EVERTRUST

EverTrust Horizon documentation v2.6

Administration Guide

EVERTRUST

Table of Contents

1. Introduction

1.1. Description

Horizon is EverTrust's Certificate lifecycle management solution and is powered up by:

- Pekko
- BouncyCastle
- MongoDB
- Kamon
- Play! Framework
- Scala
- NGINX
- Vue.js
- Quasar

This document is specific to Horizon version **2.6**.

1.2. Scope

This document is an administration guide detailing how to configure and operate Horizon.

1.3. Out of Scope

This document does not describe how to install and bootstrap a Horizon instance. Please refer to the installation guide for installation related tasks.

2. User Information

This is the section where to find all your profile information (identifier, email, name, authentication type, role and permissions), your preferences and change your account password (local account authentication only).

2.1. Profile access

1. Log in to Horizon.

2. Access your profile from the header by clicking on your account name.

2.2. How to change your password

1. Profile access

2. Fill your local password and confirm it.

3. Click on the 'Change Password' button.

CAUTION Changing your password is only available if you are using a local account.

2.3. How to change your preferences

1. Profile access

2. Change your preferences:

- Appearance (light/dark mode)
- Horizon default language

3. Click on the 'Save' button.

3. Certification Authorities

This section details how to configure the Certification Authorities known by EverTrust Horizon.

3.1. Prerequisites

Certification Authorities will be needed beforehand, in one of these formats:

- Certificate file (PEM or DER).
- Certificate string (PEM).

You might also need the URL of the CRL issued by the CA, and/or the URL of the OCSP Responder for that CA.

3.2. How to configure a Certification Authority

1. Log in to Horizon Administration Interface.

2. Access Certification Authorities from the drawer or card: **Certification Authorities**.

3. Click on ⊕ .

Certificate Tab:

4. Either

- Fill in the certificate section with certificate string (PEM) OR
- Import the certificate file (PEM or DER).

Then click on the next button.

Details Tab:

5. Check the information from your CA certificate. Then click on the next button.

Configuration Tab:

6. Fill in the information you want to add.

- **Name*** *(string input)*:
  Enter a meaningful certificate authority name. It must be unique for each certificate authority.
- **OCSP responder URL** *(string)*:
  URL to request an OCSP responder.
- **CRL URL** *(string)*:
  URL to download the CA CRL.
- **Refresh Period** *(finite duration)*:
  CRL or OCSP Refresh Period. Must be a valid finite duration.

- **Timeout** *(finite duration)*:
  Connection timeout when reaching CRL or OCSP. Must be a valid finite duration.
- **Proxy** *(string select)*:
  The HTTP/HTTPS proxy to use to reach the CRL or the OCSP Responder, if any.
- **Is exposed on Registration Authority** *(boolean)*:
  Display the CA in the Trust chains view on the RA side. The default value is set to false.
- **Is trusted for server authentication** *(boolean)*:
  Tells whether the CA should be trusted for server authentication, aka SSL/TLS server trust. The default value is set to false.
- **Is trusted for client authentication** *(boolean)*:
  Tells whether the CA should be trusted for client authentication. The default value is set to false.
- **Outdated Revocation Status Policy** *(option)*:
  Select "Revoked" if you want all certificates to be handled as revoked if the CRL/OCSP are unavailable. Select "Last available status" if you want Horizon to use the last available revocation status for the certificates.

7. Click on the import button.

You can edit ✎ , download ⤓ or delete 🗑 the Certification Authorities.

CAUTION

You will not be able to delete a Certification Authority if it is referenced in any other configuration element. Pay also attention that the CA might be used (e.g. for TLS trust chain building), even if it is not explicitly referenced in configuration items.

4. PKIs

4.1. PKI Queue

This section details how to configure a PKI Queue.PKI Queue are used to limit the PKI requests (enrollment, revocation)

4.1.1. PKI Queue Configuration

1. Log in to Horizon Administration Interface.

2. Access PKI Queues from the drawer or card: **PKI › PKI Queues**.

3. Click on ⊕ .

4. Fill in the fields:

- **Name*** *(string input)*:
  Choose a meaningful queue name. It must be unique.
- **Description** *(string input)*:
  The description for the PKI Queue.
- **Throttle Parallelism** *(int input)*:
  Number of requests processed at the same time.
- **Throttle Duration** *(finite duration)*:
  Maximum requests processed at the same time in a given duration. Parallelism must be set.
- **Max Size*** *(int input)*:
  Maximum requests stored in the queue
- **Cluster Wide** *(boolean)*:
  If not enabled, then the `throttleParallelism` and `throttleDuration` will be the same for all nodes in the cluster. If enabled, then the `throttleParallelism` and `throttleDuration` is generalized for all clusters.

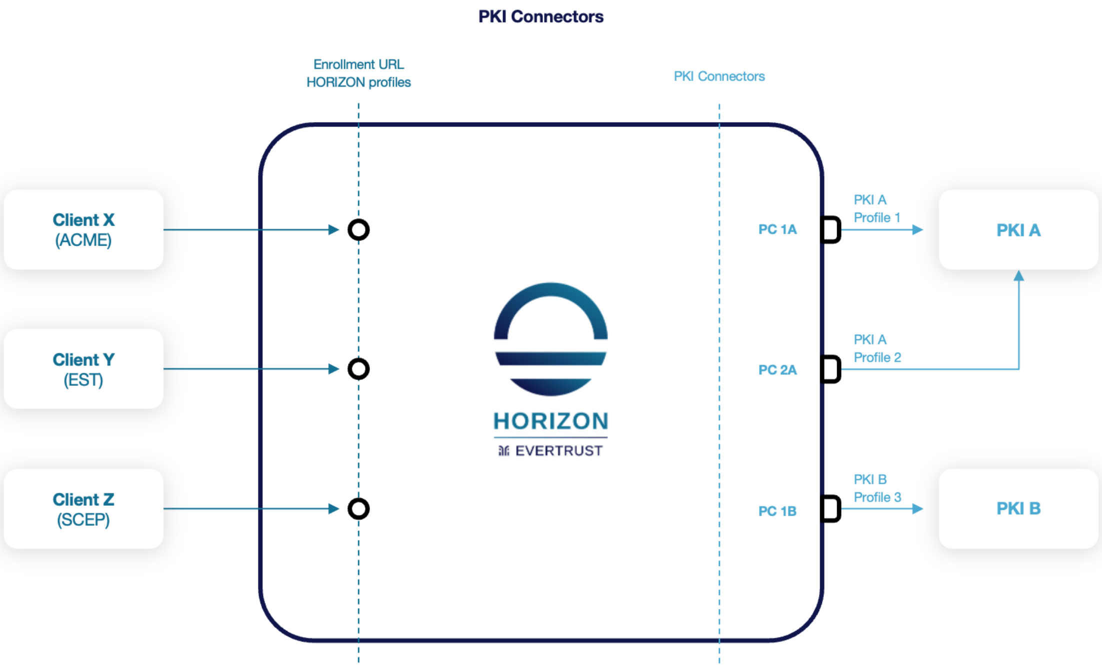
CAUTION If the queue is full every new request will be discarded.

4.2. PKI Connectors

4.2.1. PKI Connectors

Description

A "PKI Connector" is a configuration piece that allows to establish the communication with any supported PKI. Additionally, it enables to map a specific certificate profile within the connected PKI.



Common prerequisites

To grant "Horizon" proper access to a given PKI, three categories of requirements must be gathered:

- **Credentials**: It could be either certificates (PKCS#12 format) or technical accounts (login/password) allowing to authenticate against the PKI API.
- **Permissions**: The credentials must be granted with the proper permissions on the PKI in order to be able to manage certificate lifecycle (enroll, revoke, renew).
- **Profile/Certificate information**: This information is used to map certificate types and/or certificate fields.

4.2.2. AWS PKI

Prerequisites

- You need to create a user using AWS IAM, and give it the `AWSCertificateManagerPrivateCAUser` right.
- You need to retrieve the Private CA ARN from ACM Private CA console.

NOTE Refer to the editor’s documentation to configure the PKI side here.

Create the PKI connector

1. Log in to Horizon Administration Interface.

2. Access PKI from the drawer or card: **PKI › PKI Connectors**.

3. Click on ⊕ .

4. Select the correct PKI type.

5. Click on the next button

General tab

6. Fill in the common mandatory fields:

- **Connector Name*** *(string input)*:
  Choose a meaningful connector name allowing to identify the mapping between the PKI and the Certificate Profile. It must be unique and must not contain spaces.
- **Proxy** *(string select)*:
  If the PKI is not directly reachable from Horizon, you can set up an HTTP/HTTPS proxy to properly forward the traffic.
- **PKI Queue** *(string select)*:
  The PKI Queue used to manage the PKI Requests (enrollment, revocation).
- **Timeout** *(finite duration)*:
  Represents a predefined interval of time without a PKI response, when the time has passed "Horizon" will cease trying to establish the communication. Must be a valid finite duration.

7. Click on the next button

Details tab

8. Fill in all mandatory fields:

- **AWS Region*** *(string input)*:
  AWS region to use.
- **AWS PCA ARN*** *(string input)*:
  Amazon Resource Name (ARN) is a file naming convention used to identify a particular resource in AWS public cloud. To be retrieved from AWS ACM Console.
- **AWS PCA Template ARN** *(string input)*:
  A template is a declaration of the AWS resources that make up a stack. The default value is set to: `arn:aws:acm-pca:::template/EndEntityCertificate/V1`.
- **AWS PCA Role ARN** *(string input)*
- **Certificate Policy OID** *(string input)*:
  An identifying number, in the form of an "object identifier" that is included in the `certificatePolicies` field of a certificate.
- **Certificate signing hash** *(select)*:
  Select the hash function that will be used.
- **Certificate Usage** *(select)*:
  Select the certificate usage.
- **Number of valid days** *(finite duration)*:

Certificate validity duration in days. Must be a valid finite duration. The default value is set to 365 days.

- **Retry Interval** *(finite duration)*:
  Predefined interval of time before retrying to retrieve the certificate from AWS. Must be a valid finite duration. The default value is set to 3 seconds.

9. Click on the next button

Authentication tab

10. Fill in the PKI-authentication fields:

- **AWS Access Credentials*** *(select)*:
  Select `Login` credentials containing the AWS user access key ID and the AWS user secret key (see the AWS Account and Access Keys documentation).

11. Click on the save button.

You can edit ✎, duplicate ⧉ or delete 🗑 the AWS PKI connector.

4.2.3. CertEurope PKI

Prerequisites

- A technical account should be created.
- This technical account must have permissions to enroll and revoke SSL certificates on the desired domain(s).

Limitations

- Only the following fields are managed: `commonName` and `subjectAltName DNS`.
- For multi-valued fields (SAN DNS), if more data items are provided than configured in CCS for the given "Offer Identifier", the exceeding items will be ignored.
- All limitations induced by the use of the CCS REST Connector.

Create the PKI connector

1. Log in to Horizon Administration Interface.

2. Access PKI from the drawer or card: **PKI › PKI Connectors**.

3. Click on ⊕ .

4. Select the correct PKI type.

5. Click on the next button

General tab

6. Fill in the common mandatory fields:

- **Connector Name*** *(string input)*:
  Choose a meaningful connector name allowing to identify the mapping between the PKI and the Certificate Profile. It must be unique and must not contain spaces.
- **Proxy** *(string select)*:
  If the PKI is not directly reachable from Horizon, you can set up an HTTP/HTTPS proxy to properly forward the traffic.
- **PKI Queue** *(string select)*:
  The PKI Queue used to manage the PKI Requests (enrollment, revocation).
- **Timeout** *(finite duration)*:
  Represents a predefined interval of time without a PKI response, when the time has passed "Horizon" will cease trying to establish the communication. Must be a valid finite duration.

7. Click on the next button

Details tab

8. Fill in all mandatory fields:

- **Endpoint URL to the CSS partner API*** *(string input)*:
  URL to access the CertEurope web service API.
- **Technical account credentials*** *(select)*:
  Select `Login` credentials containing your technical account created in CCS. The login is usually an email address.
- **CCS offer identifier*** *(string input)*:
  The identifier of the offer within CCS.
- **Organization ID*** *(string input)*:
  Customer organization ID. For French companies, it’s usually the "SIREN".
- **Revocation reason** *(string select)*:
  Select from the drop down the default revocation reason.
- **Interval before retrying to retrieve certificate** *(finite duration)*:
  The default value is set to 21 seconds.

9. Click on the next button.

Authentication tab

10. Fill in the PKI-authentication fields:

- **Authentication Credentials*** *(select)*:
  Select `Certificate` credentials containing the authentication certificate used to connect to the PKI.

11. Click on the save button.

You can edit ✎, duplicate ⧉ or delete 🗑 the CertEurope PKI connector.

4.2.4. CS-Novidy's TrustyKey PKI

Prerequisites

- A technical account should be created.
- This technical account must have permissions to enroll and revoke SSL certificates on the desired certificate profiles.
- An authentication and a signature certificate must be issued under as PKCS#12 files for this account.

Limitations

- Only the following fields are managed: `commonName (as mail_lastname)`, `contactEmail (as mail_email)`, `OU (as org_unit)`, `O (as corp_company)`, `C (as country)`, `UID (as employeeID)`, `subjectAltNames DNS` and `msUPN`.
- For multi-valued fields (SAN DNS), if more data items are provided than configured in TrustyKey for the given `PGC`, the exceeding items will be ignored.
- All limitations induced by the use of the TrustyKey CMP Connector.

Create the PKI connector

1. Log in to Horizon Administration Interface.

2. Access PKI from the drawer or card: **PKI › PKI Connectors**.

3. Click on ⊕.

4. Select the correct PKI type.

5. Click on the next button

General tab

6. Fill in the common mandatory fields:

- **Connector Name*** *(string input)*:
  Choose a meaningful connector name allowing to identify the mapping between the PKI and the Certificate Profile. It must be unique and must not contain spaces.
- **Proxy** *(string select)*:
  If the PKI is not directly reachable from Horizon, you can set up an HTTP/HTTPS proxy to properly forward the traffic.
- **PKI Queue** *(string select)*:
  The PKI Queue used to manage the PKI Requests (enrollment, revocation).

- **Timeout** *(finite duration)*:
  Represents a predefined interval of time without a PKI response, when the time has passed "Horizon" will cease trying to establish the communication. Must be a valid finite duration.

7. Click on the next button

Details tab

8. Fill in all mandatory fields:

- **API endpoint URL*** *(string input)*:
  URL to access the CS-Novidy's TrustyKey web service.
- **PGC*** *(string input)*:
  Enter name of the `PGC` to be used.
- **TrustyKey PKI server DN*** *(string input)*:
  Enter the DN of the TrustyKey PKI server, starting from the CN.
- **TrustyKey PKI server Certificate*** *(string input)*:
  Enter the PEM representing the certificate of the CA issuing the certificates.
- **CN mapping** *(string input)*:
  Enter a CN to be mapped.
- **Email mapping** *(string input)*: Enter an email address or domain to be mapped.
- **SAN DNS mapping** *(string input)*:
  Enter a SAN DNS to be mapped.
- **Profile mapping** *(string input)*:
  Enter a profile to be mapped.
- **Issuer mapping** *(string input)*:
  Enter an issuer to be mapped.
- **Legacy CMP Style** *(boolean)*
  Chose whether to use the legacy CMP style.

9. Click on the next button.

Authentication tab

10. Fill in the PKI-authentication fields:

- **Authentication Credentials*** *(select)*:
  Select `Certificate` credentials containing the authentication certificate used to connect to the PKI.
- **Signer Credentials*** *(select)*:
  Select `Certificate` credentials containing the signature certificate used to sign the CMP messages.

11. Click on the save button.

You can edit ✎, duplicate ⧉ or delete 🗑 the CS-Novidy's TrustyKey PKI connector.

4.2.5. DigiCert CertCentral PKI

Prerequisites

- You need to validate the domain(s) for which you will issue certificates prior to their issuance. This can be done in DigiCert CertCentral in the Certificates > Domains menu.
- You need to retrieve the `organizationId` from DigiCert CertCentral in the Certificates > Organizations menu.
- You need to generate an API Key in DigiCert CertCentral using the Account > Account Access menu.

Limitations

- Only the following fields are managed: `commonName` and `subjectAltName DNS` and `RFC822Name`.
- For multi-valued fields (SAN DNS and `RFC822Name`), if more data items are provided than configured in DigiCert CertCentral for the given type of certificate, the exceeding items will be ignored.
- All limitations induced by the use of the DigiCert CertCentral REST Connector.

Create the PKI connector

1. Log in to Horizon Administration Interface.

2. Access PKI from the drawer or card: **PKI › PKI Connectors**.

3. Click on ⊕ .

4. Select the correct PKI type.

5. Click on the next button

General tab

6. Fill in the common mandatory fields:

- **Connector Name*** *(string input)*:
  Choose a meaningful connector name allowing to identify the mapping between the PKI and the Certificate Profile. It must be unique and must not contain spaces.
- **Proxy** *(string select)*:
  If the PKI is not directly reachable from Horizon, you can set up an HTTP/HTTPS proxy to properly forward the traffic.
- **PKI Queue** *(string select)*:
  The PKI Queue used to manage the PKI Requests (enrollment, revocation).
- **Timeout** *(finite duration)*:
  Represents a predefined interval of time without a PKI response, when the time has passed "Horizon" will cease trying to establish the communication. Must be a valid finite duration.

7. Click on the next button

Details tab

8. Fill in all mandatory fields:

- **DigiCert CertCentral API endpoint*** *(string input or select)*:
  URL to access DigiCert CertCentral API along with the certificate type to issue. To do so you can select from the drop down menu or type in your "certificate offer" value, then press "Enter" the corresponding URL will be automatically fetched.
- **DigiCert CertCentral Customer Organization ID*** *(int)*:
  Enter customer organization ID.
- **DigiCert CertCentral CA Cert ID** *(int)*:
  Enter CA Cert ID, to be used for private CA only.
- **Interval before retrying to retrieve certificate** *(finite duration)*:
  Use for private CA only. The default value is set to 9 seconds.
- **Skip Approval** *(boolean)*:
  The default value is set to false.

9. Click on the next button.

Custom tab

10. Click on ⊕ if custom data mapping is needed.

11. Fill in the PKI-custom data mapping:

- **Custom data field*** *(string input)*:
- **Label field*** *(select)*:
  Any existing Horizon Label

12. Click on the next button.

Authentication tab

13. Fill in the PKI-authentication fields:

- **DigiCert CertCentral API Key*** *(select)*:
  Select `API Token` credentials containing the API Key.

14. Click on the save button.

You can edit ✎ , duplicate ⧉ or delete 🗑 the DigiCert CertCentral PKI connector.

4.2.6. EJBCA PKI

Prerequisites

- A certificate profile should be created, e.g. reusing the default "SERVER" certificate profile.
- An authentication certificate should be issued for Horizon, and it should be given certificate issuance and revocation permissions on the aforementioned certificate procedure.

Limitations

- Only the following fields are managed: all Subject DN fields and `subjectAltNames DNS, IPaddress, RFC822Name, msUPN` and `msGUID`.
- For multi-valued fields (SAN DNS and `RFC822Name`), if more data items are provided than configured in EJBCA for the given *End Entity* profile, the exceeding items will be ignored.
- All limitations induced by the use of the EJBCA RA SOAP Connector.

Create the PKI connector

1. Log in to Horizon Administration Interface.

2. Access PKI from the drawer or card: **PKI › PKI Connectors**.

3. Click on ⊕ .

4. Select the correct PKI type.

5. Click on the next button

General tab

6. Fill in the common mandatory fields:

- **Connector Name*** *(string input)*:
  Choose a meaningful connector name allowing to identify the mapping between the PKI and the Certificate Profile. It must be unique and must not contain spaces.
- **Proxy** *(string select)*:
  If the PKI is not directly reachable from Horizon, you can set up an HTTP/HTTPS proxy to properly forward the traffic.
- **PKI Queue** *(string select)*:
  The PKI Queue used to manage the PKI Requests (enrollment, revocation).
- **Timeout** *(finite duration)*:
  Represents a predefined interval of time without a PKI response, when the time has passed "Horizon" will cease trying to establish the communication. Must be a valid finite duration.

7. Click on the next button

Details tab

8. Fill in all mandatory fields:

- **EJBCA RA URL*** *(string input)*:
  Enter SOAP endpoint URL of the EJBCA WebService.
- **EJBCA Certificate Profile Name*** *(string input)*:
  Enter EJBCA Certificate Profile to map for certificate issuance.
- **EJBCA CA Name*** *(string input)*:
  Enter CA to use for certificate issuance.
- **EJBCA End Entity Profile*** *(string input)*:
  Enter EJBCA End Entity profile.

9. Click on the next button.

Authentication tab

10. Fill in the PKI-authentication fields:

- **Authentication Credentials*** *(select)*:
  Select `Certificate` credentials containing the authentication certificate used to connect to the PKI.

11. Click on the save button.

You can edit ✎ , duplicate ⧉ or delete 🗑 the EJBCA PKI connector.

4.2.7. Entrust Certificate Services PKI

Prerequisites

- A technical account should be created to be used with the API.
- This technical account must have permissions to enroll and revoke SSL certificates on the desired certificate profiles (superadmin role).

Limitations

- Only the following fields are managed: `commonName (as cn, for SMIME certs)`, `contactEmail (as requester email address)`, `OU (only one)` and `subjectAltName DNS (for SSL certs)` and `RFC822Name (for SMIME)`.
- For multi-valued fields (SAN DNS), if more data items are provided than configured in ECS for the given certificate type, the exceeding items will be ignored.
- All limitations induced by the use of the ECS REST Connector.

Create the PKI connector

1. Log in to Horizon Administration Interface.

2. Access PKI from the drawer or card: **PKI › PKI Connectors**.

3. Click on ⊕ .

4. Select the correct PKI type.

5. Click on the next button

General tab

6. Fill in the common mandatory fields:

- **Connector Name*** *(string input)*:
  Choose a meaningful connector name allowing to identify the mapping between the PKI and the Certificate Profile. It must be unique and must not contain spaces.
- **Proxy** *(string select)*:
  If the PKI is not directly reachable from Horizon, you can set up an HTTP/HTTPS proxy to properly forward the traffic.
- **PKI Queue** *(string select)*:
  The PKI Queue used to manage the PKI Requests (enrollment, revocation).
- **Timeout** *(finite duration)*:
  Represents a predefined interval of time without a PKI response, when the time has passed "Horizon" will cease trying to establish the communication. Must be a valid finite duration.

7. Click on the next button

Details tab

8. Fill in all mandatory fields:

- **Technical account credentials*** *(select)*:
  Select `Login` credentials containing the technical account login/password.
- **Certificate Type** *(select)*:
  Select the Certificate Type to issue.
- **Requester's default email*** *(string input)*:
  Enter the requester default email address.
- **Requester's name** *(string input)*:
  Enter the requester name to register.
- **Requester's phone** *(string input)*:
  Enter the requester phone to register.
- **Certificate lifetime** *(finite duration)*: Enter Certificate lifetime, in days. For `SMIME_ENT` it is the number of years. The default value is set to 90 days.
- **Client ID** *(int)*:
  Enter Client ID. The default value is set to 1.

9. Click on the next button.

Authentication tab

10. Fill in the PKI-authentication fields:

- **Authentication Credentials*** *(select)*:
  Select `Certificate` credentials containing the authentication certificate used to connect to the PKI.

11. Click on the save button.

You can edit ✎, duplicate ⧉ or delete 🗑 the Entrust Certificate Services PKI connector.

4.2.8. Eviden IDCA

Prerequisites

- A certificate profile should be created.
- An authentication certificate should be issued for Horizon, and it should be given certificate issuance and revocation permissions on the aforementioned certificate profile.

Limitations

Create the PKI connector

1. Log in to Horizon Administration Interface.

2. Access PKI from the drawer or card: **PKI › PKI Connectors**.

3. Click on ⊕ .

4. Select the correct PKI type.

5. Click on the next button

General tab

6. Fill in the common mandatory fields:

- **Connector Name*** *(string input)*:
  Choose a meaningful connector name allowing to identify the mapping between the PKI and the Certificate Profile. It must be unique and must not contain spaces.
- **Proxy** *(string select)*:
  If the PKI is not directly reachable from Horizon, you can set up an HTTP/HTTPS proxy to properly forward the traffic.
- **PKI Queue** *(string select)*:
  The PKI Queue used to manage the PKI Requests (enrollment, revocation).
- **Timeout** *(finite duration)*:
  Represents a predefined interval of time without a PKI response, when the time has passed "Horizon" will cease trying to establish the communication. Must be a valid finite duration.

7. Click on the next button

Details tab

8. Fill in all mandatory fields:

- **IDCA RA Connector URL*** *(string input)*:
  Must point to the "RA" connector URL.
- **IDCA Certificate template name*** *(string input)*:
  The IDCA certificate template to use.
- **IDCA partition** *(string input)*:
  Specify a partition (if used).

9. Click on the next button.

Authentication tab

10. Fill in the PKI-authentication fields:

- **Authentication Credentials*** *(select)*:
  Select `Certificate` credentials containing the authentication certificate used to connect to the PKI.

11. Click on the save button.

You can edit ✎ , duplicate ⧉ or delete 🗑 the OpenTrust PKI connector.

4.2.9. EverTrust integrated CA

Create the PKI connector

1. Log in to Horizon Administration Interface.

2. Access PKI from the drawer or card: **PKI › PKI Connectors**.

3. Click on ⊕ .

4. Select the correct PKI type.

5. Click on the next button

General tab

6. Fill in the common mandatory fields:

- **Connector Name*** *(string input)*:
  Choose a meaningful connector name allowing to identify the mapping between the PKI and the Certificate Profile. It must be unique and must not contain spaces.
- **Proxy** *(string select)*:

If the PKI is not directly reachable from Horizon, you can set up an HTTP/HTTPS proxy to properly forward the traffic.

- **PKI Queue** *(string select)*:
  The PKI Queue used to manage the PKI Requests (enrollment, revocation).
- **Timeout** *(finite duration)*:
  Represents a predefined interval of time without a PKI response, when the time has passed "Horizon" will cease trying to establish the communication. Must be a valid finite duration.

7. Click on the next button

Details tab

8. Fill in all mandatory fields:

- **Certificate Type*** *(select)*:
  Specify the certificate type to issue.
- **Signing algorithm*** *(select)*:
  Specify the signing algorithm.
- **CA Certificate** *(string input)*:
  Enter CA certificate.
- **CA Key** *(string input)*:
  Enter CA key.
- **CRL save path** *(string input)*:
  Path to save the CRL on the Horizon server.
- **CRL lifetime** *(finite duration)*:
  CRL lifetime in days. Must be a valid finite duration.
- **Certificate Back Date** *(finite duration)*:
  Certificate Back Date. Must be a valid duration.
- **Check Proof of Possession** *(boolean)*

9. Click on the save button.

You can edit ✎ , duplicate ⧉ or delete 🗑 the EverTrust integrated CA PKI connector.

4.2.10. EverTrust Stream CA

Prerequisites

- A certificate template should be created in Stream for Horizon to enroll certificates upon.
- A dedicated Horizon account should be created in Stream and should have all lifecycle permissions on the desired CA. The credentials of this account should be either login and password or a PKCS#12 authentication certificate.

Create the PKI connector

1. Log in to Horizon Administration Interface.

2. Access PKI from the drawer or card: **PKI › PKI Connectors**.

3. Click on ⊕ .

4. Select the correct PKI type.

5. Click on the next button

General tab

6. Fill in the common mandatory fields:

- **Connector Name*** *(string input)*:
  Choose a meaningful connector name allowing to identify the mapping between the PKI and the Certificate Profile. It must be unique and must not contain spaces.
- **Proxy** *(string select)*:
  If the PKI is not directly reachable from Horizon, you can set up an HTTP/HTTPS proxy to properly forward the traffic.
- **PKI Queue** *(string select)*:
  The PKI Queue used to manage the PKI Requests (enrollment, revocation).
- **Timeout** *(finite duration)*:
  Represents a predefined interval of time without a PKI response, when the time has passed "Horizon" will cease trying to establish the communication. Must be a valid finite duration.

7. Click on the next button

8. Fill all mandatory fields:

- **Endpoint*** *(string input)*:
  Fill in the Stream endpoint url.
- **Template*** *(sting input)*:
  Fill in the Stream certificate template to enroll upon.
- **CA** *(string input)*:
  Fill in the Stream CA enrolling certificate (internal name).

9. Click on the next button.

Authentication tab

- **Authentication Credentials*** *(select)*:
  Select `Certificate` credentials containing the authentication certificate used to connect to the PKI, or `Credentials` containing the dedicated Horizon account on Stream.

10. Click on the save button.

You can edit ✎ , duplicate ⧉ or delete 🗑 the Evertrust Stream PKI connector.

4.2.11. FISId PKI

Create the PKI connector

1. Log in to Horizon Administration Interface.

2. Access PKI from the drawer or card: **PKI › PKI Connectors**.

3. Click on ⊕ .

4. Select the correct PKI type.

5. Click on the next button

General tab

6. Fill in the common mandatory fields:

- **Connector Name*** *(string input)*:
  Choose a meaningful connector name allowing to identify the mapping between the PKI and the Certificate Profile. It must be unique and must not contain spaces.
- **Proxy** *(string select)*:
  If the PKI is not directly reachable from Horizon, you can set up an HTTP/HTTPS proxy to properly forward the traffic.
- **PKI Queue** *(string select)*:
  The PKI Queue used to manage the PKI Requests (enrollment, revocation).
- **Timeout** *(finite duration)*:
  Represents a predefined interval of time without a PKI response, when the time has passed "Horizon" will cease trying to establish the communication. Must be a valid finite duration.

7. Click on the next button

Details tab

8. Fill in all mandatory fields:

- **FISId endpoint URL*** *(string input)*:
  URL to access the API.
- **Template ID*** *(int)*:
  Enter the template ID.
- **Default owner ID*** *(string input)*:
  Enter a default owner ID.
- **Authentication domain ID*** *(int)*:
  Enter an authentication domain ID.

- **Owner groups** *(string input)*:
  Enter one or several, separated by commas
- **To delete after revocation** *(boolean)*:
  The default value is set to false.

9. Click on the next button.

Authentication tab

10. Fill in the PKI-authentication fields:

- **API Key*** *(select)*:
  Select `API Token` credentials containing the API Key.

11. Click on the save button.

You can edit ✎ , duplicate ⧉ or delete 🗑 the FISId PKI connector.

4.2.12. GlobalSign Atlas PKI

Create the PKI connector

1. Log in to Horizon Administration Interface.

2. Access PKI from the drawer or card: **PKI › PKI Connectors**.

3. Click on ⊕ .

4. Select the correct PKI type.

5. Click on the next button

General tab

6. Fill in the common mandatory fields:

- **Connector Name*** *(string input)*:
  Choose a meaningful connector name allowing to identify the mapping between the PKI and the Certificate Profile. It must be unique and must not contain spaces.
- **Proxy** *(string select)*:
  If the PKI is not directly reachable from Horizon, you can set up an HTTP/HTTPS proxy to properly forward the traffic.
- **PKI Queue** *(string select)*:
  The PKI Queue used to manage the PKI Requests (enrollment, revocation).
- **Timeout** *(finite duration)*:
  Represents a predefined interval of time without a PKI response, when the time has passed "Horizon" will cease trying to establish the communication. Must be a valid finite duration.

7. Click on the next button

Details tab

8. Fill in all mandatory fields:

- **Hash Algorithm** *(select)*:
  Select the hash algorithm for the certificate to be issue.
- **API Credentials*** *(select)*:
  Select `Login` credentials containing the key and password that allows to authenticate against GlobalSign Atlas API.
- **Certificate Usage** *(select)*:
  Select a usage from the drop down list.
- **Retry Interval** *(finite duration)*:
  The default value is set to 3 seconds.

9. Click on the next button.

Authentication tab

10. Fill in the PKI-authentication fields:

- **Authentication Credentials*** *(select)*:
  Select `Certificate` credentials containing the authentication certificate used to connect to the PKI.

10. Click on the save button.

You can edit ✎ , duplicate ⧉ or delete 🗑 the GlobalSign Atlas PKI connector.

4.2.13. GlobalSign MSSL PKI

Prerequisites

- A technical account should be created.
- This technical account must have permissions to enroll and revoke SSL certificates on the desired domain.

Limitations

- Only the following fields are managed: `contactEmail` and `subjectAltName DNS`.
- For multi-valued fields (SAN DNS), if more data items are provided than configured in GlobalSign MSSL for the given "Product", the exceeding items will be ignored.
- All limitations induced by the use of the GlobalSign MSSL SOAP Connector.

Create the PKI connector

1. Log in to Horizon Administration Interface.

2. Access PKI from the drawer or card: **PKI › PKI Connectors**.

3. Click on ⊕ .

4. Select the correct PKI type.

5. Click on the next button

General tab

6. Fill in the common mandatory fields:

- **Connector Name*** *(string input)*:
  Choose a meaningful connector name allowing to identify the mapping between the PKI and the Certificate Profile. It must be unique and must not contain spaces.
- **Proxy** *(string select)*:
  If the PKI is not directly reachable from Horizon, you can set up an HTTP/HTTPS proxy to properly forward the traffic.
- **PKI Queue** *(string select)*:
  The PKI Queue used to manage the PKI Requests (enrollment, revocation).
- **Timeout** *(finite duration)*:
  Represents a predefined interval of time without a PKI response, when the time has passed "Horizon" will cease trying to establish the communication. Must be a valid finite duration.

7. Click on the next button

Details tab

8. Fill in all mandatory fields:

- **GlobalSign endpoint*** *(string select)*:
  Select from the drop-down list: the value must be "prod" for GlobalSign Production endpoint or "test" for the test environment.
- **GlobalSign profile ID*** *(string input)*:
  To be retrieved from the URL in the GlobalSign MSSL console.
- **GlobalSign domain ID*** *(string input)*:
  The ID of the domain to manage. Displayed in the GlobalSign MSSL console.
- **Certificate validity** *(int input)*:
  Certificate validity in months.
- **Default email address** *(string input)*:
  Choose a default email address.
- **Default phone number** *(string input)*:
  Choose a default phone number.

- **Interval before retrying to retrieve certificate** *(finite duration)*:
  The default value is set to 9 seconds.

9. Click on the next button.

Authentication tab

10. Fill in the PKI-authentication fields:

- **Technical account credentials*** *(select)*:
  Select `Login` credentials containing the login/password of the account created in GlobalSign MSSL.

11. Click on the save button.

You can edit ✎ , duplicate ⧉ or delete 🗑 the GlobalSign MSSL PKI connector.

4.2.14. MetaPKI

Prerequisites

Endpoint issuing CA

Create the PKI connector

1. Log in to Horizon Administration Interface.

2. Access PKI from the drawer or card: **PKI › PKI Connectors**.

3. Click on ⊕ .

4. Select the correct PKI type.

5. Click on the next button

General tab

6. Fill in the common mandatory fields:

- **Connector Name*** *(string input)*:
  Choose a meaningful connector name allowing to identify the mapping between the PKI and the Certificate Profile. It must be unique and must not contain spaces.
- **Proxy** *(string select)*:
  If the PKI is not directly reachable from Horizon, you can set up an HTTP/HTTPS proxy to properly forward the traffic.
- **PKI Queue** *(string select)*:
  The PKI Queue used to manage the PKI Requests (enrollment, revocation).
- **Timeout** *(finite duration)*:

Represents a predefined interval of time without a PKI response, when the time has passed "Horizon" will cease trying to establish the communication. Must be a valid finite duration.

7. Click on the next button

Details tab

8. Fill in all mandatory fields:

- **Endpoint*** *(string input)*:
  The MetaPKI Endpoint.
- **Endpoint Issuing CA*** *(string select)*:
  Select the CA that will be issuing the certificates for this connector (from the imported Horizon CAs)
- **Profile*** *(string input)*:
  Example: Applications_Auth_Client_Serveur_SSL.
- **Profile Cle*** *(string input)*:
  Example: Serveur_SSL
- **Workflow*** *(string input)*:
  Example: S_LOCAL_SOFT
- **Form Porteur Name** *(string input)*
- **Valid Days** *(finite duration)*
  Certificate lifetime in days (must be a valid finite duration).

9. Click on the next button.

Authentication tab

10. Fill in the PKI-authentication fields:

- **Authentication Credentials*** *(select)*:
  Select `Certificate` credentials containing the authentication certificate used to connect to the PKI.

11. Click on the save button.

You can edit ✎ , duplicate ⧉ or delete 🗑 the MetaPKI PKI connector.

4.2.15. Microsoft Active Directory Certificate Services PKI

Setup of the ADCS Connector

To set up a Microsoft Active Directory Certificate Services Connector (EverTrust ADCS Connector), you will need to have a machine running on Windows Server 2016+ with .NET 4.5.2 or later installed.

NOTE	The connector can be installed on the ADCS server itself or on another machine in

> the same domain. For the latter, you will need to copy the `C:\Windows\System32\certadm.dll` file from the ADCS server to that server at the same place and call `regsvr32 C:\Windows\System32\certadm.dll` from an elevated command prompt.

If all the pre-requisites are met then follow these steps:

1. Download the MSI installer from the EverTrust Repo (in the horizon-win folder) on the machine you wish to install it onto;

2. Start the setup and follow the installation wizard;

3. Once completed, enroll a TLS Web Server certificate with a SAN DNS that will have the DNS name you are going to use for this ADCS machine and import it in the certificate store of the ADCS machine;

4. Retrieve the hash of that certificate through certlm.msc. Be careful as some special characters may be copied alongside with the hash, so ensure that you get rid of them should they be present;

5. Edit the `C:\Program Files\EverTrust\ADCSConnector\EverTrustADCSConnector.exe.config` file and paste the previously copied hash to be the value of the "CertHash" line, then save the file;

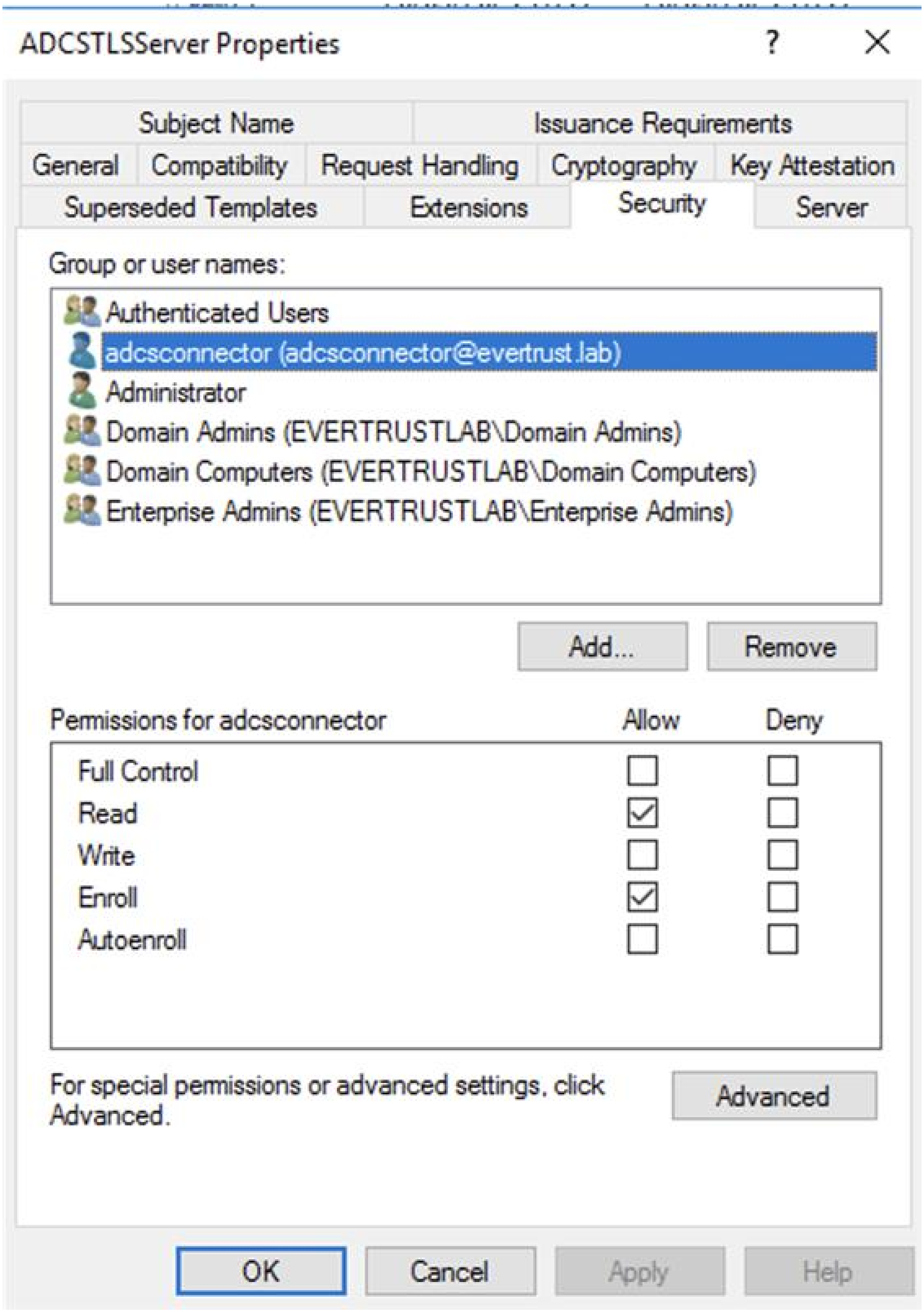
6. Ensure that the port 4443 is opened in the firewall of this machine and that the machine can indeed be reached from the Horizon machine;

7. Using `services.msc`, start the "EverTrust ADCS Connector" service. To see whether the service started successfully, start Internet Explorer and go to `https://localhost:4443/api/certificate`. This should download a json file that says "OK" if everything is good;

8. Create a new certificate template on the ADCS (or use an existing one) that the connector will use to enroll the certificates;

9. Create a technical account that needs to be able to log-in on the machine where the connector is installed;

9.1 Give it the right to enroll on the previously created template;



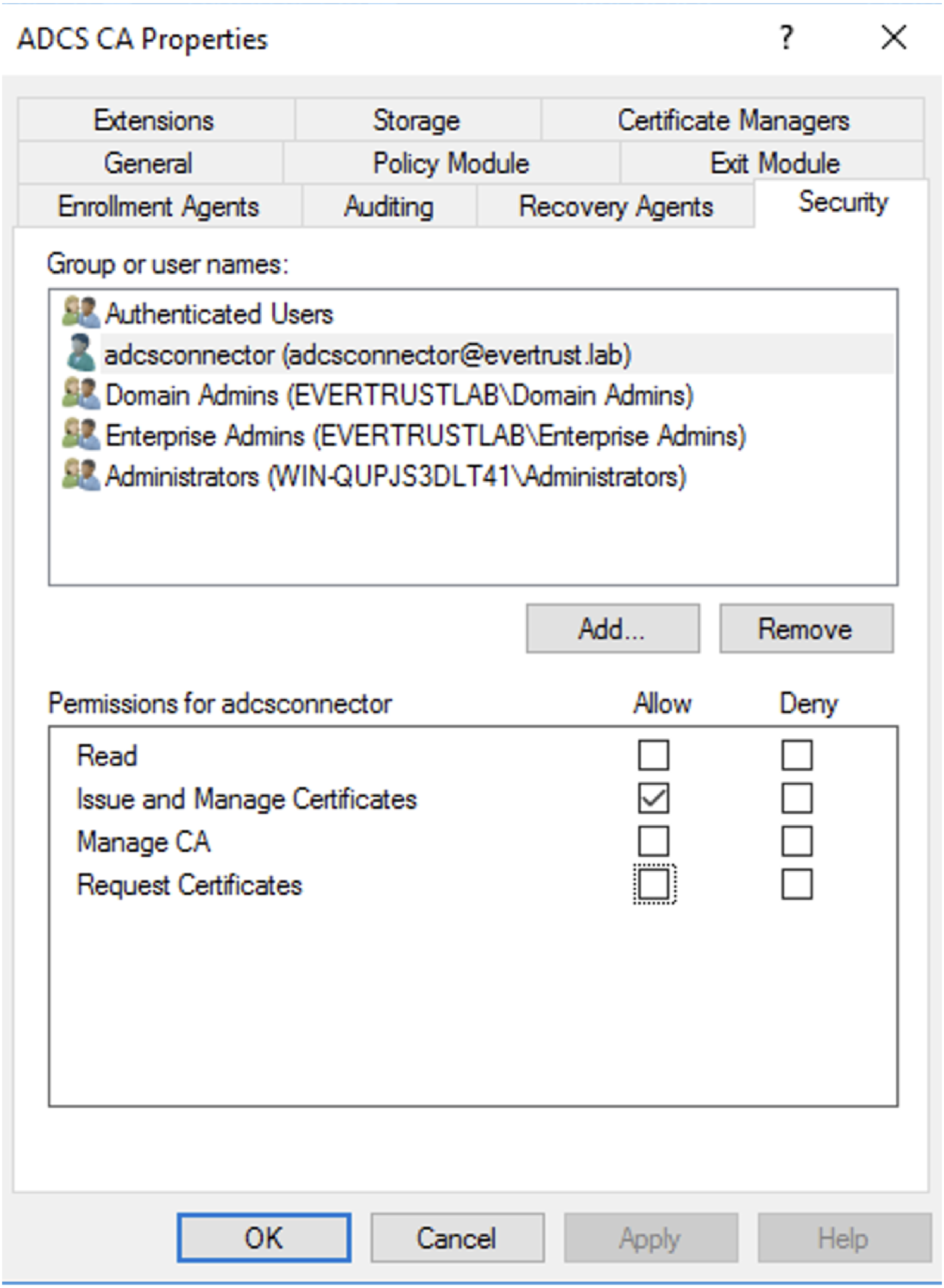
Group or user names:

- Authenticated Users
- adcsconnector (adcsconnector@evertrust.lab)
- Administrator
- Domain Admins (EVERTRUSTLAB\Domain Admins)
- Domain Computers (EVERTRUSTLAB\Domain Computers)
- Enterprise Admins (EVERTRUSTLAB\Enterprise Admins)

Permissions for adcsconnector	Allow	Deny
Full Control	□	□
Read	☑	□
Write	□	□
Enroll	☑	□
Autoenroll	□	□

For special permissions or advanced settings, click Advanced.

9.2 Give it the right to `Issue and Manage Certificates` on the ADCS;



Group or user names:

- Authenticated Users
- adcsconnector (adcsconnector@evertrust.lab)
- Domain Admins (EVERTRUSTLAB\Domain Admins)
- Enterprise Admins (EVERTRUSTLAB\Enterprise Admins)
- Administrators (WIN-QUPJS3DLT41\Administrators)

Permissions for adcsconnector	Allow	Deny
Read	□	□
Issue and Manage Certificates	☑	□
Manage CA	□	□
Request Certificates	□	□

10. Create an enrollment agent certificate and export it as PKCS#12. This certificate will be the one used to sign the CMC messages from Horizon.

Now that the configuration on ADCS-side is over, we can tackle the configuration on Horizon side.

Creating the ADCS PKI Connector in Horizon

The previous steps are considered as pre-requisites to continue the setup. If you haven't yet configured the ADCS Connector on the ADCS side, please refer to the Setup of the ADCS Connector. The rest of this section assumes that the EverTrust ADCS Connector is installed and correctly set-up on the ADCS side.

Limitations

- All limitations induced by the use of ADCS.

Create the PKI connector

1. Log in to Horizon Administration Interface.

2. Access PKI from the drawer or card: **PKI › PKI Connectors**.

3. Click on ⊕ .

4. Select the correct PKI type.

5. Click on the next button

General tab

6. Fill in the common mandatory fields:

- **Connector Name*** *(string input)*:
  Choose a meaningful connector name allowing to identify the mapping between the PKI and the Certificate Profile. It must be unique and must not contain spaces.
- **Proxy** *(string select)*:
  If the PKI is not directly reachable from Horizon, you can set up an HTTP/HTTPS proxy to properly forward the traffic.
- **PKI Queue** *(string select)*:
  The PKI Queue used to manage the PKI Requests (enrollment, revocation).
- **Timeout** *(finite duration)*:
  Represents a predefined interval of time without a PKI response, when the time has passed "Horizon" will cease trying to establish the communication. Must be a valid finite duration.

7. Click on the next button

Details tab

8. Fill in all mandatory fields:

- **Endpoint*** *(string input)*:
  URL to access the machine where the ADCS connector is running on port 4443.

- **Active Directory Domain Netbios Name*** *(string input)*:
  The NETBIOS name of the Active Directory domain where to find the technical user and the ADCS server.

- **Profile*** *(string input)*:
  The technical name of the template that you created at step 8 of the Setup of the ADCS Connector section. Example: WebServer

- **CA Config*** *(string input)*:
  The `CaConfig` string, as given out by `certutil -getconfig` for the considered ADCS CA. It's usually in the form `<ADCS Hostname>\<CA CommonName>`

9. Click on the next button.

Authentication tab

10. Fill in the ADCS authentication fields:

- **Enrollment agent certificate*** *(select)*:
  Select `Certificate` credentials containing the PKCS#12 enrollment agent certificate that was exported at step 10 of the Setup of the ADCS Connector section.

- **MS ADCS user account*** *(select)*:
  Select `Login` credentials containing the username and password of the technical account created at step 9 of the Setup of the ADCS Connector section.

> **NOTE**
>
> Specify only the username of the technical account on the ADCS machine, without the Netbios domain name.
> For example, in **PKI\Technical** do not include the **PKI**\ part.

11. Click on the save button.

You can edit ✎, duplicate ⧉ or delete 🗑 the Microsoft Active Directory Certificate Services PKI connector.

4.2.16. Nameshield

Prerequisites

- A dedicated Horizon account with enroll and revoke permissions must be set up
- An authentication token must be obtained using Nameshield's procedure

Limitations

- For decentralized enrollment, CSR data cannot be modified by user inputs
- CSR must contain at least a FQDN CN and DNS SAN

Create the PKI connector

1. Log in to Horizon Administration Interface.

2. Access PKI from the drawer or card: **PKI › PKI Connectors**.

3. Click on ⊕ .

4. Select the correct PKI type.

5. Click on the next button

General tab

6. Fill in the common mandatory fields:

- **Connector Name*** *(string input)*:
  Choose a meaningful connector name allowing to identify the mapping between the PKI and the Certificate Profile. It must be unique and must not contain spaces.
- **Proxy** *(string select)*:
  If the PKI is not directly reachable from Horizon, you can set up an HTTP/HTTPS proxy to properly forward the traffic.
- **PKI Queue** *(string select)*:
  The PKI Queue used to manage the PKI Requests (enrollment, revocation).
- **Timeout** *(finite duration)*:
  Represents a predefined interval of time without a PKI response, when the time has passed "Horizon" will cease trying to establish the communication. Must be a valid finite duration.

7. Click on the next button

8. Fill all mandatory fields:

- **Environment*** *(select)*:
  Fill in the environment of the nameshield instance (Production or Testing).
- **Organization ID*** *(number input)*:
  Fill in the Nameshield Organization ID.
- **Product ID*** *(number input)*:
  Fill in the Nameshield Product ID.
- **Customer ID*** *(number input)*:
  Fill in the Nameshield Customer ID.

9. Click on the next button.

Authentication tab

- **API Key*** *(select)*:
  Select `API Token` credentials containing the authentication token used to connect to Nameshield.

10. Click on the save button.

You can edit ✎ , duplicate ⧉ or delete 🗑 the Nameshield connector.

4.2.17. Nexus Certificate Manager PKI

Prerequisites

- A certificate procedure and a token procedure should be created.
- An authentication certificate should be issued for Horizon, and it should be given certificate issuance and revocation permissions on the aforementioned token procedure.
- Nexus Endpoint CA

Limitations

- Only the following fields are managed: `commonName, UID, OU, O, C` and `subjectAltNames DNS, IPaddress, RFC822Name` and `msUPN`.
- For multi-valued fields (SAN DNS, RFC822Name and IP address), if more data items are provided than configured in Nexus CM Procedure, the exceeding items will be ignored.
- All limitations induced by the use of the Nexus CM SDK.

Create the PKI connector

1. Log in to Horizon Administration Interface.

2. Access PKI from the drawer or card: **PKI** › **PKI Connectors**.

3. Click on ⊕ .

4. Select the correct PKI type.

5. Click on the next button

General tab

6. Fill in the common mandatory fields:

- **Connector Name*** *(string input)*:
  Choose a meaningful connector name allowing to identify the mapping between the PKI and the Certificate Profile. It must be unique and must not contain spaces.
- **Proxy** *(string select)*:
  If the PKI is not directly reachable from Horizon, you can set up an HTTP/HTTPS proxy to properly forward the traffic.
- **PKI Queue** *(string select)*:
  The PKI Queue used to manage the PKI Requests (enrollment, revocation).

- **Timeout** *(finite duration)*:
  Represents a predefined interval of time without a PKI response, when the time has passed "Horizon" will cease trying to establish the communication. Must be a valid finite duration.

7. Click on the next button

Details tab

8. Fill all mandatory fields:

- **Nexus CM DNS name*** *(string input)*:
  URL to access the Nexus Certificate Manager. Two modes are available:
  - Direct connection, you can specify the IP:PORT
  - Using PGWY, you will need to specify the PGWY url as following https://<pgwy_url>/sdkproxy
- **Nexus endpoint CA*** *(select)*:
  Select the endpoint CA.
- **Nexus CM Certificate procedure name*** *(string input)*:
  The token procedure name to use.
  Should point to the appropriate certificate procedure, and must be on PKCS#10 format.

9. Click on the next button.

Authentication tab

10. Fill in the PKI-authentication fields:

- **Authentication Credentials*** *(select)*:
  Select `Certificate` credentials containing the authentication certificate used to connect to the PKI.

11. Click on the save button.

You can edit ✎ , duplicate ⧉ or delete 🗑 the Nexus Certificate Manager PKI connector.

4.2.18. OpenTrust PKI

Prerequisites

- A certificate profile should be created.
- An authentication certificate should be issued for Horizon, and it should be given certificate issuance and revocation permissions on the aforementioned certificate profile.

Limitations

- Only the following fields are managed: `commonName`, `userID`, `serialNumber`, `organizationalUnit`, `organization`, `country`, `adminEmail` or `contactEmail`, `msCertTemplateName` and `subjectAltNames`

`DNS`, `IPadress`, `RFC822Name`, `msUPN` and `msGUID`.

- For multi-valued fields (SAN DNS, IP address and RFC822Name), if more data items are provided than configured in OTPKI 'certificate template name', the exceeding items will be ignored.
- All limitations induced by the use of the RA SOAP Connector.

Create the PKI connector

1. Log in to Horizon Administration Interface.

2. Access PKI from the drawer or card: **PKI › PKI Connectors**.

3. Click on ⊕ .

4. Select the correct PKI type.

5. Click on the next button

General tab

6. Fill in the common mandatory fields:

- **Connector Name*** *(string input)*:
  Choose a meaningful connector name allowing to identify the mapping between the PKI and the Certificate Profile. It must be unique and must not contain spaces.
- **Proxy** *(string select)*:
  If the PKI is not directly reachable from Horizon, you can set up an HTTP/HTTPS proxy to properly forward the traffic.
- **PKI Queue** *(string select)*:
  The PKI Queue used to manage the PKI Requests (enrollment, revocation).
- **Timeout** *(finite duration)*:
  Represents a predefined interval of time without a PKI response, when the time has passed "Horizon" will cease trying to establish the communication. Must be a valid finite duration.

7. Click on the next button

Details tab

8. Fill in all mandatory fields:

- **OTPKI RA Connector URL*** *(string input)*:
  Must point to the "RA" connector URL.
- **OTPKI Certificate template name*** *(string input)*:
  The OTPKI certificate template to use.
- **OTPKI zone** *(string input)*:
  Specify a zone (if used).
- **Contact email mapping** *(string input)*:

Allows to change the default fields names accordingly to certificate profiles.

- **SAN DNS mapping** *(string input)*:
  Allows to change the default fields names accordingly to certificate profiles.
- **SAN Email mapping** *(string input)*:
  Allows to change the default fields names accordingly to certificate profiles.
- **UID mapping** *(string input)*:
  Allows to change the default fields names accordingly to certificate profiles.

9. Click on the next button.

Authentication tab

10. Fill in the PKI-authentication fields:

- **Authentication Credentials*** *(select)*:
  Select `Certificate` credentials containing the authentication certificate used to connect to the PKI.

11. Click on the save button.

You can edit ✎, duplicate ⧉ or delete 🗑 the OpenTrust PKI connector.

4.2.19. Sectigo SCM PKI

Prerequisites

- For publicly trusted certificates, you need to validate the domain(s) for which you will issue certificates prior to their issuance.
- You need to retrieve the `customerUri` and the `organizationId` from Sectigo SCM.
- You need to create a technical account with appropriate permissions including the `allow ssl auto approve` permission. You need to set a password for the technical account.

Limitations

- Only the `subjectAltName DNS` field is managed.
- The certificate Subject DN will be set to whatever is specified in the PKCS#10 CSR.
- All limitations induced by the use of the Sectigo SCM REST Connector.

Create the PKI connector

1. Log in to Horizon Administration Interface.

2. Access PKI from the drawer or card: **PKI › PKI Connectors**.

3. Click on ⊕.

4. Select the correct PKI type.

5. Click on the next button

General tab

6. Fill in the common mandatory fields:

- **Connector Name*** *(string input)*:
  Choose a meaningful connector name allowing to identify the mapping between the PKI and the Certificate Profile. It must be unique and must not contain spaces.
- **Proxy** *(string select)*:
  If the PKI is not directly reachable from Horizon, you can set up an HTTP/HTTPS proxy to properly forward the traffic.
- **PKI Queue** *(string select)*:
  The PKI Queue used to manage the PKI Requests (enrollment, revocation).
- **Timeout** *(finite duration)*:
  Represents a predefined interval of time without a PKI response, when the time has passed "Horizon" will cease trying to establish the communication. Must be a valid finite duration.

7. Click on the next button

Details tab

8. Fill in all mandatory fields:

- **Customer URI*** *(string input)*:
  Enter the Customer URI. An integer is expected.
- **Organization ID*** *(int input)*:
  Enter the Organization ID.
- **Profile (Certificate Type)*** *(string input)*:
  Enter the Profile (Certificate Type). An integer is expected.
- **Retry interval** *(finite duration)*:
  Predefined interval of time before retrying to retrieve the certificate from Sectigo. Must be a valid finite duration. No default value is set.
- **Valid Days** *(finite duration)*:
  Certificate validity duration in days. Must be a valid finite duration. No default value is set.

9. Click on the next button.

Authentication tab

10. Fill in the PKI-authentication fields:

- **Authentication credentials*** *(select)*:
  Select `Login` credentials containing your Sectigo SCM login and password.

11. Click on the save button.

You can edit ✎ , duplicate ⧉ or delete 🗑 the Sectigo SCM PKI connector.

5. Security

5.1. Local Accounts

This section details how to configure the EverTrust Horizon local accounts and set their password.

NOTE

Local accounts are useful to create technical accounts, such as required by `horizon-cli` for some scenarios (e.g. Scan/Discovery)

5.1.1. How to create local accounts

1. Log in to Horizon Administration Interface.

2. Access Local accounts from the drawer or card: **Security** › **Access Management** › **Local Accounts**.

3. Click on ⊕ .

4. Fill in the mandatory fields.

- **Identifier*** *(string input)*:
  Enter a meaningful identifier for the account holder. It will be used as a login to access to the solution.
- **Name** *(string input)*:
  Enter a meaningful name for the account holder.
- **Email** *(string input)*:
  Enter the account holder email.

5. Click on the save button.

5.1.2. How to set a password to a local account

1. Once a local account is created. Click on 🔒 .

2. Fill in the mandatory fields.

- **Password*** *(string input)*:
  Set a password.
- **Confirm password*** *(string input)*:
  Confirm the password.

3. Click on the save button.

You can edit ✎ or delete 🗑 a local account. You can manage 🔒 a local account password.

NOTE

You can not delete yourself from local accounts.

5.2. Authorization

This section details how to configure the permissions granted to an account, either directly or through a configured role.

5.2.1. Prerequisites

According to the context, you might need to set up:

- Roles
- Local accounts

5.2.2. How to add an authorization manually or from a certificate

1. Log in to Horizon Administration Interface.

2. Access Authorizations from the drawer or card: **Security** › **Access Management** › **Authorizations**.

3. Click on ⊕ .

4. Click on Add Authorization Manually

5. Fill the mandatory fields.

- Either:
  - Fill in an **Identifier*** *(string input or import)*:
    Enter a meaningful identifier. It can be either a local account identifier or an OpenID Connect identifier (usually email address).
  - Import a certificate by clicking on certificate button 🏅 .
- **Contact email** *(string input)*:
  Enter the contact email for the account.

6. Click on add button.

5.2.3. How to add an authorization from a search

1. Log in to Horizon Administration Interface.

2. Access Authorizations from the drawer or card: **Security** › **Access Management** › **Authorizations**.

3. Click on ⊕ .

4. Click on Search and Add Authorization

5. Fill one of the fields.

- **Identifier*** *(string input)*:
  Enter the identifier of the account to look for.
- **Email*** *(string input)*:
  Enter the email of the account to look for.

6. Click on search button.

7. Choose the identifier you want to add.

8. Click on add button.

You can update ✎ or delete 🗑 Authorization.

5.2.4. How to grant a permission

1. Click on 🔒 .

Role

2. Select a role previously created (if needed).

Team

3. Select a team previously created (if needed).

Configuration

You can build here a configuration permission. The permission follows the pattern: Section / Module / Right.

4. Click on add button.

5. Select a section, then a module, then a submodule if there is, and a right.

6. Click on add button (Don’t forget to save).

7. Click on the save button if you are done.

Lifecycle

You can build here a lifecycle permission. The permission follows the pattern: Module / Profile / Right. You can further restrict the permission by adding a filter from the "Horizon Permission Query Language".

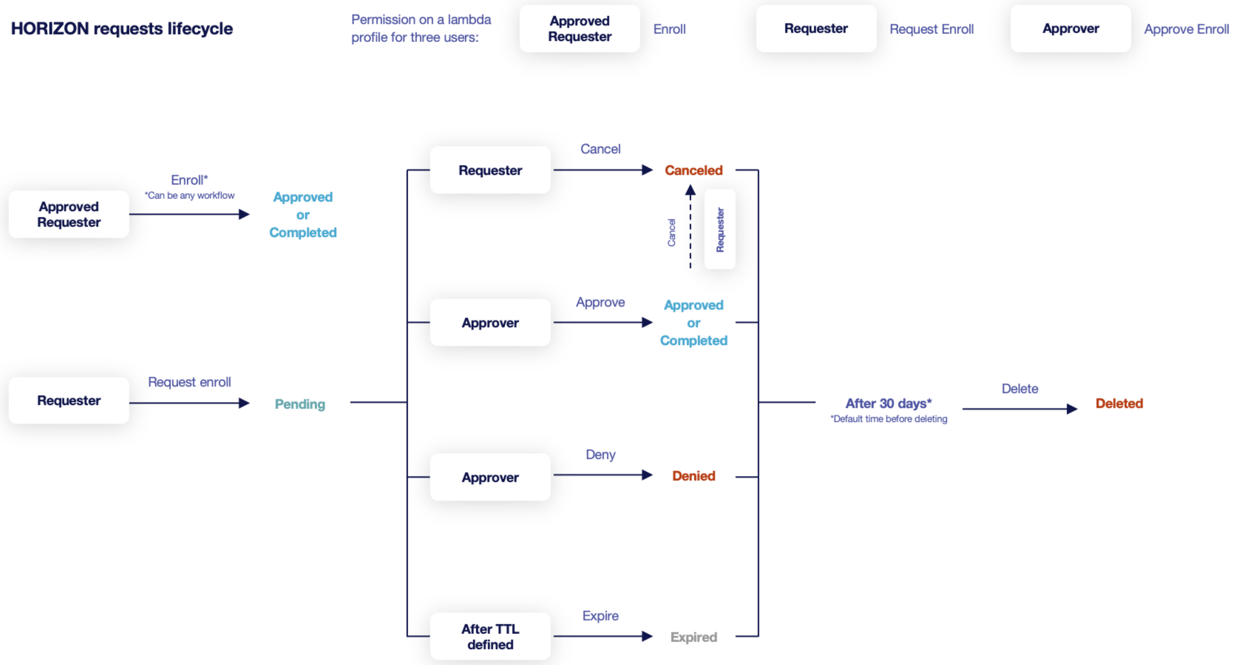
4. Click on add button.

5. Select a module, then a profile, and a right.

6. Click on add button. (don’t forget to save).

7. Click on the save button if you are done.

Horizon requests lifecycle:



HPQL

The *Horizon Permission Query Language* allows you to restrict lifecycle permissions on labels and team.

The following keywords are available:

Name	Value
x equals y	true if x 's value equals y
x contains y	true if x 's value contains y
x in y	true if x 's value is contained in y (array)
x matches y	true if x 's value matches y (regex)
x within y	true if x 's value matches a value in y (regex array)

These can be combined with the following keywords:

Name	Value
x and y	true if x and y are true
x or y	true if x or y are true
x not `expression` y	true if x `expression` y is false

Examples

To filter on the `myLabel` label:

```
label.myLabel equals "labelValue"
    => myLabel: label = false
    => myLabel: labelValue =  true
label.myLabel contains "label"
    => myLabel: label = true
    => myLabel: labelValue =  true
label.myLabel within [ "\d+", "other\d?Value" ]
    => myLabel: 12345 = true
    => myLabel: otherValue =  true
```

To filter on the team:

```
team matches "team[A-Z]"
    => team: teamA = true
    => team: bestTeam =  false
team in ["teamA", "teamB"]
    => team: teamA = true
    => team: bestTeam =  false
```

Discovery

You can build here a discovery permission. The permission follows the pattern: Module / Discovery campaign name / Right.

4. Click on add button.

5. Select a module, then a campaign, and a right.

6. Click on add button. (don't forget to save)

7. Click on the save button if you are done.

5.3. Roles

This section details how to configure the roles. Roles are groups of permissions that can be configured for authorizations.

5.3.1. How to create a role

1. Log in to Horizon Administration Interface.

2. Access Roles from the drawer or card: **Security › Access Management › Roles**.

3. Click on ⊕ .

4. Fill in at least the mandatory fields.

- **Name*** *(string input)*:
  Enter a meaningful name.
- **Description** *(string input)*:
  Enter a description.

6. Configuration permissions

7. Lifecycle permissions

8. Discovery permissions

9. Click on the save button.

You can get the list of members ☰ .

You can update ✎ or delete 🗑 the Role.

5.4. Teams

This section details how to configure teams. Teams are groups of horizon objects owner (certificates, requests) and does not define permissions.

5.4.1. How to create a team

1. Log in to Horizon Administration Interface.

2. Access Teams from the drawer or card: **Security** › **Access Management** › **Teams**.

3. Click on ⊕ .

4. Fill at least the mandatory fields.

- **Name*** *(string input)*:
  Enter a meaningful name.
- **Description** *(string input)*:
  Enter a description.
- **Contact email** *(string input)*:
  Enter a valid email.
- **Manager email** *(string input)*:
  Enter a valid email.
- **Messaging tool** *(select)*:
  Select one of Webhook Messaging tools supported

- **URL** *(string input)*:
  Enter the webhook messaging URL for the team(used by Groupware notifications)

5. Click on the save button.

You can get the list of members ☰.

You can update ✎ or delete 🗑 the Team.

5.5. Identity Providers Configuration

This section details how to configure Identity Providers. Identity Providers are going to be used by Horizon to verify the identity of an end-user based on the authentication performed by an external authorization server.

5.5.1. How to configure an Identity Provider

1. Log in to Horizon Administration Interface.

2. Access Identity Providers from the drawer or card: **Security** › **Access Management** › **Identity Providers**.

3. Click on ⊕.

General tab

4. Select an identity provider type. Currently only OpenID is supported

OpenID connect

5. Fill in all mandatory fields:

- **Name*** *(string input)*:
  Enter a meaningful identity provider name.
- **Provider metadata URL*** *(string input)*:
  Enter the OpenID Connect provider metadata URL.
- **Client ID*** *(string input)*:
  Identifier generated on the OpenID Connect IDP when setting up a new application (Horizon) to authenticate users on the identity provider.
- **Client Secret*** *(string input)*:
  Password associated to the aforementioned identifier (Client ID);
- **Scope*** *(string input)*:
  Scope used by Horizon during authentication on the identity provider to authorize access to user’s details.
- **Proxy** *(string select)*:

Proxy used to access Provider metadata URL, if any.

- **Timeout** *(finite duration)*:
  Timeout used for authentication on the identity provider. Must be a valid finite duration. By default 10 seconds.

- **Identifier Claim*** *(string input)*:
  Dynamic expression defining how to construct the identifier from the OpenID Connect claims. Claim names must be declared between `{{` and `}}` characters. For example, if the user identifier is contained in the `login` claim, then the configured value should be `{{login}}`.

- **Email Claim*** *(string input)*:
  Dynamic expression defining how to construct the user email from the OpenID Connect claims. Claim names must be declared between `{{` and `}}` characters. For example, if the user email is contained in the 'email' claim, then the configured value should be `{{email}}`. If the email is not available directly from the claims but can be computed from the 'login' claim by appending a domain, the configured value should be `{{login}}@evertrust.fr`.

- **Name Claim*** *(string input)*:
  Dynamic expression defining how to construct the username from the OpenID Connect claims. Claim names must be declared between `{{` and `}}` characters. For example, if the user name must be constructed as `family name, given name` and family name is available in the `family_name` claim, given name is available in the `given_name` claim, then the configured value should be `{{family_name}}, {{given_name}}`.

- **Enable*** *(boolean)*:
  Enable/Disable the identity provider.

- **Enabled on `UI`*** *(boolean)*:
  Enable/Disable the identity provider on user interface.

Languages tab

You can add more languages by clicking ⊕ .

- **Language*** *(select)*:
  Select a language. Supported languages are:
  - **en**: English
  - **fr**: French

- **Display Name** *(string input)*:
  Enter a display name. This will be the localized name of the provider on the login page.

- **Description** *(string input)*:
  Enter a description. This will be displayed in a tooltip when the provider is chosen on the login page.

You can delete 🗑 the localization.

6. Click on the save button.

You can update ✎ or delete 🗑 the Identity Provider.

> **CAUTION** You won’t be able to delete an Identity Provider if it is referenced in any other configuration element.

5.6. Credentials

This section details how to configure credentials. Credentials are where credentials for all integrations are regrouped.

5.6.1. How to create credentials

1. Log in to Horizon Administration Interface.

2. Access Credentials from the drawer or card: **Security › Credentials**.

3. Click on ⊕ .

4. Fill at least the mandatory fields.

- **Type*** *(select)*:
  Select the credentials type: `Certificate` for certificate based authentication, `Login` for login with password credentials or `API Token` for a single value secret (JSON or other).
- **Name*** *(string input)*:
  Enter a meaningful name.
- **Description** *(string input)*:
  Enter a description.
- **Expiration date** *(date input)*:
  Enter an expiration date. This will be taken from the certificate for `Certificate` credentials.
- **Expiration notifications** *(select)*:
  Select Email, Groupware or REST notifications on event `Credentials expiration` that will run on expiration. Notifications configured here will be sent by the internal monitoring action.
- Certificate:
  - **PKCS#12*** *(file select)*:
    Select your PKCS#12 file containing the authentication certificate and its key.
  - **PKCS#12 Password*** *(string input)*:
    Enter the password of the PKCS#12.
- Credentials:
  - **Login*** *(string input)*:
    Enter the account login.
  - **Password*** *(string input)*:
    Enter the account password.

- JSON Token:
  - **JSON Token*** *(string input)*:
    Enter the token.

5. Click on the save button.

You can update ✎ or delete 🗑 the Credentials.

5.7. Password Policies

This section details how to configure password policies that will be used by Horizon.

5.7.1. How to configure a Password Policy

1. Log in to Horizon Administration Interface.

2. Access Password Policies from the drawer or card: **Security › Password Policies**.

3. Click on ⊕ .

4. Fill in the mandatory fields.

- **Name***:
  Enter a meaningful password policy name;
- **Password range length*** *(int)*:
  Password length (0 is unlimited);
- **Minimum of lowercase** *(int)*:
  Minimum of lowercase characters in the password;
- **Minimum of uppercase** *(int)*:
  Minimum of uppercase characters in the password;
- **Minimum of digit** *(int)*:
  Minimum of digit in the password;
- **Minimum of special character** *(int)*:
  Minimum of special characters in the password;
- **Special characters accepted** *(string input)*:
  Whitelist of special characters accepted in the password.

5. Click on the save button.

You can update ✎ or delete 🗑 the Password Policy.

CAUTION You won’t be able to delete a Password Policy if it is referenced in any other configuration element.

5.8. SCIM

5.8.1. SCIM Introduction

This section refers to SCIM 2.0 integration with Horizon, used to provision users and groups in Horizon.

Description

SCIM (System for Cross-domain Identity Management) is an open standard protocol for automating the exchange of groups and users identity information between identity domains and Horizon, users and groups are synchronized between the two systems with a rich but simple set of operations:

- GET
- POST
- PUT
- PATCH
- DELETE

The SCIM protocol is detailed in the following RFCs:

- RFC7642 (System for Cross-domain Identity Management: Definitions, Overview, Concepts, and Requirements)
- RFC7643 (System for Cross-domain Identity Management: Core Schema)
- RFC7644 (System for Cross-domain Identity Management: Protocol)

NOTE Horizon does not support the full RFC, Horizon only supports the minimum of the RFC and ensures compatibility with Azure Ad and Okta.

Prerequisites

According to the context, you need:

- An application that is compatible with **SCIM 2.0**.
- Have users or groups configured in your identity manager for provisioning

Authentication with Horizon

- Have a bearer token or basic Auth

To build the bearer token you must encode in base 64 → Login:Password

Endpoint

SCIM 2.0 Base Url corresponds to: `https://<horizonUrl>/security/scim/<scimProfileName>/`

Limitations

Endpoints

List of endpoints supported:

- Users
- Groups
- ServiceProviderConfig
- ResourceTypes

Filters

List of operators supported for filtering:

- eq
- and
- ( )
- [ ]

List of attributes supported for filtering:

- userName
- displayName

Password

Horizon does not manage the password assignment.

Email

Horizon only have one email for SCIM user, it is the mail type in SCIM Profile.

SCIM User

The id of a SCIM User corresponds to the identifier of a Principal Info.

SCIM Group

Horizon does not support the creation and deletion of SCIM groups.

Supported Attributes

The list of objects and their representations :

SCIM User

- schemas
- userName
- id
- emails
- meta
- active

SCIM Group

- schemas
- id
- displayName
- members

Synchronization in Horizon

To synchronize between the SCIM groups and the roles and teams there is an object called a SCIM Profile. This object serves as an intermediary between SCIM and Horizon.

5.8.2. SCIM Profiles

This section details how to configure the SCIM profiles, it allows you to manage SCIM identity in Horizon.

How to create a SCIM Profile

1. Log in to Horizon Administration Interface.

2. Access SCIM Profiles from the drawer or card: **Security › SCIM Profiles**.

3. Click on ⊕ .

4. Fill at least the mandatory field.

- **Name*** *(string input)*:
  Enter a meaningful name.

5. Click on the save button.

You can update ✎ or delete 🗑 the SCIM Profile.

How to create a SCIM specific parameters

1. Log in to Horizon Administration Interface.

2. Access SCIM Profiles from the drawer or card: **Security › SCIM Profiles**.

3. Click on ⊕ .

4. Fill at least the mandatory field.

- **Name*** *(string input)*:
  Enter a meaningful name.

5. Fill the at least the optional field.

- **Mail type** *(string input)*:
  Enter a meaningful mail type. The mail type corresponds to the mail coming from a SCIM provider that must be synchronised in horizon. By default, the mail type is "work".

6. Click on ⊕ **Add a mapping**.

The mapping corresponds to the fields allowing synchronization between Horizon and SCIM provider.

7. Fill the SCIM group name, it is referred to the SCIM group coming from the SCIM provider that must be synchronised in horizon.

8. You must choose either a role or a team for the SCIM group, but **you cannot select both**.

You can add more mappings by clicking ⊕ .

10. Click on the save button.

You can update ✎ or delete 🗑 the SCIM Profile.

CAUTION You won’t be able to choose a role or team if it is referenced in any other Horizon user.

6. Notifications

6.1. Email

This section details how to configure the email notifications.

6.1.1. How to create an email notification

1. Log in to Horizon Administration Interface.

2. Access emails from the drawer or card: **Notifications › Emails**.

3. Click on ⊕ .

4. Fill in all mandatory fields.

- **Name*** *(string input)*:
  Enter a meaningful email notification name.
- **Event type*** *(select)*:
  Select the event type to notify (certificate or request).
- **Event*** *(select)*:
  Select the event to notify.
- **Retries in case of error** *(int)*:
  Select the number of times Horizon should retry to send the notification in case of error. The default value is set to 10.
- **From***: *(string input)*
  Enter the email address that will appear in the email "From" field.
- **To***: *(select multiple & input multiple)*
  Select one or several recipients. You may also enter an email address.
- **Subject*** *(string input)*:
  Enter the email subject. You may use dynamic attributes, that will be automatically replaced by the appropriate values upon email generation.
- **Body*** *(string input)*:
  Enter the email body. You may use dynamic attributes, that will be automatically replaced by the appropriate values upon email generation.
- **Is HTML** *(boolean)*:
  Sets whether the email body contains HTML code (true) or plain text (false). The default value is set to false.

NOTE

You can click on the "+" next to "How to use dynamic attributes" in order to get a range of possibilities from which one or more may be chosen.

*In case you selected a **Request** type event on any **Approval** event, or a **Certificate** event:*

- **Attachments** *(list)*:
  Sets whether to attach the certificate to the email notification and which format to use for the attached certificate (if any).
  - Attach certificate (PEM) attaches the certificate under PEM format
  - Attach bundle (PEM) attaches the certificate as well as the entire trust chain used to sign it in PEM format
  - Attach certificate (PKCS#7) attaches the certificate under PKCS#7 format
  - Attach bundle (PKCS#7) attaches the certificate as well as the entire trust chain used to sign it in PKCS#7 format
  - Attach certificate (DER) attaches the certificate under DER format

*In case you selected **Certificate Expiration**:*

- **Duration before certificate expiration causing the notification*** *(finite duration)*:
  Sets how long before certificate expiration the email notification should be sent. The default value is set to 5 days.
- **Run on renewed** *(boolean)*:
  Sets whether the expiration notification should be sent even though the certificate has been renewed. Default value is set to false (if the certificate has been renewed, the notification will not be sent).

*In case you selected as an Event **Enroll request Approval** or **Renew request Approval** or **Recover request Approval**:*

- **Attach PKCS#12** (set at false) *(boolean)*:
  Sets whether the certificate in PKCS#12 format (certificate + private key encrypted by password) should be attached to the email. The default value is set to false.
- **Send email if** *(select unique)*:
  Select either Always - Centralized (Horizon generates the private key) - Decentralized (a CSR is provided to Horizon). The default value is set to Always.

*In case you selected as an Event **Enroll request Pending** or **Renew request Pending** or **Revoke request Pending** or **Recover request Pending** or **Update request Pending** or **Migrate Request Pending**:*

- **Duration after request submission causing the notification*** *(finite duration)*:
  Duration after request submission causing the notification to be sent, in case the request was not approved in the meantime. The default value is set to 5 days.

6. Click on the save button.

You can edit ✎ , duplicate ⧉ or delete 🗑 the Email Notification .

6.2. Groupware

This section details how to configure the groupware notifications.

The supported groupwares are:

- Slack
- Mattermost
- Microsoft Teams

6.2.1. Prerequisites

You will need a webhook URL from the groupware tools in order to send notification:

- Slack
- Mattermost
- Microsoft Teams

6.2.2. How to create a Groupware notification

1. Log in to Horizon Administration Interface.

2. Access Groupware from the drawer or card: **Notifications › Groupware**.

3. Click on ⊕ .

4. Fill in all mandatory fields.

- **Name*** *(string input)*:
  Enter a meaningful email notification name.
- **Event type*** *(select)*:
  Select the event type to notify (certificate or request).
- **Event*** *(select)*:
  Select the event to notify.
- **Retries in case of error** *(int)*:
  Select the number of times Horizon should retry to send the notification in case of error. The default value is set to 10.
- **Timeout*** *(finite duration)*:
  The time before Horizon stop trying to connect to Webhook or Proxy.
- **Proxy** *(string select)*:
  The HTTP/HTTPS proxy to use to reach the groupware tool, if any.
- **To*** *(select)*:
  Select one of:
  - Static

        - **Groupware*** *(select)*:
          Select the groupware on which to send the message. Supported options are:
            - Slack
            - Mattermost
            - Microsoft Teams
        - **URL*** *(select)*:
          The webhook URL allowing the publication of messages. See the prerequisites to obtain one.
    - Teams webhook

- **Title*** *(string input)*:
  Enter the title of the instant message. You may use dynamic attributes, that will be automatically replaced by the appropriate values upon notification generation.
- **Body*** *(string input)*:
  Enter the body of the instant message. You may use dynamic attributes, that will be automatically replaced by the appropriate values upon notification generation.

NOTE	You can click on the "+" next to "How to use dynamic attributes" in order to get a range of possibilities from which one or more may be chosen.

In case you selected as an Event ***Certificate Expiration****:*

- **Duration before certificate expiration causing the notification*** *(finite duration)*:
  Sets how long before certificate expiration the groupware notification should be sent. The default value is set to 5 days.

In case you selected as an Event ***Enroll request Pending*** *or* ***Renew request Pending*** *or* ***Revoke request Pending*** *or* ***Recover request Pending*** *or* ***Update request Pending*** *or* ***Migrate request Pending****:*

- **Duration after request submission causing the notification*** *(finite duration)*:
  Duration after request submission causing the groupware notification to be sent, in case the request was not approved in the meantime. The default value is set to 5 days.

6. Click on the save button.

You can edit ✎ , duplicate ⧉ or delete 🗑 the Groupware Notification.

6.3. REST

This section details how to configure REST notifications.

6.3.1. How to create a REST notification

1. Log in to Horizon Administration Interface.

2. Access REST from the drawer or card: **Notifications › REST**.

3. Click on ⊕ .

4. Fill in all mandatory fields.

- **Name*** *(string input)*:
  Enter a meaningful REST notification name.
- **Event type*** *(select)*:
  Select the event type to notify (certificate or request).
- **Event*** *(select)*:
  Select the event to notify.
- **Retries in case of error** *(int)*:
  Select the number of times Horizon should retry to send the notification in case of error. The default value is set to 10.
- **HTTP Method and URL***: *(select & string input)*
  Choose the HTTP method and the destination URL for your notification.
- **Proxy**: *(select)*
  Define a proxy for this REST API call.
- **Timeout*** *(finite duration)*:
  Connection timeout when executing the REST API call. Must be a valid finite duration.
- **Accepted response HTTP code*** *(multiselect | input)*:
  Response codes meaning the REST call was a success. If another one is received, a failure will be logged.
- **Authentication type and credentials*** *(select & select)*:
  Choose the authentication type and the credentials to perform the authentication. Custom authentication allows the credentials values to be accessible in headers.
- **Headers** *(input string & input string)*:
  Choose the header name and value. The value can be enriched using dynamic attributes.
- **Body*** *(string input)*:
  Enter the REST body. You may use dynamic attributes, that will be automatically replaced by the appropriate values upon REST call.

> **NOTE** You can click on the "Dynamic attributes" drawer in order to get a range of possibilities from which one or more may be chosen.

6. Click on the save button.

You can edit ✎ , duplicate ⧉ or delete 🗑 the REST Notification.

7. Discovery

This section details how to configure Discovery campaigns. An EverTrust Horizon Discovery campaign will contain all certificates discovered on a specific scope.

CAUTION

A discovered certificate can be:

- An unknown certificate.
  > All certificate information will be stored and this certificate will appear as an '`unmanaged`' certificate.
- An already discovered certificate (due to another Discovery campaign).
  > Discovery campaign metadata will be added to the existing certificate.
- A managed certificate.
  > Discovery campaign metadata will be added to the existing certificate.

7.1. How to create a Discovery Campaign

1. Log in to Horizon Administration Interface.

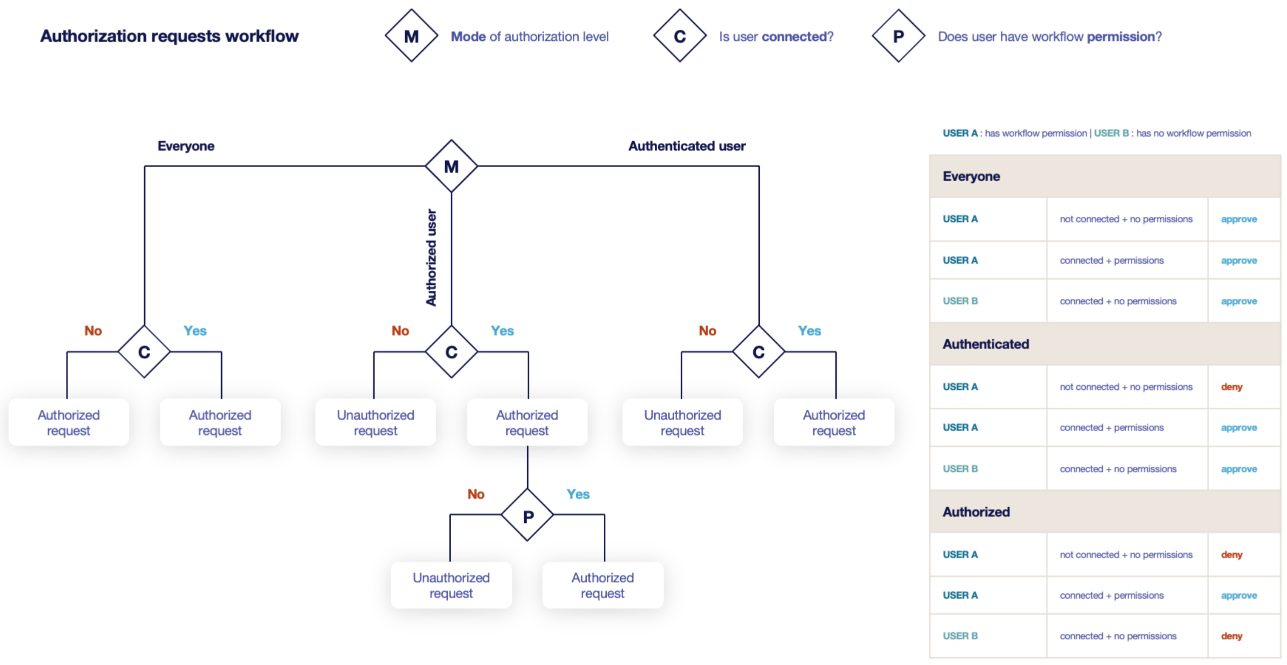
2. Access Discovery from the drawer or card: **Discovery**.

3. Click on ⊕ .

4. Fill in all mandatory fields.

General tab

- **Campaign name*** *(string input)*:
  Enter a meaningful Discovery campaign name.
- **Description** *(string input)*:
  Enter Discovery campaign description.
- **Enable** *(boolean)*:
  Enable/Disable this Discovery campaign.
- **Grading policy** *(select)*:
  The grading policy to apply to every discovered certificate on this campaign.
- **Search** *(select)*:
  Select an authorization level to search this Discovery campaign.
- **Feed** *(select)*:
  Select an authorization level to feed this Discovery campaign.



USER A : has workflow permission | USER B : has no workflow permission

Everyone		
USER A	not connected + no permissions	approve
USER A	connected + permissions	approve
USER B	connected + no permissions	approve
Authenticated		
USER A	not connected + no permissions	deny
USER A	connected + permissions	approve
USER B	connected + no permissions	approve
Authorized		
USER A	not connected + no permissions	deny
USER A	connected + permissions	approve
USER B	connected + no permissions	deny

- **Log event on success*** *(boolean)*:
  Enable/Disable discovery event on success.
- **Log event on failure*** *(boolean)*:
  Enable/Disable discovery event on failure.
- **Log event on warning*** *(boolean)*:
  Enable/Disable discovery event on warning.

Host tab

- **Hosts** *(string input or int)*:
  Specify the target to scan. Can be hostname(s), IP address(es), IP range or CIDR address(es). It is possible to add several hostnames separated by commas.

Port tab

- **Ports** *(string input or int)*:
  Enter the port(s) to scan on hosts. It is possible to add several ports separated by commas or to add a port range separated by an hyphen (ex: 1-1000 to go from 1 to 1000). If no ports are specified, ports 25, 443, 663, 8443 are scanned by the Horizon Client.

NOTE Hosts and ports should only be set if you intend to perform a network scan using `horizon-cli` in order to discover the certificates. These parameters are ignored in all other discovery modes (local scan, third party import).

6. Click on the save button.

You can edit ✎ , flush ↻ or delete 🗑 the Discovery.

7.2. How to flush a Discovery Campaign

Flushing a Discovery campaign is the action to remove Discovery campaign reference from all discovered certificates.

CAUTION

There are three different cases:

- If the certificate is not managed by Horizon (only discovered by a Discovery campaign) AND only referenced by the campaign you are willing to flush → The certificate will be removed from the Horizon database.
- If the certificate is not managed by Horizon but is referenced by at least another Discovery campaign → The certificate will NOT be removed from the database and only the Discovery metadata will be removed from the certificate.
- If the certificate is managed by Horizon → Only the Discovery metadata will be removed from the certificate.

1. Log in to Horizon Administration Interface.

2. Access Discovery from the drawer or card: **Discovery**.

3. Click on ↻.

4. Click on the Confirm button to perform the flush.

8. Automation

8.1. Automation Introduction

Certificate Lifecycle Automation allows your certificates to always be up to date with your security policy without interrupting your services unexpectedly, and without need of tiring manual operations.

The following elements are needed to allow this behavior to take place:

- **horizon-cli**: The horizon client. Installed on your server machine, it will communicate with Horizon to know when to perform the certificate change, and it will install the new certificate automatically. This feature is only available since client version `1.6.0`.

On EverTrust Horizon side:

- **Execution policies**: Define when the interruption of service to switch the certificate should take place.
- **Automation policies**: Define what profile to enroll the certificates on, and also various cryptographic parameters.
- **Profiles**: Define on which protocol and PKI your certificate is enrolled, and its contents. Automation is available on SCEP, ACME and EST profiles.

8.2. Execution Policy

Execution policies are a way to define time periods during which execution of automated actions are permitted. This allows you to avoid a service interruption during business hours.

8.2.1. Configure your execution policies

1. Log in to Horizon Administration Interface.

2. Access Execution policy from the drawer or card: **Automation › Execution Policy**.

3. Click on ⊕ .

4. Fill in the mandatory fields.

General

- **Name*** *(string input)*:
  Enter a meaningful policy name. It must be unique for each execution policy. Horizon use the name to identify the policy.
- **Description** *(string input)*:
  Enter a description for your policy. It will be displayed in a tooltip on the policy list view.

Authorized periods

The Horizon Client can perform automation operations in the following time frames.

Click on ⊕ .

- **Start Date** *(date: yyyy-mm-dd)*:
  Enter the start date of this period. If no start and no end date are defined, all dates are in this period.
- **End Date** *(date: yyyy-mm-dd)*:
  Enter the end date of this period. If no start and no end date are defined, all dates are in this period.
- **Start Time** *(time: hh:mm:ss)*:
  Enter the start time of this period. If no start and no end time are defined, all times are in this period.
- **End Time** *(time: hh:mm:ss)*:
  Enter the end time of this period. If no start and no end time are defined, all times are in this period.
- **Day selector**:
  Enter the authorized days of the week.

  CAUTION Selecting no weekdays means no weekdays are in this period.

You can delete 🗑 periods.

Forbidden periods

The Horizon Client cannot perform automation operations in the following time frames.

Click on ⊕ .

- **Start Date** *(date: yyyy-mm-dd)*:
  Enter the start date of this period. If no start and no end date are defined, all dates are in this period.
- **End Date** *(date: yyyy-mm-dd)*:
  Enter the end date of this period. If no start and no end date are defined, all dates are in this period.
- **Start Time** *(time: hh:mm:ss)*:
  Enter the start time of this period. If no start and no end time are defined, all times are in this period.
- **End Time** *(time: hh:mm:ss)*:
  Enter the end time of this period. If no start and no end time are defined, all times are in this period.
- **Day selector**:

Enter the authorized days of the week.

> **CAUTION** Selecting no weekdays means no weekdays are in this period.

You can delete 🗑 periods.

5. Click on the save button.

You can edit ✎ or delete 🗑 the policy.

8.3. Automation Policy

Automation policies allow you to choose when and how to automate your certificate renewal, while also providing additional security policy parameters.

8.3.1. Configure your automation policies

1. Log in to Horizon Administration Interface.

2. Access Automation policy from the drawer or card: **Automation › Automation Policy**.

3. Click on ⊕ .

4. Fill in the mandatory fields.

General

- **Name*** *(string input)*:
  Enter a meaningful policy name. It must be unique for each automation policy. Horizon use the name to identify the policy.
- **Profile*** *(select)*:
  Select an existing SCEP, EST or ACME profile on which to enroll the certificates.

> **NOTE** Cryptographic information, such as the key types for certificate enrollment are taken from the profile Crypto Policy.

- **Execution policy** *(select)*:
  Select a preexisting execution policy. If no policy is selected, renewal actions are always allowed.

Compliance

- **Authorized CAs** *(multiselect)*:
  Select CAs on which the certificate will be considered as compliant if its issuer is in the list. An empty list means all issuing CAs are authorized.
- **Authorized hash algorithms** *(multiselect)*:

Select algorithms on which the certificate will be considered as compliant if its hash algorithm is in the list. An empty list means all hash algorithms are authorized.

- **Trust chains** *(multiselect)*:
  Select trust chains that will be installed on the machine at the same time as the certificate installation. If no chain is specified, only the one optionally needed by the server will be installed.

5. Click on the save button.

You can edit ✎ or delete 🗑 the policy.

9. Protocols

9.1. ACME

9.1.1. ACME Introduction

This section details how to configure and consume the ACME protocol.

Horizon implements an ACME service respecting the RFC 8555 and more specifically the following lifecycle workflows:

- Enrollment;
- Renewal (which is equivalent to an enrollment);
- Revocation.

Managing certificate lifecycle through the ACME protocol involves up to three components:

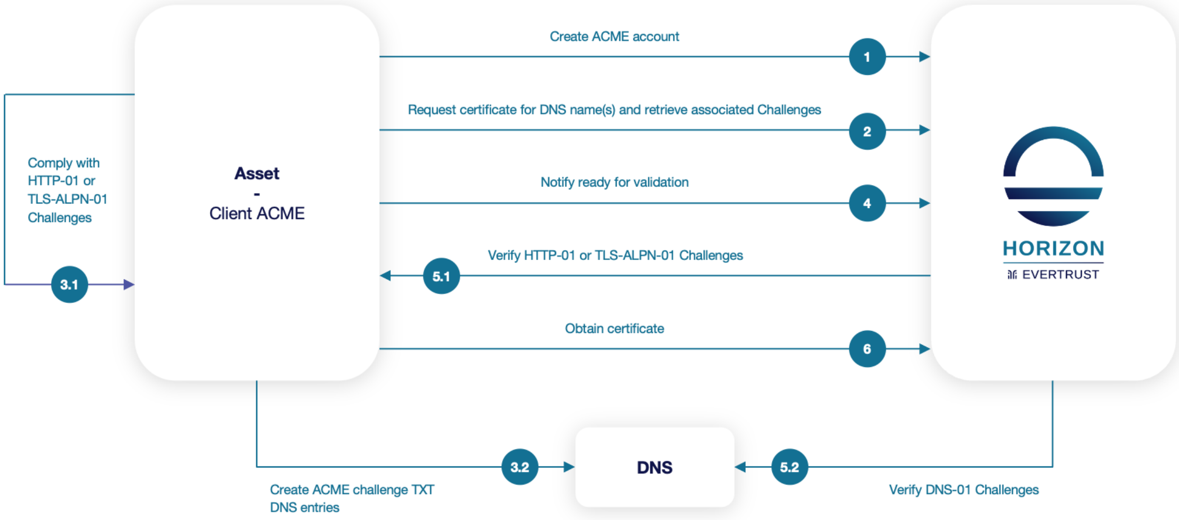
- Horizon as the ACME endpoint;
- An asset executing an ACME client or directly integrating the ACME protocol;
- When the ACME validation method is ' `dns-01` ', DNS server(s).

NOTE ACME validation modes will be detailed later on.

The protocol paradigm can be described as follows: '**if the asset can prove it has authority on the DNS names (called identifiers in ACME) it is requesting for, the certificate should be automatically enrolled / renewed**', which is basically equivalent to a **Domain Validation**.

The following schema is a simplified workflow of an ACME enrollment:

Simplified workflow of an ACME enrollment


Create ACME account
1
Request certificate for DNS name(s) and retrieve associated Challenges
2
Notify ready for validation
4
Verify HTTP-01 or TLS-ALPN-01 Challenges
5.1
Obtain certificate
6
Comply with HTTP-01 or TLS-ALPN-01 Challenges
3.1
Asset
-
Client ACME
HORIZON
EVERTRUST
DNS
3.2
Create ACME challenge TXT DNS entries
5.2
Verify DNS-01 Challenges

The protocol is based on the notion of challenge and offers three validation modes to actually verify challenges and prove that the asset owns authority on the requested DNS name(s), i.e. ACME identifiers:

- **http-01**: For each requested identifier, Horizon will validate the challenge by connecting back in **HTTP** on the configured http-01 *validation port* (TCP/80 by default) and retrieve the response to the challenge;
- **tls-alpn-01**: For each requested identifier, Horizon will validate the challenge by connecting back in **HTTPS** on the configured tls-alpn-01 *validation port* (TCP/443 by default) and extract the response to the challenge from an **ALPN** extension in the asset / client **HTTPS** response;
- **dns-01**: For each requested identifier, Horizon will validate the challenge through a DNS request and look for a specific TXT entry containing the response corresponding to the challenge for the considered identifier.

Therefore, validation modes have the following constraints:

- **http-01 and tls-alpn-01**:
  - Horizon must be able to access the asset on the validation port;
  - The validation port must be available and opened on the asset;
- **dns-01**: the ACME client must be configured with DNS credentials owning the permission to create TXT records on the requested domain(s).

NOTE For **http-01** and **tls_alpn-01** validation modes, it is possible to configure an HTTP proxy to proxify the ACME validation tentative(s). Using an HTTP proxy is useful when http-01 and/or tls-alpn-01 validation need to be performed on asset(s) hosted within a DMZ where incoming network streams must be limited. In this scenario, an HTTP proxy is configured to relay ACME validations coming from the Horizon nodes within the DMZ and a unique incoming stream needs to be open to allow communication from Horizon node to the HTTP proxy.

The choice of the validation mode to use mainly depends on the architecture. Here are the EverTrust recommendations:

- If the requester is not the asset, prefer the **dns-01** validation mode;
- If the requester is the asset:
  - If the asset is reachable from Horizon nodes, prefer the **http-01**;
  - If the asset is not reachable from Horizon nodes, prefer the **dns-01**;
- **tls-alpn-01** is the most complicated validation mode to implement and therefore should only be used when no other validation mode is an option.

Qualified ACME clients

EverTrust qualifies the following ACME clients for any release of the Horizon product:

- Linux ACME clients:

  - acme.sh
  - certbot
  - lego
  - Horizon CLI
- Windows ACME client:
  - lego
  - WinCertes: this open source client is developed and maintained by EverTrust, therefore officially supported
  - Horizon CLI
- Kubernetes: cert-manager

> **NOTE** If an ACME client is not listed above, it does not necessarily mean that the client will not work with Horizon, only that the client is not included in the list of clients tested in Horizon's continuous integration test cases.

9.1.2. ACME Profile

This section details how to configure an ACME Profile.

Prerequisites

PKI Connector

How to configure ACME Profile

1. Log in to Horizon Administration Interface.

2. Access ACME Profile from the drawer or card: **Protocols › ACME**.

3. Click on ⊕ .

4. Fill in the mandatory fields.

ACME Profile Specific Configuration

General

- **Name*** *(string input)*:
  Enter a meaningful profile name. It must be unique for each profile. Horizon uses the name to identify the profile. As the name will be part of a URL, it is advisable to use only lower case letters and dashes.
- **Enable*** *(boolean)*:
  Indicates whether the profile is enabled or not. The default value is set to true.

- **PKI Connector** *(string select)*:
  Select a PKI connector previously created.

Validations

- **Validation Methods** *(select)*:
  Select the authorized ACME validation method(s) on the considered profile (`HTTP-01` and/or `TLS-ALPN-01` and/or `DNS-01`).
- **`HTTP_01` validation port** *(int)*:
  HTTP port to perform the `http-01` validation (only if HTTP-01 has been selected). The default value is set to 80.
- **`TLS-ALPN_01` validation port** *(int)*:
  HTTPS port to perform the `tls-alpn-01` validation (only if TLS-ALPN-01 has been selected). The default value is set to 443.
- **Challenge verification attempts*** *(int)*:
  Specify the number of times Horizon should try to validate an ACME challenge. The default value is set to 3.
- **Challenge verification retry delay*** *(finite duration)*:
  Specify the time duration Horizon should wait between two consecutive validations for the same challenge. The default value is set to 3 seconds.
- **Proxy** *(string select)*:
  Specify an HTTP proxy to use when performing `http-01` or `tls-alpn-01` validations.
- **Timeout*** *(finite duration)*:
  Specify the time duration Horizon should wait when performing `http-01`, `tls-alpn-01` or `dns-01` validations.

Max Certificate per Holder Policy

- **Maximum** *(int)*:
  When specified, define the maximum number of active certificates for a given holder.
- **Behavior** *(select)*:
  What behavior to have when the maximum number is reached:
  - **revoke** the previous certificates.
  - **reject** the current request.
- **Revocation reason** *(select)*:
  When the revoke behavior is selected, the revocation reason to revoke the certificate with.

Requests management

- **Authorized short name** *(boolean)*:
  Specify if using short name is authorized when requesting certificate. If set to yes, one verifiable FQDN must be requested for each specified short name. The default value is set to false.
- **Authorized empty contact** *(boolean)*:
  Specify if an ACME account can be registered without specifying a contact email address.

Default to false.

- **Default contacts email** *(string input multiple)*:
  Specify a list of default contact email addresses when registering an ACME account with no specified contact email address.
- **Max DNS name** *(int)*:
  If specified, enforce the maximum number of requested DNS name(s).

Meta

- **Is required terms of service** *(boolean)*:
  Specify if explicitly agreeing to the terms of service is required when registering an ACME account. The default value is set to false.
- **Terms of service** *(string input)*:
  Specify an URL identifying the current terms of service.
- **Website** *(string input)*:
  Specify an HTTP or HTTPS URL locating a website providing more information about the ACME server.
- **CAA Identities** *(string input)*:
  The hostnames that the ACME server recognizes as referring to itself for the purposes of CAA record validation as defined in RFC6844.

Crypto policy

- **Default Key Type** *(select)*:
  Key Type that will be used by horizon-cli in certificate enrollment.
- **Authorized Key Types** *(multiselect)*:
  Key Types that can be used for enrollment. An empty value means no restrictions.

Common configuration for profiles

Languages

You can add more languages by clicking ⊕ .

- **Language*** *(select)*:
  Select a language. Supported languages are:
  - **en**: English
  - **fr**: French
- **Display Name** *(string input)*:
  Enter a display name. This will be the localized name of this profile.
- **Description** *(string input)*:
  Enter a description. This will be displayed on the list view of the profiles.

You can delete 🗑 the localization.

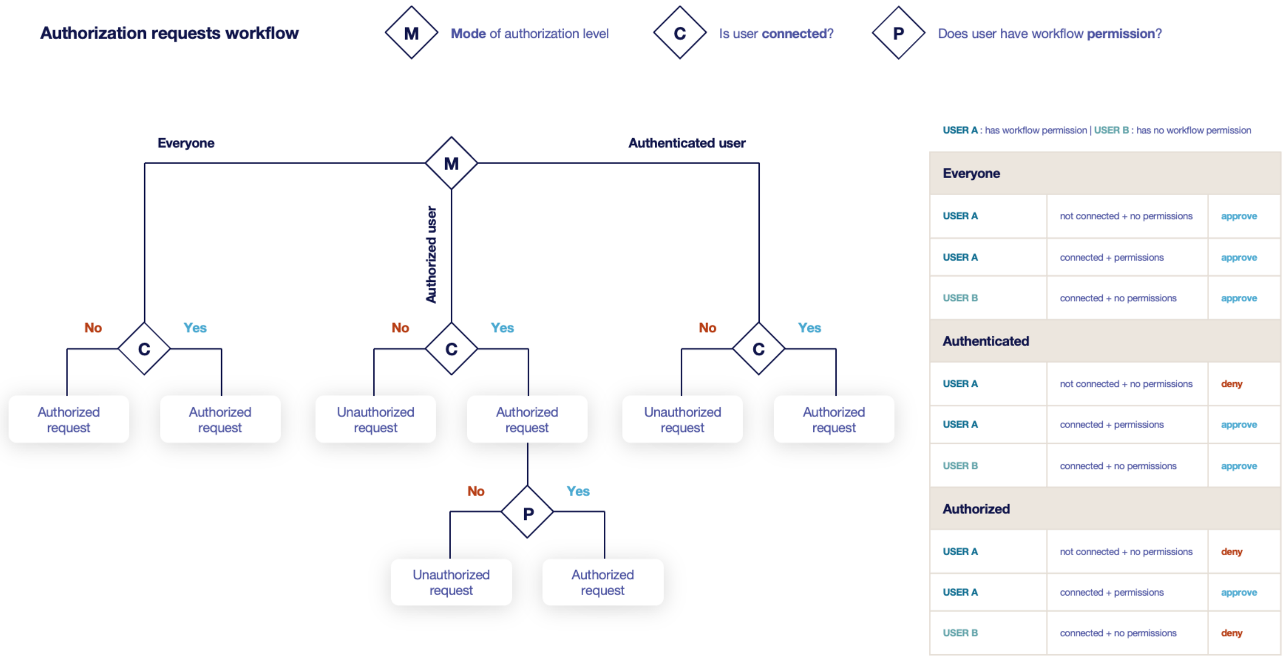
Grading Policies

You can select grading policies that will grade your certificate for a quick overview of its quality. For more information about the inner working of the grading policies in Horizon, please refer to the grading rules page.

Workflows builder

Configure custom rights for actions on this profile.

1. Select an authorization level for each workflow.



USER A : has workflow permission | USER B : has no workflow permission

Everyone		
USER A	not connected + no permissions	approve
USER A	connected + permissions	approve
USER B	connected + no permissions	approve
Authenticated		
USER A	not connected + no permissions	deny
USER A	connected + permissions	approve
USER B	connected + no permissions	approve
Authorized		
USER A	not connected + no permissions	deny
USER A	connected + permissions	approve
USER B	connected + no permissions	deny

- **Everyone**:
  No authentication is required.
- **Authenticated**:
  User has to be authenticated.
- **Authorized**:
  User has to be authenticated and have an explicit authorizations.

2. Select an access level for identity providers.

You can remove the access level for an identity provider by clicking on 'x'.

Requests time to live

Configure the time your requests have before expiring.

NOTE

After expiration, requests are stored for an additional 30 days. This can be changed using configuration files.

- **Enrollment request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Renewal request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Revocation request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Update request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Migration request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Recover request** *(finite duration)*:
  *Enabled on **escrow**:* Must be a valid finite duration. The default value is set to seven days.

Owner-related permissions

These permissions apply to the owners of a certificate (team or owner). An owner can always request the following actions, but this permission allows them to perform the action without validation.

- **Revoke** *(boolean)*:
  Grant self revoke permission. The default value is set to false.
- **Revoke (pop)** *(boolean)*:
  Grant self revoke permission with owner being determined by Proof of Possession. This is used by horizon-cli. The default value is set to false.
- **Update** *(boolean)*:
  Grant self update permission. The default value is set to false.
- **Update (pop)** *(boolean)*:
  Grant self update permission with owner being determined by Proof of Possession. This is used by horizon-cli. The default value is set to false.

Constraints

- **Allowed email domains** *(string input)*:
  Enter a valid regular expression that the inputted emails should match. This includes RFC822NAME and UPN SANs as well as the contact email

  NOTE This matches the domain of the email, not including anything before `@`.

- **Allowed DNS domains** *(string input)*:
  Enter a valid regular expression that the inputted domain should match.

CSR Data Mapping

1. Click on ⊕ to add a mapping.

2. Select a field and enter a value.

You can delete 🗑 the CSR Data Mapping.

Certificate Template

This section details how to define a custom structure for the fields `subject DN`, `SAN` *&* `extensions` *of the requested certificate in order to match the configuration on the PKI side.*

> **NOTE**
>
> Defining a template will use the CSR to fill the available field. A CSR with unexpected fields will be rejected. Using a template also disables CSR Data Mapping.

Subject DN composition

You can add more elements by clicking ⊕ .

- **Element*** *(select)*:
  Select an attribute from the elements list.
- **Mandatory** *(boolean)*:
  Should the element be mandatory. The default value is set to false.
- **Editable by requester** *(boolean)*:
  Tells whether the element should be editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the element should be editable by the approver. The default value is set to false.
- **Default value** *(string input)*:
  Set a default value to the element.
- **Regex** *(regex)*:
  Enter a regular expression that the element should match.
- **Computation rule** *(Computation rule input)*:
  Set the value of this element to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

You can remove an element by clicking the delete button 🗑 or reorder (drag and drop) ⠿ the Subject DN template.

> **CAUTION**
>
> When a template is defined, at least one mandatory Common Name must be added to the DN Elements.

SAN composition

You can add more elements by clicking ⊕ .

- **Element*** *(select)*:
  Select an attribute from the element list.
- **Editable by requester** *(boolean)*:

Tells whether the element should be editable by the requester. The default value is set to false.

- **Editable by approver** *(boolean)*:
  Tells whether the element should be editable by the approver. The default value is set to false.
- **Minimum** *(int)*:
  The minimum number of value that this SAN must have.
- **Maximum** *(int)*:
  The maximum number of value that this SAN must have.
- **Regex** *(regex)*:
  Enter a regular expression that the element should match.
- **Computation rule** *(Computation rule input)*:
  Set the value of this element to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

You can remove an element by clicking the delete button 🗑 or reorder (drag and drop) ⠿ the SAN template.

Extensions

You can add more elements by clicking ⊕ .

- **Element*** *(select)*:
  Select an attribute from the elements list.
- **Mandatory** *(boolean)*:
  Should the element be mandatory. The default value is set to false.
- **Editable by requester** *(boolean)*:
  Tells whether the element should be editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the element should be editable by the approver. The default value is set to false.
- **Default value** *(string input)*:
  Set a default value to the element.
- **Computation rule** *(Computation rule input)*:
  Set the value of this element to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

You can remove an element by clicking the delete button 🗑 or reorder (drag and drop) ⠿ the Extensions template.

CAUTION When adding a SAN, a DN element or an Extension and making it mandatory, make sure to either give it a default value or a computation rule or make it editable, otherwise the template will be unusable.

Certificate Metadata

This section details how to define a custom structure for the labels, ownership policy and technical metadata, allowing certificates to hold rich information.

Labels

You can add more labels by clicking ⊕ .

- **Name** *(select)*:
  Select a preexisting label.
- **Mandatory** *(boolean)*:
  Should the label be mandatory. The default value is set to false.
- **Editable by requester** *(boolean)*:
  Tells whether the label should be editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the label should be editable by the approver. The default value is set to false.
- **Default value** *(string input)*:
  Set a default value to the label.
- **Label value restriction**
  - **Whitelist** *(string input multiple)*:
    The label value will have to be in the whitelist. Open the popup, enter the label value and press "enter" to add this value to the accepted value list. An empty whitelist means no restriction.
  - **Suggestions** *(string input multiple)*:
    Add suggestions that will be displayed to the user. The user will be able to choose one of these values or enter its own. Open the popup, enter your suggestions and press enter to add this value to the suggestions. An empty suggestions list means no restriction.
  - **Regex** *(regex)*:
    The label value will have to match the regex. Open the popup, enter the regular expression and click on the submit button to set the regex. An empty regex means no restrictions.
- **Computation rule** *(Computation rule input)*:
  Set the value of this label to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

You can delete 🗑 or reorder (drag and drop) ⠿ the label template.

Ownership policy

- **Owner**
  - **Mandatory** *(boolean)*:
    Specify if the certificate’s owner is mandatory when submitting a request.
  - **Editable by requester** *(boolean)*:
    Specify if the certificate’s owner can be overridden by the requester when submitting a

request.

  - **Editable by approver** *(boolean)*:
    Specify if the certificate’s owner can be overridden by the requester when approving a request.

  - **Computation rule** *(Computation rule input)*:
    Set the value of the owner to the value of the evaluated computation rule. This value will override any other value including the user input.

- **Contact email**

  - **Mandatory** *(boolean)*:
    Specify if the certificate’s contact email is mandatory when submitting a request.

  - **Editable by requester** *(boolean)*:
    Specify if the certificate’s contact email can be overridden by the requester when submitting a request.

  - **Editable by approver** *(boolean)*:
    Specify if the certificate’s contact email can be overridden by the requester when approving a request.

  - **Default contact email** *(string input)*:
    Set a default contact email. This value must comply with the contact email restriction.

  - **Contact email restriction**

    - **Whitelist** *(string input multiple)*:
      The contact email will have to be in the whitelist. Open the popup, enter the email and press "enter" to add this value to the accepted whitelist. An empty whitelist means no restriction.

    - **Regex** *(regex)*:
      The contact email will have to match the regex. Open the popup, enter the regular expression and click on the submit button to set the regex. An empty regex means no restrictions.

  - **Computation rule** *(Computation rule input)*:
    Set the value of the contact email to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

- **Team**

  - **Mandatory** *(boolean)*:
    Specify if the certificate’s team is mandatory when submitting a request.

  - **Editable by requester** *(boolean)*:
    Specify if the certificate’s team can be overridden by the requester when submitting a request.

  - **Editable by approver** *(boolean)*:
    Specify if the certificate’s team can be overridden by the requester when approving a request.

  - **Default team** *(string input)*:
    Set a default team. This value must comply with the team restriction.

  - **Team restriction**
    - **Whitelist** *(string input multiple)*:
      The team will have to be in the whitelist. Enter the team and press "enter" to add this value to the accepted whitelist. An empty whitelist means no restriction.
    - **Regex** *(regex)*:
      The team will have to match the regex. Open the popup, enter the regular expression and click on the submit button to set the regex. An empty regex means no restrictions.
  - **Computation rule** *(Computation rule input)*:
    Set the value of the team to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

Metadata policy *(overridable metadata)*

WARNING These metadata are technical metadata. They are used by Horizon or Third party connectors, updating them should be done with utmost care.

NOTE Metadata edition is not allowed on enroll.

NOTE Metadata edition is not available via the User Interface. It must be changed with API, using horizon-cli.

You can allow the override of technical metadata by clicking ⊕ .

- **Metadata*** *(select)*:
  Select a metadata.
- **Editable by requester** *(boolean)*:
  Tells whether the metadata is editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the metadata is editable by the approver. The default value is set to false.

You can delete 🗑 a metadata policy. This will not delete the metadata but will make it non editable.

Notifications

This section details how to configure notifications on certificate and request lifecycle events.

Certificate lifecycle notifications

Notifications are sent when one of the following event is triggered by a certificate:

Enrollment	Revocation	Expire	Update	Migrate	Renew

Select a preexisting email, REST or groupware notification to associate it with an event.

Request lifecycle notifications

Notifications are sent when one of the following event is triggered by an Enroll/Revocation/Update/Migrate/Renew request:

Submit	Cancel	Revoke	Approve	Pending

Select a preexisting email, REST or groupware notification to associate it with an event.

NOTE Submit request events are not triggered when the user has the permission to perform the action directly.

5. Click on the save button.

You can edit ✎ , duplicate ⧉ or delete 🗑 the ACME Profile.

CAUTION You won't be able to delete an ACME Profile if it is referenced somewhere else.

9.1.3. ACME client usages

This section details how to use the most common Linux and Windows ACME clients.

Linux ACME clients

This section details how to use the **acme.sh** and **certbot** ACME clients.

Overview

Certbot is able to run on any recent UNIX-like operating system equipped with Python 2.7 or 3.4+, while acme.sh can also run on any recent Linux distribution running either bash, dash or sh.

They both fully support the latest ACMEv2 protocol including its main latest feature: wildcard certificates (*.example.com).

Both clients supports different modes for obtaining a certificate and in some cases automatically installing it.

The following tables lists the different modes for each clients:

Modes	certbot	acme.sh	Notes
apache	Y	Y	Obtains and automatically installs a certificate using the running Apache server. (For acme.sh, this mode will only obtain a certificate without installing it)
nginx	Y	Y	Obtains and automatically installs a certificate using the running NGINX server. (For acme.sh, this mode will only obtain a certificate without installing it)

Modes	certbot	acme.sh	Notes
webroot	Y	Y	Obtains a certificate by writing to the webroot directory of an already running web server
standalone	Y	Y	Uses a "standalone" web server managed by Certbot or acme.sh. This mode is useful on system with no web servers or if using the running web server is not desired
DNS	Y	Y	This mode automates obtaining a certificate by modifying a DNS record to prove the control over a domain
`tls-alpn`	N	Y	Uses a TLS server to validate the control over a domain

Requesting a certificate

Both clients must be started using administrative privileges (`sudo`), except for acme.sh when using the webroot or DNS modes.

Each client requires only a few parameters to request a certificate.

acme.sh parameters:

Parameter	Description
`-issue`	Obtain or renew a certificate, but does not install it
`-w [VALUE]`	Path of the server’s webroot folder
`-d [VALUE]`	The domain(s) to enroll.

certbot parameters:

Parameter	Description
`certonly`	Obtain or renew a certificate, but does not install it
`▯webroot`	Place files in a server’s webroot folder for authentication
`-w [VALUE]`	Path of the server’s webroot folder
`-d [VALUE]`	The domain(s) to enroll.

Requesting a certificate for Apache using certbot:

```
(sudo) certbot run --apache --no-eff-email --agree-tos --server <Horizon ACME
endpoint, example: https://horizon.evertrust.fr/acme/profile1/directory> -m <contact
email address, example: kma@evertrust.fr> --domain <DNS name, example:
apache.evertrust.fr>
```

Where:

- `--apache`: Enables the Apache mode
- `--no-eff-email`: Does not share your email address with EFF
- `--agree-tos`: Explicitly agrees to the terms of service
- `--server`: Horizon ACME profile endpoint
- `-m`: Contact email address
- `--domain`: Requested DNS name (can be specified several times)

Requesting a certificate for nginx using certbot:

```
(sudo) certbot run --nginx --no-eff-email --agree-tos --server <Horizon ACME endpoint,
example: https://horizon.evertrust.fr/acme/profile1/directory> -m <contact email
address, example: kma@evertrust.fr> --domain <DNS name, example: nginx.evertrust.fr>
```

Where:

- `--nginx`: Enables the nginx mode
- `--no-eff-email`: Does not share your email address with EFF
- `--agree-tos`: Explicitly agrees to the terms of service
- `--server`: Horizon ACME profile endpoint
- `-m`: Contact email address
- `--domain`: Requested DNS name (can be specified several times)

Requesting a certificate for nginx using acme.sh:

```
(sudo) acme.sh --issue --nginx --server <Horizon ACME endpoint, example:
https://horizon.evertrust.fr/acme/profile1/directory> --accountemail <contact email
address, example: kma@evertrust.fr> -d <DNS name, example: nginx.evertrust.fr>
```

Where:

- `--issue`: Specifies that this is a certificate request
- `--nginx`: Enables the nginx mode
- `--server`: Horizon ACME profile endpoint
- `--accountemail`: Contact email address
- `-d`: Requested DNS name (can be specified several times)

Requesting a certificate in standalone mode using certbot:

```
(sudo) certbot certonly --standalone --no-eff-email --agree-tos --server <Horizon ACME
endpoint, example: https://horizon.evertrust.fr/acme/profile1/directory> -m <contact
```

```
email address, example: kma@evertrust.fr> --domain <DNS name, example:
apache.evertrust.fr>
```

Where:

- `--standalone`: Enables the standalone mode, i.e. certbot will start a local web server to server the response
- `--no-eff-email`: Does not share your email address with EFF
- `--agree-tos`: Explicitly agrees to the terms of service
- `--server`: Horizon ACME profile endpoint
- `-m`: Contact email address
- `--domain`: Requested DNS name (can be specified several times)

Requesting a certificate in standalone mode using acme.sh:

```
(sudo) acme.sh --issue --standalone --server <Horizon ACME endpoint, example:
https://horizon.evertrust.fr/acme/profile1/directory> --accountemail <contact email
address, example: kma@evertrust.fr> -d <DNS name, example: apache.evertrust.fr>
```

Where:

- `--issue`: Specifies that this is a certificate request
- `--standalone`: Enables the standalone mode, i.e. acme.sh will start a local web server to server the response
- `--server`: Horizon ACME profile endpoint
- `--accountemail`: Contact email address
- `-d`: Requested DNS name (can be specified several times)

Revoking a certificate

Revoking a certificate using certbot:

```
(sudo) certbot revoke --cert-path <path of the certificate to revoke> --server
<Horizon ACME endpoint, example: https://horizon.evertrust.fr/acme/profile1/directory>
```

Where:

- `--cert-path`: Specifies the path of the certificate to revoke
- `--server`: Horizon ACME profile endpoint

Revoking a certificate using acme.sh:

```
(sudo) acme.sh --server <Horizon ACME endpoint, example:
```

```
https://horizon.evertrust.fr/acme/profile1/directory> --revoke -d <DNS name, example:
apache.evertrust.fr>
```

Where:

- `--server`: Horizon ACME profile endpoint
- `-d`: DNS name of the certificate to revoke

Windows ACME clients

This section details how to use the **WinCertes** ACME client.

Overview

WinCertes is a simple and efficient CLI-based client made to run on any Windows Server (> Windows Server 2008 R2 SP1 (64 bits)) and running .NET 4.6.1 or higher.

The client fully supports ACMEv2 including its latest feature, along with the support of wildcard certificates (*.example.com).

WinCertes eases certificate installation and renewal by automatically binding them to the appropriate web site on IIS and by creating a Scheduled Task that will check the expiration date of the certificates and trigger a renewal if necessary.

WinCertes offers the possibility to launch a PowerShell script upon the successful retrieval of a certificate. This feature enables advanced deployment on Exchange or multi-servers for instance.

The client supports two validation modes for validating the identity of the certificate requester:

1. HTTP challenge validation
   - With the ability to support the running IIS web server or to use an embedded standalone web server for easier configuration.
2. DNS challenge validation
   - Support for Windows DNS Server
   - Support for `acme-dns`

Requesting a certificate

To request a certificate using WinCertes, the Windows command line (`cmd.exe`) must be run as Administrator.

Then WinCertes requires only a few parameters to request a certificate:

Parameter	**Description**
`-d [VALUE]`	The domain(s) to enroll

Parameter	Description
`-w`	toggle the local web server use and sets its ROOT directory (default `c:\inetpub\wwwroot`). Activates HTTP validation mode.
`-b [VALUE]`	The name of the IIS web site to bind the certificate to
`-p`	Used to make WinCertes create a Scheduled Task to handle certificate renewal

There are many more options to customize the requests to specific needs.

Requesting a certificate for IIS using WinCertes:

```
(as administrator) wincertes -s <Horizon ACME endpoint, example:
https://horizon.evertrust.fr/acme/profile1/directory> -w -b <IIS Site Name, example:
"Default Web Site"> -p -e <contact email address, example: kma@evertrust.fr> -d <DNS
name, example: iis.evertrust.fr>
```

Where:

- `-s`: Horizon ACME profile endpoint
- `-w`: Enables standalone mode, i.e. WinCertes will start a local web server to serve the response
- `-b`: IIS Web Site name
- `-p`: Registers a scheduled task to enable certificate automated renewal
- `-e`: Contact email address

9.2. ACME External

9.2.1. ACME External Introduction

This section details how to configure the ACME protocol to be managed by Horizon but enrolled on an external ACME endpoint.

The certificate are not enrolled on Horizon but managed thanks to automatic import by third parties such as the Horizon Client.

External ACME enrollment allows to configure:

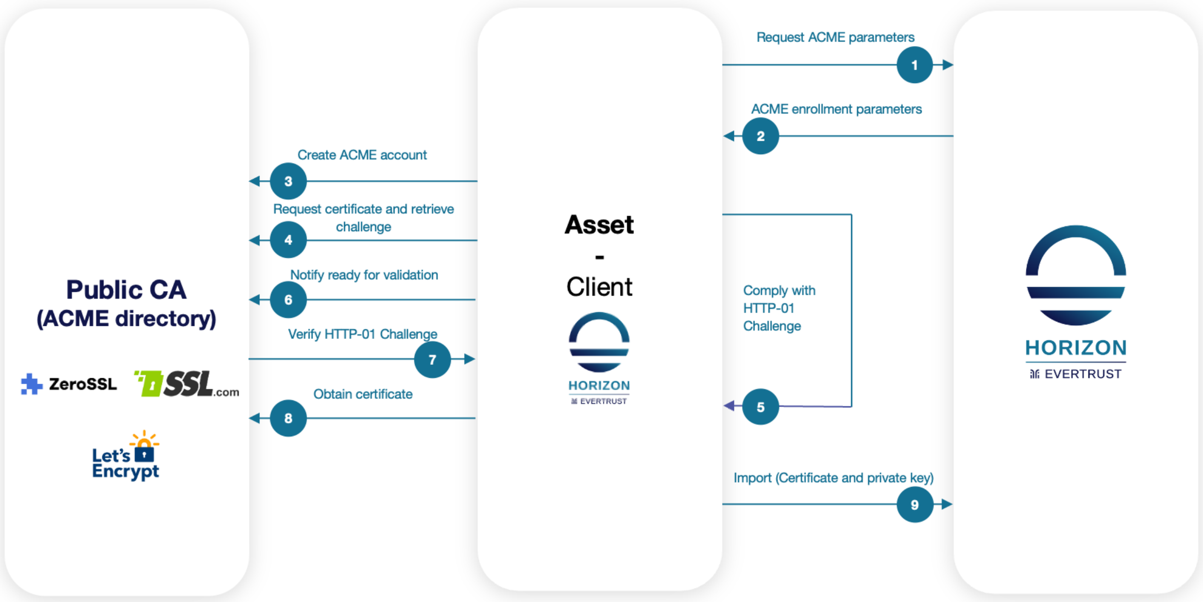
- Enrollment (will be performed by the third party);
- Renewal (will be performed by the third party, depending on the Horizon defined renewal period);
- Revocation (will be performed by Horizon).

NOTE ACME validation modes will be detailed later on. As of today, only `http-01` validation is supported.

The protocol paradigm can be described as follows: '**if the asset can prove it has authority on the DNS names (called identifiers in ACME) it is requesting for, the certificate should be automatically enrolled / renewed**', which is basically equivalent to a **Domain Validation**.

The following schema is a simplified workflow of an ACME External enrollment:

Simplified workflow of an ACME External enrollment



The protocol is based on the notion of challenge and offers three validation modes to actually verify challenges and prove that the asset owns authority on the requested DNS name(s), i.e. ACME identifiers:

> **NOTE** As of today, only `http-01` validation is supported.

- `http-01`: For each requested identifier, the ACME repository will validate the challenge by connecting back in **HTTP** on the configured `http-01` *validation port* (TCP/80 by default) and retrieve the response to the challenge;
- `tls-alpn-01`: For each requested identifier, the ACME repository will validate the challenge by connecting back in **HTTPS** on the configured `tls-alpn-01` *validation port* (TCP/443 by default) and extract the response to the challenge from an **ALPN** extension in the asset / client **HTTPS** response; (not yet supported)
- `dns-01`: For each requested identifier, the ACME repository will validate the challenge through a DNS request and look for a specific TXT entry containing the response corresponding to the challenge for the considered identifier. (not yet supported)

Therefore, validation modes have the following constraints:

- `http-01 and tls-alpn-01`:
  - The ACME Repository must be able to access the asset on the validation port;
  - The validation port must be available and opened on the asset;
- `dns-01`: the ACME client must be configured with DNS credentials owning the permission to

create TXT records on the requested domain(s).

Supported third parties

The following third party ACME directories have been tested with horizon-cli and Horizon:

> **CAUTION** Most third party vendors only support `keyCompromise` as revocation reason. Revocation with another reason will be rejected or ignored.

- Let’s encrypt
- ZeroSSL
- SSL.com

9.2.2. ACME External Profile

This section details how to configure an ACME External Profile.

Prerequisites

PKI Connector

How to configure ACME External Profile

1. Log in to Horizon Administration Interface.

2. Access ACME External Profile from the drawer or card: **Protocols › ACME External**.

3. Click on ⊕ .

4. Fill in the mandatory fields.

ACME Profile Specific Configuration

General

- **Name*** *(string input)*:
  Enter a meaningful profile name. It must be unique for each profile. Horizon uses the name to identify the profile. As the name will be part of a URL, it is advisable to use only lower case letters and dashes.
- **Enable*** *(boolean)*:
  Indicates whether the profile is enabled or not. The default value is set to true.
- **PKI Connector** *(select)*:
  Select a PKI connector previously created. Only ACME connectors can be selected for this profile as only ACME revocation is supported.

Validations

- **Validation Methods*** *(select)*:
  Select the authorized ACME validation method(s) on the considered profile (`HTTP-01`).
- **ACME URL endpoint*** *(string input)*:
  Enter the ACME repository endpoint. It should end in `/acme/directory`.
- **Authorized CAs** *(select multiple)*:
  Select the authorized CAs for enrollment. Certificates not emitted on these CAs will not be able to be imported.
- **Require External Account Binding** *(boolean)*:
  Enable the requirement for the Horizon Client to ask for external account binding.

Max Certificate per Holder Policy

- **Maximum** *(int)*:
  When specified, define the maximum number of active certificates for a given holder.
- **Behavior** *(select)*:
  What behavior to have when the maximum number is reached:
  - **revoke** the previous certificates.
  - **reject** the current request.
- **Revocation reason** *(select)*:
  When the revoke behavior is selected, the revocation reason to revoke the certificate with.

Crypto policy

- **Default Key Type** *(select)*:
  Key Type that will be used by horizon-cli in certificate enrollment.
- **Authorized Key Types** *(multiselect)*:
  Key Types that can be used for enrollment. An empty value means no restrictions.
  - **Private key escrowing** *(boolean)*:
    Tells whether the private key should be escrowed by Horizon. This is true for ACME External profiles as the private key is required for revocation.

Common configuration for profiles

Languages

You can add more languages by clicking ⊕ .

- **Language*** *(select)*:
  Select a language. Supported languages are:
  - **en**: English
  - **fr**: French
- **Display Name** *(string input)*:

Enter a display name. This will be the localized name of this profile.

- **Description** *(string input)*:
  Enter a description. This will be displayed on the list view of the profiles.

You can delete 🗑 the localization.

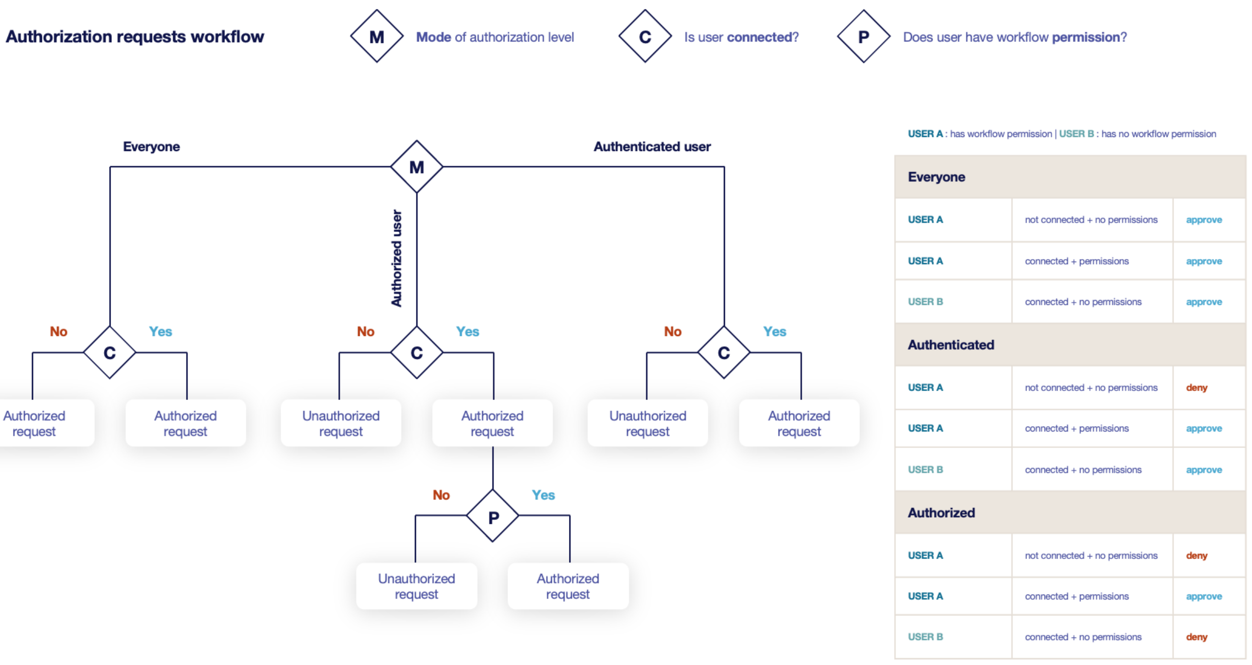
Grading Policies

You can select grading policies that will grade your certificate for a quick overview of its quality. For more information about the inner working of the grading policies in Horizon, please refer to the grading rules page.

Workflows builder

Configure custom rights for actions on this profile.

1. Select an authorization level for each workflow.



USER A : has workflow permission | USER B : has no workflow permission

Everyone		
USER A	not connected + no permissions	approve
USER A	connected + permissions	approve
USER B	connected + no permissions	approve
Authenticated		
USER A	not connected + no permissions	deny
USER A	connected + permissions	approve
USER B	connected + no permissions	approve
Authorized		
USER A	not connected + no permissions	deny
USER A	connected + permissions	approve
USER B	connected + no permissions	deny

- **Everyone**:
  No authentication is required.
- **Authenticated**:
  User has to be authenticated.
- **Authorized**:
  User has to be authenticated and have an explicit authorizations.

2. Select an access level for identity providers.

You can remove the access level for an identity provider by clicking on 'x'.

Requests time to live

Configure the time your requests have before expiring.

NOTE After expiration, requests are stored for an additional 30 days. This can be changed using configuration files.

- **Enrollment request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Renewal request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Revocation request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Update request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Migration request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Recover request** *(finite duration)*:
  *Enabled on **escrow**:* Must be a valid finite duration. The default value is set to seven days.

Owner-related permissions

These permissions apply to the owners of a certificate (team or owner). An owner can always request the following actions, but this permission allows them to perform the action without validation.

- **Revoke** *(boolean)*:
  Grant self revoke permission. The default value is set to false.
- **Revoke (pop)** *(boolean)*:
  Grant self revoke permission with owner being determined by Proof of Possession. This is used by horizon-cli. The default value is set to false.
- **Update** *(boolean)*:
  Grant self update permission. The default value is set to false.
- **Update (pop)** *(boolean)*:
  Grant self update permission with owner being determined by Proof of Possession. This is used by horizon-cli. The default value is set to false.

Constraints

- **Allowed email domains** *(string input)*:
  Enter a valid regular expression that the inputted emails should match. This includes RFC822NAME and UPN SANs as well as the contact email

  NOTE This matches the domain of the email, not including anything before `@`.

- **Allowed DNS domains** *(string input)*:
  Enter a valid regular expression that the inputted domain should match.

CSR Data Mapping

1. Click on ⊕ to add a mapping.

2. Select a field and enter a value.

You can delete 🗑 the CSR Data Mapping.

Certificate Metadata

This section details how to define a custom structure for the labels, ownership policy and technical metadata, allowing certificates to hold rich information.

Labels

You can add more labels by clicking ⊕ .

- **Name** *(select)*:
  Select a preexisting label.
- **Mandatory** *(boolean)*:
  Should the label be mandatory. The default value is set to false.
- **Editable by requester** *(boolean)*:
  Tells whether the label should be editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the label should be editable by the approver. The default value is set to false.
- **Default value** *(string input)*:
  Set a default value to the label.
- **Label value restriction**
  - **Whitelist** *(string input multiple)*:
    The label value will have to be in the whitelist. Open the popup, enter the label value and press "enter" to add this value to the accepted value list. An empty whitelist means no restriction.
  - **Suggestions** *(string input multiple)*:
    Add suggestions that will be displayed to the user. The user will be able to choose one of these values or enter its own. Open the popup, enter your suggestions and press enter to add this value to the suggestions. An empty suggestions list means no restriction.
  - **Regex** *(regex)*:
    The label value will have to match the regex. Open the popup, enter the regular expression and click on the submit button to set the regex. An empty regex means no restrictions.
- **Computation rule** *(Computation rule input)*:
  Set the value of this label to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

You can delete 🗑 or reorder (drag and drop) ⠿ the label template.

Ownership policy

- **Owner**
    - **Mandatory** *(boolean)*:
      Specify if the certificate’s owner is mandatory when submitting a request.
    - **Editable by requester** *(boolean)*:
      Specify if the certificate’s owner can be overridden by the requester when submitting a request.
    - **Editable by approver** *(boolean)*:
      Specify if the certificate’s owner can be overridden by the requester when approving a request.
    - **Computation rule** *(Computation rule input)*:
      Set the value of the owner to the value of the evaluated computation rule. This value will override any other value including the user input.
- **Contact email**
    - **Mandatory** *(boolean)*:
      Specify if the certificate’s contact email is mandatory when submitting a request.
    - **Editable by requester** *(boolean)*:
      Specify if the certificate’s contact email can be overridden by the requester when submitting a request.
    - **Editable by approver** *(boolean)*:
      Specify if the certificate’s contact email can be overridden by the requester when approving a request.
    - **Default contact email** *(string input)*:
      Set a default contact email. This value must comply with the contact email restriction.
    - **Contact email restriction**
        - **Whitelist** *(string input multiple)*:
          The contact email will have to be in the whitelist. Open the popup, enter the email and press "enter" to add this value to the accepted whitelist. An empty whitelist means no restriction.
        - **Regex** *(regex)*:
          The contact email will have to match the regex. Open the popup, enter the regular expression and click on the submit button to set the regex. An empty regex means no restrictions.
    - **Computation rule** *(Computation rule input)*:
      Set the value of the contact email to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.
- **Team**
    - **Mandatory** *(boolean)*:
      Specify if the certificate’s team is mandatory when submitting a request.
    - **Editable by requester** *(boolean)*:
      Specify if the certificate’s team can be overridden by the requester when submitting a

    request.

  - **Editable by approver** *(boolean)*:
    Specify if the certificate’s team can be overridden by the requester when approving a request.

  - **Default team** *(string input)*:
    Set a default team. This value must comply with the team restriction.

  - **Team restriction**

    - **Whitelist** *(string input multiple)*:
      The team will have to be in the whitelist. Enter the team and press "enter" to add this value to the accepted whitelist. An empty whitelist means no restriction.

    - **Regex** *(regex)*:
      The team will have to match the regex. Open the popup, enter the regular expression and click on the submit button to set the regex. An empty regex means no restrictions.

  - **Computation rule** *(Computation rule input)*:
    Set the value of the team to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

Metadata policy *(overridable metadata)*

WARNING These metadata are technical metadata. They are used by Horizon or Third party connectors, updating them should be done with utmost care.

NOTE Metadata edition is not allowed on enroll.

NOTE Metadata edition is not available via the User Interface. It must be changed with API, using horizon-cli.

You can allow the override of technical metadata by clicking ⊕ .

- **Metadata*** *(select)*:
  Select a metadata.

- **Editable by requester** *(boolean)*:
  Tells whether the metadata is editable by the requester. The default value is set to false.

- **Editable by approver** *(boolean)*:
  Tells whether the metadata is editable by the approver. The default value is set to false.

You can delete 🗑 a metadata policy. This will not delete the metadata but will make it non editable.

Notifications

This section details how to configure notifications on certificate and request lifecycle events.

Certificate lifecycle notifications

Notifications are sent when one of the following event is triggered by a certificate:

Enrollment	Revocation	Expire	Update	Migrate	Renew

Select a preexisting email, REST or groupware notification to associate it with an event.

Request lifecycle notifications

Notifications are sent when one of the following event is triggered by an Enroll/Revocation/Update/Migrate/Renew request:

Submit	Cancel	Revoke	Approve	Pending

Select a preexisting email, REST or groupware notification to associate it with an event.

NOTE Submit request events are not triggered when the user has the permission to perform the action directly.

5. Click on the save button.

You can edit ✎ , duplicate ⧉ or delete 🗑 the ACME Profile.

CAUTION You won’t be able to delete an ACME External Profile if it is referenced somewhere else.

9.3. CRMP

9.3.1. CRMP Introduction

This section refers to the CRMP protocol, used by the OpenTrust CMS. It can be used to enroll, revoke and recover certificates on physical supports such as cards.

This integration involves the following components:

- OpenTrust CMS
- EverTrust Horizon
- Cards to be enrolled

The integration is very simple, Horizon acting as an OpenTrust PKI for OpenTrust CMS. We will first see how to configure the Horizon CRMP Profile, which will define how to enroll your certificates, and then a step by step guide for a quick integration.

9.3.2. CRMP Profile

This section details how to configure the CRMP Profile

Prerequisites

PKI Connector

How to configure CRMP Profile

1. Log in to Horizon Administration Interface.

2. Access CRMP Profile from the drawer or card: **Protocol › CRMP**.

3. Click on ⊕ .

4. Fill in the mandatory fields.

CRMP profile specific configuration

General

- **Name*** *(string input)*:
  Enter a meaningful profile name. It must be unique for each profile. Horizon use the name to identify the profile.
- **Enable** *(boolean)*:
  Tells whether the profile is enabled for enrollment or not. The default value is set to true.
- **PKI Connector*** *(string select)*:
  Select a PKI connector previously created.
- **Data field identifier** *(select)*:
  *Enabled on **Escrow**:* When **recovering** a certificate, select on which Horizon field the field named `userprincipalname` on the CMS will be mapped. This will be used to identify a user, so this data should be a unique identifier on the CMS side, in a field named `userprincipalname`, and mapped to the corresponding Horizon field in application configuration on the CMS.

Crypto policy

- **Default Key Type*** *(select)*:
  Key Type that will be used by the CMS in certificate enrollment.
- **Centralized enrollment** *(boolean)*:
  Enable centralized enrollment. In CRMP, only one enrollment mode can be enabled.
  - **Private key escrowing** *(boolean)*:
    Enable key escrow. Only available in centralized enrollment mode.
  - **PKCS#12 Password generation Mode*** *(select)*:
    *For certificate **recovery**:* Select a mode for PKCS#12 password generation:
    - **manual**: prompt the user to choose its password.
    - **random**: have the password generated on Horizon side.
  - **Password policy** *(select)*:

    Select a previously created password policy. It will be enforced on PKCS#12 password for recovery and CMS centralized enrollments.

    - **Store encryption type*** *(select)*:
      Select an encryption algorithm from the list. The PKCS#12 will use this algorithm. For CRMP it is enforced on DES Average because of CMS support.

- **Decentralized enrollment** *(boolean)*:
  Enable decentralized enrollment. In CRMP, only one enrollment mode can be enabled.

Max Certificate per Holder Policy

- **Maximum** *(int)*:
  When specified, define the maximum number of active certificates for a given Holder.
- **Behavior** *(select)*:
  What behavior to have when the maximum number is reached:
    - **revoke** the previous certificates.
    - **reject** the current request.
- **Revocation reason** *(select)*:
  When the revoke behavior is selected, the revocation reason to revoke the certificate with.

Common configuration for profiles

Languages

You can add more languages by clicking ⊕ .

- **Language*** *(select)*:
  Select a language. Supported languages are:
    - **en**: English
    - **fr**: French
- **Display Name** *(string input)*:
  Enter a display name. This will be the localized name of this profile.
- **Description** *(string input)*:
  Enter a description. This will be displayed on the list view of the profiles.

You can delete 🗑 the localization.

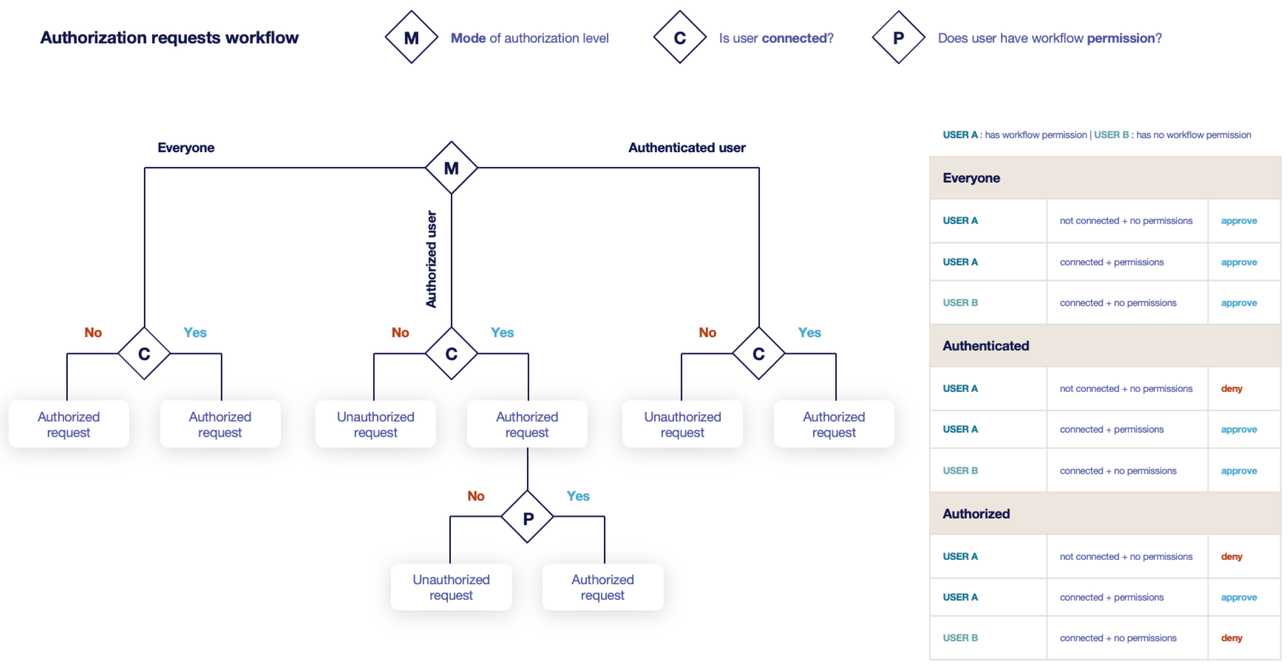
Grading Policies

You can select grading policies that will grade your certificate for a quick overview of its quality. For more information about the inner working of the grading policies in Horizon, please refer to the grading rules page.

Workflows builder

Configure custom rights for actions on this profile.

1. Select an authorization level for each workflow.



USER A : has workflow permission | USER B : has no workflow permission

Everyone		
USER A	not connected + no permissions	approve
USER A	connected + permissions	approve
USER B	connected + no permissions	approve
Authenticated		
USER A	not connected + no permissions	deny
USER A	connected + permissions	approve
USER B	connected + no permissions	approve
Authorized		
USER A	not connected + no permissions	deny
USER A	connected + permissions	approve
USER B	connected + no permissions	deny

- **Everyone**:
  No authentication is required.
- **Authenticated**:
  User has to be authenticated.
- **Authorized**:
  User has to be authenticated and have an explicit authorizations.

2. Select an access level for identity providers.

You can remove the access level for an identity provider by clicking on 'x'.

Requests time to live

Configure the time your requests have before expiring.

> **NOTE** After expiration, requests are stored for an additional 30 days. This can be changed using configuration files.

- **Enrollment request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Renewal request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Revocation request*** *(finite duration)*:

Must be a valid finite duration. The default value is set to seven days.

- **Update request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Migration request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Recover request** *(finite duration)*:
  *Enabled on **escrow**:* Must be a valid finite duration. The default value is set to seven days.

Owner-related permissions

These permissions apply to the owners of a certificate (team or owner). An owner can always request the following actions, but this permission allows them to perform the action without validation.

- **Revoke** *(boolean)*:
  Grant self revoke permission. The default value is set to false.
- **Recover** *(boolean)*:
  Grant self recover permission. The default value is set to false.
- **Update** *(boolean)*:
  Grant self update permission. The default value is set to false.

Constraints

- **Allowed email domains** *(string input)*:
  Enter a valid regular expression that the inputted emails should match. This includes RFC822NAME and UPN SANs as well as the contact email

  NOTE This matches the domain of the email, not including anything before `@`.

- **Allowed DNS domains** *(string input)*:
  Enter a valid regular expression that the inputted domain should match.

Certificate Template

This section details how to define a custom structure for the fields `subject DN`, `SAN` *&* `extensions` *of the requested certificate in order to match the configuration on the PKI side.*

CAUTION In a CRMP profile, defining a template is mandatory.

Subject DN composition

You can add more elements by clicking ⊕ .

- **Element*** *(select)*:
  Select an attribute from the elements list.
- **Mandatory** *(boolean)*:
  Should the element be mandatory. The default value is set to false.

- **Editable by requester** *(boolean)*:
  Tells whether the element should be editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the element should be editable by the approver. The default value is set to false.
- **Default value** *(string input)*:
  Set a default value to the element.
- **Regex** *(regex)*:
  Enter a regular expression that the element should match.
- **Computation rule** *(Computation rule input)*:
  Set the value of this element to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

You can remove an element by clicking the delete button 🗑 or reorder (drag and drop) ⠿ the Subject DN template.

CAUTION When a template is defined, at least one mandatory Common Name must be added to the DN Elements.

SAN composition

You can add more elements by clicking ⊕ .

- **Element*** *(select)*:
  Select an attribute from the element list.
- **Editable by requester** *(boolean)*:
  Tells whether the element should be editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the element should be editable by the approver. The default value is set to false.
- **Minimum** *(int)*:
  The minimum number of value that this SAN must have.
- **Maximum** *(int)*:
  The maximum number of value that this SAN must have.
- **Regex** *(regex)*:
  Enter a regular expression that the element should match.
- **Computation rule** *(Computation rule input)*:
  Set the value of this element to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

You can remove an element by clicking the delete button 🗑 or reorder (drag and drop) ⠿ the SAN template.

Extensions

You can add more elements by clicking ⊕ .

- **Element*** *(select)*:
  Select an attribute from the elements list.
- **Mandatory** *(boolean)*:
  Should the element be mandatory. The default value is set to false.
- **Editable by requester** *(boolean)*:
  Tells whether the element should be editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the element should be editable by the approver. The default value is set to false.
- **Default value** *(string input)*:
  Set a default value to the element.
- **Computation rule** *(Computation rule input)*:
  Set the value of this element to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

You can remove an element by clicking the delete button 🗑 or reorder (drag and drop) ⠿ the Extensions template.

CAUTION When adding a SAN, a DN element or an Extension and making it mandatory, make sure to either give it a default value or a computation rule or make it editable, otherwise the template will be unusable.

Certificate Metadata

This section details how to define a custom structure for the labels, ownership policy and technical metadata, allowing certificates to hold rich information.

Labels

You can add more labels by clicking ⊕ .

- **Name** *(select)*:
  Select a preexisting label.
- **Mandatory** *(boolean)*:
  Should the label be mandatory. The default value is set to false.
- **Editable by requester** *(boolean)*:
  Tells whether the label should be editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the label should be editable by the approver. The default value is set to false.
- **Default value** *(string input)*:
  Set a default value to the label.

- **Label value restriction**
  - **Whitelist** *(string input multiple)*:
    The label value will have to be in the whitelist. Open the popup, enter the label value and press "enter" to add this value to the accepted value list. An empty whitelist means no restriction.
  - **Suggestions** *(string input multiple)*:
    Add suggestions that will be displayed to the user. The user will be able to choose one of these values or enter its own. Open the popup, enter your suggestions and press enter to add this value to the suggestions. An empty suggestions list means no restriction.
  - **Regex** *(regex)*:
    The label value will have to match the regex. Open the popup, enter the regular expression and click on the submit button to set the regex. An empty regex means no restrictions.
- **Computation rule** *(Computation rule input)*:
  Set the value of this label to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

You can delete 🗑 or reorder (drag and drop) ⠿ the label template.

Ownership policy

- **Owner**
  - **Mandatory** *(boolean)*:
    Specify if the certificate’s owner is mandatory when submitting a request.
  - **Editable by requester** *(boolean)*:
    Specify if the certificate’s owner can be overridden by the requester when submitting a request.
  - **Editable by approver** *(boolean)*:
    Specify if the certificate’s owner can be overridden by the requester when approving a request.
  - **Computation rule** *(Computation rule input)*:
    Set the value of the owner to the value of the evaluated computation rule. This value will override any other value including the user input.
- **Contact email**
  - **Mandatory** *(boolean)*:
    Specify if the certificate’s contact email is mandatory when submitting a request.
  - **Editable by requester** *(boolean)*:
    Specify if the certificate’s contact email can be overridden by the requester when submitting a request.
  - **Editable by approver** *(boolean)*:
    Specify if the certificate’s contact email can be overridden by the requester when approving a request.
  - **Default contact email** *(string input)*:
    Set a default contact email. This value must comply with the contact email restriction.

  - **Contact email restriction**

    - **Whitelist** *(string input multiple)*:
      The contact email will have to be in the whitelist. Open the popup, enter the email and press "enter" to add this value to the accepted whitelist. An empty whitelist means no restriction.

    - **Regex** *(regex)*:
      The contact email will have to match the regex. Open the popup, enter the regular expression and click on the submit button to set the regex. An empty regex means no restrictions.

  - **Computation rule** *(Computation rule input)*:
    Set the value of the contact email to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

- **Team**

  - **Mandatory** *(boolean)*:
    Specify if the certificate's team is mandatory when submitting a request.

  - **Editable by requester** *(boolean)*:
    Specify if the certificate's team can be overridden by the requester when submitting a request.

  - **Editable by approver** *(boolean)*:
    Specify if the certificate's team can be overridden by the requester when approving a request.

  - **Default team** *(string input)*:
    Set a default team. This value must comply with the team restriction.

  - **Team restriction**

    - **Whitelist** *(string input multiple)*:
      The team will have to be in the whitelist. Enter the team and press "enter" to add this value to the accepted whitelist. An empty whitelist means no restriction.

    - **Regex** *(regex)*:
      The team will have to match the regex. Open the popup, enter the regular expression and click on the submit button to set the regex. An empty regex means no restrictions.

  - **Computation rule** *(Computation rule input)*:
    Set the value of the team to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

Metadata policy *(overridable metadata)*

WARNING These metadata are technical metadata. They are used by Horizon or Third party connectors, updating them should be done with utmost care.

NOTE Metadata edition is not allowed on enroll.

NOTE Metadata edition is not available via the User Interface. It must be changed with

API, using horizon-cli.

You can allow the override of technical metadata by clicking ⊕ .

- **Metadata*** *(select)*:
  Select a metadata.
- **Editable by requester** *(boolean)*:
  Tells whether the metadata is editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the metadata is editable by the approver. The default value is set to false.

You can delete 🗑 a metadata policy. This will not delete the metadata but will make it non editable.

Notifications/Triggers

This section details how to configure notifications and triggers to perform actions on certificate and request lifecycle events.

Certificate lifecycle notifications

Notifications are sent when one of the following event is triggered by a certificate:

Enrollment	Revocation	Expire	Update	Migrate	Renew

Select a preexisting email, REST or groupware notification to associate it with an event.

Request lifecycle notifications

Notifications are sent when one of the following event is triggered by an Enroll/Revocation/Update/Migrate/Renew request:

Submit	Cancel	Revoke	Approve	Pending

Select a preexisting email, REST or groupware notification to associate it with an event.

NOTE Submit request events are not triggered when the user has the permission to perform the action directly.

Triggers

Horizon support the use of third-party triggers in the form of callbacks on specific events happening on the profile, giving a way to synchronize the third party repositories and Horizon.

- **Enrollment** *(select)*:
  Select the preexisting third party or MDM trigger(s) to call whenever a certificate is enrolled on this profile.

- **Renewal** *(select)*:
  Select the preexisting third party or MDM trigger(s) to call whenever a certificate is renewed on this profile.
- **Revocation** *(select)*:
  Select the preexisting third party or MDM trigger(s) to call whenever a certificate gets revoked on this profile.
- **Expire** *(select)*:
  Select the preexisting third party or MDM trigger(s) to call whenever a certificate expires on this profile.

The available triggers are the following:

AKV Triggers	AWS Triggers	F5 Triggers	LDAP Triggers	*On **WebRA** and **Intune PKCS** only:* Intune PKCS Triggers

5. Click on the save button.

You can edit ✎ , duplicate ⧉ or delete 🗑 the CRMP Profile.

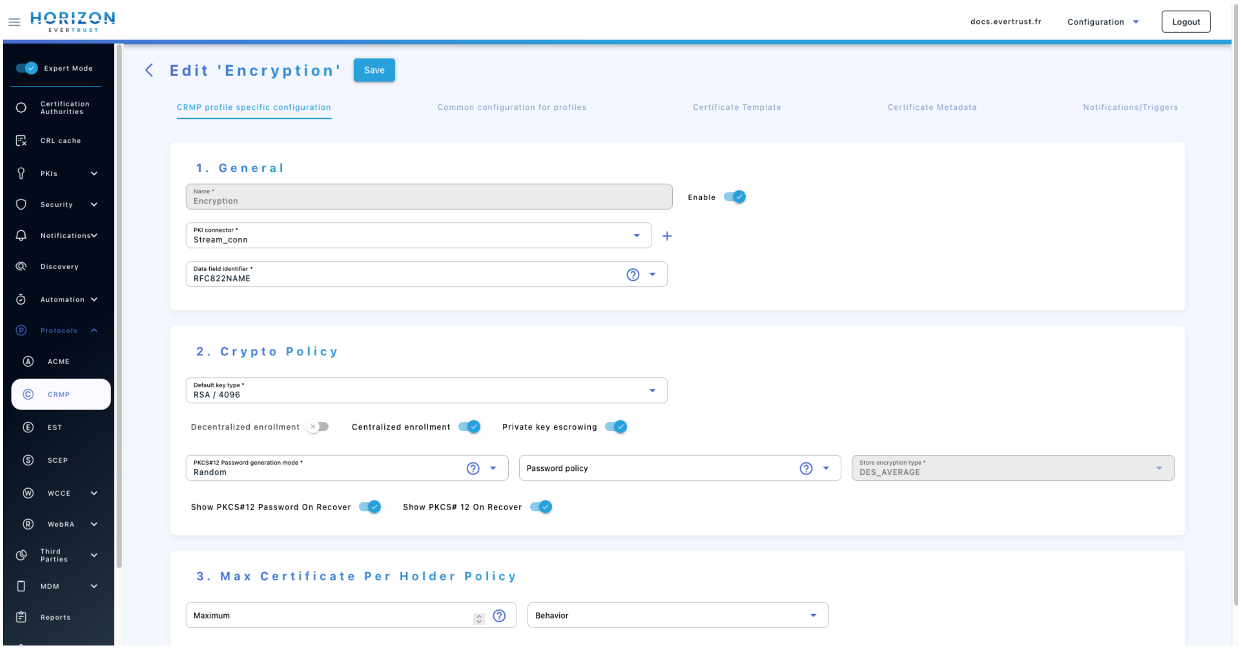
CAUTION You won’t be able to delete a CRMP Profile if a certificate is enrolled on it.

9.3.3. Enroll your first card with OpenTrust CMS

A step by step guide for a perfect integration between Horizon and OpenTrust CMS

I. Configure Horizon

1. Create your profiles.
   In **Configuration › Protocols › CRMP** you will have the possibility to setup your profiles.
   Let’s create three profiles, that will later result in 3 certificates for each user: an authentication certificate, a signing certificate and an encryption certificate.
   The first two will be decentralized profiles, and the encryption one will be centralized with escrow, so that we can always decrypt the user communications later. All configuration options are available in the profile section.



Edit 'Encryption'

1. General

Name *: Encryption

Enable ☑

PKI connector *: Stream_conn

Data field identifier *: RFC822NAME

2. Crypto Policy

Default key type *: RSA / 4096

Decentralized enrollment ☐ Centralized enrollment ☑ Private key escrowing ☑

PKCS#12 Password generation mode *: Random

Password policy

Store encryption type *: DES_AVERAGE

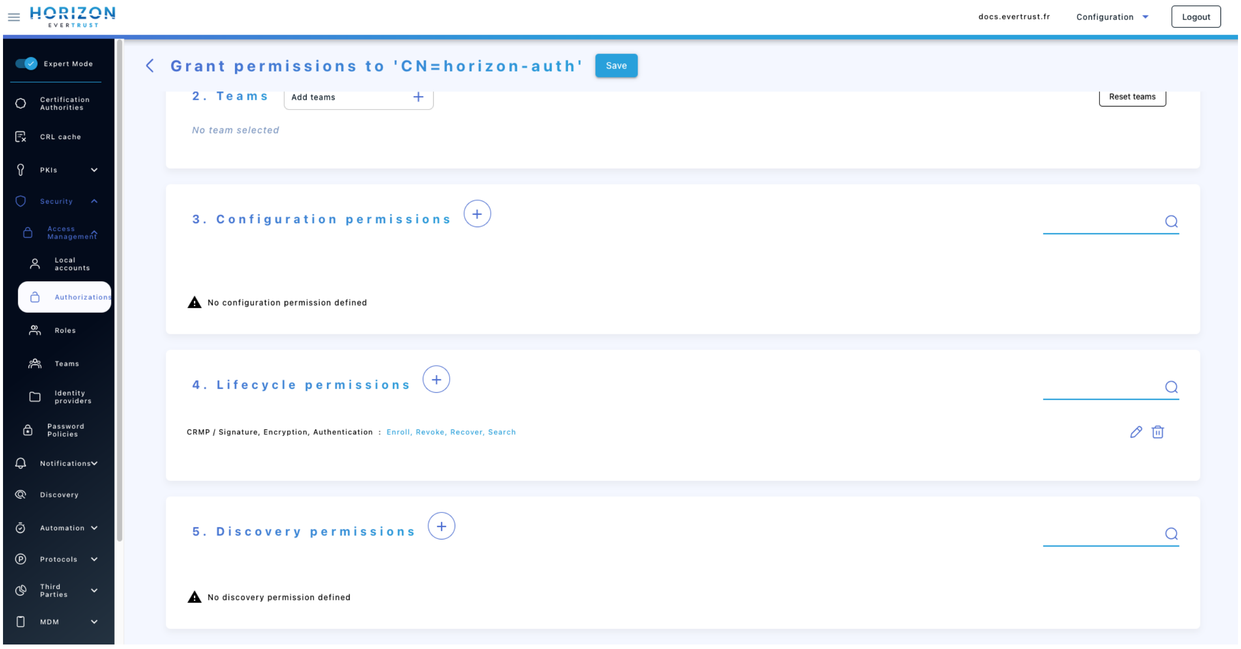
Show PKCS#12 Password On Recover ☑ Show PKCS# 12 On Recover ☑

3. Max Certificate Per Holder Policy

Maximum

Behavior

2. Create an account.
   OpenTrust CMS will need access to Horizon in order to manage your cards certificates.
   In order to do so, a certificate needs to be enrolled on a CA trusted by Horizon for client authentication.
   This certificate should be able to **enroll**, **revoke**, **recover** and **search** on the CRMP profiles you want it to manage.
   My certificate will here have for DN: `CN=horizon-auth`, and I will give it the appropriate rights.



Grant permissions to 'CN=horizon-auth'

2. Teams

No team selected

3. Configuration permissions

No configuration permission defined

4. Lifecycle permissions

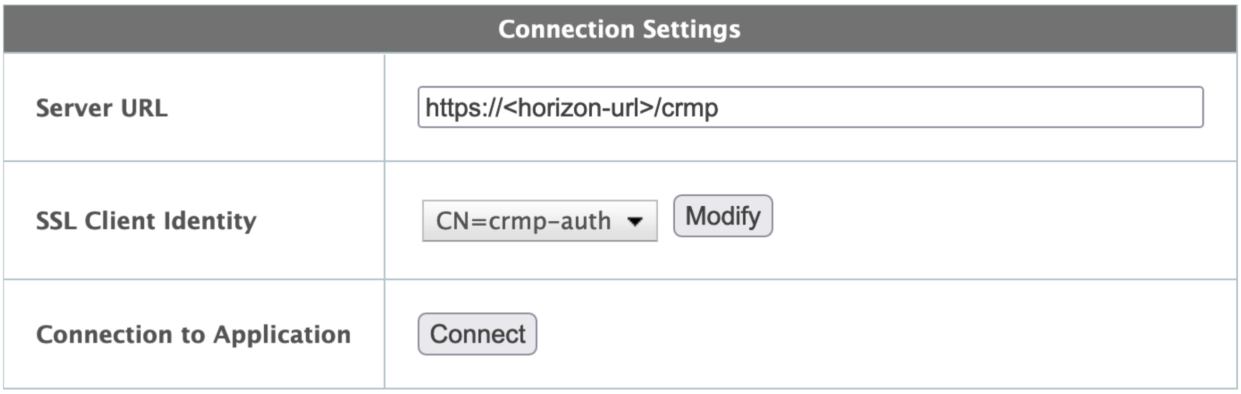
CRMP / Signature, Encryption, Authentication : Enroll, Revoke, Recover, Search

5. Discovery permissions

No discovery permission defined

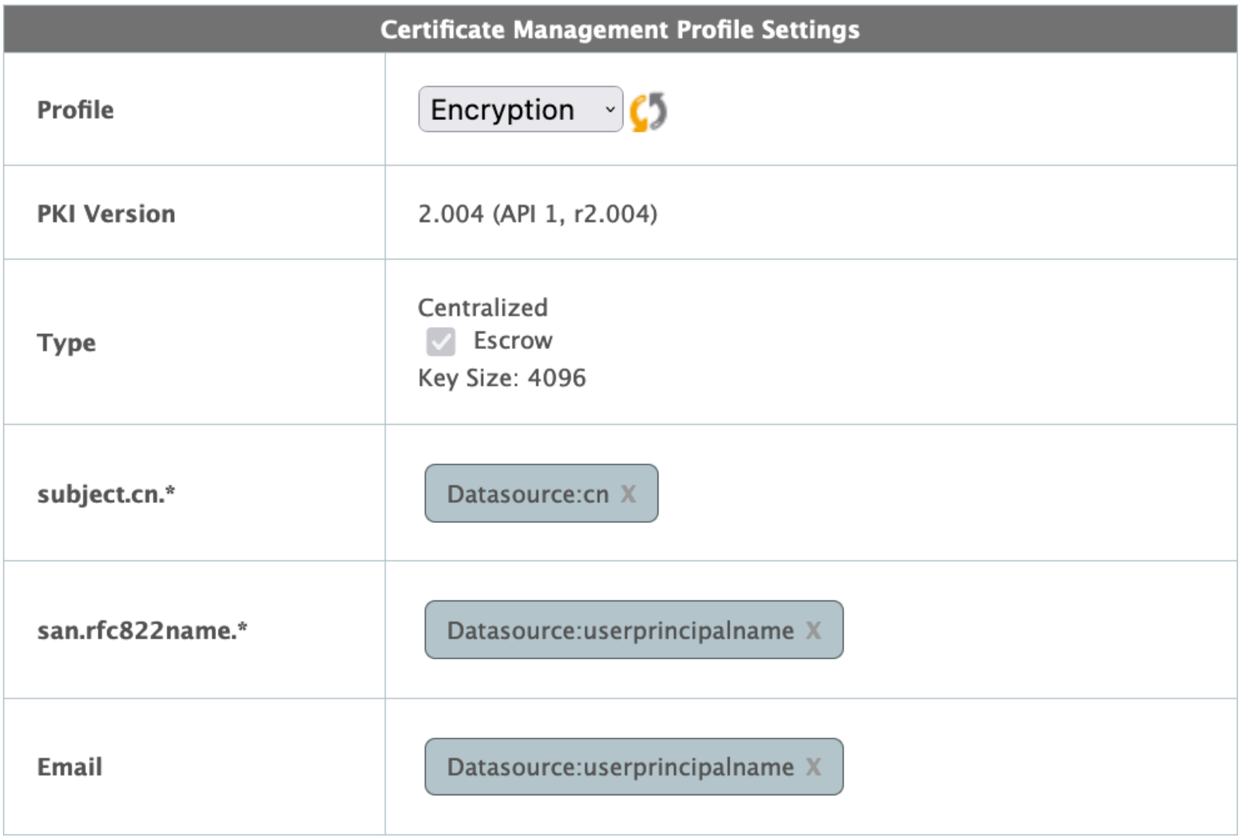
II. Configure your CMS applications

1. Connect your applications.
   For each of the profiles on Horizon, a **CRMP** application must be created (If **CRMP** is not available, it must be installed on your CMS: refer to OpenTrust CMS documentation).
   It first needs to be able to connect.
   The server url must be set to `https://<horizon-url>/crmp`.
   The SSL client identity must then be set to the certificate created in step I.2.



Connection Settings	
Server URL	https://<horizon-url>/crmp
SSL Client Identity	CN=crmp-auth ▾ Modify
Connection to Application	Connect

2. Map your applications.
   The information setup on Horizon will be displayed, and the fields can be mapped.
   The enrolled certificate on Horizon will be the result of the values mapped in the Horizon Fields on the left.
   It should be noted that some Horizon fields are indexed, but the CMS does not display numbers. They are ordered in the same order as on Horizon, with mandatory fields first and then optional fields.



Certificate Management Profile Settings	
Profile	Encryption ⌄
PKI Version	2.004 (API 1, r2.004)
Type	Centralized ☑ Escrow Key Size: 4096
subject.cn.*	Datasource:cn X
san.rfc822name.*	Datasource:userprincipalname X
Email	Datasource:userprincipalname X

CAUTION

Escrow: Due to a technical limitation in the CMS, for certificates that are escrowed, a field with technical name `userprincipalname` must be mapped to the selected `Data Field Identifier` in the CRMP Profile. Otherwise, the user will not be able to recover its certificates. The field `userprincipalname` must then be able to uniquely identify each user.

9.4. EST

9.4.1. EST Introduction

This section refers to the EST protocol, as described by RFC 7030.

9.4.2. EST Profile

This section details how to configure the EST Profile

Prerequisites

PKI Connector

How to configure EST Profile

1. Log in to Horizon Administration Interface.

2. Access EST Profile from the drawer or card: **Protocol › EST**.

3. Click on ⊕ .

4. Fill in the mandatory fields.

EST Specific Configuration

General

- **Name*** *(string input)*:
  Enter a meaningful profile name. It must be unique for each profile. Horizon use the name to identify the profile.
- **Enable** *(boolean)*:
  Tells whether the profile is enabled or not. The default value is set to true.
- **PKI Connector** *(string select)*:
  Select a PKI connector previously created.

Authorization and validation

- **Authorization mode** *(select)*:
  Select from the list.
- **Authorized**:
  - **Enable whitelist** *(boolean)*:
    Tells whether whitelist is enabled or not. The default value is set to false.
  - **CA*** *(select)*:
    Select a Certificate Authority previously created.

- **X509**:
  - **Enrollment CAs** *(select)*:
    Available only if mode at x509. Select a Certificate Authority previously created.
  - **Enable whitelist** *(boolean)*:
    Tells whether whitelist is enabled or not. The default value is set to false.
  - **CA*** *(select)*:
    Select a Certificate Authority previously created.
- **Challenge**:
  - **Password policy** *(select)*:
    Select a password policy previously created. It is used for the challenge generation.
  - **Enable whitelist** *(boolean)*:
    Tells whether whitelist is enabled or not. The default value is set to false.
  - **CA*** *(select)*:
    Select a Certificate Authority previously created.
- **Auto Validation**:

This enables auto validation.

- **CA*** *(select)*:
  Select a Certificate Authority previously created.

Max Certificate per Holder Policy

- **Maximum** *(int)*:
  When specified, define the maximum number of active certificates for a given Holder.
- **Behavior** *(select)*:
  What behavior to have when the maximum number is reached:
  - **revoke** the previous certificates.
  - **reject** the current request.
- **Revocation reason** *(select)*:
  When the revoke behavior is selected, the revocation reason to revoke the certificate with.

Renewal management

- **Renewal period** *(finite duration)*:
  Must be a valid finite duration.
- **Renewal CAs** *(select)*:
  Select a Certificate Authority previously created.

Crypto Policy

- **Default Key Type** *(select)*:
  Select the default type of key to generate when using centralized enrollment mode.
- **Authorized Key Types** *(multiselect)*:

Key Types that can be used for enrollment. An empty value means no restrictions.

- **Centralized enrollment** *(boolean)*:
  Tells whether the profile should be used with a centralized enrollment, i.e providing a PKCS#12. The default value is set to false.
  - **Private key escrowing** *(boolean)*:
    Tells whether the private key should be escrowed by Horizon. The default value is set to false.
    - **Show PKCS#12 Password On Recover** *(boolean)*:
      Tells whether the PKCS#12 password should be displayed on recover. The default value is set to false.
    - **Show PKCS#12 On Recover** *(boolean)*:
      Tells whether the PKCS#12 should be displayed on recover. The default value is set to false.
  - **PKCS#12 Password Mode*** *(select)*:
    Select how to generate PKCS#12 password:
    - **manual**: prompt the user to choose its password. This is the default behavior.
    - **random**: have the password generated on Horizon side.
  - **Password policy** *(select)*:
    Select a previously created password policy. It will be enforced on PKCS#12 password for recovery and centralized enrollments.
  - **Store encryption type*** *(select)*:
    Select an encryption algorithm from the list. The PKCS#12 will use this algorithm. The default value is set to DES_AVERAGE.
- **Decentralized enrollment** *(boolean)*:
  Tells whether the profile should be used with a decentralized enrollment mode, i.e CSR (PKCS#10) signing by the PKI. The default value is set to true.

Common configuration for profiles

Languages

You can add more languages by clicking ⊕ .

- **Language*** *(select)*:
  Select a language. Supported languages are:
  - **en**: English
  - **fr**: French
- **Display Name** *(string input)*:
  Enter a display name. This will be the localized name of this profile.
- **Description** *(string input)*:
  Enter a description. This will be displayed on the list view of the profiles.

You can delete 🗑 the localization.

Grading Policies

You can select grading policies that will grade your certificate for a quick overview of its quality. For more information about the inner working of the grading policies in Horizon, please refer to the grading rules page.

Requests time to live

Configure the time your requests have before expiring.

NOTE	After expiration, requests are stored for an additional 30 days. This can be changed using configuration files.

- **Enrollment request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Renewal request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Revocation request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Update request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Migration request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Recover request** *(finite duration)*:
  *Enabled on **escrow**:* Must be a valid finite duration. The default value is set to seven days.

Constraints

- **Allowed email domains** *(string input)*:
  Enter a valid regular expression that the inputted emails should match. This includes RFC822NAME and UPN SANs as well as the contact email

NOTE	This matches the domain of the email, not including anything before `@`.

- **Allowed DNS domains** *(string input)*:
  Enter a valid regular expression that the inputted domain should match.

CSR Data Mapping

1. Click on ⊕ to add a mapping.

2. Select a field and enter a value.

You can delete 🗑 the CSR Data Mapping.

Workflow

Auto-validation

Configure auto validation rules to avoid needing permissions configuration.

> **NOTE** A request permission must be available in order for the request to be created and then auto-validated. See workflows to modify request permissions.

1. To enable auto-validation, switch the profile mode to `auto-validation`

2. Add rules that will be evaluated on each request. For more details, see the validation rules reference.

3. Add the threshold. This is the number of rules that must pass in order for the request to be validated.

Data source flow

Configure which data sources to execute and in which order.

1. Select a data source to execute first, and fill its inputs with a computation rule.

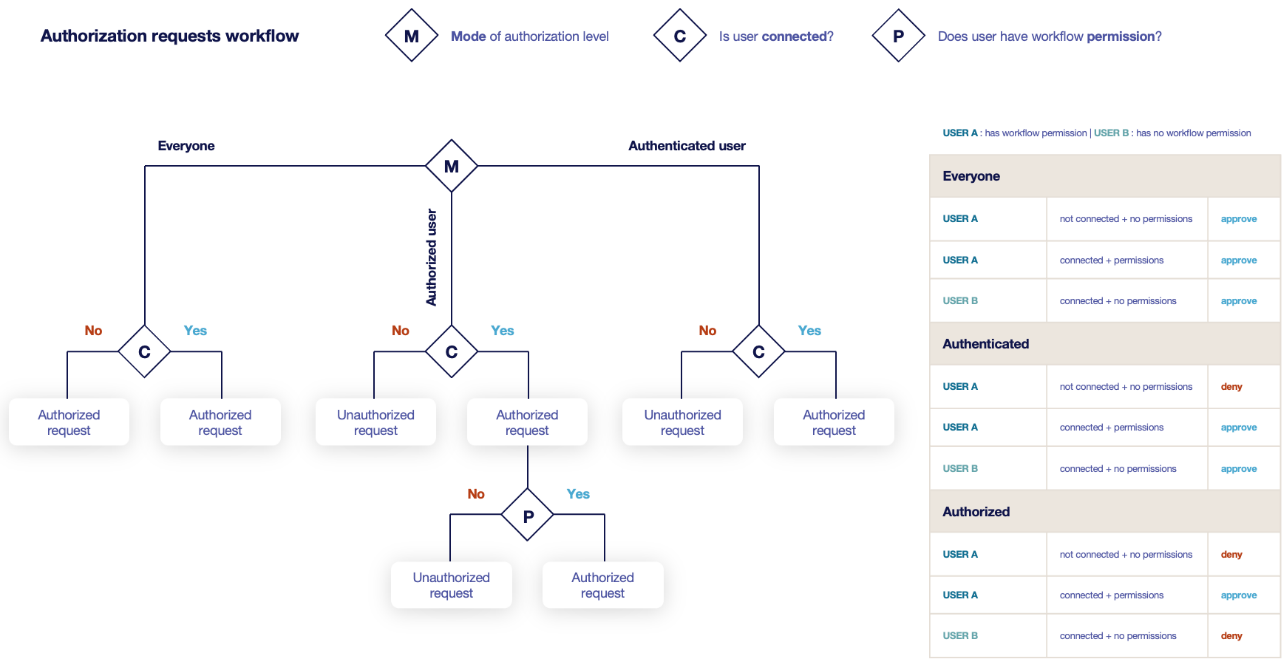
2. Add other data sources if needed. Each datasource input can use outputs from previously executed data sources.

3. All data sources output are available in computation rules throughout the certificate template and metadata.

Workflows builder

Configure custom rights for actions on this profile.

1. Select an authorization level for each workflow.



USER A : has workflow permission | USER B : has no workflow permission

Everyone		
USER A	not connected + no permissions	approve
USER A	connected + permissions	approve
USER B	connected + no permissions	approve
Authenticated		
USER A	not connected + no permissions	deny
USER A	connected + permissions	approve
USER B	connected + no permissions	approve
Authorized		
USER A	not connected + no permissions	deny
USER A	connected + permissions	approve
USER B	connected + no permissions	deny

- **Everyone**:
  No authentication is required.
- **Authenticated**:
  User has to be authenticated.
- **Authorized**:
  User has to be authenticated and have an explicit authorizations.

2. Select an access level for identity providers.

You can remove the access level for an identity provider by clicking on 'x'.

Owner-related permissions

These permissions apply to the owners of a certificate (team or owner). An owner can always request the following actions, but this permission allows them to perform the action without validation.

- **Revoke** *(boolean)*:
  Grant self revoke permission. The default value is set to false.
- **Revoke (pop)** *(boolean)*:
  Grant self revoke permission with owner being determined by Proof of Possession. This is used by horizon-cli. The default value is set to false.
- **Recover** *(boolean)*:
  Grant self recover permission. The default value is set to false.
- **Update** *(boolean)*:
  Grant self update permission. The default value is set to false.
- **Update (pop)** *(boolean)*:
  Grant self update permission with owner being determined by Proof of Possession. This is used by horizon-cli. The default value is set to false.

Certificate Template

This section details how to define a custom structure for the fields `subject DN`, `SAN` *&* `extensions` *of the requested certificate in order to match the configuration on the PKI side.*

NOTE Defining a template will use the CSR to fill the available field. A CSR with unexpected fields will be rejected. Using a template also disables CSR Data Mapping.

Subject DN composition

You can add more elements by clicking ⊕ .

- **Element*** *(select)*:
  Select an attribute from the elements list.
- **Mandatory** *(boolean)*:
  Should the element be mandatory. The default value is set to false.
- **Editable by requester** *(boolean)*:
  Tells whether the element should be editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the element should be editable by the approver. The default value is set to false.
- **Default value** *(string input)*:
  Set a default value to the element.
- **Regex** *(regex)*:
  Enter a regular expression that the element should match.
- **Computation rule** *(Computation rule input)*:
  Set the value of this element to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

You can remove an element by clicking the delete button 🗑 or reorder (drag and drop) ⠿ the Subject DN template.

CAUTION When a template is defined, at least one mandatory Common Name must be added to the DN Elements.

SAN composition

You can add more elements by clicking ⊕ .

- **Element*** *(select)*:
  Select an attribute from the element list.
- **Editable by requester** *(boolean)*:
  Tells whether the element should be editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the element should be editable by the approver. The default value is set to false.

- **Minimum** *(int)*:
  The minimum number of value that this SAN must have.
- **Maximum** *(int)*:
  The maximum number of value that this SAN must have.
- **Regex** *(regex)*:
  Enter a regular expression that the element should match.
- **Computation rule** *(Computation rule input)*:
  Set the value of this element to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

You can remove an element by clicking the delete button 🗑 or reorder (drag and drop) ⠿ the SAN template.

Extensions

You can add more elements by clicking ⊕ .

- **Element*** *(select)*:
  Select an attribute from the elements list.
- **Mandatory** *(boolean)*:
  Should the element be mandatory. The default value is set to false.
- **Editable by requester** *(boolean)*:
  Tells whether the element should be editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the element should be editable by the approver. The default value is set to false.
- **Default value** *(string input)*:
  Set a default value to the element.
- **Computation rule** *(Computation rule input)*:
  Set the value of this element to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

You can remove an element by clicking the delete button 🗑 or reorder (drag and drop) ⠿ the Extensions template.

CAUTION

When adding a SAN, a DN element or an Extension and making it mandatory, make sure to either give it a default value or a computation rule or make it editable, otherwise the template will be unusable.

Certificate Metadata

This section details how to define a custom structure for the labels, ownership policy and technical metadata, allowing certificates to hold rich information.

Labels

You can add more labels by clicking ⊕ .

- **Name** *(select)*:
  Select a preexisting label.
- **Mandatory** *(boolean)*:
  Should the label be mandatory. The default value is set to false.
- **Editable by requester** *(boolean)*:
  Tells whether the label should be editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the label should be editable by the approver. The default value is set to false.
- **Default value** *(string input)*:
  Set a default value to the label.
- **Label value restriction**
  - **Whitelist** *(string input multiple)*:
    The label value will have to be in the whitelist. Open the popup, enter the label value and press "enter" to add this value to the accepted value list. An empty whitelist means no restriction.
  - **Suggestions** *(string input multiple)*:
    Add suggestions that will be displayed to the user. The user will be able to choose one of these values or enter its own. Open the popup, enter your suggestions and press enter to add this value to the suggestions. An empty suggestions list means no restriction.
  - **Regex** *(regex)*:
    The label value will have to match the regex. Open the popup, enter the regular expression and click on the submit button to set the regex. An empty regex means no restrictions.
- **Computation rule** *(Computation rule input)*:
  Set the value of this label to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

You can delete 🗑 or reorder (drag and drop) ⠿ the label template.

Ownership policy

- **Owner**
  - **Mandatory** *(boolean)*:
    Specify if the certificate’s owner is mandatory when submitting a request.
  - **Editable by requester** *(boolean)*:
    Specify if the certificate’s owner can be overridden by the requester when submitting a request.
  - **Editable by approver** *(boolean)*:
    Specify if the certificate’s owner can be overridden by the requester when approving a request.

  - **Computation rule** *(Computation rule input)*:
    Set the value of the owner to the value of the evaluated computation rule. This value will override any other value including the user input.

- **Contact email**
  - **Mandatory** *(boolean)*:
    Specify if the certificate’s contact email is mandatory when submitting a request.
  - **Editable by requester** *(boolean)*:
    Specify if the certificate’s contact email can be overridden by the requester when submitting a request.
  - **Editable by approver** *(boolean)*:
    Specify if the certificate’s contact email can be overridden by the requester when approving a request.
  - **Default contact email** *(string input)*:
    Set a default contact email. This value must comply with the contact email restriction.
  - **Contact email restriction**
    - **Whitelist** *(string input multiple)*:
      The contact email will have to be in the whitelist. Open the popup, enter the email and press "enter" to add this value to the accepted whitelist. An empty whitelist means no restriction.
    - **Regex** *(regex)*:
      The contact email will have to match the regex. Open the popup, enter the regular expression and click on the submit button to set the regex. An empty regex means no restrictions.
  - **Computation rule** *(Computation rule input)*:
    Set the value of the contact email to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

- **Team**
  - **Mandatory** *(boolean)*:
    Specify if the certificate’s team is mandatory when submitting a request.
  - **Editable by requester** *(boolean)*:
    Specify if the certificate’s team can be overridden by the requester when submitting a request.
  - **Editable by approver** *(boolean)*:
    Specify if the certificate’s team can be overridden by the requester when approving a request.
  - **Default team** *(string input)*:
    Set a default team. This value must comply with the team restriction.
  - **Team restriction**
    - **Whitelist** *(string input multiple)*:
      The team will have to be in the whitelist. Enter the team and press "enter" to add this value to the accepted whitelist. An empty whitelist means no restriction.

        - **Regex** *(regex)*:
          The team will have to match the regex. Open the popup, enter the regular expression and click on the submit button to set the regex. An empty regex means no restrictions.

    - **Computation rule** *(Computation rule input)*:
      Set the value of the team to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

Metadata policy *(overridable metadata)*

WARNING These metadata are technical metadata. They are used by Horizon or Third party connectors, updating them should be done with utmost care.

NOTE Metadata edition is not allowed on enroll.

NOTE Metadata edition is not available via the User Interface. It must be changed with API, using horizon-cli.

You can allow the override of technical metadata by clicking ⊕ .

- **Metadata*** *(select)*:
  Select a metadata.
- **Editable by requester** *(boolean)*:
  Tells whether the metadata is editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the metadata is editable by the approver. The default value is set to false.

You can delete 🗑 a metadata policy. This will not delete the metadata but will make it non editable.

Notifications

This section details how to configure notifications on certificate and request lifecycle events.

Certificate lifecycle notifications

Notifications are sent when one of the following event is triggered by a certificate:

Enrollment	Revocation	Expire	Update	Migrate	Renew

Select a preexisting email, REST or groupware notification to associate it with an event.

Request lifecycle notifications

Notifications are sent when one of the following event is triggered by an Enroll/Revocation/Update/Migrate/Renew request:

Submit	Cancel	Revoke	Approve	Pending

Select a preexisting email, REST or groupware notification to associate it with an event.

NOTE Submit request events are not triggered when the user has the permission to perform the action directly.

5. Click on the save button.

You can edit ✎ , duplicate ⧉ or delete 🗑 the EST Profile.

CAUTION You won't be able to delete a EST Profile if this one is referenced somewhere else.

9.5. SCEP

9.5.1. SCEP Introduction

This section refers to the SCEP protocol, as described by RFC 8894.

9.5.2. SCEP Authorities

This section details how to configure SCEP Authorities.

The draft-nourse-scep-23 as well as RFC 8894 define how SCEP communications are secured. This involves using a SCEP Authority, which is a certificate and its associated private key, used to sign and encrypt communications between SCEP server and client.

Two setups are possible:

- the **CA mode** in which the SCEP Authority is a self-signed certificate. In that mode the SCEP server returns the self-signed certificate as `application/x-x509-ca-cert` when the client uses the `GetCaCert` call.
- the **RA mode** in which the SCEP Authority is a certificate signed by the CA that will issue certificates using the considered SCEP profile. In that mode, the SCEP server returns the SCEP Authority certificate and its issuing CA chain as `application/x-x509-ca-ra-cert` when the client uses the `GetCaCert` call.

Therefore, it is important in each SCEP or MDM Profile to align the SCEP mode with the characteristics of the SCEP Authority configured in the current section.

Prerequisites

- PKCS#12 containing the SCEP Authority certificate and private key. See above for explanation about the SCEP contents.

How to configure a SCEP Authority

SCEP Authorities are configured as credentials.

9.5.3. SCEP Profile

This section details how to configure the SCEP Profile

Prerequisites

PKI Connector	SCEP Authority

How to configure SCEP Profile

1. Log in to Horizon Administration Interface.

2. Access SCEP Profile from the drawer or card: **Protocol › SCEP**.

3. Click on ⊕ .

4. Fill in the mandatory fields.

SCEP Profile Specific Configuration

General

- **Name*** *(string input)*:
  Enter a meaningful profile name. It must be unique for each profile. Horizon use the name to identify the profile.
- **Enable** *(boolean)*:
  Tells whether the profile is enabled or not. The default value is set to true.
- **PKI Connector*** *(string select)*:
  Select a PKI connector previously created.
- **Authorize POST enrollment** *(boolean)*:
  Enable scep enrollment routes with HTTP POST method. The defaults value is set to false.
- **Authorization mode*** *(select)*:
  Select `Challenge` mode to allow enrollment using a pre validated request containing a challenge, `NDES` mode to use challenge validation but allow automatic request creation by a user with enroll permissions, `Authorized` to allow enrollment by a challenge containing credentials of a user with enroll permissions. In this mode, you can generate the credentials in the appropriate format by clicking on the shield icon. You will be asked to enter a username and password, then hit the 'Generate' button to display and/or copy the payload in the clipboard. Select `Auto Validation` to use auto validation.
- **Enable DN Whitelist*** *(boolean)*:
  Tells whether the DN whitelist is enabled or not. The default value is set to false.

SCEP protocol parameters

- **Mode*** *(select)*:
  Choose from the two modes RA or CA. The default value is set to RA.
- **SCEP Authority*** *(select)*:
  Select a previously created SCEP Authority.
- **CAPS*** *(select)*:
  Select a caps from the list. The default value is set to SHA.
- **Encryption algorithm*** *(select)*:
  Select an encryption algorithm from the list.
    - **Password policy** *(select)*:
      Select a previously created password policy. It is used for the challenge generation.

Max Certificate per Holder Policy

- **Maximum** *(int)*:
  When specified, define the maximum number of active certificates for a given Holder.
- **Behavior** *(select)*:
  What behavior to have when the maximum number is reached:
    - **revoke** the previous certificates.
    - **reject** the current request.
- **Revocation reason** *(select)*:
  When the revoke behavior is selected, the revocation reason to revoke the certificate with.

Renewal management

- **Renewal period** *(finite duration)*:
  Must be a valid finite duration.

Crypto Policy

- **Default Key Type** *(select)*:
  Key Type that will be used by horizon-cli in certificate enrollment.
- **Authorized Key Types** *(multiselect)*:
  Key Types that can be used for enrollment. An empty value means no restrictions.

Common configuration for profiles

Languages

You can add more languages by clicking ⊕ .

- **Language*** *(select)*:
  Select a language. Supported languages are:
    - **en**: English

  - **fr**: French
- **Display Name** *(string input)*:
  Enter a display name. This will be the localized name of this profile.
- **Description** *(string input)*:
  Enter a description. This will be displayed on the list view of the profiles.

You can delete 🗑 the localization.

Grading Policies

You can select grading policies that will grade your certificate for a quick overview of its quality. For more information about the inner working of the grading policies in Horizon, please refer to the grading rules page.

Requests time to live

Configure the time your requests have before expiring.

> **NOTE** After expiration, requests are stored for an additional 30 days. This can be changed using configuration files.

- **Enrollment request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Renewal request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Revocation request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Update request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Migration request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Recover request** *(finite duration)*:
  *Enabled on **escrow**:* Must be a valid finite duration. The default value is set to seven days.

Constraints

- **Allowed email domains** *(string input)*:
  Enter a valid regular expression that the inputted emails should match. This includes RFC822NAME and UPN SANs as well as the contact email

  > **NOTE** This matches the domain of the email, not including anything before `@`.

- **Allowed DNS domains** *(string input)*:
  Enter a valid regular expression that the inputted domain should match.

CSR Data Mapping

1. Click on ⊕ to add a mapping.

2. Select a field and enter a value.

You can delete 🗑 the CSR Data Mapping.

Workflow

Auto-validation

Configure auto validation rules to avoid needing permissions configuration.

> **NOTE** A request permission must be available in order for the request to be created and then auto-validated. See workflows to modify request permissions.

1. To enable auto-validation, switch the profile mode to `auto-validation`

2. Add rules that will be evaluated on each request. For more details, see the validation rules reference.

3. Add the threshold. This is the number of rules that must pass in order for the request to be validated.

Data source flow

Configure which data sources to execute and in which order.

1. Select a data source to execute first, and fill its inputs with a computation rule.

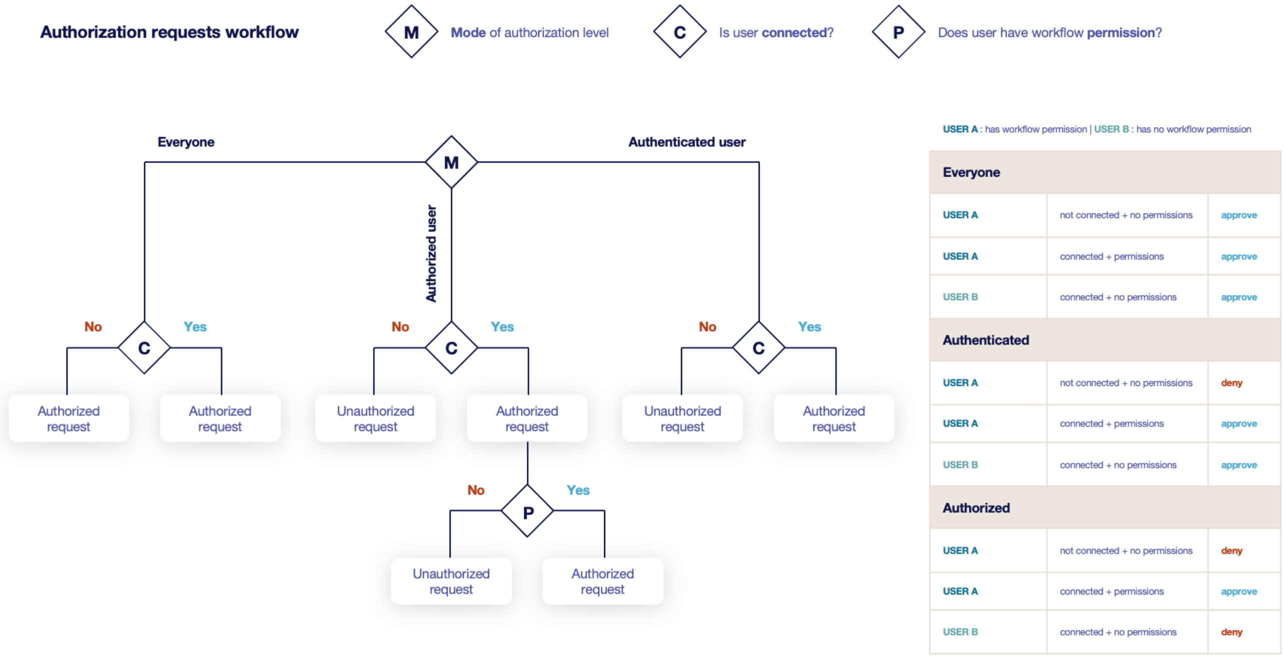
2. Add other data sources if needed. Each datasource input can use outputs from previously executed data sources.

3. All data sources output are available in computation rules throughout the certificate template and metadata.

Workflows builder

Configure custom rights for actions on this profile.

1. Select an authorization level for each workflow.



USER A : has workflow permission | USER B : has no workflow permission

Everyone		
USER A	not connected + no permissions	approve
USER A	connected + permissions	approve
USER B	connected + no permissions	approve
Authenticated		
USER A	not connected + no permissions	deny
USER A	connected + permissions	approve
USER B	connected + no permissions	approve
Authorized		
USER A	not connected + no permissions	deny
USER A	connected + permissions	approve
USER B	connected + no permissions	deny

- **Everyone**:
  No authentication is required.
- **Authenticated**:
  User has to be authenticated.
- **Authorized**:
  User has to be authenticated and have an explicit authorizations.

2. Select an access level for identity providers.

You can remove the access level for an identity provider by clicking on 'x'.

Owner-related permissions

These permissions apply to the owners of a certificate (team or owner). An owner can always request the following actions, but this permission allows them to perform the action without validation.

- **Revoke** *(boolean)*:
  Grant self revoke permission. The default value is set to false.
- **Revoke (pop)** *(boolean)*:
  Grant self revoke permission with owner being determined by Proof of Possession. This is used by horizon-cli. The default value is set to false.
- **Update** *(boolean)*:
  Grant self update permission. The default value is set to false.
- **Update (pop)** *(boolean)*:
  Grant self update permission with owner being determined by Proof of Possession. This is used by horizon-cli. The default value is set to false.

Certificate Template

This section details how to define a custom structure for the fields `subject DN`, `SAN` *&* `extensions` *of the requested certificate in order to match the configuration on the PKI side.*

> **NOTE** Defining a template will use the CSR to fill the available field. A CSR with unexpected fields will be rejected. Using a template also disables CSR Data Mapping.

Subject DN composition

You can add more elements by clicking ⊕ .

- **Element*** *(select)*:
  Select an attribute from the elements list.
- **Mandatory** *(boolean)*:
  Should the element be mandatory. The default value is set to false.
- **Editable by requester** *(boolean)*:
  Tells whether the element should be editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the element should be editable by the approver. The default value is set to false.
- **Default value** *(string input)*:
  Set a default value to the element.
- **Regex** *(regex)*:
  Enter a regular expression that the element should match.
- **Computation rule** *(Computation rule input)*:
  Set the value of this element to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

You can remove an element by clicking the delete button 🗑 or reorder (drag and drop) ⠿ the Subject DN template.

> **CAUTION** When a template is defined, at least one mandatory Common Name must be added to the DN Elements.

SAN composition

You can add more elements by clicking ⊕ .

- **Element*** *(select)*:
  Select an attribute from the element list.
- **Editable by requester** *(boolean)*:
  Tells whether the element should be editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the element should be editable by the approver. The default value is set to false.

- **Minimum** *(int)*:
  The minimum number of value that this SAN must have.
- **Maximum** *(int)*:
  The maximum number of value that this SAN must have.
- **Regex** *(regex)*:
  Enter a regular expression that the element should match.
- **Computation rule** *(Computation rule input)*:
  Set the value of this element to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

You can remove an element by clicking the delete button 🗑 or reorder (drag and drop) ⠿ the SAN template.

Extensions

You can add more elements by clicking ⊕ .

- **Element*** *(select)*:
  Select an attribute from the elements list.
- **Mandatory** *(boolean)*:
  Should the element be mandatory. The default value is set to false.
- **Editable by requester** *(boolean)*:
  Tells whether the element should be editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the element should be editable by the approver. The default value is set to false.
- **Default value** *(string input)*:
  Set a default value to the element.
- **Computation rule** *(Computation rule input)*:
  Set the value of this element to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

You can remove an element by clicking the delete button 🗑 or reorder (drag and drop) ⠿ the Extensions template.

CAUTION	When adding a SAN, a DN element or an Extension and making it mandatory, make sure to either give it a default value or a computation rule or make it editable, otherwise the template will be unusable.

Certificate Metadata

This section details how to define a custom structure for the labels, ownership policy and technical metadata, allowing certificates to hold rich information.

Labels

You can add more labels by clicking ⊕ .

- **Name** *(select)*:
  Select a preexisting label.
- **Mandatory** *(boolean)*:
  Should the label be mandatory. The default value is set to false.
- **Editable by requester** *(boolean)*:
  Tells whether the label should be editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the label should be editable by the approver. The default value is set to false.
- **Default value** *(string input)*:
  Set a default value to the label.
- **Label value restriction**
  - **Whitelist** *(string input multiple)*:
    The label value will have to be in the whitelist. Open the popup, enter the label value and press "enter" to add this value to the accepted value list. An empty whitelist means no restriction.
  - **Suggestions** *(string input multiple)*:
    Add suggestions that will be displayed to the user. The user will be able to choose one of these values or enter its own. Open the popup, enter your suggestions and press enter to add this value to the suggestions. An empty suggestions list means no restriction.
  - **Regex** *(regex)*:
    The label value will have to match the regex. Open the popup, enter the regular expression and click on the submit button to set the regex. An empty regex means no restrictions.
- **Computation rule** *(Computation rule input)*:
  Set the value of this label to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

You can delete 🗑 or reorder (drag and drop) ⠿ the label template.

Ownership policy

- **Owner**
  - **Mandatory** *(boolean)*:
    Specify if the certificate's owner is mandatory when submitting a request.
  - **Editable by requester** *(boolean)*:
    Specify if the certificate's owner can be overridden by the requester when submitting a request.
  - **Editable by approver** *(boolean)*:
    Specify if the certificate's owner can be overridden by the requester when approving a request.

    - **Computation rule** *(Computation rule input)*:
      Set the value of the owner to the value of the evaluated computation rule. This value will override any other value including the user input.

- **Contact email**
    - **Mandatory** *(boolean)*:
      Specify if the certificate’s contact email is mandatory when submitting a request.
    - **Editable by requester** *(boolean)*:
      Specify if the certificate’s contact email can be overridden by the requester when submitting a request.
    - **Editable by approver** *(boolean)*:
      Specify if the certificate’s contact email can be overridden by the requester when approving a request.
    - **Default contact email** *(string input)*:
      Set a default contact email. This value must comply with the contact email restriction.
    - **Contact email restriction**
        - **Whitelist** *(string input multiple)*:
          The contact email will have to be in the whitelist. Open the popup, enter the email and press "enter" to add this value to the accepted whitelist. An empty whitelist means no restriction.
        - **Regex** *(regex)*:
          The contact email will have to match the regex. Open the popup, enter the regular expression and click on the submit button to set the regex. An empty regex means no restrictions.
    - **Computation rule** *(Computation rule input)*:
      Set the value of the contact email to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

- **Team**
    - **Mandatory** *(boolean)*:
      Specify if the certificate’s team is mandatory when submitting a request.
    - **Editable by requester** *(boolean)*:
      Specify if the certificate’s team can be overridden by the requester when submitting a request.
    - **Editable by approver** *(boolean)*:
      Specify if the certificate’s team can be overridden by the requester when approving a request.
    - **Default team** *(string input)*:
      Set a default team. This value must comply with the team restriction.
    - **Team restriction**
        - **Whitelist** *(string input multiple)*:
          The team will have to be in the whitelist. Enter the team and press "enter" to add this value to the accepted whitelist. An empty whitelist means no restriction.

        - **Regex** *(regex)*:
          The team will have to match the regex. Open the popup, enter the regular expression and click on the submit button to set the regex. An empty regex means no restrictions.
    - **Computation rule** *(Computation rule input)*:
      Set the value of the team to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

Metadata policy *(overridable metadata)*

WARNING These metadata are technical metadata. They are used by Horizon or Third party connectors, updating them should be done with utmost care.

NOTE Metadata edition is not allowed on enroll.

NOTE Metadata edition is not available via the User Interface. It must be changed with API, using horizon-cli.

You can allow the override of technical metadata by clicking ⊕ .

- **Metadata*** *(select)*:
  Select a metadata.
- **Editable by requester** *(boolean)*:
  Tells whether the metadata is editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the metadata is editable by the approver. The default value is set to false.

You can delete 🗑 a metadata policy. This will not delete the metadata but will make it non editable.

Notifications

This section details how to configure notifications on certificate and request lifecycle events.

Certificate lifecycle notifications

Notifications are sent when one of the following event is triggered by a certificate:

Enrollment	Revocation	Expire	Update	Migrate	Renew

Select a preexisting email, REST or groupware notification to associate it with an event.

Request lifecycle notifications

Notifications are sent when one of the following event is triggered by an Enroll/Revocation/Update/Migrate/Renew request:

Submit	Cancel	Revoke	Approve	Pending

Select a preexisting email, REST or groupware notification to associate it with an event.

NOTE Submit request events are not triggered when the user has the permission to perform the action directly.

5. Click on the save button.

You can edit ✎, duplicate ⧉ or delete 🗑 the SCEP Profile.

CAUTION You won’t be able to delete a SCEP Profile if this one is referenced somewhere else.

9.6. WCCE

9.6.1. WCCE Introduction

This section details how to configure and consume the Windows Client Certificate Enrollment (WCCE) protocol.

Managing certificate lifecycle through the WCCE protocol involves up to three components:

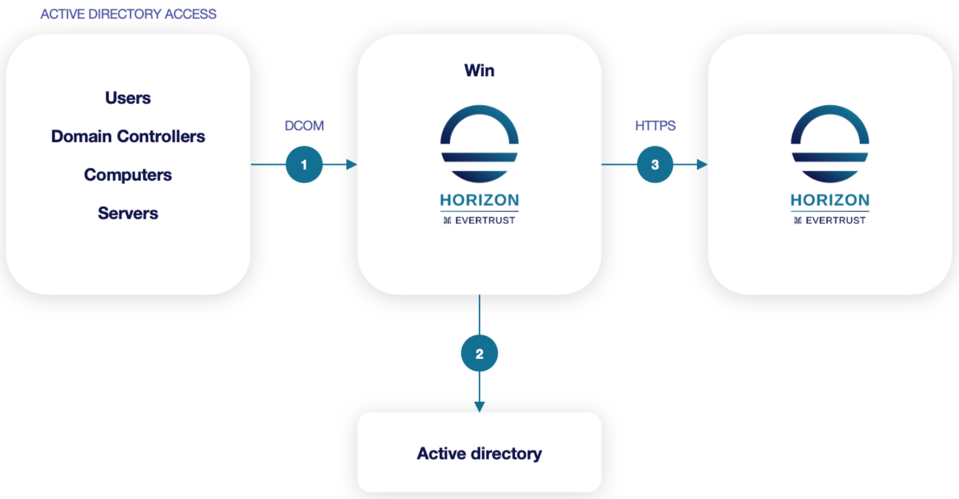
- Active Directory asset (domain controller, server, workstation, user) as WCCE Client;
- WinHorizon as the Active Directory enrollment service;
- Horizon as the WCCE proxy;

NOTE WCCE enrollment modes will be detailed later on.

The protocol paradigm can be described as follows: '**every Windows Active Directory member (machines, users) can use DCOM interfaces to interact with a CA to request certificate enrollment**'.

The following schema is a simplified workflow of an WCCE enrollment:

Simplified workflow of an WCCE enrollment



The protocol is based on the notion of Active Directory membership and configuration. Active Directory clients (such as machines and users) having rights on **Microsoft Certificate Templates** can use Active Directory **enrollment service** through DCOM interface to request certificate enrollment.

Horizon supports different WCCE enrollment modes:

- **Entity**: Certificate's elements are built using Active Directory content;
- **Enrollment On Behalf of Others (EOBO)**: Certificate signing request (CSR) is signed by one/many Certificate Enrollment Agent(s);
- **Trust request**: Certificate signature request (CSR) content is fully trust and certificate will be created using its content.

NOTE	For **Enrollment On Behalf of Others (EOBO)** enrollment mode, it is possible to configure a whitelist of **Authorized CAs** trusted as issuers of enrollment agent certificates.

Windows official resources

EverTrust WCCE implementation is based on official WCCE documentation provided by Microsoft:

- MS-WCCE: Windows Client Certificate Enrollment Protocol

Prerequisites

- WinHorizon should be installed using WinHorizon installation guide;
- WinHorizon and Active Directory should be configured using WinHorizon administration guide.

9.6.2. WCCE Forest

The first step is to register WCCE Forest on which you want to use WCCE protocol through Horizon.

Uses

MSAD Connector

How to configure WCCE Forest

1. Log in to Horizon Administration Interface.

2. Access WCCE Forest from the drawer or card: **Protocol** › **WCCE** › **Forest**.

3. Click on ⊕ .

4. Fill the mandatory fields.

- **Forest Name*** *(string input)*: Enter the Active Directory forest name.

5. Click on the save button.

You can duplicate ⧉ or delete 🗑 the WCCE Forest.

CAUTION You won’t be able to delete an WCCE Forest if it is referenced somewhere else.

9.6.3. WCCE Profile

*The second step details how to create and configure a WCCE Horizon profile. This profile is an **internal** Horizon profile.*

Uses

WCCE Template Mapping	WCCE Scheduled Task

Prerequisites

PKI Connector

How to configure a WCCE Profile

1. Log in to Horizon Administration Interface.

2. Access WCCE Profile from the drawer or card: **Protocol** › **WCCE** › **Profiles**.

3. Click on ⊕ .

4. Fill the mandatory fields.

WCCE Profile Specific Configuration

General

- **Name*** *(string input)*:
  Enter a meaningful profile name. It must be unique for each profile. Horizon uses the name to identify the profile. As the name will be part of an URL, it is advisable to use only lower case letters and dashes.
- **Enable*** *(boolean)*:
  Indicates whether the profile is enabled or not. The default value is set to true.
- **PKI Connector** *(string select)*:
  Select a PKI connector previously created.
- **Exchange certificate*** *(select)*:

  *Enabled on **escrow**:* Select a preexisting Exchange Certificate or create one with the ⊕ .

Crypto Policy

- **Authorized Key Types** *(multiselect)*:
  Key Types that can be used for enrollment. An empty value means no restrictions.
  - **Private key escrowing** *(boolean)*:
    Tells whether the private key should be escrowed by Horizon. The default value is set to false.

    > **NOTE** This can only be enabled using an Evertrust Stream PKI connector.

    - **Show PKCS#12 Password On Recover** *(boolean)*:
      Tells whether the PKCS#12 password should be displayed on recover. The default value is set to false.
    - **Show PKCS#12 On Recover** *(boolean)*:
      Tells whether the PKCS#12 should be displayed on recover. The default value is set to false.
  - **PKCS#12 Password Mode*** *(select)*:
    Select how to generate PKCS#12 password:
    - **manual**: prompt the user to choose its password. This is the default behavior.
    - **random**: have the password generated on Horizon side.
  - **Password policy** *(select)*:
    Select a previously created password policy. It will be enforced on PKCS#12 password for recovery and centralized enrollments.
  - **Store encryption type*** *(select)*:
    Select an encryption algorithm from the list. The PKCS#12 will use this algorithm. The default value is set to DES_AVERAGE.

Max Certificate per Holder Policy

- **Maximum** *(int)*:
  When specified, define the maximum number of active certificates for a given Holder.
- **Behavior** *(select)*:
  What behavior to have when the maximum number is reached:
  - **revoke** the previous certificates.
  - **reject** the current request.
- **Revocation reason** *(select)*:
  When the revoke behavior is selected, the revocation reason to revoke the certificate with.

Common configuration for profiles

Languages

You can add more languages by clicking ⊕ .

- **Language*** *(select)*:
  Select a language. Supported languages are:
  - **en**: English
  - **fr**: French
- **Display Name** *(string input)*:
  Enter a display name. This will be the localized name of this profile.
- **Description** *(string input)*:
  Enter a description. This will be displayed on the list view of the profiles.

You can delete 🗑 the localization.

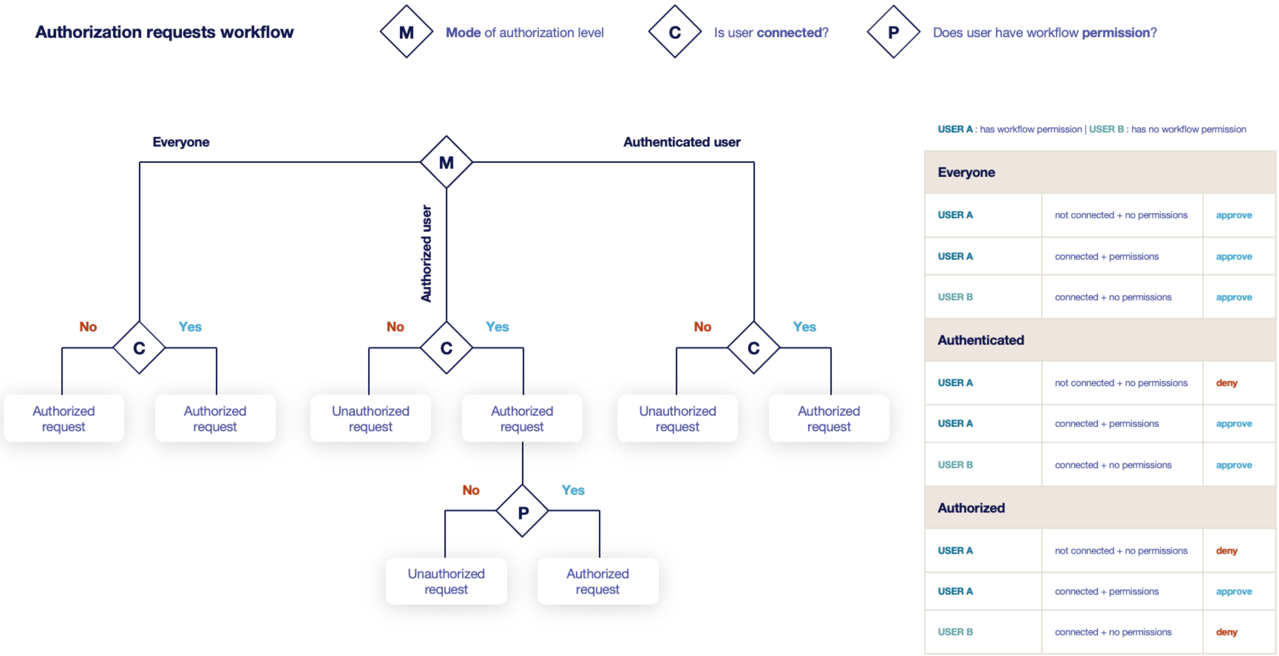
Grading Policies

You can select grading policies that will grade your certificate for a quick overview of its quality. For more information about the inner working of the grading policies in Horizon, please refer to the grading rules page.

Workflows builder

Configure custom rights for actions on this profile.

1. Select an authorization level for each workflow.



USER A : has workflow permission | USER B : has no workflow permission

Everyone		
USER A	not connected + no permissions	approve
USER A	connected + permissions	approve
USER B	connected + no permissions	approve
Authenticated		
USER A	not connected + no permissions	deny
USER A	connected + permissions	approve
USER B	connected + no permissions	approve
Authorized		
USER A	not connected + no permissions	deny
USER A	connected + permissions	approve
USER B	connected + no permissions	deny

- **Everyone**:
  No authentication is required.
- **Authenticated**:
  User has to be authenticated.
- **Authorized**:
  User has to be authenticated and have an explicit authorizations.

2. Select an access level for identity providers.

You can remove the access level for an identity provider by clicking on 'x'.

Requests time to live

Configure the time your requests have before expiring.

NOTE	After expiration, requests are stored for an additional 30 days. This can be changed using configuration files.

- **Enrollment request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Renewal request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Revocation request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Update request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Migration request*** *(finite duration)*:

Must be a valid finite duration. The default value is set to seven days.

- **Recover request** *(finite duration)*:
  Enabled on ***escrow****:* Must be a valid finite duration. The default value is set to seven days.

Owner-related permissions

These permissions apply to the owners of a certificate (team or owner). An owner can always request the following actions, but this permission allows them to perform the action without validation.

- **Revoke** *(boolean)*:
  Grant self revoke permission. The default value is set to false.
- **Recover** *(boolean)*:
  Grant self recover permission. The default value is set to false.
- **Update** *(boolean)*:
  Grant self update permission. The default value is set to false.

Constraints

- **Allowed email domains** *(string input)*:
  Enter a valid regular expression that the inputted emails should match. This includes RFC822NAME and UPN SANs as well as the contact email

  > **NOTE** This matches the domain of the email, not including anything before `@`.

- **Allowed DNS domains** *(string input)*:
  Enter a valid regular expression that the inputted domain should match.

CSR Data Mapping

1. Click on ⊕ to add a mapping.

2. Select a field and enter a value.

You can delete 🗑 the CSR Data Mapping.

Certificate Template

This section details how to define a custom structure for the fields `subject DN`, `SAN` *&* `extensions` *of the requested certificate in order to match the configuration on the PKI side.*

> **NOTE** Defining a template will use the CSR to fill the available field. A CSR with unexpected fields will be rejected. Using a template also disables CSR Data Mapping.

Subject DN composition

You can add more elements by clicking ⊕ .

- **Element*** *(select)*:
  Select an attribute from the elements list.
- **Mandatory** *(boolean)*:
  Should the element be mandatory. The default value is set to false.
- **Editable by requester** *(boolean)*:
  Tells whether the element should be editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the element should be editable by the approver. The default value is set to false.
- **Default value** *(string input)*:
  Set a default value to the element.
- **Regex** *(regex)*:
  Enter a regular expression that the element should match.
- **Computation rule** *(Computation rule input)*:
  Set the value of this element to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

You can remove an element by clicking the delete button 🗑 or reorder (drag and drop) ⠿ the Subject DN template.

CAUTION When a template is defined, at least one mandatory Common Name must be added to the DN Elements.

SAN composition

You can add more elements by clicking ⊕ .

- **Element*** *(select)*:
  Select an attribute from the element list.
- **Editable by requester** *(boolean)*:
  Tells whether the element should be editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the element should be editable by the approver. The default value is set to false.
- **Minimum** *(int)*:
  The minimum number of value that this SAN must have.
- **Maximum** *(int)*:
  The maximum number of value that this SAN must have.
- **Regex** *(regex)*:
  Enter a regular expression that the element should match.
- **Computation rule** *(Computation rule input)*:
  Set the value of this element to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

You can remove an element by clicking the delete button 🗑 or reorder (drag and drop) ⠿ the SAN template.

Extensions

You can add more elements by clicking ⊕ .

- **Element*** *(select)*:
  Select an attribute from the elements list.
- **Mandatory** *(boolean)*:
  Should the element be mandatory. The default value is set to false.
- **Editable by requester** *(boolean)*:
  Tells whether the element should be editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the element should be editable by the approver. The default value is set to false.
- **Default value** *(string input)*:
  Set a default value to the element.
- **Computation rule** *(Computation rule input)*:
  Set the value of this element to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

You can remove an element by clicking the delete button 🗑 or reorder (drag and drop) ⠿ the Extensions template.

CAUTION When adding a SAN, a DN element or an Extension and making it mandatory, make sure to either give it a default value or a computation rule or make it editable, otherwise the template will be unusable.

Certificate Metadata

This section details how to define a custom structure for the labels, ownership policy and technical metadata, allowing certificates to hold rich information.

Labels

You can add more labels by clicking ⊕ .

- **Name** *(select)*:
  Select a preexisting label.
- **Mandatory** *(boolean)*:
  Should the label be mandatory. The default value is set to false.
- **Editable by requester** *(boolean)*:
  Tells whether the label should be editable by the requester. The default value is set to false.

- **Editable by approver** *(boolean)*:
  Tells whether the label should be editable by the approver. The default value is set to false.
- **Default value** *(string input)*:
  Set a default value to the label.
- **Label value restriction**
  - **Whitelist** *(string input multiple)*:
    The label value will have to be in the whitelist. Open the popup, enter the label value and press "enter" to add this value to the accepted value list. An empty whitelist means no restriction.
  - **Suggestions** *(string input multiple)*:
    Add suggestions that will be displayed to the user. The user will be able to choose one of these values or enter its own. Open the popup, enter your suggestions and press enter to add this value to the suggestions. An empty suggestions list means no restriction.
  - **Regex** *(regex)*:
    The label value will have to match the regex. Open the popup, enter the regular expression and click on the submit button to set the regex. An empty regex means no restrictions.
- **Computation rule** *(Computation rule input)*:
  Set the value of this label to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

You can delete 🗑 or reorder (drag and drop) ⠿ the label template.

Ownership policy

- **Owner**
  - **Mandatory** *(boolean)*:
    Specify if the certificate’s owner is mandatory when submitting a request.
  - **Editable by requester** *(boolean)*:
    Specify if the certificate’s owner can be overridden by the requester when submitting a request.
  - **Editable by approver** *(boolean)*:
    Specify if the certificate’s owner can be overridden by the requester when approving a request.
  - **Computation rule** *(Computation rule input)*:
    Set the value of the owner to the value of the evaluated computation rule. This value will override any other value including the user input.
- **Contact email**
  - **Mandatory** *(boolean)*:
    Specify if the certificate’s contact email is mandatory when submitting a request.
  - **Editable by requester** *(boolean)*:
    Specify if the certificate’s contact email can be overridden by the requester when submitting a request.

  - **Editable by approver** *(boolean)*:
    Specify if the certificate’s contact email can be overridden by the requester when approving a request.

  - **Default contact email** *(string input)*:
    Set a default contact email. This value must comply with the contact email restriction.

  - **Contact email restriction**

    - **Whitelist** *(string input multiple)*:
      The contact email will have to be in the whitelist. Open the popup, enter the email and press "enter" to add this value to the accepted whitelist. An empty whitelist means no restriction.

    - **Regex** *(regex)*:
      The contact email will have to match the regex. Open the popup, enter the regular expression and click on the submit button to set the regex. An empty regex means no restrictions.

  - **Computation rule** *(Computation rule input)*:
    Set the value of the contact email to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

- **Team**

  - **Mandatory** *(boolean)*:
    Specify if the certificate’s team is mandatory when submitting a request.

  - **Editable by requester** *(boolean)*:
    Specify if the certificate’s team can be overridden by the requester when submitting a request.

  - **Editable by approver** *(boolean)*:
    Specify if the certificate’s team can be overridden by the requester when approving a request.

  - **Default team** *(string input)*:
    Set a default team. This value must comply with the team restriction.

  - **Team restriction**

    - **Whitelist** *(string input multiple)*:
      The team will have to be in the whitelist. Enter the team and press "enter" to add this value to the accepted whitelist. An empty whitelist means no restriction.

    - **Regex** *(regex)*:
      The team will have to match the regex. Open the popup, enter the regular expression and click on the submit button to set the regex. An empty regex means no restrictions.

  - **Computation rule** *(Computation rule input)*:
    Set the value of the team to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

Metadata policy *(overridable metadata)*

WARNING These metadata are technical metadata. They are used by Horizon or Third party connectors, updating them should be done with utmost care.

NOTE Metadata edition is not allowed on enroll.

NOTE Metadata edition is not available via the User Interface. It must be changed with API, using horizon-cli.

You can allow the override of technical metadata by clicking ⊕ .

- **Metadata*** *(select)*:
  Select a metadata.
- **Editable by requester** *(boolean)*:
  Tells whether the metadata is editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the metadata is editable by the approver. The default value is set to false.

You can delete 🗑 a metadata policy. This will not delete the metadata but will make it non editable.

Notifications/Triggers

This section details how to configure notifications and triggers to perform actions on certificate and request lifecycle events.

Certificate lifecycle notifications

Notifications are sent when one of the following event is triggered by a certificate:

Enrollment	Revocation	Expire	Update	Migrate	Renew

Select a preexisting email, REST or groupware notification to associate it with an event.

Request lifecycle notifications

Notifications are sent when one of the following event is triggered by an Enroll/Revocation/Update/Migrate/Renew request:

Submit	Cancel	Revoke	Approve	Pending

Select a preexisting email, REST or groupware notification to associate it with an event.

NOTE Submit request events are not triggered when the user has the permission to perform the action directly.

Triggers

Horizon support the use of third-party triggers in the form of callbacks on specific events happening on the profile, giving a way to synchronize the third party repositories and Horizon.

- **Enrollment** *(select)*:
  Select the preexisting third party or MDM trigger(s) to call whenever a certificate is enrolled on this profile.
- **Renewal** *(select)*:
  Select the preexisting third party or MDM trigger(s) to call whenever a certificate is renewed on this profile.
- **Revocation** *(select)*:
  Select the preexisting third party or MDM trigger(s) to call whenever a certificate gets revoked on this profile.
- **Expire** *(select)*:
  Select the preexisting third party or MDM trigger(s) to call whenever a certificate expires on this profile.

The available triggers are the following:

AKV Triggers	AWS Triggers	F5 Triggers	LDAP Triggers	*On **WebRA** and **Intune PKCS** only:* Intune PKCS Triggers

5. Click on the save button.

You can edit ✎ , duplicate ⧉ or delete 🗑 the WCCE Profile .

CAUTION You won't be able to delete a WCCE Profile if this one is referenced somewhere else.

9.6.4. WCCE Template Mapping

The third and last step is to configure mapping between Microsoft Certificate Template configured on Active Directory and Horizon WCCE profile. A mapping is created using a specific enrollment mode. As a result of this mapping, every Microsoft Certificate Template can issue certificate from different PKI (using PKI connector of WCCE profile associated to Microsoft Certificate Template).

Prerequisites

WCCE Forest	WCCE Profile

How to configure WCCE Template Mapping

1. Log in to Horizon Administration Interface.

2. Access WCCE Forest from the drawer or card: **Protocol › WCCE › Forest**.

3. Identify the section corresponds to the forest for which you want to add mapping. Click on + button.

4. Fill the mandatory fields.

- **Microsoft Template Name*** *(string input)*:
  Enter the Microsoft Certificate Template name created on Active Directory side.
- **Enrollment mode** *(select)*:
  Specify the enrollment mode of this mapping.
- **EOBO CAs** *(select)*:
  Specify the CA(s) to use for EOBO enrolment.
- **Profile*** *(select)*:
  Select a previously created WCCE profile.

5. Click on the save button.

You can edit ✏ or delete the WCCE Template mapping.

9.6.5. WCCE Test enrollment

This section details how to use the Microsoft Management Console (MMC) to manually retrieve a certificate through WCCE using different enrollment modes. If you want to enroll **machine certificate** you need to perform the following actions using Administrator Account.

1. Launch `mmc.exe`

2. Click on **File** > **Add/Remove or Remove Snap-ins**

3. On the left panel, click on **Certificates** then **Add**

> **NOTE**
>
> If you don’t have administrative privileges, the **User certificate store** will be automatically chosen. If your account has administrative privileges, it will be prompted a window to choose Microsoft Certificate Store to use. If you want to enroll **User** certificate please chose **My user account**. If you want to enroll **Machine** certificate (computer or IIS for example) please chose **Computer account**.

4. Navigate to **Personal › Certificates**

5. Right click on Windows and chose **All tasks › Request certificate**

6. Click on **Next**

7. On the next step, let default enrollment policy configuration, then click on **Next**

The next step lists all Microsoft Certificate Templates on which you have enrollment rights. The Microsoft Certificate template selection and last parts of this testing procedure are specific to the enrollment mode you want to perform.

Please refer to the proper section below.

Requesting a certificate using 'Entity' enrollment mode

8. Select the Microsoft Certificate Template configured on Horizon side as a part of a **Template Mapping** using **Entity** enrollment mode. Click on **Next**

9. Click on **Enroll** to request Enrollment.

10. Enrollment is requested to WinHorizon. Few seconds later, if enrollment is successful it will be displayed **STATUS: Succeeded**. Click on **Finish**.

11. Your certificate is displayed and available.

Requesting a certificate using 'Enrollment On Behalf of Others' enrollment mode

8. Identify the Microsoft Certificate Template configured on Horizon side as a part of a **Template Mapping** using **Enrollment On Behalf of Others (EOBO)** enrollment mode. Click on **Details** then **Properties**.

9. Navigate to **Extensions** tab and select **Enrollment Agent Certificate** (to be used to sign Certificate Request). Click on **OK**.

10. Click on **Enroll** to request Enrollment.

11. Enrollment is requested to WinHorizon. Few seconds later, if enrollment is successful it will be displayed **STATUS: Succeeded**. Click on **Finish**.

12. Your certificate is displayed and available.

Requesting a certificate using 'Trust request' enrollment mode

8. Identify the Microsoft Certificate Template configured on Horizon side as a part of a **Template Mapping** using **Trust request** enrollment mode. Click on **Details** then **Properties**.

9. Navigate to **Subject** tab to build your Certificate request manually. Click on **OK**.

10. Click on **Enroll** to request Enrollment.

11. Enrollment is requested to WinHorizon. Few seconds later, if enrollment is successful it will be displayed **STATUS: Succeeded**. Click on **Finish**.

12. Your certificate is displayed and available.

9.6.6. WCCE MSAD Connector

This section details how to to configure the Microsoft Active Directory Connectors.

Uses

WCCE Scheduled Task

Prerequisites

WCCE Forest

How to configure an MSAD Connector

1. Log in to Horizon Administration Interface.

2. Access MSAD Connectors from the drawer or card: **Protocol › WCCE › MSAD Connectors**.

3. Click on ⊕ .

4. Fill in the mandatory fields.

General

- **Name*** *(select)*:
  Select the Active Directory Forrest you want to use to set up the connector.
- **Hostname*** *(string input)*:
  DNS name or IP of the Active Directory domain.
- **Port** *(string input)*:
  Port to connect to the Active Directory. The default value is set to 636.
- **Proxy** *(string select)*:
  Select a proxy to connect to the Active Directory, if needed.
- **LDAP Credentials*** *(select)*:
  Select `Login` credentials containing the DN and password of the Active Directory account. Must have right privileges to browse and list objects.
- **Timeout*** *(finite duration)*:
  The time before Horizon stop trying to connect to Active Directory. Must be a valid finite duration.
- **Max stored certificate per holder** *(int)*:
  When specified, define the maximum number of active certificates for a given Holder.

Assets identification

- **Base DN*** *(string input)*:
  It can be the root of your domain or a restriction.
- **LDAP Filter** *(string input)*:
  This filter must respect LDAP filter syntax.

Actor management

These configuration elements mainly define the number of authorized interactions with the remote service on a defined period. For example, one needs to ensure that the remote service will not be requested more than 5 times per 3 seconds. *Throttle parallelism* defines the number of times and *Throttle duration* the period of time. Therefore, on the above example, throttle parallelism would be set to 5 and throttle duration would be set to 3 seconds.

- **Throttle duration*** *(finite duration)*:
  The default value is set to 3 seconds. Must be a valid finite duration.
- **Throttle parallelism*** *(int)*:
  The default value is set to 3.

5. Click on the save button.

You can update ✎ or delete 🗑 the MSAD Connector.

CAUTION You won’t be able to delete a MSAD Connector if this one is referenced somewhere else.

9.6.7. WCCE Scheduled Tasks

This section details how to schedule tasks that will run periodically on your WCCE profiles. You will be able to use MSAD Connector to browse Active Directory and retrieve changes (basically computer removal) to trigger certificate revocation. This mechanism works using comparison between Active Directory content (using MSAD connector) and Horizon certificate list based on a specific WCCE profile. If Horizon has a certificate for a holder that does not exist on Active Directory side a revocation will be triggered automatically.

Prerequisites

WCCE Forest	WCCE Profile	WCCE MSAD Connector

How to configure WCCE Scheduled Tasks

1. Log in to Horizon Administration Interface.

2. Access WCCE scheduled tasks from the drawer or card: **Protocol › WCCE › Scheduled Tasks**.

3. Click on ⊕ .

4. Fill the mandatory fields.

- **WCCE Profile*** *(select)*:
  Select the target WCCE profile.
- **Target Connector*** *(select)*:
  Select the MSAD connector to use as **golden source** of active Active Directory objects.

- **Cron scheduling in Quartz format** *(cron expression)*:
  Enter a Cron scheduling expression (in Quartz format). Default value is every 5 hours.
- **Revoke** *(boolean)*:
  If true, will revoke all certificate that do not exist on the AD side.
- **Dry run** *(boolean)*:
  If enabled, revocation actions will not be performed. Instead, a message will be logged, explaining what would have been done.

5. Click on the save button.

You can run ▷ , update ✎ or delete 🗑 the Schedules Tasks.

9.7. WebRA

9.7.1. WebRA Introduction

WebRA is a powerful protocol designed by EverTrust.It allows a validation process with edition of all certificate fields, to perform enrollments with user friendly web interfaces on Horizon Registration Authority portal.

9.7.2. WebRA Profile

This section details how to configure the WebRA Profile.

Required By

WebRA Scheduled Task

Prerequisites

PKI Connector

How to configure WebRA Profile

1. Log in to Horizon Administration Interface.

2. Access WebRA Profiles from the drawer or card: **Protocols › WebRA › Profiles**.

3. Click on ⊕ .

4. Fill in the mandatory fields.

Profile specific configuration

General

- **Name*** *(string input)*:
  Enter a meaningful profile name, this setting will be the profile identifier. It must be unique for each profile.
- **Enable** *(boolean)*:
  Should the profile be enabled. The default value is set to true.
- **PKI Connector** *(string select)*:
  Select a previously created PKI connector.
- **Authorization Mode*** *(select)*:
  This concern enrollment requests only: Select `Authorized` to use the configured permissions, `Auto Validation` to use auto validation or `Auto Validation → Authorized` to use auto validation then fallback on the configured permissions.

Crypto Policy

- **Default Key Type** *(select)*:
  Select the default type of key to generate when using centralized enrollment mode.
- **Authorized Key Types** *(multiselect)*:
  Key Types that can be used for enrollment. An empty value means no restrictions.
- **Centralized enrollment** *(boolean)*:
  Tells whether the profile should be used with a centralized enrollment, i.e providing a PKCS#12. The default value is set to false.
  - **Private key escrowing** *(boolean)*:
    Tells whether the private key should be escrowed by Horizon. The default value is set to false.
    - **Show PKCS#12 Password On Enroll** *(boolean)*:
      Tells whether the PKCS#12 password should be displayed on enroll. The default value is set to false.
    - **Show PKCS#12 On Enroll** *(boolean)*:
      Tells whether the PKCS#12 should be displayed on enroll. The default value is set to false.
    - **Show PKCS#12 Password On Recover** *(boolean)*:
      Tells whether the PKCS#12 password should be displayed on recover. The default value is set to false.
    - **Show PKCS#12 On Recover** *(boolean)*:
      Tells whether the PKCS#12 should be displayed on recover. The default value is set to false.
  - **PKCS#12 Password Mode*** *(select)*:
    Select how to generate PKCS#12 password:
    - **manual**: prompt the user to choose its password. This is the default behavior.
    - **random**: have the password generated on Horizon side.
  - **Password policy** *(select)*:
    Select a previously created password policy. It will be enforced on PKCS#12 password for

    recovery and centralized enrollments.

    - **Store encryption type*** *(select)*:
      Select an encryption algorithm from the list. The PKCS#12 will use this algorithm. The default value is set to DES_AVERAGE.

- **Decentralized enrollment** *(boolean)*:
  Tells whether the profile should be used with a decentralized enrollment mode, i.e CSR (PKCS#10) signing by the PKI. The default value is set to true.

Max Certificate per Holder Policy

- **Maximum** *(int)*:
  When specified, define the maximum number of active certificates for a given holder.
- **Behavior** *(select)*:
  What behavior to have when the maximum number is reached:
    - **revoke** the previous certificates.
    - **reject** the current request.
- **Revocation reason** *(select)*:
  When the revoke behavior is selected, the revocation reason to revoke the certificate with.

Common configuration for profiles

Languages

You can add more languages by clicking ⊕ .

- **Language*** *(select)*:
  Select a language. Supported languages are:
    - **en**: English
    - **fr**: French
- **Display Name** *(string input)*:
  Enter a display name. This will be the localized name of this profile.
- **Description** *(string input)*:
  Enter a description. This will be displayed on the list view of the profiles.

You can delete 🗑 the localization.

Grading Policies

You can select grading policies that will grade your certificate for a quick overview of its quality. For more information about the inner working of the grading policies in Horizon, please refer to the grading rules page.

Requests time to live

Configure the time your requests have before expiring.

NOTE After expiration, requests are stored for an additional 30 days. This can be changed using configuration files.

- **Enrollment request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Renewal request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Revocation request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Update request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Migration request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Recover request** *(finite duration)*:
  *Enabled on **escrow**:* Must be a valid finite duration. The default value is set to seven days.

CSR Data Mapping

1. Click on ⊕ to add a mapping.

2. Select a field and enter a value.

You can delete 🗑 the CSR Data Mapping.

Workflow

Auto-validation

Configure auto validation rules to avoid needing permissions configuration.

NOTE A request permission must be available in order for the request to be created and then auto-validated. See workflows to modify request permissions.

1. To enable auto-validation, switch the profile mode to `auto-validation`

2. Add rules that will be evaluated on each request. For more details, see the validation rules reference.

3. Add the threshold. This is the number of rules that must pass in order for the request to be validated.

Data source flow

Configure which data sources to execute and in which order.

1. Select a data source to execute first, and fill its inputs with a computation rule.

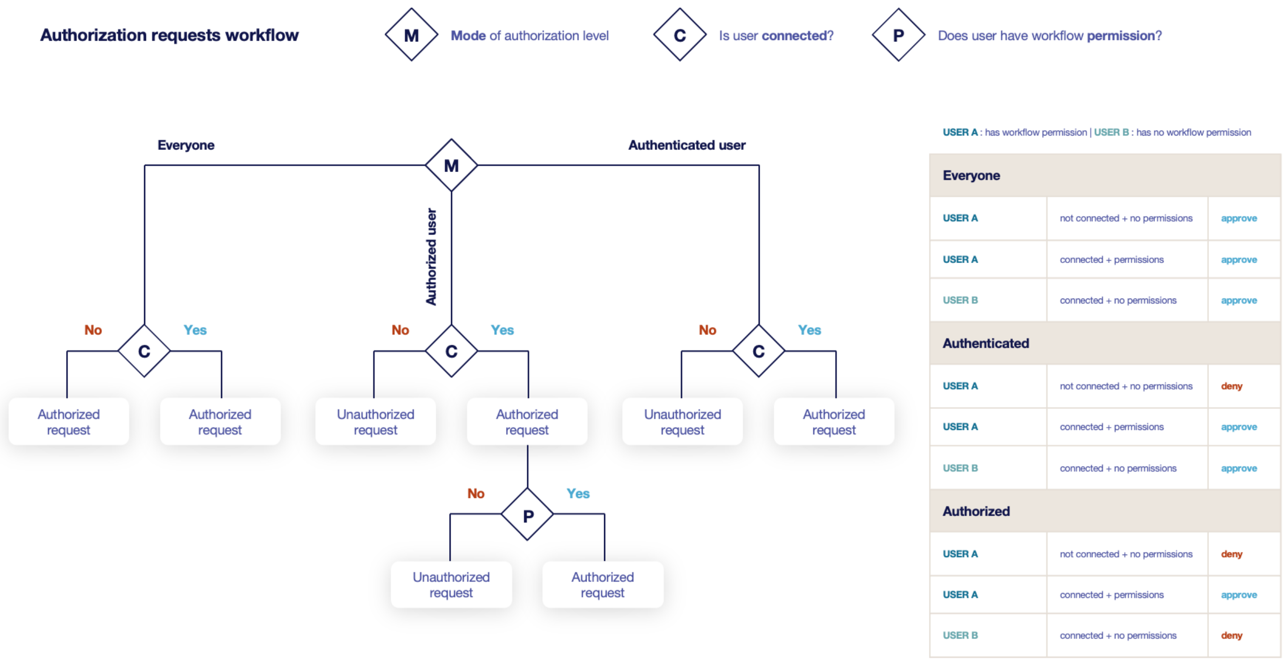
2. Add other data sources if needed. Each datasource input can use outputs from previously executed data sources.

3. All data sources output are available in computation rules throughout the certificate template and metadata.

Workflows builder

Configure custom rights for actions on this profile.

1. Select an authorization level for each workflow.



USER A : has workflow permission | USER B : has no workflow permission

Everyone		
USER A	not connected + no permissions	approve
USER A	connected + permissions	approve
USER B	connected + no permissions	approve
Authenticated		
USER A	not connected + no permissions	deny
USER A	connected + permissions	approve
USER B	connected + no permissions	approve
Authorized		
USER A	not connected + no permissions	deny
USER A	connected + permissions	approve
USER B	connected + no permissions	deny

- **Everyone**:
  No authentication is required.
- **Authenticated**:
  User has to be authenticated.
- **Authorized**:
  User has to be authenticated and have an explicit authorizations.

2. Select an access level for identity providers.

You can remove the access level for an identity provider by clicking on 'x'.

Owner-related permissions

These permissions apply to the owners of a certificate (team or owner). An owner can always request the following actions, but this permission allows them to perform the action without validation.

- **Revoke** *(boolean)*:
  Grant self revoke permission. The default value is set to false.
- **Revoke (pop)** *(boolean)*:
  Grant self revoke permission with owner being determined by Proof of Possession. This is used by horizon-cli. The default value is set to false.
- **Recover** *(boolean)*:
  Grant self recover permission. The default value is set to false.
- **Update** *(boolean)*:
  Grant self update permission. The default value is set to false.
- **Update (pop)** *(boolean)*:
  Grant self update permission with owner being determined by Proof of Possession. This is used by horizon-cli. The default value is set to false.
- **Renew** *(boolean)*:
  Grant self renew permission. The default value is set to false.
- **Renew (pop)** *(boolean)*:
  Grant self renew permission with owner being determined by Proof of Possession. This is used by horizon-cli. The default value is set to false.

Certificate Template

This section details how to define a custom structure for the fields `subject DN`, `SAN` *&* `extensions` *of the requested certificate in order to match the configuration on the PKI side.*

CAUTION In a WebRA profile, defining a template is mandatory.

Subject DN composition

You can add more elements by clicking ⊕ .

- **Element*** *(select)*:
  Select an attribute from the elements list.
- **Mandatory** *(boolean)*:
  Should the element be mandatory. The default value is set to false.
- **Editable by requester** *(boolean)*:
  Tells whether the element should be editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the element should be editable by the approver. The default value is set to false.
- **Default value** *(string input)*:
  Set a default value to the element.

- **Regex** *(regex)*:
  Enter a regular expression that the element should match.
- **Computation rule** *(Computation rule input)*:
  Set the value of this element to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

You can remove an element by clicking the delete button 🗑 or reorder (drag and drop) ⠿ the Subject DN template.

> **CAUTION** When a template is defined, at least one mandatory Common Name must be added to the DN Elements.

SAN composition

You can add more elements by clicking ⊕ .

- **Element*** *(select)*:
  Select an attribute from the element list.
- **Editable by requester** *(boolean)*:
  Tells whether the element should be editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the element should be editable by the approver. The default value is set to false.
- **Minimum** *(int)*:
  The minimum number of value that this SAN must have.
- **Maximum** *(int)*:
  The maximum number of value that this SAN must have.
- **Regex** *(regex)*:
  Enter a regular expression that the element should match.
- **Computation rule** *(Computation rule input)*:
  Set the value of this element to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

You can remove an element by clicking the delete button 🗑 or reorder (drag and drop) ⠿ the SAN template.

Extensions

You can add more elements by clicking ⊕ .

- **Element*** *(select)*:
  Select an attribute from the elements list.
- **Mandatory** *(boolean)*:
  Should the element be mandatory. The default value is set to false.

- **Editable by requester** *(boolean)*:
  Tells whether the element should be editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the element should be editable by the approver. The default value is set to false.
- **Default value** *(string input)*:
  Set a default value to the element.
- **Computation rule** *(Computation rule input)*:
  Set the value of this element to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

You can remove an element by clicking the delete button 🗑 or reorder (drag and drop) ⠿ the Extensions template.

> **CAUTION** When adding a SAN, a DN element or an Extension and making it mandatory, make sure to either give it a default value or a computation rule or make it editable, otherwise the template will be unusable.

Certificate Metadata

This section details how to define a custom structure for the labels, ownership policy and technical metadata, allowing certificates to hold rich information.

Labels

You can add more labels by clicking ⊕ .

- **Name** *(select)*:
  Select a preexisting label.
- **Mandatory** *(boolean)*:
  Should the label be mandatory. The default value is set to false.
- **Editable by requester** *(boolean)*:
  Tells whether the label should be editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the label should be editable by the approver. The default value is set to false.
- **Default value** *(string input)*:
  Set a default value to the label.
- **Label value restriction**
  - **Whitelist** *(string input multiple)*:
    The label value will have to be in the whitelist. Open the popup, enter the label value and press "enter" to add this value to the accepted value list. An empty whitelist means no restriction.
  - **Suggestions** *(string input multiple)*:
    Add suggestions that will be displayed to the user. The user will be able to choose one of

    these values or enter its own. Open the popup, enter your suggestions and press enter to add this value to the suggestions. An empty suggestions list means no restriction.

  - **Regex** *(regex)*:
    The label value will have to match the regex. Open the popup, enter the regular expression and click on the submit button to set the regex. An empty regex means no restrictions.

- **Computation rule** *(Computation rule input)*:
  Set the value of this label to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

You can delete 🗑 or reorder (drag and drop) ⠿ the label template.

Ownership policy

- **Owner**
  - **Mandatory** *(boolean)*:
    Specify if the certificate’s owner is mandatory when submitting a request.
  - **Editable by requester** *(boolean)*:
    Specify if the certificate’s owner can be overridden by the requester when submitting a request.
  - **Editable by approver** *(boolean)*:
    Specify if the certificate’s owner can be overridden by the requester when approving a request.
  - **Computation rule** *(Computation rule input)*:
    Set the value of the owner to the value of the evaluated computation rule. This value will override any other value including the user input.
- **Contact email**
  - **Mandatory** *(boolean)*:
    Specify if the certificate’s contact email is mandatory when submitting a request.
  - **Editable by requester** *(boolean)*:
    Specify if the certificate’s contact email can be overridden by the requester when submitting a request.
  - **Editable by approver** *(boolean)*:
    Specify if the certificate’s contact email can be overridden by the requester when approving a request.
  - **Default contact email** *(string input)*:
    Set a default contact email. This value must comply with the contact email restriction.
  - **Contact email restriction**
    - **Whitelist** *(string input multiple)*:
      The contact email will have to be in the whitelist. Open the popup, enter the email and press "enter" to add this value to the accepted whitelist. An empty whitelist means no restriction.
    - **Regex** *(regex)*:

        The contact email will have to match the regex. Open the popup, enter the regular expression and click on the submit button to set the regex. An empty regex means no restrictions.

    - **Computation rule** *(Computation rule input)*:
      Set the value of the contact email to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

- **Team**
    - **Mandatory** *(boolean)*:
      Specify if the certificate’s team is mandatory when submitting a request.
    - **Editable by requester** *(boolean)*:
      Specify if the certificate’s team can be overridden by the requester when submitting a request.
    - **Editable by approver** *(boolean)*:
      Specify if the certificate’s team can be overridden by the requester when approving a request.
    - **Default team** *(string input)*:
      Set a default team. This value must comply with the team restriction.
    - **Team restriction**
        - **Whitelist** *(string input multiple)*:
          The team will have to be in the whitelist. Enter the team and press "enter" to add this value to the accepted whitelist. An empty whitelist means no restriction.
        - **Regex** *(regex)*:
          The team will have to match the regex. Open the popup, enter the regular expression and click on the submit button to set the regex. An empty regex means no restrictions.
    - **Computation rule** *(Computation rule input)*:
      Set the value of the team to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

Metadata policy *(overridable metadata)*

WARNING These metadata are technical metadata. They are used by Horizon or Third party connectors, updating them should be done with utmost care.

NOTE Metadata edition is not allowed on enroll.

NOTE Metadata edition is not available via the User Interface. It must be changed with API, using horizon-cli.

You can allow the override of technical metadata by clicking ⊕ .

- **Metadata*** *(select)*:
  Select a metadata.
- **Editable by requester** *(boolean)*:

Tells whether the metadata is editable by the requester. The default value is set to false.

- **Editable by approver** *(boolean)*:
  Tells whether the metadata is editable by the approver. The default value is set to false.

You can delete 🗑 a metadata policy. This will not delete the metadata but will make it non editable.

Notifications/Triggers

This section details how to configure notifications and triggers to perform actions on certificate and request lifecycle events.

Certificate lifecycle notifications

Notifications are sent when one of the following event is triggered by a certificate:

Enrollment	Revocation	Expire	Update	Migrate	Renew

Select a preexisting email, REST or groupware notification to associate it with an event.

Request lifecycle notifications

Notifications are sent when one of the following event is triggered by an Enroll/Revocation/Update/Migrate/Renew request:

Submit	Cancel	Revoke	Approve	Pending

Select a preexisting email, REST or groupware notification to associate it with an event.

NOTE Submit request events are not triggered when the user has the permission to perform the action directly.

Triggers

Horizon support the use of third-party triggers in the form of callbacks on specific events happening on the profile, giving a way to synchronize the third party repositories and Horizon.

- **Enrollment** *(select)*:
  Select the preexisting third party or MDM trigger(s) to call whenever a certificate is enrolled on this profile.
- **Renewal** *(select)*:
  Select the preexisting third party or MDM trigger(s) to call whenever a certificate is renewed on this profile.
- **Revocation** *(select)*:
  Select the preexisting third party or MDM trigger(s) to call whenever a certificate gets revoked on this profile.
- **Expire** *(select)*:

Select the preexisting third party or MDM trigger(s) to call whenever a certificate expires on this profile.

The available triggers are the following:

AKV Triggers	AWS Triggers	F5 Triggers	LDAP Triggers	*On **WebRA** and **Intune PKCS** only:* Intune PKCS Triggers

5. Click on the save button.

You can edit ✎ , duplicate ⧉ or delete 🗑 the WebRA Profile.

CAUTION You won’t be able to delete a WebRA Profile if it is referenced somewhere else.

9.7.3. WebRA Scheduled Tasks

This section details how to schedule tasks that will run periodically with your WebRA profiles.

Prerequisites

AWS Connector	Azure AKV Connector	F5 Connector	GCM Connector	WebRA Profile

How to configure WebRA Scheduled Tasks

1. Log in to Horizon Administration Interface.

2. Access the "Scheduled tasks" from the drawer or card: **Protocols › WebRA › Scheduled Tasks**.

3. Click on ⊕ .

4. Fill in the mandatory fields.

- **WebRA Profile*** *(select)*:
  Select a previously created WebRA profile.
- **Target Connector*** *(select)*:
  Select a previously created third party connector.
- **Cron scheduling** *(cron expression)*:
  Enter a Cron scheduling expression (in Quartz format). The default expression is built to run the task every 5 hours.
- **Revoke** *(boolean)*:
  If enabled, will revoke all certificate whose container was deleted from the third party repository. The default value is set to false.
- **Renew** *(boolean)*:

If enabled, will renew all certificate who are about to expire. The default value is set to false.

- **Dry run** *(boolean)*:
  If enabled, revocation and renewal actions will not be performed. Instead, a message will be logged, explaining what would have been done.

5. Click on the save button.

You can run ▷ or edit ✎ or delete 🗑 the Schedules Tasks.

9.8. Auto Validation

Auto-validation can be enabled on the following protocols:

- WebRA
- EST
- SCEP

When the auto-validation mode is enabled, Validation Rules are evaluated to allow or deny an enrollment request.

Multiple rules can be defined on each profile, and the minimum number of passing rules can be defined.

9.8.1. Validation Rules

A validation rule is a condition that can be true or false. Inputs are taken from dictionary entries and can be manipulated using Computation Rules and validation functions and operators.

For example, to allow all requests coming from a subnet and having all DNS SANs that resolves on the Horizon server, the following rule can be used:

`{{http.request.ip}} in 154.12.45.0/24 and [[csr.san.dnsname]] resolvesDNS`

Here, two expressions are combined using the `and` operator:

- `{{http.request.ip}} in 154.12.45.0/24`: a computation rule `{{http.request.ip}}` fetches the value of the IP from the incoming request, and checks that this ip is `in` the `154.12.45.0/24` subnet, using the In operator.
- `[[csr.san.dnsname]] resolvesDNS` : a computation rule `[[csr.san.dnsname]]` fetches the values of the DNS SANs from the csr in the incoming request, and checks that these SANs `resolvesDNS`.

Examples

DNS Validation

To validate that all DNS requested in a WebRA enroll request resolve on the DNS Server with IP 192.10.132.2, the following rule can be used:

`[[webra.enroll.san.dnsname]] resolvesDNS(192.10.132.2:53)`

Here all the dns sans from the request are fetched and are submitted to the dns server.

Email validation

To validate that the Email SAN requested in an EST enroll request are associated to the requester’s Common Name, using a datasource that fetches the emails on an external LDAP server, the following rule can be used:

`{{csr.san.rfc822name.1}} = {{ds.1.1.mail}}`

Here the condition checks if the first Email SAN from the CSR is equal to the mail fetched from the LDAP datasource (supposing the LDAP datasource is the first in the flow).

Quick Reference

The table below lists the possible operators for a validation rule:

Operator Name	Syntax
And	`expression` and `expression`
Or	`expression` or `expression`
Equals	`expression` equals `expression`
In	`expression` in `expression`
Exists	`expression` exists
Contains	`expression` contains `expression`
Matches	`expression` matches `expression`
Within	`expression` within `expression`
Is Empty	`expression` is empty
Starts With	`expression` starts with `expression`
Ends With	`expression` ends with `expression`
Resolves DNS	`expression` resolvesDNS

And

```
left:<expression> and right:<expression>
```

This outputs the logical and operation on the result evaluated from `left` and `right`

```
"left" = "left" and "right"="right" => true
Upper("left") = "left" and "right"="right" => false
```

Or

```
left:<expression> or right:<expression>
```

This outputs the logical or operation on the result evaluated from `left` and `right`

```
"left" = "left" or "right"="right" => true
Upper("left") = "left" or "right"="right" => true
```

Equals

```
left:<expression> equals right:<expression>
left:<expression> = right:<expression>
```

This tests the equality operation on the result evaluated from `left` and `right`

```
"left" = "left"  => true
Upper("left") equals "left" => false
```

Not Equals

```
left:<expression> not equals right:<expression>
left:<expression> != right:<expression>
```

```
"left" != "left"  => false
Upper("left") not equals "left" => true
```

In

Element inclusion

```
elem:<single expression> in array:<multi expression>
```

This tests if `elem` is contained in `array`

```
"left" in [ "left" ]  => true
Upper("left") in ["left"] => false
```

Multiple Element inclusion

```
any of elems:<multi expression> in array:<multi expression>
all of elems:<multi expression> in array:<multi expression>
```

This tests if `all` or `any` element in `elems` is contained in `array`

```
any of ["left", "right"] in [ "left" ]  => true
all of ["left", "right"] in [ "left" ]  => false
```

Ip in CIDR

```
ip:<single expression> in subnet:<subnet>
```

This tests if `ip` is contained in `subnet` (cidr notation)

```
"128.12.13.14" in 128.12.15.0/24  => false
"2001:0db8:85a3:0000:0000:0000:0000:0001" in 2001:db8:85a3::8a2e:370:7334/64 => true
```

Multiple Ips in CIDR

```
any of ips:<multi expression> in subnet:<subnet>
all of ips:<multi expression> in subnet:<subnet>
```

This tests if `all` or `any` ip in `ips` is contained in `subnet` (cidr notation)

```
any of ["128.12.13.14", "128.12.15.32" ] in 128.12.15.0/24  => true
all of [ "2001:0db8:85a3:0000:0000:0000:0000:0001",
"2002:0db8:85a3:0000:0000:0000:0000:0001"] in 2001:db8:85a3::8a2e:370:7334/64 => false
```

Element not included

```
elem:<single expression> not in array:<multi expression>
any of elems:<multi expression> not in array:<multi expression>
all of elems:<multi expression> not in array:<multi expression>
```

```
"left" not in ["right"]  => true
any of ["left", "right"] not in [ "left" ] => true
all of ["left", "right"] not in [ "left" ] => false
"128.12.13.14" not in 128.12.15.0/24  => true
```

Exists

```
elem:<expression> exists
```

This tests if `elem` exists. For single values, this will be true if the dictionary key is defined and for multi values if the array is not empty.

```
"" exists  => true
{{will.not.exist}} exists => false
[[will.not.exist]] exists => false
```

Not exists

```
elem:<expression> not exists
```

```
"" not exists  => false
{{will.not.exist}} not exists => true
[[will.not.exist]] not exists => true
```

Is Empty

```
elem:<expression> is empty
```

This tests if `elem` is empty. For single values, this will be true if the dictionary key is not defined or if the value is empty, and for multi values if the array is empty or if all values are empty.

```
"" is empty  => true
{{will.not.exist}} is empty => true
[[will.not.exist]] is empty => true
[ "", ""] is empty => true
```

Is Not empty

```
elem:<expression> is not empty
```

```
"" is not empty  => false
{{will.not.exist}} is not empty => false
[[will.not.exist]] is not empty => false
```

Contains

Single element contained

```
containing:<expression> contains elem:<single expression>
```

This tests if `containing` contains the `elem` value. If `containing` is a string, the presence of the substring elem is checked, if it is an array the presence of the element in the array is checked.

```
"google.com" contains "google"  => true
["google.com", "google.fr"] contains "google.fr"  => true
"google.com" contains {{does.not.exist}}  => true
```

Multiple elements contained

```
containing:<multi expression> contains all of elems:<multi expression>
containing:<multi expression> contains any of elems:<multi expression>
```

This tests if `containing` contains `all` or `any` of the elements of the `elems` array. An empty `elems` array will always be contained.

```
["google.com", "google.fr"] contains all of ["test.com"]  => false
["google.com"] contains all of [[does.not.exist]]  => true
["google.com", "google.fr"] contains any of ["google.com", "a"]  => true
```

Single element not contained

```
containing:<expression> not contains elem:<single expression>
```

```
"google.com" not contains ""  => false
```

Multiple elements not contained

```
containing:<multi expression> not contains all of elems:<multi expression>
containing:<multi expression> not contains any of elems:<multi expression>
```

```
["test", "abc" ] not contains all of ["abc", "d"] => true
["test", "abcf", "d"] not contains any of ["f", "e"] => true
```

Matches

Single element match

```
elem:<single expression> matches regex:<single expression>
elem:<single expression> ~ regex:<single expression>
```

This tests if `elem` matches the `regex`. If `regex` is None, this will output false.

```
"left" matches "\d+"  => false
Upper("left") ~ "[A-Z]+" => true
```

Multiple element match

```
any of elems:<multi expression> matches regex:<single expression>
all of elems:<multi expression> matches regex:<single expression>
```

This tests if `any` or `all` element in `elems` matches the `regex`. If `regex` is None, this will output false.

```
any of ["left", "42"] matches "\d+"  => true
all of Upper(["left", "42"]) matches "[A-Z]+" => false
```

Not matching

```
elem:<single expression> not matches regex:<single expression>
any of elems:<multi expression> not matches regex:<single expression>
all of elems:<multi expression> not matches regex:<single expression>
```

```
"left" not matches "\d+" => true
any of ["left", "aaaaa"] not matches "a+"  => true
all of ["left", "aaaaa"] not matches "a+"  => false
```

Within

Element matching

```
elem:<single expression> within array:<multi expression>
```

This tests if `elem` matches a regex in `array`.

```
"left" within [ "\d+", "[a-z]+" ]  => true
Upper("left") within [ "\d+", "[a-z]+" ] => false
```

Multiple Element matching

```
any of elems:<multi expression> within array:<multi expression>
all of elems:<multi expression> within array:<multi expression>
```

This tests if `all` or `any` element in `elems` matches a regex in `array`

```
any of ["left", "aaaaa"] within [ "\d+", "a+" ]  => true
all of ["left", "aaaaa"] within [ "\d+", "a+" ]  => false
```

Element not matching

```
elem:<single expression> not within array:<multi expression>
any of elems:<multi expression> not within array:<multi expression>
all of elems:<multi expression> not within array:<multi expression>
```

```
"left" not within [ "\d+", "[a-z]+" ]  => false
any of ["left", "aaaaa"] not within [ "\d+", "a+" ]  => true
all of ["left", "aaaaa"] not within [ "\d+", "a+" ]  => false
```

Starts With

Element matching

```
elem:<single expression> starts with start:<single expression>
```

This tests if `elem` starts with `start` value. An empty `elem` will send return false.

```
"left" starts with "le"  => true
Upper("left") starts with "le" => false
```

Multiple Element matching

```
any of elems:<multi expression> starts with start:<single expression>
all of elems:<multi expression> starts with start:<single expression>
```

This tests if `all` or `any` element in `elems` starts with `start`

```
any of ["left", "aaaaa"] starts with "aaa" => true
all of ["left", "aaaaa"] starts with "aaa" => false
```

Element not matching

```
elem:<single expression> starts not with start:<single expression>
any of elems:<multi expression> starts not with start:<single expression>
all of elems:<multi expression> starts not with start:<single expression>
```

```
"left" starts not with "le"  => false
any of ["left", "aaaaa"] starts not with "a"  => true
all of ["left", "aaaaa"] starts not with "a"  => false
```

Ends With

Element matching

```
elem:<single expression> ends with start:<single expression>
```

This tests if `elem` ends with `start` value. An empty `elem` will send return false.

```
"left" ends with "ft"  => true
Upper("left") ends with "ft" => false
```

Multiple Element matching

```
any of elems:<multi expression> ends with start:<single expression>
all of elems:<multi expression> ends with start:<single expression>
```

This tests if `all` or `any` element in `elems` ends with `start`

```
any of ["left", "aaaaa"] ends with "aaa" => true
all of ["left", "aaaaa"] ends with "aaa" => false
```

Element not matching

```
elem:<single expression> ends not with start:<single expression>
any of elems:<multi expression> ends not with start:<single expression>
all of elems:<multi expression> ends not with start:<single expression>
```

```
"left" ends not with "ft"  => false
any of ["left", "aaaaa"] ends not with "a"  => true
all of ["left", "aaaaa"] ends not with "a"  => false
```

Resolves DNS

```
host:<expression> resolvesDNS
host:<expression> resolvesDNS(101.12.13.14:53)
```

This tests if `host` resolves on the DNS server. An optional DNS server in the `ip:port` format can be used. If `host` is an array, DNS must resolve for each value. An empty array returns `false`.

```
"google.com" resolvesDNS  => true
["google.com", "google.fr"] resolvesDNS  => true
["google.com", "not.resolving"] resolvesDNS  => false
```

Not Resolves DNS

```
host:<expression> not resolvesDNS
host:<expression> not resolvesDNS(101.12.13.14:53)
```

```
"google.com" not resolvesDNS  => false
["google.com", "google.fr"] not resolvesDNS  => false
```

10. Datasources

10.1. Datasource Introduction

Datasources are external assets that contain data useful for certificate enrollment or enrollment validation.

Datasources are fetched on enrollment request **submission** and are used to fill the request dictionary. This dictionary is then available to use in computation rules and validation rules.

To define which datasources to fetch, and with which parameters, a **datasource flow** is available on all certificate profiles. A Datasource Flow is a collection of datasources to fetch, with inputs from the request.

> **NOTE** Keys fetched from datasources are prefixed with `ds.<i>`, `i` being the index of the datasource in the flow (starting from 1).

10.2. DNS Datasource

This section details how to configure a DNS datasource.

10.2.1. How to configure DNS datasource

1. Log in to Horizon Administration Interface.

2. Access Datasources from the drawer or card: **Datasources**.

3. Click on ⊕ .

4. Select **DNS Type**

5. Fill in the mandatory fields.

Datasource specific configuration

General

- **Name*** *(string input)*:
  Enter a meaningful datasource name, this setting will be the datasource identifier. It must be unique for each datasource.
- **Description** *(string input)*:
  Enter a description to describe this datasource usage.

DNS Parameters

- **Hostname and port***: *(select & string input)*
  Choose the hostname and port of your DNS server.

- **Record types** *(select string)*:
  Choose the record types to fetch. If none are selected, all record types are fetched.
- **Lookup*** *(string input)*:
  Lookup to query. It is a template string and can contain keys for parametrization.
- **Timeout** *(finite duration)*:
  Set by default at 10 seconds. Must be a valid finite duration.

10.3. LDAP Datasource

This section details how to configure a LDAP datasource.

10.3.1. How to configure LDAP datasource

1. Log in to Horizon Administration Interface.

2. Access Datasources from the drawer or card: **Datasources**.

3. Click on ⊕ .

4. Select **LDAP Type**

5. Fill in the mandatory fields.

Datasource specific configuration

General

- **Name*** *(string input)*:
  Enter a meaningful datasource name, this setting will be the datasource identifier. It must be unique for each datasource.
- **Description** *(string input)*:
  Enter a description to describe this datasource usage.

LDAP Parameters

- **Hostname*** *(string input)*:
  Enter the URL pointing to LDAP.
- **Port** *(int)*:
  Enter the port where to reach the running LDAP instance (default values are 389 for LDAP and 636 for LDAPS).
- **LDAP Credentials*** *(select)*:
  Select `Login` credentials containing the technical user created for Horizon login DN and password.
- **Base DN*** *(string input)*:
  Enter the Base DN where Horizon should publish the certificate. It is a template string and can

contain keys for parametrization.

- **Filter** *(string input)*:
Enter the custom filter. It is a template string and can contain keys for parametrization.

- **Proxy** *(string select)*:
The HTTP/HTTPS proxy used to reach LDAP, if any.

- **Timeout** *(finite duration)*:
Set by default at 10 seconds. Must be a valid finite duration.

- **Follow referrals** *(boolean)*:
Allow publication to follow LDAP referrals.

- **Secure** *(boolean)*:
Only use secure LDAP connection

- **Disable hostname validation** *(boolean)*:
Allow non validated hostname during LDAP connection

- **Limit*** *(integer)*:
Maximum number of results for the LDAP query

10.4. REST Datasource

This section details how to configure a REST datasource.

10.4.1. How to configure REST datasource

1. Log in to Horizon Administration Interface.

2. Access Datasources from the drawer or card: **Datasources**.

3. Click on ⊕ .

4. Select **REST Type**

5. Fill in the mandatory fields.

Datasource specific configuration

General

- **Name*** *(string input)*:
Enter a meaningful datasource name, this setting will be the datasource identifier. It must be unique for each datasource.

- **Description** *(string input)*:
Enter a description to describe this datasource usage.

REST Parameters

- **HTTP Method and URL***: *(select & string input)*
  Choose the HTTP method and the destination URL for your notification. The URL is a template string and can contain keys for parametrization.
- **Proxy**: *(select)*
  Define a proxy for this REST API call.
- **Timeout*** *(finite duration)*:
  Connection timeout when executing the REST API call. Must be a valid finite duration.
- **Accepted response HTTP code*** *(multiselect | input)*:
  Response codes meaning the REST call was a success. If another one is received, a failure will be logged.
- **Authentication type and credentials*** *(select & select)*:
  Choose the authentication type and the credentials to perform the authentication. Custom authentication allows the credentials values to be accessible in headers.
- **Headers** *(input string & input string)*:
  Choose the header name and value. Header values are template strings and can contain keys for parametrization.
- **Body*** *(string input)*:
  Enter the REST body. It is a template string and can contain keys for parametrization.

11. Third parties

11.1. AWS

11.1.1. AWS Introduction

This section refers to the AWS Certificate Manager (ACM) integration with Horizon, used to enroll certificates held in ACM.

This integration involves at least two infrastructure components:

- AWS Certificate Manager
- EverTrust Horizon

11.1.2. AWS Connector

Here is the section to manage the AWS Connector.

Required By

AWS Trigger

Prerequisites

On Horizon side, you might need to set up a Proxy , used to reach AWS, if necessary.

On AWS side, you need to create a user using the AWS IAM module, and following AWS guide. You should create an access key for that user, and give him appropriate permissions. The created user should hold the following permissions:

- `AWSResourceGroupsReadOnlyAccess`
- `ResourceGroupsandTagEditorReadOnlyAccess`
- `AWSCertificateManagerFullAccess`

After performing these steps, you will get the following information, required later:

- the AWS Region
- the User Access Key ID
- the User Access Key Secret

On top of that, you need to define a Resource Group, using AWS Resource Groups and Tags Editor, with the following characteristics:

- Group Type: Tag based
- Resource Type: `AWS::CertificateManager::Certificate`

- Tag key and value (e.g. key=manage and value=HRZ)

After performing this steps, you will get the following information, required later:

- The Resource Group name
- the Tag name
- the Tag value

How to configure AWS Connector

1. Log in to Horizon Administration Interface.

2. Access AWS Connectors from the drawer or card: **Third Parties › AWS › Connectors**.

3. Click on ⊕ .

4. Fill the mandatory fields.

Connection

- **Name*** *(string input)*:
  Enter a meaningful connector name. It must be unique for each connector. Horizon uses the name to identify the connector.
- **Region*** *(string input)*:
  Enter a valid AWS region. Here's the region list from AWS.
- **AWS Access Key Credentials** *(select)*:
  Select `Login` credentials containing the User Access Key ID and secret used by Horizon to connect to AWS.
- **Proxy** *(string select)*:
  The HTTP/HTTPS proxy to use to reach AWS, if any.
- **Timeout*** *(finite duration)*:
  The timeout for Horizon-initiated connections to AWS. Must be a valid finite duration.

Assets identification

- **Resource group name** *(string input)*:
  Name of the resource group pointing to the tag name and value.
- **Role ARN** *(string input)*:
  Name of the AWS role Horizon will impersonate in ACM.
- **Tag key** *(string input)*:
  Name of the tag used to identify certificates managed by Horizon in ACM.
- **Tag value** *(string input)*:
  Value of the tag used to identify certificates managed by Horizon in ACM.

Actors and renewal management

- **Throttle duration*** *(finite duration)*:
  Set by default at 3 seconds. Must be a valid finite duration.
- **Renewal period** *(finite duration)*:
  Certificate renewal period (time before expiration to trigger renewal). Must be a valid finite duration.

5. Click on the save button.

You can update ✎ or delete 🗑 the AWS Connector.

CAUTION You won’t be able to delete an AWS Connector if it is referenced somewhere else.

Synchronize your third party

Your third-party certificates can be synchronized with Horizon using scheduled tasks.

Scheduled tasks are a functionality of WebRA that allows to synchronize automatic renewal or revocation events with a third party periodically with what occurs on a WebRA profile. To be more specific, it will periodically check whether the certificate has entered the "renewal period" that was defined in the connector’s configuration, and renew it automatically if necessary.

1. Refer to the third party connector documentation to create a third party connector.

2. Ensure you have an existing WebRA Profile: renewal will be automated on the selected profile.

3. Follow the documentation of the WebRA Scheduled Tasks section to properly configure a scheduled task.

AWS Trigger

Here is the section to manage the Triggers that will be used by Profiles to push or delete certificates to/from AWS ACM.

Prerequisites

AWS Connector

How to configure AWS Trigger

1. Log in to Horizon Administration Interface.

2. Access AWS Triggers from the drawer or card: **Third Parties › AWS › Triggers**.

3. Click on ⊕.

4. Fill the mandatory fields.

- **Name*** *(string input)*:
  Enter a meaningful trigger name. It must be unique for each trigger. Horizon uses the name to identify the trigger.
- **AWS Connector*** *(select)*:
  Select an AWS connector previously created.
- **Retries in case of error** *(int)*:
  Number of times to retry to push the change on the AWS repository in case of error. Must be an integer between 1 and 15.

5. Click on the save button.

You can update ✎ or delete 🗑 the AWS Trigger.

> **CAUTION** You won’t be able to delete an AWS Trigger if it is referenced somewhere else.

Synchronization using triggers

Triggers are a functionality of WebRA, Intune PKCS, WCCE and CRMP profiles that allows to push lifecycle events into a third party whenever they occur on a profile.

1. Refer to the trigger documentation to create a trigger.

2. Create or modify the profile you wish to use the triggers on.

3. Go to the **Triggers** tab, then on **Certificate lifecycle triggers**

4. Chose which lifecycle events you wish to use triggers upon (enrollment, revocation, expiration)

5. Select one or more existing triggers from the menu (if several are selected, they will all be called whenever the selected event occurs)

6. Click on the Save button.

From now on, whenever a selected lifecycle event will occur on the configured profile, the trigger will be called and the and the certificate will be pushed into or removed from the third party container.

11.2. AKV

11.2.1. AKV Introduction

This section refers to the Azure Key Vault (AKV) integration with Horizon, used to enroll certificates held in AKV.

This integration involves at least three infrastructure components:

- Azure Key Vault
- Azure Active Directory
- EverTrust Horizon

Azure AD is used to authenticate Horizon, which should be a registered application.

11.2.2. Azure AKV Connector

Here is the section to manage the Azure AKV Connector.

Required By

Azure AKV Trigger

Prerequisites

On Horizon side, you might need to set up a Proxy used to reach Azure, if necessary.

On Azure AD side, it is required to set up an application by following Microsoft’s guide.

NOTE Horizon supports only client secret authentication

After performing these steps, you will get the following information, required later:

- the Tenant ID
- the Application ID
- the Application Authentication Key

Finally, you should give all Certificate Permissions to the Application you created for Horizon inside the target Azure Key Vault "Access policies" menu entry, using the "Add Access Policy" link.

How to configure AKV Connector

1. Log in to Horizon Administration Interface.

2. Access AKV Connectors from the drawer or card: **Third Parties › AKV › Connectors**.

3. Click on ⊕ .

4. Fill the mandatory fields.

Connection

- **Name*** *(string input)*:
  Enter a meaningful Connector Name.
- **Azure Tenant*** *(string input)*:

Enter the Tenant, which is the domain name after the @ sign in your account.

- **App Registration Credentials*** *(select)*:
  Select `Login` credentials containing your app registration ID and secret key.
- **Proxy** *(string select)*:
  The HTTP/HTTPS proxy used to reach Azure AD and AKV, if necessary.
- **Timeout** *(finite duration)*:
  Set on the connections used to reach Azure AD and AKV. Configured by default at 10 seconds. Must be a valid finite duration.
- **Vault fully qualified domain name*** *(string input)*:
  Fully qualified domain name used to reach the Azure Key Vault to be managed by Horizon.

Assets identification and management

- **Prefix** *(string input)*: Used to filter the certificates managed by Horizon in the specified Azure Key Vault. Defaults to "HRZ-"

Actors and renewal management

These configuration elements mainly define the number of authorized interactions with the remote service on a defined period. For example, one needs to ensure that the remote service will not be contacted more than 5 times per 3 seconds. *Throttle parallelism* defines the number of times and *Throttle duration* the period of time. Therefore, on the above example, throttle parallelism would be set to 5 and throttle duration would be set to 3 seconds.

- **Throttle duration*** *(finite duration)*:
  Set by default at 3 seconds. Must be a valid finite duration.
- **Throttle parallelism*** *(int)*:
  Set by default at 3.
- **Renewal period** *(finite duration)*:
  Must be a valid finite duration.

5. Click on the save button.

You can update ✎ or delete 🗑 the AKV Connector.

CAUTION You will not be able to delete an AKV Connector if it is referenced in any other configuration element.

Synchronize your third party

Your third-party certificates can be synchronized with Horizon using scheduled tasks.

Scheduled tasks are a functionality of WebRA that allows to synchronize automatic renewal or revocation events with a third party periodically with what occurs on a WebRA profile. To be more specific, it will periodically check whether the certificate has entered the "renewal period" that was defined in the connector’s configuration, and renew it automatically if necessary.

1. Refer to the third party connector documentation to create a third party connector.

2. Ensure you have an existing WebRA Profile: renewal will be automated on the selected profile.

3. Follow the documentation of the WebRA Scheduled Tasks section to properly configure a scheduled task.

AKV Trigger

This section details how to configure the Triggers that will be used by Profiles to push or delete certificates to/from AKV.

Prerequisites

Azure AKV Connector

How to configure AKV Trigger

1. Log in to Horizon Administration Interface.

2. Access AKV Triggers from the drawer or card: **Third Parties › AKV › Triggers**.

3. Click on ⊕ .

4. Fill the mandatory fields.

- **Name*** *(string input)*:
  Enter a meaningful trigger name. It must be unique for each trigger. Horizon uses the name to identify the trigger.
- **Azure Key Vault Connector*** *(select)*:
  Select an AKV connector previously created.
- **Retries in case of error** *(int)*:
  Number of times to retry to push the change on the AKV repository in case of error. Must be an integer between 1 and 15.

5. Click on the save button.

You can update ✎ or delete 🗑 the AKV Trigger.

Synchronization using triggers

Triggers are a functionality of WebRA, Intune PKCS, WCCE and CRMP profiles that allows to push lifecycle events into a third party whenever they occur on a profile.

1. Refer to the trigger documentation to create a trigger.

2. Create or modify the profile you wish to use the triggers on.

3. Go to the **Triggers** tab, then on **Certificate lifecycle triggers**

4. Chose which lifecycle events you wish to use triggers upon (enrollment, revocation, expiration)

5. Select one or more existing triggers from the menu (if several are selected, they will all be called whenever the selected event occurs)

6. Click on the Save button.

From now on, whenever a selected lifecycle event will occur on the configured profile, the trigger will be called and the and the certificate will be pushed into or removed from the third party container.

11.3. F5

11.3.1. F5 Introduction

This section refers to the F5 BigIP integration with Horizon, used to enroll certificates used by F5 BigIP.

This integration involves at least two infrastructure components:

- F5 BigIP
- EverTrust Horizon

Horizon connects to the F5 BigIP using the iControl REST administration API in order to manage the lifecycle of certificates associated to Client SSL Profiles within the BigIP.

11.3.2. F5 Connector

This section details how to configure the F5 Connector.

Required By

F5 Trigger

Prerequisites

On the F5 BigIP side, you need to create a technical user for Horizon, and give it full administrator rights. This is required because only full admins have the right to upload certificates on an F5 BigIP.

After performing these steps, you will get the following information, required later:

- the technical user login/username
- the technical user password

How to configure F5 Connector

1. Log in to Horizon Administration Interface.

2. Access F5 Connectors from the drawer or card: **Third Parties › F5 › Connectors**.

3. Click on ⊕ .

4. Fill the mandatory fields.

General

- **Name*** *(string input)*:
  Enter a meaningful connector name. It must be unique for each connector. Horizon uses the name to identify the connector.
- **F5 BigIP hostname*** *(string input)*:
  Enter the F5 BigIP hostname (DNS or IP address).
- **F5 BigIP credentials*** *(select)*:
  Select `Login` credentials containing the username and password created for Horizon in the F5 BigIP. Must have administrator rights.
- **F5 Login Provider** *(string input)*:
  Login provider to use in TACACS authentication mode. `tmos` for example
- **Proxy** *(string select)*:
  The HTTP/HTTPS proxy to use.
- **Timeout** *(finite duration)*:
  Set by default at 10 seconds. Must be a valid finite duration.
- **Max stored certificates per holder** *(int)*:
  When specified, define the maximum number of certificates stored in the third party for a given holder.

Assets identification

- **Partition** *(string input)*:
  F5 BigIP partition to manage. *Common* by default.
- **SSL parent** *(string input)*:
  Name of the parent Client SSL Profile. *Common* by default.
- **Prefix** *(string input)*:
  Used to filter the certificates managed by Horizon in the specified F5 Client. *hrz-* by default.
- **Cipher group** *(string input)*:
  Name of the Cipher group. *None* by default.
- **Version** *(string input)*:
  Major version of the F5 BigIp instance. For a F5 instance in `15.1.10` use `15.0.0` for example. *None* by default.

Actors and renewal management

These configuration elements mainly define the number of authorized interactions with the remote service on a defined period. For example, one needs to ensure that the remote service will not be contacted more than 5 times per 3 seconds. *Throttle parallelism* defines the number of times and *Throttle duration* the period of time. Therefore, on the above example, throttle parallelism would be set to 5 and throttle duration would be set to 3 seconds.

- **Throttle duration*** *(finite duration)*:
  Set by default at 3 seconds. Must be a valid finite duration.
- **Throttle parallelism*** *(int)*:
  Set by default at 3.
- **Renewal period*** *(finite duration)*:
  Must be a valid finite duration.

5. Click on the save button.

You can update ✎ or delete 🗑 the F5 Connector.

CAUTION You will not be able to delete an F5 Connector if it is referenced in any other configuration element.

Synchronize your third party

Your third-party certificates can be synchronized with Horizon using scheduled tasks.

Scheduled tasks are a functionality of WebRA that allows to synchronize automatic renewal or revocation events with a third party periodically with what occurs on a WebRA profile. To be more specific, it will periodically check whether the certificate has entered the "renewal period" that was defined in the connector’s configuration, and renew it automatically if necessary.

1. Refer to the third party connector documentation to create a third party connector.

2. Ensure you have an existing WebRA Profile: renewal will be automated on the selected profile.

3. Follow the documentation of the WebRA Scheduled Tasks section to properly configure a scheduled task.

F5 Trigger

This section details how to configure the Triggers that will be used by Profiles to push or delete certificates to/from F5 BigIP.

Prerequisites

F5 Connector

How to configure F5 Trigger

1. Log in to Horizon Administration Interface.

2. Access F5 Triggers from the drawer or card: **Third Parties › F5 › Triggers**.

3. Click on ⊕ .

4. Fill the mandatory fields.

- **Name*** *(string input)*:
  Enter a meaningful trigger name. It must be unique for each trigger. Horizon uses the name to identify the trigger.
- **F5 Connector*** *(select)*:
  Select a F5 connector previously created.
- **Retries in case of error** *(int)*:
  Number of times to retry to push the change on the F5 BigIP repository in case of error. Must be an integer between 1 and 15.

5. Click on the save button.

You can update ✎ or delete 🗑 the F5 Trigger.

Synchronization using triggers

Triggers are a functionality of WebRA, Intune PKCS, WCCE and CRMP profiles that allows to push lifecycle events into a third party whenever they occur on a profile.

1. Refer to the trigger documentation to create a trigger.

2. Create or modify the profile you wish to use the triggers on.

3. Go to the **Triggers** tab, then on **Certificate lifecycle triggers**

4. Chose which lifecycle events you wish to use triggers upon (enrollment, revocation, expiration)

5. Select one or more existing triggers from the menu (if several are selected, they will all be called whenever the selected event occurs)

6. Click on the Save button.

From now on, whenever a selected lifecycle event will occur on the configured profile, the trigger will be called and the and the certificate will be pushed into or removed from the third party container.

11.4. GCM

11.4.1. GCM Introduction

This section refers to the Google Certificate Manager integration with Horizon, used to enroll certificates used by Google Certificate Manager.

This integration involves at least two infrastructure components:

- Google Certificate Manager
- EverTrust Horizon

11.4.2. GCM Connector

This section details how to configure the Google Certificate Manager Connector.

Required By

GCM Trigger

Prerequisites

On Horizon side, you might need to set up a Proxy , used to reach GCM, if necessary.

On Google Cloud side, you need to create a service account using the IAM, and grant that SA the appropriate permissions, as documented here. Typically, these can be granted through the Certificate Manager Owner role (`roles/certificatemanager.owner`), or through the individual following permissions:

- `certificatemanager.certs.create`
- `certificatemanager.certs.list`
- `certificatemanager.certs.get`
- `certificatemanager.certs.update`
- `certificatemanager.certs.delete`

After performing these steps, you will get the following information, required later:

- the GCP Project
- the GCP Location
- the API token for the GCP Service Account

How to configure GCM Connector

1. Log in to Horizon Administration Interface.

2. Access GCM Connectors from the drawer or card: **Third Parties › GCM › Connectors**.

3. Click on ⊕.

4. Fill the mandatory fields.

General

- **Name*** *(string input)*:
  Enter a meaningful connector name. It must be unique for each connector. Horizon uses the name to identify the connector.
- **GCM Service Account Credentials*** *(select)*:
  Select `API Token` credentials containing the authentication information.
- **Proxy** *(string select)*:
  The HTTP/HTTPS proxy to use.
- **Timeout** *(finite duration)*:
  Set by default at 10 seconds. Must be a valid finite duration.

Assets identification

- **Project name*** *(string input)*:
  Name of the GCM project.
- **Location*** *(string input)*:
  Location of the GCM server.
- **Label** *(string inputs)*:
  Used to filter the certificates managed by Horizon in GCM.
  - **Key** *(string input)*:
    The label key. *manage* by default.
  - **Value** *(string input)*:
    The label value. *horizon* by default.

Actors and renewal management

These configuration elements mainly define the number of authorized interactions with the remote service on a defined period. For example, one needs to ensure that the remote service will not be contacted more than 5 times per 3 seconds. *Throttle parallelism* defines the number of times and *Throttle duration* the period of time. Therefore, on the above example, throttle parallelism would be set to 5 and throttle duration would be set to 3 seconds.

- **Throttle duration*** *(finite duration)*:
  Set by default at 3 seconds. Must be a valid finite duration.
- **Throttle parallelism*** *(int)*:
  Set by default at 3.
- **Renewal period*** *(finite duration)*:
  Must be a valid finite duration.

5. Click on the save button.

You can update ✎ or delete 🗑 the GCM Connector.

CAUTION You will not be able to delete a GCM Connector if it is referenced in any other configuration element.

Synchronize your third party

Your third-party certificates can be synchronized with Horizon using scheduled tasks.

Scheduled tasks are a functionality of WebRA that allows to synchronize automatic renewal or revocation events with a third party periodically with what occurs on a WebRA profile. To be more specific, it will periodically check whether the certificate has entered the "renewal period" that was defined in the connector’s configuration, and renew it automatically if necessary.

1. Refer to the third party connector documentation to create a third party connector.

2. Ensure you have an existing WebRA Profile: renewal will be automated on the selected profile.

3. Follow the documentation of the WebRA Scheduled Tasks section to properly configure a scheduled task.

GCM Trigger

This section details how to configure the Triggers that will be used by Profiles to push or delete certificates to/from Google Certificate Manager.

Prerequisites

GCM Connector

How to configure GCM Trigger

1. Log in to Horizon Administration Interface.

2. Access GCM Triggers from the drawer or card: **Third Parties** › **GCM** › **Triggers**.

3. Click on ⊕ .

4. Fill the mandatory fields.

- **Name*** *(string input)*:
  Enter a meaningful trigger name. It must be unique for each trigger. Horizon uses the name to identify the trigger.
- **GCM Connector*** *(select)*:
  Select a GCM connector previously created.
- **Retries in case of error** *(int)*:
  Number of times to retry to push the change on the GCM repository in case of error. Must be an integer between 1 and 15.

5. Click on the save button.

You can update ✎ or delete 🗑 the GCM Trigger.

Synchronization using triggers

Triggers are a functionality of WebRA, Intune PKCS, WCCE and CRMP profiles that allows to push lifecycle events into a third party whenever they occur on a profile.

1. Refer to the trigger documentation to create a trigger.

2. Create or modify the profile you wish to use the triggers on.

3. Go to the **Triggers** tab, then on **Certificate lifecycle triggers**

4. Chose which lifecycle events you wish to use triggers upon (enrollment, revocation, expiration)

5. Select one or more existing triggers from the menu (if several are selected, they will all be called whenever the selected event occurs)

6. Click on the Save button.

From now on, whenever a selected lifecycle event will occur on the configured profile, the trigger will be called and the and the certificate will be pushed into or removed from the third party container.

> **CAUTION**
>
> Google Certificate Manager requires the certificate CN to be a valid DNS hostname. If you try to push a certificate with a CN that is not a valid DNS hostname, you may receive a validation error stating that the "domain name doesn’t comply with RFC 1034 3.5 preferred name syntax (relaxed by RFC 1123 2.1)".
>
> Therefore, we recommend validating the certificate CN using Horizon validation rules to ensure consistency between certificates in Horizon and on Google Certificate Manager.

11.5. LDAP

11.5.1. LDAP Introduction

This section details the LDAP integration with Horizon, used to publish and unpublish certificates on LDAP.

The integration will require to set up the following elements (on Horizon side):

- an LDAP Connector, which holds the configuration items required by Horizon to connect to LDAP
- an LDAP Trigger, which holds the configuration items specifying how Horizon should

publish/unpublish certificates for the specified LDAP connector

CAUTION Only SMIME Certificates can be published

11.5.2. LDAP Connector

This section details how to configure an LDAP Connector.

Required By

LDAP Trigger

Prerequisites

On the LDAP side, it is required to create a technical user with permissions to write in the LDAP sub DN, so that Horizon will be able to search by email, to publish and to unpublish certificates using that technical user. The following information will be required later:

- LDAP Hostname
- a login DN
- a password
- Base DN to publish SMIME certificates

How to configure LDAP Connector

1. Log in to Horizon Administration Interface.

2. Access LDAP Connector from the drawer or card: **Third Parties** › **LDAP** › **Connectors**.

3. Click on ⊕ .

4. Fill the mandatory fields.

Connection

- **Name*** *(string input)*:
  Enter a meaningful connector name. It must be unique for each connector. Horizon uses the name to identify the connector.
- **Hostname*** *(string input)*:
  Enter the URL pointing to LDAP.
- **LDAP Credentials*** *(select)*:
  Select `Login` credentials containing the technical user created for Horizon login DN and password.
- **Base DN*** *(string input)*:
  Enter the Base DN where Horizon should publish the certificate.

- **Max stored certificates per holder*** *(int)*:
  When specified, define a maximum number of certificates stored in the third party.
- **Port** *(int)*:
  Enter the port where to reach the running LDAP instance (default values are 389 for LDAP and 636 for LDAPS).
- **Proxy** *(string select)*:
  The HTTP/HTTPS proxy used to reach LDAP, if any.
- **Timeout** *(finite duration)*:
  Set by default at 10 seconds. Must be a valid finite duration.

Assets identification and management

- **Filter** *(string input)*:
  Enter the custom filter. By default, LDAP Identities are filtered by `(objectclass=user)`. If you are using inetOrgPerson as type, you will have to manually set the following filter: `(objectclass=inetOrgPerson)`.
- **Target LDAP publication attribute** *(string input)*:
  When specified, the certificate will be published on the specified attribute. In most LDAP applications you will have to set the field to: `userCertificate;binary` but in MSAD the field is already well managed.
- **Target LDAP user identifier attribute** *(string select)*:
  The LDAP attribute that will be used to identify a user for publication. Possible values are `CN`, `MAIL`, `UID`.
- **Certificate user identifier attribute** *(string select)*:
  The Certificate attribute value that will be used to identity a user for publication, possible values are `UID` `SERIALNUMBER` `SURNAME` `GIVENNAME` `T` `UNSTRUCTUREDADDRESS` `UNSTRUCTUREDNAME` `E` `OU` `ORGANIZATIONIDENTIFIER` `PSEUDONYM` `UNIQUEIDENTIFIER` `STREET` `ST` `L` `O` `C` `DESCRIPTION` `DC` `RFC822NAME` `DNSNAME` `URI` `IPADDRESS` `OTHERNAME_UPN` `OTHERNAME_GUID`.
- **Follow referrals** *(boolean)*:
  Allow publication to follow LDAP referrals.
- **Create LDAP entry** *(boolean)*:
  If true, an LDAP entry will be created for this certificate if no entry matching the filter and the identifier attribute are detected. This entry will have its `objectClass` set to the filter value.

Actors and renewal management

These configuration elements mainly define the number of authorized interactions with the remote service on a defined period. For example, one needs to ensure that the remote service will not be contacted more than 5 times per 3 seconds. *Throttle parallelism* defines the number of times and *Throttle duration* the period of time. Therefore, on the above example, throttle parallelism would be set to 5 and throttle duration would be set to 3 seconds.

- **Throttle duration*** *(finite duration)*:
  Set by default to 3 seconds. Must be a valid finite duration.
- **Throttle parallelism*** *(int)*:

Set by default to 3.

5. Click on the save button.

You can update ✎ or delete 🗑 the LDAP Connector.

CAUTION You won't be able to delete a LDAP Connector if it is referenced in any other configuration element.

LDAP Triggers

Here is the section to manage the Triggers that will be used by profiles to publish or unpublish certificates into LDAP.

Prerequisites

LDAP Connector

How to configure LDAP trigger

1. Log in to Horizon Administration Interface.

2. Access LDAP triggers from the drawer or card: **Third Parties** › **LDAP** › **Triggers**.

3. Click on ⊕ .

4. Fill the mandatory fields.

- **Name*** *(string input)*:
  Enter a meaningful trigger name. It must be unique for each trigger. Horizon uses the name to identify the trigger.
- **LDAP Connector Certificate Publication*** *(select)*:
  Select an LDAP connector previously created.
- **Retries in case of error** *(int)*:
  Number of times to retry to push the change on the Intune PKCS repository in case of error. Must be an integer between 1 and 15.

5. Click on the save button.

You can run ▷ or update ✎ or delete 🗑 the trigger.

Synchronization using triggers

Triggers are a functionality of WebRA, Intune PKCS, WCCE and CRMP profiles that allows to push lifecycle events into a third party whenever they occur on a profile.

1. Refer to the trigger documentation to create a trigger.

2. Create or modify the profile you wish to use the triggers on.

3. Go to the **Triggers** tab, then on **Certificate lifecycle triggers**

4. Chose which lifecycle events you wish to use triggers upon (enrollment, revocation, expiration)

5. Select one or more existing triggers from the menu (if several are selected, they will all be called whenever the selected event occurs)

6. Click on the Save button.

From now on, whenever a selected lifecycle event will occur on the configured profile, the trigger will be called and the and the certificate will be pushed into or removed from the third party container.

12. MDM

12.1. Intune

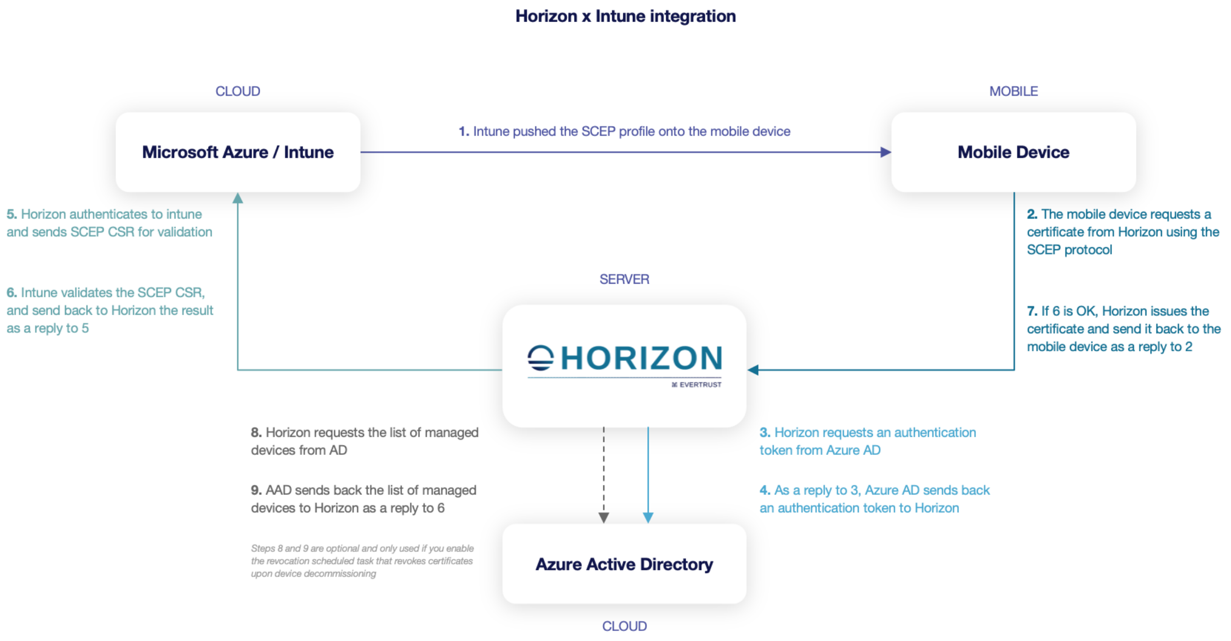
12.1.1. Intune Introduction

This section details the Microsoft Endpoint Manager - Intune SCEP integration with Horizon, used to enroll, renew and revoke certificates on Intune managed devices.

This integration involves at least three infrastructure components:

- Microsoft Endpoint Manager / Intune
- Azure Active Directory
- EverTrust Horizon

The enrolled devices interface with these components in order to retrieve their certificate.


Horizon x Intune integration
CLOUD
Microsoft Azure / Intune
MOBILE
Mobile Device
1. Intune pushed the SCEP profile onto the mobile device
2. The mobile device requests a certificate from Horizon using the SCEP protocol
SERVER
HORIZON
EVERTRUST
3. Horizon requests an authentication token from Azure AD
4. As a reply to 3, Azure AD sends back an authentication token to Horizon
5. Horizon authenticates to intune and sends SCEP CSR for validation
6. Intune validates the SCEP CSR, and send back to Horizon the result as a reply to 5
7. If 6 is OK, Horizon issues the certificate and send it back to the mobile device as a reply to 2
8. Horizon requests the list of managed devices from AD
9. AAD sends back the list of managed devices to Horizon as a reply to 6
Steps 8 and 9 are optional and only used if you enable the revocation scheduled task that revokes certificates upon device decommissioning
Azure Active Directory
CLOUD

The diagram displays these components as well as the various flows involved in an enrollment.

Microsoft describes the integration principles on their website: https://docs.microsoft.com/en-us/mem/intune/protect/certificate-authority-add-scep-overview

Finally, this integration will require to set up, on Horizon side, the following elements:

- an Intune Connector, which holds the configuration items required for Horizon to connect to Azure AD and Intune
- an Intune Profile, which holds the configuration items specifying how Horizon should issue certificates for the specified Intune Connector

- an Intune Scheduled Task, which holds configuration items defining the scheduled task in charge of performing revocation upon decommissioning devices from Azure AD. This is optional.

12.1.2. Intune Connector

This section details how to configure an Intune Connector.

Required By

Intune Profile	Intune Connector

Prerequisites

On Horizon side, you might need to set up a Proxy, used to reach Azure/Intune, if necessary. Note that the Horizon instance must also be reachable from the Azure AD endpoint, hence being reachable from the Internet.

On Azure AD side, it is required to set up an application by following Microsoft's guide. Please note that you must add the **Microsoft Graph / Application.Read.All** permission as well for the revocation feature to work properly. After performing these steps, you will get the following information, required later:

- the Tenant ID
- the Application ID
- the Application Authentication Key

How to configure Intune Connector

1. Log in to Horizon Administration Interface.

2. Access Intune Connector from the drawer or card: **Third Parties › Intune › Connectors**.

3. Click on ⊕.

4. Fill the mandatory fields.

Connection

- **Name*** *(string input)*:
  Enter a meaningful connector name. It must be unique for each connector. Horizon uses the name to identify the connector.
- **Azure Tenant*** *(string input)*:
  Enter the Tenant ID.
- **App Registration Credentials***(*select*):
  Select `Login` credentials containing your app registration ID and secret key.

- **Proxy** *(string select)*:
  The HTTP/HTTPS proxy used to reach Azure AD and Intune.
- **Timeout** *(finite duration)*:
  Timeout set on the connection used to reach Azure AD and Intune. Configured by default at 10 seconds. Must be a valid finite duration.

Assets identification and management

- **OS query string** *(string input)*:
  This allows to restrict devices by OS when performing the devices listing used for the revocation feature. Leave blank to use the default setting if unsure.
- **Intune resource URL** *(string input)*:
  This allows to point at a specific Intune installation. Used only in Hybrid Intune setups, leave blank otherwise.
- **Legacy revocation mode** *(boolean)*:
  Activate the legacy revocation mode. Default value is set to false.

Actors management

These configuration elements mainly define the number of authorized interactions with the remote service on a defined period. For example, one needs to ensure that the remote service will not be contacted more than 5 times per 3 seconds. *Throttle parallelism* defines the number of times and *Throttle duration* the period of time. Therefore, on the above example, throttle parallelism would be set to 5 and throttle duration would be set to 3 seconds.

- **Throttle duration*** *(finite duration)*:
  Set by default to 3 seconds. Must be a valid finite duration.
- **Throttle parallelism*** *(int)*:
  Set by default to 3.

5. Click on the save button.

You can update ✎ or delete 🗑 the Intune Connector.

CAUTION	You will not be able to delete an Intune Connector if it is referenced in any other configuration element.

12.1.3. Intune Profile

This section details how to configure an Intune Profile.

Required By

Intune Scheduled Tasks

Prerequisites

Intune Connector	PKI Connector	SCEP Authority

Setting up an SCEP Authority requires you to issue a certificate from the underlying PKI with the following characteristics:

- the issuing CA should be the same as the one that will issue certificates through the PKI Connector that will be linked to the Intune Profile
- the certificate key usages must include **Digital Signature** and **Key Encipherment**
- the certificate must be issued as PKCS#12 and then imported into Horizon

How to configure Intune Profile

1. Log in to Horizon Administration Interface.

2. Access Intune Profile from the drawer or card: **Third Parties › Intune › Profiles**.

3. Click on ⊕ .

4. Fill the mandatory fields.

Intune Profile Specific Configuration

General

- **Name*** *(string input)*:
  Enter a meaningful profile name. It must be unique for each profile. Horizon uses the name to identify the profile. As the name will be part of an URL, it is advised to use only lower case letters and dashes.
- **Enable*** *(boolean)*:
  Indicates whether the profile is enabled or not. Set to true by default.
- **Intune Connector*** *(select)*:
  Select an Intune Connector previously created.
- **PKI Connector** *(string select)*:
  Select a PKI connector previously created.

Assets identification

- **Device ID field name** *(string input)*:
  Subject DN field used to retrieve the Device ID. The selected field must be set to `{{AAD_Device_ID}}` on Intune side, e.g. if you select "L", the configured Subject DN in the SCEP profile in Intune must then contain `L={{AAD_Device_ID}}`. This is required to use the automated revocation feature upon device decommission.
- **Device ID separator** *(string input)*:
  Separator used to retrieve the Device ID in the device id field (if defined). This field is present

for backward compatibility reasons and should normally be left to blank.

SCEP protocol parameters

- **Mode*** *(select)*:
  Choose from one of the two modes RA or CA. Usually this should be set to **RA**.
- **SCEP Authority** *(select)*:
  Select a SCEP Authority previously created. See Prerequisites for details.
- **CAPS** *(select)*:
  Select one or many SCEP Capabilities from the list. If unsure, leave the default.
- **Encryption algorithm** *(select)*:
  Select a SCEP Encryption Algorithm algorithms from the list. If unsure, leave the default.

Crypto Policy

- **Authorized Key Types** *(multiselect)*:
  Key Types that can be used for enrollment. An empty value means no restrictions.

Max Certificate per Holder Policy

- **Maximum** *(int)*:
  When specified, define the maximum number of active certificates for a given Holder.
- **Behavior** *(select)*:
  What behavior to have when the maximum number is reached:
  - **revoke** the previous certificates.
  - **reject** the current request.
- **Revocation reason** *(select)*:
  When the revoke behavior is selected, the revocation reason to revoke the certificate with.

Common configuration for profiles

Languages

You can add more languages by clicking ⊕ .

- **Language*** *(select)*:
  Select a language. Supported languages are:
  - **en**: English
  - **fr**: French
- **Display Name** *(string input)*:
  Enter a display name. This will be the localized name of this profile.
- **Description** *(string input)*:
  Enter a description. This will be displayed on the list view of the profiles.

You can delete 🗑 the localization.

Grading Policies

You can select grading policies that will grade your certificate for a quick overview of its quality. For more information about the inner working of the grading policies in Horizon, please refer to the grading rules page.

Requests time to live

Configure the time your requests have before expiring.

> NOTE After expiration, requests are stored for an additional 30 days. This can be changed using configuration files.

- **Enrollment request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Renewal request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Revocation request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Update request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Migration request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Recover request** *(finite duration)*:
  *Enabled on **escrow**:* Must be a valid finite duration. The default value is set to seven days.

Constraints

- **Allowed email domains** *(string input)*:
  Enter a valid regular expression that the inputted emails should match. This includes RFC822NAME and UPN SANs as well as the contact email

  > NOTE This matches the domain of the email, not including anything before `@`.

- **Allowed DNS domains** *(string input)*:
  Enter a valid regular expression that the inputted domain should match.

CSR Data Mapping

1. Click on ⊕ to add a mapping.

2. Select a field and enter a value.

You can delete 🗑 the CSR Data Mapping.

Workflow

Data source flow

Configure which data sources to execute and in which order.

1. Select a data source to execute first, and fill its inputs with a computation rule.

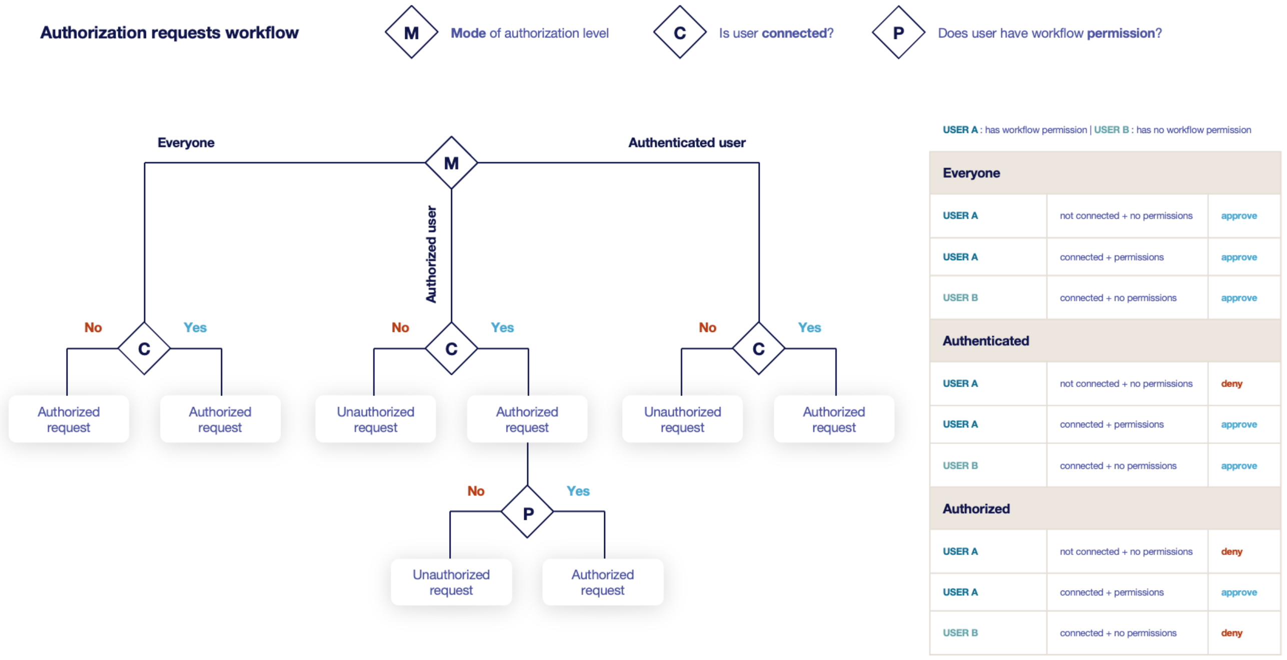
2. Add other data sources if needed. Each datasource input can use outputs from previously executed data sources.

3. All data sources output are available in computation rules throughout the certificate template and metadata.

Workflows builder

Configure custom rights for actions on this profile.

1. Select an authorization level for each workflow.



USER A : has workflow permission | USER B : has no workflow permission

Everyone		
USER A	not connected + no permissions	approve
USER A	connected + permissions	approve
USER B	connected + no permissions	approve
Authenticated		
USER A	not connected + no permissions	deny
USER A	connected + permissions	approve
USER B	connected + no permissions	approve
Authorized		
USER A	not connected + no permissions	deny
USER A	connected + permissions	approve
USER B	connected + no permissions	deny

- **Everyone**:
  No authentication is required.
- **Authenticated**:
  User has to be authenticated.
- **Authorized**:
  User has to be authenticated and have an explicit authorizations.

2. Select an access level for identity providers.

You can remove the access level for an identity provider by clicking on 'x'.

Owner-related permissions

These permissions apply to the owners of a certificate (team or owner). An owner can always request the following actions, but this permission allows them to perform the action without validation.

- **Revoke** *(boolean)*:
  Grant self revoke permission. The default value is set to false.
- **Update** *(boolean)*:
  Grant self update permission. The default value is set to false.

Certificate Template

This section details how to define a custom structure for the fields `subject DN`, `SAN` *&* `extensions` *of the requested certificate in order to match the configuration on the PKI side.*

Subject DN composition

You can add more elements by clicking ⊕ .

- **Element*** *(select)*:
  Select an attribute from the elements list.
- **Mandatory** *(boolean)*:
  Should the element be mandatory. The default value is set to false.
- **Editable by requester** *(boolean)*:
  Tells whether the element should be editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the element should be editable by the approver. The default value is set to false.
- **Default value** *(string input)*:
  Set a default value to the element.
- **Regex** *(regex)*:
  Enter a regular expression that the element should match.
- **Computation rule** *(Computation rule input)*:
  Set the value of this element to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

You can remove an element by clicking the delete button 🗑 or reorder (drag and drop) ⠿ the Subject DN template.

CAUTION	When a template is defined, at least one mandatory Common Name must be added to the DN Elements.

SAN composition

You can add more elements by clicking ⊕ .

- **Element*** *(select)*:
  Select an attribute from the element list.
- **Editable by requester** *(boolean)*:
  Tells whether the element should be editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the element should be editable by the approver. The default value is set to false.
- **Minimum** *(int)*:
  The minimum number of value that this SAN must have.
- **Maximum** *(int)*:
  The maximum number of value that this SAN must have.
- **Regex** *(regex)*:
  Enter a regular expression that the element should match.
- **Computation rule** *(Computation rule input)*:
  Set the value of this element to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

You can remove an element by clicking the delete button 🗑 or reorder (drag and drop) ⠿ the SAN template.

Extensions

You can add more elements by clicking ⊕ .

- **Element*** *(select)*:
  Select an attribute from the elements list.
- **Mandatory** *(boolean)*:
  Should the element be mandatory. The default value is set to false.
- **Editable by requester** *(boolean)*:
  Tells whether the element should be editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the element should be editable by the approver. The default value is set to false.
- **Default value** *(string input)*:
  Set a default value to the element.
- **Computation rule** *(Computation rule input)*:
  Set the value of this element to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

You can remove an element by clicking the delete button 🗑 or reorder (drag and drop) ⠿ the Extensions template.

CAUTION

When adding a SAN, a DN element or an Extension and making it mandatory, make sure to either give it a default value or a computation rule or make it editable, otherwise the template will be unusable.

Certificate Metadata

This section details how to define a custom structure for the labels, ownership policy and technical metadata, allowing certificates to hold rich information.

Labels

You can add more labels by clicking ⊕ .

- **Name** *(select)*:
  Select a preexisting label.
- **Mandatory** *(boolean)*:
  Should the label be mandatory. The default value is set to false.
- **Editable by requester** *(boolean)*:
  Tells whether the label should be editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the label should be editable by the approver. The default value is set to false.
- **Default value** *(string input)*:
  Set a default value to the label.
- **Label value restriction**
  - **Whitelist** *(string input multiple)*:
    The label value will have to be in the whitelist. Open the popup, enter the label value and press "enter" to add this value to the accepted value list. An empty whitelist means no restriction.
  - **Suggestions** *(string input multiple)*:
    Add suggestions that will be displayed to the user. The user will be able to choose one of these values or enter its own. Open the popup, enter your suggestions and press enter to add this value to the suggestions. An empty suggestions list means no restriction.
  - **Regex** *(regex)*:
    The label value will have to match the regex. Open the popup, enter the regular expression and click on the submit button to set the regex. An empty regex means no restrictions.
- **Computation rule** *(Computation rule input)*:
  Set the value of this label to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

You can delete 🗑 or reorder (drag and drop) ⠿ the label template.

Ownership policy

- **Owner**

  - **Mandatory** *(boolean)*:
    Specify if the certificate’s owner is mandatory when submitting a request.
  - **Editable by requester** *(boolean)*:
    Specify if the certificate’s owner can be overridden by the requester when submitting a request.
  - **Editable by approver** *(boolean)*:
    Specify if the certificate’s owner can be overridden by the requester when approving a request.
  - **Computation rule** *(Computation rule input)*:
    Set the value of the owner to the value of the evaluated computation rule. This value will override any other value including the user input.
- **Contact email**
  - **Mandatory** *(boolean)*:
    Specify if the certificate’s contact email is mandatory when submitting a request.
  - **Editable by requester** *(boolean)*:
    Specify if the certificate’s contact email can be overridden by the requester when submitting a request.
  - **Editable by approver** *(boolean)*:
    Specify if the certificate’s contact email can be overridden by the requester when approving a request.
  - **Default contact email** *(string input)*:
    Set a default contact email. This value must comply with the contact email restriction.
  - **Contact email restriction**
    - **Whitelist** *(string input multiple)*:
      The contact email will have to be in the whitelist. Open the popup, enter the email and press "enter" to add this value to the accepted whitelist. An empty whitelist means no restriction.
    - **Regex** *(regex)*:
      The contact email will have to match the regex. Open the popup, enter the regular expression and click on the submit button to set the regex. An empty regex means no restrictions.
  - **Computation rule** *(Computation rule input)*:
    Set the value of the contact email to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.
- **Team**
  - **Mandatory** *(boolean)*:
    Specify if the certificate’s team is mandatory when submitting a request.
  - **Editable by requester** *(boolean)*:
    Specify if the certificate’s team can be overridden by the requester when submitting a request.
  - **Editable by approver** *(boolean)*:

    Specify if the certificate’s team can be overridden by the requester when approving a request.

  - **Default team** *(string input)*:
    Set a default team. This value must comply with the team restriction.

  - **Team restriction**

    - **Whitelist** *(string input multiple)*:
      The team will have to be in the whitelist. Enter the team and press "enter" to add this value to the accepted whitelist. An empty whitelist means no restriction.

    - **Regex** *(regex)*:
      The team will have to match the regex. Open the popup, enter the regular expression and click on the submit button to set the regex. An empty regex means no restrictions.

  - **Computation rule** *(Computation rule input)*:
    Set the value of the team to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

Metadata policy *(overridable metadata)*

> **WARNING** These metadata are technical metadata. They are used by Horizon or Third party connectors, updating them should be done with utmost care.

> **NOTE** Metadata edition is not allowed on enroll.

> **NOTE** Metadata edition is not available via the User Interface. It must be changed with API, using horizon-cli.

You can allow the override of technical metadata by clicking ⊕ .

- **Metadata*** *(select)*:
  Select a metadata.

- **Editable by requester** *(boolean)*:
  Tells whether the metadata is editable by the requester. The default value is set to false.

- **Editable by approver** *(boolean)*:
  Tells whether the metadata is editable by the approver. The default value is set to false.

You can delete 🗑 a metadata policy. This will not delete the metadata but will make it non editable.

Notifications

This section details how to configure notifications on certificate and request lifecycle events.

Certificate lifecycle notifications

Notifications are sent when one of the following event is triggered by a certificate:

Enrollment	Revocation	Expire	Update	Migrate	Renew

Select a preexisting email, REST or groupware notification to associate it with an event.

Request lifecycle notifications

Notifications are sent when one of the following event is triggered by an Enroll/Revocation/Update/Migrate/Renew request:

Submit	Cancel	Revoke	Approve	Pending

Select a preexisting email, REST or groupware notification to associate it with an event.

NOTE Submit request events are not triggered when the user has the permission to perform the action directly.

5. Click on the save button.

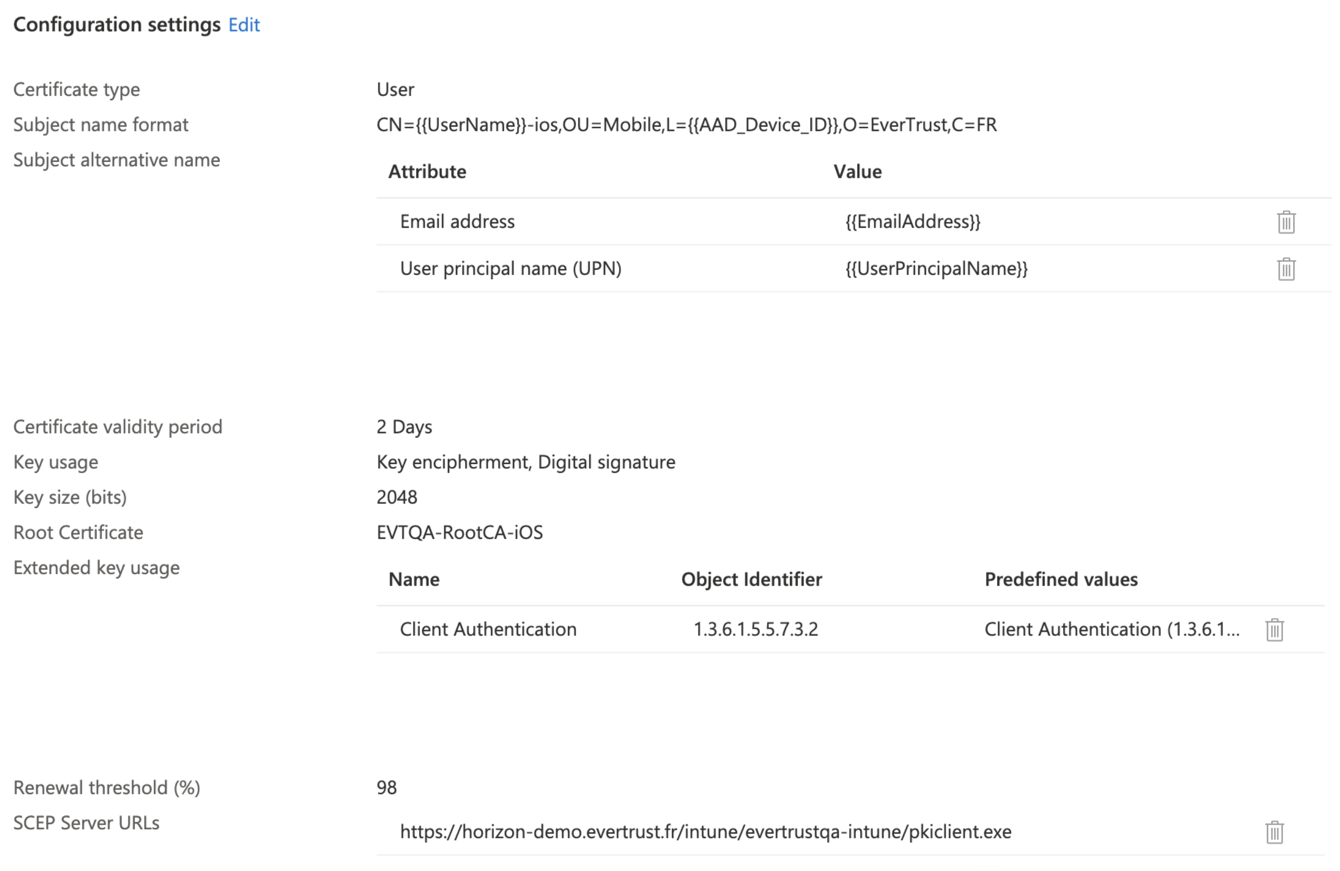
You can update ✎ or delete 🗑 the Intune Profile once it has been created.

CAUTION You won’t be able to delete an Intune Profile if it is referenced somewhere else.

Last steps

Once the profile is created in Horizon, you need to setup a SCEP profile in Intune by following Microsoft documentation. You will need to match the parameters in the Intune SCEP profile with what has been set up in Horizon and in the underlying PKI. You need to pay special attention to:

- the certificate lifetime and renewal interval, which must match throughout the solution
- the Subject and Subject Alternative Name settings must match throughout the solution. In the end, the issued certificate must contain exactly what was configured in Intune for these fields, or the renewal will not work.
- the SCEP server URL, where you need to input the URL given in the Intune Profile that you created in Horizon



Configuration settings Edit

Certificate type	User
Subject name format	CN={{UserName}}-ios,OU=Mobile,L={{AAD_Device_ID}},O=EverTrust,C=FR
Subject alternative name	

Attribute	Value
Email address	{{EmailAddress}}
User principal name (UPN)	{{UserPrincipalName}}

Certificate validity period	2 Days
Key usage	Key encipherment, Digital signature
Key size (bits)	2048
Root Certificate	EVTQA-RootCA-iOS
Extended key usage	

Name	Object Identifier	Predefined values
Client Authentication	1.3.6.1.5.5.7.3.2	Client Authentication (1.3.6.1...

Renewal threshold (%)	98
SCEP Server URLs	https://horizon-demo.evertrust.fr/intune/evertrustqa-intune/pkiclient.exe

CAUTION To enroll **Windows** machines or users using Intune, you need to remove the trailing " `pkiclient.exe` " from the SCEP server URL

12.1.4. Intune Scheduled Tasks

This section details how to configure scheduled tasks which will run periodically on your Intune profiles, in order to manage automatic revocation upon device decommission.

Prerequisites

Intune Connector	Intune Profile

How to configure Intune Scheduled Tasks

1. Log in to Horizon Administration Interface.

2. Access Intune Scheduled Tasks from the drawer or card: **Third Parties › Intune › Scheduled Tasks**.

3. Click on ⊕ .

4. Fill the mandatory fields.

- **Intune Profile*** *(select)*:
  Select an Intune profile previously created.

- **Target Connector*** *(select)*:
  Select an Intune connector previously created.
- **Cron scheduling** *(cron expression)*:
  Set to every 5 hours by default.
- **Revoke** *(boolean)*:
  Set to false by default. If true, Horizon will revoke any certificate associated to a device that has been deleted from Azure AD (and hence decommissioned).
- **Dry run** *(boolean)*:
  If enabled, revocation actions will not be performed. Instead, a message will be logged, explaining what would have been done.

5. Click on the save button.

You can run ▷ , update ✎ or delete 🗑 the Scheduled Tasks.

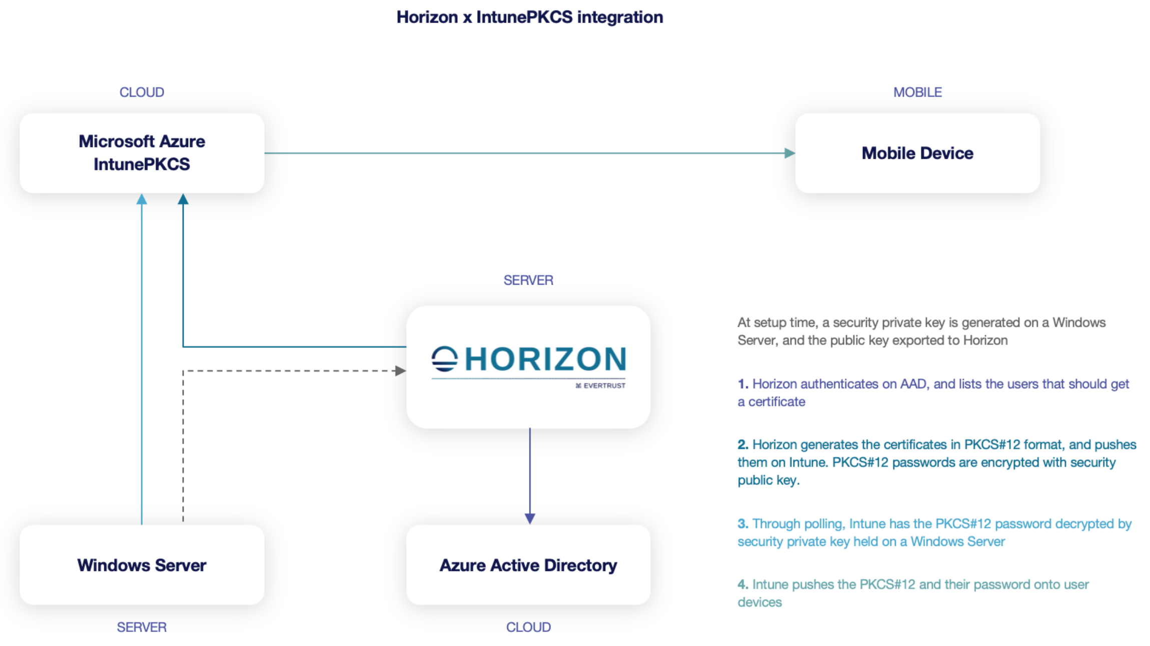
12.2. Intune PKCS

12.2.1. Intune PKCS Introduction

This integration involves at least three infrastructure components:

- Microsoft Endpoint Manager / Intune
- Azure Active Directory
- EverTrust Horizon

The enrolled devices interface with these components in order to retrieve their certificate.



At setup time, a security private key is generated on a Windows Server, and the public key exported to Horizon

1. Horizon authenticates on AAD, and lists the users that should get a certificate

2. Horizon generates the certificates in PKCS#12 format, and pushes them on Intune. PKCS#12 passwords are encrypted with security public key.

3. Through polling, Intune has the PKCS#12 password decrypted by security private key held on a Windows Server

4. Intune pushes the PKCS#12 and their password onto user devices

The diagram displays these components as well as the various flows involved in an enrollment. The integration is further explained in the Microsoft Intune PKCS documentation.

12.2.2. Intune PKCS Connector

This section details how to configure the Intune PKCS Connector.

Required By

Intune PKCS Profile

How to configure Intune PKCS Connector

Configuring the Microsoft Certificate Connector

The first step of the Intune PKCS connector is to actually understand the workflow that it bears, explained in introduction. Working with IntunePKCS requires the Microsoft Certificate Connector MSI to be uploaded to any Windows machine connected to the Internet. This connector is available on the Microsoft Documentation.

1. Run the certificate connector MSI on the machine and click on "Configure now". Configure the connector to fit your infrastructure, just remember to only check the *PKCS imported* box whenver prompted. This step should end with a connection to Azure;

2. Retrieve the **Horizon Key Manager** from Horizon and upload it to the same machine where the **Microsoft Certificate Connector** was installed;

3. Open a command-line prompt as Administrator;

4. Generate an import key through the command-line tool:

```
$ PKCSImport.exe generate [KeyName]
```

Replace [KeyName] with what you want to name your key as. The next steps of the documentation will assume that the name is set to "PKCSImportKey".

5. Use the tool to export the generated key:

```
$ PKCSImport.exe export PKCSImportKey PKCSImportKey.pub
```

This will export the public key part of the PKCS Import Key to the *PKCSImportKey.pub* file as base64 format.

Configuring the IntunePKCS Connector in Horizon

This step assumes that the previous one has been thoroughly followed. The only extra pre-requisite for this step is to retrieve the Azure resource ID of the group that will be using the escrowed certificates. Note that the app registration for Horizon must have the "*DeviceManagementConfiguration.ReadWrite.All*" permission granted as tenant admin. Read more about that in the Microsoft documentation

1. Log in to Horizon Administration Interface.

2. Access Intune PKCS Connectors from the drawer or card: **Third Parties › Intune PKCS › Connectors**.

3. Click on ⊕ .

4. Fill the mandatory fields.

General

- **Name*** *(string input)*:
  Enter a meaningful connector name. It must be unique for each connector. Horizon uses the name to identify the connector.
- **Azure Tenant*** *(string input)*:
  Value must be set to the Azure tenant.
- <strong>App Registration Credentials</strong>*<em class="mysize">(select)</em>:<br> Select <code>Login</code> <a anchor="<em>credentials">credentials</a> containing your app registration ID and secret key. The app registration must have the "_DeviceManagementConfiguration.ReadWrite.All</em>" permission granted as tenant admin.
- **Proxy** *(string select)*:
  The HTTP/HTTPS proxy to use.
- **Timeout** *(finite duration)*:
  Set by default at 10 seconds. Must be a valid finite duration.
- **Search Filter** *(string input)*:
  This value must be set to : groups/{Azure AD group object ID}/members This will apply the PKCS Import policy to all the members of the referenced Azure AD group object ID (the one that was retrieved at the beginning of the step).
- **Max stored certificates per holder** *(int)*:
  When specified, define the maximum number of certificates stored in the third party for a given holder. As an example, when set to 2, Intune will store the current certificate as well as the previous one (whether expired or revoked), so that it can still be used to decrypt resources. When a third one is going to be enrolled, the older one will be flushed out of Intune.

Assets identification and management

- **Key Name** *(string input)*:
  Enter the key name that was specified in the **Horizon Key Manager** (*PKCSImportKey* in the example).
- **Key Type** *(select)*:
  Select one key type from the list. If the **Horizon Key Manager** was used, select **RSA-2048**.

- **Provider Name** *(string input)*:
  Enter provider name. If the **Horizon Key Manager** was used, leave it blank.
- **Public Key** *(string input)*:
  Paste the base64 exported public key generated at step 5 of the previous part.
- **Intended Purpose** *(select)*:
  Select one intended certificate usage from the list. As an example, if you want to use the escrowed certificates through this connector to encrypt email, select S/MIME.

Actors and renewal management

These configuration elements mainly define the number of authorized interactions with the remote service on a defined period. For example, one needs to ensure that the remote service will not be contacted more than 5 times per 3 seconds. *Throttle parallelism* defines the number of times and *Throttle duration* the period of time. Therefore, on the above example, throttle parallelism would be set to 5 and throttle duration would be set to 3 seconds.

- **Throttle duration*** *(finite duration)*:
  Set by default at 3 seconds. Must be a valid finite duration.
- **Throttle parallelism*** *(int)*:
  Set by default at 3.
- **Renewal period** *(finite duration)*:
  Must be a valid finite duration.

5. Click on the save button.

You can update ✎ or delete 🗑 the Intune PKCS Connector.

CAUTION	You won’t be able to delete an Intune PKCS Connector if it is referenced in any other configuration element.

12.2.3. Intune PKCS Profile

This section details how to configure the Intune PKCS Profile

Required By

Intune PKCS Scheduled Tasks

Prerequisites

PKI Connector

How to configure Intune PKCS Profile

1. Log in to Horizon Administration Interface.

2. Access Intune PKCS Profiles from the drawer or card: **Third Parties › Intune PKCS › Profiles**.

3. Click on ⊕ .

4. Fill the mandatory fields.

Intune PKCS Profile Specific Configuration

General

- **Name*** *(string input)*:
  Enter a meaningful profile name. It must be unique for each profile. Horizon uses the name to identify the profile.
- **Enable*** *(boolean)*:
  Is the profile enabled or not. Set at true by default.
- **PKI Connector** *(string select)*:
  Select a PKI connector previously created. CAUTION: The selected PKI connector must support the msUPN SAN and, if used for S/MIME encryption, the RFC822NAME SAN (email).
- **Intune PKCS Connector*** *(select)*:
  Select an Intune PKCS Connector previously created.

Crypto Policy

- **Default Key Type** *(select)*:
  Select the default type of key to generate when using centralized enrollment mode.
- **Authorized Key Types** *(multiselect)*:
  Key Types that can be used for enrollment. An empty value means no restrictions.
- **Private key escrowing** *(boolean)*:
  Tells whether the private key should be escrowed by Horizon. The default value is set to false.
  - **Show PKCS#12 Password On Recover** *(boolean)*:
    Tells whether the PKCS#12 password should be displayed on recover. The default value is set to false.
  - **Show PKCS#12 On Recover** *(boolean)*:
    Tells whether the PKCS#12 should be displayed on recover. The default value is set to false.
- **PKCS#12 Password Mode*** *(select)*:
  Select how to generate PKCS#12 password:
  - **manual**: prompt the user to choose its password. This is the default behavior.
  - **random**: have the password generated on Horizon side.
- **Password policy** *(select)*:
  Select a previously created password policy. It will be enforced on PKCS#12 password for recovery and centralized enrollments.
- **Store encryption type*** *(select)*:
  Select an encryption algorithm from the list. The PKCS#12 will use this algorithm. The default

value is set to DES_AVERAGE.

Max Certificate per Holder Policy

- **Maximum** *(int)*:
  When specified, define the maximum number of active certificates for a given Holder.
- **Behavior** *(select)*:
  What behavior to have when the maximum number is reached:
  - **revoke** the previous certificates.
  - **reject** the current request.
- **Revocation reason** *(select)*:
  When the revoke behavior is selected, the revocation reason to revoke the certificate with.

Common configuration for profiles

Languages

You can add more languages by clicking ⊕ .

- **Language*** *(select)*:
  Select a language. Supported languages are:
  - **en**: English
  - **fr**: French
- **Display Name** *(string input)*:
  Enter a display name. This will be the localized name of this profile.
- **Description** *(string input)*:
  Enter a description. This will be displayed on the list view of the profiles.

You can delete 🗑 the localization.

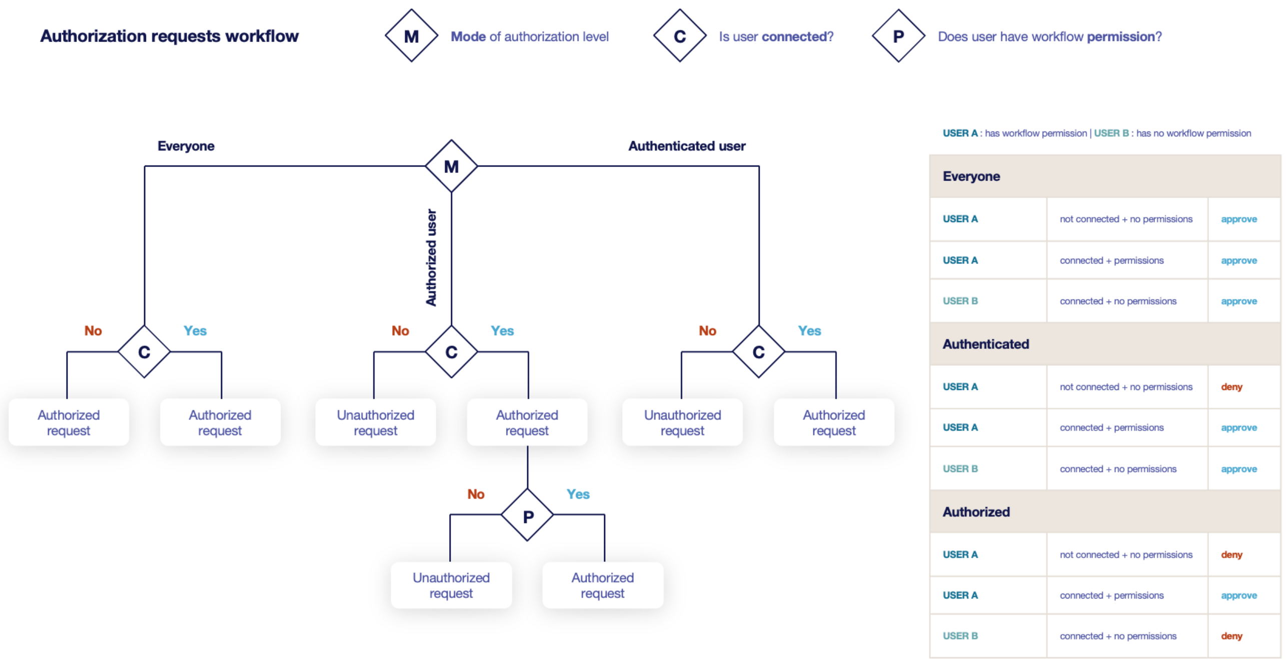
Grading Policies

You can select grading policies that will grade your certificate for a quick overview of its quality. For more information about the inner working of the grading policies in Horizon, please refer to the grading rules page.

Workflows builder

Configure custom rights for actions on this profile.

1. Select an authorization level for each workflow.



USER A : has workflow permission | **USER B** : has no workflow permission

Everyone		
USER A	not connected + no permissions	approve
USER A	connected + permissions	approve
USER B	connected + no permissions	approve
Authenticated		
USER A	not connected + no permissions	deny
USER A	connected + permissions	approve
USER B	connected + no permissions	approve
Authorized		
USER A	not connected + no permissions	deny
USER A	connected + permissions	approve
USER B	connected + no permissions	deny

- **Everyone**:
  No authentication is required.
- **Authenticated**:
  User has to be authenticated.
- **Authorized**:
  User has to be authenticated and have an explicit authorizations.

2. Select an access level for identity providers.

You can remove the access level for an identity provider by clicking on 'x'.

Requests time to live

Configure the time your requests have before expiring.

NOTE After expiration, requests are stored for an additional 30 days. This can be changed using configuration files.

- **Enrollment request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Renewal request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Revocation request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Update request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Migration request*** *(finite duration)*:

Must be a valid finite duration. The default value is set to seven days.

- **Recover request** *(finite duration)*:
  *Enabled on **escrow**:* Must be a valid finite duration. The default value is set to seven days.

Owner-related permission

These permissions apply to the owners of a certificate (team or owner). An owner can always request the following actions, but this permission allows them to perform the action without validation.

- **Revoke** *(boolean)*:
  Grant self revoke permission. The default value is set to false.
- **Recover** *(boolean)*:
  Grant self recover permission. The default value is set to false.
- **Update** *(boolean)*:
  Grant self update permission. The default value is set to false.

Constraints

- **Allowed email domains** *(string input)*:
  Enter a valid regular expression that the inputted emails should match. This includes RFC822NAME and UPN SANs as well as the contact email

  > **NOTE** This matches the domain of the email, not including anything before `@`.

- **Allowed DNS domains** *(string input)*:
  Enter a valid regular expression that the inputted domain should match.

CSR Data Mapping

1. Click on ⊕ to add a mapping.

2. Select a field and enter a value.

You can delete 🗑 the CSR Data Mapping.

Certificate Metadata

This section details how to define a custom structure for the labels, ownership policy and technical metadata, allowing certificates to hold rich information.

Labels

You can add more labels by clicking ⊕ .

- **Name** *(select)*:
  Select a preexisting label.
- **Mandatory** *(boolean)*:

Should the label be mandatory. The default value is set to false.

- **Editable by requester** *(boolean)*:
  Tells whether the label should be editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the label should be editable by the approver. The default value is set to false.
- **Default value** *(string input)*:
  Set a default value to the label.
- **Label value restriction**
  - **Whitelist** *(string input multiple)*:
    The label value will have to be in the whitelist. Open the popup, enter the label value and press "enter" to add this value to the accepted value list. An empty whitelist means no restriction.
  - **Suggestions** *(string input multiple)*:
    Add suggestions that will be displayed to the user. The user will be able to choose one of these values or enter its own. Open the popup, enter your suggestions and press enter to add this value to the suggestions. An empty suggestions list means no restriction.
  - **Regex** *(regex)*:
    The label value will have to match the regex. Open the popup, enter the regular expression and click on the submit button to set the regex. An empty regex means no restrictions.
- **Computation rule** *(Computation rule input)*:
  Set the value of this label to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

You can delete 🗑 or reorder (drag and drop) ⠿ the label template.

Ownership policy

- **Owner**
  - **Mandatory** *(boolean)*:
    Specify if the certificate’s owner is mandatory when submitting a request.
  - **Editable by requester** *(boolean)*:
    Specify if the certificate’s owner can be overridden by the requester when submitting a request.
  - **Editable by approver** *(boolean)*:
    Specify if the certificate’s owner can be overridden by the requester when approving a request.
  - **Computation rule** *(Computation rule input)*:
    Set the value of the owner to the value of the evaluated computation rule. This value will override any other value including the user input.
- **Contact email**
  - **Mandatory** *(boolean)*:
    Specify if the certificate’s contact email is mandatory when submitting a request.

  - **Editable by requester** *(boolean)*:
    Specify if the certificate's contact email can be overridden by the requester when submitting a request.
  - **Editable by approver** *(boolean)*:
    Specify if the certificate's contact email can be overridden by the requester when approving a request.
  - **Default contact email** *(string input)*:
    Set a default contact email. This value must comply with the contact email restriction.
  - **Contact email restriction**
    - **Whitelist** *(string input multiple)*:
      The contact email will have to be in the whitelist. Open the popup, enter the email and press "enter" to add this value to the accepted whitelist. An empty whitelist means no restriction.
    - **Regex** *(regex)*:
      The contact email will have to match the regex. Open the popup, enter the regular expression and click on the submit button to set the regex. An empty regex means no restrictions.
  - **Computation rule** *(Computation rule input)*:
    Set the value of the contact email to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.
- **Team**
  - **Mandatory** *(boolean)*:
    Specify if the certificate's team is mandatory when submitting a request.
  - **Editable by requester** *(boolean)*:
    Specify if the certificate's team can be overridden by the requester when submitting a request.
  - **Editable by approver** *(boolean)*:
    Specify if the certificate's team can be overridden by the requester when approving a request.
  - **Default team** *(string input)*:
    Set a default team. This value must comply with the team restriction.
  - **Team restriction**
    - **Whitelist** *(string input multiple)*:
      The team will have to be in the whitelist. Enter the team and press "enter" to add this value to the accepted whitelist. An empty whitelist means no restriction.
    - **Regex** *(regex)*:
      The team will have to match the regex. Open the popup, enter the regular expression and click on the submit button to set the regex. An empty regex means no restrictions.
  - **Computation rule** *(Computation rule input)*:
    Set the value of the team to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

Metadata policy *(overridable metadata)*

WARNING These metadata are technical metadata. They are used by Horizon or Third party connectors, updating them should be done with utmost care.

NOTE Metadata edition is not allowed on enroll.

NOTE Metadata edition is not available via the User Interface. It must be changed with API, using horizon-cli.

You can allow the override of technical metadata by clicking ⊕ .

- **Metadata*** *(select)*:
  Select a metadata.
- **Editable by requester** *(boolean)*:
  Tells whether the metadata is editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the metadata is editable by the approver. The default value is set to false.

You can delete 🗑 a metadata policy. This will not delete the metadata but will make it non editable.

Notifications/Triggers

This section details how to configure notifications and triggers to perform actions on certificate and request lifecycle events.

Certificate lifecycle notifications

Notifications are sent when one of the following event is triggered by a certificate:

Enrollment	Revocation	Expire	Update	Migrate	Renew

Select a preexisting email, REST or groupware notification to associate it with an event.

Request lifecycle notifications

Notifications are sent when one of the following event is triggered by an Enroll/Revocation/Update/Migrate/Renew request:

Submit	Cancel	Revoke	Approve	Pending

Select a preexisting email, REST or groupware notification to associate it with an event.

NOTE Submit request events are not triggered when the user has the permission to perform the action directly.

Triggers

Horizon support the use of third-party triggers in the form of callbacks on specific events happening on the profile, giving a way to synchronize the third party repositories and Horizon.

- **Enrollment** *(select)*:
  Select the preexisting third party or MDM trigger(s) to call whenever a certificate is enrolled on this profile.
- **Renewal** *(select)*:
  Select the preexisting third party or MDM trigger(s) to call whenever a certificate is renewed on this profile.
- **Revocation** *(select)*:
  Select the preexisting third party or MDM trigger(s) to call whenever a certificate gets revoked on this profile.
- **Expire** *(select)*:
  Select the preexisting third party or MDM trigger(s) to call whenever a certificate expires on this profile.

The available triggers are the following:

AKV Triggers	AWS Triggers	F5 Triggers	LDAP Triggers	*On **WebRA** and **Intune PKCS** only:* Intune PKCS Triggers

5. Click on the save button.

You can update ✎ or delete 🗑 the Intune PKCS Profile.

CAUTION

You won't be able to delete an Intune PKCS Profile if it is referenced in any other configuration element.

12.2.4. Intune PKCS Scheduled Tasks

This section details how to schedule tasks that will run periodically on your Intune PKCS profiles.

Prerequisites

Intune PKCS Connector	Intune PKCS Profile

How to configure Intune PKCS Scheduled Tasks

1. Log in to Horizon Administration Interface.

2. Access Intune PKCS Scheduled Tasks from the drawer or card: **Third Parties › Intune PKCS › Scheduled Tasks**.

3. Click on ⊕ .

4. Fill the mandatory fields.

- **Enable** *(boolean)*:
  Tells whether the Scheduled task should be enabled. Set by default at true.
- **Intune PKCS Profile*** *(select)*:
  Select an Intune PKCS profile previously created.
- **Target Connector*** *(select)*:
  Select an Intune PKCS connector previously created.
- **Cron scheduling** *(cron expression)*:
  By default set at every 5 hours.
- **Enroll?** *(boolean)*:
  If enabled, will enroll all certificate from the third party repository. Set to false by default.
- **Revoke?** *(boolean)*:
  If enabled, will revoke all certificate whose container was deleted from the third party repository. Set to false by default.
- **Renew?** *(boolean)*:
  If enabled, will renew all certificate who are about to expire. Set to false by default.
- **Dry run** *(boolean)*:
  If enabled, enroll, revocation and renewal actions will not be performed. Instead, a message will be logged, explaining what would have been done.

5. Click on the save button.

You can run ▷ or update ✎ or delete 🗑 the Schedules Tasks.

12.2.5. Intune PKCS Trigger

This section details how to configure the Triggers that will run automatically on your Intune PKCS connectors.

Prerequisites

Intune PKCS Connector

How to configure Intune PKCS Trigger

1. Log in to Horizon Administration Interface.

2. Access Intune PKCS Triggers from the drawer or card: **Third Parties** › **Intune PKCS** › **Triggers**.

3. Click on ⊕ .

4. Fill the mandatory fields.

- **Name*** *(string input)*:
  Enter a meaningful trigger name. It must be unique for each trigger. Horizon uses the name to identify the trigger.
- **Intune PKCS Connector*** *(select)*:
  Select an Intune PKCS connector previously created.
- **Retries in case of error** *(int)*:
  Number of times to retry to push the change on the Intune PKCS repository in case of error. Must be an integer between 1 and 15.

5. Click on the save button.

You can update ✎ or delete 🗑 the Intune PKCS Trigger.

CAUTION You won’t be able to delete an Intune PKCS Trigger if it is referenced in any other configuration element.

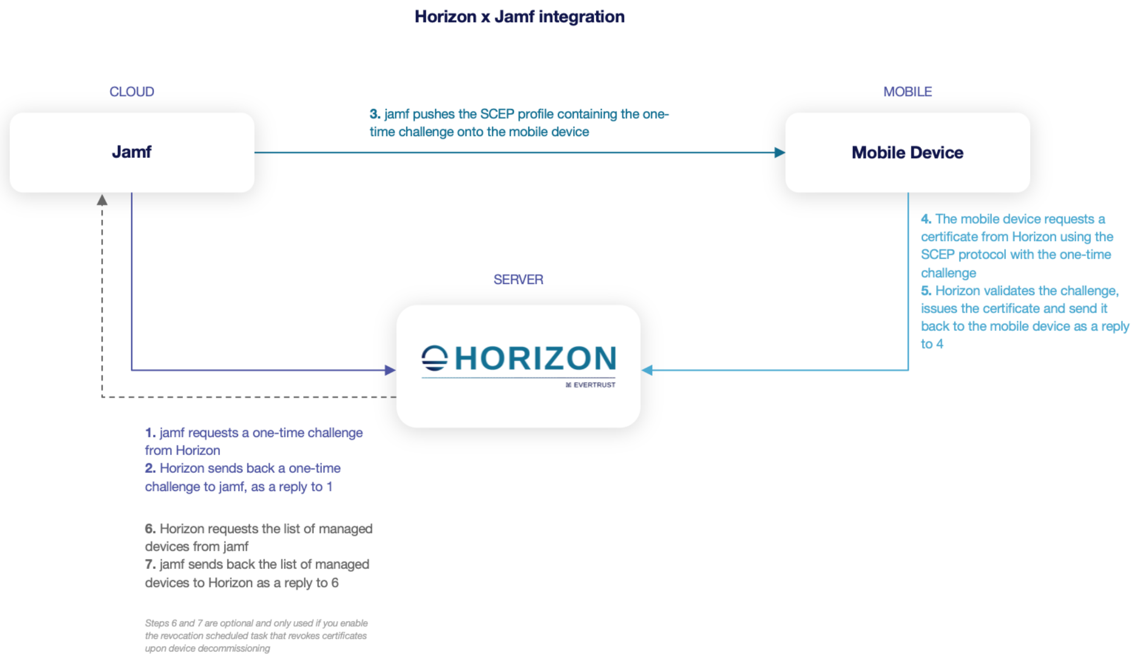
12.3. Jamf

12.3.1. Jamf Introduction

This section details the Jamf Pro integration with Horizon, used to enroll, renew and revoke certificates on Jamf Pro managed devices.

This integration involves the following components:

- Jamf Pro server or Cloud instance
- EverTrust Horizon
- Devices to be enrolled


Horizon x Jamf integration
CLOUD
Jamf
MOBILE
Mobile Device
SERVER
HORIZON
EVERTRUST
3. jamf pushes the SCEP profile containing the one-time challenge onto the mobile device
4. The mobile device requests a certificate from Horizon using the SCEP protocol with the one-time challenge
5. Horizon validates the challenge, issues the certificate and send it back to the mobile device as a reply to 4
1. jamf requests a one-time challenge from Horizon
2. Horizon sends back a one-time challenge to jamf, as a reply to 1
6. Horizon requests the list of managed devices from jamf
7. jamf sends back the list of managed devices to Horizon as a reply to 6
Steps 6 and 7 are optional and only used if you enable the revocation scheduled task that revokes certificates upon device decommissioning

The diagram displays these components as well as the various flows involved in an enrollment.

Finally, this integration will require to setup, on Horizon side, the following elements:

- a Jamf Connector, which holds the configuration items required for Horizon to connect to Jamf Pro
- a Jamf Profile, which holds the configuration items specifying how Horizon should issue certificates for the specified Jamf Connector
- a Jamf Schedule Task, which holds configuration items defining the scheduled task in charge of performing revocation upon decommissioning devices from Jamf Pro. This is optional.

12.3.2. Jamf Connector

This section details how to configure a Jamf Connector.

Required By

Jamf Profile

Prerequisites

On Horizon side, you might need to set up a Proxy used to reach Jamf Pro, if necessary.

On Jamf Pro side, it is required to create a technical user for Horizon, and give it **Auditor** rights, so that Horizon will be able to list the devices managed by Jamf Pro and thus be able to trigger certificate revocation upon decommissioning. Please follow the steps from the Jamf Pro documentation. After performing these steps, you will be given the following information, required later:

- a login
- a password

How to configure Jamf Connector

1. Log in to Horizon Administration Interface.

2. Access Jamf Connector from the drawer or card: **Third Parties › Jamf › Connectors**.

3. Click on ⊕ .

4. Fill the mandatory fields.

Connection

- **Name*** *(string input)*:
  Enter a meaningful connector name. It must be unique for each connector. Horizon uses the name to identify the connector.
- **Jamf endpoint URL*** *(string input)*:
  Enter the URL pointing to the Jamf deployment or the Jamf Cloud instance.
- **Jamf user account credentials*** *(select)*:
  Select `Login` credentials containing the username and password created for Horizon in Jamf.
- **Proxy** *(string select)*:
  The HTTP/HTTPS proxy used to reach Jamf Pro, if any.
- **Timeout** *(finite duration)*:
  Set by default at 10 seconds. Must be a valid finite duration.

Actors management

These configuration elements mainly define the number of authorized interactions with the remote service on a defined period. For example, one needs to ensure that the remote service will not be contacted more than 5 times per 3 seconds. *Throttle parallelism* defines the number of times and *Throttle duration* the period of time. Therefore, on the above example, throttle parallelism would be set to 5 and throttle duration would be set to 3 seconds.

- **Throttle duration*** *(finite duration)*:
  Set by default to 3 seconds. Must be a valid finite duration.
- **Throttle parallelism*** *(int)*:
  Set by default to 3.

5. Click on the save button.

You can update ✎ or delete 🗑 the Jamf Connector.

CAUTION You won’t be able to delete a Jamf Connector if it is referenced in any other configuration element.

12.3.3. Jamf Profile

This section details how to configure a Jamf Profile

Prerequisites

Jamf Connector	PKI Connector	SCEP Authority

The SCEP Authority setup requires you to issue a certificate from the underlying PKI with the following characteristics:

- to issue certificates for iOS:
  - the issuing CA should be the same as the one that will issue certificates through the PKI Connector that will be linked to the Jamf Profile
  - the certificate key usages must include **Digital Signature** and **Key Encipherment**
  - the certificate must be issued as PKCS#12 and then imported into Horizon
- to issue certificates for macOS:
  - the certificate should be self-signed
  - the certificate key usages must include **Digital Signature** and **Key Encipherment**
  - the certificate must be issued as PKCS#12 and then imported into Horizon

How to configure Jamf Profile

1. Log in to Horizon Administration Interface.

2. Access Jamf Profiles from the drawer or card: **Third Parties › Jamf › Profiles**.

3. Click on ⊕ .

4. Fill the mandatory fields.

Jamf Profile Specific Configuration

General

- **Name*** *(string input)*:
  Enter a meaningful profile name. It must be unique for each profile. Horizon uses the name to identify the profile. As the name will be part of an URL, it is advised to use only lower case letters and dashes.
- **Enable** *(boolean)*:
  Is the profile enabled or not. Set at true by default.
- **Jamf Connector** *(select)*:
  Select a Jamf connector previously created.
- **PKI connector*** *(string select)*:

Select a PKI connector previously created.

Assets identification

- **DN field containing the device** UDID* *(select)*:
  Field used to retrieve the Device ID. The selected field must be set to $UDID/$COMPUTERNAME on Jamf side, e.g. if you select "L", the configured Subject DN in the SCEP profile in Jamf pro must then contain L=$UDID for iOS or L=$COMPUTERNAME for macOS devices. This allows to use the automated revocation upon device decommissioning feature.

SCEP protocol parameters

- **Mode*** *(select)*:
  Choose from the two modes RA or CA. To enroll certificates on **iOS** devices, select the **RA** mode. To enroll certificates on **macOS**, select the **CA** mode.
- **SCEP Authority*** *(select)*:
  Select a SCEP Authority previously created. See Prerequisites for details.
- **CAPS** *(select)*:
  Select one or many SCEP Capabilities from the list. If unsure, leave the default.
- **Encryption algorithm*** *(select)*:
  Select a SCEP Encryption Algorithm algorithms from the list. If unsure, leave the default.
- **Password policy** *(select)*:
  Choose from the password policy you might have previously created. If unsure, leave the default.

Crypto Policy

- **Authorized Key Types** *(multiselect)*:
  Key Types that can be used for enrollment. An empty value means no restrictions.

Max Certificate per Holder Policy

- **Maximum** *(int)*:
  When specified, define the maximum number of active certificates for a given Holder.
- **Behavior** *(select)*:
  What behavior to have when the maximum number is reached:
  - **revoke** the previous certificates.
  - **reject** the current request.
- **Revocation reason** *(select)*:
  When the revoke behavior is selected, the revocation reason to revoke the certificate with.

Common configuration for profiles

Languages

You can add more languages by clicking ⊕ .

- **Language*** *(select)*:
  Select a language. Supported languages are:
  - **en**: English
  - **fr**: French
- **Display Name** *(string input)*:
  Enter a display name. This will be the localized name of this profile.
- **Description** *(string input)*:
  Enter a description. This will be displayed on the list view of the profiles.

You can delete 🗑 the localization.

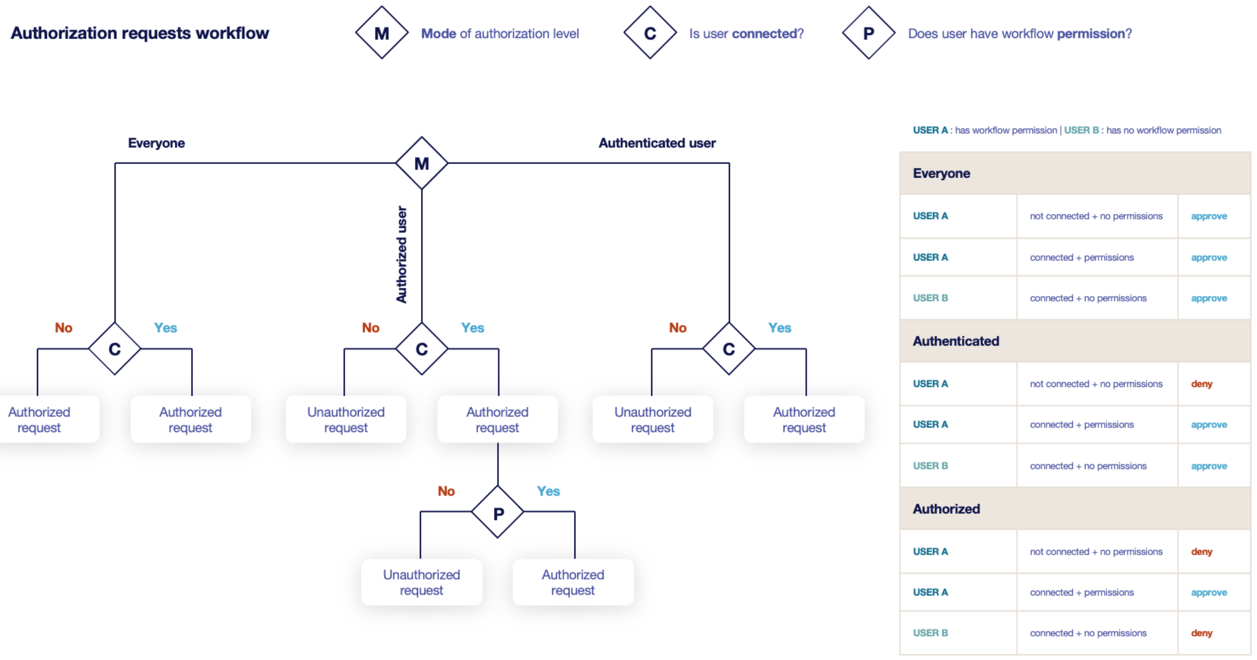
Grading Policies

You can select grading policies that will grade your certificate for a quick overview of its quality. For more information about the inner working of the grading policies in Horizon, please refer to the grading rules page.

Workflows builder

Configure custom rights for actions on this profile.

1. Select an authorization level for each workflow.



USER A : has workflow permission | **USER B** : has no workflow permission

Everyone		
USER A	not connected + no permissions	approve
USER A	connected + permissions	approve
USER B	connected + no permissions	approve
Authenticated		
USER A	not connected + no permissions	deny
USER A	connected + permissions	approve
USER B	connected + no permissions	approve
Authorized		
USER A	not connected + no permissions	deny
USER A	connected + permissions	approve
USER B	connected + no permissions	deny

- **Everyone**:
  No authentication is required.

- **Authenticated**:
  User has to be authenticated.
- **Authorized**:
  User has to be authenticated and have an explicit authorizations.

2. Select an access level for identity providers.

You can remove the access level for an identity provider by clicking on 'x'.

Requests time to live

Configure the time your requests have before expiring.

> **NOTE** After expiration, requests are stored for an additional 30 days. This can be changed using configuration files.

- **Enrollment request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Renewal request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Revocation request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Update request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Migration request*** *(finite duration)*:
  Must be a valid finite duration. The default value is set to seven days.
- **Recover request** *(finite duration)*:
  *Enabled on **escrow**:* Must be a valid finite duration. The default value is set to seven days.

Owner-related permission

These permissions apply to the owners of a certificate (team or owner). An owner can always request the following actions, but this permission allows them to perform the action without validation.

- **Revoke** *(boolean)*:
  Grant self revoke permission. The default value is set to false.
- **Update** *(boolean)*:
  Grant self update permission. The default value is set to false.

Constraints

- **Allowed email domains** *(string input)*:
  Enter a valid regular expression that the inputted emails should match. This includes RFC822NAME and UPN SANs as well as the contact email

  > **NOTE** This matches the domain of the email, not including anything before `@`.

- **Allowed DNS domains** *(string input)*:
  Enter a valid regular expression that the inputted domain should match.

CSR Data Mapping

1. Click on ⊕ to add a mapping.

2. Select a field and enter a value.

You can delete 🗑 the CSR Data Mapping.

Certificate Metadata

This section details how to define a custom structure for the labels, ownership policy and technical metadata, allowing certificates to hold rich information.

Labels

You can add more labels by clicking ⊕ .

- **Name** *(select)*:
  Select a preexisting label.
- **Mandatory** *(boolean)*:
  Should the label be mandatory. The default value is set to false.
- **Editable by requester** *(boolean)*:
  Tells whether the label should be editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the label should be editable by the approver. The default value is set to false.
- **Default value** *(string input)*:
  Set a default value to the label.
- **Label value restriction**
  - **Whitelist** *(string input multiple)*:
    The label value will have to be in the whitelist. Open the popup, enter the label value and press "enter" to add this value to the accepted value list. An empty whitelist means no restriction.
  - **Suggestions** *(string input multiple)*:
    Add suggestions that will be displayed to the user. The user will be able to choose one of these values or enter its own. Open the popup, enter your suggestions and press enter to add this value to the suggestions. An empty suggestions list means no restriction.
  - **Regex** *(regex)*:
    The label value will have to match the regex. Open the popup, enter the regular expression and click on the submit button to set the regex. An empty regex means no restrictions.
- **Computation rule** *(Computation rule input)*:
  Set the value of this label to the value of the evaluated computation rule. This value will

override any other value including the user input and the default value.

You can delete 🗑 or reorder (drag and drop) ⠿ the label template.

Ownership policy

- **Owner**
  - **Mandatory** *(boolean)*:
    Specify if the certificate’s owner is mandatory when submitting a request.
  - **Editable by requester** *(boolean)*:
    Specify if the certificate’s owner can be overridden by the requester when submitting a request.
  - **Editable by approver** *(boolean)*:
    Specify if the certificate’s owner can be overridden by the requester when approving a request.
  - **Computation rule** *(Computation rule input)*:
    Set the value of the owner to the value of the evaluated computation rule. This value will override any other value including the user input.
- **Contact email**
  - **Mandatory** *(boolean)*:
    Specify if the certificate’s contact email is mandatory when submitting a request.
  - **Editable by requester** *(boolean)*:
    Specify if the certificate’s contact email can be overridden by the requester when submitting a request.
  - **Editable by approver** *(boolean)*:
    Specify if the certificate’s contact email can be overridden by the requester when approving a request.
  - **Default contact email** *(string input)*:
    Set a default contact email. This value must comply with the contact email restriction.
  - **Contact email restriction**
    - **Whitelist** *(string input multiple)*:
      The contact email will have to be in the whitelist. Open the popup, enter the email and press "enter" to add this value to the accepted whitelist. An empty whitelist means no restriction.
    - **Regex** *(regex)*:
      The contact email will have to match the regex. Open the popup, enter the regular expression and click on the submit button to set the regex. An empty regex means no restrictions.
  - **Computation rule** *(Computation rule input)*:
    Set the value of the contact email to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.
- **Team**

  - **Mandatory** *(boolean)*:
    Specify if the certificate’s team is mandatory when submitting a request.
  - **Editable by requester** *(boolean)*:
    Specify if the certificate’s team can be overridden by the requester when submitting a request.
  - **Editable by approver** *(boolean)*:
    Specify if the certificate’s team can be overridden by the requester when approving a request.
  - **Default team** *(string input)*:
    Set a default team. This value must comply with the team restriction.
  - **Team restriction**
    - **Whitelist** *(string input multiple)*:
      The team will have to be in the whitelist. Enter the team and press "enter" to add this value to the accepted whitelist. An empty whitelist means no restriction.
    - **Regex** *(regex)*:
      The team will have to match the regex. Open the popup, enter the regular expression and click on the submit button to set the regex. An empty regex means no restrictions.
  - **Computation rule** *(Computation rule input)*:
    Set the value of the team to the value of the evaluated computation rule. This value will override any other value including the user input and the default value.

Metadata policy *(overridable metadata)*

WARNING These metadata are technical metadata. They are used by Horizon or Third party connectors, updating them should be done with utmost care.

NOTE Metadata edition is not allowed on enroll.

NOTE Metadata edition is not available via the User Interface. It must be changed with API, using horizon-cli.

You can allow the override of technical metadata by clicking ⊕ .

- **Metadata*** *(select)*:
  Select a metadata.
- **Editable by requester** *(boolean)*:
  Tells whether the metadata is editable by the requester. The default value is set to false.
- **Editable by approver** *(boolean)*:
  Tells whether the metadata is editable by the approver. The default value is set to false.

You can delete 🗑 a metadata policy. This will not delete the metadata but will make it non editable.

Notifications

This section details how to configure notifications on certificate and request lifecycle events.

Certificate lifecycle notifications

Notifications are sent when one of the following event is triggered by a certificate:

Enrollment	Revocation	Expire	Update	Migrate	Renew

Select a preexisting email, REST or groupware notification to associate it with an event.

Request lifecycle notifications

Notifications are sent when one of the following event is triggered by an Enroll/Revocation/Update/Migrate/Renew request:

Submit	Cancel	Revoke	Approve	Pending

Select a preexisting email, REST or groupware notification to associate it with an event.

NOTE Submit request events are not triggered when the user has the permission to perform the action directly.

5. Click on the save button.

You can update ✎ or delete 🗑 the Jamf Profile .

CAUTION You won’t be able to delete a Jamf Profile if it is referenced somewhere else.

Last Steps

The integration between Jamf Pro and Horizon can be done in the following modes:

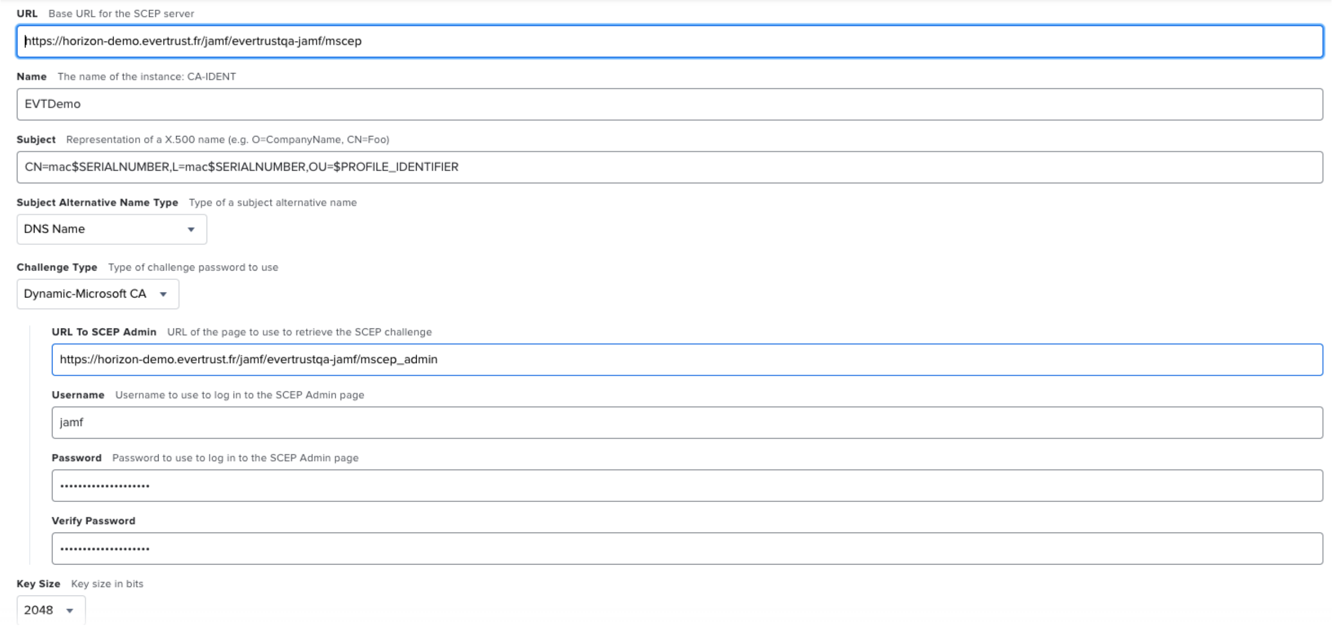
- Jamf Pro SCEP Proxy mode
- iOS SCEP Profile
- macOS SCEP Profile
- macOS SCEP Profile with Proxy

In all these modes, the Challenge type to use on Jamf Pro side is **Dynamic-Microsoft CA**, and you should point to the corresponding `mscep` and `mscep_admin` URI on Horizon side, that can be found in the Jamf Profile after it has been created.

Jamf Pro SCEP Proxy mode

This mode requires to provide the SCEP Authority PKCS#12 to Jamf Pro, so that it can be uploaded in the appropriate profile.

Other than that, the configuration looks like the following on Jamf Pro side:



URL Base URL for the SCEP server
https://horizon-demo.evertrust.fr/jamf/evertrustqa-jamf/mscep

Name The name of the instance: CA-IDENT
EVTDemo

Subject Representation of a X.500 name (e.g. O=CompanyName, CN=Foo)
CN=mac$SERIALNUMBER,L=mac$SERIALNUMBER,OU=$PROFILE_IDENTIFIER

Subject Alternative Name Type Type of a subject alternative name
DNS Name

Challenge Type Type of challenge password to use
Dynamic-Microsoft CA

URL To SCEP Admin URL of the page to use to retrieve the SCEP challenge
https://horizon-demo.evertrust.fr/jamf/evertrustqa-jamf/mscep_admin

Username Username to use to log in to the SCEP Admin page
jamf

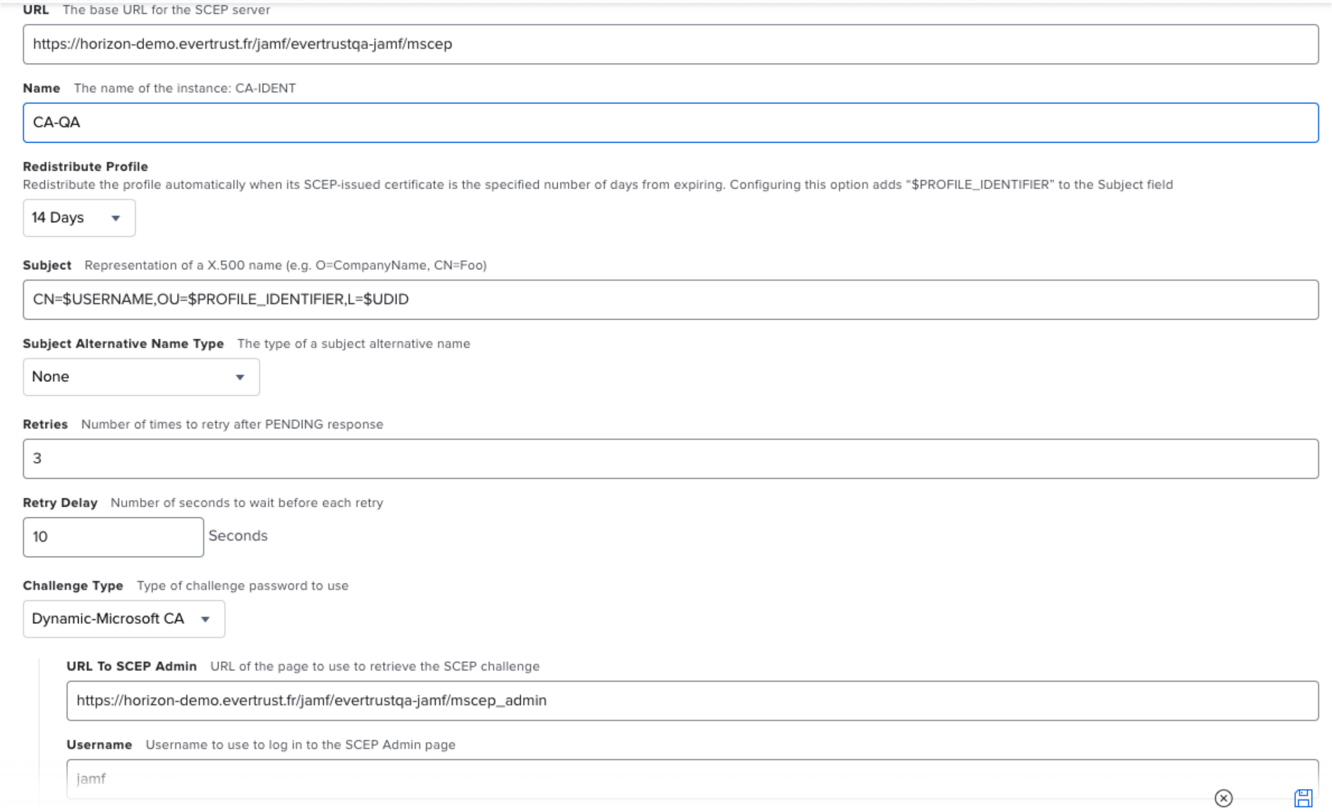
Password Password to use to log in to the SCEP Admin page
•••••••••••••••••••••

Verify Password
•••••••••••••••••••••

Key Size Key size in bits
2048

iOS/macOS SCEP Profile

On Jamf Pro side, the profile configuration looks like the following:



URL The base URL for the SCEP server
https://horizon-demo.evertrust.fr/jamf/evertrustqa-jamf/mscep

Name The name of the instance: CA-IDENT
CA-QA

Redistribute Profile
Redistribute the profile automatically when its SCEP-issued certificate is the specified number of days from expiring. Configuring this option adds "$PROFILE_IDENTIFIER" to the Subject field
14 Days

Subject Representation of a X.500 name (e.g. O=CompanyName, CN=Foo)
CN=$USERNAME,OU=$PROFILE_IDENTIFIER,L=$UDID

Subject Alternative Name Type The type of a subject alternative name
None

Retries Number of times to retry after PENDING response
3

Retry Delay Number of seconds to wait before each retry
10 Seconds

Challenge Type Type of challenge password to use
Dynamic-Microsoft CA

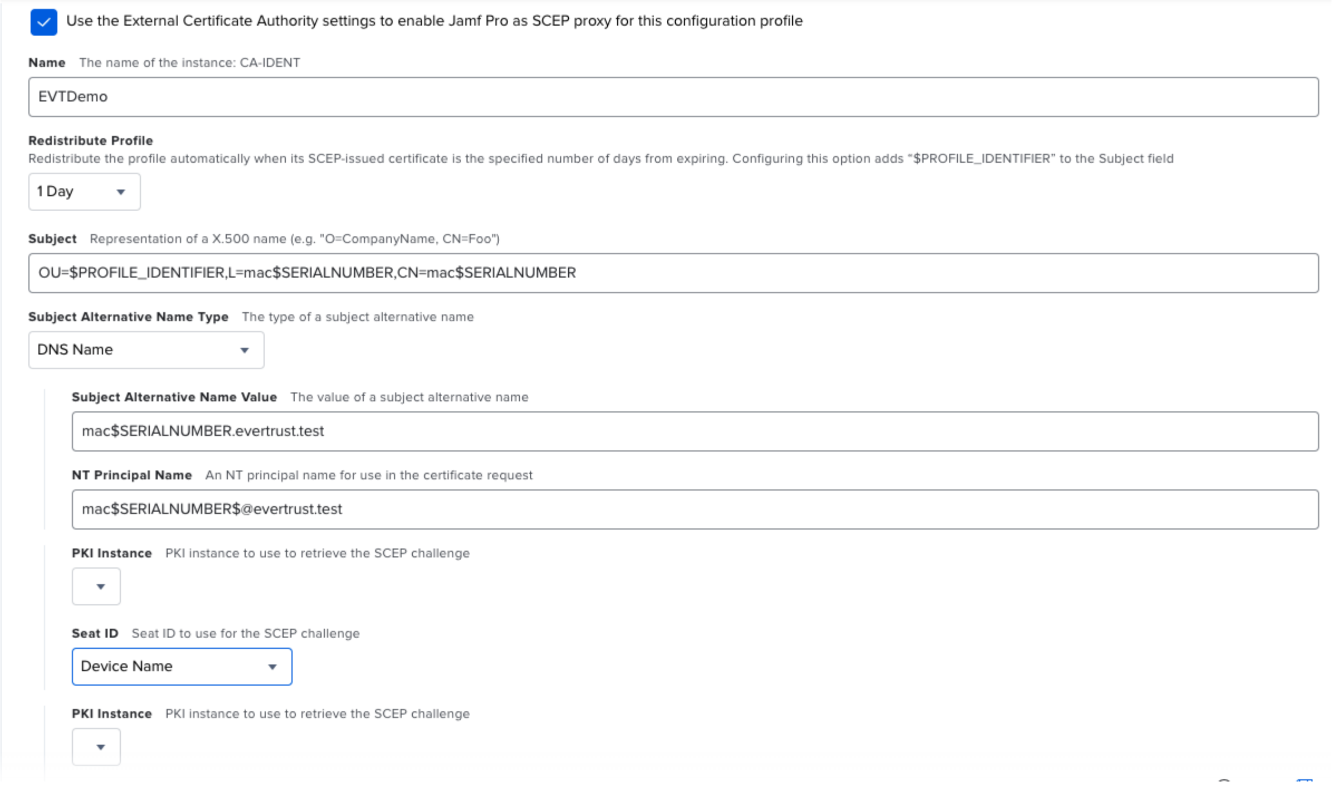
URL To SCEP Admin URL of the page to use to retrieve the SCEP challenge
https://horizon-demo.evertrust.fr/jamf/evertrustqa-jamf/mscep_admin

Username Username to use to log in to the SCEP Admin page
jamf

macOS SCEP Profile with Proxy

This mode requires:

1. to set up the SCEP Proxy mode on Jamf Pro side
2. to configure a profile on Jamf Pro side, that looks like the following:



☑ Use the External Certificate Authority settings to enable Jamf Pro as SCEP proxy for this configuration profile

Name The name of the instance: CA-IDENT

EVTDemo

Redistribute Profile
Redistribute the profile automatically when its SCEP-issued certificate is the specified number of days from expiring. Configuring this option adds "$PROFILE_IDENTIFIER" to the Subject field

1 Day ▾

Subject Representation of a X.500 name (e.g. "O=CompanyName, CN=Foo")

OU=$PROFILE_IDENTIFIER,L=mac$SERIALNUMBER,CN=mac$SERIALNUMBER

Subject Alternative Name Type The type of a subject alternative name

DNS Name ▾

Subject Alternative Name Value The value of a subject alternative name

mac$SERIALNUMBER.evertrust.test

NT Principal Name An NT principal name for use in the certificate request

mac$SERIALNUMBER$@evertrust.test

PKI Instance PKI instance to use to retrieve the SCEP challenge

▾

Seat ID Seat ID to use for the SCEP challenge

Device Name ▾

PKI Instance PKI instance to use to retrieve the SCEP challenge

▾

12.3.4. Jamf Scheduled Tasks

This section details how to schedule tasks that will run periodically on your jamf profiles, in order to manage automatic revocation upon device decommissioning.

Prerequisites

Jamf Connector	Jamf Profile

How to configure Jamf Scheduled Tasks

1. Log in to Horizon Administration Interface.

2. Access Jamf Scheduled Tasks from the drawer or card: **Third Parties › Jamf › Scheduled Tasks**.

3. Click on ⊕ .

4. Fill the mandatory fields.

- **Enable** *(boolean)*:
  Tells whether the Scheduled task should be enabled. Set by default at true.
- **Jamf Profile*** *(select)*:
  Select a Jamf profile previously created.
- **Target Connector*** *(select)*:
  Select an Jamf connector previously created.
- **Cron scheduling** *(cron expression)*:
  Set to every 5 hours by default.

- **Revoke** *(boolean)*:
  Set to false by default. If true, Horizon will revoke any certificate associated to a device that has been deleted from Azure AD (and hence decommissioned).
- **Dry run** *(boolean)*:
  If enabled, revocation actions will not be performed. Instead, a message will be logged, explaining what would have been done.

5. Click on the save button.

You can run ▷ or update ✎ or delete 🗑 the Scheduled Tasks.

13. System configuration

13.1. Labels

This section details how to configure the labels. Labels are metadata used to store information provided by the en-users in Horizon database, associated to a given certificate, but not contained in the certificate.

You will be able to associate the labels created in this section with your profiles in order to enrich the certificates that will be issued from them.

13.1.1. How to configure a Label

1. Log in to Horizon Administration Interface.

2. Access Labels from the drawer or card: **System** › **Labels**.

3. Click on ⊕ .

4. Fill the following fields:

- **Name*** *(string input)*:
  Enter a meaningful Label name.

You can add more languages by clicking ⊕ .

- **Language*** *(select)*:
  Select a language. Supported languages are:
  - **en**: English
  - **fr**: French
- **Display Name** *(string input)*:
  Enter a display name. This will be the localized name of the label.
- **Description** *(string input)*:
  Enter a description. This will be displayed when making a request with this label and when adding it to a profile.

You can delete 🗑 the localization.

5. Click on the create button to save.

You can update ✎ or delete 🗑 Labels.

CAUTION You won’t be able to delete a Label if it is referenced in any other configuration element.

13.2. HTTP Proxy

In this section you will be able to set up HTTP Proxies. HTTP Proxies may be used by Horizon to establish connection to various services.

13.2.1. How to configure an HTTP Proxy

1. Log in to Horizon Administration Interface.

2. Access HTTP Proxy from the drawer or card: **System › HTTP Proxies**.

3. Click on ⊕ .

4. Fill the mandatory fields.

- **Name*** *(string input)*:
  Enter a meaningful HTTP Proxy name.
- **Host*** *(string input)*:
  The Hostname or IP Address of the HTTP/HTTPS proxy to use.
- **Port*** *(int)*:
  The Port of the HTTP/HTTPS proxy to use.
- **Credentials** *(select)*:
  Select `Password` credentials containing basic authentication credentials for the proxy.

5. Click on the create button to save.

You can update ✎ or delete 🗑 the HTTP Proxy.

> **CAUTION** You won't be able to delete an HTTP Proxy if it is referenced in any other configuration element.

13.3. Grading Rules

The grading rules feature enhances the governance capabilities of Horizon, clearly displaying the quality of a certificate using different criteria. Currently, there is only one grading policy which is the Horizon grading policy designed by EverTrust experts using common reference documents.

The grading mechanism works as following:

1. Each rule is evaluated individually;
2. The score of each ruleset is calculated by adding the scores of each of its rules and dividing it by the max note for each ruleset, giving a score `s_i \in [-1,1]` for each ruleset;
3. The effective score for the grading policy is calculated through a weighted sum: `S = \sum_i w_i * s_i` with `w_i` being the weight of each ruleset;
4. The sum of the weights is calculated: `W = \sum_i w_i` ;

5. The score of the certificate for this grading policy is then calculated by dividing S by W: `cert\_score = \frac{S}{W} \in [-1,1]` , then the score is put back over 100 and the certificate grade is applied with the following scale:

13.3.1. Breakdown of the grading rules

ANSSI Cryptographic Content

The ANSSI Cryptographic Content Ruleset is created from the good practices advocated by the French ANSSI to ensure good cryptographic material when dealing with X509 certificates (based on the RGS). This ruleset has a maximum possible score of 70 and has a weight of 50 in the Horizon Grading Policy.

▼ *Details*

Rule	Score if satisfied	Score if not satisfied
Certificate Policy OID should be specified	10	0
Certificate should contain at least a CRLDP or an AIA OCSP URL	10	0
Certificate should contain the subject key identifier extension	10	0
Certificate subject and issuer should differ and authority key identifier should be defined	10	0
Certificate issuer should contain the country element ('C')	5	0
Certificate issuer should contain the organization element ('O')	5	0
Certificate issuer should contain the organizational unit element ('OU') or organisational identifier ('organizationIdentifier')	5	0
Certificate subject should contain the country element ('C')	5	0
Certificate subject should contain the organization element ('O')	5	0
Certificate subject should contain the organizational unit element ('OU') or organisational identifier ('organizationIdentifier')	5	0

CA/B Forum Ruleset

The CA/B Forum Ruleset contains good practices for certificates from the CA/B Forum advocations.
This ruleset has a maximum possible score of 40 and has a weight of 60 in the Horizon Grading Policy.

▼ *Details*

Rule	Score if satisfied	Score if not satisfied
CP OID extension is not empty	10	0
Character '_' is forbidden in SAN DNS *(penalty rule)*	0	-10
SAN DNS field must not end with '.' *(penalty rule)*	0	-10
Certificate lifetime is less than 397 days	10	0
Certificate serial number is longer than 8 bytes	10	0
SAN DNS field is not empty	10	0
Total	40	-20

NIST and ANSSI ECDSA Cryptographic Ruleset (Weight 100, Maximum score 35)

The NIST and ANSSI ECDSA Cryptographic Ruleset contains good practices when dealing with elliptic curves cryptography for the certificate’s private key.
This ruleset has a maximum possible score of 35 and has a weight of 100 in the Horizon Grading Policy.

▼ *Details*

Rule	Score if satisfied	Score if not satisfied
EC key algorithm should be P-256, P-384 or P-521	25	0
Signing hash algorithm should be SHA-256, SHA-384, SHA-512, SHA-3-256, SHA-3-384 or SHA-3-512	10	0
Total	35	0

EMail Certificate Ruleset

The EMail Certificate Ruleset contains good practices written by the EverTrust experts regarding the use of S/MIME certificates.
This ruleset has a maximum possible score of 20 and has a weight of 60 in the Horizon Grading Policy.

▼ *Details*

Rule	Score if satisfied	Score if not satisfied
Certificate with extended key usages 'emailProtection' should contain any of the following key usages: 'digitalSignature', 'nonRepudiation', 'keyEncipherment', 'dataEncipherment'	10	0
SAN Email (RFC822Name) field is not empty	10	0
Total	20	0

IETF PKIX Ruleset

The IETF PKIX Ruleset contains good practices from the IETF PKIX advocations.
This ruleset has a maximum possible score of 30 and has a weight of 100 in the Horizon Grading Policy.

▼ *Details*

Rule	Score if satisfied	Score if not satisfied
An entity certificate should not contain a pathlen	10	0
Issuer must not be empty	5	0
Subject must not be empty	5	0
Subject key identifier extension should not be empty	5	0
Certificate subject and issuer should differ and authority key identifier should be defined	5	0
If defined, AIA OCSP URL should use HTTP *(penalty rule)*	0	-10
If defined, CRLDP should use LDAP or HTTP *(penalty rule)*	0	-10
Non-CA certificate cannot be self-signed *(penalty rule)*	0	-30
The certificate is issued by an untrusted CA *(penalty rule)*	0	-15
Certificate KeyUsage cannot be empty *(penalty rule)*	0	-10
Total	30	-75

NIST and ANSSI RSA Cryptographic Ruleset

The NIST and ANSSI RSA Cryptographic Ruleset contains good practices when dealing with RSA cryptography for the certificate’s private key.
This ruleset has a maximum possible score of 35 and has a weight of 100 in the Horizon Grading Policy.

▼ *Details*

Rule	Score if satisfied	Score if not satisfied
RSA key size should be greater or equals to 2048 bits	10	0

Rule	**Score if satisfied**	**Score if not satisfied**
RSA key size should be greater or equals to 3072 bits	5	0
RSA key exponent should be greater than 2^16	10	0
Signing hash algorithm should be SHA-256, SHA-384, SHA-512, SHA-3-256, SHA-3-384 or SHA-3-512	10	0
RSA key size should not be less than 2048 bits *(penalty rule)*	0	-10
RSA key size must not be less than 1024 bits *(penalty rule)*	0	-10
Total	35	-20

TLS Certificate Ruleset

The TLS certificate ruleset contains good practices for certificates used to identify web servers.
This ruleset has a maximum possible score of 20 and has a weight of 60 in the Horizon Grading Policy.

▼ *Details*

Rule	**Score if satisfied**	**Score if not satisfied**
Certificate with extended key usages 'TLSServer' should contain key usage 'digitalSignature'	10	0
Certificate with extended key usage 'TLSServer' should not have a subject containing the following elements: 'givenname', 'surname' *(penalty rule)*	0	-5
SAN DNS field is not empty	10	0
Total	20	-5

13.3.2. Applying the grading policy

All certificates that are in Horizon can be graded using grading policies, whether they are discovered or fully managed by the product. If you want to add a grading policy to a profile, simply go to the profile settings then in the "Common configuration for profile" tab select the grading policies that will be used to grade certificates on this profile.
To remove a grading policy from a profile you just have to unselect it from the drop-down menu.

You can also grade discovered certificates: in the **Discovery** menu, click on the campaign that you want to apply the grading policies on and then select the grading policies that you want to apply from the drop-down menu.
Again, to remove a grading policy from a discovery campaign, just unselect it from the same drop-down menu.

13.3.3. Manually re-grading certificates

In case anything went wrong in the initial grading of certificates, or if you manually added a new grading policy to an existing profile and you want to manually re-evaluate a grading policy, follow these steps:

- **1.** Go to **System › Grading Rules**;
- **2.** Select the Grading Policy that you want to manually relaunch and click the ▷ .

All certificates concerned by this grading policy will now be re-graded.

13.4. Global configuration

These configurations handle various Horizon global parameters directly via the Web Interface.

13.4.1. Internal monitoring

This parameter configures the internal monitoring execution interval. Internal monitoring refers to an action that will check the expiration and usage status of credentials and license, and send the configured notifications if needed.

By default, this action will be executed every day at midnight UTC. The notifications will keep being sent each day for as long as an action is needed.

13.4.2. License configuration

The license configuration panel allows to configure Email, Groupware or REST notifications to be sent. These can be configured on:

- license expiration: using a notification on the `License Expiration` event and the `Delay before notification sending` field in the notification configuration, notifications configured here will be sent by the internal monitoring action.
- license usage: using the `usage threshold`, if this threshold is exceeded, notifications configured here will be sent by the internal monitoring action.

13.4.3. Interface configuration

An image can be defined here. It will be added on the Web interface in the header and in the login menu.

14. Common configuration elements

14.1. Cron Expression

Cron expressions are composed of 6 required fields and one optional field separated by white spaces. The fields are respectively described as follows:

Field Name	Allowed Values	Allowed Special Character
Seconds	0-59	- * /
Minutes	0-59	- * /
Hours	0-23	- * /
Day-of-month	1-31	- * ? / L W
Month	1-12 or JAN-DEC	- *
Day-of-Week	1-7 or SUN-SAT	- * ? / L #
Year (Optional)	empty, 1970-2199	- * /

Special characters

- * ("all values") - used to select all values within a field. For example, "*" in the minute field means *every minute*.
- **?** ("no specific value") - useful when you need to specify something in one of the two fields in which the character is allowed, but not the other. For example, if I want my trigger to fire on a particular day of the month (say, the 10th), but don't care what day of the week that happens to be, I would put "10" in the day-of-month field, and "?" in the day-of-week field. See the examples below for clarification.
- **-** - used to specify ranges. For example, "10-12" in the hour field means "the hours 10, 11 and 12".
- **,** - used to specify additional values. For example, "MON,WED,FRI" in the day-of-week field means "the days Monday, Wednesday, and Friday".
- / - used to specify increments. For example, "0/15" in the seconds field means "the seconds 0, 15, 30, and 45". And "5/15" in the seconds field means "the seconds 5, 20, 35, and 50". You can also specify '/' after the '*' character - in this case '*' is equivalent to having '0' before the '/'. '1/3' in the day-of-month field means "fire every 3 days starting on the first day of the month".
- **L** ("last") - has different meaning in each of the two fields it is allowed into. For example, the value "L" in the day-of-month field means "the last day of the month" - day 31 for January, day 28 for February on non-leap years. If used in the day-of-week field by itself, it simply means "7" or "SAT". But if used in the day-of-week field after another value, it means "the last xxx day of the month" - for example "6L" means "the last Friday of the month". You can also specify an offset from the last day of the month, such as "L-3" which would mean the third-to-last day of the calendar month. When using the 'L' option, it is important not to specify lists, or ranges of values, as you'll get confusing/unexpected results.
- **W** ("weekday") - used to specify the weekday (Monday-Friday) nearest the given day. As an example, if you were to specify "15W" as the value for the day-of-month field, the meaning is:

"the nearest weekday to the 15th of the month". So if the 15th is a Saturday, the trigger will fire on Friday the 14th. If the 15th is a Sunday, the trigger will fire on Monday the 16th. If the 15th is a Tuesday, then it will fire on Tuesday the 15th. However if you specify "1W" as the value for day-of-month, and the 1st is a Saturday, the trigger will fire on Monday the 3rd, as it will not 'jump' over the boundary of a month's days. The 'W' character can only be specified when the day-of-month is a single day, not a range or list of days.

- # - used to specify "the nth" XXX day of the month. For example, the value of "6#3" in the day-of-week field means "the third Friday of the month" (day 6 = Friday and "#3" = the 3rd one in the month). Other examples: "2#1" = the first Monday of the month and "4#5" = the fifth Wednesday of the month. Note that if you specify "#5" and there is not 5 of the given day-of-week in the month, then no firing will occur that month.

NOTE The 'L' and 'W' characters can also be combined in the day-of-month field to yield ' `LW` ', which translates to **"last weekday of the month"**.

14.2. Finite Duration

The format of a *Finite Duration* is `"<length><unit>"`, where:

- White space is allowed between the parts.
- Length is a positive integer without the "+" sign.
- Valid possible units are described in the below table:

Unit	Short name	Long names
DAYS	d	day days
HOURS	h	hour hours
MINUTES	m	min mins minute minutes
SECONDS	s	sec secs second seconds
MILLISECONDS	ms	milli millis millisecond milliseconds

For example, 10 seconds will be written as "10 s", "10s", "10 sec" or "10 seconds".

14.3. Regex

Regex are a powerful tool to match patterns in strings.Horizon is developed in Java, so regex must follow the Java regex engine syntax.

In Horizon, most regex input are used for validation of a field input.As such, they validate one line, and a good practice is enforced:

CAUTION Regex must start with `^` and end with `$`.

14.4. Dictionaries

Here is the list of available dictionary keys to use in computation rules, depending on the usage.

14.4.1. In notifications

Certificate dictionary

This dictionary is available for notifications on the following events:

- `on_enroll`
- `on_revoke`
- `on_update`
- `on_recover`
- `on_migrate`
- `on_expire`
- `on_renew`

Key	Description	Type	Available in Computation Rule
certificate.id	Horizon Id of the certificate	Single value	Yes
certificate.module	Module of the certificate	Single value	Yes
certificate.not_after	Expiration date of the certificate	Single value	Yes
certificate.not_before	Start date of the certificate	Single value	Yes
certificate.serial	Serial number of the certificate	Single value	Yes
certificate.thumbprint	Thumbprint of the certificate	Single value	Yes
certificate.public_key_thumbprint	Thumbprint of the public key of the certificate	Single value	Yes
certificate.revoked	true if the certificate is revoked, false otherwise	Single value	Yes
certificate.key_type	Key Type of the certificate	Single value	Yes

Key	Description	Type	Available in Computation Rule
certificate.signing_algorithm	Signing Algorithm of the certificate	Single value	Yes
certificate.holder_id	Holder Id of the certificate	Single value	Yes
certificate.friendly_name	Friendly name of the certificate	Single value	Yes
certificate.pem	PEM Encoded certificate	Single value	Yes
certificate.profile	The profile of the certificate	Single value	Yes
certificate.revocation_date	The revocation date of the certificate	Single value	Yes
certificate.revocation_reason	The revocation reason of the certificate	Single value	Yes
certificate.dn	The Distinguished Name of the certificate	Single value	No
certificate.sans	All the SANs of the certificate, in <type>: <value> comma separated format	Single value	No
certificate.extensions	All the extensions of the certificate, in <type>: <value> comma separated format	Single value	No
certificate.metadata	All the metadata of the certificate, in <type>: <value> comma separated format	Single value	No
certificate.labels	All the labels of the certificate, in <name>: <value> comma separated format	Single value	No
certificate.metadata.<metadata name>	The value of metadata `metadata name` defined in the certificate	Single value	Yes
certificate.subject	The values of the certificate subject	Subject dictionary	Yes
certificate.san	The values of the certificate sans	Sans dictionary	Yes

Key	Description	Type	Available in Computation Rule
certificate.extension	The values of the certificate extensions	Extensions dictionary	Yes
certificate.label	The values of the certificate label	Labels dictionary	Yes

Request dictionary

This dictionary is available for notifications on the following events:

- `on_submit_enroll`
- `on_cancel_enroll`
- `on_approve_enroll`
- `on_deny_enroll`
- `on_pending_enroll`
- `on_submit_revoke`
- `on_cancel_revoke`
- `on_approve_revoke`
- `on_deny_revoke`
- `on_pending_revoke`
- `on_submit_update`
- `on_cancel_update`
- `on_approve_update`
- `on_deny_update`
- `on_pending_update`
- `on_submit_recover`
- `on_cancel_recover`
- `on_approve_recover`
- `on_deny_recover`
- `on_pending_recover`
- `on_submit_migrate`
- `on_cancel_migrate`
- `on_approve_migrate`
- `on_deny_migrate`
- `on_pending_migrate`
- `on_submit_renew`

- `on_cancel_renew`
- `on_approve_renew`
- `on_deny_renew`
- `on_pending_renew`

Key	Description	Type	Available in Computation Rule
request.id	Horizon Id of the request	Single value	Yes
request.workflow	Workflow of the request	Single value	Yes
request.module	Module of the request	Single value	Yes
request.status	Status of the request	Single value	Yes
request.profile	Profile of the request	Single value	Yes
request.requester	Requester of the request	Single value	Yes
request.approver	Approver of the request	Single value	Yes
request.requester_comment	Comment of the requester	Single value	Yes
request.approver_comment	Comment of the approver	Single value	Yes
request.registration_date	Registration date of the request	Single value	Yes
request.last_modification_date	Last modification date of the request	Single value	Yes
request.password	PKCS#12 password or challenge value of the request	Single value	Yes
request.team	Team owning the request	Single value	Yes
request.my.url	Generates the link to access the request in the 'My Requests' drawer. Should be used after specifying the hostname without trailing slash: https://horizon.fr{{request.my.url}}	Single value	No

Key	Description	Type	Available in Computation Rule
request.manage.url	Generates the link to access the request in the 'Manage Requests' drawer. Should be used after specifying the hostname without trailing slash: https://horizon.fr{{request.my.url}}	Single value	No
request.dn	The Distinguished Name of the request	Single value	No
request.sans	All the SANs of the request, in <type>:<value> comma separated format	Single value	No
request.extensions	All the extensions of the request, in <type>:<value> comma separated format	Single value	No
request.metadata	All the metadata of the request, in <type>:<value> comma separated format	Single value	No
request.labels	All the labels of the request, in <name>:<value> comma separated format	Single value	No
request.subject	The values of the request subject	Subject dictionary	Yes
request.san	The values of the request sans	Sans dictionary	Yes
request.extension	The values of the request extensions	Extensions dictionary	Yes
request.label	The values of the request label	Labels dictionary	Yes
request.metadata.<metadata name>	The value of metadata `metadata name` defined in the request	Single value	Yes
request.certificate	The value of the certificate contained in the request	Certificate Dictionary	Yes

Previous Certificate dictionary

This dictionary is available for notifications on the following events:

- `on_renew`

Key	Description	Type	Available in Computation Rule
previous.certificate	The value of the certificate that is being renewed	Certificate dictionary	Yes

Credentials dictionary

This dictionary is available for notifications on the `on_credentials_expiration` event.

Key	Description	Type	Available in Computation Rule
credentials.name	Name of the credentials	Single value	Yes
credentials.description	Description of the credentials	Single value	Yes
credentials.type	Type of the credentials	Single value	Yes
credentials.expiration_date	Expiration date of the credentials	Single value	Yes

Profile dictionary

Key	Description	Type	Available in Computation Rule
profile.name	Technical name of the profile	Single value	Yes
profile.module	Module of the profile	Single value	Yes
profile.displaynames	Display names of the profile in <lang>:<value> comma separated format	Single value	No
profile.descriptions	Descriptions of the profile in <lang>:<value> comma separated format	Single value	No

Key	Description	Type	Available in Computation Rule
profile.<name>.displayname.<lang>	Display name of the profile in <lang> (two letter identifier) language	Single value	No
profile.<name>.description.<lang>	Description of the profile in <lang> (two letter identifier) language	Single value	No

License dictionary

This dictionary is available for notifications on the `on_license_expiration` and `on_license_usage` event.

Key	Description	Type	Available in Computation Rule
license.expiration_date	Expiration date of the license	Single value	Yes
license.used	Number of holders on the license (only available on `on_license_usage` event)	Single value	Yes
license.percent_used	Percent of the license used (only available on `on_license_usage` event)	Single value	Yes

Failed trigger dictionary

This dictionary is available for notifications on the `on_trigger_error` event.

Key	Description	Type
trigger.name	Name of the trigger	Single value
trigger.event	Event on which the trigger was run	Single value
trigger.lastExecutionDate	Last execution date of the trigger	Single value
trigger.status	Status of the trigger	Single value
trigger.retryable	true if the trigger is retryable, false otherwise	Single value
trigger.type	Type of the trigger	Single value

Key	Description	Type
trigger.retries	Number of remaining retries	Single value
trigger.nextExecutionDate	Date at which the trigger will be rerun	Single value
trigger.nextDelay	Delay between the current and next iteration	Single value
trigger.detail	Details about the failure	Single value

14.4.2. In profile

The following dictionaries are available in a certificate template in profile configuration, for auto validation and datasource flow configuration.

General

The dictionary keys listed here are available in all protocols.

NOTE All indexes start at 1.

Principal

This dictionary regroups the information of the user making the request, the 'principal'.

Key	Description	Type
principal.identifier	The identifier of the user	Single value
principal.team	The teams of the user	Multi valued
principal.team.<index>	The team at index `index`	Single value
principal.name	The name of the user	Single value
principal.mail	The email of the user	Single value
principal.provider.name	The name of the identity provider of the principal	Single value
principal.certificate.subject	The values of the principal certificate subject	Subject dictionary
principal.certificate.san	The values of the principal certificate sans	Sans dictionary
principal.certificate.extension	The values of the principal certificate extensions	Extensions dictionary

CSR

This dictionary regroups the information of the csr used for enrollment. It can be sent via a client (horizon-cli, estclient, sscep) or via web interfaces with WebRA protocol.

WARNING This only concerns decentralized enrollment.

Key	Description	Type
csr.subject	The values of the csr subject	Subject dictionary
csr.san	The values of the csr sans	Sans dictionary
csr.extension	The values of the csr extensions	Extensions dictionary

HTTP Request

This dictionary regroups the information of the http request that initiated the enrollment.

Key	Description	Type
http.request.ip	The IP from which the request originated	Single value
http.request.method	The HTTP method used by the request	Single value
http.request.path	The path requested	Single value
http.request.host	The host requested	Single value
http.request.header.<header name>	Value of the <header name> header	Multi value

WebRA

Enrollment request

Certificate fields can be filled by the user on Horizon interface. This information is available through the following dictionary.

Key	Description	Type
webra.enroll.subject	The values of the subject defined in the challenge request	Subject dictionary
webra.enroll.san	The values of the sans defined in the challenge request	Sans dictionary
webra.enroll.extension	The values of the extensions defined in the challenge request	Extensions dictionary
webra.enroll.label.<label name>	The value of label `label name` defined in the challenge request	Single value
webra.enroll.metadata.<metadata name>	The value of metadata `metadata name` defined in the challenge request	Single value

Key	Description	Type
webra.enroll.mail	The value of the contact email defined in the challenge request	Single value
webra.enroll.owner	The value of the owner defined in the challenge request	Single value
webra.enroll.team	The value of the team defined in the challenge request	Single value

EST

Enrollment request

In case of a prevalidated enroll, certificate fields can be filled by the user on Horizon interface. This information is available through the following dictionary.

Key	Description	Type
est.enroll.subject	The values of the subject defined in the challenge request	Subject dictionary
est.enroll.san	The values of the sans defined in the challenge request	Sans dictionary
est.enroll.extension	The values of the extensions defined in the challenge request	Extensions dictionary
est.enroll.label.<label name>	The value of label `label name` defined in the challenge request	Single value
est.enroll.metadata.<metadata name>	The value of metadata `metadata name` defined in the challenge request	Single value
est.enroll.mail	The value of the contact email defined in the challenge request	Single value
est.enroll.owner	The value of the owner defined in the challenge request	Single value
est.enroll.team	The value of the team defined in the challenge request	Single value

Url passed parameters

Horizon allows the use of url parameters to pass certificate metadata info. These are notably used by the horizon-cli client.

Key	Description	Type
url.enroll.label.<label name>	The value of label `label name` passed in the url	Single value

Key	Description	Type
url.enroll.metadata.<metadata name>	The value of metadata `metadata name` passed in the url	Single value
url.enroll.mail	The value of the contact email passed in the url	Single value
url.enroll.owner	The value of the owner passed in the url	Single value
url.enroll.team	The value of the team passed in the url	Single value

SCEP

Enrollment request

In case of a prevalidated enroll, certificate fields can be filled by the user on Horizon interface. This information is available through the following dictionary.

Key	Description	Type
scep.enroll.subject	The values of the subject defined in the challenge request	Subject dictionary
scep.enroll.san	The values of the sans defined in the challenge request	Sans dictionary
scep.enroll.extension	The values of the extensions defined in the challenge request	Extensions dictionary
scep.enroll.label.<label name>	The value of label `label name` defined in the challenge request	Single value
scep.enroll.metadata.<metadata name>	The value of metadata `metadata name` defined in the challenge request	Single value
scep.enroll.mail	The value of the contact email defined in the challenge request	Single value
scep.enroll.owner	The value of the owner defined in the challenge request	Single value
scep.enroll.team	The value of the team defined in the challenge request	Single value

Url passed parameters

Horizon allows the use of url parameters to pass certificate metadata info. These are notably used by the horizon-cli client.

Key	Description	Type
url.enroll.label.<label name>	The value of label `label name` passed in the url	Single value
url.enroll.metadata.<metadata name>	The value of metadata `metadata name` passed in the url	Single value
url.enroll.mail	The value of the contact email passed in the url	Single value
url.enroll.owner	The value of the owner passed in the url	Single value
url.enroll.team	The value of the team passed in the url	Single value

ACME

Order

This dictionary regroups the information of the acme order used for enrollment.

Key	Description	Type
acme.order.initialip	The initial IP of the acme order	Single value
acme.order.label.<label name>	The value of label `label name`	Single value
acme.order.metadata.<metadata name>	The value of metadata `metadata name`	Single value
acme.order.mail	The value of the contact email of the acme order	Single value
acme.order.owner	The value of the owner of the acme order	Single value
acme.order.team	The value of the team of the acme order	Single value

Account

This dictionary regroups the information of the acme account used for enrollment.

Key	Description	Type
acme.account.initialip	The initial IP of the acme account	Single value
acme.account.contact.<index>	The value of contact email address of the account at index `index`	Single value

CRMP

Enrollment request

Certificate fields can be filled by the user on CMS interface. This information is available through the following dictionary.

Key	Description	Type
crmp.enroll.subject	The values of the subject defined in the challenge request	Subject dictionary
crmp.enroll.san	The values of the sans defined in the challenge request	Sans dictionary
crmp.enroll.extension	The values of the extensions defined in the challenge request	Extensions dictionary
crmp.enroll.label.<label name>	The value of label `label name` defined in the challenge request	Single value
crmp.enroll.metadata.<metadata name>	The value of metadata `metadata name` defined in the challenge request	Single value
crmp.enroll.mail	The value of the contact email defined in the challenge request	Single value
crmp.enroll.owner	The value of the owner defined in the challenge request	Single value
crmp.enroll.team	The value of the team defined in the challenge request	Single value

WCCE

Caller identity

The information of the caller identity in a WCCE enroll.

Key	Description	Type
calleridentity.dn	The dn of the caller identity	Single value
calleridentity.subject	The dn of the caller identity, splitted in adressable form	Subject dictionary
calleridentity.cn	The cn of the caller identity	Single value
calleridentity.msguid	The guid of the caller identity	Single value
calleridentity.msupn	The upn of the caller identity	Single value
calleridentity.c	The country of the caller identity	Single value

Key	Description	Type
calleridentity.company	The company of the caller identity	Single value
calleridentity.department	The department of the caller identity	Single value
calleridentity.description	The description of the caller identity	Single value
calleridentity.displayname	The display name of the caller identity	Single value
calleridentity.dnshostname	The dns host name of the caller identity	Single value
calleridentity.employeeid	The employee id of the caller identity	Single value
calleridentity.employeenumber	The employee number of the caller identity	Single value
calleridentity.mail	The email of the caller identity	Single value
calleridentity.o	The organization of the caller identity	Single value
calleridentity.ou	The OU of the caller identity	Single value
calleridentity.samaccountname	The sam account name of the caller identity	Single value
calleridentity.serialnumber	The serial number of the caller identity	Single value
calleridentity.sn	The sn of the caller identity	Single value
calleridentity.title	The title of the caller identity	Single value
calleridentity.uid	The uid of the caller identity	Single value
calleridentity.sid	The sid of the caller identity	Single value

14.4.3. Sub dictionaries

These dictionary cannot be used alone but can be completed with one of the other ones. For example, a valid key is:

```
principal.certificate.subject.cn.1
```

Subject dictionary

Key	Description	Type
subject.<dn field type>	All values of subject field of type `dn field type`	Multi valued
subject.<dn field type>.<index>	Value of subject field of type `dn field type` at index `index`	Single value

NOTE The valid dn field types are: cn, uid, serialnumber, surname, givenname, unstructuredaddress, unstructuredname, e, ou, organizationidentifier, uniqueidentifier, street, st, l, o, c, description, dc.

Sans dictionary

Key	Description	Type
san.<san field type>	All values of san fields of type `san field type`	Multi valued
san.<san field type>.<index>	Value of subject field of type `san field type` at index `index`	Single value

NOTE The valid san field types are: rfc822name, dnsname, uri, ipaddress, othername_upn, othername_guid.

Extensions dictionary

Key	Description	Type
extension.<extension type>	Value of extension of type `extension type`	Single value

NOTE The valid extension types are: ms_sid, ms_template.

Labels dictionary

Key	Description	Type	Available in Computation Rule
label.<name>	Value of the <name> label	Single value	Yes
label.<name>.displaynames	Display names of the label in <lang>: <value> comma separated format	Single value	No
label.<name>.descriptions	Descriptions of the label in <lang>: <value> comma separated format	Single value	No

Key	Description	Type	Available in Computation Rule
label.<name>.displayname.<lang>	Display name of the label in <lang> (two letter identifier) language	Single value	No
label.<name>.description.<lang>	Description of the label in <lang> (two letter identifier) language	Single value	No

NOTE The valid extension types are: ms_sid, ms_template.

14.5. Computation rule

Computation Rules are expressions that describe operations to apply to dictionary keys. These keys can come from diverse data sources such as a certification request or a user entry. The available operations and their usage are detailed in this part.

14.5.1. Example

Let's start by an example:

My CSR contains a DNSNAME subject alternate name with the following value:

```
host.evertrust.fr
```

I want my final certificate to have 2 SANs, this value and its short name: "host".

In order to do that, in **Profile › Certificate Template › Subject Alternate Names** , I add a DNSNAME SAN with the following computation rule:

```
[{{csr.san.dnsname.1}}, Extract({{csr.san.dnsname.1}}, "(.*?)\.", 1)]
```

This will output, in my final certificate, two SANs with values:

```
host.evertrust.fr, host
```

To explain this result, the value "host.evertrust.fr" was retrieved by choosing the first DNSNAME SAN of the CSR: `{{csr.san.dnsname.1}}`. The function `Extract` extracted the first catching group from the regex `(.*?)\.`, resulting in the "host" value.

The computation rule language has a lot more possible operations, allowing complex use cases to become reality.

14.5.2. Dictionary keys

Dictionary keys are a way to name the information from the available sources. For instance, for a webra enroll, the available sources are the given csr, the webra enroll form data and the principal information if it is authenticated. The full list of available dictionary keys is available on the dictionary page.

Enrollment

A key can reference a single element or a list of elements. It is separated in three main parts: the source of data (csr, webra enroll data form), the section of the data, and an optional number

For example, the following is a valid key with these 3 parts:

`{{csr.subject.cn.1}}`

The csr is the data source, the subject.cn the requested information and the 1 is the index. It allows to retrieve the first, common name from the subject, from the CSR.

Without an index, the key is still valid, but it will output all the corresponding values. For example

`[[csr.subject.ou]]`

This retrieves all the ou from the subject, from the CSR.

WARNING When a key is expected to output a single value it should be written as a single dictionary key, and one outputing a list of values as a multi dictionary key, otherwise it will be none.

14.5.3. Basic expressions

Basic string expressions
The following expressions are evaluated as a string or None.

Expression Name	Syntax	Allowed Values	Description	Example
Single dictionary key	{{<key>}}	key: a-zA-A-._	This retrieves a key value from the dictionary, none if it does not exist	{{csr.subject.cn.1}}
Number	<number>	number: -\d+	This will output the given number	-4
Literal	"<literal>"	literal: any string	This will output the given literal	"iAmAString"
Null	NULL	NULL	This will output None	NULL
Now	NOW	NOW	This will output the current instant	NOW

Basic list expressions

The following expressions are evaluated as a list of string or None.

Expression Name	Syntax	Allowed Values	Description	Example
Multi dictionary key	[[<key>]]	key: a-zA-A-._	This retrieves all values that start with key from the dictionary	[[csr.subject.cn]]
Array	[<simpleExpression>, … <simpleExpression>]	simpleExpression: any expression that will be evaluated to a single element	This will output a multi expression composed of all inserted simple expressions	["iAmAString", {{csr.san.dnsname.1}}]

Quick reference

NOTE Function names are not case sensitive but keys are

Function Name	Syntax
Upper	Upper(`expression`:<expression>)
Lower	Lower(`expression`:<expression>)
Trim	Trim(`expression`: <expression>)
Substr	Substr(`expression`: <expression>, `start`: <number>)
Substr	Substr(`expression`: <expression>, `start`: <number>, `end`: <number>)
Concat	Concat(`expression`: <expression>, … <expression>)
Extract	Extract(`expression`: <expression>, `regex`: <literal>)
Extract	Extract(`expression`: <expression>, `regex`: <literal>, `group`: <number>)
Replace	Replace(`expression`: <expression>, `regex`: <literal>, `replacement`: <expression>)
OrElse	OrElse(`expression`: <expression>, … <expression>)
Match	Match(`expression`: <simpleExpression>, `regex`: <literal>)
DateTimeFormat	DateTimeFormat(`expression`: <simpleExpression>, `format`: <literal>)
Get	Get(`expression`: <multiExpression>, `index`: <number>)

Function Name	Syntax
First	First(expression: <multiExpression>)
Last	Last(expression: <multiExpression>)
Filter	Filter(expression: <multiExpression>, regex: <literal>)
Slice	Slice(expression: <multiExpression>, start: <number>)
Slice	Slice(expression: <multiExpression>, start: <number>, end: <number>)
Join	Join(expression: <multiExpression>, separator: <literal>)
Sort	Sort(expression: <multiExpression>)
ShortenDNS	ShortenDNS(expression: <singleExpression>)
DomainDNS	DomainDNS(expression: <singleExpression>)
EmailUser	EmailUser(expression: <singleExpression>)
EmailDomain	EmailDomain(expression: <singleExpression>)
SamAccountNameUser	SamAccountNameUser(expression: <singleExpression>)
SamAccountNameDomain	SamAccountNameDomain(expression: <singleExpression>)

14.5.4. Any expression functions

Upper

```
Upper(expression:<expression>)
```

This outputs the result evaluated from `expression` with only upper case characters and None if no value was evaluated

```
Upper("string") => "STRING"
Upper(["string1", "string2"]) => ["STRING1", "STRING2"]
```

Lower

```
Lower(expression:<expression>)
```

This outputs the result evaluated from `expression` with only lower case characters and None if no value was evaluated

```
Lower("STRING") => "string"
Lower(["STRING1", "STRING2"]) => ["string1", "string2"]
```

Trim

```
Trim(expression:<expression>)
```

This outputs the trimmed result evaluated from `expression` and None if no value was evaluated

```
Trim(" STRING") => "STRING"
Trim(["string1 ", "  string2  "]) => ["string1", "string2"]
```

Substr

```
Substr(expression: <expression>, start: <number>)
```

This outputs the substring from index `start` to the end of the string evaluated from `expression` and None if no value was evaluated or the result of substring is empty. `start` can be negative and it will be computed from end of string.

```
Substr("STRING", 2) => "TRING"
Substr(["string", "longerString", "s"], -2) => ["ng", "ng", "s"]
Substr("tooShort", 15) => None
```

Substr

```
Substr(expression: <expression>, start: <number>, end: <number>)
```

This outputs the substring from index `start` to `end` of the string evaluated from `expression` and None if no value was evaluated or the result of substring is empty. `start` and `end` can be negative and it will be computed from end of string.

```
Substr("STRING", 2, 4)  => "TRI"
Substr(["string", "longerString", "s"], 2, -2) => ["tri", "ongerStri"]
Substr("tooShort", -2, 4) => None
```

Concat

```
Concat(expression: <expression>, ...<expression>)
```

This outputs the concatenation of evaluated expressions: if they are all simple expression, a string concatenation will take place, otherwise an array with all the values will be evaluated. If the final result is empty, None will be returned.

```
Concat("start", " middle ", "end") => "start middle end"
Concat(["string1", "string2", "string3"], "string4") => ["string1", "string2",
"string3", "string4"]
```

Extract

```
Extract(expression: <expression>, regex: <literal>)
```

This extracts from the evaluated `expression` string(s) the part that matches the `regex`

```
Extract("abcd@domain.com", ".*@") => "abcd@"
Extract(["string1", "string2", "string3"], "\d") => ["1", "2", "3"]
```

Extract

```
Extract(expression: <expression>, regex: <literal>, group: <number>)
```

This extracts from the evaluated `expression` string(s) the group at index `group` that matches the `regex`

```
Extract("abcd@domain.com", "(.*)@", 1) => "abcd"
Extract(["string1", "string2", "string3"], "(.*)\d", 1) => ["string", "string",
"string"]
```

Replace

```
Replace(expression: <expression>, regex: <literal>, replacement: <expression>)
```

This replaces parts of the evaluated `expression` string(s) that matches the `regex` with the evaluated `replacement`. If `replacement` is None, values will be replaced by an empty string.

```
Replace("abcdATdomain.com", "AT", "@") => "abcd@domain.com"
Replace(["string1", "string2", "string3"], "\d", CONCAT("This", " was ", " a number"))
=> ["stringThis was a number", "stringThis was a number", "stringThis was a number"]
```

OrElse

```
OrElse(expression: <expression>, ...<expression>)
```

This outputs the first non None result of the given expressions, or None if they are all None

```
OrElse({{not.a.value}}, "abcd@domain.com") => "abcd@domain.com"
OrElse([[no.values]], "value") => ["value"]
OrElse([[no.values]], {{not.a.value}}) => None
```

14.5.5. String functions

NOTE The following functions output a string or None.

Match

```
Match(expression: <simpleExpression>, regex: <literal>)
```

This outputs the expression if it matches the regex, otherwise None

```
Match("abcd", "[a-z]+") => "abcd"
Match("abcd", "\d+") => None
```

DateTimeFormat

```
DateTimeFormat(expression: <simpleExpression>, format: <literal>)
```

This outputs the expression formatted as format. If expression is not a date, no formatting takes place. Available formats are:

- Custom format in Java DateFormatter syntax
- MILLIS
- BASIC_ISO_DATE
- ISO_LOCAL_DATE
- ISO_OFFSET_DATE
- ISO_DATE
- ISO_LOCAL_TIME
- ISO_OFFSET_TIME
- ISO_TIME

- ISO_LOCAL_DATE_TIME
- ISO_ZONED_DATE_TIME
- ISO_DATE_TIME
- ISO_ORDINAL_DATE
- ISO_WEEK_DATE
- ISO_INSTANT
- RFC_1123_DATE_TIME

  ```
  DateTimeFormat(NOW, "MILLIS") => "1709290260764"
  DateTimeFormat(NOW, "hh:mm:ss") => "10:54:57"
  ```

Get

```
Get(expression: <multiExpression>, index: <number>)
```

This outputs the string at `index` index in the `expression` list, and None if the index does not exist. The index can be negative to get from the end of the list.

```
Get(["string1", "string2", "string3", "string4"], -2) => "string3"
Get(["string1", "string2"], 3) => None
```

First

```
First(expression: <multiExpression>)
```

This outputs the first string of the `expression` list, and None if it does not exist. The index can be negative to get from the end of the list.

```
First(["string1", "string2", "string3", "string4"]) => "string1"
First([[no.values]]) => None
```

Last

```
Last(expression: <multiExpression>)
```

This outputs the last string of the `expression` list, and None if it does not exist. The index can be negative to get from the end of the list.

```
Last(["string1", "string2", "string3", "string4"]) => "string4"
```

```
Last([[no.values]]) => None
```

14.5.6. List of string functions

NOTE The following functions output a list of string or None.

Filter

```
Filter(expression: <multiExpression>, regex: <literal>)
```

This outputs a list of string from `expression` that matches the `regex`, None if none matches

```
Filter(["string1", "string2", "match"], "[a-z]+") => ["match"]
Filter(["string1", "string2"], "[a-z]+") => None
```

Slice

```
Slice(expression: <multiExpression>, start: <number>)
```

This outputs the slice of the `expression` list between `start` index and its end, or None if the slice is invalid. The index can be negative to get from the end of the list.

```
Slice(["string1", "string2", "string3", "string4"], -2) => ["string3", "string4"]
Slice(["string1", "string2"], 3) => None
```

Slice

```
Slice(expression: <multiExpression>, start: <number>, end: <number>)
```

This outputs the slice of the `expression` list between `start` and `end` index, or None if the slice is invalid. The index can be negative to get from the end of the list.

```
Slice(["string1", "string2", "string3", "string4"], 1, 3) => ["string1", "string2",
"string3"]
Slice(["string1", "string2"], 3) => None
```

Join

```
Join(`expression`: <multiExpression>, `separator`: <literal>)
```

This outputs the values of the `expression` joined with the `separator` string.

```
Join(["string1", "string2"], ".") => "string1.string2"
```

Sort

```
Sort(`expression`: <multiExpression>)
```

This outputs the values of the `expression` sorted in alphabetical order.

```
Sort(["b", "a"]) => ["a","b"]
```

ShortenDNS

```
ShortenDNS(`expression`: <singleExpression>)
```

This retrieves the first element of the DNS FQDN from `expression`.

```
ShortenDNS("subdomain.domain.com") => "subdomain"
```

DomainDNS

```
DomainDNS(`expression`: <singleExpression>)
```

This retrieves the domain FQDN of the DNS FQDN from `expression`.

```
DomainDNS("subdomain.domain.com") => "domain.com"
```

EmailUser

```
EmailUser(`expression`: <singleExpression>)
```

This retrieves the part before the @ from `expression`.

```
EmailUser("user@domain.com") => "user"
```

EmailDomain

```
EmailDomain(`expression`: <singleExpression>)
```

This retrieves the part after the @ from `expression`.

```
EmailDomain("user@domain.com") => "domain.com"
```

SamAccountNameUser

```
SamAccountNameUser(`expression`: <singleExpression>)
```

This retrieves the user part (before the \) from a SamAccountName in `expression`.

```
SamAccountNameUser("DOMAIN\User") => "User"
```

SamAccountNameDomain

```
SamAccountNameDomain(`expression`: <singleExpression>)
```

This retrieves the domain part (before the \) from a SamAccountName in `expression`.

```
SamAccountNameDomain("DOMAIN\User") => "DOMAIN"
```

14.6. Template Strings

Template Strings are augmented strings. They can be used as normal text but can also be augmented:

14.6.1. Using dictionary values

Using the following format, a dictionary key will be interpreted to its value when sending the notification:

```
{{<dictionary key>}}
```

Example:

```
I am enrolling on {{ca.name}}
```

Depending on the notification event, values will be added to context to be interpreted.

NOTE If the value is not available in the context, the dictionary value will not be replaced

14.6.2. Using computation rules

Using the following format, a computation rule will be interpreted to its value when sending the notification:

```
{{<computation rule>}}
```

Example:

```
I am enrolling on {{ Lower({{ca.name}}) }}
```

Depending on the notification event, values will be added to context to be interpreted in the computation rule.

NOTE If the computation rule result is None, an empty string will be displayed. If it is an array, it will be in a comma separated string

15. Reports

A report is a CSV file sent in a scheduled email. The CSV content is managed by:

- *HCQL query (certificates), HRQL query (requests)*
- *CSV fields shown*

15.1. Prerequisites

You may need Teams.

15.2. How to configure Reports

1. Log in to Horizon Administration Interface.

2. Access Reports from the drawer or card: **Reports**.

3. Click on ⊕ .

4. Fill in the mandatory fields.

15.2.1. Details

- **Enable** *(boolean)*:
  Tells whether the reporting task should be enabled. Set by default at true.
- **Name*** *(string input)*:
  Enter a meaningful report name. It must be unique.
- **Cron scheduling expression in Quartz format*** *(cron expression)*:
  Enter a Cron scheduling expression (in Quartz format). The default expression is built to run the task every hour.

15.2.2. Recipients

Click on ⊕ to add a recipient.

You can either target:

- A static (recipient): you will need to set a valid email address.
- A team contact: you will need to select one of the enabled teams.
- A team manager: you will need to select one of the enabled teams.

15.2.3. Email

- **From*** *(string input)*:

Enter the email address that will appear in the "From" field of the email.

- **Subject*** *(string input)*:
  Enter the subject of the email.
- **Body** *(string input)*:
  Enter the body of the email.
- **CSV file name** *(string input)*:
  Enter the name that will be given to the attached csv file.
- **Is HTML** *(boolean)*: (boolean):
  Sets whether the email body contains HTML code (true) or plain text (false). The default value is set to false.

15.2.4. HQL

- **HQL Type*** *(select)*:
  Either chose Certificate or Request. It will define the HQL Query type to set and the enabled CSV fields.
- **Query** *(string input or select)*:
  HCQL (Certificate) or HRQL (Request). You can select one of your saved queries.

15.2.5. CSV

You can select which fields will appear on the CSV file.

5. Click on the save button.

You can run ▷ , edit ✎ or delete 🗑 the report .

16. Overridable configuration parameters

This page presents the overridable parameters from the Horizon configuration.

16.1. Overriding the parameters

To override one of these parameters, simply:

1. Access the EVERTRUST Horizon server through SSH using an account with administrative privileges;

2. With an editor like vi, open the **/opt/horizon/etc/conf.d/horizon-extra.conf** file;

3. For each parameter you wish to override, create a newline and use the following syntax:

```
<parameter>=<value>
```

As an example, if you want to modify the file extension that DER certificates will have when sent as email attachments and set it to CRT, you need to add:

```
horizon.notification.mail.attachment.extension.der="crt"
```

4. Save your modifications and restart the Horizon service:

```
$ systemctl restart horizon
```

NOTE One added line means one modified option, you need to add as many lines at the end of the file as there are values that you want to override.

You'll find below an exhaustive list of overridable parameters.

16.2. Parameters

16.2.1. ACME parameters

Parameter	Default value	Description
horizon.acme.max.timeout	"5 minutes"	Maximum configurable timeout in the ACME profiles
horizon.acme.max.retry.delay	"1 hour"	Maximum configurable delay in the ACME profiles
horizon.acme.max.retry.count	15	Maximum configurable retry count in the ACME profiles

Parameter	Default value	Description
horizon.acme.order.updater.prefix	"Actor OrderUpdater#"	Name of the ACME order updater in the technical logs
horizon.acme.response.verifier.prefix	"Actor ResponseVerifier#"	Name of the ACME response verifier in the technical logs
horizon.acme.response.verifier.worker	5	Number of instances that will be started for each Horizon node to perform the ACME validation
horizon.acme.order.updater.worker	5	Number of instances that will be started for each Horizon node to perform the ACME validation
horizon.acme.authorization.methods	["http-01", "dns-01", "tls-alpn-01"]	Allowed methods that can be used to complete an ACME challenge
horizon.acme.order.ttl	"1 minute"	TTL of an ACME order
horizon.acme.challenge.entropy	32	Size of the ACME authorization value (challenge size)
horizon.acme.http.json.prettify	true	Whether the ACME responses should be JSON-pretty formatted
horizon.acme.behavior.emulate.boulder	true	Defines whether Horizon should behave like the Boulder ACME implementation (if set to false, Horizon will strictly follow the RFC). Only applicable if horizon.acme.http.json.prettify is set to "true".
horizon.acme.url.scheme	https	Protocol to use to calculate the ACME base URL if there isn’t any X-Forwarded-Proto nor X-Forwarded-Host in the header of the request
horizon.acme.url.prefix	"/acme"	Prefix used to calculate the ACME base URL
horizon.acme.behavior.enable.post-as-get.legacy	true	Whether the ACME API can be used with GET requests instead of POST ones

16.2.2. Bootstrap parameters

Parameter	Default value	Description
horizon.bootstrap.timeout	"1 minute"	Duration after which the bootstrap of Horizon times out
horizon.bootstrap.administrator.name	"administrator"	Default administrator account name

Parameter	Default value	Description
horizon.bootstrap.administrator.display.name	"Horizon Administrator"	Default administrator account display name
horizon.bootstrap.local.identity.provider	"local"	Default administrator account identity provider to use
horizon.bootstrap.administrator.password.path	"var/run/adminPassword"	Relative path of the file where the initial admin password should be stored into (starting at /opt/horizon/)
horizon.bootstrap.administrator.password.length	24	Length (in bytes) of the initial admin password

16.2.3. Certificate Authorities parameters

Parameter	Default value	Description
horizon.ca.manager.cache-idletime	"1 minute"	Default idle time after which a CA crl is removed from cache
horizon.ca.manager.timeout	"1 minute"	Duration that the CA manager actor will wait to retrieve information about certificates (trust status, trust chain, …)
horizon.ca.maximum.timeout	"5 minutes"	Maximum configurable timeout for CRL/OCSP request for a CA
horizon.ca.maximum.refresh	"7 days"	Maximum configurable refresh for a CA’s CRL
horizon.crl.updater.parallelism	5	Number of certificates processed in parallel when Horizon synchronizes its database with the CRL

16.2.4. Default PKI queue parameters

Parameter	Default value	Description
horizon.pki.manager.default.parallelism	5	Number of parallel certificate requests (enrollment, revocation…) on the default queue
horizon.pki.manager.default.size	100	Number of certificate requests (enrollment, revocation) that can be queued on the default queue

16.2.5. Events parameters

Parameter	Default value	Description
horizon.discovery.event.ttl	(No default value)	Time to live of the discovery events. If not set, events never expire
horizon.event.chainsign	true	Specify whether to chain and sign the Horizon events to ensure they haven’t been tampered with
horizon.event.seal.algorithm	"HS_512"	Algorithm to use to hash the signature of the events in Horizon (other possible values are "HS_384" and "HS_256")

16.2.6. HTTP parameters

Parameter	Default value	Description
horizon.http.header.enforce_connection_close	true	Defines whether HTTP connections should remain open
horizon.http.header.real-ip	"X-Real-IP"	Name of the HTTP header to use as Real IP
horizon.http.header.scheme	"X-Forwarded-Proto"	Name of the HTTP header containing the scheme requested - used for ACME
horizon.http.header.host	"X-Forwarded-Host"	Name of the HTTP header containing the host requested - used for ACME
play.http.session.maxAge	Duration after which the authentication session expires	"15 minutes"
horizon.security.http.headers.xapi.key	"X-API-KEY"	Name of the HTTP header to use as XAPI-KEY
horizon.security.http.headers.xapi.id	"X-API-ID"	Name of the HTTP header to use as XAPI-ID
horizon.security.http.headers.xapi.idprov	"X-API-IDPROV"	Header used to store the identity provider that will be used to validate the authentication

16.2.7. Email notifications parameters

Parameter	Default value	Description
horizon.notification.mail.attachment.extension.der	"der"	File extension that DER certificates sent as email attachments (through the notifications feature) will be given
horizon.notification.mail.attachment.extension.p7b	"p7b"	File extension that PKCS#7 certificates sent as email attachments (through the notifications feature) will be given
horizon.notification.mail.attachment.extension.pem	"pem"	File extension that PEM certificates sent as email attachments (through the notifications feature) will be given

16.2.8. Nonce parameters

Parameter	Default value	Description
horizon.[nonce type].nonce.ttl	"5 seconds"	Duration for which a nonce stays in Horizon before being removed ([nonce type] can be replaced by either automation, acme, openid or request)
horizon.[nonce type].nonce.size	32	Size (in bytes) of the challenge stored in the nonce ([nonce type] can be replaced by either automation, acme, openid or request)
horizon.security.[nonce type]_certificate_pop.iat.future	"5 seconds"	Difference of time allowed between the "Issued At Time" and the validation time (or the server time) (in the future only) ([nonce type] can be replaced by either automation or request)
horizon.security.[nonce type]_certificate_pop.iat.past	"5 seconds"	Difference of time allowed between the "Issued At Time" and the validation time (or the server time) (in the past only) ([nonce type] can be replaced by either automation or request)
horizon.security.[nonce type]_certificate_pop.iat.clock-skew	"30 seconds"	Difference of time allowed between the client time and the server time ([nonce type] can be replaced by either automation or request)
horizon.security.identity.provider.openid.nonce.size	32	Size of the nonce value used for the OpenID authentication
horizon.security.identity.provider.openid.nonce.ttl	"60 seconds"	"Time to live" of the OpenID authentication nonce
horizon.automation.nonce.size	32	Size of the nonce value used for the JWT authentication token
horizon.automation.nonce.ttl	"5 seconds"	Time to live of the nonce used to validate the JWT authentication token

16.2.9. Password reset parameters

Parameter	Default value	Description
horizon.security.local.identity.password.reset.duration	"2 minutes"	Time to live of a password reset request (from the login prompt)

16.2.10. Requests/Automation parameters

Parameter	Default value	Description
horizon.request.grace_period.[module].[request workflow]	"30 days"	Time to live of an expired or completed request. Module can be est, scep or webra and the request workflow can be enroll, revoke, update, recover, migrate or renew.
horizon.request.default_duration.[module].[request workflow]	"7 days"	Duration that a request can stay in either pending or approved state before expiring. Module can be est, scep or webra and the request workflow can be enroll, revoke, update, recover, migrate or renew.
horizon.automation.policy.default.keyType	"rsa-2048"	Default key type sent to the automation client on ACME, EST and SCEP

16.2.11. Search parameters

Parameter	Default value	Description
horizon.certificate.search.csv.delimiter	';'	The CSV delimiter to use when exporting an HCQL query result to a CSV file
horizon.certificate.search.item.separator	'\n'	The CSV item separator to use when exporting an HCQL query result to a CSV file
horizon.certificate.search.item.attribute.separator	':'	The CSV item attribute separator to use when exporting an HCQL query result to a CSV file
horizon.security.principal.search.page.size	50	How many elements to retrieve in a security principals search query if no pageSize has been specified
horizon.security.principal.search.page.max-size	(No default value)	How big can the pageSize parameter be in a security principals search query (Must be a positive integer)
horizon.discovery.event.search.page.default_size	50	How many elements to retrieve in a discovery events search query if no pageSize has been specified

Parameter	Default value	Description
horizon.discovery.event.search.page.max_size	(No default value)	How big can the pageSize parameter be in a discovery events search query (Must be a positive integer)
horizon.discovery.event.search.csv.delimiter	';'	The CSV delimiter to use when exporting a discovery events query result to a CSV file
horizon.request.search.page.default_size	50	How many elements to retrieve in a request search query if no pageSize has been specified
horizon.request.search.page.max_size	(No default value)	How big can the pageSize parameter be in a request search query (Must be a positive integer)
horizon.request.search.csv.delimiter	';'	The CSV delimiter to use when exporting an HRQL query result to a CSV file
horizon.event.search.page.default_size	50	How many elements to retrieve in an event search query if no pageSize has been specified
horizon.event.search.page.max_size	(No default value)	How big can the pageSize parameter be in an event search query (Must be a positive integer)

16.2.12. Trigger parameters

Parameter	Default value	Description
horizon.trigger.retry.initial_delay	"5 minutes"	How long must a trigger that fails for the first time wait before retrying
horizon.trigger.max.retry	15	Maximum amount of failed attempts that a trigger can have before canceling

16.2.13. Internal Horizon parameters

CAUTION These parameters are used internally in Horizon. Please be very careful when overriding them as it may prevent the product from working correctly.

Parameter	Default value	Description
horizon.security.trustmanager.enforce-serverauth	false	If set to true, enforces the use of the serverAuth EKU in the server authentication certificates (when Horizon accesses a service through TLS)
horizon.ca.maximum.timeout	"5 minutes"	Maximum configurable timeout duration on certificate authorities

Parameter	Default value	Description
horizon.ca.maximum.refresh	"7 days"	Maximum configurable CRL refresh period on certificate authorities
horizon.pki-connector.max.timeout	"5 minutes"	Maximum configurable timeout on the PKI connectors
horizon.trigger.manager.interval	"5 minutes"	How often does the trigger manager check for triggers to run
horizon.trigger.manager.timeout	"60 seconds"	Trigger manager timeout
horizon.grading.manager.timeout	"5 seconds"	Duration after which the grading manager times out when retrieving the grading configuration from the database
horizon.grading.manager.queue.size	100	How large can the grading manager queue can get before it discards new grading requests
horizon.grading.timeout	"5 seconds"	Duration after which the grading actor times out when grading a certificate (upon enrolment)
horizon.security.manager.timeout	"1 minute"	Duration after which the security manager times out when trying to authenticate a principal with its session
horizon.ca.manager.timeout	"1 minute"	Duration that the CA manager actor will wait to retrieve information about certificates (is it trusted ? its trust chain ?)
horizon.ca.manager.prefix	"Actor "	Name of the CA manager actor in the technical logs
horizon.security.manager.prefix	"Actor "	Name of the security manager actor in the technical logs
horizon.grading.actor.prefix	"Actor "	Name of the grading actor in the technical logs
horizon.grading.manager.prefix	"Actor "	Name of the grading manager actor in the technical logs
horizon.crl.updater.prefix	"Actor CRLUpdater-"	Name of the CRL updater actor in the technical logs
horizon.scheduler.manager.timeout	"60 seconds"	Duration after which the Scheduler manager actor times out when retrieving scheduled tasks in the database
horizon.scheduler.manager.prefix	"Actor "	Name of the Scheduler manager actor in the technical logs
horizon.event.timeout	"30 seconds"	Duration after which the event manager times out when trying to retrieve the last signed event in the database

Parameter	Default value	Description
horizon.event.manager.prefix	"Actor "	Name of the Event manager actor in the technical logs
horizon.event.manager.interval	"5 seconds"	How often will the Event Manager actor check in the database if new a new event appeared to sign it and display it in the "Events" section of Horizon
horizon.vault.manager.timeout	"1 minute"	Duration after which the Vault Manager times out when trying to encipher or decipher a configuration or certificate secret
horizon.pki.manager.timeout	"1 minute"	Duration after which the PKI Manager times out when trying to enroll or revoke a certificate

16.2.14. Other parameters

Parameter	Default value	Description
horizon.security.identity.provider.openid.state.separator	"#"	Separator character of the OpenID state

17. Endpoint configuration

17.1. Basic configuration

The basic configuration sets allowed hosts for all protocols using the `horizon-config` utility in RPM mode, and using the `ALLOWED_HOSTS` helm parameters

17.2. Advanced configuration

Endpoints can be configured to only allow certain capabilities using the `horizon.endpoints` config parameter.

The format is the following:

```
horizon.endpoints = [{
    # Hostname to allow
    host = "host.evertrust"
    # Allow configuration endpoints - default: false
    configuration = true
    # Allow event endpoints - default: false
    events = true
    # Allow discovery feed endpoints - default: false
    discovery = true
    # Allow WebRA endpoints - default: false
    webra = true
    # Allow WCCE endpoints - default: false
    wcce = true
    # Allow ACME endpoints - default: false
    acme = true
    # Allow EST endpoints - default: false
    est = true
    # Allow SCEP endpoints - default: false
    scep = true
    # Allow SCIM endpoints - default: false
    scim = true
    # Allow JAMF and Intune endpoints - default: false
    mdm = true
}, ...]
```

NOTE When `horizon.endpoints` is set, the hosts allowed with basic configuration are ignored

The following details each route that is authorized by the above capabilities.

17.2.1. EST

```
GET     /.well-known/est/:profile/cacerts
POST    /.well-known/est/:profile/simpleenroll
POST    /.well-known/est/:profile/simplereenroll
```

17.2.2. SCEP

```
GET     /certsrv/:profile/mscep_admin
GET     /certsrv/:profile/mscep_admin/*restUri
GET     /certsrv/:profile/mscep
POST    /certsrv/:profile/mscep
GET     /certsrv/:profile/mscep/*restUri
POST    /certsrv/:profile/mscep/*restUri
GET     /certSrv/:profile/mscep_admin
GET     /certSrv/:profile/mscep_admin/*restUri
GET     /certSrv/:profile/mscep
POST    /certSrv/:profile/mscep
GET     /certSrv/:profile/mscep/*restUri
POST    /certSrv/:profile/mscep/*restUri
GET     /scep/:profile/pkiclient.exe
POST    /scep/:profile/pkiclient.exe
```

17.2.3. MDM

```
GET     /intune/:profile/pkiclient.exe
GET     /jamf/:profile/mscep_admin
GET     /jamf/:profile/mscep_admin/*restUri
GET     /jamf/:profile/mscep
POST    /jamf/:profile/mscep
GET     /jamf/:profile/mscep/*restUri
POST    /jamf/:profile/mscep/*restUri
```

17.2.4. ACME

```
GET     /acme/:profile/directory
GET     /acme/:profile/new-nonce
HEAD    /acme/:profile/new-nonce
POST    /acme/:profile/new-acct
POST    /acme/acct/:profile/:accountId
POST    /acme/:profile/key-change
POST    /acme/:profile/new-order
GET     /acme/acct/:profile/:accountId/order/:orderId/finalize
GET     /acme/order/:profile/:orderId
POST    /acme/order/:profile/:orderId
POST    /acme/acct/:profile/:accountId/order/:orderId/finalize
GET     /acme/acct/:profile/:accountId/orders
```

```
POST  /acme/acct/:profile/:accountId/orders
POST  /acme/acct/:profile/:accountId/*restUri
GET   /acme/authz/:profile/:id
POST  /acme/authz/:profile/:id
POST  /acme/authz/:profile/:id/:challengeType
GET   /acme/authz/:profile/:id/:challengeType
GET   /acme/cert/:profile/:orderId
POST  /acme/cert/:profile/:orderId
POST  /acme/:profile/revoke-cert
```

17.2.5. WCCE

```
POST  /api/v1/wcce/enroll
GET   /api/v1/wcce/exchanges/:profile
```

17.2.6. WEBRA

```
GET   /api/v1/certificates/$id<[0-9a-fA-F]{24}>
GET   /api/v1/certificates/:pem
POST  /api/v1/certificates/
POST  /api/v1/certificates/aggregate
POST  /api/v1/certificates/csv
POST  /api/v1/certificates/search
GET   /api/v1/certificates/search/dictionary
PATCH /api/v1/certificates/run/$id<[0-9a-fA-F]{24}>/:triggerName/:event
GET   /api/v1/licenses/modules
POST  /api/v1/requests/aggregate
POST  /api/v1/requests/approve
POST  /api/v1/requests/cancel
POST  /api/v1/requests/csv
POST  /api/v1/requests/deny
GET   /api/v1/requests/profiles
POST  /api/v1/requests/search
POST  /api/v1/requests/submit
POST  /api/v1/requests/template
GET   /api/v1/requests/:id
GET   /api/v1/requests/search/dictionary
GET   /api/v1/rfc5280/crl/:pem
POST  /api/v1/rfc5280/crl
GET   /api/v1/rfc5280/pkcs10/:pem
POST  /api/v1/rfc5280/pkcs10
POST  /api/v1/rfc5280/pkcs12
GET   /api/v1/rfc5280/x509/:pem
POST  /api/v1/rfc5280/x509
GET   /api/v1/rfc5280/tc/:pem
POST  /api/v1/rfc5280/tc
POST  /api/v1/crypto/detect
```

```
GET    /api/v1/openssh/:base64
POST   /api/v1/openssh/
GET    /api/v1/rfc3161/:base64
POST   /api/v1/rfc3161/
GET    /api/v1/rfc6960/:base64
POST   /api/v1/rfc6960/
PATCH  /api/v1/security/identity/locals/
POST   /api/v1/security/identity/locals/password
GET    /api/v1/security/identity/locals/password/:identifier
GET    /api/v1/security/identity/providers/dynamic/enabled
GET    /api/v1/security/passwordpolicies/:name/generate
GET    /api/v1/security/principals/self
GET    /api/v1/security/principals/authenticate
GET    /api/v1/security/principals/logout
GET    /api/v1/security/principals/queries
GET    /api/v1/security/principals/queries/:name
POST   /api/v1/security/principals/queries
DELETE /api/v1/security/principals/queries/:name
GET    /api/v1/security/principals/dashboards
GET    /api/v1/security/principals/dashboards/:name
POST   /api/v1/security/principals/dashboards
PUT    /api/v1/security/principals/dashboards
DELETE /api/v1/security/principals/dashboards/:name
POST   /api/v1/security/principals/preferences
GET    /api/v1/security/principals/dictionary
GET    /api/v1/trustchains/
GET    /api/v1/trustchains/:anchor
GET    /api/v1/ui/
GET    /api/v1/endpoints/
```

17.2.7. EVENTS

```
POST   /api/v1/discovery/events/search
POST   /api/v1/discovery/events/csv
GET    /api/v1/discovery/events/:id
GET    /api/v1/discovery/events/search/dictionary
GET    /api/v1/events/integrity/run
GET    /api/v1/events/integrity/
POST   /api/v1/events/search
POST   /api/v1/events/csv
GET    /api/v1/events/:id
GET    /api/v1/events/search/dictionary
GET    /api/v1/licenses/modules
GET    /api/v1/rfc5280/crl/:pem
POST   /api/v1/rfc5280/crl
GET    /api/v1/rfc5280/pkcs10/:pem
POST   /api/v1/rfc5280/pkcs10
POST   /api/v1/rfc5280/pkcs12
GET    /api/v1/rfc5280/x509/:pem
```

```
POST   /api/v1/rfc5280/x509
GET    /api/v1/rfc5280/tc/:pem
POST   /api/v1/rfc5280/tc
POST   /api/v1/crypto/detect
GET    /api/v1/openssh/:base64
POST   /api/v1/openssh/
GET    /api/v1/rfc3161/:base64
POST   /api/v1/rfc3161/
GET    /api/v1/rfc6960/:base64
POST   /api/v1/rfc6960/
PATCH  /api/v1/security/identity/locals/
POST   /api/v1/security/identity/locals/password
GET    /api/v1/security/identity/locals/password/:identifier
GET    /api/v1/security/identity/providers/dynamic/enabled
GET    /api/v1/security/passwordpolicies/:name/generate
GET    /api/v1/security/principals/self
GET    /api/v1/security/principals/authenticate
GET    /api/v1/security/principals/logout
GET    /api/v1/security/principals/queries
GET    /api/v1/security/principals/queries/:name
POST   /api/v1/security/principals/queries
DELETE /api/v1/security/principals/queries/:name
POST   /api/v1/security/principals/preferences
GET    /api/v1/security/principals/dictionary
GET    /api/v1/ui/
GET    /api/v1/endpoints/
```

17.2.8. CONFIGURATION

```
GET    /api/v1/adoc/
GET    /api/v1/automation/executions/
GET    /api/v1/automation/executions/:name
POST   /api/v1/automation/executions/
PUT    /api/v1/automation/executions/
DELETE /api/v1/automation/executions/:name
GET    /api/v1/automation/policies/
GET    /api/v1/automation/policies/:name
POST   /api/v1/automation/policies/
PUT    /api/v1/automation/policies/
DELETE /api/v1/automation/policies/:name
GET    /api/v1/cas/
GET    /api/v1/cas/:name
POST   /api/v1/cas/
PUT    /api/v1/cas/
DELETE /api/v1/cas/:name
GET    /api/v1/caches/crls
GET    /api/v1/caches/crls/:ca
GET    /api/v1/certificate/grading/policies/
GET    /api/v1/certificate/grading/policies/:name
GET    /api/v1/certificate/grading/policies/:policy/explain/:input
POST   /api/v1/certificate/grading/policies/:policy/explain
GET    /api/v1/certificate/grading/policies/:policy/run
GET    /api/v1/certificate/grading/rulesets/
GET    /api/v1/certificate/grading/rulesets/:name
GET    /api/v1/certificate/grading/rulesets/:ruleset/explain/:input
POST   /api/v1/certificate/grading/rulesets/:ruleset/explain
GET    /api/v1/certificate/labels/
GET    /api/v1/certificate/labels/:name
POST   /api/v1/certificate/labels/
PUT    /api/v1/certificate/labels/
DELETE /api/v1/certificate/labels/:name
GET    /api/v1/certificate/profiles/
GET    /api/v1/certificate/profiles/:name
POST   /api/v1/certificate/profiles/
PUT    /api/v1/certificate/profiles/
DELETE /api/v1/certificate/profiles/:name
GET    /api/v1/datasources/
GET    /api/v1/datasources/:name
POST   /api/v1/datasources/
PUT    /api/v1/datasources/
DELETE /api/v1/datasources/:name
PATCH  /api/v1/datasources/
POST   /api/v1/datasource/flows/
POST   /api/v1/datasource/flows/template
POST   /api/v1/templatestring/playground
GET    /api/v1/discovery/campaigns/
GET    /api/v1/discovery/campaigns/:name
POST   /api/v1/discovery/campaigns/
PUT    /api/v1/discovery/campaigns/
DELETE /api/v1/discovery/campaigns/:name
PATCH  /api/v1/discovery/campaigns/:name
GET    /api/v1/licenses/modules
GET    /api/v1/pki/queues/
GET    /api/v1/pki/queues/:name
POST   /api/v1/pki/queues/
PUT    /api/v1/pki/queues/
DELETE /api/v1/pki/queues/:name
GET    /api/v1/pki/connectors/
GET    /api/v1/pki/connectors/:name
POST   /api/v1/pki/connectors/
PUT    /api/v1/pki/connectors/
DELETE /api/v1/pki/connectors/:name
PATCH  /api/v1/pki/connectors/connect
PATCH  /api/v1/pki/connectors/materials
GET    /api/v1/proxy/httpproxies/
GET    /api/v1/proxy/httpproxies/:name
POST   /api/v1/proxy/httpproxies/
PUT    /api/v1/proxy/httpproxies/
DELETE /api/v1/proxy/httpproxies/:name
GET    /api/v1/rfc5280/crl/:pem
```

```
POST   /api/v1/rfc5280/crl
GET    /api/v1/rfc5280/pkcs10/:pem
POST   /api/v1/rfc5280/pkcs10
POST   /api/v1/rfc5280/pkcs12
GET    /api/v1/rfc5280/x509/:pem
POST   /api/v1/rfc5280/x509
GET    /api/v1/rfc5280/tc/:pem
POST   /api/v1/rfc5280/tc
POST   /api/v1/crypto/detect
GET    /api/v1/openssh/:base64
POST   /api/v1/openssh/
GET    /api/v1/rfc3161/:base64
POST   /api/v1/rfc3161/
GET    /api/v1/rfc6960/:base64
POST   /api/v1/rfc6960/
GET    /api/v1/security/identity/locals/
GET    /api/v1/security/identity/locals/:identifier
POST   /api/v1/security/identity/locals/
PUT    /api/v1/security/identity/locals/
DELETE /api/v1/security/identity/locals/:identifier
PATCH  /api/v1/security/identity/locals/
POST   /api/v1/security/identity/locals/password
GET    /api/v1/security/identity/locals/password/:identifier
GET    /api/v1/security/identity/providers/
GET    /api/v1/security/identity/providers/:name
POST   /api/v1/security/identity/providers/
PUT    /api/v1/security/identity/providers/
DELETE /api/v1/security/identity/providers/:name
GET    /api/v1/security/identity/providers/dynamic/enabled
POST   /api/v1/security/identity/providers/search
GET    /api/v1/security/passwordpolicies/
GET    /api/v1/security/passwordpolicies/:name
GET    /api/v1/security/passwordpolicies/:name/generate
POST   /api/v1/security/passwordpolicies/
PUT    /api/v1/security/passwordpolicies/
DELETE /api/v1/security/passwordpolicies/:name
GET    /api/v1/security/principals/self
GET    /api/v1/security/principals/authenticate
GET    /api/v1/security/principals/logout
GET    /api/v1/security/principals/queries
GET    /api/v1/security/principals/queries/:name
POST   /api/v1/security/principals/queries
DELETE /api/v1/security/principals/queries/:name
POST   /api/v1/security/principals/preferences
GET    /api/v1/security/principals/dictionary
GET    /api/v1/security/principalinfos/:identifier
POST   /api/v1/security/principalinfos/
PUT    /api/v1/security/principalinfos/
DELETE /api/v1/security/principalinfos/:identifier
POST   /api/v1/security/principalinfos/search
GET    /api/v1/security/roles/
GET    /api/v1/security/roles/:name
POST   /api/v1/security/roles/
PUT    /api/v1/security/roles/
DELETE /api/v1/security/roles/:name
GET    /api/v1/security/teams/
GET    /api/v1/security/teams/:name
POST   /api/v1/security/teams/
PUT    /api/v1/security/teams/
DELETE /api/v1/security/teams/:name
PATCH  /api/v1/security/teams/:previousTeam/:newTeam
GET    /api/v1/security/credentials/
GET    /api/v1/security/credentials/:name
POST   /api/v1/security/credentials/
PUT    /api/v1/security/credentials/
DELETE /api/v1/security/credentials/:name
GET    /api/v1/scheduler/tasks/
GET    /api/v1/scheduler/tasks/:id/run
GET    /api/v1/scheduler/tasks/:id
POST   /api/v1/scheduler/tasks/
PUT    /api/v1/scheduler/tasks/
DELETE /api/v1/scheduler/tasks/:id
GET    /api/v1/thirdparty/connectors/
GET    /api/v1/thirdparty/connectors/:name
POST   /api/v1/thirdparty/connectors/
PUT    /api/v1/thirdparty/connectors/
DELETE /api/v1/thirdparty/connectors/:name
GET    /api/v1/triggers/
GET    /api/v1/triggers/:name
POST   /api/v1/triggers/
PUT    /api/v1/triggers/
DELETE /api/v1/triggers/:name
PATCH  /api/v1/triggers/
GET    /api/v1/security/scim/profiles/
GET    /api/v1/security/scim/profiles/:name
PUT    /api/v1/security/scim/profiles/
DELETE /api/v1/security/scim/profiles/:name
POST   /api/v1/security/scim/profiles/
GET    /api/v1/wcce/forests
GET    /api/v1/wcce/forests/:name
POST   /api/v1/wcce/forests
PUT    /api/v1/wcce/forests
DELETE /api/v1/wcce/forests/:name
GET    /api/v1/system/configuration/
GET    /api/v1/system/configuration/:type
PUT    /api/v1/system/configuration/
GET    /api/v1/ui/
POST   /api/v1/ui/cr/format
GET    /api/v1/endpoints/
```

17.2.9. SCIM

```
GET    /security/scim/:scimProfile/ServiceProviderConfig
GET    /security/scim/:scimProfile/ResourceTypes
GET    /security/scim/:scimProfile/Users
GET    /security/scim/:scimProfile/Users/:identifier
POST   /security/scim/:scimProfile/Users
PATCH  /security/scim/:scimProfile/Users/:userName
PUT    /security/scim/:scimProfile/Users/:identifier
DELETE /security/scim/:scimProfile/Users/:identifier
GET    /security/scim/:scimProfile/Groups/:groupName
GET    /security/scim/:scimProfile/Groups
PATCH  /security/scim/:scimProfileName/Groups/:GroupName
```

18. Logging

Horizon is able to format output log thanks to logback.

Horizon defines a default rolling file appender named **RUN**.

The default configuration keeps the technical logs for 30 days into files with the following naming convention:

```
horizon.log-<yyyy-MM-dd>.log
```

Those files are available under the **/opt/horizon/var/log** directory

18.1. Directly sending logs to your syslog server

1. Access the EverTrust Horizon server through SSH with an account with administrative privileges;

2. Using an editor like vi, open the *horizon-logback.xml* file located at **/opt/horizon/etc/horizon-logback.xml** ;

3. Edit the appender named "SYSLOG" to change the IP address for the syslogHost to redirect to your own syslog server. As an example, if your syslog server is on 192.168.1.2 and the Horizon logs must be processed by the LOCAL6 facility, the syslog appender should look like this:

```xml
<appender name="SYSLOG" class="ch.qos.logback.classic.net.SyslogAppender">
    <syslogHost>192.168.1.2</syslogHost>
    <facility>LOCAL6</facility>
    <suffixPattern>%msg%n</suffixPattern>
</appender>
```

4. Still in the **horizon-logback.xml** file, update any logger with the SYSLOG appender ref and ensure that the log level is set to "INFO":

```xml
<logger name="event" level="INFO">
    <appender-ref ref="SYSLOG"/>
</logger>
```

5. Save your modifications and restart the Horizon service:

```
$ systemctl restart horizon
```

The functional logs from Horizon should now be received by your remote syslog server:

```
horizon {"code": "SERVICE-
STOP","details":[{"key":"horizonVersion","value":"2.6.0"},{"key":"message","value":"Se
rvice successfully
stopped"}],"module":"service","node":"horizon","timestamp":1674054152149,"status":"suc
cess"}
horizon {"code": "SERVICE-
START","details":[{"key":"horizonVersion","value":"2.6.0"},{"key":"message","value":"S
ervice successfully
started"}],"module":"service","node":"horizon","timestamp":1674054170567,"status":"suc
cess"}
```

18.2. Using the local syslog server for filtering and forwarding

Alternatively, you might want to use a local syslog instance to add grok filtering to your logs before forwarding them to your own syslog server. To do so, ensure that you have a syslog instance running (like **rsyslog**), then:

1. Access the EverTrust Horizon server through SSH with an account with administrative privileges;

2. With an editor like vi, edit the **/etc/rsyslog.d/horizon.conf** (or create it if it does not exist yet) to add this line:

```
local6.*                                                @REMOTE_SYSLOG_HOSTNAME
```

Don't forget to replace the *REMOTE_SYSLOG_HOSTNAME* to the IP or DNS name of your remote syslog server. As an example, if your syslog server is on 192.168.1.2, the line should look like this:

```
local6.*                                                @192.168.1.2
```

Note that you must set up your syslog host to accept UDP traffic on a specific port (here, we are going to use the default port which is 514) and catch the local6 facility logs, however the configuration of your own syslog host is out of the scope of this document.

3. Edit the */etc/rsyslog.conf* file to uncomment the module and input lines of the UDP section:

```
#module(load="imudp") # needs to be done just once
#input(type="imudp" port="514")
```

They should look like this after uncommenting:

```
module(load="imudp") # needs to be done just once
input(type="imudp" port="514")
```

4. Restart your syslog service:

```
$ systemctl restart rsyslog
```

18.3. JSON Logging

1. Access the EverTrust Horizon server through SSH with an account with administrative privileges;

2. Using an editor like vi, open the *horizon-logback.xml* file located at **/opt/horizon/etc/horizon-logback.xml** ;

3. To send the JSON logs to a syslog server: Edit or create the appender named "JSON_SYSLOG" to change the IP address for the syslogHost to redirect to your own syslog server. As an example, if your syslog server is on 192.168.1.2 and the Horizon logs must be processed by the LOCAL6 facility, the syslog appender should look like this:

```
<conversionRule conversionWord="syslogStart"
converterClass="ch.qos.logback.classic.pattern.SyslogStartConverter"/>
<appender name="JSON_SYSLOG"
class="net.logstash.logback.appender.LogstashUdpSocketAppender">
    <host>192.168.1.2</host>
    <port>514</port>
    <layout class="net.logstash.logback.layout.LogstashLayout">
      <prefix class="ch.qos.logback.classic.PatternLayout">
        <pattern>%syslogStart{LOCAL6}</pattern>
      </prefix>
      <fieldNames>
        <timestamp>time</timestamp>
        <logger>logger</logger>
        <thread>thread</thread>
        <level>severity</level>
        <stackTrace>exception</stackTrace>
      </fieldNames>
      <customFields>{"app":"horizon", "hostname":"${HOSTNAME}"}</customFields>
    </layout>
</appender>
```

To send the JSON logs to the local console: Edit or create the appender named "STDOUT" to change the encoder to a JSON one. The final configuration should look like this.

```
<appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">
    <encoder class="net.logstash.logback.encoder.LogstashEncoder">
        <fieldNames>
          <timestamp>time</timestamp>
          <logger>logger</logger>
          <thread>thread</thread>
          <level>severity</level>
          <stackTrace>exception</stackTrace>
        </fieldNames>
        <customFields>{"app":"horizon", "hostname":"${HOSTNAME}"}</customFields>
    </encoder>
  </appender>
```

4. Still in the **horizon-logback.xml** file, update any logger with the appender ref and ensure that the log level is not OFF:

```
<logger name="event" level="INFO">
    <appender-ref ref="JSON_SYSLOG"/>
</logger>
```

or

```
<logger name="controllers" level="INFO">
    <appender-ref ref="STDOUT"/>
</logger>
```

5. Save your modifications and restart the Horizon service:

```
$ systemctl restart horizon
```

The functional logs from Horizon should now be in JSON Format:

```
Console : {"time": "2023-08-16T16:12:54.481+02:00","@version":"1","message":"[Actor
pkimanager] - Registering PKI Queue 'slowed-queue' (cluster wide:
'false')","logger":"actors.pki.PKIManagerActor","thread":"application-blocking-io-
dispatcher-
43","severity":"INFO","level_value":20000,"HOSTNAME":"horizon.evertrust","application.
home":"/opt/horizon","kamonSpanId":"c5a74b959971c7ee","kamonTraceId":"b1ccb54c9eb7e493
","kamonSpanName":"/ui","app":"horizon","hostname":"horizon.evertrust"}
Syslog : 2023-08-16T16:12:54+02:00 horizon.evertrust {"time": "2023-08-
16T16:12:54.482+02:00","@version":"1","message":"[Actor pkimanager] - Registering PKI
Queue 'test' (cluster wide:
'false')","logger":"actors.pki.PKIManagerActor","thread":"application-blocking-io-
dispatcher-
43","severity":"INFO","level_value":20000,"HOSTNAME":"horizon.evertrust","application.
home":"/opt/horizon","kamonSpanId":"c5a74b959971c7ee","kamonTraceId":"b1ccb54c9eb7e493
","kamonSpanName":"/ui","app":"horizon","hostname":"horizon.evertrust"}
```

19. Events

All events displayed in this document work in a similar manner. In case of failure, the event will display the reason of said failure. This behavior is also valid for warning-status events.

19.1. ACME

- **ACME-ACCOUNT-KEY-CHANGE**
- **ACME-ACCOUNT-REGISTER**
  This event is triggered when an account is unsuccessfully registered. Mainly due to errors in registration parameters (mail, name, …)
- **ACME-ACCOUNT-UPDATE**
  This event is triggered when an account is unsuccessfully updated. Mainly due to errors in updated parameters (mail, name, …)
- **ACME-AUTHORIZATION-DEACTIVATE**
  This event is triggered when an authorization is unsuccessfully deactivated.
- **ACME-CHALLENGE-REQUEST-VERIFY**
  This event is triggered when trying to use the challenge feature as an authentication method. It issues a warning if this is not applicable to this authentication case.
- **ACME-CHALLENGE-VERIFY**
  This event is triggered when a challenge is used as an authentication method. It issues a warning if the challenge is invalid or if the user doesn't correspond to the challenge.
- **ACME-ORDER-CERTIFICATE**
  This event is triggered when a user tries to access a certificate. It presents a failure in case the user doesn't have the necessary rights and permissions.
- **ACME-ORDER-FINALIZE**
  This event traces the status of the certificate's status. It presents a failure if the certificate is pending or if is not valid.
- **ACME-ORDER-NEW**
  This event is triggered when a user tries to order a certificate. It issues a failure if the user doesn't have the necessary rights and permissions for requesting this type of new certificate.
- **ACME-ORDER-UPDATE**
  This event is triggered when a user tries to order an update on certificate. It issues a failure if the user doesn't have the necessary rights and permissions to update this type of certificates.
- **ACME-REVOKE**
  This event is triggered when Horizon tries revoking a certificate using the ACME protocol. A warning can occur if the certificate is already revoked. A failure can occur if the certificate cannot be found based on the provided thumbprint.

19.2. ACTOR

IMPORTANT	ACTOR

Replaced by PKI-CONNECTOR.

19.3. BOOTSTRAP

Bootstrap events relate to the initial setup of the Horizon platform.

- **BOOTSTRAP-ADMINISTRATOR-ACCOUNT**
  This event is triggered when installing Horizon, it corresponds to the creation of the administrator local identity on Horizon.
- **BOOTSTRAP-ADMINISTRATOR-PRINCIPAL**
  This event is triggered when installing Horizon, it corresponds to the creation of a link between the administrator account and its right.
- **BOOTSTRAP-GRADING-POLICY**
  This event is triggered when installing Horizon, it corresponds to the creation of the "Horizon Grading Policy" which itsell contains different grading rulesets.
- **BOOTSTRAP-GRADING-RULESET**
  This event is triggered when installing Horizon, it corresponds to the creation of different grading rulesets which are:
  "ANSSI Cryptographic Content",
  "CA/B Forum Ruleset",
  "NIST and ANSSI ECDSA Cryptographic Ruleset",
  "E-Mail Certificate Ruleset",
  "IETF PKIX Ruleset",
  "NIST and ANSSI RSA Cryptographic Ruleset" &
  "TLS Certificate Ruleset".
  For more information about those grading rule sets, click here.
- **BOOTSTRAP-LOCAL-IDENTITY-PROVIDER**
  This event is triggered when installing Horizon, it corresponds to the creation of a provider of type Local so that the administrator can connect after startup.
- **BOOTSTRAP-PASSWORD-POLICY**
  This event is triggered when installing Horizon, it corresponds to the creation of the Horizon-Default password policy.
- **BOOTSTRAP-SYSTEM-CONFIGURATION**
  This event is triggered when installing Horizon, it corresponds to the creation of internal configuration elements such as the CRON internal monitor.

19.4. CA

- **CA-CERT-SYNC**
  This event is triggered when a Certification Authority is revoked and certificates managed in Horizon are subsequently revoked. The synchronization revokes all the underlying certificates.
- **CA-CRL-UPDATE**
  This event is triggered when Horizon tries fetching a CRL from a specified CRLDP.

19.5. CONF

CONF EVENTS are triggered when users interact with configuration elements. This includes certificate templates, notification triggers, Certification Authorities…

- **CONF-ADD**
  This event is triggered when a user tries to add a configuration element.
- **CONF-DELETE**
  This event is triggered when a user tries to delete a configuration element.
- **CONF-TEST**
  This event is triggered when a request of notification happens.
- **CONF-UPDATE**
  This event occurs when a user tries to modify a configuration element.

19.6. CRMP

- **CRMP-AUTHENTICATION**
  This event occurs when a user tries to authenticate. It fails if the authentication is invalide.
- **CRMP-BAD-REQUEST**
  This event occurs when a wrong request is issued. For instance if an unavailable action is requested.
- **CRMP-ENROLL**
  This event occurs when a enrolling request happens. It fails if the CRMP enrollment is unsuccessful.
- **CRMP-LIST**
  This event occurs when a user tries to access the profiles list. Fails if he doesn’t have the required rights and authorisations.
- **CRMP-PROFILE-PROPERTIES**
  This event occurs when a user tries to access a profile. Fails if he doesn’t have the required rights and authorisations or if the profile doesn’t exist.
- **CRMP-RECOVER**
  This event occurs when a user tries to recover a CRMP. It fails if its not technically possible or if the user doesn’t have the necessary rights and permissions.
- **CRMP-RETRIEVE**
  This event occurs when a user tries make a research. It issues a warning if the research field is badly used (empty).
- **CRMP-REVOKE**
  This event occurs when a user tries to revoke a certificate. It fails or issues a warning respectively if the user doesn’t have the necessary rights and permissions or if the certificate is expired.

19.7. DATASOURCE

- **DATASOURCE-IGNORED**
  This event is triggered when a datasource is ignored. For example, when an input is missing.

19.8. DISCOVERY

- **DISCOVERY-CAMPAIGN-FLUSH**
  This event is triggered when running a *Discovery* campaign.

19.9. EST

- **EST-CACERTS**
  This event is triggered when an error occurs during the call to the CACert endpoint when using the EST protocol.
- **EST-REVOKE-ON-RENEW**
  This event is triggered when a certificate is revoked following its renewal.
- **EST-SIMPLE-ENROLL**
  This event is triggered when enrolling a certificate through the EST protocol.
- **EST-SIMPLE-REENROLL**
  This event is triggered when re-enrolling a certificate through the EST protocol.

19.10. GRADING

- **GRADING-END**
  This event is triggered at the end of the grading process of a certificate.
- **GRADING-ERROR**
  This event is triggered if an error occurs while grading a certificate.
- **GRADING-START**
  This event is triggered at the beginning of the grading process of a certificate.

19.11. INTERNAL-MONITOR

- **INTERNAL-MONITOR-INIT**
  This event occurs when a bad initialization of the internal monitor happens. It is a failure case, happening for instance when it is not configured
- **INTERNAL-MONITOR-RUN**
  This event occurs when the internal monitor completed successfully.

19.12. INVALID-SEAL

- **INVALID-SEAL-PENDING-EVENT**
  This event occurs when an invalid seal is pending.

19.13. LICENSE

- **LICENSE-ERROR**
  This event occurs when an error is related to the *License*.
  For example, when the license in use is expired.
- **LICENSE-LIMIT-REACHED**
  This event is triggered when a limit built into the license is the reached.
  For example, if only one discovery campaign is available, then reaching that threshold will trigger an error saying *"Maximum number of discovery campaign(s) reached (x)"* where *x* is the availability threshold.

19.14. LIFECYCLE

- **LIFECYCLE-ENROLL**
  This event is triggered when a user tries to enroll an end-entity certificate. The event specifies the Distinguished Name of the enrolled certificate, its serial number as well as the Certificate Authority that enrolled said certificate in case of success. In case of failure, the reason of the failure is specified (e.g.: "Unauthorized DN element").
- **LIFECYCLE-ESCROW**
  This event is triggered when *Horizon* tries to escrow a key for an issued certificate.
- **LIFECYCLE-IMPORT**
  This event is triggered when trying to import a certificate in Horizon.
- **LIFECYCLE-MAX-CERT-PER-HOLDER**
  This event is triggered when an error occurs trying to enforce the max certificates per holder parameter.
- **LIFECYCLE-MIGRATE**
  This event is triggered when trying to migrate certificates. This means, making an external certificate managed by *Horizon*.
- **LIFECYCLE-RECOVER**
  This event is triggered *Horizon* tries to recover a certificate.
- **LIFECYCLE-RENEW**
  This event is triggered when *Horizon* tries to renew a certificate.
- **LIFECYCLE-REVOKE**
  This event occurs when a user tries to revoke a certificate.
  Note that no event is triggered when a certificate expires.
- **LIFECYCLE-UPDATE**
  This event is triggered when a user tries updating the details related to a certificate. The *Labels* and the *Ownership* can be edited.

19.15. PKI CONNECTOR

- **PKI-CONNECTOR**
  This event is triggered when a PKI connector cannot be properly built between Horizon and the

chosen PKI.

19.16. REQUEST

- **REQUEST-APPROVE**
  This event is triggered when approving a request.
- **REQUEST-CANCEL**
  This event is triggered when canceling a request.
- **REQUEST-DENY**
  This event is a success when a request is denied.
- **REQUEST-SUBMIT**
  This event is triggered when submitting a request.
- **REQUEST-TEMPLATE**
  This event is triggered when requesting a template. It can fail when trying to enroll a workflow without a module.

19.17. SCEP

- **SCEP-ENROLL**
  This event is triggered when enrolling a certificate via SCEP. Fails when missing mandatory certificate’s elements or when missing rights and/or permissions to enroll the certificate.
- **SCEP-GET-CA-CERT**
  This event is triggered when requesting a CA certificate via SCEP. Fails when missing mandatory certificate’s elements or when missing rights and/or permissions to enroll the certificate.
- **SCEP-GET-CERT-INITIAL**
  This event is triggered when requesting the initial certificate via SCEP. Fails when missing mandatory certificate’s elements or when missing rights and/or permissions to enroll the certificate.
- **SCEP-NDES-EMULATION**
  This event requesting a certificate with the scep profile template using the NDES server. It fails if the two don’t comply with one another.
- **SCEP-PKI-CLIENT**
  This event is triggered when using the pkiclient profile. It fails if the request is invalid, if the operation is not allow for the type of certificate the user wants to manage, or if the user doesn’t have the necessary rights and permissions to execute the action.
- **SCEP-PKI-OPERATION**
  This event is triggered when operating through the PKI.
- **SCEP-RENEW**
  This event is triggered when renewing a certificate. It fails he is system fails to enroll the new certificate.
- **SCEP-REVOKE-ON-RENEW**
  This event is triggered when revoking a certificate at its renewal.

19.18. SCHEDULED-TASK

- **SCHEDULED-TASK-COMPLETE**
  This event is triggered when a scheduled task end. It fails if the task fails.
- **SCHEDULED-TASK-RUN**
  This event is triggered when trying to pass a scheduled task to "running" status. Fails if this status is not achieved.

19.19. SEC

- **SEC-AUTHENTICATION**
  This event is triggered when a user tries to connect. The local or OpenID identifier is specified whether it is a failure or a success.

19.19.1. AUTHORIZATION

NOTE These events relate to the Security>Access Management>Authorizations tab under configuration.

- **SEC-AUTHORIZATION-ADD**
  This event is triggered when a user tries to create a an authorization profile.
- **SEC-AUTHORIZATION-DELETE**
  This event is triggered when a user tries to delete an authorization profile.
- **SEC-AUTHORIZATION-UPDATE**
  This event is triggered when a user tries to modify elements inside an authorization profile. The event specifies the modified fields.

19.19.2. CREDENTIAL

NOTE These events relate to the Security>Credentials tab under configuration.

- **SEC-CREDENTIALS-ADD**
  This event occurs when a user tries creating a new Credential.
- **SEC-CREDENTIALS-DELETE**
  This event occurs when a user tries deleting a Credential.
- **SEC-CREDENTIALS-UPDATE**
  This event occurs when a user tries modifying a Credential.

19.19.3. IDENTITY

NOTE These events relate to the Security>Access Management>Identity tab under configuration.

- **SEC-IDENTITY-PROVIDER-ADD**

This event occurs when a user tries creating an identity provider profile.

- **SEC-IDENTITY-PROVIDER-DELETE**
  This event occurs when a user tries deleting an identity provider profile.
- **SEC-IDENTITY-PROVIDER-UPDATE**
  This event occurs when a user tries modifying an identity provider profile. The modified fields are specified in the event.

19.19.4. LOCAL-IDENTITY

NOTE These events relate to the Security>Access Management>Local accounts tab under configuration.

- **SEC-LOCAL-IDENTITY-ADD**
  This event is triggered when a user tries creating a local account.
- **SEC-LOCAL-IDENTITY-DELETE**
  This event is triggered when a user tries to delete a local account.
- **SEC-LOCAL-IDENTITY-RESET**
  This event is triggered when executing the reset password workflow.
- **SEC-LOCAL-IDENTITY-UPDATE**
  This event is triggered when a user tries modifying a local account. The modified fields are specified. Updating the password falls in this event.

19.19.5. PASSWORD-POLICY

NOTE These events relate to the Security>Password Policies tab under configuration.

- **SEC-PASSWORD-POLICY-ADD**
  This event is triggered when a user tries creating a new password policy.
- **SEC-PASSWORD-POLICY-DELETE**
  This event is triggered when a user tries deleting a password policy.
- **SEC-PASSWORD-POLICY-UPDATE**
  This event is triggered when a user tries modifying a password policy.

19.19.6. ROLE

NOTE These events relate to the Security>Access Management>Roles tab under configuration.

- **SEC-ROLE-ADD**
  This event is triggered when a user tries to create a new role.
- **SEC-ROLE-DELETE**
  This event is triggered when a user tries to delete a role.
- **SEC-ROLE-UPDATE**

This event is triggered when a user tries to modify a role. The modified fields are specified in the event.

19.19.7. SCIM PROFILES

NOTE These events relate to the Security>SCIM Profiles tab under configuration.

- **SEC-SCIM-PROFILE-ADD**
  This event is triggered when a user tries creating a new SCIM profile.
- **SEC-SCIM-PROFILE-DELETE**
  This event is triggered when a user tries deleting a SCIM profile.
- **SEC-SCIM-PROFILE-UPDATE**
  This event is triggered when a user tries modifying a SCIM profile.

19.19.8. TEAM

NOTE These events relate to the Security>Teams tab under configuration.

- **SEC-TEAM-ADD**
  This event is triggered when a user tries creating a team.
- **SEC-TEAM-DELETE**
  This event is triggered when a user tries deleting a team.
- **SEC-TEAM-SWITCH**
- **SEC-TEAM-UPDATE**
  This event is triggered when a user tries modifying a team element (that does not include adding/removing users).

19.20. SERVICE

- **SERVICE-START**
  This event is triggered when the Horizon service is started.
- **SERVICE-STOP**
  This event is triggered when the Horizon service is manually stopped.

19.21. SUPERVISOR

IMPORTANT **SUPERVISOR**
This event has been deprecated.

19.22. SYNC

- **SYNC-ENROLL**
  This event is triggered when syncing an enrollment with a third party.

- **SYNC-RENEW**
  This event is triggered when syncing an renewal with a third party.
- **SYNC-REVOKE**
  This event is triggered when syncing an revocation with a third party.

19.23. TEAM

- **TEAM-SWITCH**
  This event is triggered when using the switch workflow.

19.24. THIRD-PARTY

- **THIRD-PARTY-CONNECTOR**
  This event is triggered as a *warning* when Horizon cannot build a connection with a third party.

19.25. TRIGGER

Trigger events relate to *Notifications* and can occur based on configurations made under *Third Parties* or under *Protocols*.

- **TRIGGER-DELETE**
  This event occurs when Horizon tries deleting a certificate within a third party.
- **TRIGGER-EMAIL**
  This event occurs when a *Trigger* that sends an email is activated. The event specifies to whom the email is addressed.
- **TRIGGER-NOTIFICATION**
  This event occurs when a *Trigger* that sends a notification is activated.
- **TRIGGER-PUSH**
  This event occurs when Horizon tries to push a certificate to a third party.
- **TRIGGER-REMOVE**
  This event occurs when Horizon tries removing a certificate within a third party.

19.26. UNSEALED

- **UNSEALED-PENDING-EVENT**
  This event is triggered when an event in the *pending events* collection (in the database) has not been signed.

19.27. WCCE

- **WCCE-ENROLL**
  This event is triggered when a client tries to enroll a certificate through Horizon using the WCCE protocol.