Analyse et proposition d'identifiant permanent pour le SINP

Historique des versions du documents

Version	***Date***	***Auteur***	***Commentaire***
1.0	25/10/2013	Marie-Elise LECOQ	Version initiale

Historique des relectures

Date	***Relecteur***	***Commentaire***
19/09/2013	Régine Vignes-Lebbe	
09/10/2013	Julie Chataigner	

Acronyme

Acronyme	***Définition***
URI	Uniform Resource Identifier
PURL	Persistent Uniform Resource Locator
LSID	Life Science IDentifier
DOI	Digital Object Identifier
UUID	Universally Unique Identifier
GUID	Globally Unique IDentifier
HTTP	HyperText Transfer Protocol
URN	Uniform Resource Name
URL	Uniform Resource Locator
SINP	Système d'Information sur la Nature et les Paysages
DEE	Données Elémentaires d'Echange

Sommaire

I. Introduction

Au préalable de ce document, nous avons défini les différentes caractéristiques pouvant décrire un identifiant unique (*2013_05_06_GBIF_Synthese_Identifiant_Persistant.doc*). Tout d'abord, nous avons décrit la **persistance** de l'identifiant soit le comportement de celui-ci face à la modification ou la suppression des données qui lui sont liées. Ensuite, nous avons décrit l'**indépendance** de l'identifiant soit la génération de celui-ci à un niveau national ou à un niveau régional. Dans le contexte du SINP, il serait généré chez le fournisseur ou sur la plateforme. Puis, nous avons décrit l'**opacité** de l'identifiant soit la lisibilité et la compréhension par l'homme. Ainsi cela peut être une suite de chiffres et lettres compréhensible juste par la machine ou un code « 56MEL » soit une observation faite dans le Morbihan (56) par Marie-Elise Lecoq (MEL). Après, nous avons décrit la **résolution** de l'identifiant soit quels sont les services ou outils à mettre en place pour récupérer les informations liées à l'identifiant. Par la suite, nous avons décrit l'**unicité** de l'identifiant soit le lien unique qui lie un identifiant à une occurrence. Enfin, la **génération** de l'identifiant soit le mécanisme utilisé pour créer l'identifiant (script, algorithme...).

Dans ce document, nous commencerons par décrire les cinq types d'identifiants uniques existants suivants : URI, PURL, LSID, DOI et UUID. Ce sont les identifiants les plus utilisés que ce soit dans les systèmes d'informations (UUID), pour les publications (DOI) ou sur Internet (URI et PURL). Nous parlerons aussi des LSID car il s'agit d'un outil créé spécifiquement pour les problématiques d'identification dans le domaine de la science. Puis, nous les analyserons en évaluant s'ils répondent ou non aux caractéristiques présentées précédemment : persistance, indépendance, opacité, résolution, l'unicité et génération de l'identifiant et pour nous aider dans ce travail, nous définirons des poids (de 1 (faible) à 5 (élevé)) pour chaque point. Lors de notre conclusion, nous ferons une proposition de travail afin d'avancer dans la mise en place de l'identifiant des DEE du SINP.

II. Description

Tout d'abord, nous pouvons décrire le point caractérisant l'identifiant qui n'est pas lié au choix de la technologie : la structure de l'identifiant. Nous proposons d'utiliser le schéma suivant : Autorité, Contexte et Objet. Pour le SINP, voici la correspondance proposée :

- Autorité : plateforme
- Contexte : fournisseur ou base de données
- Objet : occurrence.

Afin d'aider à la compréhension de la suite du document, nous allons rapidement décrire cinq types d'identifiants existants.

URL

L'URL ou HTTP URI est un type d'identifiant basé sur une norme d'Internet mise en place par le World Wide Web. URL est un sous-ensemble d'URI. Il identifie l'**emplacement de la ressource et non la ressource elle-même**. Nous avons décidé de décrire le HTTP URI car il est dynamique grâce au protocole HTTP. Ainsi, l'identifiant sera facilement accessible via Internet. De plus, il est possible de développer facilement, quel que soit le langage utilisé, un outil permettant de publier et de gérer les identifiants.

D'après le document des bonnes pratiques de la publication d'URI proposé par la communauté du service JoinUp de la commission européenne (source : https://joinup.ec.europa.eu/community/semic/document/10-rules-persistent-uris), il est nécessaire de suivre les points suivants:

- un HTTP URI ne doit pas contenir d'extension de fichier (.csv, .doc ….). En effet ce qui est stable aujourd'hui peut ne pas l'être dans le futur. Par exemple, l'évolution des extensions de fichiers Word .doc en .docx. Ainsi, en ne le prenant pas en compte dans l'identifiant, les documents peuvent être mis à jour sans que cela ne l'impacte.
- Ne pas donner d'information sur la nature de la ressource (document, fichier...).
- Le HTTP URI doit pointer sur la dernière version de la ressource. La gestion de version ne se fait donc pas au niveau de l'identifiant mais en amont (base de données).

Le modèle d'un HTTP URI peut suivre le schéma suivant : {URI Root}/{Resource Path}/{ID} où :

- URI Root : information à propos du fournisseur.
- Resource Path : définition du périmètre de cette URI. Dans notre cas, pour une observation d'un jeu de données, cela serait data.
- ID : représentera l'identifiant de la ressource elle-même. Deux choix sont possibles :
    - Diviser l'ID en deux : une partie dédiée au jeu de données (DOI, code fournisseur, etc.) et une partie dédiée à l'occurrence (collection identifiant, UUID…),
    - Utiliser un identifiant persistant qui englobe les deux informations comme les LSID.

Le site iNaturalist utilise les HTTP URI comme identifiants persistants. Exemple : http://www.inaturalist.org/observations/422968

PURL

Le PURL utilise le standard HTTP et est basé sur le protocole URI. Il est promulgué par l'Online Computer Library Center (OCLC). C'est une des premières implémentations des identifiants persistants basés sur les spécifications URN.

PURL est un type d'URL qui, au lieu de donner accès directement à l'emplacement de la ressource, passe par un **service de résolution intermédiaire**. Il se base sur les redirections du standard HTTP. Il est possible de supprimer la ressource liée au PURL mais pas le PURL lui-même. Nous pouvons diviser les PURL en trois parties : le protocole, l'adresse de traduction et le nom.

La structure du PURL est la suivante :

PURL ::=<protocol><resolver address><name>

avec :

- "http://" comme protocole,
- "purl.<nom_autorité>.org" comme resolver address et
- "<jeu_donnée>?id=<identifiant>" comme nom.

De façon concrète, si notre site internet est **www.plateforme.fr** et qu'il est lié à **http://purl.plateforme.org** alors **http://purl.plateforme.org/jeu_donnee?id=4** est lié à **www.plateforme.fr/jeu_donnee?id=4**.

Afin d'éviter tout détournement, il serait nécessaire de n'utiliser que des PURL créés par le SINP ou les plateformes.

Plusieurs administrations proposent des outils de résolution de PURL comme le GBIF Norvège ou le gouvernement américain. Une des interfaces possibles des administrateurs PURL est : http://purl.oclc.org/docs/index.html.

LSID

Le LSID a été créé afin de gérer un protocole d'identifiants indépendants pour la communauté des sciences de la vie. Il utilise le **protocole URN** et non pas les protocoles liés à HTTP. Les URN **identifient les ressources** et non les emplacements.

La structure du LSID est :

URN:LDIS:<autority>:<context>:<object>.

Dans ce schéma, nous pouvons définir que "autority" représente la plateforme, "context" représente le

jeu de données ou le fournisseur de données et enfin "object" représente l'identifiant unique soit le numéro de collection ou un UUID.

Le site internet ZooBank et la plateforme WoRMS utilisent des LSID comme identifiants uniques. Cependant ce sont des checklists de noms (ZooBank est un registre de nomenclature) ou de taxons marins (WoRMS) et non des occurrences comme pour le SINP. L'analyse sur les LSID faite par le GBIF et le TDWG est qu'il est très difficile de mettre en place les LSID au niveau des occurrences ; ainsi le GBIF Espagne a créé des outils pour gérer les LSID pour les noms de taxons.
Exemple de LSID dans WoRMS : urn:lsid:marinespecies.org:taxname:360406 pour Robertgurneya dactylifer (Wilson C.B., 1931).

DOI

Le DOI est une **structure gérée et contrôlée** par la Fondation internationale DOI Foundation (source: http://www.doi.org/). Celui-ci est "ouvert à toutes les organisations ayant des intérêts dans la publication électronique" (source: http://en.wikipedia.org/wiki/Digital_object_identifier). Le système DOI fournit des outils pour gérer et identifier de façon pérenne les objets numériques.

La structure du DOI est standardisée (ISO 26324) et divisée en deux parties séparées par un slash. La première partie s'appelle le préfixe DOI et la seconde le suffixe DOI. Le nom du DOI est insensible à la casse mais surtout incorpore tous les caractères affichables par UNICODE.
Le préfixe DOI doit être composé de l'indicateur du répertoire suivi du numéro de l'agence de registration. Afin que l'identifiant soit reconnu comme DOI, l'indicateur du répertoire doit être 10. Le suffixe DOI est une chaîne de caractères choisie par l'agence d'enregistrement. Le format du DOI est donc :

10.<registration_agencies>/<code_suffix>

Pour générer un DOI, il est nécessaire de créer sa propre "Registration Agency" mais pour cela il faut être membre de la fondation (payant) ou se rapprocher d'une agence déjà existante (source: http://www.doi.org/registration_agencies.html). **Chaque agence gère les identifiants et les prix** pour générer un identifiant.

Le GBIF International préconise d'utiliser les DOI afin d'identifier les jeux de données. Cela permettra que chaque jeu de données soit cité de façon plus efficace et ainsi de mieux valoriser les données de celui-ci.

*NB : Il est possible d'utiliser le "Handle System" directement via le **CNRI de façon gratuite**. Il sera cependant nécessaire d'enregistrer le nom d'autorité qui, lui, aura un certain coût. Dans ce cas là, il sera possible de mettre en place la gestion des identifiants au niveau des plateformes et non au niveau national. Cependant, il est nécessaire d'avoir des compétences informatiques, plus précisément d'administration système sur les plateformes.*

UUID/GUID

Le GUID est un identifiant **généré aléatoirement via un algorithme**. Actuellement, nous sommes à la version 4 de celui-ci. Cela permet de créer des identifiants uniques et persistant dans le temps. La plupart du temps, les UUID/GUID sont couplés avec un protocole d'accès aux données type URN ou URI. L'UUID a la forme suivante : XXXXXXXX-YYYY-ZZZZ-AAAA-BBBBBBBBBBBB. Il est composé de 33 caractères hexadécimaux qui sont calculés aléatoirement en fonction de la date et de l'adresse MAC. Il existe 2^{122} différents identifiants. La probabilité est donc quasiment nulle de créer deux identifiants permanents identiques (source: http://en.wikipedia.org/wiki/Globally_unique_identifier).

Le projet E-Recolnat prévoit d'utiliser le système URI + UUID pour mettre en place les identifiants uniques. Le Muséum d'Histoire Naturelle d'Oslo a mis en place les identifiants uniques en utilisant les UUID et un service PURL pour les résoudre. Le service PURL est basé au GBIF Norvège.

III. Caractéristiques d'un Identifiant Persistant

Pour nous aider à faire notre analyse, nous nous sommes basés sur différents documents et sites internet dont "A Beginner's Guide to Persistant Identifiers" écrit par le GBIF (http://www.gbif.org/orc/?doc_id=2428&l=fr) et Wikipédia. Nous comparons les avantages de chaque type d'identifiant pour chacune des caractéristiques des identifiants présentées en introduction afin de déterminer quel est celui qui satisfait le mieux les besoins du SINP.

A. Persistance

La **persistance de l'identifiant est un des points les plus importants et les plus compliqués** à mettre en place. En effet, c'est cette caractéristique qui permet à l'identifiant d'être pérenne dans le temps. Il est nécessaire d'établir des **règles précises** sur le devenir de l'identifiant lorsque les données associées à celui-ci sont supprimées ou modifiées. Pouvons-nous mettre à nouveau en service l'identifiant lorsque les données sont supprimées ? Si oui, il sera donc possible de lier un identifiant à deux occurrences distinctes. Si non, il faudra mettre en place une gestion des identifiants qui ne sont plus liés à des données mais qui ne peuvent plus être utilisés. Que se passe-t-il lorsque les données de l'occurrence sont modifiées ? Il y a différentes possibilités :

- mettre en place un gestion des versions qui permet de garder un certain historique sur les identifiants donc par translation les occurrences.
- créer un nouvel identifiant lors de modifications. Il est possible de distinguer les modifications majeures et mineures et de ne recréer un identifiant que lorsque les modifications sont majeures.
- ne pas prendre en compte les modifications de l'occurrence au niveau de l'identifiant. Lors de modification, il suffira de définir un algorithme de parcours de type "liste chaînée" au niveau de la base de données ou bien de partir du principe que les données avant la modification sont supprimées par la modification.

Du point de vue du SINP, la persistance est importante, néanmoins la gestion des versions n'est pas envisagée dans cette première étape. En effet, celle-ci étant un travail compliqué, il est peut être judicieux de commencer par mettre en place un processus robuste permettant d'identifier les occurrences de taxon et dans un second temps, lorsque le processus sera opérationnel, d'affiner la gestion de l'identifiant en mettant en place les versions. Comme la persistance est un des points critiques, il est nécessaire que l'identifiant offre des réponses concrètes, opérationnelles et surtout pérennes.
Poids : 4

Dans la suite de cette partie, nous allons définir le comportement des cinq éléments face aux différents événements possibles sur la partie "Autorité", "Contexte" et "Objet". Sachant que nous partons du principe que le niveau le plus haut de l'identifiant "Autorité" représente la plateforme, puis le niveau intermédiaire représente le fournisseur ou le jeu de données et enfin le niveau le plus bas représente

l'occurrence. Le récapitulatif des situations possibles est le suivant :

- Au niveau de la plateforme, nous avons les modifications suivantes :
    - La modification du nom de la structure
    - Le changement de structure
    - La suppression de la structure
- Au niveau du contexte, nous avons les modifications suivantes :
    - un jeu de données est migré sur une autre plateforme
    - un jeu de données est renommé
    - un jeu de données est divisé en différents jeux de données distincts
    - différents jeux de données sont regroupés en un seul jeu de données.
- Au niveau de l'objet, nous avons les modifications suivantes :
    - Les champs des données sont modifiés
    - Les données en elles-mêmes sont modifiées
    - La ressource est supprimée

Nous partons du principe que toutes ces fonctionnalités sont possibles afin de prendre en compte l'ensemble des cas possibles.

URI

Plateforme

Dans le premier cas, il est nécessaire que la plateforme garde l'ancien DNS et redirige celui-ci vers le nouveau. Dans le second cas, la nouvelle structure devra prendre en charge le site internet qui génère les données par l'identifiant. Pour la suppression d'une plateforme, il n'est pas possible de garder les DNS supprimés ; cependant, le protocole URI met en place une gestion d'erreur pour prévenir l'utilisateur.

Jeu de données

Au niveau du contexte, la gestion des modifications peut se diviser en deux parties : la migration du jeu de données et les modifications apportées au jeu de données lui-même.

Pour la migration des jeux de données sur une autre plateforme, il est possible que la nouvelle plateforme maintienne les anciens identifiants mais pour cela, il est nécessaire que l'ancienne plateforme redirige le jeu de données vers la nouvelle plateforme.

Pour les modifications liées au jeu de données, il est nécessaire de créer une gestion des redirections entre les anciens jeux de données et les nouveaux.

Occurrence

Pour la modification des champs du jeu de données (nom de colonne de la table par exemple) ou des données elles-mêmes, les changements se font au niveau de la ressource donc il n'y a pas de changement au niveau de l'identifiant. Si la ressource est supprimée, il est nécessaire de définir dans le protocole URI que la donnée est passée à un statut "supprimé" et donc de renvoyer une page d'erreur.

PURL

Plateforme

Dans le premier cas, le service qui résout le système PURL redirige directement l'ancien DNS vers le nouveau DNS. Dans le second cas, la nouvelle structure devra prendre en charge le site internet qui génère les données par l'identifiant. Pour la suppression d'une plateforme, comme pour le premier point, le service qui gère les identifiants redirige l'utilisateur vers le nouveau DNS.

Jeu de données

Au niveau du contexte, la gestion des modifications peut se diviser en deux parties : la migration du jeu de données et les modifications apportées au jeu de données lui-même.
Pour la migration des jeux de données sur une autre plateforme, le service lié au PURL redirige l'utilisateur vers la nouvelle plateforme. Pour les modifications liées au jeu de données, le service PURL redirige vers le ou les nouveaux jeux de données.

Occurrence

Pour toutes les modifications ainsi que pour la suppression, les changements se font au niveau de la ressource donc il n'y a pas de changement au niveau de l'identifiant.

LSID

Plateforme

Dans le premier cas, il est nécessaire de mettre à jour l'annuaire du LSID ainsi que le service de résolution du LSID qui redirige l'utilisateur vers la nouvelle plateforme. Dans le second cas, la nouvelle structure prend le contrôle de la gestion des LSID de l'ancienne plateforme. Pour la suppression d'une plateforme, il n'est pas encore possible de le faire.

Jeu de données

Au niveau du contexte, la gestion des modifications peut se diviser en deux parties : la migration du jeu de données et les modifications apportées au jeu de données lui-même.
Pour la migration des jeux de données sur une autre plateforme, le service lié au LSID redirige l'utilisateur vers la nouvelle plateforme. Pour les modifications liées au jeu de données, le service LSID redirige vers le ou les nouveaux jeux de données.

Occurrence

Pour la modification des champs du jeu de données (nom de colonne de la table par exemple) ou des données elles-mêmes, il est nécessaire de mettre à jour les métadonnées du LSID car il y a une gestion des versions. Si la ressource est supprimée, il est nécessaire de modifier les métadonnées liées au LSID ; de plus, il est aussi possible de mettre en place une gestion des versions dans le cas où la ressource est remplacée.

DOI

Plateforme

Dans le premier cas, le service qui résout le système DOI redirige directement l'ancien DNS vers le nouveau DNS. Dans le second cas, la nouvelle structure devra prendre en charge le site internet qui génère les données par l'identifiant. Pour la suppression d'une plateforme, comme pour le premier point, le service qui gère les identifiants redirige l'utilisateur vers le nouveau DNS.

Jeu de données

Au niveau du contexte, la gestion des modifications peut se diviser en deux parties : la migration du jeux de données et les modifications apportées au jeu de données lui-même.
Pour la migration des jeux de données sur une autre plateforme, le service lié au DOI redirige l'utilisateur vers la nouvelle plateforme. Pour les modifications liées au jeu de données, le service DOI redirige vers le ou les nouveaux jeu de données.

Occurrence

Pour toutes les modifications ainsi que pour la suppression, les changements se font au niveau de la ressource donc il n'y a pas de changement au niveau de l'identifiant.

UUID/GUID

Le UUID/GUID est un code généré de façon unique et aléatoire. Il n'est donc lié ni aux plateformes, ni à une base de données, et est persistant dans le temps.

Cependant, il n'est pas possible d'accéder à la donnée ni à son emplacement. Aussi, il est nécessaire de le coupler avec une URL ou une PURL afin de pouvoir avoir accès aux données. Les URL étant des URI, elles répondent aux mêmes contraintes que ces derniers.

Conclusion

Plateforme

	URI HTTP	**PURL**	**LSID**	**DOI**	**UUID/GUID**
Renommer une structure	Historique de l'ancien DNS et redirection vers le nouveau	Le service redirige directement vers le nouveau DNS	Mise à jour de l'annuaire et les services de résolution	Redirection direct par le service de résolution	Pas dynamique
Changer de structure	La nouvelle structure prend en charge le DNS	La nouvelle structure prend en charge le DNS et le service	La nouvelle structure prend le contrôle de la gestion des LSID de l'ancienne	La nouvelle structure prend en charge le DNS	Pas dynamique

			plateforme		
Supprimer une structure	Pas possible	Le service redirige directement vers le nouveau DNS	Pas possible	Redirection direct par le service de résolution	Pas dynamique

Contexte

	URI HTTP	**PURL**	**LSID**	**DOI**	**UUID/GUID**
migrer un jeu de données	Redirection du jeu de données vers la nouvelle plateforme	Redirection du jeu de données vers la nouvelle plateforme	Redirection vers la nouvelle plateforme	Redirection vers la nouvelle plateforme	Pas dynamique
renommer un jeu de données	Redirection du jeu de données vers le nouveau	Redirection du jeu de données vers le nouveau	Redirection vers le nouveau jeu de données	Redirection vers le nouveau jeu de données	Pas dynamique
diviser un jeu de donnée	Redirection du jeu de données vers le nouveau	Redirection du jeu de données vers le nouveau	Redirection vers le nouveau jeu de données	Redirection vers le nouveau jeu de données	Pas dynamique
Fusionner des jeux de données	Redirection des jeux de données fusionnés vers le nouveau	Redirection du jeu de données vers le nouveau	Redirection vers le nouveau jeu de données	Redirection vers le nouveau jeu de données	Pas dynamique

Objet

	URI HTTP	**PURL**	**LSID**	**DOI**	**UUID/GUID**
Modification des champs	Identification de la localisation de la ressource et non la ressource elle-même	Identification de la location de la ressource et non la ressource elle-même	Mise à jour des métadonnées et de la version.	Identification de la location de la ressource et non la ressource elle-même	Les données ne sont pas liées au UUID car il est généré aléatoirement
Modification de la donnée	Identification de la location de la ressource et non la ressource elle-même	Identification de la location de la ressource et non la ressource elle-même	Mise à jour des métadonnées et de la version.	Identification de la location de la ressource et non la ressource elle-même	Les données ne sont pas liées au UUID car il est généré aléatoirement

Suppression de la donnée	Identification de la location de la ressource et non la ressource elle-même	Identification de la location de la ressource et non la ressource elle-même	Mise à jour des métadonnées et de la version.	Identification de la location de la ressource et non la ressource elle-même	Les données ne sont pas liées au UUID car il est généré aléatoirement

B. Indépendance

Pour dire qu'un identifiant est indépendant, il doit être mis en œuvre au niveau des systèmes d'information des données sources et non par un système central. Cependant, cela n'exclut pas de gérer un dictionnaire de l'existant à un niveau plus haut.

D'un point de vue SINP, l'intérêt de l'indépendance est important et obligatoire pour les occurrences et peut-être non obligatoire pour les jeux de données et les fournisseurs. En effet, que ce soit au niveau du fournisseur ou au niveau de la plateforme, l'identifiant de l'occurrence doit être généré de façon automatique et indépendante d'une plateforme à l'autre. Cependant, au niveau des jeux de données et des fournisseurs (s'ils ont un identifiant unique), l'intérêt de l'indépendance est plus limité. Il sera tout à fait possible de le définir à un niveau plus élevé (national par exemple).
Poids : 4

URI
Le protocole URI peut être administré et géré, de façon technique, au niveau de chaque plateforme. Ces plateformes pourront gérer le nom de domaine qui leur est appliqué. Cependant, il faudra mettre en place un annuaire au niveau national pour répertorier toutes les plateformes existantes.
Ensuite, chaque plateforme gère ses jeux de données et ses identifiants.

PURL
Le protocole PURL est indépendant car il utilise les noms de domaines. Ainsi chaque nom de domaine, qui peut représenter une des plateformes, gère ses identifiants. Cependant, afin de garder une trace des noms de domaine, il serait intéressant de créer un annuaire pour les lister tous.

LSID
Les services pour générer et décrire les LSID peuvent être mis en place et gérés au niveau de la plateforme. Mais comme l'URI, il est nécessaire de mettre en place des systèmes d'annuaire au niveau de l'autorité à un niveau plus élevé afin d'éviter les doublons. La gestion des jeux de données et des occurrences, en revanche, se fait au niveau de la plateforme.

DOI
Pour générer un identifiant DOI, il est nécessaire de mettre en place ce que nous appelons un "Handle

System infrastructure" qui gère la création et la vie de l'identifiant. Chaque plateforme SINP pourrait utiliser son système à condition d'être membre de la fondation. Cependant, cela est coûteux (cotisation pour être membre et infrastructure importante). Il serait aussi possible d'utiliser une infrastructure déjà existante avec un coût pour la création d'un identifiant. Dans ce cas là, le DOI ne répond plus du tout à la problématique de l'indépendance.

UUID/GUID
Un UUID ou GUID est créé par un algorithme qui génère une suite de 32 caractères hexadécimaux aléatoirement. Il est donc complétement indépendant.

C. Opacité

Initialement, l'identifiant unique persistant est destiné aux ordinateurs pour que ces derniers puissent communiquer entre eux de façon rapide. Comme ce n'est pas une information nécessaire à l'utilisateur, sa lisibilité n'est pas obligatoire et est même, dans certains cas, déconseillée comme pour les adresses IP par exemple. Pour être interprété, il est nécessaire d'avoir d'autres informations. Dans le cas d'un numéro de téléphone, la ressource n'est compréhensible que si nous avons le nom de la personne lié au numéro. Cela ajoute une certaine sécurité à l'accès aux ressources.

Dans le cas du SINP, les données liées à l'identifiant unique ne sont pas critiques comme peuvent l'être les informations liées à une adresse IP ou à un numéro de téléphone. L'**identifiant peut donc être semi-opaque** c'est-à-dire compréhensible par l'homme (des mots, des noms...) mais nécessitant l'aide d'un dictionnaire ou d'un annuaire pour gérer le tout (annuaire des institutions, des bases de données ...).
Poids : 2

URI - PURL - LSID
Par principe, URI, PURL et le LSID utilisent des noms de domaine, des noms de contexte et parfois des noms descriptifs décrivant les occurrences. Ils ne sont pas opaques.

DOI
A la différence des trois types précédents, le composant du DOI est très opaque. En effet, seul le service qui gère le DOI peut comprendre les parties autorité, contexte et objet.

UUID/GUID
Un UUID/GUID est une suite de chiffres générée de façon aléatoire. Il est totalement opaque car il ne représente rien.

D. Résolution

Afin qu'un utilisateur puisse accéder à la donnée, il est nécessaire de créer un service qui permette d'accéder aux informations grâce aux identifiants. Ce mécanisme s'appelle la **résolution de l'identifiant**. Cela peut être fait grâce à des services déjà existants (HTTP) ou à la mise en place d'un système de résolution spécifique à notre projet.

Pour le SINP, il serait plus intéressant d'utiliser un protocole déjà existant afin de ne pas ajouter des contraintes à la mise en place du SINP. Il est nécessaire alors de trouver un processus compatible avec le maximum de navigateurs (porte d'accès des utilisateurs aux plateformes SINP).
Poids : 5

URI
Comme expliqué précédemment, l'URI utilise le protocole HTTP ainsi nous pouvons mettre en place des web services utilisant les requêtes HTTP (GET, PUT, POST, DELETE). Pour cela, il est conseillé de créer une architecture REST afin d'accéder aux données en dissociant la partie cliente de la partie serveur, et en se basant sur les requêtes HTTP.

PURL
PURL utilise le protocole HTTP ; ainsi il peut mettre en place des web services utilisant les requêtes HTTP. Il est possible de mettre en place une API REST afin d'accéder aux données comme nous pouvons le voir sur la page Google Code du PURL (source : https://code.google.com/p/persistenturls/wiki/PURLFederationArchitecture).

LSID
La résolution du LSID se compose de trois parties qui peuvent être matérialisées par trois web services :

- le premier est dédié à l'autorité (voir la partie sur la structure). Le client utilise le service DNS pour localiser le point d'entrée de la partie "Autorité" du LSID.
- le deuxième est focalisé sur les données. Le client fait un appel, via le protocole SOAP, au serveur déterminé par le premier web service.
- le dernier s'occupe des métadonnées. Le client se connecte, via une des fonctions spécifiques du web service précédent, pour accéder aux métadonnées de l'identifiant demandé.

IBM propose un outil pour aider au développement de ces trois web services (source : http://www.ibm.com/developerworks/opensource/library/os-lsid/).

DOI
Pour nommer et générer des composants, il est possible d'utiliser les "Registration Agencies" existantes ou devenir membre de la fondation et créer la sienne. Cela consiste à mettre en place une infrastructure informatique nommée "Handle System" qui permet de générer et de récupérer des identifiants.

UUID/GUID
Comme expliqué dans la partie “Persistance”, l’UUID devra être couplé à un autre mécanisme pour fonctionner comme un identifiant unique et accessible. Pour cela, il est nécessaire de créer une URL. Ainsi, comme pour les URI, il est possible de mettre en place des web services utilisant les requêtes HTTP (GET, PUT, POST et DELETE). Dans l’URL, il est possible de retrouver les parties “plateforme” (nom de domaine) et “jeu de données”. Par exemple, cela pourrait avoir la forme http://www.nom_plateforme.fr/nom_jeudonnee/UUID.

E. Unicité

Nous générons un **identifiant unique afin de faciliter l'accès à la donnée associée**. Ainsi, si un utilisateur fait des recherches sur deux portails qui gèrent les identifiants uniques, il pourra facilement reconnaître les doublons via cet identifiant. Cependant, il est nécessaire d'avoir une gestion de celui-ci lors de la suppression (ce point est plus développé dans la partie "persistance").

Une des règles du SINP est qu'une donnée d'échange puisse être liée à plusieurs données sources. Cela pourra compliquer la mise en place de l'unicité de l'identifiant car même si nous décidons de gérer la création de l'identifiant au niveau de la plateforme, il faudra le disperser au niveau des fournisseurs, et donc des données sources pourront avoir le même identifiant unique. Si l'identifiant est généré au niveau des fournisseurs, quel identifiant sera gardé au niveau de la plateforme ? Faudra-t-il recréer un identifiant ? Ce n'est qu'un cas particulier mais qui remet en cause l'unicité de l'identifiant. Cependant, nous avons pris la décision de réfléchir en fonction du cas 1:1 et non de ces cas particuliers.
Poids : 3

URI - PURL
HTTP URI et URL ne peuvent être liés qu’à une seule localisation donc ils respectent le principe d’unicité.

LSID
LSID est basé sur le protocole URN qui ne peut être lié qu’à une seule ressource.

DOI
Les DOI sont gérés via des systèmes, “Handle System”, qui prônent dans leur architecture l’unicité des identifiants.

UUID/GUID
L'algorithme qui génère les UUID se base sur l’adresse MAC de l’ordinateur ou du serveur ainsi que sur l’heure. L’unicité garde une très haute probabilité. De plus, comme nous le lions à une URL, l’unicité est donc respectée.

F. Génération

Nous avons déjà analysé la partie résolution de l'identifiant, il est aussi important de définir la génération de l'identifiant. Ce sera le **mécanisme qui nous permettra de créer l'identifiant** pour chaque occurrence. Cela peut être fait via un organisme central ou via un algorithme diffusé à plus bas niveau.

Du point du vue du SINP, nous partons du principe que toutes les plateformes se ressembleront d'un point de vue technique (même structure) et donc que la génération de l'identifiant sera facile à mettre en place et pérenne.

Poids : 4

URI

La structure d'une URI pourra être : nom_domaine_plateforme/data/id_collection/id_occurrence.
Afin de générer automatiquement l'identifiant lors de l'intégration des données d'échange, il sera nécessaire d'écrire un programme simple pour récupérer le DNS, le code du jeu de données à intégrer d'une part et d'autre part, et récupérer dans les données à insérer le numéro d'occurrence. A la sortie de ce programme, nous aurons notre URI bien formatée que nous intégrerons dans la base de données.

PURL

Il est possible de générer des PURL sur le site purl.oclc.org. Pour cela, il est nécessaire de suivre le cheminement suivant :

1. Créer un utilisateur
2. Créer un nom de domaine : pour le SINP, ce sera le DNS de la plateforme
3. Définir le chemin pour accéder à l'information
4. Remplir l'URL cible : www.dns_plateforme.com
5. Définir la liste des identifiants : la première étape sera le code de la collection et la seconde sera l'identifiant de l'occurrence.

Vous pouvez retrouver la documentation d'aide à la création de PURL ici : http://purl.oclc.org/docs/help.html.

Il est possible de créer son propre service PURL afin de générer les identifiants. Nous conseillons de tester les outils proposés sur la page Google code (source : http://code.google.com/p/persistenturls/). Cependant, il semble que les derniers développements remontent à 2010.

LSID

Les web services pour résoudre les LSID sont aussi là pour générer et maintenir chaque partie du LSID soit l'autorité, le contexte et l'objet.

Le GBIF Espagne propose des outils en PHP et en Java (en cours de développement) pour générer et résoudre les LSID. Néanmoins cela n'est que pour la partie taxonomie et non pour la partie occurrence.

DOI

Comme pour la récupération d'une ressource liée à un identifiant, il est nécessaire d'avoir accès à un système "Handle System" pour générer l'identifiant.

UUID/GUID

La génération de l'UUID se fait à partir d'un algorithme qu'il est possible d'appliquer à n'importe quel langage. De plus, la plupart des bases de données (PostgreSQL, MySQL, Oracle, etc.) ont un type spécifique UUID. Il sera donc très facile de l'implémenter dans la base de données.

Remarque

L'INPN et le GBIF France vont avoir une phase de test afin de mettre en place la création de l'identifiant avec la technologie choisie.

IV. Récapitulatif

Avant de conclure, voici un tableau récapitulatif des éléments décrits dans ce document permettant de clarifier les avantages de chaque type d'identifiant.

Poids	Persistance 4	Indépendance 4	Opacité 2	Résolution 5	Unicité 3	Génération 4
HTTP URI	87.50%	100.00%	25.00%	100.00%	100.00%	75.00%
PURL	100.00%	100.00%	25.00%	100.00%	100.00%	60.00%
DOI	100.00%	25.00%	100.00%	50.00%	100.00%	50.00%
LSID	87.50%	100.00%	25.00%	0.00%	100.00%	75.00%
UUID	NA	100.00%	100.00%	50.00%	100.00%	100.00%

Le pourcentage est calculé en fonction des réponses données par la technique analysée. Par exemple, pour la persistance, nous avons vu qu'il y avait 10 cas possibles, donc on retire 10% à chaque fois que l'outil ne répond pas à la fonction attendue.

Remarques

URI HTTP :

- Il n'est pas possible de gérer la suppression ; en revanche il est possible de dire à l'utilisateur que la plateforme n'existe plus.
- Les identifiants ne sont pas opaques et il est nécessaire d'avoir l'annuaire des plateformes et des codes des collections.
- La génération de l'identifiant nécessitera un développement, cependant il sera facile à dupliquer dans les autres plateformes.

PURL

- A noter que pour le PURL, il est nécessaire d'avoir un système de résolution au niveau de chaque plateforme si on veut une réelle indépendance.
- Les identifiants ne sont pas opaques, cependant il est nécessaire d'avoir l'annuaire des plateformes et des codes des collections.
- Afin de mettre en place un service qui génère des identifiants, il est nécessaire de tester les serveurs PURL existants. Si cela ne répond pas à nos attentes, nous serons peut-être amenés à développer notre propre serveur qui pourra être déployé sur toutes les plateformes SINP.

DOI

- Il est possible de mettre en place un structure au niveau de chaque plateforme, cependant cela est

très coûteux (financièrement et techniquement).

- Il y a un choix possible pour résoudre un identifiant : utiliser une Registration Agency déjà existante ou créer son propre système. Dans le deuxième cas, cela ne répondrait pas aux attentes du SINP.
- La génération de l'identifiant sera facile à mettre en place si on utilise une agence d'enregistrement déjà existante, sinon il sera nécessaire d'avoir des compétences en administration réseau pour mettre en place le système.

LSID

- Les identifiants ne sont pas opaques, néanmoins il est nécessaire d'avoir l'annuaire des plateformes et des codes des collections.
- Il est nécessaire de mettre en place toute l'architecture pour générer et résoudre l'identifiant.
- Il existe des services qui résolvent les LSID comme sur le site du TDWG (source : http://lsid.tdwg.org/).

UUID

- L'identifiant est persistant dans le temps car il n'est lié à rien, étant généré aléatoirement.

Il est à noter que la partie "Génération" et la partie "Résolution" restent théoriques. Nous ne pourrons faire nos conclusions qu'après une phase de tests.

V. Conclusion

Comme nous l'avons vu, le LSID répond à un maximum de contraintes mais il est très difficile à mettre en place, surtout à l'échelle des occurrences.

Les URI HTTP ont des lacunes au niveau de la pérennité. Il est nécessaire de mettre en place un vrai système pour gérer les modifications et les suppressions de DNS. Il est possible de rajouter une contrainte au niveau des institutions pour qu'elles ne puissent pas modifier leurs noms au niveau technique et non institutionnel.

Les PURL répondent à tous les points car ils gèrent de façon intéressante les suppressions de DNS qui sont la lacune des URI HTTP. Cependant, ils sont liés à une organisation OCLC qui ajoute des contraintes.

Les DOI sont les éléments les plus stables, cependant, si nous installons un mécanisme au niveau du SINP, il faut compter de nombreux coûts (cotisation et matériel informatique). Dans le cas où nous utilisons une Registration Agency déjà existante, nous ne répondons plus au critère de l'indépendance. Ainsi les plateformes ne pourront plus gérer les informations à leurs niveaux.

Les UUID répondent à tous les critères, cependant ils ne sont pas dynamiques et ils ne sont pas résolubles par internet. Néanmoins, il est possible de les lier à une autre architecture pour pallier ce problème (URI, PURL, etc.).

Il est plus intéressant de proposer une solution "clé en main" qui permet aux utilisateurs non avertis de mettre en place leur infrastructure. Cela doit se faire sans empêcher les plateformes ayant les ressources nécessaires d'améliorer l'architecture tout en gardant les choix faits.

Nous proposons deux solutions viables :

- Mettre en place les UUID avec une architecture URI HTTP
- Mettre en place les UUID avec une architecture PURL

Pourquoi l'UUID ?
Ce mécanisme permet de **générer des identifiants uniques, pérennes** car ils sont aléatoires et indépendants. L'algorithme permettant de générer le code n'est pas lié à un langage de programmation. Enfin, cela permet d'avoir un standard pour toutes les occurrences.
C'est la solution utilisée par le Muséum d'Histoire Naturelle d'Oslo, par le GBIF et préconisé par le projet E-Recolnat.

Pour le rendre dynamique, nous avons deux choix : les URI HTTP ou les PURL.
Il est nécessaire de mettre en place des tests pour décider quel mécanisme utiliser. Nous savons que les

URI HTTP sont pérennes hormis dans le cas où le DNS est supprimé. Il est nécessaire de mettre en place un outil de résolution des identifiants, ce qui rajoute une complexité à la mise en place. Nous savons aussi que les PURL sont contraignants à maintenir. A plusieurs étapes, il est nécessaire d'avoir une intervention humaine donc d'avoir les compétences techniques à ce moment là.

Nous garderons la structure autorité-contexte-objet du LSID. Cela pourra prendre la forme suivante : www.nom_plateforme.org/data/nom_jeu_donnee/UUID.

Autorité

La création, la gestion et la maintenance des noms de domaines se feront à un niveau national, sauf si la plateforme est capable de gérer son infrastructure. Au niveau national, seul l'annuaire devra être impérativement mis à jour. La gestion des codes des jeux de données ou des fournisseurs pourra être gérée au niveau des plateformes car elles auront des DNS différents.

Contexte

Dans le SINP, Le contexte représentera les jeux de données ou les fournisseurs. Ils seront identifiés via un code. Cette partie sera gérée au niveau des plateformes. Nous proposons deux solutions : les DOIs ou un code. Si nous désirons garder une homogénéité au niveau du territoire, le code devra être défini au niveau national, sinon nous pouvons laisser libre choix aux plateformes tout en préconisant une solution.

Objet

L'objet représentera l'occurrence. C'est à ce niveau que nous mettrons en place l'UUID. Aujourd'hui, la plupart des SGBD ont pris en compte un champ spécifique UUID. Les fournisseurs de données n'auront pas de mal à ajouter cette information. Attention, l'UUID ne remplacera pas la clé primaire. Cela ne doit absolument pas être lié au type de technologie utilisée.