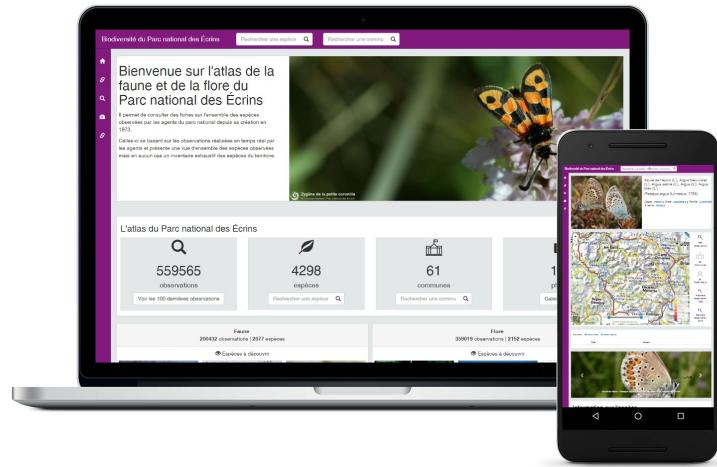




## Rapport de stage Création d'un atlas dynamique de la faune et de la flore au Parc National des Écrins

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



**Rapport présenté par :**  
Théo Lechémia  
Master 2 DCISS

**Maître de stage :**  
Camille Monchicourt  
Parc National des Écrins  
**Tuteur pédagogique**  
Jean-Michel Adam  
Responsable du Master DCISS

## **Remerciements**

Je tiens tout d'abord à remercier mes encadrants durant ce stage : Camille Monchicourt et Gil Deluermoz pour le temps qu'ils m'ont consacré. Merci d'avoir fait confiance à un « novice » en développement web pour ce projet ! J'ai énormément appris pendant cette expérience et ce fut vraiment agréable de travailler au quotidien avec vous ! Je pense que je vous dois beaucoup dans ce poste trouvé en Guadeloupe, donc un grand MERCI !

Je remercie également l'ensemble des collègues du PNE, et plus particulièrement Richard Bonet, pour son accueil au sein du service.

Merci à mes examinateurs Jean-Michel Adam et Jérôme David , à l'équipe pédagogique du master et à toute la promotion DCISS pour cette année sympathique !

Enfin, merci à toute la clique de bobets du parc, ça m'a plû !

## Table des matières

Remerciements.....	2
Lexique :.....	4
.....	4
Introduction :.....	5
1. Présentation de la structure d'accueil et du service.....	6
Le Parc National des Écrins.....	6
Les missions du SI.....	7
2. L'atlas faune – flore du Parc national des Écrins : cahier des charges et architecture.....	9
2.1 Historique et contexte applicatif du projet de l'atlas faune-flore.....	9
2.2 Données naturalistes et dynamiques de l'open-data.....	12
2.3 Cahier des charges de l'atlas.....	13
3. Le projet GeoNature Atlas – choix thématiques et technologiques.....	15
3.1 Gestion de projet et définition du contenu de l'atlas.....	15
Revu des atlas faune-flore existant.....	15
Réunions de coordination du projet.....	15
3.2 Choix des technologies et de l'architecture.....	16
Choix de l'architecture web mise en place.....	16
Architecture de la base de données.....	19
4. L'application GeoNature-Atlas.....	22
4.1 Le modèle logique de données de l'atlas.....	22
4.2 Architecture et structuration du code.....	27
4.3 Configuration de l'application.....	29
5. Présentation de l'application.....	30
5.1 La page d'accueil.....	30
5.2 La fiche espèce.....	32
5.2.1 Les informations issues de la base de données « vm_observations ».....	35
5.2.2 La partie « manuelle » de la fiche espèce.....	39
5.3 La fiche communes.....	41
5.4 La fiche du rang taxonomique.....	42
6. La phase de test.....	43
6.1 Test de la générativité de l'application.....	43
6.2 Test de performances.....	44
6.3 Les tests à réaliser.....	44
Conclusion et perspectives.....	45
Annexes.....	47

## **Lexique :**

**Taxonomie** : la taxonomie et la science de la classification hiérarchique des êtres vivants.

**Taxon** : un taxon est une unité quelconque (genre, famille, espèce, sous-espèce, etc.) issu de la hiérarchie taxonomique. Généralement le terme est employé pour désigner l'espèce ou la sous-espèce.

**Base de données spatiale** : base de données permettant de stocker des informations géolocalisées (points, lignes, polygones) grâce à un type de données nommé « Geometry ».

**Vue-matérialisée** : à la différence des « vues » traditionnelles qui sont des représentations virtuelles d'une requête sur une table existante, les « vues-matérialisées » contiennent physiquement les données issues d'une requête.

**FDW (Foreign Data Wrapper)** : mécanisme de base de données consistant à créer une base fille, vide de toute données, à partir d'une table « mère » existante. La base fille est un simple miroir de la base mère sur laquelle on peut régler les droit d'admission à la base mère

**PostgreSQL** : système de gestion de base de données relationnelle (SGBD) distribué sous licence libre.

**PostGIS** : extension SGBD PostgreSQL, permettant la manipulation d'informations géolocalisées

## **Introduction :**

Le Parc national des Écrins (PNE), où je réalise actuellement mon stage, est un établissement public national dépendant du ministère de l'Environnement. En France, les Parcs nationaux ont été créés par la loi du 22 juillet 1960 ; ils ont pour principale mission la gestion et la protection d'un territoire singulier en terme de faune, de flore, ou encore de paysage.

Le Parc national des Écrins a lui été officiellement créé en 1973. Ses missions de gestion et de protection du milieu naturel ont amené les agents du parc à collecter depuis cette date des centaines de milliers de données naturalistes (faune, flore). Historiquement, ces données servaient à alimenter des études scientifiques et étaient avant tout utilisées en interne. Depuis 2006 et l'instauration d'une « charte des Parcs Nationaux », une dynamique d'ouverture et de diffusion des données s'est mise en place. Ces données doivent non seulement être mises à disposition des partenaires institutionnels (ministères, régions), mais également être accessibles au grand public.

Ces obligations, concordantes à l'arrivée de nouvelles technologies, devaient nécessairement entraîner un bouleversement dans la collecte, l'organisation, et le traitement de l'information. Le PNE a ainsi procédé à une réorganisation de ses services en regroupant le service informatique et SIG<sup>1</sup> dans un pôle SI (Système d'Informations), afin de mener à bien ces missions. Depuis une dizaine d'années le SI a développé une chaîne de travail complète : passant de la collecte de données sur papier et la saisie manuelle sur des tableurs à une collecte numérique et un transfert automatique dans des bases de données spatiales centralisées.

Cette architecture a permis au Parc de développer des outils de diffusion et de vulgarisation des données scientifiques. Une des stratégies récentes de l'établissement est le SIT (Système d'information territorial). Elle consiste à développer des applications web qui mettent en lumière auprès du grand public la singularité du territoire du PNE ainsi que le travail des agents du parc. Deux exemples récemment mis en ligne illustrent cette stratégie :

- l'application « Bouquetin » qui permet de suivre en temps réel le parcours de bouquetins équipés de capteurs GPS (<http://bouquetins.ecrins-parcnational.fr/>) .
- « Rando Écrins » : une application permettant la visualisation et la recherche d'itinéraires de randonnées, de patrimoines et d'hébergements dans le parc (<http://rando.ecrins-parcnational.fr/fr/>).

Dans le cadre du SIT, le PNE souhaite aujourd'hui développer une application de visualisation de ses données faunes et flores à travers un atlas en ligne. C'est sur ce projet que porte mon stage de 5 mois au PNE.

Dans ce rapport de stage, je présenterai d'abord brièvement le service et l'équipe avec lesquels j'ai travaillé ainsi que mon rôle durant ce stage. Dans un deuxième temps je m'attarderai sur la présentation du projet de l'atlas et de son contexte (cahier des charges, réflexions thématiques et techniques). Enfin, je présenterai l'architecture de l'application (base de données et code), puis j'exposerai les différentes fonctionnalités de

---

<sup>1</sup> SIG : Système d'Information Géographique

l'atlas ainsi que les développements effectués.

## 1. Présentation de la structure d'accueil et du service

### ■ Le Parc National des Écrins

Le Parc national des Écrins est un établissement public qui a pour première vocation la préservation de la biodiversité et la mise en place d'une politique de développement durable sur son territoire. Ce territoire de haute-montagne s'étend sur 53 communes entre les départements des Hautes-Alpes - région PACA - et de l'Isère - région Rhône-Alpes-Auvergne.

L'équipe permanente du PNE est constituée d'environ 100 personnes réparties entre le siège à Gap, et sept secteurs qui couvrent l'ensemble du territoire du parc (figure 1). Le siège est composé de quatre services : le service scientifique, le service aménagement, le service communication et le service général.

Mon stage se déroule au sein du service scientifique du PNE. Ce service est lui-même divisé en deux pôles :

- le « pôle connaissance » qui travaille sur la mise en place de protocoles de suivi scientifique (faune, flore et mesures physiques),

- le pôle système d'informations, dans lequel je me trouve, qui s'occupe de la géomatique et de l'informatique. Il est constitué d'un chargé de mission base de données et développement web, d'un chargé de mission administration réseau, téléphonie et informatique, et d'un géomaticien, chef du pôle.

Durant ce stage, ma mission a été le développement web de l'atlas (voir annexe 1 : fiche de poste du stage). J'ai été amené à travailler en grande partie avec Gil Deluerrmoz (chargé de mission développement Web et BDD) et Camille Monchicourt (chef du SI) qui a coordonné le projet. Pour la partie thématique, nous avons également travaillé avec le pôle connaissance du service scientifique et avec le service communication.



Figure 1: Carte de localisation du PNE et de ses secteurs

## ■ Les missions du SI

Le pôle SI occupe une position transversale puisqu'il est amené à travailler avec tous les services du Parc. Il assiste aussi bien le service scientifique dans la mise en place de protocoles de suivi faune-flore, que le service aménagement dans le suivi du patrimoine bâti et de l'agriculture, ou encore le service communication dans la mise en place d'outils de mise en valeur des sentiers de randonnées et l'animation du site web.

De part ses missions de protection de la faune et de la flore, le parc national est amené à collecter des quantités importantes de données **spatialisées**. Le rôle du SI au sein du parc est donc d'organiser et de faciliter la collecte de ces données, de les gérer mais également de créer des outils pour les analyser. Une grande composante métier du SI tient donc dans l'administration de bases de données.

Le PNE a été novateur dans la mise en place de la collecte et le stockage des données sur informatique et possède aujourd'hui une architecture de base de données et des outils structurés.

Le schéma ci-dessous (figure 2) résume la modernisation de la stratégie générale du SI, du recueil de la donnée jusqu'à son traitement et sa consultation. En fonction des protocoles et des besoins des agents, plusieurs chaînes de travail ont été mises en place. La collecte sur le terrain, anciennement effectuée sur papier (en blanc sur le schéma) est aujourd'hui saisie sur des outils nomades (applications mobiles sur tablette). Pour les protocoles importants, l'ensemble de ces données sont ensuite centralisées dans des bases de données PostgreSQL (avec extension PostGIS pour les besoins spatiaux). A partir de ces bases de données, des applications de consultation métier ou grand public sont développées soit par des prestataires extérieurs, soit directement par le SI.

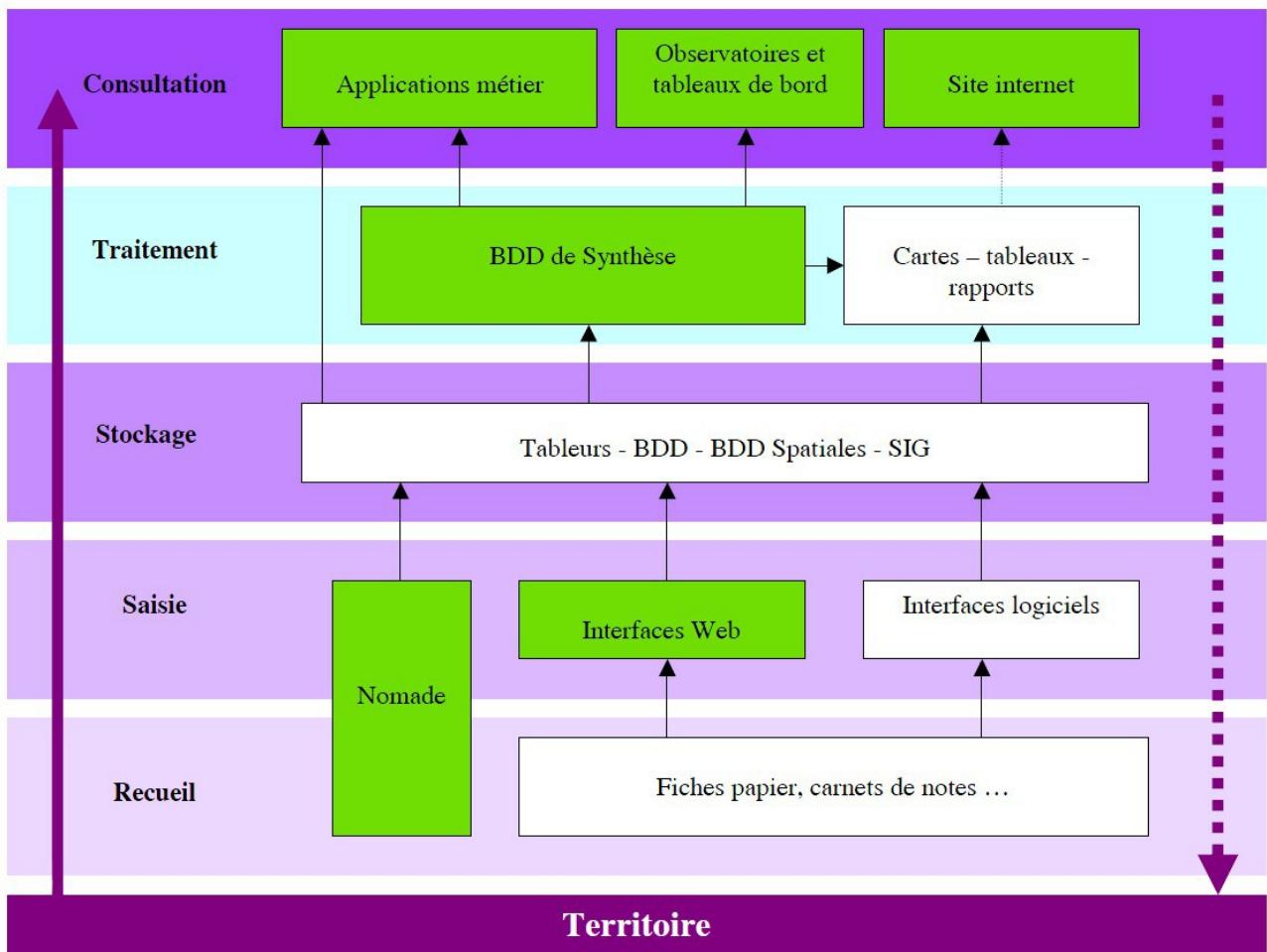


Figure 2: La chaîne de travail du SI : du recueil à la consultation des données (document interne). On distingue en vert la chaîne de travail actuelle, et en blanc l'ancienne.

## **2. L'atlas faune – flore du Parc national des Écrins : cahier des charges et architecture**

L'atlas faune – flore que souhaite développer le PNE s'inscrit dans un contexte applicatif et une architecture de base de données existante s'articulant autour des applications GeoNature et TaxHub (2.1). Il s'inscrit également dans une dynamique plus générale d'ouverture des données au grand public (2.2)

### **2.1 Historique et contexte applicatif du projet de l'atlas faune-flore**

- GeoNature

GeoNature (<https://github.com/PnEcrins/GeoNature>) est un ensemble d'applications web et mobiles qui sert d'outil de gestion des données concernant la faune et la flore au Parc national des Écrins.

Depuis 1973 le parc réalise un « inventaire permanent » de la faune et de la flore. En plus des protocoles particuliers de suivi de certaines espèces, les agents notent quotidiennement les espèces qu'ils observent sur le terrain. Le parc dispose ainsi d'environ 360 000 données d'observation faune et 200 000 de flore représentant en tout plus de 5 000 espèces différentes observées dans les Écrins.

L'architecture de base de données de GeoNature est construite autour du principe : un protocole = un outil = une BDD + BDD de SYNTHÈSE regroupant les données des différents protocoles sur la base de champs communs à tous les protocoles (QUI a vu QUOI, OU et QUAND) (figure 3).

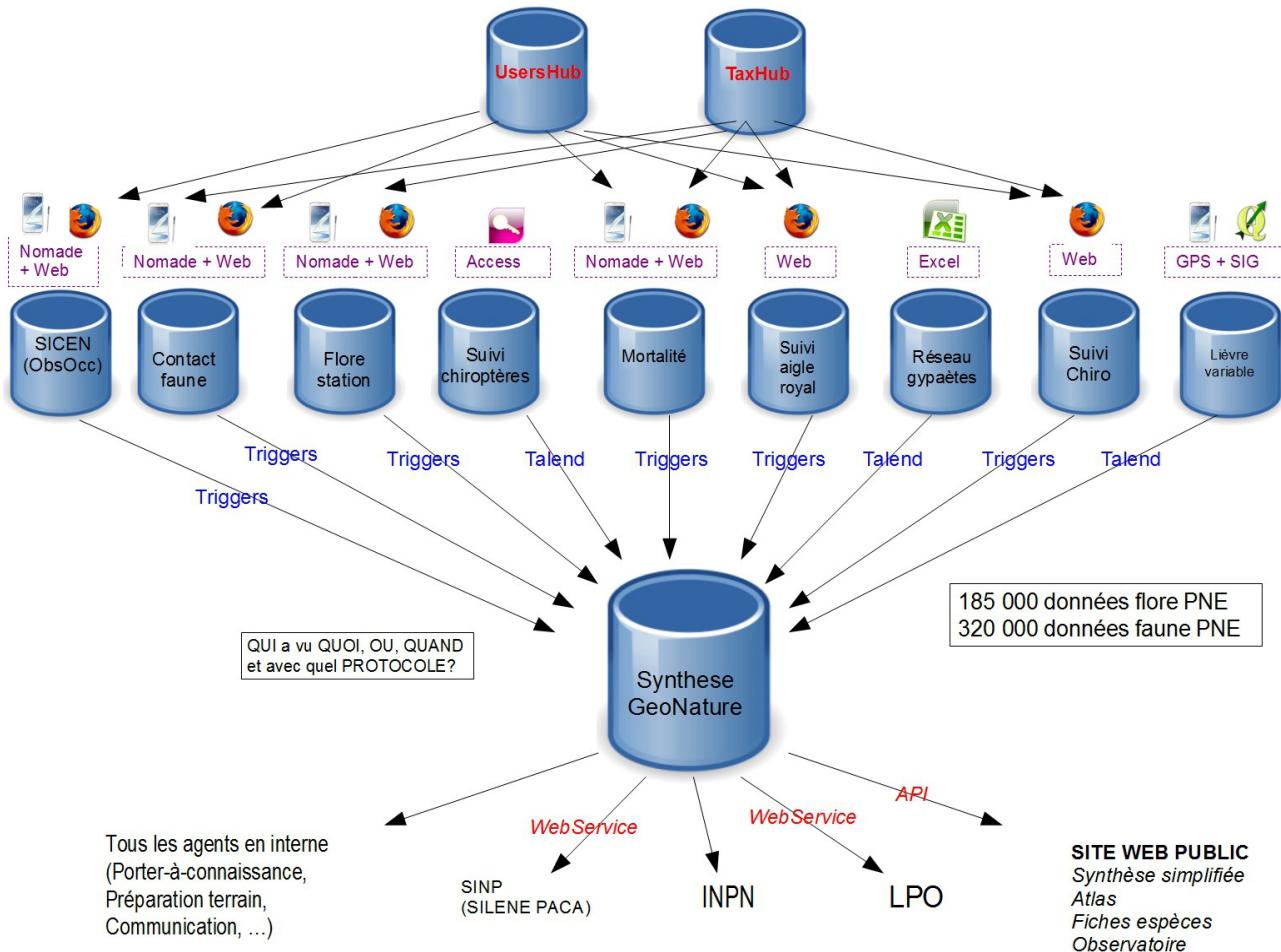


Figure 3: Architecture de la BDD faune-flore du PNE (document interne)

Chaque protocole de suivi faune-flore dispose de son propre outil de stockage des données (base PostgreSQL, Access, Excel etc.). En amont se situent deux bases : UsersHub pour la gestion centralisée des utilisateurs et TaxHub qui gère les caractéristiques de chaque espèce présente dans le parc. Par des triggers (actions automatiques déclenchées en BDD), ou des outils ETL et Talend (outils de traitement automatisé des données), l'ensemble des données sont regroupées dans une grande BDD « synthèse faune-flore ». Cette base centralisée sert de point de départ aux applications métier et grand public du parc et à la diffusion vers les partenaires. Elle est également le point de départ de l'atlas.

Cinq applications mobiles : (Faune, Recherche Flore, Mortalité, Invertébré et Flore Prioritaire) permettent aux agents de saisir les observations directement sur le terrain et d'alimenter les différentes bases de données. GeoNature possède une version « web » de chacune de ces applications, afin que les agents puissent saisir les données lorsqu'ils sont au bureau. La partie « web » comporte également une autre application nommée « synthèse » qui est celle qui se rapproche le plus du futur atlas.

Cette application permet de visualiser toutes les données de la base « synthèse faune-flore ». Elle offre la

possibilité de savoir où a été vue une espèce à un moment donné, puis d'exporter ces résultats sous différents formats (format shapefile, .kmz, tableaux, etc.). C'est un outil technique intéressant pour l'étude du territoire et de son milieu naturel : les agents peuvent en effet à tout moment exporter les données qui les intéressent pour rédiger des rapports, faire des portées à connaissance auprès d'élus ou du grand public. Cependant, cette application est actuellement accessible simplement aux agents du parc et paraît peu adaptée pour être diffusée telle quelle au grand public (figure 4).

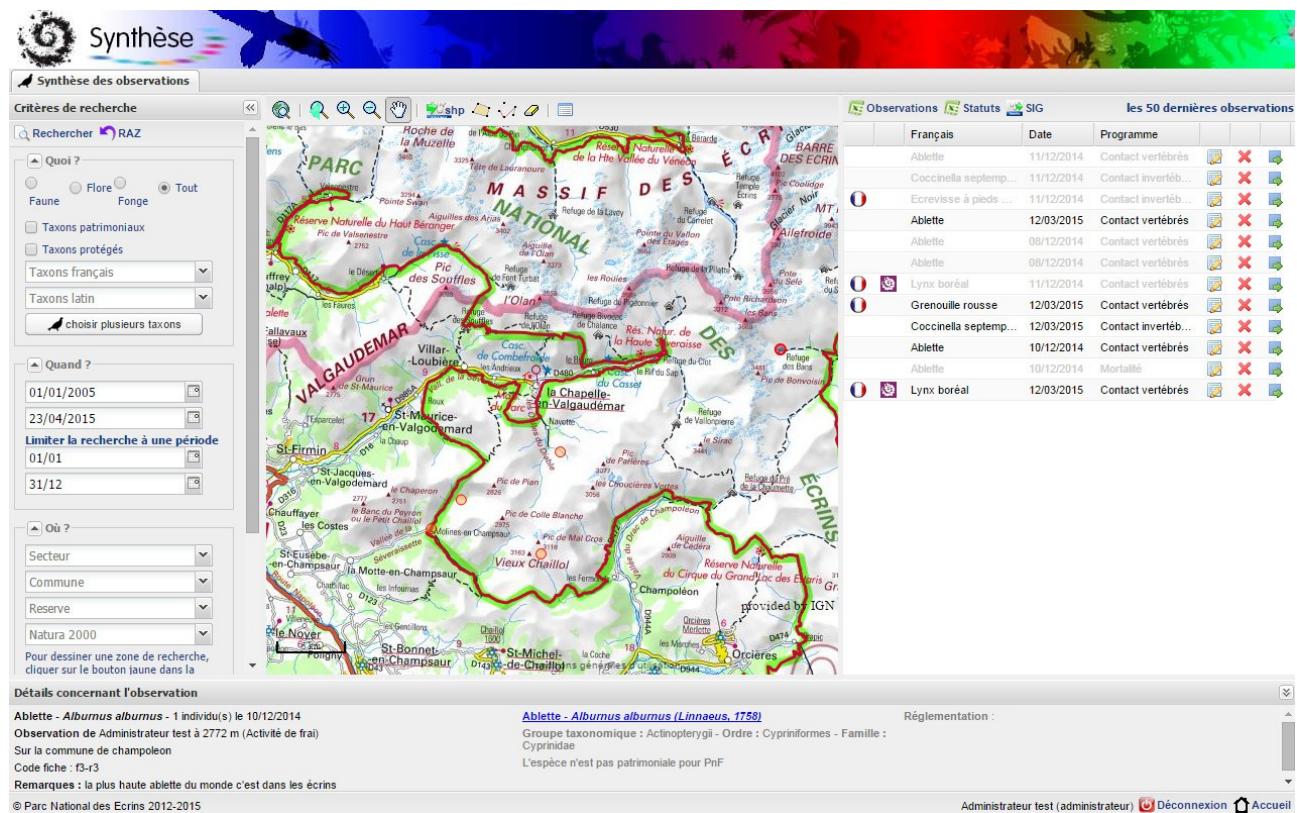


Figure 4: Aperçu de l'application "Synthèse" de GeoNature

L'application « GeoNature-Synthèse » a été développé entre 2012 et 2015 par le SI. Elle fait la synthèse de toutes les observations faune-flore effectuées dans le PNE. Les agents peuvent réaliser des recherches en filtrant par espèce, commune, date etc. Les données issues de l'application peuvent être exportées par les agents ce qui en fait un outil puissant pour la connaissance, la gestion et l'aide à la décision. L'application a été déployée dans les parcs nationaux du Mercantour, de la Vanoise et de Guyane.

- TaxHub

L'application TaxHub, récemment développée, permet de gérer les informations relatives aux espèces et d'administrer la base de données des taxons.

TaxHub s'appuie sur une base de données nationale de référence nommée TAXREF (<https://inpn.mnhn.fr/programme/referentiel-taxonomique-taxref>) administrée par le Muséum d'Histoire

Naturel. Elle centralise toutes les espèces existantes sur le territoire français. Dans la base, chaque espèce est identifiée par un identifiant unique : le « cd\_nom », ce qui permet de grandement faciliter l'échange des données naturalistes entre les structures.

Dans l'application, les chargés de mission faune et flore peuvent ainsi ajouter une espèce nouvellement découverte dans le Parc, lui associer une description ou encore renseigner des caractéristiques spécifiques. Une fonctionnalité en lien avec l'atlas permet d'associer à chaque espèce des médias (photo, vidéos, son) ou encore des articles (voir partie partie 5.2).

- La photothèque et le site interne

Le service communication du PNE est en charge de l'administration d'une photothèque qui regroupe plus de 60 000 photos. Il est également en charge du site internet du parc sur lequel on retrouve de nombreuses actualités se référant à la faune et la flore (actualités sur le suivi des bouquetins, interview radio d'un agent sur les aigles royaux, etc.)

L'objectif de l'atlas est donc de développer un outil à destination du grand public qui puisse remobiliser l'ensemble des données du parc : les données issues d'observations centralisées dans GeoNature, les données relatives aux espèces saisies dans TaxHub, mais également l'ensemble des médias présents sur les différents supports du parc.

L'atlas sera en quelque sorte la vitrine du travail et de la connaissance du parc et de ses agents sur la faune et la flore.

## 2.2 Données naturalistes et dynamiques de l'open-data

Longtemps certaines données faune-flore étaient considérées comme sensibles et les naturalistes n'étaient pas toujours enclins à les partager ou les rendre publiques. Aujourd'hui, la mouvance de l'open-data a quelque peu fait changer les mentalités et une dynamique inverse s'opère. Cette dynamique est actuellement renforcée par des contraintes réglementaires : des directives (notamment Inspire) obligent les organismes publics à diffuser leurs données et fournir des catalogues de métadonnées.

Les données du parc national sont actuellement disponibles sur différentes plates-formes internet notamment SILENE (Système d'Information et de Localisation des Espèces Naturels) (<http://flore.silene.eu/index.php?cont=application&event=init>). Sur ce site, les utilisateurs peuvent rechercher une espèce (faune ou flore) et observer où celle-ci a été vue dans la région PACA (figure 5). Les données sont également accessibles au téléchargement sur demande pour les bureaux d'étude, les étudiants ou encore les chercheurs.



Figure 5 : Capture d'écran de l'application SILENE

Cet outil permet au parc de répondre à ses obligations législatives mais a néanmoins quelques limites :

- la visibilité du parc et de son travail apparaît très faible et ses données sont noyées autour de centaines d'autres.
- les données sont agrégées par maille et sont donc peu précises.
- l'outil est clairement destinée aux spécialistes et aux professionnels et est peu compréhensible pour le grand public.

Le parc souhaite donc aujourd'hui développer une application qui mette réellement en lumière le patrimoine naturel du territoire des Écrins. L'application se devra d'être ergonomique et facilement accessible pour le grand public.

### 2.3 Cahier des charges de l'atlas

- **Le contenu**

À mon arrivée en stage, le contenu global de l'application avait déjà été discuté. L'objectif est de réaliser un « atlas » de la faune et de la flore du PNE. L'atlas devait comporter une fiche descriptive par espèce présente dans le parc. Chaque fiche serait composée de contenu généré automatiquement depuis la base de donnée « synthèse faune-flore » de GeoNature et du contenu complété manuellement espèce par espèce, par les spécialistes et les agents du parc à partir de TaxHub.

Voici les contenus prévus à mon arrivée :

**Contenu généré automatiquement pour chaque espèce :**

- Nom de l'espèce et informations sur la taxonomie
- Carte de répartition des observations
- Graphiques altitudinaux
- Répartitions communales
- Répartitions par massif / secteur / unité géographique
- Graphiques de phénologies (période d'observations)
- Statuts de protection de l'espèce
- Observateurs

#### **Informations saisies manuellement sur chaque espèce:**

- Description de l'espèce
- Commentaire libre agent
- Répartition / distribution (régions)
- Habitats / milieux (liste prédéfinie)
- Médias concernant l'espèce (photo, vidéo, son, articles)

Enfin, une page d'accueil devrait comporter du contenu dynamique sur l'actualité des observations de la faune et de la flore : par exemple les 10 dernières espèces observées, les espèces les plus observées ou encore les espèces patrimoniales, etc. On y trouvera une entrée « grand public » (géographique) où les utilisateurs pourront rechercher les espèces par commune, une entrée scientifique avec une recherche par la taxonomie (nom latin, genre, groupe de l'espèce), et une entrée présentant l'actualité des prospections naturalistes (dernières espèces observées dans le parc etc).

Malgré ces idées assez claires du contenu de l'atlas, plusieurs points faisaient encore débat et nécessitaient d'être discutées avec les thématiciens faune et flore et avec le service communication. Cette partie du travail réalisé durant la première partie du stage est résumé dans la partie 2.3 et davantage détaillée dans le [rapport de mi-stage](#).

- **La générnicité**

En plus de ces aspects thématiques, le PNE avait également des attentes techniques sur le développement de l'atlas. Le parc, s'est engagé depuis le début de ses développements dans une démarche open-source. À l'image de GeoNature ou TaxHub, les diverses applications développées sont amenées à être déployées dans d'autres parcs naturels ou dans d'autres structures dont les besoins pourraient être similaires. Chaque application est donc développée dans un soucis de générnicité afin qu'elles puissent être facilement

partagées. Cette démarche est actuellement commune à de nombreux Parcs nationaux et permet le partage et la mutualisation des coûts de développement.

Un effort important est donc fait sur la généricité et la documentation du code. La totalité des codes-sources sont publiées sous licence libre (GPL V3) sur Github (<https://github.com/PnEcrins/> et <https://github.com/PnX-SI>).

L'ensemble de ces points ont donc dû être pris en compte lors du développement de l'atlas.

### **3. Le projet GeoNature Atlas – choix thématiques et technologiques**

#### **3.1 Gestion de projet et définition du contenu de l'atlas**

##### **Revue des atlas faune-flore existant**

Durant les premières semaines du stage, une partie du travail a consisté à faire une revue des applications et ouvrages existants pouvant se rapprocher de l'atlas souhaitant être développé. En analysant les différents contenus et fonctionnalités, nous avons tenté de faire une synthèse afin de mieux définir nos besoins. Nous avons également essayé d'identifier les points forts et faibles de ces outils et les points sur lesquels notre atlas pourrait se démarquer.

Par des recherches sur internet et dans les ouvrages du parc, nous avons identifié une dizaine d'applications ou livres se rapprochant d'un atlas faune-flore. Pour chacun d'entre-eux j'ai identifié et analysé plusieurs points :

- le public cible (grand public, scientifique)
- Le contenu des fiches descriptives des espèces (graphique, texte descriptif, photographie etc..)
- la présence d'une cartographie (si oui, avec quelle échelle et quelle précision des observations)
- la possibilité de télécharger les données
- le design

##### **Réunions de coordination du projet**

Le travail préparatoire décrit ci-dessus a permis de mieux aborder les futurs réunions de définition du projet en relation avec toutes les parties prenantes, à savoir le pôle connaissance du service scientifique (notamment les thématiciens faune et flore), le service communication, et le pôle système d'informations.

Lors de ces réunions plusieurs points encore en suspens concernant l'atlas ont été discutés.

Un des points importants portait sur la définition du public cible de notre atlas. Bien qu'il était acté que l'application devait être accessible pour le grand public, il était important de définir clairement quel serait les cibles de l'atlas afin d'en adapter le contenu et la maquette.

Il a été décidé que l'atlas se devait de toucher deux grandes catégories de personnes : le grand public (touristes, curieux ou habitants du territoire) et les naturalistes (scientifiques, étudiants, curieux sur une thématique). Ce choix nous a donc amené à définir deux entrées distinctes pour l'atlas : l'entrée « grand public » se voudra géographique et didactique : recherche des espèces par la commune, le secteur du parc, panel des dernières observations, afin que l'utilisateur puisse visualiser les espèces qui se trouvent autour de lui ; et une entrée par la taxonomie : recherche par le nom scientifique ou l'ordre, la famille du taxon.

La sensibilité des données était également un point important abordé lors de ces réunions. La question était de savoir si le parc ouvrirait ou non au grand public la localisation exacte de toutes les observations faites par les agents du parc, notamment dans le cas d'espèces rares ou protégées. Dans les différents atlas investigués, les observations sont dégradées à l'échelle de mailles (de 500 mètre à un kilomètre de côté).

Le choix du parc a finalement été celui d'impulser une dynamique d'ouverture des données : chaque observation d'espèce pourrait donc être localisée sur une carte. Dans un soucis de générnicité, une option d'affichage par maille sera également développée pour permettre à d'autres structures de faire ses propres choix d'affichage des données.

Enfin, d'autres points ont été discutés, notamment sur le contenu exact de chaque fiche espèce, la possibilité de télécharger les données d'observation, ou encore le pilotage du projet en lien avec les autres services impliquées.

Une réunion finale qui s'est déroulée le 6 juin (voir annexe 2 du compte-rendu de la réunion) a permis de fixer les dernières orientations et de commencer le développement.

## 3.2 Choix des technologies et de l'architecture

### ■ Choix de l'architecture web mise en place

A mon arrivée en stage, le choix des technologies pour le développement de l'atlas n'était pas arrêté. On m'a demandé de réfléchir aux technologies et à l'architecture adaptée aux problématiques de ce projet.

La plupart des applications développées par le SI utilisent PhP et Symfony pour le back-office et AngularJS pour le front-end. Cependant en développant récemment l'application « GeoSites », le parc n'était pas tout à fait satisfait du référencement (SEO) par les moteurs de recherche. Les sites développés avec AngularJS (de manière générale avec un framework JS), où l'intégralité des templates sont générés en Javascript côté client pose en effet des problèmes de refinancement. Lorsque l'on ouvre le débogueur de son navigateur, on s'aperçoit en effet qu'en dehors des balises html, le contenu de la page est presque vide. Les moteurs de recherche ne peuvent donc pas « crawler » et donc référencer le résultat de ces pages. De plus, les applications développés avec AngularJS tendent à faire des « Single Page App » (SPA) davantage typé

« application » que site web, ce qui paraît davantage adapté pour les besoins métiers et moins pour le grand public.

Je me suis donc intéressé aux solutions permettant de palier ce problème de référencement.

Plusieurs solutions existent :

- le polymorphisme : le back-end et le front-end sont tous les deux écrits en JavaScript, ce qui permet de générer du contenu HTML à la fois côté serveur que côté client (le framework ReactJS permet notamment cette pratique).

Appliquée à notre projet, l'architecture pourrait donc être la suivante : (figure 6). Le framework Javascript est présent la fois côté serveur et côté client. La BDD est interrogée grâce à un ORM. On peut ensuite générer directement du contenu HTML à partir des résultats de la base côté serveur ou alors construire une API qui sera interrogée côté client. Le framework client générera ensuite les templates HTML.

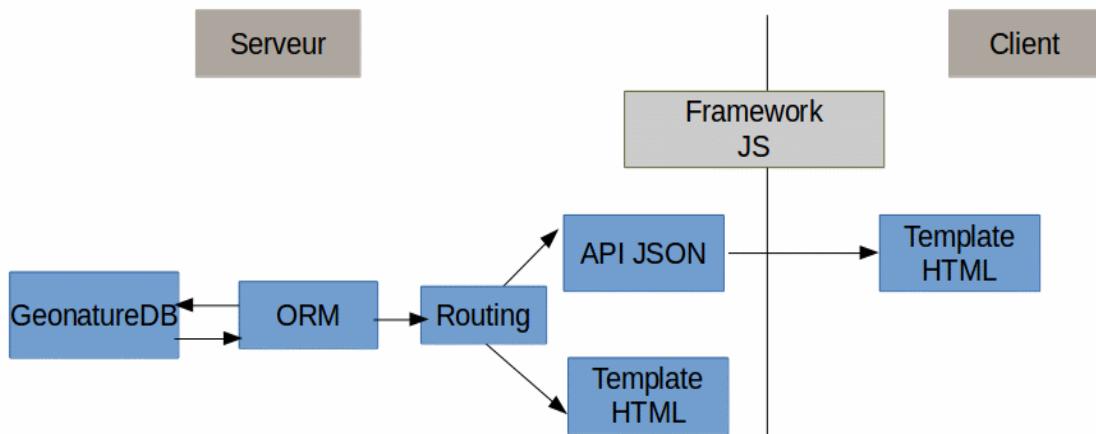


Figure 6: Schéma d'une application web isomorphe

- La seconde solution consiste à générer l'ensemble des templates côté serveur en utilisant un framework web full-stack (Symfony, Django), sans framework lourd côté client (figure 7)

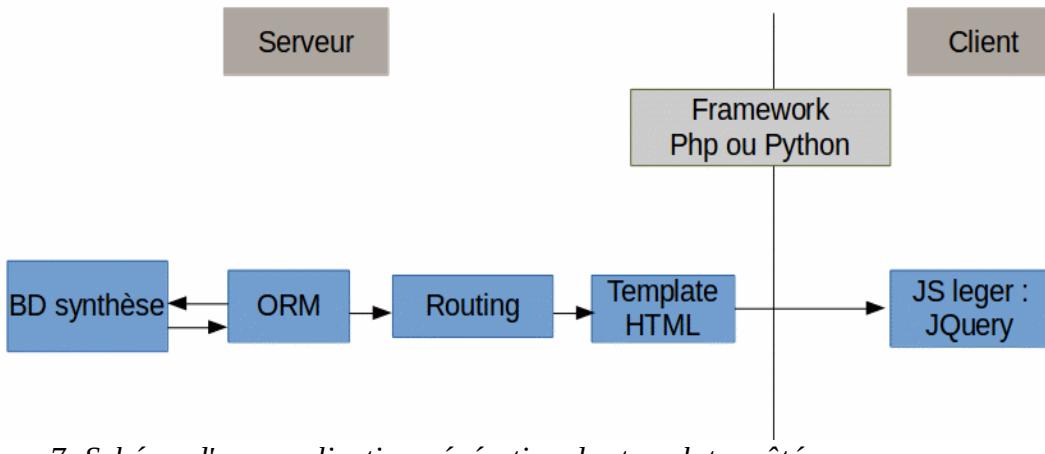


Figure 7: Schéma d'une application génération des templates côté serveur

Concernant le développement, le PNE travaille beaucoup avec les autres parc nationaux français. Comme mentionné plus haut, la logique open-source des parcs pousse à la mutualisation. Le choix d'une nouvelle technologie est donc souvent discuté avec les équipes SI des autres Parcs nationaux. Nous les avons donc consultés afin de prendre une décision sur l'architecture les technologies de l'atlas. Après concertation et présentation de nos problématiques, nous avons décidé de partir sur une solution de génération de template côté serveur (voir annexe 3 : compte-rendu de la réunion SI inter-parc). Le choix de se former sur des framework relativement compliqués tel que ReactJS nous paraissait ambitieux et lourd. Nous avons également fait le choix de nous tourner vers Python pour plusieurs raisons :

- la souplesse et la simplicité de ce langage
- c'est le langage préférentiel de la communauté open-source SIG : ils se prête donc très bien aux problématiques spatiales (de nombreuses librairies spatiales sont disponibles, de plus l'ensemble des plug-in QGIS sont codés en Python).

Ces deux points ont fait pencher la balance en faveur de Python, collant bien aux problématiques métiers des parcs. Concernant, le framework nous avons fait le choix d'une solution légère : le micro-framework Flask, qui contrairement à Django à le mérite d'être souple et plus adapté au petit projet. Mon stage a donc été l'occasion pour le parc national de faire de nouveaux choix en terme de technologie, en accord les SI inter-parc.

Pour finir, la figure 8 illustre l'ensemble des technologies utilisées pour le développement de l'atlas. Pour le côté back-end, les bases sont gérées avec PostgreSQL et PostGIS et interrogées grâce à l'ORM SQLAlchemy. Flask permettra de faire le routing ainsi que la génération de template grâce au module Jinja2. Côté front-end, nous utiliserons Jquery pour la gestion du DOM et les animations. Les librairies Leaflet et

MorrisJS, serviront respectivement à la génération de la cartographie et des graphiques. Enfin, le framemrok Bootstrap gérera le CSS.



Figure 8: Illustrations des technologies utilisées

## Architecture de la base de données

- Foreign-Data-Wrapper et vues matérialisées

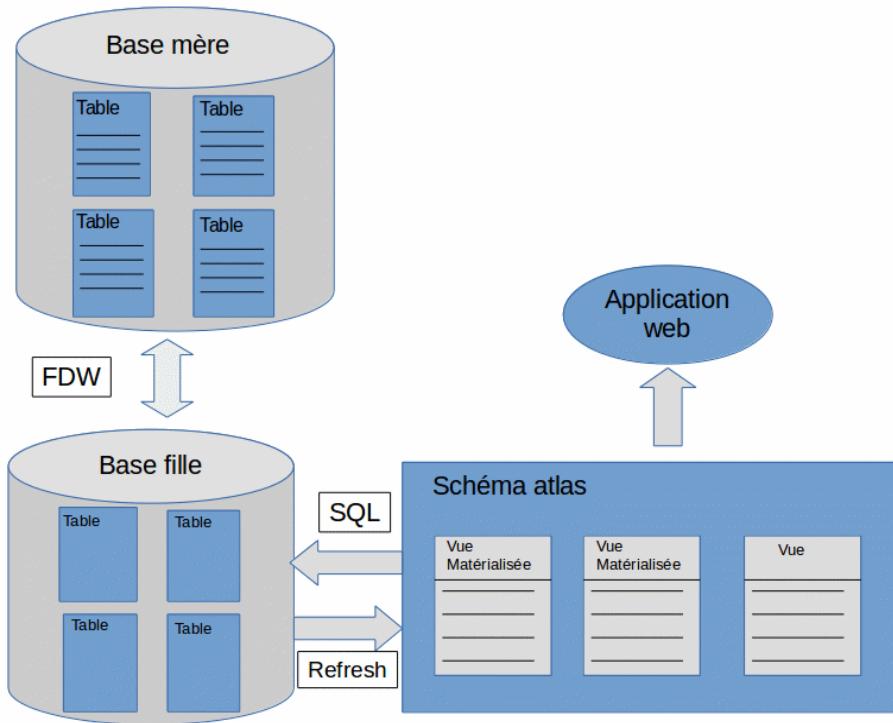
L'application GeoNature du PNE possède déjà une base qui correspond parfaitement aux besoins de l'atlas : elle comporte notamment la table « Synthèse-Faune-Flore » qui regroupe toutes les données d'observations du parc et un schéma nommé « taxonomie » qui renseigne les caractéristiques de chaque espèce. La base de données de GeoNature, embarque cependant toute une série de tables qui ne sont pas nécessaires pour l'atlas (table de chaque protocole, table de gestion des utilisateurs, etc.).

Ne voulant pas se connecter directement à la base de production de GeoNature ni créer deux bases jumelles ayant les mêmes fonctions, nous avons donc choisi d'utiliser le mécanisme des « foreign-data-

wrapper » (FDW) (figure 9). Ce mécanisme consiste à créer une base miroir (base fille) qui se connecte à une base mère, la première pouvant se mettre à jour suivant les modifications de la base mère. La base fille ne contient elle aucune donnée : c'est simplement un miroir de la base mère.

Les foreign-data-wrapper ont notamment une utilité en termes de sécurité. On peut régler les permissions d'accès à la base mère depuis la base « fille ». Dans le cas de l'atlas, aucun besoin en écriture n'est nécessaire, celui-ci a seulement besoin de récupérer les données. Notre base fille a donc seulement des droits en lecture sur la base de GeoNature.

À partir de ces tables distantes, il est possible de construire l'application, mais les performances sont largement dégradées. La solution trouvée a donc été de créer des vues-matérialisées. Les vues matérialisées, reprennent le mécanisme des vues, sauf qu'à la différence de ces dernières, les données sont stockées en dur dans la base fille. Ce mécanisme nous permet donc de simuler un « cache » des données dans la base de données de l'atlas. En effet, toutes les nuits (grâce à une tâche planifiée « Cron » exécutée par Linux), un « refresh » est fait sur, les vues matérialisées sont recalculées en interrogeant la base mère via la base fille : ce mécanisme permet de toujours avoir des données à jour stockées en dur dans la base de l'atlas.



*Figure 9: Schéma du mécanisme "foreign data wrapper" mis en place. La base mère est cloning en une base fille qui ne contient pas directement les données. Des vues matérialisées permettent de stocker en dur dans un schéma nommé « atlas », les données nécessaires. Chaque nuit, un mécanisme de « refresh » recrée les vues matérialisées avec les nouvelles données de la base mère.*

L'ensemble de l'atlas interroge donc uniquement des vues matérialisées. Ce point est très important pour la généricité. Notre objectif pour le développement de l'atlas est que n'importe quelle structure disposant d'une base de données des observations de ses espèces puisse installer l'atlas. Nous ne voulions donc pas que l'atlas dépende de la présence de GeoNature dans la structure en question. La création de « vues-matérialisées » contenant uniquement les données concernant l'atlas permet de se dissocier totalement du schéma de BDD de GeoNature.

#### ➔ Précision sur la généricité et l'installation de l'application

- Le mécanisme du FDW permet de dissocier la logique de la BDD « mère » des observations de la structure, de la logique de l'atlas : il n'est pas obligatoire pour le fonctionnement de l'application.
- Pour faire fonctionner l'application, la structure doit créer les vues-matérialisées qui collent au schéma des vues définie dans l'atlas à partir de sa base de données d'observations. La seule

condition est que la table des observations comporte le champs « cd\_ref » issu de Taxref (table du référentiel national du Museum d'histoire Naturel). Cette table est en effet aujourd'hui utilisée par la plupart des structures qui gèrent des données faune-flore.

## 4. L'application GeoNature-Atlas

### 4.1 Le modèle logique de données de l'atlas

Bien que l'atlas soit construit sur des vues-matérialisées, celles-ci simulent l'existence de tables et l'atlas possède son propre modèle logique de données qu'il convient de présenter en détail (figure 10).

Comme mentionné plus haut, l'ensemble du schéma des BDD de l'atlas part de deux table principales :

- une table des observations naturalistes de la structure
- la table nationale Taxref qui permet d'identifier chaque taxon par un entier unique : le « cd\_nom »

Toutes les vues de l'atlas sont construites à partir de deux dernières tables.

#### - « *vm-taxref* »

La vue « *vm\_taxref* » est une simple copie de la table Taxref du Muséum d'histoire naturelle. Dans cette vue on retrouve tous les taxons présents sur le sol français. Un taxon est identifié par un entier unique: le « *cd\_nom* », son nom scientifique et son nom « vernaculaire » (nom français). Comme un taxon peut posséder plusieurs noms, la synonymie est gérée par le champ « *cd\_ref* » : tous les synonymes d'une espèce sont renseignés sous le même entier nommé « *cd\_ref* ». Le nom de référence du taxon correspond à l'enregistrement où le « *cd\_ref* » est égal au « *cd\_nom* » (tableau 1).

Chaque taxon possède ensuite des champs de description qui renseigne sur sa hiérarchie taxonomique, son ou ses statut(s) de protection.

Dans l'exemple ci-dessous, les trois enregistrements représentent le même taxon : ils possèdent le même « *cd\_ref* ». Chaque enregistrement est cependant identifié par un « *cd\_nom* » différent. La ligne en vert correspond donc au taxon de référence puisque le « *cd\_ref* » est égal au « *cd\_nom* ». « *Saxifraga aretioides* » est donc le nom de référence de ce taxon. Les deux enregistrements en rouge représentent donc des synonymes du premier.

cd_nom	cd_ref	Nom scientifique	Nom vernaculaire
120977	120977	Saxifraga aretioides	Saxifrage de Burser
90972	120977	Chondrosea aretioides	Saxifrage de Burser
120990	120977	Saxifraga burseriana	Saxifrage de Burser

Tableau 1: Exemple d'enregistrement de la vue « vm\_taxref »

#### - « vm\_observations » :

Cette vue regroupe toutes les données d'observations naturalistes de la structure. Dans la vue, une observation est identifiée par un entier unique: l'`« id_observation »`. Chaque observation concerne un taxon qui est identifié par le champs `« cd_ref »` (clé étrangère de la table `« vm_taxref »`: qui est le taxon de référence de la table Taxref). La table renseigne également toute une série d'informations sur l'observation telles que la date, l'altitude, ou encore la localisation exacte (en format `« Geometry » Point – projection 3957` - pour les traitements PostGIS, et en format GeoJson pour être servi à l'application).

La vue des observations est relié à la vue `« vm-communes »`, via la clé étrangère `« insee »`. Dans cette vue, on retrouve la géolocalisation (en polygones) de chaque commune de la structure. Cette vue est construite en faisant une intersection entre la table des communes de France et la vue `« vm_observations »` (fonction Post-GIS `st_intersects`).

#### - « vm\_taxons »

La vue `« vm_taxons »` est construite à partir des vues `« vm_observations »` et `« vm_taxref »`. C'est une réduction de la vue `« vm_taxref »` dans laquelle on ne sélectionne seulement les espèces déjà observées dans la structure et où on retient seulement les noms de référence (tous les synonymes désignant une même espèce sont retirés : on a que les taxons où le `« cd_ref »` est égal au `« cd_nom »`). Dans cette vue, le `« cd_ref »` devient unique et est la clé primaire.

La `« vm_taxons »` est associée à deux vues qui renseignent les caractéristiques de chaque taxon. On a la table `« vm_medias »` où l'on retrouve l'ensemble des ressources associées à un taxon. Il peut s'agir de photos, de vidéos ou encore d'articles web. Dans la vue, un média est associé à un taxon via le `« cd_ref »`, et à un `« id_type »` (clé étrangère de la table `« bib_media »`) qui renseigne le type de médias dont il s'agit (photo, vidéo, etc.). Le chemin vers ce média peut être distant (champs `« url »`), ou local (champs `« chemin »` qui correspond à une adresse relative vers le serveur d'hébergement des médias). Enfin un média est associé à une description, un auteur, une date, etc. (tableau 2).

id_medias	cd_ref	Description	auteur	date	id_type	url	chemin
34	212	Sonneur à ventre jaune	Damien Dupont	06/06/2012	1		static/images/34/212
35	61098	Des bouquetins géolocalisés	PNE	09/08/2014	6	<a href="http://www.dailymotion.com/video/x11yk cw_des-bouquetins-geolocalises_animals">http://www.dailymotion.com/video/x11yk cw_des-bouquetins-geolocalises_animals</a>	

Tableau 2: Exemple d'enregistrements de la vue «vm\_medias »

La vue « *vm\_cor\_taxons\_attributs* » renseigne les caractéristiques descriptives de chaque taxon (description, commentaire, milieu, etc.) et reprend le même schéma que la vue précédente (le « *cd\_ref* » pour l'association à un taxon et le champs « *type\_attribut* » pour l'association au type d'attribut de la « *bib\_attributs* »).

Enfin, dans un soucis de généréricité, nous avons dû complexifier le modèle concernant la table « *vm-observations* ». Cette table renseigne en effet les positions exactes de chaque observation. Or certaines structures susceptibles d'utiliser l'atlas ne souhaitant pas afficher l'observation de manière aussi précise. Nous avons développé un système alternatif de « maillage » qui permet d'agréger les données des observations dans des carrés de 1, 5 ou 10 km.

Le base de l'atlas intègre donc 3 tables fournies à l'installation pour chacun des formats de maillage à l'échelle de la France métropolitaine (« *t\_maille1* », « *t\_maille5* » et « *t\_maille10* »). Lors de l'installation, l'utilisateur choisit la taille de ses mailles (voir plus loin pour le système de configuration). La table « *t\_maille\_territoire* » ne contient que les mailles de l'emprise de la structure. Enfin la table « *vm\_observations\_maille* » est une intersection (fonction *ST\_Intersect* dePostGIS) entre « *the\_geom\_point* » de la table « *vm\_observations* » et la table « *t\_maille\_territoire* » : elle regroupe toutes les observations agrégées par maille. Dans cette vue, chaque maille est caractérisée par son identifiant unique : « *id\_maille* », et on peut retrouver une observation via la clé étrangère « *id\_observation* ».

En plus des vues précédemment décrites, nous avons créé des vues matérialisées pour certains besoins récurrents nécessitant des grosses requêtes en base de données (en vert sur la figure ). La vue « *search\_taxon* » remonte l'ensemble des nom vernaculaires et noms scientifiques ainsi que leurs synonymes des taxons de la structure (soit 32 000 données dans notre cas). Les vues « *vm\_observations\_mois* » et « *vm\_observations\_altitudes* » renvoient pour chaque espèce une répartition

altitudinale et mensuelle, et la vue « *vm\_taxon\_plus\_observes* » renvoie les espèces les plus observées du moment (-15 jours + 15 jours par rapport à la date du jour, toutes années confondues).

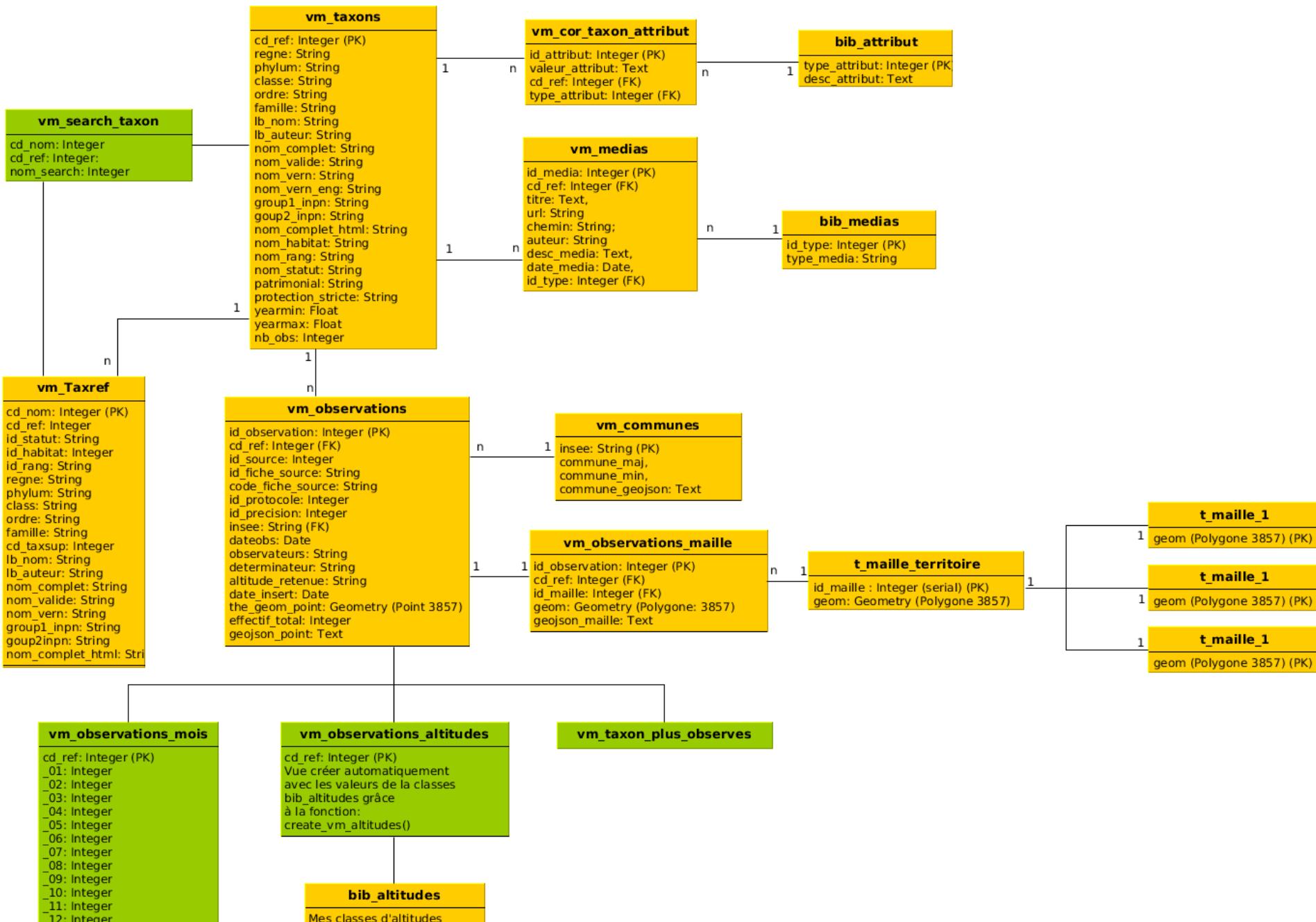


Figure 10: Modèle logique de donnée de l'atlas

- Installation de la base de données

Afin de faciliter et d'automatiser l'installation de la base de données, nous avons écrit un script Linux Bash : « [\*install\\_db.sh\*](#) ». Le langage de commande de PostGreSQL, « psql » permet en effet d'exécuter des commandes SQL directement dans le terminal Linux.

Le fichier « *install\_db.sh* » fait appel au fichier « [\*atlas.sql\*](#) » qui installe la base de données, le mécanisme de FDW, les vues-matérialisées et met en place les droits et les utilisateurs de la base.

En plus de commandes SQL, le fichier « *install\_db.sh* » utilise des librairies pour les besoins spatiaux. L'atlas est livré avec les trois fichiers « *shapefiles* » (couche vectoriel au format ArcGIS) des mailles du territoire français (1,5 et 10 km). Une commande PostGIS : « *shp2pgsql* », permet de transformer ces fichiers « *shapefiles* », en table dans la base de données.

Enfin une autre librairie nommée GDAL (Geospatial Data Abstraction Library : <http://www.gdal.org/ogr2ogr.html>), et plus particulièrement la commande « *ogr2ogr* » a été utilisée pour transformer le fichier « *shapefile* » de l'emprise de la structure en format GeoJson (JSON comportant des coordonnées géographique), afin d'être utilisé directement par la librairie de cartographie Leaflet du côté applicatif.

Le fichier « *install\_db.sh* » fait appel au fichier « [\*settings.ini\*](#) » dans lequel l'utilisateur renseigne tous les paramètres nécessaires à l'installation de la base :

- adresse du serveur de la base de donnée
- nom de la base
- nom utilisateur
- etc.

C'est ici également que l'utilisateur choisit la taille des mailles qu'il souhaite installer dans l'atlas ainsi que d'autres paramètres nécessaires lors de l'installation de la base.

## 4.2 Architecture et structuration du code

L'ensemble du code présenté ici peut être consulté sur le GitHub du projet à l'adresse suivante : <https://github.com/PnEcrins/GeoNature-atlas>

À l'installation de Flask, le framework ne livre pas une arborescence de dossiers déjà préparée. Chacun est libre de structurer son code comme il l'entend. M'étant documenté en première partie de stage sur le framework Symfony, j'ai choisi de reproduire son architecture en adoptant un « patern MVC » (Modèle-Vue-Contrôleur). Cette structure de séparation du code permet en effet de développer de manière claire et structurée, tout en facilitant la maintenabilité.

- La partie « [\*modèles\*](#) »

On retrouve dans la partie modèle le dossier « [\*repositories\*](#) » qui contient le code destiné aux interactions avec la base de données.

Pour segmenter le code il y a un fichier par vues-matérialisées (VmTaxonRepository,

VmObservationsRepository etc.). Chaque interrogation à la base est définie par une fonction qui renvoie un objet ou un tableau d'objets contenant les informations requérées.

- *La partie contrôleur :*

On distingue pour la partie contrôleur deux entités indépendantes.

- Le fichier de routing associé à la génération des templates : « [atlasRoute.py](#) »
- le fichier de routing servant d'API générant des fichiers au format JSON : « [atlasAPI.py](#) »

- Le routing des templates

Dans ce fichier, chaque URL est associée à une fonction et retourne un template HTML. Sur la figure 8, la fonction « index » est associée à la route « / »(la racine du projet) et renvoie le template « *index.html* ». Cette fonction fait appel aux méthodes définies dans les « *repositories* » pour récupérer les informations dans la base.

```
@main.route('/', methods=['GET', 'POST'])
def index():
    listeTaxons = vmSearchTaxonRepository.listeTaxons()
    observations = vmObservationsRepository.lastObservations(100)
    return render_template('index.html', listeTaxons=listeTaxons,
                           observations=observations)
```

Figure 11: Exemple de routing avec Flask

L'atlas dispose de quatre routes pour les quatre templates de l'application

- la page d'accueil : « / » (ou « /index.html » – redirection –)
- la fiche espèce : « /espece/cd\_ref »
- la fiche commune : « /commune/code\_insee »
- la fiche du rang taxonomique : « /liste/cd\_ref »

- Le *routing* de l'API

Afin d'alléger la charge du serveur et d'accélérer le chargement des pages, une partie des besoins de l'application ont été déportée côté client grâce à la réalisation d'une API : [atlasAPI.py](#).

Des requêtes AJAX permettent de charger ce contenu au format JSON à l'application.

L'API est notamment utilisée pour les besoins lourds :

- la liste de toutes les observations d'une espèce (15 000 observations au maximum)
- la liste de tous les taxons du parc (utilisée pour dans le module d'autocomplétion aidant à la recherche : 32 000 taxons pour le PNE)

- *La partie « vue »*

On retrouve dans cette partie les templates HTML (dans le dossier « [template](#) »), et les fichiers JavaScript et CSS (dans le dossier « [static](#) »). Comme expliqué dans le premier rapport, nous avons privilégié une solution avec génération des templates côté serveur. Le moteur de template intégré dans Flask est nommé Jinja2 (<http://jinja.pocoo.org/>), il permet d'utiliser les variables passées dans le template puis de les gérer grâce à des fonctionnalités similaires à JavaScript (boucle, condition, filtre, etc.). Jinja2 permet aussi de

factoriser des « briques » HTML qui sont souvent réutilisées.

On retrouve donc les quatre templates principaux (page d'accueil, fiche espèce, fiche commune et fiche du rang taxonomique). Plusieurs briques HTML séparées sont utilisées pour factoriser les parties répétées dans tous les templates telle que la barre latérale de navigation et la barre supérieure de recherche.

Le code JavaScript regroupé dans le dossier « *static* » est quant à lui segmenté par fonctionnalité. On dispose de fichiers JavaScript pour les fonctionnalités cartographiques, un pour les graphiques, et également un fichier par template pour la gestion du DOM.

Enfin, la partie « *static* » contient également toutes les librairies JavaScript et CSS externes utilisées dans le projet (Leaflet, D3, MorrisJS, Bootstrap).

### 4.3 Configuration de l'application

Comme je l'ai déjà mentionnée à plusieurs reprises, une place importante est accordée à la générnicité dans le travail afin de pouvoir partager et adapter facilement l'application.

La générnicité est d'abord construit dans l'architecture de la base de données, mais l'application en elle même doit également être modulaire.

- Le fichier de [configuration](#)

Dans le code, un maximum d'options sont passés en paramètres afin que l'application soit configurable.

L'ensemble de ces paramètres sont regroupés dans un fichier nommé « [\*config.py\*](#) ».

On y retrouve :

- les identifiants de connexion à la base de données
- les paramètres de configuration des modules cartographiques : choix des fonds de carte, emprise géographique, niveau de zoom, choix de l'affichage du mode « point » ou « maille », etc.
- l'ensemble des paramètres spécifiques à la structure : nom et description de la structure, l'URL d'hébergement de l'application, URL de stockage des médias.
- les paramètres de personnalisation : choix du nombre des « dernières observations » sur la page d'accueil, choix des statistiques à afficher, etc.

- Le fichier de sur-couchage CSS

L'atlas est livré avec un style par défaut. Un dossier nommé « [\*custom\*](#) » comportant notamment un fichier CSS, permet de surcoucher les classes que l'on souhaite afin de personnaliser le style de l'application, les couleurs, la tailles ou l'affichage des blocs.

- Documentation de l'installation et gestion des versions

Enfin, l'effort de générnicité se traduit par la rédaction régulière de documentation. Une documentation (encore en construction : <https://github.com/PnEcrins/GeoNature-atlas/tree/master/docs>), d'installation complète a été publiée sur le GitHub de l'application.

Concernant la possibilité d'évolution de l'atlas, les futurs versions seront gérées grâce au dépôt GitHub. Un utilisateur de l'atlas pourra ainsi télécharger la nouvelle version et simplement rapatrier dans son nouveau dossier les fichiers de configuration « *config.py* », « *settings.ini* » et le dossier *custom* pour que l'application

soit à nouveau fonctionnelle.

## 5. Présentation de l'application

Dans cette partie, les différentes fonctionnalités de l'atlas seront présentées, en spécifiant les développements effectués pour chacune d'entre-elles.

### 5.1 La page d'accueil

La page d'accueil –le point d'entrée de l'atlas – se veut être le portail de l'actualité des observations naturalistes de la structure. On retrouve les trois entrées qui avaient été définies dans le cahier des charges : l'entrée grand public et géographique (avec la carte des dernières observations et la recherche par commune), l'entrée taxonomique avec la recherche des espèces par le nom latin, et une entrée par l'image avec une galerie photos des espèces les plus observées du moment.

- Le bloc de présentation de l'application

Dans cette partie on retrouve simplement le titre et la présentation de l'application. Chaque structure définit dans le fichier de configuration son nom, le texte de présentation ainsi que la photographie de l'accueil (figure 12).

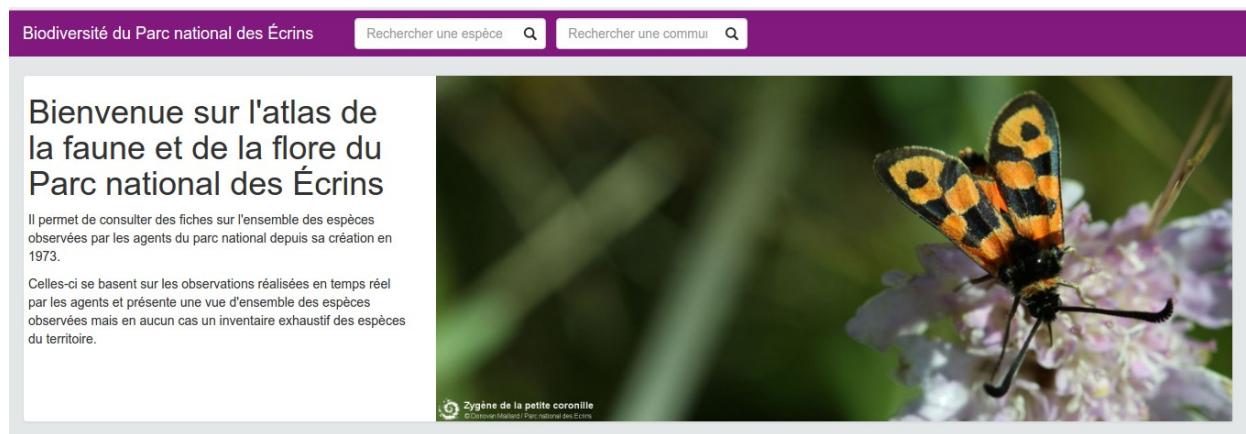


Figure 12: Le bloc de présentation de la page d'accueil

- Le bloc de statistiques de la base de données

On retrouve dans ce bloc un tableau de bord de l'atlas avec le nombre d'observations, le nombre d'espèces, le nombre de communes et de photographies présents dans l'atlas. L'utilisateur peut directement rechercher une espèce ou une commune avec une autocomplétion qui l'aide dans sa saisie (figure 13).

Sous ce tableau de bord se trouve un second bloc de statistiques qui propose de zoomer sur une « famille » d'espèces : ici on voit les statistiques pour la flore et la faune. Dans le fichier de configuration, on peut choisir les deux rangs taxonomique du bloc (insecte, papillons, faune, flore, etc...). La requête remonte ainsi les 10 espèces les plus observées du rang taxonomique choisi et affiche 3 espèces de manière aléatoire pour lesquelles on peut directement accéder à la fiche espèce.

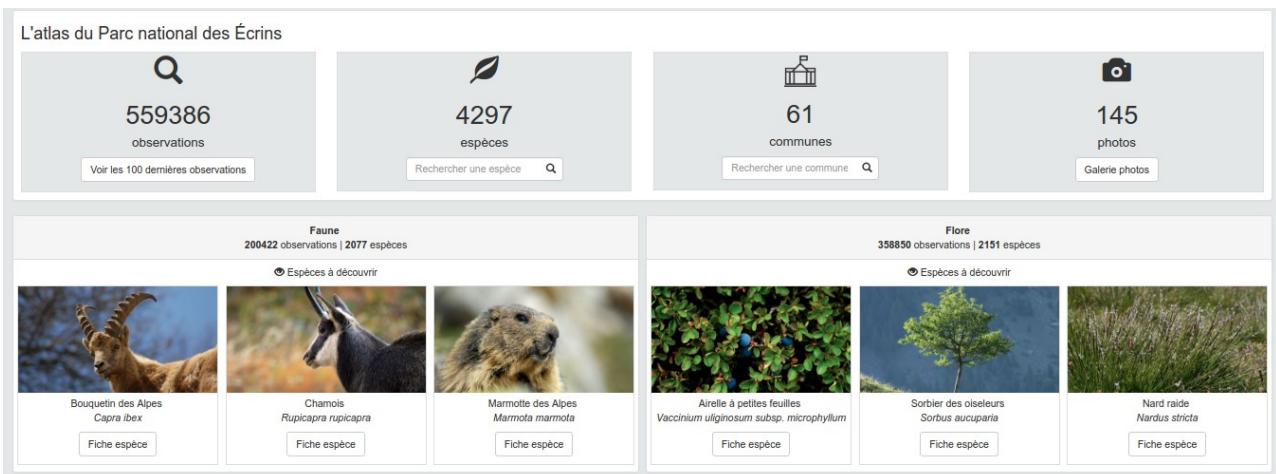


Figure 13: Le bloc de statistiques de la page d'accueil

Ce module permet de guider les utilisateurs « grand public » vers des espèces (souvent emblématiques du territoire), sans qu'ils aient forcément en tête le nom d'une espèce.

- Le bloc cartographie

Ce second bloc constitue l'entrée géographique de la page d'accueil.

On y retrouve une carte et une liste des 100 dernières observations (seuil paramétrable). La liste est interactive : au clic sur une espèce, la carte zoome sur l'observation choisie.

Comme l'application nécessite de nombreux besoins en cartographie, nous avons factorisé la création d'une carte de la librairie Leaflet dans une fonction nommée « `generateMap()` ».

Cette fonction permet de créer la carte avec deux fonds de carte définis dans le fichier de configuration (IGN, OpenStreetMap, orthophotographie, etc.), d'y ajouter une légende, le module permettant de passer en plein écran, et de changer de fond de carte .

Leaflet fournit nativement un bouton permettant de changer de fond de carte, mais celui-ci nous paraissait peu intuitif. Nous avons donc développé un petit module à la façon « `google maps` » qui permet de changer facilement le fond de carte (figure 14).



Figure 14:Le bloc cartographique de la page d'accueil

- La galerie photos

Ce dernier bloc constitue l'entrée « photographique » de la page d'accueil. On retrouve ici une photo des espèces les plus observées du moment. Pour calculer ce résultat il est nécessaire de parcourir l'ensemble de la table « *vm\_observations* » ce qui représente un travail assez lourd. Nous avons donc créé une vue matérialisée pour ce besoin : « *vm\_taxon\_plus\_observees* ». Il s'agit des espèces les plus observées sur la période « plus 15 jours », « moins 15 jours » en partant de la date du jour, toutes années confondues. Ce pas de temps est paramétrable dans le fichier « *settings.ini* » puisqu'elle est calculée à l'installation de la base de données.

Au survol de chaque photographie, le nom de l'espèce apparaît et on peut cliquer pour découvrir sa fiche espèce (figure 15).

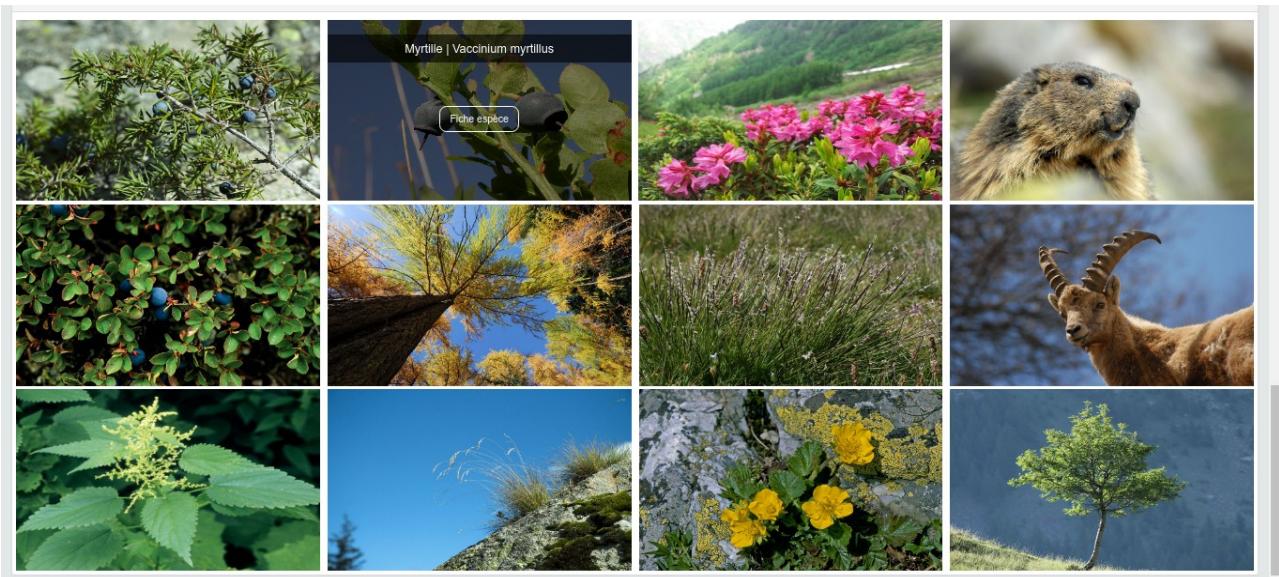


Figure 15: La galerie photos des espèces les plus observées du moment

## 5.2 La fiche espèce

[La fiche espèce](#) est la page qui regroupe toutes les informations dont la structure dispose sur le taxon sélectionné : informations taxonomiques, description, cartographie des observations, médias, articles etc. (figure 16).

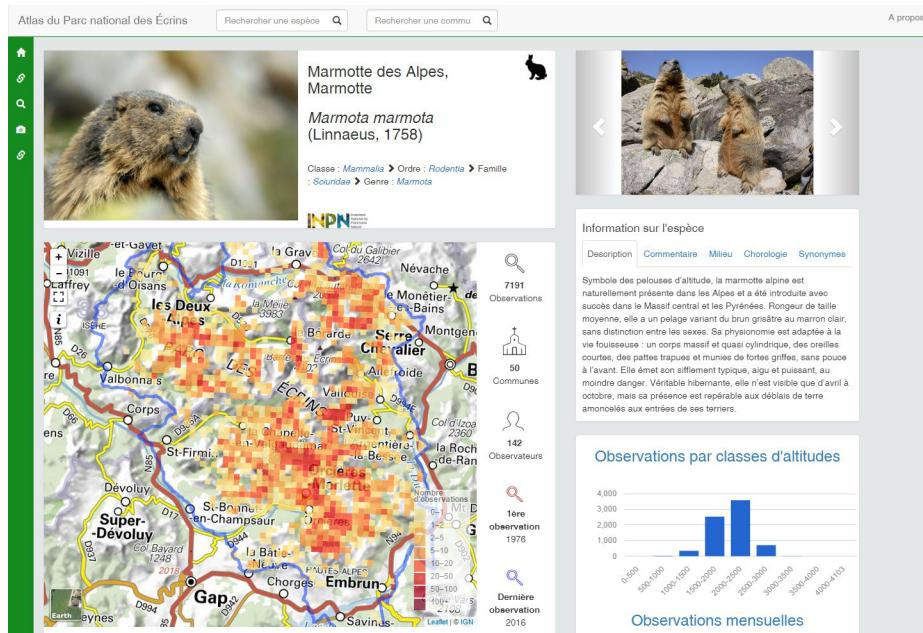


Figure 16: Aperçu global de la fiche espèce

L'atlas fournit une fiche par espèce observée dans la structure (chaque « `cd_ref` » dispose de sa fiche et pas chaque « `cd_nom` » : c'est à dire qu'il n'y a pas une fiche par synonyme, mais que tous les synonymes d'une espèce sont regroupés sur la même fiche). L'outil de recherche permet cependant de faire une recherche sur les synonymes pour les personnes connaissant une espèce sous un nom qui n'est pas celui de référence.

- Récursivité et fiche espèce

Un point important à souligner ici concerne la hiérarchie taxonomique et son incidence sur le contenu des fiches espèces. Le principe de la taxonomie est un bon exemple de structures de données arborescentes. Le sommet de l'arbre représente le monde vivant, vient ensuite le règne « animal » ou « végétal » et on descend ainsi jusqu'à l'espèce, la sous-espèce, la variété etc... Chaque « feuille » de l'arbre représente un taxon. Un taxon a un père, et peut avoir plusieurs fils (figure 17).

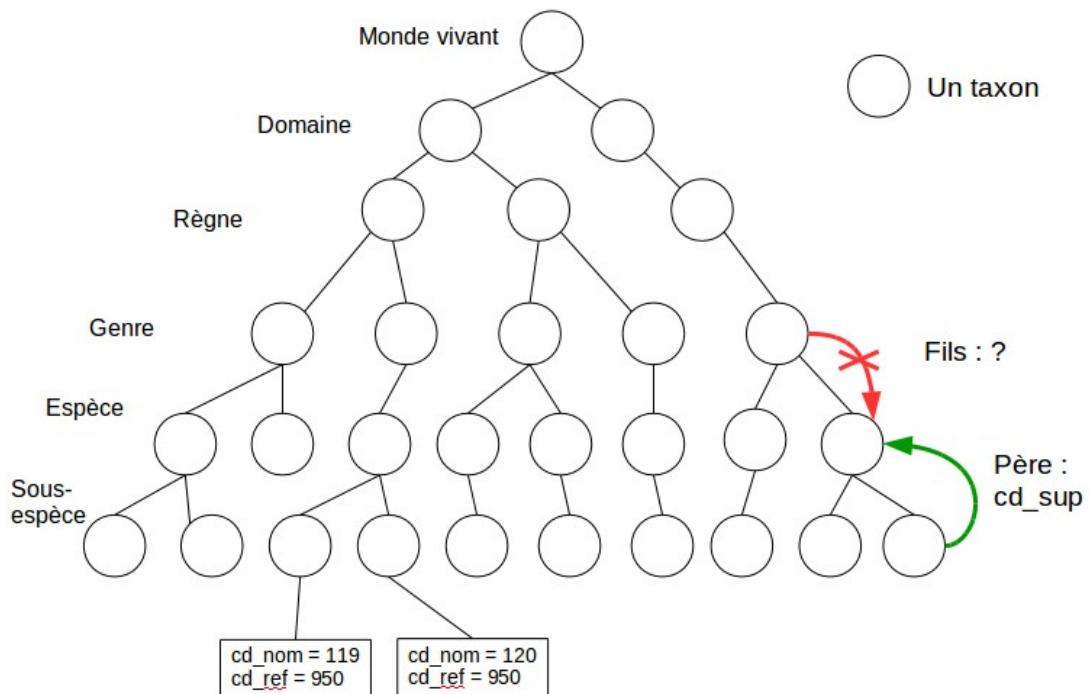


Figure 17: Schéma illustrant l'arbre de la hiérarchie taxonomique

Lorsqu'un taxon a été observé et saisi dans la base de données à un rang taxonomique donné : ce taxon bénéficie d'office d'une fiche dans l'atlas. Par exemple, il peut arriver qu'un garde saisisse l'observation d'une plante au niveau du « genre » car il n'arrive pas bien à l'identifier, puis que plus tard un autre garde rencontre une plante similaire et la saisisse au niveau de la « sous-espèce ». Dans l'atlas il est important que la fiche au niveau du « genre » agrège l'ensemble des données existantes aux niveaux inférieurs (espèce, sous-espèce etc.).

La base de données national Taxref sur laquelle s'appuie l'atlas dispose d'un champs « `cd_tax_sup` » qui permet de connaître le père d'un taxon, mais ne permet pas de connaître les fils de ce taxon. Or c'est précisément de ce champs dont nous aurions eu besoin pour l'application. Nous avons donc réalisé une fonction récursive : « `find_all_taxons_child(cd_ref)` » en base de données qui permet de trouver cette information.

PostgreSQL permet de faire des requêtes récursives grâce à la syntaxe « WITH RECURSIVE » (figure 18 ).

```

Fonction : find_all_taxons_child(id) :

DECLARE
    inf RECORD;
    c integer;
BEGIN
FOR inf IN
    WITH RECURSIVE descendants AS (
        SELECT tx1.cd_nom FROM atlas.vm_taxref tx1
        WHERE tx1.cd_tаксsup = id
    UNION ALL
        SELECT tx2.cd_nom FROM descendants d
        JOIN atlas.vm_taxref tx2 ON tx2.cd_tаксsup = d.cd_nom )
        SELECT cd_nom FROM descendants
    LOOP
        RETURN NEXT inf.cd_nom;
    END LOOP;
END;

```

Figure 18: Fonction permettant de calculer tous les fils d'un taxon

La syntaxe WITH RECURSIVE permet d'utiliser le résultat d'une requête dans une boucle, tant que ce résultat n'est pas NULL.

La fonction prend en paramètre l'identifiant du taxon : le « cd\_ref », la requête recherche ensuite dans un premier temps tous les fils d'un taxon (en recherchant tous les taxons dont le « cd\_tаксsup » est celui passé en paramètres). Le résultat de cette requête est ensuite utilisé pour rechercher les fils des taxons renvoyés et ainsi de suite.

Cette fonction renvoie donc un tableau de « cd\_nom » correspondant à tous les fils du taxon recherché. Cette fonction est ainsi utilisée pour l'ensemble des informations relatives à une fiche espèce.

- Contenu de la fiche espèce

Sur chaque fiche espèce on distingue les données calculées automatiquement à partir de la base de données des observations (4.2.1) et les données renseignées manuellement par les agents de la structure (4.2.2)

### 5.2.1 Les informations issues de la base de données « vm\_observations »

- Fiche d'identité

Ce premier bloc présente rapidement l'espèce. On y retrouve une photographie de l'espèce, son nom commun (ou nom vernaculaire), son nom scientifique, sa hiérarchie taxonomique, et éventuellement le nom de l'ensemble des taxons fils qui sont agrégés sur la fiche (figure 19).



Figure 19: Aperçu de la fiche d'identité du taxon « Marmotte » de la fiche espèce

Chaque rang de la hiérarchie taxonomique est un lien qui renvoie, soit vers une autre fiche espèce du père de cette espèce, soit vers une autre page qui présente une liste de l'ensemble des espèces observées appartenant à ce rang taxonomique ou à ces fils (voir partie 4.4 pour plus de détails).

Il est possible de paramétriser le rang à partir duquel on passe d'une fiche espèce à une simple liste, en effet certaines structures souhaitent présenter une fiche espèce jusqu'au niveau du « genre », alors que certaines souhaitent simplement afficher ces fiches jusqu'au rang de « l'espèce » et des listes pour les rangs supérieurs.

- Le bloc cartographique

Ce second bloc présente une cartographie de l'ensemble des observations du taxon (et éventuellement de ses fils), sur le territoire de la structure.

Comme précédemment évoqué, il existe deux options en fonction de la précision d'affichage des données souhaitées par la structure : un mode point qui montre la précision exacte de chaque observation et un mode maille qui agrège les données dans des carrés de 1,5 ou 10 kilomètres. Dans le fichier du configuration c'est une variable booléenne qui gère cette option : un fichier JavaScript différent est chargé en fonction de l'option choisie.

En mode point, au chargement de la page, les observations sont d'abord affichées en format mailles : cela permet d'avoir une vue d'ensemble de la répartition des observations (figure 20). On voit en rouge-orangé les mailles qui ont beaucoup d'observations, et en jaune-beige les mailles dont l'occurrence d'observation de l'espèce est plus faible.

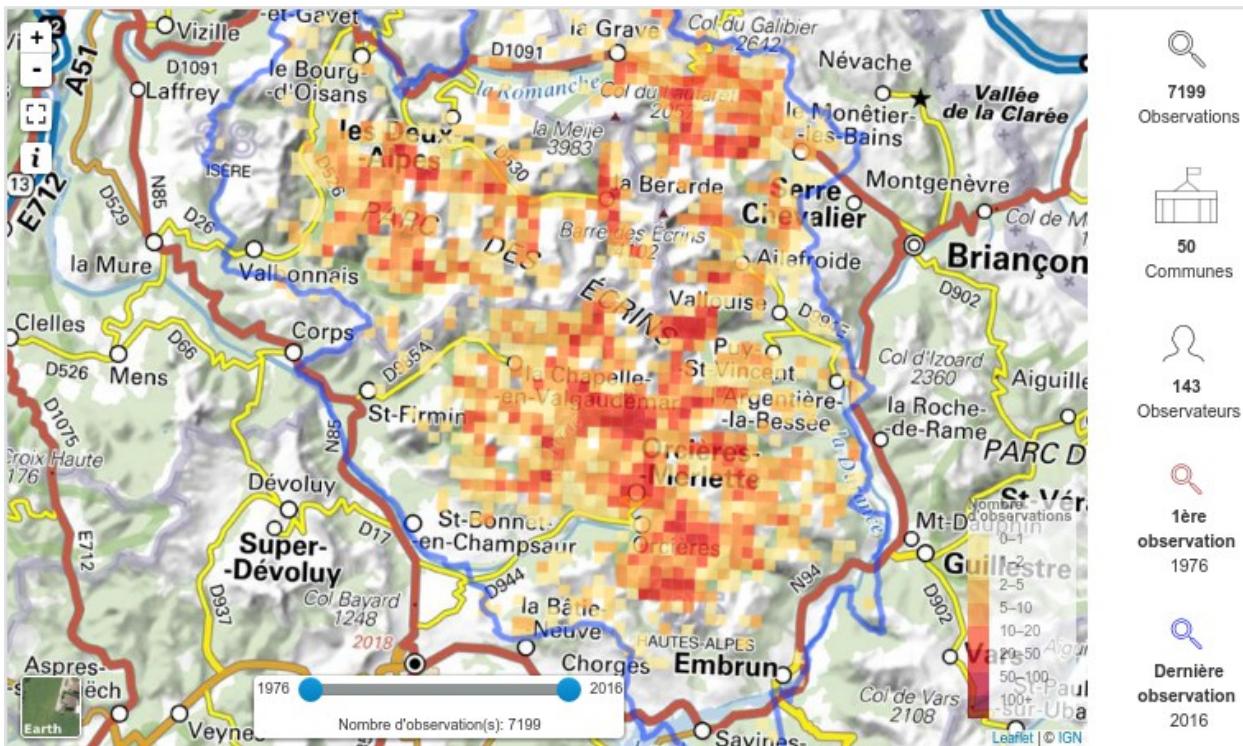


Figure 20: La carte des observations d'un taxon en mode maille

En zoomant, on passe en mode point (le niveau de changement de l'affichage des observations est paramétrable). Si la fiche espèce contient plus de 1 000 observations, pour des soucis de performance, les points proches sont agrégés dans des « clusters » (figure 18). Au zoom maximum, chaque observation est représentée par un rond bleu (figure 21)

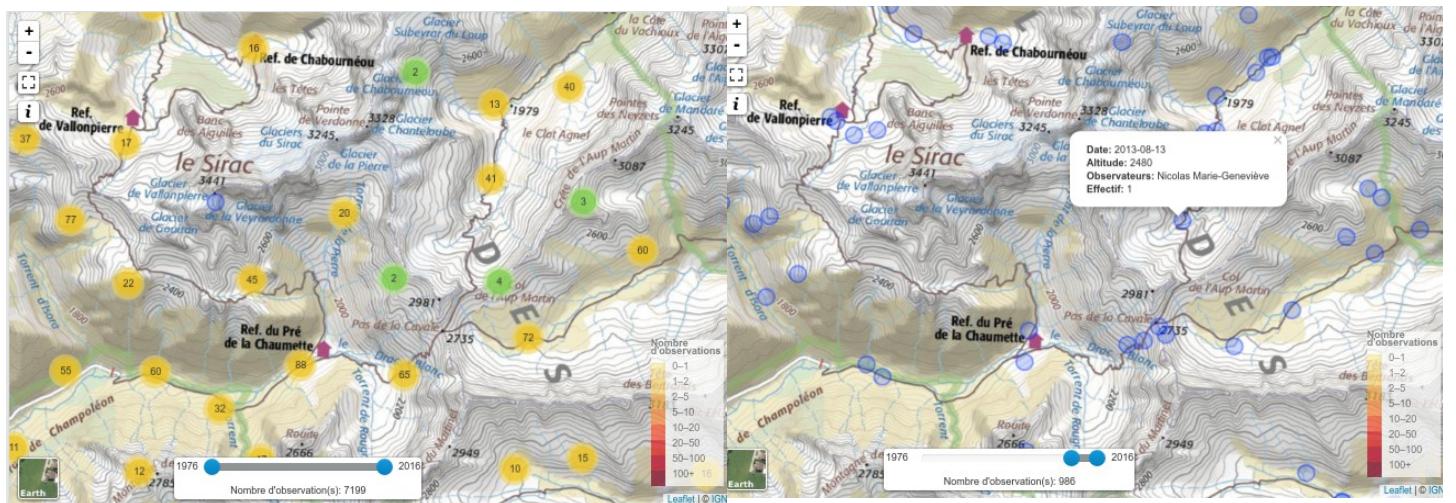


Figure 21: La carte des observations en mode cluster (à gauche) et en mode point (à droite)

L'utilisateur peut également cliquer sur le point de l'observation pour faire apparaître une pop-up qui renseigne sur l'observateur, l'altitude, la date et le nombre d'individus de l'observation.

En mode maille, on ne peut pas afficher la localisation exacte de l'observation. Pour chaque maille une pop-up renseigne le nombre d'observations et la date de la dernière observation à l'intérieur de la maille (figure 22).

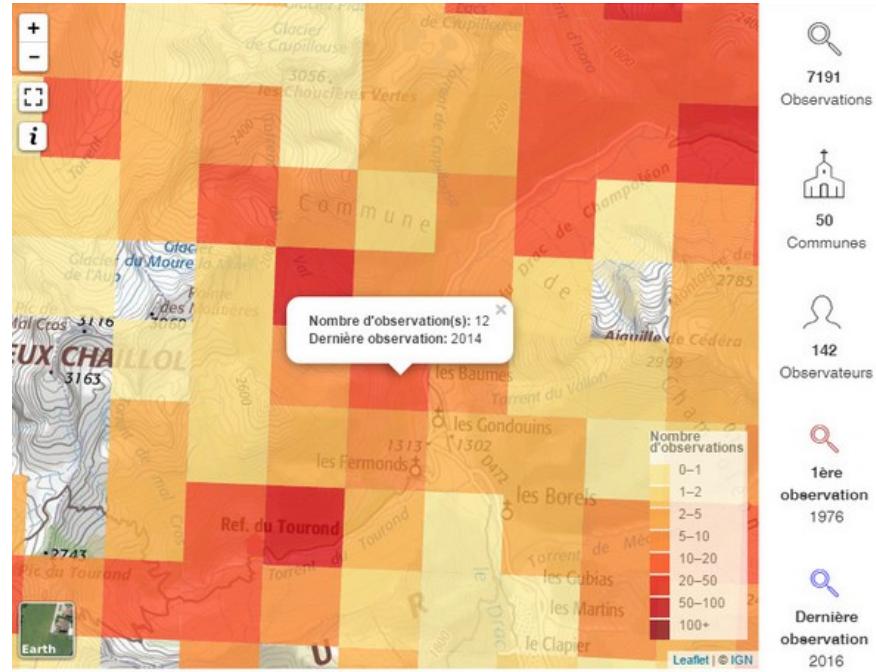


Figure 22: Affichage des observations en mode maille

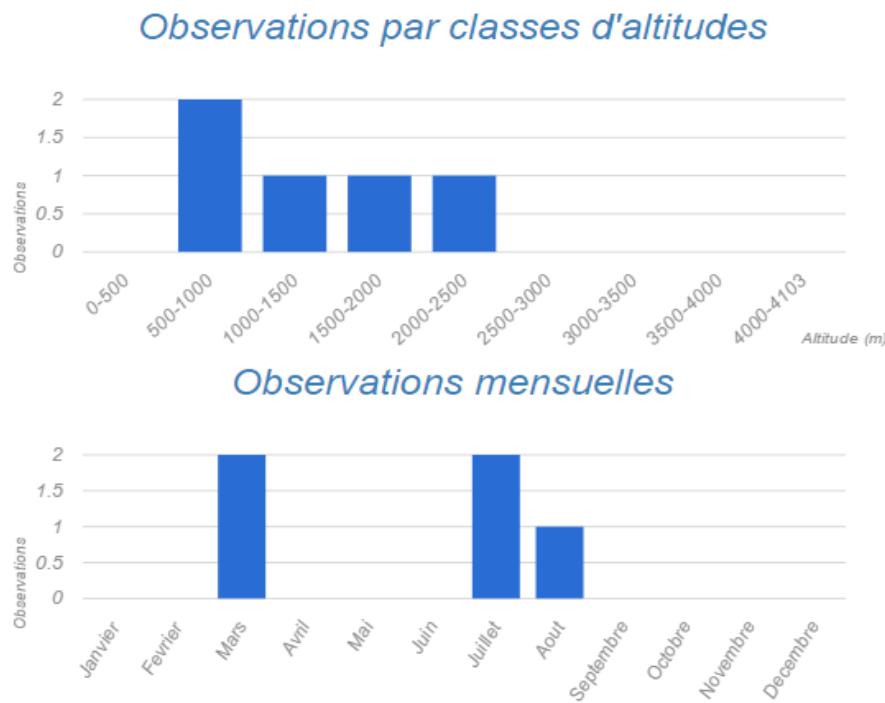
Enfin en bas de la carte, un « slider » permet de filtrer les observations en fonction de leur date. Seule les observations appartenant au pas de temps défini apparaissent. On peut ainsi visualiser par exemple seulement les observations de l'année en cours, ou celle de la dernière décennie .

- Le bloc statistique

La fiche espèce contient deux graphiques issus de la base de données des observations : un pour les altitudes d'observations, et un pour la mensualité (figure 23). Ces données sont pré-calculées dans des vues matérialisées pour améliorer les performances (« `vm_observations_mois` » et « `vm_observations_altitudes` »)

Ces graphiques ont été construits avec la librairie MorrisJS et D3 qui permettent de générer des balises SVG dans la page HTML.

Les classes d'altitudes sont construites dynamiquement à partir des plages remplies par l'administrateur dans la table « `bib_altitudes` ».



*Figure 23: Graphique d'observations d'altitudes / graphique d'observation mensuelle /*

### 5.2.2 La partie « manuelle » de la fiche espèce

La fiche espèce comporte également tout une série d'informations qui sont renseignées manuellement par les agents de la structure.

C'est l'application TaxHub qui centralise la saisie de toutes ces données. Mon stage ne portait pas sur le développement de cette application, mais j'ai néanmoins participé à la modélisation de la base de données qui accueille les données se rapportant à l'atlas dans TaxHub (voir le [ticket](#) GitHub traitant de cette problématique). Comme je l'ai évoqué dans la première partie, cette application a dans un premier temps été développée pour simplifier la gestion de la base de donnée des taxons. Elle a ensuite évoluée pour s'adapter aux besoins de l'atlas. On peut aujourd'hui saisir tous les caractéristiques d'une espèce affichée dans l'atlas : à savoir une description, un commentaire personnalisé, le milieu de l'espèce et sa chorologie (distribution spatiale à grande échelle du taxon), mais également des médias rattachés à une espèce : photos, vidéos, enregistrements sonores, articles, etc.

Lors de l'installation de l'atlas, l'application TaxHub sera fourni permettra d'administrer le contenu de l'application.

- Le bloc « information sur l'espèce »

Dans ce bloc on retrouve des informations complémentaires sur l'espèce sous forme de 5 onglets : la description générale de l'espèce, un commentaire libre pouvant raconter un anecdote, le milieu et la répartition à l'échelle mondiale (figure 24). L'onglet synonyme qui renseigne les noms associés à l'espèce est lui calculé automatiquement à partir de la vue « `vm_taxref` ».

### Informations sur l'espèce

Description    Commentaire    Milieu    Répartition    Synonymes

Plante à fleurs de couleur **rose** à mauve, à petites feuilles triangulaires étroitement imbriquées les unes sur les autres.

Espèce proche : Saxifrage tronquée (*Saxifraga retusa* Gouan)

Figure 24: Le bloc « information sur l'espèce »"

- Le bloc médias

Le bloc médias apparaît éclaté en plusieurs entités sur la fiche espèce.

En haut à droite se trouve un carrousel de photos de l'espèce, au clic : celle-ci s'affiche en plein écran dans une lightbox (figures 25 et 26).

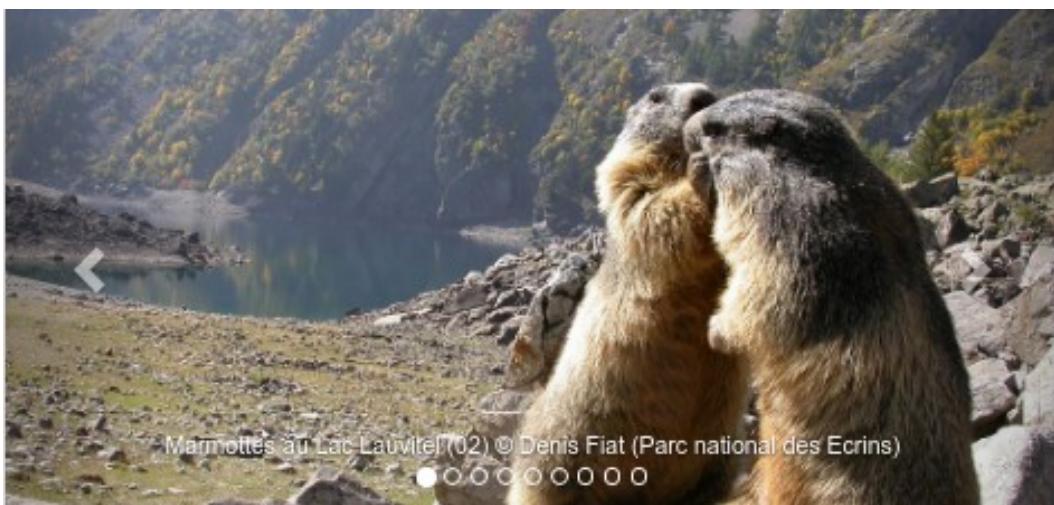


Figure 25: Le carrousel sur les photos du taxon

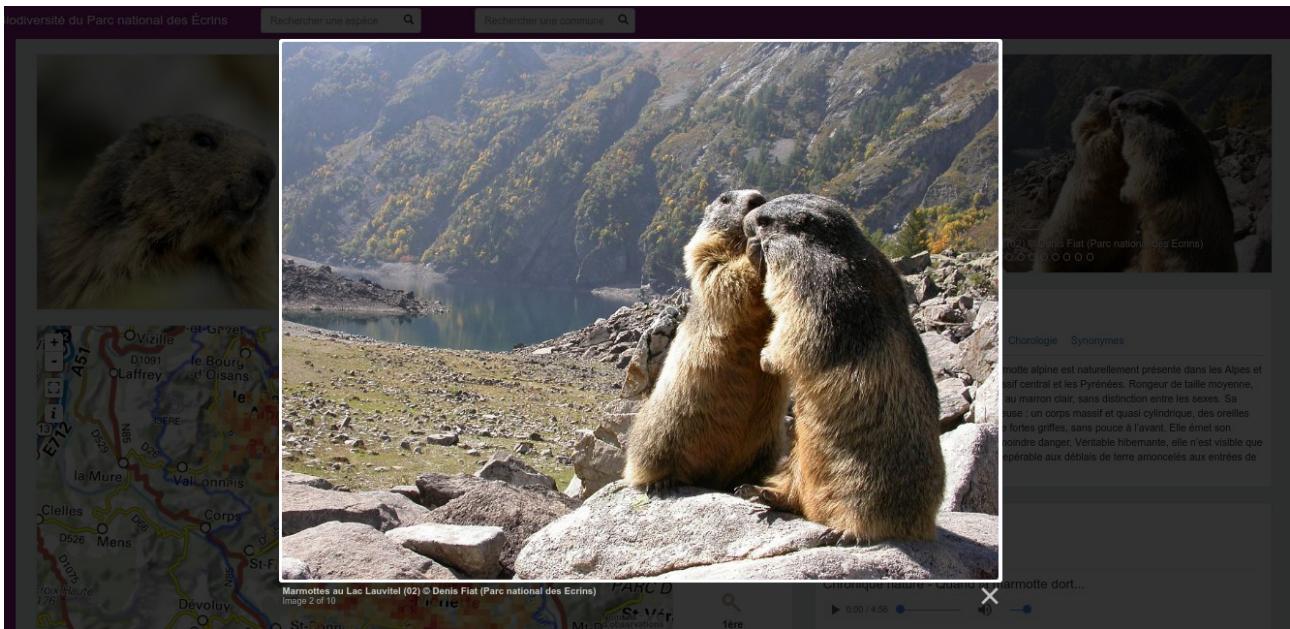


Figure 26: La photo affichée en plein écran dans une « lightbox »

On retrouve ensuite le bloc vidéo et audio.

Dans la base de données il est possible de renseigner des vidéos en local au format .mp4, ou alors des vidéos hébergées sur les plate-formes Youtube, Daylimotion et Viméo. Dans le premier cas, c'est la balise <video> HMTL 5 qui est utilisée pour afficher la vidéo, dans les autres, on génère une balise « iframe » à partir du code fourni par les plate-formes d'hébergement.

De la même manière des sons d'enregistrement des espèces ou des interviews peuvent être intégrés, avec des fichiers en local ou des liens d'enregistrements hébergés en externe. Ils sont rendus disponibles grâce à la balise HTML <audio>.

### 5.3 La fiche communes

L'atlas intègre une page pour chacune des communes où des observations ont été recensées. Sur celle-ci on retrouve un petit bloc statistique qui résume l'état des observations sur la commune (nombre d'observations, nombre d'espèces différentes et nombre d'observateurs).

Dessous se situe une liste de toutes les espèces observées sur la commune avec une série de renseignements (nom commun et nom scientifique, statuts de protection, nombre d'observations, date de la dernière observation, et un lien vers la fiche espèce).

La fiche commune est associée à une carte des observations : au chargement de la page on affiche les 100 dernières observations de la commune. On peut ensuite cliquer sur une espèce pour voir toutes les observations du taxon réalisées sur le territoire de la commune (figure 27).

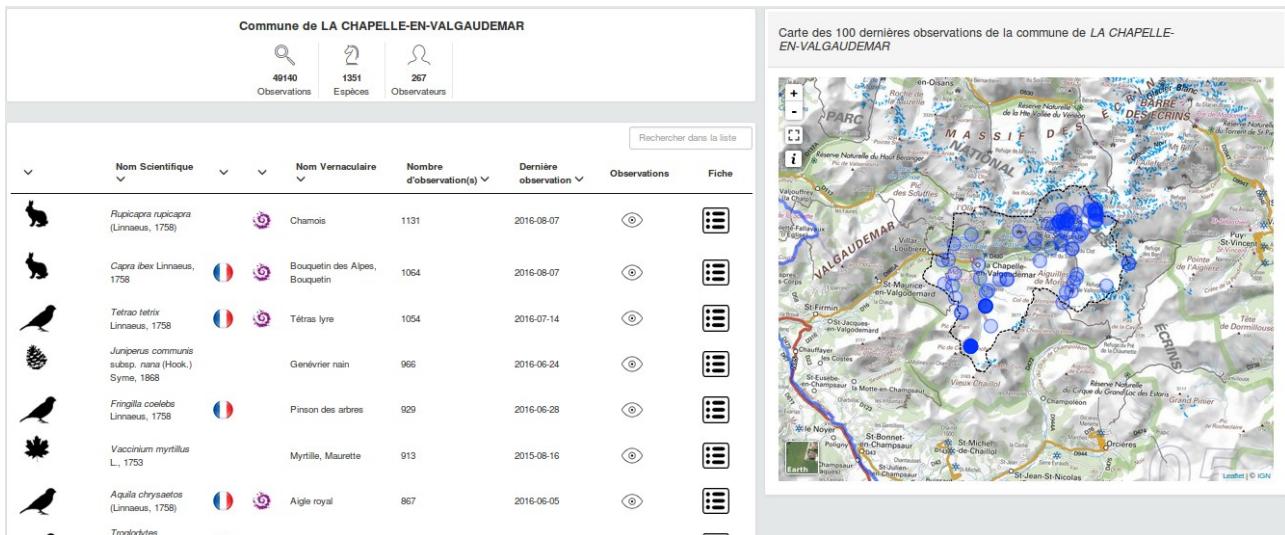


Figure 27: Aperçu de la fiche commune

#### 5.4 La fiche du rang taxonomique

De la même manière que sur la fiche commune, la fiche du rang taxonomique fournit une liste de toutes les taxons fils observées sur le territoire de la structure pour le rang sélectionné. Il existe donc une fiche par rang taxonomique.

Le module de liste utilisé pour la fiche commune et la fiche du rang taxonomique a ainsi été factorisé en un seul [fichier HTML](#) que l'on insert dans chacun des templates.

L'exemple ci-dessous (figure 28) montre la liste des taxons observés pour la famille des « bovins ». On peut ensuite remonter la hiérarchie taxonomique et afficher la liste pour les *certartiodactyla* et les *mammifères*.

Enfin, on peut rechercher une espèce dans la liste, trier par ordre alphabétique, par nombre d'observations, etc.

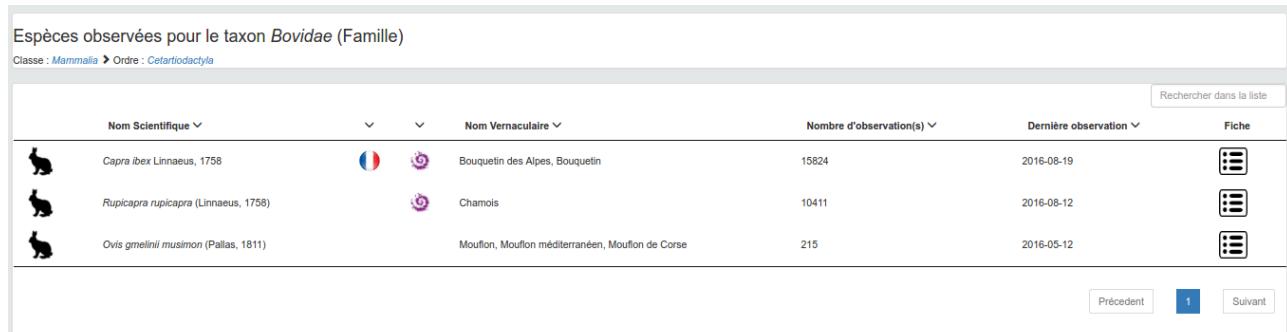


Figure 28: Aperçu de la fiche « rang taxonomique »

## 6. La phase de test

### 6.1 Test de la générnicité de l'application

Afin de tester la générnicité de l'application, nous avons récupéré la base de données des observations du Parc national des Pyrénées, un structure qui n'utilise pas GeoNature et qui possède une architecture de base de données relativement différente de celle du PNE.

A partir de leur base de données des observations nous avons construit les différentes vues-matérialisées nécessaires au fonctionnement de l'application, grâce au fichier « *atlas.sql* » qui regroupe l'ensemble des requêtes SQL nécessaire à l'installation de la base de données. Puis nous avons « branché» cette nouvelle base de données au code de l'application.

Cette procédure s'est révélée très utile puisqu'elle a mis en lumière toutes les spécificités de nos bases de données ainsi que les erreurs de générnicité que j'avais réalisé dans le code et qui n'était pas adaptable à une autre base.

Les tests ont notamment permis de faire remonter des problèmes d'imports Python qui empêchaient d'avoir deux « *atlas* » sur le même serveur. L'ensemble des imports ont donc été remaniés et réécrit en « relatif ».

Cela a également permis de repenser le processus d'installation pour les non-utilisateurs de GeoNature.

L'ensemble de la procédure a été tracé, et nous avons corrigé les différents bugs rencontrés (chemins écrit en dur, nécessité de passer en paramètres certaines variables, code à réécrire etc.)

Après ces corrections, un second atlas faune-flore fonctionnel est né grâce à la base de donnée des Pyrénées (figure 29) !

The screenshot shows the homepage of the 'Atlas du Parc national des Pyrénées'. At the top, there's a green header bar with the text 'Biodiversité du Parc national des Pyrénées', search bars for 'Rechercher une espèce' and 'Rechercher une commune', and a 'A propos' link. Below the header, a large banner features a photograph of several ibexes on a rocky mountain slope. To the left of the banner, a sidebar contains text about the atlas's purpose and history, mentioning it was created in 1973. The main content area is titled 'L'atlas du Parc national des Pyrénées' and displays four summary boxes: '64581 observations' (with a magnifying glass icon), '6438 espèces' (with a leaf icon), '121 communes' (with a building icon), and '20 photos' (with a camera icon). Each box has a corresponding search bar below it.

Figure 29: Aperçu de l'application générée à partir de la base de données du Parc national des Pyrénées

## 6.2 Test de performances

Après les tests de généricté, nous avons également testé les performances de l'application. Le chargement d'une fiche espèce était en effet relativement lent (notamment sur les espèces ayant beaucoup d'observations).

Avant cette phase d'analyse, l'ensemble des traitements étaient générés côté serveur, ce qui laissait une « page blanche » pendant relativement longtemps avant l'affichage. C'est à ce moment que nous avons décidé de réaliser l'API (permettant de charger des informations côté client), notamment pour les données d'observations (affichées sur la cartes) et pour la liste des espèces du module de recherche.

Le chargement de la page s'est largement accéléré (passant de 6 à 1 seconde pour certaines espèces), et les données additionnelles sont chargées au moment où l'utilisateur commence à découvrir la page.

## 6.3 Les tests à réaliser

Un série de tests restent encore à réaliser. Depuis un semaine, le travail d'affichage « responsive » a été amélioré : les fiches espèces et la page d'accueil s'affichent correctement sur format mobile et tablette. Un travail reste à réaliser sur les fiches communes et des rangs taxonomique.

Enfin un phase de test utilisateur sera menée durant le dernier mois de stage pour améliorer l'ergonomie et corriger les derniers problèmes.

## Conclusion et perspectives

L'application répond aujourd'hui en une grande partie aux attentes fixées en début de stage.

Au niveau du contenu, l'atlas comporte en premier lieu la « fiche espèce » qui permet de remonter l'ensemble des informations disponibles sur un taxon. Les espèces emblématiques du Parc ont fait l'objet d'un travail particulier concernant le contenu manuel, et on dispose aujourd'hui d'une vingtaine de fiches « complètes ». Un travail considérable sur le contenu reste à réaliser par les agents du parc et pourra faire l'objet d'un projet commun de l'établissement.

Durant cette deuxième partie de stage, un travail important a été réalisé sur la page d'accueil afin de la rendre dynamique. Cette page qui devait être le « portail d'actualité » des observations du Parc national des Écrins est aujourd'hui fonctionnelle. L'entrée « géographique » et grand public de l'atlas a été renforcé avec la réalisation d'une « fiche commune », et une dimension plus technique a été développée avec la fiche du rang taxonomique. Enfin une galerie photos a été développée dans un dernier temps permettant de visualiser toutes les photos d'espèces disponible dans l'atlas.

Sur la générativité, les phases de tests ont montré que l'atlas était modulaire et qu'il pouvait s'adapter à une base de données provenant d'une autre structure. Un travail, sur la documentation et la procédure d'installation a été réalisé durant les dernières semaines (<https://github.com/PnEcrins/GeoNature-atlas/blob/master/docs/installation.rst>).

D'un point de vue personnel, ce stage m'a permis de développer de nombreuses nouvelles compétences. J'ai énormément appris, autant sur la partie architecture et développement web, que sur la partie base de données, que j'ai pu plus largement explorer durant cette seconde partie de stage (modélisation de base de données, gestion de BDD spatiales). J'ai également eu la chance de suivre et participer à ce projet de sa conception (définition du cahier des charges), à sa réalisation (choix des technologies et de l'architecture, développement), ce qui fut très enrichissant.

Pour finir, plusieurs perspectives s'annoncent pour l'atlas. Aujourd'hui de nombreuses structures disposent de base de données faune-flore (Parc nationaux et régionaux, conservatoires botanique, réserves naturelles). Les données issues de ces structures sont aggrégées au niveau régional et national par les SINP (Système Information sur la Nature et les Paysages) et l'[INPN](#) (Inventaire National du Patrimoine Naturel). L'atlas est lui un outil qui pourrait permettre à de telles structures de valoriser leurs données et leur ressources au niveau territorial, afin de leur donner une meilleure visibilité.

À l'image de l'application « [GeoTrek](#) », (développée sous l'impulsion du PNE) qui a été déployée dans plus de 30 structures, le PNE souhaite encourager le déploiement de l'atlas afin de faire bénéficier de cet outil générique, mais également de mutualiser les ressources pour de futurs développements.

Plusieurs structures contactées sont aujourd'hui intéressées pour déployer l'atlas... à suivre.

# Index des illustrations

Figure 1: Carte de localisation du PNE et de ses secteurs.....	6
Figure 2: La chaîne de travail du SI : du recueil à la consultation des données (document interne). On distingue en vert la chaîne de travail actuelle, et en blanc l'ancienne.....	8
Figure 3: Architecture de la BDD faune-flore du PNE (document interne).....	10
Figure 4: Aperçu de l'application "Synthèse" de GeoNature.....	11
Figure 5 : Capture d'écran de l'application SILENE.....	13
Figure 6: Schéma d'une application web isomorphique.....	17
Figure 7: Schéma d'une application génération des templates côté serveur.....	18
Figure 8: Illustrations des technologies utilisées.....	19
Figure 9: Schéma du mécanisme "foreign data wrapper" mis en place. La base mère est clonée en une base fille qui ne contient pas directement les données. Des vues matérialisées permettent de stocker en dur dans un schéma nommé « atlas », les données nécessaires. Chaque nuit, un mécanisme de « refresh » recrée les vues matérialisées avec les nouvelles données de la base mère.	21
Figure 10: Modèle logique de donnée de l'atlas.....	26
Figure 11: Exemple de routing avec Flask.....	28
Figure 12: Le bloc de présentation de la page d'accueil.....	30
Figure 13: Le bloc de statistiques de la page d'accueil.....	31
Figure 14:Le bloc cartographique de la page d'accueil.....	31
Figure 15: La galerie photos des espèces les plus observées du moment.....	32
Figure 16: Aperçu global de la fiche espèce.....	33
Figure 17: Schéma illustrant l'arbre de la hiérarchie taxonomique.....	34
Figure 18: Fonction permettant de calculer tous les fils d'un taxon.....	35
Figure 19: Aperçu de la fiche d'identité du taxon « Marmotte » de la fiche espèce.....	36
Figure 20: La carte des observations d'un taxon en mode maille.....	37
Figure 21: La carte des observations en mode cluster (à gauche) et en mode point (à droite).....	37
Figure 22: Affichage des observations en mode maille.....	38
Figure 23: Graphique d'observations d'altitudes / graphique d'observation mensuelle / .....	39
Figure 24: Le bloc « information sur l'espèce »".....	40
Figure 25: Le carrousel sur les photos du taxon.....	40
Figure 26: La photo affichée en plein écran dans une « lightbox ».....	41
Figure 27: Aperçu de la fiche commune.....	42
Figure 28: Aperçu de la fiche « rang taxonomique ».....	42
Figure 29: Aperçu de l'application générée à partir de la base de données du Parc national des Pyrénées.....	43

## **Annexes**

### Annexe 1 : Fiche de proposition de stage

#### **CONTEXTE :**

Le Parc national des Ecrins est un établissement public de l'Etat. Il comprend une centaine d'agents, répartis sur 8 sites : le siège à Gap (05), 2 secteurs dans le département de l'Isère et 5 secteurs des Hautes-Alpes.

Le stage est placé au sein du pôle Système d'informations, lui même rattaché au service scientifique.

Le service a pour mission de définir la politique scientifique de l'établissement et de conduire les actions du Parc national en matière d'ingénierie écologique. A ce titre, il est en charge de l'élaboration des protocoles, de l'administration des données, de la restitution et de la diffusion des informations.

Le pôle Système d'informations est composé d'un géomaticien, chef du pôle SI, d'un administrateur systèmes et réseaux et d'un développeur web/bases de données.

Le pôle SI a développé plusieurs bases de données et applications métier pour la gestion des données relatives aux principales missions du parc national (inventaires et suivis faune et flore, inventaire des patrimoines, programme d'animation, gestion et valorisation des sentiers...).

La plupart des applications ont été libérés sous licence libre : <https://github.com/PnEcrins>, <https://github.com/makinacorpus/Geotrek>, <https://github.com/PnX-SI/TaxHub>.

Certaines applications sont déjà destinées au grand public : <http://rando.ecrins-parcnational.fr>, <http://bouquetins.ecrins-parcnational.fr>.

Le Parc national des Ecrins souhaite étendre cette mise à disposition des données en mettant en ligne un atlas de la faune et de la flore.

Celui-ci s'appuiera sur les données présentes dans l'application GeoNature : <https://github.com/PnEcrins/GeoNature>.

Il sera publié sous licence libre, générique et documenté pour pouvoir être déployé par les autres utilisateurs de GeoNature (parcs nationaux, conservatoires, associations...).

#### **THEME :**

Développement web d'un atlas dynamique de la flore et de la faune du Parc national des Ecrins.

#### **ENCADREMENT :**

Camille MONCHICOURT (Parc national des Ecrins) / Responsable du pôle Système d'Informations / Géomaticien

Le parc national accueille chaque année quelques stagiaires. Ces stages doivent avoir un caractère obligatoire dans le cursus universitaire. Ils donnent lieu à une convention tripartite entre l'établissement d'enseignement, le stagiaire et le parc national. Ils doivent obligatoirement faire l'objet d'un rapport de stage dont le contenu sera précisé au début et en cours de stage.

Il est indispensable qu'un enseignant apporte son concours au suivi du stagiaire avant et durant le stage.

## **OBJECTIFS DU STAGE :**

### **Principaux :**

Dans le cadre de la mise à disposition des données et du partage des connaissances acquises par le parc national depuis sa création en 1973, les observations faune et flore constituent une importante partie de ces données (plus de 500,000 observations).

L'objectif du stage est de développer en web et de déployer un atlas de la faune et de la flore interrogeant dynamiquement les données stockées dans l'outil métier GeoNature.

Le stagiaire réalisera une

- Analyse des solutions existantes
- Définition des besoins et des fonctionnalités
- Proposition d'architecture
- Maquettage et ergonomie
- Développement
- Documentation et publication sur Github

### **Secondaires :**

- Le stagiaire participera à la définition de l'architecture globale du système d'informations du parc national et de sa stratégie de développement.
- Le stagiaire pourra être amené à travailler sur d'autres projets de développement WEB.

## **LIEU :**

La résidence administrative du stagiaire est fixée au siège du Parc national des Ecrins (Gap).

## **PROFIL et COMPETENCES REQUISSES :**

**Formation en développement WEB ou géomatique.**  
**Bac +3 ou bac +5**

### **Compétences et connaissances requises :**

- Maîtrise des langages HTML, PHP, javascript, CSS et SQL
- Maîtrise en ergonomie et architecture web
- Maîtrise des concepts du développement
- Connaissance de Bootstrap, AngularJS, Git
- Connaissance d'un framework PHP (Symfony ou autre)
- Connaissance de l'architecture REST
- Connaissance en base de données (PostgreSQL/PostGIS de préférence)
- Connaissance en administration et sécurité de serveurs web linux
- Connaissance appréciée en SIG (QGIS ou autre)
- Intérêt pour les thématiques environnementales et naturalistes
- Rigueur
- Aptitude à travailler en autonomie et en équipe.

## **INFORMATIONS PRATIQUES :**

**Indemnité de stage :** Selon le barème légal en vigueur.

**Durée :** 5 ou 6 mois à partir de février 2016.

## Annexe 2 : Compte rendu réunion atlas - thématique

Date, lieu : 06/06/2016, Charance

Objet : L'atlas web faune et flore du PNE : Quel public, quel contenu, quel ressources ?

### \*\*\*\*\* PUBLIC CIBLE \*\*\*\*\*

Après rappel du contexte et des objectifs du projet (SIT, outils existants, SINP, CS, CA, CODIR), consensus sur 2 publics cibles : grand public (curieux / visiteurs) et naturalistes  
Ces deux publics impliquent 2 voire 3 entrées:

- une entrée botanique/thématique: recherche par la taxonomie
- une entrée plus géographique / territoriale: recherche par commune / secteur
- une entrée par l'image.

Les approches par détermination (couleur, forme...) pourront être envisagées dans un second temps (pas réalisable à l'heure actuelle).

Pour les autres publics : rediriger vers d'autres sites qui proposent de télécharger ou de participer (SILENE, Bdflore 05, LPO etc.)

### \*\*\*\*\* CONTENU DE L'ATLAS\*\*\*\*\*

Rappel des contenus générés automatiquement et manuellement pour chaque espèce

#### Automatiquement :

- Noms et taxonomie
- Liens vers INPN (statuts de protection etc.)
- Graphique altitude et phénologie (penser à un seuil de pertinence du nombre d'obs: fiches avec peu d'observation : données potentiellement erronées)
- Carte de répartition : consensus sur le fait de faire une carte par maille ou de chaleur à petite échelle et à l'observation à grande échelle (voir au cas par cas pour les espèces sensibles)  
=> Important : pour les deux derniers points, rappeler que ces données sont des observations et non pas des inventaires exhaustifs.

Pour la génération automatique : abandon de l'approche par milieu et de la carte de probabilité de présence: nous ne disposons pas des données nécessaires pour que ce soit pertinent.

#### Manuellement :

- Description : courte et simple
- Champs commentaire / contextualisation
- Chorologie à une échelle plus large (plante méditerranéenne, alpine etc.) à dire d'expert. Possibilité d'afficher une petite carte.
- Habitat / milieu à dire d'expert, basé sur une liste prédéfinie
- Photos : une photo fixe pour l'espèce + une galerie de photos

Autres points sur le contenu :

- Gestion des photos dans la photothèque Arjaris ? Ajouter le cd\_nom dans la photothèque  
Faire un inventaire des photos existantes dans la photothèque ou repartir de zéro ?
- Page d'accueil : tableau de bord de l'actualité des observations : almanach, traitements statistiques intéressants pour le grand public
- Se concentrer sur une famille de fiches espèces « exemple » (ex : arbre, fougère) pour générer une dynamique concernant le contenu.
- Réunion maquettage du site 27/06

Rédacteur : Theo Lechemia

Diffusion : SSC, SCOM, DIR

### **Annexe 3 : Compte rendu réunion atlas-technologie**

Date, lieu : Charance, 23/05/2016, conférence téléphonique

Objet : Architecture et technologiques de GeoNature-atlas

## **Ordre du jour :**

- Infrastructure de la BD
- Architecture
- Technologies web

- **Base de données**

- Toutes les nouvelles informations relatives à chaque espèce seront saisies dans TaxHub. On ne crée pas une autre base atlas avec sa propre administration. La partie atlas de TaxHub (photo, description) sera saisie directement dans le formulaire taxon (avec droits spécifiques si besoin).

- Duplication de la base GeoNature pour l'atlas (schéma synthèse et taxonomie) ou création d'un schéma « atlas » dans GeoNature ?

Solution envisagée : duplication de la BDD de GeoNature sur un autre serveur (sous réserve de performance : foreign data wrapper). Uniquement les schémas et tables utilisés par l'atlas (synthèse et taxonomie, à affiner selon les fonctionnalités et besoins de l'atlas). Dans cette base fille intégrée à l'atlas, des champs et des vues supplémentaires pourront être ajoutés pour optimiser les calculs (répartition phénologiques, altitudinale).

- Dans TaxHub : est-ce que l'on utilise la bib\_attributs ou est ce que l'on crée une table spécifique aux champs de l'atlas ( sachant que ceux-ci seront nécessaire pour que l'atlas fonctionne ) ? Dans tous les cas il est acté que l'atlas interrogera des vues et pas directement les tables pour que chacun ait de la souplesse. Le groupe semble préférer de garder la générnicité de TaxHub en utilisant la table bib\_attributs.

Interrogations : Comment l'atlas saura dans quel attribut il retrouve la description ? Comment l'atlas pourra ressortir tous les taxons d'un milieu, s'il n'a pas une bib\_milieux ? Comment gérer des droits spécifiques pour les champs de l'atlas ?

- Solution de générer des vues pour les besoins de l'atlas : l'API se servira directement dans ces vues. Avantage : chacun peut « personnaliser » ses vues en fonction des besoins et des différences dans les tables. L'ensemble des vues seront regroupées dans un schéma spécifique « atlas », ce qui laisse la possibilité à terme de les remplir avec autre chose que GeoNature.

- Question de la sensibilité des données : est ce qu'on le gère au niveau du taxon ou au niveau de l'observation ? Le PNC souligne que juridiquement, c'est plutôt au niveau de l'observation, mais pour autant ce n'est pas à l'observateur de le décider au moment de la saisie. La solution serait de rajouter un champs dans la synthèse de GeoNature qui est rempli par l'administrateur. Cette information permettra à l'atlas de totalement masquer une observation ou de la dégrader.

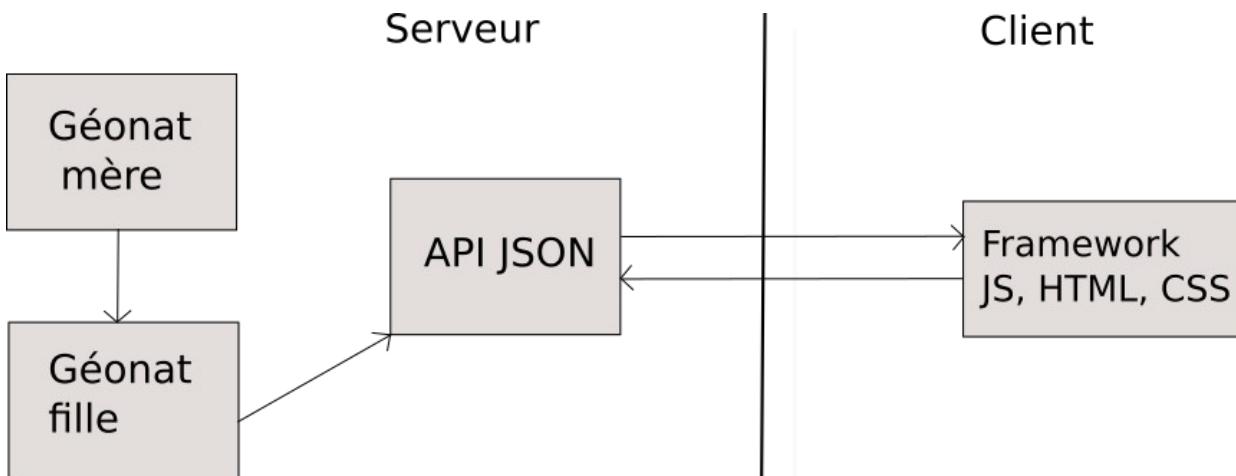
- Faire un point sur les fonctionnalités de l'atlas pour préciser les contenus de l'API.

- **Architecture**

Point abordé rapidement durant la réunion.

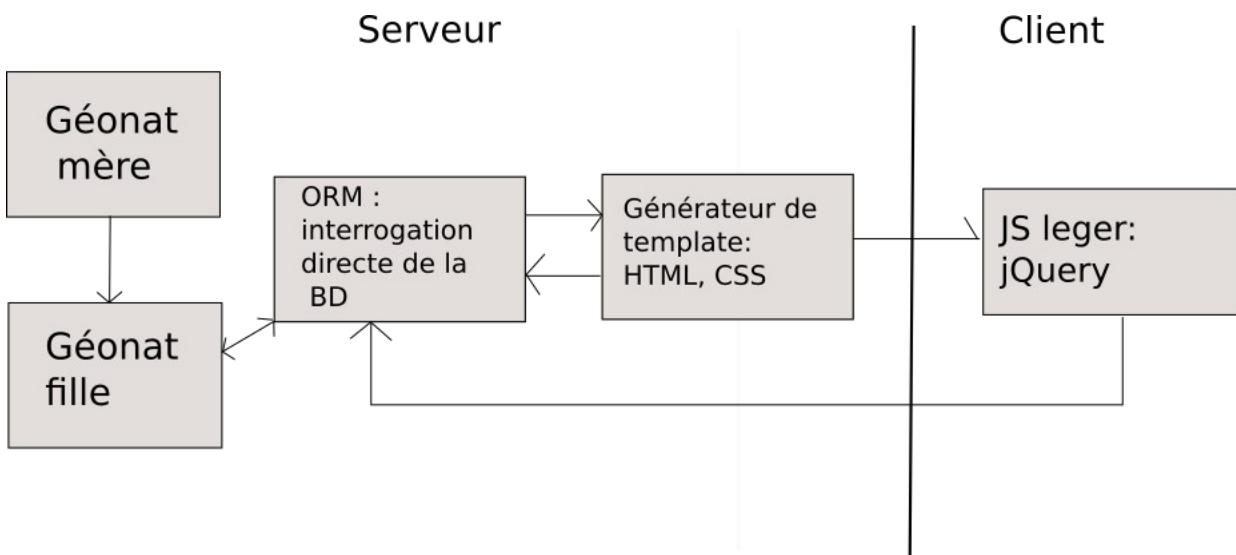
Trois solutions d'architectures sont envisagées :

- 1ère solution : Architecture de TaxHub ou GeoTrek



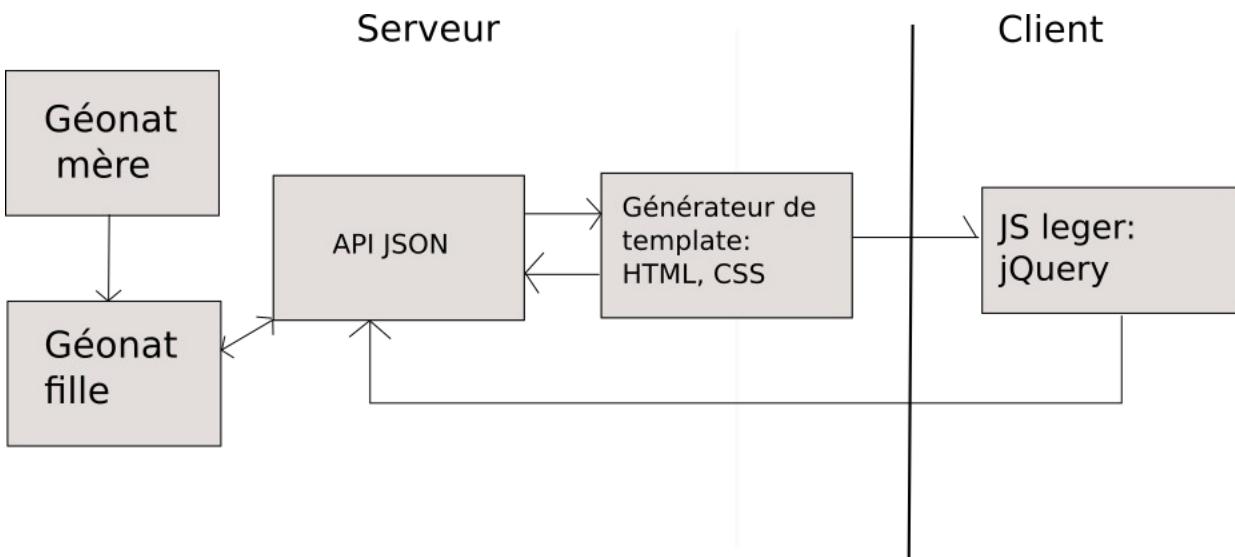
L'API se sert dans la base de données fille et génère des fichiers Json. Pour gagner en performance, ces fichiers peuvent être générés chaque nuit en dur. Le côté client, appelle cet API pour générer les pages HTML du côté front. Nécessite un framework JS lourd comme AngularJS ou React.

- 2ème solution :



Le framework serveur se sert directement dans la BD (ORM). A chaque requête du client, la BD est interrogée, et les pages HTML sont générées par le moteur de templates côté serveur. Possibilité de gérer les réponses des requêtes par de la mise en cache. Côté client, du JS léger du type Jquery pour les animations du site.

- 3eme solution :



Architecture semblable à la 2ème solution, avec génération des pages HTML côté serveur. Différence : le moteur de templates sollicite une API qui renvoie des fichiers en JSON en fonction de la requête. Possibilité de stocker chaque nuit en dur les réponses de l'API pour des questions de performance. Même solution légère côté client JS.

- **Technologies web**

Retour sur le problème de référencement des applications web rendues côté client, du type AngularJS (faisable mais compliqué car ce sont des frameworks fait pour générer des applications web singlepage).

De plus ce type de framework JS, sont davantage appropriés à un des « applications web » avec de grosses interactions côté client, ce qui n'est pas notre cas.

Pour le projet, une solution de génération des pages HTML côté serveur paraît plus adaptée. Idée initiale d'utiliser Symfony 2 pour cela.

Le PNC, de son côté ne souhaite pas développer de nouvelle application avec Symfony : trop lourd pour nos besoins. Elle propose de migrer vers Python : plus simple et plus adapté aux fonctionnalités « spatiales ». Elle propose le « micro-framework » Flask. Ce n'est pas un framework complet du type Django : il est plus souple et plus modulaire, avec une communauté et un support important.

Nécessite de se former à Python et à ce framework dans la durée du stage. Pose la question maintenabilité du projet si Théo se forme tout seul sur Python.

Rédacteur : Théo Lechémia

Diffusion : SI interparcs