presto	**PRotocole d'Echanges Standard et Ouvert** **Référence Technique**

Protocole PRESTO
Référence technique

Version 1.0

Document de travail

Date: 27/06/2006

Résumé

La spécification « PRotocole d'Echanges STandard et Ouvert » 1.0 (PRESTO) est la spécification d'un profil Web Service (ensemble de spécifications de services Web). Ces spécifications sont précisées, amendées ou restreintes afin de favoriser l'interopérabilité.

Ce document constitue une référence technique pour l'implémentation du protocole PRESTO. Ce document est accompagné du guide PRESTO [PRESTO-Guide] qui fournit une description non normative du modèle d'échanges de messages PRESTO.

Etat du document

Ce document est un document de travail, il reflète la version 1 du protocole basé sur les discussions issues des groupes de travail organisés par la DGME. Des travaux complémentaires pourront faire évoluer ce protocole dans une nouvelle version.

Notice

Ce document est destiné aux architectes, chefs de projets et plus généralement à tout public souhaitant avoir la spécification technique complète de PRESTO. Le public visé doit avoir des connaissances sur les technologies Web Services.

presto	**PRotocole d'Echanges Standard et Ouvert** **Référence Technique**

presto	PRotocole d'Echanges Standard et Ouvert Référence Technique

Table des matières

1. Introduction

Ce document définit le profil « Protocole d'Echanges STandard et Ouvert » 1.0 (PRESTO) qui consiste en un ensemble de spécifications de services Web. Ces spécifications sont accompagnées de clarifications, amendements et restrictions afin de favoriser l'interopérabilité.

Le chapitre § 1 "Introduction" introduit le profil PRESTO et décrit ses relations avec d'autres profiles existants.

Le chapitre § 2 "Conventions" décrit les conventions de notations utilisées dans ce profil.

Le chapitre § 3 "Conformité au profil" décrit les conditions de conformité au profil.

Le chapitre § 4 "PRESTO: PRotocole d'Echanges Standard et Ouvert" décrit chaque élément constituant le profil PRESTO. Chaque sous chapitre adresse une composante du profile et est constituée de deux parties : une vue d'ensemble présentant les spécifications retenues pour la composante suivie de sections traitant des points précis des spécifications. Il est à noter qu'il n'y a aucun lien entre les numéros de sections de ce document et les numéros de section des spécifications référencées.

Les Patterns d'échange de messages PRESTO (MEP) supportés par le protocole PRESTO sont présentés dans le Guide PRESTO [PRESTO-Guide]. Il est recommandé de se référer aux modèles d'échanges décrits dans le Guide PRESTO en support de ce document technique.

1.1. Relations avec les profiles du WS-I

En raison de la prolifération des plateformes et technologies de l'Administration électronique, il est important de s'assurer que les différentes implémentations de services Web PRESTO soient interopérables. Cela doit être indépendant de la technologie retenue pour l'implémentation. L'utilisation des profils WS-I définit une base de standards auxquels les implémentations doivent adhérer.

Plus particulièrement, le Basic Profile spécifie un ensemble minimum de spécifications que les services Web doivent supporter pour garantir l'intéropérabilité à travers différentes plateformes. La version actuelle du Basic Profile (BP), c'est-à-dire la version 1.1 (http://www.ws-i.org/Profiles/BasicProfile-1.1.html) intègre SOAP 1.1 (bien que SOAP 1.2

est une recommandation W3C depuis 2003), XML 1.0 et HTTP 1.1 pour la messagerie et le format des messages, WSDL 1.1 et XML Schema 1.0 pour la description des services et enfin UDDI v2 pour la découverte et la publication.

La version initiale du protocole PRESTO est basée sur les spécifications de services Web SOAP 1.2 [SOAP 1.2], WS-Addressing [WS-Addressing], MTOM [MTOM], WS-ReliableMessaging [WS-ReliableMessaging], WS-Security [WS-Security].

Ces spécifications ne sont pas prises en compte dans le Basic Profile 1.1, toutefois la plupart des exigences identifiées font apparaître la nécessité de les considérer. Par exemple les exigences concernant SOAP 1.1 sont prises en compte dans SOAP 1.2.

Le protocole PRESTO se réfère autant que possible au Basic Profile 1.1.

Par ailleurs, la version 2.0 du Basic Profile est attendue pour la fin 2006. Elle sera basée sur les spécifications suivantes : SOAP 1.2, XML InfoSet, WSDL 1.1 and XML Schema 2001 pour la description des services, WS-Addressing, MTOM, XOP, UDDI v2 & v3 pour la découverte et la publication. Un autre profil nommé Reliable Secure Profile (RSP) 1.0 prendra en compte WS-ReliableMessaging et WS-SecureConversation.

Par conséquent, certaines exigences PRESTO citées dans ce document sont amenées à disparaître dans le futur, au bénéfice du Basic Profile 2.0 et du Reliable Secure Profile 1.0. Le protocole PRESTO reprendra simplement les exigences de ces profils WS-I et en sera un cas d'usage.

Le même principe s'applique à la sécurité. Le protocole PRESTO repose sur la sécurité des messages qui apporte des avantages par rapport à la sécurité uniquement située au niveau du transport. La sécurité au niveau du transport n'est plus nécessaire car la sécurité est appliquée directement au message transmis. Une telle approche est rendue possible par le protocole PRESTO afin de permettre par exemple un routage à travers des relais PRESTO, voire des domaines de communication différents (voir les patterns d'échange du guide PRESTO [PRESTO-Guide]).

Le protocole PRESTO s'appuie sur le Basic Security Profile (actuellement en cours de finalisation) pour la sécurité au niveau des messages (http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html). Ce profil permet d'assurer l'interopérabilité des niveaux de sécurité et de définir les règles d'application de la sécurité sur les messages. Le Basic Security Profile spécifie les mécanismes de sécurité acceptables lors des communications par services Web. Cela inclut la sécurité au niveau transport (SSL et TLS), la sécurité des messages SOAP à travers WS-Security [WS-Security] et les types de jetons d'identification supportés. Il existe des profils supplémentaires qui spécifient l'utilisation d'autres jetons tels que les assertions SAML (Security Assertion Markup Language) que le protocole PRESTO peut véhiculer dans ses messages pour gérer les droits d'accès aux données.

Au fur et à mesure de l'avancement des travaux du WS-I, ce profil sera mis à jour pour refléter ces évolutions.

1.2. Relations avec le profile WS-RAMP

Ce document est en partie basé sur le profil WS-RAMP qui est une initiative similaire menée par IBM, Ford et DaimlerChrysler.
(ftp://www6.software.ibm.com/software/developer/library/ramp.pdf).

Les chapitres de ce document marqués par deux astérisques (**) sont des éléments extraits du profil WS-RAMP et sont sujets du copyright suivant.

1.2.1. Copyright de WS-RAMP

1.3. Identification et versions du profile

Ce document est identifié par un nom (ici, Profil PRESTO) et un numéro de version (ici, 1.0). Ensemble, ils identifient une *instance de profil* particulière.

Les numéros de versions sont composés d'un numéro majeur et d'un numéro mineur selon la forme « majeure.mineure ». Ils peuvent être utilisés pour déterminer la précédence d'une instance de profil ; un numéro de version supérieur (selon les éléments majeur et mineur) indique d'une instance est plus récente et donc remplace les instances précédentes.

Les instances de profils avec le même nom (par exemple « Profil Exemple 1.1 » et « Profil Exemple 5.0 ») adressent des problèmes d'interopérabilité d'un même cadre général (bien que les évolutions puissent exiger une modification du cadre exact entre deux instances).

Cette information peut être utilisée pour déterminer si deux instances sont compatibles en amont; c'est-à-dire déterminer si la conformité avec une instance de profile entraîne la compatibilité avec une instance ultérieure. Des instances avec un même nom et un même numéro majeur de version (par exemple « Profil Exemple 1.0 » et « Profil Exemple 1.1 ») PEUVENT être considérées compatibles. Il est à noter que cela n'implique rien en terme de compatibilité dans d'autres directions, c'est-à-dire qu'on ne peut assurer que la conformité avec une instance de profile ultérieure implique la conformité avec une instance récente.

2. Conventions du document

2.1. Conventions de notation

Les mots-clés « DOIT » ("MUST", "SHALL"), « NE DOIT PAS » ("MUST NOT", "SHALL NOT"), « OBLIGATOIRE » ("REQUIRED"), « DEVRAIT » ("SHOULD"), « NE DEVRAIT PAS » ("SHOULD NOT"), « RECOMMANDÉ » ("RECOMMENDED"), « PEUT » ("MAY"), et « OPTIONNEL » ("OPTIONAL") utilisés dans ce document doivent être interprétés conformément à la description du document RFC 2119 [RFC 2119].

Les exigences dans le profil (i.e., celles qui impactent la conformité, comme précisé dans le chapitre « Exigences de conformité ») sont présentées de la manière suivante :

Rnnnn *Texte de l'exigence*.

où "nnnn" est remplacé par un numéro unique au sein des exigences du profil, constituant de ce fait un identifiant unique d'exigence.

Les identifiants d'exigence peuvent être considérés comme associés à un espace de nommage, et ainsi être compatibles avec les noms QNames issus de la spécification Namespaces in XML (http://www.w3.org/TR/REC-xml-names). Si il n'y a pas de préfixe explicite sur un identifiant d'exigence (par exemple « R9999 » en opposition à « bp10 :R9999 »), ce dernier doit être interprété comme appartenant à l'espace de nommage identifié par l'URI de conformité du document en cours. S'il est qualifié, le préfixe doit être interprété selon la correspondance des espaces de nommage, comme documenté plus loin.

Certaines exigences clarifient les spécifications référencées mais n'ajoutent pas de contraintes supplémentaires. Les clarifications sont annotées de la manière suivante : C

Certaines exigences sont dérivées de spécifications en cours de standardisation. Ces exigences sont annotées de la manière suivante : xxxx, où « xxxx » est l'identifiant de la spécification (par exemple « WSDL20 » pour WSDL version 2.0). Il est à noter que de tels travaux n'étant pas achevés au moment de la publication de ce document, la spécification d'où l'exigence est issue peut changer ; cette information n'est donc incluse qu'a titre indicatif pour ceux qui implémentent.

2.2. Espaces de nommage

Cette spécification utilise un certain nombre de préfixes d'espaces de nommage. Les URI correspondantes sont listées plus bas. Le choix d'un préfixe d'espace de nommage est arbitraire et n'a pas de sens sémantique. Les URI d'espace de nommage de la forme générale « some-URI » représentent des URI dépendant d'une application ou d'un contexte comme défini dans la RFC 2396 [RFC 2396].

Prefix	XML Namespace	Reference(s)
s11	http://schemas.xmlsoap.org/soap/envelope	SOAP 1.1 [SOAP 1.1]
s12	http://www.w3.org/2003/05/soap-envelope	SOAP 1.2 [SOAP 1.2]
wsdl	http://schemas.xmlsoap.org/wsdl	WSDL 1.1 [WSDL 1.1]
soap	http://schemas.xmlsoap.org/wsdl/soap	WSDL 1.1 [WSDL 1.1]
uddi	urn:uddi-org:api_v2	UDDI 2.0 [UDDI 2.0]
xs	http://www.w3.org/2001/XMLSchema	XML Schema [Part 1, 2]

ds	http://www.w3.org/2000/09/xmldsig#	XML Digital Signatures
xenc	http://www.w3.org/2001/04/xmlenc#	XML Digital Encryption
wsa	http://schemas.xmlsoap.org/ws/2004/08/addressing	WS-Addressing [WS-Addressing]
wsrm	http://schemas.xmlsoap.org/ws/2005/02/rm	WS-ReliableMessaging [WS-ReliableMessaging]
wssc	http://schemas.xmlsoap.org/ws/2005/02/sc	WS-SecureConversation [WS-SecureConversation]
wsu	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd	WS-SecurityUtility
wsse	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.1.xsd	WS-Security Extensions [WS-Security]
wst	http://schemas.xmlsoap.org/ws/2005/02/trust	WS-Trust [WS-Trust]
wsp	http://schemas.xmlsoap.org/ws/2004/09/policy	WS-Policy [WS-Policy]
sp	http://schemas.xmlsoap.org/ws/2005/07/securitypolicy	WS-SecurityPolicy [WS-SecurityPolicy]

3. Conformité au profil

La conformité au profil est définie par l'adhérence à l'ensemble des *exigences* définies pour une *cible* spécifique, dans le *cadre* du profil. Cette section explique ces termes et décrit comment la conformité doit être définie et utilisée.

3.1. Exigences de conformité

Les exigences font état des critères de conformité au Profil. Elles réfèrent typiquement à une spécification existante et intègrent des réajustements, amplifications, interprétations et clarifications dans le but d'améliorer l'intéropérabilité. Toutes les exigences du profil sont considérées comme normatives et celles des spécifications référencées appartenant au

cadre (voir le chapitre « cadre de conformité ») doivent également être considérées comme normatives. Quand les exigences du Profil et des spécifications référencées se contredisent, les exigences du profil sont prioritaires.

Les niveaux d'exigences, utilisant le vocabulaire de la RFC [RFC2119] (par exemple DOIT, PEUT…) indique la nature de l'exigence et son impact sur la conformité. Chaque exigence est identifiée individuellement (par exemple R9999).

Par exemple ;

R9999 *les COMPOSANTS GRAPHIQUES DEVRAIENT être de forme arrondie*.

Cette exigence est identifiée par « R9999 », s'applique à la cible COMPOSANT GRAPHIQUE (voir plus loin), et place une exigence conditionnelle pour les composants graphiques ; i.e. bien que cette exigence doit être satisfaite pour assurer la conformité dans la majorité des cas, il y a quelques situations dans lesquelles il peut y avoir des raisons valides de ne pas la satisfaire (raisons expliquées dans l'exigence où dans le texte associé).

Chaque constat d'exigence contient exactement un mot clé de niveau d'exigence (par exemple « DOIT ») et une cible de conformité (par exemple « MESSAGE »). Un texte supplémentaire peut être inclus pour illustrer une exigence ou un groupe d'exigences ; dans tous les cas le texte autour d'un constat d'exigence ne doit pas être considéré comme déterminant pour la conformité.

Les définitions des termes dans le Profil sont considérées comme faisant autorité pour déterminer la conformité.

Aucune des exigences du Profil, sans tenir compte de leur niveau de conformité, ne doit être interprétée comme une limitation de la capacité d'une implémentation conforme à appliquer des mesures de sécurité en réponse à une réelle ou possible menace (par exemple une attaque de dénis de service).

3.2. Cibles de conformité

Les cibles de conformité identifient quels artéfacts (par exemple, message SOAP, description WSDL, donnée de registre UDDI) ou quels partis (par exemple un moteur SOAP, un utilisateur final) font l'objet de l'exigence.

Cela permet de définir la conformité dans différents contextes, d'assurer l'absence d'ambiguïté dans l'interprétation de l'application des exigences. Cela permet également de tester la conformité des artéfacts (par exemple les messages SOAP ou les descriptions WSDL) et du comportement des différents partis d'un service Web (par exemple les clients et les instances de service).

Les cibles de conformité des exigences sont des artéfacts physiques dans la mesure du possible afin de permettre les tests et d'écarter les ambiguïtés.

Les cibles d'exigences suivantes sont utilisées dans le Profil :

- **MESSAGE** – éléments du protocole transportés au sein d'une enveloppe (e.g., SOAP/HTTP messages)
- **ENVELOPPE** – la sérialisation de l'élément s12:Envelope et de son contenu.
- **DESCRIPTION** – description des types, messages, interfaces, protocoles, formats de messages et points d'accès réseau associés avec les services Web (par exemple une description WSDL)
- **MANDATAIRE_EMETTEUR -** élément logiciel qui génère un message en fonction du protocole associé et qui émet le message vers un mandataire récepteur selon un chemin d'envoi pouvant mettre en jeu un ou plusieurs relais.
- **MANDATAIRE_RECEPTEUR –** élément logiciel qui reçoit un message en fonction du protocole associé.
- **RELAI –** élément logiciel pouvant intervenir dans le chemin d'un envoi vers un mandataire récepteur.

3.3. Cadre de conformité

Le cadre du Profil délimite les technologies qu'il adresse. En d'autres mots, le Profil vise uniquement à améliorer l'interopérabilité au sein de son propre périmètre. Généralement, le périmètre d'un Profil est délimité par les spécifications qu'il référence.

Le périmètre du Profil est autrement réajusté par des points d'extensibilité. Les spécifications référencées fournissent souvent des mécanismes d'extensions et des paramètres de configuration non spécifiés ou ouverts. Quand il est identifié comme un point d'extension, un mécanisme ou paramètre est sorti du cadre du Profil et son utilisation (ou non utilisation) n'est pas prise en compte pour la conformité.

Il est à noter que le Profil peut toujours placer des exigences sur l'utilisation d'un point d'extension. Les utilisations spécifiques de points d'extensions peuvent également être restreints par d'autres Profils pour améliorer l'interopérabilité quand ils sont utilisés en conjonction avec le Profile.

L'utilisation de points d'extension pouvant causer des problèmes d'interopérablité, leur utilisation doit être négociée ou documentée d'une manière ou d'une autre par les partis d'un service web, par exemple cela peut prendre la forme d'un accord « hors bande ».

Le cadre du Profile est défini par les spécifications référencées dans l'appendice A, et étendu par les points d'extension de l'appendice B.

4. PRESTO: PRotocole d'Echanges Standard et Ouvert

Cette section décrit le "PRotocole d'Echanges Standard et Ouvert" 1.0 (PRESTO) en tant que Profil normatif pour un ensemble de spécifications de services Web référencées dans les chapitres suivants.

4.1. Protocoles de transport supportés par PRESTO

Le protocole PRESTO est fondamentalement agnostique par rapport au transport. Le véhicule du protocole PRESTO peut être un protocole de haut niveau comme http (situé plus haut dans la pile qu'un protocole réseau comme TCP), ou n'importe quel autre protocole réseau tel que TCP ou UDP.

La section § 5 « Lien avec le transport » de ce document décrit les liens vers les protocoles de transports supportés.

La première version de PRESTO définit un lien pour le transport HTTP uniquement. Les futures versions du Profil pourront considérer d'autres transports (FTP, SMTP…).

4.2. Messages PRESTO

Le format des messages PRESTO est XML, qui est le format de fait pour les communications entre applications. XML fournit une structure définie qui donne un sens aux données.

Le profil rend obligatoire le respect de la recommandation XML du W3C dans sa version 1.1. Cette spécification XML se trouve à l'adresse http://www.w3.org/XML.

Ainsi, l'utilisation de XML pour fournir la structure de base des messages assure l'interopérabilité entre les systèmes et les applications dans un espace hétérogène « eGouvernement ».

Le format d' « enveloppe » pour les messages permet de positionner des méta-données à côté de la charge utile. Le message PRESTO utilise le format SOAP en tant qu'enveloppe pour encapsuler les méta-données et la charge utile.

Ce Profile rend obligatoire l'utilisation de SOAP dans la version 1.2 [SOAP 1.2].

Cette section du profil incorpore les spécifications suivantes par référence, et défini leurs points d'extensions :

- SOAP 1.2 (W3C Recommendation 24 June 2003) [SOAP 1.2]

Le message SOAP PRESTO contient dans ses entêtes les méta-données techniques qui concernent l'adressage, la fiabilité et la sécurité. Ces sujets sont couverts respectivement dans les sections suivantes § 4.4 "Adressage des Messages", § 4.6 "Livraison fiable des messages et qualité de service" et § 4.7 "Sécurité".

Les données métier sont OPAQUES[1] pour le protocole PRESTO. Les données métier sont placées dans le corps du message SOAP PRESTO.

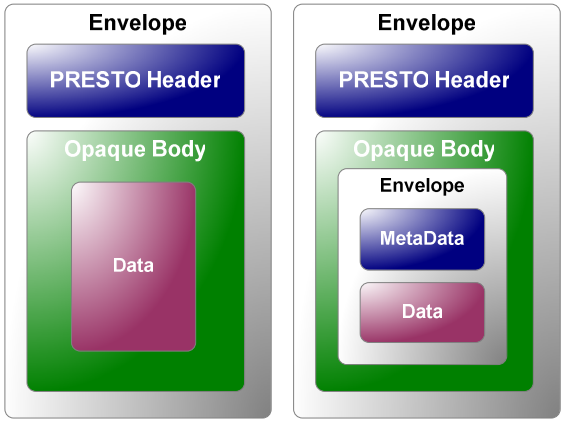
Un seul élément racine DOIT être présent au sein de l'élément s12:Body. L'enveloppe technique PRESTO permet à n'importe quelle charge utile d'être imbriquée dans le message SOAP PRESTO en tant que donnée métier.

Si des méta-données métier (méta-données associées aux données métier telles que des documents binaires) sont exigées, ces méta-données métier ne doivent pas être considérées comme une partie de l'entête PRESTO mais comme une partie de la charge utile (corps de message opaque).

La charge utile peut donc être :

- des données métier,
- une enveloppe constituée de méta-données et des données métier.

[1] Opaque signifie que les données présentes dans le corps du message ne sont pas utilisées pour assurer les services du profil PRESTO (adressage, fiabilité du message, etc.).



4.3. Envoi et reception de Messages

4.3.1. Utilisation de Document-Literal WSDL**

Le Basic Profile du WS-I définit les termes « document-literal » et « rpc-literal » (http://ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html#WSDLMSGS) s'appliquant sur les styles d'écriture autorisés pour les WSDL. Ce Profil limite le choix du style d'écriture WSDL autorisé par l'exigence R2705 (http://ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html#R2705) au style "document-literal".

R3001 *Un wsdl:port dans une DESCRIPTION DOIT utiliser un binding "document-literal"*.

Etant donné que le Profil exige l'utilisation du style WSDL « document-literal », il est à noter que la convention nommée « document-literal wrapped » peut être utilisée pour reproduire une sémantique RPC.

Les caractéristiques du mode « document-literal wrapped » sont les suivants :

- les wsdl:messages d'entrée et de sortie n'ont qu'une seule partie,
- les éléments wsdl:part des messages d'entrée et de sortie ont chacun un attribut @element,
- la déclaration de l'élément xsd:element référencé pour le message d'entrée wsdl:part/@element a un attribut « nom » avec la même valeur que l'opération pour laquelle le wsdl:message sera utilisé,

- la declaration de l'élément xsd:element référencé pour le message d'entrée wsdl:part/@element a un attribut « nom » avec la même valeur que l'opération pour laquelle le wsdl:message sera utilisé, suivi par « Response »,
- le type complexe de l'élément n'a pas d'attribut,
- le « binding » WSDL est du style « document-literal »

Par exemple,

CORRECT:

```
<types>
   <schema>
      <element name="myMethod"/>
         <complexType>
            <sequence>
               <element name="x" type="xsd:int"/>
            </sequence>
         </complexType>
      </element>
      <element name="myMethodResponse"/>
         <complexType>
            <sequence>
               <element name="y" type="xsd:string"/>
            </sequence>
         </complexType>
      </element>
   </schema>
</types>
<message name="myMethodRequest">
   <part name="body" element="myMethod"/>
</message>
<message name="myMethodResponse">
   <part name="body" element="myMethodResponse"/>
</message>
<message name="empty"/>
<portType name="myPortType">
   <operation name="myMethod">
      <input message="myMethodRequest"/>
      <output message="myMethodResponse"/>
   </operation>
</portType>
```

4.3.2. Redéfinition des exigences du WS-I Basic Profile 1.1**

Normalement, un Profil ne doit JAMAIS contredire une exigence de spécification référencée. Toutefois, dans le cas présent, cela est nécessaire afin d'accomoder la composition entre le Basic Profile du WS-I et les spécifications WS-Addressing et WS-ReliableMessaging. Nous esperons que le WS-I traitera ces questions liées à la composition dans une prochaine révision du Basic Profile.

4.3.2.1. Enveloppe SOAP dans un message HTTP Response **

Typiquement, le code réponse HTTP pour un WSDL « one-way » devrait être « 202 Accepted » sans enveloppe SOAP dans le corps de la réponse HTTP. Toutefois, des spécifications de services Web telles que WS-ReliableMessaging peuvent causer la génération de messages SOAP qui ne sont pas considérés comme des réponses applicatives mais qui doivent être transmis à l'émetteur du message HTTP. Par exemple, quand on utilise WS-ReliableMessaging, si l'élément de référence wsrm:AcksTo a la valeur « URI anonyme », alors le destinataire (par exemple le mandataire récepteur PRESTO) doit retourner des messages d'acquittement de séquence à la source (par exemple le mandataire émetteur PRESTO) dans le message de réponse http. Si le message est un message « one-way », il n'y a pas de message réponse à renvoyer, ce qui indique que le destinataire doit générer une enveloppe SOAP de réponse afin de transmettre l'acquittement de séquence.

Cela n'est pas conforme aux exigences du Basic Profile R2714 et R2750 qui statuent que la réponse à un message « one-way » doit inclure un code de réponse HTTP « 2xx » sans enveloppe SOAP et que l'émetteur du flux doit ignorer toute envelope retournée.

Toutefois, étant donné que la source a utilisé l'adresse anonyme, il est à prévoir que ce comportement est attendu. Pour cette raison le Profil redéfini les R2714 and R2750 du Basic Profile 1.1 du WS-I Basic Profile.

R3002 *Un MESSAGE de Réponse HTTP correspondant à une opération WSDL one-way PEUT contenir une enveloppe SOAP dans son corps de message.*

4.4. Adressage des Messages**

Afin de répondre aux différents types d'échanges (MEPs), il est nécessaire de disposer d'une solution d'adressage.
Ce Profil rend obligatoire l'utilisation de la proposition au W3C WS-Addressing. Nous envisagerons d'utiliser la Recommandation W3C lorsque cette dernière sera définie par le groupe de travail W3C.

Cette section du profil incorpore les spécifications suivantes par référence, et définit leurs points d'extensions :

- WS-Addressing (W3C Member Submission 10 August 2004) [WS-Addressing]

4.4.1. Utilisation des blocks d'entête d'adressage **

4.4.1.1. Présence des blocs d'entête

R0001 *Une ENVELOPPE DOIT contenir exactement un entête wsa:To.*

R0002 *Une ENVELOPPE DOIT contenir exactement un entête wsa:MessageId.*

R0003 *Une ENVELOPPE DOIT contenir exactement un entête wsa:Action.*

Par exemple,

CORRECT:

```
<s12:Envelope
      xmlns:s11="http://schemas.xmlsoap.org/soap/envelope"
      xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" >
  <s12:Header>
    <wsa:To>http://example.com/service</wsa:To>
    <wsa:ReplyTo s12:mustUnderstand='1'>
      <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
    </wsa:ReplyTo>
    <wsa:MessageId>uuid:aaaabbbb-cccc-dddd-eeee-ffffffffffff</wsa:MessageId>
    <wsa:Action>http://example.com/service/tbd</wsa:Action>
  </s12:Header>
  <s12:Body>

      <!-- body contents OPAQUE to the PRESTO protocol -->

  </s12:Body>
</s12:Envelope>
```

4.4.1.2. Liens entre une opération d'entrée et une opération de sortie

Dans le contexte d'une messagerie asynchrone, il est nécessaire que l'enveloppe contienne des informations de corrélation suffisantes pour permettre au récepteur d'un message de réponse de corréler cette réponse avec le message de requête original.

R0010 *N'importe quelle ENVELOPPE PEUT contenir un entête wsa:ReplyTo.*

R0011 *Si, dans une description WSDL, une wsdl:operation est décrite avec un élément wsdl:output, l'ENVELOPPE du message correspondant au wsdl:input de cette opération DOIT contenir un élément d'entête wsa:ReplyTo.*

R0012 *Si, dans une description WSDL, une wsdl:operation est décrite avec un élément wsdl:output, l'ENVELOPPE du message correspondant au wsdl:output de cette operation DOIT contenir un élément d'entête wsa:RelatesTo contenant la valeur wsa:MessageId du message d'entrée correspondant.*

4.4.1.3. Utilisation de l'attribut s12:mustUnderstand

Un MANDATAIRE_EMETTEUR qui émet un message de requête dont une réponse doit être remise à l'adresse spécifiée dans l'élément d'entête wsa:ReplyTo doit assurer que le destinataire du message comprend la sémantique de l'entête wsa:ReplyTo.

R0020 *Si une ENVELOPPE contient un élément d'entête wsa:ReplyTo, ce dernier DOIT avoir un attribut s12:MustUnderstand avec la valeur '1'.*

4.4.2. Considérations pour l'asynchronisme Request/Response**

4.4.2.1. Attentes d'une réponse HTTP

Dans le contexte du Basic Profile du WS-I, la seule raison pour que la réponse d'une requête HTTP soit vide est qu'il s'agit d'un envoi one-way. De toute manière, la spécification WS-Addressing introduit l'élément d'entête wsa:ReplyTo qui change la nature de l'opération requête-réponse pour une mise en œuvre sur http. Si un message de requête inclut un élément d'entête wsa:ReplyTo qui ne contient pas l'URI anonyme de WS-Addressing, alors le message de réponse est attendu à l'adresse indiquée dans l'élément wsa:ReplyTo et non dans la réponse HTTP. Comme la réponse n'est pas transmise dans le message de réponse HTTP, la réponse HTTP sera du type « 202 Accepted ». Cela ne contredit pas les exigences du Basic Profile de WS-I, il s'agit d'un changement de comportement.

R0100 *Si une enveloppe correspondant à un message wsdl:input contient un élément d'entête wsa:RelyTo contenant une valeur autre que l'URI Anonyme de WS-Addressing, alors le message de réponse HTTP PEUT avoir un code retour « 202 Accepted » et un contenu vide.*

4.4.3. Composition avec WS-Security**

Les exigences de cette section s'appliquent quand WS-Security est utilisé en composition avec WS-Addressing.

4.4.3.1. Signature des blocs d'entête

R2000 *Si ils sont présents dans l'ENVELOPPE, chacun des éléments d'entête SOAP suivants DOIT être inclus dans la signature quand le s12:Body est signé : wsa:To, wsa:From, wsa:Action, wsa:ReplyTo, wsa: FaultsTo, wsa:MessageId, wsa:RelatesTo.*

4.5. Gestion des pièces jointes

Cette section du profil incorpore les spécifications suivantes par référence, et définit leurs points d'extensions :

- MTOM 1.0 (W3C Recommendation 25 January 2005) [MTOM]
- XOP 1.0 (W3C Recommendation 25 January 2005) [XOP]

4.5.1. Transport de documents binaires

Afin de respecter l'infoset XML, les documents binaires doivent être inclus dans le document XML sous la forme d'un élément binaire base 64.

Le contenu binaire de l'élément XML peut être décrit en utilisant un attribut xmime:ContentType comme défini dans la note du W3C « Describing Media Content of Binary Data in XML » (http://www.w3.org/TR/xml-media-types/).

R2000 *Un document binaire DOIT être encodé en base 64 puis inclus dans le message.*

4.5.2. Optimisation du transport des données binaires

L'utilisation de l'encodage base 64 augmente la taille des documents. L'utilisation du mécanisme d'optimisation MTOM/XOP facilite le transport des documents en sélectionnant et en optimisant les éléments binaires de la donnée XML.

Les éléments encodés en base 64 doivent être dans une forme canonique du type de donnée XML xs:base64Binary.

R2000 *Pour être optimisés, les caractères de tous les éléments binaires du MESSAGE doivent être dans une forme canonique du type de donnée XML xs:base64Binary.*

4.6. Livraison fiable des messages et qualité de service

Le protocole PRESTO exige comme qualité de service que la livraison des messages soit garantie, comme le propose la spécification WS-ReliableMessaging (WS-RM). L'utilisation du protocole WS-RELIABLEMESSAGING est obligatoire.

Cette section du profil incorpore les spécifications suivantes par référence, et définit leurs points d'extensions :

- WS-ReliableMessaging (WS-RM) [WS-ReliableMessaging]

4.6.1. Bloc d'entête de séquence**

4.6.1.1. Utilisation de l'attribut s12:mustUnderstand

Quand un message est envoyé de manière fiable, avec WS-ReliableMessaging, il est impératif que le destinataire puisse traiter correctement l'élément d'entête wsrm:Sequence.

R1010 *Si une ENVELOPPE contient un élément d'entête wsrm:Sequence, ce dernier DOIT avoir un attribut s12:MustUnderstand avec une valeur de '1'.*

Par exemple,

INCORRECT:

```
<s12:Envelope
      xmlns:s11="http://schemas.xmlsoap.org/soap/envelope"
      xmlns:wsrm="http://schemas.xmlsoap.org/ws/2005/02/rm"
      xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
  <s12:Header>
    <wsa:MessageID>http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546817</wsa:MessageID>
    <wsa:To>http://example.org/service</wsa:To>
    <wsa:From>http://example.com/client/rm</wsa:From>
    <wsa:Action>http://example.org/service/credit</wsa:Action>
    <wsa:ReplyTo s12:mustUnderstand='1'>
      <wsa:Address>http://example.com/client</wsa:Address>
    </wsa:ReplyTo>
    <wsrm:Sequence>
      <wsrm:Identifier>
        http://example.org/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546817
      </wsrm:Identifier>
      <wsrm:MessageNumber>42</wsrm:MessageNumber>
    </wsrm:Sequence>
  </s12:Header>
  <s12:Body>
      <!-- body contents OPAQUE to the PRESTO protocol -->
  </s12:Body>
```

```
</s12:Envelope>
```

CORRECT:

```
<s12:Envelope
      xmlns:s11="http://schemas.xmlsoap.org/soap/envelope"
      xmlns:wsrm="http://schemas.xmlsoap.org/ws/2005/02/rm"
      xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
  <s12:Header>
    <wsa:MessageID>http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546817</wsa:MessageID>
    <wsa:To>http://example.org/service</wsa:To>
    <wsa:From>http://example.com/client/rm</wsa:From>
    <wsa:Action>http://example.org/service/credit</wsa:Action>
    <wsa:ReplyTo s12:mustUnderstand='1'>
      <wsa:Address>http://example.com/client</wsa:Address>
    </wsa:ReplyTo>
    <wsrm:Sequence s12:mustUnderstand='1'>
      <wsrm:Identifier>
        http://example.org/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546817
      </wsrm:Identifier>
      <wsrm:MessageNumber>42</wsrm:MessageNumber>
    </wsrm:Sequence>
  </s12:Header>
  <s12:Body>
      <!-- body contents OPAQUE to the PRESTO protocol -->
  </s12:Body>
</s12:Envelope>
```

4.6.2. Bloc d'entête d'acquittement de séquence**

4.6.2.1. Piggy-backing des acquittements de séquence

Quand une enveloppe SOAP est transmise à la même adresse que celle précisée par un élément d'entête wsrm:CreateSequence/wsrm:AcksTo, alors la destination (par exemple un mandataire récepteur PRESTO) peut ajouter un acquittement de séquence dans l'entête SOAP (pour chaque séquence idoine) au lieu de les transmettre à la source (par exemple un mandataire émetteur PRESTO) dans des messages séparés. Cette pratique est couramment appelée « piggy-backing » des acquittements.

R1021 *Un élément d'entête wsrm:SequenceAcknowledgement, pour chaque séquence idoine, PEUT être inclus dans n'importe quelle ENVELOPPE transmise au point de terminaison spécifié dans l'élément d'entête wsrm:CreateSequence/wsrm:AcksTo.*

4.6.2.2. Transport d'acquittement dans une réponse HTTP d'opération one-way

Les exigences du Basic Profile R2714 (http://ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html#R2714) et R2750 (http://ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html#R2750) sont en conflit avec la possibilité de transmettre des acquittements de séquence dans les messages de réponse HTTP pour les envois one-way avec fiabilité.

Cette exigence est susceptible d'être courante pour les cas d'utilisation où la source de l'échange fiable est située derrière un firewall, ce Profil redéfinit les exigences du Basic Profile R2714 et R2750.

R1022 *Un MESSAGE correspondant à la réponse d'un message de type one-way PEUT contenir une enveloppe SOAP avec un élément d'entête wsrm:SequenceAcknowledgement et aucun élément dans le s12:Body.*

R1023 *Un CONSOMMATEUR qui spécificie l'URI Anonyme de WS-Addressing comme adresse de l'élément d'entête wsrm:CreateSequence/wsrm:AcksTo DOIT pouvoir recevoir et traiter des messages réponses HTTP contenant une enveloppe SOAP pour des opérations one-way.*

4.6.3. Composition avec WS-Addressing**

4.6.3.1. Utilisation de l'URI Anonyme de WS-Addressing

R1050 *Quand l'URI Anonyme de WS-Addressing est utilisée dans l'élément d'entête wsrm:AcksTo d'une enveloppe wsrm:CreateSequence, tous les messages d'acquittement de la séquence DOIVENT être renvoyés dans le flux de réponse de la connexion HTTP utilisé pour la réception des messages contenant l'élément d'entête wsrm:Sequence.*

La capacité de retenter l'envoi de messages est un élément clé de WS-ReliableMessaging. Pour cela, il est nécessaire que l'émetteur d'un message puisse établir une connexion avec le destinataire. Ainsi, l'utilisation d'une URI anonyme dans l'élément d'entête wsa:ReplyTo de la requête enfreint cette règle car l'émetteur de la réponse ne peux initier une connexion en retour pour supporter ReliableMessaging.

R1051 *Pour un échange de messages dont le message d'entrée (la requête) et le message de sortie (la réponse) doivent être transmis avec fiabilité (WS-ReliableMessaging), l'ENVELOPPE du message d'entrée NE DOIT PAS être l'URI anonyme de WS-Addressing.*

4.6.4. Composition avec WS-Security**

Afin d'assurer un échange sécurisé de bout en bout entre partenaires, le cas d'usage classique compose WS-ReliableMessaging, WS-Addressing et WS-Security comme exigé dans le Basic Security Profile (BSP) 1.0 de WS-I. Les exigences de cette section s'appliquent à chaque fois que WS-Security est utilisé en composition avec WS-ReliableMessaging.

4.6.4.1. Signature des blocs d'entête

Afin de garantir l'intégrité des éléments du protocole WS-ReliableMessaging (par exemple pour assurer qu'ils n'ont pas été interceptés par une attaque de type « man-in-the-middle »), ces éléments doivent être signés.

R2010 Quand les éléments d'entête suivants sont présents dans une ENVELOPPE, ils DOIVENT être inclus dans la signature si l'élément s12:Body est signé : *wsrm:Sequence, wsrm:SequenceAcknowledgement.*

4.6.4.2. Permettre la détection des attaques de rejeu

La valeur de l'élément wsu:Timestamp doit être changée pour les retransmissions afin de permettre à la couche de sécurité du destinataire (par exemple le mandataire récepteur PRESTO) d'utiliser cette information pour distinguer un renvoi valide de message non acquitté d'une possible attaque de rejeu par un tiers.

R2011 *L'élément d'entête wsse:Security dans une ENVELOPPE DOIT contenir un élément wsu:TimeStamp.*

R2012 *La valeur de l'élément wsu:Timestamp dans une ENVELOPPE doit être celle de l'heure système au moment de l'envoi fiable pour chaque envoi de message (envoi initial et chaque envoi suivant pour un message non acquitté).*

4.6.5. Garantie de livraison

Afin d'assurer une qualité de service maximum pendant les échanges de documents, l'assurance ExactlyOnce (Exactement Une Fois) est exigée. L'assurance InOrder (Dans l'Ordre) est optionnelle.

R2011 *L'assurance de livraison des MESSAGES DOIT être ExactlyOnce.*

4.7. Sécurité

Le scénario d'utilisation PRESTO exige une sécurité des messages de bout en bout telle que proposée par le Basic Security Profile (BSP) 1.0 (http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html).

L'utilisation du BSP est obligatoire pour l'intégrité des messages et optionnelle pour la confidentialité des données métier. En cas de besoin de sécurité, le profile BSP est le mécanisme retenu.

Au moment de cette publication, le Profil BSP est encore à l'état de document de travail. Ce profil prendra en compte les évolutions du BSP.

Cette section du profil incorpore les spécifications suivantes par référence, et définit leurs points d'extensions :

- WS-I Basic Security Profile (BSP) 1.0 (WS-I Working Group Draft 20 January 2006) (http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html)

4.7.1. Contraintes d'application du Basic Security Profile

Ce profil contraint certaines options autorisées par le Profil BSP afin d'améliorer l'interopérabilité.

4.7.1.1. Utilisation de certificats X.509v3 pour la signature et le chiffrement

Les certificats X.509v3 sont utilisés pour garantir l'intégrité et la confidentialité des messages PRESTO.

R2011 Un certificat X509v3 DOIT être utilisé si le MESSAGE est signé.

R2012 Un certificat X509v3 DOIT être utilisé si le MESSAGE est chiffré.

4.7.1.2. Signature du message

Les données métier sont OPAQUES pour le protocole PRESTO, ainsi quand une signature est requise, la totalité des données doit être signée.

R2011 *Quand le MESSAGE est signé, la totalité de s12:Body DOIT être signée.*

Conformément aux recommandations du W3C, les algorithmes suivants sont recommandés pour la signature des messages PRESTO : RSAwithSHA1 for signature, Canonical with Comments for canonicalization, XPath for transformations.

Les algorithmes de signature doivent être conformes avec le Basic Security Profile 1.0.

4.7.1.3. Chiffrement du message

Les données métier sont OPAQUES pour le protocole PRESTO, ainsi quand une signature est requise, la totalité des données doit être chifrée.

R2011 *Quand le MESSAGE est chiffré, la totalité de s12:Body DOIT être chifrée.*

Les algorithmes de chiffrement doivent être conformes avec le Basic Security Profile 1.0.

5. Lien avec le transport

Cette section décrit comment mettre en œuvre ce Profil avec les transports supportés.

5.1. Lien au transport HTTP

La mise en œuvre du Profil PRESTO sur le transport http se fait en conformité avec le Basic Profile 1.1 en tenant compte des exceptions listées dans les chapitres précédents de ce document.

Annexe A: Spécifications référencées

[MTOM]
Ed. Noah Mendelsohn et al, "SOAP Message Transfer Optimization Mechanism," July 2004.

[PRESTO-Ref]
"A Technical Reference for the PRESTO protocol," April 2006.

[RFC 2119]
S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119, Harvard University, March 1997.

[RFC 2396]
T. Berners-Lee, et al, "Uniform Resource Identifier (URI): Generic Syntax," RFC 2396bis, W3C/MIT, July 2004.

[SOAP 1.1]
Don Box et al, W3C Note "Simple Object Access Protocol (SOAP) 1.1", May 2000.

[SOAP 1.2]
M. Gudgin et al, "SOAP Version 1.2 Part 1: Messaging Framework," June 2003.

[UDDI]
Ed. Tom Bellwood, "UDDI Version 2.04 API Specification," July 2002.

[WSDL 1.1]
Ed. Erik Christensen et al, "Web Service Description Language (WSDL) 1.1", March 2001.

[WS-Addressing]
D. Box et al, "Web Services Addressing (WS-Addressing)," August 2004.

[WS-Policy]
D. Box et al, "Web Services Policy Framework (WS-Policy)," September 2004

[WS-MetadataExchange]
Keith Ballinger et al, "Web Services Metadata Exchange (WS-MetadataExchange)," September 2004

[WS-RM]
Ruslan Bilorusets et al, "Web Services Reliable Messaging (WS-ReliableMessaging)," February 2005.

[WS-RMPolicy]
Stefan. Batres et al, "Web Services Reliable Messaging Policy Assertion (WS-RM Policy)," February 2005.

[WS-SecureConv]
Steve Anderson et al, "Web Services Secure Conversation Language (WS-SecureConversation)," February 2005.

[WS-Security]
A. Natalin et al, "Web Services Security: SOAP Message Security 1.0", May 2004.

[WS-SecX509]
Ed. Phillip Hallam-Baker et al, "Web Services Security: X.509 Token Profile V1.0", March 2004.

[WS-SecurityPolicy]
Giovanni Della-Libera et al, "Web Services Security Policy Language (WS-SecurityPolicy)," July 2005.

[WS-Trust]
Steve Anderson et al, "Web Services Trust Language (WS-Trust)," February 2005.

[XMLDSIG]
Eastlake III, D., Reagle, J., and Solo, D., "XML-Signature Syntax and Processing", http://www.ietf.org/rfc/rfc3275.txt, March 2002.

[XMLENC]
Imamura, T., Dillaway, B., and Simon, E., "XML Encryption Syntax and Processing", http://www.w3.org/TR/xmlenc-core/, August 2002.

[XML Schema, Part 1]
H. Thompson et al, "XML Schema Part 1: Structures," May 2001.

[XML Schema, Part 2]
P. Biron et al, "XML Schema Part 2: Datatypes," May 2001.

[XOP]
Ed. Noah Mendelsohn et al, "XML-binary Optimized Packaging (XOP)," June 2004.

Annexe B: Points d'extensibilité

Cette section identifie des points d'extensions du profile PRESTO.

Bien que leur utilisation peut avoir un impact sur l'interopérabilité, ces éléments ne font pas partie la version 1.0 du profile PRESTO. Ils devront être étudiés et pourront être intégrés à des versions ultérieures de PRESTO.

La liste actuelle non limitative des extensions de PRESTO comprend les points suivants :

- Inclure dans les entête PRESTO le jeton d'horodatage qui sera défini dans le Référentiel Général de Sécurité,
- Etudier l'application de PRESTO pour les documents de taille importante (ordre du Go), définition d'un algorithme de découpage au besoin,
- Intégration de WS-SecureConversation
- Intégration de WS-Policy
- Routage de messages

Annexe C: Glossaire et acronymes

Algorithme à clé symétrique

C'est un algorithme de chiffrement utilisant la même clé pour chiffrer et déchiffrer un message.

Authentification

L'authentification a pour but de vérifier l'identité dont une entité se réclame..

Autorisation

L'autorisation désigne le processus visant à contrôler (accorder ou refuser) l'accès à une ressource en fonction des droits d'accès possédés par l'utilisateur.

Canonisation

La canonisation est le processus de conversion d'un document XML dans un format commun. La canonisation est utilisée avant de signer les documents et de contrôler les signatures.

Composition de protocoles

La composition de protocoles est la capacité à combiner des protocoles en garantissant leur cohérence technique et l'absence d'effets de bord induits.

Contexte de sécurité

Un contexte de sécurité est un ensemble d'éléments nécessaires pour assurer les services de sécurité. Dans le cas de PRESTO il s'agit de l'authentification et de clés négociées pouvant avoir des propriétés complémentaires liés à la sécurité.

Désérialisation

La désérialisation est le processus de construction d'un XML Infoset à partir d'un flux d'octets.

Factory

Une factory est un Web service qui peut créer une ressource à partir de sa représentation en XML.

Intermédiaire SOAP

Un intermédaire SOAP est un nœud SOAP qui n'est ni l'émetteur, ni le destinataire du message SOAP.

Jeton de sécurité

Un jeton de sécurité représente un ensemble d'informations représentant une identité.

Mandataire PRESTO

Un mandataire PRESTO est un service.

Mandataire émetteur PRESTO

Un mandataire émetteur PRESTO est un service permettant de générer un message conforme aux spécifications du protocole PRESTO et capable de l'envoyer à un mandataire récepteur PRESTO. Ce message peut être envoyé à travers d'un ou plusieurs relais PRESTO.

Mandataire récepteur PRESTO

Un mandataie récepteur PRESTO est un service capable de recevoir des messages conformes aux spécifications du protocole PRESTO.

Message

Un message est une unité complète de données prête à être envoyée ou réceptionnée par un service. Dans ce document, un message contient toujours une enveloppe SOAP et peut comporter des parties MIME additionnelles tel que spécifié dans MTOM ainsi que des en-têtes spécifique au protocole de transport.

Pattern d'échange

Un pattern d'échange est un modèle utilisé pour les échanges de messages entre des mandataires PRESTO.

Policy

Une policy est un ensemble de policy alternatives.

Policy Alternative

une policy alternative est un ensemble de 'policy assertions'.

Policy Assertion

Une 'policy assertion' est pour un domaine donné, une exigence, une capacité, une propriété ou un comportement.

Policy Expression

Une 'policy expression' est la représentation en XML Infoset d'une 'policy'.

Relais PRESTO

Un relais PRESTO est un intermédiaire SOAP situé sur la route entre un mandataire émetteur et un mandataire récepteur.

Ressource

Une ressource est une entité adressable via un endpoint référence. Cette entité peut fournir une représentation XML.

Route

Une route est l'ensemble des nœuds SOAP entre le mandataire émetteur PRESTO initial et le mandataire destinataire PRESTO final et par lesquels transitent les messages PRESTO.

Security Context Token

Un Security Context Token (SCT) est la représentation technique d'un contexte de sécurité permettant de nommé ce contexte par une URI respectant [WS-Security].

Sérialisation

C'est le procédé consistant à coder un objet depuis le format machine vers le format réseau dans un format transparent, pour restitution à l'arrivée dans un format machine potentiellement différent.

Service

Composant applicatif intéragissant avec d'autres services via des messages.

Service de jeton de sécurité

Un service de jeton de sécurité (STS) est un web service fournissant des jetons de sécurité (cf [WS-Security]).

Service web

Composant applicatif accessible via un réseau, par l'entremise d'une interface standard, qui peut interagir dynamiquement avec d'autres applications en utilisant des protocoles de communication basés sur le XML, et cela indépendamment du système d'exploitation et des langages de programmation utilisés.

Signature

Une signature est un sceau électronique (valeur) calculée avec un algorithme de cryptographie, liée à des données. Ce sceau garantit à la fois l'origine et l'intégrité du

message. Une signature peut être calculée et vérifiée aussi bien avec un algorithme à clé symétrique qu'un algorithme à clé asymétrique.