

Groupe de travail Réseau  
Request for Comments : 3470  
BCP : 70  
Catégorie : Bonnes pratique actuelles  
Traduction Claude Brière de L'Isle

S. Hollenbeck, VeriSign, Inc.  
M. Rose, Dover Beach Consulting, Inc.  
L. Masinter, Adobe Systems Incorporated  
janvier 2003  
octobre 2007

# Lignes directrices pour l'utilisation du langage de balisage extensible (XML) dans les protocoles de l'IETF

## Statut du présent mémo

Le présent document spécifie les bonnes pratiques actuelles de l'Internet pour la communauté de l'Internet, et appelle à des discussions et suggestions pour son amélioration. La distribution du présent mémo n'est soumise à aucune restriction.

## Notice de Copyright

Copyright (C) The Internet Society (2003). Tous droits réservés.

## Résumé

Le langage de balisage extensible (XML, *Extensible Markup Language*) est un cadre pour structurer les données. Alors qu'il est une évolution de la norme de langage de balisage généralisé (SGML, *Standard Generalized Markup Language*) – un langage de balisage principalement orienté sur la structuration des documents -- XML a évolué pour devenir un mécanisme largement utilisé pour représenter des données structurées.

Il y a une grande variété de protocoles de l'Internet qui sont développés ; beaucoup ont des besoins de représentation de données structurées pour leur application. Il y a eu beaucoup d'intérêt pour l'utilisation de XML comme méthode de représentation. Le présent document décrit les concepts de base de l'XML, analyse diverses solutions de remplacement à l'utilisation de XML, et donne des lignes directrices pour l'utilisation de XML au sein des protocoles de l'IETF en cours de normalisation.

## Table des matières

Lignes directrices pour l'utilisation du langage de balisage extensible (XML) dans les protocoles de l'IETF.....	1
1 Introduction et généralités.....	2
1.1 Public visé.....	2
1.2 Domaine d'application.....	2
1.3 Évolution de XML.....	3
1.4 Utilisateurs XML, groupes de soutien, et informations supplémentaires.....	3
2 Considérations sur le choix de XML.....	3
3 Solutions de remplacement à XML.....	4
4 Considérations et recommandations d'utilisation de XML.....	4
4.1 Syntaxe XML et bonne formation.....	5
4.2 Ensemble d'informations XML.....	5
4.3 Restrictions syntaxiques.....	5
4.4 Déclarations XML.....	6
4.5 Instructions de traitement XML.....	6
4.6 Commentaires XML.....	6
4.7 Validité et extensibilité.....	6
4.8 Sémantique et syntaxe.....	7
4.9 Espaces de noms.....	7
4.10 Considérations sur la conception des éléments et des attributs.....	8
4.11 Données binaires et texte avec des caractères de contrôle.....	9

4.12 Traitement incrémental.....	9
4.13 Déclarations d'entité et références d'entité.....	10
4.14 Références externes.....	10
4.15 Traitement d'URI.....	10
4.16 Espaces.....	10
4.17 Interaction avec l'IANA.....	11
5 Considérations pour l'internationalisation.....	11
5.1 Jeux de caractères et codages.....	11
5.2 Déclaration de langage.....	12
5.3 Autres considérations d'internationalisation.....	12
6 Considérations relatives à l'IANA.....	12
7 Considérations sur la sécurité.....	12
8 Remerciements.....	13
9 Références normatives.....	13
10 Références informatives.....	13
11 Adresse des auteurs.....	15
12 Déclaration de copyright.....	16

## Conventions utilisées dans le présent document

Le présent document recommande, en tant que politique, ce que les spécifications des protocoles de l'Internet -- et en particulier, les documents de protocoles en cours de normalisation de l'IETF -- devraient inclure comme langage normatif. Les mots clés en majuscules "DEVRAIT", "DOIT", "EXIGE", etc. sont utilisés selon le sens où il seraient utilisés dans d'autres documents avec les significations spécifiées dans le BCP 14, RFC 2119 [1].

## 1 Introduction et généralités

Le langage de balisage extensible (XML, [8]) est un cadre de structuration des données. Alors qu'il est une évolution de la norme de langage de balisage généralisé (SGML, *Standard Generalized Markup Language*) – langage de balisage principalement orienté sur la structuration des documents -- XML a évolué pour devenir un mécanisme largement utilisé pour représenter des données structurées dans les échanges de protocoles. Voir "XML en 10 points" [47] pour une introduction à XML.

### 1.1 Public visé

De nombreux concepteurs de protocoles de l'Internet envisagent d'utiliser XML et des fragments de XML dans le contexte des protocoles Internet existants et nouveaux. Le présent document est destiné à servir de guide de l'utilisation de XML et de politique de l'IETF pour les documents en cours de normalisation. Ceux qui sont expérimentés dans la pratique de XML seront vraisemblablement déjà familiarisés avec les matériaux de base présentés ici, mais les lignes directrices sont destinées également à ces lecteurs.

### 1.2 Domaine d'application

Le présent document est destiné à donner des lignes directrices pour l'utilisation de contenus XML au sein d'un plus grand protocole. Le but n'est pas de suggérer que XML est la "meilleure" façon ou la façon "préférée" de représenter des données ; le but est plutôt d'exposer le contexte d'utilisation de XML au sein d'un protocole une fois que les autres facteurs désignent XML comme solution possible de représentation des données. Le protocole de résolution des noms communs (CNRP, *Common Name Resolution Protocol* [24]) est un exemple de protocole qui serait visé par ces lignes directrices si il devait être nouvellement défini. Le présent document ne vise pas l'utilisation de protocoles comme SMTP ou HTTP pour l'envoi de documents XML comme de la messagerie électronique ou du contenu ordinaire de la toile.

Un certain nombre de cadres de protocole sont déjà utilisés ou en cours de développement qui sont entièrement centrés sur le "protocole XML" – sur l'utilisation exclusive de XML comme représentation des données dans le protocole. Par exemple, le World Wide Web Consortium (W3C) développe un cadre de protocole XML qui se fonde sur SOAP ([45] et [46]). L'applicabilité de tels protocoles est en dehors du domaine d'application du présent document.

De plus, il y a des cadres de représentation de plus haut niveau, fondés sur XML, qui ont été conçus comme porteurs de certaines classes d'information ; par exemple, le cadre de description de ressource (RDF, *Resource Description Framework* [38]) est une représentation fondée sur XML pour les assertions logiques. Le présent document ne fournit pas de lignes directrices pour l'utilisation de tels cadres.

### 1.3 Évolution de XML

XML 1.0 a été publié à l'origine comme recommandation du W3C en février 1998 [35], et a été révisé dans une 2<sup>ème</sup> édition [8] en octobre 2000. Plusieurs facilités supplémentaires ont été aussi définies, qui s'appuient sur la spécification de base. Bien que ces additions aient été conçues pour être en cohérence avec XML 1.0, elles ont des niveaux de stabilité, de consensus, et de mise en œuvre variables. En conséquence, le présent document identifie les caractéristiques majeures de l'évolution de XML et fait des suggestions quant aux circonstances dans lesquelles chaque caractéristique devrait être utilisée.

### 1.4 Utilisateurs XML, groupes de soutien, et informations supplémentaires

Il y a beaucoup de groupes de soutien à XML, dont certains sont entièrement consacrés à l'industrie XML [51], certains sont consacrés aux développeurs [52], certains consacrés aux applications d'affaires de XML [53], et beaucoup de groupes consacrés à l'utilisation de XML dans un contexte particulier.

Il sort du domaine d'application du présent document de fournir une liste complète des références. Les lecteurs intéressés se tourneront vers les trois références ci-dessus comme point de départ, ainsi que vers leur moteur de recherche Internet favori.

## 2 Considérations sur le choix de XML

XML est un outil qui fournit les moyens d'atteindre un objectif. Choisir le bon outil pour une tâche donnée est un élément essentiel pour s'assurer que la tâche peut être menée à bien de façon satisfaisante. La présente section décrit les facteurs qui permettent de savoir quand on peut considérer XML comme un outil à utiliser dans les protocoles de l'IETF :

1. XML est un métalangage de balisage qui peut être utilisé pour définir des langages de balisage pour des domaines spécifiques et des espaces de problèmes.
2. XML fournit à la fois une structure logique et une structure physique pour décrire des données. Le tramage des données est incorporé.
3. Les instances XML peuvent être validées par rapport à la définition formelle d'une spécification de protocole.
4. XML accepte l'internationalisation.
5. XML est extensible. À la différence d'autres langages de balisage (comme HTML), de nouvelles étiquettes (et donc de nouveaux éléments de protocole) peuvent être définis sans exiger de changements de XML lui-même.
6. XML est toujours en évolution. Les spécifications formelles sont mises à jour au fur et à mesure que l'expérience de son utilisation est acquise et appliquée.
7. XML ne fournit pas de mécanismes d'origine pour la prise en charge de la frappe détaillée des données. Des mécanismes supplémentaires (tels que ceux décrits au paragraphe 4.7) sont nécessaires pour spécifier les types de données de protocoles abstraits.
8. XML est fondé sur du texte, aussi les fragments de XML sont facilement créés, édités, et gérés en utilisant des facilités courantes. De plus, être fondé sur le texte signifie qui prend plus facilement en charge le développement incrémentiel, l'élimination des fautes, et la connexion. Un simple fragment XML "en boîte" peut être incorporé dans un programme comme une constante de chaîne, plutôt que d'avoir à être reconstruit.
9. Les données binaires sont à coder sous une forme textuelle pour être représentées en XML.
10. XML est prolix comparé à beaucoup d'autres langages de représentation de données structurés. Une représentation avec l'extensibilité des éléments et lisibilité par l'homme exige normalement plus de bits comparée à un langage optimisé pour un traitement machine efficace.
11. Les mises en œuvre XML sont encore relativement nouvelles. À mesure que les concepteurs et les développeurs gagnent de l'expérience, il n'est pas rare de trouver des défauts dans les premiers produits courants.
12. La prise en charge de XML est disponible dans un grand nombre d'utilitaires de développement de logiciels, disponibles à la fois dans des produits libres et dans des produits brevetés.
13. La vitesse de traitement de XML peut être un problème dans certains environnements. Le traitement de XML peut être plus lent parce que les flux de données XML peuvent être plus gros que ceux d'autres représentations, et l'utilisation d'analyseurs XML tout venant va ajouter une couche de logiciel avec ses propres coûts de fonctionnement (bien que ces coûts puissent être réduits par l'utilisation cohérente d'un analyseur optimisé). Dans certaines situations, le traitement de XML exige l'examen de chaque octet du flux de données XML tout entier, avec une redondance plus élevée qu'avec les représentations où les segments inintéressants peuvent être sautés.

### 3 Solutions de remplacement à XML

Le présent document se concentre sur les lignes directrices de l'utilisation de XML. Il est utile de considérer pourquoi il faut utiliser XML plutôt qu'un autre mécanisme. La présente section considère quelques autres mécanismes de représentation couramment utilisés et compare XML à ces autres solutions.

Pour beaucoup de protocoles fondamentaux, l'exigence d'extensibilité est modeste, et les exigences de performances sont assez élevées pour que des blocs de données binaires de taille fixe soient la représentation appropriée ; des mécanismes tels que XML ne font qu'ajouter du gras. La RFC 3252 [23] décrit un exemple humoristique de XML comme protocole bouffi.

De plus, il y a d'autres cadres de représentation et d'extensibilité qui ont été utilisés avec succès dans des protocoles de communication. Par exemple, la notation de syntaxe abstraite n° 1 (ASN.1) [28] avec les règles de codage de base (BER, *Basic Encoding Rules* [29]) correspondantes font partie de la suite de protocoles de communication OSI, et ont été utilisées dans de nombreuses normes de communications ultérieures (par exemple, le protocole ANSI de restitution d'informations [27] et le protocole simple de gestion de réseau (SNMP, *Simple Network Management Protocol* [13]). La représentation de données externes (XDR, *External Data Representation* [14]) et ses variantes ont été utilisées dans de nombreuses autres applications réseau distribuées (par exemple, le protocole système de fichiers réseau (NFS, *Network File System* [22]). Avec certains types de codage ASN.1, les types de données sont explicites dans la représentation, alors que avec XDR, les types de données des composants sont décrits en externe au titre d'une spécification d'interface.

De nombreux autres protocoles utilisent directement les structures des données (sans encapsulation des données) en décrivant la structure des données avec le format Backus Naur (BNF, [25]) ; de nombreux protocoles de l'IETF utilisent la forme Backus-Naur augmentée (ABNF, [16]). Le protocole simple de transfert de messagerie (SMTP, [21]) est un exemple de protocole spécifié avec l'ABNF.

ASN.1, XDR, et BNF sont décrits ici comme exemples de solutions de remplacement à l'utilisation de XML dans les protocoles de l'IETF. Il y a d'autres alternatives, mais il n'entre pas dans le domaine d'application du présent document de faire une récapitulation complète de toutes les solutions de remplacement possibles.

Les autres méthodes de représentation peuvent différer de XML de plusieurs façons importantes :

**Codage du texte et jeux de caractères :** le codage de caractères utilisé pour représenter une spécification formelle. XML définit un modèle de caractère cohérent fondé sur le jeu de caractères universel (UCS, *Universal Character Set* [31] et [33]), et exige que les analyseurs XML acceptent au moins UTF-8 [4] et UTF-16 [20], et permettent d'autres codages. Alors que ASN.1 et XDR peuvent porter des chaînes de tout codage, il n'y a pas de mécanisme commun pour définir les codages de caractères en leur sein. Normalement les définitions ABNF tendent à être définies en termes d'octets ou de caractères en ASCII.

**Codage des données :** XML est défini comme une séquence de caractères, plutôt que comme une séquence d'octets. Le schéma XML [42] inclut des mécanismes pour représenter certains types de données (entier, date, matrice, etc.) mais beaucoup de types de données binaires sont codés en Base64 [15] ou en hexadécimal. ASN.1 et XDR ont des mécanismes riches pour coder une grande variété de types de données.

**Extensibilité :** XML a un modèle d'extensibilité riche tel que les spécifications XML puissent fréquemment avoir des versions indépendantes. Les spécifications peuvent être étendues en ajoutant de nouveaux noms d'élément et d'attributs (si il y a compatibilité) ; d'autres extensions peuvent être ajoutées en définissant de nouveaux espaces de nom XML [9], bien qu'il n'y ait pas de mécanisme standard dans XML pour indiquer s'il est ou non obligatoire de reconnaître les nouvelles extensions. De même, plusieurs techniques sont disponibles pour étendre les spécifications ASN.1. Les spécifications XDR tendent à n'être pas extensibles indépendamment par les différentes parties parce que le tramage et les types de données sont implicites et non auto-descriptifs. L'extensibilité des éléments de protocole fondés sur BNF doit nécessairement être planifiée explicitement.

**Lisibilité des éléments de protocole :** Comme noté ci-dessus, XML est fondé sur le texte, et donc porte des avantages (et désavantages) des éléments de protocole fondés sur le texte. Normalement, il partage ceci avec les éléments de protocole définis en (A)BNF. ASN.1 et XDR utilisent des codages binaires qui ne sont pas facilement lisibles par l'homme.

### 4 Considérations et recommandations d'utilisation de XML

La présente section note plusieurs aspects de XML et fait des recommandations pour son utilisation. Depuis la publication

de XML version 1 [35] en 1988, une seconde édition rédactionnelle [8] a été publiée en 2000 ; cette section se réfère à la seconde édition.

#### 4.1 Syntaxe XML et bonne formation

XML [8] est défini en termes de syntaxe concrète : une séquence de caractères, utilisant les caractères "<", "=", "&", etc. comme délimiteurs. Une instance est XML si et seulement si elle est bien formée, c'est-à-dire, si tous les caractères et données de balisage sont conformes aux règles structurelles définies au paragraphe 2.1 de [8].

Les caractères et les données de balisage mal formés ne sont pas de l'XML ; la bonne formation est la base de la compatibilité syntaxique avec XML. Sans bonne formation, tous les avantages de l'utilisation de XML disparaissent. Pour cette raison, il est recommandé que les spécifications de protocole exigent explicitement la bonne formation XML ("DOIT être bien formé").

L'IETF a une tradition établie de longue date d'être "libéral dans ce qu'on accepte" qui peut sembler être à l'opposé de cette recommandation. Étant donné que XML exige la bonne formation, les analyseurs qui se conforment à XML sont intolérants des erreurs de bonne formation. Lors de la spécification du traitement d'éléments de protocole XML erronés, un concept de protocole ne doit jamais recommander de tenter d'interpréter partiellement des instances d'un élément mal formé à qui il est demandé d'être XML. Les comportements raisonnables dans de tels scénarios pourraient comporter de tenter une retransmission ou d'interrompre une session en cours.

#### 4.2 Ensemble d'informations XML

En plus de la syntaxe concrète de XML, il y a un modèle abstrait du contenu XML connu sous le nom de "Ensemble d'informations" (infoset) [37]. On pourrait penser un analyseur XML comme un consommateur de syntaxe concrète et producteur d'ensembles d'informations XML pour le traitement ultérieur.

En utilisation normale de XML, la définition des documents XML admissibles est souvent définie en termes d'ensemble d'informations de XML et pas de syntaxe concrète. La notion est que toute représentation syntaxique qui donne le même ensemble d'informations sera traitée de façon équivalente.

Dans certains cas, les protocoles ont été définis seulement en termes d'ensemble d'informations XML, ou en permettant d'autres représentations de syntaxe concrète. Cependant, comme le contexte de XML incorporé au sein d'autres protocoles de l'Internet exige une définition sans ambiguïté de la syntaxe concrète, définir un élément de protocole XML dans les seuls termes de son ensemble d'informations XML et permettre d'autres représentations de syntaxe concrète est en dehors du domaine d'application du présent document.

#### 4.3 Restrictions syntaxiques

Dans certaines circonstances, un concepteur de protocole peut être tenté de définir un élément de protocole fondé sur XML comme "XML", tout en imposant en même temps des restrictions supplémentaires, au delà de celles imposées par la recommandation XML elle-même -- par exemple, en restreignant le codage de caractères du document, ou en évitant les sections CDATA, les références d'entité de caractère, en imposant des restrictions supplémentaires à l'utilisation des espaces, etc. La catégorie générale des restrictions visées par ce paragraphe est celles qui permettraient certaines, mais pas d'autres, de l'ensemble des représentations syntaxiques qui ont la même représentation canonique conformément au XML canonique décrit dans la RFC 3076 [6].

Faire ce type de restrictions dans une définition de protocole aura l'inconvénient que celui qui mettra le protocole en œuvre pourrait n'être pas capable d'utiliser un processeur par ailleurs conforme à XML pour analyser des éléments de protocole fondés sur XML. Dans certains cas, la motivation de la mise en sous-ensembles de XML est de permettre aux mises en œuvre de construire des processeurs dédiés moins lourds qu'un processeur conforme à toute la gamme de XML. Il y a un certain nombre de bons analyseurs conformes à XML qui sont petits, rapides et disponibles, alors que les processeurs dédiés sont fréquemment réputés pour leurs défaillances lors du traitement de certains cas de syntaxe XML légale.

En général, de telles restrictions syntaxiques devraient être évitées. Dans des circonstances dans lesquelles des restrictions sur la variabilité de la représentation syntaxique de XML sont nécessaires pour une raison ou pour une autre, les concepteurs devraient envisager d'utiliser le "XML canonique" [6] comme la définition de l'élément de protocole, car une telle variabilité a été entièrement retirée. Certaines questions spécifiques sont exposées aux paragraphes 4.4, 4.13, et 5.1.

#### 4.4 Déclarations XML

Une déclaration XML (définie au paragraphe 2.8 de [8]) est un petit en-tête au début d'un flux de données XML qui indique la version XML et le codage de caractères utilisé. Par exemple, `<?xml version="1.0" encoding="UTF-8"?>` spécifie l'utilisation de la version 1 de XML et le codage de caractères UTF-8.

Dans certaines utilisations de XML comme élément de protocole incorporé, le XML utilisé est un petit fragment dans un plus grand contexte, où la version XML est fixée à "1.0" et le codage de caractères est connu comme "UTF-8". Dans ces cas, une déclaration XML peut ajouter une redondance supplémentaire. Dans les cas où le XML est un plus grand composant qui peut trouver son chemin de façon autonome comme un corps d'entité externe (transporté comme un message MIME, par exemple), la déclaration XML est un marqueur important qui est utile pour la fiabilité et l'extensibilité. La déclaration XML est aussi un marqueur important pour le jeu/codage de caractères (voir au paragraphe 5.1), si un codage autre que UTF-8 ou UTF-16 est utilisé. Noter que dans le cas d'UTF-16, XML exige que cette entité commence par une marque d'ordre d'octet (BOM, *Byte Order Mark*), qui ne fait pas partie des données de caractère. Noter que la déclaration XML elle-même ne fait pas partie de l'ensemble d'informations du document XML.

Les spécifications doivent être claires sur l'utilisation des déclarations XML. XML [8] note que les "documents XML devraient commencer par une déclaration XML qui spécifie la version de XML utilisée." En général, une déclaration XML devrait être encouragée ("DEVRAIT être présente") et doit toujours être permise ("PEUT être envoyée"). Une déclaration XML devrait être exigée dans les cas où, si il est permis, le codage de caractères est quelque chose d'autre que UTF-8 ou UTF-16.

#### 4.5 Instructions de traitement XML

Une instruction de traitement XML (définie au paragraphe 2.6 de [8]) est un composant d'un document XML qui signale des informations "hors bande" au receveur ; une utilisation courante des instructions de traitement XML est pour les applications de documents. Par exemple, l'application XML2RFC utilisée pour générer le présent document et qui est décrite dans la RFC 2629 [19] prend en charge une instruction de traitement "tableau des contenus" : `<?rfc toc="yes"?>`

Comme décrit au paragraphe 2.6 de [8], les instructions de traitement ne font pas partie des données de caractère du document, mais doivent être passées à l'application. Par conséquent, il est recommandé que les instructions de traitement soient ignorées lorsque elles se rencontrent dans le traitement normal du protocole. Il est donc aussi recommandé que les instructions de traitement ne soient pas utilisées pour définir des structures de données de protocole normatives ou des extensions pour les raisons suivantes :

- o Les instructions de traitement ne sont pas averties des espaces de nom ; il n'y a aucun moyen pour qualifier une cible d'instruction de traitement avec un espace de nom,
- o L'utilisation des instructions de traitement ne peut pas être contraint par la plupart des langages de schéma,
- o Les références de caractère ne sont pas reconnues au sein d'une instruction de traitement.
- o Les instructions de traitement n'ont pas de structure définie dans XML au-delà de la division entre la cible et tout le reste. Cela signifie que les applications ont normalement à analyser le contenu des instructions de traitement d'une façon dépendante du système ; si au contraire, le contenu a été fourni au sein d'un élément, la structure pourrait être exprimée dans l'XML et l'analyse pourrait être faite par l'analyseur XML.

#### 4.6 Commentaires XML

Un commentaire XML (défini au paragraphe 2.5 de [8]) est un composant d'un document XML qui fournit des informations descriptives qui ne font pas partie des données de caractère du document. Les commentaires XML, comme les commentaires utilisés dans les langages de programmation, sont souvent utilisés pour fournir des informations explicatives en termes compréhensibles par l'homme. Exemple :

```
<!--Ceci est un exemple de commentaire. -->
```

Les commentaires XML peuvent être ignorés par les processeurs conformes. Par conséquent, il est fortement recommandé que les commentaires ne soient pas utilisés pour définir des structures normatives de données de protocole ou des extensions. Il est donc aussi fortement recommandé que les commentaires soient ignorés s'ils sont rencontrés dans le traitement normal du protocole.

#### 4.7 Validité et extensibilité

Un atout important de XML est qu'il y a des mécanismes formels pour définir les contraintes structurelles et de contenu des données ; elles contraignent l'identité des éléments ou attributs ou les valeurs qu'ils contiennent. Ceci est plus qu'un simple formalisme :

- o Une "définition de type de document" (DTD) est définie au paragraphe 2.8 de [8] ; le concept provient d'un mécanisme similaire de SGML. Il y a une expérience significative de l'utilisation des DTD, y compris dans les protocoles de l'IETF.
- o Le schéma XML (défini dans [41] et [42]) donne des caractéristiques supplémentaires pour permettre une spécification plus serrée et plus précise des spécifications admissibles de syntaxe de protocole et de type de données.
- o Il y a aussi un certain nombre d'autres mécanismes pour décrire la validité d'une instance XML ; cela inclut, par exemple, Schematron [49] et RELAX NG [48]. La partie 2 de la norme ISO/CEI Langage de définition de schéma de document (DSDL, [32]) se fonde sur RELAX NG.

Des discussions (et des controverses) sont en cours au sein de la communauté XML sur l'utilisation et l'applicabilité de divers mécanismes de contrainte de validité. Le choix de l'outil dépend des besoins d'extensibilité ou d'un langage et mécanisme formel pour contraindre les valeurs permmissibles et valider l'adhésion aux contraintes.

Il y a des cas où les protocoles ont une validité définie en utilisant un mécanisme de validité ou l'autre, mais les définitions de protocole n'ont pas précisé que tous les éléments de protocole correspondants soient "valides". La décision dépend en partie de la conception de l'extensibilité du protocole. Chaque formalisme a une façon différente de permettre les extensions futures ; de plus, une conception de protocole peut avoir son propre mécanisme d'affichage des versions, sa façon de mettre à jour le schéma, ou de pointer sur un nouveau schéma. Par exemple, l'utilisation des espaces de noms XML (paragraphe 4.9) avec le schéma XML permet d'autres sortes d'extensibilité sans compromettre la validité de schéma.

Quel que soit le formalisme choisi, il y a habituellement des contraintes syntaxiques supplémentaires, et inévitablement, des contraintes sémantiques supplémentaires, sur la validité des éléments XML qui ne peuvent pas être exprimés dans le formalisme.

Le présent document fait les recommandations suivantes pour la définition des protocoles qui utilisent XML :

- o Les protocoles devraient utiliser un formalisme approprié pour définir la validité des éléments de protocole XML.
- o Les protocoles peuvent ou non insister pour que tous les éléments de protocole correspondants soient valides, conformément au mécanisme de validité choisi ; dans tous les cas, le concept d'extensibilité devrait être clair. Qu'arrive-t-il si les données ne sont pas valides ?
- o Comme décrit à la Section 3, il n'y a pas de mécanisme standard dans XML pour indiquer s'il est ou non obligatoire de reconnaître les nouvelles extensions. Les spécifications de protocole fondées sur XML devraient donc décrire explicitement les mécanismes et exigences d'extension pour reconnaître ou ignorer les extensions.

Un modèle idéal de traitement XML pourrait d'abord vérifier la bonne formation ; si cela va, appliquer le formalisme principal et, si les instances "passent", appliquer les autres contraintes de sorte que tout l'ensemble (ou autant qu'en peut traiter la machine) puisse être vérifié au même moment.

Cependant, il est raisonnable de permettre aux mises en œuvre conformes d'éviter de faire de la validation pendant la durée d'exécution et de s'appuyer plutôt sur un code ad hoc pour éviter une dépense plus élevée, par exemple, de validation de schéma, étant donné en particulier qu'il y aura vraisemblablement de la validation de sémantique artisanal supplémentaire.

#### 4.8 Sémantique et syntaxe

Bien que la définition d'un élément de protocole XML en utilisant le formalisme de validité soit utile, elle n'est pas suffisante. XML par lui-même ne fournit pas la sémantique. Tout document définissant un élément de protocole avec XML DOIT aussi avoir une prose suffisante dans le document qui décrit la sémantique quel que soit le XML que le document a choisi de définir.

#### 4.9 Espaces de noms

Les espaces de nom XML, définis dans [9], donnent le moyen d'allouer des balises à un vocabulaire spécifique. Si deux éléments ou attributs provenant de vocabulaires différents ont le même nom, ils peuvent être distingués sans ambiguïté si ils appartiennent à des espaces de nom différents. De plus, les espaces de nom fournissent un soutien significatif à l'extensibilité de protocole car ils peuvent être définis, réutilisés, et traités de façon dynamique.

Les collisions de vocabulaire de balisage sont très possibles lorsque les espaces de nom ne sont pas utilisés pour séparer et identifier de façon univoque des vocabulaires. Les définitions de protocole devraient utiliser les espaces de nom XML existants lorsque c'est approprié. Lorsque un nouvel espace de nom est nécessaire, le "nom d'espace de nom" est un URI qui est utilisé pour identifier l'espace de nom ; il est aussi utile que cet URI pointe sur une description de l'espace de nom. Normalement (et c'est une pratique recommandée par le W3C) on alloue des noms d'espace de nom en utilisant des URI http persistants.

Dans le cas d'espaces de nom dans les document de l'IETF en cours de normalisation, il serait utile qu'une partie permanente du propre espace de l'IETF sur la toile puisse être utilisée à cette fin. Au lieu de cela, d'autres URI permanents peuvent être utilisés, par exemple, des URN dans l'espace de nom d'URN de l'IETF (voir [11] et [12]). Bien qu'il y ait des instances de spécifications de l'IETF qui créent des nouveaux schémas d'URI pour définir les espaces de nom XML, cette pratique est fortement déconseillée.

#### 4.9.1 Espaces de nom et attributs

Il y a un aspect fréquemment incompris des relations entre les attributs sans préfixe et l'espace de nom XML par défaut – l'hypothèse naturelle est qu'un attribut sans préfixe est qualifié par l'espace de nom par défaut, mais ce n'est pas vrai. L'attribut sans préfixe n'appartient plutôt à aucun espace de nom du tout. Et donc, dans l'exemple suivant

```
<ns1:fox a="xxx" ns1:b="qqq" xmlns="http://example.org"/>
<fox a="xxx" ns1:b="qqq" xmlns="http://example.org" xmlns:ns1="http://example.org"/>
```

l'attribut "a" n'est dans aucun espace de nom, alors que "ns1:b" est le même espace de nom que l'élément contenant. Une description spécifique de la relation entre les espaces de nom par défaut et les attributs se trouve au paragraphe 5.2 de [9]. Les implications pratiques de la relation entre espace de nom et attribut sont qu'il faut veiller à ce qu'aucun élément ne contienne plusieurs attributs qui aient des noms identiques ou qui aient des noms qualifiés avec la même partie locale et avec des préfixes qui soient liés à des noms d'espaces de noms identiques.

Dans les applications XML, le choix entre attributs avec et sans préfixe est fréquemment fondé sur le fait qu'ils apparaissent toujours dans des éléments du même espace de nom (auquel cas on utilise les noms sans préfixe et donc sans nom d'espace de nom) ou qu'il est exigé qu'ils puissent être appliqué aux éléments dans d'autres espaces de nom arbitraires (et dans ce cas on utilise un nom à préfixe). Les deux situations surviennent dans le langage XSLT [43] : alors que les attributs sont sans préfixe lorsque ils surviennent dans des éléments de l'espace de nom XSLT, tels que :

```
<xsl:value-of select="."/>
```

ils sont munis d'un préfixe lorsque ils apparaissent comme des éléments non XSLT, comme l'attribut "xsl:version" lorsque on utilise des "feuilles de style d'élément de résultat littéral" :

```
<html xsl:version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns="http://www.w3.org/TR/xhtml1/strict">
<head>
<title>Expense Report Summary</title>
</head>
<body>
<p>Total: <xsl:value-of select="exp-rep/total"/></p>
</body>
</html>
```

#### 4.10 Considérations sur la conception des éléments et des attributs

XML donne beaucoup de souplesse en permettant à un concepteur d'utiliser aussi bien des éléments, des attributs, ou le contenu d'un élément pour porter des données. Ce paragraphe effleure les considérations touchant à la conception ; la littérature XML a beaucoup d'écrits sur ce sujet. L'utilisation cohérente des éléments, attributs, et valeurs est une caractéristique importante d'une conception bien menée.

Les attributs sont généralement destinés à contenir des métadonnées qui décrivent l'élément, et comme tels sont soumis aux restrictions suivantes :

- o Les attributs ne sont pas ordonnés,
- o Il ne peut y avoir plus d'une instance d'un attribut donné dans un élément donné, bien qu'un attribut puisse contenir plusieurs valeurs séparées par un espace ([8], paragraphes 2.3 et 3.3.1),
- o Les valeurs d'attribut peuvent n'avoir aucune balise XML interne pour fournir la structure interne, et
- o Les valeurs d'attribut sont normalisée ([8], paragraphe 3.3) avant traitement

Considérons l'exemple suivant qui décrit une adresse IP en utilisant un attribut pour décrire la valeur d'adresse :

```
<address addrType="ipv4">10.1.2.3</address>
```

On pourrait coder les mêmes informations en utilisant un élément `<addrType>` à la place d'un attribut "addrType" :

```
<address>
  <addrType>ipv4</addrType>
  <value>10.1.2.3</value>
</address>
```

Une autre façon de coder les mêmes informations serait d'utiliser une balise pour le "addrType" :

```
<address>
  <addrType><ipv4/></addrType>
  <value>10.1.2.3</value>
</address>
```

Le choix entre ces conceptions implique des arbitrages concernant, entre autres considérations, les schémas d'extensibilité vraisemblable et la capacité du formalisme à contraindre les valeurs de façon appropriée. Dans le premier exemple, l'attribut peut être vu comme méta-donnée de l'élément qu'il modifie, et fournir une sorte d'"extensibilité d'élément". Le troisième exemple permet une différente sorte d'extensibilité : l'espace "ipv4" peut être étendu en utilisant d'autres espaces de nom, et l'élément `<ipv4>` peut inclure des balises supplémentaires.

De nombreux protocoles incluent des paramètres qui sont choisis dans un ensemble de valeurs désignées. De telles valeurs désignées peuvent être codées comme éléments, attributs, ou chaînes au sein des valeurs d'élément. Toute conception de protocole devrait considérer comment l'ensemble des valeurs désignées sera étendu : en révisant le protocole, en incluant des valeurs différentes dans des espaces de nom XML différents, ou en établissant un registre IANA (comme selon la RFC 2434 [18]). De plus, une pratique courante dans XML est d'utiliser un URI comme valeur ou contenu d'attribut XML.

Les langages qui décrivent la validité syntaxique (y compris de schéma et de DTD XML) offrent souvent un mécanisme pour spécifier les valeurs par "défaut" pour un attribut. Si un élément ne spécifie pas une valeur pour l'attribut, la valeur par "défaut" est alors utilisée. L'utilisation de valeurs par défaut pour les attributs est déconseillée par le présent document. Bien que l'utilisation de ce dispositif puisse réduire à la fois la taille et l'encombrement des documents XML, il a un impact négatif sur les logiciels qui ne connaissent pas les contraintes de validité du document (par exemple, pour le suivi des paquets ou pour les signatures numériques).

#### 4.11 Données binaires et texte avec des caractères de contrôle

XML se définit comme un flux de caractères plus qu'un flux d'octets. Il n'y a aucun moyen d'incorporer des données binaires brutes directement au sein d'un flux de données XML ; toutes les données binaires doivent être codées comme des caractères. Il y a un certain nombre de codages possibles ; par exemple, le schéma XML [42] définit des codages qui utilisent des chiffres décimaux pour des chiffres entiers, des chiffres en Base64 [15], ou des chiffres hexadécimaux. De plus, les données binaires peuvent être transmises en utilisant un autre canal de communication, et référencées au sein des données XML elles-mêmes en utilisant un URI.

Les protocoles qui ont besoin d'un contenant qui puisse détenir à la fois les données structurelles et de grandes quantités de données binaires devraient considérer avec attention si XML est approprié, car les codages Base64 et hexadécimal sont inefficaces. Autrement, les protocoles devraient utiliser le mécanisme du schéma XML pour représenter les données binaires ; le codage Base64 est meilleur pour de plus grandes quantités de données.

XML ne permet pas les caractères de "contrôle" (0x00 à 0x1F) excepté TAB (0x09), CR (0x0A), et LF (0x0D). Ils ne peuvent pas être spécifiés même en utilisant des références d'entité de caractère. IL n'y a actuellement aucune façon commune de les coder au sein de ce qui est autrement du texte ordinaire. Cela signifie que les chaînes qui peuvent être considérées comme "texte" au sein d'un élément de protocole défini en ABNF peuvent avoir besoin d'être traitées comme binaires au sein d'une représentation XML, ou quelque autre mécanisme de codage devrait être inventé.

#### 4.12 Traitement incrémental

Dans certaines situations, il est possible de traiter de façon incrémentale un document XML au fur et à mesure de la réception de chaque étiquette ; ceci est analogue au processus par lequel les logiciels de navigation rendent incrément par incrément les pages HTML au fur et à mesure qu'elles sont reçues. Noter que le traitement incrémental est difficile à mettre en œuvre si il est parsemé à travers de multiples interactions. En d'autres termes, si un protocole exige un traitement incrémental à travers les deux directions d'un flux bidirectionnel, cela peut faire porter un fardeau inhabituel sur ceux qui mettront en œuvre le protocole.

### 4.13 Déclarations d'entité et références d'entité

En plus de son rôle comme mécanisme de validité, un DTD XML fournit une facilité pour les "déclarations d'entité" ([8], paragraphe 4.2). Une déclaration d'entité définit, dans le DTD, une sorte de capacité à établir des macro-instructions où une "référence d'entité" peut être utilisée pour évoquer et inclure le contenu de la déclaration d'entité.

Cette caractéristique ajoute de la complexité au traitement XML, et semble plus appropriée dans le traitement des documents XML que dans la représentation des données. C'est pourquoi le présent document recommande d'éviter les déclarations d'entité dans les spécifications de protocole.

D'un autre côté, cinq références d'entité standard sont intégrées à XML : "&"; "&lt;"; "&gt;"; "&apos;"; et "&quot;". XML a aussi la capacité d'écrire des données de caractères en utilisant des références d'entité numériques (en utilisant la valeur Unicode [33] pour le caractère). Les références d'entité sont normalement étendues avant le calcul de l'ensemble d'informations XML. Restreindre l'utilisation de ces références d'entité introduirait une restriction syntaxique supplémentaire (voir au paragraphe 4.3) sans nécessité ; ces références d'entité devraient être admises.

### 4.14 Références externes

Lorsqu'on utilise XML dans le contexte d'un protocole sans états, qu'il soit le protocole lui-même (par exemple, SOAP), ou simplement comme contenu transféré par un protocole existant (par exemple, XML/HTTP), il faut veiller à ne pas faire dépendre la signification d'un message d'informations en dehors du message lui-même. XML fournit des entités externes (voir au paragraphe 4.13), qui sont un moyen commode pour faire dépendre la signification d'un message de quelque chose d'externe. Une autre façon est d'utiliser des langages de schéma qui puissent changer l'ensemble d'informations, comme XML Schema.

### 4.15 Traitement d'URI

La spécification de base XML [36] définit un attribut "xml:base" dans l'espace de nom XML qui est destiné à affecter la "base" à utiliser pour le traitement d'URI relatif décrit dans la RFC 2396 [17]. Les dispositions de xml:base pour contrôler le traitement d'URI peut être utile aux concepteurs de protocoles, mais si xml:base est permis, les interactions avec toutes les autres dispositions de protocole pour établir le contexte d'URI doivent être spécifiées clairement. Noter que l'utilisation des URI relatives dans les déclarations d'espace de nom a été déconseillée par le W3C ; quelques questions spécifiques des URI relatifs dans les déclarations d'espace de nom et le XML canonique se trouve au paragraphe 1.3 de la RFC 3076 [6].

Noter aussi que, dans de nombreux cas, le terme de "URI" et l'utilisation syntaxique des URI au sein de XML permet des caractères non ASCII dans les URI. Par exemple, le schéma XML de type de données "anyURI" ([42] paragraphe 3.2.17) permet un codage direct des caractères en dehors de la gamme de l'US-ASCII. La plupart des protocoles et spécifications courants de l'IETF ne permettent pas cette syntaxe. Les spécifications de protocole devraient être claires sur la gamme des caractères spécifiée, par exemple, en ajoutant une restriction à la gamme des caractères permis dans le type de données de schéma anyURI, ou en spécifiant que les caractères en dehors de la gamme de l'US-ASCII devraient être esquivés lorsqu'ils sont passés à de plus vieux protocoles ou API.

### 4.16 Espaces

Le comportement de traitement des espaces prescrit par XML peut être une source de confusion entre les concepteurs de protocole et ceux qui les mettent en œuvre. Dans les instances XML, toutes les espaces sont considérées comme significatives et sont par défaut visibles aux applications de traitement. Considérons cet exemple tiré du paragraphe 4.10 :

```
<address>
  <addrType><ipv4/></addrType>
  <value>10.1.2.3</value>
</address>
```

Ce fragment contient un élément <address> et deux éléments fils. Il contient aussi une espace pour les besoins de l'esthétique typographique :

- o au moins trois séparateurs de ligne, qui seront convertis par le processeur XML en caractères de nouvelle ligne (U+000A) (voir le paragraphe 2.11 de [8]), et
- o un ou plusieurs caractères espace en préfixe aux éléments <addrType> et <value>, qu'un processeur XML rendra visibles au logiciel qui lira l'instance.

Les mises en œuvre peuvent supposer en toute sécurité qu'ils peuvent ignorer les espaces dans l'exemple ci-dessus, mais les espaces utilisés pour l'esthétique typographique peuvent être une source de confusion dans d'autres situations. Considérons un changement mineur à l'élément `<value>` :

```
<value>
 10.1.2.3
</value>
```

où une espace se trouve des deux côtés de l'adresse IP. Les processeurs XML traitent l'espace entourant "10.1.2.3" comme partie intégrante de l'élément `<value>`. L'échec à reconnaître de comportement peut conduire à de la confusion et des erreurs à la fois dans la conception et dans la mise en œuvre.

Toute espace est considérée comme significative dans les instances XML. Par conséquent, il est recommandé que les concepteurs de protocoles fournissent des lignes directrices spécifiques pour régler le traitement des espaces au sein des protocoles qui utilisent XML.

#### 4.17 Interaction avec l'IANA

Lorsque XML est utilisé dans un protocole de l'IETF, plusieurs facteurs peuvent exiger une action de l'IANA, à savoir :

- o Types de support XML. Un morceau de XML dans un élément de protocole est parfois intrinsèquement lié au contexte de protocole dans lequel il apparaît, et en particulier pourrait être directement dérivé de et/ou entré de la mise en œuvre de l'automate d'états du protocole. Dans les cas où le contenu XML n'a pas de signification pertinente en dehors de son contexte de protocole d'origine, il n'y a pas de raison d'enregistrer un type MIME. Lorsque il est possible que le contenu XML soit interprété en dehors de son contexte d'origine (comme lorsque ce contenu XML est mémorisé dans un système de fichiers ou tunnelé sous un autre protocole), un type MIME peut alors être enregistré pour spécifier le format spécifique pour les données et pour fournir un conseil sur la façon de le traiter.

Si un étiquetage MIME est nécessaire, l'avis de la RFC 3023 [5] s'applique alors. En particulier, si le XML représente un nouveau type de langage ou de document, un nouveau type de support MIME devrait être enregistrés pour les raisons décrites dans la section 7 et au paragraphe A.1 de la RFC 3023. Dans les situations où XML est utilisé pour coder des données génériques structurées (par exemple, une application orientée document qui implique de combiner XML avec une feuille de style), "application/xml" peut être approprié ("PEUT être utilisé"). Le type de support "text/xml" n'est pas recommandé ("NE DEVRAIT PAS être utilisé") à cause des problèmes de comportement d'affichage et d'ensembles de caractères par défaut.

- o Enregistrement d'URI. Il y a un effort constant ([11], [12]) pour créer un espace de nom d'URN explicitement pour définir des URI pour les noms d'espace de nom et les autres éléments de protocole destinés aux URI à utiliser dans les document en voie de normalisation de l'IETF ; cela pourrait aussi établir une politique de l'IETF pour un tel usage.

## 5 Considérations pour l'internationalisation

La présente section décrit les considérations d'internationalisation pour l'utilisation de XML pour représenter les données dans les protocoles de l'IETF. En plus des recommandations données ici, la politique de l'IETF sur l'utilisation des jeux de caractères et des langages décrite dans la RFC 2277 [3] s'applique aussi.

### 5.1 Jeux de caractères et codages

Les protocoles de l'IETF parlent fréquemment de "jeu de caractères" ou de "charset" d'une chaîne, qui sont utilisés pour noter à la fois le répertoire de caractères et le codage utilisé pour représenter les séquences de caractères sous forme de séquences d'octets.

XML effectue tous les traitements de caractères en termes d'ensemble universel de caractères (UCS, *Universal Character Set*, [31] et [33]). XML exige que tous les processeurs XML prennent en charge à la fois les codages UTF-8 [4] et UTF-16 [20] de UCS, bien que d'autres codages (charsets) compatibles avec UCS puissent être admis. Les documents et entités externes analysées codées en UTF-16 doivent commencer par une marque d'ordre des octets ([8] paragraphe 4.3.3).

La politique de l'IETF [3] exige que l'ensemble de caractères UTF-8 soit accepté pour tous le texte.

Le présent document exige que les protocoles de l'IETF qui utilisent XML permettent le codage UTF-8 des données XML. Comme les processeurs XML conformes sont obligés d'accepter aussi le codage UTF-16, il est aussi recommandé de

permettre le codage UTF-16 (avec la marque d'ordre des octets obligatoire). Certaines applications XML utilisent une marque d'ordre des octets avec le codage UTF-8, mais cette utilisation ne devrait pas être encouragée et elle n'est pas appropriée pour XML incorporé dans d'autres protocoles.

Restreindre les données XML à n'être exprimées qu'en UTF-8 est une restriction syntaxique supplémentaire (voir au paragraphe 4.3), qui, selon les circonstances, pourrait ajouter une complexité de mise en œuvre supplémentaire. Lorsque des codages autres que UTF-8 ou UTF-16 sont utilisés, le codage doit être spécifié en utilisant un attribut "encoding" dans la déclaration XML (paragraphe 4.4), même si il peut y avoir d'autres mécanismes de protocole pour désigner le codage.

## 5.2 Déclaration de langage

Le texte encapsulé dans XML peut être représenté dans de nombreux langages humains différents, et il est souvent utile d'identifier explicitement le langage utilisé pour présenter le texte. XML définit un attribut spécial dans l'espace de nom "xml", `xml:lang`, qui peut être utilisé pour spécifier le langage utilisé pour représenter les données dans un document XML. L'attribut `xml:lang` (qui doit être explicitement déclaré pour être utilisé dans un DTD ou un schéma XML) et les valeurs qu'il peut prendre sont définies au paragraphe 2.12 de [8].

Il est vivement recommandé que les protocoles qui représentent des données dans un langage humain rendent obligatoire l'utilisation d'un attribut `xml:lang` si l'instance XML peut être interprétée dans des contextes dépendants d'un langage.

## 5.3 Autres considérations d'internationalisation

Il y a des mécanismes standard dans la typographie de certains langages humains qu'il peut être difficile de représenter en utilisant simplement des types de données de chaîne de caractères XML. Par exemple, des indices de prononciation peuvent être fournis en utilisant l'annotation Ruby [39], et des commandes incorporées (telles que celles décrites au paragraphe 3.4 de [34]) ou un attribut XHTML [40] "dir" peut être utilisé pour noter la direction d'affichage appropriée pour le texte bidirectionnel.

Un certain nombre de problèmes délicats peuvent survenir lors de l'utilisation de jeux de caractères étendus avec des formats de document XML. Par exemple :

- o Il y a différentes façons de représenter les caractères qui consistent en caractères combinés, et
- o Il y a des débats pour savoir si les URI devraient être représentés en utilisant un sous-ensemble US-ASCII restreint ou de l'Unicode arbitraire (par exemple, "séquence de caractères d'URI" contre "séquence de caractères originale" dans la RFC 2396 [17]).

Certaines de ces questions sont en discussion, avec des recommandations, dans le document "Modèle de caractères pour la Toile mondiale" du W3C [44].

Il est vivement recommandé que les protocoles qui représentent des données dans un langage humain réutilisent les mécanismes existants en tant que de besoin pour assurer un affichage approprié du texte lisible par l'homme.

## 6 Considérations relatives à l'IANA

Le présent mémoire n'a par lui-même aucun impact sur l'IANA. Le paragraphe 4.17 note quelques facteurs qui pourraient exiger une action de l'IANA lors de la définition de protocoles qui utilisent XML.

## 7 Considérations sur la sécurité

Les protocoles réseau font face à de nombreuses sortes différentes de menaces, parmi lesquelles la divulgation involontaire, la modification, et la répétition. Les attaques passives, comme le reniflage de paquet, permettent à un attaquant de capturer et visionner les informations destinées à quelqu'un d'autre. Les données capturées peuvent être modifiées et répétées au destinataire original prévu, le receveur n'ayant aucun moyen de savoir que les informations ont été compromises, de détecter les modifications, d'être assuré de l'identité de l'expéditeur, ou de confirmer quelle instance de protocole est légitime.

Plusieurs options de service de sécurité sont disponibles pour XML pour aider à atténuer ces risques. Bien que XML n'inclue aucun service de sécurité incorporé, d'autres protocoles et couches de protocole fournissent des services qui peuvent être utilisés pour protéger les protocoles XML. Le chiffrement XML [10] fournit des services de confidentialité pour empêcher la divulgation involontaire. XML canonique [6] et les signatures numériques XML [7] fournissent des

services d'intégrité pour détecter les modifications et des services d'authentification pour confirmer l'identité de la source des données. D'autres protocoles de sécurité de l'IETF (par exemple, le protocole de sécurité de la couche Transport (TLS, *Transport Layer Security* [2]) sont aussi disponibles pour protéger les données et les points d'extrémité de service en tant que de besoin.

Étant donné le manque de services de sécurité dans XML, il est impératif que les spécifications de protocole rendent obligatoire des services de sécurité supplémentaires pour contrer les menaces et attaques courantes ; les services spécifiques requis vont dépendre du modèle de menace du protocole.

L'expérience a montré qu'un code qui analyse le trafic du réseau est souvent une "cible douce" pour les pirates informatiques. En conséquence, les développeurs DOIVENT veiller à s'assurer que leur code de traitement XML est robuste par rapport au XML mal formé, aux débordements de mémoire tampon, au mauvais usages des déclarations d'entité, et ainsi de suite.

Les mécanismes XML qui suivent les références externes (paragraphe 4.14) peuvent aussi exposer une mise en œuvre à diverses menaces en causant l'accès automatique à des ressources externes par la mise en œuvre. Il est important d'interdire l'accès arbitraire à de telles références externes au sein des données XML à partir d'une source qui n'est pas de confiance. De nombreuses grammaires XML définissent des constructions qui utilisent des URI pour des références externes ; dans de tels cas, les mêmes précautions doivent être prises.

## 8 Remerciements

Les auteurs tiennent à remercier les personnes suivantes qui ont fourni des contributions significatives au développement du présent document :

Mark Baker, Tim Berners-Lee, Tim Bray, James Clark, Josh Cohen, John Cowan, Alan Crouch, Martin Duerst, Jun Fujisawa, Christian Geuer-Pollmann, Yaron Goland, Graham Klyne, Dan Kohn, Rick Jelliffe, Chris Lilley, Murata Makoto, Michael Mealling, Jean-Jacques Moreau, Andrew Newton, Julian Reschke, Jonathan Rosenberg, Miles Sabin, Rich Salz, Peter Saint-Andre, Simon Saint-Laurent, Margaret Wasserman, et Daniel Veillard.

## 9 Références normatives

- [1] Bradner, S., "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, RFC 2119, mars 1997.
- [2] Dierks, T. et C. Allen, "Protocole TLS, Version 1.0", RFC 2246, janvier 1999.
- [3] Alvestrand, H., "Politique de l'IETF sur les jeux de caractères et les langages", BCP 18, RFC 2277, janvier 1998.
- [4] Yergeau, F., "UTF-8, un format de transformation de ISO 10646", RFC 2279, janvier 1998.
- [5] Murata, M., St. Laurent, S. et D. Kohn, "Types de supports XML", RFC 3023, janvier 2001.
- [6] Boyer, J., "XML canonique, Version 1.0", RFC 3076, mars 2001.
- [7] Eastlake, D., Reagle, J. et D. Solo, "Syntaxe et traitement des signatures XML (Langage de balisage extensible)", RFC 3275, mars 2002.
- [8] Bray, T., Paoli, J., Sperberg-McQueen, C. et E. Maler, "Langage de balisage extensible (XML) 1.0 (2<sup>ème</sup> édition)", W3C REC-xml, octobre 2000, <<http://www.w3.org/TR/REC-xml>>.
- [9] Bray, T., Hollander, D. et A. Layman, "Espaces de noms en XML", W3C REC-xml-names, janvier 1999, <<http://www.w3.org/TR/REC-xml-names>>.
- [10] Imamura, T., Dillaway, B., Schaad, J. et E. Simon, "Syntaxe et traitement du chiffrement XML", W3C REC-xmlenc-core, octobre 2001, <<http://www.w3.org/TR/xmlenc-core/>>.

## 10 Références informatives

- [11] Masinter, L., Mealling, M., Klyne, G. et T. Hardie, "An IETF URN Sub-namespace for Registered Protocol Parameters", Travail en cours.
- [12] Sealling, M., "The IETF XML Registry", Travail en cours.
- [13] Case, J., Fedor, M., Schoffstall, M. et C. Davin, "Simple Network Management Protocol (SNMP)", STD 15, RFC 1157, mai 1990.
- [14] Srinivasan, R., "XDR: External Data Representation Standard", RFC 1832, août 1995.
- [15] Freed, N. et N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, novembre 1996.
- [16] Crocker, D. (Ed.) et P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, novembre 1997.
- [17] Berners-Lee, T., Fielding, R. et L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, août 1998.
- [18] Narten, T. et H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, octobre 1998.
- [19] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, juin 1999.
- [20] Hoffman, P. et F. Yergeau, "UTF-16, an encoding of ISO 10646", RFC 2781, février 2000.
- [21] Klensin, J. (Ed.), "Simple Mail Transfer Protocol", RFC 2821, avril 2001.
- [22] Shepler, S., Callaghan, B., Robinson, D., Thurlow, R., Beame, C., Eisler, M. et D. Noveck, "NFS version 4 Protocol", RFC 3010, décembre 2000.
- [23] Kennedy, H., "Binary Lexical Octet Ad-hoc Transport", RFC 3252, avril 2002.
- [24] Popp, N., Mealling, M. et M. Moseley, "Common Name Resolution Protocol (CNRP)", RFC 3367, août 2002.
- [25] Backus, J., "The syntax and semantics of the proposed international algebraic language of the Zurich ACM-GAMM conference", juin 1959.
- [26] American National Standards Institute, "Code Extension Techniques for Use with the 7-bit Coded Character Set of American National Standard Code (ASCII) for Information Interchange", ANSI X3.41, FIPS PUB 35, 1974.
- [27] American National Standards Institute, "Information Retrieval: Application Service Definition and Protocol Specification", ANSI Z39.50, ISO Standard 23950, 1995.
- [28] Organisation Internationale de normalisation, "Information Processing Systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1)", ISO Standard 8824, décembre 1990.
- [29] Organisation Internationale de normalisation, "Information Processing Systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)", ISO Standard 8825, décembre 1990.
- [30] Organisation Internationale de normalisation, "Information processing - Text and office systems - Standard Generalized Markup Language (SGML)", ISO Standard 8879, 1988.
- [31] Organisation Internationale de normalisation, "Information Technology - Universal Multiple-octet coded Character Set (UCS) - Part 1: Architecture and Basic Multilingual Plane", ISO Standard 10646-1, May 1993.
- [32] Organisation Internationale de normalisation, "DSDL Part 0 - Overview", décembre 2001, <<http://www.jtc1.org/FTP/Public/SC34/DOCREG/0275.htm>>.
- [33] Unicode Consortium, "The Unicode Standard, as it may from time to time be revised or amended", March 2002,

<<http://www.unicode.org/unicode/standard/standard.html>>.

- [34] Duerst, M. et A. Freytag, "Unicode in XML and other Markup Languages", février 2002, <<http://www.w3.org/TR/unicode-xml/>>.
- [35] Bray, T., Paoli, J. et C. Sperberg-McQueen, "Extensible Markup Language (XML) 1.0", W3C REC-xml-1998, février 1998, <<http://www.w3.org/TR/1998/REC-xml-19980210/>>.
- [36] Marsh, J., "XML Base", W3C REC-xmlbase, juin 2001, <<http://www.w3.org/TR/xmlbase/>>.
- [37] Cowan, J. et R. Tobin, "XML Information Set", W3C REC-infoset, octobre 2001, <<http://www.w3.org/TR/xml-infoset/>>.
- [38] Lassila, O. et R. Swick, "Resource Description Framework (RDF) Model and Syntax Specification", W3C REC-rdf-syntax, février 1999, <<http://www.w3.org/TR/REC-rdf-syntax>>.
- [39] Suignard, M., Ishikawa, M., Duerst, M. et T. Texin, "Ruby Annotation", W3C REC-RUBY, mai 2001, <<http://www.w3.org/TR/ruby/>>.
- [40] Pemberton, S., "XHTML 1.0: The Extensible HyperText Markup Language", W3C REC-XHTML, janvier 2000, <<http://www.w3.org/TR/xhtml1/>>.
- [41] Thompson, H., Beech, D., Maloney, M. et N. Mendelsohn, "XML Schema Part 1: Structures", W3C REC-xmlschema-1, mai 2001, <<http://www.w3.org/TR/xmlschema-1/>>.
- [42] Biron, P. et A. Malhotra, "XML Schema Part 2: Datatypes", W3C REC-xmlschema-2, mai 2001, <<http://www.w3.org/TR/xmlschema-2/>>.
- [43] Clark, J., "XSL Transformations (XSLT) Version 1.0", W3C REC-xslt, novembre 1999, <<http://www.w3.org/TR/xslt>>.
- [44] Duerst, M., Yergeau, F., Ishida, R., Wolf, M., Freytag, A. et T. Texin, "Character Model for the World Wide Web 1.0", avril 2002, <<http://www.w3.org/TR/charmod/>>.
- [45] Gudgin, M., Hadley, M., Moreau, JJ. et H. Nielsen, "SOAP Version 1.2 Part 1: Messaging Framework", juin 2002, <<http://www.w3.org/TR/soap12-part1/>>.
- [46] Gudgin, M., Hadley, M., Moreau, JJ. et H. Nielsen, "SOAP Version 1.2 Part 2: Adjuncts", juin 2002, <<http://www.w3.org/TR/soap12-part2/>>.
- [47] W3C Communications Team, "XML in 10 points", novembre 2001, <<http://www.w3.org/XML/1999/XML-in-10-points>>.
- [48] OASIS Technical Committee: RELAX NG, "RELAX NG Specification", décembre 2001, <<http://www.oasis-open.org/committees/relax-ng/spec-20011203.html>>.
- [49] Jelliffe, R., "The Schematron", novembre 2001, <<http://www.ascc.net/xml/schematron/>>.

## URI

- [50] <<http://www.imc.org/ietf-xml-use/>>
- [51] <<http://xml.org/>>
- [52] <<http://xmlhack.com/>>
- [53] <<http://oasis-open.org/>>

## 11 Adresse des auteurs

Scott Hollenbeck	Marshall T. Rose	Larry Masinter
VeriSign, Inc.	Dover Beach Consulting, Inc.	Adobe Systems Incorporated
21345 Ridgetop Circle	POB 255268	Mail Stop W14
Dulles, VA 20166-6503	Sacramento, CA 95865-5268	345 Park Ave.
US	US	San Jose, CA 95110
		US
téléphone : +1 703 948 3257	téléphone : +1 916 483 8878	téléphone: +1 408 536 3024
mél : shollenbeck@verisign.com	mél : mrose@dbc.mtview.ca.us	mél : LMM@acm.org
		URI : <a href="http://larry.masinter.net">http://larry.masinter.net</a>

## 12 Déclaration de copyright

Copyright (C) The Internet Society (2003). Tous droits réservés.

Le présent document et ses traductions peuvent être copiés et fournis aux tiers, et les travaux dérivés qui les commentent ou les expliquent ou aident à leur mise en œuvre peuvent être préparés, copiés, publiés et distribués, en tout ou partie, sans restriction d'aucune sorte, pourvu que la déclaration de copyright ci-dessus et le présent et paragraphe soient inclus dans toutes telles copies et travaux dérivés. Cependant, le présent document lui-même ne peut être modifié d'aucune façon, en particulier en retirant la notice de copyright ou les références à la Internet Society ou aux autres organisations Internet, excepté autant qu'il est nécessaire pour le besoin du développement des normes Internet, auquel cas les procédures de copyright définies dans les procédures des normes Internet doivent être suivies, ou pour les besoins de la traduction dans d'autres langues que l'anglais.

Les permissions limitées accordées ci-dessus sont perpétuelles et ne seront pas révoquées par la Internet Society ou successeurs ou ayant droits.

Le présent document et les informations qu'il contient sont fournies sur une base "EN L'ÉTAT" et LE CONTRIBUTEUR, L'ORGANISATION QU'IL OU ELLE REPRÉSENTE OU QUI LE/LA FINANCE (S'IL EN EST), LA INTERNET SOCIETY ET LA INTERNET ENGINEERING TASK FORCE DÉCLINENT TOUTES GARANTIES, EXPRIMÉES OU IMPLICITES, Y COMPRIS MAIS NON LIMITÉES À TOUTE GARANTIE QUE L'UTILISATION DES INFORMATIONS CI-ENCLOSES NE VIOLENT AUCUN DROIT OU AUCUNE GARANTIE IMPLICITE DE COMMERCIALISATION OU D'APTITUDE À UN OBJET PARTICULIER.

### Remerciement

Le financement de la fonction d'édition des RFC est actuellement fourni par Internet Society.