



# Demande de devis l'industrialisation des processus de déploiement et de mise à jour de GeoNature

Suivi par : Camille MONCHICOURT  
[camille.monchicourt@ecrins-parcnational.fr](mailto:camille.monchicourt@ecrins-parcnational.fr)

04-92-40-20-30

Responsable du pôle Système d'informations / Géomaticien

Parc national des Ecrins

12-09-2019

## 1. Contexte général du projet

Cette consultation vise à faire évoluer l'architecture et le processus de déploiement de l'application GeoNature.

GeoNature est un outil open source dont la communauté d'utilisateurs est grandissante et diverse. Il est souhaité que son déploiement, sa maintenance et sa mise à jour soient simplifiés pour faciliter le travail des développeurs mais aussi pour sa prise en main par des structures aux compétences diverses.

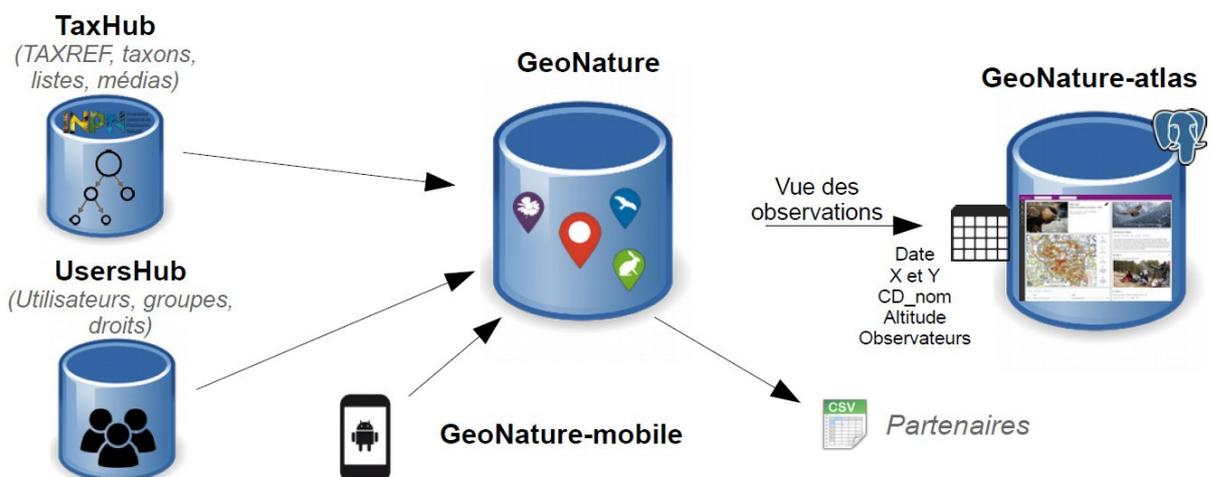
La mission consiste à faire une analyse du fonctionnement actuel et des préconisations d'améliorations puis à conduire un atelier de développements des solutions retenues avec le comité technique de GeoNature.

Le prestataire devra fournir, avant le 1<sup>er</sup> octobre 2019, un devis correspondant à 10 jours de travail, une proposition technique et méthodologique ainsi qu'une proposition de calendrier de réalisation.

### GEONATURE

GeoNature est un ensemble d'applications WEB et mobile pour saisir, gérer, synthétiser et diffuser les données faune et flore multi-protocoles d'un territoire (<http://geonature.fr>).

Elles sont publiées sous licence libre et sont maintenues par le collectif de développeurs du réseau des Parcs Nationaux.

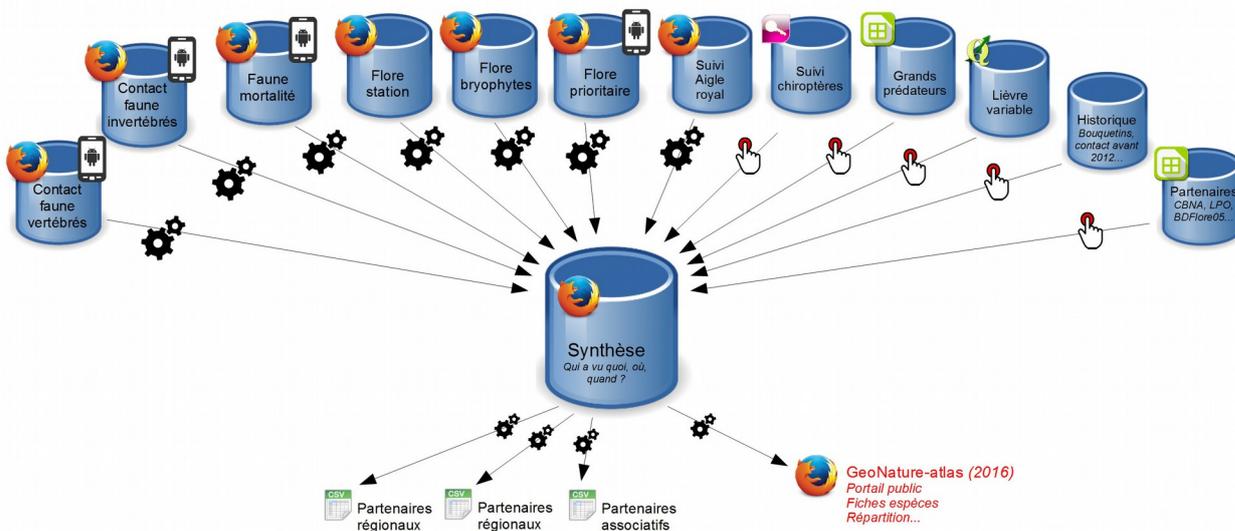


GeoNature permet ainsi de déployer un système d'informations complet pour la gestion des données Faune/Flore d'une structure, allant de :

- la gestion des référentiels (taxonomiques et utilisateurs),
- à la saisie web et mobile dans différents protocoles,



- à la gestion de leurs métadonnées,
- à l'intégration de données de partenaires,
- à l'export des données selon les formats attendus par chaque partenaires
- à la synthétisation des données des différents protocoles sous forme de DEE
- à la diffusion des données sur un portail web grand public



GeoNature regroupe différentes structures utilisatrices dont le PnMercantour, PnEcrins, PnCevennes, PnVanoise, PnGuyane, PNR Normandie-Maine, PNR du Pilat, CREA (Association Mont-Blanc), CBNA, , Flavia (association papillons), LPO Ardèche etc...

Beaucoup d'organismes souhaitent aujourd'hui mettre en place GeoNature sur leur territoire (Parcs nationaux, CBN, PNR, CEN). <https://github.com/PnX-SI/GeoNature/>  
Chaque structure dispose de sa propre instance de GeoNature.

GeoNature peut être testé sur <http://demo.geonature.fr> (admin / admin)

GeoNature (et ses dépendances) peut être installé de manière automatisée à l'aide de scripts de déploiement : <http://docs.geonature.fr>

GeoNature-atlas permet de publier un atlas en ligne pour mettre à disposition de tous les observations saisies dans GeoNature (<https://github.com/PnEcrins/GeoNature-atlas>).

Exemples : <http://biodiversite.ecrins-parcnational.fr> / <http://biodiversite.mercantour-parcnational.fr>

TaxHub permet de gérer le référentiel taxonomique (taxref) pour y renseigner la liste des taxons présents dans chaque structure, d'y greffer des informations spécifiques, de définir des listes de taxons et des filtres en fonction des besoins (<https://github.com/PnX-SI/TaxHub>).

TaxHub permet aussi d'associer des médias et des descriptions aux différents taxons, en vue de leur publication sur un portail web GeoNature-atlas.

TaxHub peut être testé sur <http://demo.geonature.fr/taxhub/> (admin / admin)

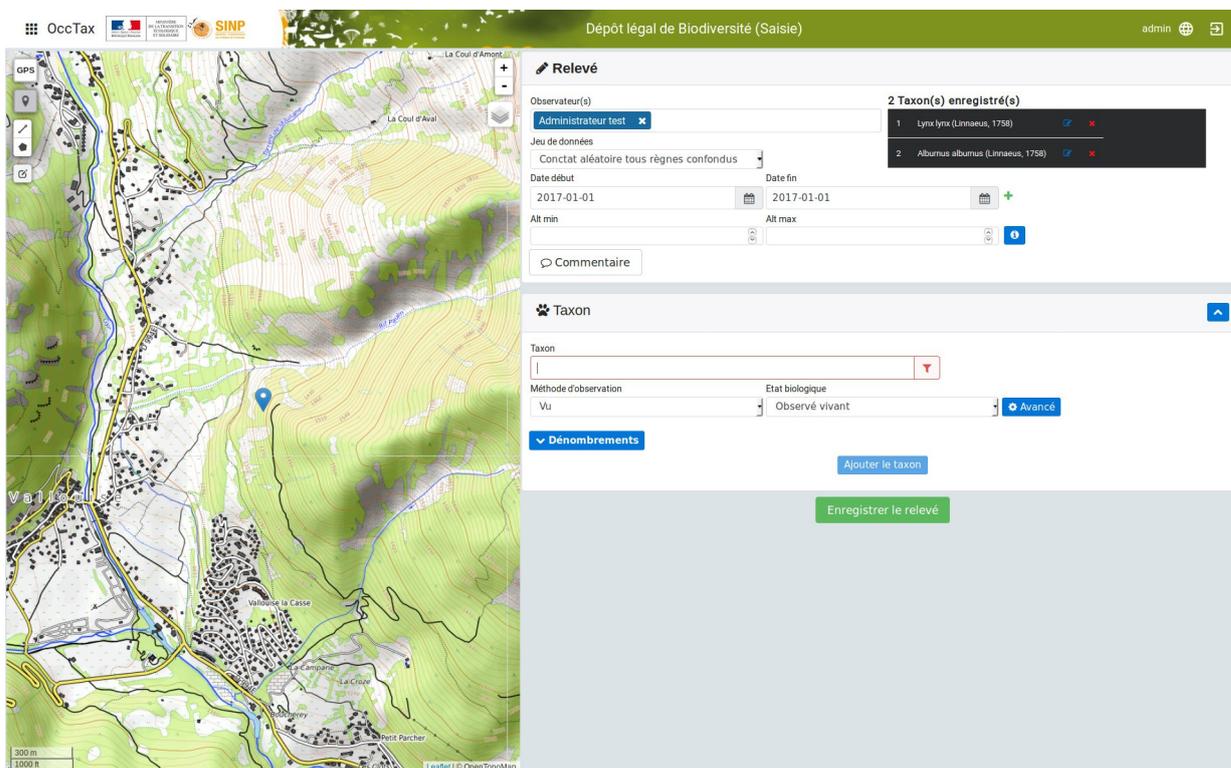
## GEONATURE V2

Fin 2016, l'ensemble des parcs nationaux ont décidé de réaliser une refonte complète de GeoNature pour le moderniser, le rendre plus générique et plus modulaire. Une version 2.0.0 a ainsi été publiée en février 2019 : <https://github.com/PnX-SI/GeoNature/releases>.

Les développements sont principalement réalisés par 3 développeurs (1 PnCevennes + 2 PnEcrins) accompagnés d'un chef de projet (PnEcrins), de contributeurs externes de plus en plus nombreux et d'un groupe projet interparcs nationaux.

Il a été décidé de faire évoluer ses technologies pour s'orienter vers celles de projets plus récents comme TaxHub ou GeoNature-atlas.

- Base de données : PostgreSQL / PostGIS
- Langage : Python
- Framework backend : Flask
- Framework Python : Flask
- Framework Javascript : Angular 4
- Framework JS carto : Leaflet
- Framework CSS : Material / Bootstrap



GeoNature s'appuie sur plusieurs sous-modules génériques permettant de factoriser le code :

- Un [sous-module de gestion des nomenclatures](#) (API et backoffice Flask) permettant de récupérer et d'administrer le vocabulaire applicatifs
- Un [sous-module d'authentification](#). Il s'agit d'une API Flask permettant de gérer l'authentification et la gestion des droits (route de login et logout). Il est utilisé par GeoNature, TaxHub et UsersHub pour contrôler l'accès et les droits. Ce module s'appuie sur un modèle de données commun (modèle de donnée de [UsersHub](#))

## 2. Déploiement de GeoNature

Actuellement le déploiement de GeoNature est effectué grâce à un [script d'installation globale](#) qui installe l'ensemble des dépendances de l'application (GeoNature / TaxHub / UsersHub) ou un script d'installation autonome (n'installant que GeoNature). Ce sont des scripts bash fonctionnels mais manquant largement de souplesse, de robustesse et de retour utilisateur (absence de logs complet, le script ne s'arrête pas si des paquets indispensables n'ont pas été correctement installés). L'utilisateur doit également remplir un fichier « .install\_all.ini » assez touffu, souvent en ligne de commande, étape qui n'est pas toujours évidente pour certains utilisateurs peu familiers avec la manipulation de fichiers sur un serveur.

Cette phase de déploiement gagnerait à passer sur des technologies plus modernes et plus robustes (Ansible, Docker, ou scripts Python – librairie Fabric ...), et éventuellement mettre en place une interface d'installation.

Côté frontend aucun « build » n'est fourni, c'est le script d'installation qui package l'application sur le serveur où est installé GeoNature. Cela nécessite d'embarquer de nombreuses librairies (webpack, angular-cli) qui ne sont pas nécessaires dans un environnement de production. Il serait intéressant de fournir ce build, tout en permettant à l'utilisateur de pouvoir customiser son application (voir [librairies](#)). La même réflexion pourrait être faite côté backend.

### 3. Montée en version et migration

#### Montée en version

La montée en version de GeoNature est actuellement laborieuse. En plus du téléchargement de la nouvelle version de l'application depuis le dépôt GitHub, l'administrateur doit également exécuter des scripts SQL pour faire évoluer le modèle de données (<http://docs.geonature.fr/installation-standalone.html#mise-a-jour-de-l-application>).

GeoNature étant une suite logicielle ayant des dépendances. Lors d'une montée de version, il est parfois nécessaire d'effectuer des mises à jour successives de TaxHub, UsersHub en exécutant des scripts SQL indépendants. La démarche est potentiellement assez complexe si on a laissé passer plusieurs versions et que l'on doit faire successivement faire plusieurs mises à jours.

La modularité de l'architecture (fonctionnement indépendant des différentes briques logicielles) entraîne une complexité dans la démarche finale.

Pour les développeurs, la maintenance manuelle de scripts de mise à jour de la base de données est lourde.

#### Pistes envisagées :

Construire une architecture plus « intégrée » pour GeoNature, tout en gardant la flexibilité initiale voulue.

Deux « modes » pourraient cohabiter, :

- un mode intégré (qui correspondra à 80 % des utilisateurs) ou l'ensemble de la suite logicielle GeoNature serait « packagée »,
- un mode « à la carte » où chaque application pourrait être installée indépendamment, sur des bases de données différentes, des serveurs différents etc.

Dans la version « intégrée », TaxHub et UsersHub pourrait être installés comme des dépendances Python de GeoNature, à l'image de ce qui est déjà fait avec les sous-module d'authentification et nomenclatures.

#### Problématiques liées à l'architecture intégrée :

- Est-il possible d'installer TaxHub comme une des dépendances Python alors qu'elle a un frontend séparé dans une technologie indépendante (AngularJS) ?
- Comment gérer l'authentification et la gestion des droits si toutes les applications sont installées en un seul bloc et partagent le même sous-module d'authentification ?
- TaxHub et UsersHub doivent pouvoir continuer à fonctionner et être installés de manière autonomes car ils sont parfois utilisés dans d'autres projets, en lien avec d'autres applications.

Actuellement le module d'authentification stocke l'utilisateur courant en cookie ainsi que ses droits dans l'application sur lequel il s'est connecté. Le sous-module d'authentification devra prévoir de gérer des droits pour plusieurs application simultanément. Le mécanisme d'authentification par cookie est également quelque peu déprécié, des mécanismes comme le JWT sont actuellement plus souples.

#### Architecture

GeoNature tel qu'il a été construit depuis le début se veut être surtout un « SI biodiversité » avant d'être une application WEB. Cela implique dans son architecture une forte dimension « base de données ». L'application n'a pas été construite avant la base de données (« code first »), c'est véritablement l'application qui vient se brancher sur une BDD existante (« DB first »).

C'est une volonté forte que la base de données puissent fonctionner indépendamment de l'application. Il est très courant que des administrateurs intègrent des données directement dans la base. La cohérence de la base de données et les différents flux existants entre les tables (triggers notamment) doit fonctionner sans rapport avec l'application.

En gardant cette architecture « DB first » on pourrait cependant largement améliorer les migrations SQL en utilisant davantage les possibilités offerte par les ORM. En se basant davantage sur les modèles python (SQLAlchemy) qui décrivent la base de données, et sur des bibliothèques de gestion de migration tels de [Alambic](#) ou [Flask-Migrate](#) on pourrait simplifier les processus de montée en version en évitant d'exécuter des scripts SQL.

Quelques points restent en suspens, notamment celui des fonctions SQL, couramment utilisé dans GeoNature, et de leur portabilité par les ORM (voir ce [sujet stackoverflow](#))

## 4. Description de la mission

La mission sera découpée en 3 phases :

- Réunion avec le comité technique GeoNature pour préciser les besoins
- Audit et analyse du fonctionnement et préconisations sur les solutions à mettre en place
- Atelier de développement des solutions retenues avec le comité technique GeoNature

En effet il est souhaité que les développements soient réalisés avec l'équipe de développement lors d'un atelier de plusieurs personnes dans une logique de collaboration et de transfert de compétences. Industrialisation du processus de déploiement de GeoNature

Le prestataire proposera des solutions permettant d'améliorer le processus d'installation de GeoNature. Il sera libre de choisir (en lien avec le client) les meilleurs choix technologiques, qui devront être adaptés au contexte et compétences des développeurs GeoNature.

Le processus d'installation devra répondre aux contraintes suivantes :

- La définition des paramètres de configuration pré-installation (URL, login/pass, options) doit être simple et explicite (interface?)
- La possibilité de déployer l'application et la base de données sur des serveurs différents
- Le processus doit s'arrêter si des dépendances requises n'ont pas été correctement installées
- L'ensemble des tâches exécutées doit être écrit dans un fichier de logs
- Le script doit pouvoir être lancé plusieurs fois de suite sans que cela n'altère le bon fonctionnement de l'installation et de l'application
- Le processus assurera un retour utilisateur « parlant » durant l'installation
- L'installation devra être garantie sur différentes distributions définies en amont (Debian 8, 9, 10, Ubuntu...)
- Le processus d'installation devra être accessible à un utilisateur « non expert » en administration de serveurs
- Les scripts écrits devront être maintenables et testables (découpez en plusieurs fonctions à responsabilité unique permettant de les tester)

### Architecture, montée de version et migration

Le prestataire proposera les améliorations nécessaires à l'architecture de GeoNature permettant d'aboutir à une version « intégrée » de GeoNature et de ses dépendances. Il proposera aux besoins des évolutions dans chacune des dépendances (TaxHub, UsersHub) et dans GeoNature lui-même. Des évolutions du sous-module d'authentification sont à prévoir pour faire cohabiter la gestion des droits entre les différentes applications.

La revue d'architecture devra aboutir aux exigences suivantes :

- Une version « intégrée » de GeoNature et de ses dépendances sera opérationnelle et installable
- L'installation des dépendances se fera via le gestionnaire de paquet python (pip) à l'image de ce qui est déjà fait avec les sous-modules nomenclatures et d'authentification
- L'utilisateur aura le choix d'installer indépendamment les dépendances s'il le souhaite via une configuration préalable à l'installation
- La montée de version de GeoNature « intégrée » devra intégrer la montée de version de ses dépendances et de leurs éventuelles modifications du modèle de données. Elle pourra également

proposer une MAJ des modules et éventuellement une désactivation des modules non compatibles avec la nouvelle version installée.

- Les montés de version ne devront pas nécessiter le lancement de script SQL (sauf cas extrême comme le changement de version de Taxref par exemple : voir [migration Taxref v11](#))
- Lorsqu'une situation particulière interdit ou complexifie fortement l'automatisation de la montée de version, un processus permettant une intervention manuelle devrait trouver sa place de la manière la plus simple possible. Cette situation peut notamment se présenter lors des update et montées de version.
- Le prestataire proposera des choix de librairies assurant les migrations SQL et les modifications du modèle de données
- Le sous-module d'authentification devra gérer la gestion des droits entre plusieurs applications

## 5. Données fournies par le maître d'ouvrage :

L'ensemble des codes sources des développements existants ainsi que leur documentation sont disponibles sur la forge Github : <https://github.com/PnX-SI>

Des documents complémentaires du projet GeoNature V2 sont aussi disponibles en ligne et pourront être fournis si nécessaire.

Un jeu de données de test sera fourni au prestataire.

## 6. Solutions de développement

Les applications préexistantes ont été développées avec des outils libres et font l'objet d'une libération sur une forge. Le présent marché s'inscrit dans la continuité et il est attendu du prestataire la proposition de solutions open source, réutilisables par d'autres opérateurs sous licences libres et qui ne généreront pas l'achat de licences d'utilisation récurrents.

Si des développements complémentaires sont réalisés, ils doivent être conduits de manière générique et documentée de manière à pouvoir être utilisés par les différents utilisateurs de GeoNature. De la même manière toute amélioration des bibliothèques libres utilisées devront être publiées.

A l'issue du marché les applicatifs développés devront être publiés sur une forge et être réutilisables par d'autres opérateurs sous licences libres.

## 7. Environnement informatique

Le candidat devra respecter les pré-requis techniques existants pour faire fonctionner les applications GeoNature V2, UsersHub, TaxHub.

## 8. Modalités d'échanges

La mise en œuvre de la prestation sera placée sous l'égide d'un groupe de travail technique associant le Parc National des Ecrins, maître d'ouvrage, et le prestataire. Les interlocuteurs seront Camille Monchicourt, responsable du pôle système d'information du PnEcrins et chef du projet GeoNature et Théo Lechémiat, développeur GeoNature au PnEcrins.

Le prestataire désignera un chef de projet et un interlocuteur technique développeur.

Le début de la prestation sera cadré par une réunion fixant les modalités d'échanges et de suivi ainsi que le planning opérationnel. Cette réunion sera soit physique soit par conférence téléphonique associée à du partage d'écrans.

Les développements seront validés par étapes à travers des échanges numériques et téléphoniques autant que de besoins.

## 9. Livrables

Suite à son audit, le prestataire fournira un document présentant son analyse et les solutions proposés.

Il conduira ensuite un atelier de développement avec le comité technique GeoNature pour mettre en œuvre les solutions qui ont été retenues et priorisées.

Si des développements sont réalisés, le prestataire fournira sur la forge Github (<https://github.com/PnX-SI>) :

- l'ensemble des développements
- la documentation des développements réalisés