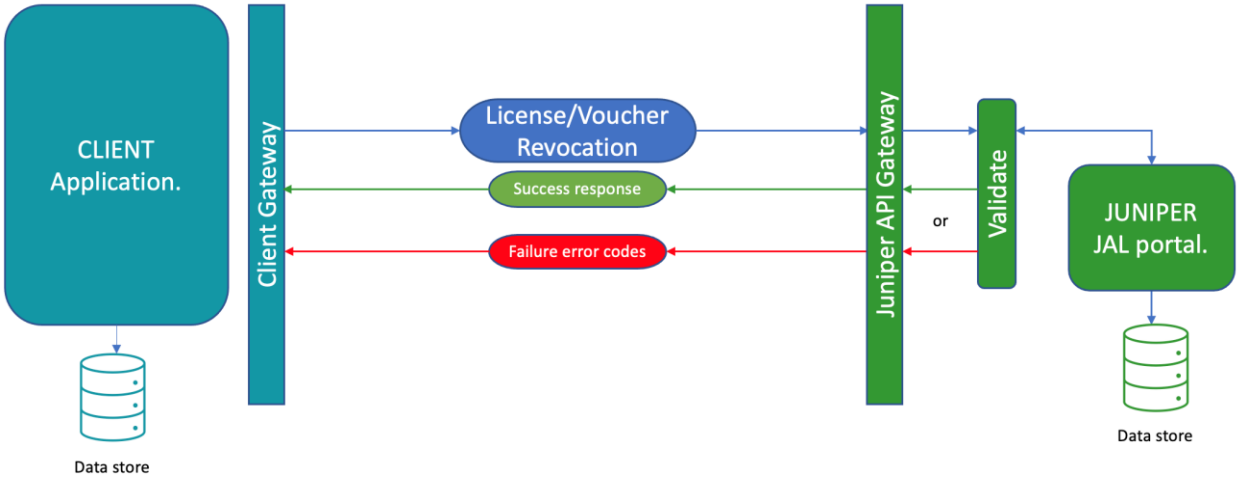
ACTIVATION API MODEL



VOUCHER ACTIVATION API MODEL



REVOCATION API MODEL



Supported API Authentication Mechanisms

Clients can choose from three authentication mechanisms for connecting to Juniper: OpenID Connect (OIDC), OAuth2.0 and Certificate based. All services are RESTful and invoked over HTTPS.

OIDC (OpenID Connect) Authentication Mechanism

For Clients who have already deployed OIDC within their organization, OIDC would be the preferred choice of authentication mechanism to connect to Juniper Service APIs. Clients who choose this authentication mechanism need to provide Juniper an ID token as part of the Onboarding process.

A good reference describing OpenID Connect can be found at https://connect2id.com/learn/openid-connect

OAuth2.0 Authentication Mechanism

Juniper Service APIs support the popular OAuth2.0 Authorization protocol. Juniper provides a clientid, secret and a token endpoint. Token endpoint is used to obtain an access token by presenting clientid, secret.

Each API request is made with header containing the access token information. Juniper will rotate the secret yearly to enforce enhanced security.

A good reference describing OAuth2.0 can be found at https://auth0.com/docs/protocols/oauth2

Certificate-based Authentication Mechanism

Juniper Service APIs support Mutual SSL authentication or certificate based mutual authentication. Clients choosing this method for authentication need to create a X509 public/private key pair and share the CA signed public certificate with Juniper. Certificate is to be presented during the SSL handshake on each API request. Client needs to provide Juniper with a new CA signed public certificate prior to expiry of the current certificate.

A good reference describing Certificate based (aka MTLS - Mutual SSL authentication) can be found https://www.codeproject.com/Articles/326574/An-Introduction-to-Mutual-SSL-Authentication

Authentication and Authorization Aspects on API Requests

The request follows these authentication steps:

1. Trusted Channel: Received with valid encrypted signature over HTTPS (TLS1.2+).

2. Rate Control: The Client must be within the rate limits.

The request follows these authorization steps:

1. API Key (sapbpID/contactID/customerGroupID): is a valid API Key in the Request header as established during the onboarding process.

2. Account (accountIds): Valid account identifiers as established during the onboarding process.

3. Contact Email (userEmail): is a valid registered user identifier associated with the account.

Json Key Details

Terminology

Basic terminology used for integrating with licensing APIs.

Term	Functional Description
API	Application Programming Interface. A set of protocols used by programmers to create applications for a specific operating system or to interface between the different modules of an application.
Client	Customer or Partner integrating their Cloud Orchestration application with Juniper's EMS system using this API channel.
JAL Portal	Juniper Agile Licensing portal.
Host	Juniper JAL/EMS back-office system.
REST API	API based on Representational State Transfer using HTTP/HTTPS.
Activation Code	Secure alpha numeric token used to uniquely identify each customer licensing entitlement.
SSRN	Software Support Reference Number used to uniquely identify each customer owned software product/entitlement. Note: All software licensing Activation Codes are associated to a unique SSRN.
Device Serial Number	Serial number to uniquely identify the customer device.
Entitlement	A Record describing what an end customer has a right to (Product, quantity, dates. Etc.) and the conditions under which they bought that right. Entitlements can be of several types of Subscription; Perpetual and Voucher Entitlements.
License	• The legal right to use a product. NOTE: The term 'license' may be used interchangeably in this document when referring to such things as an activated entitlement.
License File	The key, file, etc. that results from an activation and enables the application based on what is activated from an entitlement.
Product Key	An encrypted, unique, alpha-numeric string of characters used by a customer to enable the purchased product features and capabilities. The product key can be entered into the product itself which will initiate a call to the EMS or within a web portal

	to retrieve the license file that will be later installed on the product. (Assumes a Connected scenario is being used.)

License Compliance	• Devices/applications are in compliance when a valid license is present for all features and capacities being used.
	• Devices/applications are out of compliance in the following cases:
	  ○ A feature is being used that does not have a valid license present.
	  ○ A capacity surpasses that enabled by a valid license.