Herkenning

Agreements System eRecognition

Interface DV HM

Version 1.4

1 Document information

2 Colophon

Author	Status
Management Organization Agreements System eRecognition	Final
Project	**Date**
Agreements System eRecognition	April 28, 2012
Organization	**Classification**
ICTU	Public
Title	**Version**
Agreements System eRecognition - Interface DV HM	1.4

3 History

Date	Version	Change	Status	Processed by
10/03/29	0.8def		Tbv pilot implementations	Project
09/06/10	1.0	Shape changes and implement several RFCs	To approval	Project
10/12/17	1.0a	RFCs processed in accordance with decision Core 6 December	Final	Project
06/11/17	1.1	RFCs processed in accordance with decision Core Team May 31	Final	Project
11/12/11	1.2	RFCs processed in accordance with decision Core Team October 11	Final	Project
11/12/23	1.3	RFCs processed in accordance with decision core 13 December	Final	Project
01/12/05	1.3a	Corrections to RFC102, RFC105 and RFC124 implemented	Final	Project
28/04/12	1.4	RFCs processed in accordance with decision core team March 20	Final	Management Organisation

4 Distribution List

Date	Distribution	Presentation	Version
	Core team, eherkenning		

5 Approval

Date	Name

20/03/12 All RFCs for version 1.4 approved by core team 1.4

Ministerie van Economische Zaken eHerkenning

Content

Ministerie van Economische Zaken eHerkenning

1 Introduction

This document is part of the Agreements System eRecognition. It can not be separated seen from the other documents of the Agreements System. For a general introduction to, and An overview of all documents within eRecognition the reader of this document recommended the document first [eRecognition - General introduction] to read.

1.1 Purpose and target audience of this document

This document describes the interface between the service and the eRegocnition Broker. The is intended for anyone who needs the most detailed technical specifications.

1.2 Structure

The remainder of this paper is as follows. Chapter 2 describes the general requirements for to the coupling plane. Chapter 3 describes the technical information security requirements. In chapter 4 , the error handling described. Chapter 5 contains the message envelope. Chapter 6 describes the service catalog and chapter 8 the metadata. Chapter 9 provides an overview Data of the used data elements. The paper concludes with several appendices from which

the text refers.

1.3 Glossary

Within eRecognition a glossary used. See the appendix in document [eRecognition - General introduction]. In this list are singular forms of self-nouns and verbs included. References in this document to the verb form of this nouns is applied, this has the same meaning as defined nouns. The same is also true: where in the document the self-name word-form of a verb is used, it has the same meaning as defined verb.

1.4 Terminology

For the sake of readability of the text is everywhere 'he wrote,' he or she 'meant.

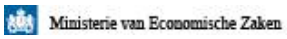
The words "MUST", "MUST NOT", "SHOULD", "SHOULD NOT" and "MAY" in this document should be interpreted similar to their English equivalents ("MUST", "MUST NOT / SHALL NOT "," SHOULD "," SHOULD NOT ", and" MAY ") as described in *RFC 2119* (http://www.ietf.org/rfc/rfc2119.txt). Where these exact terms are intended to be they main-letters. The meaning of these words:

- MUST: a prerequisite

- MAY NOT: an absolute prohibition
- SHOULD: strong desire, unless there are valid reasons in particular cases to deviate
- SHOULD NOT: undesirable, unless there is valid reason in the particular case to let
- MAY: a free choice, an option

1.5 Typography

In the more technical parts of the documentation, the words "MUST", "MUST NOT", "SHOULD", "SHOULD NOT" and "MAY" in block capitals listed.



2 General requirements

The interface described in this document is used to implement the use case "authentication acting service recipient" and MUST (except section 2.2.2.2 and 5.3) by any eRegocnition Broker implemented and offered to its users, the service [1] .

2.1 Alternative interfaces and / or binding

A eRegocnition Broker MAG other interfaces and / or bindings and implement offer than are described in this document, but MUST then be used to demonstrate that the management organization security and functionality of at least the same standard as those of the document in this described interface and the internal pseudonym is not shared.

Under the same level of security is understood that:

- use of asymmetric encryption based on a PKI
- 2048 bit key lengths of at least enforced
- minimum SHA256 hashing algorithm is used as
- measures are taken against replay attacks
- Messages / statements are perishable

MUST also use the eRegocnition Broker for a description of all the alternative interface to the service to provide data delivery to the management organization.

2.2 Use of SAML 2.0

This interface uses SAML 2.0. A service provider is seen as a service, provider. A eRegocnition Broker is seen as an identity provider.

67 2.2.1 SAML Web Browser SSO Profile

68 For in this document interface MUST SAML Web Browser SSO Profile are
69 is used. Interrogate for the optional use of attributes is made of an extension.

70 2.2.2 Bindings

71 Within SAML, different bindings are used to convey messages
72 between the parties.

1 This is going to lock-in and draw up against so-called middleware vendors to build generic pieces of software which can be used in all eRegocnition Brokers.

73 *2.2.2.1 HTTP POST Binding*

74 For in this document any interface MUST eRegocnition Broker the HTTP POST
75 bond [two] implementing and offering to its customers, the service providers. A
76 eRegocnition Broker MAY also implement other bindings and services.

77 *2.2.2.2 Alternative binding*

78 In order to meet service providers in this document, an alternative, optional
79 binding described. This is that during the binding assay has been used for the implementations
80 interface between providers and eRegocnition Broker. MAY eRegocnition Brokers this
81 Implementing and binding offer.

82 The above-described alternative bond is a combination of the HTTP Redirect [3] and HTTP Artifact [4]
83 binding use, the question is asked with an HTTP Redirect binding and
84 answer is given with an HTTP Artifact binding.

85 The Recommendations concerning the compilation of an artifact (Section 3.6.4 of the
86 SAML 2.0 specification bond [5]) MUST be implemented.

87 2.2.3 Relay State

88 Each SAML request message MUST relay state data. The response to a SAML request with
89 State Relay Relay State this data MUST also contain data. The contents of the Relay State MUST NOT
90 greater than 80 bytes and MUST be the party that State Relay creates protected against
91 changes.

92 2.2.4 Namespace aliases

93 Needless to say, it is noted that the parties are free to choose which aliases are
94 used for the abbreviation of namespaces in tags.

95 *2.3 HTTP Headers*

96 For all content to a browser of an acting individual is sent
97 MUST use the following HTTP headers:
98 • Cache-Control with value "no-cache, no-store "

99 • Pragma with value "no-cache"

---

[2] urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST
[3] urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect
[4] urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact
[5] http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf

100 2.4 Optional elements and attributes

101 Optional elements and attributes MAY be included in messages. These elements
102 MUST then be filled in accordance with specifications and therefore MUST NOT be empty.

103 2.5 Sessions

104 Participants MUST NOT a session of the acting individual track longer
105 takes than is strictly necessary for the execution of the use case.

106 Exclusively serving Single Sign-On , and then only serving proof-of-concept/pilot-achtige situations
107 MAY one eRegocnition Broker, authentication and authorization service registry for a session
108 maximum duration of 60 minutes tracking. In this session MUST include the following information
109 are maintained:

110 • one eRegocnition Broker holds user preferences (chosen authentication service and
111 elected authorization register) at.

112 • an authentication service, the identity of the natural person acting for. On
113 Based on this session, the authentication service MUST immediately a new statement on the
114 authentication issue.

115 • an authorization registry holds user preferences (chosen to represent
116 interested) in.

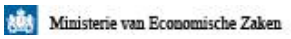
117 It is obvious that service obv eRecognition obtained from the Declaration on the
118 Authentication of holding a session with the acting individual start.

119 2.5.1 Logout

120 Participants MUST an acting individual the opportunity to log off.
121 Service MUST take action itself after a certain amount of inactivity the
122 natural person acting automatically log out.

123 2.6 Version control

124 Because different versions of the Agreements System at interface level from each other
125 can be distinguished MUST be used versioning of messages
126 implemented in the coupling plane. Because the SAML 2.0 messages for this field does not
127 is available and it is not desirable for this extension in the message MUST be used
128 participants the URL that SAML messages can be presented in the published
129 metadata linked to a version of the Agreements System. For two different versions of
130 the Agreements System so DO NOT use the same URL. Eg.



131 http://www.deelnemer.nl/SAML-endpoint/v1.0/

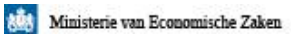
132 See also Chapter 8.

133 2.7 Language Preference

134 Within eRecognition it is possible to change the language preference of the acting individual
135 to pass so that dialogue can be conducted in this language. Because the SAML 2.0 messages
136 this no field is available and it is not desirable for this extension in messages
137 MAG EherkenningPreferredLanguage use as a query variable in the URL or POST variable
138 are given. See also section 9.4.

139 2.8 Character set and encoding

140 For all messages MUST be used for the Unicode character set in UTF-8
141 encoding.



3 Technical information security requirements

This chapter describes the requirements that the measures under Information taken are implemented.

3.1 Security Connection

For all connections between two systems must meet the following requirements:

- All connections MUST use SSL 3.0 or TLS.
- In the WS-I basic security profile [6] are single cipher suites discouraged. This MAY NOT be used for SSL or TLS.
- For SSL or TLS SHOULD a participant PKIoverheid G2 SSL certificate MUST be used. If no PKIoverheid G2 SSL certificate is used MUST be a participant EV SSL certificate with a key length of at least 2048 bits. The (extended) key usage of the certificate used MUST use SSL / TLS to allow ..
- For SSL or TLS SHOULD a service an EV SSL certificate with a key length of at least 2048 bits MUST be used. A service provider MUST use an SSL certificate with a key length of at least 1024 bits.

NB The use of SSL certificates other than PKIoverheid G2 select certificates will eventually not more are allowed.

3.2 Signing messages

For authenticity, integrity and non-repudiation SHOULD ensure that every message is described in this document are provided with an electronic signature sender of the message. The recipient of a message MUST all electronic signatures in the message to validate the message before further processing.

For the generation of an electronic signature, the following requirements apply:

- The electronic signature MUST be put across the message with the Enveloped Signature Transform [7] .
- Canonicalization MUST be under the exclusive c14n method [8] .
- Digests MUST be calculated using the SHA-256 algorithm.

6 http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html

7 http://www.w3.org/2000/09/xmldsig # enveloped-signature

8 http://www.w3.org/TR/xml-exc-c14n/

- The Signature Value MUST be calculated using the RSA-SHA256 algorithm.
- For signing messages MUST be a participant PKIoverheid G2 certificate with a key length of at least 2048 bits. The (extended) usage of key MUST use the certificate used for signing allow ..
- For signing messages MUST be a service provider PKIoverheid G1

certificates with a key length of at least 1024 bits or PKIoverheid G2 certificate with a key length of at least 2048 bits.

- The signature MUST be a keyinfo element, containing a KeyName. The KeyName MUST match a KeyName specified in the metadata of the sender for the respective roller. The signature MUST NOT contain other keyinfo (such as X509Data).

3.3 Use PKIoverheid data

In order to make good use of PKIoverheid receivers of messages MUST conform the requirements for the receiving party as described in the PKIoverheid Program Requirements. Here, the following aspects are important:

- The Tribe Certificate "State of the Netherlands Root CA - G2" [9] confidence.
- All Domain Certificates and all CSP certificates [10] know and trust.
- The PKIoverheid CRL [11] regularly consult.

3.4 Synchronization system clocks

The network is working with Coordinated Universal Time, UTC called. All timestamps in the messages are formatted in the form yyyy-MM-DDThh: mm: SSZ. (The T (time) and Z (Zulu) are fixed values).

Each participant MUST through synchronization with a reliable source of accurate time the deviation of the system time less than or equal to 2 seconds let.

[9] See http://www.pkioverheid.nl

[10] See http://www.pkioverheid.nl

[11] See http://crl.pkioverheid.nl

4 Error Handling

This section describes how errors MUST be handled in the network, so that the users and participants to be informed satisfaction and operated. At error handling is on the principle that errors are handled where the error within the network.

4.1 Cancel

In the normal message flow, the acting individual kill the process by clicks on the "Cancel" button. In the chapter Use Cases are the scenarios explicit where an end user on the "Cancel" button to print.

If the user cancels the participant MUST user automatically return to the dispatcher, with a valid SAML message with a Status Code Value = AuthnFailed. A sender MAG a Status Message Record (eg "Authentication Cancelled").

4.2 Incorrectly formatted messages

When a Party incorrectly formatted message MUST be the participant of the process immediately abort. Under an incorrectly formatted message (among others) means invalid XML, no SAML, invalid signature, digest invalid and incorrect version SAML.

The receiver MUST use the acting individual reporting that a fatal error has occurred.

The recipient MUST error in the research and take the sender MUST inform state that an error has occurred. The sender MUST error in the research process.

4.3 Functional inadequate messages

There may be a question to one participant stated, functionally invalid. For instance because a wrong issuer is staged. If a party receives a message that functional specification is not according to the receiving party MUST kill the process. The receiving party MUST the user is automatically returned to the sender with a valid SAML message with a SAML Requester status code, and a first-level status code RequestDenied. The receiver MUST in the SAML status message indicating what the problem is (for example, "missing or unknow issuer").

There may be a question to one participant stated, where a participant does not answer

has. For example, because a confidence level is requested that an AD can not meet. If a party receives a message that is not functional by the party to be handled than the receiving party MUST kill the process. The receiving party MUST the user is automatically returned to the sender with a valid SAML message, with a SAML responder status code, and a first-level status code RequestUnsupported. The receiver MUST in the SAML status message indicating what the problem is (eg "level not supported ").

If a party receives a message that is too old (instant issue), or if this message does not expect at that time (unknown inresponseto) in the process, then the receiving party MUST abort. The receiving party MUST user automatically return to the dispatcher, with a valid SAML message with a SAML responder status code, and a first level status code RequestDenied. The receiver MUST in the SAML status message indicating what the problem (eg "invalid message").

235 The sender MUST take these errors in research. The sender notifies the user of the
236 aware of the reason for the failure. The sender MAY user an alternative.

237 5 Message Specifications

238 This chapter describes the messages of the described interface.

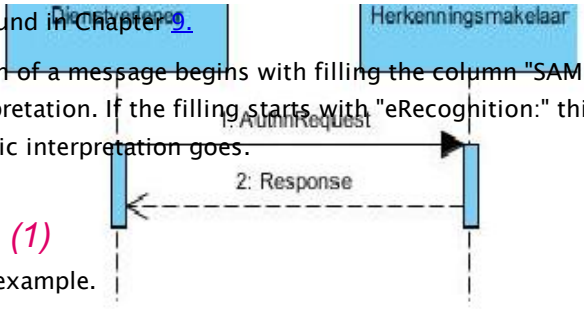
239 The use case "acting authentication service recipient" is described in the interface
240 filled with SAML 2.0 and AuthnRequest and Response.

Figure 1: Sequence diagram DV HM

241 The specific content of these messages is described below. Detailed information about the

field contents can be found in Chapter 9.

Where, in the description of a message begins with filling the column "SAML", this means that This is a standard interpretation. If the filling starts with "eRecognition:" this means that it eRecognition one specific interpretation goes.



AuthnRequest 5.1 (1)

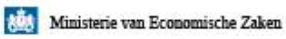
See section 11.1 for an example.

Data element	Delivering
@ Id	SAML: Unique feature of message
@ Version	SAML: Version of the SAML protocol. The value MUST be "2.0" are.
@ Instant Issue	SAML: Time the message was created
@ Destination	SAML: URL eRegocnition Broker which the message is offered. MUST match the metadata of the eRegocnition Broker.
@ Consent	eRecognition: MAY NOT be included.
@ ForceAuthn	eRecognition: The value MUST be "true" unless Single Sign-On is used (see Section 2.5 for limitations).
@ IsPassive	eRecognition: MAY NOT be included.

Data element	Delivering
@ Protocol Binding	SAML: MUST NOT be included because Assertion Consumer Service Index is within eRecognition prescribed.
@ Assertion Consumer Service Index	eRecognition: This attribute element indicates which URL the HM the answer for the DV controls. This index refers to a url the service metadata. The HM and DV must, for the use of this field, specify indices urls. The value of Assertion Consumer Service Index MUST correspond to an index of the assertion consumer service the metadata of the service.
@ AssertionConsumerServiceURL	SAML: MUST NOT be included because Assertion Consumer Service Index is within eRecognition prescribed.
@ AttributeConsumingServiceX	eRecognition: MUST be a Service ID (in the short format) contain. See section 9.2.2.
@ ProviderName	eRecognition: MAY be included but MUST be ignored by the eRegocnition Broker [12] .
Issuer	eRecognition: MUST contain the EntityID of the service. See Section 9.2.3. The attributes NameQualifier, SPNameQualifier, Format and SPProviderID MUST NOT be included.
Signature	eRecognition: MUST include the electronic signature of the service over the whole message. See section 3.2 for specific requirements.
Extensions	

	eRecognition: Optional element. If additional attributes MAY be out yet here RequestedAttributes element are included. In the RequestedAttributes element MUST only attributes be included which are included in the catalog attribute (See Chapter 7) . These attributes MUST be included as Name of RequestedAttribute. Other XML attributes MUST NOT be included. Other elements MUST NOT be included. A recognition agent MAY ignore this field, but it MUST message as able-bodied accept.
Subject	eRecognition: MUST NOT be included
NameIDPolicy	eRecognition: MAY NOT be included.
Conditions	eRecognition: MAY NOT be included.

[12] The eRegocnition Broker also uses the element ProviderName but complements it in another way.



Data element	Delivering
RequestedAuthnContext	eRecognition: an attribute MUST Comparison = 'minimum' and AuthnContextClassRef element containing recorded by the service required minimum confidence level contain. See section 9.2.1.
Scoping	eRecognition: MUST NOT be included

248 5.2 Response (2)

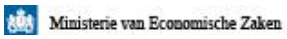
249 See section 11.2 for an example.

Data element	Delivering
@ Id	SAML: Unique feature of the message.
@ InResponseTo	SAML: Unique feature of the AuthnRequest which this Response message is the answer.
@ Version	SAML: Version of the SAML protocol. The value MUST be "2.0" are.
@ Instant Issue	SAML: Time the message was created.
@ Destination	SAML: URL of the service to which the message is offered. MUST match the metadata of the provider.
@ Consent	eRecognition: MAY NOT be included.
Issuer	eRecognition: MUST be EntityID of eRegocnition Broker contain. See section 9.2.3. The attributes NameQualifier, SPNameQualifier, Format and SPProviderID MUST NOT be included.
Signature	eRecognition: MUST include the electronic signature of the eRegocnition Broker the entire message. See Section 3.2 for specific requirements.
Extensions	eRecognition: MAY NOT be included.
Status	eRecognition: MUST contain an element Status code containing

	the status of the authentication. In case of cancellation or error This element SHALL be filled with the value AuthnFailed. See the descriptions in chapter 4. Status Detail MAY NOT be included.
Assertion	eRecognition: MUST be a statement about the authentication include a statement on the competence contain (see next section).

250 5.2.1 Declaration on Authentication

Data element		Delivering
Assertion	@ Version	SAML: Version of the SAML protocol. The value MUST be "2.0" are.
	@ Id	SAML: Unique reference to the assertion
	@ Instant Issue	SAML: Time at which the assertion is created
	Issuer	eRecognition: MUST be EntityID of eRegocnition Broker contain. See section 9.2.3. The attributes NameQualifier, SPNameQualifier, Format and SPProviderID MUST NOT be included.
	Signature	eRecognition: MUST NOT be included
	Subject	eRecognition: MUST be a NameID element containing the specific pseudonym of the acting individual. See Section 9.2.4.1. The NameID element MUST be a NameQualifier attribute, which is filled with the EntityID the authorization registry. A SubjectConfirmation element that conforms to the Web Browser SSO profile MUST be included. Other SubjectConfirmation or SubjectConfirmationData elements MUST NOT be included.
	Conditions	eRecognition: MUST be included. The attributes notBefore NotOnOrAfter and must be filled with, respectively, the time of issuance of the assertion and 120 seconds after the issuance of the assertion. An Audience element in the AudienceRestriction element complies with the Web Browser SSO profile MUST be included. Other Audience elements MUST NOT be included. Other Conditions MUST NOT be included.
	Advice	eRecognition: MUST NOT be included



Data element		Delivering
	AuthnStatem graft	eRecognition: The attribute MUST AuthnInstant the time of contain authentication. The AuthnContext element MUST be a AuthnContextClassRef element containing the confidence level is used * contain. See section 9.2.1 or section 9.1. The AuthenticatingAuthority element MUST be filled with the EntityID the authentication service authentication yielded carried out. Other elements and attributes MUST NOT be included. * The confidence level is used MUST be the lowest the reliability levels of the Declaration on the authentication of the authentication service and the Declaration on the jurisdiction of the authorization registry.
	AttributeState ment	eRecognition: MUST be a statement about the power of the acting contain natural person. (See next section). If additional attributes are required by the service provider and attributes by authentication service and / or authorization register delivered This MAY be included here. Other AttributeStatement elements MUST NOT be included.

251 5.2.2 Declaration on competence

Data element		Delivering
Part of statement about authentication	AttributeSta tement	eRecognition: MUST attributes EntityConcernedID and Service ID (Service ID in the long format) and MUST contain the attributes EntityConcernedSubID and OldEntityConcernedSubID contain. During the migration period for establishment numbers of the Chamber (See also section 8.1.4) MUST be in the case of a definition of to a power plant as well EntityConcernedSubID OldEntityConcernedSubID be given. Other attributes MUST NOT be included.

252 *5.3 Alternative binding*

253 When the reports described in Section 2.2.2.2 describes alternative binding
254 exchange subject to the following requirements.

5.3.1 HTTP Redirect binding

For the implementation for the HTTP Redirect binding the following requirements apply:

- The AuthnRequest message equals the message described in section 5.1, but MAY NOT a <ds:Signature> element.
- The message MUST be compressed using the DEFLATE method then Base64 encoding MUST be used.
- The compressed and encrypted message MUST be added to the URL as a query string parameter and MUST be marked as SAMLRequest.
- If relay state data is passed in the HTTP Redirect message must be separately encoded to the URL as a query string parameter and MUST these are referred to as Sate Relay. If no State Relay is specified MUST be all missing parameter in the URL.
- On the part of the URL SAMLRequest = value & Relay State = value MUST electronic signatures are calculated. This electronic signature MUST generate as described in section 3.2. The signature MUST be included as a query string parameter. This parameter is referred to as Signature.

5.3.2 HTTP Artifact binding

For the implementation of the HTTP Artifact binding are all requirements for the ArtifactResolve and Artifact Response messages.

5.3.2.1 ArtifactResolve

Data element	Delivering
@ Id	SAML: Unique feature of message
@ Version	SAML: Version of the SAML protocol. The value MUST be "2.0" are.
@ Instant Issue	SAML: Time the message was created
@ Destination	eRecognition: MUST NOT be included
@ Consent	eRecognition: MUST NOT be included
Issuer	eRecognition: MUST contain the EntityID of the service. See Section 9.2.3. The attributes NameQualifier, SPNameQualifier, Format and SPProviderID MUST NOT be included.
Signature	eRecognition: MUST include the electronic signature of the service over the whole message. See section 3.2 for specific requirements.

Data element	Delivering
Extensions	eRecognition: MUST NOT be included
Artifact	SAML: Contains the artifact as a query parameter that is received.

276 5.3.2.2 *Artifact Response*

Data element	Delivering
@ Id	SAML: Unique feature of the message.
@ InResponseTo	SAML: Unique feature of the AuthnRequest which this Response message is the answer.
@ Version	SAML: Version of the SAML protocol. The value MUST be "2.0" are.
@ Instant Issue	SAML: Time the message was created.
@ Destination	eRecognition: MUST NOT be included
@ Consent	eRecognition: MUST NOT be included
Issuer	eRecognition: MUST be EntityID of eRegocnition Broker contain. See section 9.2.3. The attributes NameQualifier, SPNameQualifier, Format and SPProviderID MUST NOT be included.
Signature	eRecognition: MUST include the electronic signature of the eRegocnition Broker the entire message. See Section 3.2 for specific requirements.
Extensions	eRecognition: MUST NOT be included
Status	eRecognition: MUST contain an element Status code containing the status of the authentication. In case of cancellation or error This element SHALL be filled with the value AuthnFailed. See the descriptions in chapter 4. Status Detail MAY NOT be included.
# # Any Any	eRecognition: Response message MUST contain one (see section 5.2) . This message MUST NOT a <ds:Signature> element.

277 6 Service Catalogue

278 In this chapter, the format and the publication of the service catalog.

6.1 Format

The service catalog MUST conform to the following format:

- Instant Issue (Time when the service catalog is created)
- NotOnOrAfter (Time when the service catalog is no longer valid)
- Version (version of the service catalog in the format en: eRecognition: <version Agreements System>: <serial number dienstencatalogus>. Eg. en: eRecognition: 0.8def: 1)
- Signature (Signature by management organization, or eRegocnition Broker service, for authenticity, integrity and non-repudiation).
- Per service:
  - IsPublic (attribute that indicates whether the service eRecognition publicly use has)
  - ServiceProviderID (The OIN of the service)
  - OrganizationDisplayName (The name of the service as defined by participants MUST be shown, max 64 characters). This element MUST for different languages are included.
  - each service:
    - IsPublic (attribute that indicates whether the service eRecognition publicly use it)
    - Service ID (A assigned by the service provider and delivered unique number of 1-64000 in the format urn: en: eRecognition: DV: <OIN>: Services: <unique service number>
    - Service Name (Name of the service, determined by the service provider, max 64 characters) This element MAY be included for different languages.
    - Service Description (Brief description of service, max 1024 characters, determined by provider. Permission Registers MAY use this text to administrators to assist in the capture of powers). This element MUST for different languages can be recorded.
    - ServiceDescriptionURL (max. 512 characters of a URL where an extensive defining service is found, determined by service.

Permission Registers MAY include this link to help administrators in capturing permissions). This element MUST for different languages included.

- AuthnContextClassRef (Confidence level required for the service, determined by provider)

The format of the service catalog is the form of an XML Schema included in Chapter 10.

6.2 Publication

The management organization publishes the catalog of services at a fixed location.

One participant MUST periodically in advance by the management organization chosen time, the service catalog process. Information about the URL and periodicity are described in [Operating Manual].

One participant SHOULD MUST process the metadata with an automatic process. A participant MUST relating to rollback, or other changes, this automatic process also intermediate times can perform.

7 Attribute Catalogue

This chapter defines the maximum available eRecognition additional attributes can be obtained. It concerns only non-validated attributes. Supplying of these attributes is optional.

per attribute is a name and format specified. Also indicates whether the attribute by an authentication or authorization service registry can be delivered.

All attribute strings are formatted in the Unicode character set in UTF-8, like the rest of message. The RequestedAttributes the namespace: "urn: en: eRecognition: 1.3a: attributeextension: RequestedAttributes". Supplied attributes

attributenaam one that starts with "urn: en: eRecognition: 1.3: AdditionalAttribute".
For readability is not included in the table below.
(see eRecognition XML schema extensions, chapter 12) .

Attribute Name	Format	Definition	AD / MR
Business Address	String up to 256	Address of the agent natural person, the used in the context of competence	MR
Business Mail	String up to 256	E-Mail address of the acting course person, in the context of the used competence	MR
Business Phone	String max 128	Telephone number of the acting course person, in the context of the used competence	MR
Acting Person Name	String max 128	Name of the agent natural person	AD
Personal Mail	String up to 256	E -Mail address of the acting course person registered with agent issue (would thus Private may be)	AD

Personal Phone	String max 128	Telephone number of the acting course person registered with agent issue (would thus Private may be)	AD
Business Name	String max 128	Name of the negotiated company.	MR
User Defined	String up to 256	A user defined attribute is a text field that the registration by one that indicates to is recorded.	MR
Business Address City	String max 128	City of acting course person, in the context of competence.	MR
BusinessAddressPostalCo the	String max 128	Postcode acting course person, in the context	MR

		of competence.	
BusinessAddressHouseNumber	String max 128	Number of acting course person, in the context of jurisdiction.	MR

The attributes Business Address City Business Address Postal Code Business Address and House Number in addition to the attribute Business Address redundant allowed.

The attributes in the table above may for acting natural persons whose powers are limited to a particular branch by branch also be recorded.

8 Metadata

In eRecognition network is the use of SAML metadata between participants required for describing the URLs and certificates used by the different interfaces. Between service and eRegocnition Brokers is not obliged to Making SAML metadata. When it uses SAML metadata can use can be made of it in the next section is not normative format.

8.1 Format metadata

The SAML metadata is a valid file, according to urn: oasis: names: tc: SAML: 2.0: metadata containing a signed EntityDescriptor element. The signing is performed in accordance which is described in chapter 3.

The element contains the EntityDesciptor entityID and shall not contain SAML attributes. The entityID has the form urn: en: eRecognition: ROLE: OIN: INDEX, which ROL one of DV or HM depends of the roll, the OIN OIN of the service provider or eRegocnition Broker, and INDEX is an self-selected index is four digits.

Example:

```
<? xml version = "1.0" encoding = "UTF-8"?>
<md: EntityDescriptor
ID = "[reference for dsig]"
xmlns: md = "urn: oasis: names: tc: SAML: 2.0: metadata"
```

```
xmlns: attr = "urn: oasis: names: tc: SAML: metadata: attribute"
xmlns: ds = "# http://www.w3.org/2000/09/xmldsig"
entityID = "urn: en: eRecognition: HM: 9999990000010000:0001" >
   <ds:Signature> ... </ ds: Signature>
...
</ md: EntityDescriptor>
```

In the EntityDescriptor one or more elements of type Contact Person listed which name, email address, and telephone number of persons who are described in the case problems, can be contacted.

The metadata includes data on the organization, through the inclusion of a element of type Organization, which name (OrganizationName), the readable name for user (OrganizationDisplayName), and website (OrganizationURL) are described.

A role can perform multiple systems. For each system separately then

metadata supplied. The metadata contains different EntityID, where the element Organization same.

When a system different versions of the Agreements System must support each version metadata are supplied separately.

A eRegocnition Broker takes a IDPSSODescriptor element containing a SingleSignOnService element and no other elements.

Example:

```
...
<md: IDPSSODescriptor WantAuthnRequestsSigned = "true"
protocolSupportEnumeration = "urn: oasis: names: tc: SAML: 2.0: protocol" >
        <md: SingleSignOnService Binding = "urn: oasis: names: tc: SAML: 2.0: bindings: HTTP-POST"
Location = "https:// ...." />
</ md: IDPSSODescriptor>
...
```

A service takes a SPSSODescriptor and no other elements.

Example eRegocnition Broker:

```
...
<md: SPSSODescriptor AuthnRequestsSigned = "true"
Cause Assertion Signed = "true"
 protocolSupportEnumeration = "urn: oasis: names: tc: SAML: 2.0: protocol" >
...
   <md: Assertion Consumer Service Binding = "urn: oasis: names: tc: SAML: 2.0: bindings: HTTP-POST"
Location = "https:// ..." Index = "1" />
</ md: SPSSODescriptor>
...
```

A IDPSSODescriptor contains attribute WantAuthnRequestsSigned = true and contains no other

optional attributes. A SPSSODescriptor element contains an attribute AuthnRequestsSigned = true Signed and Cause Assertion = true and no other optional attributes.

A IDPSSODesriptor SPSSODescriptor or contains one or more elements with the KeyDescriptor attribute use = "signing". Each KeyDescriptor KeyName element contains a valid and G2 PKIoverheid certificate, which the SAML messages from the participant can be authenticated. NB: Service and eRegocnition Broker all listed KeyDescriptors process. In the signatures in the protocol messages by means of the KeyName indicated which certificate from the metadata is used for signing.

Example:


```
...
<md: KeyDescriptor use = "signing" >
      <ds:KeyInfo>
          <ds:KeyName>
2fd4e1c6 7a2d28fc ed849ee1 bb76e739 1b93eb12
          </ ds: KeyName>
          <ds:X509Data>
            <ds:X509Certificate>
...
             </ Ds: X509Certificate>
           </ ds: X509Data>
      </ ds: KeyInfo>
</ md: KeyDescriptor>
...
```

9 Data elements

This chapter describes all eRecognition defined data elements. Applied SAML pure elements according to the SAML standard used are not included here.

9.1 OIN format

Within eRecognition the OIN format used to service customers, branches, participants and service providers to indicate. The OIN format is defined within Digi Clutch. A OIN consists of the following concatenated elements:

- An 8-digit prefix that the register indicates that the number is defined
- A number whose interpretation depends on the register

Within eRecognition be -FI numbers, Chamber of numbers, branch numbers and numbers from the Digi Kope Ling registry used. Numbers of foreign trade registers or similar public records can be added after it with a specific prefix issued. The corresponding numbers on the number indicated by "authorized foreign number". Below is the exact definition of each number described below.

9.1.1 Number FI

The FI-number prefix is 00000001 or 00000002. The number consists of the 9-digit number with a suffix of 3 digits. For prefix 00000001 This suffix is always 000. Example: 000000011234567890000 or 000000021234567890001

9.1.2 Registration number

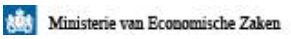
The Chamber prefix number is 00000003.
The number consists of the 8-digit Registration number with a suffix of 0000, for example:

00000003123456780000

9.1.3 Branch number (new format)

The new office number is 000 000 prefix? (Is still tbc in consultation with Logius). The number is the 12-digit confirmation number, for example:

Four hundred and forty-seven 0 million? 123456789012



448 9.1.4 Branch number (old format)

449 The old establishment prefix number is 00000003.
450 The number consists of the 8-digit number Chamber of Commerce and the 4-digit branch number,
451 for example:

452 00000003123456780008

453 NB Until 31 December 2013 the Chamber for all sites (existing and newly registered)
454 an old and a new branch track record.

455 9.1.5 Digi Clutch registry

456 This number is Logius issued to service providers who do not (own) Registration number
457 have.

458 The number of the Digi Clutch registry prefix is 00000004.
459 The number is a 9-digit number followed by 000, for example:

460 00000004123456789000

461 9.1.6 Foreign registers

462 The use of eRecognition by foreign service customers from other EU Member States
463 specific prefixes can be defined in collaboration with the Dutch and Logius
464 service desk (Answer for Businesses). It is a register of respective country prefix
465 issued. If applicable, there is both a prefix for the company level
466 (corporate and individual) as a prefix for the site level issued. By country
467 go to the website of eRecognition be communicated to service users of the
468 country can make use, referring to the Dutch
469 service counter and the specific requirements of the respective country.

470 It is the rule that if a CoC number is issued to the relevant
471 customer service that always that number should be used. That is to say that in the
472 Dutch commercial register service customers and branches of foreign
473 service customers is not the issue in which they are registered in a foreign registry may
474 used within eRecognition. If any part of the service recipient in Dutch
475 Commerce is registered then the CoC number used.

476 *identifying characteristics 9.2*

477 eRecognition Within the following identifying characteristics defined.

478 9.2.1 Confidence level

479 To ensure the reliability levels of eRecognition in messages is distinguished
480 following subset of the SAML 2.0 specifications for AuthnContextClassRef element
481 used values allowed. Other values MUST NOT be used.

eRecognition level	SAML2 AuthnContextClassRef element
1	urn: oasis: names: tc: SAML: 2.0: ac: classes: Password Protected Transport
2 [13]	urn: oasis: names: tc: SAML: 2.0: ac: classes: MobileTwoFactorUnregistered
3	urn: oasis: names: tc: SAML: 2.0: ac: classes: MobileTwoFactorContract
4	urn: oasis: names: tc: SAML: 2.0: ac: classes: SmartcardPKI

482 See document [eRecognition - Reliability Levels].

483 9.2.2 Service ID

484 Within eRecognition, all services indicated by a in the context of the
485 service unique number, the Service ID. Each Service ID in the service catalog
486 , as part of the format of an urn

487 urn: en: eRecognition: DV: <OIN>: Services: <ServiceID>

488 <OIN> which represents the OIN of the service providing the service. By means of
489 urn services are unique within eRecognition defined.

490 Registered services MUST have a Service ID of 1 or higher. One-stop shop or portal function
491 messages is indicated by the reserved Service ID with value '0 '. This is not
492 included in the service catalog.

493 The Service ID is used messages in two formats:
494 1. The short format. Here, only the Service ID included.
495 2. The long format. Here, the full urn recorded.

496 9.2.3 EntityID

497 Within eRecognition all systems of participants and service providers indicated by a

13 level was during the trial implementations given the name "level NL". The old level 2 has lapsed.

EntityID included in the metadata. Is the size of the EntityID

urn: en: eRecognition: <ROL>: <OIN>: Entities: <Index>

where the values <ROL> DV, HM, AD or MR may have <OIN> stands for the Chamber of Commerce number or Digi Link number of the participant or the FI number, CoC number or Digi Link number service. The <Index> is a free service by participant or to choose number from 0 t / m 8999 for various endpoints to define them. Songs of 9000 t / m nine thousand nine hundred ninety-nine are reserved for test systems.

9.2.4 Pseudonyms

Within eRecognition a natural person acting in two different ways indicated:

1. within the network with an internal pseudonym by the authentication service, and
2. inside and outside the network with a specific alias by
   authorizing registry

To protect the privacy of the acting individual to assure to a service only a specific pseudonym provided.

The internal pseudonym is not relevant for this interface and is not further described.

9.2.4.1 Specific pseudonym

The pseudonym is specifically authorized by the registry once and subsequently to the acting individual linked. The specific pseudonym MUST be unique within eRecognition and for each different combination of service, acting individual and acting service consumer MUST different specific pseudonym are recorded and used. The authorization MUST register a pseudonym behalf of a service when generating a natural person acting in one or more permissions for services that are registered service. As long as the combination of the natural person acting and the acting customer service remains unchanged, there no obligation to define new pseudonyms. At the request of the legal agent, manager and / or the acting individual MUST always new pseudonyms are generated. Once used specifically pseudonym MAG NOT be reused. Portability of specific pseudonyms is possible because once generated pseudonyms for the relevant natural person acting in other authority records MUST be included.

The format of the pseudonym MUST specific hexadecimal value of 32 bytes followed by

an @ and a hexadecimal value of 16 bytes. Eg.

ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890 @ ABCDEF1234 567890ABCDEF1234567890

The value of 32 bytes MUST be a random value. This can be achieved a SHA256 hash on the following elements (in this order and separated by a separator stabbing) to compute:

- The OIN of the service
- The OIN of the agent service consumer
- A in the context of negotiating service recipient unique (but not necessarily exclusive) identifier of the acting individual. This attribute MAY by the Administrator or by the registry authority determined, but also the internal CAN pseudonym.
- A 16-byte random number.

The value of 16 bytes MUST be a MD5 hash on the OIN of the acting department buyer. NB There may be other formats in circulation. Users of the specific pseudonym is not recommended (parts of) the pseudonym be used for purposes other than identifying the acting individual.

9.3 SAML Attributes

This section describes the data elements as SAML Attribute element messages prevented.

The eRecognition specific attributes are marked with an urn. This contains the urn version number of the Agreements System which (who) version of the relevant attribute for the is first recorded.

9.3.1 EntityConcernedID

Definition	A SAML Attribute element containing the OIN of the Chamber number or number of the authorized foreign acting customer service by the acting individual represented.
Name	urn: en: eRecognition: 1.0: EntityConcernedID
Type	http://www.w3.org/2001/XMLSchema # string
AttributeValue	See section 9.1.
Notes	Note the use of the OIN format, the Chamber of Commerce number followed by 0000. This is emphatically not a demarcation of competence to an establishment involved. The OIN is intended only to indicate the number Chamber.

9.3.2 EntityConcernedSubID 1.2

Definition	An optional SAML Attribute element containing the OIN of the branch number in the new format (see Section 9.1.3) or allowed foreign establishment number (see section 9.1.6) which the power of the acting individual is delineated.

Name	urn: en: eRecognition: 1.2: EntityConcernedSubID
Type	http://www.w3.org/2001/XMLSchema # string
AttributeValue	See section 9.1.3 and 9.1.6
Notes	

555 ### 9.3.3 OldEntityConcernedSubID 1.2

Definition	An optional SAML Attribute element containing the OIN of the branch number in the old format (see Section 9.1.4) which the power of the acting individual is delineated.
Name	urn: en: eRecognition: 1.2: OldEntityConcernedSubID
Type	http://www.w3.org/2001/XMLSchema # string
AttributeValue	See section 9.1.4
Notes	Note: The last four digits indicate always explicitly a branch to.

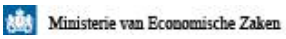
556 ### 9.3.4 Service ID

Definition	A SAML Attribute element containing the Service ID of the service which the authorization applies. See section 9.2.2.
Name	urn: en: eRecognition: 1.0: Service ID
Type	http://www.w3.org/2001/XMLSchema # string
AttributeValue	See section 9.2.2. The value MUST match the value of @ AttributeConsumingServiceID from the AuthnRequest. See section 5.1.

557 ## *9.4 URL or POST variable: EherkenningPreferredLanguage*

Definition	Preferred Language of acting natural person
Name	EherkenningPreferredLanguage

Format	According to ISO 639-1:2002



10 Appendix XML Schema service catalog

```
<? xml version = "1.0" encoding = "UTF-8"?>
<xs: schema xmlns: xs = "http://www.w3.org/2001/XMLSchema" xmlns: md = "urn: oasis: names: tc: SAML: 2.0: metadata"
xmlns: ds = "http://www.w3.org/2000/09/xmldsig #" xmlns: saml2 = "urn: oasis: names: tc: SAML: 2.0: assertion"
xmlns: eh = "urn: en: eRecognition: 1.0" target namespace = "urn: en: eRecognition: 1.0" form element default = "qualified"
attributeFormDefault = "unqualified">
        <xs: import namespace = "# http://www.w3.org/2000/09/xmldsig"
schema location = "http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/xmldsig-core-schema.xsd" />
        <xs: import namespace = "urn: oasis: names: tc: SAML: 2.0: assertion" schema location = "SAML schema-assertion-
2.0.xsd "/>
        <xs: import namespace = "urn: oasis: names: tc: SAML: 2.0: metadata" schema location = "SAML schema metadata
2.0.xsd "/>
        <! - Elements ->
        <xs:element name="Service">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element ref="eh:ServiceID"/>
                                <xs:element ref="eh:ServiceName" maxOccurs="unbounded"/>
                                <xs:element ref="eh:ServiceDescription" maxOccurs="unbounded"/>
                                <Xs: element ref = "eh: ServiceDescriptionURL" minOccurs = "0"
maxOccurs = "unbounded" />
```

```
                                <xs:element ref="saml2:AuthnContextClassRef"/>
                        </ Xs: sequence>
                        <xs:attribute ref="eh:IsPublic" use="required"/>
                </ Xs: complexType>
        </ xs: element>
        <xs:element name="ServiceCatalogue">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element ref="ds:Signature"/>
                                <xs:element ref="eh:ServiceProvider" maxOccurs="unbounded"/>
                        </ Xs: sequence>
                        <xs:attribute ref="eh:IssueInstant" use="required"/>
                        <xs:attribute ref="eh:NotOnOrAfter" use="required"/>
                        <xs:attribute ref="eh:Version" use="required"/>
                        <xs:attribute name="ID" type="xs:string"/>
                </ Xs: complexType>
        </ xs: element>
        <xs:element name="ServiceDescription">
                <xs:complexType>
                        <xs:simpleContent>
                                <xs:restriction base="md:localizedNameType">
                                        <xs:maxLength value="512"/>
                                </ Xs: restriction>
                        </ Xs: simple content>
                </ Xs: complexType>
        </ xs: element>
        <xs:element name="ServiceDescriptionURL">
```



```
                <xs:complexType>
                        <xs:simpleContent>
                                <xs:restriction base="md:localizedURIType">
                                        <xs:maxLength value="512"/>
                                </ Xs: restriction>
                        </ Xs: simple content>
                </ Xs: complexType>
        </ xs: element>
        <xs:element name="ServiceID" type="xs:anyURI"/>
        <xs:element name="ServiceName">
                <xs:complexType>
                        <xs:simpleContent>
                                <xs:restriction base="md:localizedNameType">
                                        <xs:maxLength value="64"/>
                                </ Xs: restriction>
                        </ Xs: simple content>
                </ Xs: complexType>
        </ xs: element>
        <xs:element name="ServiceProvider">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element ref="eh:ServiceProviderID"/>
                                <xs:element ref="eh:OrganizationDisplayName" maxOccurs="unbounded"/>
                                <xs:element ref="eh:Service" maxOccurs="unbounded"/>
                        </ Xs: sequence>
                        <xs:attribute ref="eh:IsPublic" use="required"/>
                </ Xs: complexType>
        </ xs: element>
        <xs:element name="ServiceProviderID" type="xs:anyURI"/>
        <xs:element name="OrganizationDisplayName">
```

```
                    <xs:complexType>
                        <xs:simpleContent>
                            <xs:restriction base="md:localizedNameType">
                                <xs:maxLength value="64"/>
                            </ Xs: restriction>
                        </ Xs: simple content>
                    </ Xs: complexType>
                </ xs: element>
                <! - Attributes ->
                <xs:attribute name="IssueInstant" type="xs:dateTime"/>
                <xs:attribute name="IsPublic" type="xs:boolean"/>
                <xs:attribute name="NotOnOrAfter" type="xs:dateTime"/>
                <xs:attribute name="Version" type="xs:anyURI"/>
</ xs: schema>
```

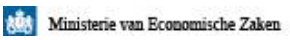

11 Appendix Example messages

This annex describes two sample messages given. There are no sample values for elements and attributes filled.

11.1 AuthnRequest

```
<? xml version = "1.0" encoding = "UTF-8"?>
<samlp: AuthnRequest xmlns: samlp = "urn: oasis: names: tc: SAML: 2.0: protocol" ForceAuthn = "true" Destination = ""
Assertion Consumer Service Index = "" AttributeConsumingServiceIndex = "2" ID = "" Instant Issue = "" Version = "2.0"
ProviderName = "">
        <saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
 </ SAML: Issuer>
        <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
            <ds:SignedInfo>
                <Ds: Canonicalization Method Algorithm = "http://www.w3.org/2001/10/xml-exc-c14n #" />
                <Ds: Signature Method Algorithm = "http://www.w3.org/2001/04/xmldsig-more # rsa-
sha256 "/>
                <ds:Reference URI=" ">
                    <ds:Transforms>
                        <Ds: Transform
Algorithm = "http://www.w3.org/2000/09/xmldsig # enveloped-signature" />
                        <Ds: Transform Algorithm = "http://www.w3.org/2001/10/xml-exc-c14n #" />
                    </ Ds: Transforms>
                    <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                    <ds:DigestValue>
         </ Ds: Digest Value>
                </ Ds: Reference>
            </ Ds: SignedInfo>
            <ds:SignatureValue>
        </ Ds: Signature Value>
            <ds:KeyInfo>
                <ds:KeyName>
            </ Ds: KeyName>
```

```
                    </ Ds: KeyInfo>
          </ ds: Signature>
          <samlp:Extensions>
            <ehsamlp:RequestedAttributes>
                    <md:RequestedAttribute Name="urn:nl:eherkenning1.3:AdditionalAttribute:ActingPersonName"/>
              <md:RequestedAttribute Name="urn:nl:eherkenning1.3:AdditionalAttribute:PersonalEmail"/>
              <md:RequestedAttribute Name="urn:nl:eherkenning1.3:AdditionalAttribute:PersonalPhone"/>
            </ ehsamlp: RequestedAttributes>
          </ samlp: Extensions>
          <samlp:RequestedAuthnContext Comparison="minimum">
                    <saml:AuthnContextClassRef xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
      </ SAML: AuthnContextClassRef>
          </ samlp: RequestedAuthnContext>
</ samlp: AuthnRequest>
```



11.2 Response

```
<? xml version = "1.0" encoding = "UTF-8"?>
<samlp: Response xmlns: samlp = "urn: oasis: names: tc: SAML: 2.0: protocol" id = "" InResponseTo = "" Version = "2.0"
Destination = "" Instant Issue = "">
        <saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
  </ SAML: Issuer>
        <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                <ds:SignedInfo>
                        <Ds: Canonicalization Method Algorithm = "http://www.w3.org/2001/10/xml-exc-c14n #" />
                        <Ds: Signature Method Algorithm = "http://www.w3.org/2001/04/xmldsig-more # rsa-
sha256 "/>
                        <ds:Reference URI=" ">
                                <ds:Transforms>
                                        <Ds: Transform
Algorithm = "http://www.w3.org/2000/09/xmldsig # enveloped-signature" />
                                        <Ds: Transform Algorithm = "http://www.w3.org/2001/10/xml-exc-c14n #" />
                                </ Ds: Transforms>
                                <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                                <ds:DigestValue>
        </ Ds: Digest Value>
                        </ Ds: Reference>
                </ Ds: SignedInfo>
                <ds:SignatureValue>
    </ Ds: Signature Value>
                <ds:KeyInfo>
                        <ds:KeyName>
                </ Ds: KeyName>
                </ Ds: KeyInfo>
        </ ds: Signature>
        <samlp:Status>
                 <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success">
     </ samlp: Status code>
        </ samlp: Status>
        <saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" Version="2.0" id=" "IssueInstant=" ">
                 <saml:Issuer>
     </ SAML: Issuer>
                 <saml:Subject>
                         <saml:NameID>
       </ SAML: NameID>
                         <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
                                 <saml:SubjectConfirmationData Recipient=" "NotOnOrAfter=" ">
         </ SAML: SubjectConfirmationData>
                         </ SAML: SubjectConfirmation>
                 </ SAML: Subject>
                 <saml:Conditions NotBefore=" "NotOnOrAfter=" ">
                         <saml:AudienceRestriction>
                                 <saml:Audience>
        </ SAML: Audience>
```




```
                                </ SAML: AudienceRestriction>
                        </ SAML: Conditions>
                        <saml:AuthnStatement AuthnInstant=" ">
                                <saml:AuthnContext>
                                        <saml:AuthnContextClassRef>
        </ SAML: AuthnContextClassRef>
                                        <saml:AuthenticatingAuthority>
        </ SAML: AuthenticatingAuthority>
                                </ SAML: AuthnContext>
                        </ SAML: AuthnStatement>
                        <saml:AttributeStatement>
                                <saml:Attribute Name="urn:nl:eherkenning:1.0:ServiceID">
                                        <SAML: attributeValue xmlns: xs = "http://www.w3.org/2001/XMLSchema"
xmlns: xsi = "http://www.w3.org/2001/XMLSchema-instance" xsi: type = "xs: string">
         </ SAML: attributeValue>
                                </ SAML: Attribute>
                                <saml:Attribute Name="urn:nl:eherkenning:1.0:EntityConcernedID">
                                        <SAML: attributeValue xmlns: xs = "http://www.w3.org/2001/XMLSchema"
xmlns: xsi = "http://www.w3.org/2001/XMLSchema-instance" xsi: type = "xs: string">
         </ SAML: attributeValue>
                                 </ SAML: Attribute>
                                 <saml:Attribute Name="urn:nl:eherkenning1.3:AdditionalAttribute:ActingPersonName">
                                        <SAML: attributeValue xmlns: xs = "http://www.w3.org/2001/XMLSchema"
xmlns: xsi = "http://www.w3.org/2001/XMLSchema-instance" xsi: type = "xs: string">
                                         </ SAML: attributeValue>
                                 </ SAML: Attribute>
                                 <saml:Attribute Name="urn:nl:eherkenning1.3:AdditionalAttribute:BusinessName">
                                        <SAML: attributeValue xmlns: xs = "http://www.w3.org/2001/XMLSchema"
xmlns: xsi = "http://www.w3.org/2001/XMLSchema-instance" xsi: type = "xs: string">
                                         </ SAML: attributeValue>
                                 </ SAML: Attribute>
                        </ SAML: AttributeStatement>
        </ SAML: Assertion>
</ samlp: Response>
```


12 eRecognition XML Schema extensions

12.1 XML schema attribute extension

```
<? xml version = "1.0" encoding = "UTF-8"?>
<xs: schema xmlns: ehsamlp = "urn: en: eRecognition: 1.3a: attributeextension"
                target namespace = "urn: en: eRecognition: 1.3a: attributeextension"
                xmlns: xs = "http://www.w3.org/2001/XMLSchema"
                xmlns: md = "urn: oasis: names: tc: SAML: 2.0: metadata"
                default element form = "qualified" attributeFormDefault = "unqualified">
  <xs:element name="RequestedAttributes">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="md:RequestedAttribute" maxOccurs="unbounded"/>
      </ xs: sequence>
    </ xs: complexType>
  </ xs: element>
</ xs: schema>
```