CATCert
Agència Catalana
de Certificació

SAML Signature Statement

Referència:	
Versió:	1.0
Data:	28/08/2006

AOC
Consorci
Administració Oberta
de Catalunya

Generalitat
de Catalunya

Consorci de governs locals
per a la societat de la informació

Informació general

Control documental

Projecte:	N/A
Entitat de destinació:	CATCert, GUIDE Project
Títol:	SAML Signature Statement
Codi de referència:	N/A
Versió:	1.0
Data:	28/08/2006
Fitxer:	Signature Statement v1r0.doc
Eina/es d'edició:	Word 2002
Autor/s:	Nacho Alamillo / Daniel Martínez
Resum:	

Estat formal

Preparat per:	Revisat per:	Aprovat per:
Nom: AiR Data: 28/08/2006	Nom: Alamillo Data: 28/8/2006	Nom: Alamillo Data: 28/08/2006

Control de versions

Versió	Parts que canvien	Descripció del canvi	Data
1.0	Tot	Creació del document	28/08/2006

Índex

1. Introduction

1.1 Signature Statement concepts

This specification extends the SAML assertion framework defining some new elements used in support of signature federations.

Using these elements, the SAML asserting authority grants a relying party that a valid signature has been produced for a concrete purpose in a concrete jurisdiction, and proper evidence has been produced and is archived.

1.2 Notation

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as described in IETF RFC 2119 [RFC 2119].

This specification uses schema documents conforming to W3C XML Schema [Schema1] and normative text to describe the syntax and semantics of XML-encoded SAML assertions and protocol messages regarding a Signature Statement.

1.3 Schema organization and namespaces

The Signature Statement structures are defined in a schema [SAMLSS-XSD] associated with the following XML namespace:

```
urn:catcert:samlss:1.0:assertion
```

The Signature Statement request-response protocol structures are defined in a schema [SAMLSSP-XSD] associated with the following XML namespace:

```
urn:catcert:samlss:1.0:protocol
```

XMLdSig, DSS and XAdES schemas are imported in this schema. They are defined in following namespaces respectively:

```
http://www.w3.org/2000/09/xmldsig#
```

```
urn:oasis:names:tc:dss:1.0:core:schema
```

```
http://uri.etsi.org/01903/v1.2.2#
```

2. Signature statement SAML schema extension

2.1 Element <SignatureStatement>

The `<SignatureStatement>` element describes a statement by a Signature legitimation authority asserting that the assertion subject produced a signature at a particular time, for a concrete purpose. Assertions containing `<SignatureStatement>` elements MUST contain a `<Subject>` element.

It is of type **SignatureStatementType**, which extends **StatementAbstractType** with the addition of the following elements and attributes:

`LegalizationInstant` [Required]

> Specifies the time at which the signature was legitimated by the Signature legitimation authority. The time value is encoded in UTC, as described in Section 1.3.3 of SAML.

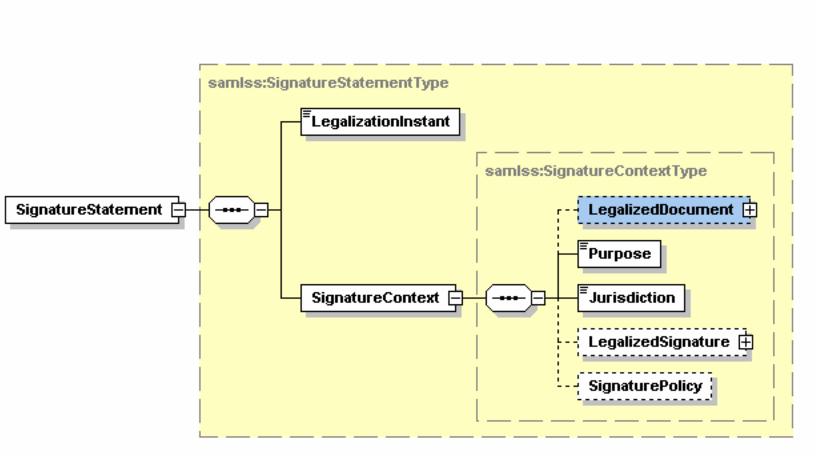
`<SignatureContext>` [Required]

> Specifies the context used by the Signature legitimation authority to produce the statement (see section 2.2)

The following schema fragment defines the `<SignatureStatement>` element and its **SignatureStatementType**:

```
<element name="SignatureStatement" type="samlss:SignatureStatementType"/>
<complexType name="SignatureStatementType">
      <complexContent>
      <extension base="saml:StatementAbstractType">
            <sequence>
                  <element name="LegalizationInstant" type="xs:dateTime"/>
                  <element name="SignatureContext"
                          type="samlss:SignatureContextType"/>
            </sequence>
      </extension>
      </complexContent>
</complexType>
```

The next graph shows the SignatureStatement structure, with its SignatureContext object.



2.2 Element <SignatureContext>

The `<SignatureContext>` element specifies the context of a signature legitimation event. It is defined as the information, additional to the signature assertion itself that the relying party may require before it makes decision with respect to a signature assertion.

It is of complex type **SignatureContextType**, which has the following attributes:

`Purpose` [Required]

Specifies the concrete purpose for which the signature has been legitimated.

`Jurisdiction` [Required]

Specifies the concrete jurisdiction according to which the legalization process has been performed. It includes the two-letter code for the country, as defined by ISO 3316.

`<LegalizedDocument>` [Optional]

Specifies the document received and legalized without signature. It is of type DocumentType from DSS schema, and should contain XML documents, or binary ones base64 encoded into a dss:Base64Data element.

`<LegalizedSignature>` [Optional]

Specifies the document legalized and signed. It is of type SignatureType (see section 2.3)

`<SignaturePolicy>` [Optional]

Specifies the signature policy applied in the signature legitimation process, to yield the statement.

The signature policy, defined in the XAdES schema, is a set of rules for the creation and validation of an electronic signature, under which the signature can be determined to be valid.

The following schema fragment defines **SignatureContextType**:

```
<complexType name="SignatureContextType">
<sequence>
      <element name="LegalizedDocument" type="dss:DocumentType"
            minOccurs="0"/>
      <element name="Purpose" type="xs:string"/>
      <element name="Jurisdiction" type="xs:string"/>
      <element name="LegalizedSignature" type="samlss:SignatureType"
            minOccurs="0"/>
      <element name="SignaturePolicy"
            type="xsd:SignaturePolicyIdentifierType" minOccurs="0"/>
</sequence>
</complexType>
```

2.3 Type SignatureType

The complexType **SignatureType** defines digital signature objects into a SAML 1.1 signature statement profile.

A SignatureType element MUST contain one of the next elements:

`ds:Signature`

Signature object defined in XMLdSig schema.

`dss:Base64Signature`

A base64 encoding of some non-XML signature, such as a PGP or signature. The type of signature is specified by its `Type` attribute.

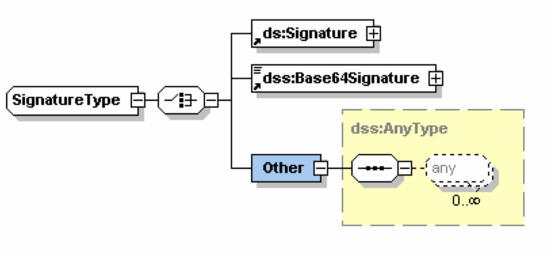
`Other`

Other may contain arbitrary content that may be specified in a profile and can also be used to extend the Protocol. It is of type `dss:AnyType`

The following schema fragment defines SignatureType SAML Signature Statement profile objects:

```
<complexType name="SignatureType">
      <choice>
            <element ref="ds:Signature"/>
            <element ref="dss:Base64Signature"/>
            <element name="Other" type="dss:AnyType"/>
      </choice>
</complexType>
```

Next chart shows SignatureType structure.



3. Signature Statement SAML protocol extension

3.1 Element <LegalizationQuery>

The <LegalizationQuery> is a request element which allows requiring the legalization of a document, with or without digital signature.

It is of type **LegalizationQueryType**, which extends **SubjectQueryAbstractType** with the addition of the following elements and attributes:

DocumentToLegalize

Must contain the document to be signed and legalized. It is a DSS DocumentType object, and should contain XML documents, or binary ones base64 encoded into a dss:Base64Data element. The type of data is specified by its MimeType attribute, that may be required when using DSS with other signature types..

SignatureToLegalize

May contain a legalized electronic signature. It is of type SignatureType.

When a SignatureToLegalize is sent, the receiver system MUST grant that it is a valid signature for the concrete requested purpose.

When SignatureToLegalize is not provided, the receiver system MUST produce or request the digital signature of the document to the end user, and grant that it is a valid signature for the concrete requested purpose.

SignatureStatementOptions

Defines the content of the SignatureStatement to be produced. It is of type SignatureStatementOptionsType and has two elements:

<ReturnLegalizedDocument>

Requests <LegalizedDocument> in SignatureStatement response.

<ReturnLegalizedSignature>

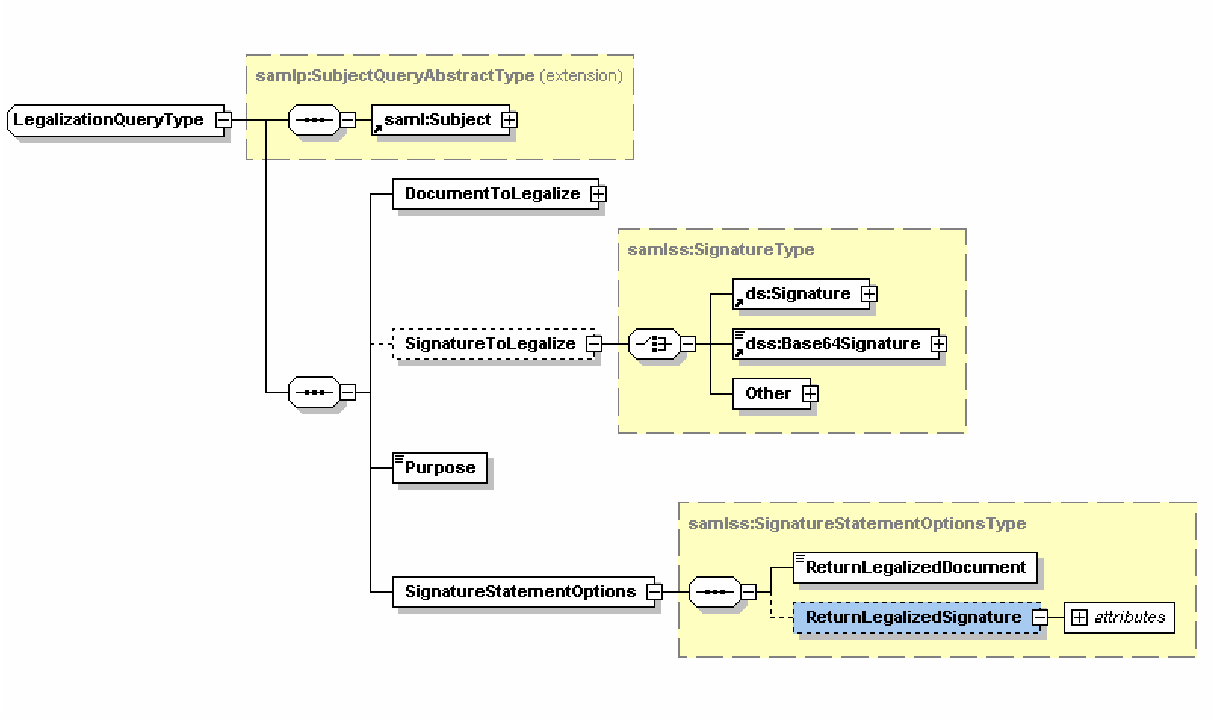
Requests <LegalizedSignature> in SignatureStatement response. It has the attribute ReturnLegalizationPolicy, which defaut value is set to true, and would be used to request the inclusion of the <SignaturePolicy> element in the SignatureContext.

Every `<LegalizationQuery>` object MUST contain a `<DocumentToLegalize>`.

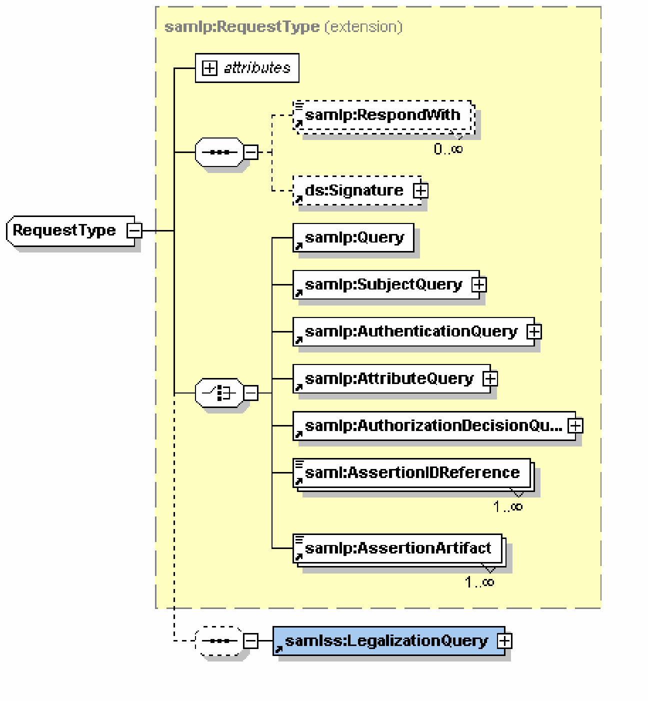
The following schema fragment defines the `<LegalizationQuery>` element, its **LegalizationQueryType** and **SignatureStatementOptionsType**:

```
<element name="LegalizationQuery" type="samlss:LegalizationQueryType"/>
<complexType name="LegalizationQueryType">
      <complexContent>
      <extension base="samlp:SubjectQueryAbstractType">
            <sequence>

                  <element name="DocumentToLegalize"
                              type="dss:DocumentType" minOccurs=”1”/>
                  <element name="SignatureToLegalize"
                              type="samlss:SignatureType" minOccurs=”0”/>

                  <element name="Purpose" type="xs:string" minOccurs=”1”/>
                  <element name="SignatureStatementOptions"
                        type="samlss:SignatureStatementOptionsType"/>
            </sequence>
      </extension>
      </complexContent>
</complexType>
<complexType name="SignatureStatementOptionsType">
      <sequence>
            <element name="ReturnLegalizedDocument" type="boolean"
                  default="false"/>
            <element name="ReturnLegalizedSignature" minOccurs="0">
                  <complexType>
                        <attribute name="ReturnLegalizationPolicy"
                              type="xs:boolean" default="true"/>
                  </complexType>
            </element>
      </sequence>
</complexType>
```

Next chart shows its structure.



A LegalizationQuery object has been included in the SAML 1.1 RequestType definition, in order to allow the request of digital signatures legalization using it.


samlp:RequestType (extension)
attributes
samlp:RespondWith
0..∞
ds:Signature
RequestType
samlp:Query
samlp:SubjectQuery
samlp:AuthenticationQuery
samlp:AttributeQuery
samlp:AuthorizationDecisionQu...
saml:AssertionIDReference
1..∞
samlp:AssertionArtifact
1..∞
samlss:LegalizationQuery