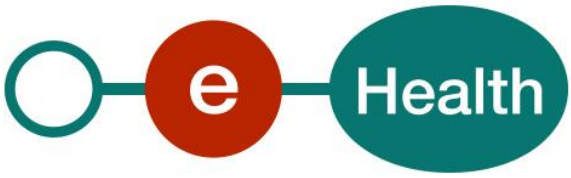


eHealth Certificates Manager (CertRa - EtkRa) V2

Cookbook V0.3

This document is provided to you free of charge by the

eHealth platform

Willebroekkaai 38

38, Quai de Willebroek

1000 BRUSSELS

Table of contents

To the attention of: “IT expert” willing to integrate this web service.

1. Document management

1.1 Document history

Version	Date	Author	Description of changes / remarks
0.1	04/10/2017	Smals	Initial version
0.2	05/10/2017	Smals	Reviewed Smals
0.3	13/12/2017	Smals	Reviewed eHealth platform

1.2 Use of the keywords

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in IETF RFC 2119.

(***https://www.ietf.org/rfc/rfc2119.txt***)

2. Introduction

2.1 Goal of the service

The services EtkRA and CertRA form the back end for the Certificate Manager of the eHealth platform. From now on, the developers of the software have the possibility to integrate the eHealth Certificate Manager in their software. This document describes the functionalities of those services in order to integrate them in a software package.

An eHealth certificate is composed of two key pairs: one for the authentication and one for the encryption i.e. the encryption token (ETK).

- The 'CertRA' contains the necessary operations to request and to complete an authentication key i.e. certificate, and to revoke both certificates.
- The 'EtkRA' offers the operations to request, to complete and to activate the encryption certificate.

2.2 Goal of the document

This document describes the services of eHealth certificates manager as provided by the eHealth platform. In the cookbook, you will find an explanation of the structure and the content aspects of the possible requests and the replies of the CertRA and the EtkRa web services. An example illustrates each of those messages. In addition, a list of possible errors can be found in this document.

This information should allow (the IT department of) an organization to develop and to integrate those web services to their software.

Some technical and legal requirements must be met in order to allow the integration of the web services of the eHealth platform in the client applications.

This document is neither a development nor a programming guide for the internal applications; eHealth partners always keep a total freedom within those fields. Nevertheless, in order to interact in a smooth, homogeneous and risk controlled way with a maximum of partners, eHealth partners must commit to comply with the specifications, the data format, and the release processes described within this document.

2.3 eHealth platform document references

All the document references can be found in the technical library on the portal of the eHealth platform[1]. These versions or any following versions can be used for the service of the eHealth platform.

ID	Title	Version	Date	Author
1	Glossary.pdf			eHealth platform
2	Service contract (WSDL + XSD)			eHealth platform

2.4 External document references

All documents can be found through the internet. They are available to the public, but not supported by eHealth.

[1] ***www.ehealth.fgov.be***

ID	Title	Source	Date	Author
1	eID SDK	***http://eid.belgium.be/fr/developper_des_applications_eid/eid_software_development_kit***		BOSA
2	eID Commons Library	***https://www.e-contract.be/sites/commons-eid/***		e-contract

3. Support

3.1 For issues in production

eHealth platform contact center:

- Phone: 02/788 51 55
- Mail: ***support@ehealth.fgov.be***
- *Contact Form :*
  - ***https://www.ehealth.fgov.be/nl/neem-contact-op-met-de-openbare-instelling-eHealth-platform*** (Dutch)
  - ***https://www.ehealth.fgov.be/fr/contactez-institution-publique-plate-forme-eHealth*** *(French)*

3.2 For issues in acceptance

Integration-support@ehealth.fgov.be

3.3 For business issues

- regarding an existing project: the project manager in charge of the application or service
- regarding a new project and other business issues: ***info@ehealth.fgov.be***

3.4 Certificates

- In order to access the secured eHealth platform environment you have to obtain an eHealth platform certificate, used to identify the initiator of the request. In case you do not have one, please consult:

  Dutch version: ***https://www.ehealth.fgov.be/nl/support/basisdiensten/ehealth-certificaten***

  French version: ***https://www.ehealth.fgov.be/fr/support/services-de-base/certificats-ehealth***
- For technical issues regarding eHealth platform certificates

  Acceptance: ***acceptance-certificates@ehealth.fgov.be***

  Production: ***support@ehealth.fgov.be***

4. Step-by-step

4.1 Technical requirements

In order to call the CertRA services, the consumer will have to produce the digital signatures by using the eID signature certificate. The signing with the eID is beyond the scope of this cookbook; therefore please refer to the eID SDK documentation to learn how to sign in your application.

Most of the software requires the key pair contained in a PKCS12 type of the key store. The authentication key gets the 'Alias' of the authentication certificate and the encryption key gets its serial number being the decimal representation of the 'Alias'.

4.2 Request/Response examples

In annex, you can find for each soap operation a sample of request/response.

4.3 Security of the web services by XML Signature

4.3.1 Introduction

Any secured webservice uses the XML Enveloped Signature specification.

Two types of private key are used for the signature:

- The eID card
  - The "Authentication" is used for operations that require authentication
  - The "Signature" key is used for operations that require non-repudiation
- The authentication private key associate to an ETK

4.3.2 Specification Signature

The specifications of the two types of signature (EID and ETK) are identical. All signatures require the use of an XML Signature in the body of the message. The XML Signature specification can be found on the website of the W3C: ***https://www.w3.org/TR/xmldsig-core/#ref-XML-exc-C14N***

The signature used by clients of eHealth Certificates Manager must use the following configuration:

- Enveloped Signature : see https://www.w3.org/TR/xmldsig-core/#def-SignatureEnveloped
- Signature method http://www.w3.org/2001/04/xmldsig-more#rsa-sha256 see https://www.ietf.org/rfc/rfc4051.txt
- Digest method : SHA256 http://www.w3.org/2001/04/xmlenc#sha256
- Use the "Exclusive" CanonicalizationMethod for the SignedInfo element: see https://www.w3.org/TR/xmldsig-core/#ref-XML-exc-C14N
- Use the "Exclusive" CanonicalizationMethod for the Reference element
- The client should add a X509Data element in the KeyInfo element containing the certificate of the key pair they used for creating the signature. The X509Data must contain the entire chain of certificates up to a certificate issued by the Root CA (excluding the Root CA itself). The certificate chain should start with the certificate chain and contain the issuers in ascending order.

  Example

  In annexes you can find an example of an xml request of the operation GetActorQualities secured by Xml Signature using EID. See file GetActorQualitiesRequest.xml

4.3.3 Process overview

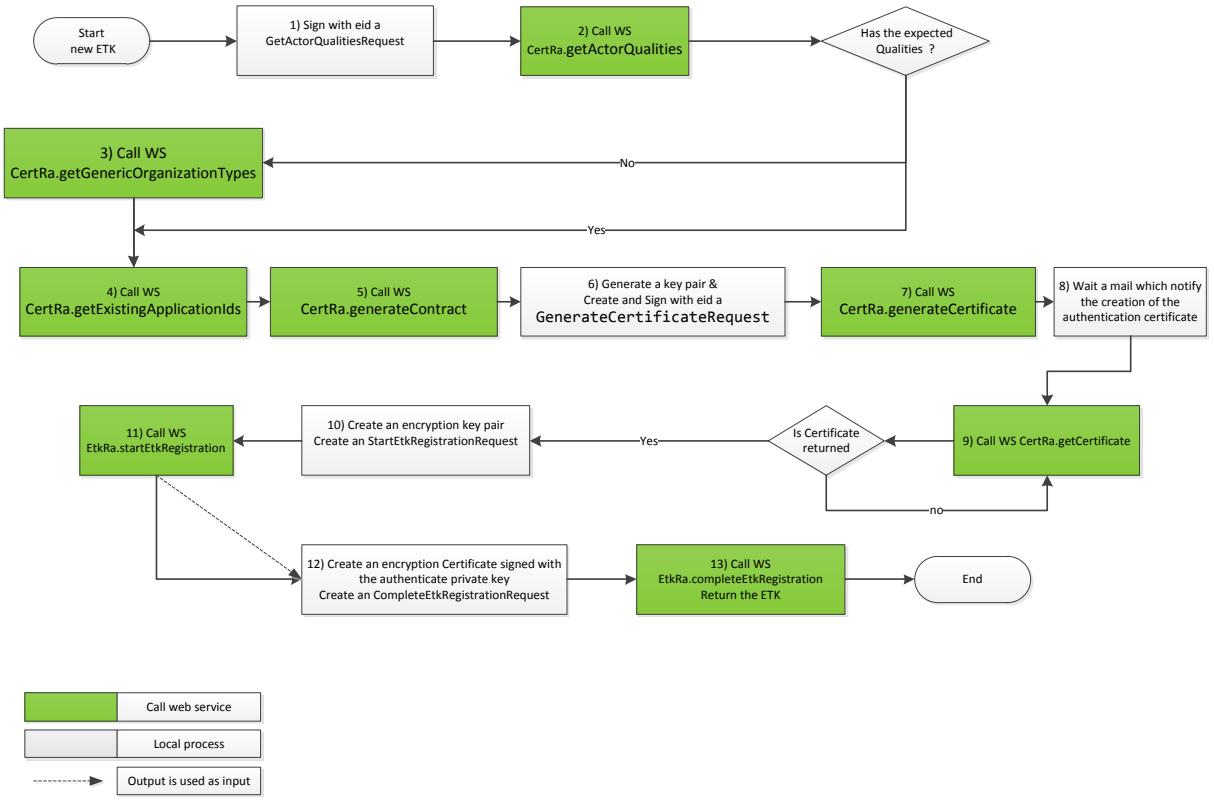
Three mains processes are involved in the ETK management

- Create a new ETK

- Renew an existing ETK
- Revoke an ETK

Each process needs to do local things like create key pair, sign some data request and call sequentially some web services provided by the eHealth platform.

4.3.4 Create a New ETK



Steps:

1) Create a GetActorQualitiesRequest, this request allows a user to get their health care qualities from the web service. It will check both the natural person qualities and the organizations for which they are authorized to manage the eHealth certificate. This request provides the NISS number of the person. The request must be signed with the “Authentication” key of the eID card of that person.
2) Send the GetActorQualitiesRequest
3) If the user does not have any authorizations, he can still request a certificate for an organization and follow the manual approval process. A call to CertRA.getGenericOrganizationTypes will return a list of organization types plus validation rules for which a certificate can be requested.
4) For an Organization’s ETK it is possible to add to the DN an application Id value, which is useful in case of multiple ETK for a same organization. To know the list of application ids already used for that organization you can call optionally the service getExistingOrganizationIds.
5) Obtain a personalized contract from the eHealth platform for requesting an eHealth certificate and encryption token by creating a GenerateContractRequest. In the request, you specify the identification details of the person or organization that will be the subject of the generated certificate; the contact data of the person(s) responsible for the certificate and the SSIN and name of the person that will sign the contract. The web service will return the generated contract that contains a personalized HTML contract that must be read and then next signed by the user.
6) Generate eHealth Certificate Request:
   a. Generate key pair
   b. Generate Distinguished Name
   c. Generate CSR

d. Add the contract returned by the step 5
e. Sign the xml request with eID card's "signature" key to guarantee the non-repudiation of the request

7) Send signed GenerateCertificateRequest document to the eHealth platform.
8) Your CSR will be treated by the eHealth platform. In the case of automatic validation, the CSR is automatically signed. In the case of the manual process, you have to wait for an eHealth operator to validate your request. In the response, you will find the "Public Key Identifier" that uniquely identifies your public key and associated certificate in Certificates Manager of the eHealth platform.
9) The certificate generation may take some time. After a while you can call CertRA.getCertificate with the "Public Key Identifier" received in step 8. If the certificate has been generated, it is returned. If not, be patient and try again.
When you received the certificate you can add the certificate + private key as a key pair with alias 'authentication' in your keystore.
10) Create an encryption key pair. Create a "StartETKRegistrationRequest" in which you provide the public key part of the new encryption key you want to register. This public key must be a 2048 bit RSA key in encoded form. This request must have an enveloped signature signed with a valid eHealth authentication certificate the user obtained in a previous step from the Cert RA web service.
11) Send your "StartETKRegistrationRequest" to EtkRA web service. The response contains a "challenge" that must be decrypted by the user. The web service has generated a random serial number and then encrypted it with the PublicEncryptionKey that was supplied by the user in the request. The user now needs to decrypt the challenge in order to prove their possession of the private key. The number that results from this decryption needs to be used in the next step.
12) Create a "CompleteETKRegistrationRequest". This request completes the ETK registration process. This is done by sending an X509 certificate that the client generated by signing with their own eHealth private authentication key. Additionally the X509 certificate must have the serial number that was decrypted from the challenge returned by the StartETKRegistration operation.
This request supplies an enveloped XML signature, the signature must be created with the same eHealth authentication certificate the user used in the StartETKRegistration step 11.
13) Send your certificate to EtkRA.completeRegistration. The certificate is now stored in the eHealth database.

4.3.4.1 Operation getActorQualities

4.3.4.1.1 Functional description

Service name	getActorQualities
WSDL	certra-proxy-v2.wsdl
Purpose	Request type that allows a user to get their health care qualities from the web service. This will check both the natural person qualities and the organizations for which the user is authorized to manage the eHealth certificate. This allows a client to determine if a certificate can be requested for the natural person or for one or more organizations.
Request	GetActorQualitiesRequest
Response	GetActorQualitiesResponse
Possible errors	The SOAP message is not correct. The signature is not correct. The NISS number of the EID card used for signer doesn't belong to the NISS referenced in the request

4.3.4.1.2 Request Example

See GetActorQualitiesRequest.xml

4.3.4.1.3 Response Example

See GetActorQualitiesResponse.xml

4.3.4.2 Operation getGenericOrganizationTypes

If the GetActorQualitiesResponse does not contain an authorization (or the authorization does not contain the organization the user represents), it is still possible to request a manual validation for an organization certificate. This "getGenericOrganizationTypes" operation allows you to get a list of organization types supported by eHealth Certificates Manager. The operation will return details of each available organization type such as the type of identifier (for example: "NIHII-HOSPITAL") , the expected pattern of the identifier (for example: "^710[0-9]{5}|719[0-9]{5}|720[0-9]{5}$" and the Check Digit Algorithm used for validating the correctness of the identifier (for example: "modulo 97" algorithm).

Personal certificates can only be validated automatically, so when the user (a health care professional in Belgium) does not receive an authorization for EntityType Natural, they should contact the eHealth Contact Center in order to verify their authorization.

4.3.4.2.1 Functional description

Service name	getGenericOrganizationTypes
WSDL	certra-proxy-v2.wsdl
Purpose	Request a certificate for an organization.
Request	Requesting the list of authorized identifier type, valid value pattern and check digit algorithm for organization that is not present in the authorization returned by "GetActorQualitiesResponse".
Response	Return the list with all possible organization types for which a certificate can be requested. The response also contains regex validation rules for the identifier (so the client can check the identifier before sending to eHealth), as well as a list of possible (optional) service usage types. See below (in the description of the eHCSR) for more information on the usage types.

Possible errors	Soap Fault with internal server error in case of backend trouble.

4.3.4.2.2 Request Example

See GetGenericOrganizationTypesRequest.xml

4.3.4.2.3 Response Example

See GetGenericOrganizationTypesResponse.xml

4.3.4.3 Operation getExistingApplicationIds

4.3.4.3.1 Functional descriptions

Service name	getExistingApplicationIds
WSDL	certra-proxy-v2.wsdl
Purpose	To know the list of the application ids being already used for a specific organization.
Request	When requesting a certificate for an organization you can create a new unique Distinguished Name DN by using an application id. This is useful when you need multiple ETKs for the same organization. Of course, you cannot use the same application id value for multiple ETK belonging to the same organization. - Type: the organization's identifier type (e.g. NIHII-PHARMACY, NIHII-HOSPITAL, CBE …) depending on the organization type (e.g. pharmacy, hospital, general organization…). - Value: the organization identifier value.
Response	Returns the list of the application ids that are already used.
Possible errors	Soap Fault with internal server error in case of backend trouble.

4.3.4.3.2 Request Example

See GetExistingApplicationIdsRequest.xml

4.3.4.3.3 Response Example

See GetExistingApplicationIdsResponse.xml

4.3.4.4 *Operation generateContract*

4.3.4.4.1 Functional descriptions

Service name	generateContract
WSDL	certra-proxy-v2.wsdl
Purpose	Generate a contract for a specific ETK request. This contract is a legal text to be read by the requestor. This contract is digitally signed with the requestor's eID card in the next operation "generateCertificate".
Request	Specify the identification details of the person or organization that will be the subject of the generated certificate, the contact data (email and phone number), the name and SSIN of the requestor.
Response	The contract between the eHealth platform and the requestor.
Possible errors	Soap Fault with internal server error in case of backend trouble.

4.3.4.4.2 Request Example

- See GenerateContractRequest_ForNaturalPerson.xml
- See GenerateContractRequest_ForOrganization.xml

4.3.4.4.3 Response Example

- See GenerateContractResponse.xml

4.3.4.5 *Operation generateCertificate*

4.3.4.5.1 Functional description

Service name	generateCertificate
WSDL	certra-proxy-v2.wsdl
Purpose	This method should be used to send a CSR for the authentication certificate part of an ETK. The sender of the request will receive an email when the certificate has been produced. The certificate can be recovered by using the returned PublicKeyIdentifier.

Request	- Contact data : email , phone number
	- CSR as a DER encoded representing an PKCS#10 Certificate Signing Request with following constraints :
	  ○ Generate Key pair : Key size = 2048
	  ○ Csr signature Algorithm = SHA2 & RSA
	  ○ The 'distinguished name' of the CSR must match the distinguished name that was returned in the "DN" element of the response of generateContract. More details about the DN format are describe in this document: ***https://www.ehealth.fgov.be/sites/default/files/en-savoir-plus/fiche/ehealthcertificatstest_fr_060214.pdf***
	- Contract: the contract received by "generateContract" operation. Be careful to copy the contract as-is in your request because the server will check if the content complies with the expected contract.
	- Signature: an xml enveloped signature signed with the "signature" key of the eID card.

Response	- PublicKeyIdentifier: The returned PublicKeyIdentifier should be kept or computed to be used when invoking the web service getCertificate to retrieve

	the certificate generated by eHealth. The PublicKeyIdentifier is the Hexa Decimal representation of the Subject Public Key Identifier of the csr. - AutomaticallyValidated: Indicates if the authorization to request the certificate was validated automatically. If not then a manual verification is required which may take several days.
Possible errors	- Wrong signature: The given CSR is not a valid PKCS10 CSR. - The given CSR does not contain a subject(=owner) - …

4.3.4.5.2 Request Example

See GenerateCertificateRequest.xml

4.3.4.5.3 Response Example

See GenerateCertificateResponse_Success.xml

4.3.4.6 Operation getCertificate

4.3.4.6.1 Functional description

Service name	getCertificate
WSDL	certra-proxy-v2.wsdl
Purpose	This service should be used to retrieve the certificate once the CA has produced it. If the certificate is not ready yet, then the client will receive a business error. If the certificate request was automatically validated, then clients can poll the operation until the certificate is returned. Otherwise, the delivery might take several days so clients should only get the certificate after they received a notification that the certificate was approved.
Request	This certificate PublicKeyIdentifier returned by the service generateCertificate described in the previous paragraph.
Response	Return the produced certificate and the certificate chain for the given PublicKeyIdentifier.
Possible errors	The certificate is not yet available.

4.3.4.6.2 Request Example

See GetCertificateRequest.xml

4.3.4.6.3 Response Example

See GetCertificateResponse.xml

4.3.4.7 Operation startEtkRegistration

4.3.4.7.1 Functional description

Service name	StartEtkRegistration
WSDL	etkra-proxy-v2.wsdl
Purpose	This method should be used to register an encryption public key for a given authentication certificate.
Request	- PublicEncryptionKey: The public key part of the new encryption key you want to register. This public key must be a 2048 bit RSA key in encoded form. - Signature: An enveloped XML signature that signs the entire request element. The signature must be created with a valid eHealth authentication certificate the user obtained in a previous step from the Cert RA web service.

Response	- Challenge: A challenge created by the server. The challenge consists of an encrypted number followed by 256 bits containing the SHA-256 hash of this serial number. The client must decrypt the challenge with the private key associated with the public key to prove that they possess the correct private key. Clients can verify that their decryption was successful if the SHA-256 hash of the resulting data is equal to the last 256 bits of the challenge.
Possible errors	- The encryption token is requested for an authentication certificate not issued by the CA of eHealth. This is not allowed. - The encryption token is requested for an authentication certificate revoked by the CA of eHealth. This is not allowed. - The encryption token is requested for an authentication certificate that is expired. This is not allowed - The encryption token is requested for an authentication certificate that is not yet valid. This is not allowed.

4.3.4.7.2 Request Example

See StartETKRegistrationRequest.xml

4.3.4.7.3 Response Example

See StartETKRegistrationResponse.xml

4.3.4.8 Create an encryption Certificates

Purpose	To generate the encryption certificate. This certificate must have as serial number the value returned by the previous step.
Encryption certificate	Subject: Same as the authentication certificate Issuer: Same as the authentication certificate Certificate Revocation List Crl: null Key usage: key encipherment and data encipherment. Not Before: same as the authentication certificate Not Before field Not After : same as the authentication certificate Not After field Signature Algorithm: use the same sign algorithm which was used to generate the authentication certificate (currently SHA256withRSA)
Output	The certificate of the authentication.

4.3.4.9 Method CompleteEtkRegistration

4.3.4.9.1 Functional description

Service name	CompleteEtkRegistration
WSDL	etkra-proxy-v2.wsdl
Purpose	This service should be used to register the certificate generated in the previous step in the eHealth database.
Request	- ToBeRegistered : the encoded encryption certificate generated by the previous step - Signature : Enveloped xml signature created with the eHealth authentication private key
Response	- Information about the completion of the registration. - The ETK
Possible errors	- Signature does not belong to the expected authentication certificate - Wrong signature - The encryption token is requested for an expired authentication certificate. This is not allowed - The encryption token is requested for an authentication certificate, revoked by the CA of eHealth. This is not allowed - …

4.3.4.9.2 Input parameters

See CompleteETKRegistrationRequest.xml

4.3.4.9.3 Output parameters

See CompleteETKRegistrationResponse.xml

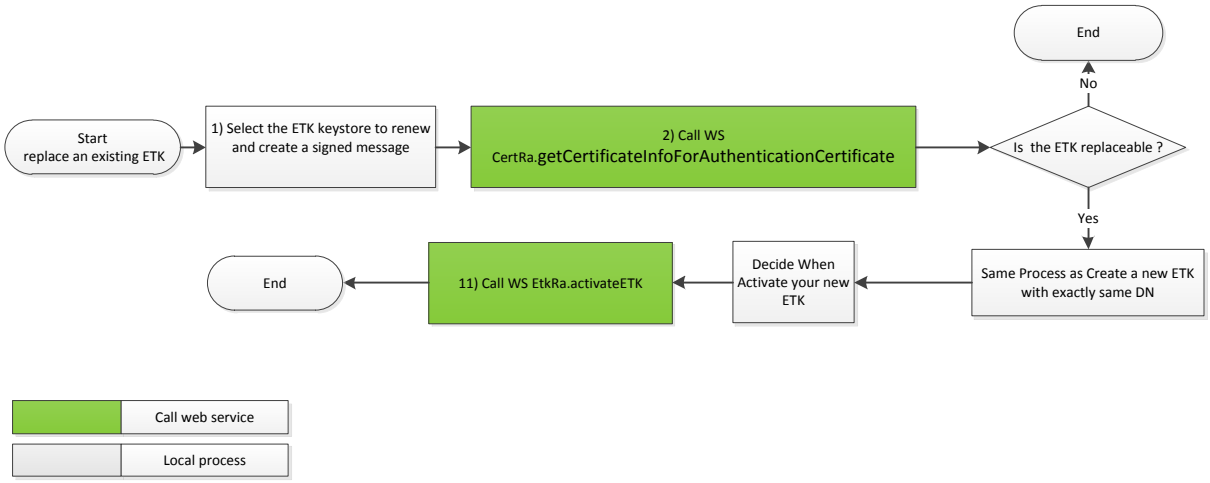
4.3.5 Replacement of an existing ETK

As of a period of about 3 months before expiration, an existing ETK can be replaced. The server defines the exact replacement period.

In order to know the expiration date of an ETK use the certRa operation getCertificateInfoForAuthenticationCertificate.

If the status of your ETK is “Replaceable == true” you can create a new certificate with the same DN as the ETK you want to replace. You create it with the same process explain in “Create a New ETK” chapter.

When the new ETK is created, you have to call the operation on etkRa web service “activateETK” before it is active for use.



4.3.5.1 Operation getCertificateInfoForAuthenticationCertificate

4.3.5.1.1 Functional description

Service name	getCertificateInfoForAuthenticationCertificate
WSDL	certra-proxy-v2.wsdl
Purpose	This service should be used to get the status of your certificate. The certificate you are requesting is determined based on the certificate used for signing. This allows clients, not registered in the authentic sources still to access information of their certificate.
Request	- A GetCertificateInfoForAuthenticationCertificateRequest - Enveloped Xml Signature done with the authentication private key of the ETK keystore to replace.
Response	The response will contain - General information about the certificate - Status of the certificate - Whether or not the certificate can be replaced (Replaceable == true if the certificate can be replaced), - The ReplacementPeriodStartDate (the start date of the replacement period) and the ValidNotAfter date (= expiration date).
Possible errors	- The signature verification fail for unknown raison - The etk belonging to the request could not be found (could append by example when you give etk from Integration environment to Acceptation environment)

4.3.5.1.2 Request Example

See GetCertificateInfoForAuthenticationCertificateRequest.xml

4.3.5.1.3 Response Example

See GetCertificateInfoForAuthenticationCertificateResponse.xml

4.3.5.2 *Method ActivateETK*

4.3.5.2.1 Functional description

Service name	ActivateETK
WSDL	etkra-proxy-v2.wsdl
Purpose	This service should be used to activate the encryption key pair (ETK) as a last step in a renewal process. In the case of a renewal, two authentication certificates can be temporarily activate at the same time. However, only one encryption certificate may be active at any given point in time. The certificate pair associated with the valid encryption certificate is the one that will be returned by the getEtk service. When registering the encryption certificate, the new certificate gets the status 'WAIT_FOR_ACTIVATION'. After this ActivateETK operation the newly created encryption certificate will have the 'VALID_ACTIVE' status. EtkDepot will return the old encryption key until the operation ActivateEtk is called. After activation, the old one will be considered expired and EtkDepot will return the new key.
Request	- A xml enveloped signature done with the authentication private key of the ETK to activate
Response	- The status of the completion
Possible errors	- The etk to activate is already actif.

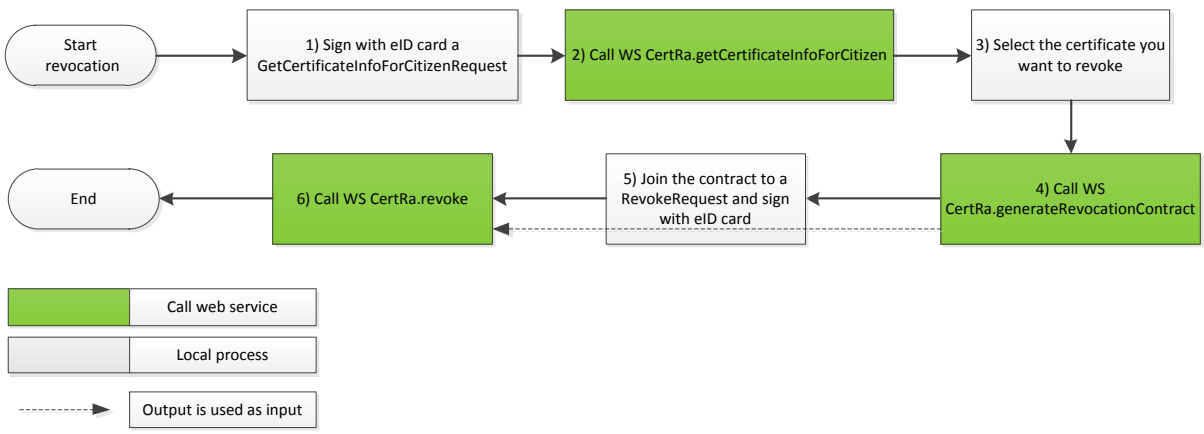
4.3.5.2.2 Request example

See ActivateETKRequest.xml

4.3.5.2.3 Response example

See ActivateETKResponse.xml

4.3.6 Revoke an ETK



4.3.6.1 *Operation GetCertificateInfoForCitizenRequest*

4.3.6.2 *Functional description*

Service name	getCertificateInfoForCitizen
WSDL	certra-proxy-v2.wsdl
Purpose	Request for obtaining information about all the eHealth certificates and encryption tokens that can be managed by the natural person that signed the request with their Belgian eID.
Request	An enveloped XML signature that signs the entire request. It must be created by the "Authentication" key of a Belgian eID. The service will extract the details of the person from the associated certificate in order to determine the certificates this person can view and manage.
Response	Contains information about all the eHealth certificates and encryption tokens this person can view and manage. This includes all of their personal certificates associated with their SSIN, but also all the certificates of organizations for which they possess a mandate according to the authentic eHealth sources.
Possible errors	Wrong signature

4.3.6.2.1 Request example

See GetCertificateInfoForCitizenRequest.xml

4.3.6.2.2 Response example

See GetCertificateInfoForCitizenResponse.xml

4.3.6.3 *Operation generateRevocationContract*

4.3.6.3.1 Functional description

Service name	generateRevocationContract
WSDL	certra-proxy-v2.wsdl
Purpose	Operation for requesting a revocation contract generated by the server. The output of this operation is a revocation contract that must be read and signed by the user. This revocation contract must be included as part of the revocation request.
Request	- PublicKeyIdentifier value of the ETK to be revoked - Name, Firstname, NISS of the revoker - The reason for the revoking
Response	- The contract generated by the server. Inside the XML contract are HTML contents that must be presented to the user before they sign the revocation request.
Possible errors	- Missing value in the request

4.3.6.3.2 Request Example

See GenerateRevocationContractRequest.xml

4.3.6.3.3 Response Example

See GenerateRevocationContractResponse.xml

4.3.6.4 Operation revoke

4.3.6.4.1 Functional description

Service name	Revoke
WSDL	certra-proxy-v2.wsdl
Purpose	Request for revoking an eHealth authentication certificate and its associated Encryption Token.
Request	The request must contain a XML contract that was obtained from the generateRevocationContract operation. Enveloped XML signature that signs the entire request element. The signature must be created with the "Signature" key of a valid Belgian eID and the signer of this request must be a person that is allowed to revoke the certificate that is identified by the PublicKeyIdentifier element.
Response	The certificate and encryption token are revoked successfully if this response contains no error message in the status of the response.

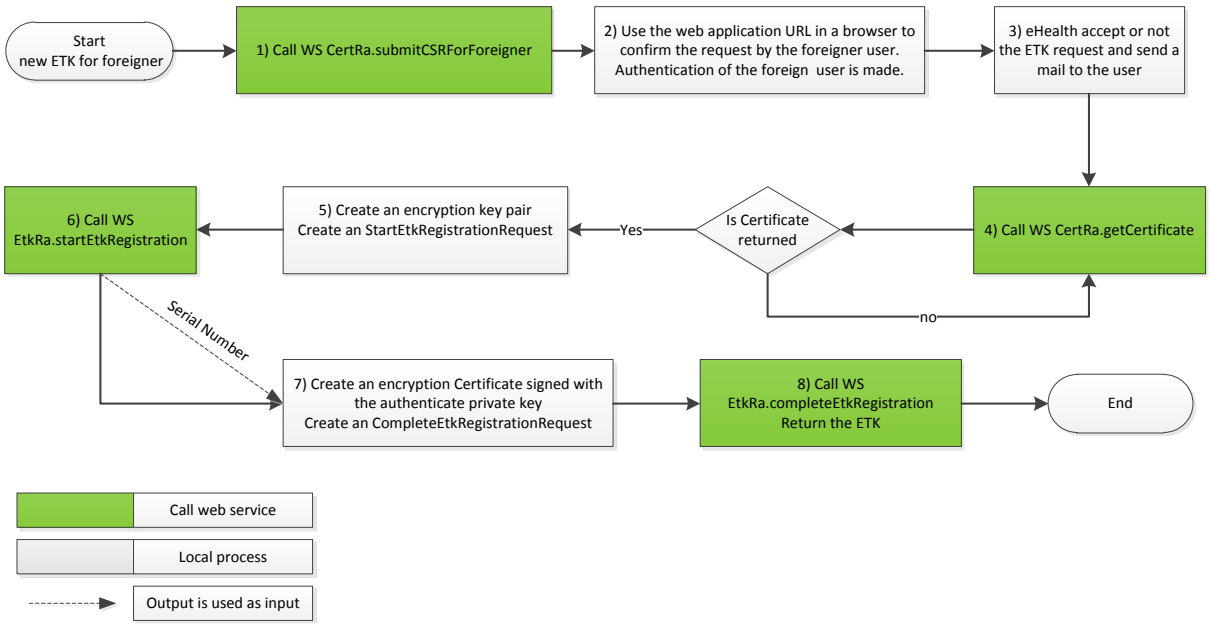
4.3.6.4.2 Request example

See RevokeRequest.xml

4.3.6.4.3 Response example

See RevokeResponse.xml

4.3.7 Create a New ETK for foreigner



The operations 4 -> 8 are already explained in the chapter “Create a New ETK”.

4.3.7.1 Operation submitCSRForForeigner

Service name	submitCSRForForeigner
WSDL	certra-proxy-v2.wsdl
Purpose	Request an ETK for a foreign person. This request should only be used by foreign persons, not residing in Belgium and thus do not have a SSIN number. This request will only "submit" the CSR, there is no certificate generation until the foreigner has confirmed his identity using the eHealth IDP The response supplies a web application URL. The foreign user who requested the ETK must open a browser with the URL he received. The application invites the user to authenticate himself through the eHealth IDP. The user has to confirm his request. When confirmed, the user will receive a mail to achieve his request.
Request	- The SSIN BIs number, firstname , name of the foreign person - Contact data , mail, phone - CSR with DN like CN=SSIN=85210123617,OU=eHealth-platform Belgium,OU=DUPONT DUPOND,OU=SSIN=85210123617,O=Federal Government,C=BE
Response	- A web application URL - Expiration date of the URL

4.3.7.1.1 Request example

See SubmitCSRForForeignerRequest.xml

4.3.7.1.2 Response example

See SubmitCSRForForeignerResponse.xml

5. Risks and security

5.1 Security

5.1.1 Business security

- In case the development adds an additional use case based on an existing integration, the eHealth platform must be informed at least one month in advance with a detailed estimate of the expected load. This will ensure an effective capacity management.
- In case of technical issues on the web service, the partner may obtain support from the contact center (See chapter 3).

In case the eHealth platform finds a bug or vulnerability in its software, the partner is advised to update his application with the newest version of the software within 10 business days.

In case the partner finds a bug or vulnerability in the software or web service that eHealth delivered, he is obliged to contact and inform eHealth immediately and he is not allowed to publish this bug or vulnerability in any case.

5.1.2 Web service

- Messages are sent securely over a one-way TLS connection.
- No WS-Security is present, but a digital XML Signature using either a Belgian eID or an eHealth Certificate must sign the service operations, where the author of the message needs to be verified.
- Please refer to the paragraph 4.3 (Security of the web service by XML Signature) for more information.

5.1.3 The use of username, password and token

- The username, password and token are strictly personal and not allowed to transfer.
- Every user takes care of his username, password and token. He is forced to confidentiality of them. Until activation, every user is also responsible of every use, including the use by a third party.

6. Test and release procedure

6.1 Procedure

This chapter explains the procedures for testing and releasing an application in acceptation or production.

6.1.1 Initiation

If you intend to use the service of the eHealth platform, please contact ***info@ehealth.fgov.be***. The Project department will provide you with the necessary information and mandatory documents.

6.1.2 Development and test procedure

You have to develop a client in order to connect to our web service. Most of the required integration info to integrate is published in the technical library on the portal of the eHealth platform.

In some cases, the eHealth platform provides you with a mock-up service or test cases in order for you to test your client before releasing it in the acceptance environment.

6.1.3 Release procedure

When development tests are successful, you can request to access the acceptance environment of the eHealth platform.

From this moment, you start integration and acceptance tests. The eHealth platform suggests testing during minimum one month.

After successful acceptance tests, the partner sends his test and performance results with a sample of “eHealth request” and “eHealth answer” by email to the point of contact at the eHealth platform.

Then the eHealth platform and the partner agree on a release date. The eHealth platform prepares the connection to the production environment and provides the partner with the necessary information. During the release day, the partner provides the eHealth platform with feedback on the test and performance tests.

For further information and instructions, please contact: ***integration-support@ehealth.fgov.be***.

6.1.4 Operational follow-up

Once in production, the partner using the service of the eHealth platform for one of his applications will always test first in the acceptance environment before releasing any adaptations of its application in production. In addition, he will inform the eHealth platform on the progress and test period.

6.2 Test cases

- The eHealth platform recommends performing tests for the entire web services described in the paragraph 4.3 (Process overview).
- In addition, the organization should also run negative test cases.

7. Error and failure messages

Error code	Message
CSR_NOT_VALID	The given CSR is not a valid PKCS10 CSR.
CSR_SIGNATURE_NOT_VALID	The given CSR has an invalid signature.
CSR_INVALID_NO_SUBJECT_PRESENT	The given CSR does not contain a subject (=owner).
CSR_INVALID_PUBLIC_KEY_INVALID	The given CSR contains an invalid public key.
CSR__CSR_DELIVERY_TO_CA_FAILED__SIGNING_FAILED	The signing of the CSR in order to pass it to the CA, failed.
CSR__CSR_DELIVERY_TO_CA_FAILED__INVALID_SIGN_KEY_PWD	The CSR could not be signed because the password for the signature key is invalid.
CSR__INPUT_NOT_PARSEBLE_INTO_CSR	The input could not be parsed in to a CSR.
CSR__INPUT_NOT_UNMARSHALLABLE_INTO_EHCSR	The input could not be unmarshalled into an eHealthCSR.
CSR_A_VALID_AUTH_CERT_ALREADY_EXISTS	A valid authentication certificate that is not expiring soon already exists for the user who is requesting a new eHealth authentication certificate.
CSR_AN_APPROVED_CSR_FOR_GIVEN_DN_IS_IN_PROCESS_OF_BEING_CERTIFIED	There is already an approved CSR in process of being certified for this DN.
CSR_INVALID_DN_INVALID	The distinguished name in the CSR is not according to the eHealth policy.
CSR_PUB_KEY_NOT_UNIQUE	The CSR is not unique.
CSR_ALREADY_VALIDATED	The CSR was already validated.
CSR_ALREADY_VALIDATED	The CSR was already validated.

CERT_NOT_YET_DELIVERED	Your certificate is not yet delivered. Try to get it later.
AUTH_CERT_NOT_YET_VALID	The authentication certificate has been delivered but is not yet valid. The certificate is valid from {0}.
ETK_REQUESTED_FOR_AUTH_CERT_NOT_ISSUED_BY_EHEALTH_CA	The encryption token is requested for an authentication certificate that is not issued by the CA of eHealth. This is not allowed.
ETK_REQUESTED_FOR_AUTH_CERT_REVOKED	The encryption token is requested for an authentication certificate that is revoked by the CA of eHealth. This is not allowed.
ETK_REQUESTED_FOR_AUTH_CERT_EXPIRED	The encryption token is requested for an authentication certificate that is expired. This is not allowed.
ETK_REQUESTED_FOR_AUTH_CERT_NOT_YET_VALID	The encryption token is requested for an authentication certificate that is not yet valid. This is not allowed.
REQUEST_HAS_NO_VALID_CMS_FORMAT	The request has no eHealth authorized CMS format. Please contact the helpdesk.
AUTH_CERT_NOT_FOUND	An authentication certificate corresponding to the given ToBeRegistered is not found in the ETK DB.
PUBLIC_KEY_NOT_YET_REGISTERED	The public key for the given ToBeRegistered is not yet registered.
NO_VALID_AUTH_CERT_FOUND	No authentication certificate, that has status valid, has been found in the ETK RA.

AUTH_CERT_NOT_VALID	The given authentication certificate is not valid since it is not identical to the one found that is registered.
VALID_ETK_ALREADY_EXISTS	An ETK already exists for the given public key.
RENEW_ACTIVATION_ETK_ALREADY_ACTIF	The ETK to activate is already active.
RENEW_REQUEST_NOT_YET_IN_EXPIRATION_PERIOD	Not yet in the expiration period to perform a renewal request
EHCSR__SIGNED_DATA_HAS_NO_CONTENT	The eHealth CSR is a CMS SignedData without content. It should contain an XML containing the CSR, contract and contact data.