

Guia d'integració PSA

Control documental

Estat formal	Elaborat per: Eduardo Luján Rubén Pérez	Aprovat per:
Data de creació	08/09/2008	
Control de versions	Versió	2.8
	Data	21/03/2011
	Descripció	Afegida missatgeria amb referències dinàmiques.
	Versió	2.7
	Data	07/09/2010
	Descripció	Afegit WS de descàrrega de tiquet de validació de signatura. Afegit canvis a les peticions per la validació síncrona a PSIS. Actualitzacions a la missatgeria degut a la introducció de les Extensions.
	Versió	2.6
	Data	30/04/2010
	Descripció	Afegit annex sobre les respostes dels WS de PSA.
	Versió	2.5
	Data	18/01/2009
	Descripció	Afegits canvis per a la configuració de les peticions de signatura en Browser Afegit informació explicativa sobre UploadAttributes
	Versió	2.4
	Data	04/11/2009
	Descripció	Canvis en la política de seguretat dels WS. Afegits exemples de peticions WS per PDF amb signatura visible.
	Versió	2.3
	Data	22/07/2009
	Descripció	Afegida una llibreria pel Client 1.4
	Versió	2.2
	Data	08/06/2009
	Descripció	Client JDK 1.4
	Versió	2.1
	Data	21/04/2009

	Descripció	Nova estructura del document. Actualització esquemes.
	Versió	2.0
	Data	25/03/2009
	Descripció	Revisió URLS de CATCert
	Data	24/03/2009
	Descripció	Reestructuració del document Actualització de WS (signatura visible, signatura múltiple, eliminació de serveis web innecessaris, etc...)
	Versió	1.7
	Data	12/12/2008
	Descripció	Canvi política de seguretat client .NET
	Versió	1.6
	Data	24/11/2008
	Descripció	Canvi política de seguretat client Java
	Versió	1.5
	Data	10/11/2008
	Descripció	Revisió del client Java
	Versió	1.4
	Data	05/11/2008
	Descripció	Revisió amb inclusió d'exemples de codi
	Versió	1.3
	Data:	10/10/2008
	Descripció:	Revisió amb els comentaris de la primera versió i completat d'informació de generació del client de Java
	Versió	1.2
	Data:	25/09/2008
	Descripció:	Versió completa amb primera revisió
	Versió	1.1
	Data:	08/09/2008
	Descripció:	Redacció inicial
Nivell accés informació	pública	
Títol	Guia d'integració PSA	
Fitxer	06-0435-OD-0001-26-Guia_integració_PSA.doc	
Control de còpies	Només les còpies disponibles a M:\NouPrometeo\Departaments\Tecnica\PSA\Desenvolupament\Integració	

	garanteixen l'actualització dels documents. Tota còpia impresa o desada en ubicacions diferents es consideraran còpies no controlades.
Drets d'autor	Aquesta obra està subjecta a una llicència Reconeixement-No comercial-Sense obres derivades 2.5 Espanya de Creative Commons. Per veure'n una còpia, visiteu http://creativecommons.org/licenses/by-nc-nd/2.5/es/ o envieu una carta a Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.

Glossari

DSS – Digital Signature Services

PSA – Programari de Signatura Avançada

PSIS – Plataforma de Serveis d'identificació i Signatura

SSL – Secure Socket Layer

URL – Uniform Resource Locator

WSE – Web Service Enhancements

WSDL – Web Service Description Language

WSS – Web Service Security

XML – Extensible Markup Language

Índex

Guia d'integració PSA	1
Control documental	2
Glossari	5
Índex	6
1. Document Guia d'Integració	9
1.1 Introducció	9
1.2 Objectius	9
1.3 Capítols	9
2. WebServices a PSA	10
2.1 Profiles WebServices	10
2.2 Seguretat WebServices	10
3. Serveis de signatura	14
3.1 SignatureAtBrowser	14
3.1.1 Funcionalitat	14
3.1.2 Conversa	15
3.1.3 Descripció Missatges	15
3.2 SignatureAtPSA	19
3.2.1 Funcionalitat	19
3.2.2 Conversa	20
3.2.3 Descripció Missatges	20
3.3 SignatureAtSignatureManager	24
3.3.1 Funcionalitat	24
3.3.2 Conversa	24
3.3.2.1 Escenari 1	24
3.3.2.2 Escenari 2	27
3.3.2.3 Escenari 3	29
3.3.3 Descripció Missatges	31
3.4 Extensió per referències dinàmiques	40
3.5 Missatges d'Error	41
4. Serveis d'utilitats	43
4.1 CipherDocument	43
4.1.1 Funcionalitat	43

4.1.2	Descripció Missatges	43
4.2	DecipherDocument	45
4.2.1	Funcionalitat.....	45
4.2.2	Descripció Missatges	45
4.3	DeleteSignature	47
4.3.1	Funcionalitat.....	47
4.3.2	Descripció Missatges	47
4.4	Missatges d'Error	48
5.	Serveis de consulta.....	50
5.1	DownloadSignatureAtPSA	50
5.1.1	Funcionalitat.....	50
5.1.2	Descripció Missatges	50
5.2	QueryActivity	51
5.2.1	Funcionalitat.....	51
5.2.2	Descripció Missatges	51
5.3	QueryActivityToSigner	53
5.3.1	Funcionalitat.....	53
5.3.2	Descripció Missatges	53
5.4	QueryAuthorizationToSignature.....	54
5.4.1	Funcionalitat.....	54
5.4.2	Descripció Missatges	54
5.5	DownloadTicketSignature	55
5.5.1	Funcionalitat.....	55
5.5.2	Descripció Missatges	55
5.6	QuerySignatureAct.....	56
5.6.1	Funcionalitat.....	56
5.6.2	Descripció Missatges	57
5.7	DownloadVerifySignatureTicket.....	58
5.7.1	Funcionalitat.....	58
5.7.2	Descripció Missatges	58
5.8	Missatges d'error.....	59
6.	Client del SI PSA	61
6.1	Requisits previs.....	61
6.1.1	Comunicacions i protocols	61

6.1.2	Software.....	62
6.1.2.1	Client Java	62
6.1.2.2	Client .NET	63
6.2	Execució dels serveis.....	63
6.2.1	Client Java	63
6.2.1.1	Client amb l'API del WS del SI PSA	64
6.2.1.2	Configuració client amb Metro	66
6.2.1.3	Preparació client WSDL2Java	69
6.2.1.4	Generació i compilació	73
6.2.2	Client Java 1.4	73
6.2.2.1	Client amb l'API del WS per Java 1.4 del SI PSA	74
6.2.2.2	Client WSDL2Java con Axis2	76
	Creació del projecte	76
	Generació i compilació.....	82
	Apache Rampart	84
6.2.3	Client .NET.....	85
6.2.3.1	Preparació	85
6.2.3.2	Generació	85
6.2.3.3	Compilació.....	88
6.2.3.4	Execució	89
6.3	Exemples de codi.....	92
6.3.1	Client Java	92
6.3.1.1	Client Java utilitzant l'API	92
6.3.1.2	Client Java ad hoc WSDL2Java	100
6.3.2	Client Java 1.4	102
6.3.2.1	Client Java utilitzant l'API	103
6.3.3	Client .NET (VB)	105
7.	Annexes	122
7.1	Generació de l'API de WS del SI PSA.....	122
7.2	Respostes retornades pels Serveis Web	123
8.	Referències	127

1. Document Guia d'Integració

1.1 Introducció

La guia bàsica d'integració amb el **Programari de Signatura Avançada (PSA)** és un document que va dirigit a desenvolupadors que vulguin integrar les seves aplicacions de gestió amb els diferents serveis del **SI PSA**.

El lector d'aquest document ha de ser un professional amb coneixements en programació avançada amb els llenguatges Java o .Net i estar familiaritzat amb l'ús de Maven2. És molt recomanable disposar també de coneixement sobre Webservices, missatgeria SOAP, WS-Security, ús de certificats i signatures digitals.

1.2 Objectius

Els objectius del present document són:

- Descriure els principals serveis web (WS) que presenta el **SI PSA**, tant a nivell de funcionalitat com a nivell d'esquema.
- Proporcionar una guia concisa però completa de com generar clients que ataquin al **SI PSA**.

No són objectius:

- No substitueix els documents d'anàlisi funcional i disseny tècnic.

1.3 Capítols

En aquest document intentem donar informació dels WS del **SI PSA** i de com realitzar la integració mitjançant un client. Hem organitzat els capítols de la següent forma:

1. [WS a PSA](#). Es presenta l'estructura i organització dels WS oferts pel **SI PSA** així com d'aspectes i característiques comunes.
2. [Serveis de Signatura](#). Descripció detallada dels WS de signatura.
3. [Serveis d'Utilitats](#). Descripció detallada dels WS d'Utilitats.
4. [Serveis de Consulta](#). Descripció detallada dels WS de Consulta.
5. [Client de PSA](#). S'explica, pas a pas, totes les operacions de configuració i creació d'un client del **SI PSA**, utilitzant Java i VB.NET. Com a part final es mostren també exemples de codi per a facilitar les tasques de desenvolupament pròpiament dites tant en Java com en VB.Net.

2. WebServices a PSA

2.1 Profiles WebServices

Les funcionalitats suportades pels diferents serveis web del **SI PSA**, es poden agrupar en tres tipus:

- Operacions de signatura
- Operacions de consulta d'informació
- Operacions d'utilitats o realització d'altres operacions

La interacció entre els clients amb el **SI PSA** a través dels serveis web consisteix en l'intercanvi de missatgeria SOAP, el contingut de la qual està definit en el següent conjunt d'esquemes XML (XML Schema) o profiles:

- **dss-directsign.xsd**: Namespace: *urn:catcert:psa:1.0:profiles:dss-directsign*
- **dss-dooperation.xsd**: Namespace: *urn:catcert:psa:1.0:profiles:dss-dooperation*
- **dss-encrypt.xsd**: Namespace: *urn:catcert:psa:1.0:profiles:dss-encrypt*
- **dss-getdata.xsd**: Namespace: *urn:catcert:psa:1.0:profiles:dss-getdata*
- **dss-getsession.xsd**: Namespace: *urn:catcert:psa:1.0:profiles:dss-getsession*
- **dss-signsession.xsd**: Namespace: *urn:catcert:psa:1.0:profiles:dss-signsession*

Aquests esquemes fan ús dels següents profiles importats, no propis del **SI PSA**:

- xmlns:xmlmime=<http://www.w3.org/2005/05/xmlmime>
- xmlns:dsp=<http://uri.etsi.org/2038/v1.1.1#>
- xmlns:ds=<http://www.w3.org/2000/09/xmldsig#>
- xmlns:xs=<http://www.w3.org/2001/XMLSchema>
- xmlns:saml=<http://docs.oasis-open.org/security/saml/v2.0/saml-schema-assertion-2.0.xsd>
- xmlns:dss=<http://docs.oasis-open.org/dss/v1.0/oasis-dss-core-schema-v1.0-os.xsd>

2.2 Seguretat WebServices

La política de seguretat dels serveis publicats pel **SI PSA** requereix la signatura de les dades de la petició i la posterior encriptació total de la mateixa petició, amb aquesta configuració s'aconsegueix la integritat i confidencialitat dels dades. En la següent figura es mostra un exemple genèric de petició amb els requisits de seguretat.

Petició de servei de PSA signada i xifrada

```
<?xml version='1.0' encoding='UTF-8'?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wssse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:exc14n="http://www.w3.org/2001/10/xml-exc-c14n#"
xmlns:xenc="http://www.w3.org/2001/04/xmllenc#">
  <S:Header>
    <wssse:Security S:mustUnderstand="1">
      <wsu:Timestamp xmlns:ns17="http://docs.oasis-open.org/ws-sx/ws-
secureconversation/200512" xmlns:ns16="http://www.w3.org/2003/05/soap-
envelope" wsu:Id="_3">
        <wsu:Created>2008-11-24T17:02:16Z</wsu:Created>
        <wsu:Expires>2008-11-24T17:07:16Z</wsu:Expires>
      </wsu:Timestamp>
      <wssse:BinarySecurityToken xmlns:ns17="http://docs.oasis-
open.org/ws-sx/ws-secureconversation/200512"
xmlns:ns16="http://www.w3.org/2003/05/soap-envelope"
wsu:Id="uuid_6f0e62ee-ef42-446a-9418-aada7caa91d5"
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
soap-message-security-1.0#Base64Binary" ValueType="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-
1.0#X509v3">MIIHeTCCBmGgAwIBAgIQOwbxm5zHySJHn1XkesQmYTANBgkqhkiG9w0BAQUFA
DCCAToxCzAJBgNV...+1RLtxsgKe8HpthPcqr+b+jG0/4lbI8iDT2H7Vck7viXn
</wssse:BinarySecurityToken>
      <xenc:EncryptedKey xmlns:ns17="http://docs.oasis-open.org/ws-sx/ws-
secureconversation/200512" xmlns:ns16="http://www.w3.org/2003/05/soap-
envelope" Id="_5003">
        <xenc:EncryptionMethod
Algorithm="http://www.w3.org/2001/04/xmllenc#rsa-oaep-mgf1p" />
        <ds:KeyInfo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="keyInfo">
          <wssse:SecurityTokenReference>
            <wssse:KeyIdentifier ValueType="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-
1.0#X509SubjectKeyIdentifier" EncodingType="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-soap-message-security-
1.0#Base64Binary">dVE29ysyFW/iD1la3ddePzM6IWo=</wssse:KeyIdentifier>
          </wssse:SecurityTokenReference>
          </ds:KeyInfo>
          <xenc:CipherData>
            <xenc:CipherValue>Pba2dxnwj6MfLrpVylpLK6ble2n786Mry2eUEfAF/PLTWwgKHdZTXxm
boda jKeqTp/ih32LxMtfCzXW/BM9hlKBaBYBhm6/6dKzJo+oveEGn29C32I3koKP61jPeV8Ce
DzsJJRv/eBFXYRX+fJqMYa1Y48nqx/MFLMpyI1ZqwuU=</xenc:CipherValue>
          </xenc:CipherData>
          <xenc:ReferenceList>
            <xenc:DataReference URI="#_5004" />
          </xenc:ReferenceList>
        </xenc:EncryptedKey>
        <ds:Signature xmlns:ns17="http://docs.oasis-open.org/ws-sx/ws-
secureconversation/200512" xmlns:ns16="http://www.w3.org/2003/05/soap-
envelope" Id="_1">
```

```

<ds:SignedInfo>
  <ds:CanonicalizationMethod
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
    <excl4n:InclusiveNamespaces PrefixList="wsse S" />
  </ds:CanonicalizationMethod>
  <ds:SignatureMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
  <ds:Reference URI="#_5002">
    <ds:Transforms>
      <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
        <excl4n:InclusiveNamespaces PrefixList="S" />
      </ds:Transform>
    </ds:Transforms>
    <ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
    <ds:DigestValue>WqZ+I7kKJMd7x6ifmiL8pK+IK/0=</ds:DigestValue>
  </ds:Reference>
  <ds:Reference URI="#_3">
    <ds:Transforms>
      <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
        <excl4n:InclusiveNamespaces PrefixList="wsu wsse S" />
      </ds:Transform>
    </ds:Transforms>
    <ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
    <ds:DigestValue>4psAFYViU3oQUjLTgmAtcjcoXlc=</ds:DigestValue>
  </ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>udiEq8zfev3EKVPB5d/jx1McVzmc+t7HK87JjZnzykEDHdZP0KVWEJ
xaKnYRVk8qZlTfpf86WglULTuqvc+BprCKV6aOM9LgoIeu5DPnRrpXk5UOYDAw8tPlnywcHg9
T05YA+3Cw9Vyd1+kncQEYSf7vddhCmWtb0CsGLUmasMs=</ds:SignatureValue>
  <ds:KeyInfo>
    <wsse:SecurityTokenReference>
      <wsse:Reference ValueType="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3"
URI="#uuid_6f0e62ee-ef42-446a-9418-aada7caa91d5" />
    </wsse:SecurityTokenReference>
  </ds:KeyInfo>
</ds:Signature>
</wsse:Security>
</S:Header>
<S:Body wsu:Id="_5002">
  <xenc:EncryptedData xmlns:ns17="http://docs.oasis-open.org/ws-sx/ws-
secureconversation/200512" xmlns:ns16="http://www.w3.org/2003/05/soap-
envelope" Type="http://www.w3.org/2001/04/xmenc#Content" Id="_5004">
    <xenc:EncryptionMethod
Algorithm="http://www.w3.org/2001/04/xmenc#aes128-cbc" />
    <xenc:CipherData>
      <xenc:CipherValue>cWlpj5L6bRJhLyTQsAQyoiDRbOZRSuOkiwC6q6ntlf+T0W+IuPrDuAU
B2ih/0eC+kJuZNMRz4EOXOsf2GikEOGW2JebFglg/8XIjLA/NiRRV6y8Jn7UEUWZdC9HQv91X
x6ioEc/Ngnv+S1x6EgaYwVcetjTs4S3Rl5JM7VbZI2jMlCQtW4VQpDLgN8/2WE2lgXTFRmO8I
GLgpFvQFPJ7fXe32/31ThR+2HuT/lQ9rJDne0gUAXS6ZBWLNPzQNKjQHL0xQ2YafHM/L2KYJ4
zUtxgP90YWkbUCK8Gj5XaRkGZXe8RaLOMoF90L9XHwzx2JSw2ZcGaJ4/haFnTA47rLhYgMHQe
FXPhyZocjkgrg6EIVxGNkuKSrPeNLMkEQCCMXXkaXsN96T4ljmRK4X/3ztA==
      </xenc:CipherValue>
    </xenc:CipherData>
  </xenc:EncryptedData>

```

En el document de seguretat (**06-0435-DS-0001-XX-Security_PSA.doc**) es dona informació detallada sobre tots els aspectes referents a la seguretat dels Serveis Web.

3. Serveis de signatura

El **SI PSA** disposa de 3 serveis de signatura. La diferència principal de cadascun d'ells és on es realitza la signatura. El **SI PSA** permet signar, doncs, a 3 llocs diferents que són:

- **SignatureAtPSA:** Signatura al **SI PSA** on la clau privada estarà en possessió del **SI PSA**. El procés de signatura es realitza íntegrament mitjançant serveis web.
- **SignatureAtBrowser:** Signatura en Browser dins el context de web del **SI PSA**, la clau privada estarà en possessió de l'usuari. La invocació al servei web es realitza per iniciar un context, la resta del procés la realitza l'usuari en una URL del **SI PSA** on, mitjançant un applet es realitza la signatura.
- **SignatureAtSignatureManager:** Signatura en Browser dins el context d'una aplicació de gestió, la clau privada estarà en possessió de l'usuari. El procés de signatura es realitza íntegrament mitjançant una conversa entre l'aplicació de gestió i el **SI PSA** a través de serveis web.

3.1 SignatureAtBrowser

3.1.1 Funcionalitat

El servei de Signatura en Browser permet realitzar signatures en Browser en el context web del **SI PSA** i amb la clau en possessió de l'usuari i consta de 2 parelles de peticions/resposta:

1. `InitSignDocumentAtBrowserRQST/ SignDocumentAtBrowserRSPNS`
2. `ContinueSignDocumentAtBrowserRQST/ ContinueSignDocumentAtBrowserRSPNS`

La petició `InitSignDocumentAtBrowserRQST` és necessària i permet a un usuari d'una aplicació client del **SI PSA**, realitzar el procés de creació de context de signatura electrònica. El **SI PSA** respondrà amb una URL a la qual, amb la presentació de les seves credencials, l'usuari podrà dur a terme la signatura. Per tant, l'usuari ha de disposar de la clau associada al seu certificat personal correctament instal·lada al repositori de claus del seu navegador Web.

La petició de `ContinueSignDocumentAtBrowserRQST` només serà necessària en cas de que la política de procediment escollida en el `InitSignDocumentAtBrowser` consisteixi en N signants. La petició permetrà recuperar el context de signatura en l'estat que l'hagi deixat l'anterior signant. Quan l'aplicació envii la petició de `ContinueSignDocumentAtBrowserRQST` estarà indicant al **SI PSA** que recuperi el context i es prepari pel següent signant. El **SI PSA** respondrà amb la URL corresponent i el següent signant podrà procedir a signar.

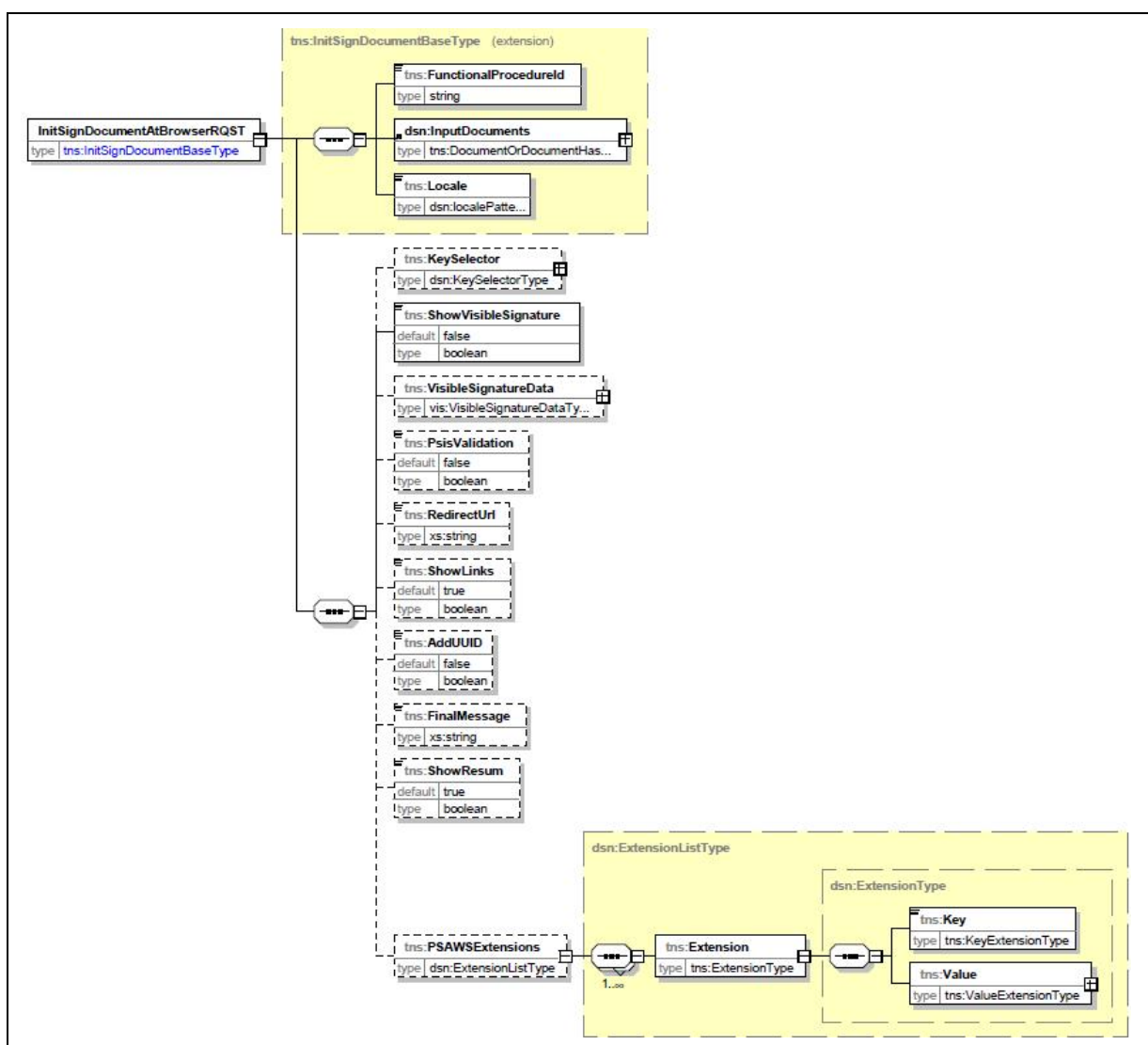
El servei i la seva informació es troba definit al profile `urn:catcert:psa:1.0:profiles:dss-signsession`.

3.1.2 Conversa

[Diagrama]

3.1.3 Descripció Missatges

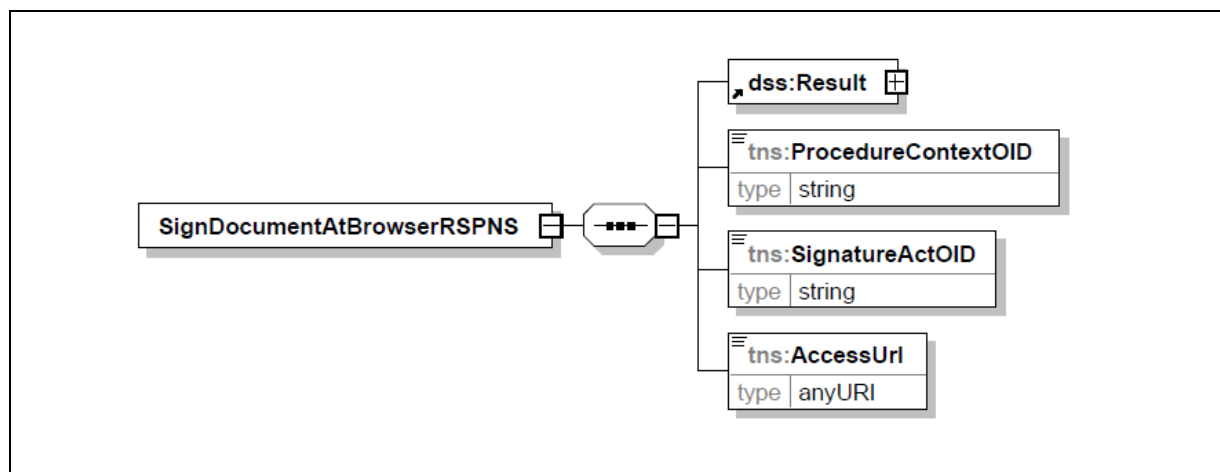
1. Petició de creació de context de signatura amb certificat en possessió de l'usuari `InitSignDocumentAtBrowserRQST`.



006_20070524

<i>FunctionalProcedureId</i>	Identificador del procediment per a la realització de la signatura
<i>InputDocuments</i>	Col·lecció de documents a signar.
<i>Locale</i>	Locale per a la definició del llenguatge dels missatges descriptius d'operació
<i>KeySelector</i>	KeySelector (Certificat o Issuer i Serial) associat al procés de signatura a realitzar.
<i>ShowVisibleSignature</i>	Indica si s'ha d'inserir Signatura visible o no (només es tindrà en compte per aquells formats que la suportin com PDF)
<i>VisibleSignatureData</i>	Configuració de la signatura visible a crear
<i>PsisValidation</i>	Especifica si la signatura final s'ha de enviar a validar a PSIS
<i>RedirectURL</i>	URL a la qual s'ha de redirigir a l'usuari després de finalitzar el procés de signatura
<i>ShowLinks</i>	Indica si es mostren o no els links al document per signar i signat
<i>AddUUID</i>	Indica si s'inclou l'UUID del document signat en la redirecció
<i>FinalMessage</i>	Missatge descriptiu a mostrar a l'usuari a la finalització del procés de signatura
<i>ShowResum</i>	Indica si s'ha de mostrar la plana resumen a la finalització del procés de signatura.
<i>PSAWSExtensions</i>	Llista d'extensions de la petició. En aquest cas s'utilitzarà per a informar a PSA de les referències dinàmiques de la signatura, en cas que apliqui. La clau (<i>Key</i>) a utilitzar és <i>DynamicSignatureReferences</i> , i el valor (<i>Value</i>) és <i>SignatureReferencesType</i> .

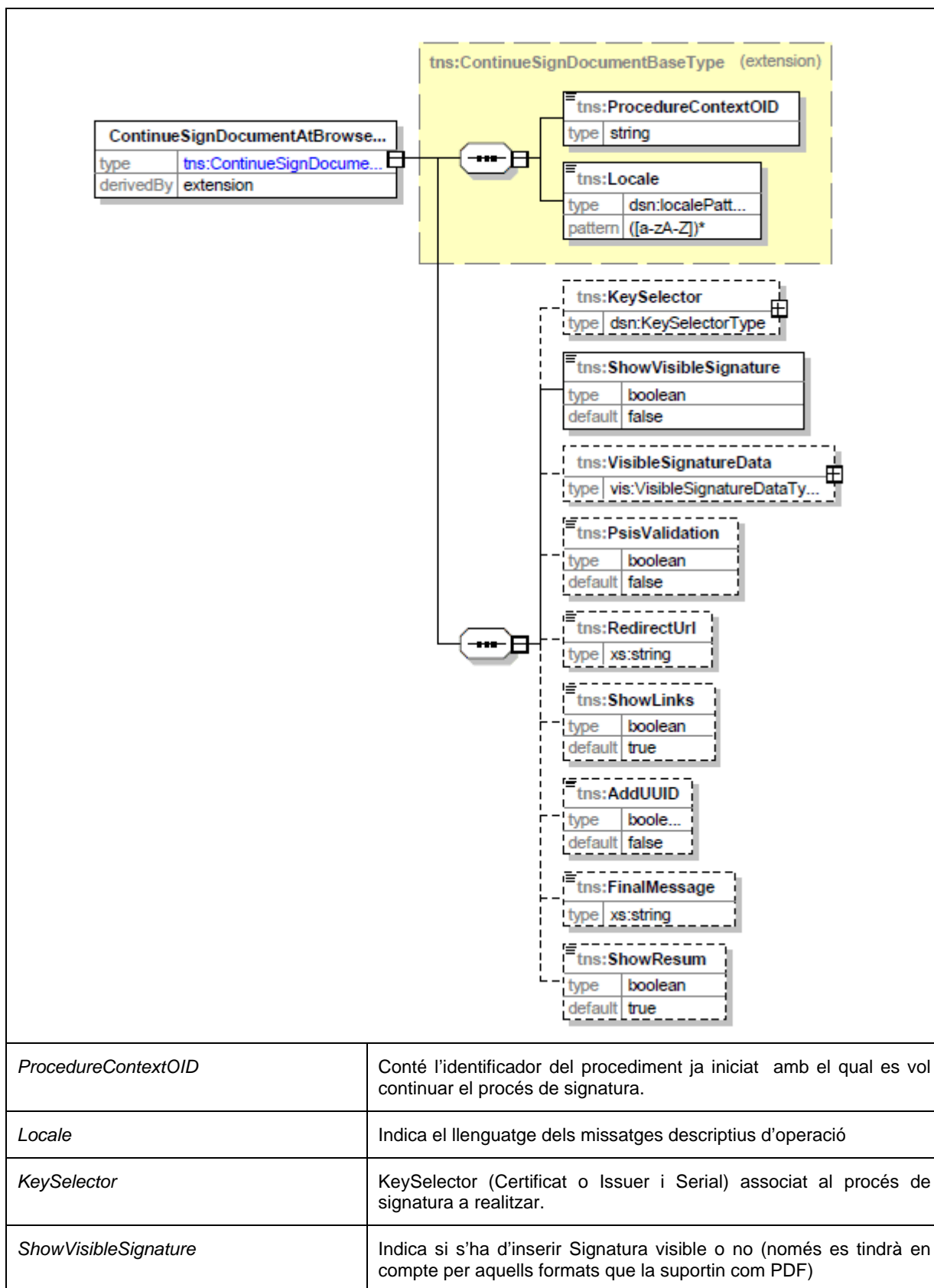
2. Resposta a les peticions de signatura amb certificat en possessió de l'usuari **SignDocumentAtBrowserRSPNS.**



006_20070524

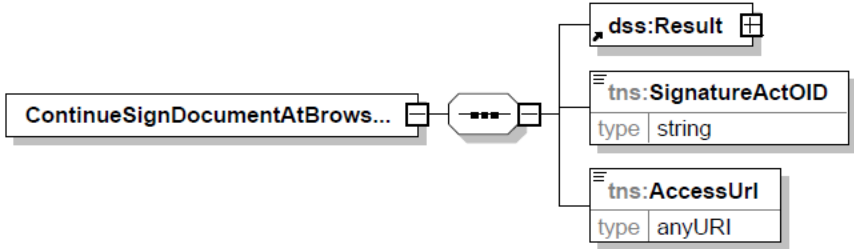
<i>Result</i>	Indica el resultat de l'operació sol·licitada
<i>ProcedureContextOID</i>	Identificador del context de signatura iniciat necessari per als següents peticions de signatura associades al mateix context.
<i>SignatureActOID</i>	Identificador del context de signatura
<i>AccesUrl</i>	URL d'accés al frontal de signatura del PSA per a la realització d'aquesta.

3. Petició de continuació de context de signatura amb certificat en possessió de l'usuari, **ContinueSignDocumentAtBrowserRQST**



<i>VisibleSignatureData</i>	Configuració de la signatura visible a crear
<i>PsisValidation</i>	Especifica si la signatura final s'ha de enviar a validar a PSIS
<i>RedirectURL</i>	URL a la qual s'ha de redirigir a l'usuari després de finalitzar el procés de signatura
<i>ShowLinks</i>	Indica si es mostren o no els links al document per signar i signat
<i>AddUUID</i>	Indica si s'inclou l'UUID del document signat en la redirecció
<i>FinalMessage</i>	Misatge descriptiu a mostrar a l'usuari a la finalització del procés de signatura
<i>ShowResum</i>	Indica si s'ha de mostrar la plana resumen a la finalització del procés de signatura.

4. Resposta a les peticions continuació de context de signatura amb certificat en possessió de l'usuari **ContinueSignDocumentAtBrowserRSPNS**.

	
<i>Result</i>	Indica el resultat de l'operació sol·licitada
<i>SignatureActOID</i>	Identificador del context de signatura
<i>AccesUrl</i>	URL d'accés al frontal de signatura del PSA per a la realització d'aquesta.

3.2 SignatureAtPSA

3.2.1 Funcionalitat

Aquest servei permet a una aplicació client del **SI PSA**, realitzar el procés de creació de N procediments de signatura electrònica. El **SI PSA** disposa de la clau sol·licitada (KeySelector) per l'aplicació de gestió i realitza la/les signatura/signatures amb aquesta.

El servei consta de dues parelles de petició/resposta:

1. `MultiInitSignDocumentAtPSARQST/ MultiSignDocumentAtPSARSPNS`
2. `MultiContinueSignDocumentAtPSARQST/ MultiSignDocumentAtPSARSPNS`

Per realitzar la signatura al **SI PSA** l'aplicació de gestió haurà d'invocar, en primer lloc la petició de `MultiInitSignDocumentAtPSARQST`. Aquesta petició és multiprocedimental, és a dir, permet fet N peticions de signatura amb el mateix KeySelector simultàniament.

Si s'han iniciat prèviament procediments de signatura amb més d'un signant, aquests es podran continuar o finalitzar (si és el darrer signant) mitjançant `MultiContinueSignDocumentAtPSARQST`. La petició a aquest servei recuperarà cadascun dels contextos sol·licitats i intentarà realitzar la signatura amb el nou firmant per cada un d'ells.

La resposta a tots dos casos, informarà, per cadascun dels procediments de signatura, quins han anat bé i quins malament.

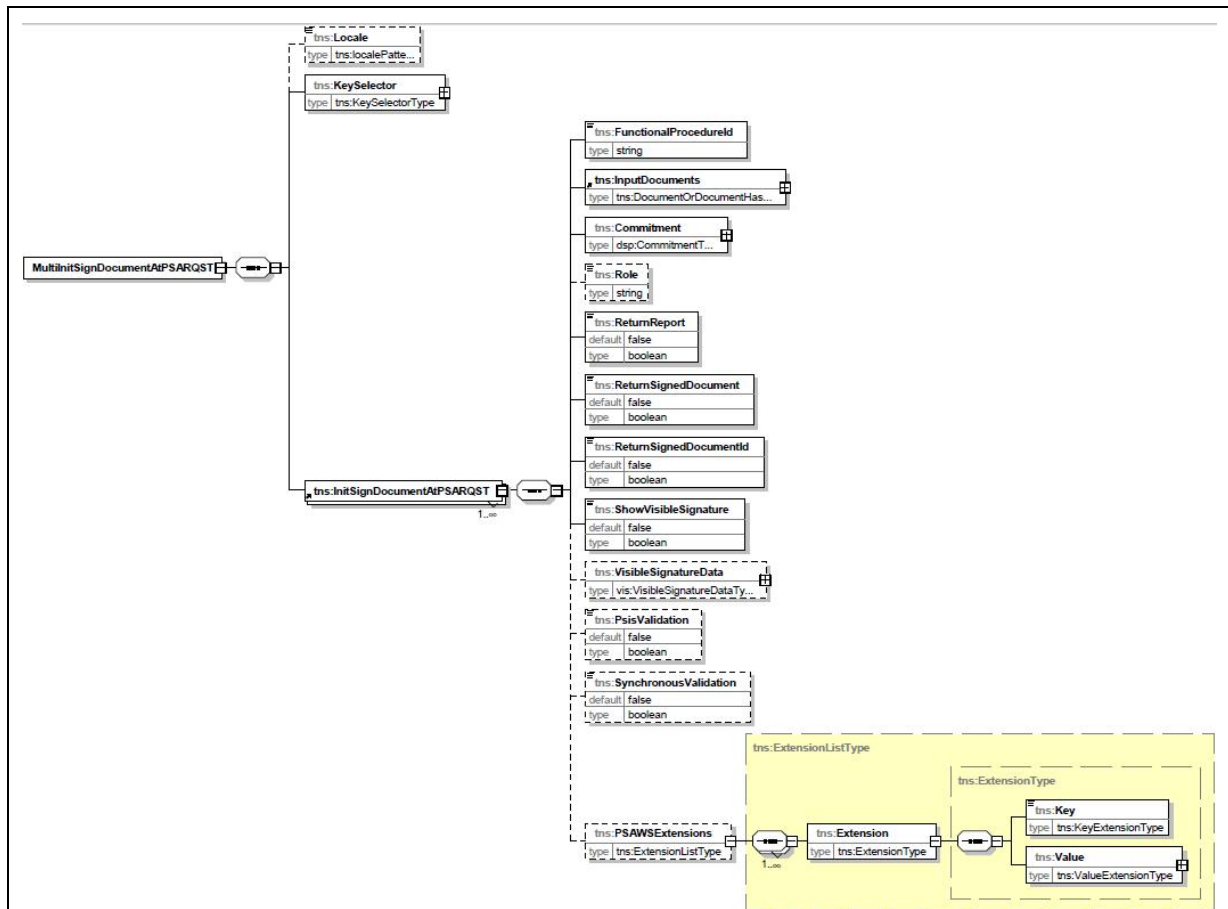
El servei i la seva informació es troba definit al profile `urn:catcert:psa:1.0:profiles:dss-directsign`.

3.2.2 Conversa

[Diagrama]

3.2.3 Descripció Missatges

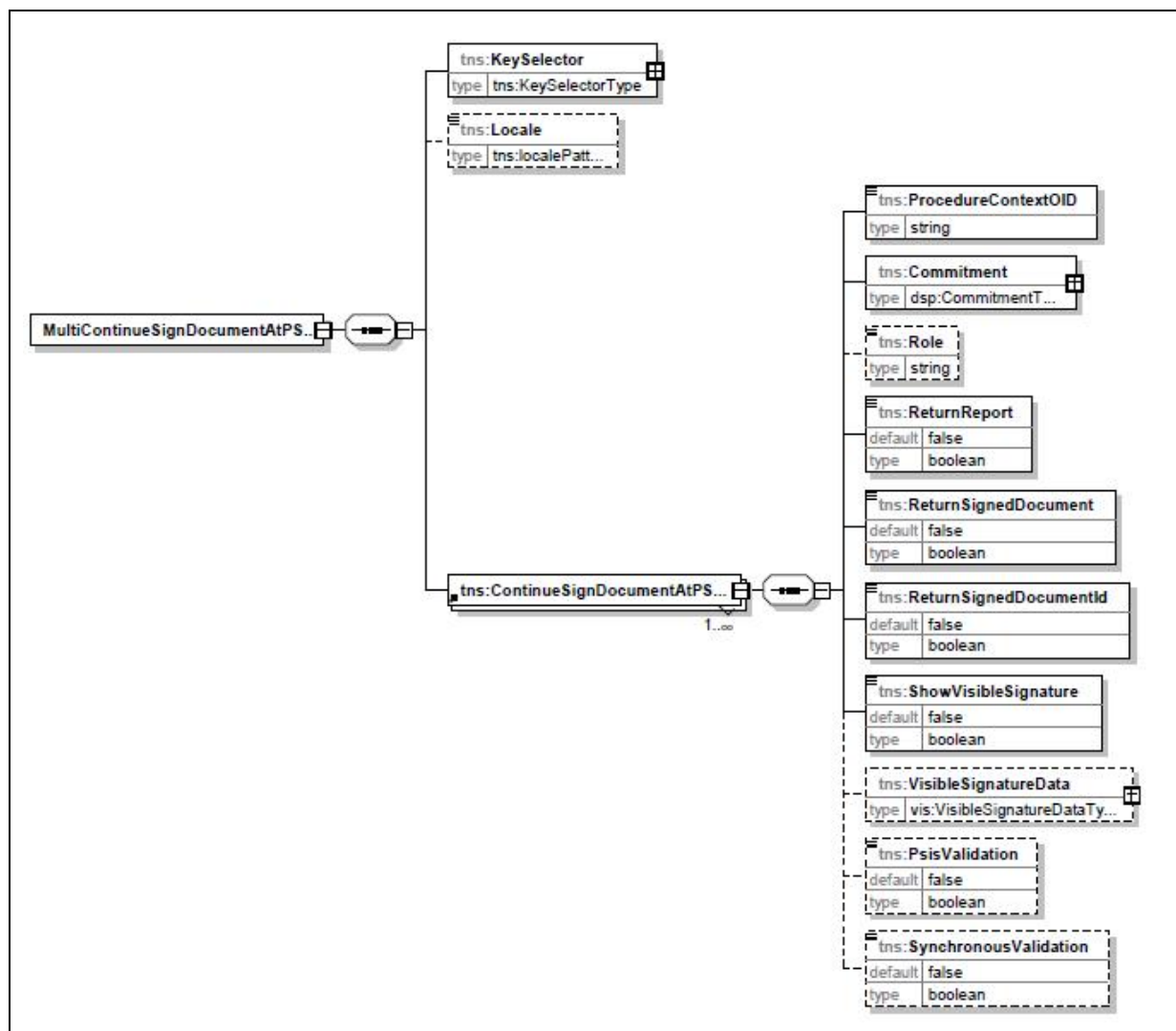
1. Petició de creació de signatura amb certificat al **SI PSA**
`MultiInitSignDocumentAtPSARQST`.



<i>Locale</i>	Locale per a la definició del llenguatge dels missatges descriptius d'operació
<i>KeySelector</i>	Identificador del certificat associat al procés de signatura a realitzar.
<i>FunctionalProcedureId</i>	Identificador del procediment per a la realització de la signatura
<i>InputDocuments</i>	Col·lecció de documents a signar.
<i>Commitment</i>	Compromís adquirit pel signant dintre d'aquest context de signatura.
<i>Role</i>	Rol corresponent al signant.
<i>ReturnReport</i>	Indica si la resposta ha de incloure el comprovant de la signatura realitzada.
<i>ReturnSignedDocument</i>	Indica si la resposta ha de incloure el document signat.
<i>ReturnSignedDocumentID</i>	Indica si la resposta ha de incloure l'identificador del document signat.
<i>ShowVisibleSignature</i>	Indica si s'ha d'inserir Signatura visible o no (només es tindrà en compte per aquells formats que la suportin com PDF)

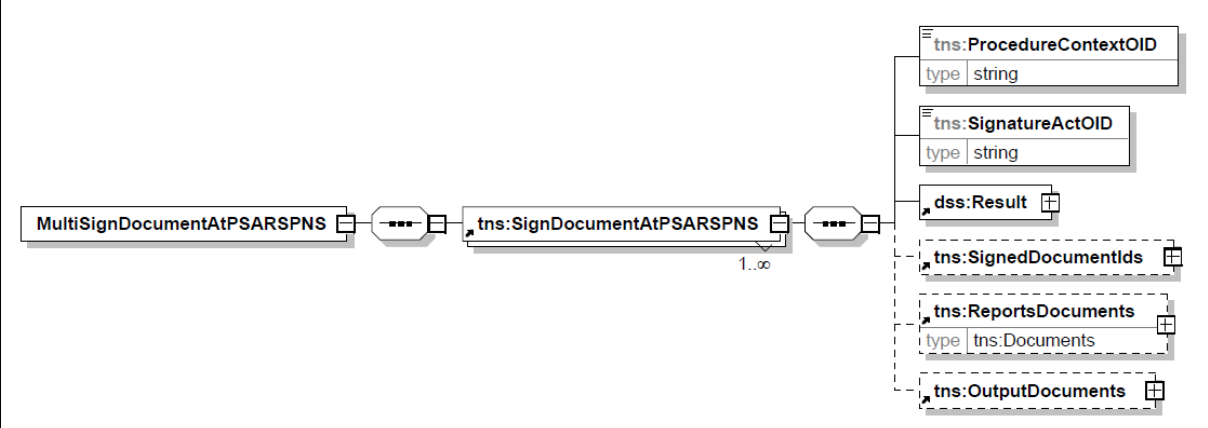
<i>VisibleSignatureData</i>	Configuració de la signatura visible a crear
<i>PsisValidation</i>	Indica si es vol validar la signatura contra PSIS.
<i>SynchronousValidation</i>	Indica si la validació contra PSIS es farà de forma síncrona.
<i>PSAWSExtensions</i>	<p>Llista d'extensions de la petició. En aquest cas s'utilitzarà per a informar a PSA de les referències dinàmiques de la signatura, en cas que apliqui.</p> <p>La clau (Key) a utilitzar és <i>DynamicSignatureReferences</i>, i el valor (Value) és <i>SignatureReferencesType</i>.</p>

2. Petició de continuació de context de signatura amb certificat al SI PSA **MultiContinueSignDocumentAtPSARQST**



<i>ProcedureContextOID</i>	Conté l'identificador del procediment ja iniciat amb el qual es vol continuar el procés de signatura.
<i>Locale</i>	Indica el llenguatge dels missatges descriptius d'operació
<i>Commitment</i>	Compromís adquirit pel signant dintre d'aquest context de signatura.
<i>Role</i>	Rol corresponent al signant.
<i>KeySelector</i>	Identificador del certificat associat al procés de signatura a realitzar.
<i>ReturnReport</i>	Indica si la resposta ha de incloure el comprovant de la signatura realitzada.
<i>ReturnSignedDocument</i>	Indica si la resposta ha de incloure el document signat.
<i>ReturnSignedDocumentID</i>	Indica si la resposta ha de incloure l'identificador del document signat.
<i>ShowVisibleSignature</i>	Indica si s'ha d'inserir Signatura visible o no (només es tindrà en compte per aquells formats que la suportin com PDF)
<i>VisibleSignatureData</i>	Configuració de la signatura visible a crear
<i>PsisValidation</i>	Indica si es vol validar la signatura contra PSIS.
<i>SynchronousValidation</i>	Indica si la validació contra PSIS es farà de forma síncrona.

3. Resposta de signatura amb certificat al SI PSA **MultiSignDocumentAtPSARSPNS.**

	
<i>ProcedureContextOID</i>	Identificador del context de signatura iniciat necessari per al les següents peticions de signatura associades al mateix context.
<i>SignatureActOID</i>	Identificador del context de signatura

<i>Result</i>	Indica el resultat de l'operació realitzada
<i>SignedDocumentIDs</i>	Identificadors dels documents signats si ho hem especificat en la petició
<i>ReportsDocuments</i>	Comprovants de la signatura realitzada si ho hem especificat a la petició
<i>OutputDocuments</i>	Documents signats si ho hem especificat a la petició.

3.3 SignatureAtSignatureManager

3.3.1 Funcionalitat

Aquest servei permet iniciar N contextos de signatura en Browser dins el context d'una aplicació de gestió (PortaSignatures) i el **SI PSA**. En funció de la informació proporcionada en aquesta petició, la conversa de signatura entre el PortaSignatures i el **SI PSA** requerirà d'un intercanvi major o menor de missatges tal i com es descriu en el següent apartat de *Conversa*.

El servei i la seva informació es troba definit al profile **urn:catcert:psa:1.0:profiles:dss-directsign**.

3.3.2 Conversa

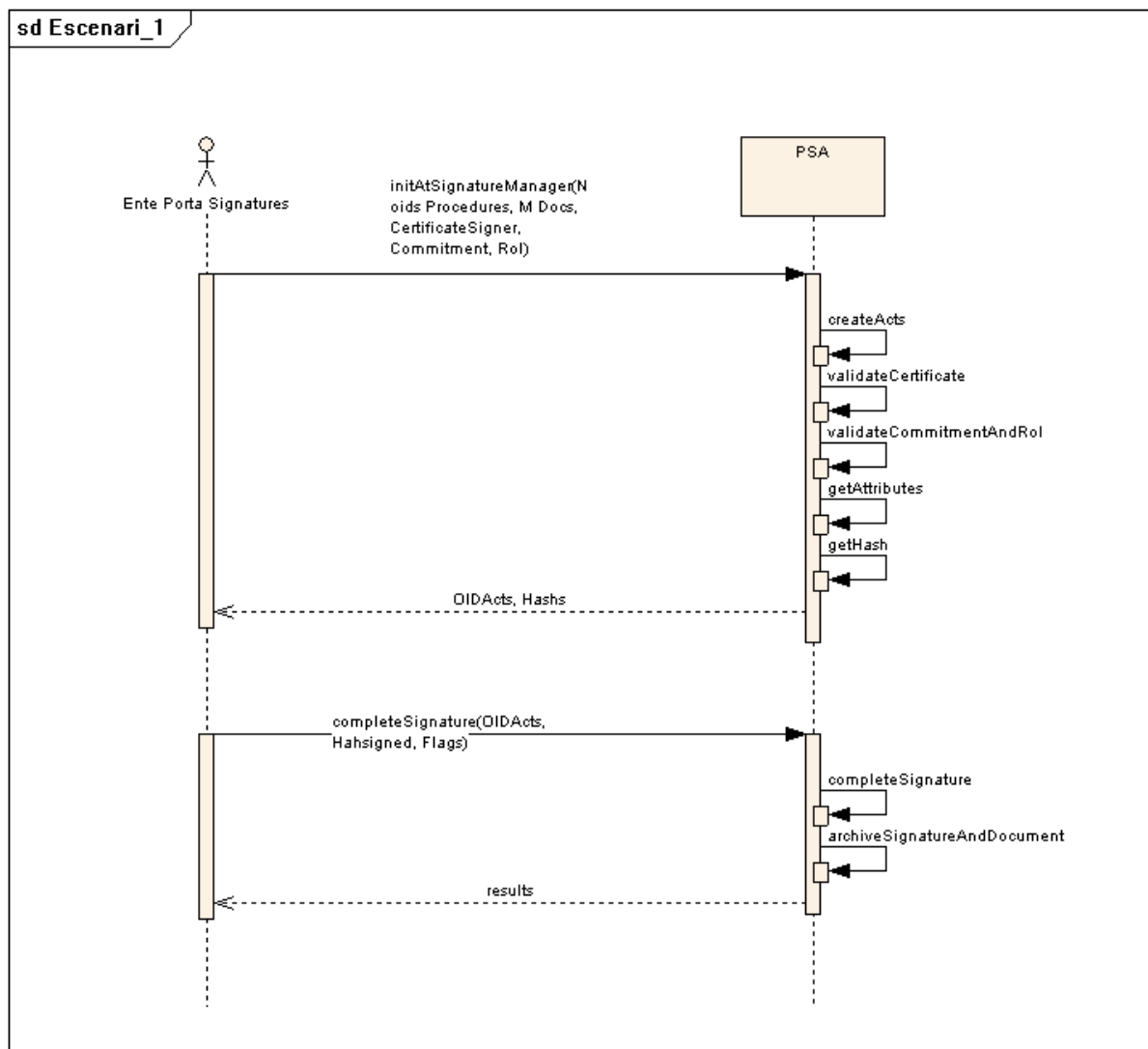
El **SI PSA** ofereix una sèrie de serveis web que possibiliten la conversa per la realització de signatura en browser per part d'una aplicació de gestió (portasignatures). Es presenten interfícies amb capacitat per treballar amb N procediments de signatura simultàniament. A continuació es detallaran 3 escenaris possibles en funció del moment en què l'aplicació de gestió proporciona diferents informacions que requereix el **SI PSA** per signar.

És molt important en tots els escenaris realitzar la gestió de emmagatzemar l'ordre d'enviament dels documents i aportar la informació corresponent amb el mateix ordre, amb el fi de conèixer posteriorment quina signatura resultant correspon a cadascú dels documents enviats!!

3.3.2.1 Escenari 1

L'escenari 1 és el cas en què es requerirà un menor intercanvi de missatges entre el **SI PSA** i el Portasignatures ja que aquest, proporciona tant el Certificat com el Commitment i el Rol en la petició inicial de `InitAtSignatureManager`.

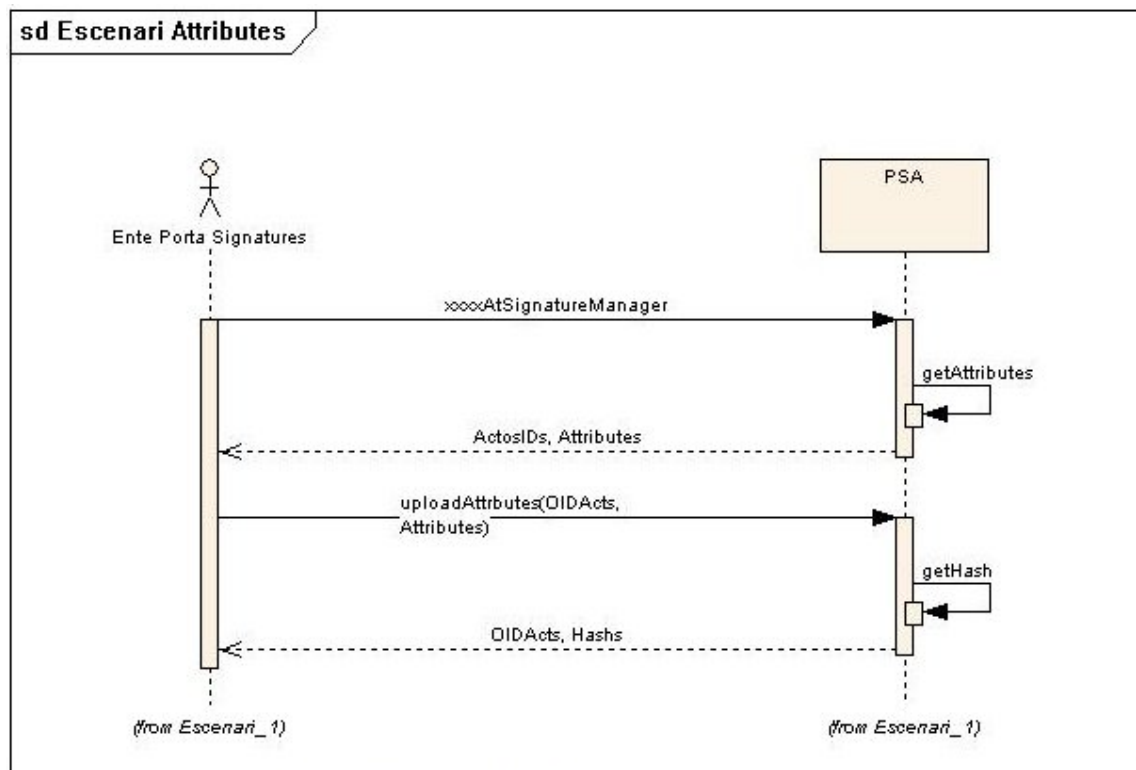
Suposant d'altra banda que no és necessari que el portasignatures proporcioni cap valor d'atribut al **SI PSA**, tenim el següent escenari:



El Portasignatures realitza la petició de `InitSignatureAtSignatureManager` incloent el Certificat i el Commitment i el rol. El **SI PSA** realitza tots els passos previs a la signatura i retorna els hashes al Portasignatures perquè aquest els signi.

El Portasignatures realitzarà llavors una petició WS de `CompleteSignature` amb els hashes signats i el **SI PSA** completarà la signatura tornant el resultat.

Si es requerís que el PortaSignatures proporcionés algú atribut al **SI PSA**, l'escenari es veuria modificat de la següent forma:



Aquest esquema és genèric per tots els escenaris (Escenari 1, Escenari 2 i Escenari 3) i mostra com es modifica la conversa en cas que el PortaSignatures hagi de passar atributs al SI PSA. S'observa com apareix un nou servei web, `UploadAttributes` que permet informar dels atributs mitjançant una petició. Un cop s'ha fet la petició, la resta de la conversa segueix normalment, tal com defineix l'escenari.

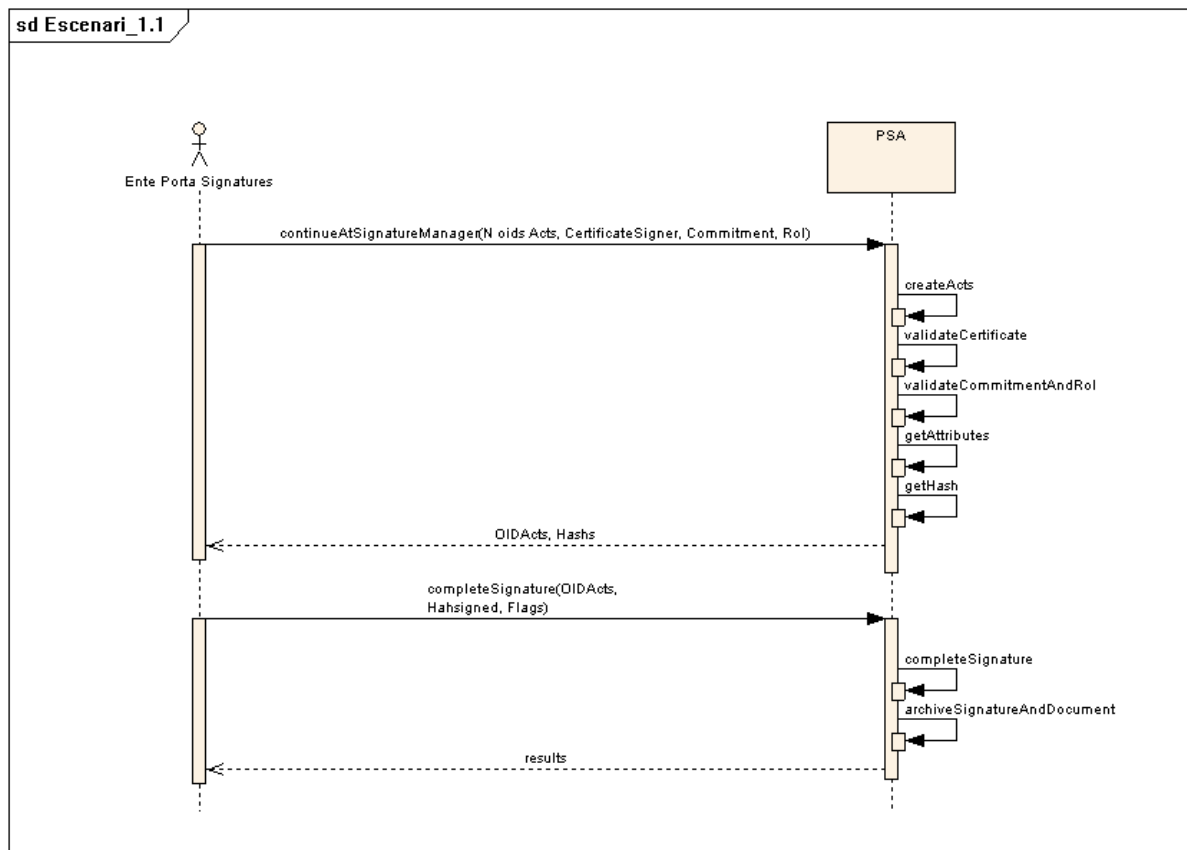
Aquest es el cas en el qual l'usuari ha de establir el valor d'algú atribut signat¹, una vegada s'ha fet l'operació de informació amb el servei `UploadInfoRitual`. Els atributs que es necessiten per a la continuació del procés de signatura s'especifiquen en la resposta de la petició de `UploadInfoRitual`.

Amb tota questa informació, el sistema ya pot generar els hashos per a la signatura (xifrat) dels mateixos pel client.

En cas que la política requereixi de més signants, el PortaSignatures podrà continuar el context de signatura mitjançant una crida a

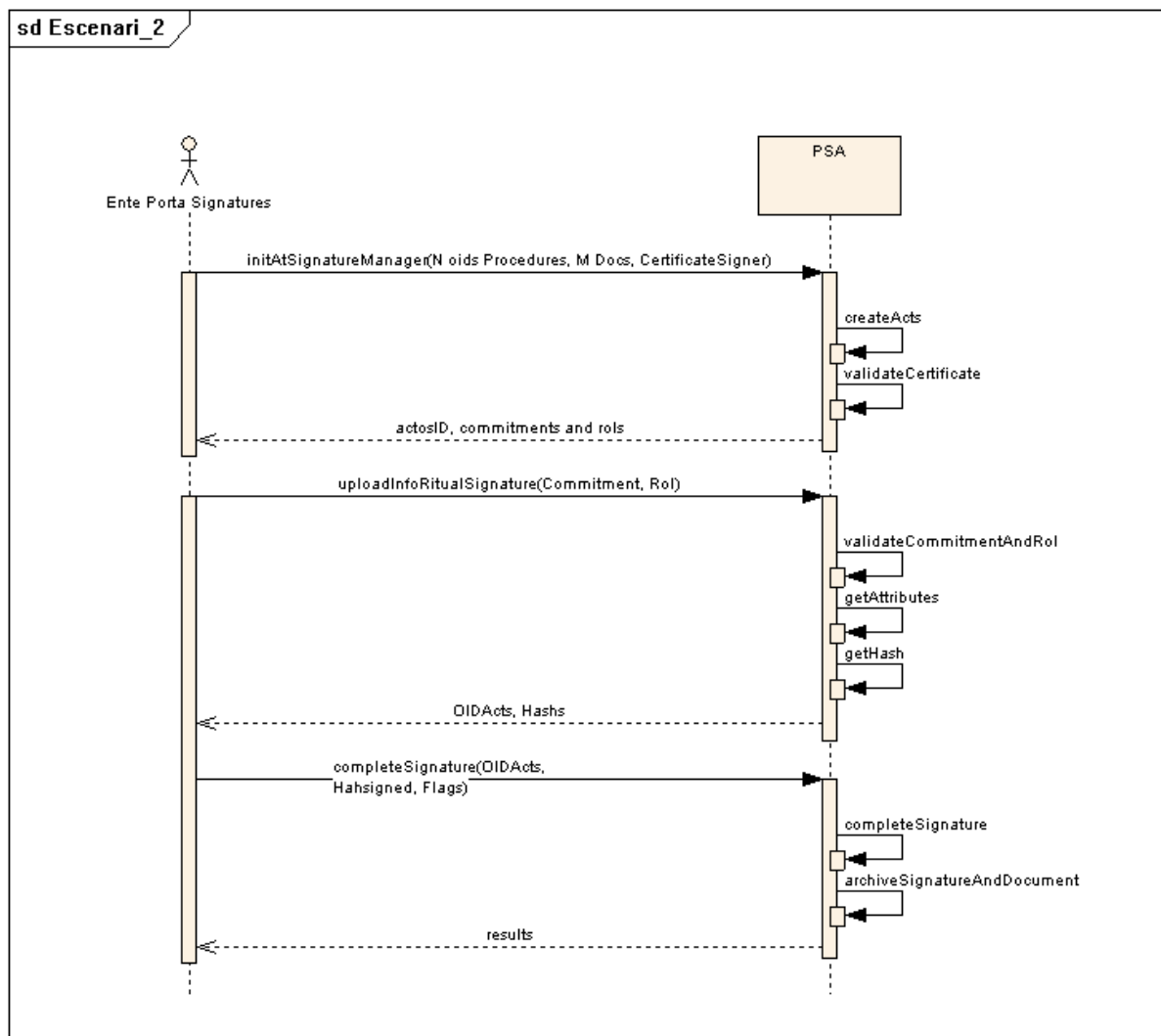
¹ Aquesta situació no és habitual

ContinueSignatureAtSignatureManager. Suposant que ens proporcionen tant el Certificat com el Commitment i el Rol el diagrama resultant és el següent:



3.3.2.2 Escenari 2

Es pot donar una altra possibilitat i és que el PortaSignatures no informi del Commitment i del Role en la petició inicial. Aquesta informació és requerida per signar de manera que la conversa serà una mica més complexa ja que requerirà un pas intermig on el Portasignatures haurà de proporcionar aquestes dades al **SI PSA**:

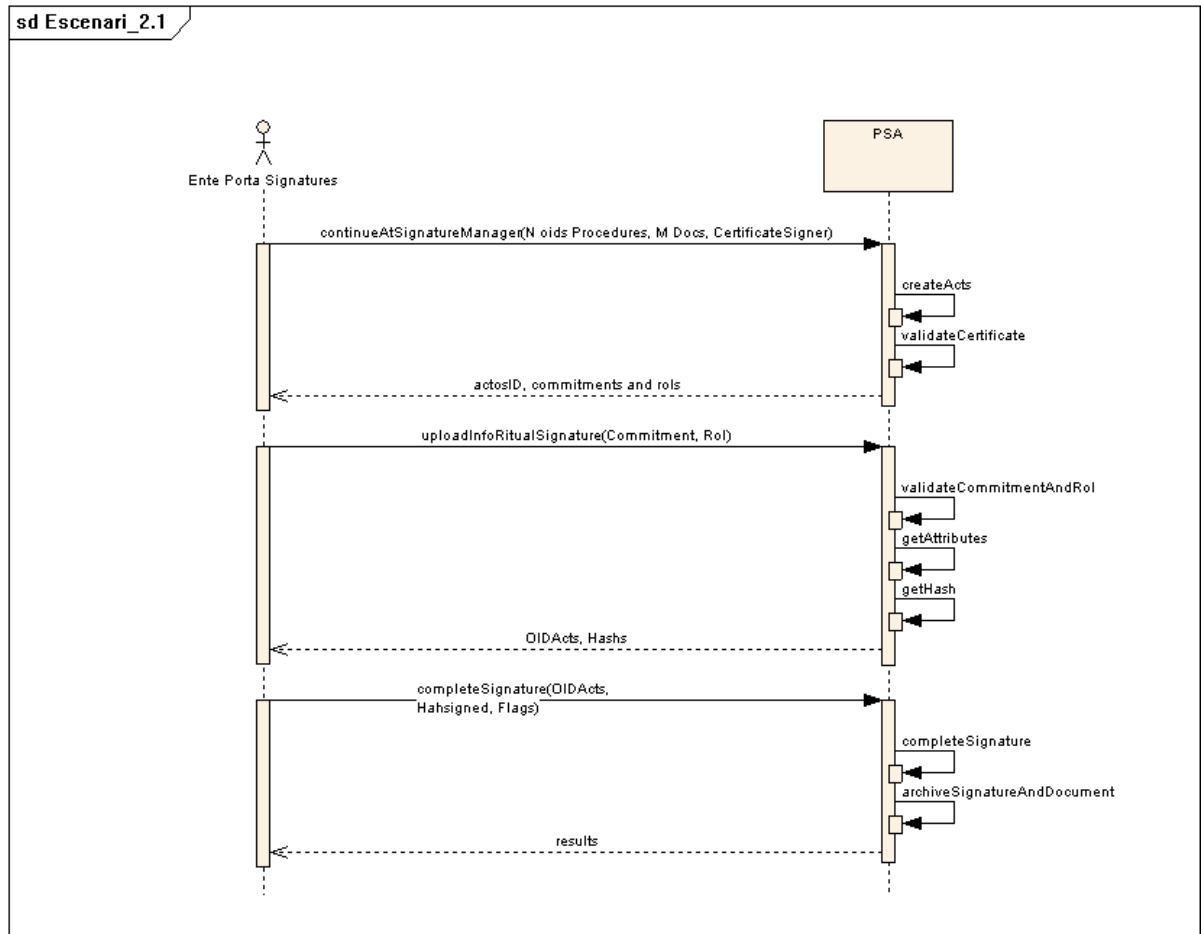


En aquesta conversa s'observa un pas intermig. Un cop el **SI PSA** rep la petició Init, després de crear els actes i validar el certificat retorna al PortaSignatures un llistat de Commitments i Roles possibles per la política de procediment seleccionada pel Portasignatures. A continuació, el Portasignatures informará el **SI PSA** del Commitment i Role escollit mitjançant el WS UploadInfoRitualSignature.

Per a establir les restriccions de política a utilitzar, es necessari conèixer en tots els casos el Commitment (pot ser també el Role), per això, per a poder realitzar l'operació de generació i retorn dels hashos, s'ha de conèixer el Certificat, Commitment i Role

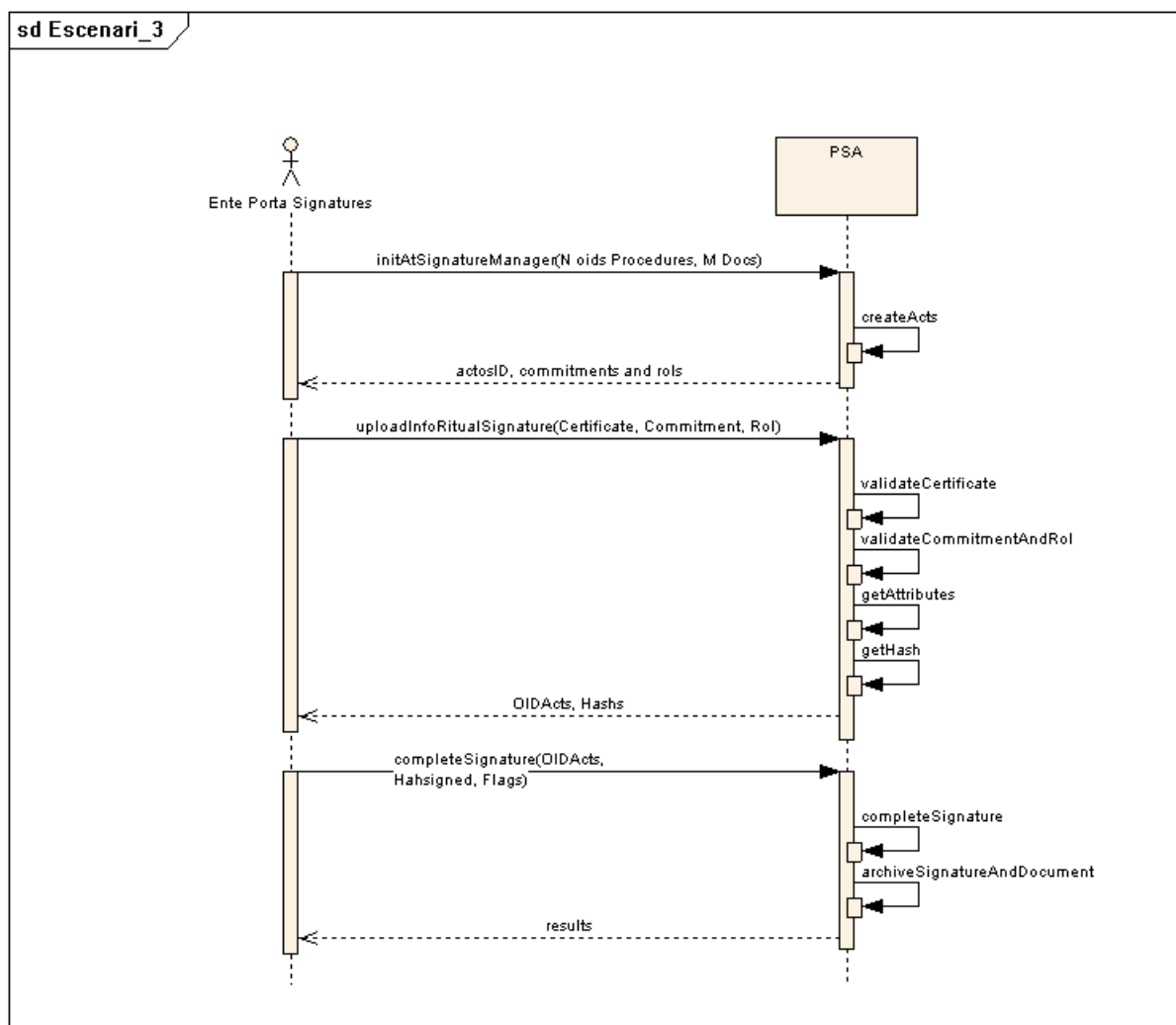
Els passos que segueixen són els mateixos que els de l'Escenari 1.

El Continue presentaria el següent aspecte:



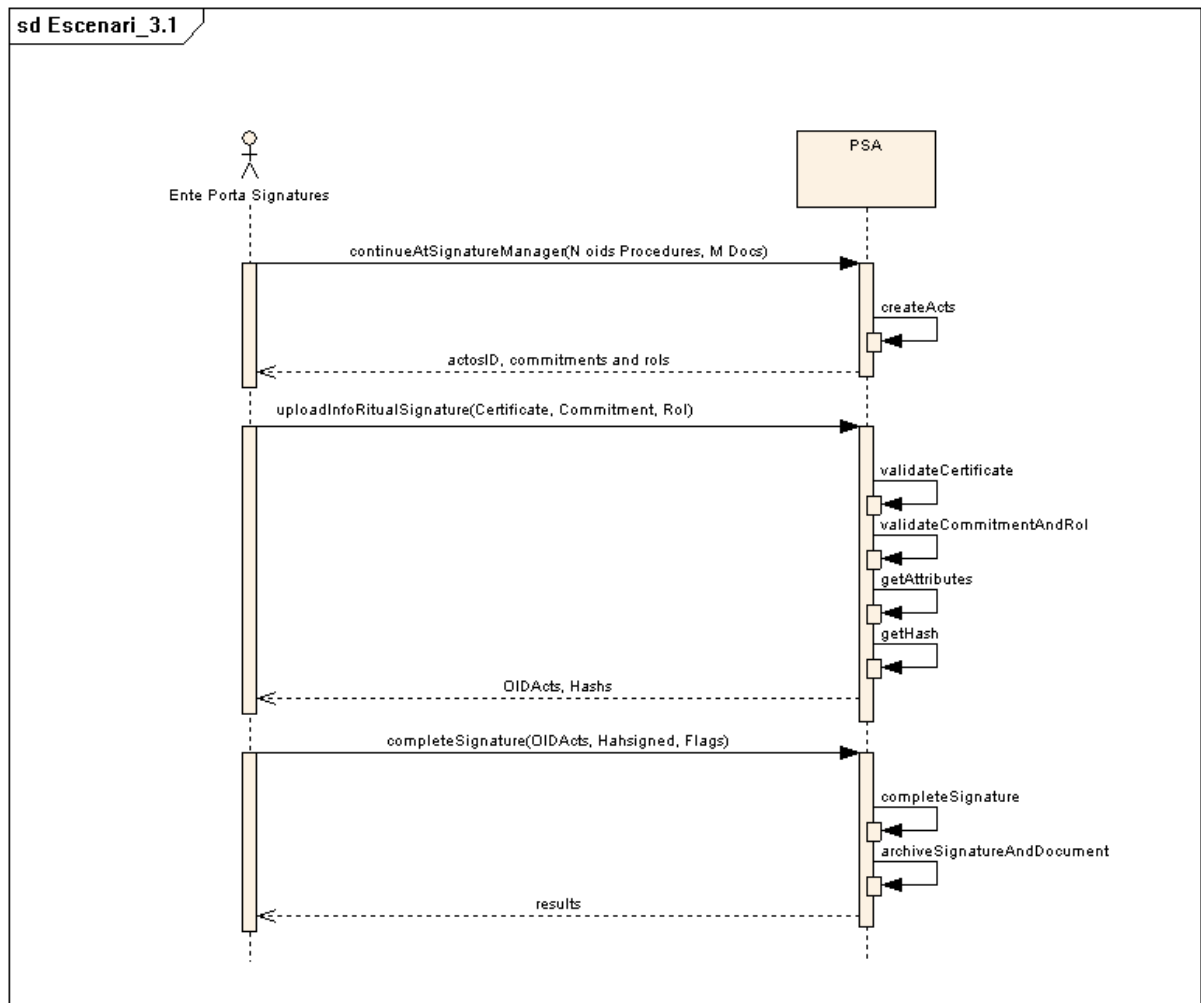
3.3.2.3 Escenari 3

Finalment queda la opció que a la petició inicial no s'informi ni del Certificat ni del Commitment i el Role. La conversa en tal cas seria la següent:



Es pot observar que la única diferència respecte l'Escenari 2 és que el Certificat, en comptes de passar-se a la petició de Init es passa posteriorment a la petició d'UploadInfoRitualSignature. La resta és igual a l'Escenari 2.

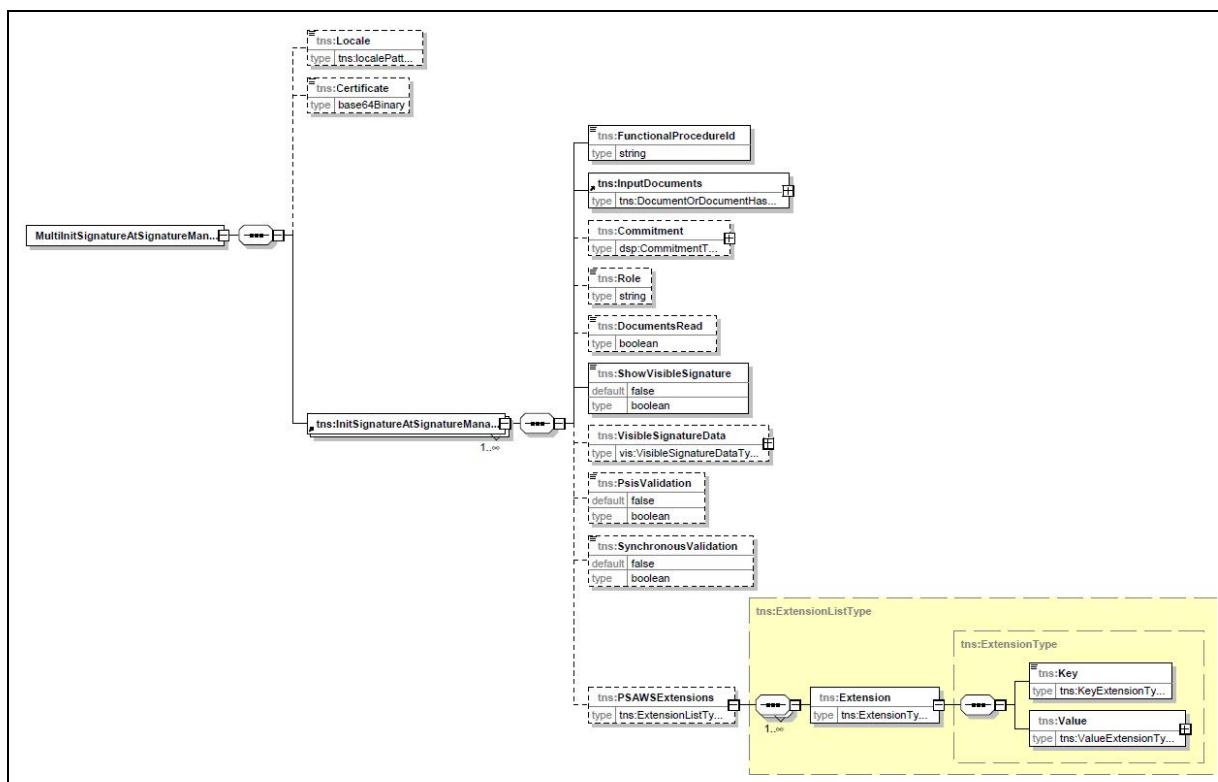
La petició de Continue presentaria el següent aspecte:



3.3.3 Descripció Missatges

A continuació es detallaran el format de les peticions i respostes implicades en la signatura per conversa entre el PortaSignatures i el **SI PSA** (signatura SignatureAtSignatureManager).

1. **MultiInitSignatureManagerRQST**: Petició de creació de N contexts de signatura:



<i>Locale</i>	Indica el llenguatge dels missatges descriptius d'operació
<i>Certificate</i>	Certificat associat al procés de signatura a realitzar.
<i>FunctionalProcedureId</i>	Identificador del procediment per a la realització de la signatura
<i>InputDocuments</i>	Col·lecció de documents a signar.
<i>Commitment</i>	Compromís adquirit pel signant dintre d'aquest context de signatura.
<i>Role</i>	Rol corresponent al signant.
<i>DocumentsRead</i>	Indica si l'usuari ha llegit els documents a signar. Requerit per firma cega.
<i>ShowVisibleSignature</i>	Indica si s'ha d'inserir Signatura visible o no (només es tindrà en compte per aquells formats que la suportin com PDF)
<i>VisibleSignatureData</i>	Configuració de la signatura visible a crear
<i>PsisValidation</i>	Indica si es vol validar la signatura contra PSIS.
<i>SynchronousValidation</i>	Indica si la validació contra PSIS es farà de forma síncrona.
<i>PSAWSExtensions</i>	<p>Llista d'extensions de la petició. En aquest cas s'utilitzarà per les referències dinàmiques de la signatura, en cas que apliqui.</p> <p>La clau (<i>Key</i>) a utilitzar és <i>DynamicSignatureReferences</i>, i el valor (<i>Value</i>) és <i>SignatureReferencesType</i>.</p>

2. MultiSignatureAtSignatureManagerRSPNS:

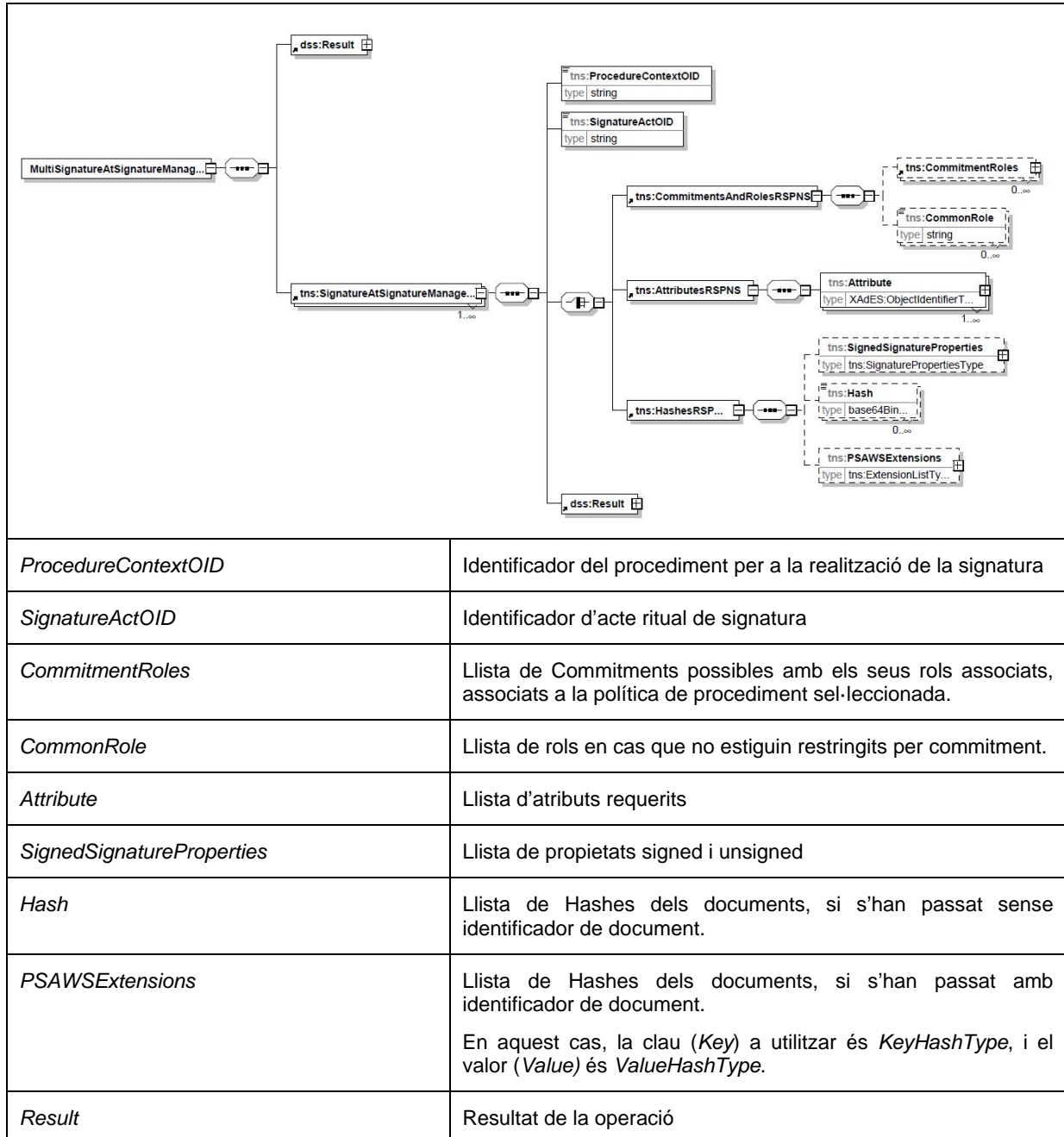
Resposta

a

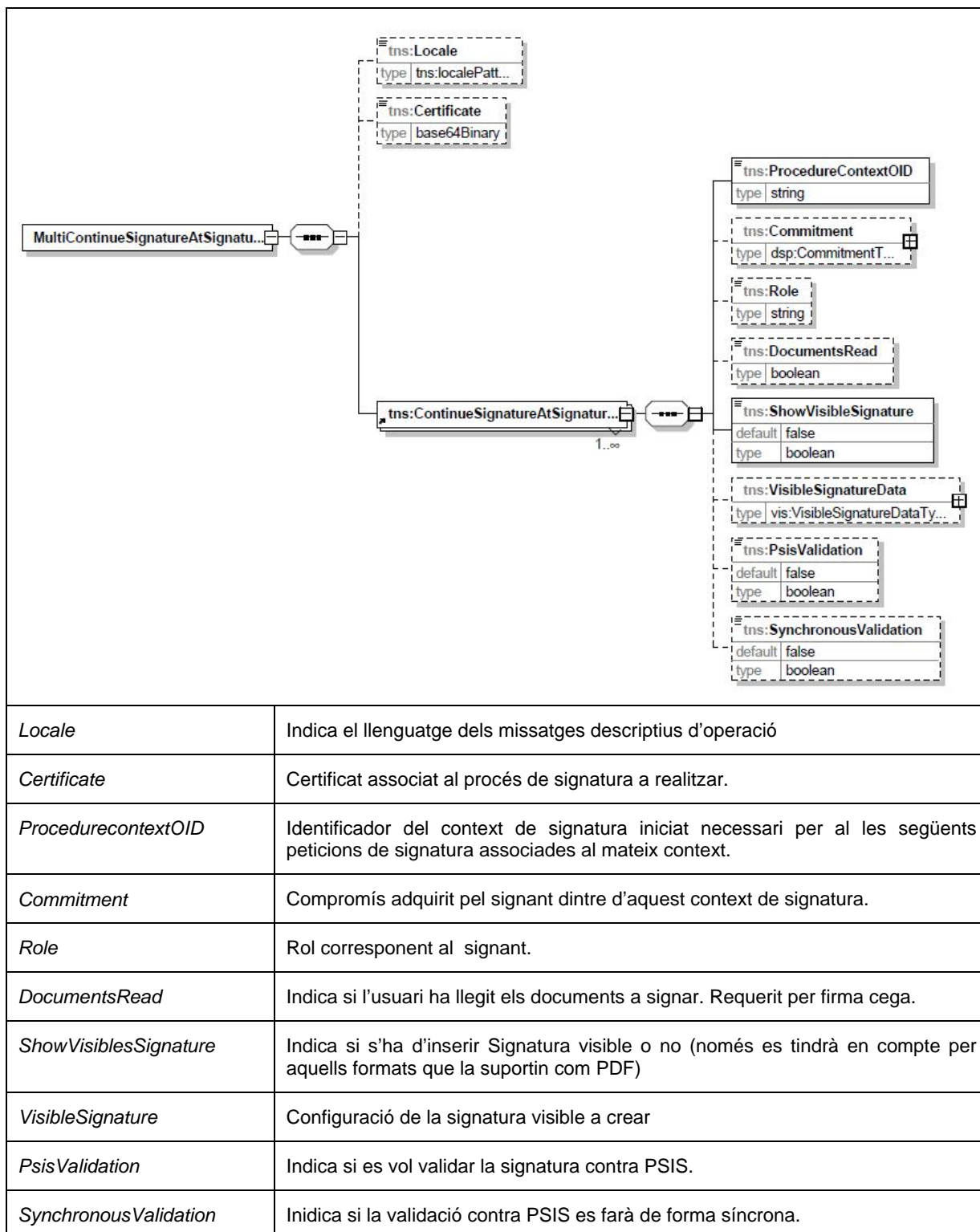
MultiInitAtSignatureManagerRQST

i

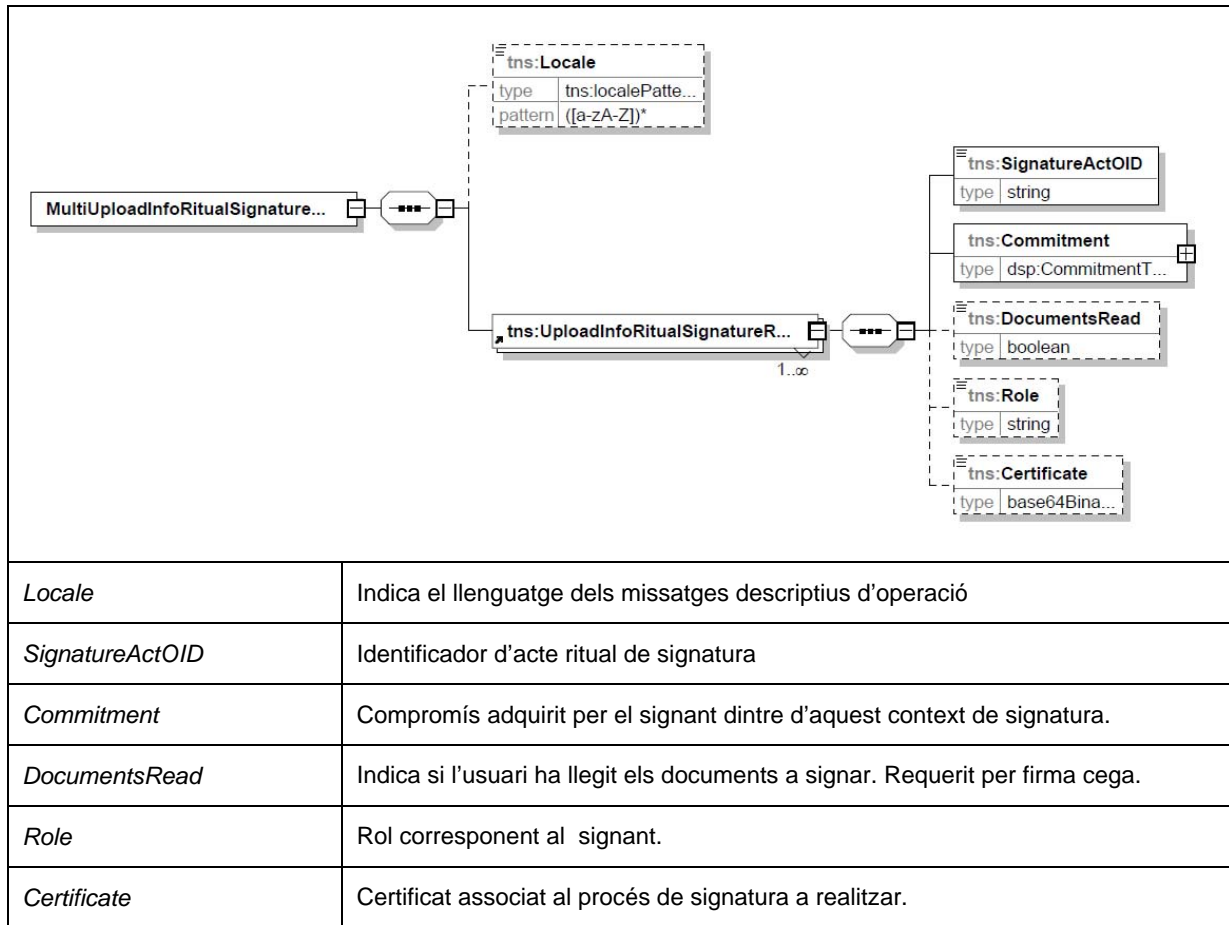
MultiContinueAtSignatureManagerRQST



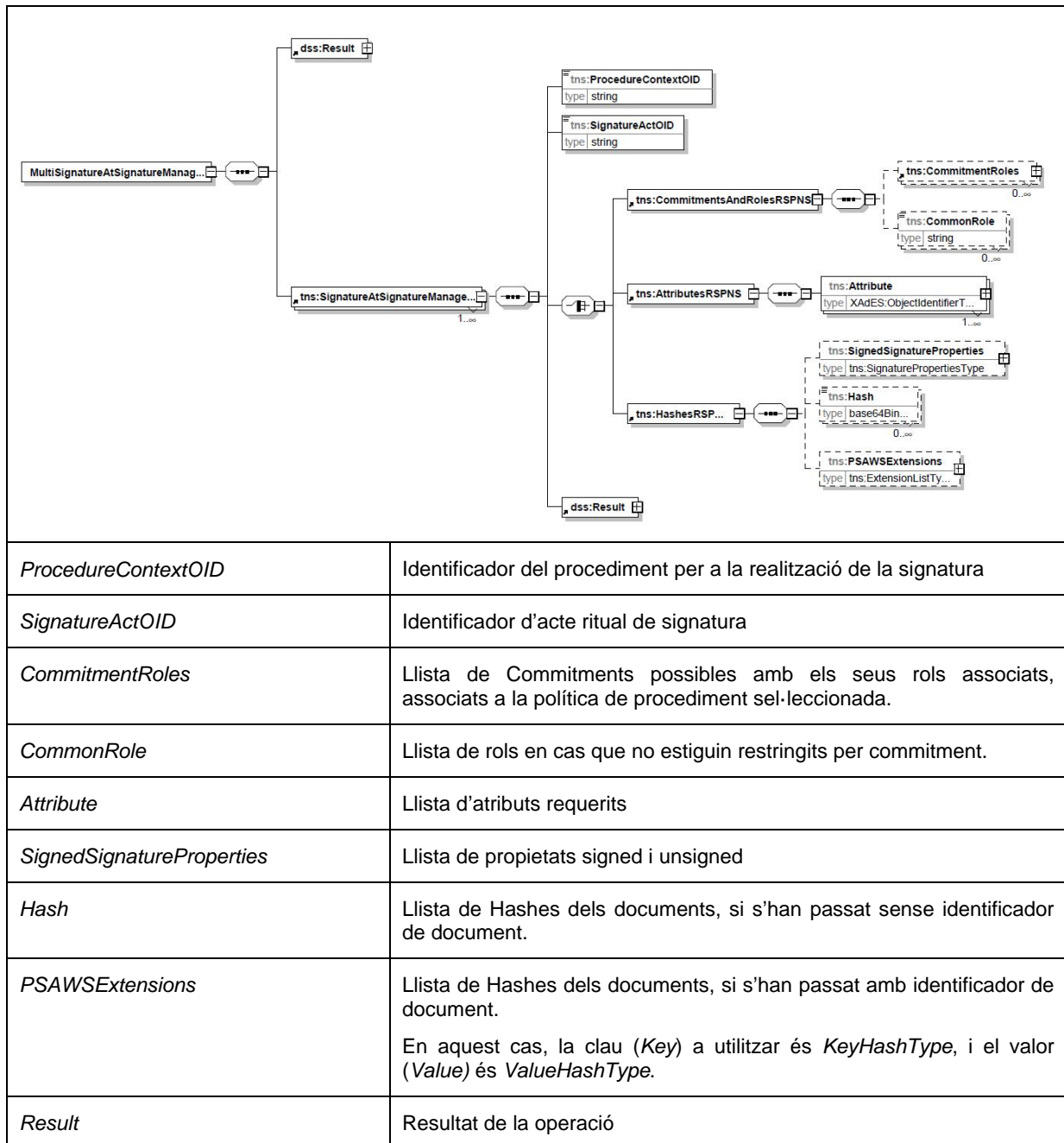
3. MultiContinueSignatureAtSignatureManagerRQST: Petició de Continuació de N contextos de signatura:



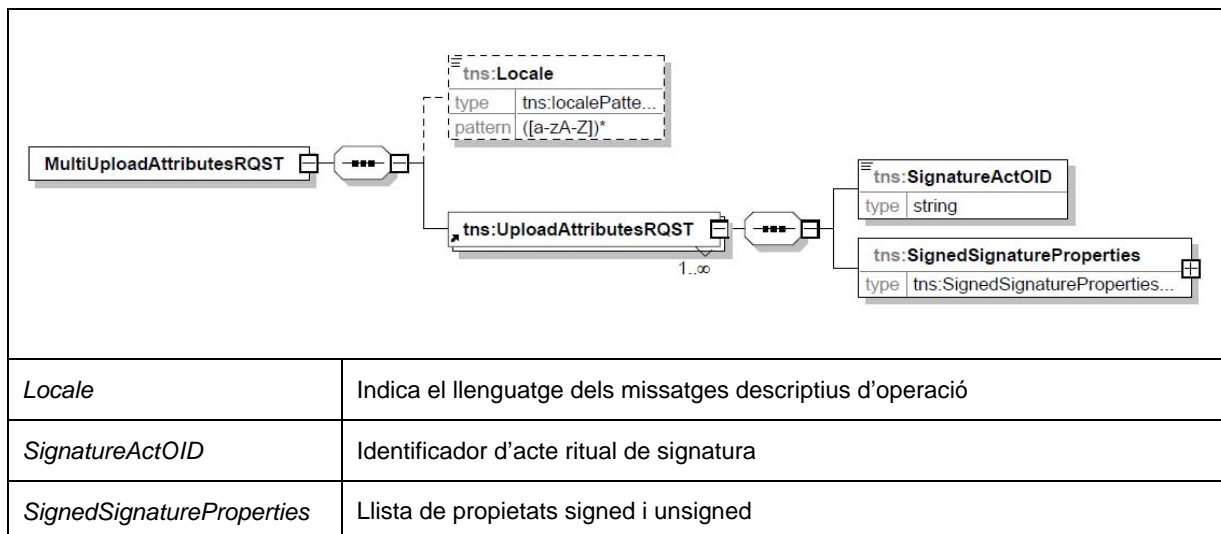
4. **MultiUploadInfoRitualSignatureRQST**: Petició amb informació necessària pel procés de signatura de N procediments:



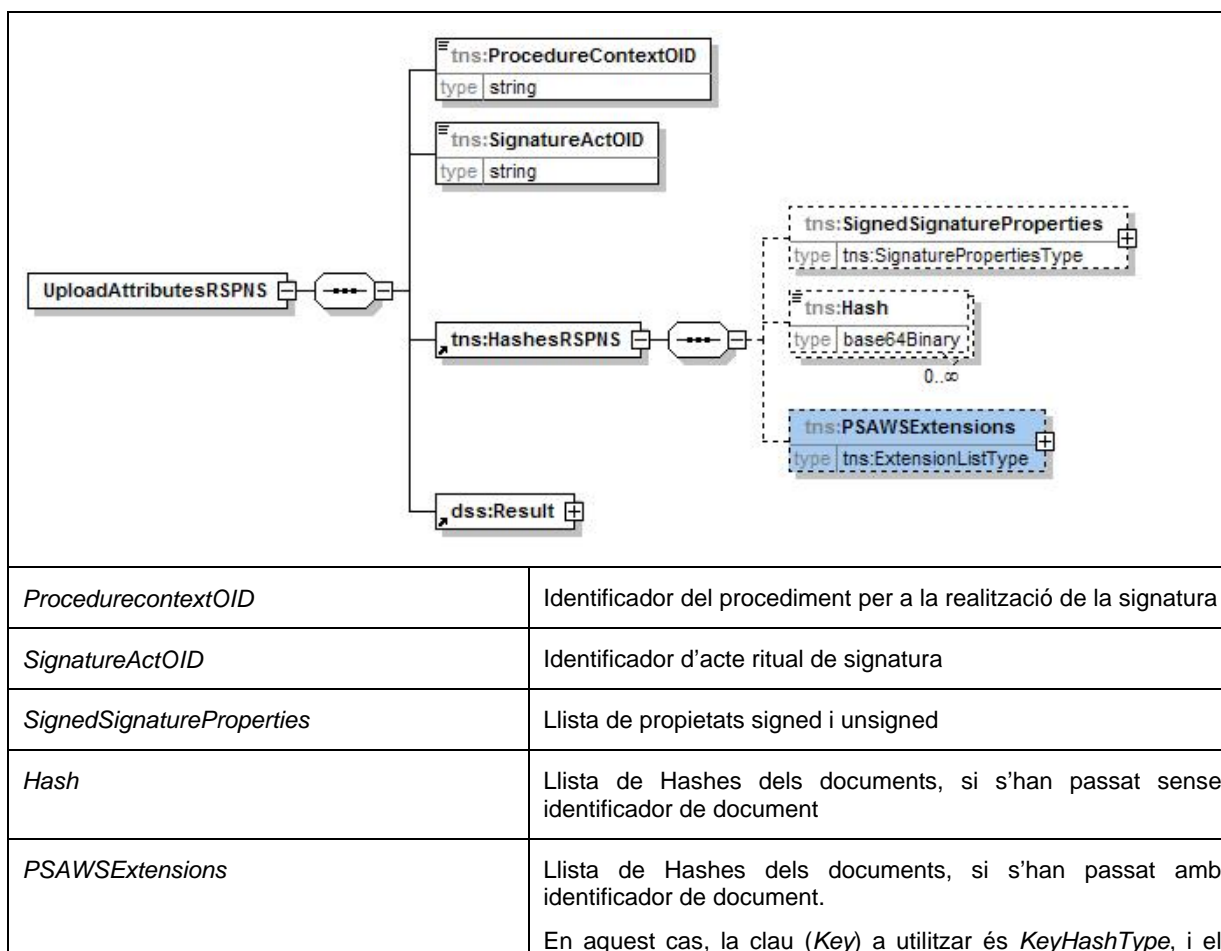
5. **MultiSignatureAtSignatureManagerRSPNS**: Resposta a MultiUploadInfoRitualSignatureRQST



6. **MultiUploadAttributesRQST**: Petició amb informació dels atributs que requereix (segons indica la política de procediment) el **SI PSA** per N procediments.

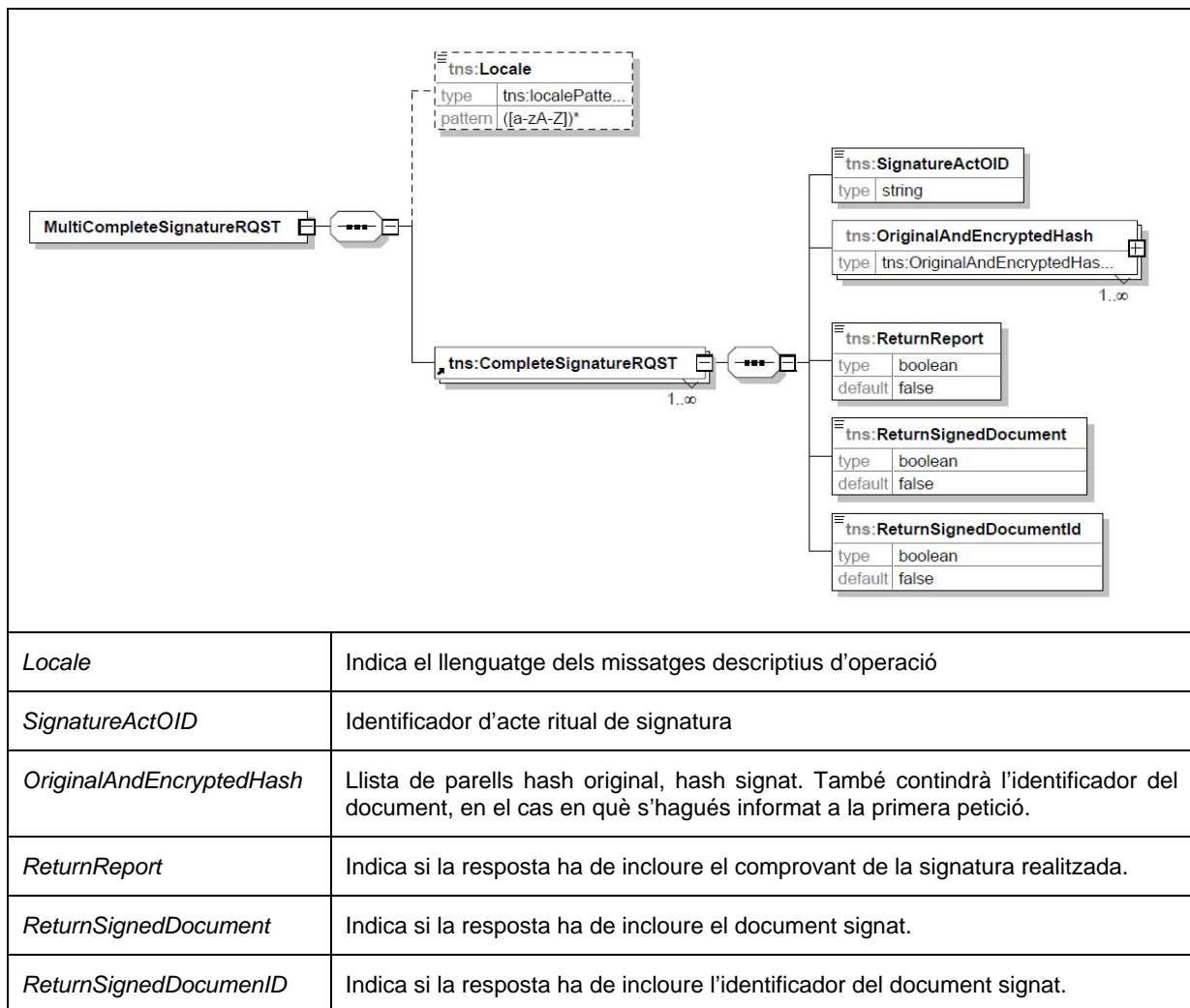


7. MultiUploadAttributesRSPNS: Resposta a MultiUploadAttributesRQST.



	valor (<i>Value</i>) és <i>ValueHashType</i> .
<i>Result</i>	Resultat de la operació

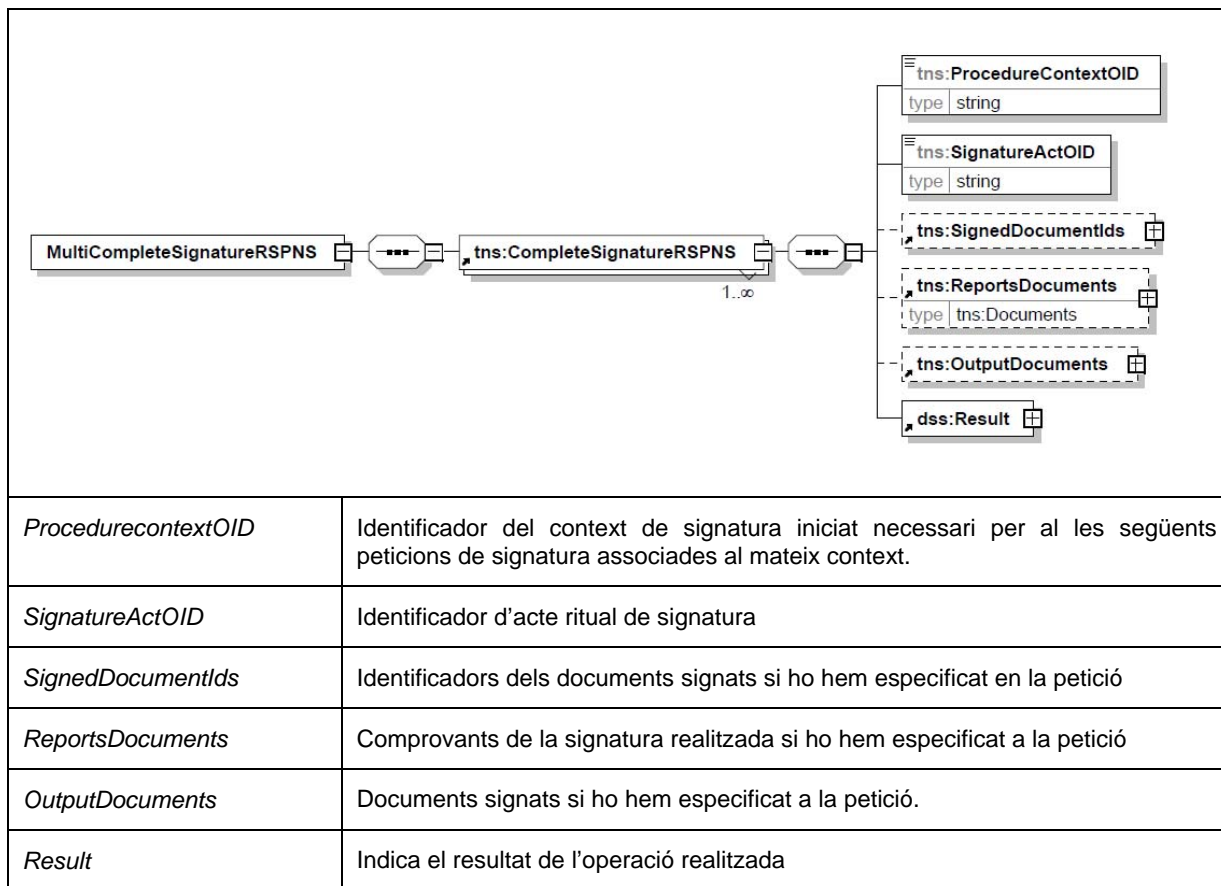
8. **MultiCompleteSignatureRQST**: Petició amb M hashes signats i flags amb sol·licitud del resultat de signatura per N procediments.



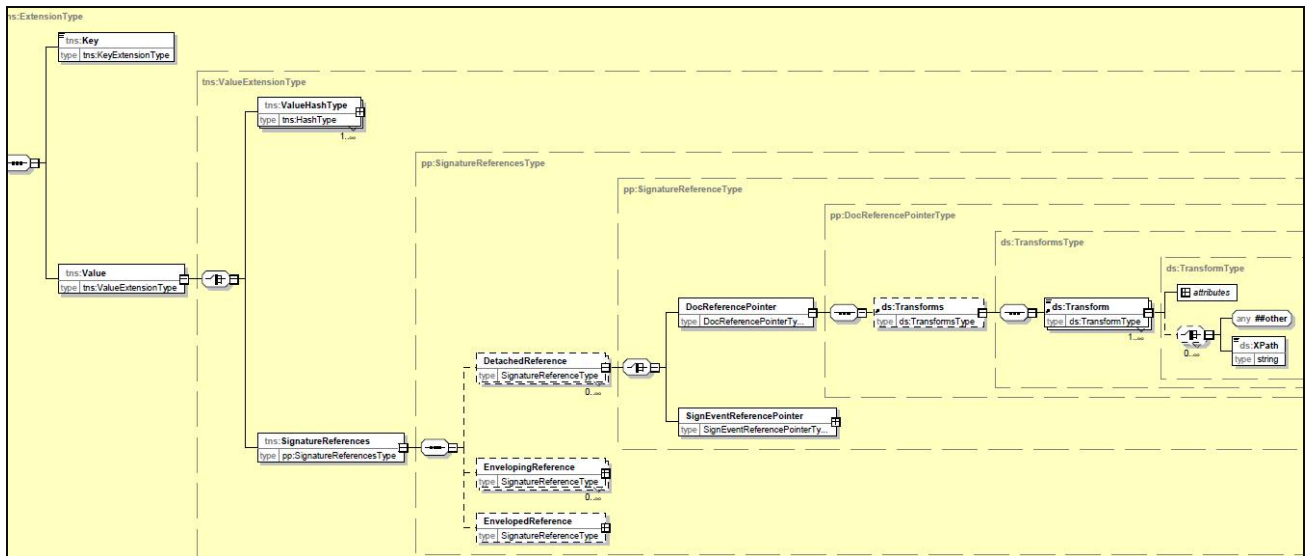
9. **MultiCompleteSignatureRSPNS**:
MultiCompleteSignatureRQST

Resposta

a



3.4 Extensió per referències dinàmiques



A la imatge anterior es defineix la estructura de la informació que es podrà enviar a aquestes peticions a PSA mitjançant les noves extensions:

- Com ja hem dit abans, la **Key** a utilitzar serà **DynamicSignatureReferences**.
- Pel que fa al valor d'aquest node, s'haurà de fer servir l'objecte de tipus **SignatureReferencesType**.

La forma d'indicar la informació de les referències és la mateixa que la que s'utilitza per indicar-ho a la política de signatura.

- Indicarem quina és la relació que volem de les dades a signar amb la signatura:
 - Detached
 - Enveloping
 - Enveloped
- També s'indicarà si es vol aplicar alguna transformació al/s document/s a signar:
 - Es podrà fer ús de la transformada XPath en cas de documents XML, per poder signar només alguna part del document, i no el document sencer.

3.5 Missatges d'Error

WS	
Missatge Error	No existeix la política de procediment {0}
Causa	La política de procediment requerida per la signatura no existeix en el sistema
Significat	La política de procediment ha d'estar donada d'alta a PSA per poder-se fer servir.
Missatge Error	Impossible crear el fitxer temporal al directori {0}
Causa	Problema associat a la creació o lectura de fitxers temporals.
Significat	Situació anòmla
Missatge Error	El format del document és invàlid. El seu pronom és: {0}
Causa	El format document que s'ha intentat signar no està suportat per PSA.
Significat	PSA pot signar un número limitat de tipus de documents. Si no és un d'ells es produeix aquest error.
Missatge Error	No existeix la política de signatura {0}
Causa	La política de signatura requerida per la signatura no existeix en el sistema
Significat	La política de signatura ha d'estar donada d'alta a PSA per poder-se fer servir.
Missatge Error	PSIS {0}. El certificat no és vàlid, té una política de certificació no suportada pel servidor. Serial {1} Issuer {2}
Causa	El certificat no és vàlid. La política de certificació no és vàlida (validació de PSIS).
Significat	El certificat té una política de certificació no suportada per PSIS.
Missatge Error	No té drets pel KeySelector Serial {0} Issuer {1}
Causa	El keyselector presentat no està associat a l'aplicació de gestió autenticada.
Significat	L'aplicació de gestió no té definit com a certificat d'operació el certificat identificat pel keyselector presentat.
Missatge Error	La política de signatura no permet signatura cega.
Causa	S'està intentat fer una signatura cega quan la política indica que no és permès.
Significat	Si a la política de procediment la signatura cega està permesa, és obligatori per part del signant d'informar que el Document a signar ha estat llegit cosa que no es pot garantir quan es fa signatura a PSA.
Missatge Error	Impossible crear el fitxer temporal al directori {0}
Causa	Problema associat a la creació o lectura de fitxers temporals.

Significat	Situació anòmla
Missatge Error	PSIS {0}. El certificat no és vàlid, no és de confiança. El certificat enviat: Serial {1} Issuer {2}
Causa	El certificat no és vàlid. El certificat no és de confiança (validació a PSIS).
Significat	El servei de certificació de PSIS no ha pogut trobar una cadena de confiança vàlida per aquest certificat.
Missatge Error	El Rol escollit, {0}, no coincideix amb el Rol del certificat, {1}.
Causa	S'ha escollit un rol que no coincideix amb el rol del certificat presentat.
Significat	Si la política de signatura obliga a una signatura amb rol certificat, el rol seleccionat ha de ser el mateix que té el certificat.
Missatge Error	El compromís escollit, {0}, no és vàlid.
Causa	Es fa una petició WS amb un compromís que no és suportat per la política de signatura.
Significat	A la política de signatura hi ha els compromisos en base als quals es pot signar i el presentat no és cap d'ells.
Missatge Error	El certificat no té el Rol adequat: {0}
Causa	Es fa una petició WS amb un rol que no és suportat per la política de signatura.
Significat	A la política de signatura hi ha els rols en base als quals es pot signar i el rol en el certificat presentat no és cap d'ells.
Missatge Error	És obligatori Rol per a aquesta signatura, i el certificat escollit ({0})no té Rol.
Causa	S'ha presentat un certificat sense rol.
Significat	A les signatures que per política de signatura requereixen de signatura amb rol certificat.requereixen.

4. Serveis d'utilitats

Els serveis d'utilitats són aquells que no realitzen cap operació de signatura digital ni de simple consulta de informació sinó que realitzen una operació en el sistema. Aquests serveis són els següents:

- **CipherDocument:** Xifra un document amb un o varis certificats especificats a la petició i retorna la llista amb el resultat de xifrar el document per cada certificat.
- **DecipherDocument:** Desxifra un document amb el KeySelector (clau privada emmagatzemada al **SI PSA**) especificat a la petició. Retorna el document desxifrat.
- **DeleteSignature:** Elimina una signatura del **SI PSA** realitzada amb anterioritat.

4.1 CipherDocument

4.1.1 Funcionalitat

Aquest servei permet a una aplicació client del **SI PSA**, realitzar una petició de xifrat de document per a un o varis certificats els quals s'especifiquen en la petició. També es permet escollir el format de xifrat del document resultant, ja sigui **S/MIME**[12] o **XMLEncrypt**[13].

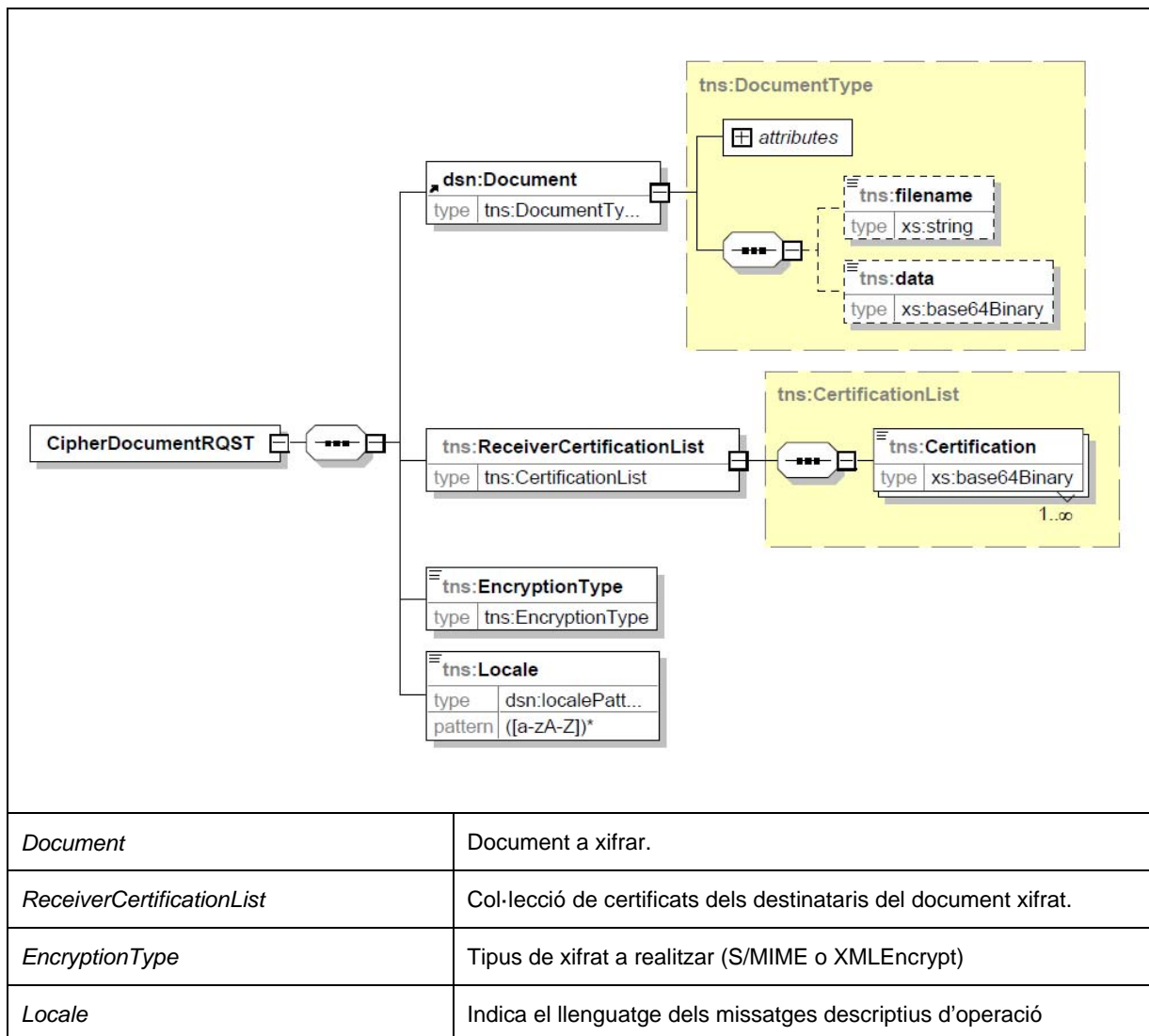
El servei i la seva informació es troba definit al profile **urn:catcert:psa:1.0:profiles:dss-encrypt**.

4.1.2 Descripció Missatges

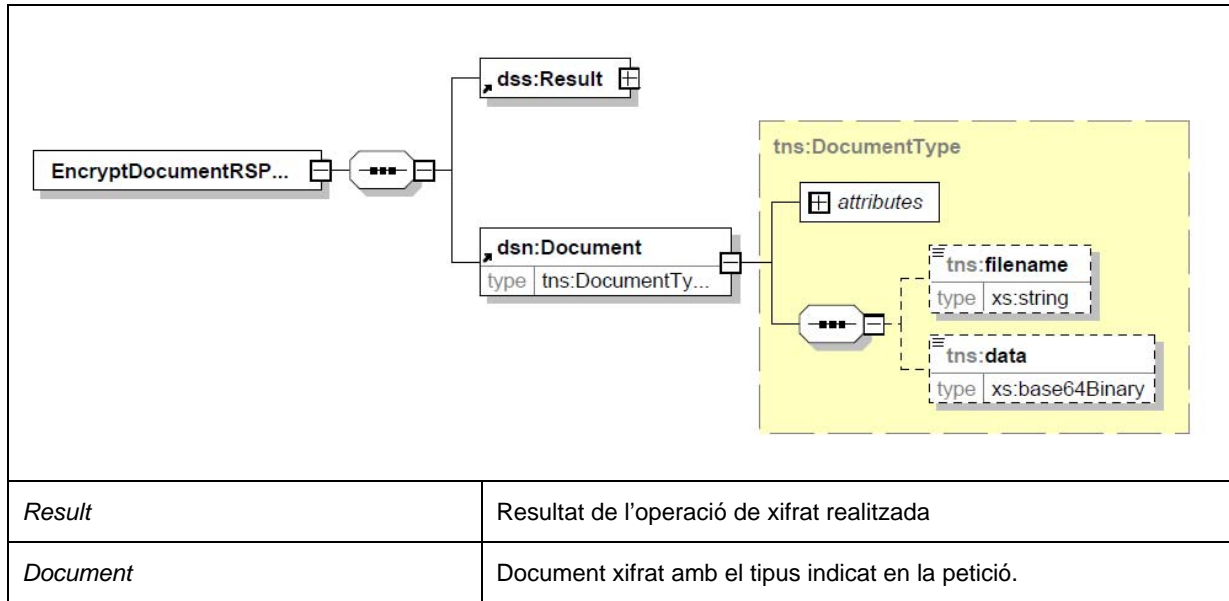
Protocol de xifrat i desxifrat de documents pel **SI PSA**.

xmlns:tns="urn:catcert:psa:1.0:profiles:dss-encrypt"

1. Petició de xifrat de document al **SI PSA CipherDocumentRQST**



2. Resposta de xifrat de document al **SI PSA** `EncryptDocumentRSPNS`



4.2 DecipherDocument

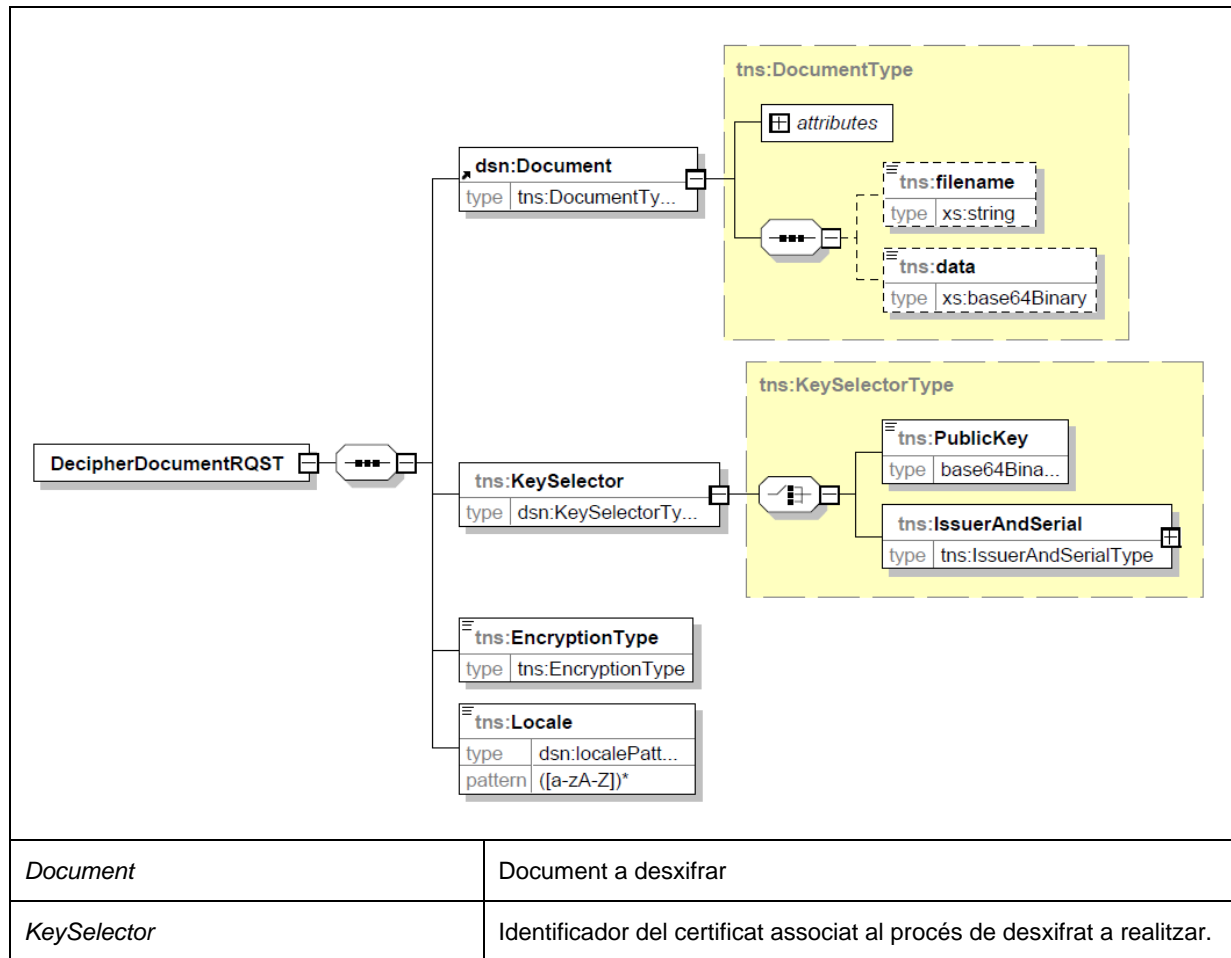
4.2.1 Funcionalitat

Aquest servei permet a una aplicació client del **SI PSA**, realitzar una petició de desxifrat de document especificant la clau privada (KeySelector) la qual s'utilitzarà en el procés de desxifrat. Aquesta clau es troba accessible pel **SI PSA**.

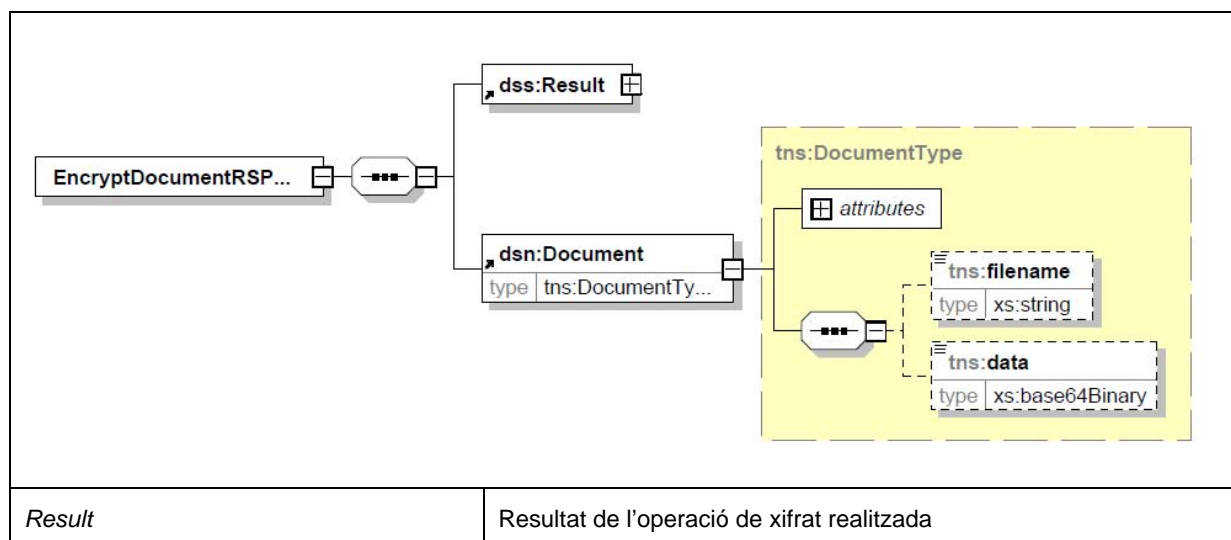
El servei i la seva informació es troba definit al profile

4.2.2 Descripció Missatges

1. Petició de desxifrat de document al **SI PSA DecipherDocumentRQST**.



1. Resposta de desxifrat de document al **SI PSA** `EncryptDocumentRSPNS`



Document	Document xifrat amb el tipus indicat en la petició.
----------	---

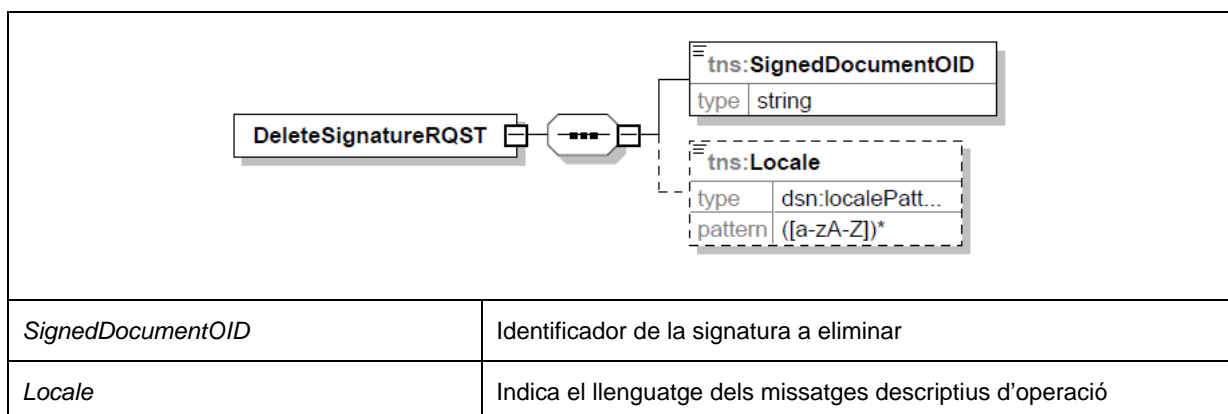
4.3 DeleteSignature

4.3.1 Funcionalitat

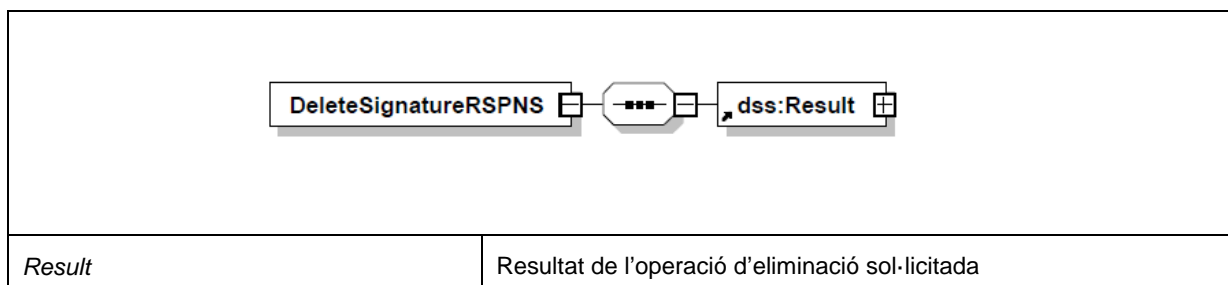
Aquest servei permet a una aplicació client del **SI PSA**, realitzar una petició d'eliminació de signatura realitzada amb anterioritat. El servei i la seva informació es troba definit al profile **urn:catcert:psa:1.0:profiles:dss-operation**.

4.3.2 Descripció Missatges

1. Petició d'eliminació de signatura **DeleteSignatureRQST**



2. Resposta d'eliminació de signatura **DeleteSignatureRSPNS**



4.4 Missatges d'Error

WS	DecipherDocument
Missatge Error	Impossible crear el fitxer temporal al directori {0}
Causa	Problema en la creació d'un fitxer temporal
Significat	Situació anòmla.
Missatge Error	CKI0011: El fitxer conté multiples claus per aquest receptor pero són diferents.
Causa	El document té xifrada la clau múltiples cops però són diferents.
Significat	És una situació improbable però podria donar-se.
Missatge Error	CKI0010: El fitxer encriptat no conté cap clau per aquest receptor.
Causa	El document xifrat no té clau per ser desxifrat.
Significat	El document no es pot desxifrar si no conté la clau. El document es invàlid.
Missatge Error	CKI0002: Ha ocorregut una excepció externa. Comproveu l'stacktrace.
Causa	Excepcions per situació anòmla diverses, consultar stacktrace.
Significat	Situació anòmla

WS	CiperDocument
Missatge Error	Fitxer temporal {0} no trobat
Causa	Problema en la lectura d'un fitxer temporal
Significat	Situació anòmla
Missatge Error	SVC0009: Proveïdor {0} no disponible.
Causa	S'ha passat un EncryptionType desconegut.
Significat	Actualment els valors possibles són CMS i XML.
Missatge Error	CKI0004: El document no pot ser null!
Causa	El document es nul.
Significat	El document a xifrar no pot se nul.
Missatge Error	CKI0002: Ha ocorregut una excepció externa. Comproveu l'stacktrace.

Causa	Problema genèric extern.
Significat	Situació anòmla

WS	DeleteSignature
Missatge Error	La signatura amb UUID {0}, no pot ser eliminada del model PSA, per que el seu Context de Signatura no està tancat
Causa	S'ha intentat esborrar una signatura abans que aquesta s'hagi completat
Significat	No es poden esborrar signatures si el context de signatura associat encara està obert.
Missatge Error	No existeix la signatura amb OID {0}
Causa	S'ha intentat esborrar una signatura que no existeix
Significat	S'ha proporcionat un OID d'una signatura que no existeix en el CMS.
Missatge Error	Error, node {0} no trobat. Error {1}
Causa	Intentar esborrar un node que no existeix al CMS.
Significat	Situació Anòmla. La signatura existeix a base de dades però no al CMS.
Missatge Error	Error creant o llegint el fitxer temporal {0}. Error {1}
Causa	Problema associat a la creació o lectura de fitxers temporals.
Significat	Situació Anòmla.
Missatge Error	No té drets per esborrar la signatura {0}
Causa	L'aplicació de gestió intenta esborrar una signatura que no li pertany
Significat	Les aplicacions de gestió només tenen dret a esborrar les seves signatures

5. Serveis de consulta

Els serveis de consulta són aquells serveis del **SI PSA** que, com el propi nom indica, permeten els processos de consulta i descàrrega d'informació del propi **SI PSA**.

Els serveis que proporciona el **SI PSA** són els següents:

- **DownloadSignatureAtPSA**: Descàrrega de Signatura.
- **QueryActivity**: Consulta d'Activitat.
- **QueryActivityToSigner**: Creació de sessió de consulta d'Activitat.
- **QueryAuthorizationToSigner**: Creació de sessió de consulta d'Autoritzacions
- **DownloadTicketSignature**: Descàrrega de comprovant.
- **QuerySignatureAct**: Consulta de informació de signatura.

5.1 DownloadSignatureAtPSA

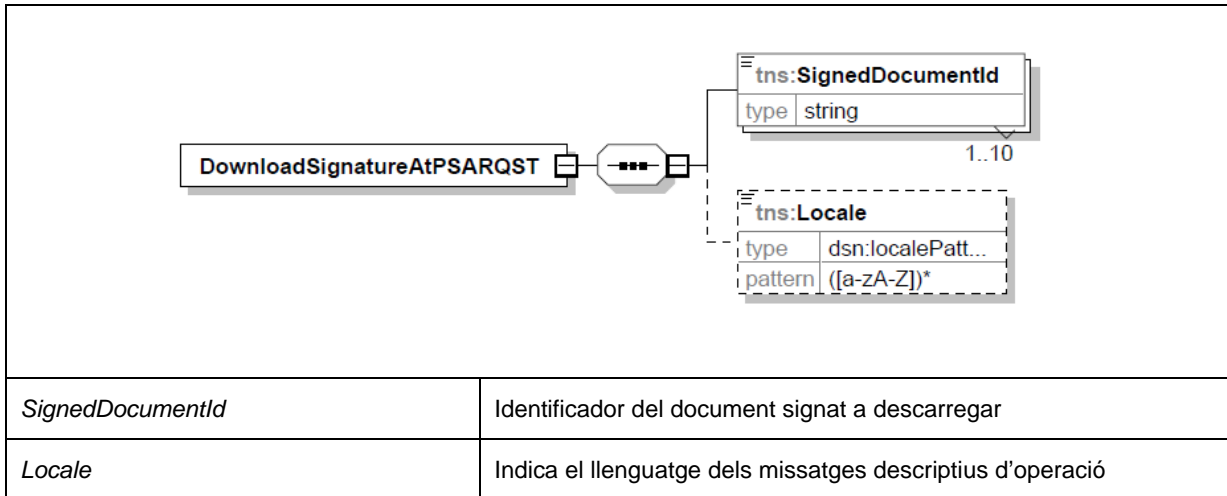
5.1.1 Funcionalitat

Aquest servei permet a una aplicació client del **SI PSA** realitzar una petició de descàrrega de N documents signats o N signatures i/o els seus identificadors emmagatzemats al **SI PSA** corresponents a un procés de signatura realitzat amb anterioritat. Pot donar-se el cas que la signatura inclogui el document i es descarregui tota la informació o únicament es descarregui la signatura si aquesta és independent del document. La invocació al servei es pot realitzar més d'una vegada, ja que la signatura es trobarà disponible fins a la seva eliminació per part de l'usuari o l'eliminació automàtica després del temps màxim establert per l'auditoria.

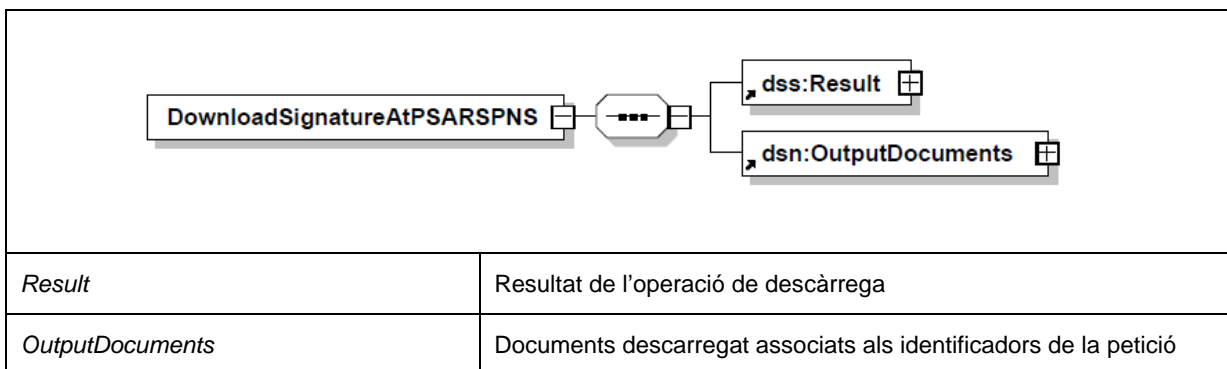
El servei i la seva informació es troba definit al profile **urn:catcert:psa:1.0:profiles:dss-getdata**.

5.1.2 Descripció Missatges

1. Petició de descàrrega de signatura **DownloadSignatureAtPSARQST**.



2. Resposta de descàrrega de signatura **DownloadSignatureAtPSARSPNS**.



5.2 QueryActivity

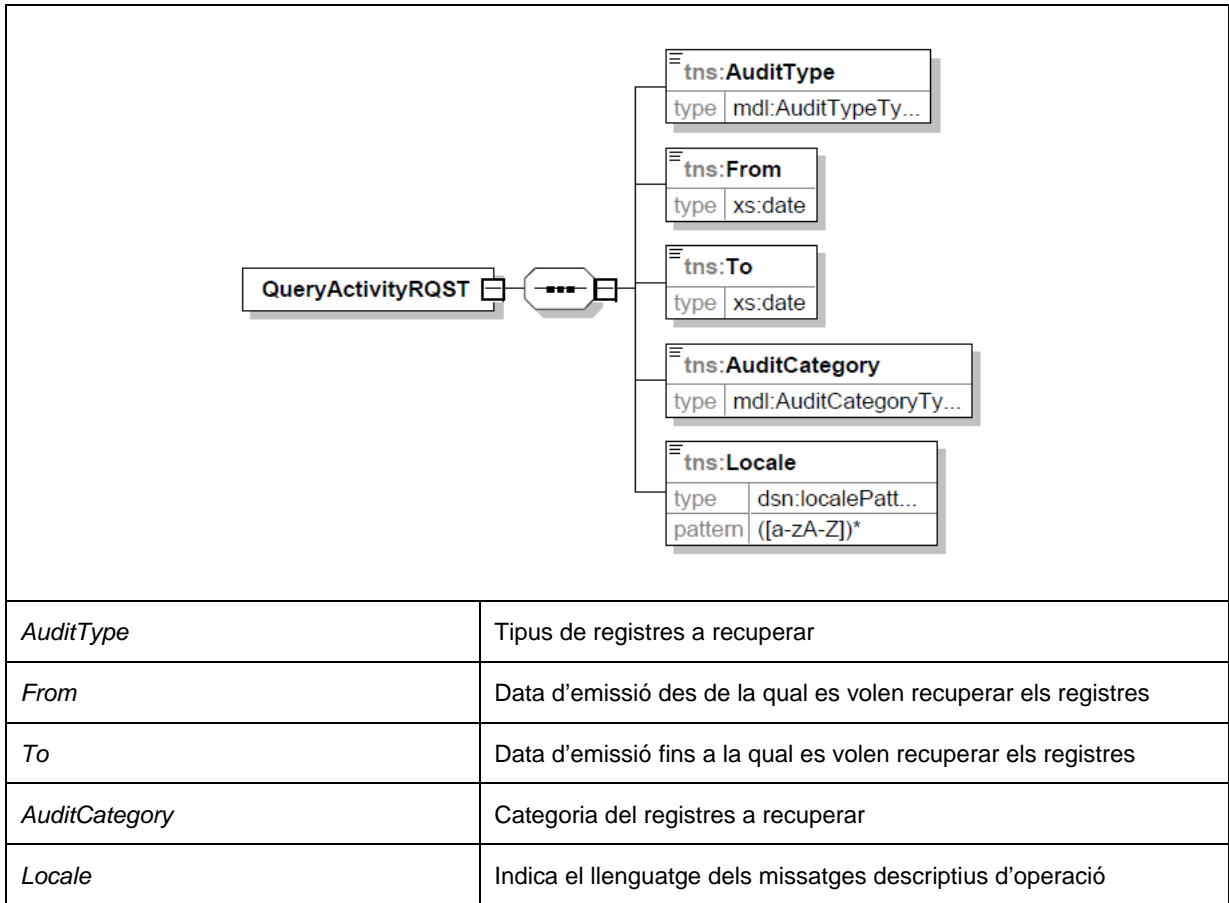
5.2.1 Funcionalitat

Aquest servei permet a una aplicació client del **SI PSA** realitzar una petició de consulta d'activitat d'operacions realitzades al **SI PSA**. La petició permet filtrar els resultats per una sèrie de paràmetres.

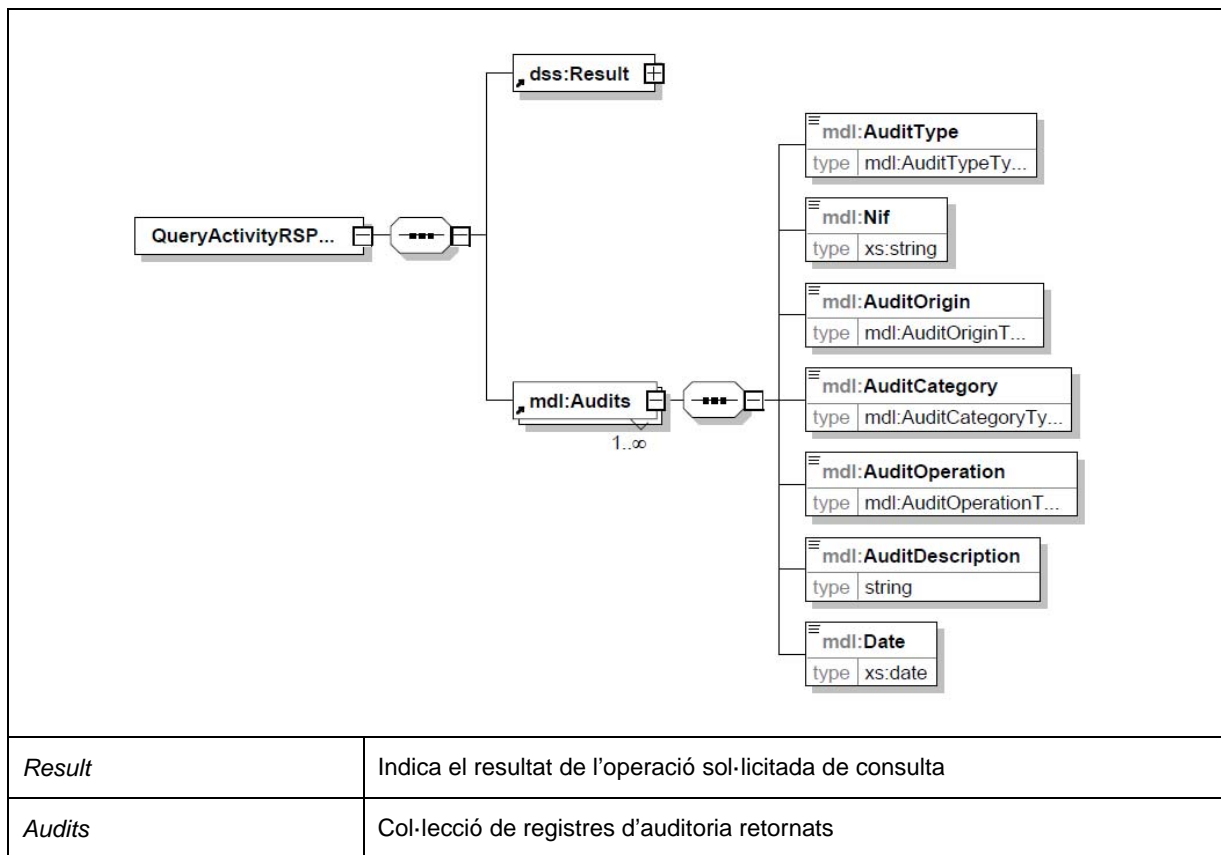
El servei i la seva informació es troba definit al profile **urn:catcert:psa:1.0:profiles:dss-getsession**.

5.2.2 Descripció Missatges

1. Petició de consulta de registres d'activitat **QueryActivityRQST**



2. Resposta de consulta de registres d'activitat **QueryActivityRSPNS**



5.3 QueryActivityToSigner

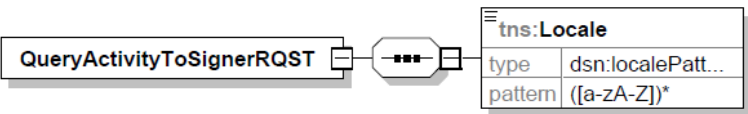
5.3.1 Funcionalitat

Aquest servei permet a una aplicació client del **SI PSA** realitzar una petició de creació de sessió per a la consulta d'activitat d'operacions realitzades al **SI PSA**. Aquesta sessió serà utilitzada posteriorment per un usuari per a la consulta interactiva de les operacions realitzades amb el seu certificat.

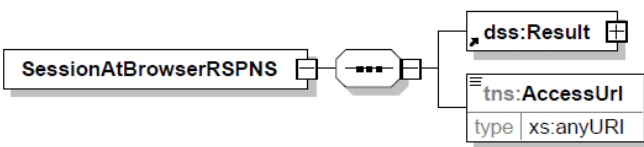
El servei i la seva informació es troba definit al profile **urn:catcert:psa:1.0:profiles:dss-getsession**.

5.3.2 Descripció Missatges

1. Petició de consulta de creació de sessió per a consulta d'activitat **QueryActivityToSignerRQST**

	
<i>Locale</i>	Indica el llenguatge dels missatges descriptius d'operació

2. Resposta de creació de sessió de consulta d'activitat **SessionAtBrowserRSPNS**

	
<i>Result</i>	Resultat de l'operació de creació de la sessió sol·licitada
<i>AccesUrl</i>	URL d'accés al frontal de signatura del PSA per a la realització d'aquesta.

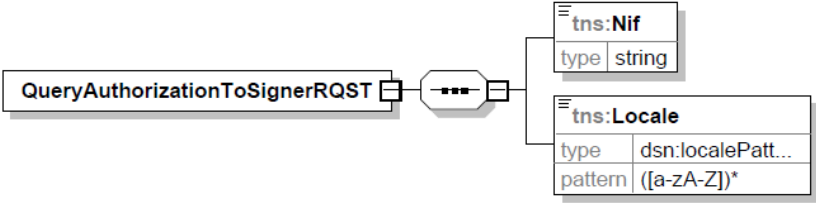
5.4 QueryAuthorizationToSignature

5.4.1 Funcionalitat

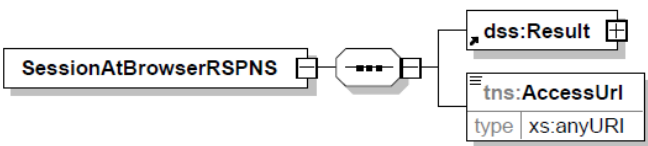
Aquest servei permet a una aplicació client del **SI PSA** realitzar una petició de creació de sessió per a la consulta d'autoritzacions de certificats. Aquesta sessió serà utilitzada posteriorment per un usuari per a la consulta i gestió interactiva d'autoritzacions dels certificats amb el NIF especificat.

5.4.2 Descripció Missatges

1. Petició de consulta de creació de sessió per a consulta d'autoritzacions
QueryAuthorizationToSignerRQST

	
<i>Nif</i>	NIF dels certificats dels quals es farà la consulta/gestió d'autoritzacions.
<i>Locale</i>	Indica el llenguatge dels missatges descriptius d'operació

2. Resposta de creació de sessió de consulta d'autoritzacions **SessionAtBrowserRSPNS**

	
<i>Result</i>	Resultat de l'operació de creació de la sessió sol·licitada
<i>AccesUrl</i>	URL d'accés al frontal de signatura del PSA per a la realització d'aquesta.

5.5 DownloadTicketSignature

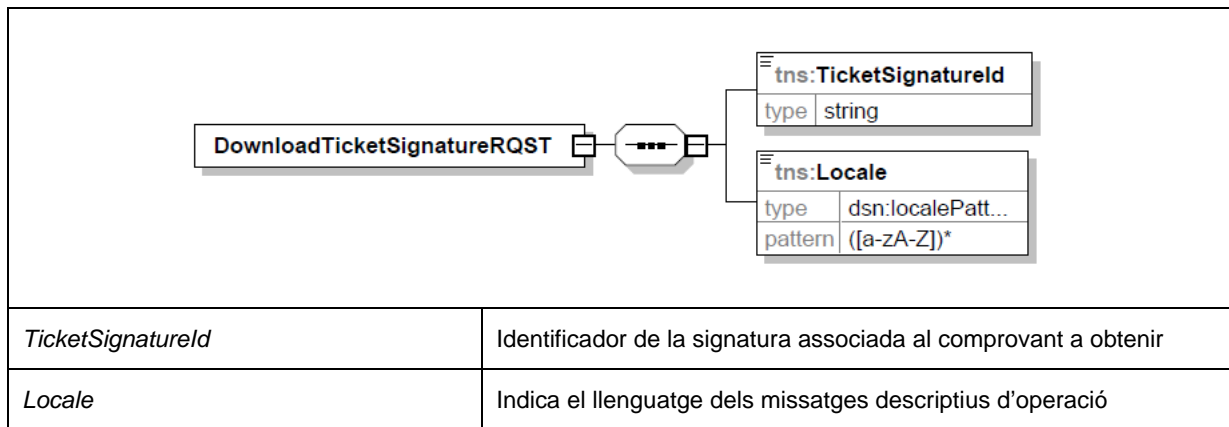
5.5.1 Funcionalitat

Aquest servei permet a una aplicació client del **SI PSA** realitzar una petició de sol·licitud de comprovant corresponent a una signatura realitzada amb anterioritat.

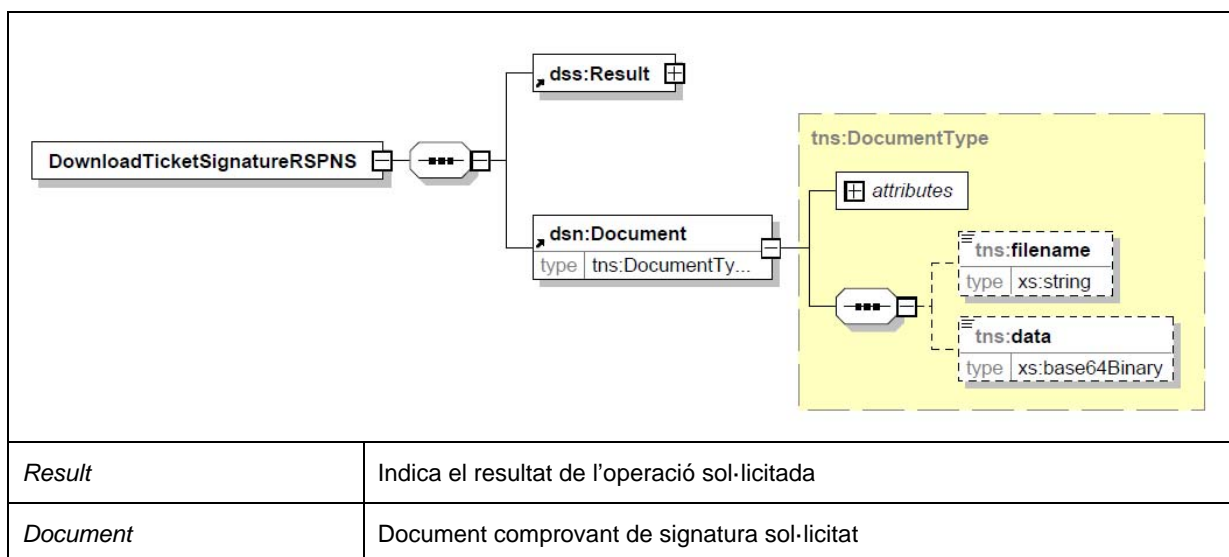
El servei i la seva informació es troba definit al profile **urn:catcert:psa:1.0:profiles:dss-getdata**.

5.5.2 Descripció Missatges

1. Petició de sol·licitud de comprovant **DownloadTicketSignatureRQST**



2. Resposta de sol·licitud de comprovant **DownloadTicketSignatureRSPNS**



5.6 QuerySignatureAct

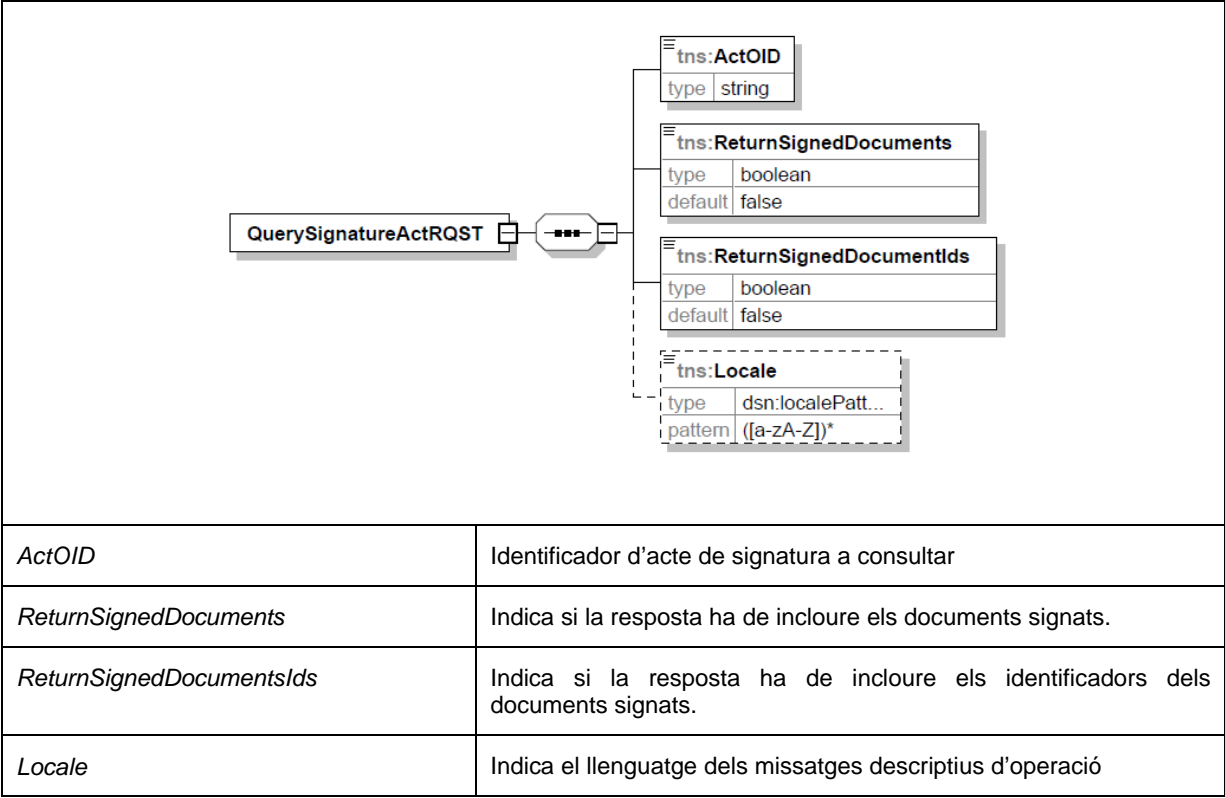
5.6.1 Funcionalitat

Aquest servei permet a una aplicació client del **SI PSA** realitzar una petició de consulta per tal de conèixer la informació associada a un acte ritual de signatura finalitzat, pendent de finalització o erroni.

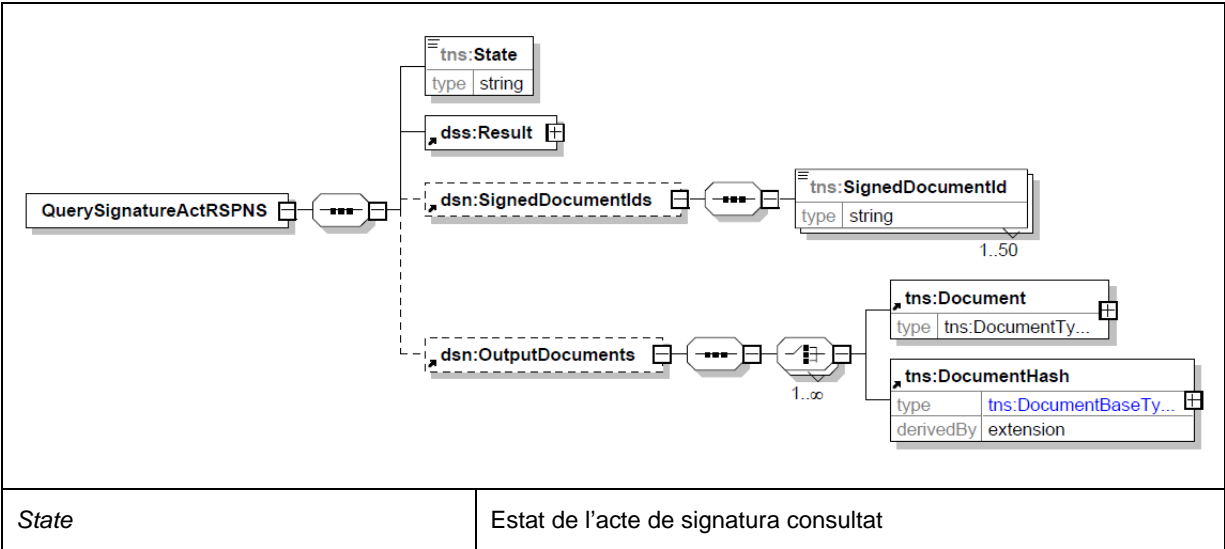
El servei i la seva informació es troba definit al profile **urn:catcert:psa:1.0:profiles:dss-getdata**.

5.6.2 Descripció Missatges

1. Petició de sol·licitud d'informació de signatura **QuerySignatureActRQST**



2. Resposta de sol·licitud d'informació de signatura **QuerySignatureActRSPNS**



<i>Result</i>	Resultat de l'operació
<i>SignedDocumentIds</i>	Identificadors de documents signats.
<i>OutputDocuments</i>	Documents signats

5.7 DownloadVerifySignatureTicket

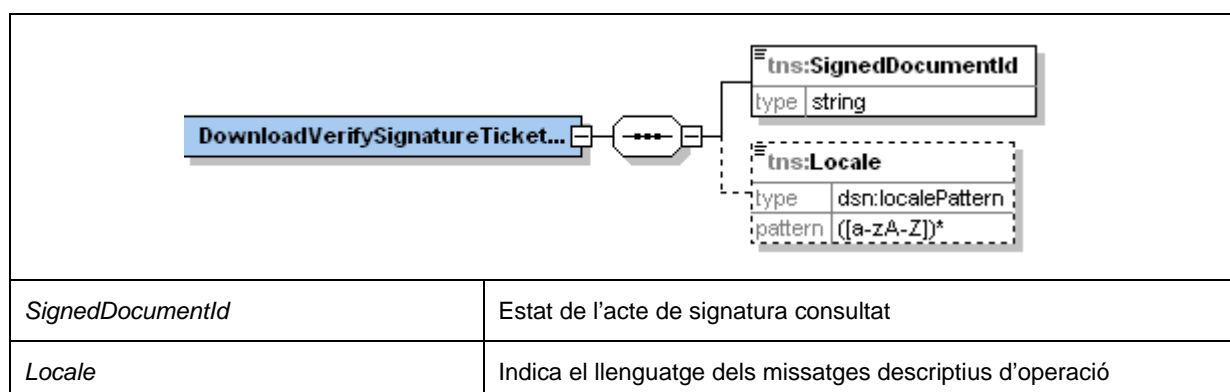
5.7.1 Funcionalitat

Aquest servei web permet a una aplicació client del **SI PSA** realitzar una petició de descàrrega del tiquet de validació (de PSIS) d'una de les seves signatures.

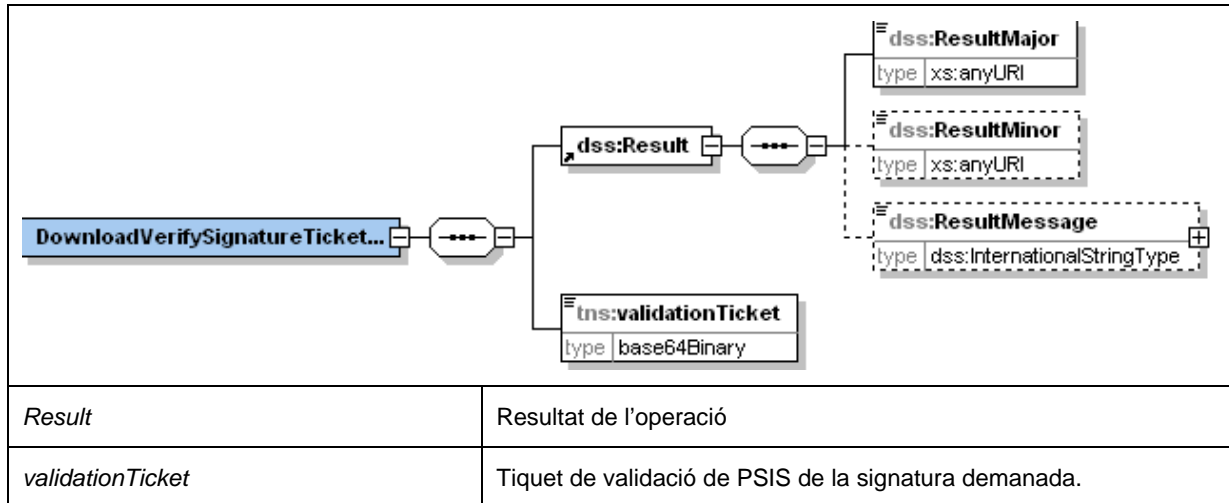
El servei i la seva informació es troba definit al profile **urn:catcert:psa:1.0:profiles:dss-getdata**.

5.7.2 Descripció Missatges

1. Petició de sol·licitud del tiquet de validació
DownloadVerifySignatureTicketRQST



2. Resposta de sol·licitud del tiquet de validació de la signatura DownloadVerifySignatureTicketRSPNS



5.8 Missatges d'error

WS	downloadSignatureAtPSA
Missatge Error	Error, node {0} no trobat. Error {1}
Causa	Intentar accedir a un node que no existeix al CMS.
Significat	La signatura demanada no existeix
Missatge Error	Error creant o llegint el fitxer temporal {0}. Error {1}
Causa	Problema associat a la creació o lectura de fitxers temporals.
Significat	Situació anòmla
Missatge Error	No té drets a accedir al document {0}
Causa	Intentar descarregar una signatura sobre la que no es tenen permisos
Significat	Les aplicacions de gestió només poden descarregar signatures realitzades per elles mateixes.

WS	downloadTicketSignature
Missatge Error	No té drets a accedir al document {0}
Causa	S'ha demanat un ticket de signatura sobre el qual no es tenen permisos
Significat	Una aplicació de gestió només pot descarregar els comprovants associats a les seves signatures.
Missatge Error	No existeix la signatura amb OID {0}
Causa	S'ha demanat un ticket que no existeix
Significat	El ticket no existeix en el CMS.
Missatge Error	Error creant o llegint el fitxer temporal {0}. Error {1}
Causa	Problema associat a la creació o lectura de fitxers temporals.
Significat	Situació Anòmala
Missatge Error	Error, node {0} no trobat. Error {1}
Causa	El ticket existeix a base de dades pero no al CMS.
Significat	Situació anòmala

WS	queryActivity
Missatge Error	
Causa	
Significat	

WS	queryActivityToSigner
Missatge Error	
Causa	
Significat	

6. Client del SI PSA

6.1 Requisits previs

En els següents punts es descriu de forma detallada els requisits necessaris pel desenvolupament d'un client per al consum dels serveis Web del **SI PSA**.

Aquests requisits són necessaris per a la realització de les següents operacions:

- Obtenció dels fitxers amb la informació descriptiva del diferents serveis (**WSDL's** [2]).
- Creació de les classes del client a partir del fitxer **WSDL**.
- Implementació de la lògica d'execució.
- Compilació de la lògica implementada.
- Execució del client i recuperació de resultat de l'operació

6.1.1 Comunicacions i protocols

Per a la realització de les tasques de desenvolupament i execució del client del **SI PSA**, es necessari disposar d'un ordinador amb connexió a Internet i accés al punt d'entrada dels serveis Web del **SI PSA**.

- Entorn Integració CATCert
 - <http://www.preproduccio.psa.cat/engineWS/signature/service>
 - <http://www.preproduccio.psa.cat/engineWS/query/service>
 - <http://www.preproduccio.psa.cat/engineWS/util/service>

Web d'administració/configuradors:

- https://www.preproduccio.psa.cat/web_psa

- Entorn d'explotació CATCert
 - <http://www.psa.cat/engineWS/signature/service>
 - <http://www.psa.cat/engineWS/query/service>
 - <http://www.psa.cat/engineWS/util/service>

Web d'administració/configuradors:

- https://www.psa.cat/web_psa

Els serveis que ofereix el **SI PSA** s'han desenvolupat seguint en compte la definició **WSDL**. **WSDL** defineix tots els mètodes, interfícies i elements necessaris per què el programador desenvolupi el codi del seu client d'una forma estàndard e independent de la tecnologia.

Per a verificar la accessibilitat a cadascú dels serveis, des d'una finestra del nostre navegador, podem introduir la **URL** dels diferents fitxers de descripció dels serveis Web, depenent del entorn al qual volem accedir.

Aquestes **URL** s'aconsegueixen afegint “?wsdl” a les adreces del servei abans indicades. (per exemple la URL de desenvolupament de les utilitats de preproducció és “<http://www.preproduccio.psa.cat/engineWS/util/service?wsdl>”)

Si l'accessibilitat és correcta, s'obrirà la plana amb la descripció del servei Web corresponent.

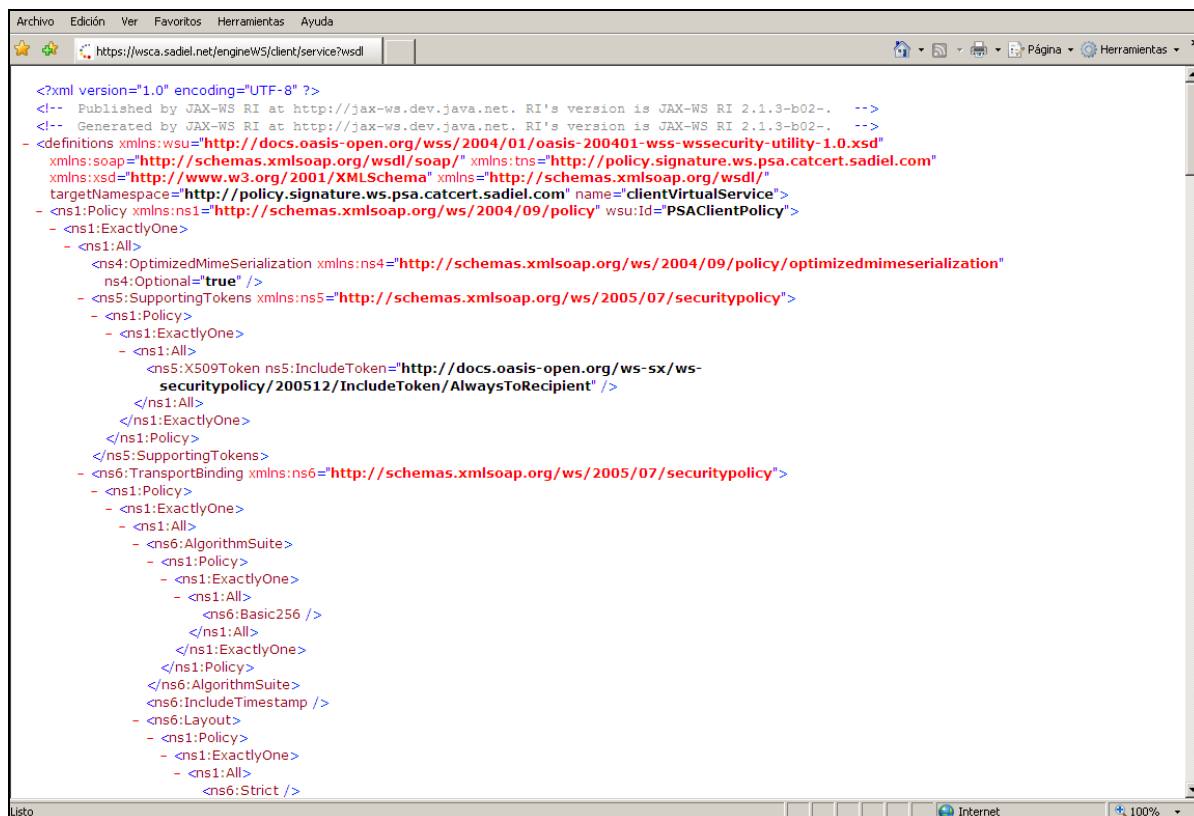


Fig. 1 – Plana amb la descripció del servei Web

6.1.2 Software

Aquest punt indica els requisits de software que seran necessaris per al desenvolupament dels clients d'accés als serveis del **SI PSA**. Aquests requeriments són diferents depenent del llenguatge a utilitzar.

6.1.2.1 Client Java

El software necessari per construir el client en Java:

- **JDK 1.5+[9]** - Recomanem JDK 6, JDK 1.5 és un producte que ha completat el SUN EOL (End Of Live). Si es vol emprar una JDK 1.4 veure el punt de [Client amb JDK 1.4](#).

- **Maven2 v2.0.9+[10]** -
- **Metro 1.2+[11]** - Metro es un stack de Web Services para Java.

6.1.2.2 Client .NET

A continuació especifiquem els requisits de software necessaris si el llenguatge de programació es VB² .NET.

- Entorn de desenvolupament **Microsoft Visual Studio 2005**[4].
- Versió de **Microsoft .NET framework** [5] 2.0.50727
- Microsoft **Web Service Enhancements** [6] v.3.0

En la part d'execució del client per a .NET s'especifiquen els passos a realitzar amb aquest software.

6.2 Execució dels serveis

En els següents punts es descriuran els passos a seguir per a la creació i execució dels clients d'accés als serveis Web del **SI PSA** per als diferents llenguatges, Java y VB .NET.

6.2.1 Client Java

Aquest document no té com objectiu ser un document introductori a **maven2**, i es pressuposa que el lector té coneixements del producte **maven2**.

L'explicació per construir el client Java, d'aquest punt està basat en l'ús de **maven2**.

Per utilitzar un WS de **SI PSA** des d'una classe Java tenim dues opcions:

1. **Treballar amb l'API de WS del SI PSA³** – Aquesta biblioteca proporciona les classes client preparades per el motor de WS **Metro** i classes d'utilitat que facilita la generació de les peticions del WS.
2. **From scratch (WSDL2Java)** – Aquesta opció requereix accedir al **WSDL**, generar les classes amb el motor de WS seleccionat i finalment generar tota la petició.

Les peticions dels WS del **SI PSA** són relativament complexes, per això recomanem la primera opció.

² Dels llenguatges suportats pel entorn de desenvolupament Visual Studio 2005 s'ha escollit el Visual Basic .NET per a la realització d'aquesta guia. De manera anàloga es poden realitzar les mateixes operacions emprant les altres llenguatges (C#, J#).

³ Veure Annex per la generació de la API de WS del PSA

6.2.1.1 Client amb l'API del WS del SI PSA

Creem un projecte Java simple amb el **standard layout de maven2**[10] i el **pom.xml** del projecte serà semblant al següent:

pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.sadiel.catcert.psa.clientws</groupId>
  <artifactId>api_psa</artifactId>
  <version>2.0-SNAPSHOT</version>
  <name>clientws_psa</name>
  <packaging>jar</packaging>
  <url>http://www.sadiel.es</url>
  <repositories>
    <repository>
      <id>maven2-repository.dev.java.net</id>
      <name>Java.net Maven 2 Repository</name>
      <url>http://download.java.net/maven/2</url>
    </repository>
  </repositories>
  <pluginRepositories>
    <pluginRepository>
      <id>maven2-repository.dev.java.net</id>
      <url>http://download.java.net/maven/2</url>
    </pluginRepository>
  </pluginRepositories>

  <profiles>
    <profile>
      <id>client</id>
      <activation>
        <activeByDefault>>false</activeByDefault>
      </activation>
      <build>
        <plugins>
          <plugin>
            <artifactId>maven-compiler-plugin</artifactId>
            <configuration>
              <source>1.5</source>
              <target>1.5</target>
            </configuration>
          </plugin>
        </plugins>
      </build>
    </profile>
  </profiles>
  <dependencies>
    <!-- METRO -->
    <dependency>
      <groupId>com.sun.xml.ws.metro</groupId>
      <artifactId>webservices-api</artifactId>
      <version>1.2</version>
    </dependency>
  </dependencies>
</project>
```



```
<dependency>
  <groupId>com.sun.xml.ws.metro</groupId>
  <artifactId>webservices-extra</artifactId>
  <version>1.2</version>
</dependency>
<dependency>
  <groupId>com.sun.xml.ws.metro</groupId>
  <artifactId>webservices-extra-api</artifactId>
  <version>1.2</version>
</dependency>
<dependency>
  <groupId>com.sun.xml.ws.metro</groupId>
  <artifactId>webservices-rt</artifactId>
  <version>1.2</version>
</dependency>
<dependency>
  <groupId>com.sun.xml.ws.metro</groupId>
  <artifactId>webservices-tools</artifactId>
  <version>1.2</version>
</dependency>
<!-- -->
<dependency>
  <groupId>com.sadiel.catcert.psa</groupId>
  <artifactId>clientws_psa</artifactId>
  <version>1.0.0.RC1</version>
</dependency>
<dependency>
  <groupId>com.sadiel.catcert.psa</groupId>
  <artifactId>commons_psa</artifactId>
  <version>1.0.0.RC1</version>
</dependency>
</dependencies>

</project>
```

Les dependències que utilitza el projecte són:

- El **Metro stack WS**, no es troben a cap repositori públic de **maven2**. Per això hem de baixar-nos **metro 1.2+** (<https://metro.dev.java.net/>) i deixar els jars al **repositori local** o a un **repositori corporatiu** amb l'estructura que hem definit al **pom.xml** (*com.sun.xml.ws.metro.web-services-**):

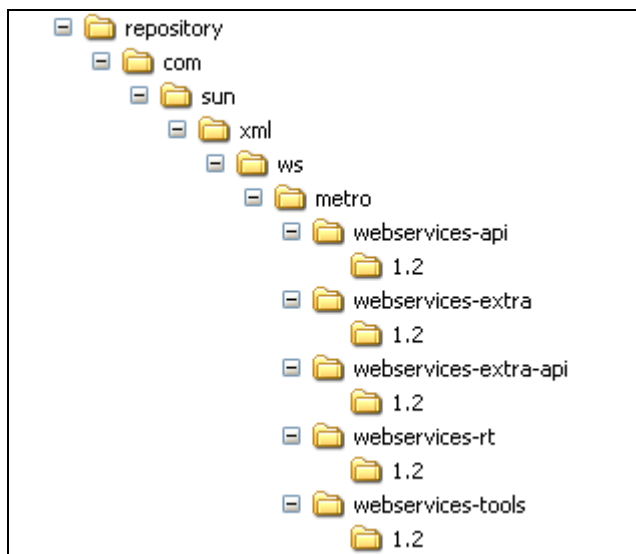


Fig. 2 Captura organització de metro al repositori local de Maven.

- **API dels WS del SI PSA**, aquestes biblioteques actualment no són accessibles a cap repositori públic de maven2 i s'han de sol·licitar a **CATCert** i hem de deixar-les al repositori local com hem explicat en el punt anterior. Les biblioteques són dos:
 - `commons_psa` – Conté classes d'utilitat.
 - `clientws_psa` – Conté les classes client dels WS del **SI PSA** i factories que permeten generar peticions. Aquestes factories faciliten la generació de les complexes peticions.

Recordem que **Maven2** treballa amb els següents tipus de repositoris:

- **Local Repository** – Repositori localitzat a la màquina del desenvolupador i es pot configurar mitjançant el fitxer `%M2_HOME%\conf\settings.xml`.
- **Public Remote External Repository** – És el repositori públic. Maven2 utilitza per defecte el repositori públic <http://repo1.maven.org/maven2>.
- **Private Remote Internal Repository** – És el repositori d'una companyia, organització o corporació que centralitza les seves llibreries.

Una bona pràctica es disposar d'un o més **Private Remote Internal Repository**, i permetre publicar les llibreries del projectes generats.

6.2.1.2 Configuració client amb Metro

Per executar els WS del **SI PSA** amb **Metro** necessitem fer les següents accions:

- Creuem el fitxer **wsit-client.xml** a la carpeta **META-INF** del jar del client a desenvolupar.
- El fitxer **wsit-client.xml** importa el fitxer **<nom servei>Service.xml** que conté el WSDL del servei WEB i la política de seguretat del client.
- El fitxer **<nom servei>Service.xml** a la carpeta **META-INF** del jar del client a desenvolupar.
- Copiem a la carpeta **META-INF** del jar del client a desenvolupar el p12 necessari per treballar amb el **X509Token**.

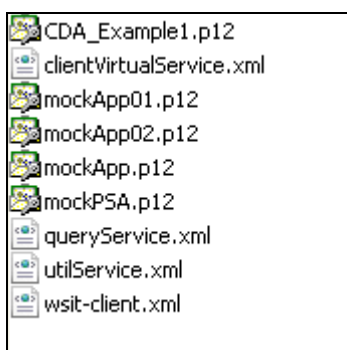


Fig. 3 Captura continguts de META-INF.

Exemple de META-INF/wsit-client.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  name="mainclientconfig">

  <import location="signatureService.xml"
    namespace="http://policy.signature.ws.psa.catcert.sadiel.com" />

</definitions>
```

Exemple de META-INF/signatureService.xml

```
<?xml version="1.0" encoding="UTF-8"?><!-- Published by JAX-WS RI at
http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.1.3-b02-. --><!--
Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is
JAX-WS RI 2.1.3-b02-. -->
<definitions
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
```

```
wssecurity-utility-1.0.xsd"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://policy.signature.ws.psa.catcert.sadiel.com"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
  xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy"
  xmlns:mtom="http://schemas.xmlsoap.org/ws/2004/09/policy/optimizedmi
meserialization"
  xmlns:sc="http://schemas.sun.com/2006/03/wss/client"
  xmlns:wspp="http://java.sun.com/xml/ns/wsdl/policy"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  targetNamespace="http://policy.signature.ws.psa.catcert.sadiel.com"
  name="signatureService"
  xsi:schemaLocation="
http://schemas.xmlsoap.org/ws/2004/09/policy/
http://schemas.xmlsoap.org/ws/2004/09/policy/ws-policy.xsd
http://schemas.xmlsoap.org/ws/2005/02/trust
http://schemas.xmlsoap.org/ws/2005/02/trust/WS-Trust.xsd
http://schemas.xmlsoap.org/ws/2005/07/securitypolicy
http://schemas.xmlsoap.org/ws/2005/07/securitypolicy/ws-
securitypolicy.xsd
http://schemas.xmlsoap.org/ws/2005/02/rm/policy
http://schemas.xmlsoap.org/ws/2005/02/rm/wsrn-policy.xsd
http://www.w3.org/2005/08/addressing
http://www.w3.org/2005/08/addressing/ws-addr.xsd
http://schemas.xmlsoap.org/ws/2004/09/mex/
http://schemas.xmlsoap.org/ws/2004/09/mex/MetadataExchange.xsd
http://schemas.xmlsoap.org/ws/2004/09/policy/optimizedmimeserializat
ion
http://schemas.xmlsoap.org/ws/2004/09/policy/optimizedmimeserialization/o
ptimizedmimeserialization-policy.xsd">

  <portType name="signature" />
  <binding name="signaturePortBinding" type="tns:signature">
    <wsp:PolicyReference URI="#PSAClientPolicyClient" />
  </binding>
  <service name="signatureService">
    <port name="signaturePort"
      binding="tns:signaturePortBinding">
    </port>
  </service>

  <wsp:Policy wsu:Id="PSAClientPolicyClient">
    <wsp:ExactlyOne>
      <wsp:All>
        <sc:KeyStore wspp:visibility="private"
storepass="1111" alias="elujan" type="PKCS12" location="CDA_Example1.pl2"
/>
        <sc:TrustStore wspp:visibility="private"
location="client-truststore.jks" storepass="changeit" peeralias="xws-
security-server" />
      </wsp:All>
    </wsp:ExactlyOne>
  </wsp:Policy>
```

```
</definitions>
```

L'etiqueta **<sc:KeyStore>** indica a **Metro Stack WS**, quin fitxer p12 utilitzar per realitzar la petició SOAP amb Seguretat **WS X509 Token Profile**[7].

L'etiqueta **<sc:TrustStore>** indica a **Metro Stack WS** quin certificat de confiança públic emprar per xifrar la petició SOAP.

6.2.1.3 Preparació client WSDL2Java

Creem un projecte Java simple amb el **standard layout de maven2**[10] i el **pom.xml** del projecte serà el següent:

pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.sadiel.catcert.psa</groupId>
  <artifactId>clientws_psa</artifactId>
  <version>1.0.0.RC3</version>
  <name>clientws_psa</name>
  <packaging>jar</packaging>
  <url>http://www.sadiel.es</url>
  <repositories>
    <repository>
      <id>maven2-repository.dev.java.net</id>
      <name>Java.net Maven 2 Repository</name>
      <url>http://download.java.net/maven/2</url>
    </repository>
  </repositories>
  <pluginRepositories>
    <pluginRepository>
      <id>maven2-repository.dev.java.net</id>
      <url>http://download.java.net/maven/2</url>
    </pluginRepository>
  </pluginRepositories>

  <profiles>
    <profile>
      <id>client</id>
      <activation>
        <activeByDefault>>false</activeByDefault>
      </activation>
      <build>
        <plugins>
          <plugin>
            <artifactId>maven-compiler-plugin</artifactId>
            <configuration>
```

```

        <source>1.5</source>
        <target>1.5</target>
    </configuration>
</plugin>
<plugin>
    <artifactId>maven-clean-plugin</artifactId>
    <configuration>
        <filesets>
            <fileset>
                <directory>

${project.build.sourceDirectory}/com/sadiel/catcert/psa/ws
            </directory>
            <includes>
            <include>**/*.java</include>
            </includes>
        </fileset>
    </filesets>
    </configuration>
</plugin>

<plugin>
    <groupId>org.codehaus.mojo</groupId>
    <artifactId>jaxws-maven-
plugin</artifactId>

    <version>1.9</version>
    <configuration>
        <sourceDestDir>

${project.build.sourceDirectory}
        </sourceDestDir>
        <extension>>false</extension>
        <genWsdL>>true</genWsdL>
        <keep>>true</keep>
        <protocol>soap1.1</protocol>
        <verbose>>true</verbose>
        <xnoCompile>>true</xnoCompile>
        <xdebug>>true</xdebug>
    </configuration>
    <executions>
        <execution>
            <id>ClientWS</id>
            <goals>

<goal>wsimport</goal>

            </goals>
        </configuration>
        <wsdlUrls>
            <wsdlUrl>

${wsdlUrlClient}
            </wsdlUrl>
        </wsdlUrls>
    </configuration>
</execution>
</execution>

```

```

<id>UtilWS</id>
<goals>

<goal>wsimport</goal>

</goals>
<configuration>
  <wsdlUrls>
    <wsdlUrl>

    </wsdlUrl>
  </wsdlUrls>
</configuration>
</execution>
<execution>
  <id>QueryWS</id>
  <goals>

  <goal>wsimport</goal>

  </goals>
  <configuration>
    <wsdlUrls>
      <wsdlUrl>

      </wsdlUrl>
    </wsdlUrls>
  </configuration>
</execution>
</executions>
</plugin>

</plugins>
</build>
<properties>
  <maven.test.failure.ignore>
    true
  </maven.test.failure.ignore>
  <maven.test.skip>false</maven.test.skip>
</properties>
</profile>
</profiles>
<properties>
  <wsdlUrlClient>
    http://localhost:8080/engineWS/signature/service?wsdl
  </wsdlUrlClient>
  <wsdlUrlUtil>
    http://localhost:8080/engineWS/util/service?wsdl
  </wsdlUrlUtil>
  <wsdlUrlQuery>
    http://localhost:8080/engineWS/query/service?wsdl
  </wsdlUrlQuery>
</properties>
<dependencies>
  <!-- METRO -->
  <dependency>

```

```

        <groupId>com.sun.xml.ws.metro</groupId>
        <artifactId>webservicess-api</artifactId>
        <version>1.2</version>
    </dependency>
    <dependency>
        <groupId>com.sun.xml.ws.metro</groupId>
        <artifactId>webservicess-extra</artifactId>
        <version>1.2</version>
    </dependency>
    <dependency>
        <groupId>com.sun.xml.ws.metro</groupId>
        <artifactId>webservicess-extra-api</artifactId>
        <version>1.2</version>
    </dependency>
    <dependency>
        <groupId>com.sun.xml.ws.metro</groupId>
        <artifactId>webservicess-rt</artifactId>
        <version>1.2</version>
    </dependency>
    <dependency>
        <groupId>com.sun.xml.ws.metro</groupId>
        <artifactId>webservicess-tools</artifactId>
        <version>1.2</version>
    </dependency>
    <!-- -->
</dependencies>
</project>

```

El **pom.xml** de l'exemple presenta les propietats:

- **wsdlUrlClient** <http://localhost:8080/engineWS/signature/service?wsdl>.
- **wsdlUrlUtil** <http://localhost:8080/engineWS/util/service?wsdl>.
- **wsdlUrlQuery** <http://localhost:8080/engineWS/query/service?wsdl>.

Òbviament haurem de canviar les URLs per les correctes que dependran de l'entorn.

Especial rellevància presenta el plugin de **maven2** per **jaxws** (**jaxws-maven-plugin**) [10]

El plugin **jaxws-maven-plugin** ens facilita completament la tasca de generar les classes JAX-WS client, arran d'un WSDL accessible on-line, però també disposem de la opció d'indicar un WSDL local amb els nodes paràmetres **<wsdlFiles>** o **<wsdlDirectory>** (poden treballar alhora).

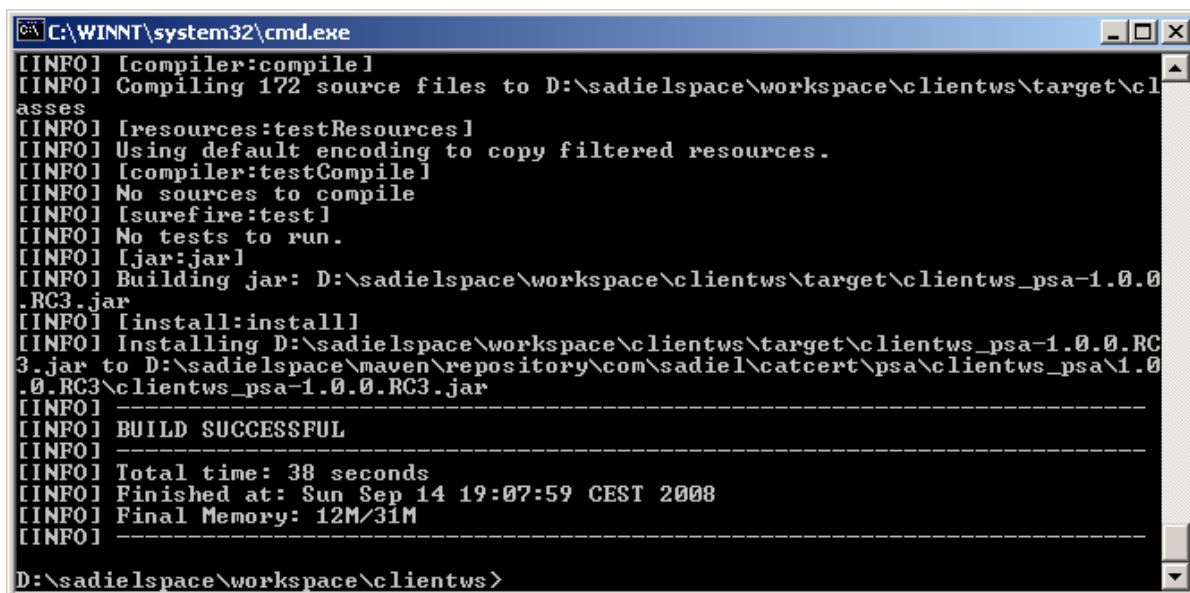
El plugin **jaxws-maven-plugin** genera les classes client dels WS del **SI PSA**, indicats a les propietats **wsdlUrl***, aquestes classes també les tenim a la biblioteca **clientws_psa**, com hem explicat al punt [Client amb l'API del WS de PSA](#)

Per construir les peticions haurem de construir la petició **ad hoc**, sense classes d'utilitats.

6.2.1.4 Generació i compilació

Obrim un *prompt* del sistema operatiu (*command line*) i executem la següent sentència:

>mvn clean install -P client



```

C:\WINNT\system32\cmd.exe
[INFO] [compiler:compile]
[INFO] Compiling 172 source files to D:\sadielspace\workspace\clientws\target\classes
[INFO] [resources:testResources]
[INFO] Using default encoding to copy filtered resources.
[INFO] [compiler:testCompile]
[INFO] No sources to compile
[INFO] [surefire:test]
[INFO] No tests to run.
[INFO] [jar:jar]
[INFO] Building jar: D:\sadielspace\workspace\clientws\target\clientws_psa-1.0.0.RC3.jar
[INFO] [install:install]
[INFO] Installing D:\sadielspace\workspace\clientws\target\clientws_psa-1.0.0.RC3.jar to D:\sadielspace\maven\repository\com\sadiel\catcert\psa\clientws_psa\1.0.0.RC3\clientws_psa-1.0.0.RC3.jar
[INFO]
[INFO] BUILD SUCCESSFUL
[INFO]
[INFO] Total time: 38 seconds
[INFO] Finished at: Sun Sep 14 19:07:59 CEST 2008
[INFO] Final Memory: 12M/31M
[INFO]
D:\sadielspace\workspace\clientws>

```

Fig. 4 Exemple del resultat d'executar mvn

El resultat és un únic fitxer d'extensió jar que el trobarem a la carpeta:

`${base.dir}\target\<finalName>.jar`

On:

base.dir – és el directori del projecte Java, on es troba el **pom.xml**.

finalName – és el valor de la etiqueta **finalName** del **pom.xml** del projecte Java per generar el client.

També trobem la nova llibreria al repositori local de **maven2**, localitzat a la màquina del desenvolupador.

6.2.2 Client Java 1.4

Per generar un client de PSA que funcioni amb Java 1.4 disposem, com en el cas anterior de Java 5 o Java 6, d'una API que ens facilitarà la feina a l'hora d'integrar-nos tot i que el seu ús no és obligatori.

Així, per utilitzar un WS de **SI PSA** des d'una classe Java 1.4 tenim dues opcions:

1. **Treballar amb l'API de WS de Java 1.4 del SI PSA**– Aquesta biblioteca proporciona les classes client preparades pel motor de WS **Axis2** amb **Apache Rampart** i classes d'utilitat que faciliten la generació de les peticions del WS.
2. **From scratch (WSDL2Java)** – Aquesta opció requereix accedir al **WSDL**, generar les classes amb el motor de WS seleccionat i finalment generar tota la petició.

Les peticions dels WS del **SI PSA** són relativament complexes, per això recomanem la primera opció.

6.2.2.1 Client amb l'API del WS per Java 1.4 del SI PSA

Creem un projecte Java simple amb el **standard layout de maven2**[10] i el **pom.xml** del projecte serà similar al següent:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <artifactId>axis2client</artifactId>
  <groupId>com.sadiel.catcert.psa.ws.client</groupId>
  <name>axis2-jdk14</name>
  <packaging>jar</packaging>
  <url>http://www.sadiel.es</url>
  <version>1.0-SNAPSHOT</version>

  <dependencies>
    <dependency>
      <groupId>com.sadiel.catcert.psa.ws.client</groupId>
      <artifactId>axis2client</artifactId>
      <version>X.Y.Z</version>
    </dependency>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.2</version>
    </dependency>
    <dependency>
      <groupId>commons-configuration</groupId>
      <artifactId>commons-configuration</artifactId>
      <version>1.5</version>
    </dependency>

    <!-- Rampart -->
    <dependency>
      <groupId>org.apache.ws.security</groupId>
      <artifactId>wss4j</artifactId>
      <version>1.5.7</version>
      <exclusions>
        <exclusion>
          <groupId>org.slf4j</groupId>
          <artifactId>jcl-over-slf4j</artifactId>
```

```

        </exclusion>
      <exclusion>
        <groupId>org.slf4j</groupId>
        <artifactId>log4j-over-slf4j</artifactId>
      </exclusion>
    <exclusion>
      <groupId>org.apache.woden</groupId>
      <artifactId>woden-api</artifactId>
    </exclusion>
    <exclusion>
      <groupId>org.apache.xmlbeans</groupId>
      <artifactId>xmlbeans</artifactId>
    </exclusion>
  </exclusions>
</dependency>
<dependency>
  <groupId>org.apache.rampart</groupId>
  <artifactId>rampart-trust</artifactId>
  <version>1.4</version>
  <exclusions>
    <exclusion>
      <groupId>org.apache.woden</groupId>
      <artifactId>woden-api</artifactId>
    </exclusion>
  </exclusions>
</dependency>
<dependency>
  <groupId>org.apache.rampart</groupId>
  <artifactId>rampart-core</artifactId>
  <version>1.4</version>
  <exclusions>
    <exclusion>
      <groupId>org.apache.woden</groupId>
      <artifactId>woden-api</artifactId>
    </exclusion>
  </exclusions>
</dependency>
</dependencies>

<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <configuration>
        <source>1.4</source>
        <target>1.4</target>
      </configuration>
    </plugin>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-surefire-plugin</artifactId>
      <version>2.4.2</version>
    </plugin>
  </plugins>
</build>

```

```
</project>
```

Les dependències que utilitza el projecte són:

- **Apache Rampart** que es el motor de WS Security que farem servir.
- **API dels clients WS amb Java 1.4 del SI PSA**, aquesta biblioteca actualment no és accessible a cap repositori públic de maven2 i s'ha de sol·licitar a **CATCert** i hem de deixar-la al repositori local. La biblioteca és:
 - `axis2client` – Conté les classes client dels WS del **SI PSA** i factories que permeten generar peticions. Aquestes factories faciliten la generació de les complexes peticions.

6.2.2.2 Client WSDL2Java con Axis2

Creació del projecte

Creem un projecte Java simple amb el **standard layout de maven2**^[10] i el **pom.xml** del projecte serà similar al següent:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <artifactId>axis2client</artifactId>
  <groupId>com.sadiel.catcert.psa.ws.client</groupId>
  <name>axis2-jdk14</name>
  <packaging>jar</packaging>
  <url>http://www.sadiel.es</url>
  <version>1.0-SNAPSHOT</version>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.2</version>
    </dependency>
    <dependency>
      <groupId>commons-configuration</groupId>
      <artifactId>commons-configuration</artifactId>
      <version>1.5</version>
    </dependency>
    <dependency>
      <groupId>org.apache.axis2</groupId>
      <artifactId>axis2</artifactId>
      <version>1.4.1</version>
    </dependency>
    <dependency>
      <groupId>org.apache.xmlbeans</groupId>
```

```

        <artifactId>xmlbeans</artifactId>
        <version>2.4.0</version>
    </dependency>

<!--AIOM Dependencies-->

<dependency>
    <groupId>org.apache.ws.commons.axiom</groupId>
    <artifactId>axiom-impl</artifactId>
    <version>1.2.8</version>
</dependency>
<dependency>
    <groupId>org.apache.ws.commons.axiom</groupId>
    <artifactId>axiom-api</artifactId>
    <version>1.2.8</version>
</dependency>
<dependency>
    <groupId>org.apache.ws.commons.axiom</groupId>
    <artifactId>axiom-dom</artifactId>
    <version>1.2.8</version>
</dependency>
<dependency>
    <groupId>org.apache.neethi</groupId>
    <artifactId>neethi</artifactId>
    <version>2.0.4</version>
</dependency>
<dependency>
    <groupId>commons-logging</groupId>
    <artifactId>commons-logging</artifactId>
    <version>1.1</version>
</dependency>
<dependency>
    <groupId>log4j</groupId>
    <artifactId>log4j</artifactId>
    <version>1.2.13</version>
</dependency>
<dependency>
    <groupId>commons-httpclient</groupId>
    <artifactId>commons-httpclient</artifactId>
    <version>3.0.1</version>
</dependency>
<dependency>
    <groupId>wsdl4j</groupId>
    <artifactId>wsdl4j</artifactId>
    <version>1.6.1</version>
</dependency>
<dependency>
    <groupId>org.apache.ws.commons.schema</groupId>
    <artifactId>XmlSchema</artifactId>
    <version>1.4.5</version>
</dependency>
<dependency>
    <groupId>stax</groupId>
    <artifactId>stax-api</artifactId>
    <version>1.0.1</version>
</dependency>

```

```
<dependency>
  <groupId>org.codehaus.woodstox</groupId>
  <artifactId>wstx-asl</artifactId>
  <version>3.2.1</version>
</dependency>
<dependency>
  <groupId>backport-util-concurrent</groupId>
  <artifactId>backport-util-concurrent</artifactId>
  <version>2.1</version>
</dependency>
<dependency>
  <groupId>commons-codec</groupId>
  <artifactId>commons-codec</artifactId>
  <version>1.3</version>
</dependency>
<dependency>
  <groupId>commons-fileupload</groupId>
  <artifactId>commons-fileupload</artifactId>
  <version>1.1.1</version>
</dependency>
<dependency>
  <groupId>org.apache.woden</groupId>
  <artifactId>woden-api</artifactId>
  <version>1.0M8</version>
</dependency>
<dependency>
  <groupId>org.apache.woden</groupId>
  <artifactId>woden-impl-dom</artifactId>
  <version>1.0M8</version>
</dependency>
<dependency>
  <groupId>org.apache.woden</groupId>
  <artifactId>woden-impl-dom</artifactId>
  <version>1.0M8</version>
</dependency>

<!-- Rampart -->

<dependency>
  <groupId>org.apache.ws.security</groupId>
  <artifactId>wss4j</artifactId>
  <version>1.5.7</version>
  <exclusions>
    <exclusion>
      <groupId>org.slf4j</groupId>
      <artifactId>jcl-over-slf4j</artifactId>
    </exclusion>
    <exclusion>
      <groupId>org.slf4j</groupId>
      <artifactId>log4j-over-slf4j</artifactId>
    </exclusion>
    <exclusion>
      <groupId>org.apache.woden</groupId>
      <artifactId>woden-api</artifactId>
    </exclusion>
  </exclusions>
</dependency>
```

```

        <groupId>org.apache.xmlbeans</groupId>
        <artifactId>xmlbeans</artifactId>
    </exclusion>
</exclusions>
</dependency>
<dependency>
    <groupId>org.apache.rampart</groupId>
    <artifactId>rampart-trust</artifactId>
    <version>1.4</version>
    <exclusions>
        <exclusion>
            <groupId>org.apache.woden</groupId>
            <artifactId>woden-api</artifactId>
        </exclusion>
    </exclusions>
</dependency>
<dependency>
    <groupId>org.apache.rampart</groupId>
    <artifactId>rampart-core</artifactId>
    <version>1.4</version>
    <exclusions>
        <exclusion>
            <groupId>org.apache.woden</groupId>
            <artifactId>woden-api</artifactId>
        </exclusion>
    </exclusions>
</dependency>
</dependencies>

<build>
    <sourceDirectory>
        ${basedir}/src/main/src/
    </sourceDirectory>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-compiler-plugin</artifactId>
            <configuration>
                <source>1.4</source>
                <target>1.4</target>
            </configuration>
        </plugin>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-surefire-plugin</artifactId>
            <version>2.4.2</version>
        </plugin>
        <plugin>
            <groupId>org.codehaus.mojo</groupId>
            <artifactId>build-helper-maven-plugin</artifactId>
            <version>1.1</version>
            <executions>
                <execution>
                    <id>add-source</id>
                    <phase>generate-sources</phase>
                </execution>
            </executions>
        </plugin>
    </plugins>
</build>

```

```

        <goals>
            <goal>add-source</goal>
        </goals>
        <configuration>
            <sources>
                <source>${basedir}/src/main/src</source>
                <source>${basedir}/src/main/java</source>
                <source>${basedir}/src/main/resources</source>
            </sources>
        </configuration>
    </execution>
</executions>
</plugin>
</plugins>
</build>

<profiles>
    <profile>
        <id>client</id>
        <build>
            <plugins>
                <plugin>
                    <artifactId>maven-clean-plugin</artifactId>
                    <configuration>
                        <filesets>
                            <fileset>
                                <directory>${basedir}/src/main/src
                                </directory>
                                <includes>
                                    <include>**/*.java</include>
                                </includes>
                            </fileset>
                            <fileset>
                                <directory>
                                    ${basedir}/src/main/resources/schemaorg_apache_xmlbeans
                                </directory>
                                <includes>
                                    <include>**/*</include>
                                </includes>
                            </fileset>
                        </filesets>
                    </configuration>
                </plugin>
                <plugin>
                    <groupId>org.apache.axis2</groupId>
                    <artifactId>axis2-wsdl2code-maven-
plugin</artifactId>

                    <version>1.4.1</version>
                    <executions>
                        <execution>
                            <id>SignatureWS</id>
                            <goals><goal>wsdl2code</goal>
                            </goals>
                            <configuration>
                                <packageName>
com.sadiel.catcert.psa.ws.client

```



```

        </packageName>
        <wsdlFile>${wsdlUrlClient}</wsdlFile>
    </dataBindingName>xmlbeans</dataBindingName>
    <outputDirectory>src/main</outputDirectory>
    <generateAllClasses>false</generateAllClasses>
    <generateServerSide>false</generateServerSide>
    <language>java</language>
    </configuration>
    </execution>
    <execution>
        <id>QueryWS</id>
        <goals>
            <goal>wsdl2code</goal>
        </goals>
        <configuration>
    </packageName>com.sadiel.catcert.psa.ws.client</packageName>
    <wsdlFile>${wsdlUrlQuery}</wsdlFile>
    <dataBindingName>xmlbeans</dataBindingName>
    <outputDirectory>src/main</outputDirectory>
    <generateAllClasses>false</generateAllClasses>
    <generateServerSide>false</generateServerSide>
    </configuration>
    </execution>
    <execution>
        <id>UtilWS</id>
        <goals>
            <goal>wsdl2code</goal>
        </goals>
        <configuration>
    </packageName>com.sadiel.catcert.psa.ws.client</packageName>
    <wsdlFile>${wsdlUrlUtil}</wsdlFile>
    <dataBindingName>xmlbeans</dataBindingName>
    <outputDirectory>src/main</outputDirectory>
    <generateAllClasses>false</generateAllClasses>
    <generateServerSide>false</generateServerSide>
    </configuration>
    </execution>
    </executions>
    </plugin>
    </plugins>
    </build>
    </profile>
    <profile>
        <id>tests</id>
        <properties>
            <maven.test.skip>false</maven.test.skip>
        </properties>
    </profile>
    </profiles>

    <properties>
        <maven.test.skip>true</maven.test.skip>

    <wsdlUrlClient>
        http://localhost:8080/engineWS/signature/service?wsdl
    </wsdlUrlClient>

```

```
<wsdlUrlUtil>
    http://localhost:8080/engineWS/util/service?wsdl
</wsdlUrlUtil>
<wsdlUrlQuery>
    http://localhost:8080/engineWS/query/service?wsdl
</wsdlUrlQuery>
</properties>

</project>
```

El **pom.xml** de l'exemple presenta les propietats:

- **wsdlUrlClient** <http://localhost:8080/engineWS/signature/service?wsdl>.
- **wsdlUrlUtil** <http://localhost:8080/engineWS/util/service?wsdl>.
- **wsdlUrlQuery** <http://localhost:8080/engineWS/query/service?wsdl>.

Òbviament haurem de canviar les URLs per les correctes que dependran de l'entorn.

Especial rellevància presenta el plugin de **maven2** per axis2 (**axis2-wsd12code**)

El plugin **axis2-wsd12code-maven-plugin** ens facilita completament la tasca de generar les classes de Stub del client, a partir d'un WSDL accessible on-line

El plugin **axis2-wsd12code-maven-plugin** genera les classes client dels WS del **SI PSA**, indicats a les propietats **wsdlUrl***, aquestes classes també les tenim a la biblioteca **axis2client**, com hem explicat al punt [Client amb l'API del WS de PSA per Java 1.4](#).

Per construir les peticions haurem de construir la petició **ad hoc**, sense classes d'utilitats.

Generació i compilació

1. El primer pas serà generar les classes del client.

Com es pot observar al POM hi ha definit un profile anomenat **client**. Així, executant:

```
mvn -P client clean install
```

es generaran las clases del client a partir dels WSDLs definits a les propietats del Pom. La carpeta on el plugin genera aquestes classes és **[Nom_Del_Projecte]/src/main/src**.

Nota d'interès: La ruta que Maven 2 recomana per ubicar totes les classes Java és **[Nom_Del_Projecte]/src/main/java** però no és possible configurar aquest

OutputDirectory ja que el plugin la cadena "src" a l'outputDirectory que li especifiquem per configuració.

2. Entre les classes generades es troben les conegudes com classes de stub que són aquelles sobre les quals farem les invocacions als WS. Estan ubicades dins de **[Nom_del_projecte]/src/main/src/com/sadiel/catcert/psa/ws/client** i són les següents:
 - a. QueryServiceStub.java
 - b. SignatureServiceStub.java
 - c. UtilServiceStub.java

Haurem de realitzar unes modificacions d'aquestes classes a fi de solucionar un problema de compatibilitat entre el WSDL generat per Metro i el parseig que en fa Apache Rampart.

Les modificacions a realitzar a les 3 classes citades anteriorment són les següents:

A les línies que comencen per:

```
(__operation).getMessage(org.apache.axis2.wsdl.WSDLConstants.MESSAGE_LABEL_OUT_VALUE)...
```

i

```
(__operation).getMessage(org.apache.axis2.wsdl.WSDLConstants.MESSAGE_LABEL_IN_VALUE)...
```

eliminar els tags <ExactlyOne> i <All> dins de <AlgorithmSuite>

Exemple il·lustratiu:

```
<ns7:AlgorithmSuite>\n<ns1:Policy>\n<ns1:ExactlyOne>\n<ns1:All>\n<ns7:Basic128/>\n</ns1:All>\n</ns1:ExactlyOne>\n</ns1:Policy>\n</ns7:AlgorithmSuite>
```

passa a ser:

```
<ns7:AlgorithmSuite>\n<ns1:Policy>\n<ns7:Basic128/>\n</ns1:Policy>\n</ns7:AlgorithmSuite>
```

3. El darrer pas serà recompilar les classes amb les modificacions realitzades. Per fer-ho executarem:

```
mvn clean install
```

Apache Rampart

El client per Java 1.4 fa ús d'Apache Rampart 1.4 com a mòdul de WS Security de Axis 2. La release 1.4 s'Apache Rampart descarregable des de <http://ws.apache.org/rampart/download.html> **NO és compatible** amb **JDK 1.4**. Al SVN d'Apache hi ha un branch compatible amb JDK 1.4 (https://svn.apache.org/repos/asf/webservices/rampart/branches/java/1_4/). El problema és que no han canviat la versió en el pom.xml i no hi ha manera de distingir la versió compatible de la no compatible.

Per facilitar la feina de l'integrador Catcert proporcionarà les dependències de Rampart següents compatibles amb Java 1.4:

1. rampart-trust-1.4.jar
2. rampart-core-1.4.jar
3. rampart-policy-1.4.jar

Amb **GroupId: com.sadiel.org.apache.rampart**

amb els corresponents Pom.xml:

1. rampart-trust-1.4.pom
2. rampart-core-1.4.pom
3. rampart.policy-1.4.pom
4. rampart-project-1.4.pom

que hauran de ser instal·lades en el repositori privat de Maven (o un repositori d'empresa com Artifactory).

Un altra component clau que requereix el client Java 1.4 és el mòdul de Rampart **rampart-1.4.mar**. En els exemples de codi veurem com es carrega abans de fer servir les classes de stub.

Si es fa servir la API de client 1.4 proporcionada per Catcert com recomanem, el recurs **rampart-1.4.mar** ja està inclòs a /META-INF/rampart-1.4.mar.

Si, d'altra banda, no es fa servir la API, es pot aconseguir des del repositori de SVN d'Apache a la branch de java 1.4 (https://svn.apache.org/repos/asf/webservices/rampart/branches/java/1_4/) fent una construcció amb Maven estàndar.

Nota: Per la construcció amb Maven del fitxer .mar és necessari disposar de les JCE a la màquina virtual.

6.2.3 Client .NET

6.2.3.1 Preparació

El pas de preparació per a la creació i execució del client de WebService del **SI PSA** únicament consisteix en verificar l'accessibilitat del fitxer de descripció del servei Web a invocar com hem especificat [anteriorment](#), per exemple, per al servei Web de signatura obrint una instància del nostre navegador amb la **URL** del **WSDL** d'aquets servei (<http://www.preproduccio.psa.cat/engineWS/signature/service?wsdl>).

6.2.3.2 Generació

Per a generar les classes per al client de servei Web del **SI PSA**, realitzarem els següents passos des de l'entorn de desenvolupament **Microsoft Visual Studio 2005**. En aquest exemple generarem el client per al servei Web de signatura del **PSA**:

5. Obrim l'entorn de desenvolupament de Visual Studio i creem un projecte de tipus Visual Basic o triem un projecte d'aquest tipus ja creat.
6. Dintre de la finestra de l'explorador de solucions, triem el node de *Web References*, polsem el botó dret i l'opció d' *Agregar referencia Web*.

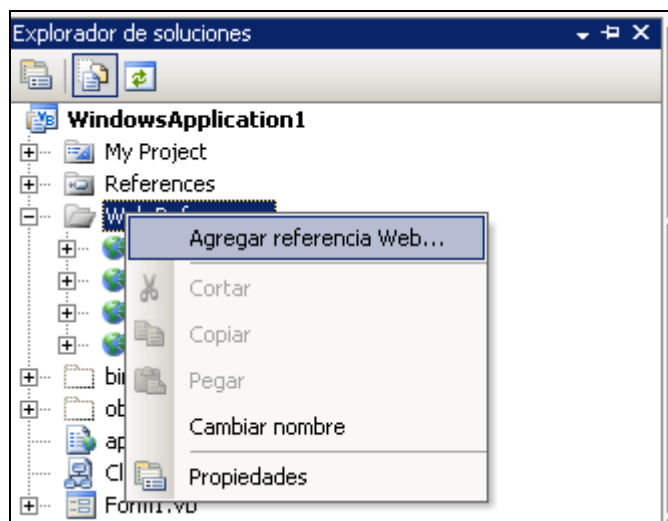


Fig. 5 – Agregar referencia Web al nostre projecte

7. A la finestra que es mostra tenim que afegir la **URL** del fitxer de descripció del servei Web (<http://www.preproduccio.psa.cat/engineWS/signature/service?wsdl>) que ja hem verificat que es troba accessible i polsem el botó *Ir*.

A continuació es mostra una nova plana amb tots el mètodes del servei Web trobat, triem un nom i polsem el botó *Agregar Referència* per a la generació de les classes d'accés al servei Web.

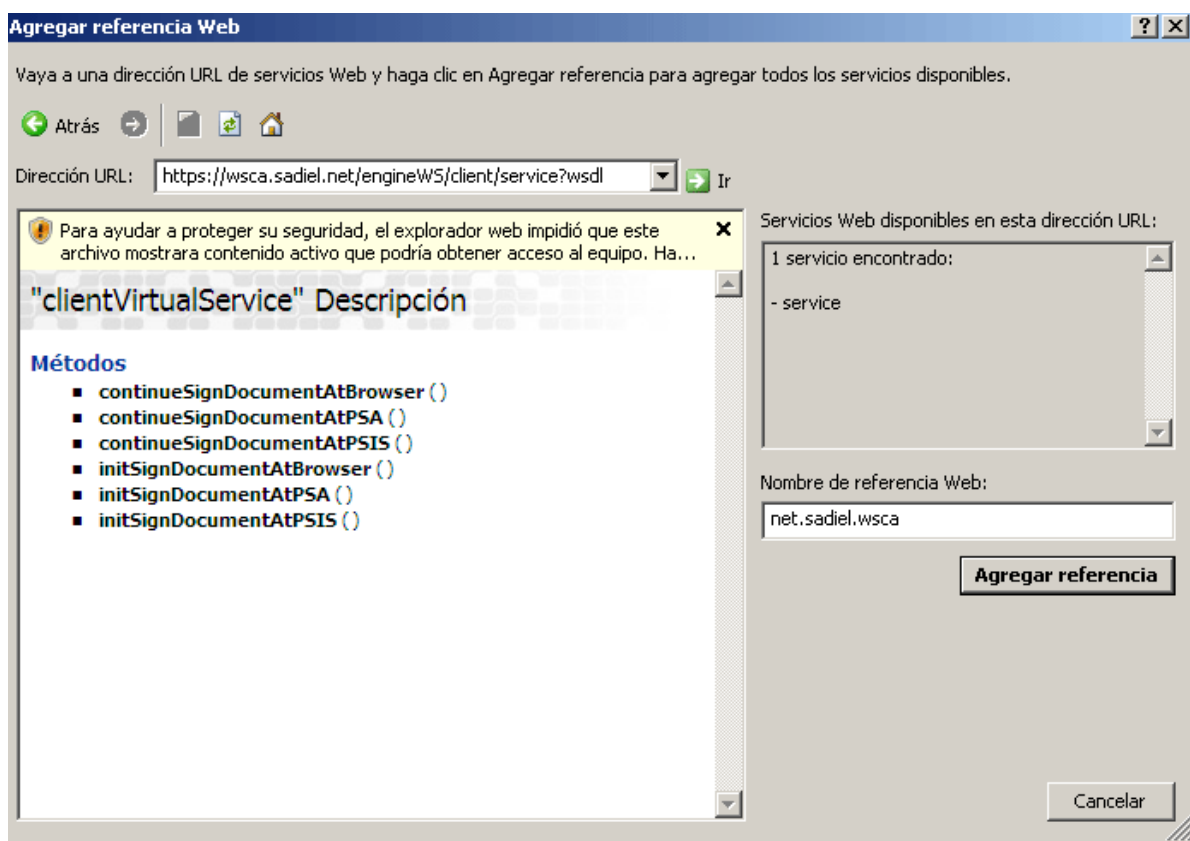


Fig. 6 – Servei Web de Signatura trobat

Si no hem configurat correctament la [confiança](#) del certificat d'aplicació del **SI PSA** o hagi algú problema amb el certificat, es mostrarà un missatge d'advertència referent a la validesa o el nom del certificat. Hem de revisar l'estat del certificat d'aplicació per a evitar aquest missatge.

Ara podem verificar la correcta creació de les classes que utilitzarem per a la invocació del servei Web a la vista de l'explorador de solucions o a la de classes.

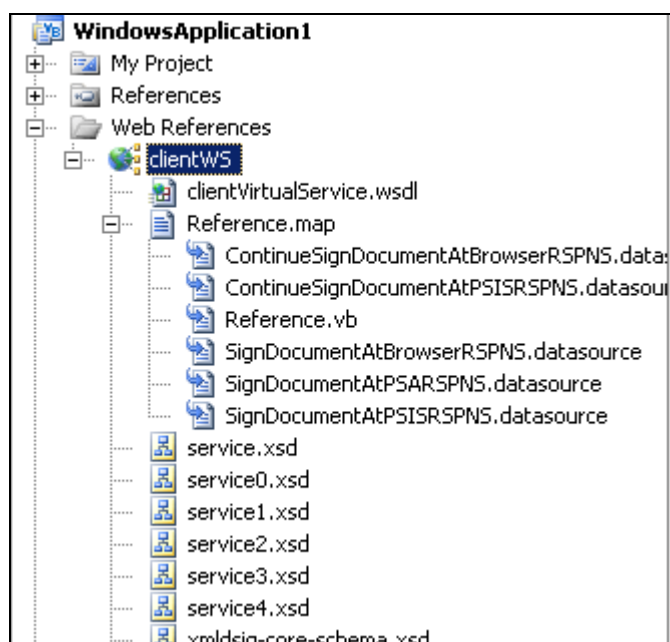


Fig. 7 – Elements del servei Web afegit

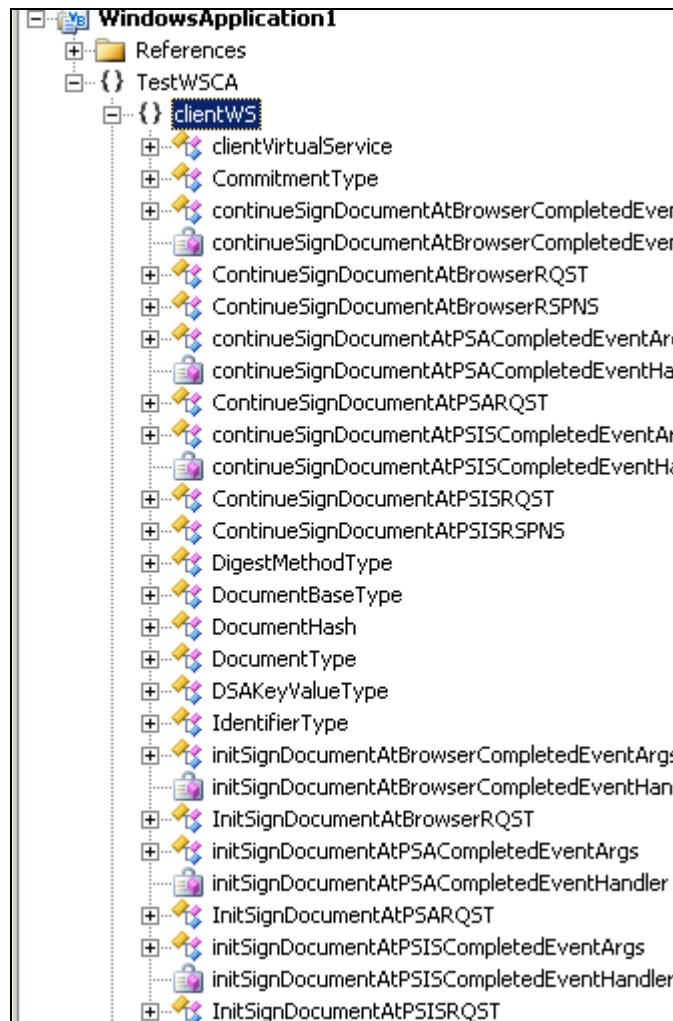


Fig. 8 – Vista de classes del servei Web afegit

6.2.3.3 Compilació

Una vegada s'ha realitzat la generació de les classes del servei Web, i encara que no hàgim obtingut cap error, hem de realitzar un procés automàtic de compilació per a confirmar que el fitxer de descripció **WSDL** del servei Web s'ha interpretat de manera correcta i no hi ha cap problema.

Per aquesta compilació de revisió, podem utilitzar l'opció de generació de Visual Studio o la de depuració, depenent del nostre tipus de projecte. A la fi, el que necessitem es que el compilador ens retorna el resultat amb cap error:

===== Generar: 1 correctos o actualizados, 0 incorrectos, 0 omitidos =====

Amb això, ja estem preparats per a la introducció de la lògica de negoci del nostre client de servei Web per a la seva execució.

6.2.3.4 Execució

Per a poder realitzar l'execució del nostre client de servei Web, ens queda pendent realitzar una mínima modificació a les classes autogenerades. Aquesta modificació es necessària per a la inclusió del token de seguretat per a la correcta composició de la petició amb seguretat de servei Web (**WSS[7]**), ja que els serveis Web del **SI PSA** han estat desenvolupats usant aquesta capacitat. Aquest comportament es troba explicat detalladament al document de seguretat del **SI PSA** (06-0435-DS-0001-02-Security_PSA.doc).

A continuació es detallen els passos necessaris per a la inclusió del component de seguretat de Web Services al nostre client:

1. Per a la inclusió de seguretat de client de web Service, es necessari el component Microsoft Web Services Enhacements (**WSE**) v 3.0, des d'aquesta URL fem la descàrrega

<http://www.microsoft.com/downloads/details.aspx?familyid=018a09fd-3a74-43c5-8ec1-8d789091255d&displaylang=en#Overview>

2. Una vegada feta la descàrrega, comencem la instal·lació del component amb el Wizard d'instal·lació. Durant aquesta instal·lació únicament es necessari la instal·lació Runtime.
3. Quan ha finalitzat aquesta instal·lació, tornem al nostre projecte de Visual Studio i a l'explorador de solucions, triem l'opció d'*Agregar referència*. Ara examinem la referència Microsoft.Web.Services3, dintre de la pestanya de .NET i polsem el botó *Aceptar*.

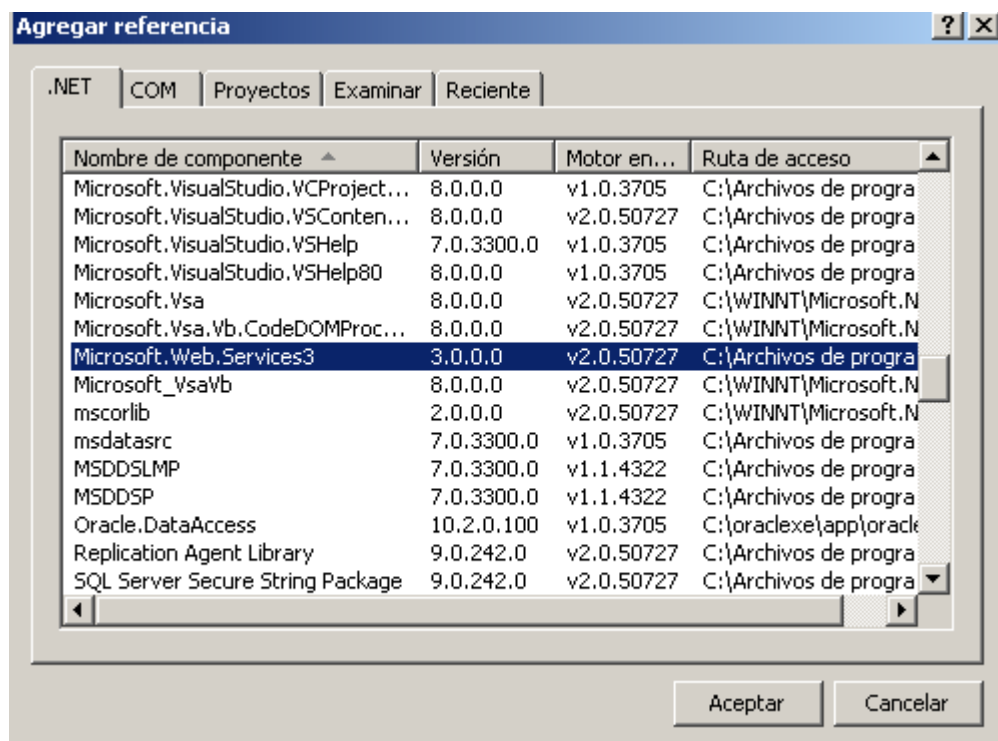


Fig. 9 – Referència Microsoft.Web.Services3

4. A continuació, tenim que realitzar una substitució d'una classe del nostre servei Web. Obrim el fitxer Reference.vb del servei Web afegit, i substituïm la línia

`Inherits System.Web.Services.Protocols.SoapHttpClientProtocol`

Per la línia

`Inherits Microsoft.Web.Services3.WebServicesClientProtocol`

5. Ara tenim que especificar dintre del nostre projecte la política de seguretat del servei Web a complir. Aquesta política de seguretat s'especifica dintre de dos fitxers, el fitxer app.config i dintre d'aquest una referència a la política (wse3policyCache.config). A continuació es mostren les parts a afegir dintre del fitxer app.config.

Política de seguretat relativa al fitxer app.config

```
<microsoft.web.services3>
  <security>
    <securityTokenManager>
      <add localName="EncryptedKey"
type="Microsoft.Web.Services3.Security.Tokens.EncryptedKeyTokenManager,
Microsoft.Web.Services3, Version=3.0.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35 "
namespace="http://www.w3.org/2001/04/xmlenc#">
      <keyAlgorithm name="AES128" />
    </add>
  </securityTokenManager>
</security>
<policy fileName="C:\wse3policyCache.config" />
</microsoft.web.services3>
```

En la següent taula es mostra el contingut complet del fitxer de política de seguretat wse3policyCache.config

Política de seguretat wse3policyCache.config

```
<policies xmlns="http://schemas.microsoft.com/wse/2005/06/policy">
  <extensions>
    <extension name="mutualCertificate10Security"
type="Microsoft.Web.Services3.Design.MutualCertificate10Assertion,
Microsoft.Web.Services3, Version=3.0.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35 " />
    <extension name="x509"
type="Microsoft.Web.Services3.Design.X509TokenProvider,
Microsoft.Web.Services3, Version=3.0.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35 " />
    <extension name="requireActionHeader"
type="Microsoft.Web.Services3.Design.RequireActionHeaderAssertion,
Microsoft.Web.Services3, Version=3.0.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35 " />
  </extensions>
```

```
<policy name="PSAClientPolicy">
  <mutualCertificate10Security establishSecurityContext="false"
  renewExpiredSecurityContext="true" requireSignatureConfirmation="false"
  messageProtectionOrder="SignBeforeEncryptAndEncryptSignature"
  requireDerivedKeys="false" ttlInSeconds="3600">
    <protection>
      <request signatureOptions="IncludeSoapBody" encryptBody="false"/>
      <response signatureOptions="IncludeSoapBody" encryptBody="false"/>
      <fault signatureOptions="IncludeSoapBody" encryptBody="false" />
    </protection>
  </mutualCertificate10Security>
  <requireActionHeader />
</policy>
</policies>
```

6. Amb això ja tenim el client correctament configurat, a continuació es mostra un exemple de codi que realitza una petició buida amb aquest client. El resultat d'aquesta petició ha de ser un missatge d'error de petició buida, en el següent punt del document es mostra la lògica de negoci per a la invocació als serveis.

Exemple d'invocació de servei Web del programari PSA

```
Dim objService As New ClientWS.clientVirtualService

Dim objRequest As New ClientWS.InitSignDocumentAtPSARQST
Dim objResponse As New ClientWS.SignDocumentAtPSARSPNS

Try

    ' Indica que el response disposa de codificació MTOM
    objService.RequireMtom = True

    ' Especifiquem la política de seguretat a utilitzar
    objService.SetPolicy("PSAClientPolicy")

    ' Certificat afegit con a X509Token, el de l'aplicació de gestió
    Dim X509TokenClient As New X509Certificate2("C:/CDA_APP1.p12", "1111")

    ' Certificat del servidor PSA per a xifrar la request
    Dim X509CertServer As New X509Certificate2("C:/CDA_PSA.cer")

    ' Afegim els certificats
    objService.SetClientCredential(New X509SecurityToken(X509TokenClient))
    objService.SetServiceCredential(New X509SecurityToken(X509CertServer))

    .....
    'Lògica de negoci
    .....

    'Enviem la petició
    objResponse = objService.initSignDocumentAtPSA(objRequest)

Catch ex As Exception
```

[End Try](#)

6.3 Exemples de codi

En els següents punts es poden consultar els exemples de codi per a la invocació dels diferents serveis del PSA per a totes les seves funcionalitats.

6.3.1 Client Java

Com hem avançat en el punt [Client Java](#) la construcció del client Java pot utilitzar una sèrie de classes que facilitem en biblioteques Java, sobre tot per la construcció de les peticions.

Algunes d'aquestes classes que faciliten la generació de les sol·licituds:

- `com.sadiel.catcert.psa.api.client.model.DocumentMetaData`
- `com.sadiel.catcert.psa.spi.client.factory.PSARequestFactory`
- `com.sadiel.commons.helper.NIOUtils`
- `com.sadiel.commons.helper.X509Utils`

Aquestes classes seran accessibles com una llibreria externa, instal·lades a un repositori privat corporatiu de **maven2**.

6.3.1.1 Client Java utilitzant l'API

Primer de tot necessitem obtenir el port del servei web. El següent exemple mostra com obtenir el port:

Exemple de creació del Port

```
public static final QName QNAME_SIGNATURE_DEV = new
QName("http://policy.signature.ws.psa.catcert.sadiel.com",
"signatureService");

public Signature getPort(){
    URL url = null;
    try {
        url = new
URL("http://localhost:8080/engineWS/signature/service?wsdl");
    } catch (MalformedURLException e) { //
        log.error("getPort()--->" + e.getMessage(), e);
    }
    // TO environment concrete:
    // DEV, INT, PRO:
    SignatureService service = new SignatureService(url,
QNAME_SIGNATURE_DEV);
    Signature port = service.getSignaturePort();
    BindingProvider bp = (BindingProvider) port;
```

```

        /*
        * to use MTOM
        */

        ((SOAPBinding)(bp.getBinding())).setMTOMEnabled(this.isMtomEnabled());

        return port;
    }

```

A continuació un exemple per demanar al **SI PSA** la creació d'un context de signatura amb certificat al **SI PSA**:

Exemple de creació de signatura amb certificat al PSA

```

public static final String FILENAME_PDF_1 = "Documentol.pdf";
public static final String PROOF_OF_ORIGIN_SEMANTICS = "Proof of origin
indicates
that the signer recognizes to have created, approved and sent the message";
public static final String PROOF_OF_ORIGIN_FIELD_OF_APP = "Proof of origin";
public static final String PROOF_OF_ORIGIN_ID =
"urn:oid:1.2.840.113549.1.9.9.16.6.1";
public static final String PROOF_OF_ORIGIN_DESCRIPTION = "Proof of
origin";

public void signSingleDocumentAtPSA(final String locale, final String ppOID,
final String fileName_cert, final String fileName_PDF){
    try{
        List<InitSignDocumentAtPSARQST> reqList = new
        ArrayList<InitSignDocumentAtPSARQST>();
        reqList.add(this.getInitAtPSARQSTSingleDocument(ppOID,
        fileName_PDF,
        false));
        X509Certificate certificate =
        X509Utils.getX509CertificateFromBytes(NIOUtils.resourceAsBytes(fileName_cert));
        MultiSignDocumentAtPSARSPNS multiResponse =
this.wsService.initSignDocumentAtPSA(PSARequestFactory._().
        createInitSignDocumentAtPSARQST(
            reqList, locale, X509Utils.getIssuer(certificate),
            certificate.getSerialNumber()));
        catch (Exception e) {
            log.error("signSingleDocumentAtPSA()---> " + e.getMessage(),e);
        }
    }
}

```

Destaquem:

- El port construït amb el mètode `getPort()`, torna el servei i el podem utilitzar com qualsevol classe Java.

- La petició es construeix utilitzant la classe d'utilitat :
PSARequestFactory.__().createInitSignDocumentAtPSARQST

La classe PSARequestFactory, genera les sol·licituds complexes a partir de classes primitives (String, BigInteger, boolean) i pròpies de la JVM (List).

A continuació un exemple per demanar al **SI PSA** una signatura PDF visible amb certificat al **SI PSA**:

Exemple de signatura visible de PDF amb certificat al PSA

```
public void signVisibleSignature(final String locale, final String ppOID,
final String fileName_cert, final String fileName_PDF) {
    try {
        List<InitSignDocumentAtPSARQST> reqList = new
            ArrayList<InitSignDocumentAtPSARQST>();
        reqList.add(this.getInitAtPSARQSTVisible(ppOID,
            fileName_PDF));
        X509Certificate certificate = 509Utils.getX509CertificateFromBytes(
            NIOUtils.resourceAsBytes(fileName_cert));
        MultiSignDocumentAtPSARSPNS multiResponse = this.signature1.
            initSignDocumentAtPSA(PSARequestFactory.__( ).
                createInitSignDocumentAtPSARQST(
                    reqList, locale, certificate));
        SignDocumentAtPSARSPNS response =
            multiResponse.getSignDocumentAtPSARSPNS().get(0);
        String procedureContextOID = response.getProcedureContextOID();
        String signatureOID = response.getSignatureActOID();
    } catch (Exception e) {
        log.error("signVisibleSignature()--->Error: "+e.getMessage(),e);
    }
}

private InitSignDocumentAtPSARQST getInitAtPSARQSTVisible (final String ppOID,
final String fileName_PDF) {
    List<DocumentMetaData<InputStream>> docs =
        new ArrayList<DocumentMetaData<InputStream>>();
    InputStream is = NIOUtils.resourceAsStream(fileName_PDF);
    DocumentMetaData<InputStream> doc =
        new DocumentMetaData<InputStream>();
    doc.setData(is);
    doc.setFilename(fileName_PDF.substring(fileName_PDF.lastIndexOf("/"),
        fileName_PDF.length()));
    docs.add(doc);
    VisibleSignatureDataType visibleSignatureData =
        this.getVisibleSignatureData();
    InitSignDocumentAtPSARQST request =
        PSARequestFactory.__( ).createInitSignDocumentAtPSARQSTInputStream(
            ppOID, null, PROOF_OF_ORIGIN_DESCRIPTION,
            PROOF_OF_ORIGIN_SEMANTICS, PROOF_OF_ORIGIN_FIELD_OF_APP,
            PROOF_OF_ORIGIN_ID, QualifierType.OID_AS_URN,
            docs, true, true, true, isVisible, visibleSignatureData);
    return request;
}
```

```

    }

    private VisibleSignatureDataType getVisibleSignatureData() {
        VisibleSignatureDataType visibleSignatureData =
            new VisibleSignatureDataType();
        visibleSignatureData.setCertificationLevel(0);
        visibleSignatureData.setDate(true);
        visibleSignatureData.setImage(false);
        visibleSignatureData.setLocation(true);
        visibleSignatureData.setReason(true);

        VisibleSignaturePosition position = new VisibleSignaturePosition();
        RectangleInfoType rectangleInfo = new RectangleInfoType();
        rectangleInfo.setLowerLeftX(BigInteger.valueOf(100));
        rectangleInfo.setLowerLeftY(BigInteger.valueOf(100));
        rectangleInfo.setUpperRightX(BigInteger.valueOf(300));
        rectangleInfo.setUpperRightY(BigInteger.valueOf(200));
        rectangleInfo.setPageNumber("last");
        position.setRectangleInfo(rectangleInfo);
        visibleSignatureData.setVisibleSignaturePosition(position);
        CertificateDataListType certList = new CertificateDataListType();
        certList.getCertificateData().add(CertificateDataType.COMMON_NAME);
        certList.getCertificateData().add(CertificateDataType.COUNTRY);
        certList.getCertificateData().add(CertificateDataType.GIVEN_NAME);
        certList.getCertificateData().add(
            CertificateDataType.ORGANIZATION_NAME);
        certList.getCertificateData().add(CertificateDataType.SERIAL_NUMBER);
        certList.getCertificateData().add(CertificateDataType.SURNAME);
        certList.getCertificateData().add(CertificateDataType.TITLE);
        visibleSignatureData.setInfoText(certList);
        return visibleSignatureData;
    }

```

Com abans, aquest exemple també aprofita `PSARequestFactory`, per generar la petició.

A continuació un exemple de creació de context de signatura amb certificat en possessió de l'usuari:

Exemple de creació de context signatura amb certificat en possessió de l'usuari

```

public static final String UC_1_SIGNER_1 = "META-INF/signer1.cer";
public static final String DOC1_UC_1 = "docs_uc1/Solicitud.pdf ";
public static final String FUNCTIONAL_CONTEXT_SIGNER1 = "PP-0012";

public void signer_1(){
    SignDocumentAtBrowserRSPNS response = null;
    try{
        response = this.getPort().initSignDocumentAtBrowser(
            this.getRequestSigner1());
        if (response.getResult() != null){

```



```

        this.studyResponse(response.getResult());
    }
    this.accessURL = response.getAccessUrl();
    this.procedureContextOID = response.getProcedureContextOID();
    log.debug("fileUpload_1()---> procedureContextOID: " +
this.procedureContextOID);
    } catch (Exception e) {
        log.error("fileUpload()--->Error: "+e.getMessage(),e);
    }
}

private InitSignDocumentAtBrowserRQST getRequestSigner1(){

    byte[] certificate = NIOUtils.resourceAsBytes(UC_1_SIGNER_1);
    File uploadFile = NIOUtils.resourceAsFile(DOC1_UC_1);
    log.debug("getRequestUC1()---> File: " + uploadFile);
    return PSAResquestFactory._().createInitSignDocumentAtBrowserRQST(
        FUNCTIONAL_CONTEXT_SIGNER1,
        "ES",
        certificate,
        uploadFile);
}

```

Aquest exemple presenta el mateix plantejament, aprofitar PSAResquestFactory, per generar la petició.

Exemple de Xifrat XML

```

public void testCipherXML() {
    log.debug("testCipher()---> Before Call Webservice: " +
this.psaUtilService);
    EncryptDocumentRSPNS response =
this.psaUtilService.cipherDocument(this.buildCipherDocumentResquest(
        DOC_CIPHER_XML_UC_6,
        UC_6_CYPHER,
        EncryptionType.XML_ENCRYPT));
    log.debug("testCipher()---> After Call Webservice " + response);
    if (response.getResult() != null) {
        this.studyResponse(response.getResult());
    }
}

private CipherDocumentRQST buildCipherDocumentResquest(String filename,
String X509Filename, EncryptionType encryptionType) {
    DocumentMetaData<InputStream> doc = new DocumentMetaData<InputStream>();
    List<byte[]> certificates = new ArrayList<byte[]>();

    InputStream fis = NIOUtils.resourceAsStream(X509Filename);
    doc.setData(NIOUtils.resourceAsStream(filename));
    doc.setFilename(filename);
}

```



```
X509Certificate cert = X509Utils.getX509Certificate(fis);
try {
    certificates.add(cert.getEncoded());
} catch (CertificateEncodingException e) {
    log.error(e.getMessage(), e);
}

CipherDocumentRQST request =
    PSAResultFactory._().createCipherDocumentRQSTInputStream(doc,
certificates, encryptionType, "es");

return request;
}
```

Exemple de Desxifrat XML

```
public void testDecipherXML(){
    log.debug("testDecipher()---> Before Call Webservice: " +
        this.psaUtilService); EncryptDocumentRSPNS response =
    this.psaUtilService.decipherDocument(
        this.buildDecipherDocumentRequest(
            DOC_DECIPHER_XML_UC_6,
            UC_6_DECYPHER,
            EncryptionType.XML_ENCRYPT));
    log.debug("testDecipher()---> After Call Webservice " + response);
    if (response.getResult() != null){
        this.studyResponse(response.getResult());
    }

    DocumentType document = response.getDocument();
    byte[] documentFromResponse = null;
    byte[] documentResult = null;
    try {
        documentFromResponse =
            NIOUtils.getBytes(document.getData().getInputStream());
        documentResult =
            NIOUtils.resourceAsBytes(DOC_CIPHER_XML_UC_6);
        NIOUtils.writeDataToTempFile(
            documentFromResponse,
            "D://Docs_decyphered");
        log.debug("testDecipher()---> Document decyphered correctly");

    } catch (IOException e) {
        log.error(e.getMessage());
    }

    private DecipherDocumentRQST buildDecipherDocumentRequest(String
filename,
    String X509Filename, EncryptionType encryptionType) {
        DocumentMetaData<byte[]> doc = new DocumentMetaData<byte[]>();
        doc.setData(NIOUtils.resourceAsBytes(filename));
        doc.setFilename(filename);
    }
}
```

```
byte[] certificate = NIOUtils.resourceAsBytes(X509Filename);
X509Certificate cert = X509Utils
    .getX509CertificateFromBytes(certificate);

DecipherDocumentRQST request = PSAResponseFactory._()
    .createDecipherDocumentRQSTArrayBytes(
        doc,
        X509Utils.getIssuer(cert),
        cert.getSerialNumber(),
        encryptionType, "es");

return request;
}
```

Exemple de Descàrrega de tiquet de validació de signatura

```
public void testDownloadPSISTicket(String signatureID) {
    DownloadVerifySignatureTicketRSPNS response;
    DownloadVerifySignatureTicketRQST request = PSAResponseFactory._().
        createDownloadVerifySignatureTicketRQST(signatureID, SPANISH_LOCALE);
    response = this.query1.downloadVerifySignatureTicket(request);
    if (response.getValidationTicket() != null) {
        try {
            byte[] documentFromResponse = response.getValidationTicket();
            NIOUtils.writeDataToTempFile(documentFromResponse, "D:/temp/");
        } catch (IOException e) {
            log.error("testDownloadPSISTicket()---> " + e.getMessage());
        }
    }
}
```

Exemple de InitSignDocumentAtBrowser

```
private void initInBrowser(String certPath) {
    /* call WebService */
    try {
        log.debug("fileUpload()--->Before Call WebService: "
            + this.psaService);
        SignDocumentAtBrowserRSPNS response = this.psaService
            .initSignDocumentAtBrowser(this
                .getSignDocumentAtBrowserRQST(certPath));
        log.debug("fileUpload()--->After Call WebService " + response);
        if (response.getResult() != null) {
            this.studyResponse(response.getResult());
        }
        this.accessURL = response.getAccessUrl();
        this.procedureContextOID = response.getProcedureContextOID();
        this.isContinue = true;
    }
}
```

```

        log.debug("fileUpload()--->After Call Webservice. URL: "
            + response.getAccessUrl());
    } catch (IOException e) {
        // TODO Auto-generated catch block
        log.error("fileUpload()--->Error: " + e.getMessage(), e);
    } catch (Exception e) {
        log.error("fileUpload()--->Error: " + e.getMessage(), e);
    }
}

private InitSignDocumentAtBrowserRQST getSignDocumentAtBrowserRQST(
    String pathToCert) throws IOException {
    log.debug("getSignDocumentAtBrowserRQST()---> Init");
    File tempDir = NIOUtils.createTempDirectoryServer(SUBDIRECTORY_TEMP);
    File upload = NIOUtils.writeDataToTempFile(this.getFile(), tempDir);
    log.debug("getSignDocumentAtBrowserRQST()---> Init-1: "
        + upload.getAbsolutePath());
    byte[] certificate = NIOUtils.resourceAsBytes(pathToCert);
    log.debug("getSignDocumentAtBrowserRQST()---> Init-2");
    return PSAResquestFactory._().createInitSignDocumentAtBrowserRQST(
        getFunctionalContextFromDemoCase(),
        Locale.getDefault().getLanguage(),
        certificate,
        upload);
}

```

Exemple de ContinueSignDocumentAtBrowser

```

private SessionAtBrowserRSPNS continueAtBrowser(String pathToCert) {
    log.debug("continueAtBrowser()---> ");
    SessionAtBrowserRSPNS response = null;
    try {
        response = this.psaService.continueSignDocumentAtBrowser(this
            .getContinueAtBrowserRQST(pathToCert));
        this.accessURL = response.getAccessUrl();
        if (response.getResult() != null) {
            this.studyResponse(response.getResult());
        }
    } catch (Exception e) {
        log.error("continueAtBrowser()---> ");
    }
    return response;
}

private ContinueSignDocumentAtBrowserRQST getContinueAtBrowserRQST(
    String pathToCert) {
    byte[] certificate = NIOUtils.resourceAsBytes(pathToCert);
    return PSAResquestFactory._().createContinueSignDocumentAtBrowserRQST(
        this.procedureContextOID, "ES", certificate);
}

```

6.3.1.2 Client Java ad hoc WSDL2Java

Per el client **ad hoc from WSDL2Java**, la feina s'incrementa, per que tot allò que realitza les classes d'utilitat proporcionades per l'API, haurà de ser implementada al client. El codi és molt extens. A continuació un exemple per demanar al **SI PSA** la creació d'un context de signatura amb certificat al **SI PSA**:

Exemple de creació de context signatura amb certificat en possessió de l'usuari

```
public static final String FILENAME_PDF_1 = "Documentol.pdf";
public static final String PROOF_OF_ORIGIN_SEMANTICS = "Proof of origin
indicates that the signer recognizes to have created, approved and sent the
message";
public static final String PROOF_OF_ORIGIN_FIELD_OF_APP = "Proof of origin";
public static final String PROOF_OF_ORIGIN_ID =
"urn:oid:1.2.840.113549.1.9.9.16.6.1";
public static final String PROOF_OF_ORIGIN_DESCRIPTION = "Proof of
origin";

public void signSingleDocumentAtPSA(final String locale, final String ppOID,
final String fileName_cert, final String fileName_PDF){
    try{
        SignDocumentAtPSARSPNS response =
            this.getPort().initSignDocumentAtPSA(
                this.getInitAtPSARQSTSingleDocument(
                    locale,
                    ppOID,
                    fileName_cert,
                    fileName_PDF));

        String procedureContextOID =
            response.getProcedureContextOID();
    } catch (Exception e) {
        log.error("signSingleDocumentAtPSA()---> " +
e.getMessage(), e);
    }
}

private InitSignDocumentAtPSARQST getInitAtPSARQSTSingleDocument(final String
locale, final String ppOID, final String fileName_cert, final String
fileName_PDF) {
    byte[] certificate = NIOUtils.resourceAsBytes(certificateToKeySelector);
    X509Certificate cert = X509Utils.getX509CertificateFromBytes(certificate);

    List<DocumentMetaData<byte[]>> docs = new
ArrayList<DocumentMetaData<byte[]>>();
    DocumentMetaData<byte[]> doc = null;
    int i = 0;

    for (byte[] temp : this.files){
```

```

        doc = new DocumentMetaData<byte[]>();
        doc.setData(temp);
        doc.setFilename(this.fileNames.get(i));
        byte[] temp2 = this.files.get(0);
        docs.add(doc);
        i++;
    }
    this.createInitSignDocumentAtPSARQSTArrayBytes(
        "PP-0012",
        "ES",
        null,
        X509Utils.getIssuer(cert),
        cert.getSerialNumber(),
        null,
        commitmentsSemantics,
        commitmentsFieldOfApplication,
        commitmentIdentifier,
        QualifierType.OID_AS_URN,
        docs,
        false,
        true,
        false);

    return request;
}

private InitSignDocumentAtPSARQST createInitSignDocumentAtPSARQSTArrayBytes(
    String functionalProcedureId, String locale, String role,
    String issuerName, BigInteger serialNumber,
    String oidTypeDescription, String commitmentsSemantics,
    String commitmentsFieldOfApplication, String
    identifierTypeValue,
    QualifierType qualifierType,
    List<DocumentMetaData<byte[]>> documents, boolean
    returnReport,
    boolean returnSignedDocumentId, boolean returnSignedDocument)
{
    InitSignDocumentAtPSARQST initSignDocumentAtPSARQST =
    buildCommonInitSignDocumentAtPSARQST(
        functionalProcedureId, locale, role, issuerName,
        serialNumber,
        oidTypeDescription, commitmentsSemantics,
        commitmentsFieldOfApplication, identifierTypeValue,
        qualifierType, returnReport, returnSignedDocumentId,
        returnSignedDocument);

    //
    try {
        initSignDocumentAtPSARQST
        .setInputDocuments(getInputDocumentsFromByteArray(documents));
    } catch (IOException e) {
        // TODO
        log.error(e.getMessage(), e);
    }
}

```

```

        return initSignDocumentAtPSARQST;
    }

    private InitSignDocumentAtPSARQST buildCommonInitSignDocumentAtPSARQST(
        String functionalProcedureId, String locale, String role,
        String issuerName, BigInteger serialNumber,
        String oidTypeDescription, String commitmentsSemantics,
        String commitmentsFieldOfApplication, String
        identifierTypeValue,
        QualifierType qualifierType, boolean returnReport,
        boolean returnSignedDocumentId, boolean returnSignedDocument)
    {
        //
        IssuerAndSerialType issuerAndSerialType = new
        IssuerAndSerialType();
        issuerAndSerialType.setIssuerName(issuerName);
        issuerAndSerialType.setSerialNumber(serialNumber);

        KeySelectorType keySelectorType = new KeySelectorType();
        keySelectorType.setIssuerAndSerial(issuerAndSerialType);

        ObjectIdentifierType objectIdentifierType =
        buildObjectIdentifierType(
            qualifierType, identifierTypeValue,
            oidTypeDescription);

        CommitmentType commitmentType = buildCommitmentType(
            commitmentsFieldOfApplication, commitmentsSemantics,
            objectIdentifierType);

        InitSignDocumentAtPSARQST initSignDocumentAtPSARQST = new
        InitSignDocumentAtPSARQST();
        initSignDocumentAtPSARQST
            .setFunctionalProcedureId(functionalProcedureId);
        initSignDocumentAtPSARQST.setLocale(locale);
        initSignDocumentAtPSARQST.setReturnReport(returnReport);

        initSignDocumentAtPSARQST.setReturnSignedDocument(returnSignedDocument);
        initSignDocumentAtPSARQST
            .setReturnSignedDocumentId(returnSignedDocumentId);
        initSignDocumentAtPSARQST.setRole(role);
        initSignDocumentAtPSARQST.setCommitment(commitmentType);
        initSignDocumentAtPSARQST.setKeySelector(keySelectorType);
        return initSignDocumentAtPSARQST;
    }

```

6.3.2 Client Java 1.4

Com hem avançat en el punt [Client Java](#) la construcció del client Java pot utilitzar una sèrie de classes que facilitem en una biblioteca Java, sobre tot per la construcció de les peticions.

Aquestes classes seran accessibles com una llibreria externa, instal·lades a un repositori privat corporatiu de **maven2**.

6.3.2.1 Client Java utilitzant l'API

Primer de tot necessitem carregar el mòdul de seguretat Rampart i iniciar el seu context per després poder fer invocacions mitjançant les classes de stub. El següent exemple mostra com fer-ho pel cas específic del WS del **SI PSA** de Query:

Exemple de carrega del mòdul Rampart i creació d'un stub pels serveis web de Query

```
private void init() {
    ConfigurationContext ctx = null;

    ctx =
ConfigurationContextFactory.createConfigurationContextFromFileSystem
("src/main/resources", null);

    QueryServiceStub stub = new QueryServiceStub(ctx);
    ServiceClient client = stub._getServiceClient();

    log.info("Engaging RAMPART From code");
    Policy rampartConfig = getRampartConfig();

    client.getAxisService().getPolicySubject().attachPolicy(rampartConfig);
    clientengageModule("rampart");
}
```

Observem que es fa una crida al mètode static **getRampartConfig()**. Aquest mètode és el que configura Rampart i li especifica el Keystore que haurà de fer servir per la encriptació/descriptació i signatura de les peticions WS. El seu codi és el següent:

Codi de configuració de Rampart

```
protected static Policy getRampartConfig() {
    log.debug("getRampartConfig()-->Init");
    RampartConfig rampartConfig = new RampartConfig();
    rampartConfig.setUserCertAlias(clientService.getSignerAlias());
    rampartConfig.setEncryptionUser(clientService.getEncryptAlias());
    rampartConfig.setPwCbClass(PWCBHandler.class.getName());

    CryptoConfig sigCrypto = new CryptoConfig();
    sigCrypto.setProvider(MagicNames.MERLIN_PROVIDER);

    Properties props = new Properties();
    props.setProperty(MagicNames.MERLIN_PROVIDER_KEYSTORE_TYPE,
```

```

clientService.getKeystoreType();
    props.setProperty(MagicNames.MERLIN_PROVIDER_KEYSTORE_FILE,
clientService.getKeystoreFile());
    props.setProperty(MagicNames.MERLIN_PROVIDER_KEYSTORE_PASSWORD,
clientService.getKeystorePassword());

    sigCrypto.setProp(props);

    rampartConfig.setSigCryptoConfig(sigCrypto);
    rampartConfig.setDecCryptoConfig(sigCrypto);
    rampartConfig.setEncrCryptoConfig(sigCrypto);
    Policy policy = new Policy();
    policy.addAssertion(rampartConfig);
    return policy;
}

```

Observem que recupera tots els valors de configuració del **keystore** d'un objecte anomenat **clientService**. Aquest objecte és de tipus **com.sadiel.catcert.psa.ws.client.PSAClientService** que no és més que un Bean que guarda els valors que li introduïm (mitjançant els mètodes 'setX()') amb la particularitat que, en cas de no haver estat fet el 'set' explícitament, recupera el valor d'un **fitxer de propietats**.

Per tant, hi ha **2 maneres de configurar el client** si fem servir la API, o bé a través del "set" de cadascun dels valors (amb Spring per exemple) o bé a través d'un fitxer de propietats.

El fitxer de propietats es troba dins del **jar** a la ruta **/src/main/resources/META-INF/config.properties**. Si optem per aquest mètode de configuració haurem de modificar aquest fitxer. Això es pot fer amb qualsevol eina de compressió com **WinZip**.

El contingut de **confi.properties** és el següent:

```

crypto.provider=org.apache.ws.security.components.crypto.Merlin
keystore.type=PKCS12
keystore.password=1111
keystore.file=src/test/resources/CDA\_Example\_mod.p12
keystore.signer.alias=elujan
keystore.encrypt.alias=psaCert

```

Nota Important: Observem que hi ha un sol **keystore/truststore** al contrari que als clients Java 5/6 o .NET que tenien 2 fitxers independents que corresponien al certificat del servidor PSA i al certificat del client. Dins d'aquest trustore haurà d'haver com a mínim 2 alias. El **keystore.signer.alias** apuntarà al certificat + clau de l'aplicació client. El **keystore.encrypt.alias** apuntarà al certificat públic del servidor PSA.

Un cop sabem que el clientService serà capaç de recuperar els valors del keystore passem al exemples concrets de invocacions a WS de PSA mitjançant les classes de **Stub**:

Exemple de petició a WS AqueryAuthorizationToSigner

```
log.debug("testQueryAuthorizationToSigner()--->Init");
QueryAuthorizationToSignerDocument queryAuthorizationDoc =
PSARequestFactory._().createQueryAuthorizationToSigner(LOCALE, AUTH_NIF);

QueryAuthorizationToSignerResponseDocument response =
stub.queryAuthorizationToSigner(queryAuthorizationDoc);

SessionAtBrowserRSPNS rspns =
response.getQueryAuthorizationToSignerResponse().getSessionAtBrowserRSPNS();

log.info("testQueryAuthorizationToSigner()--->ResultMajor: " +
rspns.getResult().getResultMajor());
log.info("testQueryAuthorizationToSigner()--->ResultMinor: " +
rspns.getResult().getResultMinor());
log.info("testQueryAuthorizationToSigner()--->La URL d'accés és:
" + rspns.getAccessUrl());
```

6.3.3 Client .NET (VB)

Exemple de creació de context de signatura amb certificat en possessió de l'usuari

```
Dim objService As New ClientWS.clientVirtualService

Dim objRequest As New ClientWS.InitSignDocumentAtBrowserRQST
Dim objResponse As New ClientWS.SignDocumentAtBrowserRSPNS

Try
    'Indica que el response disposa de codificació MTOM
    objService.RequireMtom = True

    'Especifiquem la política de seguretat a utilitzar
    objService.SetPolicy("PSAClientPolicy")

    'Certificat afegit con a X509Token, el de l'aplicació de gestió
    Dim X509TokenClient As New X509Certificate2("C:/CDA_APP1.p12", "1111")

    'Certificat del servidor PSA per a xifrar la request
    Dim X509CertServer As New X509Certificate2("C:/ CDA_PSA.cer")

    'Afegim els certificats
    objService.SetClientCredential(New X509SecurityToken(X509TokenClient))
    objService.SetServiceCredential(New X509SecurityToken(X509CertServer))
```

```
'Establim el ProcedureContext
objRequest.FunctionalProcedureId = "PP-0001"

'Establim el locale
objRequest.Locale = "es"

'Establim el certificat del signant
Dim X509SignerCert As New X509Certificate2("C:/signer.cer")
objRequest.KeySelector.PublicKey = X509SignerCert.GetRawCertData

'Establim el fitxer a signar
Dim objDocument As New ClientWS.DocumentType
objDocument.filename = "Ejemplo.xml"

objDocument.data = ReadBinaryFile("C/Ejemplo.xml")

Dim docs() As ClientWS.DocumentType
ReDim docs(0)
docs(0) = objDocument

objRequest.InputDocuments = obj

'Invocació del servei
objResponse = objService.initSignDocumentAtBrowser(objRequest)

Catch ex As Exception
    Console.WriteLine(ex.Message)
End Try

MsgBox(objResponse.AccessUrl & " " & objResponse.ProcedureContextOID)
```

Exemple de continuació de context de signatura amb certificat en possessió de l'usuari

```
Dim objService As New ClientWS.clientVirtualService

Dim objRequest As New ClientWS.ContinueSignDocumentAtBrowserRQST
Dim objResponse As New ClientWS.SessionAtBrowserRSPNS

Try

    'Indica que el response disposa de codificació MTOM
    objService.RequireMtom = True

    'Especifiquem la política de seguretat a utilitzar
    objService.SetPolicy("PSAClientPolicy")

    'Certificat afegit con a X509Token, el de l'aplicació de gestió
    Dim X509TokenClient As New X509Certificate2("C:/CDA_APP1.p12", "1111")

    'Certificat del servidor PSA per a xifrar la request
    Dim X509CertServer As New X509Certificate2("C:/CDA_PSA.cer")

    'Afegim els certificats
    objService.SetClientCredential(New X509SecurityToken(X509TokenClient))
    objService.SetServiceCredential(New X509SecurityToken(X509CertServer))

    'Establim el ProcedureContext ja iniciat
    objRequest.ProcedureContextOID = "47"
```

```
'Establim el locale
objRequest.Locale = "es"

'Establim el certificat del següent signant
Dim X509SignerCert As New X509Certificate2("C:/signer2.cer")
objRequest.KeySelector.PublicKey = X509SignerCert.GetRawCertData

'Invocació del servei
objResponse = objService.continueSignDocumentAtBrowser(objRequest)

Catch ex As Exception
    Console.Write(ex.Message)
End Try

MsgBox(objResponse.AccessUrl & " " & objResponse.ProcedureContextOID)
```

Exemple de creació de signatura amb certificat al PSA

```
Dim objService As New ClientWS.signatureService

Dim objRequests As New ClientWS.MultiInitSignDocumentAtPSARQST
Dim objRequest As New ClientWS.InitSignDocumentAtPSARQST
Dim objResponses As ClientWS.MultiSignDocumentAtPSARSPNS
Try
    'Indica que el response disposa de codificació MTOM
    objService.RequireMtom = True

    'Especifiquem la política de seguretat a utilitzar
    objService.SetPolicy("PSAClientPolicy")

    'Certificat afegit con a X509Token, el de l'aplicació de gestió
    Dim X509TokenClient As New X509Certificate2("C:/CDA_APP1.p12", "1111")

    'Certificat del servidor PSA per a xifrar la request
    Dim X509CertServer As New X509Certificate2("C:/CDA_PSA.cer")

    'Afegim els certificats
    objService.SetClientCredential(New X509SecurityToken(X509TokenClient))
    objService.SetServiceCredential(New X509SecurityToken(X509CertServer))

    'Creem la petició single
    Dim req() As ClientWS.InitSignDocumentAtPSARQST
    ReDim req(0)
    req(0) = objRequest

    objRequests.InitSignDocumentAtPSARQST = req

    'Establim el ProcedureContext
    objRequest.FunctionalProcedureId = "PP-0004"

    'Establim el locale
    objRequests.Locale = "es"

    'Establim el fitxer a signar
    Dim objDocument As New ClientWS.DocumentType
    objDocument.filename = "Informe.pdf"
```

```

objDocument.data = ReadBinaryFile("C:/Informe.pdf")

Dim docs() As ClientWS.DocumentType
ReDim docs(0)
docs(0) = objDocument

objRequest.InputDocuments = docs

'Establim el certificat del signant (Keyselector)
Dim X509SignerCert As New X509Certificate2("C:/Signer.cer")

objRequests.KeySelector = New ClientWS.KeySelectorType
objRequests.KeySelector.PublicKey = X509SignerCert.RawData

'Establim el commitment
objRequest.Commitment = New ClientWS.CommitmentType
objRequest.Commitment.CommitmentIdentifier = New
    ClientWS.ObjectIdentifierType
objRequest.Commitment.CommitmentIdentifier.Identifier = New
    ClientWS.IdentifierType

objRequest.Commitment.CommitmentIdentifier.Identifier.Qualifier =
    ClientWS.QualifierType.OIDAsURN
objRequest.Commitment.CommitmentIdentifier.Identifier.Value =
    "urn:oid:1.2.840.113549.1.9.9.16.6.1"
objRequest.Commitment.FieldOfApplication = "Proof of origin"

'Especifiquem que volem obtenir els ID's
objRequest.ReturnSignedDocumentId = True

'Invocació del servei
objResponses = objService.initSignDocumentAtPSA(objRequests)

Catch ex As Exception
    Console.Write(ex.Message)
End Try

Dim objResponse As New ClientWS.SignDocumentAtPSARSPNS
objResponse = objResponses.SignDocumentAtPSARSPNS(0)

MsgBox(objResponse.Result.ResultMessage.Value & " " &
objResponse.SignedDocumentIds(0))

```

Exemple de creació de signatura visible en PDF amb certificat al PSA

```

Dim objService As New ClientWS.signatureService

Dim objRequests As New ClientWS.MultiInitSignDocumentAtPSARQST
Dim objRequest As New ClientWS.InitSignDocumentAtPSARQST
Dim objResponses As New ClientWS.MultiSignDocumentAtPSARSPNS
Try
    'Indica que el response disposa de codificació MTOM
    objService.RequireMtom = True

    'Especifiquem la política de seguretat a utilitzar
    objService.SetPolicy("PSAClientPolicy")

```

```
' Certificat afegit con a X509Token, el de l'aplicació de gestió
Dim X509TokenClient As New X509Certificate2("C:/CDA_APP1.p12", "1111")

' Certificat del servidor PSA per a xifrar la request
Dim X509CertServer As New X509Certificate2("C:/CDA_PSA.cer")

' Afegim els certificats
objService.SetClientCredential(New X509SecurityToken(X509TokenClient))
objService.SetServiceCredential(New X509SecurityToken(X509CertServer))

' Creem la petició single
Dim req() As ClientWS.InitSignDocumentAtPSARQST
ReDim req(0)
req(0) = objRequest

objRequests.InitSignDocumentAtPSARQST = req

'Establim el ProcedureContext
objRequest.FunctionalProcedureId = "PP-0004"

'Establim el locale
objRequests.Locale = "es"

'Establim el fitxer a signar
Dim objDocument As New ClientWS.DocumentType
objDocument.filename = "Informe.pdf"

objDocument.data = ReadBinaryFile("C:/Informe.pdf")

Dim docs() As ClientWS.DocumentType
ReDim docs(0)
docs(0) = objDocument

objRequest.InputDocuments = docs

'Establim el certificat del signant (Keyselector)
Dim X509SignerCert As New X509Certificate2("C:/Signer.cer")

objRequests.KeySelector = New ClientWS.KeySelectorType
objRequests.KeySelector.PublicKey = X509SignerCert.RawData

'Establim el commitment
objRequest.Commitment = New ClientWS.CommitmentType
objRequest.Commitment.CommitmentIdentifier = New
    ClientWS.ObjectIdentifierType
objRequest.Commitment.CommitmentIdentifier.Identifier = New
    ClientWS.IdentifierType

objRequest.Commitment.CommitmentIdentifier.Identifier.Qualifier =
    ClientWS.QualifierType.OIDAsURN
objRequest.Commitment.CommitmentIdentifier.Identifier.Value =
    "urn:oid:1.2.840.113549.1.9.9.16.6.1"
objRequest.Commitment.FieldOfApplication = "Proof of origin"

'Especifiquem que volem obtenir els ID's
objRequest.ReturnSignedDocumentId = True

objRequest.ShowVisibleSignature = True
objRequest.VisibleSignatureData = New ClientWS.VisibleSignatureDataType

Dim Infotext() As ClientWS.CertificateDataType
```

```

ReDim Infotext(1)

Infotext(0) = ClientWS.CertificateDataType.CommonName
Infotext(1) = ClientWS.CertificateDataType.SerialNumber

objRequest.VisibleSignatureData.InfoText = Infotext

objRequest.VisibleSignatureData.VisibleSignaturePosition = New
ClientWS.VisibleSignaturePosition

objRequest.VisibleSignatureData.VisibleSignaturePosition.RectangleInfo = New
ClientWS.RectangleInfoType

objRequest.VisibleSignatureData.VisibleSignaturePosition.RectangleInfo.LowerLeftX
= 100

objRequest.VisibleSignatureData.VisibleSignaturePosition.RectangleInfo.LowerLeftY
= 100

objRequest.VisibleSignatureData.VisibleSignaturePosition.RectangleInfo.UpperRightX
= 200

objRequest.VisibleSignatureData.VisibleSignaturePosition.RectangleInfo.UpperRightY
= 400

objRequest.VisibleSignatureData.VisibleSignaturePosition.RectangleInfo.PageNumber
= 1

'Invocació del servei
objResponses = objService.initSignDocumentAtPSA(objRequests)

Catch ex As Exception
    Console.WriteLine(ex.Message)
End Try

Dim objResponse As New ClientWS.SignDocumentAtPSARSPNS
objResponse = objResponses.SignDocumentAtPSARSPNS(0)

MsgBox(objResponse.Result.ResultMessage.Value & " " &
objResponse.SignedDocumentIds(0))

```

Exemple de continuació de signatura amb certificat al PSA

```

Dim objService As New ClientWS.signatureService

Dim objRequests As New ClientWS.MultiContinueSignDocumentAtPSARQST
Dim objRequest As New ClientWS.ContinueSignDocumentAtPSARQST
Dim objResponses As ClientWS.MultiSignDocumentAtPSARSPNS

Try
    'Indica que el response disposa de codificació MTOM
    objService.RequireMtom = True

    'Especifiquem la política de seguretat a utilitzar
    objService.SetPolicy("PSAClientPolicy")

    'Certificat afegit con a X509Token, el de l'aplicació de gestió
    Dim X509TokenClient As New X509Certificate2("C:/CDA_APP1.p12", "1111")

```

```
' Certificat del servidor PSA per a xifrar la request
Dim X509CertServer As New X509Certificate2("C:/CDA_PSA.cer")

' Afegim els certificats
objService.SetClientCredential(New X509SecurityToken(X509TokenClient))
objService.SetServiceCredential(New X509SecurityToken(X509CertServer))

' Creem la petició single
Dim req() As ClientWS.ContinueSignDocumentAtPSARQST
ReDim req(0)
req(0) = objRequest

objRequests.ContinueSignDocumentAtPSARQST = req

'Establim el ProcedureContext
objRequest.FunctionalProcedureId = "PP-0004"

'Establim el locale
objRequests.Locale = "es"

'Establim el certificat del signant (Keyselector)
Dim X509SignerCert As New X509Certificate2("C:/Signer.cer")

objRequests.KeySelector = New ClientWS.KeySelectorType
objRequests.KeySelector.PublicKey = X509SignerCert.RawData

'Establim el commitment
objRequest.Commitment = New ClientWS.CommitmentType
objRequest.Commitment.CommitmentIdentifier = New
    ClientWS.ObjectIdentifierType
objRequest.Commitment.CommitmentIdentifier.Identifier = New
    ClientWS.IdentifierType

objRequest.Commitment.CommitmentIdentifier.Identifier.Qualifier =
    ClientWS.QualifierType.OIDAsURN
objRequest.Commitment.CommitmentIdentifier.Identifier.Value =
    "urn:oid:1.2.840.113549.1.9.9.16.6.1"
objRequest.Commitment.FieldOfApplication = "Proof of origin"

'Especifiquem que volem obtenir els ID's
objRequest.ReturnSignedDocumentId = True

'Invocació del servei
objResponses = objService.initSignDocumentAtPSA(objRequests)

Catch ex As Exception
    Console.WriteLine(ex.Message)
End Try

Dim objResponse As New ClientWS.SignDocumentAtPSARSPNS
objResponse = objResponses.SignDocumentAtPSARSPNS(0)

MsgBox(objResponse.Result.ResultMessage.Value & " " &
objResponse.SignedDocumentIds(0))
```

Exemple d'inici de signatura amb Signature Manager

```
'Inici de context de N signatures amb EPF
Dim objService As New ClientWS.signatureService
```

```

Dim objRequests As New ClientWS.MultiInitSignatureAtSignatureManagerRQST

Dim objResponses As ClientWS.MultiSignatureAtSignatureManagerRSPNS

Try
    'Indica que el response disposa de codificació MTOM
    objService.RequireMtom = True

    ' Especifiquem la política de seguretat a utilitzar
    objService.SetPolicy("PSAClientPolicy")

    'Certificat afegit con a X509Token, el de l'aplicació de gestió
    Dim X509TokenClient As New X509Certificate2("C:/CDA_APP1.p12", "1111")

    ' Certificat del servidor PSA per a xifrar la request
    Dim X509CertServer As New X509Certificate2("C:/CDA_PSA.cer")

    ' Afegim els certificats
    objService.SetClientCredential(New
X509SecurityToken(X509TokenClient))
    objService.SetServiceCredential(New
X509SecurityToken(X509CertServer))

    ' Establim el locale
    objRequests.Locale = "es"

    'Establim el certificat del signant
    Dim X509SignerCert As New X509Certificate2("C:/Signer.cer")

    objRequests.Certificate = X509SignerCert.GetRawCertData

    ' Creem la petició
    Dim objRequest As New ClientWS.InitSignatureAtSignatureManagerRQST

    ' Establim els documents
    Dim objDocument As New ClientWS.DocumentType

    objDocument.filename = "Document1.pdf"

    objDocument.data = ReadBinaryFile("C:/Document1.pdf")

    Dim docs() As ClientWS.DocumentType
    ReDim docs(1)
    objRequest.InputDocuments = docs

    objRequest.InputDocuments(0) = objDocument

    objDocument = New ClientWS.DocumentType

    objDocument.filename = "Document2.pdf"

    objDocument.data = ReadBinaryFile("C:/Document2.pdf")

    objRequest.InputDocuments(1) = objDocument

```



```
'Establim el commitment
objRequest.Commitment = New ClientWS.CommitmentType
objRequest.Commitment.CommitmentIdentifier = New
ClientWS.ObjectIdentifierType
objRequest.Commitment.CommitmentIdentifier.Identifier = New
ClientWS.IdentifierType
objRequest.Commitment.CommitmentIdentifier.Identifier.Qualifier =
ClientWS.QualifierType.OIDAsURN
objRequest.Commitment.CommitmentIdentifier.Identifier.Value =
"urn:catcert:compromisos:signatura:aprovo"
objRequest.Commitment.FieldOfApplication = "Aprovació genèrica"

'Establim el ProcedureID
objRequest.FunctionalProcedureId = "PP-0004"

'Establim la petició
Dim reqs() As ClientWS.InitSignatureAtSignatureManagerRQST
ReDim reqs(1)
objRequests.InitSignatureAtSignatureManagerRQST = reqs

objRequests.InitSignatureAtSignatureManagerRQST(0) = objRequest

'Invocació del servei
objResponses =
objService.initSignatureAtSignatureManager(objRequests)

Catch ex As Exception
    Console.Write(ex.Message)
End Try

Dim objResponse As New ClientWS.SignatureAtSignatureManagerRSPNS
objResponse = objResponses.SignatureAtSignatureManagerRSPNS(0)

MsgBox(objResponse.Result.ResultMessage.Value)

Dim encoder As New System.Text.ASCIIEncoding

' Hi ha que convertir a Base64 ja que el client realitza la
decodificació de B64

Console.Write(Convert.ToBase64String(objResponse.HashesRSPNS.Hash(0)))
Console.Write(Convert.ToBase64String(objResponse.HashesRSPNS.Hash(1)))
```

Exemple d'inici de completat amb Signature Manager

```
'Finalització de context de N signatures amb EPF
Dim objService As New ClientWS.signatureService

Dim objRequests As New ClientWS.MultiCompleteSignatureRQST
Dim objResponses As ClientWS.MultiCompleteSignatureRSPNS

Try

'Indica que el response disposa de codificació MTOM
```

```
objService.RequireMtom = True

' Especifiquem la política de seguretat a utilitzar
objService.SetPolicy("PSAClientPolicy")

'Certificat afegit con a X509Token, el de l'aplicació de gestió
Dim X509TokenClient As New X509Certificate2("C:/CDA_APP1.p12", "1111")

' Certificat del servidor PSA per a xifrar la request
Dim X509CertServer As New X509Certificate2("C:/CDA_PSA.cer")

' Afegim els certificats
objService.SetClientCredential(New
X509SecurityToken(X509TokenClient))
objService.SetServiceCredential(New
X509SecurityToken(X509CertServer))

Dim req As New ClientWS.CompleteSignatureRQST

'Establim el signature OID
req.SignatureActOID = "1"

' Establim els hashos xifrats
Dim enc As ClientWS.OriginalAndEncryptedHashType()
ReDim enc(1)
req.OriginalAndEncryptedHash = enc

    Hi ha que associar el hashos i la signatura en clar ja que el
    client realitza la conversió a B64
    req.OriginalAndEncryptedHash(0) = New
ClientWS.OriginalAndEncryptedHashType
    req.OriginalAndEncryptedHash(0).Hash =
Convert.FromBase64String("MHD...TR")
    req.OriginalAndEncryptedHash(0).EncryptedHash =
Convert.FromBase64String("MCR...V3")

    req.OriginalAndEncryptedHash(1) = New
ClientWS.OriginalAndEncryptedHashType
    req.OriginalAndEncryptedHash(1).Hash =
Convert.FromBase64String("MHD...TR")
    req.OriginalAndEncryptedHash(1).EncryptedHash =
Convert.FromBase64String("MHD...TR")

'Establim la devolució de les signatures
req.ReturnSignedDocument = True

'Establim la devolució dels identificadors
req.ReturnSignedDocumentId = True

Dim reqs() As ClientWS.CompleteSignatureRQST
ReDim reqs(1)
objRequests.CompleteSignatureRQST = reqs

objRequests.CompleteSignatureRQST(0) = req

'Invocació del servei
```

```
objResponses = objService.completeSignature(objRequests)

Catch ex As Exception
    Console.WriteLine(ex.Message)
End Try

Dim objResponse As New ClientWS.CompleteSignatureRSPNS
objResponse = objResponses.CompleteSignatureRSPNS(0)

MsgBox(objResponse.Result.ResultMessage.Value)

Dim objDoc As New ClientWS.DocumentType
objDoc = objResponse.OutputDocuments.Items(0)

WriteBinaryFile("C:/DocSigned1.pdf", objDoc.data)

objDoc = New ClientWS.DocumentType
objDoc = objResponse.OutputDocuments.Items(0)

WriteBinaryFile("C:/DocSigned2.pdf", objDoc.data)
```

Exemple de xifrat de document

```
Dim objService As New UtilWS.utilService

Dim objRequest As New UtilWS.CipherDocumentRQST
Dim objResponse As New UtilWS.CipherDocumentRSPNS

Try

    ' Especificuem la política de seguretat a utilitzar
    objService.SetPolicy("PSAClientPolicy")

    ' Certificat afegit con a X509Token, el de l'aplicació de gestió
    Dim X509TokenClient As New X509Certificate2("C:/CDA_APP1.p12", "1111")

    ' Certificat del servidor PSA per a xifrar la request
    Dim X509CertServer As New X509Certificate2("C:/CDA_PSA.cer")

    ' Afegim els certificats
    objService.SetClientCredential(New X509SecurityToken(X509TokenClient))
    objService.SetServiceCredential(New X509SecurityToken(X509CertServer))

    ' Establim el document a xifrar
    Dim objDocument As New UtilWS.DocumentType
    objDocument.filename = "Informe.pdf"

    objDocument.data = ReadBinaryFile("C:/Informe.pdf")
    objRequest.Document = objDocument

    ' Establim el locale
    objRequest.Locale = "es"

    ' Establim el tipus de xifrat
    objRequest.EncryptionType = UtilWS.EncryptionType.SMIME

    ' Establim els certificats dels destinataris
    Dim X509CipherCert As New X509Certificate2("C:/Cipher.cer")
```

```
Dim objCerts As Byte()()
ReDim objCerts(0)
objCerts(0) = X509CipherCert.RawData

objRequest.ReceiverCertificationList = objCerts

'Invocació del servei
objResponse = objService.cipherDocument(objRequest)

Catch ex As Exception
    Console.WriteLine(ex.Message)
End Try

MsgBox(objResponse.Result.ResultMessage)
Console.WriteLine(objResponse.Document.data)
```

Exemple de desxifrat de document

```
Dim objService As New UtilWS.utilService

Dim objRequest As New UtilWS.DecipherDocumentRQST
Dim objResponse As New UtilWS.DecipherDocumentRSPNS

Try

    'Especifiquem la política de seguretat a utilitzar
    objService.SetPolicy("PSAClientPolicy")

    'Certificat afegit con a X509Token, el de l'aplicació de gestió
    Dim X509TokenClient As New X509Certificate2("C:/CDA_APP1.p12", "1111")

    'Certificat del servidor PSA per a xifrar la request
    Dim X509CertServer As New X509Certificate2("C:/CDA_PSA.cer")

    'Afegim els certificats
    objService.SetClientCredential(New X509SecurityToken(X509TokenClient))
    objService.SetServiceCredential(New X509SecurityToken(X509CertServer))

    'Establim el document a xifrar
    Dim objDocument As New UtilWS.DocumentType
    objDocument.filename = "DocumentXifrat.txt"

    objDocument.data = ReadBinaryFile("C:/DocumentXifrat.txt")
    objRequest.Document = objDocument

    'Establim el locale
    objRequest.Locale = "es"

    'Establim el tipus de xifrat
    objRequest.EncryptionType = UtilWS.EncryptionType.SMIME

    'Establim el KeySelector per al desxifrat
    Dim X509DecipherCert As New X509Certificate2("C:/Decipher.cer")

    objRequest.KeySelector = New UtilWS.KeySelectorType
    objRequest.KeySelector.PublicKey = X509DecipherCert.RawData
```

```
'Invocació del servei
objResponse = objService.decipherDocument(objRequest)

Catch ex As Exception
    Console.WriteLine(ex.Message)
End Try

MsgBox(objResponse.Result.ResultMessage)
Console.WriteLine(objResponse.Document.data)
```

Exemple d'eliminació de signatura

```
Dim objService As New UtilWS.utilService

Dim objRequest As New UtilWS.DeleteSignatureRQST
Dim objResponse As New UtilWS.DeleteSignatureRSPNS

Try

    'Indica que el response disposa de codificació MTOM
    objService.RequireMtom = True

    'Especifiquem la política de seguretat a utilitzar
    objService.SetPolicy("PSAClientPolicy")

    'Certificat afegit con a X509Token, el de l'aplicació de gestió
    Dim X509TokenClient As New X509Certificate2("C:/CDA_APP1.p12", "1111")

    'Certificat del servidor PSA per a xifrar la request
    Dim X509CertServer As New X509Certificate2("C:/CDA_PSA.cer")

    'Afegim els certificats
    objService.SetClientCredential(New X509SecurityToken(X509TokenClient))
    objService.SetServiceCredential(New X509SecurityToken(X509CertServer))

    'Establim el locale
    objRequest.Locale = "es"

    'Establim l'identificador de la signatura a eliminar
    objRequest.SignedDocumentOID = "046ef040-75fa-4da2-9e4c-bd1c85351b2a"

    'Invocació del servei
    objResponse = objService.deleteSignature(objRequest)

Catch ex As Exception
    Console.WriteLine(ex.Message)
End Try

MsgBox(objResponse.Result.ResultMessage)
```

Exemple de descàrrega de signatura al PSA

```
Dim objService As New QueryWS.queryService

Dim objRequest As New QueryWS.DownloadSignatureAtPSARQST
Dim objResponse As New QueryWS.DownloadSignatureAtPSARSPNS
```

Try

```
'Indica que el response disposa de codificació MTOM
objService.RequireMtom = True

' Especifiquem la política de seguretat a utilitzar
objService.SetPolicy("PSAClientPolicy")

' Certificat afegit con a X509Token, el de l'aplicació de gestió
Dim X509TokenClient As New X509Certificate2("C:/CDA_APP1.p12", "1111")

' Certificat del servidor PSA per a xifrar la request
Dim X509CertServer As New X509Certificate2("C:/CDA_PSA.cer")

' Afegim els certificats
objService.SetClientCredential(New X509SecurityToken(X509TokenClient))
objService.SetServiceCredential(New X509SecurityToken(X509CertServer))

'Establim el locale
objRequest.Locale = "es"

'Establim els identificador del documents signats
Dim lstDocuments As String()
ReDim lstDocuments(0)
lstDocuments(0) = "cd99af4d-28b4-4809-b205-ea220e18c603"

objRequest.SignedDocumentId = lstDocuments

'Invocació del servei
objResponse = objService.downloadSignatureAtPSA(objRequest)
```

```
Catch ex As Exception
    Console.Write(ex.Message)
End Try
```

```
MsgBox(objResponse.Result.ResultMajor & " " &
objResponse.Result.ResultMessage.Value)
```

```
Dim objDoc As New QueryWS.DocumentType
objDoc = objResponse.OutputDocuments.Items(0)
```

```
WriteBinaryFile("C:/DocSignat.pdf", objDoc.data)
```

Exemple de consulta d'activitat

```
Dim objService As New QueryWS.queryService

Dim objRequest As New QueryWS.QueryActivityRQST
Dim objResponse As New QueryWS.QueryActivityRSPNS

Try

    ' Especifiquem la política de seguretat a utilitzar
    objService.SetPolicy("PSAClientPolicy")

    ' Certificat afegit con a X509Token, el de l'aplicació de gestió
    Dim X509TokenClient As New X509Certificate2("C:/CDA_APP1.p12", "1111")
```

```
' Certificat del servidor PSA per a xifrar la request
Dim X509CertServer As New X509Certificate2("C:/CDA_PSA.cer")

' Afegim els certificats
objService.SetClientCredential(New X509SecurityToken(X509TokenClient))
objService.SetServiceCredential(New X509SecurityToken(X509CertServer))

'Establim els criteris de cerca
objRequest.AuditType = QueryWS.AuditTypeType.INFO
objRequest.AuditCategory = QueryWS.AuditCategoryType.SIGN
objRequest.Nif = "00000000J"
objRequest.From = DateTime.Today
objRequest.To = DateTime.Today.AddDays(-1)

'Invocació del servei
objResponse = objService.queryActivity(objRequest)

Catch ex As Exception
    Console.WriteLine(ex.Message)
End Try

MsgBox(objResponse.Result.ResultMessage)
MsgBox(objResponse.Audits.ToString())
```

Exemple de creació de sessió per a consulta d'activitat

```
Dim objService As New QueryWS.queryService

Dim objRequest As New QueryWS.QueryActivityToSignerRQST
Dim objResponse As New QueryWS.SessionAtBrowserRSPNS

Try

    'Indica que el response disposa de codificació MTOM
    objService.RequireMtom = True

    'Especifiquem la política de seguretat a utilitzar
    objService.SetPolicy("PSAClientPolicy")

    ' Certificat afegit con a X509Token, el de l'aplicació de gestió
    Dim X509TokenClient As New X509Certificate2("C:/CDA_APP1.p12", "1111")

    ' Certificat del servidor PSA per a xifrar la request
    Dim X509CertServer As New X509Certificate2("C:/CDA_PSA.cer")

    ' Afegim els certificats
    objService.SetClientCredential(New X509SecurityToken(X509TokenClient))
    objService.SetServiceCredential(New X509SecurityToken(X509CertServer))

    'Establim el KeySelector
    Dim X509SignerCert As New X509Certificate2("C:/CDA1.cer")

    objRequest.KeySelector = New QueryWS.KeySelectorType
    objRequest.KeySelector.PublicKey = X509SignerCert.RawData

    'Invocació del servei
    objResponse = objService.queryActivityToSigner(objRequest)

Catch ex As Exception
```

```

Console.WriteLine(ex.Message)
MsgBox(ex.ToString)
End Try

MsgBox(objResponse.Result.ResultMessage.Value)
MsgBox(objResponse.AccessUrl)

```

Exemple de sol·licitud de comprovant

```

Dim objService As New QueryWS.queryService

Dim objRequest As New QueryWS.DownloadTicketSignatureRQST
Dim objResponse As New QueryWS.DownloadTicketSignatureRSPNS

Try

    'Indica que el response disposa de codificació MTOM
    objService.RequireMtom = True

    'Especifiquem la política de seguretat a utilitzar
    objService.SetPolicy("PSAClientPolicy")

    'Certificat afegit con a X509Token, el de l'aplicació de gestió
    Dim X509TokenClient As New X509Certificate2("C:/CDA_APP1.p12", "1111")

    'Certificat del servidor PSA per a xifrar la request
    Dim X509CertServer As New X509Certificate2("C:/CDA_PSA.cer")

    'Afegim els certificats
    objService.SetClientCredential(New X509SecurityToken(X509TokenClient))
    objService.SetServiceCredential(New X509SecurityToken(X509CertServer))

    'Establim el locale
    objRequest.Locale = "es"

    'Establim l'id de la signatura per a obtindre el comprovant
    objRequest.TicketSignatureId = "cd99af4d-28b4-4809-b205-ea220e18c603"

    'Invocació del servei
    objResponse = objService.downloadTicketSignature(objRequest)

Catch ex As Exception
    Console.WriteLine(ex.Message)
    MsgBox(ex.ToString)
End Try

MsgBox(objResponse.Result.ResultMessage.Value)

Dim objDoc As New QueryWS.DocumentType
objDoc = objResponse.Document

WriteBinaryFile("C:/Comprovant.pdf", objDoc.data)

```

Exemple de consulta d'acte de signatura

```

Dim objService As New QueryWS.queryService

Dim objRequest As New QueryWS.DownloadTicketSignatureRQST

```



```
Dim objResponse As New QueryWS.DownloadTicketSignatureRSPNS

Try

    'Indica que el response disposa de codificació MTOM
    objService.RequireMtom = True

    'Especifiquem la política de seguretat a utilitzar
    objService.SetPolicy("PSAClientPolicy")

    'Certificat afegit con a X509Token, el de l'aplicació de gestió
    Dim X509TokenClient As New X509Certificate2("C:/CDA_APP1.p12", "1111")

    'Certificat del servidor PSA per a xifrar la request
    Dim X509CertServer As New X509Certificate2("C:/CDA_PSA.cer")

    'Afegim els certificats
    objService.SetClientCredential(New X509SecurityToken(X509TokenClient))
    objService.SetServiceCredential(New X509SecurityToken(X509CertServer))

    'Establim el locale
    objRequest.Locale = "es"

    'Obtenim els identificadors
    objRequest.ReturnSignedDocumentIds = True

    'Obtenim el document
    objRequest.ReturnSignedDocuments = True

    'Establim l'id de l'acte a consultar
    objRequest.ActOID = "cd99af4d-28b4-4809-b205-ea220e18c603"

    'Invocació del servei
    objResponse = objService.querySignatureAct(objRequest)

Catch ex As Exception
    Console.WriteLine(ex.Message)
    MsgBox(ex.ToString)
End Try

MsgBox(objResponse.Result.ResultMessage.Value)
MsgBox(objResponse.State)
```

7. Annexes

7.1 Generació de l'API de WS del SI PSA

Per generar l'API de WS del **SI PSA** (clientws_psa.jar) es requereix:

1. El codi font del Projecte
2. Maven 2
3. Accés a una URL amb PSA desplegat

L'API pels clients de WS de PSA de Java correspon al mòdul **clientws**.

Verificarem que en el **pom.xml** de clientws, la URL dels WSDL dels WS del **SI PSA** estan accessibles. Els valors per defecte especificats en el pom.xml són els següents i hauran de modificar-se en cas que sigui necessari:

```
<properties>
  <wsdlUrlClient>
    http://localhost:8080/engineWS/client/service?wsdl
  </wsdlUrlClient>
  <wsdlUrlUtil>
    http://localhost:8080/engineWS/util/service?wsdl
  </wsdlUrlUtil>
  <wsdlUrlQuery>
    http://localhost:8080/engineWS/query/service?wsdl
  </wsdlUrlQuery>
</properties>
```

Per defecte, el client es genera per ser executat per la màquina virtual de Java 6. En cas de voler generar un client executable per Java 5 editar el POM principal del projecte ubicat a **psa-application/pom.xml** i substituir els valors de source i target de **6** (per defecte) a **1.5**.

```
<plugin>
  <artifactId>maven-compiler-plugin</artifactId>
  <configuration>
    <source>1.5</source>
    <target>1.5</target>
  </configuration>
</plugin>
```

NOTA IMPORTANT: En cas de fer servir una versió de JDK anterior a **Java 6 Update 4**, copiar les llibreries següents dins de **\$JAVA_HOME/jre/lib/endorsed**

- jaxws-api-2.1EA2.jar
- jaxb-api-2.2.jar

Finalment, per construir el JAR `clientws_psa.jar` ens situarem dins la carpeta **psa-application/clientws** i executarem la comanda:

```
mvn -P client clean install
```

El JAR es generarà a **psa-application/clientws/target/clientws_psa.jar**.

7.2 Respostes retornades pels Serveis Web

MAJOR	
urn:oasis:names:tc:dss:1.0:resultmajor:Success	El servidor ha processat correctament la petició.
urn:oasis:names:tc:dss:1.0:resultmajor:RequesterError	La petició no ha pogut ser processada per un error a la petició per part del client.
urn:oasis:names:tc:dss:1.0:resultmajor:ResponderError	La petició no ha pogut ser processada correctament per un error al servidor.

MINOR	
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:Success	El servidor ha processat correctament la petició.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:SignedHashNotFound	No s'ha enviat a PSA el hash signat.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:WrongHash	El/s hash/es proporcionats no són els calculats per PSA.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:DocumentMustBeRead	S'ha de llegir el document a signar.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:InvalidDocumentFormat	El format del document és invàlid.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:NoRoleInCertificate	És obligatori rol per a aquesta signatura, i el certificat escollit no té rol.
urn:catcert:psa:1.0:profiles:dss-	El rol escollit no coincideix amb el rol del

directsign:resultminor:RoleDoNotMatch	certificat.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:NoRightRoleInCertificate	El certificat no té el rol adequat.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:InvalidCommitment	El compromís indicat no és vàlid per la política de multesignatura indicada.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:UnsupportedBlindSignature	Signatura cega no suportada.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:InvalidProcedureContext	No existeix el context amb l'identificador especificat.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:UnfinishedPreviousSigner	No es pot començar a signar fins que el signant anterior finalitzi.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:InvalidProcedurePolicy	Política de multesignatura incorrecta.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:InvalidSignaturePolicy	Política de signatura incorrecta.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:keyselector:NoRights	La clau no hi és al sistema, o no té drets sobre aquesta.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:keyselector:NotFound	No s'ha trobat la clau al sistema.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:InvalidKeyselector	El keyselector no és vàlid.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:SignatureDoesNotExist	No s'ha trobat la signatura sol·licitada.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:delete:NoRights	L'usuari no té permisos per esborrar la signatura.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:ContextNotClosed	La signatura no pot ser eliminada del model PSA, per que el seu context de signatura no està tancat.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:download:NoRights	L'usuari no té permisos per descarregar la signatura.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:signature:act:ActDoesNotExist	L'acte de signatura no existeix.
urn:catcert:psa:1.0:profiles:dss-	L'usuari no té permisos sobre l'acte de signatura

directsign:resultminor:signature:act:NoRights	indicat.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:UnavailableExternalAuthorities	En aquests moments les autoritats externes no estan operatives.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:authorization:NoResults	No hi ha cap autorització amb el NIF especificat.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:NoRightCertificate	El certificat escollit no és el correcte per aquest procediment.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:UnauthorizedCertificate	Certificat no autoritzat.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:certificate:NotValid	El certificat no és vàlid.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:InvalidWebService	El Web Service invocat no és l'esperat per a aquest acte.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:InternalServerError	S'ha produït un error intern al servidor.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:CertificateNeeded	És necessari proveir el certificat per tal de continuar amb el procés de signatura.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:CommitmentNeeded	És necessari proveir el compromís per tal de continuar amb el procés de signatura.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:RoleNeeded	És necessari proveir el rol del signatari per tal de continuar amb el procés de signatura.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:AttributesNeeded	És necessari proveir els valors dels atributs que requereix la política de signatura.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:SignedHashNeeded	És necessari proveir el hash xifrat per tal de continuar amb el procés de signatura.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:InputHashNotAllowed	La política de signatura no admet el resum criptogràfic com a document a signar.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:InvalidIdentity	El certificat no està autoritzat, i pot no ser vàlid.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:NoMoreSignersAllowed	El flux definit a la política de mulisignatura no permet més signataris.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:NoSuchPDFField	El camp de signatura indicat no existeix al document PDF a signar.
urn:catcert:psa:1.0:profiles:dss-	El número de pàgina indicat no existeix al

directsign:resultminor:InvalidPageNumber	document PDF a signar.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:InvalidImageFormat	La imatge proporcionada (per signatura visible) no té un format vàlid.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:WrongSignature	El hash i el hash xifrat no es corresponen.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:SignedHashNotFound	No s'ha enviat a PSA el hash signat.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:NotAuthorizedCertificate	El certificat no està autoritzat.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:DocumentsReadNeeded	S'ha de llegir el document a signar.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:InvalidRequest	La <i>request</i> enviada no compleix l' <i>schema XSD</i> .
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:signature:act:AlreadyClosed	L'acte de signatura ja està tancat.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:SignatureValidationNotRequested	No es va sol·licitar la validació de la signatura a la petició.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:NoSignatureValidationRetriesLeft	No s'ha pogut validar la signatura, i ja no queden reintents disponibles.
urn:catcert:psa:1.0:profiles:dss-directsign:resultminor:NoSignatureValidationYet	No s'ha pogut validar la signatura, però encara queden reintents disponibles.

8. Referències

Codi	Descripció	Recursos
[1]	Digital Secure Services	http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=dss
[2]	Web Service Description Language	http://www.w3.org/TR/wSDL
[3]	Secure Socket Layer	http://www.windowsecurity.com/articles/secure_socket_layer.html http://www.unix.com.ua/oreilly/java-ent/servlet/ch08_03.htm http://tldp.org/HOWTO/SSL-Certificates-HOWTO/x64.html
[4]	Microsoft Visual Studio 2005	http://www.microsoft.com/spanish/msdn/vs2005/default.msp http://msdn.microsoft.com/en-us/vstudio/default.aspx
[5]	Microsoft .NET framework	http://msdn.microsoft.com/en-us/netframework/default.aspx http://www.microsoft.com/downloads/details.aspx?displaylang=es&FamilyID=0856eacb-4362-4b0d-8edd-aab15c5e04f5
[6]	Microsoft Services Enhancements	http://www.microsoft.com/downloads/details.aspx?FamilyID=018a09fd-3a74-43c5-8ec1-8d789091255d&displaylang=en
[7]	Web Services Security	http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss http://www.oasis-open.org/committees/download.php/16785/wss-v1.1-spec-os-x509TokenProfile.pdf
[8]	Java Security	http://java.sun.com/javase/6/docs/technotes/tools/index.html#security
[9]	Java	http://java.sun.com/javase/downloads/index.jsp http://java.sun.com/j2se/1.5.0/
[10]	Maven	http://maven.apache.org/download.html http://maven.apache.org/guides/introduction/introduction-to-the-standard-directory-layout.html https://jax-ws-commons.dev.java.net/jaxws-maven-plugin/
[11]	Metro	https://metro.dev.java.net/
[12]	S/MIME	http://es.kioskea.net/crypto/s-mime.php3 http://www.imc.org/ietf-smime/
[13]	XMLEncrypt	http://www.w3.org/TR/xmlenc-core/