De l'Utilisation des VPNs pour l'Administration et la Sécurité des Services Web

Lina ALCHAAL[(1)(2)] Vincent ROCA[(2)] Michel HABERT[(1)]

[(1)] *Netcelo S.A., Echirolles, France*
[(2)] *INRIA Rhône-Alpes, Planète project, France*

Ces dernières années les Services Web on commencé à s'imposer du fait de la flexibilité qu'ils offrent. Mais le manque de solutions standardisées pour leur administration est un frein majeur, de même que le fait qu'il n'existe pas de solution simple permettant la sécurisation du trafic entre les différents composants du service web. Parallèlement, les dernières années ont été le théâtre d'une révolution en matière de télécommunications. A l'origine de ce bouleversement se trouve le développement des réseaux privés virtuels (VPN) basés sur l'Internet.

Dans cet article nous présentons un modèle d'architecture hybride qui : (1) correspond bien à la nature dynamique des services web, (2) offre un service d'administration facilement intégrable, et (3) améliore grandement la sécurité des services web grâce à l'utilisation de VPNs.

Mots-clés: `services web, sécurité, VPN, administration dynamique`

1 Introduction

Un Réseau Privé Virtuel (ou VPN, Virtual Private Network) [Kos98] [Ker02] est un réseau de télécommunications bâti sur une infrastructure publique non sécurisée, tel l'Internet, chargé de transférer les flux d'informations entre sites d'une manière totalement sécurisée grâce à l'utilisation de tunnels [Har03]. Dans un précédent article [ARH02] nous avons introduit une approche dynamique pour l'établissement de VPNs entre sites distants, potentiellement gérés par différents fournisseurs d'accès Internet (ou ISP). Cette solution repose sur une nouvelle entité, *l'opérateur VPN*. Celui-ci exploite un centre d'opération, le VNOC (ou Virtual Network Operation Center) qui configure à distance les sites VPN dynamiquement, selon les besoins des utilisateurs.

Les services web prennent leurs origines dans l'informatique distribuée et dans l'avènement du Web (en particulier le commerce électronique sur l'Internet). Ils offrent des services applicatifs publiés, localisés et exécutés sur l'Internet, et accessibles via des protocoles standardisés du web par des entités distantes (applications ou utilisateurs finaux) [MSvM02]. L'adoption de standards comme XML, SOAP, WSDL et UDDI a largement contribué au succès car cela résout le problème d'interopérabilité entre les différentes plates-formes [New02]. Néanmoins des obstacles majeurs à l'adoption des services web restent, à savoir la sécurité et surtout l'administration dynamique des services web.

En fusionnant les services VPNs (et les protocoles de sécurité associés tels IPsec et SSL) dans une architecture des services web, nous introduisons un modèle hybride qui : (1) correspond bien à la nature dynamique des services web, (2) offre un service d'administration facilement intégrable, et (3) améliore grandement la sécurité des services web grâce à l'utilisation des VPNs.

La suite de cet article est organisé ainsi : la section 2 introduit une approche VPN centralisée et dynamique; la section 3 discute les problèmes d'administration et de sécurisation des services web; nous introduisons notre proposition au sein de la section 4, puis rendons compte d'une première évaluation du système en section 5. Finalement nous présentons des travaux associés puis nous concluons.

2 Une Approche VPN Centralisée et Dynamique

2.1 Architecture

Dans [ARH02] nous avons défini une architecture VPN basée sur des tunnels IPsec entre des sites distribués sur l'Internet, ces sites pouvant être un réseau local, une branche d'entreprise, un réseau partenaire, ou même un utilisateur (cas des travailleurs nomades ou à domicile). Les VPNs sont dynamiques et leur topologie varie en fonction de l'ajout ou du retrait d'un site au VPN. *Tous les services fournis* (authentification des sites, ajout et retrait d'un site, configuration) sont réellement *dynamiques*.

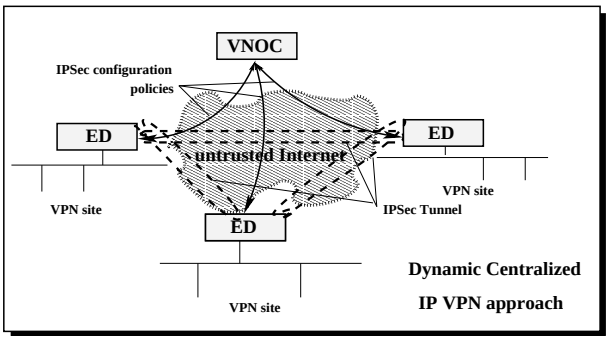

FIG. 1 – *Architecture de l'approche VPN centralisée et dynamique (cas site à site).*

Un deuxième aspect essentiel de cette architecture est son caractère centralisé [†], nécessaire afin d'appliquer les politiques à tous les sites de façon cohérente. Le centre d'administration, ou VNOC (Virtual Network Operation Center), prend en charge l'application des politiques VPNs sur les passerelles VPN (ou Edge Device, ED) des sites. Un seul ED existe pour chaque site VPN et il constitue le point d'entrée et de sortie des tunnels VPNs de ce site. Les EDs jouent un rôle actif dans cette architecture : en particulier ils envoient des requêtes au VNOC pour participer ou quitter un VPN. Ainsi, si un ED demande à participer à un VPN, et si le site de cet ED y est autorisé, le VNOC lui envoie en retour les fichiers de configuration nécessaires. Le VNOC informe également les autres EDs concernés par ce VPN afin d'établir les tunnels, et toutes les communications site-à-site sont alors sécurisées par IPsec.

Pour résumer les caractéristiques de cette approche :

- *Indépendance vis-à-vis des ISPs :* ce service peut être appliqué entre des sites liés à différents ISPs.
- *Approche centralisée :* la présence d'un centre d'administration facilite le contrôle et la configuration des VPNs, ainsi que la maîtrise des services d'accounting et de facturation.
- *Utilisation de briques standards :* le trafic entre les sites VPNs est protégé par IPsec, pendant que le trafic entre un site VPN et le VNOC est protégé par SSL.
- *Approche dynamique :* chaque site envoie dynamiquement des requêtes au VNOC afin de participer à ou de quitter un VPN, selon les besoins des utilisateurs, et le VNOC met à jour immédiatement la configuration du VPN.
- *Plusieurs topologies sont possibles pour les communications site-à-site :* dans cet article seuls les VPNs en étoiles sont considérés, néanmoins d'autres topologies sont envisageables. Les services de groupes de communication (multicast) sont également possibles [ARH02][ARESH03].
- *Administration :* l'authentification et le contrôle d'accès des utilisateurs sont effectués par le VNOC qui offre aussi un outil de configuration et de surveillance des VPNs.
- *Solution opérationnelle :* enfin cette approche est complètement implémentée et disponible commercialement.

[†] Notons qu'à des fins de robustesse, un ou plusieurs VNOCs de secours sont déployés en dehors du réseau de l'opérateur de VPNs.

2.2 Utilisation de SSL et IPsec

IPsec et SSL sont les protocoles de sécurité les plus utilisés sur l'Internet [Pat03][Bar00]. Néanmoins ces deux protocoles présentent de grandes différences [Che02]:

- *niveau d'opération :* IPsec est un protocole de niveau réseau tandis que SSL est un protocole de niveau applicatif. Cette différence est le point de divergence essentiel.
- *périmètre sécurisé :* opérant à des niveaux différents, SSL offre des services de sécurité limités à TCP tandis que IPsec supporte n'importe quel trafic, TCP, UDP ou autre. De même SSL sécurise une application donnée tandis qu'IPsec sécurise plusieurs applications simultanément.
- *support d'installation :* un grand avantage de SSL est l'absence de logiciel supplémentaire côté client, un grand nombre de navigateurs supportant nativement HTTPS (HTTP sur SSL). A l'inverse IPsec requiert à ce jour le déploiement de logiciels spécifiques.

D'un point de vue sécurité, il n'y a pas une grande différence entre SSL et IPsec qui partagent les mêmes algorithmes. Etablir un VPN SSL ou un VPN IPsec dépend alors essentiellement des besoins des utilisateurs [Arr03] et les deux types de VPNs sont utilisés dans notre approche et dans la suite de ce travail.

3 Services Web

3.1 Architecture d'un Service Web

Un service web typique comporte trois éléments (figure 2):

- un fournisseur de services web, qui offre un ou plusieurs services.
- un annuaire de services, où les fournisseurs publient la description des services fournis, et
- un client, qui demande l'exécution d'un service web. Ces clients interrogent l'annuaire, récupèrent les descriptions des services voulus, puis les invoquent.

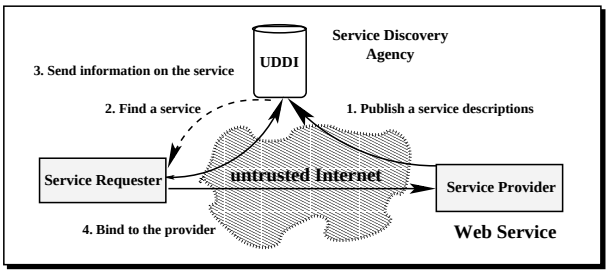

FIG. 2 – *Architecture d'un Service Web.*

L'utilisation de protocoles standards est une des raisons de la réussite des services web. C'est le cas de XML et HTTP, bien connus, mais aussi des protocoles suivants :

- SOAP : Ce protocole définit un modèle d'échange de messages XML via HTTP entre clients et fournisseurs de services web. Ces messages contiennent le nom et les paramètres du service web à exécuter, et en retour contiennent les résultats. En plus, SOAP bénéficie aussi de la tolérance des pare-feux au trafic HTML.
- UDDI : Ce protocole d'annuaire permet de trouver le service web recherché (de façon manuelle ou automatique) [OAS02], mais aussi d'en annoncer la disponibilité. Les clients d'un annuaire UDDI sont soit un fournisseur qui publie ses services web, soit un client qui recherche un service.
- WSDL [W3C01] : Cette norme dérive de XML et décrit l'interface d'utilisation d'un service web (méthodes et propriétés des composants de l'application), le système de communication sous-jacent (un service peut proposer une communication via SOAP, via HTTP GET ou POST, ou via MIME), et ses détails de déploiement.

Les plates-formes des services web sont à ce jour soit "J2EE", soit ".NET". Ces plates-formes sont complexes et nécessitent une administration précise.

3.2 Evolution des Services Web et Challenges Associés

L'évolution de service web a suivi trois phases :

- La première phase a concerné les principaux blocs fonctionnels: XML, SOAP, WSDL, et UDDI. Grâce aux efforts d'organisations de standardisation tels le W3C et le WS-I (Web Service Interoperability Organization), ces protocoles sont maintenant largement matures.
- En revanche la deuxième phase, qui concerne les problèmes de sécurité et de fiabilité, est loin d'être achevée, même si de nombreux efforts sont fait par OASIS (Organization for the Advancement of Structured Information Standards), WS-I et d'autres organisations (section 3.2.2).
- La troisième phase concerne le provisionnement, la surveillance et l'administration des services. Cette phase en est encore à ses débuts, malgré l'urgence.

Plusieurs challenges restent donc ouverts, et nous allons maintenant les détailler.

3.2.1 Challenges Liés à l'Administration

Un bon système d'administration nécessite un ensemble flexible et inter-opérable de primitives permettant de déployer et surveiller les services web. Dans cet article nous nous concentrons uniquement sur les opérations externes d'administration, entre les fournisseurs et leurs clients, à savoir :

- l'identification des clients: chaque client sous contrôle du système possède un identifiant unique.
- l'authentification et autorisation des clients: l'authentification des clients est la première étape, la deuxième consistant à autoriser ou non l'accès aux services.
- les opérations de surveillance: un système de surveillance est essentiel pour connaître à tout moment l'état des services web, la liste des clients de chaque fournisseur de service web, voire le temps moyen pour l'exécution d'un service web.
- le déploiement des fichiers de configurations: des fichiers de configurations sont à déployer pour assurer des communications sécurisées entre clients et fournisseurs.

Malheureusement les standards actuels ne permettent pas de gérer ces aspects.

3.2.2 Challenges Liés à la Sécurité

Les services web n'échappent pas aux éternels problèmes de sécurité. Les flux SOAP étant véhiculés sur le port 80, les pare-feux traditionnels qui ne pratiquent pas d'inspection du contenu, sont à priori incapables de détecter la majorité des attaques. L'essentiel des efforts dans ce domaine ont donné naissance à plusieurs langages de sécurité (mais qui ne sont pas encore mis en oeuvre) :

- Le WS-Security (Web Services Security Language) définit les extensions SOAP pour sécuriser les échanges des messages SOAP.
- XML Signature [Gok02a] définit la syntaxe (compatible XML) permettant de signer électroniquement tout ou partie d'un document (texte, image, plus généralement ressource Web) représentable par une URI (Universal Resource Identifier), et donc notamment tout ou partie d'un document XML pour assurer l'intégrité et la non répudiation des données.
- XML Encryption [Gok02b] spécifie le processus pour le cryptage de données et la représentation du résultat dans XML. Ces données peuvent être des données arbitraires (y compris un document XML), un élément XML ou le contenu d'un élément XML.
- XKMS [Gok02c] traite des services de gestion des clés et certificats (utilisé conjointement avec XML Signature).
- XACML [OAS03b] spécifie les politiques utilisées pour accéder aux documents XML selon leur contenu, le sujet et l'action (lire, écrire, créer, effacer).
- SAML [OAS03a] est un dialecte XML pour exprimer des informations d'authentification sur des systèmes sécurisés et pour définir des droits utilisateurs génériques.
- SPML [OAS03c] vise à normaliser le mode d'invocation et d'administration d'une plate-forme de provisionnement.

Tous ces standards ont pour but de sécuriser les données XML, mais ils ne visent pas à sécuriser ni le service web lui-même, ni le trafic du service web. Aussi le trafic SOAP doit-il être transporté sur HTTP/SSL ou TLS (Transport Layer Security) pour une sécurité de bout-en-bout [DA99]. IPsec [KA98] permet lui aussi de sécuriser les messages web dans un service web, avec une approche ici point-à-point.

Mais aucune de ces solutions n'offre à notre avis suffisamment de souplesse pour permettre d'administrer les politiques de sécurité entre les différents éléments du service web, tout en évitant les conflits d'interopérabilité. De plus la solution doit être dynamique et permettre de générer les politiques de configurations sans intervention humaine.

4 Notre Approche : Services Web VPN

4.1 Architecture

Notre architecture est basée sur l'intégration des services VPN et services web. On y retrouve en particulier tous les composants d'un service web ordinaire. *Un centre d'administration centralisé, ou MOP (Management Operation Point), prend en charge les fonctions d'administration des services web. Cependant le MOP n'intervient pas dans les processus des services web eux-mêmes.* Le MOP contient deux éléments :

- un annuaire UDDI, qui contient les descriptions des services web, et
- un VNOC (section 2.1) qui assure les opérations d'administration.

Notre architecture crée un service web spécifique aux opérations d'administration. On doit donc distinguer :

- les services web "commerciaux" (ou *business web services*), fournis par les fournisseurs des services web à leurs clients, et
- le service web d'administration (ou *management web service*), fourni par le MOP à chaque élément des services web commerciaux.

Il s'agit donc d'une application récursive (!) de la notion de service web. Nous trouverons donc à la fois un fichier WSDL pour décrire les services d'administration fournis par l'interface d'administration, et un fichier WSDL pour décrire les services web commerciaux. A chaque fois SOAP est utilisé pour les échanges de messages.

La sécurité des échanges avec le MOP est assurée par SSL (section 2.1). Afin de simplifier les opérations administratives liées aux services web commerciaux, ainsi que les problèmes de sécurité. Le MOP crée dynamiquement un *VPN pour chaque service web commercial*. Chaque VPN a une topologie en étoile et contient :

- un fournisseur de services web : ce fournisseur est le centre du VPN,
- les clients du service web commercial : ces clients sont à la périphérie du VPN.

Dans la suite nous utiliserons le terme "service web VPN" pour dénoter le service web commercial associé à un VPN. Dans ce travail nous associons un VPN à chaque service web. Des extensions, qui ne sont pas considérées içi pourraient consister à n'avoir qu'un VPN sur la plate-forme, partagé par tous les services web qu'elle offre. Il faudrait alors probablement ajouter des mécanismes AAA sur cette plateforme, ce qui va à l'encontre de notre approche.

4.2 Les Deux Phase d'Etablissement d'un Service Web VPN

Nous présentons maintenant plus en détails les deux phases d'établissement d'un service web VPN, et les messages SOAP utilisés.

4.2.1 Phase de Signalisation et d'Installation du VPN

Cette phase concerne le service web d'administration. On identifie deux types de messages :

- les messages échangés entre le MOP et un fournisseur du service web : le fournisseur envoie au MOP un message SOAP du type *Publish* (1), avec un fichier WSDL décrivant en détails son interface commerciale et les services fournis par celle-ci. Le message SOAP est traité par le VNOC qui vérifie

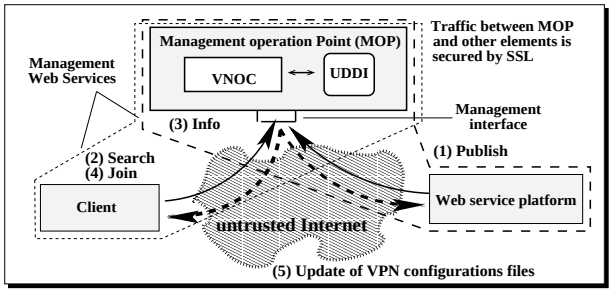

FIG. 3 – *Création d'un Service Web VPN: phase de signalisation.*

l'identité du fournisseur et son autorisation (on suppose qu'existe une liste des fournisseurs qui ont le droit d'utiliser le service d'administration). Si la réponse est positive, le VNOC enregistre le fichier de description des services dans l'annuaire UDDI, et créé un VPN centré sur ce fournisseur.

– les messages échangés entre le MOP et les clients : le client envoie un message SOAP du type *Search* (2) au MOP afin de trouver des informations sur un service web. Une fois le client identifié et autorisé par le VNOC, un autre message SOAP du type *Info* (3) est retourné au client avec le fichier de description du service demandé et l'identifiant du service web VPN associé. Le client envoie alors un second message SOAP du type *Join* (4) au MOP afin de participer au service web VPN. Le VNOC ajoute alors l'identifiant du client aux autres membres du VPN et met à jour les fichiers de configuration du VPN afin qu'un tunnel puisse être établi entre le nouveau client et le fournisseur de services (5).

Deux autres messages SOAP existent :

– *Leave* : afin de quitter un ou plusieurs services web VPN, le client envoie ce message au MOP, avec l'identifiant des VPNs. Le MOP met alors à jour les fichiers de configuration côté fournisseurs afin de désactiver les tunnels correspondants au client.

– *GetVPN* : ce message sert à récupérer la liste d'identifiants des service web VPNs auquel fait parti un client donné.

Les identifiants fournisseurs et des clients des services web VPNs sont stockés dans une base de données de l'opérateur VPN, après inscription au service. Cette génération des identifiants constitue une phase initiale incontournable dans laquelle on vérifie l'identité de chacun (par mail ou téléphone) ainsi que le niveau de sécurité désiré.

4.2.2 Phase de Transfert de Données

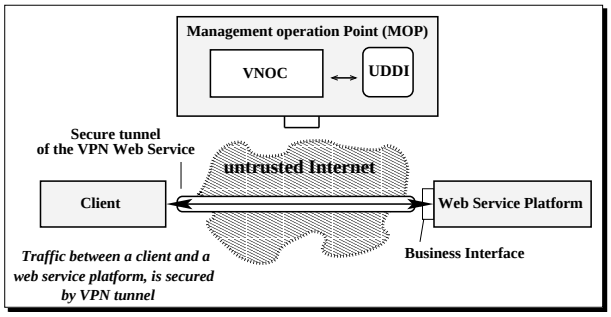

FIG. 4 – *Création d'un Service Web VPN: phase de transfert de données.*

Après déploiement des fichiers de configuration du VPN sur le client et le fournisseur, et obtention du fichier de description du service web, le client est désormais prêt à exécuter les services web fournis. L'établissement du tunnel est fait dès que le client envoie le premier message à l'interface commerciale pour

exécuter le service web. Ce tunnel restera ensuite actif jusqu'à ce que le client envoie un message *Leave* au MOP.

Selon les politiques de sécurité déployées dans la phase précédente, le trafic entre le client et le fournisseur de service web sera sécurisé par IPsec ou SSL. Le choix entre ces deux technologies dépend des niveaux de sécurité demandés par les clients au moment de leur inscription au service VPN. D'autre part l'utilisation d'IPsec ou de SSL n'empêche pas l'utilisation d'autres technologies comme XML signature ou SAML qui ajoutent un niveau de sécurité supplémentaire.

Un avantage de l'utilisation du service web VPN est qu'il ne contient que les clients qui ont été autorisés à exécuter un service web. Les fournisseurs de services web sont ainsi déchargés des taches d'authentification/autorisation/accounting de leurs clients, ces aspects étant gérés par le VNOC.

5 Implémentation et Evaluation des Services Web VPN

Nous avons implémenté une grande partie de cette approche à des fins d'évaluation et validation de l'architecture. Dans ce prototype le MOP contient uniquement un VNOC (pas de base UDDI) et est construit sur une plate-forme J2EE (qui gère les opérations d'administration). Les clients utilisent un programme Perl afin de générer des messages SOAP et les envoyer à l'interface commerciale. L'interface elle-même tourne sur un serveur Apache Java.

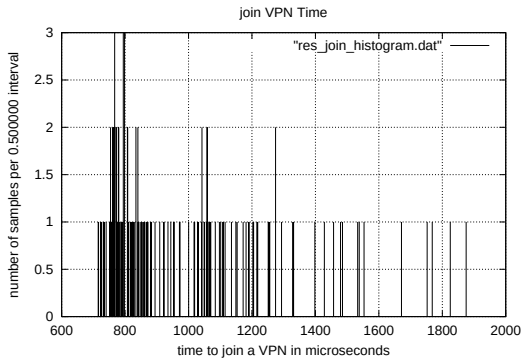

FIG. 5 – *Temps de traitement d'un message JOIN (histogramme).*

Nous avons mesuré le temps nécessaire au traitement d'opérations d'administration sur le VNOC au sein d'un environnement opérationnel. Nous nous sommes concentrés sur trois opérations : *Join*, *Leave* et *GetVPN*. Ces tests ont montré que le traitement d'un message *Join* (figure 5) et *Leave* nécessite respectivement 978 microsecondes et 938 microsecondes en moyenne. Ceci est deux fois plus long que le traitement d'un message *GetVPN* qui ne prend que 492 microsecondes. Cela s'explique par le fait que les opérations *Join*/*Leave* impliquent des modifications au sein de la base de données VPN gérée par le VNOC.

Un problème qui affecte les performances, tout particulièrement lorsque le nombre de clients augmente, est le mécanisme d'authentification au sein de SOAP. Nous avons utilisé le modèle SOAP::Lite au sein du programme client Perl. Puisque SOAP a été développé pour n'envoyer d'informations d'authentification que sur demande expresse du serveur, le premier message émis par le client n'inclue aucune information d'authentification. L'interface d'administration se doit donc de renvoyer un message d'erreur "401: authentication error" au client, qui répond alors avec les données demandées. Nous avons résolu ce problème en obligeant le client à ajouter systématiquement des informations d'authentification au premier message SOAP généré.

Nous avons alors mesuré le temps requis pour mettre en place un service web VPN. Ce temps est l'intervalle s'écoulant entre la réception d'un message *Join* d'un client, et la mise à jour des fichiers de configuration chez le client. Nous avons découvert que ce temps est en fait fortement lié à certains paramétrages actuels du VNOC qui ajoutent un délai additionnel Δ au processus de mise à jour. Dans nos tests (figure 6), ce délai Δ varie entre 10 et 20 secondes, ce qui conduit à un temps moyen de mise à jour de 24 secondes.

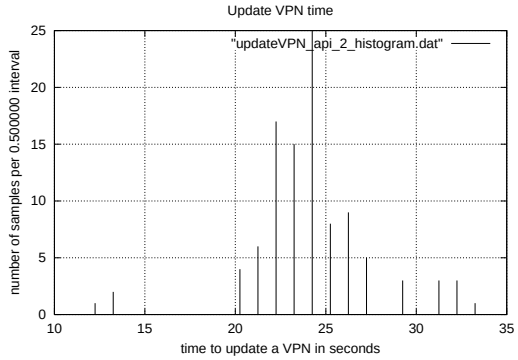

FIG. 6 – *Mise à jour d'un service web VPN sur le site client (histogramme).*

C'est le délai que doit attendre un client avant de participer à un service web VPN. Cela peut paraître long, mais (1) une fois le client connecté, alors l'invocation du service est seulement limitée par la plate-forme du service web, et non pas le VPN associé; de plus (2) nous travaillons sur ce point afin de réduire cette latence.

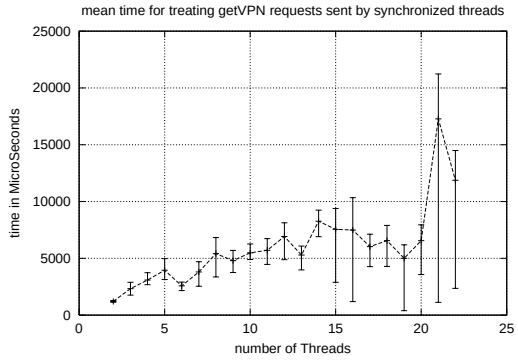

FIG. 7 – *Temps de traitement de messages GetVPN envoyés simultanément par plusieurs clients.*

Nous avons finalement étudié le passage à l'échelle de notre solution. Le figure 7 montre le temps de traitement de messages SOAP *GetVPN* lorsque le nombre de clients augmente. Les résultats sont bons jusqu'à environ 20 clients simultanés, puis le temps moyen de traitement augmente largement. Une première raison à ce comportement est le fait que pour chaque message *GetVPN* reçu, la base de données VPN doit être accédée. Comme cette base est partagée, le nombre d'accès concurrents est limité et certains messages ne peuvent être traités de suite. Une deuxième raison probable est l'impact négatif de la plate-forme Java/J2EE sur les performances, mais ceci nécessite un approfondissement.

Nos expériences ont montré que le niveau de performances global est largement impacté par différents aspects techniques. Nous sommes confiants que les performances peuvent encore largement être améliorées et nos efforts dans ce domaine complexe se poursuivent. Par conséquent cette section doit davantage être vue comme une première évaluation d'un prototype, et non comme une étude de performances d'une solution optimisée.

6 Autres Travaux dans le Domaine

Un but de l'informatique distribuée est de permettre à une application donnée d'accéder à une fonction d'une autre application sur une machine distante aussi simplement que si l'appel était local, indépendamment des plates-formes et des langages utilisés. Trois standards sont aujourd'hui utilisés : CORBA/IIOP

(Common Object Request Broker Architecture / Internet Inter-ORB Protocol), DCOM (Distributed Component Object Model) et RMI (Remote Method Invocation). Ces modèles offrent des mécanismes de récupération d'espace mémoire, de sécurité, de gestion du cycle de vie des objets, mais restent complexes, peu compatibles avec les pare-feux, et difficilement interopérables entre eux. Ils restent donc souvent confinés à l'intérieur des entreprises.

Les services web résolvent ce problème et tirent le meilleur profit de l'Internet en tant que plate-forme de distribution d'informations ubiquitaire et simple d'accès. Mais les problèmes de sécurité sont encore un obstacle important à leur adoption, et en dépit des efforts fournis aucun standard unique n'existe. De nombreux documents ont été écrits dans ce domaine, mais la vaste majorité d'entre eux restent purement théoriques. Le document WS-Security, qui est développé au sein de l'organisme de standardisation OASIS, devrait être publié d'ici quelques mois. L'organisme WS-I (Web Services Interoperability) a été formé récemment afin de promouvoir des standards ouverts pour l'interopérabilité des services web, et il devrait fournir des spécifications de sécurité.

Concernant les aspects d'administration, aucun effort sérieux n'a été fait, et la plupart sont des approches propriétaires émanant des principaux fournisseurs. Récemment un nouveau comité technique au sein d'OASIS a été créé, le Web Services Distributed Management (WSDM) Technical Committe, afin de faire le lien entre OASIS et d'autres organismes de standardisation tels le W3C Web Services Architecture Working Group et le Distributed Management Task Force (DMTF). Ceci inclue l'utilisation des technologies service web afin d'administrer des ressources distribuées. La première spécification Web Services Distributed Management (WSDM) V1.0 est attendue début 2004.

Outre ces organismes, plusieurs compagnies telles Microsoft, Actionl, Amberpoint, ont dors et déjà compris l'urgence des besoins en matière de solutions d'administration. Ainsi Microsoft doit livrer sa nouvelle solution d'administration pour .Net, Microsoft Operations Manager MOM 2004, en été 2004.

Enfin les précédentes sections ayant dors et déjà discuté avec force de détails plusieurs travaux liés à la sécurité des services web, nous ne les reprendrons pas ici.

7 Conclusions

Du fait de leur flexibilité et simplicité, les services web ont connu un intérêt croissant durant ces dernières années. Cependant la nature distribuée et dynamique des services web nécessitent des fonctions d'administration avancées. Les VPNs jouent un rôle prédominant dans l'architecture que nous proposons afin d'administrer et sécuriser ces services, puisque les VPNs offrent naturellement l'infrastructure au dessus de laquelle les services web commerciaux peuvent être déployés. La gestion centralisée, au sein d'un "Management Operation Point", ou MOP, assure les fonctions d'authentification et autorisation des clients, la gestion des politiques, et la configuration des tunnels de sécurité. Par conséquent le MOP décharge les plate-formes de service web de nombreux aspects lourds à gérer.

Ce travail est soutenu, fait assez rare vu la complexité de l'ensemble, par un prototype. Nous estimons avoir dors et déjà établi la faisabilité de notre approche. Cependant atteindre un excellent niveau de performances requiert la maîtrise de nombreux composants, une tache hautement complexe. Si nous ne sommes pas à ce jour satisfaits par certains points, les problèmes sont maintenant identifiés et nous continuons nos effort quant à l'optimisation globale du système.

Références

[ARESH03] L. Alchaal, V. Roca, A. El-Sayed, and M. Habert. A vprn solution for fully secure and efficient group communications. July 2003. 8th IEEE Symposium on Computers and Communications (ISCC'03), Kemer - Antalya, Turkey.

[ARH02] L. Alchaal, V. Roca, and M. Habert. Offering a multicast delivery service in a programmable secure ip vpn environment. October 2002. Fourth International Workshop on Networked Group Communication (NGC'02), Boston, USA.

[Arr03] Array Networks Inc. *SSL VPN vs IPSec VPN*, January 2003. white paper.

[Bar00] J. Barrett. *A Response to the Feature on IPv6 vs SSL*. Root Prompt Org., June 2000. http://rootprompt.org.

[Che02] Check Point Software Technologies Ltd. *IPSec Versus Clientless VPNs for Remote Access*, September 2002. white paper, http://www.checkpoint.com.

[Cla02] J. Clabby. *What are Web Services*. Inc. InformIT Division, September 2002. http://www.informit.com.

[DA99] T. Dierks and C. Allen. *The TLS Protocol Version 1.0*, January 1999. IETF Request for Comments, RFC 2246.

[GLH+00] B. Gleeson, A. Lin, J. Heinanen, G. Armitage, and A. Malis. *A Framework for IP based Virtual Private Networks*, February 2000. IETF Request for Comments, RFC 2764.

[Gok02a] S. Gokul. *XML Digital Signatures*. Inc. InformIT Division, August 2002. http://www.informit.com.

[Gok02b] S. Gokul. *XML Encryption*. Inc. InformIT Division, August 2002. http://www.informit.com.

[Gok02c] S. Gokul. *XML Key Management (XKMS)*. Inc. InformIT Division, September 2002. http://www.informit.com.

[Har03] J. Harrison. *VPN Technologies - a comparison*. Data Connection Ltd., February 2003. http://www.dataconnection.com.

[Hou01] L. Houston. *SOAP Security Issues*. Sun com., December 2001. http://sunonedev.sun.com/building.

[KA98] S. Kent and R. Atkinson. *Security Architecture for the Internet Protocol*, November 1998. IETF Request for Comments, RFC 2401.

[Ker02] Z. Kerraval. *The Evolution of Virtual Private Networks*, October 2002. white paper, Yankee Group.

[Kos98] D. Kosiur. *Building and Managing Virtual Private Networks*. John Wiley & Sons Inc., ISBN 0-471-29526-4, 1998.

[Mit02] N. Mitra. *SOAP Version 1.2 Part 0: Primer*. W3C Working Group, December 2002. work in progress.

[MSvM02] V. Machiraju, A. Sahai, and A. van Moorsel. Web service management network: An overlay network for federated service management. August 2002. IEEE/IFIP IM 2003, Colorado Springs, USA.

[Net] Netcelo S.A. http://www.netcelo.com.

[New02] E. Newcomer. *Understanding Web Services: XML, WSDL, SOAP, and UDDI*. Addison Wesley Professional, ISBN 0-201-75081-3, 2002.

[Ngh03] A. Nghiem. *The Basic Web Services Stack*. Inc. InformIT Division, February 2003. http://www.informit.com.

[OAS02] OASIS Standard. *UDDI Spec Technical Committee Specification Version 3.0*, July 2002.

[OAS03a] OASIS Standard. *Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) v1.1*, September 2003.

[OAS03b] OASIS Standard. *eXtensible Access Control Markup Language (XACML) version 1.0*, February 2003.

[OAS03c] OASIS Standard. *Service Provisioning Markup Language (SPML) Version 1.0*, June 2003.

[Pat03] A. Patrick. *SSL VPNs: The next hot mantra*. Network Computing company, February 2003. http://www.nc-india.com/features/.

[W3C01] W3C Standard. *Web Services Definition Language (WSDL)1.1*, March 2001.

[W3C02] W3C Working Group. *Web Services Architecture*, November 2002. work in progress.