

Groupe de travail Réseau
Request for Comments : 3981
 Catégorie : En cours de normalisation
 Traduction Claude Brière de L'Isle

A. Newton, VeriSign, Inc.
 M. Sanz, DENIC eG

janvier 2005

IRIS : Protocole central du service d'information des registres Internet

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 (2005).

Résumé

Le présent document décrit un protocole client-serveur de couche application comme un cadre de représentation des interrogations et des opérations de résultat des services d'information des registres de l'Internet. Spécifié dans le langage de balisage extensible (XML, *Extensible Markup Language*) le protocole définit les interrogations génériques et les opérations de résultat ainsi qu'un mécanisme pour étendre ces opérations pour des besoins spécifiques de service de registre.

Table des Matières

1. Introduction.....	2
1.1 Utilisation de XML.....	2
1.2 Concepts généraux.....	2
1.3 Couches du cadre.....	2
1.4 Définitions.....	3
1.5 Lectures conseillées.....	3
2. Terminologie du document.....	3
3. Identification du protocole.....	3
4. Description de l'échange.....	4
4.1 Format de demande.....	4
4.2 Format de réponse.....	4
4.3 Cadre d'extension.....	5
4.4 Sacs de relais.....	11
5. Mise en série de la base de données.....	12
6. Syntaxe XML formelle.....	13
7. URI IRIS.....	24
7.1 Définition d'URI.....	24
7.2 Schémas spécifiques du transport.....	25
7.3 Résolution d'URI.....	25
7.4 Exemples d'URI IRIS.....	26
8. Listes de contrôle.....	26
8.1 Listes de contrôle de définition de registre.....	26
8.2 Listes de contrôle de transposition de transport.....	27
9. Considérations d'internationalisation.....	27
10. Considérations relatives à l'IANA.....	27
11. Considérations sur la sécurité.....	27
12. Références.....	27
12.1 Références normatives.....	27
12.2 Références pour information.....	28
Appendice A. Utilisations de S-NAPTR et d'IRIS.....	28
A.1. Exemple de S-NAPTR avec IRIS.....	28
A.2 Utilisation de S-NAPTR pour la cohabitation.....	30
Appendice B. Philosophie de la conception d'IRIS.....	30
B.1 Principe de base.....	30
B.2 Attrait d'un client universel.....	30
B.3 Considérations de serveur.....	31

B.4 Examens, recherches, et classes d'entités.....	31
B.5 Références d'entités, continuations de recherches, et portée.....	31
Appendice C. Remerciements.....	32
Adresse des auteurs.....	32
Déclaration complète de droits de reproduction.....	32

1. Introduction

La spécification proposée dans le présent document se fonde sur les exigences fonctionnelles décrites dans CRISP [RFC3707].

1.1 Utilisation de XML

Le présent document décrit les spécifications du protocole de service d'informations des registres Internet (IRIS, *Internet Registry Information Service*) en texte XML destiné à décrire les types d'interrogations et les types de résultats de divers services d'informations de registres. IRIS est spécifié en utilisant le langage de balisage extensible (XML, *Extensible Markup Language*) 1.0 comme décrit dans [XML], la notation de schéma XML est décrite dans [XML-S2] et [XML-S1], et les espaces de noms XML sont décrits dans [XML-NS].

1.2 Concepts généraux

Chaque sorte de registre Internet est identifiée par un type de registre. L'identifiant pour un type de registre est un nom de ressource universel (URN, *Uniform Resource Name*) utilisé dans les instances XML pour identifier le schéma XML qui décrit formellement l'ensemble de classes d'interrogations, de résultats, et d'entités permis dans ce type de registre.

La structure de ces URN ne fait aucune hypothèse ni restriction sur les types de registres qu'ils identifient. Donc, IRIS peut prendre en charge plusieurs types de registres de nature différente ou similaire ; c'est seulement une affaire de définition. Par exemple, un seul type de registre peut être défini pour les registres de noms de domaines, et plusieurs types de registres pour les divers registres d'adresses IP.

Un serveur d'informations de registre peut traiter des interrogations et servir des résultats pour plusieurs types de registres. Chaque type de registre que dessert un opérateur de registre particulier est une instance de service de registre.

IRIS et le schéma XML qui décrit formellement IRIS ne spécifient aucun registre, identifiant de registre, ou connaissance d'une instance ou ensemble d'instances de services particuliers. IRIS est une spécification d'un cadre avec lequel ces registres peuvent être définis, utilisés et, dans certains cas, interopérer. Le cadre spécifie simplement les éléments d'identification des registres et les éléments qui doivent être utilisés pour déduire les interrogations et les résultats.

Ce cadre permet à un type de registre de définir sa propre structure, les dénominations, les entités, les interrogations, etc., par l'utilisation de l'espace de noms XML et les schémas XML (donc, un type de registre DOIT être identifié par le même URI qui identifie son espace de noms XML). Pour être conforme, la spécification d'un type de registre doit s'étendre à partir de ce cadre.

Le cadre définit certaines structures qui peuvent être communes à tous les types de registres, comme les références aux entités, la continuation de la recherche, et les classes d'entités. Un type de registre peut déclarer ses propres définitions pour tout cela, ou il peut mélanger ses définitions dérivées avec les définitions de base.

IRIS définit deux types de références : une référence d'entité et une continuation de recherche. Une référence d'entité indique une connaissance spécifique sur une entité individuelle, et une continuation de recherche permet des recherches réparties. Les deux références peuvent s'étendre sur différents types et instances de registres. Aucune hypothèse ni spécification n'est faite sur les racines, les bases, ou les maillages des entités.

1.3 Couches du cadre

Le cadre IRIS peut être vu comme ayant trois couches :

Spécifique du registre	domaine adresse etc...
Commun aux registres	IRIS
Application-Transport	beep iris-lwz etc...

Dans cette figure, "beep" se réfère au protocole extensible d'échange de blocs (BEEP, *Blocks Extensible Exchange Protocol*) (voir la [RFC3080]), et "iris-lwz" se réfère à un lien UDP théorique qui utilise la compression.

Les différentes couches ont les responsabilités suivantes :

Spécifique de registre : définit les classes d'interrogations, résultats, et entités d'un type spécifique de registre. Chaque type spécifique de registre est identifié par un URN.

Commun aux registres : définit les opérations de base et la sémantique communes à tous les types de registres comme des ensembles de recherches, des ensembles de résultats, et des références. Il définit aussi la syntaxe pour parler des types de registre spécifiques.

Application-Transport : définit les mécanismes d'authentification, de passage de message, de gestion de connexion et de session, etc. Il définit aussi la syntaxe d'URI spécifique du mécanisme d'application-transport.

1.4 Définitions

Pour être clair, on fournit les définitions suivantes :

- o type de registre -- Registre servant une fonction spécifique, comme un registre de domaines, ou un registre d'adresses. Chaque type de registre reçoit un URN.
- o schéma de registre -- définition d'un type de registre qui spécifie les classes d'interrogations, de résultats, et d'entités.
- o autorité -- référence au serveur ou ensemble de serveurs qui contiennent les informations.
- o méthode de résolution -- technique utilisée pour localiser une autorité.
- o classe d'entités -- groupe d'entités avec un type commun ou un ensemble commun de caractéristiques.
- o nom d'entité -- identifiant utilisé pour se référer à une seule entité au sein d'une classe d'entités.
- o référence d'entité -- pointeur sur une entité, composé d'une autorité, d'une méthode de résolution facultative, d'un type de registre, d'une classe d'entités, et d'un nom d'entité. Un type de référence d'entité est l'URI IRIS (défini Section 7).

Les termes "dérivatif", "dérive", et "dérivation" sont utilisés avec la même signification que pour dériver un type d'élément d'un autre, comme spécifié dans [XML-S1].

1.5 Lectures conseillées

L'Appendice B contient un texte qui répond à la question, "Pourquoi IRIS ?".

Le présent document décrit la structure du cœur de IRIS. Les documents suivants décrivent les autres aspects de IRIS relevant de CRISP [RFC3707] : iris-beep [RFC3983] et iris-dreg [RFC3982].

2. Terminologie du document

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119].

3. Identification du protocole

L'élément racine de toutes les instances XML de demandes DOIT être <request>. L'élément racine de toutes les instances de réponse XML DOIT être <response>. Ces éléments identifient le début des éléments IRIS, l'espace de noms XML utilisé comme identifiant pour IRIS, et, facultativement, la localisation du schéma. Ces éléments et l'étiquette de clôture associée DOIVENT être appliqués à toutes les demande et réponses envoyée par les clients et les serveurs.

L'utilisation de l'attribut de localisation de schéma 'xsi:schemaLocation' est FACULTATIVE par rapport à la présente spécification, et les mises en œuvre de IRIS PEUVENT le résoudre pour restituer le schéma ou PEUVENT utiliser une

version en antémémoire locale du schéma.

La version du protocole IRIS est de la responsabilité de la couche application-transport mais DOIT être associée à l'URI d'espace de noms XML [XML-NS] qui représente IRIS. Un changement de cet URI indique un changement du schéma sous-jacent, et donc une nouvelle version du protocole (et vice versa).

4. Description de l'échange

La présente section décrit les échanges de demandes et de réponses du protocole. Les descriptions contenues dans la présente section se réfèrent aux éléments et attributs XML et à leur relation à l'échange de données au sein du protocole. Ces descriptions contiennent aussi des spécifications qui sortent du domaine d'application de la syntaxe XML formelle. Donc, cette section utilisera les termes définis dans la [RFC2119] pour décrire les spécifications qui sortent du domaine d'application de la syntaxe XML formelle. En lisant la présente section, on se référera à la Section 6 pour les détails de la syntaxe XML formelle.

4.1 Format de demande

Un élément <request> contient un élément facultatif <control> et un ensemble d'éléments <searchSet>.

Les éléments <searchSet> permettent à un client d'interroger un type de registre particulier en utilisant l'URN qui identifie le type de registre. Ils peuvent être trouvés dans un de ses deux enfants : <lookupEntity> et <query>.

L'élément <lookupEntity> décrit l'examen d'une entité dans un registre spécifique. Cet élément a trois attributs : "registryType", "entityClass", et "entityName". L'attribut "registryType" contient l'identifiant de registre pour le type de registre dans lequel va avoir lieu l'opération. L'attribut "entityClass" contient le jeton qui identifie l'indice pour lequel l'opération d'examen va avoir lieu, et l'attribut "entityName" contient le nom de l'entité à examiner.

L'élément <query> est abstrait et ne peut pas légalement apparaître dans une instance XML. Il fournit le type de base que vont utiliser les schémas de registres pour définir les types d'interrogation dérivés. Ce mécanisme de dérivation est décrit au paragraphe 4.3.

Chaque <searchSet> peut aussi contenir un élément <bag>. Lorsque cet élément apparaît comme fils de <searchSet>, il NE DOIT PAS contenir l'attribut "id". Pour une description de l'élément <bag>, voir au paragraphe 4.4.

L'élément <control> peut contenir un élément fils de tout espace de noms XML. Cet élément fils permet à un client de signaler à un serveur des états ou traitements particuliers. Un exemple d'un tel élément fils <control> se trouve au paragraphe 4.3.8.

4.2 Format de réponse

L'élément <response> contient un élément <reaction> facultatif, un ensemble d'éléments <resultSet>, et un élément <bags> facultatif.

Les éléments <resultSet> sont des réponses à une demande <searchSet>. Le contenu de cet élément contient un élément <answer>, un élément <additional> facultatif, et des éléments d'erreur, si c'est applicable.

Les enfants de l'élément <answer> sont des types suivants :

- o <result> est un élément abstrait et ne peut pas être légalement placé dans une instance XML. Il fournit le type de base à utiliser par les schémas de registre pour définir les types de résultat dérivés. Ce mécanisme de dérivation est décrit au paragraphe 4.3.
- o <entity> est un élément qui spécifie une référence d'entité. Voir au paragraphe 4.3.5.
- o L'élément <searchContinuation> spécifie une référence d'interrogation. Son seul enfant est tout élément dérivé de <query> (voir au paragraphe 4.3.1). Pour diriger l'interrogation sur un serveur référent, <searchContinuation> a un attribut "authority" obligatoire et un attribut "resolution" facultatif. L'élément <searchContinuation> peut aussi contenir un attribut "bagRef". Pour une description de l'attribut "bagRef", voir au paragraphe 4.4.

Lorsque ils suivent les références d'entité et les continuations de recherche, les clients DEVRAIENT suivre seulement une

fois une réponse `<entity>` ou `<searchContinuation>`. Manquer à faire ainsi peut avoir pour résultat que le processus client soit englué dans une boucle sans fin d'interrogations, couramment appelée une référence en boucle.

L'élément `<additional>` contient seulement des éléments `<result>`, comme décrit ci-dessus. Cet élément permet à un serveur d'indiquer à un client des résultats qui n'ont pas fait l'objet d'une interrogation spécifique mais qui sont en rapport avec les résultats recherchés, permettant donc au client d'afficher correctement cette distinction à un utilisateur. L'utilisation de l'élément `<additional>` est facultative.

Les éléments suivants, qui représentent des conditions d'erreur, peuvent être retournés :

- o `<insufficientResources>` -- l'interrogation correspondante exige des ressources que le serveur ne peut pas atteindre.
- o `<invalidName>` -- un nom donné dans une interrogation est syntaxiquement incorrect.
- o `<invalidSearch>` -- les paramètres de l'interrogation correspondante ne sont pas sémantiquement significatifs.
- o `<queryNotSupported>` -- l'interrogation correspondante n'est pas acceptée par ce serveur.
- o `<limitExceeded>` -- l'interrogation correspondante exige plus de ressources que permis.
- o `<nameNotFound>` -- le nom donné dans une interrogation ne correspond pas à une entité connue.
- o `<permissionDenied>` -- l'authentification donnée ne permet pas l'accès à une entrée de résultat spécifique.
- o `<bagUnrecognized>` -- le contenu d'un sac n'a pas été reconnu. Voir au paragraphe 4.4.
- o `<bagUnacceptable>` -- le contenu d'un sac n'était pas et ne sera jamais acceptable. Voir au paragraphe 4.4.
- o `<bagRefused>` -- le contenu d'un sac n'était pas acceptable pour l'instant. Voir au paragraphe 4.4.
- o Un dérivé de `<genericCode>`, comme décrit au paragraphe 4.3.

La section `<resultSet>` est divisée en les sections `<answer>` et `<additional>` pour permettre un traitement plus facile et le parcours des résultats par un client. Les serveurs DOIVENT retourner les réponses directes aux interrogations dans l'élément `<answer>` et PEUVENT retourner les résultats dans l'élément `<additional>` pour lequel une référence a été faite dans l'élément `<answer>`. Les résultats dans l'élément `<additional>` DOIVENT avoir été référencés dans la `<answer>`, soit comme enfant direct de l'élément `<answer>`, soit comme descendants plus éloignés de l'élément `<answer>`.

Ceci sert deux objets. D'abord, cela peut éliminer un renouvellement d'interrogation par le client pour des références contenues dans l'élément `<answer>`. Ensuite, cela fait une distinction entre les résultats qui sont un résultat direct d'une interrogation et ceux qui auraient été retournés si le client avait suivi les références appropriées, donnant donc un conseil sur la façon dont les clients pourraient traiter ou afficher les résultats retournés. Par exemple, les clients qui construisent des affichages complexes avec des appliquestes d'arborescence de navigation vont savoir que les résultats dans l'élément `<answer>` devraient tous être directement en dessous du nœud racine de l'arborescence, tandis que les résultats dans l'élément `<additional>` sont des successeurs de ceux produits à partir de l'élément `<answer>`.

Un élément `<reaction>` (fils de `<response>`) est une réponse à un élément `<control>`, et donne un moyen à un serveur de conseiller un client sur les effets d'un élément `<control>`.

L'élément `<bags>` (fils de `<response>`) est facultatif. Il contient des éléments `<bag>`, et le contenu de chaque élément `<bag>` constitue un élément dans tout espace de noms XML. Chaque élément `<bag>` a un attribut "id", qui est référencé par l'attribut "bagRef" des références d'entité (`<entity>`) et continuations de recherche (`<searchContinuation>`). Voir au paragraphe 4.4.

4.3 Cadre d'extension

Parce que le schéma IRIS définit seulement un type d'interrogation, et deux types de résultats autonomes, et ne définit pas de structure de registre, il est d'un usage limité par lui-même. L'extension de IRIS est accomplie par l'utilisation d'un schéma IRIS de base, comme défini dans XML_SD [XML-S2] et XML_SS [XML-S1], et par son extension au moyen des schémas construits par dessus IRIS.

4.3.1 Éléments dérivés

La définition de schéma XML de IRIS exige que les schémas de type de registre dérivent les types d'élément des types de base de la définition IRIS. Les schémas de registre DOIVENT dériver des éléments pour définir des interrogations et résultats typés.

Bien que la définition de schéma IRIS n'interdise la dérivation d'aucun élément, les schémas de registre DEVRAIENT restreindre les dérivations aux types suivants :

- o `<query>` -- comme défini, cet élément n'a pas de contenu et pas d'attribut valide. Il est abstrait et donc seuls ses dérivés apparaissent dans les instances XML. Les schémas de registre dérivent de cet élément pour définir les interrogations permises.

- o <result> -- comme défini, cet élément n'a pas de contenu et a cinq attributs valides : "authority", "resolution" (facultatif), "registryType", "entityClass", "entityName", et "temporaryReference" (facultatif, voir au paragraphe 4.3.6). Il est abstrait et donc seuls ses dérivés apparaissent dans les instances XML. Les schémas de registre dérivent de cet élément pour définir les résultats qui peuvent être retournés d'une interrogation.
- o <genericCode> -- comme défini, cet élément est une instance de <codeType>. Il contient des éléments facultatifs <explanation> et <language>, qui approfondissent la description de la nature de l'erreur.
- o <entity> -- identifie une référence à une entité. Les schémas de registre DEVRAIENT utiliser les éléments dérivés de <entity> mais PEUVENT utiliser <entity> directement. L'avantage de dériver de <entity> sur l'utilisation directe est la chance de définir le nom de l'élément et d'utiliser ce nom de façon descriptive -- par exemple, comme le rôle que l'entité joue par rapport à une autre entité. Voir au paragraphe 4.3.5.
- o <seeAlso> -- indique une référence à une entité qui a une association indirecte avec un élément parent représentant une entité. Cet élément est dérivé de l'élément <entity> (paragraphe 4.3.5). Les schémas de registre PEUVENT dériver de cet élément ou PEUVENT l'utiliser directement.

4.3.2 Exigences d'identifiant de type de registre

L'identifiant pour un type de registre et l'identifiant d'espace de noms XML utilisé par le schéma XML qui décrit le registre DOIVENT être le même. Ces identifiants DOIVENT être restreints à un URN [RFC2141] enregistré dans la classe "ns" du registre IANA gouverné par XML_URN [RFC3688]. Ces identifiants sont insensibles à la casse.

C'est une restriction de XML_NS [XML-NS], qui spécifie qu'un identifiant d'espace de noms XML est tout URI valide [RFC2396].

Ces identifiants PEUVENT être abrégés à la partie qui suit le composant de classe et son séparateur de l'URN. Par exemple, l'URN complet "urn:ietf:params:xml:ns:dreg1" peut être abrégé en "dreg1".

Dans l'utilisation avec IRIS, cette abréviation NE DOIT PAS être utilisée à l'intérieur d'instances XML dans lesquelles le schéma XML [XML-S2] spécifie l'utilisation d'un URI pour l'identification de schéma ou lorsque XML_NS [XML-NS] spécifie l'utilisation d'un URI pour l'identification d'un espace de noms XML.

4.3.3 Classes d'entité

IRIS fournit des classes d'entités pour aider à éviter les collisions avec les noms d'entité au sein d'un certain type de registre. Leur spécification dans les interrogations permet aussi aux mises en œuvre de serveur de rétrécir rapidement le champ d'examen ou de recherche à un seul index.

Par exemple, le nom d'entité "192.0.2.0" peut se référer à des entités distinctes dans les classes "name-server" et "network". L'entité "192.0.2.0" dans la classe "name-server" peut se référer à l'hôte serveur de noms qui est aussi multi-rattaché par l'adresse 192.0.2.255 et connu dans le DNS comme "ns.example.com", tandis que l'entité "192.0.2.0" dans la classe "network" peut se référer au réseau 192.0.2/30.

IRIS définit les deux classes d'entités par défaut de "local" et "iris", qui NE DOIVENT PAS être redéfinies. Ces classes d'entité DOIVENT être valides dans tous les types de registres.

La classe "local" est réservée aux entités définies en local par l'opérateur de serveur et elle ne note aucun type particulier d'entité. Un examen dans cette classe d'entités PEUT résulter en une référence d'entité ou en une continuation de recherche. Par exemple, "iris:dreg1//example.com/local/myhosts" peut résulter en une continuation de recherche donnant les serveurs de noms pour example.com.

La classe "iris" est réservée aux entités spécifiques d'une instance de service particulière. Elle DOIT contenir les noms d'entité suivants (voir au paragraphe 4.3.4) :

- o "id", qui donne un résultat de <serviceIdentification> (voir au paragraphe 4.3.7.1).
- o "limits", qui donne un résultat de <limits> (voir au paragraphe 4.3.7.2). Cette classe d'entités PEUT contenir aussi d'autres entités définies en local.

Les noms des classes d'entités dans un schéma de registre sont du type jeton, comme défini par XML_SD [XML-S2]. Leur sensibilité à la casse DOIT être définie par la définition du type de registre. En général, ils DEVRAIENT être insensibles à la casse.

4.3.4 Noms des entités

Les noms des entités dans un schéma de registre sont du type jeton, comme défini par XML_SD [XML-S2].

Les noms des entités DEVRAIENT être uniques au sein d'une instance de toute classe d'entité particulière au sein d'un registre. Deux entités NE DEVRAIENT PAS avoir le même nom, mais une seule entité PEUT être connue par plusieurs noms. Dans des situations où un seul nom peut résulter en deux entités, le schéma de registre DEVRAIT en tenir compte en définissant des types de résultats qui contiennent des références d'entité aux deux entités (par exemple, "example.com" peut se référer à la fois au domaine example.com et à l'hôte example.com). Cependant, ce type de conflit DEVRAIT généralement être évité par l'utilisation appropriée des classes d'entités.

La sensibilité à la casse des noms d'entités dépend de la classe d'entités dans laquelle elles résident. La définition d'un type de registre DOIT spécifier la sensibilité à la casse des noms d'entités. Un type de registre PEUT définir les noms d'entité de différentes classes d'entités comme ayant une sensibilité à la casse différente.

4.3.5 Références aux entités

L'élément <entity> permet des références aux entités dans les ensembles de résultats, soit comme un fils direct de <resultSet> soit au sein d'une structure plus complexe dérivée de <result>. L'élément <entity> est défini par "entityType". Les schémas de registre DEVRAIENT définir les éléments dérivés de <entity> lorsque ils font référence aux entités mais peuvent utiliser l'élément <entity> directement. Dériver un nouvel élément permet à un schéma de registre d'utiliser le nom du nouvel élément pour signifier la relation qu'a l'entité référencée avec le référant. Un dérivé de <entity> NE DOIT PAS être utilisé comme substitut lorsque l'élément <entity> est déclaré (comme dans la section <answer> de <resultSet>).

L'élément <entity> (et les éléments de type 'entityType') peuvent avoir des éléments fils de <displayName> avec un attribut "language" facultatif. Ceux-ci sont fournis afin que les serveurs puissent donner aux clients une description plus lisible par l'homme de la référence d'entité. C'est souvent utile pour les usagers qui naviguent dans des structures de références.

L'élément <entity> (et ses dérivés) ont les attributs suivants :

- o "authority", "resolution" (facultatif), "registryType", "entityClass", et "entityName" -- ces attributs spécifient où l'entité peut être trouvée.
- o "temporaryReference" -- cet attribut est facultatif. Voir au paragraphe 4.3.6.
- o "referentType" -- cet attribut contient le type attendu de l'entité référencée et peut contenir le mot "ANY" ou le nom XML qualifié. À la différence des autres attributs de <entity>, cet attribut est qualifié et déclaré dans l'espace de noms XML d'IRIS. Donc, il sera aussi qualifié avec le préfixe associé à l'espace de noms XML d'IRIS (par exemple, "iris:referentType"). Cela permet aux clients de reconnaître les références d'entité en utilisant un élément dérivé de <entity>.
- o "bagRef" -- cet attribut est facultatif. Si il est présent, il doit contenir un identifiant XML à un élément <bag> dans la section <bags> de l'ensemble de résultats. Pour une description de l'attribut 'bagRef', voir au paragraphe 4.4.

4.3.6 Entités temporaires

Il peut exister des instances dans lesquelles une référence d'entité a besoin d'être temporaire. Par exemple, un type particulier de résultat ne peut n'avoir qu'une clé unique. Si cette clé contient une signification sémantique qui ne doit pas être exposée à tous les usagers, une clé synthétique devra lui être substituée.

De plus, il y a des fois où des données de la mémorisation de données ne sont pas normalisées de la même manière que celle exprimée dans le schéma de registre. Dans le schéma de registre, les objets de type A peuvent référencer des objets de type B. Mais dans le magasin de données, les objets de type A peuvent contenir des objets de type B. Là encore, une clé synthétique devra être produite temporairement.

Pour prendre en charge de tels cas d'utilisation, les résultats et références d'entité peuvent être déclarés temporairement en utilisant l'attribut "temporaryReference". Cet attribut est de type booléen [XML-S2] et a une valeur par défaut de "faux". Il est facultatif pour les dérivés et éléments <result> de type "entityType".

Lorsque cet attribut est utilisé, les données de référence d'entité (par exemple, "entityClass", "entityName") ne sont valides qu'au sein de la réponse dans laquelle elles apparaissent et peuvent n'être pas cohérentes avec les réponses suivantes. Un

serveur DOIT inclure toute référence d'entité temporaire dans la section <additional> du même <resultSet>

4.3.7 Éléments dérivés de <result>

Le cadre IRIS de base contient trois éléments directement dérivés de l'élément <result> à utiliser par tout type de registre.

4.3.7.1 <serviceIdentification>

Exemple d'un résultat <serviceIdentification> :

```
<serviceIdentification
  authority="example.com" registryType="dreg1"
  entityClass="iris"
  entityName="id" >
  <authorities>
    <authority> example.com </authority>
    <authority> example.net </authority>
    <authority> example.org </authority>
  </authorities>
  <operatorName>
    Internet Assigned Numbers Authority
  </operatorName>
  <eMail>
    iana@iana.org
  </eMail>
</serviceIdentification>
```

L'élément <serviceIdentification> est fourni pour permettre aux clients IRIS de faire référence aux instances de service IRIS. Il contient les éléments suivants :

- o <authorities> -- cet élément contient un ou plusieurs éléments <authority>. Chaque élément <authority> contient un composant d'autorité d'URI pour lequel le serveur a des résultats. Bien qu'un serveur PUISSE ne retourner qu'une liste partielle de ses zones d'autorité, selon la politique de l'opérateur, il DOIT retourner l'autorité pour laquelle le client a fait sa demande.
- o <operatorName> -- cet élément contient le nom de l'opérateur du serveur.
- o <eMail> -- ces éléments facultatifs contiennent les adresses de messagerie électronique de l'opérateur de l'instance de service.
- o <phone> -- ces éléments facultatifs contiennent les numéros de téléphone de l'opérateur de l'instance de service.
- o <seeAlso> -- voir sa définition au paragraphe 4.3.1.

4.3.7.2 <limits>

Exemple d'un résultat <limits> :

```
<limits
  authority="example.com" registryType="dreg1"
  entityClass="iris" entityName="limits">
  <totalQueries>
    <perHour>2</perHour>
    <perDay>15</perDay>
  </totalQueries>
  <totalResults>
    <perHour>25</perHour>
    <perDay>200</perDay>
  </totalResults>
  <totalSessions>
    <perHour>2</perHour>
    <perDay>15</perDay>
```

```
</totalSessions>
</limits>
```

L'élément `<limits>` fournit un mécanisme permettant à un serveur d'informer un client des limites qu'il peut rencontrer dans une surutilisation du service. Le contenu décrit les limitations de service pour un client au niveau d'accès actuel. Le contenu de cet élément est le suivant :

- o `<totalQueries>` -- cet élément décrit le nombre total d'interrogations que va accepter le serveur. Les enfants de cet élément indiquent ce nombre par unité de temps. Les enfants sont `<perSecond>`, `<perMinute>`, `<perHour>`, et `<perDay>`. Chaque enfant DOIT apparaître seulement une fois comme enfant de `<totalQueries>`, mais plus d'un enfant PEUT être présent. Par exemple, un serveur pourrait indiquer qu'il va accepter 15 interrogations par minute mais seulement 60 interrogations par jour.
- o `<totalResults>` -- cet élément décrit le nombre total de résultats que le serveur va envoyer à un client. Les enfants de cet élément indiquent ce nombre par unité de temps de la même manière que `<totalQueries>`.
- o `<totalSessions>` -- cet élément décrit le nombre total de sessions que le serveur va accepter d'un client. Les enfants de cet élément indiquent ce nombre par unité de temps de la même manière que `<totalQueries>`. La définition d'une session est faite par la couche application-transport.
- o `<otherRestrictions>` -- cet élément décrit les autres restrictions qui ne peuvent être exprimables qu'en dehors de la syntaxe structurée des autres éléments fils de `<limits>`. Cet élément peut avoir des éléments fils `<description>` facultatifs, chacun ayant un attribut "language" obligatoire.
- o `<seeAlso>` -- ces éléments sont fournis comme référence à d'autres entités, comme une `<simpleEntity>` (paragraphe 4.3.7.3) décrivant une politique publiée. Voir `<seeAlso>` (paragraphe 4.3.1).

Tous ces éléments fils sont facultatifs, et un serveur peut exprimer qu'il n'a pas de limites en utilisant un élément `<limits>` sans contenu (par exemple, `<limits authority=... />`).

4.3.7.3 `<simpleEntity>`

Exemple de résultat `<simpleEntity>` :

```
<simpleEntity
  authority="example.com" registryType="dreg1"
  entityClass="local"
  entityName="notice" >
  <property name="legal" language="en">
    Example.com is reserved according to RFC 2606.
  </property>
</simpleEntity>
```

L'élément `<simpleEntity>` est fourni afin que l'opérateur du service puisse faire de simples ajouts aux autres entités sans dériver des types de registres entièrement nouveaux. Sa définition permet aux opérateurs de service d'y faire référence à partir d'autres entités (en utilisant, par exemple, un élément `<seeAlso>`). Le `<simpleEntity>` est destiné à représenter des paires de chaînes de nom et valeur, permettant à chaque paire d'être associée à un qualificatif de langage spécifique et un URI facultatif pointant sur des informations complémentaires.

Les clients peuvent facilement afficher de telles informations dans un tableau à deux colonnes. Les applications qui utilisent des données binaires ou des structures de données plus riches sortent du domaine d'application de cet élément. Lorsque de tels scénarios d'usage se présentent, un client a probablement besoin de connaissances spécifiques pour traiter ces données, et donc d'invoquer un nouveau type de registre.

4.3.8 Éléments `<control>` et `<reaction>`

Les éléments `<control>` (paragraphe 4.1) et `<reaction>` (paragraphe 4.2) permettent au client de demander au serveur un état spécial pour le traitement des interrogations. L'intention de ces éléments est de permettre l'extensibilité afin que certaines juridictions puissent adopter des politiques de traitement des interrogations sans exiger de revoir les différentes versions d'IRIS ou un type de registre.

Le présent document définit une commande, `<onlyCheckPermissions>`, et sa réaction obligée, `<standardReaction>`, pour la

conformité à CRISP [RFC3707].

Lorsque un client envoie une commande `<onlyCheckPermissions>`, il demande seulement au serveur de vérifier si les permissions adéquates sont disponibles pour exécuter les interrogations dans la demande associée. Un serveur DOIT répondre à cette commande par un élément `<standardReaction>`.

L'élément `<standardReaction>` donne à un serveur un moyen standard pour répondre aux commandes (il peut être utilisé par d'autres commandes, mais ceci relève de leur définition). Il contient quatre enfants :

- o `<controlAccepted>` -- le traitement ou état nécessaire pour la commande a été accepté.
- o `<controlDenied>` -- le traitement ou état nécessaire pour la commande a été refusé (échec transitoire).
- o `<controlDisabled>` -- le traitement ou état nécessaire pour la commande ne peut pas être activé (échec permanent).
- o `<controlUnrecognized>` -- la commande n'est pas reconnue (échec permanent).

Si `<onlyCheckPermissions>` est rejeté, le serveur DOIT alors retourner tous les ensembles de résultats appropriés (c'est-à-dire, pour chaque ensemble de recherches dans la demande) mais tous les ensembles de résultats DOIVENT être vides de résultats et DOIVENT ne contenir aucune erreur (une réaction ne fait pas partie d'un ensemble de résultats et n'est donc pas une erreur d'ensemble de résultat). Cette commande s'applique à tous les ensembles de recherches ou à aucun d'eux ; donc un serveur DOIT produire un rejet si `<onlyCheckPermissions>` ne peut pas être accepté pour tous les ensembles de recherches d'une demande.

Voici un exemple d'échange XML IRIS qui utilise ces éléments :

```
C: <?xml version="1.0"?>
C: <request xmlns="urn:ietf:params:xml:ns:iris1"
C:   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
C:
C:   <control>
C:     <onlyCheckPermissions />
C:   </control>
C:
C:   <searchSet>
C:
C:     <lookupEntity
C:       registryType="dreg1"
C:       entityClass="local"
C:       entityName="AUP" />
C:
C:   </searchSet>
C:
C: </request>

S: <?xml version="1.0"?>
S: <response xmlns="urn:ietf:params:xml:ns:iris1"
S:   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
S:
S:   <reaction>
S:     <standardReaction>
S:       <controlAccepted />
S:     </standardReaction>
S:   </reaction>
S:
S:   <resultSet>
S:     <answer>
S:
S:     <simpleEntity
S:       authority="example.com" registryType="dreg1"
S:       entityClass="local" entityName="AUP" >
S:       <property name="legal" language="en">
S:         It is illegal to use information from this service
S:         for the purposes of sending unsolicited bulk email.
S:       </property>
S:     </simpleEntity>
S:
```

S: </answer>
 S: </resultSet>
 S:
 S: </response>

4.4 Sacs de relais

IRIS emploie des sacs pour permettre à un serveur de relayer des informations à un serveur référent via le client. Ces sacs sont générés par le serveur interrogé, passés au client comme données opaques, et ensuite passés au serveur référent pour traitement. Le contenu des sacs n'est pas défini par IRIS, et le client NE DOIT PAS faire d'hypothèse sur le contenu d'un sac lorsque il le relaye d'un serveur à un autre.

Lorsque un serveur retourne un ensemble de résultats à un client, l'élément <response> peut contenir un élément fils <bags>. Cet élément fils contient un ou plusieurs éléments <bag>. Chacun d'eux DOIT contenir un attribut "id" contenant l'identifiant de type de données XML. Les références d'entité et les continuations de recherche qui ont à spécifier un sac à utiliser quand elles sont suivies DOIVENT avoir un attribut "bagRef" contenant le type de données XML IDREF. Voir au paragraphe 4.2. Cela permet que la réponse ne spécifie qu'une seule fois un sac mais permet à chaque référence d'entité ou continuation de recherche (dans tous les ensembles de résultats) d'avoir un sac distinct, en tant que de besoin.

Lorsque il suit une référence d'entité ou une continuation de recherche qui spécifie l'utilisation d'un sac, le client DOIT inclure le sac référencé dans l'ensemble de recherches comme un fils de l'élément <searchSet>. Voir au paragraphe 4.1.

Voir au paragraphe 4.2 la liste des erreurs qu'un serveur peut retourner à un client à réception d'un sac. Un serveur NE DOIT PAS ignorer un sac lorsqu'il le reçoit. Au cas où un sac ne peut pas être reconnu ou accepté, une des erreurs du paragraphe 4.2 DOIT être retournée.

Voici un exemple d'échange XML IRIS qui utilise ces éléments :

```
C: <?xml version="1.0"?>
C: <request xmlns="urn:ietf:params:xml:ns:iris1"
C:   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
C:
C:   <searchSet>
C:
C:     <bag>
C:       <simpleBag xmlns="http://example.com/">
C:         XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
C:       </simpleBag>
C:     </bag>
C:
C:     <lookupEntity
C:       registryType="dreg1"
C:       entityClass="local"
C:       entityName="AUP" />
C:
C:   </searchSet>
C:
C: </request>

S: <?xml version="1.0"?>
S: <response xmlns="urn:ietf:params:xml:ns:iris1"
S:   xmlns:iris="urn:ietf:params:xml:ns:iris1"
S:   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
S:
S:   <resultSet>
S:     <answer>
S:
S:     <entity authority="example.com" bagRef="x1"
S:       registryType="dreg1"
S:       entityClass="local" entityName="AUP"
S:       iris:referentType="ANY" >
S:     <displayName language="en">
```

```

S:   Acceptable Usage Policy
S:   </displayName>
S:   </entity>
S:
S:   </answer>
S: </resultSet>
S:
S: <bags>
S:
S:   <bag id="x1">
S:   <simpleBag xmlns="http://example.com/">
S:     AAAAB3NzaC1yc2EAAAABIwAAAIEA0ddD+W3Agl0Lel98G1r77fZ
S:   </simpleBag>
S: </bag>
S:
S: </bags>
S: </response>

```

5. Mise en série de la base de données

Cette section décrit une méthode de mise en série des entités de registre IRIS. Les descriptions contenues dans cette section se réfèrent aux éléments et attributs XML et à leurs relations à ce processus de mise en série. Ces descriptions contiennent aussi des spécifications qui sortent du domaine d'application de la syntaxe XML formelle. Cette section utilise les termes définis par la [RFC2119] pour les décrire. Pour lire cette section, se référer à la Section 6 pour les détails sur la syntaxe XML formelle.

Une base de données des entités IRIS peut être mise en série pour la mémorisation des fichiers avec XML [XML] en utilisant l'élément <serialization> défini par IRIS. Cet élément contient les dérivés de l'élément <result> et les éléments <serializedReferral>.

Les dérivés de l'élément <result> sont des entités. Les serveurs qui chargent ces entités DOIVENT placer l'entité dans les classes d'entités spécifiées les attributs 'registryType', 'entityClass', et 'entityName' d'éléments et dans toutes classes d'entités que l'entité peut appliquer conformément aux enfants explicitement définis de cet élément. Par exemple, si un type de registre a deux classes d'entités "foo" et "bar" et si un <result> dérivé a les attributs entityClass="foo" et entityName="one" et un élément fils <bar>two</bar>, le serveur va entrer cette entité dans la classe d'entité "foo" comme nom "one" et dans la classe d'entité "bar" comme nom "two".

Les serveurs qui chargent les entités comme des dérivés en série de l'élément <result> PEUVENT traduire l'attribut d'autorité. Les serveurs auront probablement à faire cela si l'autorité pour l'entité a changé.

Les éléments <serializedReferral> permettent la mise en série de références d'entité et continuations de recherche explicites. Cet élément a un élément fils <source> qui contient les attributs "authority", "resolution" (facultatif), "registryType", "entityClass", et "entityName". Les attributs de cet élément sont utilisés pour signifier à l'entité qu'elle peut être référencée pour donner ce référent.

Comme mentionné ci-dessus, il peut se trouver qu'un serveur ait besoin de traduire l'attribut d'autorité d'une entité chargée. Les mises en œuvre doivent aussi être conscientes de ce besoin pour les référents. Durant la désérialisation, les serveurs DOIVENT changer l'attribut d'autorité d'un référent (soit <entity>, soit les éléments dérivés de <entity> ou l'enfant <source> de <serializedReferral>) pour contenir une autorité valide du serveur si l'attribut en série est vide. Durant la mise en série, les serveurs et leurs processus DOIVENT laisser vide l'attribut d'autorité pour les référents dans lesquels la référence est une entité pour laquelle le serveur répond aux interrogations.

Voici un exemple d'IRIS en série :

```

<iris.serialization
  xmlns:iris="urn:ietf:params:xml:ns:iris1"
  xmlns="urn:ietf:params:xml:ns:iris1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <serviceIdentification
    authority="iana.org" registryType="dreg1"
    entityClass="iris"

```

```

entityName="id" >
<authorities>
  <authority> iana.org </authority>
</authorities>
<operatorName>
  Internet Assigned Numbers Authority
</operatorName>
<eMail>
  dbarton@iana.org
</eMail>
<seeAlso
  iris:referentType="iris:simpleEntity"
  authority="iana.org" registryType="dreg1"
  entityClass="local"
  entityName="notice">
  <displayName language="en">
    Legal Notice
  </displayName>
</seeAlso>
</serviceIdentification>

<serializedReferral>
<source
  authority="example.com" registryType="dreg1"
  entityClass="iris"
  entityName="id"/>
<entity
  iris:referentType="iris:serviceIdentification"
  authority="iana.org" registryType="dreg1"
  entityClass="iris" entityName="id"/>
</serializedReferral>

<simpleEntity
  authority="iana.org" registryType="dreg1"
  entityClass="local"
  entityName="notice" >
  <property name="legal" language="en">
    Please use the net wisely!
  </property>
</simpleEntity>

</iris:serialization>

```

6. Syntaxe XML formelle

IRIS est spécifié en notation de schéma XML. La syntaxe formelle présentée ici est une représentation complète de schéma IRIS convenable pour une validation automatique d'instances XML IRIS.

```

<?xml version="1.0"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:iris="urn:ietf:params:xml:ns:iris1"
  targetNamespace="urn:ietf:params:xml:ns:iris1"
  elementFormDefault="qualified" >

<annotation>
  <documentation>
    Internet Registry Information Service (IRIS) Schema v1
  </documentation>
</annotation>

```

Les transactions

```

<element name="request">
  <complexType>
    <sequence>
      <element
        name="control"
        type="iris:controlType"
        minOccurs="0"
        maxOccurs="1" />
      <element
        name="searchSet"
        type="iris:searchSetType"
        minOccurs="1"
        maxOccurs="unbounded" />
    </sequence>
  </complexType>
</element>

```

```

<element name="response">
  <complexType>
    <sequence>
      <element
        name="reaction"
        type="iris:reactionType"
        minOccurs="0"
        maxOccurs="1" />
      <element
        name="resultSet"
        type="iris:resultSetType"
        minOccurs="1"
        maxOccurs="unbounded" />
      <element
        name="bags"
        type="iris:bagsType"
        minOccurs="0"
        maxOccurs="1" />
    </sequence>
  </complexType>
</element>

```

Ensembles de recherche et ensembles de résultats

```

<complexType
  name="searchSetType" >
  <sequence>
    <element
      name="bag"
      type="iris:bagType"
      minOccurs="0"
      maxOccurs="1" />
    <choice>
      <element
        name="lookupEntity"
        type="iris:lookupEntityType" />
      <element
        ref="iris:query" />
    </choice>
  </sequence>
</complexType>

```

```

<complexType
  name="resultSetType" >
  <sequence>

```

```

<element
  name="answer"
  minOccurs="1"
  maxOccurs="1">
  <complexType>
    <sequence>
      <element
        ref="iris:result"
        minOccurs="0"
        maxOccurs="unbounded" />
      <element
        ref="iris:entity"
        minOccurs="0"
        maxOccurs="unbounded" />
      <element
        ref="iris:searchContinuation"
        minOccurs="0"
        maxOccurs="unbounded" />
    </sequence>
  </complexType>
</element>
<element
  name="additional"
  minOccurs="0"
  maxOccurs="1">
  <complexType>
    <sequence>
      <element
        ref="iris:result"
        minOccurs="1"
        maxOccurs="unbounded" />
    </sequence>
  </complexType>
</element>
<choice
  minOccurs="0"
  maxOccurs="1" >
  <element
    name="insufficientResources"
    type="iris:codeType" />
  <element
    name="invalidName"
    type="iris:codeType" />
  <element
    name="invalidSearch"
    type="iris:codeType" />
  <element
    name="queryNotSupported"
    type="iris:codeType" />
  <element
    name="limitExceeded"
    type="iris:codeType" />
  <element
    name="nameNotFound"
    type="iris:codeType" />
  <element
    name="permissionDenied"
    type="iris:codeType" />
  <element
    name="bagUnrecognized"
    type="iris:codeType" />
  <element
    name="bagUnacceptable"

```

```

    type="iris:codeType" />
  <element
    name="bagRefused"
    type="iris:codeType" />
  <element
    ref="iris:genericCode"/>
</choice>
</sequence>
</complexType>

```

Commandes et réactions

```

<complexType
  name="controlType">
  <sequence>
    <any
      namespace="##any"
      processContents="skip"
      minOccurs="1"
      maxOccurs="1" />
  </sequence>
</complexType>

```

```

<complexType
  name="reactionType">
  <sequence>
    <any
      namespace="##any"
      processContents="skip"
      minOccurs="1"
      maxOccurs="1" />
  </sequence>
</complexType>

```

Interrogations et examens

```

<complexType
  name="queryType" />

```

```

<element
  name="query"
  type="iris:queryType"
  abstract="true" />

```

```

<complexType
  name="lookupEntityType" >
  <attribute
    name="registryType"
    type="anyURI"
    use="required" />
  <attribute
    name="entityClass"
    type="token"
    use="required" />
  <attribute
    name="entityName"
    type="token"
    use="required" />
</complexType>

```

Résultats

```

<complexType

```

```

name="resultType">
  <attribute
    name="authority"
    use="required"
    type="token" />
  <attribute
    name="resolution"
    type="token" />
  <attribute
    name="registryType"
    use="required"
    type="anyURI" />
  <attribute
    name="entityClass"
    use="required"
    type="token" />
  <attribute
    name="entityName"
    use="required"
    type="token" />
  <attribute
    name="temporaryReference"
    default="false"
    type="boolean" />
</complexType>

<element
  name="result"
  type="iris:resultType"
  abstract="true" />

```

Erreurs

```

<complexType
  name="codeType">
  <sequence
    minOccurs="0"
    maxOccurs="unbounded">
    <element
      name="explanation">
      <complexType>
      <simpleContent>
      <extension
        base="string">
      <attribute
        use="required"
        name="language"
        type="language" />
      </extension>
      </simpleContent>
      </complexType>
    </element>
  </sequence>
</complexType>
<element
  name="genericCode"

  type="iris:codeType"
  abstract="true" />

```

Références d'entité et continuations de recherche

```

<complexType

```

```

name="entityType">
<sequence>
  <element
    name="displayName"
    minOccurs="0"
    maxOccurs="unbounded">
    <complexType>
      <simpleContent>
        <extension
          base="string">
          <attribute
            name="language"
            use="required"
            type="language" />
        </extension>
      </simpleContent>
    </complexType>
  </element>
</sequence>
<attribute
  name="authority"
  use="required"
  type="token" />
<attribute
  name="resolution"
  type="token" />
<attribute
  name="registryType"
  use="required"
  type="anyURI" />
<attribute
  name="entityClass"
  use="required"
  type="token" />
<attribute
  name="entityName"
  use="required"
  type="token" />
<attribute
  name="referentType"
  use="required"
  form="qualified"
  type="iris:referentTypeType" />
<attribute
  name="temporaryReference"
  default="false"
  type="boolean" />
<attribute
  name="bagRef"
  type="IDREF" />
</complexType>

<element
  name="entity"
  type="iris:entityType" />

<simpleType
  name="referentTypeType">
  <union
    memberTypes="QName iris:anyLiteralType" />
</simpleType>

<simpleType

```

```

name="anyLiteralType">
  <restriction
    base="string">
      <enumeration
        value="ANY" />
      </enumeration>
    </restriction>
  </simpleType>

<complexType
  name="searchContinuationType">
  <sequence>
    <element ref="iris:query" />
  </sequence>
  <attribute
    name="bagRef"
    type="IDREF" />
  <attribute
    name="authority"
    type="token"
    use="required" />
  <attribute
    name="resolution"
    type="token" />
</complexType>

<element
  name="searchContinuation"
  type="iris:searchContinuationType" />

```

Sacs

```

<complexType
  name="bagsType">
  <sequence>
    <element
      name="bag"
      minOccurs="1"
      maxOccurs="unbounded">
      <complexType>
        <complexContent>
          <extension
            base="iris:bagType">
            <attribute
              use="required"
              name="id"
              type="ID" />
            </extension>
          </complexContent>
        </complexType>
      </element>
    </sequence>
  </complexType>

<complexType
  name="bagType">
  <sequence>
    <any
      namespace="##any"
      processContents="skip"
      minOccurs="1"
      maxOccurs="1" />
    </sequence>
  </complexType>

```

Résultats dérivés à utiliser avec tous les types de registres.**See Also**

```
<element
  name="seeAlso"
  type="iris:entityType" />
```

Identification de service

```
<complexType
  name="serviceIdentificationType">
  <complexContent>
    <extension
      base="iris:resultType">
      <sequence>
        <element
          name="authorities"
          minOccurs="1"
          maxOccurs="1">
          <complexType>
            <sequence>
              <element
                name="authority"
                type="token"
                minOccurs="1"
                maxOccurs="unbounded" />
            </sequence>
          </complexType>
        </element>
        <element
          name="operatorName"
          type="string"
          minOccurs="0"
          maxOccurs="1" />
        <element
          name="eMail"
          type="string"
          minOccurs="0"
          maxOccurs="unbounded" />
        <element
          name="phone"
          type="string"
          minOccurs="0"
          maxOccurs="unbounded" />
        <element
          ref="iris:seeAlso"
          minOccurs="0"
          maxOccurs="unbounded" />
      </sequence>
    </extension>
  </complexContent>
</complexType>

<element
  name="serviceIdentification"
  type="iris:serviceIdentificationType"
  substitutionGroup="iris:result" />
```

Limites

```
<complexType
```

```

name="limitsType">
<complexContent>
  <extension
    base="iris:resultType">
    <sequence>
      <element
        name="totalQueries"
        minOccurs="0"
        maxOccurs="1" >
        <complexType>
          <group
            ref="iris:timeLimitsGroup"
            minOccurs="1"
            maxOccurs="4" />
          </complexType>
        </element>
      <element
        name="totalResults"
        minOccurs="0"
        maxOccurs="1" >
        <complexType>
          <group
            ref="iris:timeLimitsGroup"
            minOccurs="1"
            maxOccurs="4" />
          </complexType>
        </element>
      <element
        name="totalSessions"
        minOccurs="0"
        maxOccurs="1" >
        <complexType>
          <group
            ref="iris:timeLimitsGroup"
            minOccurs="1"
            maxOccurs="4" />
          </complexType>
        </element>
      <element
        name="otherRestrictions"
        minOccurs="0"
        maxOccurs="1">
        <complexType>
          <sequence>
            <element
              name="description"
              minOccurs="0"
              maxOccurs="unbounded">
              <complexType>
                <simpleContent>
                  <extension
                    base="string">
                    <attribute
                      name="language"
                      type="language"
                      use="required" />
                  </extension>
                </simpleContent>
              </complexType>
            </element>
          </sequence>
        </complexType>
      </element>

```

```

    <element
      ref="iris:seeAlso"
      minOccurs="0"
      maxOccurs="unbounded" />
  </sequence>
</extension>
</complexContent>
</complexType>
<element
  name="limits"
  type="iris:limitsType"
  substitutionGroup="iris:result" />

<group
  name="timeLimitsGroup">
  <choice>
    <element
      name="perSecond"
      type="nonNegativeInteger" />
    <element
      name="perMinute"
      type="nonNegativeInteger" />
    <element
      name="perHour"
      type="nonNegativeInteger" />
    <element
      name="perDay"
      type="nonNegativeInteger" />
  </choice>
</group>

```

Simple entité

```

<complexType
  name="simpleEntityType">
  <complexContent>
    <extension
      base="iris:resultType">
      <sequence>
        <element
          name="property"
          minOccurs="1"
          maxOccurs="unbounded">
          <complexType>
            <simpleContent>
              <extension
                base="string">
                <attribute
                  name="name"
                  type="string"
                  use="required" />
                <attribute
                  name="language"
                  type="language"
                  use="required" />
                <attribute
                  name="uri"
                  type="anyURI" />
              </extension>
            </simpleContent>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```

    </extension>
  </complexContent>
</complexType>
<element
  name="simpleEntity"
  type="iris:simpleEntityType"
  substitutionGroup="iris:result" />

```

Commandes et réactions dérivées

Permissions de vérification seulement

```

<element
  name="onlyCheckPermissions" >
  <complexType />
</element>

```

Réaction standard

```

<element
  name="standardReaction" >
  <complexType>
    <choice>
      <element
        name="controlAccepted">
        <complexType/>
      </element>
      <element
        name="controlDenied">
        <complexType/>
      </element>
      <element
        name="controlDisabled">
        <complexType/>
      </element>
      <element
        name="controlUnrecognized">
        <complexType/>
      </element>
    </choice>
  </complexType>
</element>

```

Mise en série

```

<complexType
  name="serializedReferralType">
  <sequence>
    <element name="source">
      <complexType>
        <attribute
          name="authority"
          use="required"
          type="token" />
        <attribute
          name="resolution"
          type="token" />
        <attribute
          name="registryType"
          type="anyURI"
          use="required" />
        <attribute
          name="entityClass"
          type="token"

```

```

        use="required" />
    <attribute
        name="entityName"
        type="token"
        use="required" />
    </complexType>
</element>
<choice>
    <element
        ref="iris:searchContinuation" />
    <element
        ref="iris:entity" />
</choice>
</sequence>
</complexType>

<element
    name="serialization">
    <complexType>
    <choice
        minOccurs="1"
        maxOccurs="unbounded">
        <element
            ref="iris:result" />
        <element
            name="serializedReferral"
            type="iris:serializedReferralType" />
        </choice>
    </complexType>
</element>

</schema>

```

Figure 8

7. URI IRIS

L'URI IRIS a une structure très rigide. Tous les URI IRIS ont les mêmes champs et paraissent similaires aux utilisateurs.

Mais les URI IRIS sont souples parce qu'ils permettent l'emploi de différentes méthodes pour trouver les serveurs et l'utilisation de plusieurs transports (BEEP étant celui par défaut).

7.1 Définition d'URI

Un URI IRIS [RFC2396] a la syntaxe générale suivante.

```
iris:<registry>/<resolution>/<authority>/<class>/<name>
```

L'ABNF [RFC2234] complet suit, avec certaines valeurs incluses de la [RFC2396] et de la [RFC2732].

```

iris-uri      = scheme ":" registry-urn "/" [ resolution-method ] "/" authority [ "/" entity-class "/" entity-name ]
scheme        = "iris"
authority     = // comme spécifié par la RFC2396
registry-urn = // comme spécifié par IRIS
resolution-method = *(unreserved | escaped)
entity-class  = *(unreserved | escaped)
entity-name   = *(unreserved | escaped)
unreserved   = // comme spécifié par la RFC2396
escaped       = // comme spécifié par la RFC2396

```

Un URI IRIS NE DOIT PAS être un URI relatif. La méthode de résolution, la classe d'entité, et le nom d'entité DOIVENT être du jeu de caractères UTF-8 [UNICODE] codé avec "application/x-www-form-urlencoded", comme spécifié par URL_ENC [RFC2854].

Lorsque les composants entity-class et entity-name ne sont pas spécifiés, "iris" et "id" DOIVENT être implicites par défaut. Par exemple, "iris:dreg1//com" est interprété comme "iris:dreg1//com/iris/id".

Lorsque la méthode de résolution n'est pas spécifiée, c'est par défaut la méthode de résolution directe décrite au paragraphe 7.3.2.

7.2 Schémas spécifiques du transport

Le nom de schéma "iris" n'est pas spécifique du transport d'application. Le processus de résolution d'URI PEUT déterminer le transport d'application. Un exemple d'un tel processus est la résolution directe (paragraphe 7.3.2) qui utilise les étapes mentionnées au paragraphe 7.3.3 pour déterminer le transport d'application.

Une transposition entre un transport d'application et IRIS PEUT définir un nom de schéma signifiant son utilisation avec la sémantique de l'URI IRIS.

Les règles pour déterminer quel transport d'application utiliser sont les suivantes :

- o Si un nom de schéma spécifique de transport d'application est présent, le transport d'application qu'il signifie DEVRAIT être utilisé si possible.
- o Si un client a un transport préféré et si le processus de résolution permet son utilisation, le client PEUT utiliser ce transport d'application.
- o Autrement, le transport d'application par défaut spécifié par IRIS-BEEP [RFC3983] DOIT être utilisé.

7.3 Résolution d'URI

7.3.1 Résolution dépendant du registre

L'interprétation et la résolution du composant d'autorité d'un URI IRIS peuvent être altérées avec la spécification d'une méthode de résolution dans l'URI. Si aucun composant de méthode de résolution n'est spécifié dans l'URI, la méthode de résolution directe (voir au paragraphe 7.3.2) est celle par défaut.

Des méthodes de résolution de remplacement PEUVENT être spécifiées par les types de registres. Les identifiants de ces méthodes DOIVENT être conformes à l'ABNF du paragraphe 7.1.

7.3.2 Résolution directe

Dans le processus de résolution directe, le composant d'autorité d'un URI IRIS ne peut contenir qu'un nom de domaine, un nom de domaine accompagné par un numéro d'accès, une adresse IP, ou une adresse IP accompagnée par un numéro d'accès. Le composant d'autorité du schéma indique le serveur ou l'ensemble de serveurs d'autorité responsables d'un domaine conformément aux enregistrements du DNS (paragraphe 7.3.3) si un domaine est spécifié. Si une adresse IP est spécifiée, elle indique le serveur spécifique à interroger.

Les règles pour la résolution sont les suivantes :

- o Si le composant d'autorité est un nom de domaine accompagné par un numéro d'accès comme spécifié par la RFC 2396, le nom de domaine est converti en une adresse IP via un enregistrement A ou AAAA du DNS.
- o Si le composant d'autorité est un nom de domaine par lui-même, on utilise le processus de localisation de service/transport (paragraphe 7.3.3). Si ce processus ne produit pas de résultat, le DNS est alors interrogé sur les RR A ou AAAA correspondant au nom de domaine, et le numéro d'accès utilisé est l'accès bien connu du transport utilisé conformément au paragraphe 7.2.
- o Si le composant d'autorité est une adresse IP, le DNS n'est alors pas interrogé, et l'adresse IP est utilisée directement. Si le numéro d'accès est présent, il est utilisé directement ; autrement, le numéro d'accès utilisé est l'accès bien connu du transport utilisé conformément au paragraphe 7.2.

L'utilisation d'une adresse IPv6 dans le composant d'autorité DOIT se conformer à la [RFC2732].

7.3.3 Transport et localisation de service

La méthode de résolution directe (paragraphe 7.3.2) utilise le profil des enregistrements de ressource NAPTR et SRV défini

dans S-NAPTR [RFC3958] pour déterminer à la fois la localisation d'un ensemble de serveurs pour un certain service et l'ensemble des transports qui peuvent être utilisés. Il est RECOMMANDÉ que toute méthode de résolution qui ne rend pas explicite l'utilisation du processus de résolution directe devrait utiliser S-NAPTR [RFC3958] dans tout processus qu'elle définit.

S-NAPTR [RFC3958] requiert une étiquette de service d'application. La méthode de résolution directe (paragraphe 7.3.2) utilise la forme abrégée de l'URN de registre comme étiquette de service d'application. D'autres méthodes de résolution PEUVENT spécifier d'autres étiquettes de service d'application.

Voir à l'Appendice A un échantillon d'utilisations de S-NAPTR.

7.4 Exemples d'URI IRIS

Voici quelques exemples d'URI IRIS et leur signification :

- o iris:dreg1//example.com/domain/example.com
 - * trouve un serveur d'autorité pour "example.com" conformément aux règles de résolution directe (paragraphe 7.3.2) ;
 - * le serveur est demandé pour "example.com" dans l'index, ou classe d'entité "domain", du registre "dreg1".
- o iris:dreg1//example.com
 - * trouve un serveur d'autorité pour "example.com" conformément aux règles de résolution directe (paragraphe 7.3.2) ;
 - * le serveur est demandé pour "id" dans l'index, ou classe d'entité "iris", du registre "dreg1".
- o iris:dreg1//com/domain/example.com
 - * trouve un serveur d'autorité pour "com" conformément aux règles de résolution directe (paragraphe 7.3.2) ;
 - * le serveur est demandé pour "example.com" dans l'index, ou classe d'entité "domain", du registre "dreg1".
- o iris:dreg1//192.0.2.1:44/domain/example.com
 - * suivant les règles de résolution directe (paragraphe 7.3.2) le serveur à l'adresse IP 192.0.2.1 sur l'accès 44 est interrogé en utilisant BEEP ;
 - * le serveur est demandé pour "example.com" dans l'index, ou classe d'entité "domain", du registre "dreg1".
- o iris.lwz:dreg1//192.0.2.1:44/domain/example.com
 - * suivant les règles de résolution directe (paragraphe 7.3.2), le serveur à l'adresse IP 192.0.2.1 sur l'accès 44 est interrogé en utilisant un transport d'application léger ;
 - * le serveur est demandé pour "example.com" dans l'index, ou classe d'entité "domain" du registre "dreg1".
- o iris.beep:dreg1//com/domain/example.com
 - * trouve un serveur d'autorité pour "com" conformément aux règles de résolution directe (paragraphe 7.3.2) ;
 - * utilise le transport d'application BEEP ;
 - * le serveur est demandé pour "example.com" dans l'index, ou classe d'entité "domain", du registre "dreg1".
- o iris:dreg1/bottom/example.com/domain/example.com
 - * trouve un serveur d'autorité pour "example.com" conformément aux règles de la méthode de résolution "bottom" comme défini par le type de registre urn:ietf:params:xml:ns:dreg1 ;
 - * le transport d'application utilisé est déterminé par la méthode de résolution "bottom".
 - * le serveur est demandé pour "example.com" dans l'index, ou classe d'entité "domain", du registre "dreg1".
- o iris.beep:dreg1/bottom/example.com/domain/example.com
 - * trouve un serveur d'autorité pour "example.com" conformément aux règles de la méthode de résolution "bottom" comme défini par le type de registre urn:ietf:params:xml:ns:dreg1.
 - * utilise le transport d'application BEEP ;
 - * le serveur est demandé pour "example.com" dans l'index, ou classe d'entité "domain", du registre "dreg1".

8. Listes de contrôle

8.1 Listes de contrôle de définition de registre

Les spécifications de type de registre DOIVENT inclure les définitions explicites suivantes :

- o Syntaxe formelle XML dérivant du XML IRIS.
- o URN identifiant le registre.
- o Toutes méthodes de résolution spécifiques du registre.
- o Un enregistrement de l'URN de registre abrégé comme étiquette de service d'application pour la conformité à S-NAPTR [RFC3958]. Noter qu'il y a un registre IANA différent du registre IANA d'URN de type de registre.
- o Une liste des classes d'entités bien connues.
- o Une déclaration concernant la sensibilité à la casse des noms dans chaque classe d'entité.

8.2 Listes de contrôle de transposition de transport

Les spécifications des transpositions de transport DOIVENT inclure les définitions explicites suivantes :

- o Un nom de schéma d'URI spécifique du transport.
- o Une étiquette de protocole d'application pour la conformité à S-NAPTR [RFC3958]. Voir au paragraphe 7.3.3. Noter que bien que ceci soit un registre IANA différent du registre IANA des noms de schéma d'URI, il est RECOMMANDÉ qu'ils aient la même chaîne de caractères.
- o L'ensemble des codages de jeu de caractères admis pour l'échange de XML (voir la Section 9).
- o L'ensemble des mécanismes de sécurité.

9. Considérations d'internationalisation

IRIS est représenté en XML. Les processeurs de XML sont obligés de reconnaître les codages UTF-8 et UTF-16 [UNICODE]. XML fournit les mécanismes pour identifier et utiliser d'autres codages de caractères au moyen de l'attribut "encoding" dans la déclaration <xml>. L'absence de cet attribut ou d'une marque d'ordre des octets (BOM) indique par défaut un codage UTF-8 [RFC3629]. Donc, pour des raisons de compatibilité et selon la [RFC2277], l'utilisation de UTF-8 [RFC3629] est RECOMMANDÉE avec IRIS.

La liste complète des identifiants de codage de jeu de caractères est tenue par l'IANA à [CHARSET].

La couche application-transport DOIT définir un ensemble commun de jeux de caractères pour être comprise par le client et le serveur.

La localisation de chaînes internationalisées peut exiger des informations supplémentaires de la part du client. Les définitions d'entités DEVRAIENT utiliser le type "language" défini par XML_SD [XML-S2] pour aider les clients dans le processus de localisation. Voir un exemple au paragraphe 4.3.7.3.

10. Considérations relatives à l'IANA

Le présent document propose un registre d'espace de noms XML et de schéma spécifiés dans XML_URN [RFC3688]. En conséquence, les informations d'enregistrement suivantes sont fournies pour l'IANA :

- o URN/URI : urn:ietf:params:xml:ns:iris1
- o Contact : Andrew Newton <andy@hxr.us>
Marcos Sanz <sanz@denic.de>
- o XML : le schéma XML spécifié à la Section 6.

11. Considérations sur la sécurité

La couche XML IRIS ne fournit pas par elle-même de facilités d'authentification ou de confidentialité. Elle s'appuie sur la couche application-transport pour toutes ces capacités. Les transports d'application devraient définir explicitement leurs mécanismes de sécurité (voir au paragraphe 8.2).

Les résultats de registre de référents IRIS peuvent contenir des examens d'entité et des continuations de recherche qui résultent en une opération d'interrogation de client à l'égard d'un autre service de registre. Les clients NE DEVRAIENT PAS utiliser des accreditifs et mécanismes d'authentification sujets à des attaques en répétition pour conduire les examens d'entité et continuations de recherche suivants.

12. Références

12.1 Références normatives

[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.

[RFC2234] D. Crocker et P. Overell, "BNF augmenté pour les spécifications de syntaxe : ABNF", novembre 1997.

(*Obsolète, voir [RFC5234](#)*)

- [RFC2277] H. Alvestrand, "Politique de l'IETF en matière de [jeux de caractères et de langages](#)", BCP 18, janvier 1998.
- [RFC2396] T. Berners-Lee, R. Fielding et L. Masinter, "Identifiants de ressource uniformes (URI) : Syntaxe générique", août 1998. (*Obsolète, voir [RFC3986](#)*)
- [RFC2732] R. Hinden, B. Carpenter et L. Masinter, "Format pour les adresses littérales IPv6 dans les URL", décembre 1999. (*Obsolète, voir [RFC3986](#)*) (P.S.)
- [RFC2854] D. Connelly et L. Masinter, "Type de support 'text/html'", juin 2000. (*Information*)
- [RFC3629] F. Yergeau, "[UTF-8, un format de transformation](#) de la norme ISO 10646", STD 63, novembre 2003.
- [RFC3688] M. Mealling, "[Registre XML de l'IETF](#)", BCP 81, janvier 2004.
- [RFC3958] L. Daigle, A. Newton, "[Localisation de service d'application](#) fondée sur le domaine avec les enregistrements de ressource de SRV et le service de recherche dynamique de délégation (DDDS)", janvier 2005. (P.S.)
- [RFC3983] A. Newton, M. Sanz, "[Utilisation du service d'information des registres Internet](#) (IRIS) sur le protocole extensible d'échange de blocs (BEEP)", janvier 2005. (P.S.)
- [UNICODE] The Unicode Consortium, "The Unicode Standard, Version 3", ISBN 0-201-61633-5, 2000
- [XML] World Wide Web Consortium, "Extensible Markup Language (XML) 1.0", W3C XML, février 1998, <<http://www.w3.org/TR/1998/REC-xml-19980210>>.
- [XML-NS] World Wide Web Consortium, "Namespaces in XML", W3C XML Namespaces, janvier 1999, <<http://www.w3.org/TR/1999/REC-xml-names-19990114>>.
- [XML-S1] World Wide Web Consortium, "XML Schema Part 1: Structures", W3C XML Schema, octobre 2000, <<http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>>.
- [XML-S2] World Wide Web Consortium, "XML Schema Part 2: Datatypes", W3C XML Schema, octobre 2000, <<http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>>.

12.2 Références pour information

- [CHARSET] <<http://www.iana.org/assignments/character-sets>>
- [RFC3080] M. Rose, "Cœur du [protocole extensible d'échange de blocs](#) (BEEP)", mars 2001. (P.S.)
- [RFC3707] A. Newton, "Exigences pour le protocole d'enregistrement croisé de service Internet (CRISP)", février 2004. (*Info.*)
- [RFC3912] L. Daigle, "[Spécification du protocole WHOIS](#)", septembre 2004. (*D.S.*)
- [RFC3982] A. Newton, M. Sanz, "IRIS : [un type de registre de domaines \(dreg\)](#) pour le service d'information des registres Internet (IRIS)", janvier 2005. (P.S.)

Appendice A. Utilisations de S-NAPTR et d'IRIS

A.1. Exemple de S-NAPTR avec IRIS

Cette section donne un exemple d'utilisation de S-NAPTR [RFC3958] par IRIS. Dans cet exemple, il y a deux types de registres : REGA et REGB. Il y a aussi deux transports d'application IRIS : iris-a et iris-b. Ceci étant donné, l'utilisation de S-NAPTR offre :

1. Un moyen par lequel un opérateur peut séparer l'ensemble des serveurs qui traitent REGA de l'ensemble de serveurs qui traitent REGB. C'est à dire que l'opérateur est capable de séparer l'ensemble de serveurs qui desservent les données pour

REGA de l'ensemble de serveurs qui desservent les données pour REGB.

2. Un moyen par lequel un opérateur peut distinguer l'ensemble des serveurs qui traitent iris-a de l'ensemble de serveurs qui traitent iris-b. C'est à dire que l'opérateur est capable de séparer l'ensemble de serveurs qui fonctionnent avec le protocole iris-a desservant les données de REGA et REGB de l'ensemble de serveurs qui fonctionnent avec le protocole iris-b pour desservir les données de REGA et REGB.
3. Un moyen par lequel un opérateur peut spécifier quel ensemble de serveurs faire fonctionner et quel ensemble des serveurs ci-dessus déléguer à un autre opérateur.

Pour mettre en œuvre la première caractéristique, l'opérateur déploie ce qui suit dans sa zone DNS :

example.com.

```
;;          order  pref  flags  service          re  replacement
IN NAPTR  100    10   ""     "REGA:iris-a:iris-b" ""   rega.example.com
IN NAPTR  100    10   ""     "REGB:iris-a:iris-b" ""   regb.example.com
```

Pour mettre en œuvre la seconde caractéristique, l'opérateur ajoute alors ce qui suit dans leur zone DNS :

reg.example.com.

```
;;          order  pref  flags  service          re  replacement
IN NAPTR  100    10   "s"    "REGA:iris-a"    ""   _iris-a_udp.example.com
```

regb.example.com.

```
IN NAPTR  100    10   "s"    "REGA:iris-b"    ""   _iris-b_tcp.example.com
```

_iris-a_udp.example.com.

```
;;          pref  weight  port  target
IN SRV    10      0       34   big-a.example.com.
IN SRV    20      0       34   small-a.example.com.
```

_iris-b_tcp.example.com.

```
;;          pref  weight  port  target
IN SRV    10      0       34   big-b.example.com.
IN SRV    20      0       34   small-b.example.com.
```

Finalement, un opérateur peut décider de faire fonctionner les services REGA tout en déléguant les services REGB à quelqu'un d'autre. Voici comment cela est fait :

example.com.

```
;;          order  pref  flags  service          re  replacement
IN NAPTR  100    10   ""     "REGA:iris-a:iris-b" ""   rega.example.com
IN NAPTR  100    10   ""     "REGB:iris-a:iris-b" ""   quelquundautre.com
```

Ou l'opérateur peut décider de faire fonctionner les services REGB sous le protocole/transport iris-a tout en déléguant les services REGB sous le protocole/transport iris-b à quelqu'un d'autre.

example.com.

```
;;          order  pref  flags  service          re  replacement
IN NAPTR  100    10   ""     "REGB:iris-a:iris-b" ""   regb.example.com
IN NAPTR  100    10   "s"    "REGB:iris-a"    ""   _iris-a_udp.example.com
IN NAPTR  100    10   "s"    "REGB:iris-b"    ""   _iris-b_tcp.quelquundautre.com
```

_iris-a_udp.example.com.

```
;;          pref  weight  port  target
IN SRV    10      0       34   big-a.example.com.
IN SRV    20      0       34   small-a.example.com.
```

Noter qu'alors que ce dernier exemple est possible, il n'est probablement pas à conseiller à cause des problèmes de fonctionnement qu'il implique pour la synchronisation des données entre example.com et quelquundautre.com. Il n'est donné ici que comme exemple de ce qui est possible.

A.2 Utilisation de S-NAPTR pour la cohabitation

Étant donné les exemples de l'Appendice A.1, l'utilisation de S-NAPTR pourrait faire partie d'une stratégie de transition pour la cohabitation de protocoles résolvant les problèmes de CRISP [RFC3707].

Par exemple, le type de données pour les informations de domaine pourrait recevoir l'étiquette de service d'application de "DREG1". Cela étant, le champ de service d'un enregistrement NAPTR conforme à S-NAPTR pourrait se lire : "DREG1:whois:iris-beep".

Ce champ de service porte les données de domaine, comme défini par CRISP, et est disponible via les deux protocoles iris-beep et whois. L'étiquette de protocole d'application whois se réfère à la RFC 954 [RFC3912].

Appendice B. Philosophie de la conception d'IRIS

Au delà des arguments concrets qui pourraient être avancés derrière une analyse détaillée des bits qui traversent l'éther, il y a d'autres raisons abstraites au développement d'IRIS. Le présent appendice tente de les expliquer.

B.1 Principe de base

IRIS a été conçu comme un service d'annuaire pour des registres publics des ressources de l'Internet. Les prémisses sont :

- o Un client devrait être capable d'examiner tout ensemble de données de tout type de registre. Cet examen devrait impliquer une définition directe et cohérente pour trouver le registre et devrait entraîner une touche pour un seul indice de données dans le registre.
- o Tout ce qu'il y a en plus, comme une recherche dans l'arborescence du DNS pour trouver le registre ou chercher à travers plusieurs index dans un registre, exige d'un client qu'il ait des connaissances particulières sur les relations entre les données contenues dans un registre.

Donc, IRIS fait les choses suivantes :

- o Il spécifie le langage de base de schéma utilisé par tous les registres pour spécifier leurs schémas.
- o Il fournit le cadre de base pour qu'un registre fasse référence à une entité dans un autre type de registre.

Et donc, IRIS ne fait pas ce qui suit :

- o Il ne spécifie pas un langage d'interrogation commun à tous les types de registres. Un langage d'interrogation commun imposé à tous les types de registres résulte généralement en la désactivation de certaines fonctions par un opérateur de serveur afin de satisfaire des niveaux de performances acceptables, ce qui donne un langage d'interrogation commun qui ne fonctionne pas de façon commune.
- o Il n'impose aucune relation entre les ensembles de données dans aucun type de registre, comme de spécifier une arborescence. Il y a de nombreux types de ressources de l'Internet, et elles ne partagent pas toutes le même style de relations avec les ensembles de données qu'elles contiennent. Si il n'y a pas une harmonisation naturelle, l'imposition d'une relation commune est souvent une source de problèmes et non un avantage.

B.2 Attrait d'un client universel

Les prémisses de la conception d'IRIS signifient que, pour les services d'annuaire, il n'y a rien qui ressemble à un client universel (ou si il y en a un, c'est ce qu'on appelle généralement le "navigateur de la Toile").

Pour IRIS, la chose la plus proche d'un client universel est celle qui peut "examiner" les données et peut être capable d'afficher les données d'une façon rudimentaire. Pour qu'un client soit capable de "chercher" les données ou de les afficher d'une façon qui soit vraiment acceptable pour l'utilisateur, il doit avoir une connaissance spécifique du type de données qu'il veut restituer.

Les tentatives pour équiper un client universel d'un langage commun d'interrogation ne sont pas non plus très utiles. Un langage commun d'interrogation peut être appliqué à un domaine de problème spécifique, qui exigerait d'un utilisateur qu'il soit expert la fois dans le langage commun d'interrogation et dans le domaine du problème. À la fin, le résultat est généralement le développement d'un client spécifique du domaine de problème mais encombré de la traduction des désirs de l'utilisateur et de l'aspect plus petit commun dénominateur du langage d'interrogation .

B.3 Considérations de serveur

Comme mentionné ci-dessus, IRIS a été conçu pour les besoins de service d'annuaire de registres mis à la disposition du

public. Sous cet éclairage, certains aspects de services d'annuaire plus généralisés sont un obstacle dans un environnement qui n'a pas les mêmes considérations de contrôle et de sécurité qu'un réseau géré.

Par exemple, un langage commun d'interrogation peut donner une grande souplesse aussi bien à l'utilisateur puissant qu'à l'utilisateur abusif. Celui-ci pourrait facilement soumettre une interrogation à travers de multiples index avec des valeurs partielles. Une telle interrogation n'aurait aucune autre utilité que de causer un déni de service aux autres usagers. Pour combattre cela, un opérateur de service doit restreindre les types d'interrogations qui causent des dommages aux performances globales, et cet acte annule le bénéfice d'un langage commun d'interrogation.

Une autre considération pour les performances de serveur est l'absence d'une relation exigée entre les données. Parce que des ensembles de données ont souvent des relations qui diffèrent, une approche de type taille unique ne va pas bien avec tous les types de registres. De plus, les services publics tendent à avoir des exigences de niveau de service qui ne peuvent pas être raisonnablement satisfaites en transformant des magasins de données complets d'un format natif en un format qui met en application un ensemble artificiel de relations.

Pour faire face à ces problèmes, les opérateurs de services au public tendent à créer leurs propres analyseurs personnalisés d'interrogations et leurs magasins de données de l'autre côté. Mais faire cela met en question l'utilisation d'un service de répertoires généralisé.

Finalement, IRIS est construit sur un ensemble de couches de technologies standard. Cela permet aux opérateurs de services d'échanger les composants pour satisfaire aux besoins de leur environnement particulier.

B.4 Examens, recherches, et classes d'entités

IRIS prend en charge à la fois les examens et les recherches. Conceptuellement, la différence entre les deux est la suivante :

Un "examen" est une seule interrogation qui a une valeur discrète sur un seul indice.

Tout ce qui va plus loin, comme des interrogations sur une valeur partielle, des interrogations à travers plusieurs indices, ou plusieurs interrogations à un seul indice sont une "recherche".

Les examens sont accomplis par l'interrogation définie <lookupEntity>. Cette interrogation spécifie un nom discret, appelé le nom d'entité, à interroger dans un seul index, appelé la classe d'entité. Donc, les mises en œuvre peuvent considérer un type de registre comme étant composé de plusieurs indices, un pour chaque classe d'entité définie.

Il n'y a pas de recherches standard dans IRIS. Chaque type de registre définit son propre ensemble de recherches.

B.5 Références d'entités, continuations de recherches, et portée

À cause de cet effet sur le comportement du client et des effets collatéraux qu'un tel comportement peut avoir sur les serveurs, IRIS fait une claire distinction entre les références d'entité (<entity>) et les continuations de recherche (<searchContinuation>). Ce n'est pas un ajout, mais le cœur fondamental du protocole.

La distinction entre les assertions suivantes est très importante pour un client : "Examinez cela et vous allez trouver ce que vous cherchez", "Examinez cela et il est possible que vous trouviez ce que vous cherchez, ou vous pourriez trouver autre chose, ou peut être rien".

Finalement, comme IRIS ne fait pas d'hypothèse et n'impose aucune exigence sur les relations entre les données dans un registre, cela s'étend aussi aux données du même type de registre étalées à travers de multiples zones d'autorité. Cela signifie que IRIS n'impose aucune exigence à la portée des références d'entité ou continuations de recherche. La portée est déterminée par ce dont a besoin le type de registre et par ce que le type de registre permet à l'opérateur de service.

Appendice C. Remerciements

La terminologie utilisée dans le présent document pour décrire les espaces de noms et les espaces de nom des espaces de noms est maintenant beaucoup plus claire grâce aux habiles tactiques de débat de Leslie Daigle. Avant, c'était beaucoup plus confus. De plus, Leslie a fourni beaucoup de précisions sur les détails des URI, des URN, et des enregistrements de ressources NAPTR/SRV.

L'inutilité de nombreuses autres complexités techniques a été démontrée par David Blacka et elles ont été supprimées. Et sa

mise en œuvre de IRIS a aidé a arrondir les angles.

Adresse des auteurs

Andrew L. Newton
VeriSign, Inc.
21345 Ridgetop Circle
Sterling, VA 20166
USA
téléphone : +1 703 948 3382
mél : anewton@verisignlabs.com ; andy@hxr.us
URI : <http://www.verisignlabs.com/>

Marcos Sanz
DENIC eG
Wiesenhuettenplatz 26
D-60329 Frankfurt
Germany
mél : sanz@denic.de
URI : <http://www.denic.de/>

Déclaration complète de droits de reproduction

Copyright (C) The Internet Society (2005).

Le présent document est soumis aux droits, licences et restrictions contenus dans le BCP 78, et à www.rfc-editor.org, et sauf pour ce qui est mentionné ci-après, les auteurs conservent tous leurs droits.

Le présent document et les informations contenues sont fournis 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.

Propriété intellectuelle

L'IETF ne prend pas position sur la validité et la portée de tout droit de propriété intellectuelle ou autres droits qui pourraient être revendiqués au titre de la mise en œuvre ou l'utilisation de la technologie décrite dans le présent document ou sur la mesure dans laquelle toute licence sur de tels droits pourrait être ou n'être pas disponible ; pas plus qu'elle ne prétend avoir accompli aucun effort pour identifier de tels droits. Les informations sur les procédures de l'ISOC au sujet des droits dans les documents de l'ISOC figurent dans les BCP 78 et BCP 79.

Des copies des dépôts d'IPR faites au secrétariat de l'IETF et toutes assurances de disponibilité de licences, ou le résultat de tentatives faites pour obtenir une licence ou permission générale d'utilisation de tels droits de propriété par ceux qui mettent en œuvre ou utilisent la présente spécification peuvent être obtenues sur le répertoire en ligne des IPR de l'IETF à <http://www.ietf.org/ipr> .

L'IETF invite toute partie intéressée à porter son attention sur tous copyrights, licences ou applications de licence, ou autres droits de propriété qui pourraient couvrir les technologies qui peuvent être nécessaires pour mettre en œuvre la présente norme. Prière d'adresser les informations à l'IETF à ietf-ipr@ietf.org .

Remerciement

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