Analyse de la gestion des droits dans GeoNature

Point de Camille M. - 25 septembre 2019

```
Pour mémoire, je repose aussi l'historique et le contexte (qui me fait
tourner la tête) :

Dans GeoNature V1, on avait 3 profils avec des droits fixes (rédacteur,
référent, administrateur).
Lors de la refonte en V2, on a souhaité pouvoir définir des droits bien plus
souples, génériques, adaptables à chaque instance et par module.

C'est là qu'on a posé les bases du CRUVED sur des portées.
Le détail des besoins et de sa modélisation est disponible ici :
https://github.com/PnX-SI/GeoNature/issues/238

Pour gérer cela de manière générique, une notion de tags a été intégrée au
niveau de UsersHub :  https://github.com/PnX-SI/UsersHub/issues/28

Après quelques usages, on s'est rendu compte que l'approche était trop
générique mais qu'en même temps elle intégrait des concepts de permissions
spécifiques à GeoNature dans UsersHub. On a donc travaillé à la
réintégration du CRUVED dans GeoNature. Détails ici :
https://github.com/PnX-SI/GeoNature/issues/517

C'est le fonctionnement qui est actuellement en place (documenté sur
http://docs.geonature.fr/admin-manual.html#gestion-des-droits) et qui devait
permettre d'implémenter d'autres types de droits (par précision des données,
taxonomiques, géographiques...).
Il semble avoir des limites.
Par ailleurs, Jean-Brieuc a de nouveau remonté le fait quel'on ne peut pas
associer des JDD à des groupes :
https://github.com/PnX-SI/GeoNature/issues/399#issuecomment-534611221

Et il y a encore quelques autres tickets sur le sujet, le CRUVED dans le
module Synthèse, les empilements de permissions etc... :
https://github.com/PnX-SI/GeoNature/search?q=CRUVED&type=Issues
```

Ressources

- Best Practices for Application Security (Identity and Access Management)
- Keycloak - open source Identity and Access Management solution

- Authentik : alternative en Python à Keycloak.
- Présentation de ce mémo : 2019-10-16 - GeoNature et permissions

Solution OAuth existante

- Keycloak - open source Identity and Access Management solution
- Du coup, il y aurait aussi https://goauthentik.io/ Authentik comme alternative à Keycloak, c'est en Python. Selon cette discussion https://www.reddit.com/r/selfhosted/comments/ub7dvb/authentik_or_keycloak/ Authentik a l'air moins orienté sécurité maximum par rapport à Keycloak. Mais je pense que ce n'est pas forcément un problème dans notre cadre d'utilisation...

Dictionnaire

- **Authentification** : prend généralement la forme d'un couple *identifiant/mot de passe* et permet de créer une session entre l'utilisateur et le système.
- **Autorisation** : procédure au cours de laquelle le droit de l'utilisateur à accéder à une ressource est vérifié.
- **Identification** : procédure au cours de laquelle l'utilisateur fournit un *identifiant*, représentation de son identités.
- **Permission** : permet à un utilisateur d'accéder aux ressources des applications du système.
- **Ressource** : élément, fonction d'une application.
- **Système** : ensembles des applications et ressources dont l'accès dépend du GIA.
- **Utilisateur** : correspond à une personne utilisant une application.

UsersHub

UsersHub se positionne en tant que solution de **Gestion des Identités et des Accès** (GIA) (*Identity and Access Management (IAM) solution*).

Les buts de ce type d'outils sont :

- L'*identification* : procédure au cours de laquelle l'utilisateur fournit un *identifiant*, représentation de son identités.
- L'*authentification* : prend généralement la forme d'un couple *identifiant/mot de passe* et permet de créer une session entre l'utilisateur et le système.
- L'*autorisation* : procédure au cours de laquelle le droit de l'utilisateur à accéder à une ressource est vérifié.
- La *gestion de l'utilisateur* : consultation, ajout, modification, suppression des utilisateurs, des groupes et des rôles.
- être un *annuaire central* d'utilisateurs

Avantages :

- financier/temps : éviter de développer dans chaque appli une nouvelle GIA.

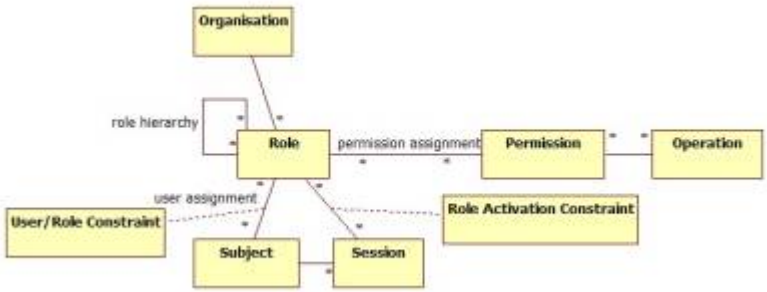
- souplesse : une modification d'identité est rapidement effectuée sur l'ensemble des applications du système.
- sécurité : secondaire pour GeoNature (d...

En lien avec l'authentification :

- *Authentification unique* [*Single-Sign-On* (SSO)] : permet à l'utilisateur de ne procéder qu'à une seule authentification pour accéder à plusieurs applications informatiques.
- *Json Web Tocken* (JWT) : plutôt que d'utiliser une session classique web à base de cookie (ne fonctionnant pas avec les applis mobiles), JWT permet d'échanger des jetons de manière sécurisé entre l'utilisateur et le système. Un des avantages et que cela permet d'avoir un mécanisme de session identique pour les applis web et mobile. Des informations concernant l'utilisateur peuvent transiter dans le JWT. Le standard prévoie certains informations classiques (nom, prénom, email...) et permet d'en ajouter des personnalisées (permissions par exemple).

En lien avec l'autorisation (ce qui nous intéresse ici) :

- *Contrôle d'accès basé sur les rôles* [Role-Based Access Control (RBAC)] : modèle de contrôle d'accès à un système d'information dans lequel chaque décision d'accès est basée sur le rôle auquel l'utilisateur est associé.



    - Dans ce modèle, toutes les *permissions* sont attribuées à un utilisateur via les rôles.

La notion de *permission* :

- Dans GeoNature correspond aux lignes de la table `gn_permissions.t_actions`, c'est à dire au CRUVED : Créer (C), Lire (R), Mettre à jour (U), Valider (V), Exporter (E), Supprimer (D).
- Dans les autres systèmes, les permissions sont libres et correspondent plutôt à une action du type "Voir les observations". Chaque action est associé à un code utilisé dans les applications pour déterminer si l'utilisateur peut y accéder. Nous pouvons découper une permission en 2 parties, un verbe et un complément. Le verbe correspond à l'*action* et le complément à la *ressource* d'une *application*.

Différences de GeoNature vis à vis de la notion de permission classique :

- Dans GeoNature, une permission c'est l'association d'un utilisateur, d'une action (CRUVED), d'un filtre, d'un module et d'un objet.
    - L'"objet" par défaut d'id 1 correspond à tous les objets d'un module. Les "objets" correspondent aujourd'hui aux sous-modules d'un module. Ex. le module "Admin" contient les sous-modules "Backoffice des permissions" et "Backoffice des nomenclatures".
    - un "filtre" permet limiter la *ressource* sur laquelle l'action va avoir lieu. Ex. Lire les observations sur la commune Gap.

Différences de UsersHub vis à vis d'un GIA classique :

- la gestion des permissions n'y est pas inclue ⇒ déportée dans des applications "chapeaux" (GeoNature, GeoTreck (?)).
- la notion de "role" du RBAC correspond à la notion de "profil".
- la notion d'application rentre en jeu alors qu'elle n'existe pas habituellement.
- l'association entre un utilisateur et son/ses profils se fait aussi vis à vis d'une application.
- pas de nécessité d'avoir une sécurité extrêmement poussée pour répondre a des questions tel quel : quel utilisateur a eu accès à quelle ressource à quel instant ?

Questions :

- Pourquoi les profils sont ils liés aux applications dans le schéma "utilisateurs" ?

Notes du COTECH GeoNature du 16 octobre 2019

Lien vers la présentation : 2019-10-16 - GeoNature et permissions

UsersHub et schéma "utilisateurs"

Concernant UsersHub et le schéma en base de données "utilisateurs" :

- UsersHub est une solution de Gestion des Identités et Accès (GIA) mais avec des spécificités :
  - gestion des permissions déportée dans les appli "chapeau". Ex. : GeoNature, GeoTrek.
  - notion d'application permettant d'indiquer les appli auxquelles un utilisateur peut se connecter
- Dans les GIA, le système de contrôle d'accès basé sur les rôles (RBAC) est le plus utilisé
- La notion de **rôle** du RBAC des GIA classiques est ici subdivisée en 3 notions : *profil*, *groupe* et *liste*.
  - Dans GeoTrek, les "*profils*" sont utilisés pour attribuer des permissions
  - Dans GeoNature :
    - les "*groupes*" sont utilisés pour attribuer des permissions,
    - la notion de *profil* liée à celle d'*application* sert juste à déterminer si l'*utilisateur* peut se connecter.
  - A noter que :
    - dans la base de données une table "`t_roles`" contient à la fois les informations des *utilisateurs* et *groupes*
    - la notion de *groupe* peut être hiérarchisée comme l'est la notion de *rôle* dans un GIA classique
- GeoNature est dépendant du schéma utilisateur de UsersHub pour fonctionner. UsersHub permet d'administrer le schéma "utilisateur",
- Il est envisageable de connecter le schéma "utilisateur" de GeoNature à un autre GIA mais cela nécessite de réaliser un développement sur mesure pour maintenir la synchro avec le schéma utilisateur de la base GeoNature.

Permissions GeoNature et besoins SINP Régionaux

Concernant la gestion des permissions dans GeoNature vis à vis des besoins des SINP régionaux

(PACA et AURA) :

- Il va être nécessaire de revoir entièrement l'interface de gestion des permissions qui est trop générique. Les administrateurs peuvent associer aux groupes et utilisateurs des permissions qui ne sont pas implémentées et n'ont aucun effet sur les interfaces.
- Les données et méta-données des SINP seront traitées en amont pour respecter les règles du standard SINP concernant les données sensibles et les niveaux de diffusions des jeux de données de type "privés".
- L'interface du module Synthèse sera modifiée en priorité pour respecter les règles du SINP (les autres modules pourront adaptés aussi par la suite) :
  - Licence fermé SINP : dans le cadre de la diffusion des données sensibles.
  - Formulaire d'accès aux données d'observation sur les espèces (SINP) : exemple de formulaire.
  - Ginco - Floutage des données
- Une interface spécifique de demande d'accès aux données géo-confidentielles (observations des jeux de données privés avec niveau de diffusion et observations sensibles) sera développée sur le modèle des demandes d'inscription à GeoNature. L'activation de cette interface sera lié à un paramètre de config. Par défaut, elle sera désactivée.
- Une attention particulière sera portée au maintient des performances d'affichage des observations dans GeoNature durant l'implémentation de ses fonctionnalités.

Modifications envisagées dans la base de données GeoNature

Les permissions permettant d'accéder aux données géo-confidentielles nécessites des modifications dans la base de données. Après discussion, il est retenu à priori ce qui suit :

- Schéma `gn_permissions` :
  - Ajout d'une table permettant de stocker les demandes d'accès et les informations liées à la motivation de cette demande (description du projet, type d'étude, durée...)
  - Modification de la table `cor_role_action_filter_module_object`
    - ajout d'un champ `group` permettant de cumuler des filtres pour une même permission (Ex. : Accès aux obs sensibles pour le taxon X et la commune Y).
    - ajout d'un champ `end_date` permettant d'indiquer une date de fin à la permission. Si NULL pas de fin prévu à la permission.
- Schéma `gn_synthese` :
  - Ajout d'un champ de type geom à la table `synthese` pour y stocker la géométrie floutée correspondant aux règles liées aux niveaux de diffusion et de sensibilités.
- Afin de facilité la création de l'interface de gestion des permissions :
  - une permission se compose d'un "verbe" et d'un "complément". Les verbes sont aujourd'hui stockées dans la table `t_actions` mais nous savons pas où placer le "complément" :
    - la table `t_filters` est une mauvais endroit car elle contient des infos qui joutent sur la clause "WHERE" d'une requête SQL et restreigne donc l'effet d'une permission.
    - Il a été évoqué d'ajouter de nouvelles actions plus précises (verbe + complément) à la table `t_actions` mais cela complexifie la gestion des permissions au niveau des routes.
    - Finalement, la table `t_objects` est retenue. Elle permettra de stocker les "ressources" des modules sur lesquelles une action agit. Les "sous-modules" actuellement stockés dans cette table peuvent être considéré comme des

ressources du module "Admin" de GeoNature.
    - Afin d'être certain des permissions implémentées dans un module, nous créerons une nouvelle table (qui sera renseigné par les développeurs) pour indiquer les actions possibles vis à vis d'un objet pour un module donnée et les types de filtres applicables : `cor_module_action_object_filter`. Cela implique la suppression des tables : `cor_object_module`, `cor_filter_type_module`. La proposition de nouvelle table `cor_action_module` est donc inutile.

Échange de données entre instances de GeoNature

- Échange de données entre instances de GeoNature :
    - travail en cours pour pouvoir synchroniser 2 GeoNature sur le principe "parent"-"enfant".
    - Modification des données possibles seulement sur le parent pour éviter une trop grande complexité de synchronisation.
- Pour faciliter cette synchronisation, il serait nécessaire d'avoir une table d'historique des modifications de la table `synthese`.
- Il a été évoqué d'utiliser la table `gn_commons.t_history_actions` qui y est destinée.
    - Problème : le champ json stockant les informations historisées ne permet pas de réaliser des requêtes facilement sur la géométrie des observations.
    - Piste de solution : une table d'historique dédiée à synthèse. Pour la remplir 2 possibilités : triggers classiques ou SQL 2011 "temporal tables".