

## Technology Services

**INERIS**



**Audit Geod'air**

#### Suivi des modifications

Version	Rédaction	Description	Vérification	Date
0.1	Jean-François MONY	Initialisation		28 octobre 2016
0.2	Jean-François MONY	Ajout des éléments d’analyse du site internet		02 novembre 2016
1.3	Jean-François MONY / Richard AFFI	Compléments		
1.4	Jean-François MONY / Richard AFFI	Version finalisée sur constats		
1.5	Jean-François MONY / Richard AFFI	Prise en compte des remarques clients Ajout des préconisations		05 Janvier 2017
1.6	Jean-François MONY / Richard AFFI	Prise en compte des remarques clients Ajout de la suite « Elastic Slack »		27 Janvier 2017

#### Approbation du document

Organisme ou entreprise	Nom (fonction)	Date	Visa
CAPGEMINI	Jean-François MONY		
INERIS			

#### Diffusion du document

Organisme ou entreprise	Nom (fonction)	Nombre	Mode	Pour
INERIS	Jean-Yves CHATELIER		✉	Avis
INERIS	Jonathan BARRIER		✉	Avis
INERIS	Christophe DECROMBECQUE		✉	Avis

## 1 SOMMAIRE

<b>1</b>	<b>Sommaire .....</b>	<b>3</b>
<b>2</b>	<b>Introduction .....</b>	<b>5</b>
2.1	Contexte et Objectifs .....	5
<b>3</b>	<b>Méthodologie et périmètre.....</b>	<b>6</b>
3.1	Méthodologie.....	6
3.2	Périmètre.....	6
3.3	Rapport d’étonnement .....	6
<b>4</b>	<b>Etat général de la documentation .....</b>	<b>8</b>
4.1	Documents fonctionnelles .....	8
4.2	Documents techniques .....	8
<b>5</b>	<b>Analyse de l’architecture générale.....</b>	<b>12</b>
5.1	Architecture physique.....	12
5.2	Briques applicatives.....	12
5.2.1	<i>L’absence des composants Bonita et OpenLDAP .....</i>	<i>13</i>
5.2.2	<i>L’utilisation de l’ESB Fuse (ServiceMix) .....</i>	<i>14</i>
5.2.3	<i>Le choix de MongoDB .....</i>	<i>15</i>
5.2.4	<i>Couche d’accès aux données .....</i>	<i>16</i>
5.2.5	<i>Monitoring .....</i>	<i>17</i>
<b>6</b>	<b>Analyse détaillée .....</b>	<b>21</b>
6.1	Qualité de code générale.....	21
6.2	Bases de données .....	24
6.2.1	<i>Base Postgresql.....</i>	<i>24</i>
6.2.1	<i>Bases MongoDB .....</i>	<i>26</i>
6.3	ETL.....	27
6.3.1	<i>Les Jobs TALEND.....</i>	<i>27</i>
6.3.2	<i>Statistiques périodiques .....</i>	<i>35</i>
6.4	Webservices (ServiceMIX) / ESB .....	36
6.4.1	<i>Service CUD Référentiels .....</i>	<i>36</i>
6.4.2	<i>Services Rapportage .....</i>	<i>39</i>
6.5	BPM.....	41
6.6	Site web .....	41
6.6.1	<i>Analyse statique .....</i>	<i>41</i>
6.6.2	<i>Analyse dynamique .....</i>	<i>45</i>
6.7	Synthèse des constats.....	63
<b>7</b>	<b>Récapitulatif des constats .....</b>	<b>65</b>
7.1	Points forts .....	65
7.2	Points d’amélioration.....	65
7.3	Non conformité.....	67
<b>8</b>	<b>Préconisations .....</b>	<b>70</b>
8.1	Architecture cible préconisée.....	70
8.1.1	<i>Architecture cible en mode page blanche .....</i>	<i>70</i>
8.1.2	<i>Architecture cible préconisée.....</i>	<i>72</i>
8.1.3	<i>Zone « Front » .....</i>	<i>73</i>
8.1.4	<i>Zone « Middle » .....</i>	<i>74</i>
8.1.5	<i>Zone « Back » .....</i>	<i>75</i>
8.2	Approche Incrémentale.....	76
8.2.1	<i>Incrément 1 : Découpler le serveur WEB1 des serveurs DB .....</i>	<i>76</i>
8.2.2	<i>Incrément 2 : Transfert des fichiers depuis les AASQA via l’ESB .....</i>	<i>77</i>
8.2.3	<i>Incrément 3 : Optimisation des bases de données .....</i>	<i>78</i>
8.3	Actions prioritaires pour mise en production rapide.....	79



## 2 INTRODUCTION

### 2.1 CONTEXTE ET OBJECTIFS

---

L’application Geod’Air permet d’intégrer et de centraliser les données d’observation de la qualité de l’air, données produites au niveau local. Elle met ces données à disposition des différents acteurs et supporte les processus de rapportage. Geod’Air maintient également à jour un référentiel du dispositif de surveillance de la qualité de l’air.

Par ses fonctionnalités de rapportage, Geod’air permet également à la France de répondre à ses obligations vis-à-vis des directives européennes sur la surveillance de la qualité de l’air. Elle permettra également à terme de répondre aux obligations de la directive INSPIRE en publiant un certain nombre de services sur les données de la qualité de l’air.

L’application est en production depuis Septembre 2015, mais n’est pas encore largement ouverte aux utilisateurs. Elle comporte environ 20 gabarits avec 4 IHM par objets, soit 80 écrans + 3 écrans de cartographies, 1 écran de données, 1 onglet mesure/cartographie. La volumétrie conservée à 4 ans devrait impliquer une croissance x5 du volume actuel en base de données. Le nombre d’utilisateurs cible est de 100 utilisateurs environ (<10 aujourd’hui).

Confronté à certains dysfonctionnements et en anticipation de la fin de la TMA assurée par ATOS, l’INERIS a souhaité réaliser un audit technique du système.

## 3 METHODOLOGIE ET PERIMETRE

---

### 3.1 METHODOLOGIE

---

La méthodologie et le planning initiaux ont été adaptés afin de prendre en compte les contraintes imposées par ATOS.

La logique initiale de l’audit reposait sur 3 phases principales :

- Une phase de prise de connaissance documentaire et d’échange avec l’INERIS
- Une phase d’analyse intégrant un atelier avec ATOS
- Une phase de rédaction et de restitution

Ces 3 phases étaient prévues pour se chevaucher afin de réduire le délai global de l’audit.

La phase 1 s’est déroulée conformément aux prévisions.

La phase 2 a débuté par une analyse du site sur la base d’un accès standard à ce dernier (sans les sources) et l’audit a été mis en pause, les codes applicatifs n’étant pas fournis.

La phase 2 a repris après 7 mois d’interruption, le code et des accès aux machines de recette ont été fournis et l’atelier d’échange avec ATOS a pu se dérouler.

Des points de suivi et de restitution intermédiaire ont été mis en place avec l’INERIS.

La phase de rédaction a été subdivisée en deux avec la fourniture d’un rapport intermédiaire correspondant à l’analyse et aux constats.

### 3.2 PERIMETRE

---

Cet audit couvre le périmètre suivant :

- Etat général de la documentation de référence pour cette application
- Choix d’architectures techniques
- Audit de conception technique générale
- Revue de code par qualimétrie
- Revue de code par échantillonnage
- Revue du modèle de données (2 bases de données SQL et NoSQL)
- Revue des composants techniques
- Analyse des principaux points de contention
- Cinématique générale de navigation entre les pages
- Architecture d’information
- Ergonomie générale des gabarits

### 3.3 RAPPORT D’ETONNEMENT

---

Le rapport d’étonnement, comme son nom le laisse présager, est une première approche de surface sans analyse.

Il s’agit de tracer les premières impressions, le ressenti lors de la prise de connaissance initiale.

Le périmètre de la conduite de projet ne faisait pas partie du périmètre de l’audit. Cependant le fait que le système Géod’air soit en production tout en n’étant pas ouvert aux utilisateurs dans une version qui repose pour partie sur des spécifications qui n’ont jamais été validées nous a posé beaucoup de questions par rapport aux étapes standards d’un projet informatique.

L’utilisation du site en mode naïf, sans avoir consulté la documentation fonctionnelle au préalable, nous amène à certains constats. Ces constats reposent principalement sur notre expertise dans le domaine

cartographique en général et plus particulièrement sur les sites permettant l’exploitation et la diffusion de données d’inventaire (manuelles ou automatiques) :

- Le premier chargement est très lent et le site connaît des lenteurs sur certaines actions.
- La navigation est peu intuitive et le lien entre la carte et les données attributaire n’est ni très évidente, ni très riche
- Le comportement de la carte est parfois surprenant (changement de niveau de zoom, changement de visibilité des couches) lors des allers / retours entre les données de référence et de mesure.
- Les critères de recherche ne couvrent pas l’ensemble des référentiels disponibles ou des champs disponibles dans le tableau de présentation des mesures (choix métier ?)
- L’interaction avec la carte est pauvre et limitée au filtrage des stations et des zas. On pourrait attendre, sans être exhaustif :
  - Une meilleure interaction entre tableau et carte (sélection / highlight sur les 2 supports)
  - Sélection de stations via la carte
  - Représentation des valeurs de mesure associées aux stations (analyses thématiques)
  - La manipulation des données pour réaliser des agrégations ou des représentations graphiques
  - ....

La présence d’une carte dynamique sur un site ayant toujours un coût non négligeable sur les performances, sa valorisation sur le site ne nous semble en rapport avec son impact sur la navigation.

## 4 ETAT GENERAL DE LA DOCUMENTATION

### 4.1 DOCUMENTS FONCTIONNELLES

---

- DRC-12-120045-13093A CCTP GEODAIR\_VF.pdf

Ce document a permis de mieux comprendre le besoin initial mais n'a pas été analysé lors de l'audit car l'adéquation du produit aux exigences et aux documents contractuels n'était pas dans le périmètre.

- DV13AV - INERIS - GEOD'AIR - réponse technique v1.0.pdf

Ce document a permis de comprendre la réponse initiale d'ATOS au CCTP mais n'a pas été analysé lors de l'audit car l'adéquation du produit aux exigences et aux documents contractuels n'était pas dans le périmètre.

- GEODAIR - SFG - v1.1.pdf

Il s'agit des spécifications fonctionnelles générales produites par l'INERIS : aucun commentaire à faire.

- GEODAIR-MEX\_140206\_Matrice\_Exigences\_Generales\_V1.1.xlsx

Il s'agit du document de suivi et de phasages des exigences élaborées à partir des spécifications fonctionnelles. Nous remarquons l'absence d'éléments de suivi du document (version, destinataires...). De plus, il ne précise pas la version du document des SFG prise en compte.

- GEODAIR-MANUEL-UTILISATEURS-V05.pptx

Ce document est le manuel d'utilisation du site web. Il ne présente cependant aucun cas métier ni la corrélation entre les règles de gestion et les différents écrans et interactions du site.

- UserGuide2\_AQD\_XML\_v3.2.0.pdf et AQD Model\_V1.0.7\_V2.pdf

Il s'agit des documents décrivant le schéma de données pour produire les rapports : aucun commentaire à faire.

### 4.2 DOCUMENTS TECHNIQUES

---

- GEOD'AIR-STG-131001\_Conception Technique Générale V1.0.docx

Il s'agit du document de conception technique générale de la solution Géod'Air. Il présente les composants de la solution Bull Géo Live et leur utilisation pour Geod'Air.

Il souffre d'un défaut de mise à jour. Certains composants décrits comme Bonita ne sont pas utilisés. De plus, il ne donne pas les justifications du choix des composants. Les paragraphes dédiés aux différents modules sont trop détaillés et auraient dû faire l'objet de document séparé (par exemple, les paragraphes "Gestion des données référentielles", "Module de reporting"). En conséquence, le document est trop volumineux.

- GEODAIR-SD-140117\_Conception Technique Détaillée- Introduction & Architecture V1.1.docx



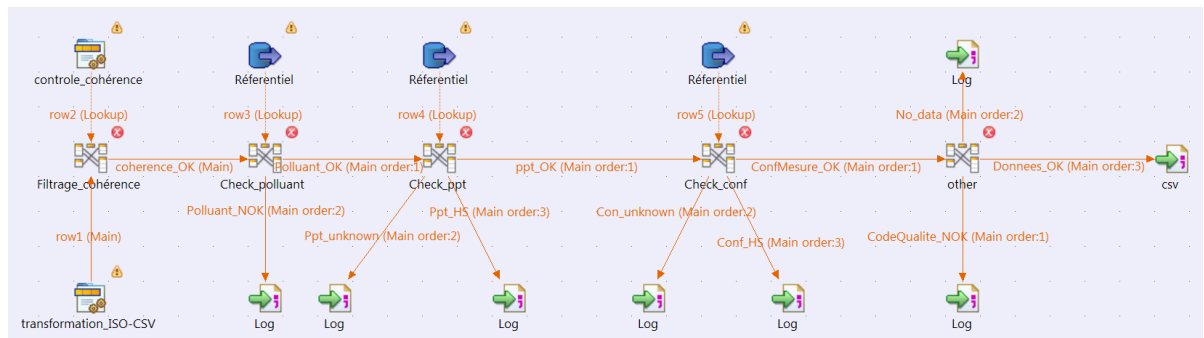
- Document chapeau présentant et référençant la documentation projet de Géod’air
  - Document présentant les principes généraux du système
  - Les documents référencés ne sont pas dans la bonne version
  - Certains documents référencés semblent ne pas exister (GEODAIR-SD-131115\_Conception détaillée Données élaborées)
  - Le document fait référence à des briques inexistantes (Bonita, LDAP)
  - Le document ne porte pas son statut (validé ou non validé)
- GEODAIR-SD-170313\_Conception Détaillée des IHM-V1.8\_final.docx
  - Le document indique que les maquettes d’écran ne sont qu’indicatives. C’est assez surprenant pour un document de spécifications détaillées d’IHM. Nous n’avons pas identifié d’autre document où ce sujet aurait été traité séparément.
  - Le doc n’est pas à jour par rapport au site actuellement mis à disposition (la partie 6.6. Ecrans conformité du parc de mesure est par exemple absente du site)
  - Le document a une approche très statique dans la description des IHM. Il nous semble qu’une approche par use cases / user stories est plus à même de spécifier un site en prenant en compte l’interaction et les usages ciblés.
  - Le document ne porte pas son statut (validé ou non validé)
- GEODAIR-SD-140210-CTD Gestion des Référentiels-V2.6.docx
  - Document de bonne facture et complet.
  - La cohérence du document avec la structure de base actuelle n’a pas été analysée.
  - Le document ne porte pas son statut (validé ou non validé)
- GEODAIR-SD-131206\_Conception Détaillée Module Rapportage V1.51.docx
  - Document de bonne facture et complet.
  - Le document fait mention du document « GEODAIR DCD Gestion des logs, alarmes et rejets » qui ne semble pas exister. Comme indiqué par ailleurs ce document est indispensable pour assurer un monitoring cohérent de l’ensemble du système dont chaque brique produit ses logs.
  - Le document ne porte pas son statut (validé ou non validé)
- GEODAIR-SD-131129\_Conception Technique Détaillée Flux acquisition et production de données V1.3.docx
  - Spécification du module de collecte et d’intégration de données – document de bonne facture globale mais souffre d’un manque de mise à jour (Mention de l’utilisation d’un annuaire LDAP par exemple).
  - Des cas d’erreur nécessitant une reprise sur incident sont identifiés et reportés au DEX qui ne les traite pas.
  - Le paragraphe reprise en cas d’erreur est vide
  - La production des logs est abordée mais reste partielle. Un document dédié à la supervision et à l’agrégation de l’ensemble des logs du système serait nécessaire.
  - Le document ne porte pas son statut (validé ou non validé)
- GEODAIR-SD-131122\_Bilan Données et Procédure Reprise V1.4.pdf

- Document concernant la reprise des données pour intégration dans Geod'air
- Le document était valable au moment de la migration, mais n'est plus conforme à la structure actuelle de la base référentielle
- Le document donne une bonne vision de l'appariement entre données sources et structure cible
- Statut du document faux puisque l'INERIS nous indique que ce document est validé alors que le statut interne du document est « en cours »

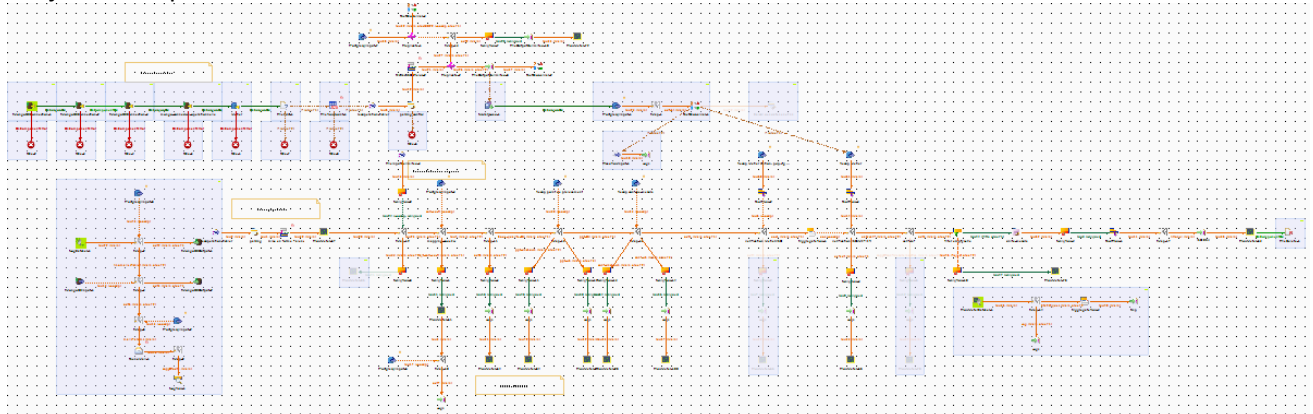
- GEODAIR-SD-151104-Descriptiontraitements des ETL des flux automatiques.docx

Les grands principes décrits dans ce document sont toujours d'actualité. Par contre il y a un décalage important entre les captures du document et la réalité des jobs :

Capture du document



Capture du job correspondant :



De plus les points suivants sont relevés :

- Il n'y a aucune description de la stratégie de log / monitoring
- Il manque une description des logiques de redémarrage sur erreur, de garantie de l'intégrité des données...
- Toute la logique globale est traitée dans GEODAIR-SD-131129\_Conception Technique Détaillée Flux acquisition et production de données V1.3.docx
- Le document ne porte pas son statut (validé ou non validé)

- 20141212 MCD GEOD'AIR - impression.pdf

Le document n'est pas totalement exploitable. Les colonnes de certaines tables ne sont pas affichées. En comparant avec le dump reçu d'ATOS, il manque certaines tables (fos\_\*, rapport\_ue, rapport\_historique, etc...)

- GEODAIR-DAT-131004\_Dossier Architecture Technique\_v1 4\_INERIS.docx

- Le DAT est de bonne qualité et couvre les sujets du périmètre d’un DAT. Il souffre par contre d’un défaut de mise à jour puisqu’il fait toujours référence à des briques qui ont été sorties (Bonita, LDAP).
  - Peu d’éléments donnés sur le monitoring et la supervision des applications
  - Le document ne porte pas son statut (validé ou non validé)
- 
- GEODAIR-DEX-Exploitation et sauvegarde-1.1.docx
    - La stratégie de sauvegarde est indiquée, mais il n’est pas fait mention des procédures de restauration
    - Le document souffre d’un défaut de mise à jour, puisqu’il fait toujours référence à des briques qui ont été sorties (Bonita, LDAP).
    - La stratégie de sauvegarde / rotation des logs techniques n’est pas mentionnée ici.
    - Le document ne porte pas son statut (validé ou non validé)
- 
- GEODAIR\_Doc\_installation\_v0 2.docx

En résumé, il existe une variété de documents couvrant les différentes phases du projet Geod’Air. La complétude et l’état de dernière version de ces documents a été confirmé par l’INERIS et par ATOS. Dans le cadre d’une réversibilité, un gros travail sera nécessaire pour à la fois remettre les documents en phase avec la réalité du système et pour combler les lacunes observées. La première de ces lacunes réside dans la rédaction d’une procédure d’installation complète et détaillée du système.

De plus, nous n’avons trouvé aucun document sur la présentation des sources, l’usine logicielle et des outils de développement peuvent être intégrés dans ce document. Il est aussi vrai qu’il n’existe aucun document sur la stratégie de tests. Ce qui explique aussi le peu de tests trouvés dans le code source.

## 5 ANALYSE DE L’ARCHITECTURE GENERALE

---

### 5.1 ARCHITECTURE PHYSIQUE

---

L’architecture en front / back mise en place présente l’intérêt de protéger la couche de persistance des accès directs depuis l’extérieur. Il en va de même des services mis en place via l’ESB. Tous ces éléments ne sont accessibles que sur des IP locales.

Intrinsèquement, ce choix n’est guère contestable puisqu’il protège le composant le plus sensible du système : les données.

A l’usage, il pose problème puisqu’il interdit à l’INERIS de se connecter via des clients autres que le client applicatif de Géod’air ou le client natif en ligne de commande aux données.

Ce besoin n’est pas une exigence initiale de l’INERIS mais un besoin identifié comme un palliatif aux différents problèmes rencontrés par l’INERIS sur la plateforme Geod’air.

Par ailleurs, en ce qui concerne les problématiques de montée en charge et de haute disponibilité, l’absence de mécanisme de load balancing dans l’architecture physique freine la scalabilité horizontale.

### 5.2 BRIQUES APPLICATIVES

---

Le schéma ci-dessous représente la vision logique de l’architecture telle que l’audit nous a permis de constater :

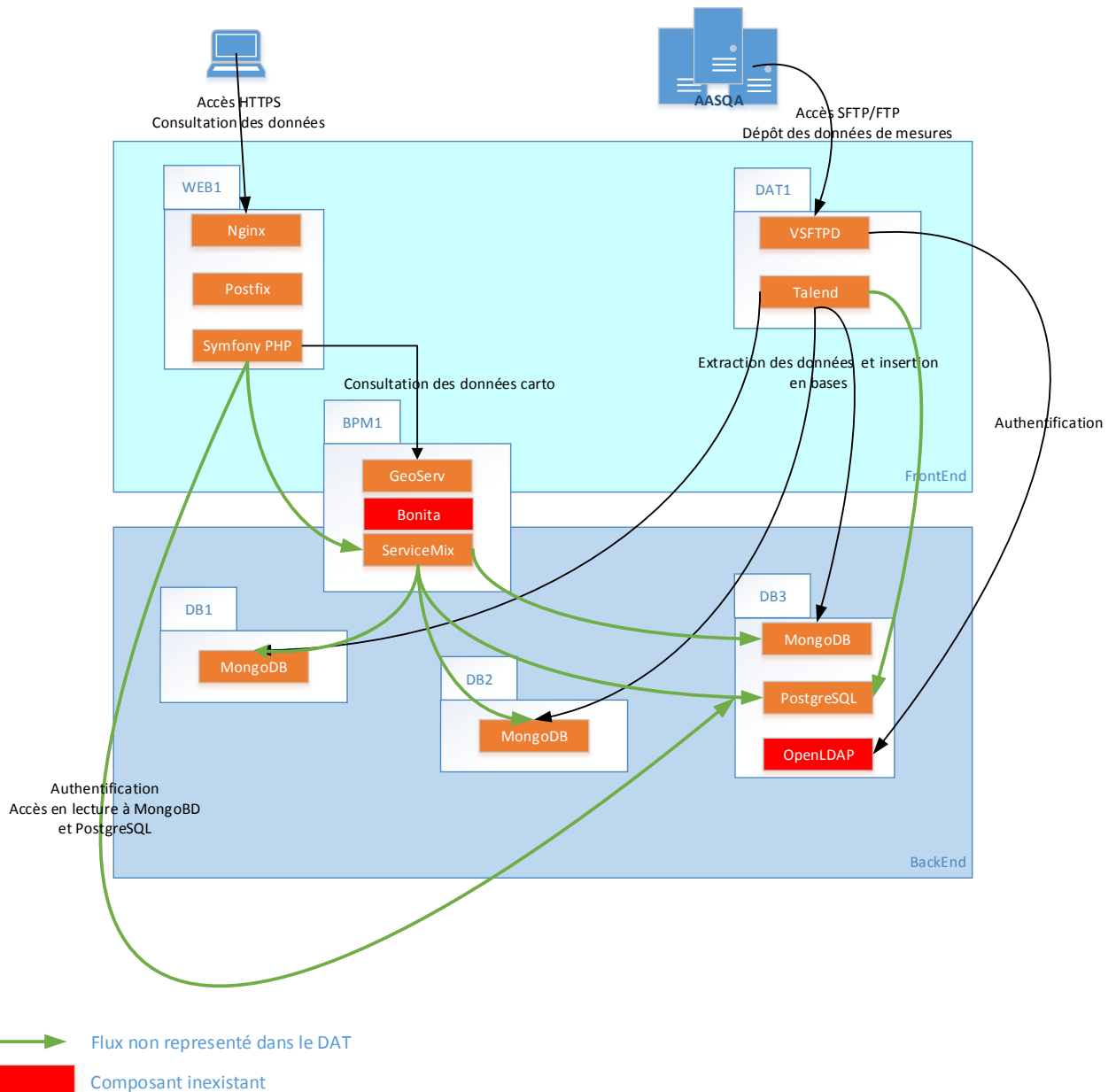


Figure 1: Architecture logique réelle

L'analyse de la vue globale de l'architecture applicative mise en place fait ressortir les points suivants :

### 5.2.1 L'absence des composants Bonita et OpenLDAP

Contrairement à ce qui a été annoncé dans la réponse technique d'ATOS, nous constatons (confirmé également par ATOS) l'absence des composants Bonita et OpenLDAP.

L'absence du BPM Bonita empêche le suivi des processus déclenchés par les événements internes ou externes. A partir des workflows du BPM, les actions suivantes auraient pu être mises en place :

- Mécanisme de « Pull/push » pour synchroniser les données avec les systèmes externes
- Déclenchement des alarmes ou des notifications vers les applications clientes en fonction de règles métiers
- Application de règles métiers pour le traitement des données (workflow de validation des modifications du référentiel)

De plus, l’absence du BPM a conduit à modifier les briques pour compenser et produire les logs.

Par ailleurs, l’utilisation de l’annuaire OpenLDAP aurait permis de centraliser toutes les données des utilisateurs pour gérer l’authentification sur l’IHM et le FTP.

### 5.2.2 L’utilisation de l’ESB Fuse (ServiceMix)

ESB Fuse est un bus de services open source basé sur la plateforme Apache ServiceMix. Il est utilisé comme outil d’intégration de systèmes hétérogènes via l’exposition d’un ensemble de services, l’utilisation de systèmes orientés messages (MOM), le routage intelligent découplant l’expéditeur du message de son destinataire.

Dans l’implémentation actuelle du projet Geod’Air, cet ESB ne sert uniquement qu’à exposer des web services (services de rapportage, services de modification des référentiels). De simples web services exposées via le framework J2EE JAX-WS auraient suffi et rendraient l’architecture plus simple. Le choix de la mise en place de cet ESB est sûrement dû à l’utilisation de la plateforme Bull « GeoLive » comme solution technique.

**Par ailleurs, l’utilité du bus de service perd son sens lorsque la couche de présentation (WEB1) l’outrepasse pour implémenter des règles métiers et accéder aux bases de données.**

De plus, la mise en œuvre d’un bus de service est séduisante et de bonne augure en termes d’urbanisation de la solution.

Dans le contexte de Geod’air, l’intégration d’un ESB aurait de l’intérêt s’il était réellement positionné au carrefour de tous les échanges entre les briques applicatives.

Or, il s’avère que Service mix doit partager le périmètre qui devrait être le sien avec :

- L’ETL pour la pris en charge et l’intégration des données
- Symphony / Sonata qui assure les services de consultation des données
- Du code spécifique (Java) pour le service de production des statistiques

Le périmètre de Service mix se trouve donc réduit

- Aux services CRUD limités à la création, la mise à jour et la suppression, le ‘Read’ étant assuré par Symphony /Sonata
- Aux services de rapportage

Ces services sont analysés dans le chapitre suivant.

Le positionnement d’un ETL ou d’un ESB pour le traitement et l’intégration des données de mesure est une vraie question. Classiquement, l’ETL est dédié à des traitements en masse en mode batch et l’ESB à des traitements plus atomiques en temps réel.

On peut modéliser de façon classique l’arbre de choix technologique suivant :

Alimentation

- Temps réel  
=> ESB
- Batch
  - Volume de données important  
=> ETL
  - Volume de données faible

- Transformations complexes  
=> ETL
- Transformations simples  
=> ESB

On voit aisément ici que le choix n’est pas tranché dans le cadre de Géod’air :

- Les données automatiques pourraient être fournies en temps réel au fur et à mesure de leur production directement par les AASQA. Le volume de chaque dépôt unitaire pourrait donc être considéré comme faible.
- Les données manuelles sont déposées sur un rythme plus aléatoire dans un volume également plus variable. Cependant ce volume reste faible en comparaison du volume des données automatiques
- Les transformations apportées aux données, dans le cadre de l’architecture actuelle sont de 3 types
  - Validation de la conformité du format
  - Validation de la conformité par rapport au référentiel.
  - Calcul des statistiques. Ce n’est pas à proprement parler une transformation des données transmises, mais plutôt une création de nouvelles données. Mais actuellement cette création est directement déclenchée et liée à l’intégration des données.

En complément, on notera que la stratégie d’un ESB est intrinsèquement synchrone. L’ESB expose un service et lorsqu’un flux lui est transmis via ce service, il est directement pris en charge par l’ESB (ceci n’empêche pas des mises en queue de traitement si cela correspond aux principes mis en œuvre dans l’ESB).

Le fonctionnement de l’ETL peut lui être synchrone ou asynchrone en mode batch. Dans le cas de Geod’air, un script sh lancé en démon fait office d’ordonnanceur et lance les scripts ETL en fonction des dépôts sur le FTP et selon un niveau de parallélisme configuré.

On notera également qu’actuellement l’ESB n’est accessible que sur l’IP privée du serveur. Ceci protège le serveur des risques d’intrusion mais impose également des limites dans l’utilisation de l’ESB. Ceci complique (implique a minima la mise en œuvre d’un proxy) en effet l’ouverture des services à l’extérieur comme l’INERIS a prévu de le faire.

### 5.2.3 Le choix de MongoDB

MongoDB est une base de données NoSQL orienté documents. Comme toutes les autres BD NoSQL, son utilité se justifie fortement dans les cas suivants :

- Flexibilité dans le modèle de données
- Capacité à mettre en place rapidement une scalabilité horizontale des données
- Limitation du SQL pour requêter de larges volumes de données
- Pas de contraintes fortes sur les propriétés ACID
- Volumétries de données très importantes et distribuées

Nous constatons que les critères cités précédemment ne s’appliquent pas aux données de la solution « Géod’Air » :

- Le modèle de données des mesures, statistiques et des référentiels évolue très peu.
- La scalabilité horizontale n’est pas un enjeu fort
- Le langage SQL suffit largement pour effectuer les requêtes
- Les propriétés ACID de la base de données sont obligatoires
- La volumétrie de données définie dans le DAT (120 Go au total pour 20 ans de données) n’est pas très importante.

Le choix de MongoDB est donc très discutable et a été contraint vraisemblablement par l’utilisation de la plateforme Bull « GéoLive » pour l’implémentation de la solution « Geod’Air ».

**Constat** : L'utilisation d'une base de données PostgreSQL aurait été largement suffisante comme support de sauvegarde des données de mesures et des statistiques.

### 5.2.4 Couche d'accès aux données

Au vu des flux entre les bases de données et les autres composants, le problème de centralisation de la couche d'accès aux données se pose. Nos analyses montrent que chaque composant (DAT1, WEB1, BPM1), voire plusieurs sous composants (ETL et code de génération des statistiques) d'un même composant possède sa propre couche d'accès aux données. Une modification de structure de table peut potentiellement impacter tous ces composants (Figure 2: Couches d'accès aux données). La maintenance globale est donc fortement alourdie avec un risque important sur le maintien de la cohérence globale.

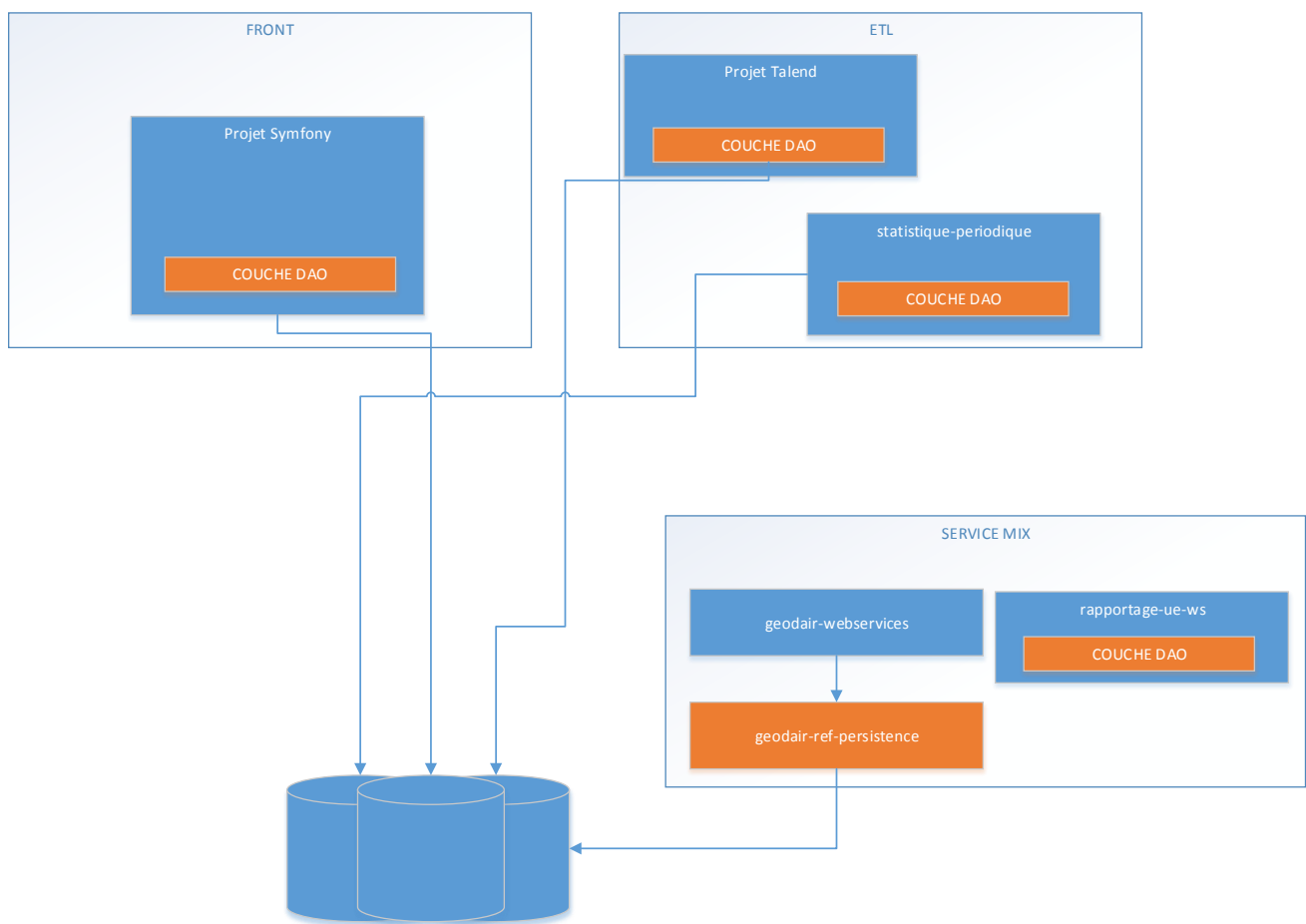


Figure 2: Couches d'accès aux données

Par ailleurs, contrairement à ce qui a été dit à l'atelier de présentation avec Atos, il n'existe pas que des requêtes de sélection dans la couche de présentation (Symfony). Pour l'administration de certaines nomenclatures (Civilité, Densité Unité Urbaine, ...) les requêtes sont directement faites via l'ORM de Symfony.



```
class CivileRepository extends EntityRepository
{
    public function updateEntity($entity, $toEntity, $logger)
    {
        $logger->debug(print_r('entrée Update', true));
        //si modification du code (IE : de la clé primaire)
        if ($entity->getCode() != $toEntity['code'])
        {
            $em = $this->_em;
            //ce morceau de code étant commun à tous les repository des nomenclatures d
            d'être gérée
            //on récupère donc sa classe
            $entityInfo = $em->getClassMetadata(get_class($entity))->getName();
            //création d'une nouvelle nomenclature
            $newCurrentEntity = new $entityInfo;
            $newCurrentEntity->setCode($toEntity['code']);
            $newCurrentEntity->setLibelle($toEntity['libelle']);
            $newCurrentEntity->setUri($toEntity['uri']);
            $newCurrentEntity->setDescription($toEntity['description']);
            $logger->debug(print_r('create new currentEntity OK', true));

            $em->persist($newCurrentEntity);
            $em->flush();
        }
    }
}
```

Figure 3: Exemple de code repository UPDATE

### 5.2.5 Monitoring

Le système Geod'air repose sur plusieurs briques applicatives (intégration de données, production de statistiques, rapportage..) qui participent toutes au respect des exigences vis-à-vis du système.

Chaque brique produit des logs, certains sont au format texte, d'autres sont mis en base.

Au niveau intégration des données, les logs mis en base sont exploités au niveau de l'IHM (onglet supervision).

**Il ne semble pas exister d'équivalent pour les autres briques, pas plus que de monitoring transverse** avec production de rapports en mode push pour assurer que l'alerte sur les dysfonctionnements est poussée vers l'administrateur applicatif qui peut ensuite suivre leur prise en charge.

Pour illustrer, nous donnons ci-dessous un exemple des logs produits pas le job d'import de données automatiques vérifiées.

Le script sh d'ordonnancement produit ses propres logs qui sont déposés dans le fichier [ftp2etl.log](#). Les logs sont de la forme suivante :

```
28/10/16 16:10:51| ETL ->
import_donnees_automatiques_1.2.57/import_donnees_automatiques_1.2.57/import_donnees_automatiq
ues.sh | AASQA -> AIR_NORMAND | Debut de chargement du fichier /opt/ftphome/08/05NUL296.16T
28/10/16 16:11:23| ETL ->
donnees_automatiques_verifiees_1.2.57/donnees_automatiques_verifiees_1.2.57/donnees_automatiques_
verifiees.sh | AASQA -> AIR_LORRAINE | Lancement calcul des statistiques
28/10/16 16:11:23| ETL ->
donnees_automatiques_verifiees_1.2.57/donnees_automatiques_verifiees_1.2.57/donnees_automatiques_
verifiees.sh | AASQA -> AIR_LORRAINE | Fichier CSV:
/opt/geodair/talend/SCRIPT_CHARGEMENT/WORK/04/01NUL156.16Vrecalcul.CSV
28/10/16 16:11:23| ETL ->
donnees_automatiques_verifiees_1.2.57/donnees_automatiques_verifiees_1.2.57/donnees_automatiques_
verifiees.sh | AASQA -> AIR_LORRAINE | Fin calcul des statistiques
```

On peut voir que leur exploitation n’est pas aisée car chaque étape n’est pas rattachée à un identifiant de thread unique qui permettrait de regrouper les logs correspondant à un même traitement.

Ce même script d’ordonnancement produit un second fichier de logs : [ftp2etlSTAT.log](#)

Sur la recette, ce fichier n’est d’ailleurs pas pris en charge par le logrotate et fait une taille de plus de 2Go. Il stocke les logs produits par le script de calcul des stats périodiques.

Les logs sont de la forme suivante :

```
2016-12-05 22:04:37-368 INFO Scheduler.java - process - DÃ©but calcul des statistiques journaliÃ¨res :  
2016-12-05 22:04:37  
  
2016-12-05 22:04:37-377 INFO StatJour.java - init - Statistiques journalieres : 2016-11-15  
2016-12-05 22:04:37-422 INFO StatJour.java - calcul - debut recherche depassement pour ppt : FR12057-081  
2016-12-05 22:04:37-423 INFO StatJour.java - calcul - fin recherche depassement pour ppt : FR12057-081  
2016-12-05 22:04:37-423 INFO StatJour.java - calcul - debut sauvegarde statistiques dMean pour ppt :  
FR12057-081  
2016-12-05 22:04:37-459 INFO StatJour.java - calcul - fin sauvegarde statistiques dMean pour ppt : FR12057-  
081  
2016-12-05 22:04:37-459 INFO StatJour.java - calcul - debut statistiques daysAbove pour ppt : FR12057-081  
[ ]  
2016-12-05 22:04:37-461 INFO StatJour.java - calcul - fin statistiques daysAbove pour ppt : FR12057-081  
2016-12-05 22:04:37-462 INFO Scheduler.java - process - Fin calcul des statistiques journaliÃ¨res : 2016-12-  
05 22:04:37  
  
2016-12-05 22:04:37-462 INFO Scheduler.java - process - DÃ©but calcul des statistiques hrsAbove : 2016-  
12-05 22:04:37  
  
2016-12-05 22:04:37-466 INFO StatHrsAbove.java - init - Statistiques annuelles : 2016
```

Lorsque ce script d’ordonnancement lance un batch Talend, il redirige les messages de ce dernier (correspondant à des system.out.println vers un fichier de log dédié à chaque job principal :  
« (\$ETL\_DIR/\$NOM\_ETL --context\_param NomFichier=\$REPE\_TRAVAIL/\$FILE\_DIR/`basename \$fichier`  
&>>/opt/geodair/talend/SCRIPT\_CHARGEMENT/LOG/`basename \$NOM\_ETL`.log; echo "\$? \${fichier}.ENC"  
> \${TEMP\_FILE}) & »

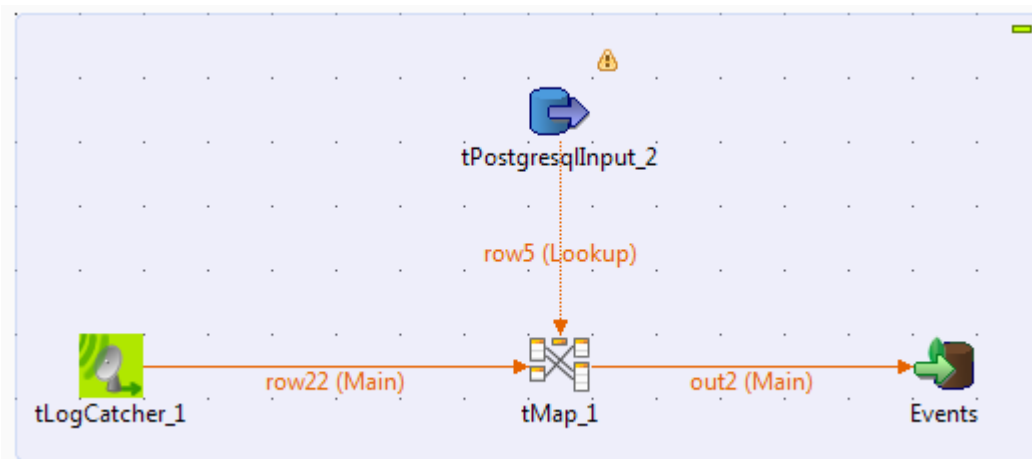
Les logs sont de la forme suivante :

```
2016-11-09 00:00:00|2016-11-15  
03:41:13|import_donnees_automatiques_verifiees|31NUL314.16V|07|AIRAQ|31|AIRAQ|INFO|ETL  
Talend|3|Le fichier 31NUL314.16V va être traité par l'ETL|0  
15/11/2016 04:41:13: Début de traitement pour le fichier 31NUL314.16V  
2016-11-10 00:00:00|2016-11-15  
03:41:13|import_donnees_automatiques_verifiees|08NUL315.16V|03|AIR LANGUEDOC-  
ROUSSILLON|08|AIR LANGUEDOC ROUSSILLON|INFO|ETL Talend|3|Le fichier 08NUL315.16V va être traité  
par l'ETL|0  
15/11/2016 04:41:13: Début de traitement pour le fichier 08NUL315.16V  
[ tChronometerStop_1 ] 16seconds 16692 milliseconds  
[ tChronometerStop_1 ] 19seconds 19674 milliseconds  
For input string: "code_reseau"  
For input string: "code_reseau"  
For input string: "code_reseau"  
[ tChronometerStop_2 ] 3808 milliseconds
```

```
04:41:34,562 INFO [XMLSerializer] Using default type string  
04:41:34,571 INFO [XMLSerializer] Using default type string
```

On peut voir que leur exploitation n’est pas aisée car chaque étape n’est pas rattachée à un identifiant de thread unique qui permettrait de regrouper les logs correspondant à un même traitement.

Il y a également les logs qui sont interceptés par le composant ci-dessous à l’intérieur de chaque job et qui sont intégrées en base.



On trouve également des traces permettant de suivre les volumétries traitées : stat\_ETL.csv et STAT\_LOG .csv dont le contenu respectif est du type :

```
date;fichier;read_data;inserted_data;TotalBloc;IgnoredBloc  
2015-05-05 20:11:17;09NUL121.15V;7776;4320;81;8  
2015-05-07 12:47:30;38NUL258.14V;8064;2592;84;8  
2015-05-07 10:37:41;38NUL234.14V;7680;2592;80;8  
2015-05-07 09:57:01;38NUL176.14V;7008;2976;73;8  
2015-05-05 17:04:46;02NUL120.15V;7968;5280;83;8
```

```
2015-06-30 13:46:24;28NUL180.15T;AUTO_INSERTION_LOG;39  
2015-06-30 13:46:31;18NUL180.15T;AUTO_INSERTION_LOG;46  
2015-06-30 13:47:18;03NUL177.15V;AUTO_INSERTION_LOG;87  
2015-06-30 13:49:10;23NUL177.15V;AUTO_INSERTION_LOG;69  
2015-06-30 13:57:45;28NUL180.15T;AUTO_INSERTION_LOG;0  
2015-06-30 13:57:45;18NUL180.15T;AUTO_INSERTION_LOG;0  
2015-06-30 13:57:45;03NUL177.15V;AUTO_INSERTION_LOG;62  
2015-06-30 14:00:36;23NUL177.15V;AUTO_INSERTION_LOG;0
```

On trouve enfin (a priori) pour chaque fichier de données traité, 4 fichiers csv produits : nomfichierlog.csv, nomfichierreject.csv, nomfichierstat\_temp.csv (utilisé pour construire stat\_ETL.csv) et nomfichier.csv. Ces fichiers contiennent respectivement des données du type :

```
import_donnees_automatiques;0001;19_FR_19005_24B;5;Format fichier incoherent ! Le bloc de donnee  
decrit dans l'entete est manquant pour le polluant 24B sur le site 19005;DEFAULT;MAIL;01;AIR BREIZH;19;AIR  
BREIZH;1;2016-12-10 00:00:00;2016-12-11 02:41:12;;;57;1;;
```

```
import_donnees_automatiques;0004;19_FR19015_02B;5;Configuration de mesure absente du
référentiel;REJET;MAIL;01;AIR BREIZH;19;AIR BREIZH;1;2016-12-10 00:00:00;2016-12-11
02:41:17;Configuration de mesure;FR19015_NO_B;135;1;;
import_donnees_automatiques;0004;19_FR19031_02E;5;Configuration de mesure absente du
référentiel;REJET;MAIL;01;AIR BREIZH;19;AIR BREIZH;1;2016-12-10 00:00:00;2016-12-11
02:41:18;Configuration de mesure;FR19031_NO_E;180;1;;
import_donnees_automatiques;0004;19_FR19015_03B;5;Configuration de mesure absente du
référentiel;REJET;MAIL;01;AIR BREIZH;19;AIR BREIZH;1;2016-12-10 00:00:00;2016-12-11
02:41:19;Configuration de mesure;FR19015_NO2_B;243;1;;
import_donnees_automatiques;0004;19_FR19031_03E;5;Configuration de mesure absente du
référentiel;REJET;MAIL;01;AIR BREIZH;19;AIR BREIZH;1;2016-12-10 00:00:00;2016-12-11
02:41:19;Configuration de mesure;FR19031_NO2_E;279;1;;
```

Format fichier incohérent ! Le bloc de donnée décrit dans l'entête est manquant pour le polluant 24B sur le site 19005;24B;19005;57

```
2016-12-11 03:41:20;19NUL345.16T;4800;0;50
2016-12-11 06:06:13;19NUL345.16T;4800;0;50
2016-12-11 23:59:18;19NUL345.16T;4800;0;50
```

```
iso_version;nom_fichier;type_fichier;code_reseau;nom_reseau;code_station;date_fichier;date_insertion;vali
dation_fichier;numero_organisme;nom_organisme;ligne1;ligne2;ligne3;ligne4;nb_polluants;nb_mesures;co
mmentaire;id_pt_prelevement;id_conf_mesure_ref;type_donnee;heure_debut_periode;duree_periode;inte
rval_temps;interval_temps_mesure_isolee;nb_mesure_par_donnee;facteur_multiplicateur;nb_donnees;nu
mero;id4geoserv_station_ref;id4geoserv_commune_ref;id4geoserv_zas_ref;nom_zas_ref;nom_commune_r
ef;id_commune_ref;id_zas_ref;identification;diff_heure_tuc;latitude;longitude;altitude;reseau_type;nature_
site_ref;code;code_fr;code_complementaire;nom;unite_mesure;unite_mesure_ref;methode;hauteur;code_
default;limite_sup_mesurage;limite_inf_detection;rapportage;date_debut_moy_hor;date_fin_moy_hor;vale
ur;code_etat;date;est_prev_air;unite_mesure_fichier;code_st;reseau_codeue;code_nature_evaluation;code
_type_mesure;technique_analytique_code;modele_appareil_de_mesure_code;methode_de_mesure_code;
methode_de_prelevement_code;modele_equipement_de_prelevement_code;code_unite_concentration;nb
_code_etat_diff;ligne
NORME ISO 7168 v1.7;19NUL345.16T;AUTO;19;AIR BREIZH;FR19002;2016-12-10;2016-12-11
23:59:09;T;01;AIR BREIZH;;28 rue des veyettes;35000 RENNES;France;23;50;SANS COMMENTAIRE;FR19002
02;20;1;2016-12-10 00:00:00;0 0 1 0 0;0 0 0 015;0 0 0 015;1;-1;0;19002;2304;14037;39;BRETAGNE-
RENNES;RENNES;35238;FR19A01;Rennes Laennec;0;+480631.52;-0013957.10;+35;0;Urbaine;02;38;C;NO;µg-
m3;µg-m3;CHIMILUMINESCENCE;3;;1000;0;;2016-12-10 00:00:00;2016-12-10 01:00:00;6.8;A;2016-12-10
00:00:00;t;microg/m3;FR19002;FR072A;1;automatic;;undetermined;chemi;;;µg-m3;1;144
NORME ISO 7168 v1.7;19NUL345.16T;AUTO;19;AIR BREIZH;FR19002;2016-12-10;2016-12-11
23:59:09;T;01;AIR BREIZH;;28 rue des veyettes;35000 RENNES;France;23;50;SANS COMMENTAIRE;FR19002
02;20;1;2016-12-10 00:00:00;0 0 1 0 0;0 0 0 015;0 0 0 015;1;-1;0;19002;2304;14037;39;BRETAGNE-
RENNES;RENNES;35238;FR19A01;Rennes Laennec;0;+480631.52;-0013957.10;+35;0;Urbaine;02;38;C;NO;µg-
m3;µg-m3;CHIMILUMINESCENCE;3;;1000;0;;2016-12-10 00:00:00;2016-12-10 01:00:00;3.9;A;2016-12-10
00:15:00;t;microg/m3;FR19002;FR072A;1;automatic;;undetermined;chemi;;;µg-m3;1;144
```

Comme on peut le voir, le système ne souffre pas d'un manque de traces et de log, mais de la description d'une stratégie transverse de monitoring qui permettrait d'identifier le but et l'usage de chaque log produit.

On voit également que de nombreux fichiers normalement temporaires et utilisés pour construire des fichiers de synthèse ne sont pas détruits comme ils le devraient (comme les xxxstat\_temp.csv par exemple).

## 6 ANALYSE DETAILLEE

### 6.1 QUALITE DE CODE GENERALE

Le code source des différents composants a été analysé pour mesurer la qualité.

L’analyse a été faite dans un premier temps manuellement en parcourant les fichiers et dans un second temps à partir de l’outil Sonar sur certains composants.

Les figures qui suivent illustrent les différents rapports SONAR obtenus. On distingue dans ces rapports trois axes de mesures :

- Bugs (liées à la fiabilité) et vulnérabilités (liées à la sécurité)
- Mauvaises pratiques de conception et programmation (Code Smells)
- Duplications de code

Les lettres dans le rapport correspondent à des taux d’erreurs:

- A = 0 erreur
- B = au moins 1 erreur mineure
- C = au moins 1 erreur majeure
- D = au moins 1 erreur critique
- E = au moins 1 erreur bloquante

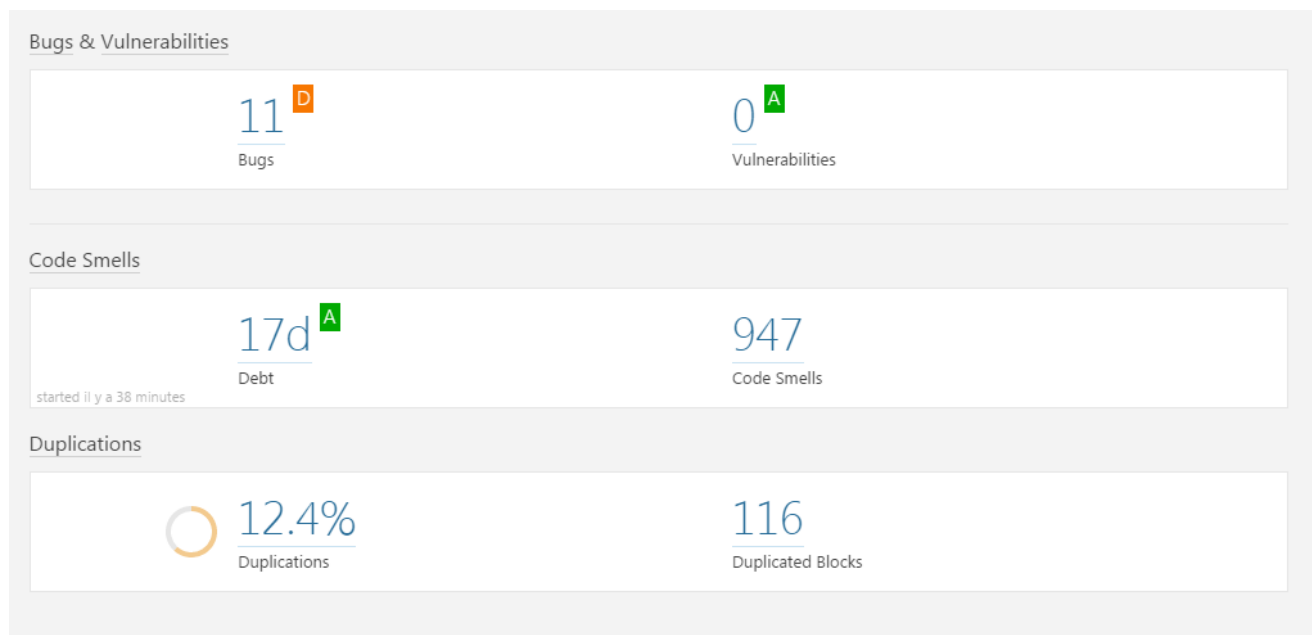


Figure 4: Projet geodair-ref-persistence

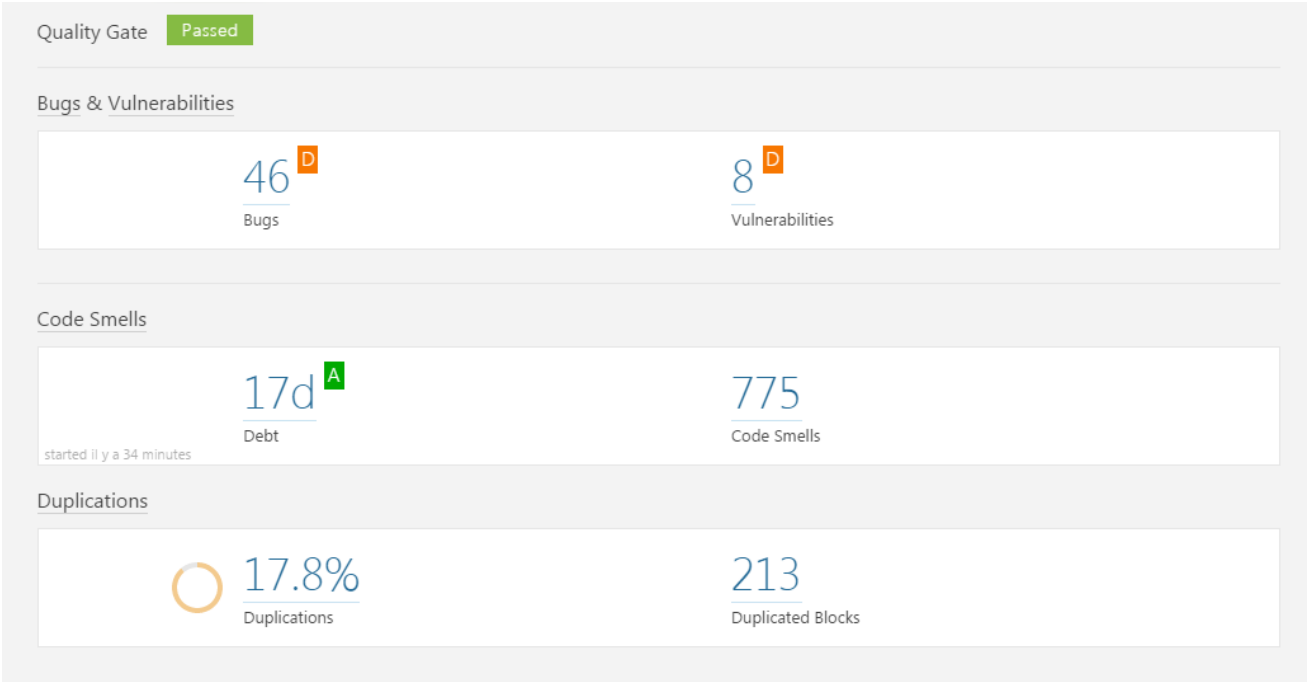


Figure 5: Projet geodairws

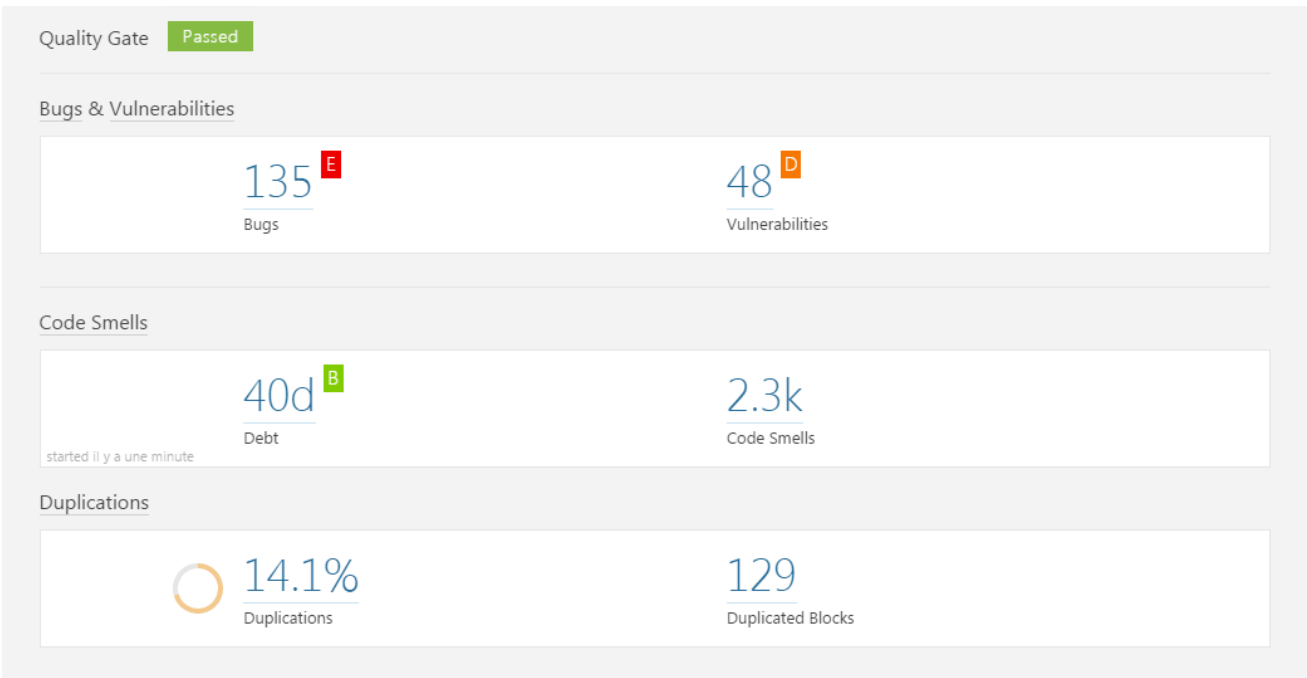


Figure 6: Projet rapportage-ue

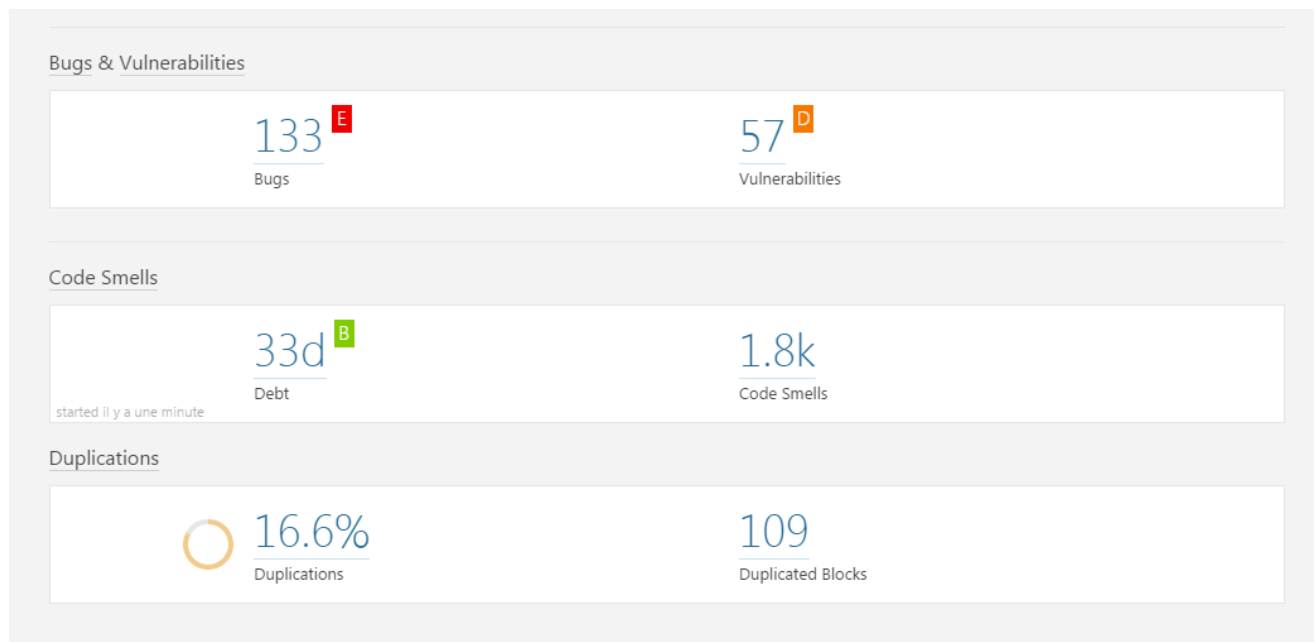


Figure 7: Projet statistique\_periodique

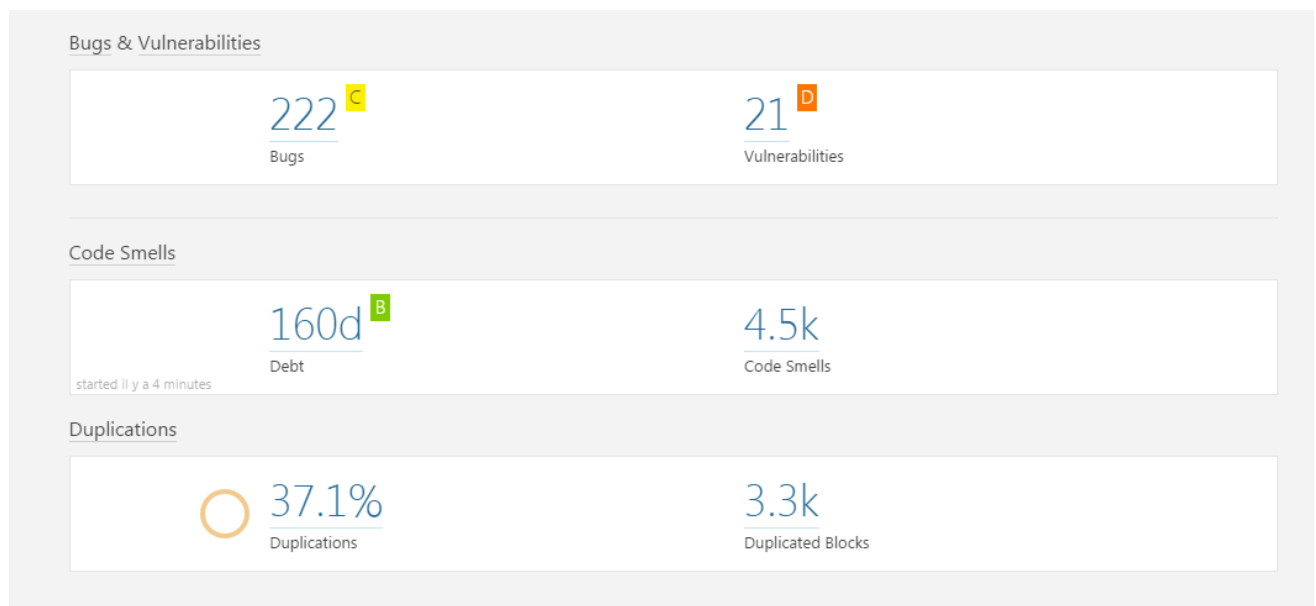


Figure 8: Projet IHM Symphony

Ces analyses ont montré que le code source n'est pas de bonne qualité. Le code est difficilement lisible et par conséquent difficile à maintenir. Les erreurs critiques et leurs conséquences seront détaillées dans le paragraphe de chaque composant.

**Remarque :** La mise en place d'une intégration continue avec mesure de la qualité de code apportera un gain de maintenabilité et performance du code. De plus, l'application des principes et patterns de bases de programmation comme les patterns SOLID permettra d'obtenir une meilleure qualité de code.

## 6.2 BASES DE DONNEES

### 6.2.1 Base Postgresql

La base PostgreSQL contient les tables suivantes :

- Les tables référentielles
- Les tables applicatives (Gestion des utilisateurs)
- Les tables des rapports

Les points suivants ont été relevés lors de l'analyse de la base de données PostgreSQL :

- **Normalisation de la table « rapport\_ue »**

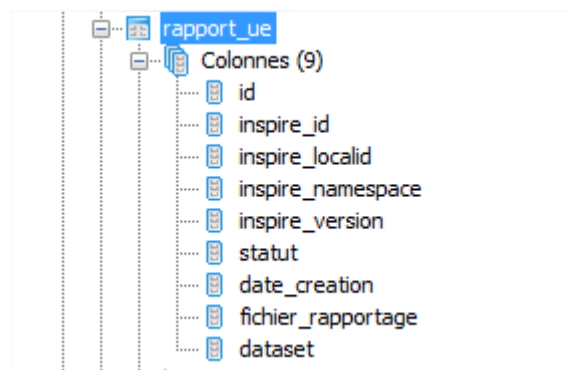


Figure 9: Table "rapport\_ue"

La colonne « inspire\_id » est multivaluée. Elle contient le nom du fichier du rapport qui se décompose ainsi <NOM\_RAPPORT><ANNEE>[0-9].xml.

Plusieurs requêtes dans le module de rapportage utilise la fonction « SIMILAR » qui vérifie que la valeur de la colonne « inspire\_id » respecte cette expression régulière.

Pour éviter l'utilisation de cette fonction, la table doit être normalisée en rajouter les colonnes « NOM\_RAPPORT » et « ANNEE »

- **Gestion du versioning des objets**

La stratégie de versioning actuel des objets référentiels est résumée par le schéma ci-dessous :



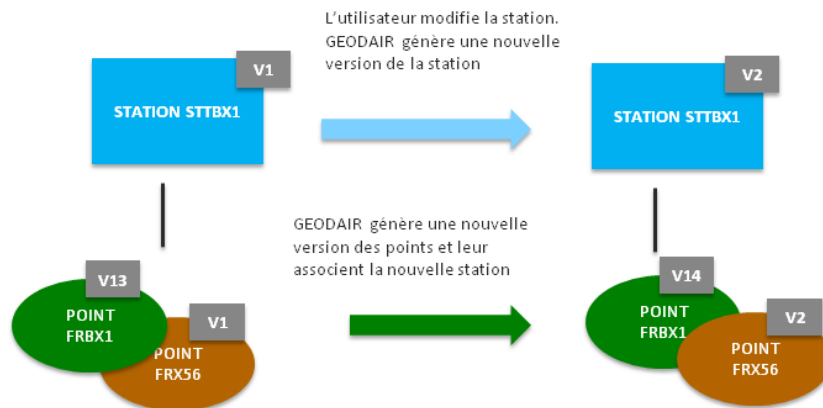
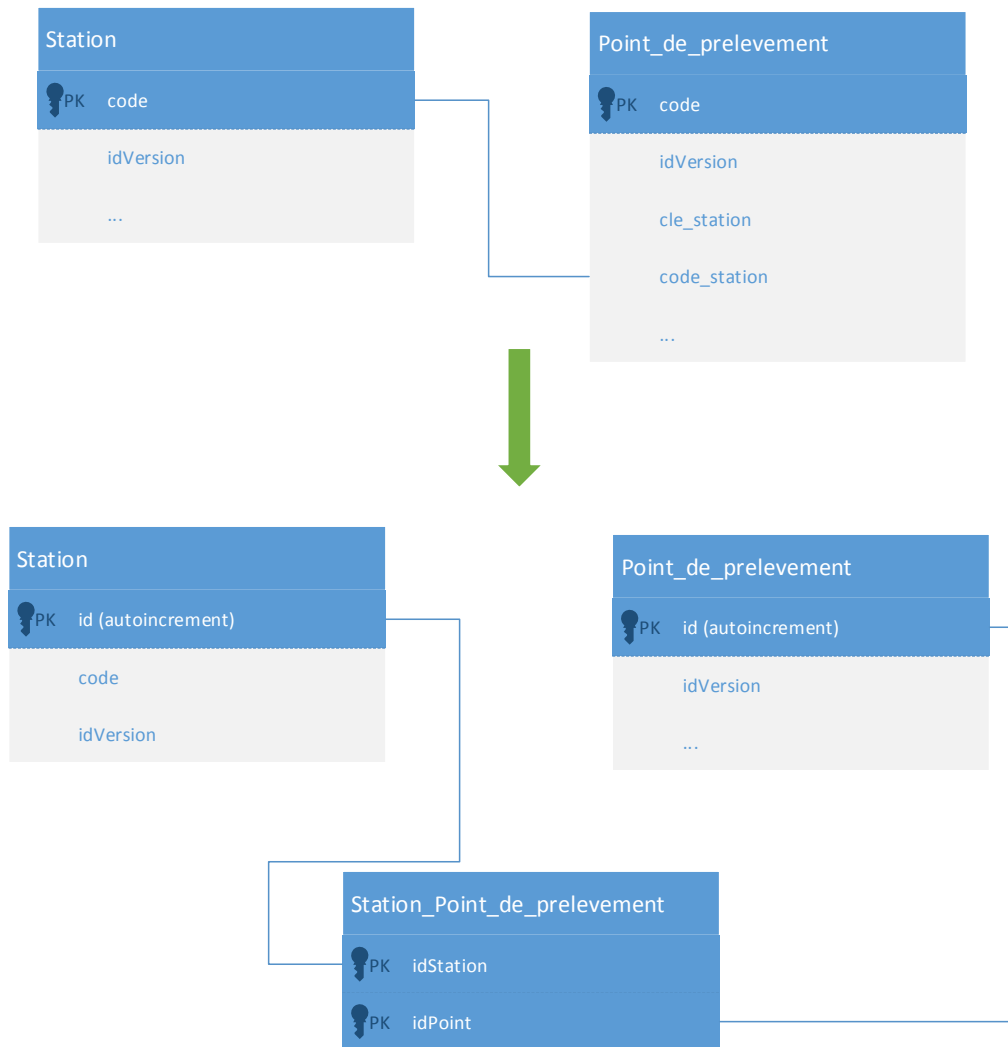


Figure 10: Versioning actuel

Par exemple, à chaque changement de version d'une station, les points de prélèvement sont dupliqués pour être associés à la nouvelle version de la station.

Cette stratégie apporte de la lourdeur au système et aucune plus-value au métier.

**Alternative :** Créer des tables d'association entre entités versionnées afin de ne pas avoir à dupliquer les entités filles. Utiliser de plus des attributs monovalués comme clé primaire pour faciliter la jointure (ID auto incrémenté).



### 6.2.1 Bases MongoDB

Il existe trois instances de base de données MongoDB réparties sur trois serveurs différents. Le tableau ci-dessous donne la liste des bases de données par serveur avec des informations sur la volumétrie et les collections (environnement de recette).

Tableau 1: Liste des bases de données

Serveur	Base de données	Volumétrie	Collections
DB1	DonneesMesuresHistoriques	14Go	DonneesMesures
DB1	RAPPORTAGE_PREVAIR	16Go	PREVAIR_AAAAMMJJ SYS_DATE_COLLECTION_PREVAIR
DB2	Audit	88Go	AuditLogs AuditLogs_AAAA Events Events_AAAA
DB2	IndiceATMO	0,20Go	INDICE_ATMO_HIST
DB2	StatsHistoriques	38Go	StatsHistoriques
DB3	DonneesMesuresOpe	9Go	DonneesMesures
DB3	IndiceATMO	0,06Go	IndiceATMO
DB3	StatsOpe	34Go	IndiceATMO StatsPeriodiques

L’existence d’une base historique sur 20 ans et d’une base opérationnelle sur 3 ans pose une question sur le choix de ce type de base : pourquoi choisir une approche NoSQL gros volume si cela se traduit par la mise en œuvre de 2 bases distinctes ?

Nous verrons plus loin que ce choix est lourd de conséquences en termes de qualité des données, le synchronisme des deux silos n’étant pas garanti.

## 6.3 ETL

### 6.3.1 Les Jobs TALEND

La brique ETL est implémentée sur la base de la déclinaison Big Data de l’ETL de Talend.

Il s’agit d’un produit robuste et reconnu qui a la caractéristique d’être un générateur de code autonome : le studio qui permet à un opérateur de créer une chaîne de traitement de façon interactive et visuelle génère un code java compilé complètement autonome et déployable.

Un projet TALEND est un projet structuré dans lequel on retrouve les jobs de traitement, la description des couches d’accès aux données, les contextes d’utilisation.... quand le système est correctement utilisé.

La méthode choisie par Atos pour nous fournir les éléments liés à l’ETL se borne à la partie jobs, donc traitement et n’inclue pas les autres composants du projet.

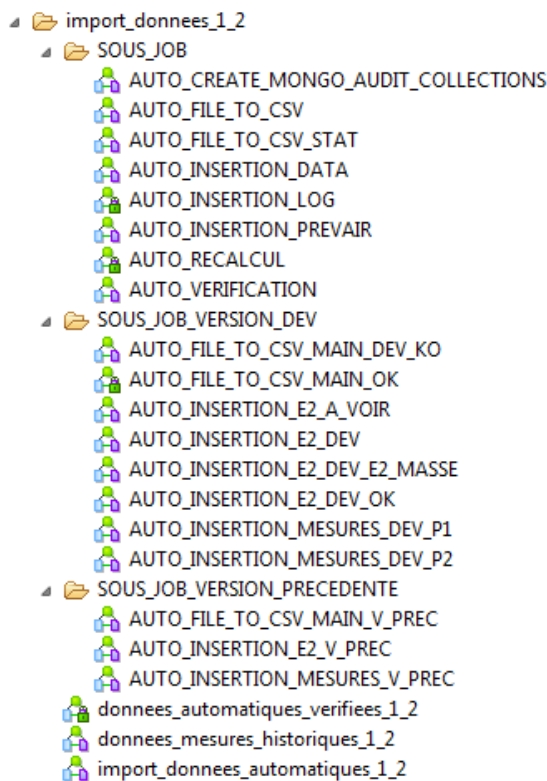
Pour cette raison, nous ne sommes pas en mesure de donner un avis sur le respect des bonnes pratiques dans la structuration des projets ETL par ATOS sur Géod’air.

Nous nous focalisons donc sur le contenu des jobs de traitement.

Nous avons identifié 2 familles de jobs :

- Des jobs liés à l’alimentation de la base référentielle notamment sur la base du COG. Nous n’avons d’ailleurs pas identifié de traitements de prise en charge des évolutions du COG et donc l’évolution possible de l’association des objets métiers avec le référentiel administratif
- Des jobs liés au traitement des données de mesure en distinguant les mesures automatiques, les mesures automatiques vérifiées et les mesures historiques

La subdivision des jobs principaux en sous jobs indique une volonté de factorisation de blocs de traitements communs entre les jobs.



Par contre, l’organisation globale est assez peu claire avec par exemple la cohabitation de versions de jobs secondaires.

L’ordonnancement des jobs est assuré par le script [Ftp2Etl.sh](#) qui est lancé en démon. Ce script scrute les répertoires de dépôt, confirme que le flux détecté n’est plus en cours de transfert et lance le traitement par le bon job ETL.

Le script contrôle le nombre de jobs lancés en parallèle selon ce qui est indiqué en conf (fichier Ftp2Etl.conf) :

- Données automatiques vérifiées : 2
- Données automatiques : 1
- Données manuelles : 1
- Données ATMO : 0 (chaîne d’intégration a priori désactivée)

Nb : Il s’agit ici de la conf des serveurs de recette

Les fichiers transmis sur le FTP sont traités par ordre chronologique.

En fin de traitement réussi d’un flux, ce dernier est déplacé vers un répertoire d’archivage

En cas d’anomalie la procédure de reprise indiquée, mais non décrite est manuelle.

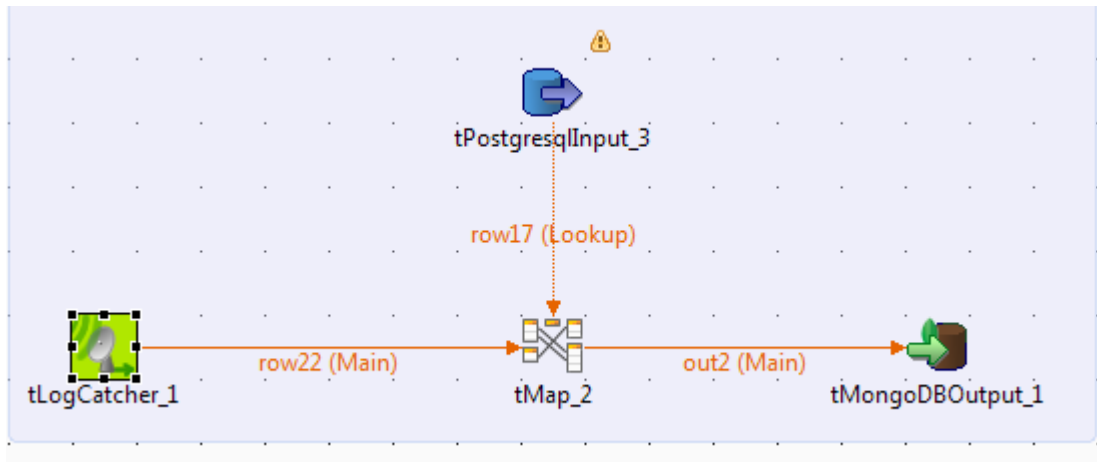
Nous ne comprenons pas la raison de l’intégration de ce script sh qui complique la conception globale dans la mesure où un job Talend dédié, lancé en démon, aurait pu couvrir le même périmètre tout en assurant une certaine cohérence technique.

L’analyse complète et la retro-ingénierie de l’ensemble des jobs n’étaient pas l’objet de cette analyse. Cependant nous constatons que si l’on prend en compte l’écart entre la documentation existante et la réalité des jobs fournis et le fait que la documentation ne couvre pas tout, il y aura un gros effort de rétro-documentation pour assurer la maintenance de cette brique par un tiers.

#### Constat n° 1 :

La stratégie de log/trace mériterait une documentation dédiée afin d’expliquer la distinction faite entre logs en base et logs en fichier.

De notre point de vue le stockage en base de logs peut s’expliquer dans le cas de logs métiers devant être exploités par une IHM d’administration fonctionnelle dédiée.  
Or on peut voir que les logs techniques sont redirigés vers la base.



Nous n’avons pu identifier comment ces logs en base sont exploités dans le cadre d’un monitoring automatisé des traitements.

En complément de ces logs catchés automatiquement, il y a des logs générés par des appels système via un composant java.

Il y’a à priori une troisième source de log, peut être issue d’une analyse / d’un reporting sur les logs déposés en base qui a été fournie par Atos et qui donne des éléments très macroscopiques de prise en charge des fichiers.

#### Constat n°2 :

Les logs transmis par ATOS correspondent aux logs produits par le script d’ordonnancement. Elles donnent donc une vision macro de l’enchaînement des traitements.

Pour rappel, la configuration du script d’ordonnancement en recette donne les niveaux de parallélisme suivants :

- donnees\_automatiques\_verifiees : 2
- import\_donnees\_automatiques : 1
- import\_donnees\_manuelles : 1
- import\_donnees\_ATMO\_1\_2\_7 : 0

L’analyse de ces logs met en évidence une certaine variabilité sur les temps de traitement ce qui peut sans doute s’expliquer par la taille des fichiers à traiter. Globalement l’ordre de grandeur du temps de traitement est de l’ordre de quelques minutes par fichier :

28/10/16 00:36:10| ETL ->

import\_donnees\_automatiques\_1.2.57/import\_donnees\_automatiques\_1.2.57/import\_donnees\_automatiques.sh | AASQA -> LIMAIR | **Debut de chargement du fichier /opt/ftphome/20/35NUL286.16T**

28/10/16 00:38:20| ETL ->

import\_donnees\_automatiques\_1.2.57/import\_donnees\_automatiques\_1.2.57/import\_donnees\_automatiques.sh | AASQA -> LIMAIR | **Succes du chargement du fichier /opt/ftphome/20/35NUL286.16T**

⇒ 2mn10s

28/10/16 05:41:46| ETL ->  
donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees.sh | AASQA -> AIR\_PACA | **Debut de chargement du fichier /opt/ftphome/05/24NUL285.16V**  
28/10/16 05:48:25| ETL ->  
donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees.sh | AASQA -> AIR\_PACA | Lancement calcul des statistiques  
28/10/16 05:48:25| ETL ->  
donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees.sh | AASQA -> AIR\_PACA | Fichier CSV:  
/opt/geodair/talend/SCRIPT\_CHARGEMENT/WORK/05/24NUL285.16Vrecalcul.CSV  
28/10/16 05:48:26| ETL ->  
donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees.sh | AASQA -> AIR\_PACA | Fin calcul des statistiques  
28/10/16 05:48:26| ETL ->  
donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees.sh | AASQA -> AIR\_PACA | **Succes du chargement du fichier /opt/ftphome/05/24NUL285.16V**  
⇒ 6mn40s

- ⇒ import\_donnees\_automatiques en cours pour 23NUL296.16T
- ⇒ donnees\_automatiques\_verifiees en cours pour 01NUL156.16V

28/10/16 16:10:51| ETL ->  
import\_donnees\_automatiques\_1.2.57/import\_donnees\_automatiques\_1.2.57/import\_donnees\_automatiques.sh | AASQA -> AIR\_NORMAND | **Debut** de chargement du fichier /opt/ftphome/08/05NUL296.16T  
⇒ 1(/1) import\_donnees\_automatiques début pour 08/05NUL296.16T  
28/10/16 16:11:23| ETL ->  
donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees.sh | AASQA -> AIR\_LORRAINE | Lancement calcul des statistiques  
28/10/16 16:11:23| ETL ->  
donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees.sh | AASQA -> AIR\_LORRAINE | Fichier CSV:  
/opt/geodair/talend/SCRIPT\_CHARGEMENT/WORK/04/01NUL156.16Vrecalcul.CSV  
28/10/16 16:11:23| ETL ->  
donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees.sh | AASQA -> AIR\_LORRAINE | Fin calcul des statistiques  
28/10/16 16:11:23| ETL ->  
donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees.sh | AASQA -> AIR\_LORRAINE | Succes du chargement du fichier /opt/ftphome/04/01NUL156.16V  
⇒ 1(/2) donnees\_automatiques\_verifiees fin pour 01NUL156.16V  
28/10/16 16:11:24| ETL ->  
donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees.sh | AASQA -> AIR\_LORRAINE | **Debut** de chargement du fichier /opt/ftphome/04/01NUL157.16V  
⇒ 2(/2) donnees\_automatiques\_verifiees début pour 01NUL157.16V  
28/10/16 16:11:46| ETL ->  
donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees.sh | AASQA -> AIR\_LORRAINE | Lancement calcul des statistiques  
28/10/16 16:11:46| ETL ->  
donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees.sh | AASQA -> AIR\_LORRAINE | Fichier CSV:  
/opt/geodair/talend/SCRIPT\_CHARGEMENT/WORK/04/01NUL157.16Vrecalcul.CSV

28/10/16 16:11:46| ETL ->  
donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees.sh | AASQA -> AIR\_LORRAINE | Fin calcul des statistiques  
28/10/16 16:11:46| ETL ->  
donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees.sh | AASQA -> AIR\_LORRAINE | Succes du chargement du fichier /opt/ftphome/04/01NUL157.16V  
⇒ 1(/2) donnees\_automatiques\_verifiees fin pour 01NUL157.16V  
28/10/16 16:11:46| ETL ->  
donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees.sh | AASQA -> AIR\_LORRAINE | **Debut** de chargement du fichier /opt/ftphome/04/01NUL158.16V  
⇒ 2(/2) donnees\_automatiques\_verifiees début pour 01NUL158.16V  
28/10/16 16:12:18| ETL ->  
donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees.sh | AASQA -> AIR\_LORRAINE | Lancement calcul des statistiques  
28/10/16 16:12:18| ETL ->  
donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees.sh | AASQA -> AIR\_LORRAINE | Fichier CSV:  
/opt/geodair/talend/SCRIPT\_CHARGEMENT/WORK/04/01NUL158.16Vrecalcul.CSV  
28/10/16 16:12:18| ETL ->  
donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees.sh | AASQA -> AIR\_LORRAINE | Fin calcul des statistiques  
28/10/16 16:12:18| ETL ->  
donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees.sh | AASQA -> AIR\_LORRAINE | Succes du chargement du fichier /opt/ftphome/04/01NUL158.16V  
⇒ 1(/2) donnees\_automatiques\_verifiees fin pour 01NUL158.16V  
28/10/16 16:12:19| ETL ->  
donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees\_1.2.57/donnees\_automatiques\_verifiees.sh | AASQA -> AIR\_LORRAINE | **Debut** de chargement du fichier /opt/ftphome/04/01NUL159.16V  
⇒ 2(/2) donnees\_automatiques\_verifiees début pour 01NUL158.16V  
28/10/16 16:12:52| ETL ->  
import\_donnees\_automatiques\_1.2.57/import\_donnees\_automatiques\_1.2.57/import\_donnees\_automatiques.sh | AASQA -> AIR\_PAYS-DE-LA-LOIRE | Succes du chargement du fichier /opt/ftphome/06/23NUL296.16T  
⇒ import\_donnees\_automatiques fin pour 23NUL296.16T  
28/10/16 16:12:53| ETL ->  
import\_donnees\_automatiques\_1.2.57/import\_donnees\_automatiques\_1.2.57/import\_donnees\_automatiques.sh | AASQA -> AIR\_NORMAND | **Debut** de chargement du fichier /opt/ftphome/08/25NUL296.16T  
⇒ import\_donnees\_automatiques Début pour 25NUL296.16T  
28/10/16 16:13:03| ETL ->  
import\_donnees\_automatiques\_1.2.57/import\_donnees\_automatiques\_1.2.57/import\_donnees\_automatiques.sh | AASQA -> AIR\_NORMAND | **Succes du chargement du fichier /opt/ftphome/08/05NUL296.16T**

Le parallélisme que l’on peut déduire de ces logs semble différent pour la production avec :

- Données automatiques vérifiées : 1
- Données automatiques brutes >= 2

Nous n’avons pas identifié dans les documents, d’analyse de la capacité du système en termes de parallélisme et donc de sa capacité à pouvoir s’ajuster à une augmentation de la charge de fichiers de mesures à traiter.

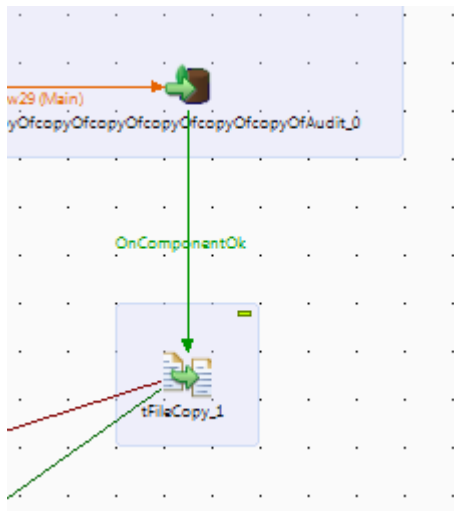
Une analyse de l’évolution du stock de fichiers de mesure à traiter dans le temps serait pertinente pour permettre d’affiner le niveau de parallélisme.

Une perspective d'optimisation des jobs réside

- dans le fait que les jobs accèdent aux bases Postgresql et MongoDB et sont amenés à réaliser des opérations qui pourraient se faire directement en base dans le cas d'une base unique.
- dans la rationalisation de la production des logs et traces

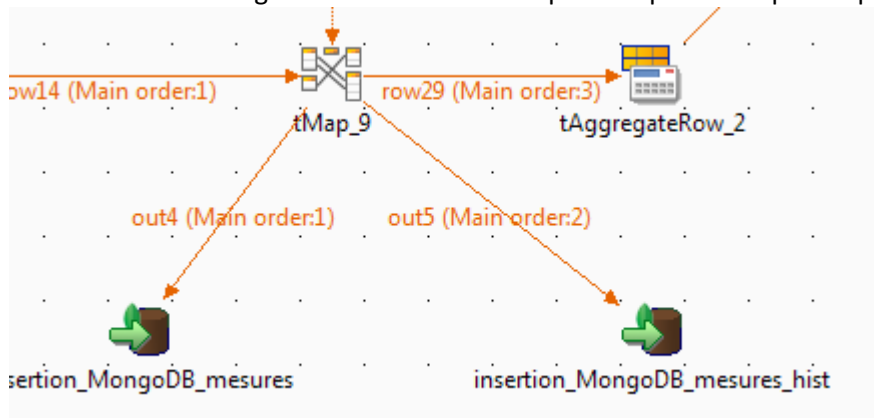
### Constat n°3 :

Un problème récurrent et connu de ce type d'outil est d'assurer une approche transactionnelle des actions ou d'intégrer une stratégie de reprise sur anomalie.



Une dernière écriture en base après l'ensemble des traitements sur le fichier déclenche sa suppression. En cas d'anomalie sur la dernière étape d'écriture en base, le job s'arrête. L'état général du système est alors que le fichier a bien été traité, intégré en base, mais comme il n'a pas été supprimé, il sera sans doute repris en charge lors d'un prochain déclenchement du script ordonnanceur.

Un autre cas de non gestion transactionnelle pose un problème plus impactant et qui de plus semble avéré.



Comme on peut le voir ici, le sous-job qui a en charge l'insertion des données en base assure l'intégration parallèle des données dans la base historique et dans la base opérationnelle.

Pour des raisons évidentes, la base historique doit rester synchrone avec la base opérationnelle, or dans cette configuration rien ne le garantit.

Une des deux écritures peut échouer alors que l'autre a réussi.

On notera que MongoDB pose une difficulté supplémentaire dans la mesure où ce n'est pas une base transactionnelle. Là où il est possible de gérer une transaction sur plusieurs insertions dans le cadre d'une connexion à une base comme PostgreSQL, MongoDB va gérer chaque écriture de façon atomique et indépendante.



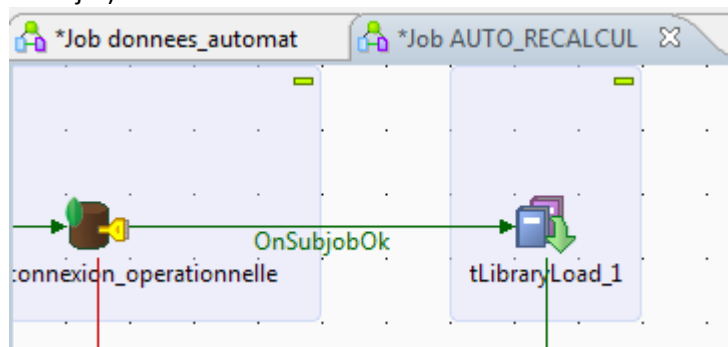
Il est donc nécessaire de mettre en place toute la stratégie de transaction au niveau des procédures d’intégration des données.

Ce problème pourrait se poser sur Geod’air en cas d’insertion de lots de données dont on souhaite s’assurer de la cohérence / complétude à l’échelle du lot dans son ensemble.

#### Constat n°4 :

Une insertion de données de mesures en base implique le recalcul des statistiques.

Ce recalcul des stats horaires est bien déclenché par l’ETL (job AUTO\_RECALCUL) mais toute la logique de calcul et l’intégration en base des statistiques est confiée à une librairie externe (statistique\_periodique-0.0.4.jar)



Ce choix implique une nouvelle couche DAO autonome à maintenir.

Dans la documentation l’argument donné pour le fait de traiter les statistiques dans un code java isolé est de pouvoir le lancer depuis Talend pour les stats horaires et de façon autonome pour les autres statistiques. Cet argument n’exclut en rien la possibilité de traiter le calcul des statistiques par un job Talend qui pourrait être appelé depuis un autre job ou de façon autonome. Ceci permettrait de rester techniquement homogène et de reposer sur la même couche DAO.

Si la complexité des calculs justifie l’usage d’un code externe à l’ETL, ce dernier aurait pu être limité aux algorithmes de calcul en laissant les accès aux données à la partie ETL, réduisant ainsi l’effort de maintien en cohérence des couches DAO.

#### Constat n°5 :

A la demande de l’INERIS nous analysons ici la stratégie d’utilisation de fichiers temporaires par les jobs et sous jobs.

Nous prenons comme exemple le job de traitement des données automatiques vérifiées et identifions les créations / utilisations et suppressions de fichiers temporaires pour ce job et ses sous jobs :

Job données automatiques vérifiées

**AUTO\_CREATE\_MONGO\_AUDIT\_COLLECTIONS**

RAS

**AUTO\_FILE\_TO\_CSV**

**AUTO\_FILE\_TO\_CSV\_MAIN\_OK**

NOMFICHIER – lecture

NOMFICHIERlog.csv -> écriture (en fin de fichier) et création si fichier inexistant

NOMFICHIERreject.csv -> écriture (en fin de fichier) et création si fichier inexistant

NOMFICHIERlog.csv -> écriture (en fin de fichier) et création si fichier inexistant

NOMFICHIERreject.csv -> lecture

NOMFICHIER.CSV -> écriture par écrasement et création si fichier inexistant

NOMFICHIERreject.csv -> suppression

NOMFICHIERstat\_temp.csv – écriture (en fin de fichier) et création si fichier inexistant

**AUTO\_FILE\_TO\_CSV\_STAT**

NOMFICHIERlog.csv -> lecture  
NOMFICHIERstat\_temp.csv – lecture  
NOMFICHIERlog.csv -> écriture (en fin de fichier) et création si fichier inexistant  
ETL\_STAT – écriture par ajout  
NOMFICHIERstat\_temp.csv – suppression

#### AUTO\_VERIFICATION

NOMFICHIERlog.csv -> lecture  
NOMFICHIERlog.csv -> lecture  
NOMFICHIERrecalcul.csv -> écriture par écrasement et création si fichier inexistant  
NOMFICHIERA.csv -> écriture par écrasement et création si fichier inexistant  
NOMFICHIERA.csv -> lecture  
NOMFICHIERlog.csv -> écriture  
NOMFICHIERA.csv -> suppression

#### AUTO\_INSERTION\_MESURES\_DEV\_P1

NOMFICHIERlog.csv -> lecture  
NOMFICHIERlog.csv -> lecture  
NOMFICHIERlog.csv -> écriture (en fin de fichier) et création si fichier inexistant  
NOMFICHIER\_insertion\_mesure\_tmp.csv -> écriture (en fin de fichier) et création si fichier inexistant  
NOMFICHIER\_E2\_PREVAIR.CSV -> écriture par écrasement et création si fichier inexistant

#### AUTO\_INSERTION\_MESURES\_DEV\_P2

NOMFICHIER\_insertion\_mesure\_tmp.csv -> lecture  
NOMFICHIER1.csv -> écriture par écrasement et création si fichier inexistant  
NOMFICHIER\_insertion\_mesure\_tmp.csv -> suppression

#### AUTO\_INSERTION\_DATA

NOMFICHIER1.csv -> lecture  
NOMFICHIERlog.csv -> écriture (en fin de fichier) et création si fichier inexistant

#### AUTO\_INSERTION\_E2\_DEV\_E2\_MASSE

NOMFICHIER\_E2\_PREVAIR.CSV -> lecture  
NOMFICHIERlog.csv -> écriture (en fin de fichier) et création si fichier inexistant  
EmptyE2 -> écriture (en fin de fichier) et création si fichier inexistant  
« Horodatage »\_E2\_data.csv -> écriture (en fin de fichier) et création si fichier inexistant

#### AUTO\_RECALCUL

NOMFICHIERrecalcul.csv -> lecture

#### AUTO\_INSERTION\_LOG

NOMFICHIERlog.csv -> lecture  
NOMFICHIER\_countRejets.csv -> écriture par écrasement  
NOMFICHIER -> lecture  
NOMFICHIER\_countRejets.csv – lecture pour écriture dans base audit  
NOMFICHIERlog.csv -> suppression  
NOMFICHIER.CSV -> suppression  
NOMFICHIER1.CSV -> suppression  
NOMFICHIER\_E2\_PREVAIR.CSV.CSV -> suppression  
NOMFICHIER\_countRejets.csv -> suppression

Cette approche permet une décomposition en sous jobs partageables entre jobs ce qui assure un minimum de factorisation des traitements.

Le choix de l’écriture en fichier est pertinent du point de vue de l’efficacité (à support physique de stockage équivalent) et permet d’éviter de surcharger les bases avec des opérations d’insertion et de suppression multiples.

Cependant, pour fonctionner, cette approche repose sur 2 hypothèses :

- Les suppressions des fichiers temporaires se font bien comme prévu en fin de job principal (nécessaire étant donné qu’une partie des écritures se fait sans écrasement)
- Il n’y a pas de possibilité qu’un fichier de données soit traité par deux flux au même instant (ceci est très peu probable vu qu’un fichier de mesure n’est déplacé vers l’archivage qu’une fin de traitement)

Il reste les fichiers non préfixés par le nom du fichier de mesure en cours de traitement : ETL\_STAT, empty2 et « Horodatage »\_E2\_data.csv pour lesquels des écritures concurrentes par plusieurs jobs sont possibles dans la mesure où Talend ne verrouille pas les fichiers en écriture.

Cependant les données des fichiers ci-dessus ont une cohérence à l’échelle de la ligne (l’ordre des lignes n’est pas important) et Talend procède à des écritures par ligne entière.

La probabilité de conflit d’écriture est donc faible vu le niveau de parallélisme actuel.

### 6.3.2 Statistiques périodiques

Le module de calcul des données statistiques est appelé par l’ETL pour la production de données statistiques. Les calculs sont effectués de deux manières :

- « Au fil de l’eau » de la réception des fichiers de mesure
- « A l’échéance »

A l’analyse de ce module, les points suivants ont été relevés.

- **Gestion des logs**

Contrairement à ce qui a été dit lors de l’atelier avec Atos, tous les logs (techniques et métiers) sont enregistrés dans la BD « Audit » (DB2). En environnement de recette, cette base a une volumétrie de 88Go représentant la base la plus volumineuse. Nous nous posons la question de l’utilité de sauvegarder des logs techniques dans une BD.

N’ayant pas accès à l’environnement de production, il serait intéressant de vérifier la volumétrie de cette base et par la même occasion vérifier le niveau de log activé.

- **Imbrication des requêtes**

Nous avons trouvé plusieurs requêtes imbriquées dans le code JAVA impactant les performances du calcul des statistiques.

Par exemple, la méthode « InitialisationDepassement » de la classe « ReferentielDB », permettant d’initialiser les dépassements de seuil imbrique des requêtes dans des boucles. Ce qui provoque des allers-retours fréquents entre l’application et la base de données.

Pour y remédier, il serait plus efficace d’utiliser le mode batch du driver JDBC Java permettant de grouper des requêtes et les soumettre en un seul appel à la base de données.

- **Utilisation du driver JDBC Java**

Pour implémenter la couche DAO pour ce module, le choix a été porté sur le driver JDBC Java. Il en ressort que le code est verbeux et difficile à maintenir. L’utilisation d’un framework tel que Spring Data ou JTA permet d’avoir un code plus maintenable, « sans réinventer la roue » et sans impact considérable sur les performances.

- **Qualité de code**

Le tableau ci-dessous montre les erreurs de code les plus critiques et leurs conséquences :

Types d'erreur	Occurrences	Conséquences/Risques	Exemple
« Resources should be closed » Les ressources I/O doivent être fermées après utilisation	28	Ne pas fermer explicitement les ressources I/O telles que les « PreparedStatement » peut conduire à une fuite mémoire et donc des problèmes de performance	
« Exception handlers should preserve the original exceptions »	71	Ne pas logger ou propager une exception catchée rend difficile le débogage en cas de problème.	
Ne pas utiliser la classe « Statement » pour effectuer une requête avec des paramètres	13	La classe « Statement » n'est pas protégée contre les attaques d'injection SQL.	
Throwable.printStackTrace(...) should not be called	24	Utiliser cette expression affiche l'objet « Throwable » et sa pile de trace sur la sortie d'erreur standard. Des données sensibles peuvent donc être affichées. Il est recommandé d'utiliser un Logger.	Classe « MsgAlerte », Méthode « SendMail », L147
16,6% de duplication de code surtout au niveau des classes « Stat »		La conception des classes représentant les différents types de calcul de statistiques n'est pas effectuée dans l'état de l'art d'où la forte duplication de code. La maintenance et la lisibilité du code sont donc impactées négativement.	Classes « StatAnnee », « StatJour »...

## 6.4 WEBSERVICES (SERVICEMIX) / ESB

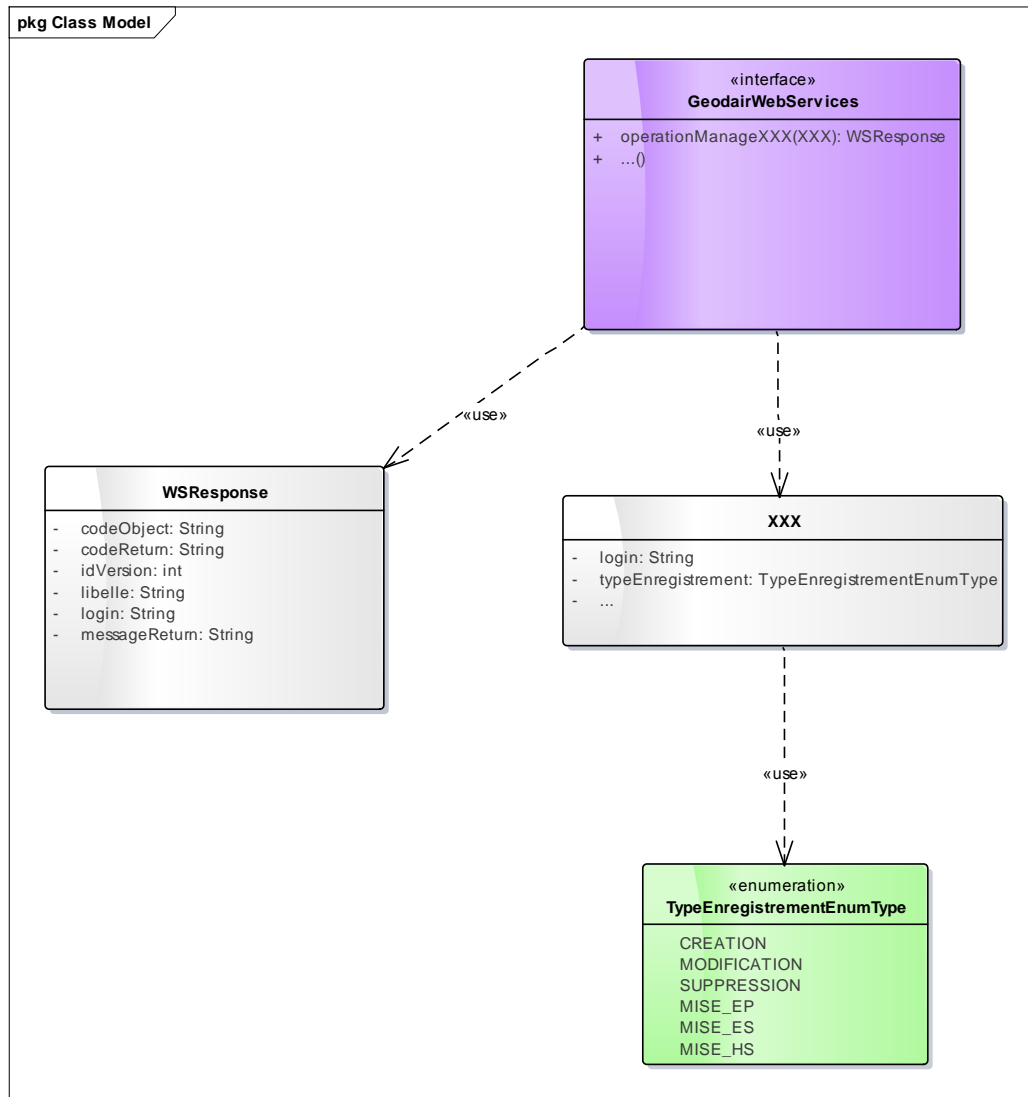
L'ESB « ServiceMix » expose deux web services :

- Service CUD (Create, Update, Delete) des référentiels pour la modification des référentiels
- Service de rapportage permettant de générer un rapport et modifier le statut d'un rapport

### 6.4.1 Service CUD Référentiels

Le service CUD Référentiels permet de créer, modifier ou supprimer des référentiels. Le choix a été fait de ne pas implémenter la lecture dans le bus de services mais plutôt dans la couche de présentation pour des raisons non connues par l'équipe actuelle d'Atos.

Les requêtes de lecture des référentiels sont donc dupliquées dans chaque composant (« WEB1 » et « DAT1 »).



**Figure 11: Conception des webservices CUD Référentiels**

Le schéma ci-dessus illustre la conception de l'exposition du service. Pour chaque table référentielle à exposer, une opération de nom « operationManageXXX » (XXX désigne le nom de la table) est créée. Cette opération prend en paramètre une requête de type XXX contenant au moins 2 attributs:

- login : Login de l'utilisateur
- typeEnregistrement : Type de l'enregistrement souhaité (Création, Suppression,...)

Des attributs supplémentaires sont rajoutés en fonction de la table.

Toutes les opérations renvoient un objet « WSResponse » défini dans la figure précédente.

Les points suivants ont été relevés sur ces services :

- **Conception des services SOAP**

L'une des bonnes pratiques en SOA est de créer un service par domaine ou sous-domaine entraînant un couplage faible entre les services métiers. Ainsi, créer un seul endpoint contenant un seul service avec toutes les opérations enfreint le principe de séparation des responsabilités (SRP). La moindre modification sur une opération entraîne la livraison de tout l'ensemble des opérations.

Par ailleurs, il est vrai que le service CUD des référentiels n'est pas un réel service métier mais technique dans le sens où ses opérations sont relatives aux tables de la base de données PostgreSQL. Créer un

webservice par table conduirait à avoir plusieurs services unitaires contenant qu’une seule opération. Ce qui est une mauvaise pratique.

**Recommandation :** L’alternative est donc de regrouper en un seul service les opérations sur des tables connexes (par ex : ProcedureMesure, ConfigurationMesure, UniteMesure). Une autre alternative apporte une meilleure granularité est de passer sur une architecture REST.

- **Sécurité des services**

Le login passé en paramètre n’est pas utilisé par les services. C’est une très mauvaise pratique d’exposer des services CRUD sans vérification de l’identité de l’appelant.

**Recommandation :** Bien qu’il soit vrai que l’accès à l’ESB ne se fait que depuis des serveurs frontaux via des IP privés, il est important d’avoir des web services autonomes et ne pas déporter la responsabilité de la sécurité chez l’appelant.

- **Couche d’accès aux données**

La couche d’accès aux données référentielles (Repository) est implémentée en utilisant le framework Apache OpenJPA. Nous avons remarqué quelques requêtes non optimisées mais sans impact fort sur les performances.

Ci-dessous un exemple de requête non optimisé :

```
@Override
public String getUserOrganisme(String userLogin) {

    LOGGER.info("getUserPerimetre login : " + userLogin);
    Query query = em.createQuery("SELECT u from FosUserUser u WHERE u.username= :userLogin");
    query.setParameter("userLogin", userLogin);

    FosUserUser fosUser = (FosUserUser) query.getSingleResult();
    String userOrganisme = null;

    if(fosUser != null){
        userOrganisme = fosUser.getCodeOrganisme();
    }

    return userOrganisme;
}
```

Figure 12: Select \*

- **Qualité du code**

Le tableau ci-dessous montre les erreurs les plus critiques et leurs conséquences :

Types d’erreur	Occurrences	Conséquences/Risques	Exemple
« Synchronized classes Vector, Hashtable, Stack and StringBuffer should not be used »	9	Les classes telles que Vector, Hashtable et StringBuffer sont « thread-safe ». Malheureusement la synchronisation a un impact négatif sur les performances. Remplacer l’utilisation de StringBuffer par StringBuilder.	Classe « ServiceReferentielImpl.java ». Méthode « canDeletePolluant »
« Exception handlers should preserve the original exceptions »	40	Ne pas logger ou propager une exception catchée rend difficile le débogage en cas de problème.	

17,8% de duplication de code dans le projet « geodairws » et 12,4% dans le projet « geodair-ref-persistence »		La duplication de code rend le code moins maintenable et donc difficile à faire évoluer. Elle permet de déduire un manque de conception.	
---------------------------------------------------------------------------------------------------------------	--	------------------------------------------------------------------------------------------------------------------------------------------	--

### 6.4.2 Services Rapportage

Les webservices « Rapportage » permettent depuis l'IHM de produire des données au format XML conformément à un schéma de données bien défini.

L'API Java JDOM 1.1 est utilisée pour générer les fichiers XML.

**Recommandation** : La version 2.0 sortie en 2012 est plus performante<sup>1</sup> et doit être utilisée pour un gain de performance.

Par ailleurs, l'analyse du code relève que plusieurs objets liés à l'I/O (connections aux bases de données, flux de fichier,...) ne sont pas nettoyés explicitement. Des ressources I/O non nettoyées peuvent entrainer des problèmes de performance.

Nous avons aussi relevé des requêtes non optimisées pouvant entrainer des problèmes de performance :

- Exemple 1

```
// On Récupère la liste des points de prélèvements associés au regime d'évaluation
// via le type d'évaluation
requete = "select distinct mep.code_point_de_prelevement,"
+ " ne.code, "
+ " mo.code, "
+ " po.codefr, "
+ " pp.code_station,"
+ " te.description, "
+ " ne.uri "
+ "from depassement "
//+ "LEFT join modele mo on te.id_modele = mo.id "
+ "LEFT JOIN objectif_environnemental oe on depassement.id_objectif_environnemental = oe.id "
+ "LEFT JOIN regime_d_evaluation re on re.id_objectif_environnemental = oe.id "
+ "LEFT JOIN type_d_evaluation te on te.id_regime_d_evaluation = re.id "
+ "LEFT join moyen_d_evaluation_pdp mep on mep.id_type_d_evaluation = te.id "
+ "INNER JOIN nature_evaluation ne on ne.id = te.code_nature_evaluation "
+ "LEFT join moyen_d_evaluation_modele mem on mem.id_type_d_evaluation = te.id "
+ "LEFT JOIN modele mo on mem.code_modele = mo.code and mem.cle_modele = mo.idversion "
+ "LEFT join point_de_prelevement pp on mep.code_point_de_prelevement = pp.code and mep.cle_point_de_prelevement = pp."
+ "LEFT join polluant po on pp.code_polluant = po.code "

+ "where depassement.id = " + resultats.getString("id") + " ";

// ne pas oublier que le point de prelevement doit etre actif pour l'annee en cours
//System.out.println(requete);
Statement stmt1 = conn.createStatement();
ResultSet resultats1 = stmt1.executeQuery(requete);
int nbppt = 0;
if (resultats1 != null) {
    while (resultats1.next()) {
        nbppt++;
    }
}
if (nbppt != 0)
{
    vAttainments.add(datasetG);
}
```

L'utilisation de la fonction count(\*) directement sur le « select » est plus performante que de récupérer les lignes et les compter par programmation

<sup>1</sup> <http://hunterhacker.github.io/jdom/jdom2/performance.html>

- Utilisation abusive de requêtes imbriquées

Nous avons aussi relevé l’utilisation de plusieurs requêtes imbriquées (par exemple la méthode « RecupererZones » de la classe « BddReferentiel ») qui peuvent être réécrites pour gagner en performance.

- Agrégation de documents MongoDB

MongoDB 2.2 a introduit un mécanisme d’agrégation de documents appelé « pipeline d’agrégation ». Il consiste à définir une série (appelées « pipeline ») d’opérations (appelées « stages ») sur des données provenant de plusieurs documents de collections multiples. Pour la production du rapportage Dataset E, la méthode d’agrégation de l’API JAVA MongoDB utilisée pour effectuer l’agrégation des statistiques ne correspond pas à celle du framework de pipeline. Cette méthode est d’ailleurs obsolète comme le montre la capture de code ci-dessous. L’opération d’agrégation est donc moins performante.

```
AggregationOutput output = this.statColl.aggregate(match,group,sort);
System.out.println("output : "+ output.toString());
BasicDBList result = (BasicDBList) output.getCommandResult().get("result");
BasicDBList resultFiltre = result;
ArrayList<String> Nonfiltre = new ArrayList<String>();
ArrayList<String> filtre ;
for (int j = 0; j<resultFiltre.size(); j++)
```

- Qualité de code

Types d’erreur	Occurrences	Conséquences/Risques	Exemple
“NullPointerException might be thrown as “XXX” is nullable here	2	Nous trouvé du code provoquant obligatoirement un « NullPointerException ». Vérifier qu’il ne s’agit pas d’un code mort.	Classe « DatasetG » L158 listeAjustements
« Floating point numbers should not be tested for equality »	1	Faire des comparaisons sur des float, double avec l’opérateur « == » est source d’erreur à cause des précisions.	Classe « DatasetE2.java » if (incertitude == 0) Record.setText("");
Math operands should be cast before assignment	13	Si le résultat de la division de deux entiers est assigné à une variable flottante, la précision sera perdue avant l’affectation. Par ex : float twoThirds = 2/3. Le résultat sera « 0.0 » Ce genre de code peut fausser les résultats de calcul	Classe « ReportDataSetE.java » tauxValide = 100 * nbMesuresValides.intValue() / nbHeures;
Throwable.printStackTrace(...) should not be called	15	Utiliser cette expression affiche l’objet “Throwable” et sa pile de trace sur la sortie d’erreur standard. Des données sensibles peuvent donc être affichées. Il est recommandé d’utiliser un Logger.	
Ne pas utiliser la classe “Statement” pour effectuer une requête avec des paramètres	8	La classe « Statement » n’est pas protégée contre les attaques d’injection SQL	



14,1% de duplication de code		La duplication de code rend le code moins maintenable et donc difficile à faire évoluer. Elle permet de déduire un manque de conception.	
------------------------------	--	------------------------------------------------------------------------------------------------------------------------------------------	--

**Recommandation :** En résumé, le module « Rapportage » nécessite une réécriture en tenant compte des erreurs SONAR, en mettant en place une meilleure conception et en optimisant certaines requêtes.

## 6.5 BPM

Le DAT annonce une brique BPM implémentée sur une solution Bonita et ayant pour objectif de « permettre aux AASQA de suivre la validation de certains processus métiers ».

L’atelier avec Atos a permis de confirmer qu’in fine ce périmètre applicatif a été limité au suivi des états de validation des rapports. Ce suivi est mis en œuvre par l’utilisation de flags en base.

**Constat : L’absence de ce composant est un frein à tout suivi de validation des processus métier.**

## 6.6 SITE WEB

### 6.6.1 Analyse statique

Le site web est construit en une seule page SPA (Single Page Application) en utilisant les briques techniques illustrées dans le schéma suivant :

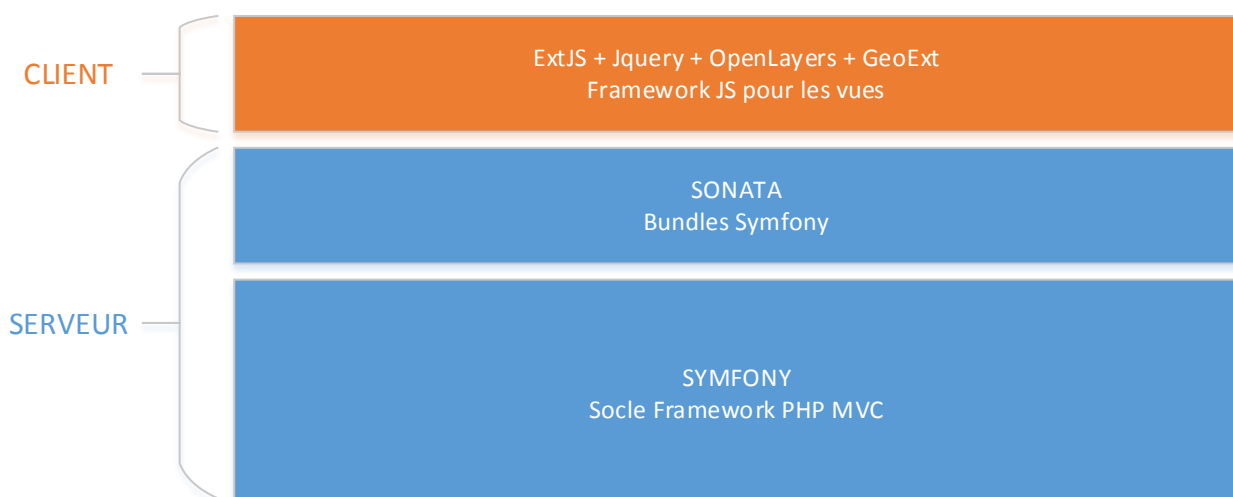


Figure 13: Briques techniques de l'application Web

Il s’agit ici des librairies et frameworks utilisés par le site internet et s’inscrivant dans le cadre des composants logiciels traités dans le cadre de l’architecture générale :

- Composants serveur :

- Symfony 2.3.11
- Bundles Sonata (Admin, Core, Doctrine2 ORM Admin, User).
- Composants client :
  - ExtJS 4.2.1
  - OpenLayers 2.13.1
  - GeoExt 2
  - JQuery 1.8.3

#### 6.6.1.1 Cohabitation ExtJS/SONATA

ExtJS est un framework Javascript permettant de construire des applications web cross-plateforme. Il est en général utilisé pour des applications web « data-intensive » (axées plus sur les données). Il dispose de composants graphiques prédéfinis (calendrier, grids,...).

SONATA est une collection de bundles Symfony apportant une rapidité de développements des écrans. Dans le projet Geod’air, les bundles suivants sont utilisés :

- Core Bundle : Bundle Noyau de SONATA
- Admin Bundle permettant de construire rapidement des pages d’administration du site web (CRUD)
- User Bundle permettant d’ajouter la gestion des utilisateurs et des rôles dans les pages d’administration

Le framework ExtJS et les bundles SONATA sont donc utilisés pour implémenter les différents écrans du site web avec une répartition illustrée dans la figure suivante :

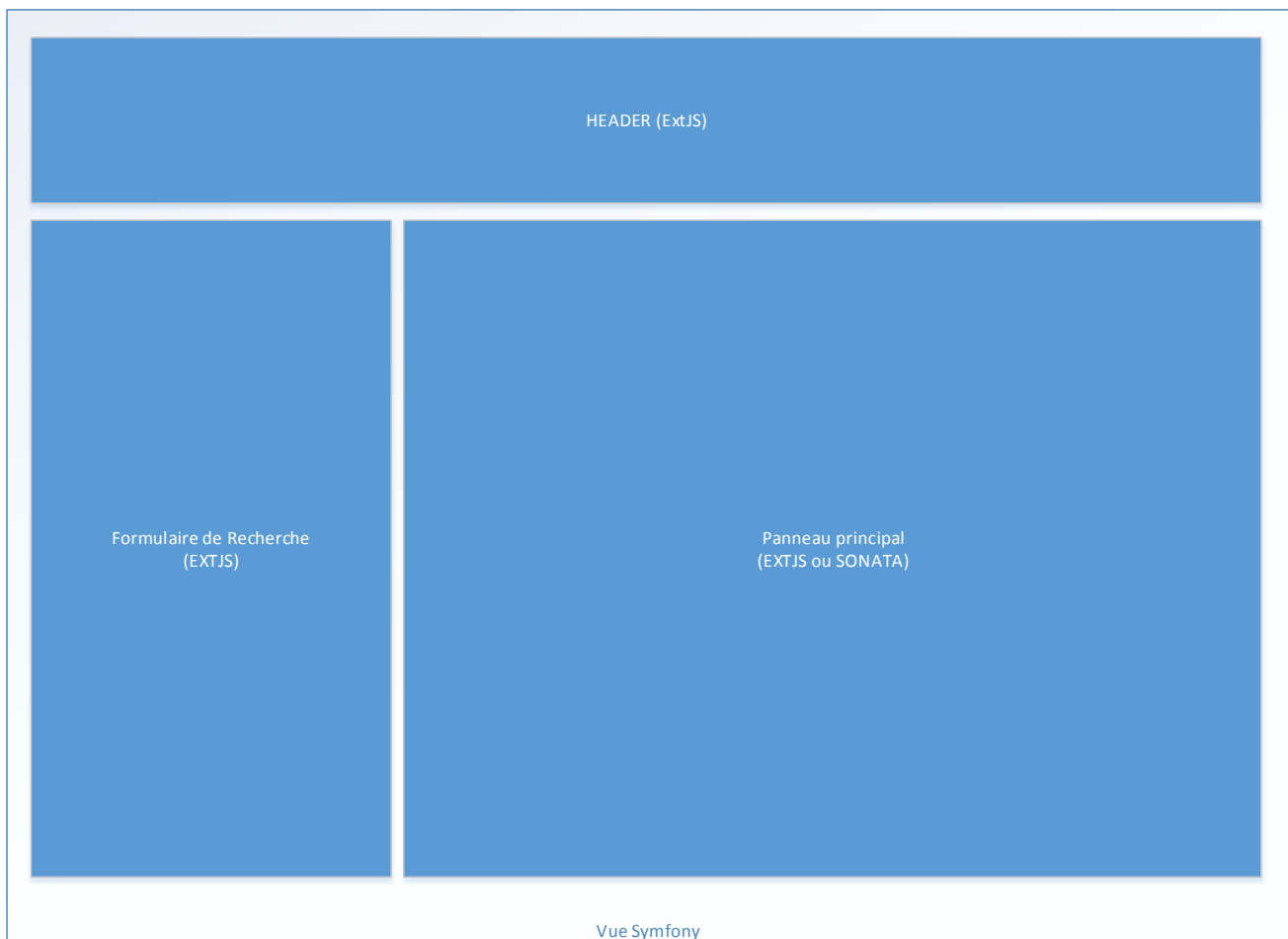


Figure 14: Répartition EXTJS/SONATA

L’implémentation du panneau principal dépend du type d’affichage. SONATA est utilisé pour l’affichage des écrans « CRUD » (la modification des référentiels) et pour l’affichage des détails du compte connecté. Tous les autres écrans sont assurés par ExtJS.

Le mix entre les 2 frameworks pour implémenter les écrans conduit à s’éloigner de l’état de l’art, à rendre le code moins lisible et le site moins performant.

En effet, de manière standard, la cohabitation entre un framework front-end (ExtJS) et un framework back-end (Symfony) se fait selon le schéma ci-dessous :



Figure 15: Echange Front/Back-End

Le framework Front-End a pour responsabilité de générer les composants HTML à partir des données brutes (en JSON ou XML) récupérées depuis le serveur en faisant des appels AJAX. Le flux transitant sur le réseau est moins volumineux et toute la vue (V de MVC) est portée uniquement pour le front-end. Ce qui n’est pas le cas sur le site Geod’Air.

Le code ExtJS ci-dessous montre un appel AJAX de l’url définie dans la variable « url ». Il s’agit d’un appel vers un contrôleur SONATA pour afficher un référentiel donné. La réponse de cet appel est du HTML et non des données brutes. (Ex :

[https://www.geodair.fr/admin/ineris/referentiels/classification/list?filter\[\]\[type\]=1&filter\[\]\[value\]=undefined&filter\[\]\[type\]=1&filter\[\]\[value\]=undefined&filter\[\]\[type\]=1&filter\[\]\[value\]=undefined&filter\[verrouOrganisme\]\[type\]=1&filter\[verrouOrganisme\]\[value\]=& dc=1481846567891](https://www.geodair.fr/admin/ineris/referentiels/classification/list?filter[][type]=1&filter[][value]=undefined&filter[][type]=1&filter[][value]=undefined&filter[][type]=1&filter[][value]=undefined&filter[verrouOrganisme][type]=1&filter[verrouOrganisme][value]=& dc=1481846567891))).

```
var url = 'admin/ineris/referentiels/' + entity + '/list?' + param0 + param1 + param2 + param3 + param4 + param5 + param6 + param7;
ajaxCustom.request(
{
  url: url,
  method: 'GET',
  success: function(response)
  {
    if (response.responseText == 'dc')
    {
      var msg = geolive.Global.translation.$disconnect_message;
      geolive.Global.sessionLost(msg);
    }
    else
    {
      nom_entite = entity
      Ext.getCmp('buttonRepositoriesExport').setDisabled(false);
      var result = response.responseText
      tmpMask.hide();
      Ext.getCmp('repositoryTabContainer').update(result);
      Ext.getCmp('repositoryTabContainer').getEl().scrollTo('top');
    }
  },
  failure: function(response, opts)
  {
    if (typeof response.status === '401')
    {
      var msg = geolive.Global.translation.$disconnect_message;
      geolive.Global.sessionLost(msg);
    }
    else
    {

```

Figure 16: Bout de code ExtJS

**Conseil :** Pour résoudre ce problème, nous conseillons de choisir un seul framework pour générer les composants HTML. Soit les écrans sont tous implémentés en ExtJS (côté poste de travail). Dans ce cas, il faudra recoder les écrans CRUD avec de l’ExtJS. Soit les vues sont toutes implémentées par du Symfony + SONATA.

**Remarque :** Le choix d’implémenter le site Geod’Air en une seule page est discutable. L’expérience utilisateur en est impactée étant donné qu’il n’est plus possible pour lui de revenir à un état précédent des écrans (surtout pour les formulaires de recherche).

Par ailleurs, ExtJS est un framework MVC. Une bonne pratique serait donc d’isoler chaque partie (Modèle, Vue, Contrôleur) dans des composants différents. Malheureusement ce principe de séparation des responsabilités n’est pas respecté. Nous nous retrouvons donc avec des fichiers JS de plus de 2000 lignes (Par exemple : Fichier repositories.js).

Une meilleure structuration du projet ExtJs comme recommandée<sup>2</sup> par l’éditeur aurait permis d’avoir un code plus lisible et donc plus facile à maintenir.

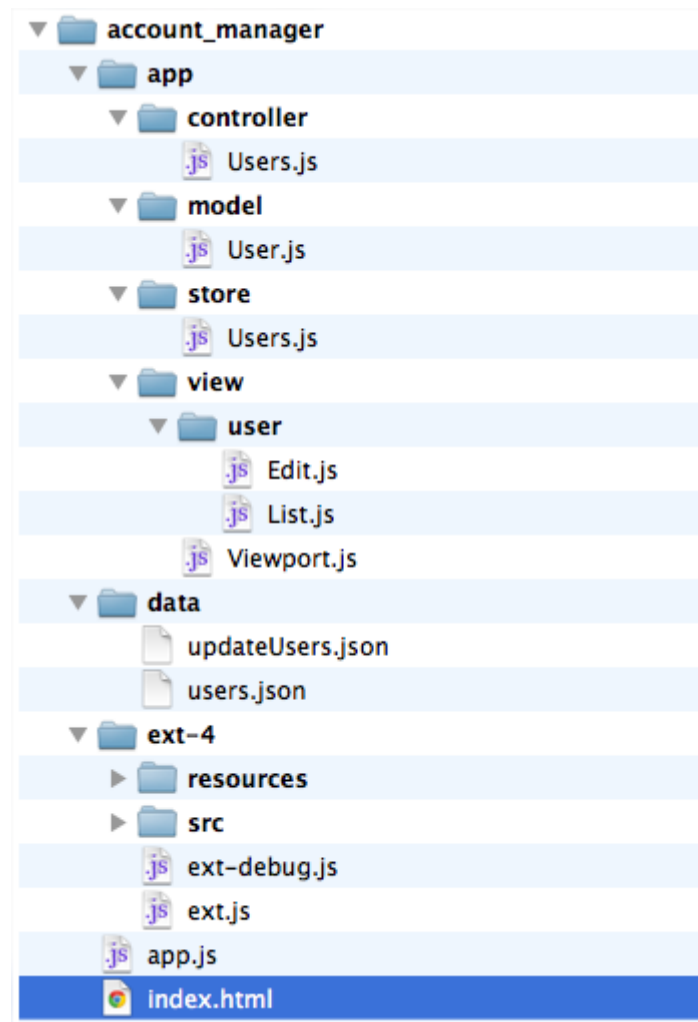


Figure 17: Recommandation de structuration d'un projet ExtJS

---

<sup>2</sup> [http://docs.sencha.com/extjs/4.2.6/#/guide/application\\_architecture](http://docs.sencha.com/extjs/4.2.6/#/guide/application_architecture)

### 6.6.1.2 Contrôleurs Symfony

L’une des bonnes pratiques de programmation Symfony est d’utiliser les classes « repository » et ne pas écrire des requêtes SQL directement dans les contrôleurs afin que ces requêtes puissent être réutilisées par plusieurs contrôleurs.

Dans le projet Geod’Air, certaines requêtes sont faites directement dans les contrôleurs Symfony (Par exemple, la classe ReferentielsAccessController).

```
/**
 * Select Organismes where code_nature matches with $nature param
 *
 * @access public
 * @param string $nature
 * @return string
 */
public function getQueryOrganismeByNature($nature)
{
    $natureOrg="";
    switch ($nature) {
        case 'AASQA':
        case 'LCSQA':
            $natureOrg = "WHERE o.code_nature_organisme=" . $nature . "' ";
            break;
    }

    $req = "SELECT DISTINCT o.code, o.nom, o.code_nature_organisme, r.code AS code_region FROM ".PgsqlDBController::tblORGANISME." o ";
    $req .= "INNER JOIN ".PgsqlDBController::tblDEPARTEMENT." d ON ( ";
    $req .= "    (substring(o.code_postal from 1 for 2) = d.code AND CAST(coalesce(o.code_postal, '0') AS integer)<97000) ";
    $req .= "    OR ";
    $req .= "    (substring(o.code_postal from 1 for 3) = d.code AND CAST(coalesce(o.code_postal, '0') AS integer)>97000) ";
    $req .= ") ";
    $req .= "INNER JOIN ".PgsqlDBController::tblREGION." r ON r.code = d.code_region ";

    return $req;
}
```

Figure 18: Extrait de code de la classe "ReferentielsAccessController"

### 6.6.1.3 Qualité de code

Le tableau ci-dessous liste des erreurs SONAR les plus critiques :

Types d'erreur	Occurrences	Conséquences/Risques	Exemple
"Remove this use of the "eval" function."	21	La fonction « eval » est un moyen d'exécuter de manière arbitraire un code. Son utilisation impacte négativement la lisibilité du code et le chemin d'exécution devient imprévisible.	
Plus de 2000 erreurs critiques SONAR liées en général à de la duplication de code ou de code trop complexe		Ces erreurs montrent clairement la difficulté à maintenir le code PHP. Le développement des futures évolutions souffrira de ce fait.	

## 6.6.2 Analyse dynamique

### 6.6.2.1 Conditions de test

#### 6.6.2.1.1 Caractéristiques de la connexion - étalonnage

Les analyses présentées ci-après intègrent des notions de durée qui sont intimement liées aux conditions de test. Nous ferons dans la mesure du possible abstraction des temps de téléchargement, mais afin de les

mettre tout de même en perspective, nous indiquons ci-dessous les résultats d'un ensemble de tests permettant de caractériser la connexion utilisée.

<http://www.speedtest.net/fr/>

	Latence	Download	Upload
Tir 1	10	43,67	21,06
Tir 2	9	46,57	23,45
Tir 3	10	47,63	22,93
Tir 4	10	46,13	23,02
Tir 5	9	46,84	24,28
Tir 6	10	43,3	22,88
Tir 7	10	37,89	21,82
Tir 8	10	43,6	22,77
Tir 9	10	46,03	22,49
Tir 10	10	45,54	23,94
Moyenne	9,8	44,72	22,864
Ecart type	0,421	2,824	0,944

#### 6.6.2.1.2 Profils de connexion

Environnement de production :

url : <https://www.geodair.fr/>

compte : [jean-francois.mony@capgemini.com](mailto:jean-francois.mony@capgemini.com) // Cs3QC47N

Environnement de recette :

url : <https://www.recette.geodair.fr>

compte : [jean-francois.mony@capgemini.com](mailto:jean-francois.mony@capgemini.com) // Cs3QC47N

#### 6.6.2.1.3 Outils de test

Afin de mener les analyses qui suivent nous utilisons les outils suivants :

- Pagespeed
- Yslow
- Developer tools de Chrome
  - Network
  - Timeline

#### 6.6.2.2 Analyse des accès ouverts

Accès au server cartographique

<https://www.geodair.fr/geoserver/web/> -> accès à l'interface d'admin de géoserver

Permet d'identifier la version de Géoserver en place : 2.4.3 (risques possibles en cas de faille de sécurité connue sur une version)

Permet d'identifier les couches existantes et de les exporter.

Toutes les couches de Geod'air sont accessibles en WMS / WFS

**N'y a-t-il pas un problème de droits sur les couches IGN qui sont mises à disposition sans contrôle ?**

Quel est le format des sources des données ign ? (cascading, images sur le serveur...)

Accès à l'admin sonata

La consultation des sources des pages du site accessible sans connexion (login et demandeCreation) permet d'identifier que le site est basé sur sonata/symfony.

Test de l'adresse que l'on peut déduire des ces informations : <https://www.geodair.fr/admin> -> redirection vers <https://www.geodair.fr/admin/login>

L'accès est protégé par authentification -> mêmes profils que pour le site geodair.fr

Connexion avec profil geodair -> redirection vers page d'accueil geod'air <https://www.geodair.fr/accueil>

Accès à la page (<https://www.geodair.fr/admin>) via son url en tant qu'utilisateur déjà connecté -> redirection vers <https://www.geodair.fr/admin/dashboard>

-> Accès à toutes les données de la base Postgresql déclarées dan SONATA en lecture (y compris les users)

### 6.6.2.3 Analyse du site geodair.fr

#### 6.6.2.3.1 Principes de navigation

Contexte : profil d'un utilisateur n'ayant que des droits en consultation

Accueil

    Référentiel

        Recherche

            Liste des référentiels

            Liste des entités (dépendant du choix de référentiel)

        Tableau

        Carte

            Affichage des fonds, des communes et des couches métier (stations, ZAS)

            Restriction de l'affichage des couches métiers en cas de filtre sur ces couches au niveau de la recherche

Accès aux données

    Recherche

    Mesure

    Statistique

    Carte

Supervision

    Recherche

    Données brutes

        Synthèse

        Détails

    Données vérifiées

        Synthèse

        Détails

    Données manuelles

        Synthèse

        Détails

    Indice ATMO

        Synthèse

        Détails

#### 6.6.2.3.2 Focus sur la cartographie

Openlayers 2.13.1 est intégré au site comme client cartographique.

3 types de services cartographiques sont accédés :

- Services WMS
- Services de tuiles
- Service WFS

### 1) Les services WMS

Ils correspondent aux couches IGN (planIGN et orthophoto). Ces 2 couches sont servies par le Geoserver du site Geod'air.

OpenLayers accède à ces couches de façon tuilée.

Geoserver intègre un gestionnaire de cache : GeoWebCache. Le principe du cache est de fixer une grille de découpage et de stocker les images correspondant à chaque cellule de la grille.

Ceci permet de gagner en efficacité puisque le serveur cartographique n'est plus amené à recalculer les images à chaque sollicitation.

L'accès à <https://www.geodair.fr/geoserver/gwc/demo> montre que geoWebCache est activé pour l'ensemble des couches disponibles.



Layer name:	Enabled:	Grids Sets:		
<b>ineris:OrthoIGN</b> <a href="#">Seed this layer</a>	true	EPSG:4326 EPSG:900913	OpenLayers: <a href="#">[jpeg, png]</a> OpenLayers: <a href="#">[jpeg, png]</a>	KML: <a href="#">[jpeg, png]</a>
<b>ineris:PlanIGN</b> <a href="#">Seed this layer</a>	true	EPSG:4326 EPSG:900913	OpenLayers: <a href="#">[jpeg, png]</a> OpenLayers: <a href="#">[jpeg, png]</a>	KML: <a href="#">[jpeg, png]</a>
<b>ineris:communeLayer</b> <a href="#">Seed this layer</a>	true	EPSG:4326 EPSG:900913	OpenLayers: <a href="#">[jpeg, png]</a> OpenLayers: <a href="#">[jpeg, png]</a>	KML: <a href="#">[jpeg, png]</a>
<b>ineris:stationLayer</b> <a href="#">Seed this layer</a>	true	EPSG:4326 EPSG:900913	OpenLayers: <a href="#">[jpeg, png]</a> OpenLayers: <a href="#">[jpeg, png]</a>	KML: <a href="#">[jpeg, png]</a>
<b>ineris:zasLayer</b> <a href="#">Seed this layer</a>	true	EPSG:4326 EPSG:900913	OpenLayers: <a href="#">[jpeg, png]</a> OpenLayers: <a href="#">[jpeg, png]</a>	KML: <a href="#">[jpeg, png]</a>

On peut également observer que le cache est activé pour les projections EPSG :4326 et EPSG :900913. Ceci est confirmé par les informations accessibles via le `getcapabilities` <https://www.geodair.fr/geoserver/gwc/service/wms?request=getcapabilities&tiled=true>.

Il semble cependant que le cache ne soit pas exploité.

En effet, Geoserver met en place deux mécanismes d'accès à son cache, soit par appel d'une url dédiée <https://www.geodair.fr/geoserver/gwc/service/wms?>, soit par appel de l'url des services wms en précisant le paramètre tiled <https://www.geodair.fr/sig/geoserver/wms?tiled=true>. Cette seconde méthode implique que ce mode appelé 'direct integration' soit activé dans la configuration de Geoserver.

L'accès à <https://www.geodair.fr/sig/geoserver/wms?request=getcapabilities&tiled=true> montre que ce mode n'est pas activé (ce qui est la configuration par défaut de Geoserver).

Enfin les appels aux services par OpenLayers <https://www.geodair.fr/sig/geoserver/wms?LAYERS=OrthoIGN&CRS=EPSG%3A3857&VERSION=1.3.0&FORMAT=image%2Fpng&EXCEPTIONS=INIMAGE&SERVICE=WMS&REQUEST=GetMap&STYLES=&BBOX=763320.24425,4763479.587375,1563352.112875,5563511.456&WIDTH=256&HEIGHT=256> montrent que l'accès ne se fait pas via l'url de GeoWebCache et qu'il n'y a pas de paramètre 'tiled=true' dans l'url.

**Il serait pertinent de modifier les appels aux services par OpenLayers en conséquence afin d'optimiser les accès sur l'ensemble des couches de base qui évoluent peu dans le temps.**

### 2) Le service de tuiles

Il correspond à la couche OpenStreetMap. Ce service est accédé directement depuis les serveurs d'OpenStreetMap.

### 3) Les services WFS

Ils correspondent aux couches des sites, des ZAS et des communes.



La première question qui se pose concerne le choix du vectoriel pour ces couches. Il est probable que ce choix soit lié à l’interactivité souhaitée (highlight et tooltips). On notera cependant qu’une approche mixte (raster et vecteur) est souvent mise en œuvre pour éviter le chargement de trop nombreux vertex qui alourdissent la navigation.

Se pose également la question des droits dans la mesure où la couche ign des communes est en accès libre dans un format qui permet sa réutilisation.

On notera que le WFS communal n’est normalement accessible qu’à certaines échelles (sauf anomalie) et qu’il est appelé dans le cadre d’un tuilage vecteur. Ces deux éléments permettent de limiter la volumétrie de données téléchargée en une fois.

La seconde question concerne le choix du WFS/gml. Il s’agit d’un service ogc normée, mais **l’état de l’art oriente vers l’usage de services geojson** pour ce type d’application. Le geojson est beaucoup moins verbeux et son interprétation par le javascript beaucoup plus efficace.

Enfin, la dernière question concerne le contenu des services WFS. Les extraits ci-dessous des trois flux (pour un seul objet : un site, une ZAS et une commune) montrent qu’en plus de la verbosité liée à la norme xml, le flux comporte également **un grand nombre d’informations qui ne semblent pas utilisées par l’application**. A cela s’ajoute le fait que **les coordonnées des points des géométries sont données avec une précision nanométrique (10<sup>-9</sup> mètres)**. Tous ces éléments augmentent inutilement la taille de ces flux.

### Sites

```
<ineris:stationLayer gml:id="stationLayer.FR40007.1">
  <gml:description>Matoury Kalou</gml:description>
  <ineris:code>FR40007</ineris:code>
  <ineris:codeue>FR40007</ineris:codeue>
  <ineris:idversion>1</ineris:idversion>
  <ineris:libversion>Version 1</ineris:libversion>
  <ineris:id_etat>1</ineris:id_etat>
  <ineris:code_commune>97307</ineris:code_commune>
  <ineris:cle_commune>2014</ineris:cle_commune>
  <ineris:code_nature_aire>suburban</ineris:code_nature_aire>
  <ineris:nom>MATOURY KALOU</ineris:nom>
  <ineris:alias>MATOURY KALOU</ineris:alias>
  <ineris:est_implantation_conforme>true</ineris:est_implantation_conforme>
  <ineris:longitude>-52.3238</ineris:longitude>
  <ineris:latitude>4.84816</ineris:latitude>
  <ineris:geometrie>
    <gml:Point srsDimension="2" srsName="http://www.opengis.net/gml/srs/epsg.xml#3857">
      <gml:pos>-5824658.772369087 540339.8868427121</gml:pos>
    </gml:Point>
  </ineris:geometrie>
  <ineris:id4geoserv>1318</ineris:id4geoserv>
  <ineris:altitude>12</ineris:altitude>
  <ineris:code_mobilite_station>FIXE</ineris:code_mobilite_station>
  <ineris:code_nature_station>2</ineris:code_nature_station>
  <ineris:code_nature_projection>WGS84</ineris:code_nature_projection>
  <ineris:date_creation>2014-07-16Z</ineris:date_creation>
  <ineris:date_es>2014-07-16Z</ineris:date_es>
  <ineris:version_debut>2014-07-16Z</ineris:version_debut>
  <ineris:verrou_organisme>23</ineris:verrou_organisme>
  <ineris:code_zas>FR40N10</ineris:code_zas>
  <ineris:cle_zas>1</ineris:cle_zas>
</ineris:stationLayer>
```

### ZAS

```
<ineris:zasLayer gml:id="zasLayer.FR39N20.1">
  <ineris:code>FR39N20</ineris:code>
  <ineris:idversion>1</ineris:idversion>
  <ineris:libversion>Version 1</ineris:libversion>
  <ineris:id_etat>1</ineris:id_etat>
  <ineris:code_organisme>21</ineris:code_organisme>
  <ineris:cle_organisme>1</ineris:cle_organisme>
  <ineris:nom_majuscule>MARTINIQUE-ZR</ineris:nom_majuscule>
</ineris:zasLayer>
```

```

<ineris:nom>MARTINIQUE-ZR</ineris:nom>
<ineris:code_type_zas>ZR</ineris:code_type_zas>
<ineris:aire>914</ineris:aire>
<ineris:population>224190</ineris:population>
<ineris:population_annee>2011</ineris:population_annee>
<ineris:id_zonage>1</ineris:id_zonage>
<ineris:geometrie>
  <gml:MultiSurface srsDimension="2" srsName="http://www.opengis.net/gml/srs/epsg.xml#3857">
    <gml:surfaceMember>
      <gml:Polygon srsDimension="2">
        <gml:exterior>
          <gml:LinearRing srsDimension="2">
            <gml:posList>-6770840.60298671 1628363.703287266 -6769857.540563515 1627189.8007291236 -
            6770639.114708375 1626307.1388008364 -6771029.400843098 1624755.8291458497 -
            6772132.688316349 1623109.6700511232 -6773238.870096362 1623145.0726799022 -
            6773523.068756358 1622879.7840422527 -6773149.8145037275 1622171.0584655292 -
            6772452.509213397 1621799.9221293367 -6772685.834866101 1621464.6512192304 -
            6773495.795481112 1621479.4779854226 -6774445.239418088 1620143.4943506136 -
            6775144.437139762 1619750.206167015 -6775551.643837083 1619686.5358826707 -
            6776364.276119875 1619751.2405231192 -6776709.143902351 1619959.0320174117 -
            6777527.230840192 1620889.8620344626 -6777730.388910889 1621247.9975133883 -
            6778637.642760854 1622484.9596382447 -6777750.203780251 1623266.2236046554 -
            6777119.8015038865 1624589.8405633194 -6777634.097551353 1625576.246985395 -
            6777462.776855022 1626136.6568181883 -6776849.963058204 1625532.5642735634 -
            6776289.24678308 1625670.739843721 -6776398.673842529 1626677.3058888556 -
            6775306.406998866 1626464.5166372887 -6775401.473844003 1627574.810907688 -
            6775623.22269663 1628262.532397509 -6776740.424679264 1628357.1501606682 -
            6779263.146979622 1625662.9229030162 -6780155.595337311 1626733.866052889 -
            6780641.727553606 1627317.983527128 -6780524.396810309 1628079.3911136405 -
            6781293.614491691 1628173.7782653393 -6783105.227884861 1627393.3989985788 -
            6785439.374967815 1627734.49584099 -6786443.47677477 1628369.9115140147 -
            6787559.899947935 1628084.794509983 -6788032.228547371 1628256.324196879 -
            6788187.741876009 1629478.449103481 -6788542.739732149 1629617.6807827263 -
            6788909.648773802 1628515.4603738475 -6789607.065383622 1627743.4630590829 -
            6790998.336379558 1629216.428871285 -6791243.573217775 1628308.5191158678 -
            6792650.317622929 1629445.9119998082 -6794074.761827121 1629002.2388158385 -
            6795450.670733326 1628189.0687868313 -6795810.34400808 1627049.3177198789 -
            6796422.601207442 1626561.0820423544 -6796776.263229691 1626364.732768781 -
            6797532.790489122 1626625.6891004676 -6798124.898860652 1627670.5757622186 -
            6799507.709575287 1628373.2455623357 -6799414.980439456 1629963.1763832546 -
            6800179.856660696 1630853.8982820867 -6800069.093767358 1632010.491993818 -
            6801392.682512888 1632791.587524908 -6801145.330604346 1633968.9678894093 -
            6800260.6746110115 1635293.8303110912 -6799095.71613986 1636412.674687882 -
            6798696.524445876 1636349.9978599644 -6797899.254252814 1636349.5378471212 -
            6796214.322440167 1638470.8606602002 -6794684.013400233 1637822.6566876348 -
            6795289.257471675 1636738.7116910324 -6793806.815812781 1636056.5113579016 -
            6791037.96611828 1636071.8066130243 -6790795.623586823 1636391.8590636614 -
            6789978.315885419 1637505.0013813365 -6789298.710394127 1637274.410500877 -
            6790174.79478667 1638540.788550928 -6790559.069668888 1638532.737632639 -
            6790561.296058703 1639701.0676102932 -6789770.148437635 1641109.7370781614 -
            6787937.273021724 1642673.4323689302 -6787315.664985135 1643705.2895049131 -
            6785796.042616315 1644166.590422382 -6784613.7183046 1644000.9352593513 -
            6784410.226275429 1644347.4318809703 -6784858.398545365 1647007.6304080454 -
            6786135.010465779 1649476.0685947887 -6787471.734911227 1650318.9182664875 -
            6788425.520308344 1652060.9736647136 -6789278.672885783 1652805.863124488 -
            6789882.692442828 1651569.6095543192 -6793144.464842561 1653778.2917892165 -
            6795087.657873848 1654109.3864365262 -6797593.904889569 1655311.5832264465 -
            6798921.501136769 1656222.648705415 -6799925.825582705 1657483.5322299688 -
            6800131.432682202 1657134.7831204655 -6799788.012053104 1655563.4004460317 -
            6800495.002139132 1655438.412241848 -6801710.16570063 1654116.4065648834 -
            6802110.581909015 1654681.36069823 -6803085.629328872 1654878.8502065046 -
            6803392.091887026 1653569.9932544758 -6804353.669648499 1653526.0321267338 -
            6805148.1568542905 1651797.9143088767 -6804954.34962082 1651110.0144259976 -
            6807629.8022625465 1649575.3680539213 -6809151.53970169 1651808.0407773915 -

```

6810976.622753245	1655309.1663457998	-6810991.539565012	1656157.9661098712	-
6810446.296699106	1658401.4676602494	-6810162.320678093	1659148.15665624	-
6810453.866424479	1660297.87328937	-6812435.4646800915	1662262.5742960668	-
6813220.489729164	1663665.253590791	-6814320.6602566745	1664045.524016101	-
6815743.8799464675	1666492.7250963217	-6815976.203723753	1668510.3064517358	-
6814927.351481499	1671154.757439633	-6813873.267223177	1672863.597089214	-
6813202.678610638	1673291.2284450629	-6810516.094019833	1674749.4686668802	-
6809019.960063571	1675041.1039433288	-6807065.635083204	1675213.2993653184	-
6804457.6420529	1674626.1121216256	-6801486.970121591	1673397.6479483112	-
6800672.445407456	1672487.2230832244	-6799906.789949779	1672278.768694131	-
6799065.4372383645	1671492.9919628133	-6797685.632149982	1671157.2910144841	-
6797392.861889196	1670566.7447218234	-6796349.575621481	1669945.810991941	-
6795365.511322868	1670370.0526146297	-6794877.37535574	1669781.9432656514	-
6793836.649436315	1669708.2434449524	-6793951.753789794	1668845.511127637	-
6793141.5705358	1668790.4686215145	-6792571.837381921	1667248.5217609257	-
6791835.904228286	1666987.0301060372	-6790878.779246446	1667295.8461709535	-
6791019.375763318	1666253.4633404652	-6789960.504766893	1664744.7163128827	-
6788431.531560848	1663573.9573135956	-6787482.978179796	1661907.1968457056	-
6786862.038060152	1661398.0269986792	-6786342.398677129	1659527.2073834513	-
6785800.384076457	1658998.9782272042	-6784466.999215735	1660936.0647629357	-
6783051.015292844	1661700.0971212205	-6781485.306654837	1661547.219713432	-
6778093.847048328	1663639.5800768812	-6776918.758503515	1663389.178352644	-
6776334.776454813	1662832.316930581	-6776299.822134704	1661597.1810527379	-
6777017.276252868	1661364.7580654614	-6777661.593465579	1662437.099051066	-
6779002.436732184	1661767.3266382802	-6778129.803243855	1660787.5663775424	-
6779235.985023867	1659767.2105799417	-6778346.876250902	1659231.494374977	-
6778913.715098021	1658517.0320687657	-6780425.211144012	1658699.703191107	-
6780689.149656683	1659780.7935633785	-6782120.606988794	1660360.03408182	-
6783328.200824919	1660439.8073247788	-6783625.423865338	1660161.6956479105	-
6784256.271419664	1659208.4728780184	-6783487.276377263	1658524.2836380517	-
6784636.650119704	1657959.7034449696	-6783565.979257255	1656463.5410473936	-
6782236.824537183	1657265.649792484	-6782655.051864093	1656623.6384785182	-
6783106.4523992585	1655859.3004137822	-6782801.548313977	1655400.8928838545	-
6781350.276112505	1656454.5636515766	-6780934.052536429	1656415.3163562706	-
6781006.298885954	1655084.7427273593	-6780343.168679297	1654599.188947402	-
6779506.268747514	1655022.9402675068	-6779573.283080972	1654148.9753827334	-
6780134.44463406	1654034.6969955452	-6780570.371760007	1653259.850290173	-
6781067.190647418	1653623.1610037836	-6782439.203371446	1653324.9857494903	-
6782068.398147611	1652170.4099281987	-6783386.7548770765	1652650.508180975	-
6784033.966396549	1651129.9216025418	-6783084.1885011	1649894.3256255658	-
6782579.354610353	1649280.3473993272	-6781570.577384784	1649314.7509561894	-
6780830.525409991	1650805.0798730296	-6778492.482144861	1650004.5579326318	-
6778425.913089365	1650723.3810537588	-6777415.132112962	1650641.1071584476	-
6777784.824141887	1649219.134497837	-6778677.272499576	1648905.133059349	-
6779667.904648147	1649032.9653460416	-6779982.159570655	1648569.044242917	-
6778356.1157686375	1647990.4204796066	-6779176.206457311	1645553.879980624	-
6775976.216374969	1645283.755768831	-6777185.368683966	1644564.2822818705	-
6775849.757433427	1643569.4316614175	-6776272.548859461	1642867.606952587	-
6775651.942698288	1642074.9284947529	-6774267.907469255	1642285.3177306787	-
6773718.657101681	1641818.5297610187	-6773797.471301162	1641459.6449109358	-
6772960.014771924	1640668.0457285456	-6774272.248929396	1640392.6826739933	-
6773935.284830764	1639539.2372402295	-6773111.631918385	1639818.271629146	-
6772759.862327478	1639228.8064645163	-6773182.876392494	1638502.2591793998	-
6772790.141228975	1638533.7727505614	-6771672.048263446	1639697.5020401939	-
6771066.9155114945	1639464.0157993697	-6772420.115241578	1637356.985947035	-
6771197.047996232	1635871.7040030365	-6772520.970700236	1635525.8982070074	-
6771106.211291744	1635431.3688083144	-6771140.9429728715	1634266.8000554508	-
6772127.790258754	1634300.1482756974	-6772165.527566134	1632815.7348317609	-
6770993.889925534	1633439.4332088719	-6770860.863134037	1631461.911176129	-
6770778.820669321	1631075.6925302471	-6770073.389056165	1630776.9778661095	-
6769866.000844816	1630441.4741921492	-6769476.048668568	1629327.1461750448	-
6769858.208480461	1628650.43296513	-6770605.273583175	1628680.4397455368	-
6770840.60298671	1628363.703287266	</gml:posList>		

</gml:LinearRing>

```

        </gml:exterior>
      </gml:Polygon>
    </gml:surfaceMember>
  </gml:MultiSurface>
</ineris:geometrie>
<ineris:id4geoserv>74</ineris:id4geoserv>
<ineris:date_creation>2009-12-31Z</ineris:date_creation>
<ineris:date_es>2009-12-31Z</ineris:date_es>
<ineris:version_debut>2009-12-31Z</ineris:version_debut>
<ineris:version_fin>2015-09-27Z</ineris:version_fin>
<ineris:motif_fin>Fin de zonage</ineris:motif_fin>
<ineris:description_changement>Importation initiale</ineris:description_changement>
<ineris:verrou_organisme>21</ineris:verrou_organisme>
</ineris:zasLayer>

```

### Communes

```

<ineris:communeLayer gml:id="communeLayer.34324.2014">
  <ineris:code>34324</ineris:code>
  <ineris:code_uts>34324</ineris:code_uts>
  <ineris:code_departement>34</ineris:code_departement>
  <ineris:code_unite_urbaine>34303</ineris:code_unite_urbaine>
  <ineris:code_epci>243400769</ineris:code_epci>
  <ineris:idversion>2014</ineris:idversion>
  <ineris:code_zas>FR08N10</ineris:code_zas>
  <ineris:geometrie>
    <gml:MultiSurface srsDimension="2" srsName="http://www.opengis.net/gml/srs/epsg.xml#3857">
      <gml:surfaceMember>
        <gml:Polygon srsDimension="2">
          <gml:exterior>
            <gml:LinearRing srsDimension="2">
              <gml:posList>364164.0683308481      5347565.0870516      363865.48838968156
              5347960.573306383      365868.3739734712      5350772.973078632      367811.7790679762
              5351323.903259357      368648.161530161      5350787.152778014      364164.0683308481
              5347565.0870516</gml:posList>
            </gml:LinearRing>
          </gml:exterior>
        </gml:Polygon>
      </gml:surfaceMember>
    </gml:MultiSurface>
  </ineris:geometrie>
  <ineris:id4geoserv>13786</ineris:id4geoserv>
  <ineris:nom_majuscule>VALRAS-PLAGE</ineris:nom_majuscule>
  <ineris:code_nature_commune>C</ineris:code_nature_commune>
  <ineris:est_bureau_distributeur>false</ineris:est_bureau_distributeur>
  <ineris:population>4465</ineris:population>
</ineris:communeLayer>

```

### Choix de la projection de travail

OpenLayers fixe la projection d’affichage à EPSG :3857. Il s’agit d’une projection mercator sphérique dérivée du WGS84 (EPSG :4326).

C’est la projection de prédilection des sites cartographiques couvrant le monde entier, ce qui est le cas d’OpenStreetMap. Le choix de cette projection permet donc d’assurer que l’ensemble des couches seront bien superposables.

Il est à noter cependant que cette projection a été créée pour permettre un affichage sans trop de déformations sur l’ensemble du globe, le corolaire est qu’elle ne peut servir à réaliser des mesures précises car la valeur du mètre est très approximative selon les latitudes.

Comme cité plus haut, le site ne proposant pas de changer la projection d’affichage, **il serait pertinent que les projections de stockage et d’affichage soient les mêmes pour éviter les calculs de reprojection inutiles.**

Remarque :

L'ajout des ZAS à la carte provoque une mise à l'échelle automatique sur France + DOM/TOM ce qui déclenche le rechargement des fonds WMS.

Ce comportement n'est pas observé pour la couche des sites

Sur sites, ZAS et communes, le survol de la souris met l'objet en évidence et un arrêt de la souris provoque l'affichage d'un tooltip contenant le code et le nom de l'objet.

### 6.6.2.3.3 Analyse des pages

#### 6.6.2.3.3.1 Remarques transverses

Nous indiquons ici les constats récurrents réalisés sur l'ensemble des pages analysées.

Ces constats sont issus des outils Yslow et Pagespeed

- Utiliser la mise en cache du navigateur : la majorité des pages ne comporte pas d'information sur la durée de mise en cache (délais d'expiration) ce qui laisse le navigateur maître de la stratégie de mise en cache et peut occasionner des variations de comportement entre les navigateurs. Certaines pages (services) forcent la non mise en cache.
- Eviter le redimensionnement des images par le navigateur et diffuser les images directement à la bonne échelle. Ceci économise la bande passante ainsi qu'un calcul de mise à l'échelle par le navigateur
- Utiliser la compression : seuls les services wfs sont compressés par geoserver. Aucune des autres ressources n'est compressée alors que les navigateurs cibles autorisent l'échange de contenus compressé ce qui permet de fortement réduire les volumes échangés. Le gain serait de 70% à 90% selon les ressources concernées.
- De nombreuses images ne sont pas optimisées autant que possible (niveau de compression)
- Optimiser les ressources (Minify) : cette mesure permet de réduire la taille des ressources en supprimant tous les caractères inutiles. Cette action peut être considérée comme redondante en cas de mise en œuvre de la compression.
- Mieux répartir les phases d'interprétation du javascript pour éviter les latences.
- Optimisation des bibliothèques : openlayers et ExtJS sont des bibliothèques de grande taille et modulaires. Elles sont proposées par défaut dans une version complète intégrant toutes les fonctionnalités afin de faciliter les actions de développement. Il est fortement recommandé de construire sa propre distribution pour les environnements de production afin de n'intégrer que les fonctionnalités utilisées et ainsi disposer de bibliothèques optimisées. Ceci a un impact sur les temps de téléchargement, sur les temps d'interprétation par le navigateur et sur l'empreinte mémoire globale.

Pour openlayers (<https://www.geodair.fr/js/e58a428.js>), la bibliothèque utilisée correspond à la version 2.13.1 standard et contient donc la quasi-totalité des fonctionnalités d'openlayers qui ne sont pour la plupart pas utilisées ici.

<http://geotribu.net/node/52>

<http://docs.openlayers.org/library/deploying.html>

Pour extjs (<https://www.geodair.fr/js/e4c801f.js>), la bibliothèque utilisée correspond à la version 4.2.1.883 et au fichier standard : ext-all.js qui correspond à la version complète d'ExtJS.

[http://docs.sencha.com/extjs/4.0.7/#/guide/getting\\_started](http://docs.sencha.com/extjs/4.0.7/#/guide/getting_started)

<http://www.rahulsingla.com/blog/2010/12/extjs-custom-build-with-selected-components>

<http://blog.mdsohelrana.com/2012/10/21/how-to-compile-extjs-for-production-to-include-only-the-needed-ui-elements/>

- Les appels en POST ou en GET ne suivent pas la logique préconisée dans le cadre d'applications REST alors que symfony est conçu pour permettre cette approche. (Rien n'indique au niveau conception de la volonté de respecter une normalisation des services, cependant les services mis en œuvre correspondent plus à une approche REST qu'à une approche SOAP. Le principe du REST est entre autre d'exploiter de la façon la plus optimale possible le protocole http. De ce fait les actions de CRUD sont mappées sur les actions http

existantes : POST = Create, GET = Read, PUT = Update/replace, Patch = Update/modifie, DELETE = Delete.). L'usage de POST pour des actions de Read est contraire à la logique REST.

- Eviter d'avoir deux feuilles de style pour arriver au même rendu : la partie « données de mesures » est affichée dans un tableau Extjs qui est mis en forme par les css ExtJS et la partie référentielles est affichée dans un tableau généré et mis en forme par Sonata avec des css Sonata surchargées dans la page pour assurer la cohérence visuelle avec la partie Extjs. Une modification de rendu sur les tableaux implique donc une double modification sur 2 css distinctes.

Nous reproduisons ci-dessous un exemple HTML pour une cellule de tableau EXTJS puis Sonata pour mettre en évidence la différence de structure et de traitement css

```
<td role="gridcell" class="x-grid-cell x-grid-td x-grid-cell-gridcolumn-1109 x-unselectable " id="ext-gen3279">
<div unselectable="on" class="x-grid-cell-inner " style="text-align:left;">2016/06/13 00:30:00
</div>
</td>
```

```
<td class="sonata-ba-list-field sonata-ba-list-field-actions" objectId="01002~2014">
<div class="btn-group">
<a href="javascript:ajaxSonata('/admin/ineris/referentiels/commune/01002~2014/show',
'01002~2014')" class="btn view_link btn-small button-application-geolive" title="Afficher"> 1,
<i class="icon-zoom-in"></i>
</a>
</div>
</td>
```

#### 6.6.2.3.3.2 Page d'authentification

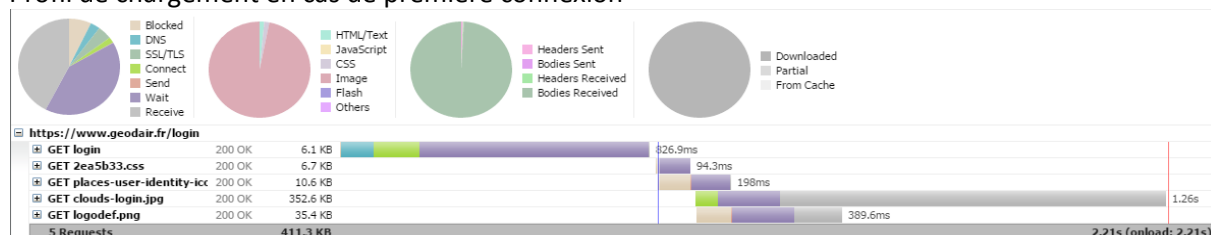
<https://www.geodair.fr/login>

Score Yslow : Grade A

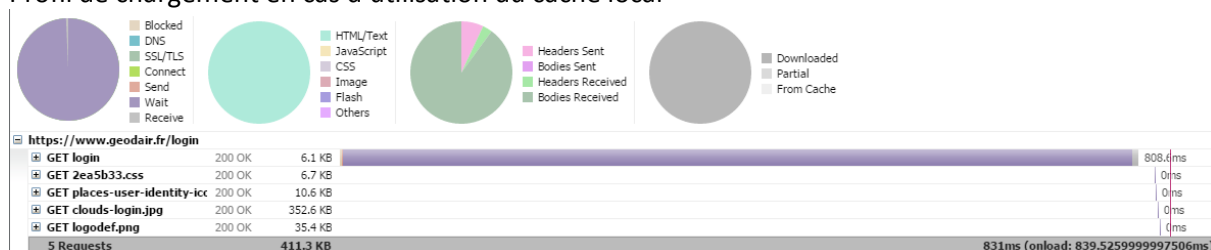
Score Pagespeed : 95/100

Il y a assez peu de choses à dire concernant cette page qui est assez simple.  
C'est une page html standard structurée en div et mise en forme par css.

#### Profil de chargement en cas de première connexion



#### Profil de chargement en cas d'utilisation du cache local



On note uniquement un temps d'attente un peu élevé sur l'accès à la page : 0,5 à 1s.



### 6.6.2.3.3 Page d’accueil

<https://www.geodair.fr/accueil>

Score Yslow : Grade D

Score Pagespeed : 13/100

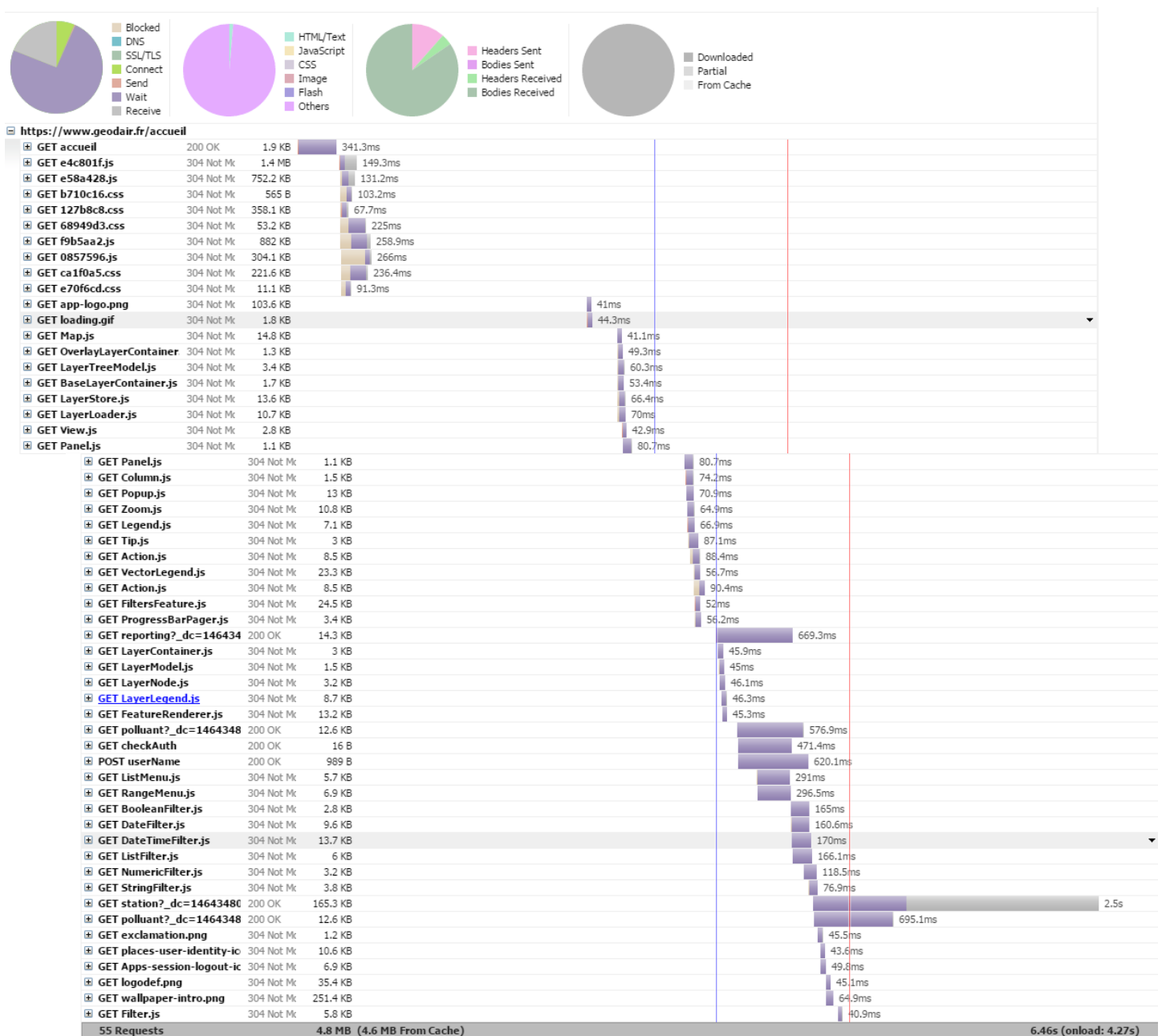
Nb requêtes au serveur : 55

Volume de données transféré : 4,9 MB

Cette page est la page principale du site.

Elle est structurée sur la base d’un viewport ExtJS.

Toute la navigation qui suivra se fera en restant au sein de cette page en exploitant les possibilités d’Extjs et de ses composants qui permettent de construire une application fenêtrée au sein du navigateur.

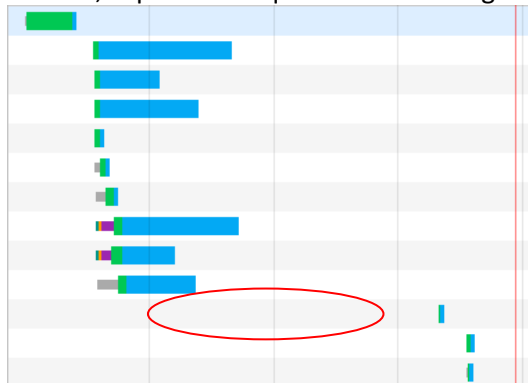


Les graphs ci-dessus montrent plusieurs choses :

- Indépendamment du sujet de la stratégie de cache explicité dans les remarques transverses, on constate un volume global chargé de 4,9Mb ce qui semble assez important, notamment lorsqu’il s’agit d’un premier accès
- Considérant le périmètre fonctionnel de la page d’accueil à ce stade, on peut se poser la question de la pertinence du téléchargement d’un certain nombre de ressources parmi lesquelles on peut citer :
  - Les fichiers javascript liés à openlayers
    - [e58a428.js](#) -> 752 KB => non nécessaire pour le rendu à ce stade
  - Les fichiers javascript liés à Extjs
    - [e4c801f.js](#) -> 1,4 MB
    - [ListMenu.js](#) -> 6,0 KB => non nécessaire pour le rendu à ce stade
    - [RangeMenu.js](#) -> 7,1 KB => non nécessaire pour le rendu à ce stade
    - [BooleanFilter.js](#) -> 3,0 KB => non nécessaire pour le rendu à ce stade
    - [DateFilter.js](#) -> 9,9 KB => non nécessaire pour le rendu à ce stade
    - [DateTimeFilter.js](#) -> 14,0 KB => non nécessaire pour le rendu à ce stade
    - [ListFilter.js](#) -> 6,2 KB => non nécessaire pour le rendu à ce stade
    - [NumericFilter.js](#) -> 3,5 KB => non nécessaire pour le rendu à ce stade
    - [StringFilter.js](#) -> 4,1 KB => non nécessaire pour le rendu à ce stade
    - [Filter.js](#) -> 6,1 KB => non nécessaire pour le rendu à ce stade
  - Les fichiers javascript liés à GeoExt
    - [Map.js](#) -> 15,1 KB => non nécessaire pour le rendu à ce stade
    - [OverlayLayerContainer.js](#) -> 1,6 KB => non nécessaire pour le rendu à ce stade
    - [BaseLayerContainer.js](#) -> 2,0 KB => non nécessaire pour le rendu à ce stade
    - [LayerTreeModel.js](#) -> 3,7 KB => non nécessaire pour le rendu à ce stade
    - [LayerStore.js](#) -> 13,9 KB => non nécessaire pour le rendu à ce stade
    - [LayerLoader.js](#) -> 10,9 KB => non nécessaire pour le rendu à ce stade
    - [View.js](#) -> 3,1 KB => non nécessaire pour le rendu à ce stade
    - [Panel.js](#) -> 1,4 KB => non nécessaire pour le rendu à ce stade
    - [Column.js](#) -> 1,7 KB => non nécessaire pour le rendu à ce stade
    - [Popup.js](#) -> 13,2 KB => non nécessaire pour le rendu à ce stade
    - [Zoom.js](#) -> 11,0 KB => non nécessaire pour le rendu à ce stade
    - [Legend.js](#) -> 7,3 KB => non nécessaire pour le rendu à ce stade
    - [Tip.js](#) -> 3,2 KB => non nécessaire pour le rendu à ce stade
    - [VectorLegend.js](#) -> 23,6 KB => non nécessaire pour le rendu à ce stade
    - [Action.js](#) -> 8,7 KB => non nécessaire pour le rendu à ce stade
    - [FiltersFeature.js](#) -> 24,7 KB => non nécessaire pour le rendu à ce stade
    - [ProgressBarPager.js](#) -> 3,6 KB => non nécessaire pour le rendu à ce stade
    - [Action.js](#) -> 8,7 KB => non nécessaire pour le rendu à ce stade
    - [LayerContainer.js](#) -> 3,2 KB => non nécessaire pour le rendu à ce stade
    - [LayerModel.js](#) -> 1,7 KB => non nécessaire pour le rendu à ce stade
    - [LayerNode.js](#) -> 3,4 KB => non nécessaire pour le rendu à ce stade
    - [LayerLegend.js](#) -> 8,9 KB => non nécessaire pour le rendu à ce stade
    - [FeatureRenderer.js](#) -> 13,5 KB => non nécessaire pour le rendu à ce stade
  - Les fichiers javascript liés à JQuery
    - [f9b5aa2.js](#) -> 882 KB
  - Les fichiers javascript (geosignal) spécifiques
    - [0857596.js](#) -> 324 KB
  - Les accès aux données
    - [https://www.geodair.fr/reporting?\\_dc=1478007209135&page=1&start=0&limit=5&sort=%5B%7B%22property%22%3A%22date\\_de\\_generation%22%2C%22direction%22%3A%22DESC%22%7D%5D](https://www.geodair.fr/reporting?_dc=1478007209135&page=1&start=0&limit=5&sort=%5B%7B%22property%22%3A%22date_de_generation%22%2C%22direction%22%3A%22DESC%22%7D%5D) -> 21,7 KB => non nécessaire pour le rendu à ce stade
    - [https://www.geodair.fr/getRefData/ComboBox/polluant?\\_dc=1478007209275&page=1&start=0&limit=25](https://www.geodair.fr/getRefData/ComboBox/polluant?_dc=1478007209275&page=1&start=0&limit=25) -> 13,1 KB => non nécessaire pour le rendu à ce stade



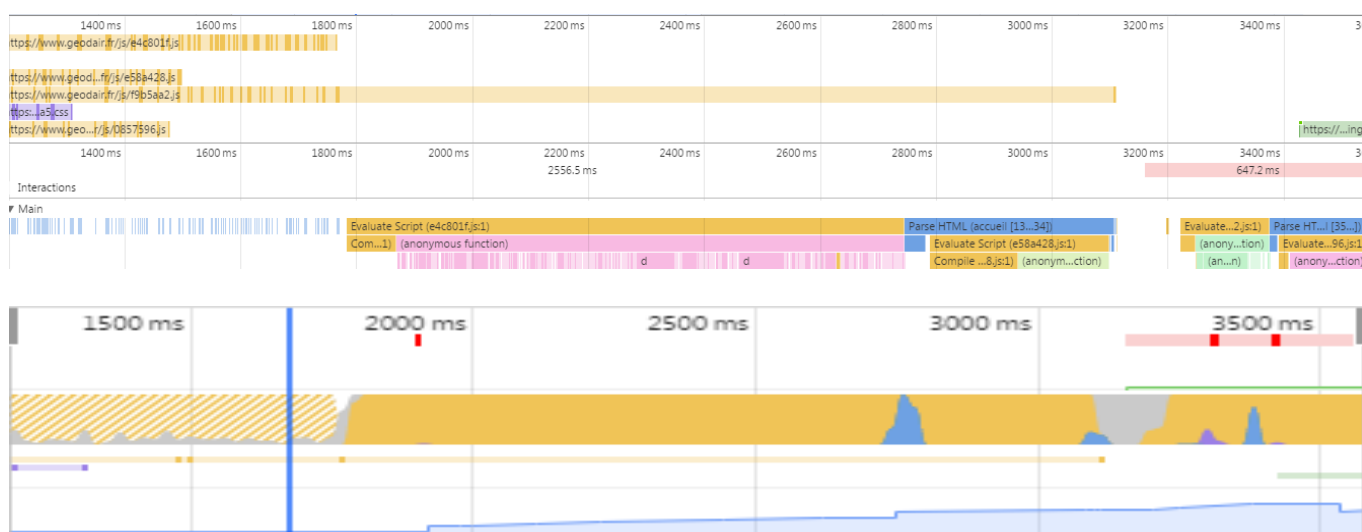
- [https://www.geodair.fr/getRefData/ComboBox/station?\\_dc=1478007209763&page=1&start=0&limit=25&sort=%5B%7B%22property%22%3A%22code\\_station%22%2C%22direction%22%3A%22ASC%22%7D%5D](https://www.geodair.fr/getRefData/ComboBox/station?_dc=1478007209763&page=1&start=0&limit=25&sort=%5B%7B%22property%22%3A%22code_station%22%2C%22direction%22%3A%22ASC%22%7D%5D) -> 183 KB => non nécessaire pour le rendu à ce stade
  - [https://www.geodair.fr/getRefData/ComboBox/polluant?\\_dc=1478007209765&page=1&start=0&limit=25](https://www.geodair.fr/getRefData/ComboBox/polluant?_dc=1478007209765&page=1&start=0&limit=25) -> 13,1 KB => non nécessaire pour le rendu à ce stade
- On peut identifier une latence de 1,5s pendant laquelle les téléchargements sont bloqués



Les graphes ci-dessous mettent en évidence que cette période de latence est principalement due à une phase d'activité intense sur le javascript. Par ordre d'importance, les scripts incriminés sont Extjs et Openlayers.

Cette latence pourrait donc être évitée si l'on prend en compte ce qui a été dit précédemment :

- en utilisant un script Extjs adapté au projet
- en décalant l'interprétation Openlayers à l'accès à la cartographie ou en fin de traitement des éléments nécessaires à la page en cours.



On note en complément que les urls d'appel générées par ExtJS comportent le paramètre `_dc`. Ce paramètre (disable caching) a pour objectif de modifier une url à chaque appel d'un service en ajoutant un timestamp afin d'éviter les mises en cache abusives du navigateur.

Ce comportement qui peut se paramétrer au niveau de la déclaration de chaque store dans ExtJS est activé de façon systématique. A cela s'ajoute le fait que ces services fixent la gestion du cache à no-cache dans leurs en-têtes. Ce point pourrait être amélioré en gérant plus finement la mise en cache des services, les services liés au référentiel étant peu soumis aux évolutions.

Dans les liens indiqués ci-dessus, on peut identifier 2 appels du service polluants. Ces 2 appels sont traités comme deux appels distincts au serveur du fait du paramètre `_dc` et du paramètre no-cache alors que le contenu récupéré est strictement le même. Nous verrons dans les pages suivantes que cette redondance peut être encore plus importante.

#### 6.6.2.3.3.4 Page d’accueil / référentiel

Clic sur référentiel -> appel de quelques images

Les actions sur les en-têtes de colonne (filtre des colonnes ou classement) déclenchent le rechargement de la page  
Les appels des services mettent en évidence des latences assez élevées de 0,5 à 2s  
Les tableaux présentés sont des tableaux HTML générés côté serveur. Il en va de même pour les fiches de détail de chaque objet. Par contre le tableau de version présenté pour chaque objet est généré côté client en Extjs sur la base d’un Json envoyé par le serveur.

##### a) Choix territoires / ZEAT

[https://www.geodair.fr/admin/ineris/referentiels/zeat/list?&\\_dc=1478008940837](https://www.geodair.fr/admin/ineris/referentiels/zeat/list?&_dc=1478008940837)

→ Tableau -> 26,8 KB

Portion de HTML contenant javascript, css et tableau dans une div

On exploite ici le fonctionnement natif de symfony.

[https://www.geodair.fr/filterlist/zeat?\\_dc=1478008941205&page=1&start=0&limit=25&sort=%5B%7B%22property%22%3A%22nom%22%2C%22direction%22%3A%22ASC%22%7D%5D](https://www.geodair.fr/filterlist/zeat?_dc=1478008941205&page=1&start=0&limit=25&sort=%5B%7B%22property%22%3A%22nom%22%2C%22direction%22%3A%22ASC%22%7D%5D)

→ liste des champs pour les combobox de filtrage -> 514 B

Format Json

<https://www.geodair.fr/sig/geoserver/wms?LAYERS=PlanIGN&CRS=EPSG%3A3857&VERSION=1.3.0&FORMAT=image%2Fpng&EXCEPTIONS=INIMAGE&SERVICE=WMS&REQUEST=GetMap&STYLES=&BBOX=-36711.624375001,5563511.456,763320.24425,6363543.324625&WIDTH=256&HEIGHT=256> -> 95,3 KB

<https://www.geodair.fr/sig/geoserver/wms?LAYERS=PlanIGN&CRS=EPSG%3A3857&VERSION=1.3.0&FORMAT=image%2Fpng&EXCEPTIONS=INIMAGE&SERVICE=WMS&REQUEST=GetMap&STYLES=&BBOX=-36711.624375001,4763479.587375,763320.24425,5563511.456&WIDTH=256&HEIGHT=256> -> 48,8 KB

<https://www.geodair.fr/sig/geoserver/wms?LAYERS=PlanIGN&CRS=EPSG%3A3857&VERSION=1.3.0&FORMAT=image%2Fpng&EXCEPTIONS=INIMAGE&SERVICE=WMS&REQUEST=GetMap&STYLES=&BBOX=763320.24425,5563511.456,1563352.112875,6363543.324625&WIDTH=256&HEIGHT=256> -> 74,2 KB

<https://www.geodair.fr/sig/geoserver/wms?LAYERS=PlanIGN&CRS=EPSG%3A3857&VERSION=1.3.0&FORMAT=image%2Fpng&EXCEPTIONS=INIMAGE&SERVICE=WMS&REQUEST=GetMap&STYLES=&BBOX=763320.24425,4763479.587375,1563352.112875,5563511.456&WIDTH=256&HEIGHT=256> -> 42,1 KB

→ tuiles PlanIGN

Format image

Inutilisé à ce niveau car carte non visible

Cf remarques transverses sur la cartographie.

##### b) Choix territoires / communes

Suite à la sélection c’est la carte qui s’affiche par défaut.

Quel est l’intérêt dans la mesure où les communes ne sont pas visibles à l’échelle d’affichage par défaut ?

Quel est l’intérêt d’une couche communal en wfs dans la mesure où cette couche n’est pas interrogeable ?

<https://www.geodair.fr/repositorytomap?entity=commune> -> 814 KB

→ Identifiant geoserver de l’ensemble des communes

Format Json

Permet a priori de faire le lien entre le javascript et les objets sur la carte. Mais le cas d’usage n’est pas identifié.

**Cet appel serait inutile si les services vectoriels étaient gérés en GeoJson.**

<https://www.geodair.fr/admin/ineris/referentiels/commune/list?& dc=1478086852461> -> 114 KB

→ Tableau des communes

Format HTML avec js et css intégrés

Non utilisé à ce stade puisque c'est la carte qui est visible

<https://www.geodair.fr/filterlist/commune? dc=1478086852767&page=1&start=0&limit=25&sort=%5B%7B%22property%22%3A%22nom%22%2C%22direction%22%3A%22ASC%22%7D%5D> -> 1,3 KB

→ Liste des champs

Format Json

Utilisé par les combobox de filtrage

Non utilisé à ce stade puisque c'est la carte qui est visible

<https://www.geodair.fr/sig/geoserver/wms?LAYERS=PlanIGN&CRS=EPSG%3A3857&VERSION=1.3.0&FORMAT=image%2Fpng&EXCEPTIONS=INIMAGE&SERVICE=WMS&REQUEST=GetMap&STYLES=&BBOX=-836743.493,5563511.456,-36711.624375001,6363543.324625&WIDTH=256&HEIGHT=256>

<https://www.geodair.fr/sig/geoserver/wms?LAYERS=PlanIGN&CRS=EPSG%3A3857&VERSION=1.3.0&FORMAT=image%2Fpng&EXCEPTIONS=INIMAGE&SERVICE=WMS&REQUEST=GetMap&STYLES=&BBOX=-36711.624375001,6363543.324625,763320.24425,7163575.19325&WIDTH=256&HEIGHT=256>

<https://www.geodair.fr/sig/geoserver/wms?LAYERS=PlanIGN&CRS=EPSG%3A3857&VERSION=1.3.0&FORMAT=image%2Fpng&EXCEPTIONS=INIMAGE&SERVICE=WMS&REQUEST=GetMap&STYLES=&BBOX=-836743.493,4763479.587375,-36711.624375001,5563511.456&WIDTH=256&HEIGHT=256>

<https://www.geodair.fr/sig/geoserver/wms?LAYERS=PlanIGN&CRS=EPSG%3A3857&VERSION=1.3.0&FORMAT=image%2Fpng&EXCEPTIONS=INIMAGE&SERVICE=WMS&REQUEST=GetMap&STYLES=&BBOX=-836743.493,6363543.324625,-36711.624375001,7163575.19325&WIDTH=256&HEIGHT=256>

<https://www.geodair.fr/sig/geoserver/wms?LAYERS=PlanIGN&CRS=EPSG%3A3857&VERSION=1.3.0&FORMAT=image%2Fpng&EXCEPTIONS=INIMAGE&SERVICE=WMS&REQUEST=GetMap&STYLES=&BBOX=-1636775.361625,5563511.456,-836743.493,6363543.324625&WIDTH=256&HEIGHT=256>

<https://www.geodair.fr/sig/geoserver/wms?LAYERS=PlanIGN&CRS=EPSG%3A3857&VERSION=1.3.0&FORMAT=image%2Fpng&EXCEPTIONS=INIMAGE&SERVICE=WMS&REQUEST=GetMap&STYLES=&BBOX=763320.24425,6363543.324625,1563352.112875,7163575.19325&WIDTH=256&HEIGHT=256>

<https://www.geodair.fr/sig/geoserver/wms?LAYERS=PlanIGN&CRS=EPSG%3A3857&VERSION=1.3.0&FORMAT=image%2Fpng&EXCEPTIONS=INIMAGE&SERVICE=WMS&REQUEST=GetMap&STYLES=&BBOX=1563352.112875,5563511.456,2363383.9815,6363543.324625&WIDTH=256&HEIGHT=256>

<https://www.geodair.fr/sig/geoserver/wms?LAYERS=PlanIGN&CRS=EPSG%3A3857&VERSION=1.3.0&FORMAT=image%2Fpng&EXCEPTIONS=INIMAGE&SERVICE=WMS&REQUEST=GetMap&STYLES=&BBOX=-1636775.361625,4763479.587375,-836743.493,5563511.456&WIDTH=256&HEIGHT=256>

<https://www.geodair.fr/sig/geoserver/wms?LAYERS=PlanIGN&CRS=EPSG%3A3857&VERSION=1.3.0&FORMAT=image%2Fpng&EXCEPTIONS=INIMAGE&SERVICE=WMS&REQUEST=GetMap&STYLES=&BBOX=-1636775.361625,6363543.324625,-836743.493,7163575.19325&WIDTH=256&HEIGHT=256>

<https://www.geodair.fr/sig/geoserver/wms?LAYERS=PlanIGN&CRS=EPSG%3A3857&VERSION=1.3.0&FORMAT=image%2Fpng&EXCEPTIONS=INIMAGE&SERVICE=WMS&REQUEST=GetMap&STYLES=&BBOX=1563352.112875,4763479.587375,2363383.9815,5563511.456&WIDTH=256&HEIGHT=256>

<https://www.geodair.fr/sig/geoserver/wms?LAYERS=PlanIGN&CRS=EPSG%3A3857&VERSION=1.3.0&FORMAT=image%2Fpng&EXCEPTIONS=INIMAGE&SERVICE=WMS&REQUEST=GetMap&STYLES=&BBOX=1563352.112875,6363543.324625,2363383.9815,7163575.19325&WIDTH=256&HEIGHT=256>

→ Fon de plan IGN

Un clic sur tableau affiche le tableau sans chargement puisqu'il a été pré-généré.

c) Choix territoires / département

Comportement comparable à ZEAT.

d) Choix territoires / EPCI

Comportement comparable à ZEAT.

e) Choix territoires / IRIS

Comportement comparable à ZEAT.

f) Surveillance / ZAS

Comportement comparable à communes.

Zoom sur la France métropolitaine.

On note des accès anormaux aux services :

- Le service WFS des stations est accédé alors que la couche n'est pas visible par défaut.
- Le service WMS PlanIGN est appelé 2 fois consécutive. Une première fois à une échelle qui couvre la France avec les DOM/TOM, puis une seconde fois à une échelle correspondant à la France métropolitaine.

g) Surveillance / Stations

Comportement comparable à communes.

Zoom sur la France + DOM/TOM.

Couche communes cochable (anormal) ce qui déclenche la récupération et l'affichage de toutes les communes françaises par WFS, soit 10,8MB de données.

h) Clic sur onglet carte

<https://www.geodair.fr/sig/geoserver/wms?LAYERS=PlanIGN&CRS=EPSG%3A3857&VERSION=1.3.0&FORMAT=image%2Fpng&EXCEPTIONS=INIMAGE&SERVICE=WMS&REQUEST=GetMap&STYLES=&BBOX=-836743.493,5563511.456,-36711.624375001,6363543.324625&WIDTH=256&HEIGHT=256> -> 35,9 KB

<https://www.geodair.fr/sig/geoserver/wms?LAYERS=PlanIGN&CRS=EPSG%3A3857&VERSION=1.3.0&FORMAT=image%2Fpng&EXCEPTIONS=INIMAGE&SERVICE=WMS&REQUEST=GetMap&STYLES=&BBOX=-36711.624375001,6363543.324625,763320.24425,7163575.19325&WIDTH=256&HEIGHT=256> -> 71,3 KB

<https://www.geodair.fr/sig/geoserver/wms?LAYERS=PlanIGN&CRS=EPSG%3A3857&VERSION=1.3.0&FORMAT=image%2Fpng&EXCEPTIONS=INIMAGE&SERVICE=WMS&REQUEST=GetMap&STYLES=&BBOX=-836743.493,4763479.587375,-36711.624375001,5563511.456&WIDTH=256&HEIGHT=256> -> 28,6 KB

<https://www.geodair.fr/sig/geoserver/wms?LAYERS=PlanIGN&CRS=EPSG%3A3857&VERSION=1.3.0&FORMAT=image%2Fpng&EXCEPTIONS=INIMAGE&SERVICE=WMS&REQUEST=GetMap&STYLES=&BBOX=-836743.493,6363543.324625,-36711.624375001,7163575.19325&WIDTH=256&HEIGHT=256> -> 47,0 KB

<https://www.geodair.fr/sig/geoserver/wms?LAYERS=PlanIGN&CRS=EPSG%3A3857&VERSION=1.3.0&FORMAT=image%2Fpng&EXCEPTIONS=INIMAGE&SERVICE=WMS&REQUEST=GetMap&STYLES=&BBOX=763320.24425,6363543.324625,1563352.112875,7163575.19325&WIDTH=256&HEIGHT=256> -> 47,2 KB

<https://www.geodair.fr/sig/geoserver/wms?LAYERS=PlanIGN&CRS=EPSG%3A3857&VERSION=1.3.0&FORMAT=image%2Fpng&EXCEPTIONS=INIMAGE&SERVICE=WMS&REQUEST=GetMap&STYLES=&BBOX=-1636775.361625,5563511.456,-836743.493,6363543.324625&WIDTH=256&HEIGHT=256> -> 781 B

<https://www.geodair.fr/sig/geoserver/wms?LAYERS=PlanIGN&CRS=EPSG%3A3857&VERSION=1.3.0&FORMAT=image%2Fpng&EXCEPTIONS=INIMAGE&SERVICE=WMS&REQUEST=GetMap&STYLES=&BBOX=-1636775.361625,4763479.587375,-836743.493,5563511.456&WIDTH=256&HEIGHT=256> -> 16,7 KB

<https://www.geodair.fr/sig/geoserver/wms?LAYERS=PlanIGN&CRS=EPSG%3A3857&VERSION=1.3.0&FORMAT=image%2Fpng&EXCEPTIONS=INIMAGE&SERVICE=WMS&REQUEST=GetMap&STYLES=&BBOX=1563352.112875,5563511.456,2363383.9815,6363543.324625&WIDTH=256&HEIGHT=256> -> 75,7 KB

<https://www.geodair.fr/sig/geoserver/wms?LAYERS=PlanIGN&CRS=EPSG%3A3857&VERSION=1.3.0&FORMAT=image%2Fpng&EXCEPTIONS=INIMAGE&SERVICE=WMS&REQUEST=GetMap&STYLES=&BBOX=-1636775.361625,6363543.324625,-836743.493,7163575.19325&WIDTH=256&HEIGHT=256> -> 18,6 KB

<https://www.geodair.fr/sig/geoserver/wms?LAYERS=PlanIGN&CRS=EPSG%3A3857&VERSION=1.3.0&FORMAT=image%2Fpng&EXCEPTIONS=INIMAGE&SERVICE=WMS&REQUEST=GetMap&STYLES=&BBOX=1563352.112875,4763479.587375,2363383.9815,5563511.456&WIDTH=256&HEIGHT=256> -> 64,0 KB

<https://www.geodair.fr/sig/geoserver/wms?LAYERS=PlanIGN&CRS=EPSG%3A3857&VERSION=1.3.0&FORMAT=image%2Fpng&EXCEPTIONS=INIMAGE&SERVICE=WMS&REQUEST=GetMap&STYLES=&BBOX=1563352.112875,6363543.324625,2363383.9815,7163575.19325&WIDTH=256&HEIGHT=256> -> 44,2 KB

#### 6.6.2.3.3.5 Page d'accueil / accès aux données

Clic sur accès aux données -> appel de quelques images

Les combobox ne sont pas pré-remplies comme sur la partie référentielle mais remplies sur sollicitation.  
Ceci permet de ne télécharger les flux que lorsqu'ils sont utiles.

- 1) Choix polluant -> appel service

<https://www.geodair.fr/getRefData/ComboBox/polluant? dc=1478100466497&query=&page=1&start=0&limit=25> qui renvoi la liste des polluants au format json (`{"codefr": "01", "notation": "01 - SO2", "est_reglementaire": "t", "reglementaire": "Polluants r\u00e9glement\u00e9s"}`)

- 2) Choix type de données -> pas d'appel de services

- 3) Choix date et heure -> pas d'appel de services

- 4) Choix organisme -> appel de 2 services :

<https://www.geodair.fr/getRefData/ComboBox/organisme? dc=1478100490698&query=&page=1&start=0&limit=25&sort=%5B%7B%22property%22%3A%22nom%22%2C%22direction%22%3A%22ASC%22%7D%5D> qui renvoi les informations sur les organismes au format json (`{"code": "14", "nom": "ATMO NORD-PAS-DE-CALAIS", "code_nature_organisme": "AASQA", "code_region": "31"}`) et

[https://www.geodair.fr/getRefData/ComboBox/station? dc=1478100493165&page=1&start=0&limit=25&sort=%5B%7B%22property%22%3A%22nom\\_station%22%2C%22direction%22%3A%22ASC%22%7D%5D](https://www.geodair.fr/getRefData/ComboBox/station? dc=1478100493165&page=1&start=0&limit=25&sort=%5B%7B%22property%22%3A%22nom_station%22%2C%22direction%22%3A%22ASC%22%7D%5D) qui renvoi les informations sur les stations au format json (`{"code_station": "FR25040", "nom_station": "FR25040 - Mesnil Esnard", "code_reseau": "25", "code_zas": "FR25A01", "code_organisme": "08", "code_region": "23"}`)

La combo box des stations n'est cependant pas remplie

- 5) Choix réseau -> appel de 2 services :

<https://www.geodair.fr/getRefData/ComboBox/reseau? dc=1478100511044&query=&page=1&start=0&limit=25&sort=%5B%7B%22property%22%3A%22nom%22%2C%22direction%22%3A%22ASC%22%7D%5D> qui renvoi les informations sur les réseaux au format json (`{"code": "01", "nom": "AERFOM", "code_organisme": "04"}`) et

[https://www.geodair.fr/getRefData/ComboBox/station? dc=1478100513034&page=1&start=0&limit=25&sort=%5B%7B%22property%22%3A%22nom\\_station%22%2C%22direction%22%3A%22ASC%22%7D%5D](https://www.geodair.fr/getRefData/ComboBox/station? dc=1478100513034&page=1&start=0&limit=25&sort=%5B%7B%22property%22%3A%22nom_station%22%2C%22direction%22%3A%22ASC%22%7D%5D) qui renvoi les informations sur les stations au format json (`{"code_station": "FR25040", "nom_station": "FR25040 - Mesnil Esnard", "code_reseau": "25", "code_zas": "FR25A01", "code_organisme": "08", "code_region": "23"}`)

La combo box des stations n'est cependant pas remplie

Tous les réseaux de France sont envoyés et le filtre sur le code organisme se fait en local pour correspondre au choix de la combo box précédente

- 6) Choix de la station -> appel du service

[https://www.geodair.fr/getRefData/ComboBox/station? dc=1478100525364&query=&page=1&start=0&limit=25&sort=%5B%7B%22property%22%3A%22code\\_station%22%2C%22direction%22%3A%22ASC%22%7D%5D&filter=%5B%7B%22property%22%3A%22code\\_reseau%22%2C%22value%22%3Aundefined%7D%5D](https://www.geodair.fr/getRefData/ComboBox/station? dc=1478100525364&query=&page=1&start=0&limit=25&sort=%5B%7B%22property%22%3A%22code_station%22%2C%22direction%22%3A%22ASC%22%7D%5D&filter=%5B%7B%22property%22%3A%22code_reseau%22%2C%22value%22%3Aundefined%7D%5D) qui renvoi les informations de station au format json (`{"code_station": "FR25040", "nom_station": "FR25040 - Mesnil Esnard", "code_reseau": "25", "code_zas": "FR25A01", "code_organisme": "08", "code_region": "23"}`)

Toutes les stations de France sont chargées et le filtre si le code réseau se fait en local.

A ce stade toute modification des critères de recherche se fait en local sans rappel des services organisme et réseau. Par contre toute modification de choix dans une combo box déclenche systématiquement le rappel du service station.

**Pour une sélection simple, il y a donc à minima 3 appels identiques du service stations qui renvoi exactement le même contenu et toute action sur les combo box déclenche un appel supplémentaire.**

- 7) Clic sur afficher les données -> appel du service <https://www.geodair.fr/getRefData/Donnees> en mode POST avec l’ensemble des critères de recherche. Le service renvoi les réponses au format json pour les données (`{"date_deb":"2016\06\12 01:00:00","date_fin":"2016\06\12 01:15:00","nom_organisme":"AIRPARIF","code_zas":"FR04A01","nom_zas":"ILE-DE-FRANCE-PARIS","code_site":"FR04004","nom_site":"PARIS 18eme","nom_polluant":"NO","discriminant":[],"type_valeur":"Donn\u00e9es valid\u00e9es","valeur":"0.0","valeur_brute":"","unite_mesure":"\u00b5g-m3","taux_saisi":"","couverture_temp":"","code_qualite":"A","flag_validite":"1"}`), les statistiques (`{"date_deb":"2016\06\12 01:00:00","date_fin":"13\06\2016 00:00:00","nom_organisme":"AIRPARIF","code_zas":"FR04A01","nom_zas":"ILE-DE-FRANCE-PARIS","code_site":"FR04004","nom_site":"PARIS 18eme","nom_polluant":"NO","discriminant":"","type_valeur":"Moy journali\u00e8re","valeur":"1.1","valeur_brute":"1.1354166666667","unite_mesure":"\u00b5g-m3","taux_saisi":"100","couverture_temp":"100","code_qualite":"R","flag_validite":"1","type_rattachement":"Urbaine : construction continue"}`) et les informations pour faire le lien avec la cartographie (`{"id4geoserv":"115"},"id4geoserv_zas":{"id4geoserv":"10"},"id4geoserv_commune":[],"msg":""}`)

Les données sont intégrées dans 2 tableaux qui sont des grid Extjs.

Ce tableau peut être modifié (tri, choix des colonnes) sans nouvel appel du service. De même la pagination est un artefact de visualisation local.

Il y a donc une différence fondamentale d’implémentation entre la partie référentielle et données :

- La première exploite la ‘mécanique’ standard de symfony qui produit les pages au niveau serveur et envoi du html préformé vers le client et le tout est intégré dans un composant Extjs de visualisation HTML.
- La seconde repose sur des services qui produisent un json brut qui est ensuite traité dans des composants Extjs en local.

Ceci explique la précaution prise de brider le requêtage pour limiter le volume du flux réponse.

Quatre remarques à ce stade :

- Ce choix est surprenant car au niveau référentiel les volumes sont connus et il n’y a pas de risque de faire des requêtes qui produiraient de gros volumes de réponse. Hors c’est côté référentiel que l’on utilise une pagination serveur alors que côté données la pagination est cliente ce qui implique un bridage des possibilités de requêtage.
- Le bridage des requêtes se fait de façon arbitraire sans lien direct avec la réalité du contenu de la base.
- La protection mise en place n’est qu’au niveau de l’application cliente. Des envois de requête via un client REST générique permet de requêter beaucoup plus largement. Nous avons pu ainsi récupérer les données d’une station pour un polluant sur un mois, deux mois de profondeur (avec les statistiques horaires) en une requête au serveur. Une tentative sur un an aboutit à un time out du serveur web nginx. Le timeout est donc la seule protection qui empêche de lancer des requêtes trop volumineuses ce qui n’empêche pas la base d’être fortement sollicitée jusqu’au timeout et normalement la fermeture de la requête lancée. L’envoi d’un grand nombre de requêtes volumineuses est un risque de surcharger le serveur de base avant que le timeout n’arrive.
- La sélection des critères de recherche se fait sans tenir compte de l’existence ou non de données dans la base pour la combinaison de critères choisie. Cette situation est une conséquence directe de la dualité des



bases de données qui cloisonnent hermétiquement données et référentiel. Les Combo box reposent sur le référentiel alors que le tableau repose sur la base MongoDB des données.

Il serait pertinent de filtrer les critères de recherche progressivement en prenant également en compte l'existence ou non de données en base.

- Les tableaux ne contiennent pas de liens vers des fiches détaillées comme pour le référentiel alors qu'une partie des informations du tableau correspondent au référentiel.

#### Comportement si la carte n'a jamais été visualisée :

En complément les services cartographiques sont également appelés :

- Service wfs des stations sur une zonelimitée
- Service WFS des ZAS sur une zone limitée
- Services WMS de fond de plan lié au zoom sur la France

**Doublement inutile puisque la carte est invisible et que les couches seront toutes rappelées lors de l'affichage de la carte et que le site connaît les identifiants des éléments pour permettre un filtrage en local.**

En complément, le filtrage utilisé ne se base pas sur un filtrage sur la valeur d'un champ en CQL comme le permet le WFS, mais sur une bbox réduite correspondant à l'objet cherché

Il serait opportun que le zoom se fasse sur la ZAS et non sur la France entière.

#### Comportement si la carte a déjà été visualisée :

Appel uniquement des services WMS correspondant au zoom sur la ZAS sélectionnée dans la recherche.

A l'affichage de la carte, aucun nouvel appel et la carte est bien centrée sur la ZAS.

**Cette différence de comportement selon les cas de figure tant au niveau de l'affichage que des flux réseau est assez surprenante.**

## 6.7 SYNTHÈSE DES CONSTATS

---

Le système souffre globalement d'un manque de conception ou de la dérive d'une conception initiale non adaptée que l'on aurait tenté de faire évoluer sans vision globale.

Les choix de solutions sont critiquables et pour certains lourds de conséquence comme la dichotomie entre PostgreSQL et MongoDB

Le code spécifique réalisé est de qualité médiocre et présente une dette technique très forte qui alourdira les phases de reprise de la TMA.

Le périmètre est sous documenté si l'on cumule les zones non documentées avec l'obsolescence d'autres documents qui nécessitera des travaux de mise à jour et de complément en cas de reprise de la TMA.

Toute la stratégie de supervision et de monitoring est soit absente soit non documentée ce qui posera un problème en cas de reprise de l'exploitation par un tiers.

Dans le chapitre suivant, nous listons tous les constats par catégorie : PF (points forts), PA (Points d'amélioration) et NC (Non-conformité) en faisant référence à la page et au paragraphe où ils sont décrits.

Au niveau technique, on peut synthétiser les points suivants :

- L'utilisation de Fuse ne se justifie pas dans l'usage (exposition de services) qui en est fait actuellement.
- Le requêtage et notamment l'agrégation ne sont pas optimisés
- Absence de garantie du synchronisme entre les bases historiques et opérationnelles
- Absence de procédure de traitement des inadéquations entre données transmises et référentiel. Ce point a un impact majeur puisqu'il implique un rechargement complet en masse ce qui provoque de délais très importants de chargement et de recalcul
- Structuration des web services discutable par rapport à une approche d'un web service par objet métier

- L’absence d’urbanisation de services malgré la présence de briques techniques (ESB) normalement dédiées à cet usage
- Démultiplication des couches DAO ce qui alourdit la maintenance et représente un risque fort d’anomalies
- Des performances moyennes sur l’intégration des données liées en partie à l’approche multibase, à l’intégration du calcul de stats dans la séquence d’intégration et non en traitement parallèle, à l’utilisation assez faible du parallélisme...
- Des performances médiocres sur les scripts (notamment calcul de statistiques) qui impliquent une forte interaction entre la base postgresql et les bases MongoDB
- Une absence totale d’optimisation du site de consultation
- Une conception hybride des écrans de façon à faire cohabiter une approche ExtJs / services avec la création de pages classiques par Sonata
- Absence d’optimisation des services cartographiques à mettre en relation avec le choix d’une approche vectorielle pour les données référentielles.
- Un choix discutable de limiter arbitrairement les possibilités de requêtage sur les données de mesures pour éviter une volumétrie de données trop importante (lié au choix d’une pagination cliente sur cette partie).



## 7 RECAPITULATIF DES CONSTATS

### 7.1 POINTS FORTS

No du constat	Résumé du constat	Page	Paragraphe
PF01	Une variété de documents couvrant les différentes phases du projet	8	4
PF02	La solution « Bull Geo Live » est complète et offre les fonctionnalités nécessaires	12	5.2

### 7.2 POINTS D'AMELIORATION

No du constat	Résumé du constat	Page	Paragraphe
PA01	Les différents documents doivent être mis à jour par rapport à l'existant pour assurer une bonne réversibilité	8	4
PA02	Compléter les documents techniques existants pour couvrir encore plus les sujets suivants : <ul style="list-style-type: none"><li>• Présentation des sources</li><li>• Présentation de l'usine logicielle</li><li>• Stratégie de tests</li><li>• Installation des différents composants</li></ul>	8	4
PA03	Ajouter à l'architecture physique un mécanisme de load balancing pour assurer une scalabilité horizontale automatique permettant ainsi de mieux gérer une montée en charge.	12	5.1
PA04	Utiliser pleinement les fonctionnalités de l'ESB Fuse afin de : <ul style="list-style-type: none"><li>• Urbaniser les services métiers</li><li>• Rajouter des processus de validation métier via les workflows</li></ul>	12	5.2

	<ul style="list-style-type: none"> <li>Servir de médiation aux systèmes externes</li> </ul>		
PA05	Optimiser la configuration du site pour l'environnement de production	53	6.6.2.3.3
PA06	Optimiser les services cartographiques (Ne pas utiliser une approche vectorielle pour les données référentielles)		
PA07	La mise en place d'une intégration continue avec mesure de la qualité de code apportera un gain de maintenabilité et performance du code. De plus, l'application des principes et patterns de bases de programmation comme les patterns SOLID permettra d'obtenir une meilleure qualité de code.	21	6.1
PA08	La table « rapport_ue » doit être normalisée en rajouter les colonnes « NOM_RAPPORT » et « ANNEE »	24	6.2.1
PA09	Créer des tables d'association entre entités versionnées afin de ne pas avoir à dupliquer les entités filles. Utiliser de plus des attributs monovalués comme clé primaire pour faciliter la jointure (ID auto incrémenté).	24	6.2.1
PA10	Définir une documentation dédiée à la stratégie de logs/traces des jobs Talend	27	6.3.1
PA11	Une perspective d'optimisation des jobs réside <ul style="list-style-type: none"> <li>dans le fait que les jobs accèdent aux bases Postgresql et MongoDB et sont amenés à réaliser des opérations qui pourraient se faire directement en base dans le cas d'une base unique.</li> <li>dans la rationalisation de la production des logs et traces</li> </ul>	27	6.3.1
PA12	Si la complexité des calculs justifie l'usage d'un code externe à l'ETL (calcul des statistiques), ce dernier aurait pu être limité aux algorithmes de calcul en laissant les accès aux données à la partie ETL, réduisant ainsi l'effort de maintien en cohérence des couches DAO.	27	6.3.1
PA13	Optimiser certaines requêtes de la couche DAO du service « CUD Référentiels »	36	6.4.1
PA14	La version 2.0 de l'API JDOM sortie en 2012 est plus performante et doit être utilisée pour un gain de performance.	39	6.4.2
PA15	Pour remédier aux requêtes imbriquées, il serait plus efficace d'utiliser le mode batch du driver JDBC Java permettant de grouper des requêtes et les soumettre en un seul appel à la base de données.	35	6.3.2
PA16	L'utilisation d'un framework tel que Spring Data ou JTA permet d'avoir un code plus maintenable, « sans réinventer la roue » et sans impact considérable sur les performances.	35	6.3.2
PA17	Passer à une architecture REST pour les services CRUD des référentiels	36	6.4.1
PA18	Bien qu'il soit vrai que l'accès à l'ESB ne se fait que depuis des serveurs frontaux via des IP privés, il est important d'avoir des web services autonomes et ne pas déporter la responsabilité de la sécurité chez l'appelant.	36	6.4.1
PA19	Le module « Rapportage » nécessite une réécriture en tenant compte des erreurs SONAR, en mettant en	39	6.4.2

	place une meilleure conception et en optimisant certaines requêtes.		
<b>PA20</b>	Il serait pertinent de modifier les appels aux services par OpenLayers en conséquence afin d'optimiser les accès sur l'ensemble des couches de base qui évoluent peu dans le temps.	47	6.6.2.3.2
<b>PA21</b>	Le choix du WFS/gml est discutable. Il s'agit d'un service ogc normée, mais l'état de l'art oriente vers l'usage de services geojson pour ce type d'application. Le geojson est beaucoup moins verbeux et son interprétation par le javascript beaucoup plus efficace.	47	6.6.2.3.2
<b>PA22</b>	Le flux WMS comporte un grand nombre d'informations qui ne semblent pas utilisées par l'application. A cela s'ajoute le fait que les coordonnées des points des géométries sont données avec une précision nanométrique (10-9 mètres). Tous ces éléments augmentent inutilement la taille de ces flux.	47	6.6.2.3.2
<b>PA23</b>	Le site ne proposant pas de changer la projection d'affichage, il serait pertinent que les projections de stockage et d'affichage soient les mêmes pour éviter les calculs de reprojection inutiles.	47	6.6.2.3.2
<b>PA24</b>	L'utilisation d'une base de données PostgreSQL aurait été largement suffisante comme support de sauvegarde des données de mesures et des statistiques.	15	5.2.3

### 7.3 NON CONFORMITE

No du constat	Résumé du constat	Page	Paragraphe
<b>NC01</b>	L'absence des composants Bonita et OpenLDAP conformément à la réponse technique d'ATOS au CCTP	13	5.2.1
<b>NC02</b>	L'utilisation de Fuse ne se justifie pas dans l'usage (exposition de services) qui en est fait actuellement.	14	5.2.2
<b>NC03</b>	Absence de garantie du synchronisme entre les bases historiques et opérationnelles	27	6.3.1
<b>NC04</b>	Absence de procédure de traitement des inadéquations entre données transmises et référentiel. Ce point a un impact majeur puisqu'il implique un rechargement complet en masse ce qui provoque de délais très importants de chargement et de recalcul		
<b>NC05</b>	Démultiplication des couches DAO ce qui alourdit la maintenance et représente un risque fort d'anomalies	16	5.2.4
<b>NC06</b>	L'utilité du bus de service perd son sens lorsque la couche de présentation (WEB1) l'outrepasse pour implémenter des règles métiers et accéder aux bases de données.	14	5.2.2

NC07	Forte volumétrie de la base MongoDB d'Audits avec insertion des logs métiers comme techniques et un niveau de logs verbeux (RECETTE)	35	6.3.2
NC08	Créer un seul service avec toutes les opérations du service pour la gestion des référentiels enfreint le principe de séparation des responsabilités (SRP). La moindre modification sur une opération entraine la livraison de tout l'ensemble des opérations.	36	6.4.1
NC09	Le mix des frameworks ExtJS et Sonata pour implémenter les écrans conduit à s'éloigner de l'état de l'art, à rendre le code moins lisible et le site moins performant.	42	6.6.1.1
NC10	Le choix d'implémenter le site Geod'Air en une seule page est discutable. L'expérience utilisateur en est impactée étant donné qu'il n'est plus possible pour lui de revenir à un état précédent des écrans (surtout pour les formulaires de recherche).	42	6.6.1.1
NC11	L'une des bonnes pratiques de programmation Symfony est d'utiliser les classes « repository » et ne pas écrire des requêtes SQL directement dans les contrôleurs afin que ces requêtes puissent être réutilisées par plusieurs contrôleurs.	42	6.6.1.1
NC12	<a href="https://www.geodair.fr/geoserver/web/">https://www.geodair.fr/geoserver/web/</a> -> accès à l'interface d'admin de géoserver	46	6.6.2.2
NC13	Les critères de recherche ne couvrent pas l'ensemble des référentiels disponibles ou des champs disponibles dans le tableau de présentation des mesures (choix métier ?)	Erreur ! Signet non défini.	Erreur ! Source du renvoi introuvable.
NC14	L'interaction avec la carte est pauvre et limitée au filtrage des stations et des zas	Erreur ! Signet non défini.	Erreur ! Source du renvoi introuvable.
NC15	Le cache GeoServer n'est pas exploité	47	6.6.2.3.2



## 8 PRECONISATIONS

Suite à différentes analyses effectuées lors de l’audit et décrites dans les paragraphes précédents, ce chapitre donne les préconisations pour obtenir une solution fiable, performante et facilement maintenable.

Notre préconisation s’articule autour de trois axes :

- Présentation de l’architecture cible la plus pertinente à notre sens
- Présentation d’une approche graduée pour atteindre progressivement la cible.
- Identification des actions prioritaires impliquant le moins d’impacts sur le système actuel permettant rapidement une ouverture large aux utilisateurs.

### 8.1 ARCHITECTURE CIBLE

---

Dans un premier temps, nous souhaitons proposer une architecture cible en mode « page blanche » autrement dit sans tenir compte de l’existant et du coût de migration vers cette cible.

Dans un second temps, nous présentons l’architecture cible évitant de pratiquement tout réécrire. C’est sur cette cible que l’approche graduée sera développée.

Pour une meilleure compréhension, nous avons séparé l’architecture du système en 3 zones :

- Front accessible depuis l’extérieur et contenant donc que les serveurs frontaux
- Middle contenant les serveurs pour les traitements métiers
- Back contenant les serveurs de bases de données

#### 8.1.1 *Architecture cible en mode page blanche*

Au vu des différents constats récapitulés dans le paragraphe précédent, le schéma ci-dessous montre l’architecture cible en mode page blanche que nous préconisons.

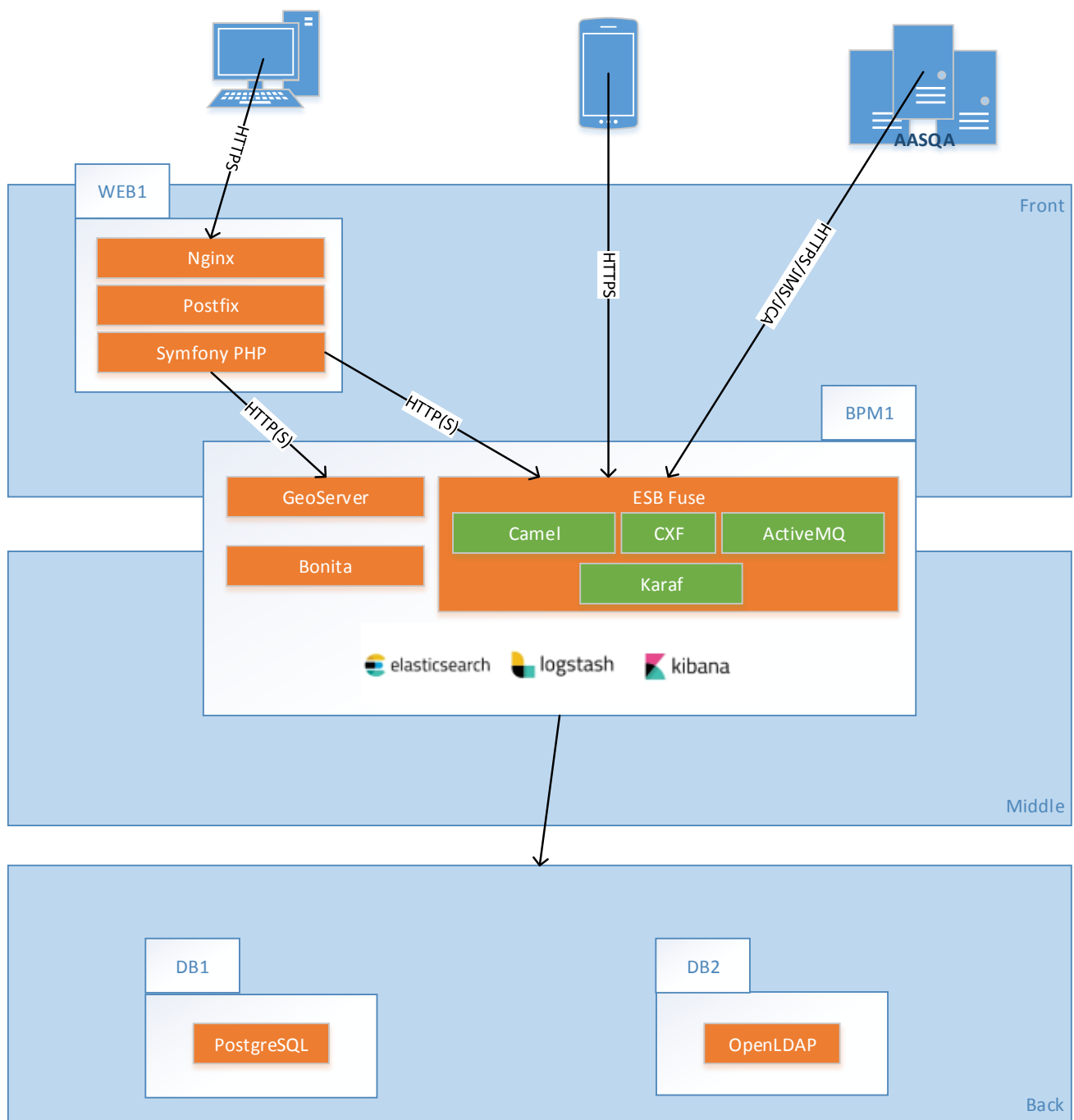


Figure 19: Architecture Cible Page blanche

Dans cette architecture, nous gardons une seule base de données PostgreSQL pour contenir toutes les données nécessaires à l'application (PA24).

De plus, nous faisons le choix de l'ESB pour l'intégration des données de mesure pour les raisons explicitées au paragraphe 5.2.2 . L'ETL Talend est donc abandonné.

Cette architecture cible remet en cause complètement les choix du système actuel sans tenir compte de l'investissement déjà consenti. En effet, sa mise en place entraine des chantiers conséquents tels que :

- Migration de toutes les bases MongoDB vers une base PostgreSQL
- Redéveloppement des services utilisant les bases MongoDB

- Redéveloppement du module d’intégration des données de mesure dans l’ESB

Par ailleurs, nous intégrons la suite « Elasticsearch – LogStash – Kibana » qui est une solution d’analyse de log performante et complète.

- Elasticsearch : Moteur de stockage et d’indexation de documents et moteur de requête/analyse de ceux-ci
- LogStash : Analyse, filtrage et découpage des logs pour les transformer en documents, parfaitement formatés notamment pour Elasticsearch
- Kibana : Dashboard interactif et paramétrable permettant de visualiser les données stockées.

### **8.1.2 Architecture cible préconisée**

Pour éviter de devoir pratiquement tout réécrire, le schéma suivant montre une architecture cible tenant compte de l’investissement déjà consenti en conservant le cœur du système :



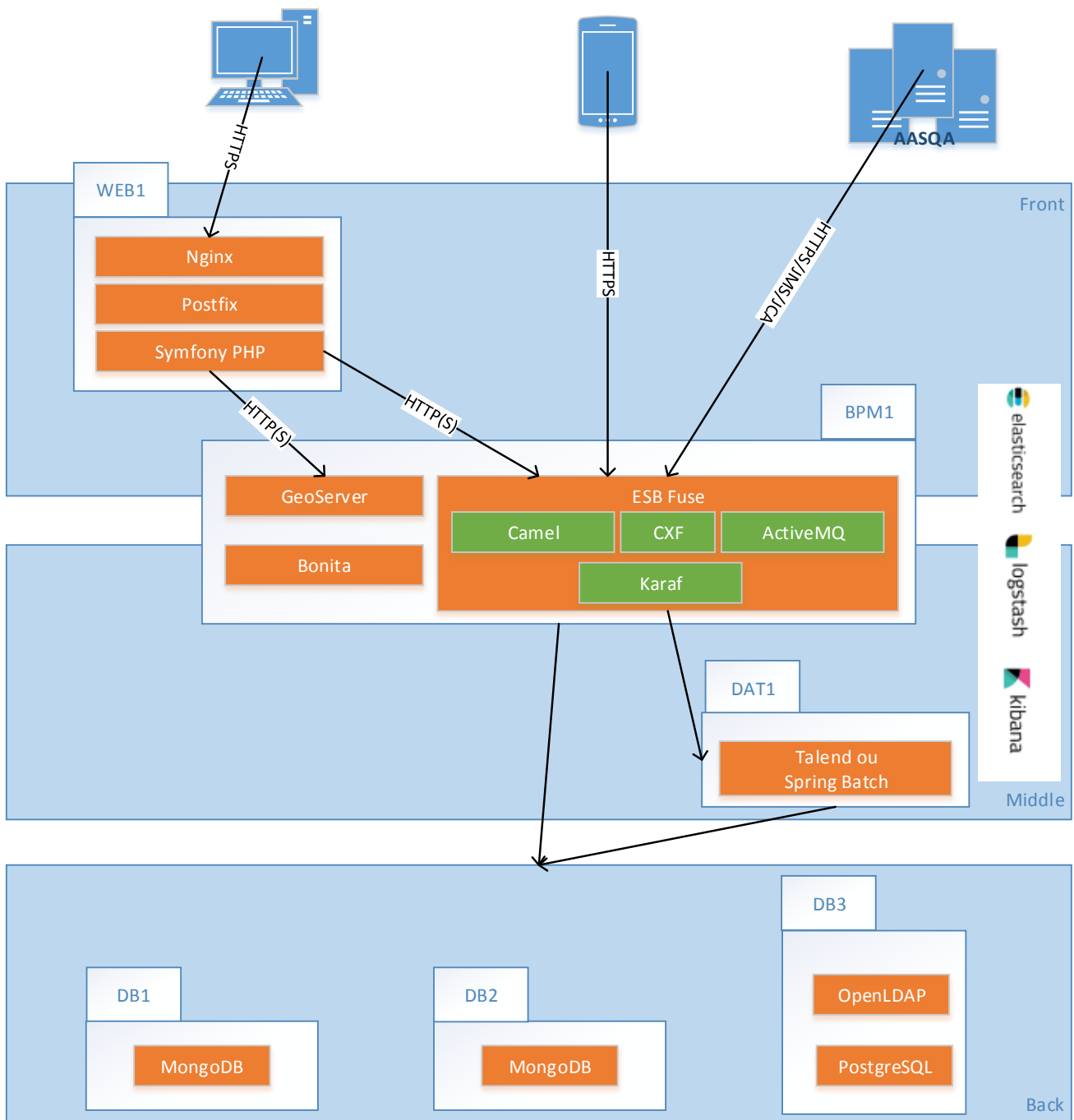


Figure 20: Architecture cible préconisée

### 8.1.3 Zone « Front »

#### 8.1.3.1 Serveur WEB1

Nous recommandons d’abandonner les bundles SONATA pour ne construire que des vues (au sens MVC) uniquement avec le framework ExtJS. Il faudra donc en contrepartie, développer un CRUD générique avec ExtJS pour les données du référentiel.

La couche Symfony devient donc une API ne renvoyant que des données en JSON.

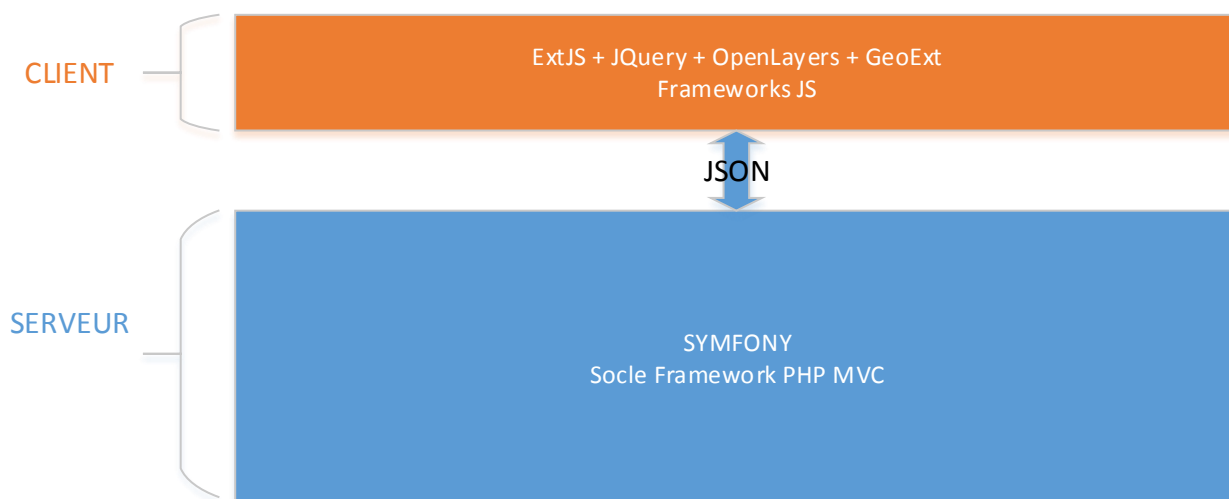


Figure 21: Cible Architecture Web

De plus, nous préconisons de transformer le site web d’une application SPA (Single Page Application) en une application multipages permettant ainsi d’améliorer l’expérience utilisateur (voir 6.6.1.1). Il est vrai que le framework ExtJS est en général utilisé pour des applications SPAs. Cependant avec l’intégration du concept de « workspaces »<sup>3</sup>, il est tout à fait possible de développer des applications multipages avec du ExtJS.

#### 8.1.3.2 Serveur BPM1

Nous recommandons de placer l’ESB au cœur de l’architecture permettant une meilleure urbanisation des services vis-à-vis des applications clientes du système Geod’Air. Le serveur BPM1 est donc dans une zone mixte « Front/Middle » étant donné que l’ESB est accessible depuis l’extérieur.

Le paragraphe suivant donne plus de détails sur la cible envisagée pour ce serveur.

### 8.1.4 Zone « Middle »

#### 8.1.4.1 Serveur BPM1

Le système actuel ne tire pas profit des fonctionnalités d’urbanisation et d’intermédiation qu’apporte l’ESB Fuse de la solution Bull « Geo Live ». Il revient donc de le replacer au cœur de l’architecture.

Bien qu’il soit toujours possible d’intégrer les données des AASQA via du transfert de fichiers en utilisant l’ESB (via des connecteurs FTP), le faire en utilisant le protocole JMS ou via des web services apporte une plus-value significative : l’intégration des données en temps réels.

Le traitement ETL des données est toujours assuré par le serveur DAT1. L’ESB a pour responsabilité de lancer les jobs Talend et de fournir les données à traiter.

<sup>3</sup> <https://docs.sencha.com/cmd/guides/workspaces.html>

Dans l’architecture cible, l’ESB devient le garant des données. Le serveur WEB1 n’a plus accès directement aux bases de données. Il doit appeler des web services (REST ou SOAP) pour récupérer les informations nécessaires au site web.

De plus, la solution BPM Bonita peut être intégrée à l’ESB pour gérer des processus métiers complexes. Il est important de bien juger de son utilité avant son intégration. L’utilisation d’un BPM doit être envisagée si l’on remplit les conditions suivantes :

- Changement fréquente des processus métiers
- Automatisation et monitoring des processus
- Permettre aux utilisateurs métiers et aux clients d’agir sur les workflows
- Collection de métriques sur les processus
- Etc...

Concernant l’authentification, nous recommandons l’utilisation de l’ESB exposant des services d’authentification aux applications clientes en s’appuyant sur l’annuaire OpenLDAP.

En termes d’évolutions futures, l’ESB permet d’envisager les éléments suivants :

- Intégration d’autres applications clients comme des applications mobiles appelant les services exposés par l’ESB
- Synchronisation des données avec les AASQA via le mécanisme de push.

Pour finir, la suite « Elastic Slack » est mise en place pour l’analyse et la visualisation des logs de tout le système.

#### **8.1.4.2 Serveur DAT1**

Dans l’architecture cible, le serveur DAT1 n’est plus accessible depuis l’extérieur d’où sa position dans la zone « Mixte ». Les jobs Talend sont exécutés via l’ESB.

De plus, il n’est plus nécessaire de disposer d’un serveur FTP étant donné que l’intégration des données dans l’architecture cible se fait en temps réels via des web services ou des messages JMS.

#### **8.1.5 Zone « Back »**

Concernant les bases de données, nous recommandons les actions suivantes :

- Ne plus distinguer les données opérationnelles (< 3 ans) des données historiques (< 20 ans). Toutes les données métiers sont sauvegardées dans une seule base MongoDB.
- Déplacer les tables liées aux reportages de la base PostgreSQL (les tables « rapport\_ue », « rapport\_historique ») vers la base MongoDB contenant les données métiers.
- Ne sauvegarder en base de données MongoDB que des logs métiers. Les logs techniques doivent être sauvegardés dans des fichiers.
- Supprimer les tables liées aux rôles et utilisateurs de la base PostgreSQL. L’authentification et l’autorisation d’accès aux ressources seront gérées via l’annuaire LDAP.

Par conséquent, dans l’architecture cible, nous avons la répartition suivante :

- DB1 : Base MongoDB contenant les logs métiers d’audit générés par Talend
- DB2 : Base MongoDB contenant les données de mesures et statistiques + les données de rapportage
- DB3 : Base PostgreSQL contenant uniquement les référentiels + OpenLDAP

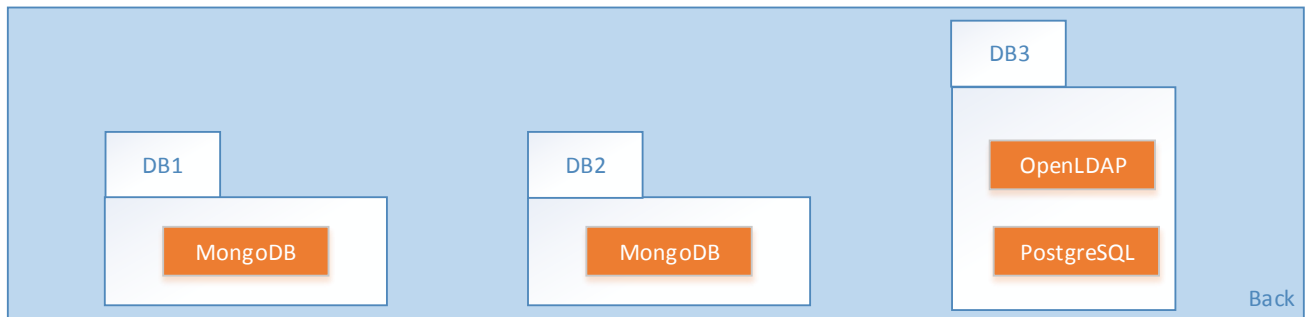


Figure 22: Architecture Back cible

## 8.2 APPROCHE INCREMENTALE

---

Pour permettre une mise en place progressive de l’architecture cible préconisée, nous proposons dans ce chapitre une approche incrémentale.

### 8.2.1 Incrément 1 : Découpler le serveur WEB1 des serveurs DB

L’objectif principal de cette première phase est de découpler le front WEB1 des serveurs DB. La Figure 23: Architecture après phase 1 montre l’architecture après l’exécution de cette étape.

Les actions suivantes doivent être effectuées :

- Intégrer l’annuaire OpenLDAP et migrer tous les rôles et utilisateurs.
- Développer des web services dans l’ESB couvrant tous les services existants dans la couche Symfony :
  - Services de consultations des données dans les bases PostgreSQL et MongoDB
  - Services d’authentification et d’autorisation.
- Modifier l’application web pour appeler les nouveaux services de l’ESB
- Supprimer les tables liées à la gestion des utilisateurs et des rôles de la base PostgreSQL

Par ailleurs, pour terminer avec le front WEB1, il est bien de profiter de cette étape pour supprimer les bundles SONATA et faire passer l’application en site multipages.

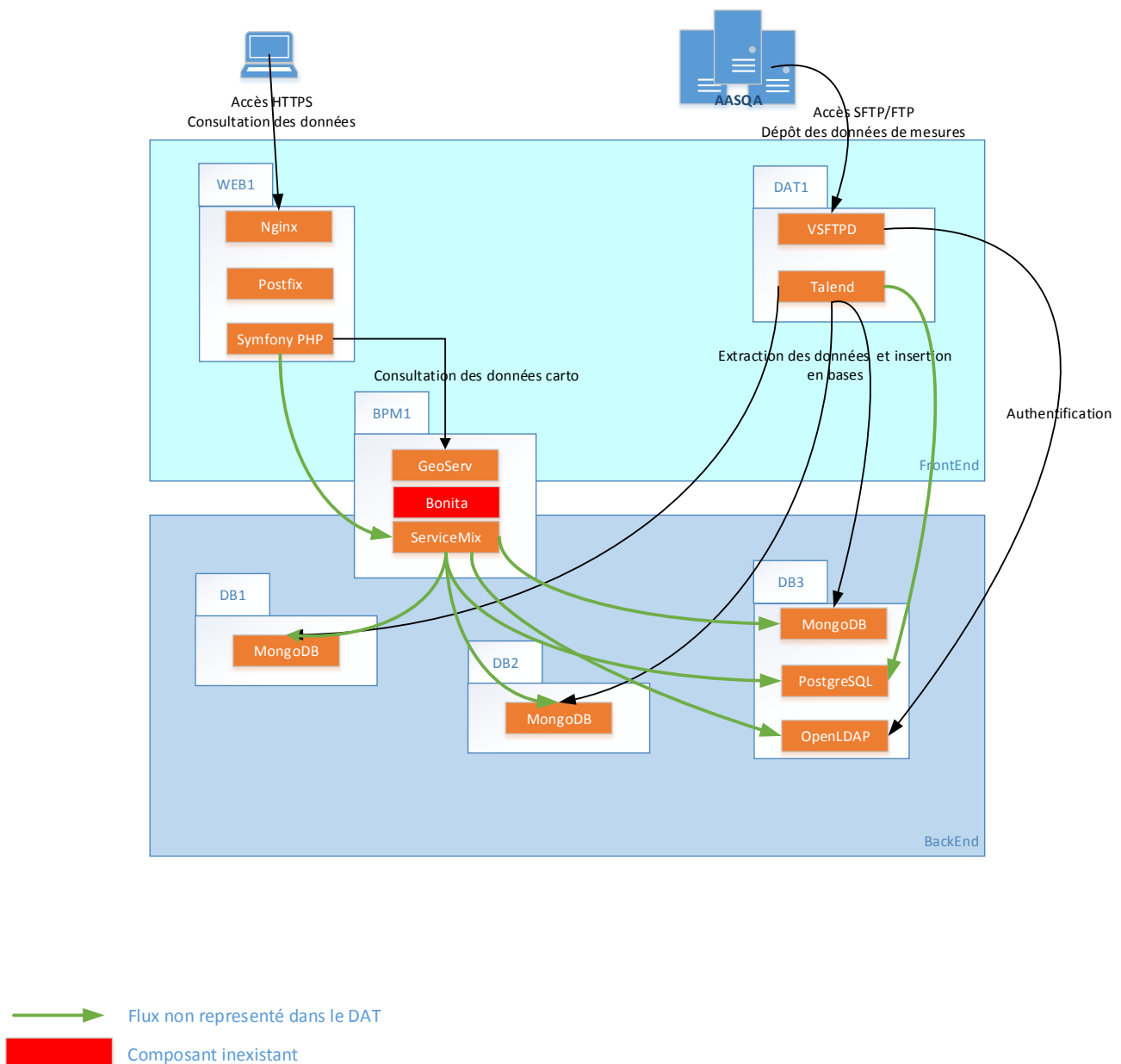


Figure 23: Architecture après phase 1

### 8.2.2 Incrément 2 : Transfert des fichiers depuis les AASQA via l'ESB

L'objectif de cette étape est de positionner l'ESB en intermédiation entre les applications AASQA et la solution Geod'Air en mode dégradé. Le transfert de fichiers ne se fera plus via connexion FTP mais via des messages. Les systèmes AASQA devront donc appeler un service JMS pour transférer les fichiers ISO. La production des fichiers ISO est donc toujours nécessaire.

Cette modification constitue une amélioration de la communication :

- Garantie de délivrance des données : les messages non délivrés seront envoyés de nouveau (un système de sauvegarde étant intégré au bus).
- Sécurité du transport des données : Le bus garantit un fort niveau de sécurité pour les données pouvant encryptant ces dernières.

Il n'est donc plus nécessaire d'utiliser le serveur FTP. Le serveur DAT1 doit être déplacé de la zone « Front »

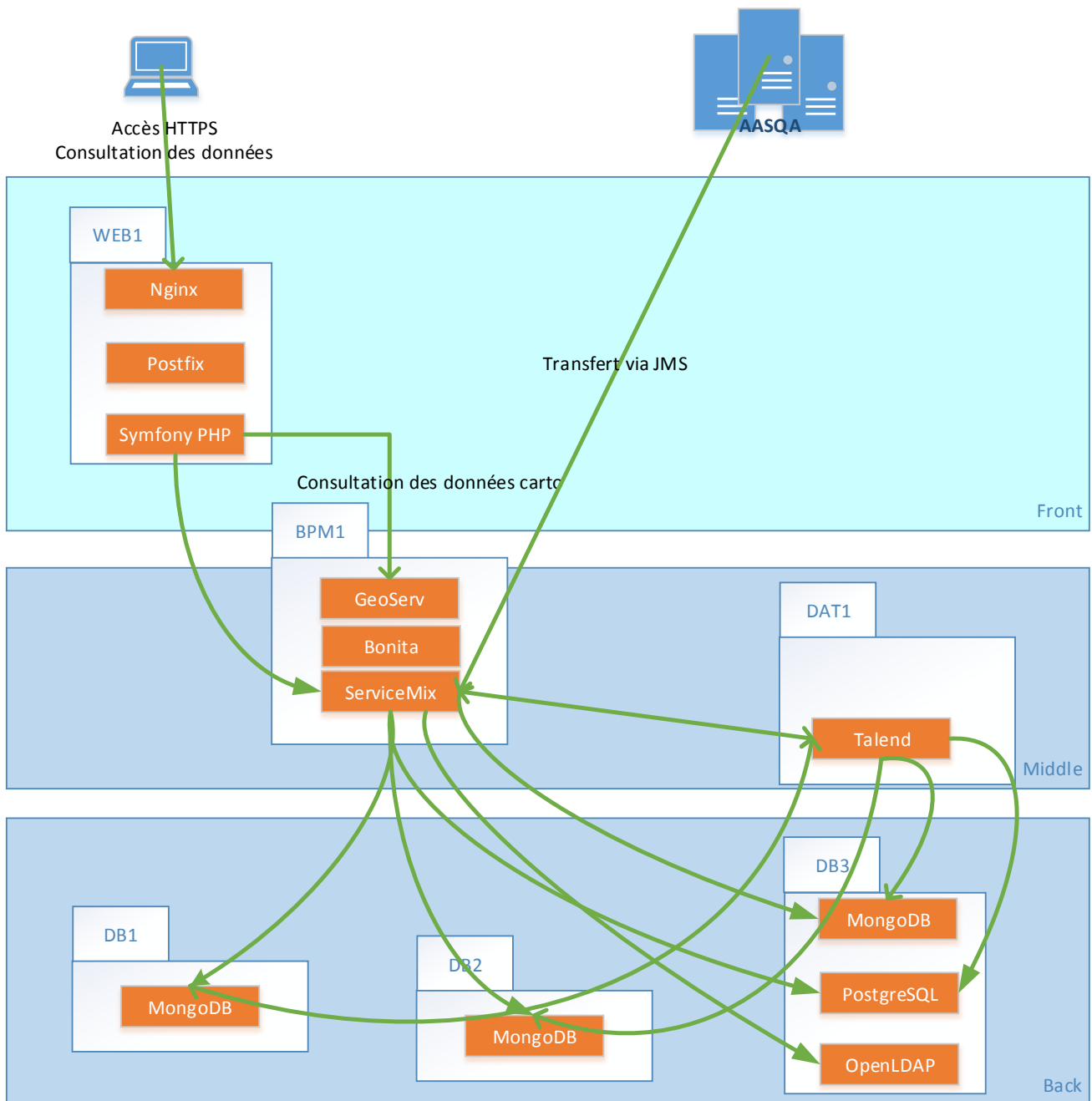


Figure 24: Architecture cible après l'étape 2

Le composant Bonita peut être intégré dans cette étape.

### 8.2.3 Incrément 3 : Optimisation des bases de données

L'objectif de cet incrément est d'obtenir l'architecture cible pour la zone « Back ». Les actions à effectuer correspondent exactement à celles décrites dans le paragraphe 8.1.5

Il faudra donc modifier la configuration des bases de données dans les applications clientes.

### **8.3 ACTIONS PRIORITAIRES POUR MISE EN PRODUCTION RAPIDE**

---

Sans devoir modifier l’architecture actuelle, corriger les différents constats (PA et NC) non liés à des problématiques d’architecture permettra d’obtenir un système tenant la charge pour l’ouverture à un plus large public.

Cependant pour obtenir une solution fiable, maintenable et évolutive, il est recommandé de mettre en place l’architecture cible préconisée.