Australian Government
Services Australia

Provider Digital Access

(PRODA)

Unattended B2B Software Vendor Developer's Guide

Version: 3.7 - 2020.11

Table of Contents

Table of Figures

Table of Tables

1 Introduction

1.1 Purpose

The Provider Digital Access (PRODA) Business to Business (B2B) unattended pattern provides an authentication and authorisation framework that can be used by organisations to submit authenticated and authorised unattended web service transactions to a range of Service Providers.

PRODA's B2B authorisation pattern leverages the OAuth 2.0 framework. This document provides a high-level overview of the framework as well as the service interface specification for the PRODA B2B Application Programming Interface (API) that must be used by software requiring authorisation to access B2B services on behalf of organisations.

1.2 Intended Audience

All organisational and 3rd party Vendor Software developers requiring B2B authentication and authorisation services from PRODA to access Service Provider APIs protected by the PRODA authorisation pattern.

1.3 Scope

The scope of this document includes:

- Overview of technologies used in the PRODA B2B authorisation pattern
- Software Instance interactions with the framework, including
  - Software Instance activation,
  - Obtaining an Access Token for authorisation purposes,
  - Refreshing a Software Instance's RSA key pair (used to sign access token requests)

1.4 Glossary

The following acronyms and terms are in this document:

Acronym/Term	Description
API	Application Programming Interface.
Authorised Provider	A person who has successfully registered, and is associated with an organisation within PRODA, through a director/associated role with the Australian Business Register (ABR).
B2B	Business to Business.
Bearer Token	An OAuth 2.0 token included in the HTTP header [IETF RFC 6750], providing proof of authorisation to use a Service Provider's services.
DAC	Device Activation Code.
Device	A device is a term to describe a physical server, a virtual server or even a desk top computer, hosting software that utilises the PRODA authorisation pattern. This term is being used interchangeably with Software Instance as the instance of software resides on the device.
Device Name	The name of the device (or software instance) registered in PRODA by an organisation during B2B device registration. It uniquely identifies a software instance associated with an organisation.
JSON	JavaScript Object Notation is a language-independent data format that contains human-readable data objects consisting of attribute-value pairs and array data types.
JWK	JSON Web Key [IETF RFC 7517]. A JSON formatted cryptographic key.
JWS	JSON Web Signature [IETF RFC 7515]. A JWT formatted digital signature, used to authenticate a client.
JWT	JSON Web Token [IETF RFC 7519].
TTL	Time To Live. A term to describe a time value in which an action needs to be completed by.
OAuth 2.0	OAuth 2.0 authorisation framework [IETF RFC 6749]. Enables 3rd party applications access to an HTTP service, either on behalf of a resource owner by orchestrating an approval interaction between the resource owner and the HTTP service, or by allowing the third-party application to obtain access on its own behalf.
PRODA	Provider Digital Access application.
REST	Representational State Transfer (REST) web services allows requesting systems to access and manipulate textual representations through stateless operations.

Acronym/Term	Description
Relying Party	A Service Provider that relies upon PRODA for authentication of clients.
RSA Cryptography	RSA (Rivest–Shamir–Adleman) is a public-key cryptosystem that is widely used for secure data transmission.
Service Provider	An entity that relies upon PRODA for authentication of clients to use their API services.
Software Instance	An instance of software that has been installed on a device, server or virtual server (cloud). It is considered to be a unique entity.
Software Vendor	A third-party entity that creates or distributes the software used to communicate with a Service Provider, utilising PRODA for authorisation.
Vendor Software	Software developed by a third-party to transact with Service Providers.

2 Concepts and System Context

This B2B framework supports a self-service approach which allows organisations to register and manage their software instances (i.e. devices) via the PRODA web application.

The B2B framework facilitates authentication and authorisation of software instances to use a Service Provider's services on behalf of the client organisation.

2.1 System Context

System context diagram provides an overview of the business scenarios that PRODA provide for the B2B Unattended solution.

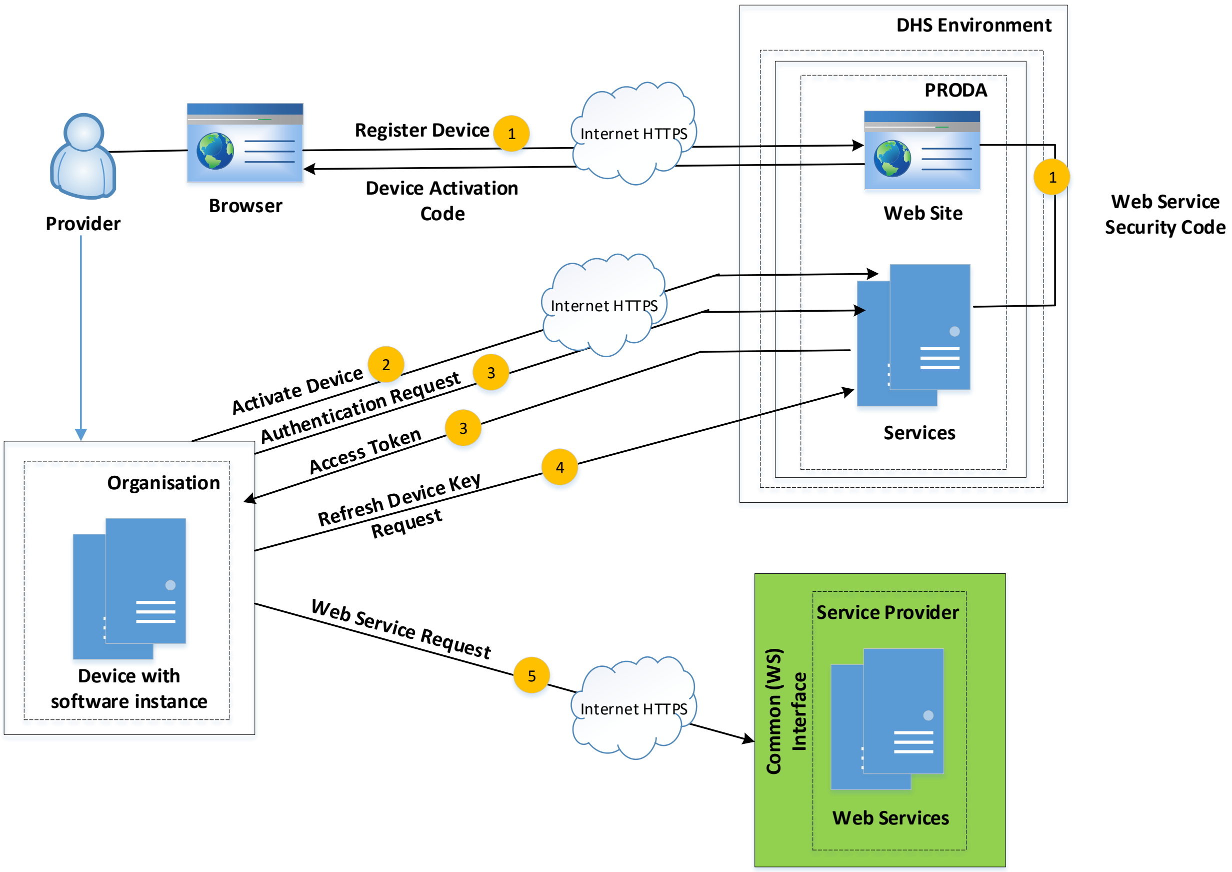


Figure 1 - System context diagram

The below provides a description of the business scenarios described in Figure 1 above.

Table 1 – Business Scenarios

ID	Scenario	Actor	Description
1	Register Device	PRODA User	The authorised PRODA user registers the B2B software instance for their organisation within the PRODA web application. Once successfully registered in PRODA, a unique Device Activation Code (DAC) is issued to the user. This DAC is required by the Vendor Software to activate the software instance within PRODA. **Note:** The DAC has an expiry date[1]. If the DAC issued during the software instance registration expires before the software instance was activated, a new one can be requested within the PRODA application.
2	Activate Device	Vendor Software	The Vendor Software instance submits software instance details to PRODA, including the DAC obtained from (1) and the associated RSA public key, to make the software instance active in PRODA. The Vendor Software must generate an RSA key pair, passing to PRODA the public component of the key as a JWK for later use.
3	Submit Authorisation Request	Vendor Software	The software instance submits an authentication request to PRODA in preparation for a Web Service request to a Service Provider. If validated, an Access Token is returned to the software instance.
4	Refresh Device Public Key	Vendor Software	The software instance generates a new RSA key-pair and uploads the public key as a JWK to PRODA. ❗ **Important Information:** The Vendor Software must regularly refresh the public key within the time parameters[1]. If the public key is not refreshed within this timeframe, the software instance will be unable to gain authorisation from PRODA to access Service Provider services.
5	Submit Request	Vendor Software	The Vendor Software instance submits an unattended B2B request to a PRODA Service Provider using the Access Token from a Submit Authorisation Request. The Service Provider can use this Access Token to verify that the organisation's software has successfully authenticated to PRODA to obtain authorisation to use the Service Provider's services. **Note:** This is not PRODA functionality; however, it has been included for completeness.

[1] Time parameters are described as 'Time to Live (TTL)'. Current values can be found here

2.2 Key Concepts

This section describes the processes at a technical level, noting the majority of process flows do not have user interaction.

The key characteristics of the PRODA B2B authentication framework are:

- PRODA provides the authorisation framework for Service Providers based on OAuth 2.0 standards.

- The organisation and its associated software instance must be registered within PRODA. A device activation code will be provided by PRODA during software instance registration. This device activation code can be used once, and must be provided together with a public key in the software instance activation process. If the activation code expires before the software instance was activated, a new one must be requested in the PRODA application.

- Signing of Authorisation Requests for unattended B2B scenarios is via an RSA key pair, generated and managed by the Vendor Software. The Vendor Software instance holds the RSA key, and provides the public component of the key to PRODA during software instance activation and the key refresh process. The public key is stored in PRODA against the organisation's software instance definition.

- The software instance signs unattended B2B authorisation requests with its private key. PRODA will verify the signature with the public key provided during software instance activation or key refresh, ensuring that the authorisation request was sent from the software instance associated with the RSA key pair.

- PRODA will issue an Access Token in response to a successful authorisation request,

- The Access Token can be used to request access to the Service Provider's services.

- The Access Token can be used to update the software instance's public key in PRODA before the key expiration date,

2.3 OAuth 2.0

The B2B authentication pattern uses the OAuth 2.0 JWT Bearer Token authorisation grant flow which allows a software instance (device) to request an Access Token from the PRODA authorisation service without sharing security information across security domains. This pattern secures web service calls without transmitting credentials.

PRODA will issue a signed OAuth2.0 Access Token, which the client software can use to request access to the Service Provider's services.

2.4 Actors and Roles

The below provides an overview of PRODA actors and their roles.

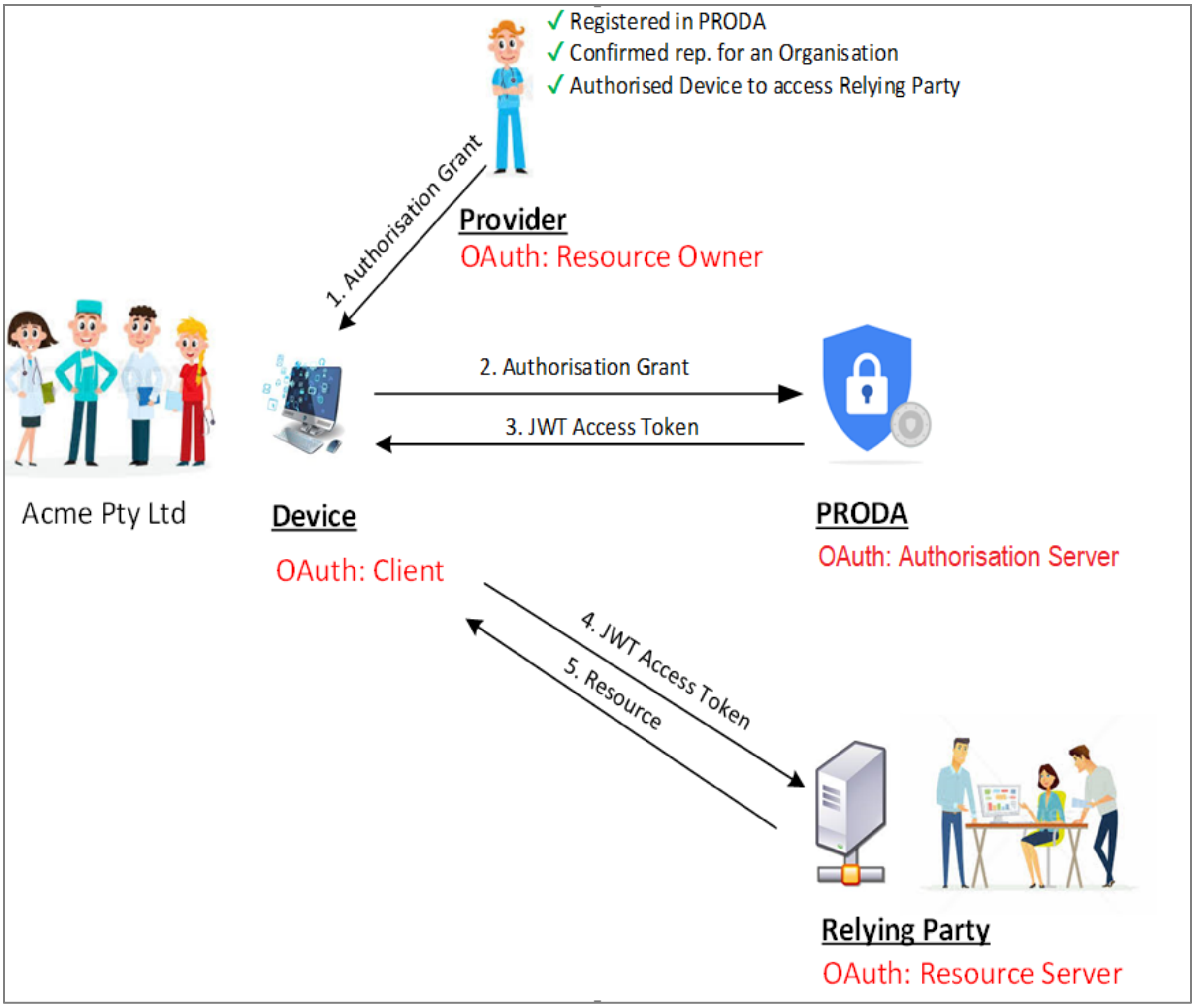


Figure 2 - OAuth 2.0 in action

The below table describes the interaction between the Actor and the OAuth 2.0 role illustrated in Figure 2.

Actor	OAuth Role	Description
PRODA	OAuth Authorisation Server	The Provider Digital Access system that authenticates and manages identity assertions for Providers.
Service Provider	Resource Server	The Service Provider.
Device	OAuth Client	A software instance which is used in the unattended B2B process and is an entity that is related to a public key. The software instance requires PRODA authorisation to submit/obtain information from a Service Provider on behalf of an Organisation.
Provider	Resource Owner	Provider is an individual person who is registered in PRODA and is associated with an organisation. The Provider grants authorisation to the device to access the Service Provider on their behalf.

Table 2 - Actors and Roles

3 Security Interactions

3.1 Overview

This section explains PRODA security interactions. The table below provides more detail regarding these interactions.

Operation Attribute	Attribute Value
Name	PRODA OAUTH2
Service Provider	PRODA
Service Consumers	Vendor Software
Authentication Model	OAuth2.0 (https://tools.ietf.org/html/rfc6749) and bearer token (https://tools.ietf.org/html/rfc6750) **Note:** unless otherwise specified, all OAuth2.0 and Bearer Token security practices are implemented in full as part of the PRODA API interactions.
Transport Protocol Type	HTTPS
Operations	• Activate software instance • Authenticate software instance • Refresh software instance public key
Pre-requisites	An authorised officer in the organisation must register the software instance for an Organisation in PRODA. Prior to software instance activation in PRODA, the following information is required: 1. Organisation Id 2. Device Name (not case sensitive) 3. Device Activation Code

Table 3 - Operation Attributes and Values

3.2 Runtime Application Interactions – High Level Process

Registered organisations may use Vendor Software to access PRODA on-boarded Service Provider APIs. The authorised person for the organisation must be registered as an individual in PRODA, and be associated with an organisation, and have appropriate access to register a software instance. In order to interact with a Service Provider, the following *runtime* process is used:

1. An authorised person for the organisation will register a software instance name in PRODA for the organisation. PRODA will return a device activation code (DAC), as well as the organisation identifier and registered software instance name. [refer section on Register B2B device]
2. The Vendor Software must generate an RSA key pair for the software instance, storing the private key securely. [refer section on Key pair generation]
3. The Vendor Software must activate the device in PRODA, prior to any B2B transactions with Service Providers. The Vendor software instance will submit an activation request to PRODA, passing the device activation code acquired in step 1, PRODA organisation id, software instance name and the public key from Step 2. [refer section on Activate device service]
4. The Vendor Software instance will send a signed B2B authorisation request to PRODA, including the client ID[2] of the software, PRODA organisation ID and software instance name. An Access Token is returned for a successfully valid request. [refer section on Authorisation service]
5. The Vendor Software instance can use the Access Token to interact with the Service Provider until the token expires; when it expires, the Vendor Software instance can obtain a new Access Token by restarting the process again at step 4.
6. The Vendor Software instance can refresh the public key in PRODA by submitting a refresh software instance key request, passing in a valid Access Token, and the new public key. **Note:** PRODA will enforce public key expiry. [refer section on Refresh device key service]
7. If the public key has expired, the Authorised Person must request a new device activation code via the PRODA web application. The Vendor Software instance must then submit a software instance activation request to upload a new public key. [refer section on Activate device service]

[2] The Client Id is provided to the Software Vendor during on boarding. It is an identifier that confirms the software can transact with PRODA authentication services.

4 Device Interactions

4.1 Register

The table below describes the registration process for B2B software instances. Registering the B2B software instance is required so the software instance can activate itself, and then authenticate to PRODA to gain authorisation to use Service Provider services. The software instance is an entity, or credential within PRODA which is associated with an organisation. An organisation's authorised officer is required to manually register the software instance in the PRODA web application.

Table 4 – B2B Device Registration

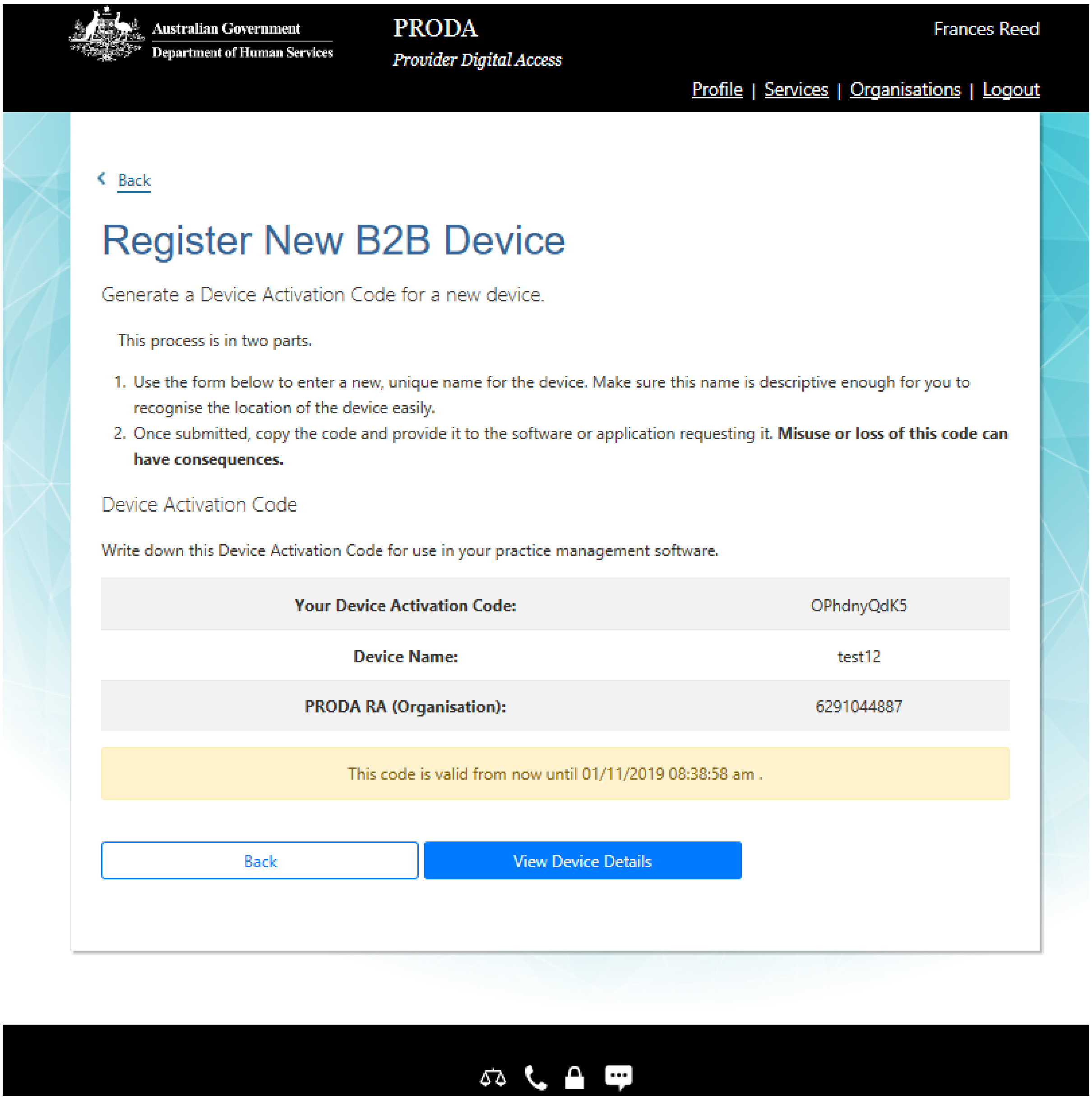
Name	B2B Device Registration
Goal	The organisation's authorised person successfully registers a software instance for the organisation.
Primary Actor	A member of the organisation with 'Device Management' access for the organisation in PRODA.
Precondition	The authorised person has successfully registered and logged onto PRODA as an individual. The associated organisation has been successfully registered in PRODA. An authorised person has linked the organisation to the relevant Service Provider the organisation does business with. The authorised person has been given access to manage software instances (devices) for the organisation.
Steps	1. The authorised person selects the organisation; 2. The authorised person selects the option to Register New B2B Device for the selected organisation; 3. The authorised person enters the device (software instance) name and description then submits using the Register Device button. Note that the device name must be unique within the organisation; 4. PRODA generates a 'Device Activation Code' (DAC) and displays it on the screen; 5. The authorised person copies the DAC, organisation identifier and device name from the screen;

	6. The authorised person supplies the DAC, organisation identifier and device name to the software instance. Note: the vendor may have their own process to achieve this, PRODA does not prescribe how this is done.
Post condition	The organisation has registered a software instance (device) in PRODA. The organisation software instance has acquired the DAC, organisation identifier (PRODA organisation RA), and the software instance name.

Table 4 - B2B Device Registration

4.1.1 Interactions

The below illustrates the screen within PRODA that provides the DAC and device name.


Australian Government
Department of Human Services
PRODA
Provider Digital Access
Frances Reed
Profile | Services | Organisations | Logout
Back

Register New B2B Device

Generate a Device Activation Code for a new device.

This process is in two parts.

1. Use the form below to enter a new, unique name for the device. Make sure this name is descriptive enough for you to recognise the location of the device easily.
2. Once submitted, copy the code and provide it to the software or application requesting it. **Misuse or loss of this code can have consequences.**

Device Activation Code

Write down this Device Activation Code for use in your practice management software.

Your Device Activation Code:	OPhdnyQdK5
Device Name:	test12
PRODA RA (Organisation):	6291044887

This code is valid from now until 01/11/2019 08:38:58 am .

Figure 3 - Screen illustrating device activation code

The table below describes the steps to register a device within the PRODA application.

#	Step
1	Login to PRODA online application
2	Select 'Organisations'
3	Select the specific organisation requiring the new device definition
4	Select 'Register New B2B Device' in the B2B Devices drop down tab
5	Populate software instance details into Register New B2B Device screen and select 'Register Device' button
6	Obtain the 'Device activation code'
7	Enter Device Activation Code, software instance name and organisation id into the Vendor Software.
8	Software instance submits an activation request to PRODA.

Table 5 - Register new B2B device

4.1.2 Device activation code

A device activation code is issued to the organisation's authorised person during Device Registration, or can be regenerated via the 'Manage Device' screen. The device activation code will be displayed on the browser.

The code is for 'one time use' only, with a validity period of **7** days (default)[1]. If the device is not activated within this time period, the authorised person is required to obtain a new activation code.

4.1.3 Device Expiry

Upon successful activation, software instances are associated with a valid time period, after which software instances are disabled by PRODA[1]. The PRODA software instance authentication services return the device expiry date in the response body. This should be used by the Vendor Software to notify owner(s) that their software instance is about to expire. The software instance's active time period can be reset at any time by using the software instance reactivation button within the PRODA web site's organisation B2B device management screen.

4.1.4 Key Pair Generation

The Vendor Software is responsible for generating an RSA key pair, and providing PRODA with the public component of the key during the device activation or key refresh process.

The software instance will need to securely store the generated RSA key pair, as it is the credential used to authenticate to PRODA on behalf of the organisation. Only the public component of the key is transmitted to PRODA. The public key stored by PRODA is used for verification of authentication requests sent from the software instance responsible for management of the RSA key pair.

The public keys are associated with time parameters[1] which defines how long a public key can be used for verification of the authentication requests. This means that a software instance will need to periodically generate new RSA key pairs and publish the new public component of the key to PRODA. If a public key is to expire before it is refreshed, the software instance will need to be reactivated using the device activation process which requires manual intervention by the software instance owner.

The public key submitted to PRODA must be in the JSON Web Key (JWK) format. This is a JSON data structure that represents a cryptographic key. The JWK must be provided in the 'Device Activation' and 'Refresh Device Key' service requests.

The key provided must be an RSA public key with the following properties:

- Key Size: 2048 (minimum size);
- Key Use: Signature;
- Algorithm RS256 (RS384 and RS512 are also accepted); and
- Key Id: Is the name of the software instance with which the key is associated.

The below details the attributes required for the public key JWK.

Member Name	Value
kty	RSA
alg	[RS256 | RS384 | RS512]
use	sig (signature)
kid	<software instance name>
n	<public key>
e	AQAB

Table 6 - Key Pair Generation

An example of a JWK follows:

```
{
    "kty": "RSA",
    "kid": "my-device-01",
    "use": "sig",
    "alg": "RS256",
    "n": "k6dcKJZvFCGxx-
MQNbrTdJMMBviHbNsp8bHjkaNty7IPajjcFDVZj2yFBfOpuDV6sJfnhgVFB-
y1jYXbZUUCqejegWtlFfmSl9_hpNNZaJEBhhVMVppKFnpxrZlYW-
wteONFi_4swx4YAxt_i7zRaUVKZWJVsHidCk4Q6aLBO7RcQso3gIxi2lSY4tPipmPIcAdrU9L8mPCNWJZ5
Y37pszzozj3yk3ysfXxXGV3S3_EvafHNmclaWSpthA0JFBzs38K1k25LuL1JqGVOpyNhWyegNReGIJv_n0
XuoceI4f4BR-H_aj3RlaeceOtv86uMUBhWs47sRAm0mGBuFAFo3Q",
    "e": "AQAB"
}
```

4.2 Device Activation

Activating a software instance involves uploading a public key and storing it against a registered device in PRODA. The new public key must not match previous public keys associated with the software instance. The below table details further information regarding software instance activation.

Name	**B2B Device Activation**
Goal	Software instance will successfully activate in PRODA, with PRODA obtaining the software instance’s public key.
Primary Actor	This will be initiated by the authorised person through the Vendor’s software on behalf of the organisation.
Precondition	The software instance is registered in PRODA. The software instance has been supplied with the one-time device activation code, PRODA organisation id and software instance name. The software instance generated a new and unique RSA key pair.
Steps	1. The software instance submits a B2B API request to PRODA for device activation. The request must include the device activation code, organisation id, software instance name, and public key. 2. PRODA will validate the device activation code. 3. PRODA will store the public key and activate the software instance.
Post condition	The organisation has a software instance registered and active in PRODA PRODA contains an active public key for the registered software instance.

Table 7 - Device Activation

4.2.1 Interactions

The figure below provides a logical representation of the interaction undertaken to activate a software instance.

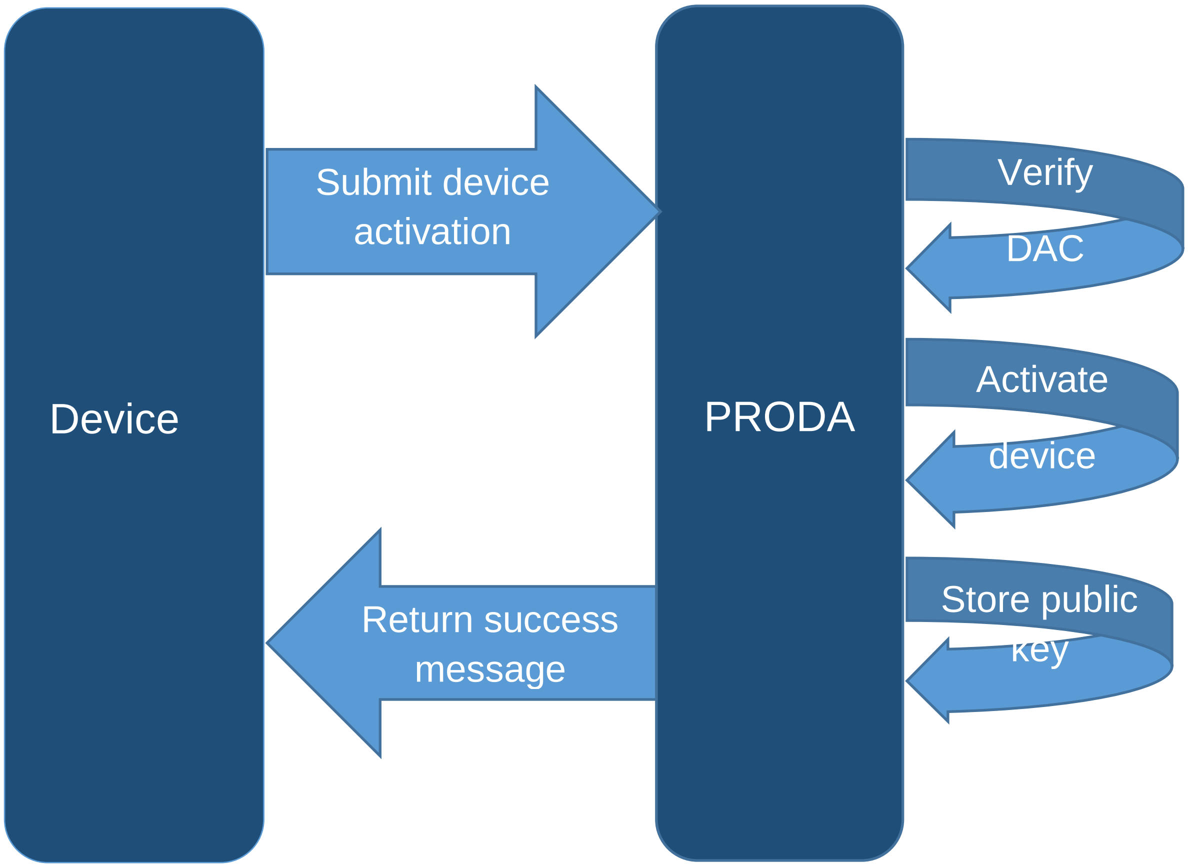


Figure 4 - Activate device service

The table below explains the above steps in detail.

Name	Description
1. Submit device activation request	The software instance prepares the request for software instance activation. The request parameters include the software instance name. The request parameters are conveyed in the body of an HTTPS request. The request body includes: • device activation code • public key • PRODA organisation id • software instance name Submit the request as a HTTPS request

2. PRODA will action the request	PRODA will verify the device activation code, activate the software instance for the organisation, and store the public key against the software instance .
3. Receive successful response	A successful response will be returned with HTTPS Status 200 (OK) and content type of application /json. The response body will contain: • organisation id – the identifier of the organisation submitting the request. • device name – identifier of the software instance submitting the request for the organisation. • device status – the status of the software instance being activated. • key status – the status of the public key associated with the software instance • key expiry – expiry date and time of the public key. • device expiry – the date and time the software instance will expire.

Table 8 - Activate device service

4.2.2 Interface specifications

4.2.2.1 Service Reference

Service	Device Activation
URL – ext test (vendor environment)	https://test.5.rsp.humanservices.gov.au/piaweb/api/b2b/v1/devices/<deviceName >/jwk
URL - prod	https://5.rsp.humanservices.gov.au/piaweb/api/b2b/v1/devices/<deviceName >/jwk
Supported Operation	HTTP PUT

4.2.2 Common HTTP Request Headers

Parameter	Required?	Description	Data type / [Value]
dhs-auditId	Yes	The audit field represents the PRODA organisation (id) that owns the Software Instance.	<org_id>
dhs-auditIdType	Yes	The auditIdType field represents the class of audit user specified. Always has a literal value of: http://humanservices.gov.au/PRODA/org	[http://humanservices.gov.au/PRODA/org]
dhs-subjectId	Yes	The subject Id field represents the software component that is formulating the request. The device name used during software instance registration on 'Register new B2B device' screen.	<deviceName>
dhs-subjectIdType	Yes	The subjectIdType field represents the class of the subject. Always has a literal value of: http://humanservices.gov.au/PRODA/device	[http://humanservices.gov.au/PRODA/device]
dhs-messageId	Yes	A unique identifier through which the message can be identified.	Urn:uuid format
dhs-correlationId	Yes	A unique identifier through which the session (a related set of messages) can be grouped.	Urn:uuid format

Parameter	Required?	Description	Data type / [Value]
dhs-productId	Yes	The product ID represents the human readable name of the software making this call. It should indicate the name and version of the software making this API call. This is not the client_id that PRODA issues as part of the vendor Notice of Integration process. This field is required, but not validated, and is for audit purposes.	<software name and version> e.g. SoftConnect.v1.0 or MCOL.v2019.10.02

4.2.2 Service HTTP Request Headers

Parameter	Required?	Description	Data type / [Value]
Content-type	Yes	Identifies type of data that is actually sent.	[application/json]

4.2.2.4 Service Request

4.2.2.4.1 Request Path Parameters

Parameter	Required?	Description	Data type / Format
deviceName	Yes	The name of the software instance created during software instance registration on PRODA's 'Register new B2B device' screen.	Alphanumeric

4.2.2.4.2 Request Body Parameters

The following parameters must be provided to the service upon software instance activation.

Parameter	Required?	Description	Data type / Format
otac	Yes	One time access code	Alphanumeric. Length 10
key	Yes	JWK – [refer to Key Pair Generation section]	JSON
orgId	Yes	Organisation ID	Numeric

Response

The request was processed successfully and JSON representation of the activated software instance is returned to the client in the response body.

Response Header

4.2.2

HTTP Status	200 (OK)
Content Type	application/json

4.2.2.5.1

Response Body

Device info returned in the response:

4.2.2

Parameter	Supplied?	Description	Data type / [Value]
orgId	Always	Uniquely identifies an organisation.	Numeric
deviceName	Always	Software instance identifier	Alphanumeric
deviceStatus	Always	Status of the software instance.	[ACTIVE]
keyStatus	Always	Status of the key that is associated with the software instance.	[ACTIVE]
keyExpiry	Always	Expiry date time of the public key. The timestamp format: "yyyy-MM-dd"HH:mm:ss, time zone: "Australia/Sydney". The software instance should use this value to determine if its RSA key needs to be refreshed in the near future, triggering an RSA key generation and Refresh Device Key API call to update PRODA with a new public key for the software instance.	Timestamp
deviceExpiry	Always	Expiry date time of the software instance. The timestamp format: "yyyy-MM-dd"HH:mm:ss, time zone: "Australia/Sydney". The software instance should use this value to determine if the definition of the software instance (device) in PRODA needs to be updated in the near future. The	Timestamp

		owner/user of the software instance needs to be notified that the device definition needs to be refreshed in PRODA, a new DAC obtained, and entered into the software instance, to effect a Device Activation API call. Failure to do this before expiry will disable the software instance definition in PRODA, and subsequent failure to obtain a PRODA access token to access Service Provider services.	

4.2.2 Error Response Format

An API error, as returned to service consumers, contains the following attributes:

Parameter	Supplied	Description	Data Type/[Values]
errors	Always	List of errors	Array
reference	Always	Identifier for the error type	Alphanumeric
status	Always	Http return status	Numeric
url	Always	The REST URL that threw the error	URI

For each error the following attributes are provided:

Parameter	Supplied	Description	Data type / [Values]
code	Always	Error code	Alphanumeric
message	Always	Device name	Alphanumeric

4.2.2.5.4 Service Error Codes

Type	Http Response Code	Description	Reason
DE.2	404	Organisation not found	The organisation's identifier provided in the request is not associated with an active organisation.

Type	Http Response Code	Description	Reason
DE.4	404	Device not found.	The software instance name provided in the service request is not associated with the identified organisation.
DE.5	500	Device is in invalid state.	Only software instances that are in the 'Inactive' state and are associated with an active 'activation code' can be activated.
DE.9	500	Organisation is not active.	The software instance must be associated with an active organisation.
DE.7	500	Invalid OTAC provided.	The activation code provided in the request does not match the software instance‘s activation code.
JWK.1	400	Parse Exception	The public key provided is not in the valid format.
JWK.2	400	Invalid Algorithm	The key provided must be an RSA public key with the following properties: • Key Size: 2048 (minimum size) • Key Use: Signature • Algorithm RS256 (RS384 and RS512 are also accepted) • Key Id: Is the name of the software instance with which the key is associated.
JWK.8	400	Invalid Key Use	The key "use" value must be provided and set to "sig".
JWK.9	400	Key In History	The key provided in the request is a historical key that was previously assigned to the software instance.
111	400	Input error	Identifies an input validation error.

4.3 Software Instance Authentication

Once the Software Instance has been registered and activated in PRODA, it is ready to be used for authentication purposes.

The software instance authentication service is a standard OAuth 2.0 token end-point, or authorisation service, which is configured to generate an OAuth2.0 JWT Bearer Token authorisation grant type. The authorisation service is an implementation of the OAuth Assertion Framework.

In order for a software instance to be able to use the service the following specific parameter values must be provided in a service requests body:

The value of the '***grant_type'*** parameter must be '**urn:ietf:params:oauth:grant-type:jwt-bearer'**.

The value of the '***assertion'*** must contain a single JWT signed by the software instance's private key.

The JWT assertion must also contain the identifier (in the token.aud field) for the Service Provider to be accessed. As PRODA access tokens are in JWT format, and contain Service Provider authorisation claims, they are only relevant and usable with the Service Provider they are requested for.

On successful verification of an Access Token request the PRODA authorisation service will return an OAuth 2.0 Access Token, which can be then used in a request to a Service Provider API in the Authorization header of the HTTP request. A token can live up to 60 minutes, although this may depend on the security requirements of the Service Provider. When it expires, the software is required to obtain another Access Token using the same service. The Access Token expiry time is included in the Access Token response. Write your code to anticipate the possibility that the token might no longer work. We suggest tracking expiration time and requesting a new token before it expires, rather than handling a token expiration error.

4.3.1 Access Token Reuse

A PRODA access token is valid for a period of time, generally 60 minutes depending on the security requirements of the Service Provider. This means an access token can be reused for calls to the Service Provider up until the expiry time of the access token is reached.

Best practice around this concept is described in the PRODA **B2B Best Practice – Developers Guide**, available from Services Australia OTS team.

4.3.2 Goals

Name	B2B Authorisation
Goal	PRODA will successfully authenticate the software instance, and issue an Access Token to authorise submission of B2B API requests to a Service Provider service.
Primary Actor	The software instance on behalf of the organisation.
Precondition	The software instance is registered and active in PRODA. The software instance contains a key pair. PRODA holds the active public key for the software instance.
Steps	1. The software instance submits a B2B request to PRODA for authorisation. The request will include the organisation id, device name, client id, Service Provider identifier and a signature. 2. PRODA will validate the signature using the public key and the software instance details. 3. PRODA will respond with an Access Token for valid requests.
Post condition	The software instance has an Access Token authorising submission of B2B requests to a Service Provider service.

4.3.3 Interactions

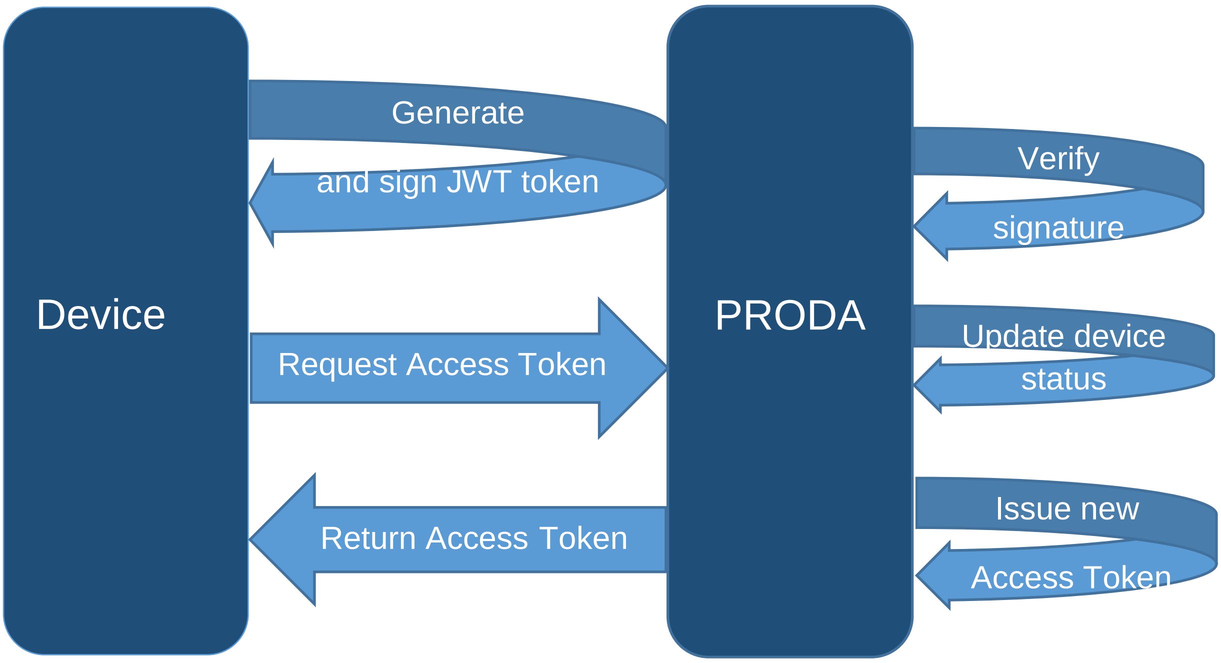


Figure 5 - Authorisation service

Name	Description
1. Generate and sign JWT token	The software instance generates a signed JWT using its active private key.
2. Request an OAuth 2.0 Access Token	The dev software instance submit the software instance authentication request (as defined in Section 5.3.41).
3. Verify the request	PRODA will verify the signature using the public key stored against the software instance, update the software instance status, and issue an Access Token.
4. Receive response containing Access Token	The response will be returned with HTTP Status 200 (OK) and content type of application /json. The response will contain: • access_token - representing the authorised request • key_expiry - the time the key will expire • device_expiry - the time the device will expire • scope - the purpose of the Access token • token_type - “bearer” • expires_in - the time the Access Token will expire (in seconds)

4.3.4 Interface Specifications

4.3.4.1 Service Reference

Service	Authentication service
URL - prod	https://proda.humanservices.gov.au/mga/sps/oauth/oauth20/token
URL – ext test	https://vnd.proda.humanservices.gov.au/mga/sps/oauth/oauth20/token
Supported Operation	HTTP POST

4.3.4.2 Service Request

4.3.4.2.1 Request Header Parameters

Parameter	Data type / [Value]
Content-type	[application/x-www-form-urlencoded]

4.3.4.2.2 Request Body Parameters

Attribute	Value

grant_type	urn:ietf:params:oauth:grant-type:jwt-bearer
assertion	<Signed JWT>
client_id	<provided during software registration>

Request Assertion

The request assertion needs to contain a JWT that has been signed by the private component of the software instance's RSA key.

4.3.4.2.3 In addition to being signed with the correct private key the JWT must contain a set of valid JWT claims and header elements.

Required JWT Claim Set:

Attribute	Value
iss	<organisation id>
sub	<device name>
aud	<audience string> Always set to: https://proda.humanservices.gov.au
token.aud	< Service Provider's audience string> **Note**: This string can be obtained from PRODA support. A list is provided at the end of this document, but may not be up to date with new Service Providers.
iat	<issued-at time timestamp>
exp	<expiry time timestamp – should be very short, 600 seconds after "iat" would be acceptable.>

💣 Expiry (exp) time must be after issued (iat) time.

Required JWT Header elements:

Attribute	Value
alg	[RS256 | RS384 | RS513]
kid	<software instance name>

💣 The public part of the RSA key provided during software instance activation is used for signature verification. It is absolutely essential that the JWT is signed with the correct private key. Attempting software instance authentication with an assertion that has been signed with the wrong private key will result in the software instance getting LOCKED after five unsuccessful authentication attempts.

Response Header

HTTP Status	200 (OK)
Content Type	application/json

4.3.4.2.3.1 Response Body

The following parameters will be provided in the response body.

4.3.4

Attribute	Value
token_type	bearer
access_token	<Access Token to supply to Service Provider> in JWT format
expires_in	<lifetime of Access Token expressed in seconds> 3600 seconds
key_expiry	<Timestamp containing expiry date-time of the public key> 11 days from software instance activation or last key refresh.
device_expiry	<Timestamp containing expiry date-time of the software instance definition in PRODA> 6 months from software instance activation.
scope	Empty

💣 The service response body contain attribute values that can be used by the Vendor Software to identify when the software instance needs to be reactivated and whether or not the software instance key refresh is required.

We suggest tracking these expiry times and taking an appropriate action before the software instance or software instance key expires. It is strongly recommended to evaluate the 'key_expiry' and 'device expiry' values on every successful software instance authentication event and perform the software instance key refresh process before the software instance's key expires. If the key expires, the software instance will be disabled and it will need to be reactivated using the software instance activation process which requires manual intervention from the software instance owner.

Error Response Format

If an assertion is not valid, has expired or is associated with a software instance that is in an invalid state, the authorisation service will respond with an error response.

The error response header will have following parameters.

4.3.4

Parameter	Supplied?	Description	Data type / [Values]
Status	Always	Http return status	[400]
Content-Type	Always		[application/JSON]
Cache-Control	Always		[no-store]

The error response body will contain a JSON object with the following parameters

Parameter	Supplied?	Description	Data type / [Values]
error	Always	Error code	Alphanumeric
error_description	Always	Description	Alphanumeric

The authorisation service may include additional information regarding the reason why the assertion was considered invalid using the “error_description” parameter.

4.3.4.2.3.4 Service Error Codes

Error	Error Description	Reason
mapping_error	Token was not valid	When provided assertion is not valid or has expired.
device_error	Device is inactive.	The software instance is not active and it must be reactivated.
device_error	Device key has expired.	The public key has expired and software instance must be reactivated.
device_error	Device has expired	The software instance has expired and must be reactivated.

💣 Device errors are returned when manual intervention by the software instance owner is required to resolve the reported error.

4.4 Refresh Software Instance Key

A key security constraint associated with the framework is that active software instances will need to refresh their RSA key pairs on a periodic basis. This security constrained is enforced by associating public keys stored in PRODA with a Time to Live (TTL) period which determines for how long a public key can be used for verification of authentication request.

The 'key_expiry' value returned by the software instance authentication service can be used to determine whether or not the software instance needs to generate new RSA key and publish its public component to PRODA.

The refresh software instance key service is an authenticated service that can be used to update the public key associated with the software instance in PRODA. We strongly recommend monitoring the key expiry and automatically refreshing the software instance key before the current key expires.

As this is an authenticated service, it can only be used when the public key is still active. If the public key has expired, it is necessary for the authorised person to login to the PRODA online application, and regenerate a device activation code. This device activation code is then manually passed to the software instance, which calls the PRODA device activation service with the device activation code and a new public key in order to replace the expired public key.

The new public key must not match any previous public keys used by the software instance.

Name	B2B Device Key Refresh
Goal	The organisation will successfully submit a new public key to PRODA.
Primary Actor	The software instance on behalf of the organisation.
Precondition	The software instance is registered and active in PRODA. The software instance has an active Access Token with an audience of https://proda.humanservices.gov.au. PRODA contains an active public key for the registered software instance. The software instance has generated a new RSA key pair.

Steps	1. The software instance submits a B2B request to PRODA to refresh the public key. The request will include the organisation id, software instance name, Access Token, and a new public key. 2. PRODA will validate the Access Token. 3. PRODA will replace the existing public key with the newly supplied public key.
Post condition	PRODA contains a new active public key for the registered software instance.

4.4.1 Interactions

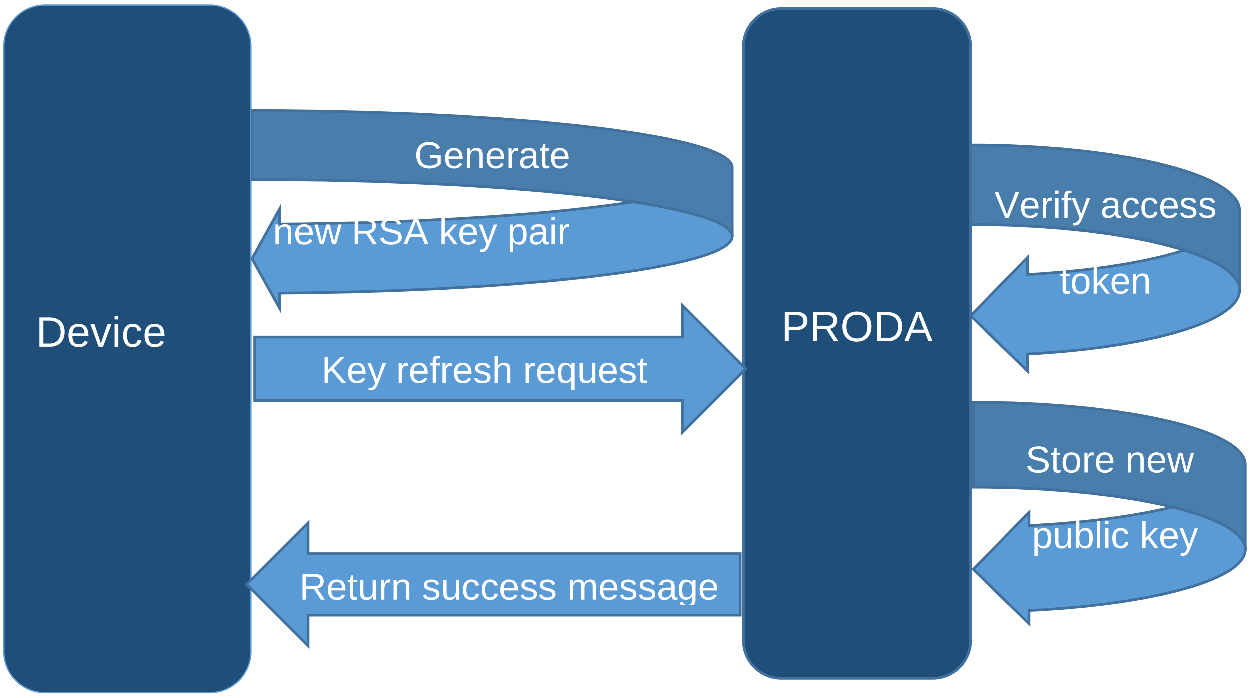


Figure 6 - Refresh device key service

Name	Description
Key refresh request	The software instance will obtain an Access Token from the authentication service. The software instance will generate new RSA Key pair. The software instance will prepare the request for device key refresh. The request will contain request path parameters in the query string: • Version – the version of the API. • Organisation id – identifier of the organisation submitting the request. • Software instance name – identifying the software instance registered in PRODA for the organisation. The request body will contain new public key in the JWK format.
PRODA request processing	PRODA will verify the Access Token, and store the new public key against the software instance.
Return success message	The response will with HTTP Status 200, and content type of application /json.

	The returned response body contains a JSON object with the following information: • orgId – the PRODA organisation ID of the owning PRODA organisation, • deviceName – the software instance name, • deviceStatus – the status of the software instance, should be "ACTIVE", • keyStatus – the status of the refreshed public key, should be "ACTIVE", • keyExpiry – timestamp indicating when the new key will expire. NOTE: The timestamp format is: "yyyy-MM-dd HH:mm:ss", time zone: "Australia/Sydney". • deviceExpiry – timestamp indicating when the software instance definition will expire. The software instance definition must be re-activated by an authorised organisation member before this date via the PRODA web application. NOTE: The timestamp format is: "yyyy-MM-ddHH:mm:ss", time zone: "Australia/Sydney".

4.4.2 Interface specifications

4.4.2.1 Service Reference

Service	JWK Update
URL – ext test	https://test.5.rsp.humanservices.gov.au/piaweb/api/b2b/v1/orgs/<org_id>/devices/<deviceName >/jwk
URL –prod	https://5.rsp.humanservices.gov.au/piaweb/api/b2b/v1/orgs/<org_id>/devices/<deviceName>/jwk
Supported Operation	HTTP PUT

4.4.2.2 Request

4.4.2.2.1 Request Header Parameters

The service request header parameters consist of a set of common request headers also used by the device activation service. The request must also contain an Authorization header.

Authorization Header

This key refresh call must be authorised using an access token. The software must request an access token from PRODA, with the token.aud value in the request indicating the access token will be presented to PRODA (https://proda.humanservices.gov.au).

The access token must be included in the refresh software instance key API call. It should be presented in the Authorization header as a bearer token. See example in section 5.3.7.

4.4.2.2.1.1

Request Path Parameters

4.4.2

Parameter	Required?	Description	Data type / Format
org_id	Required	Uniquely identifies a PRODA organisation. This is displayed on the Organisation details screen and 'Register new B2B Device' screen. Also referred to as PRODA RA (Org)	Numeric
deviceName	Required	The name of software instance used during device registration on 'Register new B2B device' screen, Uniquely identifies a software instance associated with an organisation.	Alphanumeric

4.4.2.2.3

Request Body Parameters

The public key in JWK format outlined in the Key pair generation section of the document.

4.4.2.2.4

Response

A successful result will return an HTTP 200 status code, the body of the response will contain information about the current state of the software instance.

4.4.2.2.4.1

Response Header

HTTP Status	200
Content Type	application/json

4.4.2 Response Body

Parameter	Supplied?	Description	Data type / VALUES
orgId	Always	Uniquely identifies an organisation. This is displayed on the Organisation details screen and 'Register new B2B Device' screen. Also referred to as PRODA RA. (Org)	Numeric
deviceName	Always	Software instance identifier	Alphanumeric
deviceStatus	Always	Status of the software instance	[ACTIVE]
keyStatus	Always	Status of the key that is associated with the software instance	[ACTIVE]
keyExpiry	Always	Expiry date time of the public key. NOTE: The timestamp format is: "yyyy-MM-dd HH:mm:ss", time zone: "Australia/Sydney".	Timestamp
deviceExpiry	Always	Expiry date time of the software instance definition. NOTE: The timestamp format is: "yyyy-MM-dd HH:mm:ss", time zone: "Australia/Sydney".	Timestamp

4.4.2.2.5 Errors

The same set of errors that apply to the activation service also apply to this service.

5 Appendix

5.1 Environment URLs

5.1.1 External Test

The external test environment is used by Software Vendors to test their software against the services.

The base URLs for the external test environment are:

Service	URL
Device Activation	https://test.5.rsp.humanservices.gov.au
Device Authentication	https://vnd.proda.humanservices.gov.au/
Refresh Device Key	https://test.5.rsp.humanservices.gov.au

5.1.2 Production

The base URL for the production environment are:

Service	URL
Device Activation	https://5.rsp.humanservices.gov.au
Device Authentication	https://proda.humanservices.gov.au/
Refresh Device Key	https://5.rsp.humanservices.gov.au

5.2 Security Configuration Parameters

The key security constraints associated with the system are:

- A newly registered/activated software instance is associated with a device activation code that needs to be used within the defined time period (DAC TTL).
- Once activated the software instance key must be periodically refreshed for the software instance to stay active (Key TTL).
- The software instance will stop working after a period of time and needs to be reactivated using the device activation process (Device TTL).

5.2.1 External Test (Vendor)

Parameter	Value
Device Activation Code TTL	30 days
Key Pair TTL	180 days
Device TTL	62 months

5.2.2 Production

Parameter	Value
Device Activation Code TTL	7 days
Key Pair TTL	11 days
Device TTL	6 months

5.3 Sample Requests

The following non-normative examples shows sample requests and response from the device activation, authentication and device key refresh service (with extra line breaks for display purposes only).

5.3.1 Activate Device Sample Request

```
PUT https://test.5.rsp.humanservices.gov.au/piaweb/api/b2b/v1/devices/my-device-01/jwk HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/json
dhs-auditIdType: http://ns.humanservices.gov.au/audit/type/proda/organisation
dhs-subjectId: my-device-01
dhs-productId: ACME-DummyProduct-v1.0.5
dhs-auditId: 9544531825
dhs-messageId: urn:uuid:28ee25da-53f0-4069-aca5-f0040af2423a
dhs-correlationId: uuid:64f56305-667a-4911-b1e6-fad0c4c0adbd
dhs-subjectIdType: http://ns.humanservices.gov.au/audit/type/proda/device
accept: application/json
Content-Length: 479
Host: test.5.rsp.humanservices.gov.au
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

{
       "orgId": "9544531825",
       "otac": "4tS7twZi8D",
       "key": {"kty":"RSA","kid":"my-device-
01","use":"sig","alg":"RS256","n":"pN27XWnzCfoVv5ZWSBtnwNIvKOmqXrgvPJxjx6QBEgCLQi1cUPwy7hWiPX-
TDUUG2V58VNieNiAQNLfgNg0ewijxY6_TM90tahhc6MDQUF37eieUNi4BVy6fC-UWt8-
uA76JLxzXCnwdtoilB37vEwztYyvzNQfNh3A2jzuOLMg4oqthVZ-
BQyw7y0QoiSPdaXjEci6gAuOmDsOS1wPJd_vg0tlsNfytpIUxWW5Z41gKa0R_DHmtBn8-
dazzEhbp1stlU3YyID2hThwscpwBUfy_cWa5NcZawF1tGc4j8UA5qJW8q10hQNEJku4UimahZaTlMin60XN8TYB4Ln2wjQ"
,"e":"AQAB"}
}
```

5.3.2 Activate Device Sample Response

```
HTTP/1.1 200 OK
X-Backside-Transport: OK OK,OK OK,OK OK,OK OK
Connection: Keep-Alive
Transfer-Encoding: chunked
Date: Fri, 04 Oct 2019 05:40:03 GMT
X-Frame-Options: SAMEORIGIN
X-Powered-By: Servlet/3.0
Vary: Accept-Encoding,User-Agent
Content-Type: application/json
Content-Language: en-US
X-Global-Transaction-ID: 47d391e85d96db330bc3814d

{
  "orgId" : "9544531825",
  "deviceName" : "my-device-01",
  "deviceStatus" : "ACTIVE",
  "keyStatus" : "ACTIVE",
  "keyExpiry" : "2020-04-01 15:40:03",
  "deviceExpiry" : "2024-12-04 15:40:03"
}
```

5.3.3 Activate Device Error Response

```
HTTP/1.1 400 Processed
X-Backside-Transport: FAIL FAIL,FAIL FAIL,FAIL FAIL,FAIL FAIL
Connection: Keep-Alive
Transfer-Encoding: chunked
Date: Fri, 04 Oct 2019 05:44:59 GMT
X-Frame-Options: SAMEORIGIN
X-Powered-By: Servlet/3.0
Vary: Accept-Encoding,User-Agent
Content-Type: application/json
Content-Language: en-US
X-Global-Transaction-ID: 47d391e85d96dc5b0bc3dc8d

{
  "errors" : [ {
    "code" : "JWK.9",
    "message" : "Key In History."
  } ],
  "reference" : "N/A",
  "status" : "404",
  "url" : "PUT  /piaweb/api/b2b/v1/devices/my-device-01/jwk"
}
```

5.3.4 Authorisation Service Request

```
POST https://vnd.proda.humanservices.gov.au/mga/sps/oauth/oauth20/token HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 724
Host: vnd.proda.humanservices.gov.au
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

client_id=soape-testing-client-v2&grant_type=urn:ietf:params:oauth:grant-type:jwt-bearer
&assertion=eyJraWQiOiJteS1kZXZpY2UtMDEiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiI5NTQ0NTMxODI1Iiwic3ViIjoib
XktZGV2aWNlLTAxIiwiYXVkIjoiaHR0cHM6Ly9wcm9kYS5odW1hbnNlcnZpY2VzLmdvdi5hdSIsInRva2VuLmF1ZCI6
Imh0dHBzOi8vcHJvZGEuaHVtYW5zZXJ2aWNlcy5nb3YuYXUiLCJleHAiOjE1NzAxNjgyMDQsImlhdCI6MTU3MDE2
NzYwNH0.ct62mclS-YTbv3QLVMQ16Y46KCRPmSuvSyvsM32NkYohojCTem-
tSysm3Co_TJE19fs9ImvmPaBCLVSzTptsAm6pkAQu5pf2xFfTMtgHt5FwvWsY4xRyO4NDADTifclU2rv38ojNZv0Vy
R0JWeUeQFWs3k58ymZ6A-
5uwAhN3mVjmtd9AVwS00Cml2sO5zrJCqORzId75sauDgUlzHzPB0Z0rne236UFoOw6LgHNeeoDeIppDgrHzzXiJIg
f3ykRXH27h4HkwQK15J6poG1fRYpaLzOBR2vfoUSCmxrscO-
H4MrmZ4ZW4g6hEAMQXcA4Dix2FqWgvGRivCdET5J-Eg
```

The assertion parameter is a JWT and looks like this.

Encoded:

eyJraWQiOiJteS1kZXZpY2UtMDEiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiI5NTQ0NTMxODI1Iiwic3ViIjoibXktZGV2a
WNlLTAxIiwiYXVkIjoiaHR0cHM6Ly9wcm9kYS5odW1hbnNlcnZpY2VzLmdvdi5hdSIsInRva2VuLmF1ZCI6Imh0d
HBzOi8vcHJvZGEuaHVtYW5zZXJ2aWNlcy5nb3YuYXUiLCJleHAiOjE1NzAxNjgyMDQsImlhdCI6MTU3MDE2NzYw
NH0.ct62mclS-YTbv3QLVMQ16Y46KCRPmSuvSyvsM32NkYohojCTem-
tSysm3Co_TJE19fs9ImvmPaBCLVSzTptsAm6pkAQu5pf2xFfTMtgHt5FwvWsY4xRyO4NDADTifcIU2rv38ojNZv0
VyR0JWeUeQFWs3k58ymZ6A-
5uwAhN3mVjmtd9AVwS00Cml2sO5zrJCqORzId75sauDgUlzHzPB0Z0rne236UFoOw6LgHNeeoDeIppDgrHzzXiJ
Igf3ykRXH27h4HkwQK15J6poG1fRYpaLzOBR2vfoUSCmxrscO-
H4MrmZ4ZW4g6hEAMQXcA4Dix2FqWgvGRivCdET5J-Eg

Decoded:

```
{
  "kid": "my-device-01",
  "alg": "RS256"
},
{
  "iss": "9544531825",
  "sub": "my-device-01",
  "aud": "https://proda.humanservices.gov.au",
  "token.aud": "https://proda.humanservices.gov.au",
  "exp": 1570168204,
  "iat": 1570167604
},
{ <JWT signature (not shown here)>}
```

Note: the token.aud field should represent the Service Provider the obtained access token will be presented to for authorisation to use their services. In the above example the resultant access token can only be presented to PRODA.

5.3.5 Authorisation Service Response

```
HTTP/1.1 200 OK
content-language: en-US
content-type: application/json;charset=UTF-8
date: Fri, 04 Oct 2019 05:40:04 GMT
p3p: CP="NON CUR OTPi OUR NOR UNI"
transfer-encoding: chunked
x-frame-options: DENY
x-content-type-options: nosniff
cache-control: no-store, no-cache=set-cookie
expires: Thu, 01 Dec 1994 16:00:00 GMT
x-xss-protection: 1; mode=block
content-security-policy: frame-ancestors 'none'
strict-transport-security:
pragma: no-cache
Set-Cookie: AMWEBJCT!%2Fmga!JSESSIONID=0000_MOXcZ7Xu_E4TZCgnR9Ikjb:d3e7d899-f2ea-47bf-
bde2-2f76d4952b88; Path=/
Set-Cookie: PD_STATEFUL_8dc98fae-f3fc-11e6-812f-74fe480683a9=%2Fmga; Path=/
Set-Cookie: PD-S-SESSION-ID=1_2_0_0cSnaAeE0CBVdhsTdKv1rc6o0AScvYrdrg4OsZVWZ+WF1k+s;
Path=/; Secure; HttpOnly

{"access_token":"eyJraWQiOiJWcms3MkhjWjdMYmFUdWFkdTl3NnZPYlk5bElUTF80TEJ0YjVPeWdqV3BFIiwiY
WxnIjoiUlMyNTYifQ.eyJzdWIiOiI5NTQ0NTMxODI1IiwiYXVkIjoiUFJPREEuVU5BVFRFTkRFRC5CMkIiLCJwcm9k
YS5zd2luc3QiOiJteS1kZXZpY2UtMDEiLCJwcm9kYS50eXBlIjoiVU5BVFRFTkRFRC5CMkIiLCJwcm9kYS5vcmciOi
I5NTQ0NTMxODI1IiwicHJvZGEucnAiOiJQUk9EQSIsInByb2RhLnNwIjpbIlBST0RBIl0sInByb2RhLmF1ZCI6Imh0
dHBzOi8vcHJvZGEuaHVtYW5zZXJ2aWNlcy5nb3YuYXUiLCJpc3MiOiJodHRwczovL3Byb2RhLmh1bWFuc2VydmljZX
MuZ292LmF1IiwiaWF0IjoxNTcwMTY3NjA0LCJleHAiOjE1NzAxNzEyMDR9.JI0hPY7a-x5sCFGWeQ1ko3eY1g-
8Fe8NNLIg0dK62GExzs-jlGUqonHD0AoQSYV7qH_yL8-
ks3lAmfQulbTDcQteHbumglfX1_Traj8fAcfsQl8sHSwv6t4G7uegBu5LzdHbT8MJ0FWzH-fPvjZwVx-
nZAVikr9_kGFl2_jJ6sTlmq-cBA6aKQzrWEkISlXE-
W2WKJ6t65soaSx41c_Mk6Gle6KXcFQlrNyuuG2z_lroU7_tzG1dX9r_wHaDXktL6xO42oA3LY2yXZ8CrE41NJBYL-
S0qmGjk2w2AeBcBmKw5RNtGpI3WmqpTWcKKC0I4PCxWkx9XHBLJvwzuiO7iw","scope":"","device_expiry":"
2024-12-04 15:40:03","key_expiry":"2020-04-01
15:40:03","token_type":"bearer","expires_in":3600}
```

Please note: The current PRODA authorisation service implementation returns cookie definitions. These should not be honoured, and can safely be ignored.

Decoded access token

```
{
  "kid": "Vrk72HcZ7LbaTuadu9w6vObY9lITL_4LBtb5OygjWpE",
  "alg": "RS256"
},
{
  "sub": "9544531825",
  "aud": "PRODA.UNATTENDED.B2B",
  "proda.swinst": "my-device-01",
  "proda.type": "UNATTENDED.B2B",
  "proda.org": "9544531825",
  "proda.rp": "PRODA",
  "proda.sp": [
    "PRODA"
  ],
  "proda.aud": "https://proda.humanservices.gov.au",
  "iss": "https://proda.humanservices.gov.au",
  "iat": 1570167604,
  "exp": 1570171204
}
```

5.3.6 Authorisation Service Error Response

```
HTTP/1.1 400 Bad Request
content-language: en-US
content-type: application/json;charset=UTF-8
date: Fri, 04 Oct 2019 05:55:34 GMT
p3p: CP="NON CUR OTPi OUR NOR UNI"
transfer-encoding: chunked
x-frame-options: DENY
x-content-type-options: nosniff
cache-control: no-store, no-cache=set-cookie
expires: Thu, 01 Dec 1994 16:00:00 GMT
x-xss-protection: 1; mode=block
content-security-policy: frame-ancestors 'none'
strict-transport-security:
pragma: no-cache
Set-Cookie: AMWEBJCT!%2Fmga!JSESSIONID=0000_4j-A5Rpl9qo-zh2lpFBsGW:d3e7d899-f2ea-47bf-
bde2-2f76d4952b88; Path=/
Set-Cookie: PD_STATEFUL_8dc98fae-f3fc-11e6-812f-74fe480683a9=%2Fmga; Path=/
Set-Cookie: PD-S-SESSION-ID=1_2_0_+C7OpmyTs1AhHm5ZExGU5bcFgmNBUAsCa5gY7LWvkT1fAwp1;
Path=/; Secure; HttpOnly

{"error_description":"Token was not valid","error":"mapping_error"}
```

Please note: The current PRODA authorisation service implementation returns cookie definitions. These should not be honoured, and can safely be ignored.

5.3.7 Refresh Device Key Service Request

```
PUT https://test.5.rsp.humanservices.gov.au/piaweb/api/b2b/v1/orgs/9544531825/devices/my-
device-01/jwk HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/json
Authorization: Bearer
eyJraWQiOiJWcms3MkhjWjdMYmFUdWFkdTl3NnZPYlk5bElUTF80TEJ0YjVPeWdqV3BFIiwiYWxnIjoiUlMyNTYifQ.e
yJzdWIiOiI5NTQ0NTMxODI1IiwiYXVkIjoiUFJPREEuVU5BVFRFTkRFRC5CMkIiLCJwcm9kYS5zd2luc3QiOiJteS1kZ
XZpY2UtMDEiLCJwcm9kYS50eXBlIjoiVU5BVFRFTkRFRC5CMkIiLCJwcm9kYS5vcmciOiI5NTQ0NTMxODI1IiwicHJvZ
GEucnAiOiJDQ1MiLCJwcm9kYS5zcCI6WyJDQ1MiXSwicHJvZGEuYXVkIjoiUFJPREEuVU5BVFRFTkRFRC5CMkIiLCJpc
3MiOiJodHRwczovL3Byb2RhLmh1bWFuc2VydmljZXMuZ292LmF1IiwiaWF0IjoxNTcwMTY3NjA0LCJleHAiOjE1NzAxN
zEyMDR9.lkZFk-CpssAXflCnCT8-
nT_PR3RS6jqhMjjiuK4LlAm_C0RHO74PaUPmFRPLvryhsUUWcoahWufPMROdttyircL_yi_RqgPBdUSfUWjDArASgWyl
0P6rohetPpe3jc1pYRdyaozZrxiYg_pwx9ELqj_00SqbMBEDveHyR0r2lrJb117vkM5MXt77Jy7HqUaLgT7EEGXI8xKN
Tv2GXnTSmWz3gNFNeScUO9KdDpUV4PETNKp9brMsDGL6frLgbNSrHsde9C1Vx9V28m1IsE1nLT9Qlc9NdPZBh7b_u23T
eYfFjIMoVvOArXR7hSU95dE0VhCqCzByh7QZf8m7NilMPQ
dhs-auditIdType: http://ns.humanservices.gov.au/audit/type/proda/organisation
dhs-subjectId: my-device-01
dhs-productId: ACME-DummyProduct-v1.0.5
dhs-auditId: 9544531825
dhs-messageId: urn:uuid:41be4c97-fff2-474c-b66c-cc26de1efec9
dhs-correlationId: uuid:363e3b93-e49c-4752-b2cb-6c403407d387
dhs-subjectIdType: http://ns.humanservices.gov.au/audit/type/proda/device
accept: application/json
Content-Length: 421
Host: test.5.rsp.humanservices.gov.au
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

{"kty":"RSA","kid":"my-device-
01","use":"sig","alg":"RS256","n":"5frQgOxBlZeRyE7AZRlv3vhNLFnmliSsgMY29OY0vCGXddVzLe5i7JPUu
1FxQTvjVEMUufGRMgKWeQampdyWlxN1NGgcGxl3GjLhMpgQhLwhhjPcDg7RrJ4rgAbyDTEMOORxMi92E4_qscgmssIyM
v28_45BJkJtHNVipPeJSPu72_45weBG-zujIWVbRBdG_AzW-6IpvibBL7-DdXRZJX9-
YELDqgjusYORWeCVyvY17bY_sogGlTkZ40jT2bAfqGh9E9s56EweEdRQ_G_8DLkmidQybP_YgpexCA7eXQs79YW0jUtg
8KtK2nMvEADdrQAQVw_6ZZE2C2FwASFIhw","e":"AQAB"}
```

Please note: There is no new-line in the Authorization: Bearer <JWT> header. The above example should be copied into a text editor that does not wrap the content to understand the correct structure.

5.3.8 Refresh Device Key Service Response

```
HTTP/1.1 200 OK
X-Backside-Transport: OK OK,OK OK,OK OK,OK OK
Connection: Keep-Alive
Transfer-Encoding: chunked
Date: Fri, 04 Oct 2019 05:40:05 GMT
X-Frame-Options: SAMEORIGIN
X-Powered-By: Servlet/3.0
Vary: Accept-Encoding,User-Agent
Content-Type: application/json
Content-Language: en-US
X-Global-Transaction-ID: 47d391e85d96db3500110ef1

{
  "orgId" : "9544531825",
  "deviceName" : "my-device-01",
  "deviceStatus" : "ACTIVE",
  "keyStatus" : "ACTIVE",
  "keyExpiry" : "2020-04-01 15:40:05",
  "deviceExpiry" : "2024-12-04 15:40:03"
}
```

5.3.9 Refresh Device Key Error Response

```
HTTP/1.1 500 Internal Server Error
X-Backside-Transport: FAIL FAIL,FAIL FAIL
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: application/json
Date: Fri, 04 Oct 2019 06:02:23 GMT
X-Global-Transaction-ID: 47d391e85d96e06f0bc52d8d

{"code":17,"codeType":"DHSEIN","message":"Poorly formed security credential"}
```

5.4 PRODA Service Provider Audience Strings

The following table contains the relevant Service Provider strings that should be used in the token.aud field of authorisation requests. This table is complete at the time of publishing, additional Service Providers may have been added since publication. Please contact Services Australia OTS for support with undefined values.

Service Provider	proda.rp Code	token.aud String
Medicare online	MCOL	https://medicareaustralia.gov.au/MCOL
Australian Immunisation Register (AIR)	MCOL	https://medicareaustralia.gov.au/MCOL
Child Care System	CCS	PRODA.UNATTENDED.B2B
E-Invoicing	EINV	https://proda.humanservices.gov.au/relyingparty/EINV
Pharmaceutical Benefits Scheme	PBS	https://medicareaustralia.gov.au/PBS
Aged care	AGC	https://medicareaustralia.gov.au/AGC
NDIS	NAPI	https://www.ndis.gov.au/
DET	TCSI	https://tcsi.humanservices.gov.au
PRODA	PRODA	https://proda.humanservices.gov.au

5.5 Frequently Asked Questions

1. *Is the protocol OAuth2.0 an industry standard or protocol?*

Yes, OAuth 2.0 is an industry protocol or open standard.

IT Industry leaders like Google, Facebook, and Microsoft have implemented customised version of the OAuth2.0 protocol. PRODA is trying to minimise the use of customised forms of the protocol to maximise its compatibility with other products utilising these protocols.

2. *Does DHS provides any source code for OAuth2.0?*

DHS does not supply any libraries or code for using OAuth 2.0. As it is an industry standard for authorisation, there are many resources available on the internet to help with developing against it.

3. Can I find libraries of previously created code for Oauth2.0?

There are many mature open source and commercial libraries available for developers to use to develop against PRODA OAuth 2.0 implementation.

4. *Who provides the client Id?*

PRODA provides client id (client_id) to a vendor as part of the software registration process. The client_id represents a version of software. The “client_Id” is used to identify the software making API requests to PRODA, and can be used to determine if a particular software is not working correctly. The PRODA team can customise a client Id to something meaningful to the Service Provider.

5. *Who creates private and public keys?*

Vendor software must create the private and public key pair. The public key will be passed to PRODA and PRODA stores it in its database.

Private keys must be kept safe in the vendor software or organisation software instance.

As the names suggest,

- “Private Key” is just like a password or secret,
- “Private Key” must be kept in a “vault” or safe place, in the same way as any secret or password,
- “Public Key” can be shared.

6. *Why do organisations need to provide a public key to PRODA?*

PRODA needs the public key to verify the signed identity assertion in Access Token requests. The token is signed by using the private key of the software instance, PRODA uses the public key to verify it was signed by the correct private key for the software instance.

7. *Is signing a string different from encryption or hashing?*

Signing a string is the same as encrypting a string, but when used in the signing context it only involves a short string, whereas encrypting usually refers to larger payloads of data. The original string can be obtained by decrypting the signed or encrypted string. Signing and encryption use an algorithm in conjunction with a key to work.

Hashing is different to signing and encryption, in that the result is obtained via an algorithm without the use of a key. Reversing a hash will not result in the original string, and this is not the purpose of hashing. A hash is usually used to determine if a string has been changed from its original, known content.

8. *Do I need to sign the payload (of a web service)?*

The signing or encryption of web service payloads is not a concern of PRODA; that is up to the Service Provider's requirements.

9. *What details need to be included in the "signed" JWT used to request PRODA access tokens?*

sub (subject)	<software instance name>
aud (audience)	https://PRODA.humanservices.gov.au
iss (issuer)	<Org RA number>
exp (expiry time)	Expiry time in epoch time format
iat (issued at time)	Create time in epoch time format
token.aud (access token audience)	Audience string, a unique string to identify the intended recipient (uniquely identifying a service provider)

10. *Are there any useful tags for searching in popular search engines?*

The below tags may help to request relevant information in a search engine, so that more can be learnt about the protocol.

- OAuth 2.0, RSA
- JSON Web Key
- JSON Web Token
- RSA Private Key
- RSA Public key
- RSA Public key to JWK
- OAuth 2.0 access token
- OAuth 2.0 tokens
- Signing using private key
- verifying signature by using public key
- PEM to JWK
- JWK to PEM
- single sign on
- epoch time
- UTC epoch time

5.6 Additional References

Reference	Source
OAuth 2.0	Hardt, D., “The OAuth 2.0 Authorization Framework,” Jones, M. and D. Hardt, “The OAuth 2.0 Authorization Framework: Bearer Token Usage,”
JWK	Jones, M, “JSON Web Key (JWK)”,
JWT	Jones, M., Bradley, J., Sakimura, N “JSON Web Token (JWT)”,
JWS	Jones, M., Bradley, J., and N. Sakimura, “JSON Web Signature (JWS),” RFC 7515, May 2015
JWA	Jones, M “JSON Web Algorithms (JWA)”, RFC7518, May 2015
JWT Profile Client Authentication	Jones, M, Campbell, B, Mortimore, C “JSON Web Token (JWT Profile for OAuth 2.0 Client Authentication and Authorization Grants”, RFC7523, May 2015