

Groupe de travail Réseau  
**Request for Comments : 4479**  
 Catégorie : Sur la voie de la normalisation

J. Rosenberg, Cisco Systems  
 juillet 2006  
 Traduction Claude Brière de L'Isle

## Modèle de données pour Présence

### Statut du présent mémoire

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

*(La présente traduction incorpore les errata 2963, 80, et 2131)*

### Notice de Copyright

Copyright (C) The Internet Society (2006).

### Résumé

Le présent document définit le modèle sous-jacent de données de présence utilisé par les agents de présence du protocole d'initialisation de session (SIP, *Session Initiation Protocol*) pour les extensions de développement de messagerie instantanée et de présence (SIMPLE, *SIP Instant Messaging et Presence Leveraging Extensions*). Le modèle de données donne des lignes directrices sur la façon de transposer les divers systèmes de communications en documents de présence d'une façon cohérente.

## Table des matières

1. Introduction.....	1
2. Définitions.....	2
3. Le modèle.....	3
3.1 URI de présentité.....	4
3.2 Personne.....	4
3.3 Service.....	5
3.4 Appareil.....	8
3.5 Ambiguïté de modélisation.....	10
3.7 État ou caractéristiques.....	11
3.8 Propriétés du document Présence.....	11
4. Motifs du modèle.....	12
5. Codage.....	12
5.1 Schémas XML.....	13
6. Extension du modèle de Presence.....	15
7. Exemple de document Présence.....	15
7.1 Client IM de base.....	15
8. Considérations sur la sécurité.....	17
9. Considérations d'internationalisation.....	17
10. Considérations relatives à l'IANA.....	17
10.1 Enregistrement de sous espace de noms d'URN.....	17
10.2 Enregistrements de schémas XML.....	18
11. Remerciements.....	18
12. Références.....	19
12.1 Références normatives.....	19
12.2 Références pour information.....	19
Adresse de l'auteur.....	20
Déclaration complète de droits de reproduction.....	20

## 1. Introduction

Presence porte la capacité et la volonté d'un utilisateur de communiquer à travers un ensemble d'appareils. La [RFC2778]

définit un modèle et une terminologie pour décrire les systèmes qui fournissent des informations de présence. La [RFC3863] définit un format de document XML [XML], [XML-1], [XML-2] pour représenter les informations de présence. Dans ces spécifications, les informations de présence sont modélisées comme une série de tuplets, dont chacun contient un état, une adresse de communication, et autres balises. Cependant, aucune de ces spécifications ne donne de lignes directrices sur ce qu'exactly un tuple est censé modéliser, ou comment transposer les systèmes de communications du monde réel (et en particulier, ceux construits autour du protocole d'initialisation de session (SIP, *Session Initiation Protocol*) [RFC3261]) en un document de présence.

En particulier, plusieurs concepts importants ne sont pas clairement modélisés ou bien délimités par les RFC 2778 et 3863. Ce sont les suivants :

**Service** : un service de communications, comme la messagerie instantanée (IM) ou la téléphonie, est un système pour l'interaction entre des utilisateurs, qui fournit certaines modalités ou contenus.

**Appareil** : un appareil de communications est un composant physique avec lequel un utilisateur interagit afin de faire ou recevoir des communications. Des exemples sont un téléphone, une tablette, ou un ordinateur personnel.

**Personne** : une personne est l'utilisateur final, et pour les besoins de présence, caractérisé par des états, comme "occupé" ou "triste", qui impactent sa capacité et sa volonté de communiquer.

La présente spécification définit ces concepts plus complètement au moyen d'un modèle de données de présence, et définit concrètement comment prendre les systèmes réels et les transposer en documents de présence en utilisant ce modèle. Ce modèle de données est défini en termes d'extension à la RFC3863.

## 2. Définitions

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].

Le présent document utilise de nombreux termes supplémentaires au delà de ceux définis dans les RFC 2778 et RFC 3863. Ces nouveaux termes sont les suivants :

**Appareil** : un appareil modélise l'environnement physique dans lequel les services se manifestent aux utilisateurs. Les appareils ont des caractéristiques qui sont utiles pour permettre à un utilisateur de faire un choix sur les services de communications à utiliser.

**Service** : un service modélise une forme de communication qui peut être utilisée pour interagir avec l'utilisateur.

**Personne** : une personne modélise l'utilisateur humain et ses états qui sont pertinents pour les systèmes de présence.

**Occurrence** : une seule description d'un service particulier, un appareil particulier, ou une personne. Il peut y avoir plusieurs occurrences pour un service ou appareil particulier, ou plusieurs occurrences de personnes dans un document de format de données d'informations de présence (PIDF, *Presence Information Data Format*) dans les cas où il y a une ambiguïté qui est au mieux résolue par l'observateur.

**Présentité** : une présentité combine des informations d'appareils, de services, et de personne pour un tableau complet de l'état de présence d'un utilisateur sur le réseau.

**URI de présentité** : URI qui agit comme identifiant unique pour une présentité et fournit un moyen pour obtenir des informations de présence sur cette présentité.

**Composant de données** : une des parties de l'appareil, service, ou personne d'un document de présence.

**État** : informations de présence sur un service, personne, ou appareil qui change normalement au fil du temps, à la différence des caractéristiques, qui sont généralement statiques.

**Caractéristiques** : informations de présence sur un service, personne, ou appareil qui sont généralement fixées pour longtemps, et de nature descriptives. Les caractéristiques sont utiles pour fournir un contexte qui identifie le

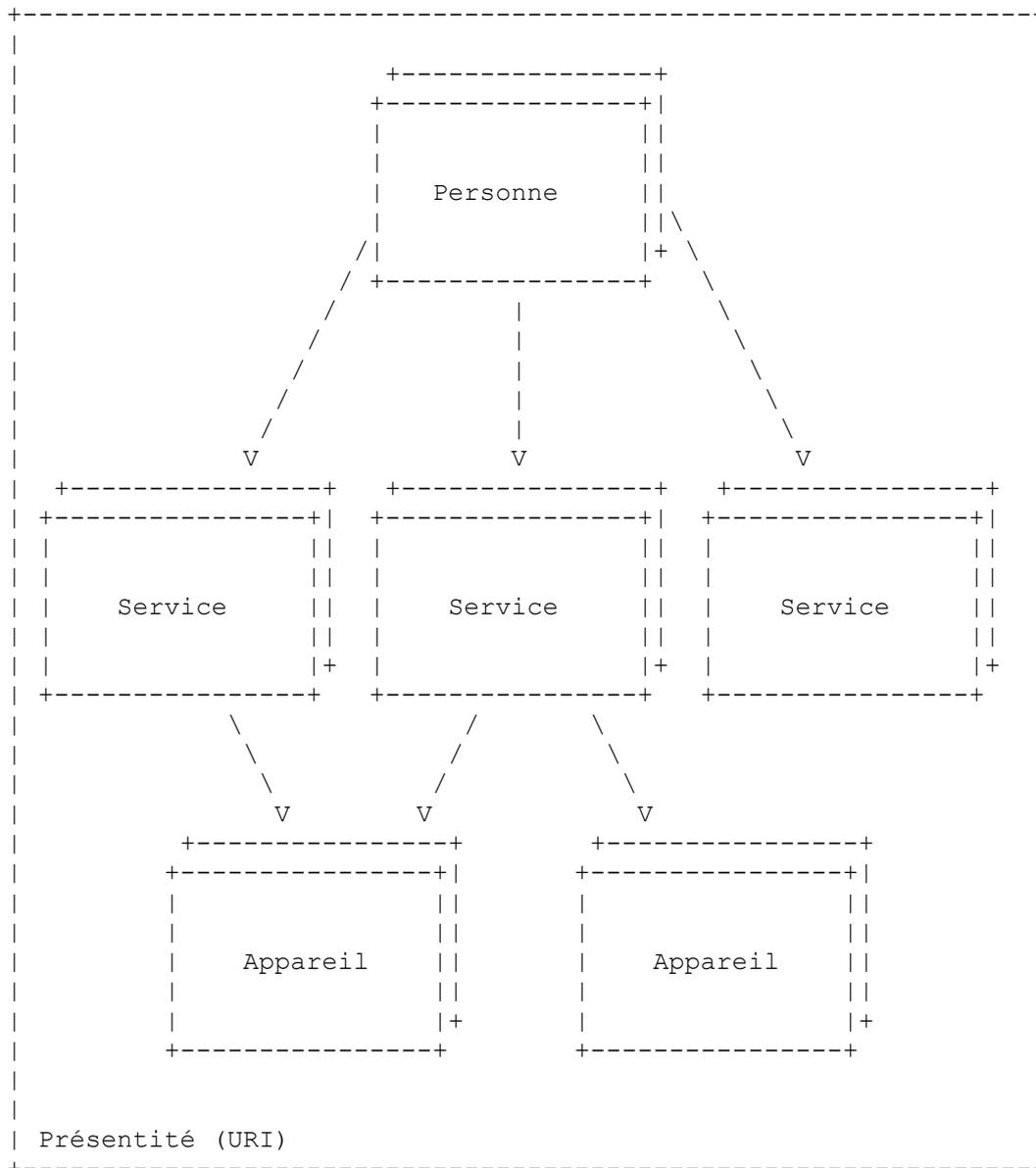
service ou appareil comme différent d'un autre service ou appareil.

Attribut : un état ou une caractéristique. Il représente un seul élément d'informations de présence.

Attribut de présence : synonyme de attribut.

Composition : acte de combinaison d'un ensemble de données de présence et d'événements sur une présentité dans un tableau cohérent de l'état de cette présentité.

### 3. Le modèle



**Figure 1**

Le modèle de données pour présence est montré à la Figure 1. Le modèle cherche à décrire la présentité, identifiée par un URI de présentité. Il y a trois composants dans le modèle : la personne, le service, et l'appareil. Ces trois composants de données contiennent des informations (appelées attributs) qui fournissent une description de certains des aspects du service, personne, ou appareil. Il est central dans ce modèle que chaque attribut soit affilié au service, personne, ou appareil parce que ils décrivent ce service, personne, ou appareil. Ceci est différent d'un modèle par lequel les attributs sont associés au service, personne, ou appareil parce qu'ils ont été rapportés par ce service, personne, ou appareil. Par exemple, si un

téléphone cellulaire rapporte qu'un utilisateur est en réunion, cela sera fait en incluant un attribut au titre des informations de personne, indiquant un état de "en réunion". Les informations de présence peuvent aussi inclure des informations sur le téléphone cellulaire comme appareil. Cependant, même si il est l'appareil qui rapporte que l'utilisateur est en réunion, "en réunion" est un fait qui décrit l'utilisateur humain, et pas son appareil physique. Par conséquent, cet attribut est placé dans le composant "personne" du document.

### 3.1 URI de présentité

L'identifiant de la présentité est un URI. Pour chaque présentité unique dans le réseau, il y a un ou plusieurs URI de présentité. Une présentité peut avoir plusieurs URI parce que elles sont identifiées par un URI provenant du schéma Presence (pres) [RFC3859] et par un URI spécifique du protocole, comme un URI SIP [RFC3261] ou un identifiant de ressource internationalisée du protocole extensible de messagerie et de présence (XMPP IRI, *Extensible Messaging et Presence Protocol Internationalized Resource Identifier*) [RFC4622]. Ou, ce peut être parce que un utilisateur a plusieurs alias dans un domaine, dont tous sont des identifiants équivalents pour la présentité.

Quand un document est construit, l'URI de présentité est idéalement réglé à l'identifiant utilisé pour demander le premier le document. Par exemple, si un document a été demandé au moyen d'une demande SIP SUBSCRIBE, l'URI de présentité va correspondre à l'URI de demande de la demande SUBSCRIBE. Cela suit le principe de moindre surprise, car l'entité qui demande le document peut ne pas connaître les autres identifiants de la présentité.

Sans considération du schéma duquel l'URI est tiré, l'URI de présentité est indépendant de tous services ou appareils que possède la présentité. Cependant, l'URI n'est pas juste un nom - il représente une ressource à laquelle elle peut être abonnée, afin de trouver l'état de l'utilisateur. Quand l'URI est un URI SIP, il va souvent être l'adresse de rattachement pour l'utilisateur, à laquelle les appels SIP peuvent être dirigés. Cette équivalence n'est pas rendue obligatoire par la présente spécification, mais est une configuration recommandée pour faciliter la tâche de mémorisation des identifiants pour les utilisateurs.

### 3.2 Personne

Le composant données de personne modélise les informations sur l'utilisateur que les données de présence essayent de décrire. Ces informations consistent en les caractéristiques de l'utilisateur, et leur état.

Les caractéristiques d'une personne sont les information statiques sur un utilisateur qui ne changent pas dans des circonstances normales. De telles informations peuvent inclure des caractéristiques physiques, comme l'âge et la taille. Un autre exemple de caractéristique d'une personne est un alias. Un alias est un URI qui identifie le même utilisateur, mais avec un URI de présentité différent. Par exemple, une présentité "sip:bob@exemple.com" pourrait avoir un document de présence avec un composant de personne qui indique un alias de "sip:robert@exemple.com" et "sip:r.smith@exemple.com".

Les informations d'état sur une présentité représentent les informations dynamiques sur un utilisateur. Cela consiste normalement en des choses que \*l'usager\* est en train de faire, les endroits où \*l'usager\* est, les sentiments que \*l'usager\* a, et ainsi de suite. Des exemples d'états normaux d'une personne sont "en réunion", "au téléphone", "sorti déjeuner", "heureux", et "en train d'écrire des projets Internet". La ligne de partage entre les informations d'état statiques et les informations d'état dynamiques est confuse, et il n'est pas important qu'une ligne soit tracée. Le modèle ne différencie pas d'une façon syntaxiquement ou sémantiquement significative entre ces deux types d'attributs.

Dans le modèle, il ne peut y avoir qu'un seul composant de personne par présentité. En d'autres termes, le composant de personne modélise un seul être humain, et inclut des caractéristiques et états qui sont relatifs aux états de communication pour un seul être humain. Bien sûr, le système n'a aucun moyen de vérifier que l'humain décrit par le composant de personne est réellement un seul être humain, par opposition à un groupe d'utilisateurs, ou même un chien. Comme on dit, "sur l'Internet, personne ne sait que tu es un chien", et la même chose est vraie ici. Le composant de personne est une facade pour une seule personne ; tout ce qui peut ressembler à une seule personne peut être modélisé avec cette facade.

Par exemple, considérons la tâche d'utiliser un document de présence pour décrire un bureau d'aide aux consommateurs. Le composant de personne peut être considéré comme étant "occupé" si aucun des personnels d'assistance n'est disponible, et "à déjeuner" si le département du bureau d'assistance a un déjeuner de groupe ensemble. L'observateur qui reçoit le document va considérer le bureau d'aide comme une seule personne ; rien dans le document (sauf peut-être l'élément "note", si sa valeur est "bureau d'aide" ou quelque chose de similaire) ne porte l'information qui indiquerait que la personne en question est en fait un bureau d'aide.

Cependant, il peut y avoir de multiples occurrences du composant de personne. Cela arrive dans des cas où l'état du composant de personne est ambigu, comme expliqué au paragraphe 3.5.

### 3.3 Service

Chaque présentité a accès à un certain nombre de services. Chacun d'eux représente un point d'accessibilité pour les communications, qui peut être utilisé pour interagir avec l'utilisateur. Des exemples de services sont la téléphonie (c'est-à-dire, le service de téléphone traditionnel fondé sur le circuit) les interphones, la messagerie instantanée, le service de messages courts (SMS), et le service de messages multimédia (MMS).

Il est difficile de donner une définition précise pour service. Une approche raisonnable est de modéliser chaque logiciel ou agent matériel dans le système comme un service. Si un utilisateur lance une application de logiciel de téléphone sur son ordinateur personnel, cela représente un service. Si un utilisateur a un appareil de vidéophonie, cela représente un autre service. Ceci est effectivement une vue physique des services. Cependant cette définition n'est plus satisfaisante quand un service est étalé sur de nombreux agents logiciels ou appareils. Par exemple, un URI SIP qui représente une adresse de rattachement peut être acheminé à un téléphone logique ou à un vidéophone, ou aux deux. Dans chaque cas, on peut tenter de définir plutôt un service sur la base de son adresse sur le réseau. Cette définition ne tient pas non plus quand on modélise les appareils ou applications qui reçoivent des appels et les répartissent à différents "assistants" fondés sur des logiques potentiellement complexes. Par exemple, un téléphone cellulaire pourrait héberger plusieurs applications SIP, dont chacune peut "enregistrer" des détenteurs différents sur la base de la méthode ou même du type de corps de la demande. Chacune de ces applications ou détenteurs peut légitimement être considérée comme un service, mais elles n'ont pas d'adresse distincte des autres sur le réseau.

À cause de cette difficulté inhérente de définir précisément un service, le modèle de données n'essaye pas de contraindre ce qui peut être considéré comme un service. C'est plutôt tout ce qui peut être considéré comme un service tant qu'il présente un ensemble de propriétés clés définies par ce modèle. En particulier, chaque service est associé à des caractéristiques qui identifient la nature et les capacités de ce service, avec les informations d'accès qui indiquent comment se connecter au service, avec des informations d'état qui représentent l'état de ce service, et des informations relatives qui décrivent les façons dont ce service se rapporte aux autres qui sont associés à la présentité.

Par conséquent, dans ce modèle, les services ne sont pas explicitement énumérés. Il n'y a pas de registre central où on trouve les identifiants de chaque service. En conséquence, chaque service n'a pas un seul attribut "service" avec des valeurs telles que "ptt" ou "téléphonie". Cela ne signifie pas que ces monistes consolidés ne soient pas utiles ; bien sûr, ils représentent un résumé essentiel de ce qu'est le service. Un tel résumé est utile pour créer des images qui permettent à un utilisateur de choisir un service plutôt qu'un autre. Un observateur est libre de créer de telles informations de résumé à partir de toutes les informations associées à un service. Les informations d'accès fournissent souvent des informations précieuses pour créer de tels résumés. Souvent, le schéma de l'URI est synonyme de la vue de ce qu'est un service. Un URI "sms" [RFC5724] indique clairement le SMS, par exemple. Pour certains URI, il peut y avoir de nombreux services disponibles, par exemple, SIP ou tel [RFC3966], et dans ce cas le schéma est moins significatif comme moyen de créer un résumé. Les informations d'accès pourraient aussi indiquer que certains logiciels d'application doivent être invoqués (comme un jeu vidéo) et dans ce cas, l'aspect des informations d'accès va être utile pour générer une image représentant le jeu.

#### 3.3.1 Caractéristiques

Chaque service est accompagné de caractéristiques qui décrivent la nature et les capacités du service qui va être expérimenté lorsque un observateur invoque cet URI. La nature d'un service est un ensemble de propriétés qui sont relativement statiques à travers les sessions de communication établies avec ce service. La nature d'un service tend à être descriptive. Des exemples de la nature d'un service sont qu'il représente une réponse vocale interactive ou un serveur de messagerie vocale, qu'il est un automate, ou qu'il est un service de téléphonie utilisé pour le travail. D'un autre côté, les capacités représentent des propriétés qui peuvent être affichées, et leur affichage dépend de la négociation et autres fonctions dynamiques qui ont lieu durant l'établissement de session. Des exemples de telles capacités sont le type de supports qui peuvent être utilisés, la directionnalité des communications qui sont permises, les extensions SIP prises en charge, et ainsi de suite. Les capacités peuvent être très complexes ; par exemple, la [RFC2533] décrit un modèle pour représenter les capacités par des fonctions booléennes N-aires. Il est difficile de différencier une capacité d'une modalité (par exemple, ce service fait seulement de la voix) d'une caractéristique qui représente la nature d'un service. Cependant, il n'est pas important de le faire.

Les caractéristiques sont importantes quand plusieurs services sont indiqués. C'est parce que l'objet d'une liste de plusieurs services dans un document de présence est de donner à l'observateur un \*choix\*. C'est à dire que la présentité offre explicitement à l'observateur une opportunité de la contacter en utilisant une multiplicité de différents services. Pour aider l'observateur à prendre une décision, le document de présence inclut des caractéristiques de chaque service qui aide à différencier les services les uns des autres et donne à l'observateur le contexte dans lequel faire un choix.

Comme cet objet est principalement de faciliter le choix, les capacités n'imposent pas d'exigence sur la façon dont un utilisateur atteint le service. Par exemple, si un document de présence inclut deux services, et que l'un supporte seulement l'audio tandis que l'autre supporte seulement la vidéo, cela ne signifie pas que, quand il contacte le premier service, un utilisateur doit offrir seulement un flux audio, ou que quand il contacte le second service, un utilisateur doit offrir seulement un flux vidéo. Un utilisateur peut utiliser la politique locale à sa discrétion pour déterminer quelles capacités ou modalités de communications sont offertes quand il choisit de se connecter à un service. Il n'est pas nécessaire qu'un observateur ajoute les préférences de l'appelant SIP [RFC3841] pour demander l'acheminement de la demande sur un service qui a les caractéristiques décrites dans le document de présence.

Si, afin d'atteindre un service, l'agent d'utilisateur doit générer une demande qui présente une capacité particulière ou contient un en-tête spécifique, elle est alors indiquée séparément des informations d'accès, décrites ci-dessous.

Une importante caractéristique de chaque service est la liste des appareils sur laquelle ce service s'exécute. Chaque appareil est identifié de façon univoque par un identifiant d'appareil. À ce titre, les caractéristiques de service peuvent inclure une liste d'identifiants d'appareils. Un document de présence peut aussi contenir des informations sur chaque appareil, mais c'est une partie séparée du document. Bien sûr, les informations sur chaque appareil peuvent n'être même pas présentes dans le document. Dans ce cas, les identifiants d'appareils mentionnés pour chaque service ne sont rien de plus que des identifiants de corrélation, utiles pour déterminer quand deux services fonctionnent sur le même appareil. L'avantage de ce modèle est que les informations sur les appareils peuvent être filtrées dans un document de présence, et donc les informations de service, qui incluent les identifiants d'appareils, restent utiles et significatives.

Il est parfaitement valide qu'un document de présence contienne juste un seul service. C'est permis même si la présentité a en fait plusieurs services à sa disposition. Qu'il n'y ait pas plusieurs services dans le document signifie simplement que la présentité n'offre pas de choix à l'observateur. Dans ce cas, les caractéristiques de service sont moins importantes, mais peuvent aider à permettre à l'observateur de décider si il veut en fait communiquer.

### 3.3.2 Informations d'accès

Les informations d'accès pour un service donnent des instructions au receveur d'un document sur comment contacter correctement ce service.

Quand un service est accessible sur un réseau de communications, les informations d'accès incluent un URI qui peut être "touché" pour accéder au service. Cet URI est appelé l'URI de service. Cependant, certains services ne sont pas accessibles sur un réseau de communications (comme des communications en personne ou une lettre écrite) et à ce titre, peuvent ne pas utiliser un URI.

Même pour les services accessibles sur un réseau de communications, l'URI seul peut n'être pas suffisant. Par exemple, deux applications peuvent fonctionner dans un téléphone cellulaire, toutes deux accessibles par l'adresse de rattachement SIP de l'utilisateur. Cependant, une application est lancée quand la demande INVITE contient un corps d'un type particulier, et l'autre est lancée pour d'autres types de corps. Un autre exemple serait un service qui peut fournir une logique d'application complexe qui ne fonctionne correctement que si elle est contactée à partir d'un logiciel d'application qui lui correspond. Dans ce cas, même si les communications entre les instances utilisent un protocole standard (comme SIP) l'expérience de l'utilisateur ne sera correcte que si les applications sont satisfaites.

Lorsque l'URI n'est pas suffisant, des attributs supplémentaires du service peuvent être présents pour définir les instructions sur la façon dont le service est atteint. Ces attributs doivent être compris pour que le service soit utilisé. Si un observateur reçoit un document de présence qui contient des informations d'accès qu'il ne comprend pas, il devrait supprimer les informations de service.

Les informations d'accès sont une partie importante du service. Quand l'observateur prend une décision sur le service de la présentité auquel il souhaite accéder, l'observateur utilise les informations d'accès pour ce service. Pour cette raison, chaque service a un ensemble unique d'informations d'accès. Si ce n'était pas le cas, l'utilisateur n'aurait aucun moyen de choisir entre les services. Cela signifie que les informations d'accès représentent un identifiant unique pour le service. Cependant, un document de présence peut contenir de multiples occurrences d'un service particulier, dont chacun contient les mêmes

informations d'accès, mais différent par son identifiant d'occurrence. Plusieurs occurrences d'un service existent dans un document quand l'état du service est ambigu, comme expliqué au paragraphe 3.5.

Parce que les informations d'accès servent d'identifiant d'un service, elles servent aussi de moyen pour découvrir si une capacité de communications devrait être représentée comme un ou plusieurs services. Quelque chose ne peut être un service si il n'y a pas un moyen d'y accéder séparément d'un autre service. Par exemple, considérons une application de téléphone logiciel qui est capable d'audio et de vidéo. Il n'est pas possible de décrire ce téléphone logiciel comme deux services - une capable juste d'audio, et une capable juste de vidéo. C'est parce que il n'y a aucun moyen d'atteindre le service de vidéo seule ; par exemple, envoyer un INVITE SIP avec juste un flux vidéo ne suffit pas, car on peut toujours ajouter le flux audio ultérieurement et cela va fonctionner. Vidéo et audio, dans ce cas, représentent des capacités pour un seul service.

Les informations d'accès représentent une forme faible de contrat ; la présentité dit à l'observateur que, si l'observateur utilise les informations d'accès incluses dans le document de présence, l'observateur pourrait être connecté à un service décrit par les caractéristiques incluses dans le document de présence. Il est important de souligner que ceci n'est en aucun cas une garantie. Ce ne peut pas être une garantie pour deux raisons. D'abord, le service dans le document pourrait en fait modéliser un certain nombre de services utilisés réellement par l'utilisateur, et il se pourrait qu'il ne soit pas possible de connecter l'observateur à un service avec toutes les caractéristiques décrites dans le document de présence. Ensuite, les préférences de la présentité ont toujours la préséance. L'appelant pourrait demander à être connecté au service vidéo, mais il est permis de le connecter à un service différent si c'est le souhait de la présentité.

Ce contrat lâche donne aussi des directives sur le type d'URI qui convient idéalement le plus à l'URI de service. Un URN [RFC4234] peut être utilisé comme URI de service. Cependant, comme un URN pourrait être résolu en potentiellement tout nombre d'URI différents, les informations de caractéristiques, état, et informations relatives doivent avoir un sens pour tous les URI qui peuvent être résolus à partir de l'URN. Comme l'URN devient de plus en plus "vague" en termes de service qu'il identifie, le nombre d'attributs de présence qui peuvent être inclus diminue de façon correspondante.

L'URI tel [RFC3261] partage des propriétés similaires avec un URN, et les mêmes considérations s'appliquent. Si, par exemple, le numéro de téléphone existe dans ENUM [RFC3761] et si plusieurs services ENUM sont définis, incluant la voix et la messagerie, il est probable que très peu d'informations de caractéristiques peuvent être incluses dans ce service. Si, cependant, un URI tel a seulement un service ENUM défini, et si il se réfère à un service de téléphonie sur le réseau téléphonique public commuté (RTPC) il peut y en avoir plus à dire sur ses caractéristiques, état, et priorité relative.

Il est important de souligner qu'il peut y avoir une transposition surjective des informations d'accès à un service. C'est-à-dire qu'un service particulier peut être accessible par un nombre infini d'ensembles d'informations d'accès. Ceci est vrai même si les informations d'accès sont juste l'URI de service ; il est permis que plusieurs URI de service accèdent au même service. Dans tout document particulier, pour un service particulier, il va y avoir un seul URI de service. Cependant, il est permis et même utile de fournir des URI de service différents aux différents observateurs, ou de changer les URI de service fournis à un certain observateur au cours du temps. Le faire apporte de nombreux avantages, en fait. Cela peut permettre au receveur d'une tentative de communications de déterminer le contexte de cette tentative - que la tentative a été faite par suite de l'essai d'accéder à un service particulier dans un certain document de présence. Ce peut être utilisé comme une technique pour empêcher les communications non désirées, par exemple [RFC5039].

Il est aussi possible qu'un document de présence contienne un service qui n'a pas d'informations d'accès du tout. Dans ce cas, la présentité indique que le service existe, mais qu'elle choisit de ne pas offrir à l'observateur l'opportunité de s'y connecter. Un exemple serait de faire savoir à un observateur qu'un utilisateur a un service de téléphonie, et qu'il est occupé, mais afin d'éviter la réception d'un appel, aucune information d'accès n'est fournie.

Dans un système idéal, l'URI seul représenterait des informations d'accès suffisantes pour chaque service. Un URI est supposé fournir un contexte suffisant pour accéder aux ressources associées à l'URI, et donc en théorie il n'y a pas besoin de contexte supplémentaire. Cependant, parfois, des informations supplémentaires sont nécessaires. Comme les informations d'accès doivent être comprises afin que le service soit utilisé, les informations d'accès au delà de l'URI devraient être définies et utilisées avec parcimonie. Les extensions à PIDF qui définissent les attributs qui sont des informations d'accès devraient clairement invoquer ces attributs en tant que tels.

### 3.3.3 Informations relatives

Chaque service est aussi associé à une priorité, qui représente la préférence qu'a l'utilisateur pour l'usage d'un service plutôt que d'un autre. Ceci ne signifie pas que quand un observateur souhaite communiquer avec la présentité, il devrait toujours utiliser le service de la plus haute priorité. Si c'était le cas, il n'y aurait pas lieu d'inclure plusieurs services dans le document de présence. La priorité dit plutôt "Si vous, l'observateur, ne pouvez décider lequel utiliser, ou si cela n'a pas d'importance

pour vous, ceci est l'ordre dans lequel je voudrais que vous me contactiez. Cependant, je vous donne le choix." Les priorités sont relatives les unes aux autres, et n'ont pas de signification dans l'absolu. Si il y a deux services, et qu'elles ont des priorités respectivement de 1 et 0,5, ceci est identique à leur donner les priorités de 0,2 et 0,1, respectivement.

### 3.3.4 État

Chaque service a aussi un état. L'état représente généralement des informations dynamiques sur la disponibilité des communications en utilisant ce service. Cela fait un contraste avec les caractéristiques, qui décrivent des propriétés très statiques des divers services. La plus simple forme d'état est l'état de base, qui est un indicateur binaire de disponibilité pour les communications qui utilisent ce service. Il peut avoir les valeurs de "fermé" ou "ouvert". "Fermé" signifie que la communication avec le service va, selon toute probabilité, échouer, ne pas atteindre la partie prévue, ou ne pas résulter en communications telles que décrites par les caractéristiques du service. Par exemple, si un appel est transmis à une boîte vocale lorsque l'utilisateur est occupé ou indisponible, le service est marqué "fermé". De même, une présentité peut inclure le numéro de téléphone d'un hôtel comme URI de service. Après la sortie de l'hôtel, le téléphone va encore sonner, mais c'est la femme de chambre ou le prochain client qui va répondre. Donc, il va être déclaré "fermé" par cette présentité. Un autre exemple, serait si un utilisateur a un URI SIP comme URI de service qui pointe sur une application de téléphone logique SIP, et que le PC ferme, les appels à cet URI SIP vont retourner un code de réponse 480. Ce service serait aussi déclaré "fermé". "Ouvert" implique le contraire - que les communications sur ce service vont probablement réussir et atteindre la cible désirée.

Il est aussi possible d'avoir des informations d'état qui dépendent des caractéristiques de la session de communications qui est finalement établie. Par exemple, un attribut d'état peut être défini pour indiquer qu'un service de téléphone logique est disponible si la messagerie instantanée est utilisée, mais indisponible si l'audio est utilisé.

D'autres informations d'état peuvent indiquer plus de détails sur la raison de la disponibilité ou l'indisponibilité du service. Par exemple, un service de téléphonie pourrait avoir un état supplémentaire pour indiquer que l'utilisateur est occupé au téléphone, ou que l'utilisateur est engagé dans trois appels pour ce service.

Les services ont par nature des tas d'états dynamiques associés. Par exemple, considérons un service de téléphonie sans fil (c'est-à-dire, un téléphone cellulaire). Il y a de nombreux états dynamiques de ce service - que le téléphone soit ou non enregistré, qu'il soit ou non en itinérance, auprès de quel fournisseur d'accès il est connecté, la force du signal, combien d'appels il a, quel état ont ces appels, quel est la durée des appels de l'utilisateur, et ainsi de suite. Un autre exemple : considérons un service IM. Les états dans ce service incluent si l'utilisateur est enregistré, depuis quand, si il y a une conversation IM en cours, combien de conversations IM sont en cours, si l'utilisateur est en train d'entrer des données, à qui elles sont destinées, et ainsi de suite.

Cependant, il n'est pas approprié d'inclure tous ces états dynamiques dans un composant de données de service d'un document de présence. L'information n'est incluse que quand elle porte une aide à l'observateur pour décider d'initier des communications avec ce service, ou pour décider quand l'initier, si ce n'est pas maintenant. Par exemple, si un téléphone cellulaire a une force de signal assez forte ou juste une bonne force de signal pour réussir à l'essai de litmus. Savoir cela n'a probablement pas d'impact sur la décision d'utiliser ce service.

## 3.4 Appareil

Les appareils modèlent l'environnement physique de fonctionnement dans lequel les services s'exécutent. Les exemples d'appareils incluent des téléphones cellulaires, des PC, des tablettes, des assistants personnels, des téléphones fixes, des extensions de PBX d'entreprise, et des consoles de répartition d'opérateur.

La transposition des services en appareils est de plusieurs à plusieurs. Un seul service peut s'exécuter sur de multiples appareils. Considérons un service de téléphonie SIP. Deux téléphones SIP peuvent s'enregistrer sur une seule adresse de rattachement pour ce service. Par suite, le service SIP est associé aux deux appareils. De façon similaire, un seul appareil peut prendre en charge une multiplicité de services. Un téléphone cellulaire peut prendre en charge un service de téléphonie SIP, un service de SMS, et un service de MMS. De même, un PC peut prendre en charge un service de téléphonie SIP et un service SIP de visiophonie.

De plus, un seul appareil peut ne prendre en charge aucun service. Dans ce cas, l'appareil n'a pas d'informations de présence utiles par lui-même. Cependant, quand il est composé avec d'autres documents qui décrivent ce même appareil en relation avec un service, un document de présence plus riche peut être créé. Par exemple, considérons une étiquette d'identifiant radio fréquence (RFID) comme un appareil. Cet appareil n'exécute aucun service. Cependant, comme appareil, il a des

propriétés, comme une localisation, et il peut avoir une connectivité réseau avec laquelle il peut rapporter son état et ses caractéristiques. Si un visiophone devait rapporter qu'il fait fonctionner un service de vidéo, et qu'une de ses propriétés était qu'il est étiqueté avec cette RFID, un composeur pourrait combiner les deux documents, et utiliser la localisation de la RFID pour dire quelque chose sur la localisation de l'appareil de visiophonie.

Les appareils sont identifiés avec un identifiant d'appareil. Un identifiant d'appareil est un URI qui est un identifiant mondialement et temporellement unique pour l'appareil. En particulier, un identifiant d'appareil est un URN. L'URN doit être unique parmi tous les autres appareils pour une certaine présentité. Cependant, il est aussi très souhaitable qu'il soit persistant dans le temps, mondialement unique, et calculable d'une façon telle que différents systèmes puissent se référer à l'appareil en utilisant le même identifiant. Avec ces propriétés, différentes sources d'informations de présence fondées sur l'état de l'appareil peuvent être combinées. La dernière de ces trois propriétés - directement calculable - est particulièrement utile. Elle permet à un composeur de combiner des sources d'informations disparates sur un appareil, toutes reliées par un identifiant d'appareil commun que chaque source a utilisé de façon indépendante pour identifier l'appareil en question.

Malheureusement, du fait de la variété des différents appareils existants, il est difficile d'utiliser un seul schéma d'URN qui ait ces propriétés. Il est prévu que plusieurs schémas soient définis, avec des différences appropriées pour les différents types d'appareils. Pour les téléphones cellulaires, le numéro de série électronique (ESN, *Electronic Serial Number*) par exemple, est un bon identifiant. Pour les appareils IP, l'adresse MAC en est un autre. L'adresse MAC a la propriété d'être directement calculable, mais manque de persistance dans le temps (elle va changer si la carte d'interface sur l'appareil change). Dans tous les cas, aucun d'eux n'est associé aux schémas d'URN pour l'instant. En attendant, l'URN d'identifiant universellement unique (UUID, *Universally Unique Identifier*) [RFC4122] peut être utilisé. Pour les appareils qui ont une adresse MAC, les UUID de version 1 sont RECOMMANDÉS, car ils résultent en un identifiant fondé sur l'heure qui fait usage de l'adresse MAC. Pour les appareils sans adresse MAC, un UUID de version 4 est RECOMMANDÉ. C'est un identifiant purement aléatoire, qui fournit l'unicité. L'UUID pour un appareil va normalement être choisi au moment de la fabrication de l'appareil, et persiste ensuite dans l'appareil dans une mémoire flash ou quelque autre sorte de mémorisation non volatile. L'URN UUID a la propriété d'être temporellement unique au monde, mais à cause de son composant aléatoire, il n'est pas du tout directement calculable, et donc sans utilité comme identifiant de corrélation avec les autres sources de présence sur un réseau. Il est prévu que de futures spécifications seront développées pour fournir des identifiants d'appareil supérieurs supplémentaires.

Bien que chaque appareil soit identifié par un identifiant unique d'appareil, il peut y avoir plusieurs occurrences d'un certain appareil représentées dans un document. Chacune va partager le même identifiant d'appareil, mais va différer dans son identifiant d'occurrence. Plusieurs occurrences d'un appareil existent dans un document quand l'état de l'appareil est ambigu, comme expliqué au paragraphe 3.5.

Bien que le présent document ne rende pas obligatoire une approche particulière de mise en œuvre, l'identifiant d'appareil est des plus utiles quand tous les services sur l'appareil ont le moyen d'obtenir l'identifiant d'appareil et obtiennent la même valeur pour lui. Ceci plaiderait en faveur de son placement comme caractéristique du système d'exploitation. Les développeurs de systèmes d'exploitation intéressés par la mise en œuvre de la présente spécification sont encouragés à fournir des API qui permettent aux applications d'obtenir l'identifiant d'appareil. En l'absence de telles API, les applications qui rapportent des informations de présence sur leurs appareils auront à générer leurs propres identifiants d'appareils. Cela conduit à la possibilité que les applications puissent choisir différents identifiants d'appareils, en utilisant des algorithmes ou données différents. Dans le pire des cas, cela peut signifier que deux services qui fonctionnent sur le même appareil, ne paraissent pas le faire.

Comme les composants de données de services et de personne, les composants de données d'appareil ont généralement des caractéristiques statiques et généralement des états dynamiques. Les caractéristiques d'un appareil incluent ses dimensions physiques et ses capacités - la taille de son affichage, la vitesse de sa CPU, et la quantité de mémoire. Les informations d'état incluent les informations dynamiques sur l'appareil. Cela inclut si l'appareil est sous tension ou non, la quantité de batterie qui reste dans l'appareil, la localisation géographique de l'appareil, et ainsi de suite.

Les informations de caractéristiques et d'état rapportés sur un appareil sont pour les besoins des choix - pour permettre à l'utilisateur de choisir le service sur la base de la connaissance de ce qu'est l'appareil. Les caractéristiques et états de l'appareil ne peuvent pas être utilisés de façon fiable pour extraire des informations sur la nature du service qui va être reçu sur l'appareil. Par exemple, si les caractéristiques de l'appareil incluent la vitesse de la CPU, et si la vitesse est suffisante pour prendre en charge une compression vidéo de haute qualité, ceci ne peut pas être interprété pour signifier que la qualité de la vidéo sera assez bonne pour un service de vidéo sur cet appareil. D'autres contraintes sur le système peuvent réduire la quantité de CPU disponible pour ce service. Si on désire indiquer que la vidéo de qualité supérieure est disponible sur un appareil, cela devrait être fait en incluant des caractéristiques de service qui le disent explicitement. La vitesse de la CPU peut être utile pour aider l'observateur à faire la différence entre un appareil qui est un ordinateur personnel et un qui est un téléphone cellulaire, dans le cas où l'observateur souhaite appeler le téléphone cellulaire de l'utilisateur.

De même, si il y a un état dynamique d'appareil (comme si l'appareil est ou non sous tension) et cet état impacte l'état du service, ce qui est représenté en ajustant l'état du service. Sauf si un consommateur d'un document de présence a une connaissance a priori du contraire (noter que les agents de présence le font souvent) l'état d'un appareil n'a pas d'impact sur l'état du service.

Tout comme pour les services, il n'y a pas d'énumération des types d'appareils - PC, tablettes téléphones cellulaires, etc. L'appareil est plutôt défini par ses caractéristiques, à partir desquelles un observateur peut extrapoler si l'appareil est une tablette, un téléphone cellulaire, ou ce qu'on a.

Il est important de souligner que l'appareil est un \*modèle\* des systèmes physiques sous-jacents dans lesquels les services s'exécutent. Il n'y a rien qui dise que ce modèle ne peut pas être utilisé pour parler des systèmes où les services fonctionnent dans des systèmes virtuels, plutôt que réels. Par exemple, si un PC exécute une machine virtuelle et offre des services au sein de cette machine virtuelle, il est parfaitement acceptable d'utiliser ce modèle pour parler de ce PC comme étant composé de deux appareils séparés.

### 3.5 Ambiguïté de modélisation

L'ambiguïté est une réalité d'un système de présence, et elle est explicitement modélisée par la présente spécification. L'ambiguïté existe quand il y a plusieurs éléments d'information sur une personne, un appareil particulier, ou un service particulier. Cette ambiguïté augmente naturellement quand plusieurs éléments publient des informations sur la personne, un service particulier, ou un appareil particulier. Dans certains cas, un composeur peut résoudre l'ambiguïté de façon automatique, et combiner les données sur la personne, l'appareil, ou le service en une seule description cohérente. Dans d'autres cas, il ne peut pas, peut-être parce que le composeur n'a pas la capacité de le faire.

Cependant, dans de nombreux cas, la résolution de cette ambiguïté est laissée à l'observateur qui consomme le document. Ce consommateur pourrait être une application qui a plus d'informations que le composeur, et donc va être capable de faire un meilleur travail de résolution de l'ambiguïté. Ou, ce peut être présenté à l'utilisateur humain, et l'humain peut souvent résoudre l'ambiguïté. On ne sera pas surpris qu'un humain puisse souvent faire cela beaucoup mieux qu'un automate.

Pour modéliser l'ambiguïté, le modèle permet que chaque composant de service, d'appareil, ou de personne contienne plusieurs occurrences. Chaque occurrence a un identifiant unique, appelé identifiant d'occurrence. Cet identifiant est unique parmi tous les autres identifiants d'occurrence pour tout service, appareil, ou personne. C'est-à-dire, la portée de son unicité est tous les éléments de services, appareils, et personnes pour une présentité particulière. L'identifiant persiste idéalement dans le temps, car il fournit un moyen précieux pour régler les politiques de composition et d'autorisation. Même si il y a une seule occurrence d'un appareil, service, ou personne particulier, l'occurrence a un identifiant d'occurrence.

L'identifiant d'occurrence ne doit pas être confondu avec l'identifiant d'instance défini dans la spécification de SIP en sortie [RFC5626]. Une instance d'agent d'utilisateur est mieux modélisée comme un service, et bien sûr, un URI d'agent d'utilisateur mondialement utilisable (GRUU, *Globally Routable User Agent URI*) [RFC5627], qui est déduit de l'identifiant d'instance, représente un choix raisonnable pour un URI de service. Cependant, si l'état d'une telle instance d'UA ne pouvait pas être déterminée sans ambiguïté, un document de présence pourrait inclure deux occurrences ou plus du service modélisant cette instance d'UA. Dans ce cas, chaque occurrence a un unique identifiant d'occurrence, mais elle partage le même URI de service, et par conséquent, le même identifiant d'instance.

Quand plusieurs occurrences existent dans un document, il est important que certains des attributs de l'appareil, service, ou personne aident le receveur à résoudre l'ambiguïté. Pour les humains, le champ Note et l'horodatage sont des outils précieux. Pour un automate, presque tous les attributs de l'appareil, service, ou personne peuvent être utilisés pour résoudre l'ambiguïté. L'horodatage en particulier est très utile pour les humains et pour les automates. Comme décrit dans la [RFC3863], l'horodatage donne l'heure du plus récent changement du tuple. La présente spécification définit aussi l'horodatage pour les composants de personne et d'appareil, avec la même signification. En l'absence d'autres informations, la personne, l'appareil, ou le service qui a changé le plus récemment peut être utilisé comme la source de données la plus fiable. Cependant, un tel algorithme de résolution n'est en aucune façon exigé normativement.

### 3.6 Signification de l'absence

Il est clair que l'existence d'un attribut de présence dans un document dit quelque chose à un observateur sur la valeur de cet attribut de présence. Cependant, que dit l'absence d'un attribut de présence ? Le présent modèle de données suit les directives de la [RFC3840], qui est utilisée pour définir les capacités pour les agents d'utilisateur SIP. Dans cette

spécification, si une déclaration de capacité omet une étiquette de caractéristique particulière, cela signifie que l'agent ne fait pas de déclaration définitive sur si cette étiquette de caractéristique est prise en charge. La même chose est vraie ici - l'absence d'un attribut de présence dans un document signifie qu'un observateur ne peut pas faire de déclaration définitive sur la valeur de cet attribut de présence. Il peut être absent parce qu'il a été retiré de l'observateur, ou il peut être absent parce que cet attribut n'est pas pris en charge par le logiciel de la présentité. Aucune conclusion ne peut être tirée.

Parce que l'absence d'un attribut de présence ne porte aucune information que ce soit, les documents de présence réalisent leur valeur maximum quand ils ont autant d'attributs de présence que possible. À ce titre, il est RECOMMANDÉ qu'un document de présence contienne autant d'attributs de présence que la présentité veut et est capable de fournir à un observateur.

### 3.7 État ou caractéristiques

Le modèle de données essaye de séparer les informations d'état des caractéristiques, généralement en définissant l'état comme un état relativement dynamique sur une personne, un appareil, ou un service, tandis qu'une caractéristique est relativement statique. Cependant, cette distinction est souvent artificielle. Presque toutes les caractéristiques peuvent changer au cours du temps, et parfois les caractéristiques peuvent changer relativement rapidement. Il en résulte que la distinction entre états et caractéristiques est simplement conceptuelle pour faciliter la compréhension des différents types d'informations de présence. Rien dans un document de présence n'indique si un élément est une caractéristique ou état, et quand un attribut de présence est défini, il n'est pas besoin qu'il soit déclaré l'un ou l'autre. Les documents de présence permettent à tout attribut de présence, qu'il soit vu comme caractéristique ou comme état, de changer à tout moment.

Malheureusement, la spécification PIDF d'origine a une partie séparée du tuple pour décrire l'état, et l'état de base a été défini comme existant au sein de cette partie du tuple. La présente spécification ne change pas PIDF ; cependant, tous les futurs attributs de présence DOIVENT être définis comme enfants de l'élément <tuple> et non de l'élément <état>. De plus, les schémas définis ici ne contiennent pas d'élément <état> ni pour l'élément <personne> ni pour l'élément <appareil>.

### 3.8 Propriétés du document Présence

Le document de présence global a plusieurs propriétés importantes qui sont essentielles pour ce modèle.

D'abord, un document de présence a une signification concrète indépendante de la façon dont il est transporté ou d'où il est trouvé. La sémantique d'un document est la même sans considération de si le document est publié par un agent d'utilisateur de présence à son composeur, ou si il est distribué à partir d'un agent de présence aux observateurs. Il n'y a pas de comportement exigé ou implicite pour un receveur d'un document. Il y a plutôt une sémantique bien définie pour le document lui-même, et un receveur d'un document peut effectuer toutes les actions de son choix sur la base de cette sémantique.

Un corollaire de cette propriété est que les systèmes de présence sont composables à l'infini. Un agent d'utilisateur de présence peut publier un document pour son serveur de présence. Ce serveur de présence peut le composer avec d'autres documents, et placer le résultat dans une notification à un observateur. Cet observateur peut en fait être un autre agent de présence, combinant ce document avec les autres qu'il a reçus, et placer encore ces résultats dans une autre notification.

Encore un autre corollaire de cette propriété est que les comportements impliqués en réaction au document ne peuvent jamais être supposés. Par exemple, juste parce que un service indique qu'il prend en charge l'audio ne signifie pas qu'un observateur va offrir l'audio dans une tentative de communications à ce service. Si le faire est nécessaire pour atteindre le service, cela doit être indiqué explicitement par les informations d'accès.

Il est aussi important de comprendre que le rôle du document de présence est d'aider un utilisateur à faire un choix parmi un ensemble de services, et de plus, de savoir à l'avance avec autant de certitude que possible si une tentative de communications va réussir ou échouer. Le succès est une combinaison de nombreux facteurs : l'observateur comprend-il l'URI de service ? Peut-il agir sur toutes les informations d'accès ? Prend-il en charge un sous-ensemble des capacités associées au service ? Les informations de personne indiquent-elles que l'utilisateur va probablement répondre ? Toutes ces vérifications devraient idéalement être faites avant de tenter la communication.

Parce que le document de présence sert à aider un utilisateur à choisir et établir les communications, l'URI de présentité - comme indice de ce document - représente une forme de communications à "un numéro". En partant de cet URI, toutes les modalités des communications et leurs URI pour un utilisateur peuvent être découvertes, et ensuite utilisées pour invoquer un service de communications particulier. Plutôt que d'avoir à formuler un numéro de téléphone, adresse de messagerie

électronique, adresse IM, voix sur Internet, adresse de protocole (VoIP), et ainsi de suite, séparés, l'URI de présentité peut être fourni, et toutes les autres peuvent être apprises à partir de là.

#### 4. Motifs du modèle

Presence est défini dans la [RFC3856] comme la capacité, la volonté, ou le désir de communiquer à travers un ensemble d'appareils. Le cœur de cette définition est le transport des informations sur la capacité, volonté, ou désir de communications. Donc, le modèle de données de présence doit être construit autour du transport des informations qui réalisent ce but.

Le composant de données de personne est ciblé sur le transport de la volonté et du désir de communications. Il est utilisé pour représenter les informations sur les utilisateurs eux-mêmes qui affectent la volonté et le désir de communiquer. Que je sois dans une réunion, au téléphone - chacune de ces informations dit quelque chose sur ma volonté de communiquer, et donc a un sens pour être incluse dans un document de présence.

Le composant de service du modèle de données vise à transporter des informations sur la capacité à communiquer. La capacité à communiquer est définie par les services par lesquels un utilisateur est accessible. Donc, les inclure est essentiel.

Comment les appareils tiennent-ils dans ce modèle ? Pour de nombreux utilisateurs, les appareils représentent la capacité de communiquer, pas des services. Fréquemment, les utilisateurs tiennent des propos du genre, "Appelle moi sur mon portable" ou "Je suis à mon bureau". Ce sont des déclarations de préférence pour des communications qui utilisent un appareil spécifique, et non un service. Donc, on prévoit que les utilisateurs voudront représenter les appareils comme faisant partie des données de présence.

De plus, le concept d'appareil ajoute la capacité de corréler les services ensemble. L'appareil modélise les plates formes sous-jacentes qui prennent en charge tous les services du téléphone. Son état impacte donc tous les services. Par exemple, si un serveur de présence peut déterminer qu'un téléphone cellulaire est éteint, cela dit quelque chose sur les services qui fonctionnent sur cet appareil : ils sont tous indisponibles. Donc, si les services incluent des indicateurs sur les appareils sur lesquels ils fonctionnent, l'état de l'appareil peut être obtenu et donc utilisé pour calculer l'état des services sur l'appareil.

Le modèle de données essaye bien de séparer appareil, service, et personne comme des concepts différents. Une partie de cette différenciation est que de nombreux attributs vont être applicables à certains de ces concepts, mais pas aux autres. Par exemple, la localisation géographique est un attribut significatif de la personne (l'utilisateur a une localisation) et d'un appareil (l'appareil a une localisation) mais pas d'un service (par nature, les services n'ont pas de localisations). Sur cette base, les informations de localisation géographique ne devraient apparaître qu'au titre d'un appareil ou d'une personne, jamais d'un service. De plus, il est possible et significatif que les informations de localisation soient convoyées pour les appareils et les personnes, et que ces localisations soient différentes. Le fait que le système de présence puisse essayer de déterminer la localisation d'une personne en extrapolant à partir de la localisation d'un des appareils n'est pas pertinent du point de vue de la modélisation des données. La localisation de la personne et la localisation de l'appareil ne sont pas la même chose.

La [RFC4119] définit l'élément XML <geopriv> pour porter les informations de localisation, et indique qu'il est porté comme enfant de l'élément <tuple> dans un document PIDF. La [RFC4119] a été développée avant la présente spécification, et malheureusement, sa recommandation d'inclure les objets de localisation en dessous de <tuple> va à l'encontre de nos recommandations. À ce titre les mises en œuvre fondées sur la présente spécification DEVRAIENT inclure les objets de localisation <geopriv> au titre des composants de personne et/ou appareil du document, mais DEVRAIENT être prêts à recevoir des documents de présence avec cet objet comme enfant de <tuple>. Un objet de localisation <geopriv> va être inclus dans un composant de personne quand le document vise à porter la localisation de l'utilisateur, et dans un composant d'appareil quand il vise à porter la localisation de l'appareil.

#### 5. Codage

Les informations représentées conformément au modèle de données décrit ci-dessus doivent être transposées en un format du réseau pour le transport et la mémorisation. Le format de données des informations de présence (PIDF, *Presence Information Data Format*) [RFC3863] est utilisé pour la représentation des données de présence.

L'élément <presence> contient les informations de présence pour la présentité. L'attribut "entity" de cet élément contient

l'URI de présentité.

L'élément existant <tuple> dans le document PIDF est utilisé pour représenter le service. Ceci est cohérent avec l'intention originale de la RFC 2778 et de la RFC 3863, et réalise la rétro compatibilité avec les mises en œuvre développées avant que le modèle décrit ici soit achevé. L'élément <contact> dans l'élément <tuple> est utilisé pour coder l'URI de service. Les nouveaux attributs de présence, qu'ils représentent un état dynamique ou des caractéristiques statiques, apparaissent directement comme enfants de <tuple>. Cependant, les attributs définis avant la publication de la présente spécification qui étaient définis comme enfants de <status> (comme <basic>) restent des enfants de <status>, pour les besoins de la rétro compatibilité. Par conséquent, un attribut de présence qui décrit un service pourrait apparaître soit comme enfant de <status> soit directement comme enfant de <tuple>, mais jamais des deux.

L'attribut "id" de l'élément <tuple> porte l'occurrence de service. Chaque élément <tuple> avec le même URI <contact> représente une occurrence différente d'un service particulier.

La présente spécification introduit l'élément <personne>, qui peut apparaître comme enfant de <presence>. Il peut y avoir zéro, une ou plusieurs occurrences de cet élément par document. Chacune a un attribut "id" obligatoire, qui contient l'identifiant d'occurrence pour la personne. Chaque élément <personne> contient un nombre quelconque d'éléments qui indiquent les informations d'état et de caractéristiques. Ceci est suivi par zéro, un ou plusieurs éléments <note> facultatifs et un <horodatage> facultatif. Plusieurs éléments <note> vont apparaître pour porter la même note dans plusieurs langages.

La RFC 3863 définit un élément <note>, dont zéro, un ou plusieurs peuvent être présents comme enfants de <presence>. Comme ils se rapportent au modèle défini ici, ces éléments "note", si il en est de présents dans un document, s'appliquent à toutes les occurrences de personnes qui n'ont pas leurs propres éléments <note>. En d'autres termes, si un élément <personne> a un ou plusieurs éléments <note>, ce sont les éléments <note> pour cet élément <personne>. Si un élément <personne> n'a pas ses propres éléments <note>, les éléments <note> qui sont des enfants directs de <presence> sont les éléments <note> pour cette <personne>. Si il n'y a pas d'élément <note> sous l'élément <personne>, et si il n'y a pas d'élément <note> qui soit un enfant direct de <presence>, alors cet élément <personne> n'a pas d'élément <note>.

La présente spécification introduit aussi l'élément <appareil>, qui peut apparaître comme enfant de <presence>. Il peut y avoir zéro, une ou plusieurs occurrences de cet élément par document. L'élément <appareil> peut apparaître soit avant, soit après l'élément <personne> ; il n'y a pas de contrainte d'ordre. Chaque élément <appareil> a un attribut "id" obligatoire, qui contient l'identifiant d'occurrence pour l'appareil. Comme <personne>, <appareil> contient un nombre quelconque d'éléments qui indiquent les informations d'état et de caractéristiques. Ceci est suivi par <deviceID>, qui contient l'URN pour l'identifiant d'appareil pour cet appareil. Ceci est suivi par zéro, un ou plusieurs éléments <note> facultatifs et un <horodatage> facultatif. Plusieurs éléments <note> vont apparaître pour porter la même note dans plusieurs langages.

Un client qui reçoit un document PIDF contenant les éléments <appareil> et <personne> , mais ne les comprend pas (parce qu'il ne met pas en œuvre la présente spécification) va les ignorer. De plus, comme la sémantique de service comme elle est définie ici est alignée sur la signification d'un tuple comme défini dans les RFC 2778 et RFC 3863, les documents qui incorporent les concepts définis dans ce modèle sont compatibles avec les plus anciennes mises en œuvre.

Il est important de noter que la transposition du modèle de données de présence en un document PIDF est simplement un exercice de syntaxe.

Les documents de présence créés en accord avec le présent modèle DOIVENT être valides, avec l'exception suivante. Il est permis à un composeur de créer un document de présence qu'il ne peut pas valider pleinement mais qui par ailleurs est validé quand il est traité conformément aux règles de traitement laxistes permises par le schéma du composeur. Cependant, il n'est pas prévu que les entités qui reçoivent ces documents effectuent de validation de schéma ; elles vont plutôt simplement accéder aux informations provenant du document aux endroits où elles les attendent. Les mises en œuvre DEVRAIENT être prêtes à recevoir des documents qui ne sont pas valides, et extraire toutes les informations qui s'y trouvent et qu'elles peuvent analyser.

## 5.1 Schémas XML

Les schémas XML sont séparés en un schéma commun, appelé common-schema.xsd, qui contient les définitions de type commun, et le reste du modèle de données, data-model.xsd.

### 5.1.1 Schéma commun

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
    schemaLocation="http://www.w3.org/2001/xml.xsd"/>
  <xs:simpleType name="Timestamp_t">
    <xs:annotation>
      <xs:documentation>Timestamp type</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:dateTime"/>
  </xs:simpleType>
  <xs:simpleType name="deviceID_t">
    <xs:annotation>
      <xs:documentation>Identifiant d'appareil, un URN</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:anyURI"/>
  </xs:simpleType>
  <xs:complexType name="Note_t">
    <xs:annotation>
      <xs:documentation>Note type</xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute ref="xml:lang"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:attributeGroup name="fromUntil">
    <xs:attribute name="from" type="xs:dateTime"/>
    <xs:attribute name="until" type="xs:dateTime"/>
  </xs:attributeGroup>
  <xs:complexType name="empty"/>
</xs:schema>
```

### 5.1.2 Modèle de données

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:ietf:params:xml:ns:pidf:data-model"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:ietf:params:xml:ns:pidf:data-model"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:includer schemaLocation="common-schema.xsd"/>
  <xs:nom d'élément="deviceID" type="deviceID_t">
    <xs:annotation>
      <xs:documentation>Identifiant d'appareil, un URN</xs:documentation>
    </xs:annotation>
  </xs:élément>
  <xs:nom d'élément="appareil">
    <xs:annotation>
      <xs:documentation>Contient les information sur l'appareil</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:tout espace de noms="##autre" processContents="lax"
          minOccurs="0" maxOccurs="unbounded"/>
        <xs:référence d'élément="deviceID"/>
        <xs:nom d'élément="note" type="Note_t" minOccurs="0"
          maxOccurs="unbounded"/>
        <xs:nom d'élément="timestamp" type="Timestamp_t" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:nom d'élément>
</xs:schema>
```

```

</xs:sequence>
<xs:nom d'attribut="id" type="xs:ID" use="required"/>
</xs:complexType>
</xs:élément>
<xs:nom d'élément="personne">
<xs:annotation>
<xs:documentation>Contient des informations sur l'utilisateur humain</xs:documentation>
</xs:annotation>
<xs:complexType>
<xs:sequence>
<xs:tout espace de noms="##autre" processContents="lax"
minOccurs="0" maxOccurs="unbounded">
<xs:annotation>
<xs:documentation>Informations de caractéristiques et d'état</xs:documentation>
</xs:annotation>
</xs:any>
<xs:nom d'élément="note" type="Note_t" minOccurs="0"
maxOccurs="unbounded"/>
<xs:nom d'élément="horodatage" type="Timestamp_t" minOccurs="0"/>
</xs:sequence>
<xs:nom d'attribut="id" type="xs:ID" use="required"/>
</xs:complexType>
</xs:élément>
</xs:schema>

```

## 6. Extension du modèle de Presence

Lorsque de nouveaux attributs de présence sont ajoutés, une telle extension doit examiner les questions suivantes :

1. Le nouvel attribut est-il applicable à un composant de données de personne, de service, ou d'appareil ? Si il est applicable à plus d'un, quelle est sa signification dans chaque contexte ? Une extension devrait s'efforcer d'avoir chaque attribut défini de façon concise pour chaque domaine d'applicabilité, afin qu'une source puisse clairement déterminer à quel type de composant de données il devrait être appliqué.
2. Appartient-il à un espace de noms nouveau, ou à un déjà existant ? Généralement, les nouveaux attributs de présence définis dans la même spécification DEVRAIENT appartenir au même espace de noms. Les attributs de présence définis dans des spécifications séparées, mais produits de façon coordonnée par une administration centralisée, PEUVENT être placés dans le même espace de noms. Faire ainsi exige cependant que l'administration centralisée s'assure qu'il n'y a pas de collision de noms d'éléments sur ces spécifications. De plus, si une nouvelle extension a des éléments destinés à être placés comme enfants d'un autre élément à un point d'extensibilité défini par <tout espace de noms="##autre">, la nouvelle extension DOIT utiliser un espace de noms différent de celui de son élément parent.
3. L'extension elle-même exige-t-elle l'extensibilité ? Si oui, les points d'extension DOIVENT être définis dans le schéma, et cela DEVRAIT être fait en utilisant la construction <tout espace de noms="##autre">.

## 7. Exemple de document Presence

Dans cette section, on donne un exemple d'un système physique, on présente le modèle de ce système en utilisant les concepts décrits ici, et on montre ensuite le document de présence résultant. L'exemple utilise les attributs de présence définis dans les [RFC4480] et [RFC5196].

### 7.1 Client IM de base

Dans ce scénario, un fournisseur offre un service très similaire aux services de messagerie instantanée offerts aujourd'hui par des fournisseurs publics comme AOL, Yahoo!, et MSN. Dans ce service, chaque utilisateur a un "nom d'écran" qui identifie l'utilisateur dans le service. Un seul client, généralement une application de PC, se connecte à la fois au service. Quand le client se connecte, ce fait est rendu disponible aux autres observateurs de cet usager dans le système. L'usager a la

capacité d'établir une note textuelle qui décrit ce qu'il fait, et cette note est vue par les observateurs du système. L'utilisateur peut établir un des messages d'état (occupé, en réunion, etc.) qui sont des notes pré-définies que le système comprend. Si un utilisateur ne rentre rien sur son clavier pendant un certain temps, l'état de l'utilisateur passe à "repos" sur les écrans des divers observateurs du système. Le système indique aussi la durée pendant laquelle l'utilisateur a été inactif.

Chaque fois qu'un utilisateur est connecté au système, il est capable de recevoir des messages instantanés. Un utilisateur peut régler son état à "invisible", ce qui signifie qu'il apparaît comme non en ligne aux autres utilisateurs. Cependant, si un IM lui est envoyé, il va quand même être livré.

Ce système est modélisé en représentant chaque présentité dans le système avec trois composants de données : un composant de personne, un composant de service, et un composant d'appareil. Le composant de personne décrit l'état de l'utilisateur, incluant la note et les messages d'état pré-définis. Cela représente les informations sur l'utilisateur humain, de sorte qu'elles sont incluses dans le composant de personne. Le tuple de service représente le service IM. Aucune caractéristique n'est incluse. L'URI de service publié par le client est réglé à l'adresse de rattachement (AOR, *Address of Record*) du client. Le composant appareil est utilisé pour modéliser le PC. Le composant appareil inclut l'élément <user-input> [RFC4480], car l'inactivité se réfère à l'usage de l'appareil, pas du service.

Le document publié par le client ressemblerait à ceci :

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
  xmlns:dm="urn:ietf:params:xml:ns:pidf:data-model"
  xmlns:rp="urn:ietf:params:xml:ns:pidf:rp"
  xmlns:caps="urn:ietf:params:xml:ns:pidf:caps"
  entity="pres:presentity@example.com">
  <tuple id="sg89ae">
    <status>
      <basic>ouvert</basic>
    </status>
    <dm:deviceID>urn:uuid:698137d0-b395-11e0-aff2-0800200c9a66</dm:deviceID>
    <caps:servcaps>
      <caps:extensions>
        <caps:supported>
          <caps:pref/>
        </caps:supported>
      </caps:extensions>
      <caps:methods>
        <caps:supported>
          <caps:MESSAGE/>
          <caps:OPTIONS/>
        </caps:supported>
      </caps:methods>
    </caps:servcaps>
    <contact>sip:someone@example.com</contact>
  </tuple>
  <dm:person id="p1">
    <rp:activities>
      <rp:au téléphone/>
    </rp:activities>
  </dm:personne>
  <dm:identifiant d'appareil="pc122">
    <rp:user-input>idle</rp:user-input>
    <dm:deviceID>urn:uuid:698137d0-b395-11e0-aff2-0800200c9a66</dm:deviceID>
  </dm:appareil>
</presence>
```

Il vaut la peine de donner un peu plus de commentaires sur l'intérêt d'avoir un élément séparé d'appareil juste pour porter l'indication d'inactivité. L'indication d'inactivité est réellement intéressante comme indicateur que l'appareil est inactif. En rendant cela explicite, l'indicateur d'inactivité peut être utilisé par le serveur de présence pour affecter l'état des autres services fonctionnant sur le même appareil. Par exemple, disons qu'il y a une application VoIP fonctionnant sur le même appareil. Cette application rapporte son état de présence séparément, mais indique qu'elle fonctionne sur le même appareil.

Comme il a indiqué qu'il fonctionne sur le même appareil, le serveur de présence peut utiliser l'état du service pour préciser l'indicateur d'inactivité de l'appareil. Précisément, si l'utilisateur utilise son application VoIP, le serveur de présence sait que l'appareil est en service, même si l'application IM rapporte que l'appareil est au repos. Normalement, le repos est déterminé par l'absence d'entrées au clavier ou à la souris, dont aucune ne va être utilisée durant un appel VoIP.

Dans un cas plus simple, rapporter l'indicateur d'inactivité au titre de l'état de l'appareil permet que l'indicateur soit utilisé pour d'autres services sur le même appareil. En prenant là encore l'exemple de l'application VoIP sur le même appareil, si l'application VoIP ne rapporte aucune information sur l'appareil, et si un observateur ne reçoit pas d'informations sur le service IM, le document de présence envoyé à l'observateur peut inclure l'état de l'appareil. Par l'usage des identifiants d'appareils et des informations d'appareil, le serveur de présence peut corrélérer l'état de l'appareil comme rapporté par l'application IM avec le service VoIP, et les utiliser ensemble.

## 8. Considérations sur la sécurité

Les informations de présence décrites par le modèle défini ici sont très sensibles. C'est pour cette raison que le filtrage de confidentialité joue un rôle clé dans le traitement des données de présence. Le filtrage de confidentialité est l'acte d'appliquer des permissions à un document de présence pour supprimer des informations qu'un observateur n'est pas autorisé à voir. En termes plus généraux, le filtrage de confidentialité est une forme d'autorisation. Le filtrage de confidentialité peut aussi assurer qu'un observateur ne peut voir aucune donnée de présence pour une présentité, et bien sûr, il peut même assurer que la présentité ne sait pas qu'elle est bloquée. Les spécifications de présence SIP [RFC3856] exigent qu'un tel traitement d'autorisation soit effectué avant de divulguer les informations de présence. Des spécifications ont aussi été définies pour porter des politiques d'autorisation aux serveurs de présence [RFC5025].

L'intégrité des informations de présence est aussi critique. La modification des données de présence par un attaquant peut conduire, par exemple, à des détournements de communications. Les protocoles utilisés pour transporter les données de présence, comme SIP pour la présence, sont utilisés pour fournir les fonctions d'intégrité nécessaires.

## 9. Considérations d'internationalisation

La présente spécification définit un modèle de données qui contient surtout des jetons qui sont destinés à être consommés par des programmes, et pas directement par des humains. Les programmes sont supposés traduire ces jetons en chaînes de texte appropriées au langage conformément aux préférences de l'observateur.

Cependant, la présente spécification définit un élément `<note>` qui peut contenir du texte libre. Cet élément et d'autres définis par les extensions à PIDF qui peuvent contenir du texte libre DEVRAIT être étiqueté avec l'attribut `"xml:lang"` pour indiquer le langage et l'écriture. La présente spécification permet plusieurs occurrences de l'élément `<note>` afin que la présentité puisse porter la note dans plusieurs écritures et langages. Si aucun attribut `"xml:lang"` n'est fourni, la valeur par défaut est `"i-default"` [RFC2277].

Comme le modèle de présence est représenté en XML, il fournit une prise en charge native pour les informations de codage qui utilisent le jeu de caractères Unicode et ses représentations plus compactes incluant UTF-8. Les processeurs conformes à XML reconnaissent UTF-8 et UTF-16. Bien que XML inclue des dispositions pour identifier et utiliser d'autres jeux de caractères par l'utilisation d'un attribut `"encoding"` dans une déclaration `<?xml?>`, l'utilisation de UTF-8 est RECOMMANDÉE dans les environnements où existent des incompatibilités de prise en charge de codage.

## 10. Considérations relatives à l'IANA

Plusieurs considérations IANA sont associées à cette spécification.

### 10.1 Enregistrement de sous espace de noms d'URN

Ce paragraphe enregistre un nouvel espace de noms XML, selon les directives de la [RFC3688]

URI : l'URI pour cet espace de noms est `urn:ietf:params:xml:ns:pidf:data-model`.

Contact d'enregistrement : IETF, groupe de travail SIMPLE, [simple@ietf.org](mailto:simple@ietf.org) , Jonathan Rosenberg [jdrosen@jdrosen.net](mailto:jdrosen@jdrosen.net) .

XML :

DÉBUT

```
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
  "http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="content-type"
    content="text/html;charset=iso-8859-1"/>
  <title>Modèle de données pour Présence</title>
</head>
<body>
  <h1>Espace de noms pour le modèle de données de Présence</h1>
  <h2>urn:ietf:params:xml:ns:pidf:data-model</h2>
  <p>Voir <a href="http://www.rfc-editor.org/rfc/rfc4479.txt">
    RFC4479</a>.</p>
</body>
</html>
```

FIN

## 10.2 Enregistrements de schémas XML

Ce paragraphe enregistre deux schémas XML selon les procédures de la [RFC3688].

### 10.2.1 Schéma commun

URI : urn:ietf:params:xml:schema:pidf:common-schema.

Contact d'enregistrement : IETF, groupe de travail SIMPLE, [simple@ietf.org](mailto:simple@ietf.org) , Jonathan Rosenberg [jdrosen@jdrosen.net](mailto:jdrosen@jdrosen.net) .

Le XML pour ce schéma se trouve dans le contenu du paragraphe 5.1.1.

### 10.2.2 Modèle de données

URI : urn:ietf:params:xml:schema:pidf:data-model.

Contact d'enregistrement : IETF, groupe de travail SIMPLE, [simple@ietf.org](mailto:simple@ietf.org) , Jonathan Rosenberg [jdrosen@jdrosen.net](mailto:jdrosen@jdrosen.net) .

Le XML pour ce schéma se trouve dans le contenu du paragraphe 5.1.2.

## 11. Remerciements

Le présent document est réellement la compilation de nombreuses idées discutées sur une longue période. Ces idées ont été apportées par les nombreux participants au groupe de travail SIMPLE. Aki Niemi, Paul Kyzivat, Cullen Jennings, Ben Campbell, Robert Sparks, Dean Willis, Adam Roach, Hisham Khartabil, et Jon Peterson ont contribué à beaucoup des concepts qui sont décrits ici. Les exemples de documents de présence viennent de "Exemples de documents de présence" de Robert Sparks, et les idées sur la définition des services par des caractéristiques, plutôt que leur énumération, viennent du document de caractéristique de service de Adam Roach. Des remerciement particuliers à Steve Donovan pour les discussions sur les sujets discutés ici, et à Elwyn Davies pour sa relecture finale du document.

## 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. (MàJ par [RFC8174](#))
- [RFC2277] H. Alvestrand, "Politique de l'IETF en matière de [jeux de caractères et de langages](#)", BCP 18, janvier 1998.
- [RFC3688] M. Mealling, "[Registre XML de l'IETF](#)", BCP 81, janvier 2004.
- [RFC3841] J. Rosenberg, H. Schulzrinne, P. Kyzivat, "[Préférences de l'appelant](#) pour le protocole d'initialisation de session (SIP)", août 2004. (P.S.)
- [RFC3863] H. Sugano et autres, "[Format des données d'information de présence](#) (PIDF)", août 2004.
- [RFC4234] D. Crocker et P. Overell, "[BNF augmenté pour les spécifications de syntaxe](#) : ABNF", octobre 2005. (Remplace [RFC2234](#), remplacée par [RFC5234](#))
- [XML] Yergeau, F., Paoli, J., Sperberg-McQueen, C., Bray, T., et E. Maler, "[Extensible Markup Language](#) (XML) 1.0 (Third Edition)", W3C REC REC-xml-20040204, février 2004.
- [XML-1] Maloney, M., Beech, D., Thompson, H., et N. Mendelsohn, "XML Schema Part 1: Structures Second Edition", W3C REC REC-xmlschema-1-20041028, octobre 2004.
- [XML-2] Malhotra, A. et P. Biron, "XML Schema Part 2: Datatypes Second Edition", W3C REC REC-xmlschema-2-20041028, octobre 2004.

### 12.2 Références pour information

- [RFC2533] G. Klyne, "Syntaxe de description des [ensembles de caractéristiques des supports](#)", mars 1999. (MàJ par [RFC2738](#), [RFC2938](#)) (P.S.)
- [RFC2778] M. Day, J. Rosenberg et H. Sugano, "[Modèle pour Presence et la messagerie instantanée](#)", février 2000.
- [RFC3261] J. Rosenberg et autres, "SIP : [Protocole d'initialisation de session](#)", juin 2002. (Mise à jour par [3265](#), [3853](#), [4320](#), [4916](#), [5393](#), [6665](#), [8217](#), [8760](#))
- [RFC3761] P. Falstrom, M. Mealling, "Application de E.164 au système de découverte dynamique de délégation (DDDS) d'identifiants de ressource uniformes (URI) (ENUM)", avril 2004. (P.S.) (Obsolète, voir la [RFC6116](#))
- [RFC3840] J. Rosenberg, H. Schulzrinne et P. Kyzivat, "[Indication des capacités d'agent d'utilisateur](#) dans le protocole d'initialisation de session (SIP)", août 2004
- [RFC3856] J. Rosenberg, "[Paquetage d'événement Presence](#) pour le protocole d'initialisation de session (SIP)", août 2004.
- [RFC3859] J. Peterson, "[Profil commun pour les services de présence](#) (CPP)", août 2004. (P.S.)
- [RFC3966] H. Schulzrinne, "[L'URI tel pour les numéros de téléphone](#)", décembre 2004. (MàJ par [RFC5341](#)) (P.S.)
- [RFC4119] J. Peterson, "[Format d'objet de localisation GEOPRIV](#) fondé sur la présence", décembre 2005. (MàJ par [RFC5139](#)) (P.S.)
- [RFC4122] P. Leach et autres, "[Espace de noms d'URN](#) d'identifiant unique au monde (UUID)", juillet 2005. (P.S.)
- [RFC4480] H. Schulzrinne et autres, "[RPID : Extensions Rich Presence](#) au format de données d'information Presence (PIDF)", juillet 2006. (P.S.)

- [RFC4622] P. Saint-Andre, "Identifiants de ressource internationalisés (IRI) et identifiants de ressource uniformes (URI) pour le protocole de messagerie et de présence extensibles (XMPP)" juillet 2006. (*Obsolète, voir [RFC5122](#)*) (P.S.)
- [RFC5025] J. Rosenberg, "Règles d'autorisation de présence", décembre 2007. (P.S.)
- [RFC5039] J. Rosenberg, C. Jennings, "Le protocole d'initialisation de session (SIP) et les pourriels", janvier 2008. (*Information*)
- [RFC5196] M. Lonnfors, K. Kiss, "Extension de capacité d'agent d'utilisateur au format de données d'information de présence (PIDF) du protocole d'initialisation de session (SIP)", septembre 2008. (P.S.)
- [RFC5626] C. Jennings, R. Mahy, F. Audet, éd., "Gestion des connexions initiées par le client dans le protocole d'initialisation de session (SIP)", octobre 2009. (*MàJ [RFC3261](#), [RFC3327](#)*) (P. S.)
- [[RFC5627](#)] J. Rosenberg, "Obtention et utilisation des URI d'agent d'utilisateur mondialement acheminable (GRUU) dans le protocole d'initialisation de session (SIP)", octobre 2009. (P. S.)
- [RFC5724] E. Wilde, A. Vaha-Sipila, "Schéma d'UR pour le service de message court (SMS) du système mondial pour les communications avec les mobiles (GSM)", janvier 2010. (P. S.)

## Adresse de l'auteur

Jonathan Rosenberg  
Cisco Systems  
600 Lanidex Plaza  
Parsippany, NJ 07054  
USA

téléphone : +1 973 952-5000  
mél : [jdrosen@cisco.com](mailto:jdrosen@cisco.com)  
URI : <http://www.jdrosen.net>

## Déclaration complète de droits de reproduction

Copyright (C) The IETF Trust (2006).

Le présent document est soumis aux droits, licences et restrictions contenus dans le BCP 78, et à [www.rfc-editor.org](http://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 encloses ne viole 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 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](mailto:ietf-ipr@ietf.org).

**Remerciement**

Le financement de la fonction d'édition des RFC est fourni par l'activité de soutien administratif (IASA) de l'IETF.