

Web Service : Réalisation d'un Extranet de l'Enseignement Supérieur et de la Recherche

Dominique Verez
Groupe Logiciel Universités - Recherche
Centre de Ressources Informatiques
Université des Sciences et Technologies de Lille
59655 Villeneuve d'Ascq Cedex
Dominique.Verez@univ-lille1.fr
Date : 13 octobre 2003

Résumé

300 000 postes, ce chiffre à lui seul donne la démesure du défi : réaliser un Extranet fédérant l'ensemble de ces postes et le rendre accessible à tous ceux qui le désirent.

Notre communauté Enseignement Supérieur Recherche est répartie administrativement en 241 établissements, physiquement sur des centaines de sites, et au niveau réseaux sur des milliers de domaines (au sens DNS).

L'utilisation de Web Services a permis de réaliser une solution pertinente et élégante pouvant être appelée de partout, depuis n'importe quel serveur Web tournant sur n'importe quel système.

Mots clefs

Web Service, Extranet, DNS.

1 Le contexte : un cadre et des chiffres

La Cellule Administrative des Ressources Informatiques (CARI) de la Direction de la Recherche (DR) du Ministère de la Jeunesse, de l'Education nationale et de la Recherche, Ministère Délégué à la Recherche et aux Nouvelles technologies, gère les ressources informatiques des 241 établissements (Universités, Ecoles, IUFM, Organismes de Recherche publics) qu'elle fédère et a créé en 1993 un pôle spécifique pour tout ce qui concerne les logiciels : le Groupe Logiciel Enseignement Supérieur – Recherche (appelé plus simplement Groupe Logiciel). Ce groupe composé de 35 membres représente l'ensemble des 241 établissements et travaille en leur nom.

Le Groupe Logiciel se réunit régulièrement pour recevoir, à tour de rôle, les éditeurs avec lesquels les établissements travaillent. L'objectif est d'obtenir les meilleurs tarifs et les meilleures conditions en se présentant unis face à eux.

Les bénéficiaires des accords sont l'ensemble des personnels administratifs, des enseignants, des chercheurs et dans une certaine mesure les étudiants de nos établissements. La totalité du parc informatique concerné représente 300.000 postes de type PC auxquels s'ajoutent nombre de serveurs.

Cette introduction, finalement assez brève, n'est pas donnée par hasard, car outre le fait qu'elle définit le cadre d'activité concerné elle donne les chiffres qui sont la clé de cet exposé.

2 L'objectif : communiquer en respectant la confidentialité

Un petit groupe qui travaille pour une large communauté en négociant des accords, doit non seulement communiquer sur ses activités, mais il doit le faire en respectant la confidentialité avec les éditeurs avec lesquels il négocie.

Trois niveaux de communication sont à considérer :

1. Communications internes au Groupe Logiciel.
2. Communications vers les Correspondants Logiciels.
3. Communications vers la Communauté Enseignement Supérieur – Recherche, le niveau qui nous concerne ici particulièrement.

Reste un point, celui des communications vers « le monde Internet », c'est-à-dire vers l'ensemble des personnes qui n'appartiennent pas à la communauté Enseignement Supérieur – Recherche ; il est facile à résoudre car, pour simplifier, on peut considérer que ces personnes extérieures à notre communauté ne doivent avoir aucun accès !

Pour bien comprendre les différences de communication et les réponses techniques adaptées, il faut regarder un peu plus en détail et brièvement ce qui distingue les trois « communautés ».

2.1 Communications internes au Groupe Logiciel

Il s'agit de faire circuler des documents de travail, des pièces de marché public, des protocoles d'accord, des tarifs en cours de négociation. Il faut également maintenir et gérer un site Web sécurisé... Ce sont des données particulièrement sensibles qui nécessitent authentification forte et cryptage.

Pour l'authentification nous utilisons des certificats émis par les Infrastructures de Gestion de Clés (IGC) du Comité Réseau des Universités (CRU) et du Centre National de la Recherche Scientifique (CNRS).

Un double usage des certificats :

1. pour l'échange de courriers électroniques signés (voire cryptés dans certains cas). Notons au passage une difficulté liée au cryptage : pour pouvoir crypter un échange avec une personne, il faut disposer de sa clé publique, autrement dit chaque membre du Groupe Logiciel doit disposer sur son poste des 34 clés publiques des autres membres, ce n'est pas si simple...
2. pour l'accès à un site Web sécurisé, accessible uniquement sur présentation d'un certificat, une correspondance est établie entre l'identité de la personne et un compte sur le serveur, les 35 « comptes » des personnes membres du Groupe Logiciel sont regroupés dans un « groupe » particulier. Les accès des répertoires et des fichiers sont donnés à ce groupe précis. Le serveur Web utilise le protocole *https* pour encrypter les transmissions.

2.2 Communications vers les Correspondants Logiciels

Il y a un « Correspondant Logiciel » par établissement et un suppléant ; ils sont désignés officiellement par leur établissement auprès de la CARI.

Les correspondants logiciels sont non seulement des relais mais également de véritables interfaces entre les personnes de leur établissement et le Groupe Logiciel. Ils ont accès à des contenus sensibles (tarifs, négociations, contacts chez l'éditeur, support technique...) et peuvent télécharger depuis un serveur de télédistribution les produits de certains éditeurs (images ISO des CD).

Nous utilisons également des certificats, avec quelques différences cette fois : ils sont plus nombreux, 480, et émanent de nombreuses IGC diverses, internes aux établissements (de Recherche notamment) ou externes du commerce. Vu le nombre, il n'est plus question de pouvoir faire de l'échange de mails cryptés.

Les certificats servent donc à accéder à un autre site Web sécurisé de la même manière que précédemment. Pour accéder au site Web de téléchargement, qui contient plus de 2.000 images ISO de CD, il a fallu prendre des mesures de sécurité particulières concernant l'encryptage, nous ne pouvons autoriser le transfert de ces produits en clair sur le réseau. Le serveur de fichier utilisé est de type NAS (3 Téraoctets), on lui a adjoint une carte matérielle (Broadcom BCM5820) pour réaliser l'encryptage à la volée lors des transferts SSL. Contrairement aux applications classiques où l'on encrypte un volume de données restreint, un numéro de carte bleue par exemple, il s'agit d'encoder cette fois des images ISO faisant 650 Mo, d'où l'utilisation de ce matériel de cryptage réseau.

2.3 Communications vers la Communauté Enseignement Supérieur – Recherche

Pour communiquer entre le Groupe Logiciel et les utilisateurs des établissements, autrement dit vers tous ceux qui font partie de notre communauté Enseignement Supérieur – Recherche, c'est cette fois l'effet de masse qui prime, avec une problématique importante : comment distinguer aisément cette catégorie du reste de la population de l'Internet ?

Un point est essentiel à considérer, il conditionne toute la suite de cet exposé, c'est la nature des documents ou plus largement des informations à protéger. Il s'agit cette fois de pièces nettement moins sensibles et les mesures de protection à prendre sont moins strictes. Avouons que ça tombe bien, car nous ne saurions gérer aisément plus d'un million de certificats...

Les documents placés sur le site Web sont cette fois plus « informatifs » ils s'adressent à tout à chacun et non plus à des professionnels. Les différents produits des éditeurs sont détaillés, des livres blancs sont accessibles, différents liens renvoient sur les sites des éditeurs. Nous détaillons les procédures de commande et les personnalisons ; comme une personne (en fait une machine) est identifiée par son nom DNS, nous savons de quel établissement elle provient et nous

pouvons lui donner des informations sur mesure : ses correspondants dans l'établissement, les sites Web locaux mis à sa disposition, etc. C'est pourquoi ne désirons pas en faire un site Web grand public, nous souhaitons le réserver à notre population.

La communication doit se faire uniquement via le Web et il n'est pas simple de restreindre l'accès d'un serveur à quelques 300.000 postes répartis sur des milliers de domaines (au sens DNS) ; ces postes sont situés dans les différents Intranet des établissements et il faut les regrouper dans un Extranet commun.

Pour résoudre le problème il faut également prendre en compte un certain nombre de points capitaux : la sacro-sainte autonomie de nos établissements, la rareté des plages IP disponibles, l'éclatement des établissements, l'historique, etc.

3 Réalisation de l'Extranet

Nous devons d'abord redéfinir ce qu'est un Extranet car, au fil du temps, le terme a été galvaudé pour finalement prendre des sens divergents. Pour nous un Extranet se comprendra dans son sens original : l'agrégation d'un certain nombre d'Intranet fédérant des populations ayant un intérêt commun.

L'Extranet de l'Enseignement Supérieur et de la Recherche correspond-il à un besoin ?

Le Groupe Logiciel a certes ses propres besoins de communication vers la communauté, mais il est loin d'être le seul. Différents besoins ont été identifiés, par exemple pour un serveur Web Apache sous Linux situé à Lyon ou pour un serveur IIS sous Windows 2003 situé à Strasbourg. Pouvoir être sûr d'adresser notre population, et uniquement elle, correspond à un vrai besoin et suscite des idées...

Le problème est ensuite de trouver la meilleure façon de réaliser un tel Extranet, comment faire pour fédérer des milliers de petits Intranet, comment faire pour rendre ce service de manière large et universelle ?

L'apparition des Web Services a été le déclic : réaliser un Web Service qui puisse être appelé de partout, depuis n'importe quel logiciel tournant sur n'importe quel système. Réaliser une « boîte noire » à destination des sites Web, qui puisse être appelée en une ligne, simplement, avec un minimum de paramètres.

Le service à rendre est assez simple : une personne désire se connecter sur un serveur Web qui ne veut être accessible qu'à notre seule communauté, cette personne le fait à partir d'une machine identifiée par son numéro IP. Le serveur Web décide alors de contacter le Web Service qui gère notre Extranet, il passe en paramètre le numéro IP de la machine candidate, le service Web lui répond en indiquant si cette machine est oui ou non dans notre Extranet, si elle fait partie de notre communauté ou non. Le serveur Web accorde ou refuse l'accès en conséquence.

Mais la simplicité apparente de la consommation du Web Service cache un traitement beaucoup plus complexe qui nécessite notamment l'utilisation d'une base de données, car certains usages demandent de retourner beaucoup plus d'informations ! Par exemple, quel est l'établissement concerné, quelles sont les coordonnées d'un certain nombre de correspondants au sein de l'établissement, etc.

Les demandes d'usage s'intensifient, non seulement des établissements ou des équipes de recherche, mais également des éditeurs et des distributeurs veulent maintenant offrir à notre communauté des sites Web spécifiques et pourront aisément le faire en appelant « simplement », « naturellement », un Web Service que nous mettrons à leur disposition.

Précisons de suite que le service n'est pour l'instant disponible qu'à des fins d'évaluation, sa mise en œuvre complète et opérationnelle se fera début 2004.

4 Considérations générales

Un certain nombre de principes ont été discutés avant de réaliser le projet. Ce sont des considérations essentielles pour atteindre les objectifs fixés :

1. Une machine qui se connecte sur un serveur Web est toujours identifiable par son numéro IP, c'est ce numéro qui sera la clé de toute l'identification. Ceci, même si le consommateur du Web Service connaît déjà le nom DNS de la machine.

2. Une machine qui ne dispose pas d'une entrée DNS est considérée comme ne faisant pas partie de notre Extranet. Il n'y a aucune alternative à ce point, toute machine de notre communauté (en dehors également) devrait disposer d'une entrée dans un DNS.
3. Nous sommes conscients qu'un numéro IP peut être falsifié, nous acceptons ce risque et les consommateurs du Web Service doivent également en être conscients. Cependant afin de le minimaliser nous recherchons le nom DNS correspondant à ce numéro IP, puis, nous faisons une recherche inverse. En cas de non concordance nous considérons que cette machine ne fait pas partie de notre Extranet, nous renvoyons un code d'erreur spécifique et le consommateur du Web Service est alors libre de traiter différemment ce cas particulier.
4. Tout consommateur devra être autorisé, il recevra un nom de service appelant et sera identifié lui-même par son propre numéro IP. Nous voulons simplement connaître les utilisateurs de notre Extranet et avoir un contact avec eux en cas de problèmes, d'autre part nous ne pouvons nous engager sur un service 24/24, 7/7... Le service est offert tel quel sans garantie particulière, nous l'avons réalisé pour nos propres besoins et le mettons simplement à disposition.

5 Réalisation du Web Service

Sans entrer dans la syntaxe du langage, voilà ce qui fait la différence entre une méthode (fonction) traditionnelle callable au sein d'un programme classique :

```
public StructRetourDNSEasy DNSEasy(string nomServiceAppelant, string numeroIP)
{
}
```

et la même méthode, devenue cette fois un Web Service dans l'outil RAD. Une et une seule ligne contenant une chaîne de caractère qui décrit le rôle du Web Service, tout le traitement est ensuite effectué automatiquement, appréciable...

```
[WebMethod(Description="Rend le nom DNS correspondant au numéro IP fourni en paramètre")]
public StructRetourDNSEasy DNSEasy(string nomServiceAppelant, string numeroIP)
{
}
```

Afin de générer automatiquement la documentation afférente au développement, ou simplement s'aider soi-même pour relire ses programmes quelques mois après leur développement, on préférera commenter la méthode dont l'aspect définitif devient alors :

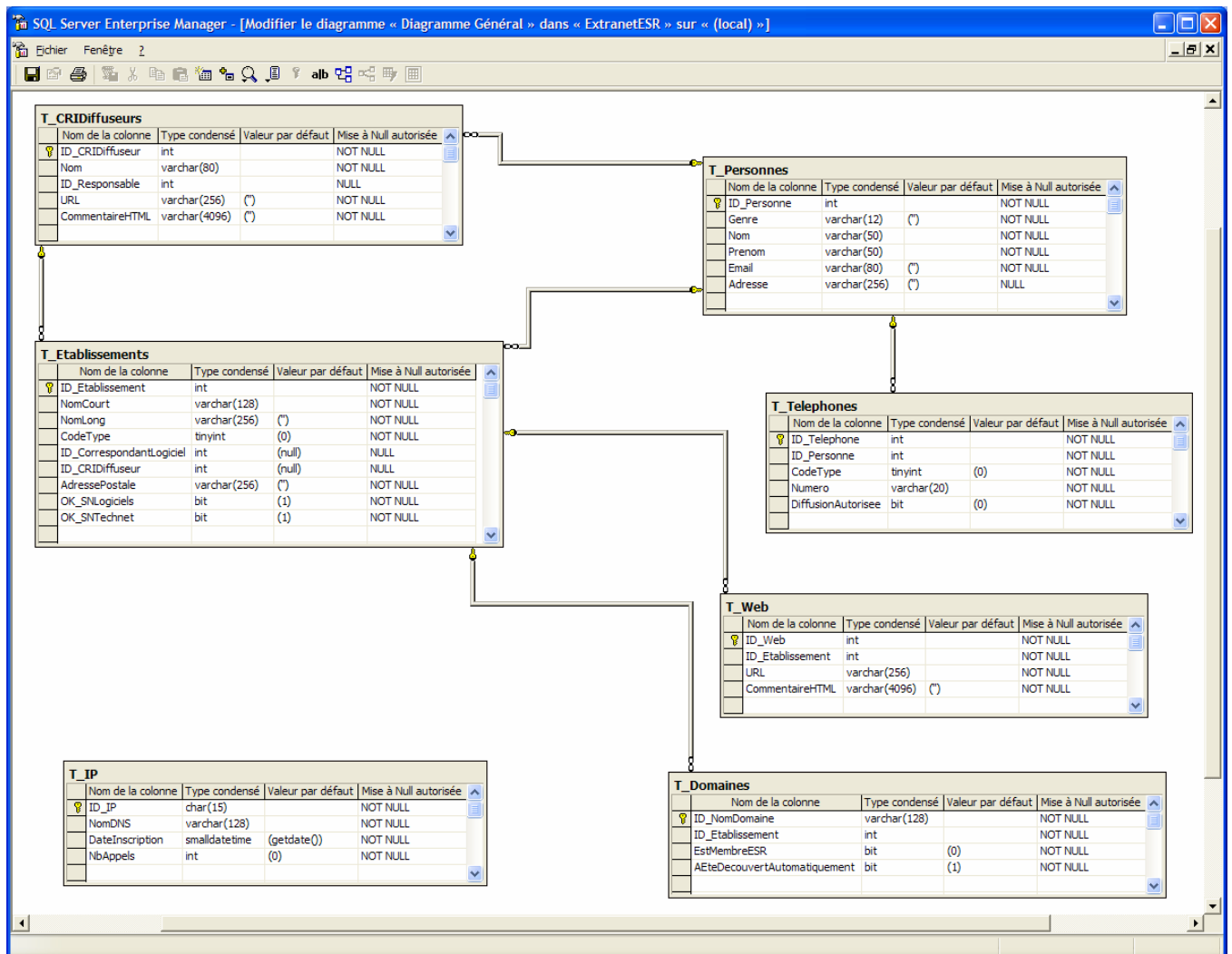
```
/// <summary>
/// Point d'entrée du Webservice
/// </summary>
/// <param name="nomServiceAppelant">Nom du service appelant cette méthode Web</param>
/// <param name="numeroIP">Numéro IP à tester sous la forme : "134.206.1.8"</param>
[WebMethod(Description="Rend le nom DNS correspondant au numéro IP fourni en paramètre")]
public StructRetourDNSEasy DNSEasy(string nomServiceAppelant, string numeroIP)
{
}
```

Notons au passage que l'intellisense (outil qui permet l'aide automatique à la saisie) prend immédiatement en compte les commentaires et les affiche à bon escient.

Bien évidemment il faut disposer d'un serveur Web afin que le Web Service devienne consommable, mais tous les fichiers nécessaires sont générés automatiquement et immédiatement utilisables. De plus on peut déboguer le service en cours de consommation, ce qui est une aide particulièrement appréciable.

6 Base de données

Une base interne est utilisée par le Web Service, elle stocke les divers renseignements concernant les établissements afin de pouvoir retourner les valeurs spécifiques liées à un numéro IP précis. Voici un diagramme général de la base (sujet à évolution, mais de toute façon totalement transparent pour le consommateur du service) :



La base est pour l'instant une base SQL Server remplie manuellement, cependant elle va être profondément modifiée pour être alimentée directement à partir de la base maintenue par la CARI. Cette base est sous un autre format, elle va être exportée en XML afin que nous puissions l'importer directement. Le diagramme de la base sera donc profondément modifié pour s'adapter à celui de la CARI afin d'automatiser les mises à jour.

Cette base du Ministère pourrait être également interrogée via un Web Service, mais l'imbrication de Web Services s'appelant les uns les autres n'est pas souhaitable pour des raisons d'efficacité (consommer un Web Service demande du temps) et de continuité de service (si la chaîne comporte trop de maillons, il suffit que l'un deux casse pour que l'ensemble s'écroule).

Sur le plan de la programmation, notons un point intéressant que ceux qui font des requêtes aux base de données apprécieront : l'existence de typage fort sur les valeurs retournées. Par exemple si un champ contient la valeur NULL ce n'est absolument pas (et définitivement pas) la valeur 0 (zéro) ou toute autre valeur, c'est simplement que le champ n'a pas été renseigné ou que le concepteur ou celui qui a alimenté la base a délibérément choisi cette valeur. Par contre le programmeur est bien embêté, il ne sait ni comment récupérer ni comment traiter cette valeur. Un autre exemple, un entier stocké dans la base est-il sur 16, 32 ou 64 bits ?

L'exemple ci-dessous montre une partie du traitement (ici la récupération) d'un enregistrement de la base. On notera les types utilisés : `sqlstring` au lieu de `string`, ainsi que le test de la valeur à NULL d'un champ. Une classe intègre les types SQL et des méthodes s'appliquent aux objets de cette classe.

```
using System.Data.SqlClient; // Préférable à OleDb pour une base SQL Server
using System.Data.SqlTypes;
...
public struct Struct_T_IP {
    public SqlString ID_IP;
    public SqlString NomDNS;
    public SqlDateTime DateInscription;
    public SqlInt32 NbAppels;
}
...
while (dataReader.Read()){
    T_IP.ID_IP = dataReader.GetSqlString (0);
    T_IP.NomDNS = dataReader.GetSqlString (1);
    T_IP.DateInscription = dataReader.GetSqlDateTime (2);
    T_IP.NbAppels = dataReader.GetSqlInt32 (3);
}
...
string nomDNS = (T_IP.NomDNS.IsNull) ? "" : T_IP.NomDNS.Value;
...
```

7 Enveloppe SOAP

Nous pouvons regarder en détail ce que donne l'enveloppe SOAP associée au Web Service simplifié DNSEasy décrit plus haut :

Source C#

```
/// <summary>
/// Point d'entrée du Webservice
/// </summary>
/// <param name="nomServiceAppelant">Nom du service appelant cette méthode Web</param>
/// <param name="numeroIP">Numéro IP à tester sous la forme : "134.206.1.8"</param>
[WebMethod(Description="Rend le nom DNS correspondant au numéro IP fourni en paramètre")]
public StructRetourDNSEasy DNSEasy(string nomServiceAppelant, string numeroIP)
{
}
}
```

Les valeurs libellés ici `length` et `string` devant être remplacées par de vrais valeurs :

Message SOAP

```
POST /ExtranetESR/ExtranetESR.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://logiciel.education.fr/ExtranetESR/DNSEasy"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <DNSEasy xmlns="http://logiciel.education.fr/ExtranetESR">
      <nomServiceAppelant>string</nomServiceAppelant>
      <numeroIP>string</numeroIP>
    </DNSEasy>
  </soap:Body>
</soap:Envelope>
```

Bien entendu, je rassure immédiatement le lecteur, vous n'avez jamais à saisir ce texte manuellement, mais vous utilisez des outils qui font ceci pour vous (notion de proxy de Web Service). L'exemple est là pour montrer comment le code XML placé dans une enveloppe SOAP est à rapprocher du code C# de l'application.

Le retour se fait conformément à la déclaration de la structure de retour :

Source C#

```
public struct StructRetourDNSEasy {
    public string numeroIP;
    public string nomDns;
    public bool estAutorise;
    public bool uneErreurEstSurvenue;
    public string messageErreur;
}
```

Ainsi que la réponse du Web Service après sa consommation :

Message SOAP

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <DNSEasyResponse xmlns="http://logiciel.education.fr/ExtranetESR">
      <DNSEasyResult>
        <numeroIP>string</numeroIP>
        <nomDns>string</nomDns>
        <estAutorise>boolean</estAutorise>
        <uneErreurEstSurvenue>boolean</uneErreurEstSurvenue>
        <messageErreur>string</messageErreur>
      </DNSEasyResult>
    </DNSEasyResponse>
  </soap:Body>
</soap:Envelope>
```

Nous pouvons voir ce que donne une consommation réussie (numeroIP = 134.206.1.8) :

```
<?xml version="1.0" encoding="utf-8" ?>
<StructRetourDNSEasy xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://logiciel.education.fr/ExtranetESR">
  <numeroIP>134.206.1.8</numeroIP>
  <nomDns>pcdv.univ-lille1.fr</nomDns>
  <estAutorise>true</estAutorise>
  <uneErreurEstSurvenue>false</uneErreurEstSurvenue>
  <messageErreur />
</StructRetourDNSEasy>
```

Ainsi que ce qui se passe en cas de transfert erroné (numeroIP = 134.206.1.888) :

```
<?xml version="1.0" encoding="utf-8" ?>
<StructRetourDNSEasy xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://logiciel.education.fr/ExtranetESR">
  <numeroIP />
  <nomDns />
  <estAutorise>false</estAutorise>
  <uneErreurEstSurvenue>true</uneErreurEstSurvenue>
  <messageErreur>Erreur: Numéro IP incorrect (134.206.1.888),
    champ non compris entre 0 et 255</messageErreur>
</StructRetourDNSEasy>
```

8 Noms de domaines

Décider que telle machine appartient ou non à notre Extranet est un tout autre challenge ; il ne s'agit plus de codage, de programmation ou de Web Service...

Non, affirmer qu'une machine est de notre communauté repose sur les données que nous fournit la communauté, sur les informations qu'elle peut nous remonter. Pour réaliser le travail nous sommes obligés d'être humble et pragmatique, nous ne sommes (seront) jamais totalement à jour ; en effet qui peut nous assurer que tel administrateur système dans son laboratoire ne va pas créer un sous domaine dont nous ignorons l'existence ?

Moyennant ces quelques précautions nous pouvons quand même connaître assez aisément plus de 99% des domaines utilisés, le reste se faisant au coup par coup, sous domaine par sous domaine.

Le point de départ est l'Annuaire de la Recherche et de l'Enseignement Supérieur (ARES), créé par le CRU [1]. Son objet est de constituer un annuaire des établissements contenant :

1. des informations publiques d'ordre général comme l'adresse, le numéro de téléphone, le serveur Web, l'annuaire, les campus,...
2. des informations plus spécifiques comme les coordonnées de responsables (présidents, secrétaires généraux, RSSI, responsables visioconférences...), les caractéristiques des raccordements réseaux des différents campus...
3. les plages d'adresses IP et les domaines DNS utilisés en fonction de leurs usages (administration, recherche,...)

C'est bien sûr le troisième point qui nous intéresse et plus spécifiquement les noms de domaines.

8.1 L'auto découverte

Le Web Service est capable de prendre un certain nombre de décisions seul. S'il connaît un domaine a.fr qui a été validé pour un établissement donné, il acceptera automatiquement les sous domaines qui en résulteront : b.a.fr, c.a.fr et d.e.a.fr par exemple.

Ceci peut conduire à des erreurs, le cas le plus flagrant est d'autoriser *.jussieu.fr, alors qu'il ne faut surtout pas autoriser *.ext.jussieu.fr. Mais ces cas restent exceptionnels, dans la grande majorité de nos domaines, tout se passe bien.

Une alerte est cependant remontée vers l'administrateur pour lui signaler un sous domaine jusque là inconnu, l'administrateur peut alors valider définitivement ce sous domaine ou l'invalider.

Sur un plan pratique, l'administrateur prendra contact avec l'établissement pour effectuer cette validation.

8.2 L'exclusion sélective

Le Web Service permet de traiter les différents cas particuliers qui peuvent se présenter :

- Isoler les domaines étudiants des autres, ceci pour tel consommateur du service tout en le laissant passer pour tel autre. En fait chaque consommateur hérite d'une configuration standard et peut modifier les choix qui sont fait, sous domaine par sous domaine.
- Exclusion de *.ext.jussieu.fr alors que *.jussieu.fr et *.ccr.jussieu.fr sont acceptés, c'est là un des cas les plus remarquables.
- Exclusion automatique de noms de domaines à des fins d'accélération, par exemple « .com »

8.3 Bufferisation et durée de vie

Une résolution DNS peut être très longue (plusieurs dizaines de secondes) et donc devenir très pénalisante si elle est effectuée systématiquement à chaque appel. Pour se prémunir de cet inconvénient les découvertes DNS sont placées dans une table de la base de données.

Afin d'harmoniser les clés de la base, une forme canonique de l'adresse est stockée : 15 caractères pour une adresse de la forme : aaa.bbb.ccc.dddd. Nous aurions pu encoder en binaire l'adresse et peut-être (?) gagner en efficacité au niveau des accès, mais la consultation manuelle de la base est beaucoup moins évidente.

Un nom associé à une adresse n'est pas éternel, les tables DNS peuvent changer, aussi une durée de vie a-t-elle été associée à chaque entrée. Un programme externe, indépendant du Web Service, inspecte les entrées dans la table IP pour faire les mises à jour éventuelles et réinitialiser les compteurs. La durée de vie, paramétrable, a été fixée à 30 jours pour l'instant.

Un journal trace les modifications de noms au fil du temps et il est bien vide pour l'instant, les seuls cas rencontrés concernant des adresses qui ne sont pas de notre communauté.

Un compteur est associé à chaque adresse et permet un minimum de statistiques. Idéalement il faudrait un compteur par service appelant, mais pour l'instant cette extension n'est pas encore implantée.

8.4 Journaux

Les responsables des services qui consomment l'Extranet ont la possibilité de récupérer les logs qui les concernent et de voir ainsi l'usage qu'ils font du Web Service.

D'autres journaux concernent eux plus spécifiquement l'administrateur du Web Service qui peut tracer à la fois l'utilisation par les consommateurs, les erreurs, les sous domaines inconnus, etc.

9 Conclusion

Des demandes de limitation d'accès à notre communauté Enseignement Supérieur – Recherche peuvent maintenant être résolues simplement, quelque soit le système d'exploitation utilisé, quelque soit le serveur Web employé. Chacun, moyennant un minimum de lignes de programme (PHP, ASP, Java...) peut savoir aisément s'il doit ou non laisser l'accès aux pages de son serveur à une machine qui se présente face à lui avec son simple numéro IP.

Pas la peine de réinventer la roue à chaque fois ou de transférer une base sur chaque machine, le service existe, il est disponible, il à la fois simple d'usage et universel.

L'utilisation de Web Services a été une solution pertinente et beaucoup de problèmes laissés en suspens depuis des années vont maintenant pouvoir être résolus de façon simple et harmonieuse. Ce qui n'avait pas été conçu à l'origine pour travailler ensemble peut maintenant le faire avec élégance.

Références

- [1] C. Claveleira, ARES : caractéristiques d'un annuaire LDAP atypique. Dans *Actes du congrès JRES2001*, pages 287-294, Lyon, Décembre 2001.