



Livre Blanc de sécurité SOA. v.1.0

Project Documentation

Table of Contents

1 Introduction	
1.1 Avant Propos	1
2 La sécurité et SOA	
2.1 Introduction	3
2.2 La démarche de sécurité	4
2.3 Les coûts de la sécurité	6
3 Un modèle de sécurité SOA, J2EE, WS-*	
3.1 Des Concepts de base	8
3.2 Un modèle de sécurité	11
3.3 Questions de sécurité	15
4 Conclusion	
5 Annexes	
5.1 Où trouver les spécifications?	18
5.2 Ressources de documentation	19

1.1 Avant Propos

«Les services Web XML vont rouvrir 70% des chemins d'attaques fermés par les pare-feu lors de la dernière décennie. Ils peuvent transporter virtuellement toutes les données utiles sur le port 80 et le pare-feu ne peut les arrêter.»

- Gartner Group , 2003

Les services Web apportent des bénéfices significatifs pour des applications basées sur l'Architecture Orientée Services, mais exposent des risques importants en terme de sécurité. Créer et gérer un environnement sécurisé pour les services web nécessite la manipulation et la maîtrise de spécifications et standards divers et variés ainsi que des technologies et logicielles et matérielles conséquentes.

Il convient d'étudier la mise en œuvre de la sécurité pour les Architectures Orientées Services (SOA) via les quatre volets suivants :

- **La sécurité niveau Transport** : pare-feu (firewall), VPN (Virtual Private Networks), authentification basique, non-répudiation et cryptage
- **La sécurité niveau Message** : Utilisation des jetons de sécurité afin de valider l'identité du consommateur du service ou du processus, utilisation des assertions d'autorisation pour valider l'accès au services.
- **Sécurité niveau application** : Sécuriser les composants appelées par les Web Services, EJBs, Servlets appelés via les services Web.
- **Sécurité niveau Données** : Cryptage et signature des messages afin de protéger les données stockées ou transmises.
- **Sécurité niveau Environnement** : Monitoring, logging et audit afin d'identifier les problèmes qui doivent être fixés et résolus et établir des communications sûres et fiables.

L'émergence des services web pose plusieurs problématiques dont celle de la sécurité des échanges de messages entre partenaires. Dans une architecture Orientée Services, les services web peuvent exposer des processus métier sensibles qui nécessitent un traitement particulier en terme de sécurité aux deux bouts du canal de communication. En plus, les services web sont des technologies récentes, ceci implique de nouvelles vulnérabilités et attaques ou menaces.

Les services web sont utilisés dans les cas suivants (la liste n'est certainement pas exhaustive) :

- Intégration de systèmes point-à-point
- Intégration d'applications entreprise
- Collaboration et partenariat Business
- E-Business
- Composition des processus métier
- Protection et ouverture des systèmes d'information
- Réduction des coûts du cycle de vie du développement et la maintenance des systèmes d'information

Les solutions de sécurité des services web doivent prendre en charge les concepts suivants :

- L'authentification
- L'autorisation
- La confidentialité
- L'intégrité
- La non-répudiation

En plus de ces concepts, le système de sécurité doit prendre en charge l'audit des actions et messages envoyés afin de tracer l'activité de la sécurité des web services.

Les utilisateurs des services web doivent être identifiés soit via un nom d'utilisateur combiné à un mot de passe soit via un certificat digital. Une fois l'utilisateur identifié, il doit posséder l'autorisation ou l'habilitation nécessaire afin d'effectuer le traitement qu'il a demandé. Toutes informations ou messages sensibles mis en jeu par le traitement doivent être confidentiels et ne doivent pas subir d'altération qui touche à son intégrité d'origine. Une fois le traitement exécuté, des mesures de non-répudiation doivent être mises en place afin d'éviter tout dénis des deux parts (consommateurs/fournisseur).

Les services web reposent sur des topologies applicatives diverses et variées telles que : l'Internet mobile, des passerelles, zones démilitarisées (DMZ), systèmes distribués La communication entre ces technologies s'effectue via des intermédiaires.

La sécurité n'était pas la priorité des organisations travaillant sur les spécifications de la stack WS. La sécurité au niveau de la couche de Transport n'étant pas suffisante, il demeurait un vide qui freinait l'adoption des services web. Heureusement, plusieurs propositions majeures ont été diffusées afin de combler ce vide dont **WS-Security** qui apporte un support globale de l'intégrité, de la confidentialité et l'authentification des messages et **SAML** qui définit un langage commun d'interopérabilité permettant de partager les informations liées à l'authentification et l'autorisation afin de faciliter la mise en place de fonctionnalités SSO et de permettre la délégation des droits. Il est intéressant à noter que d'autres propositions et spécifications tendent à émerger comme **Kerberos** , **XKMS** , **XACML** afin d'apporter un support complémentaire à la stack de sécurité.

La sécurité des messages est d'autant plus nécessaire si des intermédiaires sont présents dans une communication point-à-point. L'expéditeur d'origine et le destinataire final doivent établir des relations de confiance avec ces intermédiaires afin d'assurer la sécurité de bout en bout.

2.1 Introduction

Afin de mettre en œuvre la sécurité pour les architectures SOA, il serait intéressant de comprendre la notion de sécurité ainsi que les services tels que définis dans les normes de sécurité, par exemple ISO 7498-2.

L'adoption des nouvelles technologies dans le développement d'applications entreprise stratégiques a pour but de contribuer à l'amélioration de la compétitivité de l'entreprise. L'Architecture Orientée Services propulse l'entreprise vers une ouverture et une flexibilité de son métier. Les concepts de l'Architecture Orientée Services ne prennent pas en compte la problématique de la mise en œuvre de la sécurité d'accès aux services.

Cette sécurité doit assurer la constante disponibilité des services, l'intégrité des informations stockées et des messages échangés et la confidentialité de ces derniers. L'ouverture a un coût conséquent en terme de sécurité.

La sécurité de l'Architecture Orientée Services se fixe l'objectif suivant :

Protéger le référentiel de services ou les domaines de services contre les risques, et ce d'une manière qui est adaptée à l'entreprise, à son environnement et à l'état du référentiel de services.

- Protéger : Conception, mise en œuvre, et maintenance des contre-mesures de sécurité.
- Le référentiel de services ou les domaines de services : Identification du référentiel des services et de leurs enjeux. Ceci est considéré comme une étape suite à l'urbanisation du système d'information.
- Contre les risques : Identification des risques et des menaces significatives
- De manière adaptée à l'entreprise : Détermination du niveau de criticité des services déployés.
- A son environnement : Identification des menaces externes ou internes, d'origine accidentelle ou intentionnelle.
- L'état du référentiel de services : Identification des vulnérabilités du référentiel de services.

2.2 La démarche de sécurité

La mise en place de l'Architecture Orientée Services doit se placer après un alignement métier et une urbanisation finalisée et décrite dans un document d'Architecture Fonctionnelle. Cette phase d'urbanisation du système d'information à mettre en œuvre met en place un référentiel de services ou des domaines de services ; Ces derniers spécifient à leur tour les exigences en terme de sécurité.

Cette sécurité vise principalement le profil et les identités des clients de services ou des processus. Une autre couche est ajoutée par l'infrastructure technique qui renseigne la sécurité en terme d'accès et d'échanges des informations entre les machines du système d'information et le monde extérieur de l'entreprise.

La démarche de sécurité peut se schématiser selon le plan suivant :

1. Identification du référentiel de services ou des domaines de services
2. Identification des menaces pouvant toucher le fonctionnement et l'activité des services ou des processus métier exposés
3. Identification des vulnérabilités en terme de système informatique
4. Identification des risques en terme de système informatique
5. Détermination du niveau de criticité des domaines de services
6. Conception, mise en œuvre et maintenance des contre-mesures

Le bon fonctionnement des préconisations en terme de sécurité ne sont pas toujours techniques quoique importantes, mais sont étroitement liée à une connaissance du métier attendu du système d'information. D'où la nécessité de la démarche d'urbanisation.

En ce qui concerne les normes et les outils, au risque d'en décevoir beaucoup, je dirai que le meilleur outil est le bon sens, doublé d'une bonne connaissance du fonctionnement de l'entreprise.

Dans une étude de sécurité relative à l'architecture Orientée Services, il conviendrait avant tout de déterminer ce qui mérite d'être protégé et contre quoi avant de commencer à définir les techniques de sécurité à adopter. On peut résumer cette démarche en 3 mots: actifs, risques, contre-mesures:

Actifs	On se demande d'abord quels processus métier sont vraiment indispensables pour que le système d'information puisse remplir sa mission. En d'autres termes, quels sont les équipements, les applications, le personnel informatique dont l'entreprise.
Risques	Le risque résulte de la combinaison de menaces et de vulnérabilités. Ici, on privilégie les menaces probables, ainsi que les vulnérabilités les plus pertinentes: <ul style="list-style-type: none"> • Menaces: la menace majeure vient-elle de l'extérieur de l'entreprise ou de l'intérieur?. Vient-elle d'interventions humaines ? • Vulnérabilités: sont-elles liées à des faiblesses de l'organisation du service informatique ? Sont-elles liées aux techniques informatiques utilisées?
Contre- mesures	À ce stade, on sait quels sont les vrais risques qui pèsent sur les actifs les plus importants. On sait par où commencer, quoi protéger et contre quoi. Les contre-mesures de sécurité découlent naturellement. Ici-et seulement ici-on entre dans la technique!

2.3 Les coûts de la sécurité

La sécurisation d'une architecture SOA a des coûts assez conséquents. Ces coûts sont relatifs à la performance des applications appelant les services Web. Elle nécessite aussi une connaissance et une compétence en terme des spécifications des services Web.

Les Performances

La sécurisation d'une architecture SAO nécessite des calculs et opérations comme :

- Contrôle d'identité
- Cryptage
- Décryptage
- Authentification
- Autorisation
- Etablissement de relation de confiance
- Vérification des contextes de sécurité
- Signature des messages
- La taille des messages SOAP.

Ces opérations ajoutent un coût significatifs en matière de traitement des demandes de services Web. Ce qui implique une baisse des performances des applications.

Il convient de comprendre donc les besoins en terme de sécurité afin d'éviter l'utilisation de spécifications qui ne font qu'ajouter une couche supplémentaire et sans intérêt pour sécuriser les applications basees sur l'architecture SOA.

Ces baisses ne peuvent être évitées ou contournées que par l'évolution du matériel utilisé au niveau de l'environnement d'exécution.

La veille Technologique

Le domaine de sécurité des services web nécessite la manipulation de technologies et spécifications diverses et variées. Les compétences des administrateurs des serveurs d'applications ou du personnel attaché à la sécurité application doit maîtriser ces technologies et comprendre leur impact. Ceci nécessite aussi une veille technologique continue et sérieuse.

L'intégration et l'interopérabilité

Une architecture SOA est une architecture ouverte, même si cette ouverture se base sur des standards, elle implique plusieurs problématiques :

- Est-ce que les services Web déployés peuvent être exploités par les partenaires ou utilisateurs ?
- Comment faire pour échanger des jetons de sécurité ?
- L'infrastructure de sécurité est-elle fiable et efficace ?
- Comment faire pour intégrer d'autres services nécessitant une sécurité particulière en plus de ce qui est mis en place ?
- Comment gérer l'évolution des standards ?

Il faudra donc une remise en cause permanente de l'existant en terme de sécurité. Cette remise en cause permet de fiabiliser le contexte de sécurité mis en place et d'anticiper toute évolution interne ou externe.

3.1 Des Concepts de base

L'identité

La gestion de l'identité pour les services web est similaires à celle utilisée pour les systèmes d'information qui statue si le consommateur d'un service (une personne, une machine, un programme, un processus, etc) dispose d'un identifiant unique et non ambiguë dont la validité peut être vérifiée. L'identité d'un consommateur de service web est nécessaire afin d'établir une relation de confiance entre le consommateur et le fournisseur de service.

La gestion de l'identité au niveau des services web est plus compliquée à mettre en œuvre parce que les services web peuvent traverser les entreprises via des intermédiaires. Des problèmes d'unicité d'identité peuvent facilement surgir.

Plusieurs initiatives sponsorisées par le « The Liberty Alliance » se focalisent sur la problématique de gestion identitaire pour le domaine d'Internet.

Profile

Un profile est un document qui décrit le modèle de sécurité associé à un type ou à une classe de client. Le client peut être par exemple, un navigateur classique (appelé client passif) capable d'envoyer des requêtes http classiques ou un client actif capable d'envoyer des requêtes basées sur SOAP et de comprendre le résultat.

Claim (Réclamation)

Elle spécifie une déclaration ou une revendication faite par une entité concernant par exemple, un nom, une identité, une clef, un privilège, un attribut, un certificat ...

Jeton de sécurité

Un jeton de sécurité correspond à un ensemble de claims.

Jeton de sécurité signé

Une jeton de sécurité signé est un jeton de sécurité certifié et crypté par une autorité spécifique (par exemple un certificat X.509 ou un ticket Kerberos)

Signature

Une signature est valeur calculée via un algorithme cryptographique d'une donnée. La signature permet

de vérifier que les données n'ont pas été altérées après envoi du signataire.

STS (Security Token Service)

Un STS est un service Web capable de délivrer des jetons de sécurité (utilisé dans le modèle de sécurité avec WS-Trust). Il permet entre autres d'installer une communication de confiance entre des entités ou fournisseurs de services.

Principal

Toute identité utilisatrice peut être appliquée à une personne, machine, service web, etc.

AS (Attribute Service)

Un AS est un service Web capable de maintenir des informations (attributs) de principaux en relation avec un domaine de confiance.

Realm/Domain

Un realm ou domaine représente une unité de sécurité administrable et capable de fournir des informations afin d'établir une relation de confiance entre deux ou plusieurs fournisseurs de services.

Trust (Confiance)

La confiance signifie qu'une entité peut se baser ou avoir confiance en une autre entité pour exécuter un ensemble d'opérations relatives à la sécurité.

Trust Domain (Domaine de confiance)

Un domaine de confiance est un contexte sécurisé et administrable ouvert entre un client et un fournisseur de service leur permettant d'être d'accord si les credentials échangées satisfassent les contraintes de sécurité de chacun.

Fédération

Une fédération consiste en une collection de realms/domaines qui ont établi une relation de confiance. La confiance peut intervenir au niveau de l'authentification ou de l'autorisation.

Fournisseur d'identité (Identity Provider)

Un fournisseur d'identité est une entité qui fait foie de service d'authentification aux utilisateurs de service et aussi aux services. Cette entité est capable de fournir des informations de sécurité sur les profils demandés.

Mapping d'identité (Identity Mapping)

Cette opération consiste à définir une correspondance entre les propriétés associées à une identité entre deux fournisseurs de services ou participant à une chaîne de service web.

Single Sign On (SSO)

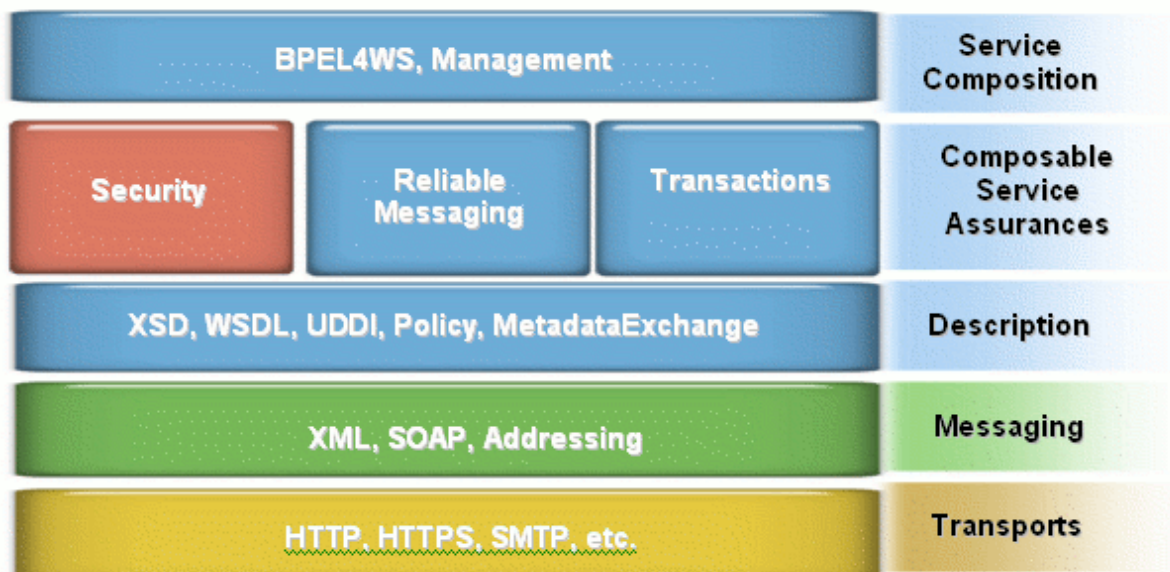
Le Single Sign On est une solution à la problématique d'authentification d'un client dont la transaction appelée met en jeu plusieurs participants. Ce qui évite une phase d'authentification pour chaque fournisseur de service à chaque échange.

Single Sign Off ou Global Logout

Solution permettant de finaliser toutes les sessions ouvertes pour un client donné au sein des fournisseurs de services participant à la transaction en cours.

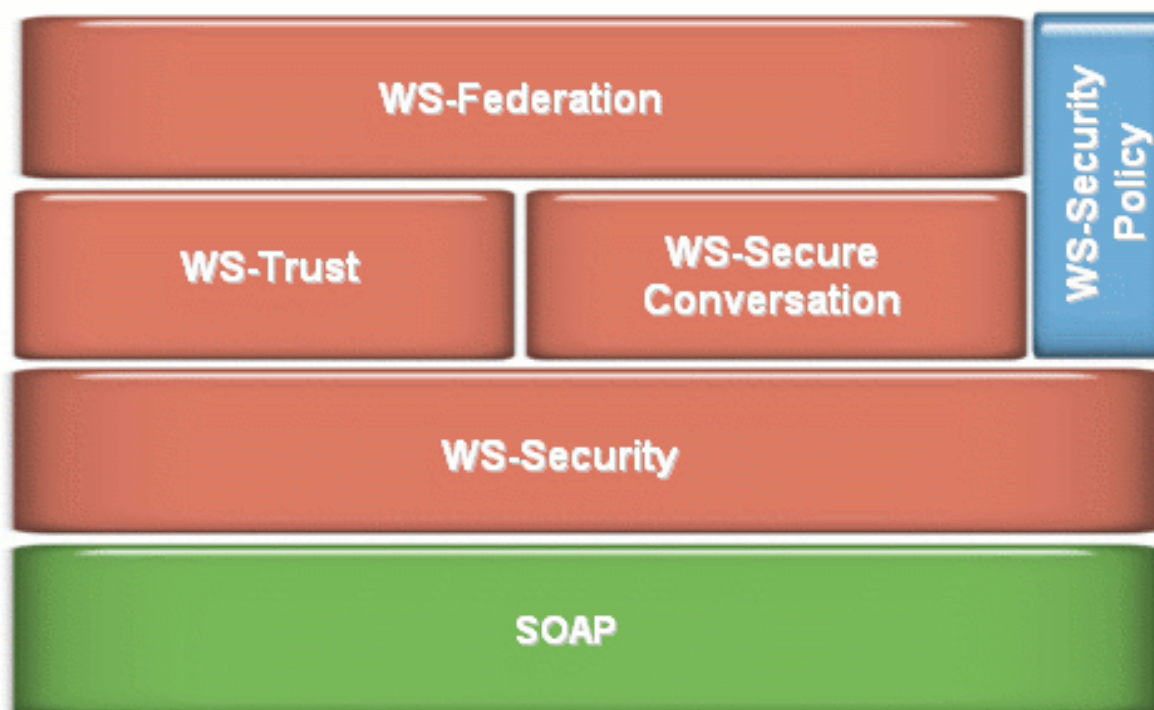
3.2 Un modèle de sécurité

Dans ce chapitre on va décrire un modèle de sécurité faisant intervenir les concepts de l'architecture orientée services, des spécifications WS-* et plus particulièrement WS-Security et la plate-forme J2EE.



La sécurité des services Web fait partie intégrante de la stack des spécifications WS-*.

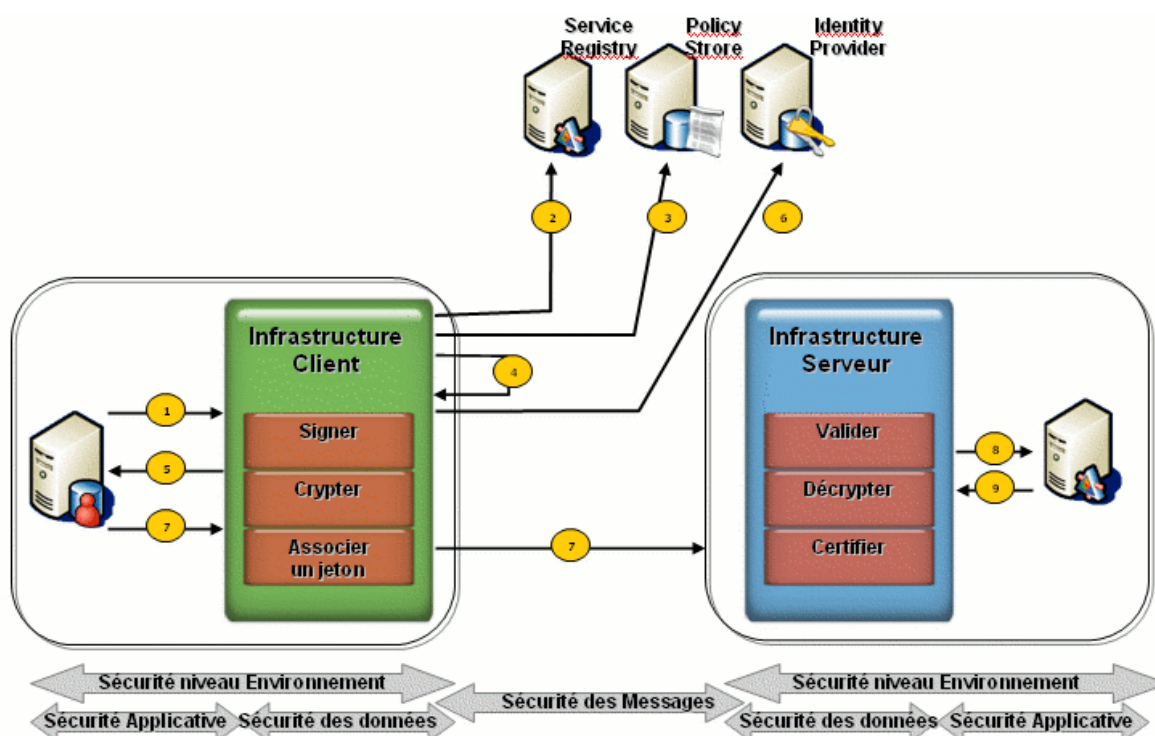
Plusieurs standards lui sont associés. Les plus importants et les plus stables en terme de spécifications sont cités ci-dessous. Pour plus de détails concernant chaque spécification se reporter aux autres livres blancs sur la sécurité niveau données et messages.



Afin de comprendre l'interaction entre les participants à une transaction basée sur un service Web, il conviendrait de dresser la cinématique relative au flux de messages transportés entre l'infrastructure Client et l'infrastructure Serveur. Ensuite seront associées les différentes spécifications, standards et outils associés à chaque étape de ce flux.

Les schémas mettent en jeu un client et un serveur, la cinématique peut être adaptée dans le cas d'un échange avec un intermédiaire ou plusieurs.

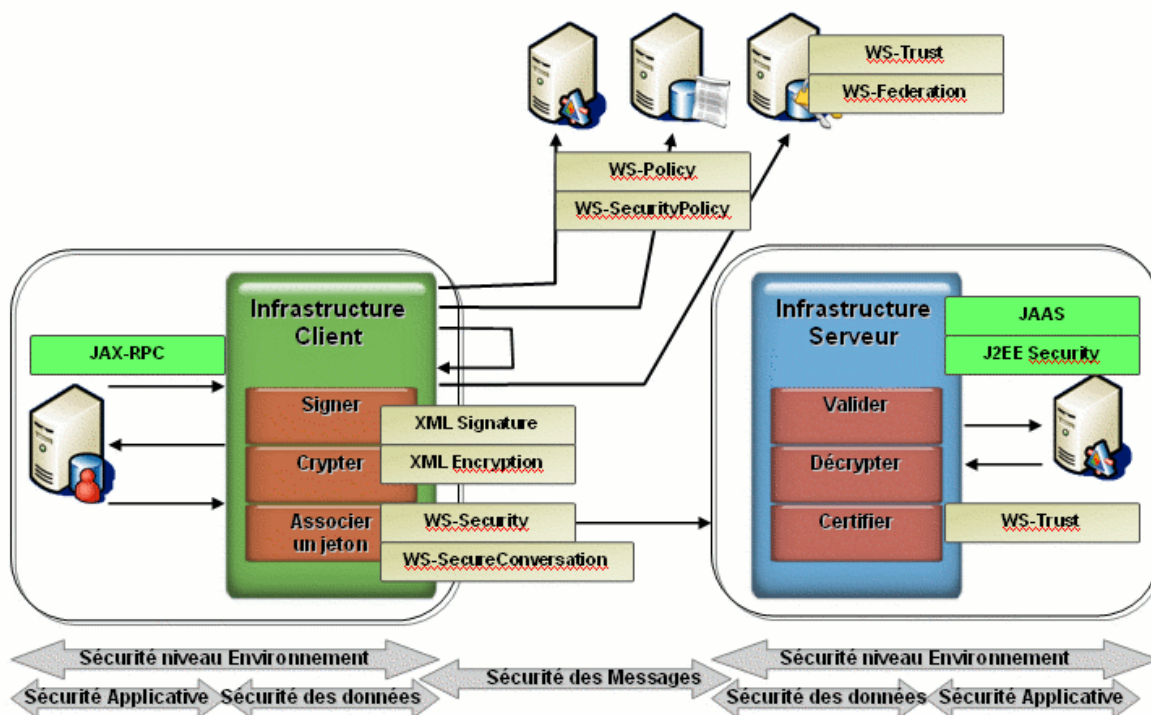
La cinématique ci-dessous ne fait pas référence au contexte de sécurité partagé et constitue un modèle dont la topologie reste simple et compréhensible.



Ci-dessous Le flux des messages

1. Le code client initialise l'environnement relatif à l'infrastructure client en récupérant des informations sur le proxy de connexion et soumet la requête désirée. Dans le contexte de la plate-forme J2EE, le client utiliserait l'api Jax-RPC.
2. Une fois l'infrastructure client alertée de la requête du client, cette dernière associe un processus à cette requête. Le processus correspond à un ensemble de services coordonnés et orchestrés. Pour chaque service, l'infrastructure client se connecte à l'annuaire des services et récupère la définition liée au service en téléchargeant le fichier WSDL représentatif.
3. une fois le fichier WSDL récupéré, l'infrastructure client se connecte à l'annuaire de policy (politique ou stratégies) puis récupérer les stratégies associées à l'appel du service Web. Ces stratégies peuvent contenir des exigences en terme de sécurité.
4. L'infrastructure Client Interprète les stratégies de sécurité de l'invocation du service, par exemple savoir quelle partie du message doit être cryptée. En ayant le fichier de policy, l'infrastructure client identifie le type de jetons de sécurité attendus par l'infrastructure serveur.
5. L'infrastructure client demande au client de spécifier son identité afin de l'autoriser.
6. Une fois l'identité du client validée, un jeton de sécurité est récupère via le fournisseur d'identité.
7. Lors de cette étape, le client constitue le message de la requête SOAP. Il signe le message, crypte les parties sensibles et associe le jeton de sécurité à l'entête du message. Une fois le message sécurisé créé il est enfin envoyé à l'infrastructure du serveur.
8. L'infrastructure serveur s'occupe de valider le message de le décrypter et de vérifier le jeton de sécurité afin d'installer une relation de confiance. Une fois la confiance établie, le service final est invoqué.

Ci-dessous sont associés à chaque étape de cette des technologies, standards et spécifications.



3.3 Questions de sécurité

La sécurisation de l'environnement n'est pas garantie seulement par l'intégrité des messages, la confidentialité et la vérification des jetons associés à un service Web. Il convient aussi de vérifier que les données ne peuvent pas faire l'objet d'attaques. La liste des considérations ci-dessous permet de trouver les points primordiaux qui aideraient à sécuriser un échange de messages. Cette liste n'est pas exhaustive mais je crois qu'elle pose les problématiques les plus répandues.

Interception des messages

Les messages des services Web sont généralement envoyés en plain texte, ce qui est facile sa lecture. Un message intercepté peut facilement être modifié. Des informations malicieuses peuvent donc être insérées au niveau de l'entête ou du corps du message. Ces informations peuvent aussi être des virus.

L'interception des messages peut aussi donner accès à des données confidentielles transportées par le messages ou les fichiers qui y sont attachés.

La solution efficace pour éviter les menaces suite à l'interception des messages est l'utilisation du cryptage ainsi que la signature digitale afin de préserver la confidentialité et l'intégrité des messages.

Usurpation d'identité

La menace d'usurpation d'identité consiste en l'envoi d'un message malicieux avec un jeton de sécurité d'une autre identité habilité à passer le système de sécurité.

Cette menace peut être évitée par la mise en place d'une politique de sécurité mutuelle.

Attaques de dénis de service

Des messages SOAP interceptés peuvent être utilisés afin d'être renvoyés au fournisseur du service correspondant de manière répétitive afin de surcharger l'environnement d'exécution du service web même si l'envoi de message cause des exceptions et erreurs d'exécution.

Cette menace peut être évité par l'installation d'un firewall dédié.

Attaques de Replay

Cette vérification permet de protéger les ressources utilisées par les services web contre les attaques du type : Replay . Cette attaque consiste à utiliser une session utilisateur déjà valide pour envoyer un message. Il existe deux solutions qui permettent d'éviter ce type d'attaques et qui font partie des spécifications WS-Security.

- Nonce : Nonce est un jeton de chiffrement généré de manière aléatoire et est généralement un

sous-élément du jeton UsernameToken. Sans l'utilisation de ce concept, un jeton de nom utilisateur et mot de passe même chiffrés peuvent être détournés et utilisés pour une attaque de type Replay.

- **TimeStamp** : L'élément TimeStamp peut être utilisé afin de marquer le message envoyé par une date de création. En définissant un délai d'expiration des messages, il est plus facile de déjouer les attaques de type Replay.

Le timestamp est représenté via l'éléments <wsu:timestamp>. L'élément nonce est représenté par l'élément <wsse:nonce>.

Ci-dessous un exemple d'utilisation de ces deux concepts :

```

<S:Envelope xmlns:S="http://www.w3.org/2001/12/soap-envelope">
  <S:Header>
    <wsse:Security> <wsse:UsernameToken">
      <wsse:Username>USER</wsse:Username>
      <wsse:Password Type="wsse:PasswordDigest">D2A12DFE8D</wsse:Password>
      <wsse:Nonce>EFD89F06CCB28C89</wsse:Nonce>
      <wsu:Created>2005-06-02T07:41:06Z</wsu:Created>
    </wsse:UsernameToken>
    <wsu:Timestamp> <wsu:Created>2005-06-02T07:41:06Z</wsu:Created>
    <wsu:Expires>2005-06-02T15:41:06Z</wsu:Expires> </wsu:Timestamp>
  </wsse:Security>
</S:Header> </S:Envelope>

```

Important : Les éléments timestamp et nonce doivent tous deux être signés. Dans le cas contraire, ils peuvent être facilement modifiés et ne peuvent plus empêcher les attaques de réexécution (replay).

Protection du jeton Nom d'utilisateur avec un mot de passe

Il est recommandé de ne pas envoyer à un serveur en aval un mot de passe dans un jeton UsernameToken sans protection. Dans ce cas, il est possible d'utiliser les mécanismes de sécurité niveau Transport ou utiliser le chiffrement XML de WS-Security pour protéger le mot de passe.

La méthode de protection à appliquer varie en fonction de l'environnement d'exécution. Toutefois, il reste possible et acceptable d'envoyer un mot de passe à un serveur en aval sous la forme d'un texte normal si l'environnements d'exécution ne présente pas de vulnérabilités à ce niveau.

Protection des jetons de sécurité

Il existe toujours un risque d'attaque avec substitution des jetons. Dans ce scénario, une signature numérique est vérifiée avec une clé définie à partir d'un jeton de sécurité et est incluse dans un message. Si le jeton est remplacé, un destinataire risque d'accepter le message associé à la clé remplacée et de ne pas obtenir ce qu'il attendait. L'une des solutions possibles est de signer le jeton de sécurité (ou les données d'identification uniques à partir desquelles la clé de signature est établie) avec les données signées. Dans certaines situations, le jeton émis par l'autorité habilitée est signé.

Vérification du certificat et l'utilisation de listes de révocation de certificats

Il est recommandé de vérifier si l'authenticité ou la validité de l'identité du jeton utilisée pour la signature numérique est digne de confiance. Pour les jetons X.509, cette procédure implique la vérification du chemin du certificat et l'utilisation d'une liste de révocation des certificats (CRL). Il est possible de mettre en place une solution qui effectue la vérification des certificats et des listes de révocation des certificats à l'aide de la base de données CRL en ligne ou du protocole OCSP (Online Certificate Status Protocol).

5.1 Où trouver les spécifications?

SAML	http://www.oasis-open.org/committees/security/
Security Services TC	http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss
WS-Federation	http://www-106.ibm.com/developerworks/webservices/library/ws-fedworld/
WS-Security	http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss
WS-SecureConversation	http://www-106.ibm.com/developerworks/webservices/library/ws-secon/
WS-SecurityPolicy	http://www-106.ibm.com/developerworks/webservices/library/ws-secpol/
WS-Trust	http://msdn.microsoft.com/library/en-us/dnglobspec/html/ws-trust.asp
XML-Encryption	http://www.w3c.org/Encryption/2001/
XML-Signature	http://www.w3c.org/Signature/

5.2 Ressources de documentation

Ci-dessous des éléments de documentation utiles pour aborder la sécurisation des Services Web.

Basic Security Profile Working Group	http://www.ws-i.org/deliverables/workinggroup.aspx?wg=basicsecurity
Public Key Infrastructure (PKI) (Anglais)	http://www.pki-page.org/
Public Key Infrastructure (PKI) (Français)	http://www.hsc.fr/ressources/cours/pki/index.html.fr
WS-Security Kerberos	http://www.oasis-open.org/committees/download.php/1049/WSS-Kerberos-03.pdf
SAML (Security Assertion Markup Language)	http://www.oasis-open.org/committees/download.php/1048/WSS-SAML-06.pdf
REL (Rights Express Language)	http://www.oasis-open.org/committees/download.php/7347/oasis-____-wss-REL-token-profile-1.0-draft08-clean.pdf
OpenSAML 1.0.1 - an Open Source Security Assertion Markup Language implementation	http://www.opensaml.org/
The XML Apache Security Project	http://xml.apache.org/security/index.html
Ehe Apache Directory Project - Kerberos	http://directory.apache.org/subprojects/kerberos.html
La légion de Bouncy Castle	http://www.bouncycastle.org/fr/index.html
AXIS WSSE Security	http://axis-wsse.sourceforge.net/#home
VeriSign Offers Open Source WS-Security Implementation and Integration Toolkit	http://www.verisign.com/verisign-inc/news-and-events/news-archive/us-news-2002/page_000810.html
FIX : Financial Information eXchange protocol	http://www.fixprotocol.org
IIOP : Internet Inter-ORB Protocol	http://www.omg.org
UDDI : Universal Description, Discovery and Integration	http://www.uddi.org
SPEC	SITE