

Groupe de travail Réseau  
**Request for Comments : 3367**  
 Catégorie : En cours de normalisation  
 Traduction Claude Brière de L'Isle

N. Popp & M. Mealling, VeriSign, Inc.  
 M. Moseley, Network, Inc.  
 août 2002

## Protocole de résolution de noms courants (CNRP)

### Statut de ce mémoire

Le présent document spécifie un protocole Internet en cours de normalisation pour la communauté de l'Internet, et appelle à des discussions et des suggestions pour son amélioration. Prière de se reporter à l'édition actuelle du STD 1 "Normes des protocoles officiels de l'Internet" pour connaître l'état de normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

### Notice de copyright

Copyright (C) The Internet Society (2002). Tous droits réservés

### Résumé

Les gens se réfèrent souvent aux choses du monde réel par un nom ou phrase courant, par exemple, un nom commercial, le nom d'une société, ou le titre d'un livre. Ces noms sont pour les gens parfois plus faciles à mémoriser et à taper que les URL. De plus, à cause de la syntaxe limitée des URL, les entreprises et les individus trouvent que les noms qui seraient les plus raisonnables pour leurs ressources sont déjà utilisés ailleurs et sont donc indisponibles. Pour les besoins du présent document, un "nom courant" est un mot ou phrase, sans structure syntaxique imposée, qui peut être associé à une ressource.

Cet essai concerne la création d'un protocole pour que les applications client communiquent avec les services de résolution de noms courants, comme en donnent des exemples l'amélioration du navigateur et les paradigmes de site de recherche. Bien que la principale fonction du protocole soit la résolution, il est aussi destiné à traiter les questions d'internationalisation et de localisation. Les services de résolution de nom ne sont pas des services de recherche génériques et ne fournissent pas d'interrogation booléenne complexe, de classement de pertinence ou de capacités similaires. Le protocole est un cœur interopérable simple et minimal. Des mécanismes d'extension sont fournis afin que puissent être ajoutées des capacités supplémentaires.

## Table des Matières

1. Introduction.....	2
2. Notes importantes.....	3
2.1 Terminologie.....	3
2.2 Le DTD est définitif.....	3
2.3 Identifiants de ressource universels.....	3
3. Modèle d'interaction.....	3
3.1 Services, serveurs, ensembles de données et référents.....	3
3.2 Demandes et réponses.....	3
3.3 Indépendance du transport.....	3
3.4 Codage des caractères.....	4
3.5 Interrogations.....	4
3.6 Indications.....	4
4. Modèle d'objets.....	5
4.1 Propriétés.....	5
4.2 Objets.....	6
5. DTD XML pour CNRP.....	14
6. Exemples.....	16
6.1 Demande de description de service.....	16
6.2 Envoi d'une interrogation et obtention d'une réponse.....	17
7. Transport.....	17
7.1 Transport HTTP.....	18
7.2 Transport SMTP.....	18
8. Enregistrement : application/cnrp+xml.....	18
9. Considérations sur la sécurité.....	18
10. Considérations relatives à l'IANA.....	19
Références.....	19

Appendice A. Propriété bien connue et gabarit d'enregistrement de type.....	20
A.1 Propriétés.....	20
A.2 Types.....	20
Appendice B Codes d'état.....	21
B.1 Codes de niveau 1 (Information).....	21
B.2 Codes de niveau 2 (Succès).....	21
B.3 Codes de niveau 3 (Succès partiel).....	22
B.4 Codes de niveau 4 (défaillance transitoire).....	22
B.5 Codes de niveau 5 (défaillance permanente).....	23
Adresse des auteurs.....	23
Déclaration complète de droits de reproduction.....	23

## 1. Introduction

Des services apparaissent qui offrent une transposition de noms courants en ressources de l'Internet (par exemple, celles identifiées par un URI). Ces services résolvent souvent des catégories de noms communs comme des noms de société, des noms commerciaux ou des mots clés courants. Donc, un tel service de résolution peut fonctionner dans une catégorie ou un petit nombre de catégories ou domaines, ou peut attendre du client qu'il limite la portée de résolution à un nombre limité de catégories ou domaines. Par exemple, la phrase "Équipe d'ingénierie de l'Internet" est un nom courant dans la catégorie "organisation", comme l'est "Moby Dick" dans la catégorie des romans.

Deux classes de clients de tels services sont construites, des améliorations de navigateur, et des services frontaux accessibles par la Toile. Les améliorations de navigateur modifient le champ "ouvrir" ou "adresse" d'un navigateur afin qu'un nom commun puisse être entré à la place d'un URL. Les sites de recherche de l'Internet intègrent les services de résolution de nom courant comme un complément de la recherche. Dans les deux cas, il peut y avoir des clients de services de résolution de finition. Dans le cas du navigateur, celui-ci doit parler à un service qui va résoudre le nom courant. Les sites de recherche sont accessibles via un navigateur. Dans certains cas, le site de recherche peut aussi être le service de finition de résolution, mais dans d'autres, le site de recherche est un frontal dans une collection de services de finition.

Le présent document concerne la création d'un protocole pour que les applications client communiquent avec les services de résolution de nom courant, dont les exemples sont le navigateur amélioré et les paradigmes de site de recherche. Les services de résolution de nom ne sont pas des services de recherche générique et n'ont donc pas besoin de fournir des interrogations booléennes complexes, des classements de pertinence ou des capacités similaires. Le protocole est un simple cœur interopérable minimal. Des mécanismes d'extension sont fournis, afin de pouvoir ajouter des capacités supplémentaires.

Plusieurs autres questions, bien qu'importantes pour le déploiement des services de résolution de noms courants, sont laissées en dehors du protocole de résolution lui-même et ne figurent pas dans la portée initiale du document proposé. Cela inclut la découverte et le choix des fournisseurs de service de résolution, l'administration des services de résolution, l'enregistrement des noms, la propriété des noms, et les méthodes de création, d'identification ou pour s'assurer de l'unicité des noms courants.

Pour les besoins du présent document, un "nom courant" est un mot ou une phrase, sans structure syntaxique imposée, qui peut être associé à une ressource. Ces noms courants seront principalement utilisés par les personnes, par opposition aux agents mécaniques. Un "service de résolution" de nom courant traite ces associations entre noms courants et données (ressources, informations sur les ressources, pointeurs sur les localisations, etc.). Un seul nom courant peut être associé à différents enregistrements de données, et plus d'un service de résolution est supposé exister. Tout nom courant peut être utilisé dans tout service de résolution.

Les noms courants ne sont pas des URI (Identifiants de ressource universels) en ce qu'il leur manque la structure syntaxique imposée par les URI ; de plus, à la différence des URN, il n'y a pas d'exigence d'unicité ou de persistance de l'association entre un nom courant et une ressource. (Noter que les noms courants peuvent être exprimés dans un URI dont la syntaxe est décrite dans la [RFC3368].)

Le présent document définit un protocole pour les paramètres de résolution nécessaires pour rendre les noms courants utiles. "Résolution" est défini comme la restitution des données associées (a priori) à des descripteurs qui correspondent à la demande d'entrée. "Paramétré" signifie la capacité à avoir un descripteur multi propriétés. Il n'est pas obligé que les descripteurs fournissent une identification univoque, donc zéro, un ou plusieurs enregistrements peuvent être retournés pour satisfaire une interrogation spécifique d'entrée.

## 2. Notes importantes

### 2.1 Terminologie

Dans le présent document, les mots clés "DOIT", "NE DOIT PAS", "EXIGÉ", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDÉ", "PEUT", et "FACULTATIF" sont à interpréter comme décrit dans le BCP 14, [RFC2119].

### 2.2 Le DTD est définitif

Les portions descriptives du présent document contiennent des éléments en XML qui sont des \*exemples seulement pour illustration\*. La Section 5 du présent document contient des déclarations de type de données (DTD, *Data Type Declaration*) XML pour CNRP, qui sont définitives. Si il y a des discordances, c'est la DTD qui s'impose.

### 2.3 Identifiants de ressource universels

Tous les URI utilisés dans CNRP DOIVENT se conformer à la production des "URI absolus" de l'ABNF de la [RFC2396]. CNRP ne définit pas la sémantique d'une Base et n'est donc pas capable d'exprimer la production de la "Référence d'URI".

## 3. Modèle d'interaction

### 3.1 Services, serveurs, ensembles de données et référents

CNRP suppose un modèle d'interaction particulier où un "service" généralisé fournit la résolution de nom courant à un ou plusieurs "serveurs" réels. Si les données contenues dans tous ces serveurs sont identiques (miroirs) le service n'a pas besoin d'identifier un sous ensemble particulier de données. Si cependant, le service fournit des collections de données différentes sur les différents serveurs (par exemple, des sous ensembles, des collections spécialisées, etc.) il DEVRAIT indiquer quels sous ensembles de ses données offre chaque serveur. On fait cela en utilisant les URI pour distinguer de façon univoque un ensemble de données d'un autre. Si le service offre une copie d'une collection de données en accord avec un service étranger, le service étranger DEVRAIT fournir un URI d'ensemble de données pour permettre que la collection soit identifiée par rapport à son offre propre.

CNRP prend en charge le concept de référents. C'est lorsque un serveur peut savoir qu'un autre service existe, au sein du même service ou ailleurs, qui peut fournir d'autres réponses sur une interrogation particulière mais décide de transmettre ce fait au client plutôt que de faire suivre l'interrogation pour le client. Un référent est envoyé avec le reste des résultats provenant d'un serveur (si il y en a). Les référents à un service DEVRAIENT indiquer l'URI de l'ensemble de données particulier qui a déclenché le référent, si il est connu. Voir au paragraphe 4.2.5 les détails sur les référents et la détection de boucle.

### 3.2 Demandes et réponses

Le protocole consiste en un mécanisme simple de demande/réponse. Un client envoie un des quelques types de demandes à un serveur qui répond par le résultat de cette demande. Toutes les demandes et réponses sont codées en XML [XML] en utilisant la DTD de la Section 5. Il y a deux types de demandes. Une est une interrogation générale pour un nom courant. L'autre est une demande pour un objet qui décrit le service et ses capacités. Il y a seulement un type de réponse qui est un ensemble de résultats. Les résultats peuvent contenir des éléments de résultat réels, des référents et/ou des messages d'état.

### 3.3 Indépendance du transport

CNRP est complètement encapsulé dans sa définition XML et est donc indépendant du transport dans sa spécification. Cependant, les clients ont besoin d'une définition claire des moyens de rattacher une connexion à un serveur.

Il est possible de définir des applications d'usage particulier qui utilisent CNRP mais n'ont jamais besoin de la méthode de rattachement HTTP présentée ci-dessous ; ces applications DOIVENT définir comment trouver le serveur/accès/protocole approprié. Les serveurs CNRP dédiés à ces applications ne peuvent fournir de service que sur les accès/protocoles de

transport définis par l'application.

Tous les autres clients et serveurs (génériques) CNRP DOIVENT prendre en charge le transport HTTP (paragraphe 7.1) sur l'accès CNRP par défaut de 1096.

Noter qu'un certain service peut choisir de passer à un transport ou accès différent via des déclarations au sein d'une demande de description de service CNRP, mais que les contacts initiaux entre un client et un serveur sont sur HTTP à l'accès 1096. Une brève explication de la façon dont CNRP emploie HTTP figure au paragraphe 7.1 du présent document. Si d'autres transports sont utilisés, ils DOIVENT être traités sur un accès autre que l'accès CNRP par défaut.

### 3.4 Codage des caractères

Pour garantir l'interopérabilité, les dispositions suivantes s'appliquent :

- o les interrogations et réponses XML DOIVENT être codées en UTF-8.
- Note : comme dans tout document XML, les références de caractères numériques peuvent être utilisées.
- o le codage des caractères dans l'URI CNRP se fonde sur UTF-8 ; voir les détails dans la [RFC3368].

Toutes les interfaces qui choisissent de présenter/accepter des éléments de protocole dans d'autres représentations sont responsables de la précision du transcodage à utiliser dans les éléments de protocole CNRP, selon les dispositions ci-dessus.

### 3.5 Interrogations

Les interrogations sont envoyées par le client au serveur. Il y a deux types d'interrogations.

1. Une interrogation initiale 'spéciale' qui établit le schéma pour une base de données CNRP particulière et le communique au client. Le client CNRP va envoyer cette interrogation, et va à son tour recevoir un document XML qui définit les propriétés de l'interrogation que prend en charge la base de données. (Dans CNRP, XML [XML] est utilisé pour définir et exprimer tous les objets.) Cette interrogation est appelée 'servicequery' dans la DTD. Dans le cas où un client ne sait rien sur le service, le client PEUT supposer qu'il peut au moins produire la demande via HTTP.
2. Une interrogation 'standard' qui est la soumission de la chaîne de recherche CNRP à la base de données. L'interrogation va se conformer au schéma qui PEUT avoir été précédemment récupéré auprès du service.

Il va y avoir un ensemble de propriétés d'interrogation, énumérées ci-dessous, qui sont traitées comme des indications par le serveur. Noter qu'une base de données CNRP va accepter toute propriété d'interrogation CNRP codée correctement ; la mesure dans laquelle un résultat d'interrogation est en rapport avec ces propriétés est un différenciateur de service. Les propriétés de base qui sont toujours prises en charge sont nom courant, langage, géographie, catégorie, et gamme (début et longueur de l'ensemble résultat). CNRP permet aux fournisseurs de service de base de données de créer des types de données uniques et de les exposer à tout client CNRP via les documents XML de schéma CNRP.

### 3.6 Indications

Une indication est une assertion de l'utilisateur à son sujet et sur le contexte dans lequel il fonctionne. Il n'y a pas de type de données 'indication' ; une indication est exprimée au sein de la structure de l'interrogation elle-même et est limitée ou est activée par la richesse de l'espace de noms défini par l'interrogation. En effet, une interrogation et toute propriété en son sein est une indication.

Par exemple, la propriété "langage" peut être donnée comme indication dans une interrogation ; cela peut être utilisé pour ordonner des résultats de recherches. Si on veut les résultats d'abord en anglais US suivi par du français et enfin de l'espagnol d'Amérique du sud, ce qui suit peut être inclus dans l'interrogation :

```
<property name="language" type="rfc1766">en-US</property>
<property name="language" type="rfc1766">fr-FR</property>
<property name="language" type="rfc1766">sp-MX</property>
```

Noter que les déclarations de propriétés ne disent rien sur le fait que le langage soit principal, secondaire, etc. Dans cet exemple, l'ordre de la déclaration contrôle que, la première déclaration, étant la première, signifie que l'anglais US est le langage principal. La seconde déclaration spécifie la seconde région/langage, et ainsi de suite. \*Mais c'est seulement un exemple.\* La mesure dans laquelle les indications sont acceptées (ou non) est un différenciateur de service.

Le fait qu'une indication existe ne signifie pas qu'une base de données CNRP doive y répondre. Cette approche au mieux est similaire au rang de pertinence d'un moteur de recherche (haute précision, faible souvenir) ; les indications sont similaires aux critères de sélection d'un moteur de recherche. Les services CNRP vont tenter de retourner les résultats "les plus proches" du critère de sélection. C'est assez différent de l'approche d'une base de données SQL où une interrogation SQL retourne l'ensemble complet de résultats et chaque résultat dans l'ensemble doit correspondre à toutes les exigences exprimées par le qualificateur (la clause SQL WHERE).

## 4. Modèle d'objets

### 4.1 Propriétés

Dans CNRP, les objets sont des listes de propriétés. Une propriété est un attribut nommé. Une propriété a aussi un type bien défini. Certaines propriétés peuvent faire partie de l'interrogation, ou de la liste des résultats, ou des deux. Pour rester simple, CNRP limite les valeurs de propriété à des valeurs de chaîne.

#### 4.1.1 Propriétés centrales

CNRP introduit un ensemble de propriétés centrales. Les propriétés centrales sont l'ensemble minimal de propriétés que tous les services CNRP DOIVENT prendre en charge afin d'être conformes à CNRP. Donc, les propriétés centrales définissent le niveau d'interopérabilité entre tous les services CNRP. Les propriétés centrales sont :

1. CommonName : le nom courant associé à une ressource.
2. ID : chaîne opaque qui sert d'identifiant univoque pour un résultat d'un service (normalement, un identifiant de base de données). L'ID n'est pas unique au monde, ni nécessairement persistant (par exemple, entre les interrogations à un certain service).
3. resourceURI : un 'URI absolu' comme défini dans l'ABNF de la [RFC2396].
4. description : une description textuelle de la ressource.

#### 4.1.2 Propriétés abstraites et personnalisées

En plus des propriétés centrales, CNRP introduit la notion de propriétés abstraites. L'élément propriété abstraite assure l'extensibilité du schéma au delà des propriétés centrales. La notion de propriété abstraite est extrêmement importante dans CNRP car elle permet une plus large gamme de services fondés sur CNRP que celle fondée sur les propriétés centrales.

Pour créer des propriétés personnalisées concrètes, un service CNRP doit définir un nom et un type de propriété. Donc, il y a réellement deux façons de créer une propriété personnalisée. La première façon est de créer un nouveau nom de propriété et de définir au moins un type pour lui. Une autre façon est d'étendre une propriété existante en définissant un nouveau type. La propriété "géographie" exposée au paragraphe suivant est un exemple de propriété multi-type. Noter qu'un type n'est applicable qu'à la propriété pour laquelle il est défini. Si une nouvelle propriété est définie, un nouveau type DOIT être défini même si la valeur établie pour ce type se trouve identique à celle d'un type existant pour une propriété existante. En d'autres termes, les types sont liés à une certaine propriété. Les propriétés personnalisées DOIVENT être enregistrées auprès de l'IANA. Les détails sur le processus d'enregistrement pour les nouvelles propriétés se trouvent à la Section 10.

Par exemple, supposons qu'un service CNRP spécialisé dans la vente de livres en ligne souhaite introduire la propriété ISBN de type "numéro". Cette propriété encapsulerait le numéro ISBN du livre en ligne et aurait la représentation XML :

```
<nom de propriété="isbn" type="numéro">92347231</propriété>
```

#### 4.1.3 Propriétés de base

Illustrant l'utilisation d'une propriété abstraite pour étendre le cœur du schéma, CNRP définit aussi un ensemble de propriétés personnalisées appelées propriétés de base. Afin de garder des exigences extrêmement simples, ces propriétés ne sont pas de mise en œuvre obligatoire pour la conformité à CNRP. Bien que ces propriétés ne soient pas exigées, on s'attend à ce que de nombreux services, en particulier les grands, les mettent en œuvre. Un but aussi important pour l'introduction de propriétés supplémentaires est de fournir un mécanisme de filtrage des résultats. C'est une exigence pour les grands espaces de noms qui contiennent plusieurs millions de noms.

Les propriétés de base et leurs types sont définis dans l'Appendice A mais on en donne ici la liste dans un souci de clarté :

o Langage :

C'est le langage associé à une ressource. Le type par défaut de cette propriété est 'RFC1766' et le vocabulaire est tiré de la

liste des langages de la [RFC1766]. Si la RFC 1766 est mise à jour, les valeurs énumérées dans la version mise à jour sont aussi valides pour ce type.

o Géographie :

C'est la région géographique ou la localisation associée à une ressource. Certains des types possibles sont énumérés ci-dessous. Voir à l'Appendice A une liste complète des types spécifiés par le présent document.

- \* 'freeform' : c'est une expression de forme libre pour une localisation géographique (par exemple, "palo alto en Californie").
- \* 'ISO3166-1' : région géographique exprimée en utilisant un code de pays standard défini par la norme ISO3166-1 (par exemple, "US").
- \* 'ISO3166-2' : valeur = région géographique exprimée en utilisant les codes de pays et de région standard définis par la norme ISO3166-2 (par exemple, "US-CA").
- \* 'lat-long' : latitude et longitude d'une localisation géographique.

o Catégorie :

C'est la catégorie associée à une ressource. Il y a un grand nombre de types possibles pour cette propriété. Parmi eux :

1. 'freeform' : expression de forme libre pour une catégorie (par exemple, "films").
2. 'NAICS' : système de code de l'industrie d'Amérique du nord (*North American Industry Code System*).

o Gamme :

La gamme est une propriété de contrôle d'ensemble de résultats. La propriété gamme est utilisée pour spécifier le point de départ et la longueur d'un ensemble de résultats (par exemple, Je veux cinq enregistrements en commençant pas le 10<sup>ème</sup> enregistrement). On ne devrait jamais avoir qu'un seul type mais, dans un souci d'extensibilité et de cohérence, d'autres peuvent être créés en cas de besoin. Le type par défaut est 'start-length' qui prend la forme de deux entiers séparés par un trait d'union. Le premier entier est le numéro de début et le second est le nombre de valeurs qu'il comporte.

o Dataseturi : URI absolu (comme défini dans la [RFC2396]) qui identifie un ensemble défini de noms courants et des données associées.

Note : Pour de nombreuses propriétés le type par défaut est "freeform". La valeur du type de forme libre est importante parce que elle permet une interface d'utilisateur très simple où l'utilisateur peut entrer une valeur dans un champ de texte. Il appartient au service d'interpréter correctement la valeur et d'en tirer parti pour augmenter la pertinence des résultats (par exemple, en utilisant des dictionnaires spécialisés).

#### 4.1.4 Codage de chaîne de nom courant et règles d'équivalence

CNRP spécifie que les chaînes de nom courant devraient être codées en UTF-8. CNRP ne spécifie aucune règle d'équivalence de chaîne pour faire correspondre un nom courant dans l'interrogation avec un nom courant d'une ressource. Les règles d'équivalence de chaîne dépendent du langage et du service. Elles sont spécifiques des algorithmes de classement d'équivalence, et donc traitées comme des services CNRP. Par conséquent, les règles d'équivalence de chaîne ne font pas partie de la spécification du protocole CNRP. Par exemple, le membre d'interrogation :

```
<commonname>bmw</commonname>
```

devrait être lu comme un critère de choix pour une ressource avec un nom courant COMME (similaire à) la chaîne "bmw" où la définition exacte de l'opérateur COMME est intuitive, et donc spécifique du service CNRP interrogé.

Il est aussi important de noter que XML traite les espaces comme un cas particulier dans de nombreuses situations. Dans certains cas, il écrase les espaces en une seule. Les mises en œuvre de client aussi bien que de serveur doivent référencer la norme XML pour viser la variante d'utilisation des espaces dans les interrogations et/ou les résultats.

## 4.2 Objets

### 4.2.1 Query

L'objet Query (*interrogation*) encapsule tous les composants d'interrogation tels que CommonName, ID, et toute propriété. Un objet Query ne peut pas être vide. Query doit contenir soit un seul nom courant, soit un seul identifiant. Query peut aussi contenir les propriétés personnalisées définies par un service CNRP spécifique.

Par exemple, une interrogation sur les cinq premières ressources dont le nom courant est comme "bmw" serait exprimée par :

```
<query>
  <commonname>bmw</commonname>
  <property name="range" type="start-length">1-5</property>
</query>
```

#### 4.2.1.1 Opérations logiques au sein de Query

La syntaxe de Query est extrêmement simple. CNRP n'apporte pas un soutien fort aux opérateurs logiques booléens OU, ET ou NON. Cependant, il existe deux opérations logiques implicites qui peuvent être exprimées à travers l'objet Query et ses propriétés. D'abord, une interrogation avec plusieurs paires propriété-valeur exprime implicitement une opération ET sur les termes de l'interrogation. Par exemple, l'interrogation CNRP pour demander toutes les ressources dont le nom courant est comme "bmw", ET dont le langage est "Allemand" peut être exprimée par :

```
<query>
  <commonname>bmw</commonname>
  <property name="language" type="rfc1766">de-DE</property>
</query>
```

Noter cependant que comme le serveur essaye seulement de trouver la meilleure correspondance au critère de l'interrogation, il n'est pas garanti que toutes les ressources des résultats, s'il en est, satisfassent aux exigences.

De plus, CNRP permet au client d'exprimer un OU logique en spécifiant plusieurs valeurs pour la même propriété au sein de l'interrogation. Par exemple, l'expression logique :

propriété = valeur1 OU propriété = valeur2 OU propriété = valeurN

sera exprimé par :

```
<propriété>valeur1</propriété>
<propriété>valeur2</propriété>
<propriété>valeurN</propriété>
```

De sorte que si différentes propriétés sont exprimées, CNRP les ajoute comme avec l'opérateur logique ET ; si il y a plusieurs instances de la même propriété qui sont exprimées, CNRP les compare comme avec l'opérateur logique OU.

Il est important de souligner que cette forme n'est applicable qu'aux propriétés (à l'exception du nom courant lui-même qui, même si il est une propriété, est l'objet entier de l'interrogation). En particulier, les opérations logiques OU sur le nom courant ne sont pas acceptées. Noter que l'ordre des paires propriété-valeur dans l'interrogation implique une préséance. Par conséquent, CNRP introduit aussi une valeur de chaîne particulière : "\*". On ne sera pas surpris que "\*" signifie toutes les valeurs admissibles pour le type de propriété. Par exemple, l'interrogation qui suit demande toutes les ressources dont le nom courant est comme BMW et dont le langage est de préférence l'allemand ou le français ou toute autre langue.

```
<query>
  <commonname>bmw</commonname>
  <property name="language" type="rfc1766">de-DE</property>
  <property name="language" type="rfc1766">fr-FR</property>
  <property name="language" type="rfc1766">*</property>
</query>
```

## 4.2.2 Results

L'objet Results est un conteneur pour les résultats de CNRP. Le type d'objets contenu dans Results peut être : ResourceDescriptor, Error, Referral et Schema. Les résultats d'un service CNRP sont rangés par ordre de pertinence décroissante. Lorsque l'ensemble de résultats contient des résultats provenant de plusieurs services CNRP, le résultat ne peut plus être rangé (parce que l'ordre de pertinence est spécifique d'un certain service). Dans ce cas, on notera cependant que les résultats générés par le même service restent ordonnés.

### 4.2.2.1 ResourceDescriptor

L'objet ResourceDescriptor (*descripteur de ressource*) décrit une ressource de l'Internet (par exemple, une page de la Toile, une personne, tout objet identifié par un URI). Donc, le ResourceDescriptor DOIT toujours comporter la propriété resourceURI. Le ResourceDescriptor peut aussi contenir le nom courant, l'URI, l'identifiant (de cette entrée dans la base de

données du service) la description, le langage, la géographie, et la catégorie de la ressource. Un ResourceDescriptor peut aussi être augmenté des propriétés personnalisées et peut faire référence à un objet de service pour indiquer son origine (en utilisant l'élément serviceRef). Comme avec les références, un bloc resourcedescriptor peut aussi contenir un attribut ID qui est utilisé par un message d'état pour se référer à un resourcedescriptor particulier. Il faut faire attention à ne pas confondre cet identifiant avec l'étiquette d'identifiant qui se réfère à l'identifiant de base de données de l'entrée réelle de la base de données.

```
<results>
  <service id="i0">
    <serviceuri>http://cnrp.bar.com/</serviceuri>
  </service>
  <resourcedescriptor id="i1">
    <commonname>bmw</commonname>
    <id>foo.com:234364</id>
    <resourceuri>http://www.bmw.de/</resourceuri>
    <serviceref ref="i0" />
    <description>BMW Motorcycles, International</description>
    <property name="language" type="rfc1766">de-DE</property>
  </resourcedescriptor>
  <referral>
    <serviceref ref="i0" />
  </referral>
</results>
```

### 4.2.3 Service

L'objet Service fournit une encapsulation d'une instance d'un service CNRP. Un service est identifié de façon univoque par l'étiquette serviceuri qui DOIT être incluse dans l'objet Service. Un objet Service PEUT inclure une brève description textuelle du service. Elle PEUT inclure des propriétés datasets, servers et custom.

```
<service>
  <serviceuri>http://cnrp.foo.com</serviceuri>
  <description>foo.com est un service CNRP spécialisé dans les recettes de cocktails</description>
</service>
```

L'objet Service PEUT aussi être étendu en incluant des propriétés existantes pour décrire le service. Par exemple, un service qui se concentre sur les sociétés françaises pourrait être exprimé par :

```
<service>
  <serviceuri>http://cnrp.foo.com</serviceuri>
  <property name="category" type="freeform">companies</property>
  <property name="geography" type="ISO3166-1">FR</property>
</service>
```

#### 4.2.3.1 Datasets

L'objet dataset représente un ensemble de transpositions de CN en URI. Par exemple, la base de données des mots clés AOL et leurs URI constitue un ensemble de données (*dataset*). L'objet dataset permet à une mise en œuvre de CNRP d'identifier de façon univoque l'identité de la ou des bases de données des transpositions qu'elle effectue. À cet égard, la notion d'ensemble de données permet une séparation entre la résolution et les données, fournissant le mécanisme pour qu'un service CNRP résolve les noms courants au nom d'un autre service ou même de plusieurs services CNRP. À l'inverse, le même ensemble de données peut être desservi par deux services CNRP distincts. Comme un service CNRP peut résoudre des noms au sein d'un ou plusieurs ensembles de données, l'objet Service peut contenir un ou plusieurs autres objets dataset (zéro si l'ensemble de données n'est pas formellement déclaré).

Au sein de l'objet Service, un ensemble de données est défini de façon univoque en utilisant la propriété dataseturi. D'autres propriétés, telles que langage et description, peuvent décrire plus avant l'ensemble de données. Comme l'objet Service, l'objet dataset a un attribut ID associé qui est unique au sein d'un message XML particulier. Comme l'attribut ID de l'objet Service, cet identifiant est utilisé par les descripteurs de ressource et les référents pour spécifier de quel service et/ou ensemble de données ils viennent ou se réfèrent.

Tout service peut être dit avoir un "ensemble de données par défaut" qui est l'ensemble de données qu'il considère avoir été

utilisé si un serveur répond simplement à l'interrogation d'un client qui ne contenait pas d'ensemble de données. L'ensemble de données par défaut peut aussi être dit être le seul ensemble de données utilisé par les services qui ne prennent pas du tout en charge les ensembles de données. Ce concept est utile pour les clients qui ont l'intention de faire une détection rigoureuse de boucles au moyen de la tenue d'une liste des nœuds de service/ensembles de données visités.

L'exemple suivant illustre à quoi ressemble l'objet Service lorsque il définit deux datasets :

```
<service id="i0">
  <serviceuri>http://acmecorp.com</serviceuri>
  <dataset id="i1">
    <property name="dataseturi">
      urn:oid:1.2.3.4.666.5.4.3.1
    </property>
    <property name="language">en-us</property>
    <property name="language">en-gb</property>
  </dataset>
  <dataset id="i2">
    <property name="dataseturi">
      urn:oid:1.2.3.4.666.10.9.8.7.6
    </property>
    <property name="language">fr</property>
  </dataset>
</service>
```

La propriété dataseturi peut aussi être utilisée au sein d'une interrogation comme un conseil au service pour l'ensemble de données au sein duquel le nom courant devrait être résolu :

```
<query>
  <commonname>toys r us</commonname>
  <property name="dataseturi">urn:oid:1.2.3.4.666.5.4.3.1</property>
</query>
```

Il est important de noter que les règles de résolution (c'est-à-dire, l'équivalence de chaîne, l'ordre de pertinence, etc.) vont vraisemblablement être spécifiques de l'ensemble de données. Ceci est vrai même si la résolution est fournie par le même service.

Une autre utilisation de la propriété dataseturi est dans un référent. Dans ce cas, l'étiquette datasetref est utilisée pour épingler un ensemble de données spécifique dans le service.

```
<referral>
  <serviceref ref="i0" /><datasetref ref="i1" />
</referral>
```

Bien que le concept d'ensemble de données soit important pour les services qui souhaitent rendre leurs données disponibles via d'autres services, il est important de se rappeler que la déclaration et l'utilisation des ensembles de données sont complètement facultatives. La conformité au protocole CNRP n'exige pas qu'un objet Service définisse ou fasse référence à un objet dataset. La seule exigence pour la conformité est qu'un client et/ou serveur connaisse le format des étiquettes XML particulières et les traite selon la syntaxe. Si il choisit de les ignorer, c'est parfaitement son droit.

#### 4.2.3.2 Serveurs

L'objet Service encapsule aussi une liste d'objets server. L'objet server est utilisé pour décrire un serveur ou un ensemble de serveurs CNRP. Un serveur est identifié par ses serveruri. L'URI utilisé pour identifier un serveur n'est pas un URI CNRP [RFC3368], mais est plutôt un URI du schéma utilisé comme mécanisme de transport CNRP. C'est-à-dire que pour un serveur CNRP qui va communiquer via le protocole HTTP avec l'hôte foo.com sur l'accès 6543, le serveruri serait http://foo.com:6543. Si d'autres informations sont nécessaires afin que le transport correct soit utilisé, ces informations peuvent alors être communiquées via d'autres propriétés. Noter qu'un service DOIT avoir au moins un serveur qui répond sur l'accès CNRP par défaut afin qu'un client obtienne l'objet Service initial.

Un serveur peut desservir un ou plusieurs ensembles de données déclarés par son service. Les bases de données des services sont spécifiées en utilisant la propriété dataseturi. Comme pour les autres objets, un serveur peut être plus décrit en utilisant des propriétés descriptives telles que geography et description. L'XML suivant complète la définition de service de

l'exemple précédent en définissant deux serveurs CNRP. Un serveur est situé aux USA et l'autre est situé en France. Le serveur US est spécialisé et ne dessert que l'ensemble de données français.

```
<servers>
  <server>
    <serveruri>cnrp://router.us.widgetco.com:4321</serveruri>
    <property name="geography" type="ISO3166-1">US</property>
  </server>
  <server>
    <serveruri>cnrp://router.fr.acmeco.com:4321</serveruri>
    <property name="geography" type="ISO3166-1">FR</property>
  </server>
</servers>
```

Comme on le verra dans une section suivante, l'objet Service peut contenir des objets Schema. Ces objets Schema décrivent pleinement les interfaces d'interrogation et de réponse mises en œuvre par un service CNRP. À cet égard, l'objet Service est essentiel pour la capacité de découverte. Il constitue le principal point d'entrée pour qu'un client CNRP découvre de façon dynamique les capacités d'un service de résolution. À cette fin, l'objet Service peut être retourné au titre de la réponse à toute interrogation de résolution. De plus, l'objet Service est la réponse dédiée à l'interrogation de service spécialisée (voir au paragraphe 4.2.6).

Une autre utilisation de Service est pour que d'autres objets indiquent leur service CNRP d'origine. Des messages système, des référents et des descripteurs de ressource peuvent inclure une référence à leur objet Service. Par exemple, imaginons un service CNRP qui agit comme mandataire pour plusieurs services CNRP. Par exemple, il est exigé que CNRP permette des agrégations de résultats provenant de différentes sources. Considérons un tel service CNRP qui agit comme mandataire pour plusieurs services CNRP. Dans ce mode, le service mandataire contacte chaque sous service CNRP en parallèle ou à la suite. Ensuite, le mandataire combine les ensembles de résultats individuels en une réponse unique retournée au client CNRP. Comme l'ensemble de résultats agrégés contient les descripteurs de ressource provenant de différents services, le mandataire ajoute une étiquette servicereference au sein de chaque résultat individuel pour indiquer leur service d'origine. Dans le cas où un des services référencés résout des noms dans plusieurs ensembles de données, il est possible que ces objets se réfèrent à un ensemble spécifique de données au sein du service en utilisant l'étiquette datasetref. Cet exemple est un résultat hybride avec des descripteurs de ressource qui font référence à leur service et dataset d'origine :

```
<?xml version="1.0"?>
<!DOCTYPE cnrp PUBLIC "-//IETF//DTD CNRP 1.0//EN" "http://ietf.org/dtd/cnrp-1.0.dtd">
<cnrp>
  <results>
    <service id="i0">
      <serviceuri>http://acmecorp.com</serviceuri>
      <dataset id="i1">
        <property name="dataseturi">urn:oid:1.2.3.4.666.5.4.3.1</property>
      </dataset>
      <dataset id="i2">
        <property name="dataseturi">urn:oid:1.2.3.4.666.10.9.8.7.6</property>
      </dataset>
    </service>
    <service id="i3">
      <serviceuri>http://serverfarm.acmecorp.com</serviceuri>
    </service>
    <service id="i4">
      <serviceuri>http://servers.acmecorp.co.uk</serviceuri>
      <dataset id="i5">
        <property name="dataseturi">urn:oid:1.2.3.4.666.5.4.3.1</property>
      </dataset>
    </service>
    <resourcedescriptor>
      <commonname>Fidonet</commonname>
      <id>1333459455</id>
      <resourceuri>http://www.fidonet.ca</resourceuri>
      <serviceref ref="i0" /><datasetref ref="i1" />
      <description>C'est le vieux Fidonet canadien</description>
    </resourcedescriptor>
  </results>
</cnrp>
```

```

    <commonname>Fidonet</commonname>
    <id>1333459455</id>
    <resourceuri>http://host:port/bla</resourceuri>
    <serviceref ref="i3" />
    <description>Un vieux nœud Fidonet</description>
  </resourcedescriptor>
  <referral>
    <serviceref ref="i0" /><datasetref ref="i2" />
  </referral>
</results>
</cnrp>

```

## 4.2.4 Messages d'état

### 4.2.4.1 État de CNRP, pas du transport

Les messages d'état définis ici ne sont applicables qu'aux opérations définies par CNRP lui-même. Si des caractéristiques ou opérations sont définies par le transport (sécurité via HTTP, échec de messagerie via SMTP, etc.) tout message d'état sur cette opération DOIT alors être envoyé en accord avec le mécanisme de rapport de ce transport et non via CNRP.

### 4.2.4.2 Codes et description

Un objet Status indique un message pour le client dans l'ensemble de résultats. L'objet encapsule deux valeurs : un code d'état et une description. La description peut contenir une description textuelle de l'état communiqué. Dans de nombreux cas, des informations supplémentaires de diagnostic pourront aussi être incluses. On n'essayera pas de normaliser la description d'un code d'état car le seul élément programmatique qui importe est le code réel.

Un message d'état peut aussi spécifier à quel autre élément CNRP il se réfère en incluant une référence à l'identifiant de l'élément en question. Par exemple, si un bloc Service a un identifiant de "i2" et si un message d'état se réfère à ce bloc, il peut alors mettre cet identifiant dans son attribut ref.

```
<status code="x.y.z" ref="i2">La base de données CNRP foo.com est temporairement inaccessible</status>
```

### 4.2.4.3 Codes d'état

L'organisation des codes d'état est tirée de la [RF1893] qui structure ses codes sous la forme x.yyy.zzz. L'ABNF des codes est tiré de la RFC1893 :

```

code d'état = classe "." sujet "." détail
classe = "2"/"3"/"4"/"5"
sujet = 1*3chiffre
détail = 1*3chiffre

```

Les codes de niveaux supérieurs notent les niveaux de sévérité de l'état :

- o 1.X.X Informatif
  - \* Les informations portées par le code n'ont pas d'incidence sur le succès ou l'échec d'une demande. Elles sont strictement à des fins d'information.
- o 2.X.X Succès
  - \* La demande a été traitée et les résultats retournés. Dans la plupart des cas, cette classe d'état ne sera pas envoyée car les résultats réels eux-mêmes révèlent la réussite. Dans d'autres cas, les résultats ont été retournés mais certaines informations doivent être retournées au client.
- o 3.X.X Succès partiel
  - \* La demande a été traitée et les résultats retournés. Dans ce cas cependant, certaines valeurs envoyées avec la demande étaient soit invalides, soit ont été ignorées, mais de telle façon que le serveur considère quand même que la réponse est un succès et n'indique aucune vraie condition d'erreur.
- o 4.X.X Échec provisoire
  - \* La demande était valide telle qu'envoyée, mais certains événements temporaires empêchent la réussite de l'achèvement de la demande et/ou l'envoi des résultats. Un envoi futur reste possible.

- o 5.X.X Échec permanent
  - \* Un échec permanent est celui qui ne pourra sans doute pas être résolu par le renvoi de la demande sous sa forme actuelle. Des changements à la demande ou à la destination doivent être faits pour qu'elle réussisse.

Les codes de second niveau indiquent le sujet des messages d'état. Cette valeur s'applique à chacune des cinq classes. Le sous code de sujet, si il est reconnu, doit être rapporté même si les détails supplémentaires apportés par le sous code de détail n'est pas reconnu. Les valeurs énumérées pour le sous code de sujet sont :

- o X.0.X Autre état ou état indéfini
  - \* Aucune information spécifique n'est disponible sur la classe de sujet à laquelle appartient ce message.
- o X.1.X En rapport avec l'interrogation
  - \* Tout état en rapport avec le codage spécifique de l'interrogation ou ses valeurs, à l'exception des propriétés.
- o X.2.X En rapport avec le service
  - \* Tout état qui se rapporte au service à la fourniture duquel ce serveur coopère.

L'Appendice B contient une liste de tous les codes d'état prédéfinis.

#### 4.2.5 Referral

Un objet Referral (*réfèrent*) dans l'ensemble de résultats est un fourre tout pour des résultats non ordonnés provenant d'un service et éventuellement ensemble de données différent. Des référents surviennent normalement lorsque un serveur CNRP connaît l'existence d'un autre service capable de fournir des résultats pertinents pour l'interrogation et veut notifier cette possibilité au client. Le client peut décider si il veut suivre la référence et résoudre les résultats supplémentaires en contactant le service référencé en utilisant les informations contenues dans l'objet Referral (un objet Service et d'éventuelles propriétés). Le Referral est un mécanisme simple pour permettre une résolution hiérarchique ainsi que pour joindre ensemble plusieurs services de résolution.

```
<results>
  <service id="i0">
    <serviceuri>http://cnrp.bar.com/</serviceuri>
    <dataset id="i1">
      <property name="dataseturi">urn:oid:1.3.6.1.4.1.782.1</property>
    </dataset>    <dataset id="i2">
      <property name="dataseturi">urn:oid:1.3.6.1.4.1.782.2</property>
    </dataset>
  </service>
  <resourcedescriptor>
    <commonname>bmw</commonname>
    <id>foo.com:234364</id>
    <resourceuri>http://www.bmw.de/</resourceuri>
    <serviceref ref="i0" /><datasetref ref="i1" />
    <description>BMW Motorcycles, International</description>
    <property name="language" type="iso646">de-DE</property>
  </resourcedescriptor>
  <referral>
    <serviceref ref="i0" /><datasetref ref="i2" />
  </referral>
</results>
```

Comme les autres objets CNRP, un référent peut être mieux décrit en utilisant des propriétés personnalisées. Comme un descripteur de ressource, un référent peut avoir un attribut ID qui est utilisé par un message d'état pour parler d'un bloc de référence particulier.

##### 4.2.5.1 Détection de boucle et traitement des ensembles de données chez les serveurs

Dans CNRP, les référents peuvent être traités de trois façons :

- o spécifique de l'application,
- o seulement comme indication,
- o détection rigoureuse de boucle.

Dans les deux premiers cas, le comportement du client, lorsque il reçoit un référent, n'est pas défini dans ce mémoire. Le client peut voir dans le référent un simple conseil. Dans ce cas, les ensembles de données peuvent être traités ou non. La détection de boucle peut n'être rien de plus que "Ai-je déjà parlé à ce nom d'hôte ?" ou "Arrêter après le 3<sup>ème</sup> référent". Ces deux cas vont très vraisemblablement s'appliquer à des mises en œuvre simples ou contraintes où les clients et les serveurs ont une connaissance a priori de leurs capacités. Sans une telle connaissance, il y a trop d'ambiguïtés vis-à-vis des services et des ensembles de données pour que les clients fassent une détection de boucle fiable.

Le dernier cas est lorsque le client s'attend à parler à plusieurs serveurs qui pourraient ne rien savoir l'un de l'autre. Ce cas exprime la sémantique de base de ce qu'un serveur devrait dire à un client si il comprend les ensembles de données ou les référents. Comme un référent spécifie l'ensemble de données exact auquel il se réfère, un nœud dans la liste des nœuds visités est constitué d'un serviceuri et d'un dataseturi. Ces deux valeurs doivent être prises en considération durant la détection de boucle. Dans le cas où un service ne prendrait pas en charge les ensembles de données, le nœud visité sera constitué du service et de l'ensemble de données par défaut.

L'élément majeur à se remémorer lors de la détection de boucle à travers plusieurs serveurs est que certains serveurs peuvent ne pas comprendre du tout les ensembles de données, alors que d'autres s'appuient spécifiquement sur eux. Pour aider à déterminer comment doivent être marqués les nœuds de détection de boucle, on a défini trois messages d'état spécifiques :

Le message d'état 3.1.3 (Ensembles de données non acceptés) est utilisé pour noter que le serveur n'accepte pas du tout les ensembles de données. Il est envoyé en réponse à une interrogation qui contient des ensembles de données. Le client devrait considérer que le serveur a ignoré les ensembles de données et le client devrait considérer ce nœud comme ayant été visité pour tous les ensembles de données possibles (y compris l'ensemble de données par défaut).

Le message d'état 3.1.4 (Premier ensemble de données seul pris en charge) est utilisé par un serveur pour indiquer une situation où un client a inclus plusieurs dataseturis dans son interrogation et où le serveur peut seulement en prendre en charge un à la fois. Dans ce cas, le serveur déclare explicitement qu'il a utilisé seulement le premier dataseturi. Le client devrait considérer que seul le premier dataseturi spécifié a été traité correctement. Le client devrait considérer que les ensembles de données restants dans l'interrogation ont été ignorés complètement. Ils devraient être envoyés individuellement comme référents si le client se soucie réellement de leurs résultats. Seul la première paire serviceuri/dataseturi devrait être marquée comme visitée.

Le message d'état 3.1.5 (Cet ensemble de données n'est pas accepté) est utilisé pour indiquer qu'un dataseturi spécifique envoyé dans une interrogation par un client n'est pas accepté par le serveur. Cette paire serviceuri/dataseturi devrait être considérée comme visitée par le client. Si ce message est envoyé en réponse à une interrogation qui spécifie plusieurs ensembles de données, le client devrait se comporter de la même façon que si il avait reçu le message 3.1.3 ci-dessus. Ce devrait être considéré comme une mauvaise forme pour un serveur de renvoyer ce message d'état en réponse à une interrogation avec plusieurs ensembles de données, parce que c'est ambigu.

Bien qu'il n'y ait pas d'algorithme exact pour la détection de boucle que les clients soient invités à prendre en charge, ces messages d'état peuvent être utilisés par le serveur pour des éclaircissement sur les services et ensembles de données qu'il considère comme ayant été interrogés. Il appartient au client de décider que faire de ces messages et du niveau de précision des tentatives de détection de boucle.

#### 4.2.6 Découvrabilité : ServiceQuery et Schema

Une sous classe de Query, l'objet ServiceQuery accepte la découverte dynamique des caractéristiques du service CNRP spécifique. Noter que la conformité à CNRP n'exige pas qu'un service mette pleinement en œuvre la découvrabilité. En particulier, retourner l'objet Service avec son serviceuri constitue une mise en œuvre minimale et donc suffisante de la conformité. Néanmoins, on s'attend à ce que des services CNRP évolués choisissent de retourner une description complète des interfaces prises en charge.

La réponse complète à une interrogation de service (*servicequery*) retourne l'objet Service décrit au paragraphe 5.3.2 avec les informations de schéma suivantes :

1. les propriétés de base et personnalisées utilisées par le service CNRP (schéma Property),
2. les propriétés utilisées pour décrire l'objet Service (schéma Service),
3. les propriétés qui appartiennent à l'interface d'interrogation (schéma Query),
4. les propriétés qui appartiennent à une ressource au sein du résultat (schéma Resource).

Cela conduit aux définitions de nouvel objet suivantes :

- o *propertyschema* – c'est un schéma de propriété qui décrit toutes les propriétés personnalisées qui font partie du service.

- o propertydeclaration – une déclaration de propriété décrit une propriété de base ou personnalisée utilisée par le service CNRP. Une déclaration de propriété a un nom et un type (le nom et le type de la propriété à laquelle ils se réfèrent). Noter qu'au titre du schéma de propriété, on DOIT déclarer les propriétés existantes ainsi que celles nouvellement définies.
- o propertyreference – c'est une référence à une déclaration de propriété afin qu'un certain schéma (de service, d'interrogation ou de ressource) puisse déclarer la propriété au sein de son interface. Noter qu'une référence de propriété spécifie si l'utilisation de la propriété est obligatoire ou seulement facultative.
- o serviceschema -- le schéma de service définit les propriétés utilisées pour décrire le service.
- o querieschema – un schéma d'interrogation décrit la structure d'une interrogation traitée par le service CNRP. Les propriétés référencées dans le schéma d'interrogation font partie de l'interface d'interrogation du service de résolution.
- o resourcedescriptorschema – un schéma de descripteur de ressource décrit la ressource retournée comme résultat par le service CNRP.

Par exemple, une interrogation CNRP pour découvrir les capacités d'un service aurait la forme suivante :

```
<cnrp> <servicequery/> </cnrp>
```

Et pour un service CNRP pour des recettes de cocktail en français, la réponse correspondante serait :

```
<service>
  <serviceuri>http://cnrp.recipe.com</serviceuri>
  <propertyschema>
    <propertydeclaration id="i1">
      <propertyname>language</propertyname>
      <propertytype>rfc1766</propertytype>
    </propertydeclaration>
    <propertydeclaration id="i2">
      <propertyname>cocktailrecipe</propertyname>
      <propertytype>freeform</propertytype>
    </propertydeclaration>
  </propertyschema>
  <querieschema>
    <propertyreference required="yes" ref="i1"/>
  </querieschema>
  <resourcedescriptorschema>
    <propertyreference required="yes" ref="i1"/>
    <propertyreference required="yes" ref="i2"/>
  </resourcedescriptorschema>
</service>
```

Cette réponse stipule que le service accepte le langage de la propriété au titre de l'interface d'interrogation et retourne des descripteurs de ressource qui contiennent les deux propriétés language et cocktailRecipe.

## 5. DTD XML pour CNRP

```
<!--Étiquette du document -->
```

```
<!ELEMENT cnrp (query|results|servicequery)>
```

```
<!--Utilisé pour demander un objet Service-->
```

```
<!ELEMENT servicequery VIDE>
```

```
<!--Une interrogation peut demander un schéma, un enregistrement spécifique par identifiant, ou un nom courant-->
```

```
<!--avec un ensemble de propriétés (ou assertions) sur l'entité qui fait l'interrogation.-->
```

```
<!ELEMENT query (id|(commonname,property*))>
```

```
<!ELEMENT id (#PCDATA)>
```

```
<!ELEMENT commonname (#PCDATA)>
```

```
<!--Note : CNRP définit plusieurs propriétés et types bien connus. Voir les détails à l'Appendice A.-->
```

```
<!ELEMENT property (#PCDATA)>
```

```
<!--Le nom de la propriété-->
```

```
<!ATTLIST property name CDATA #REQUIRED>
```

```

<!--Le type de la propriété-->
<!ATTLIST property type CDATA "freeform">
<!ELEMENT results (status? |
    ( service+,
      ( status | resourcedescriptor | referral )*
    )*
)>

<!ELEMENT resourcedescriptor (commonname,id,resourceuri,
    serviceref, datasetref?,
    description,
    property*)>
<!ATTLIST resourcedescriptor id ID #IMPLIED>
<!--C'est l'objet de tout ceci...-->
<!ELEMENT resourceuri (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT referral (serviceref, datasetref?)>
<!ATTLIST referral id ID #IMPLIED>
<!ELEMENT status (#PCDATA)>
<!ATTLIST status code CDATA #REQUIRED>
<!ATTLIST status ref IDREF #IMPLIED>

<!--serviceRef est utilisé pour pointer sur un des objets service de l'ensemble fourni. C'est ainsi qu'une ressource-->
<!--peut pointer sur le service dont elle vient. On peut inclure l'objet service entier mais il faudrait alors répéter une-->
<!--grande quantité d'informations.-->
<!ELEMENT serviceref EMPTY>
<!ATTLIST serviceref ref IDREF #IMPLIED>
<!ELEMENT service (serviceuri, dataset*,
    servers?,
    description?,
    property*,propertyschema?,queryschema?,resourcedescriptorschema?,serviceschema?)>
<!--Durée de vie du schéma en secondes depuis qu'il a été récupéré-->
<!ATTLIST service ttl CDATA "0">
<!ATTLIST service id ID #IMPLIED>
<!ELEMENT serviceuri (#PCDATA)>
<!ELEMENT servers (server+)>
<!ELEMENT server (serveruri, property*)>
<!ELEMENT serveruri (#PCDATA)>
<!ELEMENT dataset (property*)>
<!ATTLIST dataset id ID #IMPLIED>
<!ELEMENT datasetref EMPTY>
<!ATTLIST datasetref ref IDREF #IMPLIED>
<!ELEMENT propertyschema (propertydeclaration*)>
<!ELEMENT propertydeclaration (propertyname, propertytype*)>
<!ATTLIST propertydeclaration id ID #IMPLIED>
<!ELEMENT propertyname (#PCDATA)>
<!ELEMENT propertytype (#PCDATA)>
<!--Cela spécifie si le type est destiné à être celui par défaut. C'est généralement réservé pour "freeform".-->
<!ATTLIST propertytype default (no|yes) "no">
<!--Propriétés qu'on peut utiliser dans une interrogation-->
<!ELEMENT queryschema (propertyreference*)>
<!--Propriétés qu'on peut s'attendre à voir dans une ressource-->
<!ELEMENT resourcedescriptorschema (propertyreference*)>
<!--Propriétés qu'on peut s'attendre à trouver dans une définition de Service-->
<!ELEMENT serviceschema (propertyreference*)>
<!ELEMENT propertyreference EMPTY>
<!--Cela spécifie si une propriété est exigée au titre de l'interrogation-->
<!ATTLIST propertyreference ref IDREF #REQUIRED>
<!ATTLIST propertyreference required (no|yes) "no">

```

## 6. Exemples

### 6.1 Demande de description de service

C'est ce que le client envoie lorsque il demande le schéma d'un serveur.

```
<?xml version="1.0"?>
<!DOCTYPE cnrp PUBLIC "-//IETF//DTD CNRP 1.0//EN"
"http://ietf.org/dtd/cnrp-1.0.dtd">
<cnrp>
  <servicequery />
</cnrp>
```

C'est le résultat. Remarquer comment est utilisée l'étiquette Service pour permettre au service de se décrire dans ses propres termes.

```
<?xml version="1.0"?>
<!DOCTYPE cnrp PUBLIC "-//IETF//DTD CNRP 1.0//EN"
"http://ietf.org/dtd/cnrp-1.0.dtd">
<cnrp>
  <results>
    <service ttl="43200">
      <serviceuri>urn:foo:bar</serviceuri>
      <servers>
        <server>
          <serveruri>http://host1.acmecorp.com:4321/foo?</serveruri>
        </server>
        <server>
          <serveruri>smtp://host2.acmecorp.com:4321/foo?</serveruri>
        </server>
      </servers>
      <description>C'est le service CNRP Acme</description>
      <!-- Cette propriété signifie que Acme est spécialisé dans les services de noms commerciaux-->
      <property name="category" type="naics">544554</property>
      <property name="BannerAdServer" type="uri">
        http://adserver.acmecorp.com/
      </property>
      <propertyschema>
        <propertydeclaration id="i1">
          <propertyname>workgroupID</propertyname>
          <propertytype default="yes">freeform</propertytype>
          <propertytype default="no">domainname</propertytype>
        </propertydeclaration>
        <propertydeclaration id="i2">
          <propertyname>BannerAdServer</propertyname>
          <propertytype default="yes">URI</propertytype>
        </propertydeclaration>
      </propertyschema>
      <queryschema>
        <propertyreference ref="i1" required="yes" />
      </queryschema>
      <resourcedescriptorschema>
        <propertyreference ref="i1" required="yes" />
      </resourcedescriptorschema>
      <serviceschema>
        <propertyreference ref="i2" required="yes" />
      </serviceschema>
    </service>
  </results>
</cnrp>
```

## 6.2 Envoi d'une interrogation et obtention d'une réponse

C'est l'interrogation qui est envoyée du client au serveur :

```
<?xml version="1.0"?>
<!DOCTYPE cnrp PUBLIC "-//IETF//DTD CNRP 1.0//EN"
"http://ietf.org/dtd/cnrp-1.0.dtd">
<cnrp>
<query>
  <commonname>Fido</commonname>
  <property name="geography" type="iso3166-2">CA-QC</property>
  <property name="geography" type="iso3166-1">CA</property>
  <property name="language" type="rfc1766">fr-CA</property>
</query>
</cnrp>
```

C'est l'ensemble de résultats. Il est renvoyé en réponse à l'interrogation. Cet ensemble de résultats comporte un référant et une erreur non fatale.

```
<?xml version="1.0"?>
<!DOCTYPE cnrp PUBLIC "-//IETF//DTD CNRP 1.0//EN"
"http://ietf.org/dtd/cnrp-1.0.dtd">
<cnrp>
<results>
  <service id="i0">
    <serviceuri>http://acmecorp.com</serviceuri>
  </service>
  <service id="i1">
    <serviceuri>http://serverfarm.acmecorp.com</serviceuri>
  </service>
  <service id="i2">
    <serviceuri>http://servers.acmecorp.co.uk</serviceuri>
  </service>
  <resourcedescriptor>
    <commonname>Fidonet</commonname>
    <id>1333459455</id>
    <resourceuri>http://www.fidonet.ca</resourceuri>
    <serviceref ref="i0" />
    <description>This is ye olde Canadian Fidonet</description>
  </resourcedescriptor>
  <resourcedescriptor>
    <commonname>Fidonet</commonname>
    <id>1333459455</id>
    <resourceuri>http://host:port/bla</resourceuri>
    <serviceref ref="i1" />
    <description>An old Fidonet node</description>
  </resourcedescriptor>
  <referral><serviceref ref="i2" /></referral>
  <status code="3.1.1">
    La propriété langage 'fr-CA' a été ignorée.
  </status>
</results>
</cnrp>
```

## 7. Transport

Deux protocoles de transport CNRP sont spécifiés. HTTP est utilisé à cause de sa popularité et de sa facilité d'intégration aux autres applications de la Toile. SMTP est aussi utilisé comme moyen d'illustrer un protocole qui a une gamme de latence très différente de celle de la plupart des protocoles.

Dans les cas où les transports utilisent les types de support MIME (HTTP et SMTP en étant des exemples) la charge utile

CNRP DOIT utiliser le type de support 'application/cnrp+xml'. Voir à la Section 8 le gabarit d'enregistrement pour ce type de support. Une note importante au sujet de ce type de support est que, comme CNRP utilise toujours UTF-8, il n'y a pas d'attribut de jeu de caractères.

## 7.1 Transport HTTP

Le transport HTTP est très simple. Le client se connecte à un serveur CNRP fondé sur HTTP et produit une demande en utilisant la méthode POST au chemin "/" avec les en-têtes Content-type et Accept réglés à "application/cnrp+xml". Le contenu du corps POST est le document XML CNRP qui est à envoyer. Toutes les caractéristiques HTTP 1.1 sont permises durant la demande.

Les résultats sont renvoyés au client avec un type de contenu de "application/cnrp+xml". Le corps du résultat est le document XML CNRP envoyé au client.

## 7.2 Transport SMTP

Le transport SMTP est très similaire au transport HTTP. Comme il n'y a pas de méthode à spécifier, le document XML CNRP est simplement envoyé à un point d'extrémité SMTP particulier avec son type de contenu réglé à "application/cnrp+xml". Le serveur répond par l'envoi d'une réponse à l'origine de la demande avec les résultats dans le corps et le Content-Type réglé à "application/cnrp+xml". Le service DOIT spécifier au moins une cible SMTP (adresse de messagerie électronique) au contact.

## 8. Enregistrement : application/cnrp+xml

Voici le gabarit d'enregistrement pour 'application/cnrp+xml' selon la [RFC3023].

Nom du type de support MIME : application

Nom de sous type MIME : cnrp+xml

Paramètres exigés : aucun

Paramètres facultatifs : aucun

Considérations de codage : Ce type de support consiste en un texte de 8 bits qui peut nécessiter l'utilisation d'un codage de transfert de contenu approprié sur certains transports. Comme ces considérations sont les mêmes que pour XML en général, l'exposé de la [RFC3023] sur XML et MIME est applicable.

Considérations de sécurité : aucune n'est spécifique pour ce type de support. Voir à la Section 9 les considérations générales pour CNRP.

Considérations d'interopérabilité : non applicable.

Spécification publiée : Ce type de support est un sous ensemble approprié de la spécification XML 1.0 [XML] sauf pour les limitations fixées sur les étiquettes et les codages par le présent document.

Applications qui utilisent ce type de support : tout client/serveur CNRP qui souhaite envoyer ou recevoir des demandes ou réponses CNRP.

Informations supplémentaires : aucune

Contact pour complément d'informations: cf. la section "Adresse de l'auteur" du présent mémoire.

Usage prévu : utilisation limitée.

Auteur/Contrôleur des modifications : IESG.

## 9. Considérations sur la sécurité

Trois menaces existent pour la sécurité de CNRP ou des applications qui en dépendent : les attaques par interposition, les agents malveillants qui se font passer pour un service en falsifiant un objet Service, et les attaques de déni de service causées par l'ajout d'un nouveau niveau de redirection pour la résolution d'une ressource. La solution proposée pour les attaques par interposition est d'utiliser l'authentification et le chiffrement au niveau du transport, lorsque disponible. Dans les cas où le transport ne peut pas fournir le niveau d'authentification requis, des entrées individuelles ou la réponse entière peuvent être signées/chiffrées en utilisant les méthodes de signature XML qui sont développées par le groupe de travail XMLDSIG.

Dans le cas où un service tente de se faire passer pour un autre en falsifiant le serviceuri dans l'objet Service, l'objet Service devrait être signé. Un client peut alors vérifier la véracité de l'objet Service en vérifiant la signature. Comment le

client obtient cette clé publique d'autorité sort du domaine d'application de la présente spécification car cela dépend du problème de découverte du service.

Bien que le présent document ne puisse pas proposer de solution pour les attaques de déni de service (DOS), il peut illustrer que comme dans bien d'autres cas, chaque fois qu'un nouveau niveau de redirection est créé, il est créé une opportunité d'attaque de DOS. Les fournisseurs de services sont invités à veiller à cela et à agir en conséquence pour atténuer les effets d'une attaque de DOS.

## 10. Considérations relatives à l'IANA

La considération majeure pour l'IANA est qu'elle va enregistrer les propriétés bien connues, les types de propriétés et les messages d'état. Elle ne va pas enregistrer les valeurs. Comme le présent document ne parle pas de la découverte des services CNRP, l'IANA n'aura pas à enregistrer l'existence des serveurs ou des objets Server.

Il y a trois types d'entités que l'IANA peut enregistrer : les propriétés, les types de propriétés, et les messages d'état. Si une propriété ou un type n'est pas enregistré auprès de l'IANA, il doit alors commencer par "x-". Les messages d'état peuvent être créés pour consommation locale et non enregistrés. Il n'est pas exigé que les nouveaux messages d'état soit de mise en œuvre obligatoire sauf si le présent document est mis à jour. Les enregistrements de message d'état sont plutôt de nature informative.

Les informations requises pour l'enregistrement d'une nouvelle propriété sont le nom de la propriété, son type par défaut, et une description générale. Un nouveau type requiert le nom du type, les propriétés pour lesquelles il est valide, et une description. Un nouveau message d'état exige le code X.Y.ZZZ et une brève description de l'état communiqué.

Toutes les propriétés, les types et les messages d'état sont enregistrés sur la base du premier arrivé, premier servi sans révision par l'IANA ni d'aucun groupe d'experts. Le consensus du groupe de travail CNRP est que cette révision des enregistrements de propriétés devrait se faire une fois qu'aura été acquise l'expérience du fonctionnement du protocole et qu'existera un besoin réel de révision. Si, à l'avenir, cette politique devait changer, le présent document serait mis à jour.

Les gabarits d'enregistrement de propriété et de type qui se trouvent à l'Appendice A devraient être enregistrés par l'IANA au moment de la publication du présent document.

L'IANA est aussi invitée à enregistrer le type de support spécifié à la Section 8.

## Références

- [NAICS] United States, "North American Industry Classification System", janvier 1997, <<http://www.census.gov/epcd/www/naics.html>>.
- [RFC1766] H. Alvestrand, "Étiquettes pour l'identification des langues", mars 1995. (*Obsolète, voir RFC3066, RFC3282*) (P.S.)
- [RFC1893] G. Vaudreuil, "Codes d'état du système de messagerie améliorée", janvier 1996. (*Obsolète, voir RFC3463*) (P.S.)
- [RFC2119] S. Bradner, "[Mots clés à utiliser](#) dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997.
- [RFC2141] R. Moats, "[Syntaxe des URN](#)", mai 1997.
- [RFC2396] T. Berners-Lee, R. Fielding et L. Masinter, "Identifiants de ressource uniformes (URI) : Syntaxe générique", août 1998. (*Obsolète, voir RFC3986*)
- [RFC2616] R. Fielding et autres, "[Protocole de transfert hypertexte](#) -- HTTP/1.1", juin 1999. (*D.S., MàJ par 2817, 6585*)
- [RFC3023] M. Murata, S. St.Laurent et D. Kohn, "Types de support XML", janvier 2001. (*Obsolète, voir RFC7303*)
- [RFC3368] M. Mealling, "[Schéma d'URI 'go'](#) pour le protocole commun de résolution de nom", août 2002. (P.S.)
- [XML] Bray, T., Paoli, J. and C. Sperberg-McQueen, "Extensible Markup Language (XML) 1.0", février 1998.

[ISO3166] "Codes de pays et de régions", norme ISO 3166, janvier 1996.

## Appendice A. Propriété bien connue et gabarit d'enregistrement de type

### A.1 Propriétés

Nom de la propriété : geography  
Type par défaut : iso3166-1  
Description : localisation géographique

Nom de la propriété : language  
Type par défaut : rfc1766  
Description : spécification d'un langage

Nom de la propriété : category  
Type par défaut : freeform  
Description : nœud dans un système de relations sémantiques qui est considéré comme pertinent pour le nom courant.

Nom de la propriété : range  
Type par défaut : range  
Description : gamme donnée dans le format "x,y" où x est le point de départ et y est la longueur. Cette propriété est utilisée par le client pour dire au serveur qu'il demande une sous gamme des résultats.

Nom de la propriété : dataseturi  
Type par défaut : uri  
Description : URI utilisé pour distinguer entre deux ensembles de données offerts par le même service.

### A.2 Types

Type : freeform  
Propriété : category  
Description : la valeur est à interpréter par le serveur au mieux qu'il peut. Cette valeur n'a pas de structure définie.

Type : freeform  
Propriété : geography  
Description : la valeur est à interpréter par le serveur au mieux qu'il peut. Cette valeur n'a pas de structure définie.

Type : freeform  
Propriété : language  
Description : la valeur est à interpréter par le serveur au mieux qu'il peut. Cette valeur n'a pas de structure définie.

Type : iso3166-2  
Propriété : geography  
Description : combinaison des codes de pays et de sous-région de la norme ISO 3166-2 [ISO3166].

Type : iso3166-1  
Propriété : Geography  
Description : codes de pays de la norme ISO 3166-1 [ISO3166].

Type : postcode  
Propriété : Geography  
Description : code postal qui est valide pour une région. Un bon exemple est le système de code Zip des USA.

Type : lat-long  
Propriété : Geography  
Description : les valeurs pour la latitude et la longitude devront être exprimées en fractions décimales de degrés. Les degrés entiers de latitude devront être représentés par un nombre décimal de deux chiffres allant de 0 à 90. Les degrés entiers de longitude devront être représentés par un nombre décimal allant de 0 à 180. Lorsque une fraction décimale de degré est spécifiée, elle devra être séparée du nombre entier de degrés par un point. Les fractions décimales d'un degré peuvent être

exprimées à la précision désirée.

Les latitudes au nord de l'équateur devront être spécifiées par un signe plus (+), ou par l'absence d'un signe moins (-), précédant les degrés indiqués. Les latitudes au sud de l'équateur devront être désignées par un signe moins (-) précédant les deux chiffres désignant les degrés. Un point sur l'équateur devra être affecté à l'hémisphère Nord.

Les longitudes à l'est du premier méridien devront être spécifiées par un signe plus (+), ou par l'absence d'un signe moins. Les longitudes à l'ouest du premier méridien devront être désignées par un signe moins (-) précédant les chiffres qui désignent les degrés. Un point sur le premier méridien devra être affecté à l'hémisphère Est. Un point sur le 180<sup>ème</sup> méridien devra être affecté à l'hémisphère Ouest. Une exception à cette dernière convention est permise. Pour la condition particulière de description d'une bande de latitude autour de la terre, l'élément de donnée de coordonnée de limite ouest devra recevoir la valeur de +180 (180) degrés.

Toute adresse spatiale avec une latitude de +90 (90) ou -90 degrés spécifiera la position respectivement du pôle Nord ou Sud. Le composant de longitude peut avoir toute valeur légale.

À l'exception de la condition particulière décrite ci-dessus, cette forme est spécifiée dans le document Ministère du Commerce des USA, 1986, "Representation of geographic point locations for information interchange" (Federal Information Processing Standard 70-1) : Washington, Department of Commerce, National Institute of Standards and Technology.

```
DEGREES = *PLUSMINUS DIGITS '!' DIGITS
PLUSMINUS = + | -
DIGITS = DIGIT *DIGIT
DIGIT = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

Type : rfc1766

Propriété : Language

Description : codes de langues définis par la [RFC1766]

Type : naics

Propriété : Category

Description : Système de code de l'industrie nord américaine (*North American Industry Code System*) [NAICS]

Type : uri

Propriété : dataseturi

Description : URI qui adhère à la production 'absoluteURI' de l'ABNF de la [RFC2396]

## Appendice B Codes d'état

### B.1 Codes de niveau 1 (Information)

#### 1.0.0 – Information indéfinie

Ce code est utilisé pour tout message d'information n'entrant dans aucune catégorie. Si, par exemple, le serveur voulait dire au client que le chat de l'administrateur du système a une fourrure bleu, ce code serait approprié pour cette information.

#### 1.1.0 – Informations en rapport avec l'interrogation

Ce code est utilisé pour toute information concernant l'interrogation qu'envoie ce client. Par exemple, "L'interrogation envoyée est assez intéressante !".

1.2.0 -- Message d'information relatif au service. Ce message concerne le service au sens général.

### B.2 Codes de niveau 2 (Succès)

#### 2.0.0 – Quelque chose d'indéfini a réussi

Il y a eu réussite, mais la situation que concerne ce message n'est pas définie.

#### 2.1.0 – Interrogation réussie

L'interrogation a réussi. Ce message DOIT être retourné lorsque il n'y a pas de résultat qui satisfasse l'interrogation. C'est-

à-dire que l'interrogation a bien été traitée et l'ensemble de résultats correct ne contenait pas de ressource ou référent. Le manque de résultat n'est pas une erreur mais une déclaration positive sur le nom courant.

Note : Le manque apparent de code de niveau 2.X.X est causé par le fait que le succès est généralement indiqué non par un message d'état mais par le fait que le serveur retourne seulement les objets que le client a demandé.

### **B.3 Codes de niveau 3 (Succès partiel)**

3.0.0 – Quelque chose d'indefini a été partiellement réussi

Une demande du client a été seulement partiellement réussie. La situation ou cause exacte de cet échec partiel n'est pas définie.

3.1.0 – L'interrogation a été seulement partiellement réussie.

3.1.1 – L'interrogation contenait des propriétés invalides ou non acceptées

L'interrogation contenait des noms, types ou valeurs de propriétés invalides ou non acceptés. La propriété invalide a été ignorée et l'interrogation a été traitée.

3.1.2 -- Le XML était bien formé mais invalide

Le XML envoyé par le client était bien formé mais invalide. Le serveur a été assez habile pour déduire ce dont le client voulait parler et a retourné des résultats.

3.1.3 – Le serveur ne prend pas en charge les ensembles de données

Cet état devrait être généré par les serveurs qui ne traitent pas les ensembles de données. Un serveur peut envoyer ce message d'état à tout moment, mais il est particulièrement utile lorsque un serveur reçoit d'un client une interrogation qui contient un dataseturi. Dans ce cas et si le client fait une détection de boucle rigoureuse, il devrait considérer que ce service entier a été visité.

3.1.4 -- Le premier ensemble de données de la liste donnée dans l'interrogation a été le seul utilisé.

Ce message d'état est utilisé par un serveur pour indiquer une situation où un client a inclus plusieurs dataseturis dans son interrogation et où le serveur ne peut en prendre en charge qu'un seul à la fois. Dans ce cas, le serveur déclare explicitement qu'il a seulement utilisé le premier dataseturi. Le client devrait considérer que seul le premier dataseturi spécifié a été traité correctement. Le client devrait considérer que les ensembles de données restants dans l'interrogation ont été complètement ignorés.

Cela devrait être envoyé individuellement comme référents si le client tient réellement à ces résultats. Seul la première paire serviceuri/dataseturi devrait être marquée comme visitée si la détection de boucle est effectuée.

3.1.5 – Cet ensemble de données n'est pas pris en charge.

Ce message est utilisé pour indiquer qu'un dataseturi spécifique envoyé dans une interrogation par un client n'est pas accepté par le serveur. Cette paire serviceuri/dataseturi devrait être considérée comme visitée par le client. Si ce message est envoyé en réponse à une interrogation qui spécifie plusieurs ensembles de données, le client devrait se comporter de la même façon que si il avait reçu le message 3.1.3. Renvoyer ce message d'état en réponse à une interrogation avec plusieurs ensembles de données devrait être considéré comme une mauvaise forme pour un serveur parce que il est ambigu.

3.2.0 -- Le serveur a causé un événement partiellement réussi.

Du fait de quelque erreur interne du serveur, les résultats retournés sont incomplets.

3.2.1 – Un serveur de référence était indisponible.

Ce message d'état est utilisé pour noter qu'un ou plusieurs des services référencés qui sont normalement interrogés étaient indisponibles. Les résultats ont été générés, mais ils pourraient n'être pas représentatifs d'une réponse complète.

### **B.4 Codes de niveau 4 (défaillance transitoire)**

4.0.0 – Quelque chose d'indefini a causé une défaillance transitoire persistante.

4.1.0 – Il y avait une erreur dans l'interrogation qui la rend ininterprétable.

4.2.0 – L'interrogation est trop complexe.

L'interrogation telle que spécifiée était trop complexe pour que ce service la traite.

4.2.1 -- Le service est trop occupé.

Du fait des contraintes de ressources, le service entier est trop occupé pour traiter les demandes. Cela signifie que tous les serveurs qui coopèrent pour fournir ce service auraient aussi retourné ce même message.

4.2.2 -- Le serveur est en cours d'entretien.

Ce serveur est maintenant en mode de maintenance. Essayer un autre serveur à partir de ce service ou réessayer plus tard.

4.2.3 -- Le serveur a subi une erreur interne. Il y a eu une erreur interne qui a causé l'échec complet du serveur.

## **B.5 Codes de niveau 5 (défaillance permanente)**

5.0.0 – Quelque chose d'indéfini a causé une défaillance permanente.

5.1.0 – L'interrogation a subi une défaillance permanente.

5.2.0 -- Le service a subi une défaillance permanente.

5.2.1 -- Ce service n'est plus disponible. Ce service a décidé de ne plus se rendre disponible.

5.2.2 -- Le serveur a subi une défaillance permanente.

Ce serveur est en échec permanent. Essayer un autre serveur à partir de ce service.

## **Adresse des auteurs**

Nico Popp  
VeriSign, Inc.  
487 East Middlefield Road  
Mountain View, CA 94043  
téléphone : (650) 426-3291  
mél : [npopp@verisign.com](mailto:npopp@verisign.com)

Michael Mealling  
VeriSign, Inc.  
21345 Ridgetop Circle  
Sterling, VA 20166  
USA  
mél : [michael@verisignlabs.com](mailto:michael@verisignlabs.com)

Marshall Moseley  
Netword, Inc.  
702 Russell Avenue  
Gaithersburg, MD 20877-2606  
USA  
téléphone : (240) 631-1100  
mél : [marshall@netword.com](mailto:marshall@netword.com)

## **Déclaration complète de droits de reproduction**

Copyright (C) The Internet Society (2002). 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 droits de reproduction ci-dessus et le présent 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 droits de reproduction 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 droits de reproduction 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 ses successeurs ou ayant droits.

Le présent document et les informations y contenues 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 l'Internet Society.