# Implantation de la procédure pour la vérification des liens brisés

## Problématique

Mise en contexte

Avec l’arrivée du projet OSDP, des demandes grandissantes en terme de qualité de métadonnées sont reçues par l’équipe de la PGF. L’une d’elles concerne la fonctionnalité des URL inscrites dans les ressources des jeux de données. De plus en plus de jeux de données sont ajoutées dans la PGF mais aucun processus n’est en place pour assurer le cycle d’opération et de maintenance des métadonnées à travers le temps. Ainsi, les URL peuvent être valides lors de l’ajout des métadonnées dans la FGP mais rien nous permet d’assurer aux utilisateurs que les URL restent valides après cela.

Besoin

Au niveau de l’équipe de données de la FGP, nous avons besoin d’un outil automatique pour vérifier la validité des URL définies dans la section ressources dans les jeux de données contenues dans le catalogue de la FGP. L’équipe des données n’ayant pas autorité sur le contenu des métadonnées, le besoin est principalement ciblé sur l’identification des métadonnées en problème et la notification des problèmes à l’autorité concernée.

Out of Scope

* Correction des erreurs détectées par la solution
* Les URL validées sont uniquement celles présentes dans les ressources. La solution ne valide pas les URL pouvant se trouver ailleurs dans les métadonnées (contact, description, ...)
* Aucune validation au niveau de la synchronisation avec le RCS ne sera faite.
* Validation de conformité des métadonnées envers le HNAP.

## Analyse des besoins

Objectif

Mettre en place une solution validant automatiquement les URL et identifiant les jeux de données comportant des URL non fonctionnelles. La solution en place comportera le module de validation préalablement présenté ainsi qu’on module de gestion des erreurs et de notification par courriel.

Les étapes suivantes montrent une vue haut niveau de la séquence d’opération

1. Lecture des métadonnées et extraction des URL (données, service Web, lien html, etc) à partir des métadonnées
2. Identification du type d’URL et validation de manière appropriée
3. Gestion des codes de retour et identification des problèmes au besoin
4. Mise à jour du registre des erreurs identifiées
5. Envoie des notifications à l’autorité concernée

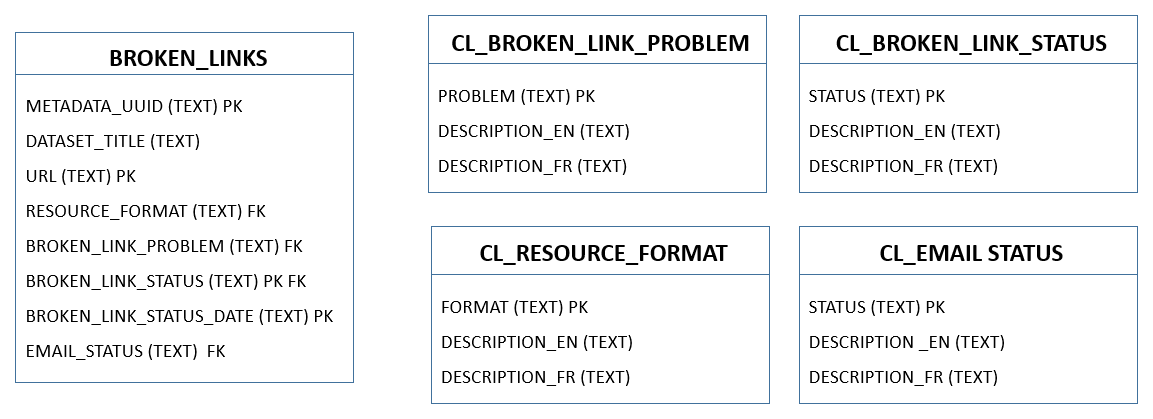
## Analyse détaillée

Sources de données

La solution développée aura comme source de données un point d’accès CSW diffusant des métadonnées sous le format HNAP. Pour l’instant, étant donné le firewall entre le GCNET et l’environnement AWS où est hébergé le FME Server de la FGP, la solution se connectera sur le catalogue externe de la FGP. Si un besoin se fait sentir pour effectuer une validation des métadonnées contenues dans le catalogue interne, il faudra déployer la solution vers un FME Server à l’intérieur du GCNET ou avoir un proxy à travers le firewall permettant l’accès au catalogue interne de la FGP à l’extérieur du GCNET.

Gestion des erreurs

Nous effectuerons la validation des liens sur une base hebdomadaire. Afin de faire une gestion adéquate des erreurs, une base de données sera mise en place permettant de faire un suivi rigoureux des erreurs détectées et des avis envoyés.



Lors d’une première détection, les liens non fonctionnels seront classés comme « observé ». Lorsque les liens observés sont à nouveau détectés par notre validation, ceux-ci seront maintenant classifiés comme « Confirmé ». C’est à ce moment que le problème sera officialisé avec le propriétaire et qu’un avis contenant des précisions sur le ou les liens brisés lui sera envoyé. Lorsque les liens confirmés ne sont plus présents dans les validations subséquentes, nous classerons ces cas comme « fixé ». De cette façon, nous éviterons de communiquer des problèmes qui proviendrait d’une interruption de réseau ou un problème de serveur (cas observé jamais confirmé) et qui se sont résolus d’eux même. Également, nous préviendrons l’envoi superflu de courriels pour des problèmes connus en cours de correction.

Envoie des notifications

Nous utiliserons le champs « metadata contact » dans les métadonnées afin d’obtenir le nom et le courriel de la personne qui recevra les avis touchant un jeu de données en particulier (lorsque les cas sont confirmés). Le courriel sera concis et ne contiendra seulement que l’information nécessaire à l’identification du jeu de données et du lien problématique (**UUID, Titre, URL**) et des informations sur le message d’erreur de nous avons obtenu via le logiciel FME qui pourront guider le responsable du jeu de données pour la correction de la situation. La **date** de notre détection sera également indiquée afin de souligner à quel moment le problème a été confirmé.

Choix technologique

Plusieurs technologies s’offrent à nous pour mettre en place la solution. Étant donné que la PGF dispose d’une infrastructure FME Server et que cette technologie nous offre toutes les capacités nécessaires autant au niveau de la validation que de la notification des erreurs, nous avons choisi cette technologie pour l’implantation de notre solution.

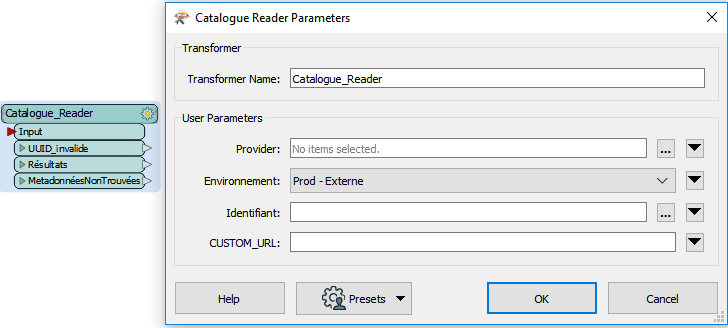
Au niveau de la gestion des erreurs et le suivi dans le temps des erreurs, nous préconisons une base de données SQL Lite. Le format de fichier offre des capacités similaires aux fonctionnalités de bases des SGDB tel que PostGreSQL ou Oracle. Pour les besoins actuels, ce type de base de données sera amplement suffisant.

Note : Étant donné l’utilisation de FME, il sera très facile de changer la BD SQL-Lite si désiré durant le cycle d’opération et de maintenance de la solution mise en place.

## Conception détaillée

Lecture de la données source

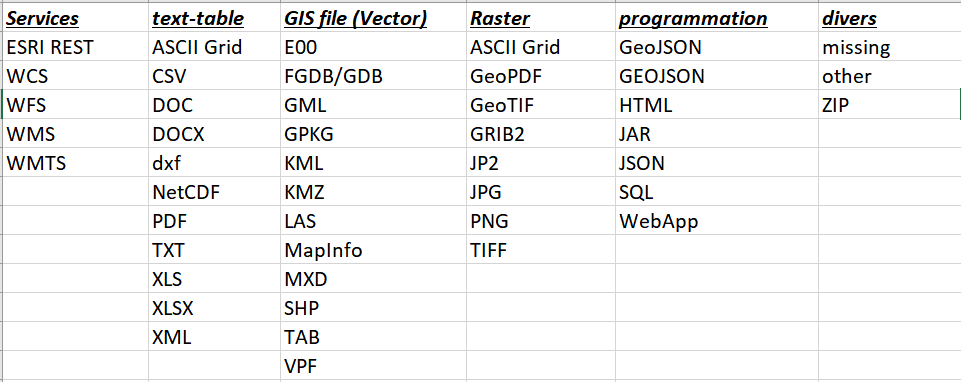
Utilisation du logiciel FME et d’un Custom Transformer (Catalogue\_Reader) afin de lire le contenu des métadonnées et de faire des recherches et requêtes au niveau des attributs. Ce programme a été développé pour obtenir des informations précises sur le contenu des catalogues de la PGF et sera réutilisé pour obtenir la liste des liens à valider pour chaque jeu de données.



Nous aurons ainsi accès au contenu du catalogue en format XML (HNAP) et nous pourrons utiliser les balises qui nous intéressent pour conserver les informations nécessaires à nos travaux. Des détails sur le format des ressources s’y retrouve alors ce sera la base pour aiguiller chaque lien vers le bon choix de validation.

Identification des URL et des formats

La section « ressources » des métadonnées sera celle dans laquelle nous obtiendrons le plus grand nombre d’informations utiles. Voici une liste des formats de fichiers regroupés par type au moment de notre analyse.



Validation des URL en fonction du type de ressources

La validation des liens sera exécutée en fonction du type de lien. Nous débuterons par les liens vers les services Web en traitant les services de type OGC WMS ainsi que les services de type ESRI REST. Une fois que nous aurons une façon de valider ceux-ci, nous pourrons mettre en place les composantes de base de notre procédure (base de donnée de suivi des problèmes, envoi de courriels, etc). Ceci nous permettra d’établir notre preuve de concept et de nous assurer que notre idée de départ est réalisable. L’ajout des formats par la suite se fera au fur et à mesure, probablement en attaquant un type de fichier à la fois pour des fins d’efficacité (vecteur, raster, texte, etc).

Pour la validation des liens WMS, plusieurs éléments peuvent être en cause lorsque le lien n’est pas fonctionnel :

* L’adresse peut comporter des erreurs ou avoir une mauvaise définition (?service=wms&query=GetCapabilities)
* Le retour obtenu lorsque l’adresse est contactée peut être nul ou avoir une structure inadéquate
* L’appel des couches doit être fait de la bonne façon avec des numéros de couches qui existe dans la définition du service.

Il y a donc beaucoup de choses à valider pour nous assurer de capter l’ensemble des problèmes et de pouvoir fournir des détails instructifs aux propriétaires des jeux de données afin de faciliter leur tâche de corriger ces problèmes. Nous aurons probablement à personnaliser la validation de la plupart des formats de données et c’est ce qui demandera beaucoup de temps et d’ingéniosité. C’est aussi la raison pour laquelle nous avons décidé d’opter pour un développement itératif et de mettre en place un système dès que nous serons confiant avec la validation d’un format en particulier.