



MINISTÈRE DE L'INTÉRIEUR

# **Cahier des Clause Technique Particulières** Tierce Maintenance Applicative de

## **FNAEG-NG**

(Fichier National Automatisé des Empreintes  
Génétique – Nouvelle Génération)

## **Annexe 2**

# **Architecture Technique**



# 1 Table des matières

Cahier des Clause Technique Particulières.....	1
2 Introduction.....	4
2.1 Cartographie du Système.....	4
3 Détail des exigences technique et de sécurité.....	6
3.1 Niveau de service attendu.....	6
3.1.1 Disponibilité.....	6
Plage d'ouverture.....	6
Continuité attendue.....	6
3.1.2 Intégrité.....	6
3.1.3 Niveau de Confidentialité.....	6
3.2 Contrôle d'accès.....	6
3.2.1 Accès à l'application FNAEG-NG.....	6
3.2.2 SSO.....	7
3.2.2.1 Différents SSO.....	7
3.2.2.2 Services offert par le SSO.....	7
3.2.3 Détails de l'authentification des utilisateurs.....	7
3.3 Traçabilité.....	8
4 ARCHITECTURE LOGIQUE.....	9
4.1 Principe de Structuration de l'application en modules applicatifs.....	9
4.1.1 Description des différents types de modules.....	10
4.1.1.1 Module applicatif IHM.....	10
4.1.1.2 Module applicatif de services.....	10
4.1.1.3 Module applicatif Batch.....	10
4.1.2 Interactions entre les modules applicatifs.....	10
4.2 Architecture Globale : Interaction entre les différents Systèmes.....	11
5 Architecture Physique.....	12
5.1 Schéma d'architecture Physique.....	12
5.1.1 Vue Générale.....	12
5.1.2 Tableau des composants logiciels.....	13
5.1.3 Flux.....	14
5.2 Dimensionnement.....	17
5.2.1 Environnement de Production.....	17
5.2.2 Environnement de développement.....	18
6 FNAEG.....	19
6.1 Architecture Logique.....	19
6.1.1 Architecture Applicative.....	19
6.1.1.1 Conception des modules applicatifs.....	19
6.1.1.1.1 Module « fnaeg-mai-web ».....	20
6.1.1.1.2 Module « fnaeg-format ».....	20
6.1.1.1.3 Module « fnaeg-domain ».....	21
6.1.1.1.4 Module « fnaeg-fwk-jpa-common ».....	21
6.1.1.1.5 Module « fnaeg-reports ».....	21
6.1.1.1.6 Module « fnaeg-mas-jms ».....	21
6.1.1.2 Traçabilité.....	21
6.1.1.3 Interfaces.....	22
6.1.1.4 Échanges Télématicues.....	22

## CCTP FNAEG-NG ANNEXE 2 ARCHITECTURE TECHNIQUE

6.1.1.5 Exports de la base de données.....	24
6.1.2 Architecture Logicielle.....	24
6.1.2.1 Schéma d'architecture logicielle.....	24
6.1.2.2 Tableau des Principaux Frameworks Java.....	25
7 PRÜM EG.....	26
7.1 Architecture Logique.....	26
7.1.1 Architecture Applicative.....	26
7.1.1.1 Conception des modules applicatifs.....	26
7.1.1.1.1 Module « prum-mai-web ».....	27
7.1.1.1.2 Module « prum-mas-metier ».....	27
7.1.1.1.3 Module « fnaeg-jpa-common-prum ».....	27
7.1.1.1.4 Module « prum-mas-edition ».....	27
7.1.1.1.5 Module « tigre-rest-lib ».....	27
7.1.1.2 Interfaces.....	27
7.1.2 Architecture Logicielle.....	28
7.1.2.1 Schéma d'architecture logicielle.....	28
7.1.2.2 Tableau des Principaux Framework Java.....	28
7.1.2.3 Batch applicatifs Java intégrés dans PRÜM.....	29
8 Statistiques.....	30
8.1 Architecture Globale.....	30
8.2 Statistiques FNAEG.....	30
8.2.1 Architecture applicative générale.....	31
8.2.2 Enchaînement des traitements.....	31
8.2.3 Dimensionnement Statistiques FNAEG.....	32
8.3 Statistiques PRUM.....	33
8.3.1 Architecture Applicative Générale.....	33
8.3.2 Enchaînement des Traitements.....	33
8.3.3 Dimensionnement Statistiques PRUM.....	34
9 TIGRE.....	35
9.1 Introduction.....	35
9.2 Présentation technique Générale de l'Application.....	35
9.2.1 Éléments Structurant de l'application.....	35
9.2.1.1 Contrainte Technique.....	35
9.2.1.2 Une application REST.....	35
9.2.1.3 Un moteur de recherche central.....	36
9.2.1.4 Une application non transactionnelle.....	37
9.2.1.5 Le cas d'utilisation de la base de données.....	37
9.2.1.6 L'interface d'administration.....	37
9.2.2 Architecture de la solution.....	38
9.2.2.1 Architecture logique.....	38
9.2.2.2 Architecture physique.....	39
9.2.2.2.1 Architecture physique du moteur de rapprochement.....	39
9.2.2.3 Description du flux réseau.....	40
9.2.3 Dimensionnement.....	41
9.2.3.1 Environnement de Production.....	41
9.2.3.2 Environnement de Développement.....	41
9.2.4 Supervision.....	42
9.3 Architecture détaillée.....	42

## CCTP FNAEG-NG ANNEXE 2 ARCHITECTURE TECHNIQUE

9.3.1	Architecture applicative.....	42
9.3.2	Composants de l'application et version.....	42
9.3.2.1	Composant de l'application.....	42
9.3.3	Architecture applicative.....	43
9.3.3.1	Moteur de recherche.....	43
9.3.3.2	Serveur d'application.....	43
9.3.3.3	L'interface d'administration.....	43
10	Communication Center.....	45
10.1	Introduction.....	45
10.1.1	Terminologie.....	45
10.2	Présentation technique générale de la solution.....	46
10.2.1	Les éléments structurants de la situation.....	46
10.2.1.1	Contrainte techniques.....	46
10.2.1.2	Un système de traitement de données NIFI.....	46
10.2.1.3	Une application API REST.....	46
10.2.2	Architecture Globale.....	46
10.3	Architecture Détaillée.....	47
10.3.1	Architecture Communication Center API RESTfull.....	47
10.3.2	Composant Communication Center API RESTfull.....	48
10.3.2.1	End-points exposés.....	48
10.3.2.2	Service XML.....	48
10.3.3	Architecture NIFI.....	49
10.3.3.1	Installation NIFI.....	49
10.3.3.2	Interface de configuration.....	49
10.3.4	Tâches et Processus Communication Center NIFI.....	50
11	Complexité de l'application FNAEG-NG et indicateurs technique.....	50
11.1	Le code.....	50
11.2	FNAEG.....	51
11.2.1	Analyse base de données.....	51
11.2.1.1	Structure.....	51
11.2.1.2	Volumétrie des tables (Avril 2025).....	51

## 2 Introduction

Ce document présente l'architecture technique du FNAEG-NG.

### 2.1 Cartographie du Système

Le système FNAEG-NG se compose des applications suivantes :

- FNAEG : Fichier National Automatisé des Empreintes Génétiques ;
- PRÜM : Gestion des échanges de données génétiques avec les états membres de l'Union Européenne ;
- TIGRE : Moteur de rapprochement d'ADN.
- STAT FNAEG : Statistiques du FNAEG
- STAT PRUM : Statistiques de PRUM
- STAT GIST : Statistiques de GIST

Pour les communications, le FNAEG-NG utilise les deux applications suivantes (hors champ du présent accord-cadre) :

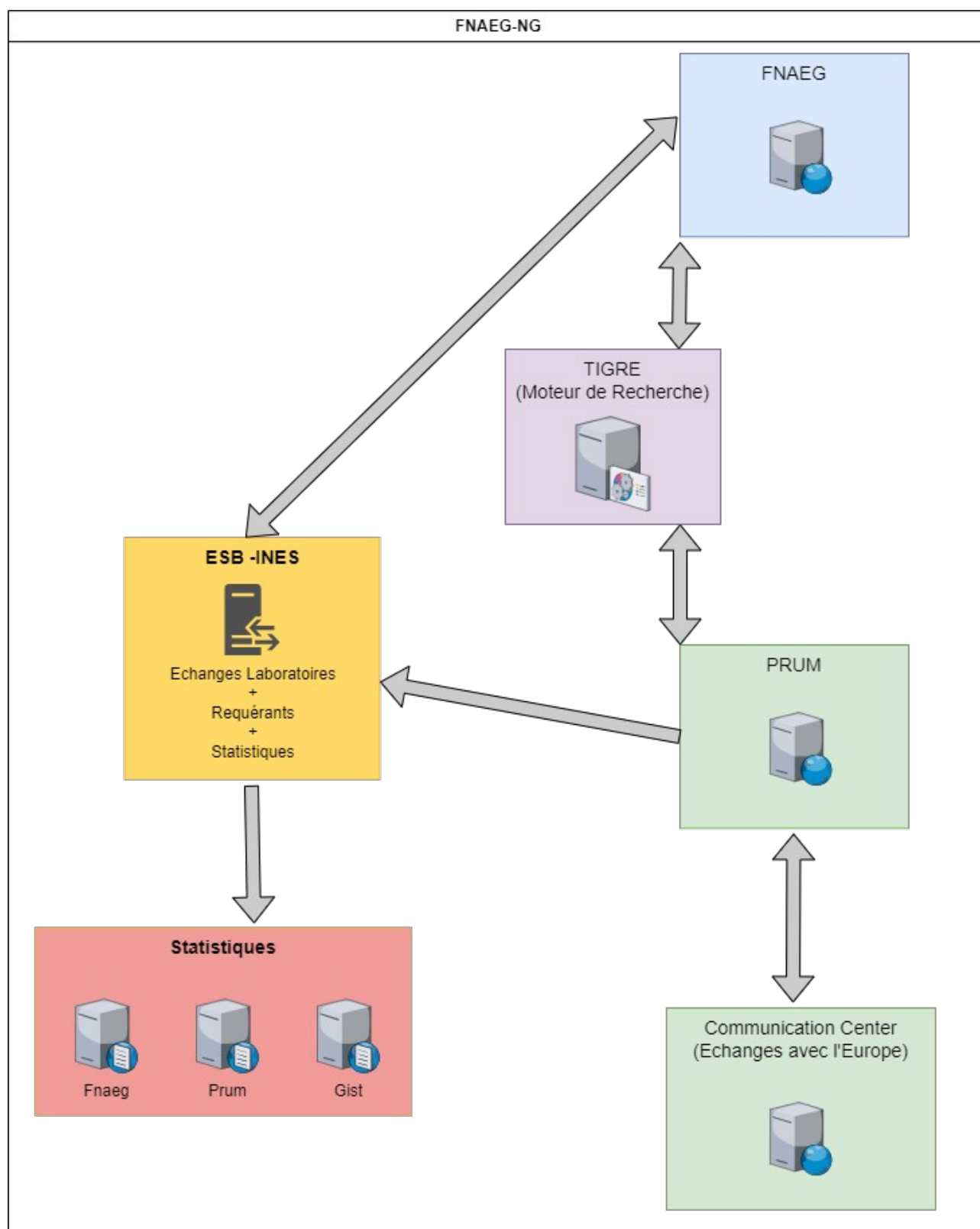
- Communication Center : centre de communication avec les États-Membres signataires du traité PRUM ;
- ESB – INES : Echanges avec les laboratoires et les requérants ainsi que les échanges avec les outils de statistiques ETL Talend et BI4.

Pour les statistiques, le FNAEG-NG utilise les outils ETL Talend et BI4 (hors champ du présent accord-cadre) .

Pour certaines recherches particulières, le service gestionnaire du FNAEG utilise l'application CODIS (hors champ du présent accord-cadre).

Une application de suivi du cycle de vie du code-barres a été développée par le service informatique du SNPS. Cette application, la « GIST1 » est alimentée par le service gestionnaire du FNAEG et est maintenue par le service informatique du SNPS.

## CCTP FNAEG-NG ANNEXE 2 ARCHITECTURE TECHNIQUE



### 3 Détail des exigences technique et de sécurité

#### 3.1 Niveau de service attendu

##### 3.1.1 Disponibilité

###### **Plage d'ouverture**

La durée maximale d'une indisponibilité est de 4 heures hors temps de bascule et temps de rechargement du moteur de rapprochement (plan de reprise d'activité en cours de test en avril 2021). Sur une année complète, le nombre maximal d'incidents entraînant une indisponibilité est de 4. Le cumul des indisponibilités annuelles doit être inférieur à 12H, soit un taux de disponibilité annuelle sur période d'ouverture du service de 99,5%.

###### **Continuité attendue**

Aucun mode dégradé n'existe pour les applications du FNAEG-NG.  
Le Ministère dispose d'un Plan de Reprise sur Activité (PRA) pour l'application FNAEG

##### 3.1.2 Intégrité

Le tableau ci-dessous résume les exigences d'intégrité du FNAEG-NG :

Intitulé	Description
Perte de données maximale (RPO)	Aucune perte de données n'est acceptée.
Niveau d'intégrité	L'information doit être parfaitement intègre.

##### 3.1.3 Niveau de Confidentialité

L'information est en diffusion restreinte

#### 3.2 Contrôle d'accès

##### 3.2.1 Accès à l'application FNAEG-NG

Les utilisateurs n'accèdent pas directement à l'application.

Ils s'adressent d'abord à la plate-forme SSO (Single Sign On) du ministère, celle-ci étant matérialisée par un Reverse Proxy. Une fois l'authentification réalisée, l'utilisateur est redirigé vers l'application et celle-ci obtient de la plate-forme SSO l'identité de cet utilisateur.

### 3.2.2 SSO

#### 3.2.2.1 *Differents SSO*

Il existe plusieurs SSO :

- le SSO GN authentifie les personnels de la Gendarmerie Nationale uniquement ;
- le SSO PN authentifie les personnels de la Police Nationale uniquement.

#### 3.2.2.2 *Services offert par le SSO*

Les services offerts par le Sso sont les suivants :

**- Authentification des utilisateurs :**

- elle fournit un système d'authentification unique pour l'ensemble des utilisateurs de la Police et de la Gendarmerie ;
- elle fournit les informations d'identité de l'utilisateur à l'application ainsi que le rôle de l'utilisateur.

**- Accounting :** ce service permet de transmettre à l'application un certain nombre d'informations (par exemple, les informations stockées dans l'annuaire) sur l'utilisateur qui s'y connecte.

**- Autorisation :** le SSO peut contrôler l'accès à certaines URL de l'application selon les utilisateurs qui essaient d'y accéder.

**- Gestion des traces :** le SSO enregistre les traces techniques d'accès à l'application. Ces traces peuvent être consultées avant une recherche dans les traces fonctionnelles.

**- Single Log Out (SLO) :** lors de la détection d'un changement des données transmises par le SSO, le FNAEG-NG purge les éventuels cookies de session de l'application.

### 3.2.3 Détails de l'authentification des utilisateurs

L'authentification est prise en charge par la plate-forme de SSO. L'identité de l'utilisateur est vérifiée au sein de l'annuaire LDAP via un des moyens



## CCTP FNAEG-NG ANNEXE 2 ARCHITECTURE TECHNIQUE

d'authentification suivants :

- le couple utilisateur / mot de passe ;
- la carte agent (comportant un certificat électronique).

Les informations envoyées par les SSO au FNAEG-NG sont sous forme des entêtes HTTP suivantes :

Paramètre	Description
Matricule	Le matricule de l'utilisateur
Nom	Le nom de l'utilisateur
Prenom	Le prénom de l'utilisateur
Institution	L'institution de l'utilisateur (Police ou Gendarmerie)
MatriculeAnonyme	Le matricule anonyme de l'utilisateur qui permet de tracer en base de données
Code-ATF	Le profil de l'utilisateur
NIA	Numéro d'immatriculation administrative

Le paramètre « Code-ATF » permet de définir le profil de connexion pour l'utilisateur. Les différents profils de l'application sont les suivants :

- Administrateur ;
- Gestionnaire ;
- Opérateur ;
- APJ (PN ou GN) ;
- OPJ (PN ou GN) ;
- SCPPB ;
- Gestionnaire chef PTS ;
- Profils des personnels scientifiques (ingénieur, technicien et agent spécialisé de PTS).
- Profil technique

Note : Pour l'application Prüm seuls les profils Administrateur, Gestionnaire et Opérateur sont opérationnels. Le profil Gestionnaire a des droits identiques à ceux du profil Opérateur.

### 3.3 Traçabilité

Il faut distinguer la traçabilité liée aux SSO de la traçabilité interne à l'application.

En effet, le module métier de traçabilité des actions des utilisateurs est implémenté directement dans les applications FNAEG et Prüm. Les traces qu'il génère sont stockées dans la base de données des applications concernées. Ce module est complémentaire des fonctions de traçabilité des systèmes en charge du contrôle d'accès des utilisateurs.

Les applications FNAEG et Prüm enregistrent chaque action des utilisateurs qui provoquent une modification ou une consultation d'informations de la base de données. Cette action est caractérisée par :

- la date (date, heure, minute et seconde) ;
- le code utilisateur anonymisé en ce qui concerne les utilisateurs de la Police Nationale ;
- la fonctionnalité utilisée ;
- des informations complémentaires.

Les actions réalisées peuvent ainsi être retrouvées (sauvegarde complète des actions) et associées à un utilisateur dans la limite du niveau d'authentification mis en place.

Les SSO (PN et GN) enregistrent aussi ce qu'ils perçoivent des actions des utilisateurs (notamment l'authentification et l'accès à l'application).

Ces traces enregistrées dans la base de données ne sont pas accessibles par l'IHM du FNAEG. Seul un administrateur de la base de données est en mesure de les extraire.

La traçabilité de l'application Prüm n'est pas partagée avec l'application Communication Center.

## 4 ARCHITECTURE LOGIQUE

### 4.1 Principe de Structuration de l'application en modules applicatifs

**Remarque** : les principes présentés dans ce chapitre sont issus du cadre de cohérence technique du Ministère.

- Module Applicatif IHM (MAI)
- Module Applicatif de Service (MAS)
- Module Applicatif Batch (MAB)

### 4.1.1 Description des différents types de modules

#### 4.1.1.1 *Module applicatif IHM*

Un Module Applicatif IHM implante un (ou plusieurs) cas d'utilisation. Il pilote les interactions entre l'utilisateur et les autres Modules Applicatifs de Services. Il peut stocker des données à des fins de traçabilité et de reprise après interruption de service et des données liées à des objets métier stockés uniquement de manière temporaire (durée de vie d'une session).

#### 4.1.1.2 *Module applicatif de services*

Un Module Applicatif de Services contient un ensemble cohérent de données métier (silo de données) et fournit les services de gestion de ces données. Il est garant de la cohérence de son silo au niveau fonctionnel et technique. Les modèles de données de deux MAS sont nécessairement disjoints (principe d'unicité de la localisation d'une donnée). Un MAS peut néanmoins stocker des identifiants informatiques exportés par un autre MAS. Un MAS expose les services nécessaires à l'implantation des cas d'utilisation d'un MAI ou d'un MAB de manière :

- ⌚ atomique : un appel de service est soit complètement exécuté, soit pas du tout exécuté ; il conduit toujours à un état fonctionnel et technique cohérent des données ;
- sans état : chaque appel de service est traité uniquement en fonction de ses paramètres, de manière indépendante et sans relation avec les appels précédents.

#### 4.1.1.3 *Module applicatif Batch*

Un Module Applicatif Batch implante un (ou plusieurs) cas d'utilisation. Il pilote les interactions entre un composant applicatif de traitement par lot et les autres Modules Applicatifs de Services. Il peut stocker des données à des fins de traçabilité, de reprise après interruption de service et des données liées à des objets métier stockées uniquement de manière temporaire (durée de vie d'un traitement).

### 4.1.2 Interactions entre les modules applicatifs

Une interaction est une action exercée par un composant applicatif d'un Module Applicatif sur un composant applicatif d'un autre Module Applicatif.

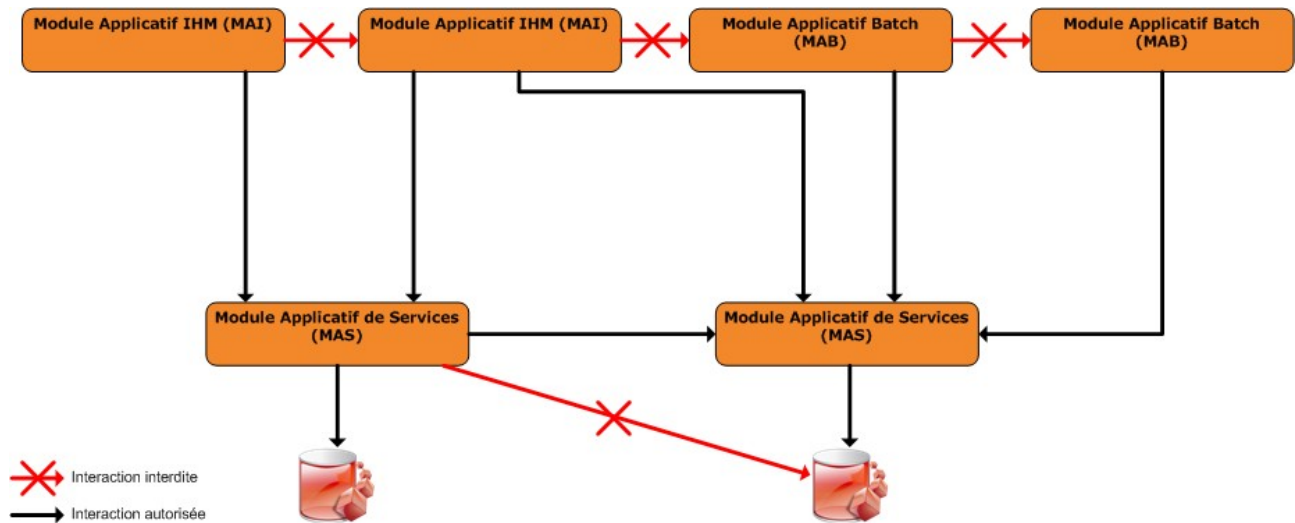
Les règles dans les interactions sont les suivantes :

- un MAI ne peut invoquer un autre MAI ou un MAB ;
- un MAB ne peut invoquer un autre MAB ou un MAI.

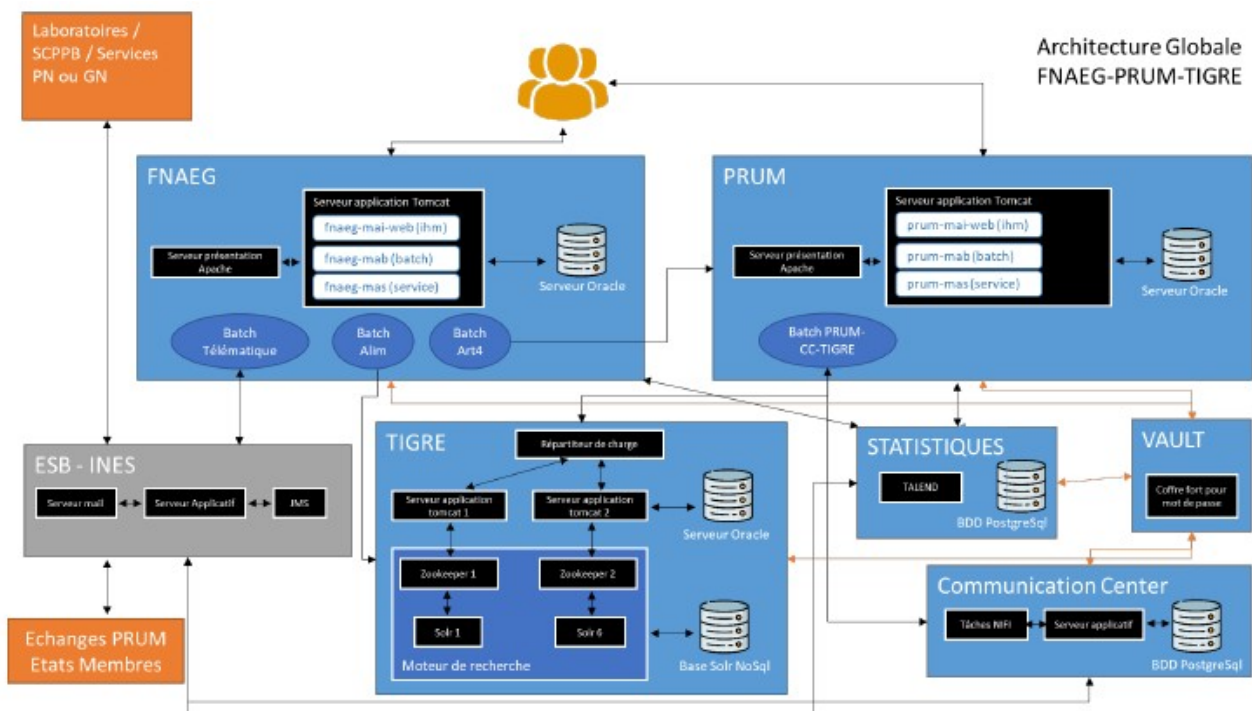
Un composant applicatif n'est pas autorisé à accéder directement aux données métier de l'application. Il est obligatoire d'utiliser les services exposés par le Module Applicatif de Services.

## CCTP FNAEG-NG ANNEXE 2 ARCHITECTURE TECHNIQUE

L'illustration ci-dessous présente de manière schématisée les interactions autorisées ou interdites entre les Modules Applicatifs :



## 4.2 Architecture Globale : Interaction entre les différents Systèmes

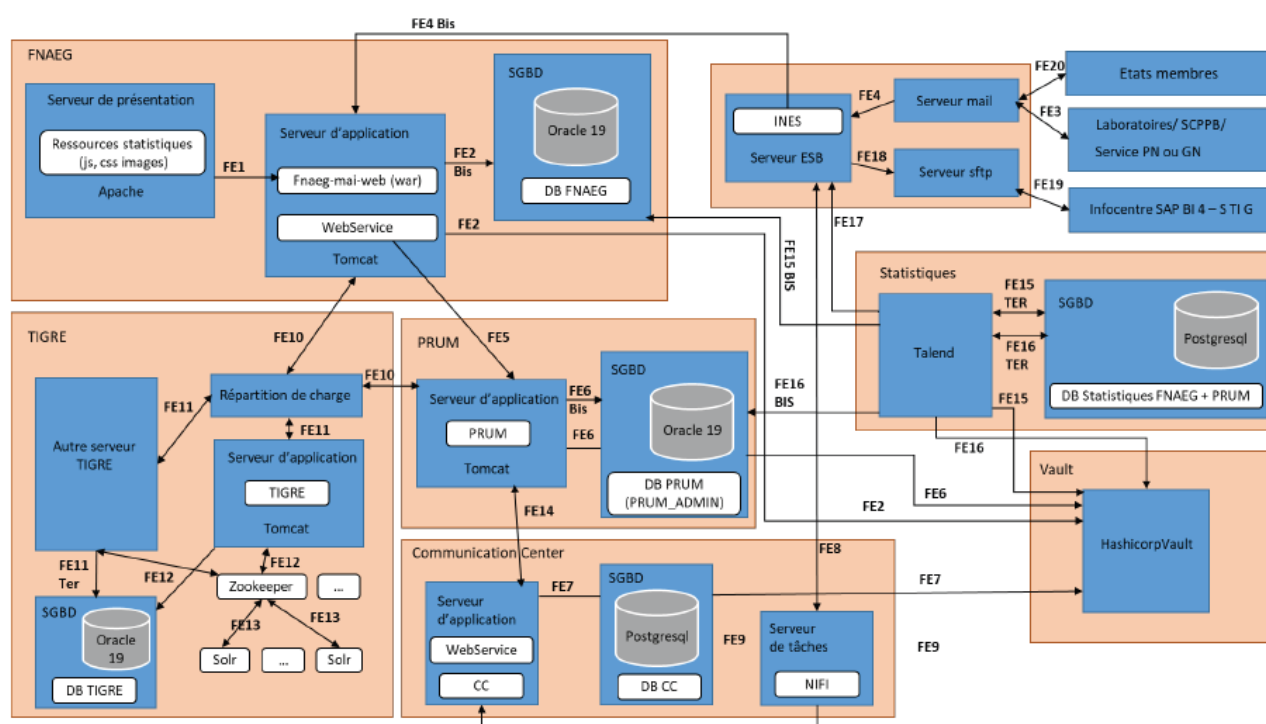


## 5 Architecture Physique

### 5.1 Schéma d'architecture Physique

#### 5.1.1 Vue Générale

Au niveau de la vue générale, l'architecture physique du FNAEG-NG se présente comme suit :



### 5.1.2 Tableau des composants logiciels

Les composants logiciels sur lesquels repose le FNAEG-NG sont récapitulés dans les tableaux ci-dessous :

#### FNAEG

Type	Produit	Version
Frontal	Apache Server	2,4,X
Chiffrement	OpenSSL	0.9.7i
Container Web	Tomcat	9.0.X
JDK	Oracle JDK	Jdk-17.0.2
SGBD	Oracle	19c

#### PRUM

Type	Produit	Version
Container Web	Tomcat	9.0.X
JDK	Oracle JDK	Jdk-17.0.2
SGBD	Oracle	19c

#### TIGRE

Type	Produit	Version
Serveur d'application	Apache Tomcat	9.0.X
Moteur de recherche	Apache SolR	8.11.1
Système distribué pour le moteur de recherche	Zookeeper	3.0.8
SGBD	Oracle	19c
JDK	Oracle JDK	Jdk-17.0.2

#### CC

Type	Produit	Version
Container Web	Spring Boot REST API (Intègre un Apache Tomcat)	3.2.0

## CCTP FNAEG-NG ANNEXE 2 ARCHITECTURE TECHNIQUE

JDK	Open JDK	21
OS	Linux Debian	12
SGBD PostGRESQL	Serveur de base de données des indicateurs statistiques	15.5
NIFI	Apache NIFI	1.25.0

### **STATISTIQUES**

Type	Produit	Version
Talend	Construction des traitements de création des indicateurs (langage Java généré)	7.3.2
JAVA	Version Java sur lequel exécuter les jobs Talend	Oracle JDK 11.0.17
SGBD PostGRE SQL	Serveur de base de données des indicateurs statistiques	13.5

### **VAULT**

Type	Produit	Version
Coffre fort	Harshicorp Vault	1.13.2

### **5.1.3 Flux**

Le tableau des flux externes est le suivant :

N° Flux	Émetteur	Destinataire	Protocole	Description
FE1	Serveur de présentation FNAEG	Serveur d'application FNAEG	AJP	Connecteur entre le serveur Apache et le serveur d'application Tomcat
FE2	Serveur d'application FNAEG	VAULT	REST / HTTPS	Récupération des mots de passe de connexion à la base de données.

## CCTP FNAEG-NG ANNEXE 2 ARCHITECTURE TECHNIQUE

N° Flux	Émetteur	Destinataire	Protocole	Description
FE2 Bis	Serveur d'application FNAEG	SGBD FNAEG	JDBC	Interrogation de la base de données FNAEG
FE3	Laboratoires / SCPPB / Services PN ou GN	Serveur mail	S/MIME	Échange télématique
FE3	Serveur Mail	Laboratoires /SCPPB / Services PN ou GN	S/MIME	Échange télématique – réponse et acquittements
FE4	Serveur Mail	Serveur d'application INES	S/MIME	Échange télématique
FE4	Serveur d'application INES	Serveur Mail	S/MIME	Échange télématique – réponse et acquittements
FE4 Bis	Serveur d'application INES	Serveur d'application FNAEG	JMS	Échange télématique
FE4 Bis	Serveur d'application FNAEG	Serveur d'application INES	JMS	Échange télématique – réponse et acquittements
FE5	Serveur d'application FNAEG	Serveur d'application PRUM	Webservice	Mise à jour base PRUM
FE5 Bis	SGBD FNAEG	SGBD PRUM	JDBC	Mise à jour base PRUM (par batch)
FE6	Serveur d'application PRUM	Vault	REST / HTTPS	Récupération des mots de passe de connexion à la base de données
FE6 Bis	Serveur d'application PRUM	SGBD PRUM	JDBC	Interrogation de la base de données PRUM
FE7	Serveur d'application CC	SGBD CC	JDBC	Interrogation de la base de données CC
FE7	Serveur d'application CC	Vault	REST / HTTPS	Récupération des mots de passe de connexion à la base de données
FE8	Serveur de tâches CC	Serveur d'application INES	JMS	Échange de fichiers



## CCTP FNAEG-NG ANNEXE 2 ARCHITECTURE TECHNIQUE

N° Flux	Émetteur	Destinataire	Protocole	Description
FE9	Serveur de tâches CC	Serveur d'application CC	REST / HTTPS	Récupération des questions/réponses des Etats membres
FE10	Serveur d'application FNAEG	Répartiteur de charge	REST	Interaction avec Tigre
FE10	Serveur d'application PRUM	Répartiteur de charge	REST	Interaction avec Tigre
FE10	Répartiteur de charge	Serveur d'application FNAEG	REST	Interaction avec Tigre
FE10	Répartiteur de charge	Serveur d'application PRUM	REST	Interaction avec Tigre
FE11	Répartiteur de charge	Serveur d'application TIGRE	REST	Transfert de la demande au serveur Tigre en fonction de la charge
FE11	Serveur d'application Tigre	Répartiteur de charge	REST	Retour de la requête Tigre
FE11 Bis	Serveur d'application	Vault	REST / HTTPS	Récupération des mots de passe de la connexion à la base de données
FE11 Ter	Serveur d'application TIGRE	SGBD TIGRE	JDBC	Interrogation de la base de données TIGRE
FE12	Serveur d'application TIGRE	Zookeeper	HTTP	Interaction avec Tigre
FE12	Zookeeper	Serveur d'application TIGRE	HTTP	Interaction avec Tigre
FE13	Zookeeper	SolR	HTTP	Interaction avec Tigre
FE13	SolR	Zookeeper	HTTP	Interaction avec Tigre
FE14	Application PRUM	Serveur d'application CC	REST/ HTTPS	Transfert des questions / réponses
FE14 Bis	Répartiteur de charge	SGBD Prum	JDBC	Transfert des questions / réponses

## CCTP FNAEG-NG ANNEXE 2 ARCHITECTURE TECHNIQUE

N° Flux	Émetteur	Destinataire	Protocole	Description
FE15	Talend	Vault	REST/ HTTPS	Récupération des mots de passe de connexion à la base de données. (Oracle + Postgresql)
FE15 Bis	SGBD FNAEG	Talend	JDBC	Transfert des données FNAEG
FE15 Ter	Talend	SGBD PostGRESQL	JDBC	Insertion des enregistrements en table statistiques FNAEG
FE16	Talend	Vault	REST/ HTTPS	Récupération des mots de passe de connexion à la base de donnée
FE16 Bis	SGBD PRUM	Talend	JDBC	Transfert des données PRUM
FE16 Ter	Talend	SGBD PostGRESQL	JDBC	Insertion des enregistrements en table statistiques PRUM
FE17	Talend	Serveur d'application INES	SFTP	Transfert de fichiers statistiques pour alimentation BI4
FE18	Serveur d'application INES	Serveur SFTP	SFTP	Transfert de fichiers statistiques pour alimentation BI4
FE19	Serveur SFTP	Infocentre BI4	SFTP	Transfert de fichiers statistiques pour alimentation BI4
FE20	Serveur Mail	Etats Membres	S/MIME	Question/réponses des états membres

## 5.2 Dimensionnement

### 5.2.1 Environnement de Production

Le tableau ci-dessous présente le dimensionnement de l'environnement de production :

Élément	OS	vCPU	RAM	Disque	Quantité
Serveur de présentation FNAEG et PRUM	Linux 64bits (Debian 11)	2	2 Go	40 Go	2
Serveur d'application FNAEG	Linux 64Bits (Debian 11)	8	4 Go	575 Go	1

## CCTP FNAEG-NG ANNEXE 2 ARCHITECTURE TECHNIQUE

Élément	OS	vCPU	RAM	Disque	Quantité
Serveur d'application PRUM	Linux 64Bits (Debian 11)	4	4 Go	30 Go	1
Serveur d'application TIGRE	Linux 64Bits (Debian 11)	4	4 Go	30 Go	2
Serveur de Moteur de recherche	Linux 64Bits (Debian 11)	10	12 Go	30 Go	6
Base de données FNAEG et PRUM	AIX 7.2	6	48 Go	800 Go	1
Vault	Linux 64Bits (Debian 11)	8	4 Go	15 Go	1
Communication Center	Linux 64Bits (Debian 12)	4	8 Go	100 Go	2
Base de données CC	Linux 64Bits (Debian 12) PostGreSql	6	12 Go	1500 Go	1

### 5.2.2 Environnement de développement

Sauf dans des cas d'usages particuliers comme les tests de performances ou les tests d'intégration, les environnements de développement peuvent être composés d'un serveur dont les caractéristiques sont :

Élément	OS	vCPU	RAM	Disque	Quantité
Serveur de présentation FNAEG et PRUM	Linux 64bits (Debian 11)	1	1 Go	30 Go	1
Serveur d'application FNAEG	Linux 64Bits (Debian 11)	2	2 Go	30 Go	2
Serveur d'application PRUM	Linux 64Bits (Debian 11)	2	2 Go	30 Go	1

## CCTP FNAEG-NG ANNEXE 2 ARCHITECTURE TECHNIQUE

Élément	OS	vCPU	RAM	Disque	Quantité
Serveur d'application TIGRE	Linux 64Bits (Debian 11)	2	2 Go	30 Go	2
Serveur de Moteur de recherche	Linux 64Bits (Debian 11)	2	2 Go	30 Go	1
Base de données FNAEG et PRUM	AIX 7.2	2	16 Go	250 Go	1
Communication Center	Linux 64Bits (Debian 12)	4	8 Go	100 Go	2
Base de données CC	Linux 64Bits (Debian 12) PostGreSql	6	12 Go	1500 Go	1

## 6 FNAEG

### 6.1 Architecture Logique

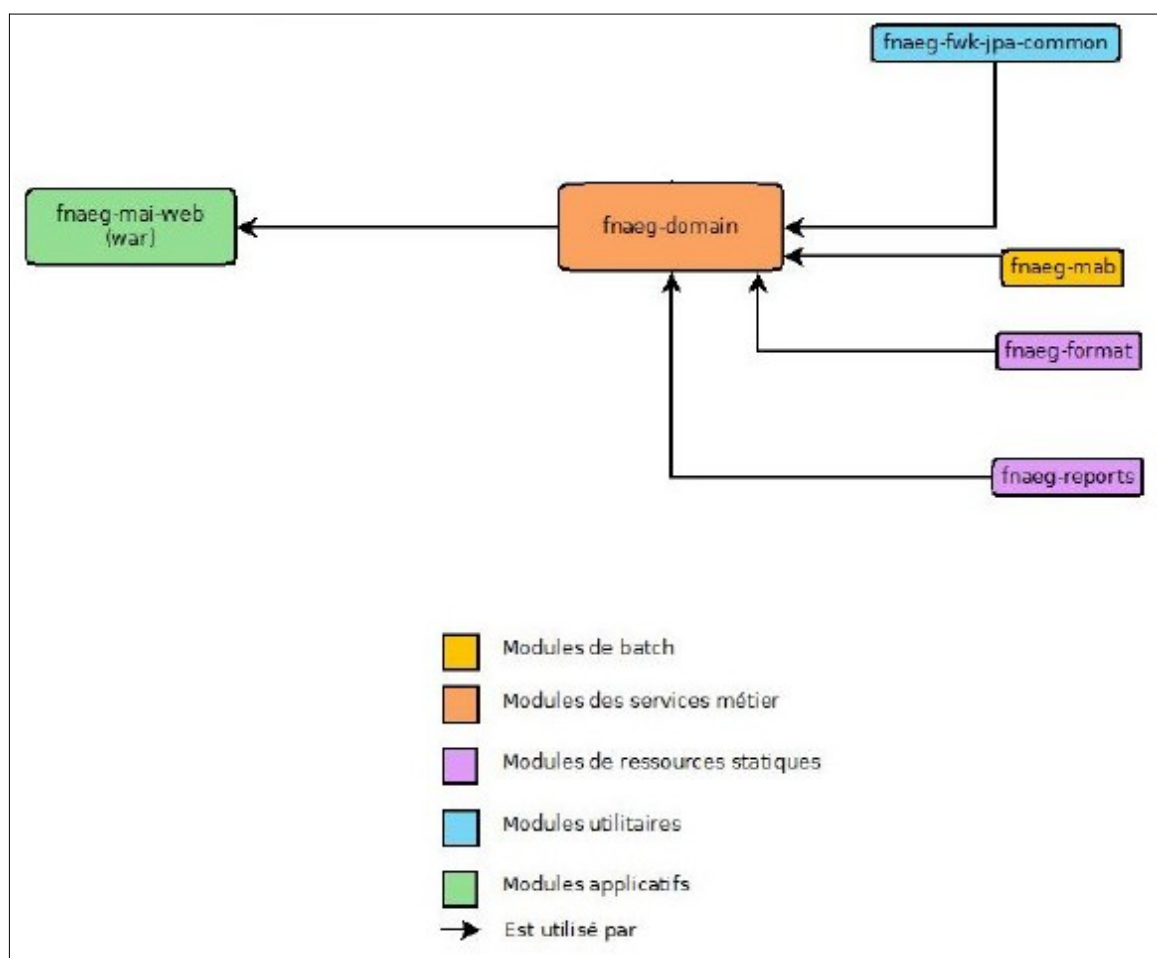
#### 6.1.1 Architecture Applicative

##### 6.1.1.1 Conception des modules applicatifs

<b>fnaeg-app</b>	Génération des livrables (war, ear)			
<b>fnaeg-conf-reporting</b>	Configuration de l'audit de code (findbugs/checkstyle)			
<b>fnaeg-db</b>	Scripts sql patch (jusqu'à la V8.1.2)			
<b>fnaeg-flyway</b>	Gestion des patchs de base de données (depuis la V8.1.2)			
<b>fnaeg-mab</b>	Gestion du batch d'alimentation de tigre, identification trace, article4, batch d'intégrité			
<b>fnaeg-mai</b>	Modules interfaces			
	<b>fnaeg-mai-web</b>	Interface Web Fnaeg		
<b>fnaeg-mas</b>	Modules services			
	<b>fnaeg-common</b>	Ressources communes Fnaeg		
		<b>fnaeg-format</b>	Formats XSD	
			<b>fnaeg-anomalie-chiffre</b>	XSD anomalie chiffre
			<b>fnaeg-codis</b>	XSD CODIS

## CCTP FNAEG-NG ANNEXE 2 ARCHITECTURE TECHNIQUE

			<b>fnaeg-requisition-analyse</b>	<i>XSD réquisition analyse</i>
			<b>fnaeg-transfert-profil</b>	<i>XSD transfert-profil</i>
		<b>fnaeg-reports</b>	<i>Rapports Jasper</i>	
	<b>fnaeg-metier</b>	<i>Services métier FNAEG</i>		
		<b>fnaeg-domain</b>	<i>Services du domaine FNAEG</i>	
	<b>Fnaeg-mas-jms</b>	<i>Gestion de l'interfaçage avec INES</i>		



### 6.1.1.1.1 Module « fnaeg-mai-web »

Le module « fnaeg-mai-web » implémente le modèle MVC contenant les vues et contrôleurs pour la gestion de l'interaction avec l'application. Il constitue le packaging Web de l'application FNAEG.

### 6.1.1.1.2 Module « fnaeg-format »

Le module « fnaeg-format » permet la génération des XSD de l'application en objet Java.

#### **6.1.1.1.3 Module « *fnaeg-domain* »**

Le module « *fnaeg-domain* » contient les services métier de l'application FNAEG. Ces services sont utilisés par le module « *fnaeg-web* » et « *fnaeg-mab* ».

#### **6.1.1.1.4 Module « *fnaeg-fwk-jpa-common* »**

Le module « *fnaeg-fwk-jpa-common* » est un module offrant des services génériques de gestion de la persistance des données selon le standard Java JPA. Ce module est utilisé suite à l'introduction du framework Hibernate dans les applications FNAEG et la refonte PRÜM EG.

#### **6.1.1.1.5 Module « *fnaeg-reports* »**

Le module « *fnaeg-reports* » contient les templates JasperReports des différents rapports de l'application. Les rapports sont bâtis avec l'outil Jasper Report. L'application FNAEG n'est pas connectée à une GED. Le moteur documentaire est implémenté directement dans l'applicatif et les rapports de rapprochement générés par le FNAEG sont stockés dans le système de fichiers du disque serveur dans le dossier :

/XXX/out/rapport/YYYY/MM (avec YYYY l'année, MM le mois)

Compte-tenu de la volumétrie de rapports stockés, ce dossier est un lien vers une partition du disque dédiée au stockage des rapports. La forme des fichiers est la suivante : Rapprochement\_XXXX\_YYYYYYYYYYYY.pdf.

Où

- XXXX correspond au numéro de travail,

- YYYYYYYYYYYY correspond au numéro de code barre échantillon.

Ces rapports peuvent être téléchargés pendant un mois depuis l'application. Ils sont conservés sur le disque pendant 3 ans.

#### **6.1.1.1.6 Module « *fnaeg-mas-jms* »**

Le module « *fnaeg-mas-jms* » permet l'interfaçage avec les files JMS fournies par INES et le déclenchement des traitements métiers associés.

### **6.1.1.2 Traçabilité**

L'application FNAEG enregistre chaque action des utilisateurs qui provoquent une modification ou une consultation d'informations de la base de données. Cette action est caractérisée par :

- la date (date, heure, minute, seconde),
- l'utilisateur (via son matricule pour la GN, via son matricule anonymisé pour la PN),
- la fonctionnalité
- des informations complémentaires.

Les actions réalisées peuvent ainsi être retrouvées (sauvegarde complète des actions) et imputées à un utilisateur dans la limite du niveau d'authentification mis en place.

Les gérants d'habilitation (Cheops, Proxyma) enregistrent aussi ce qu'ils perçoivent des actions des utilisateurs : authentification, accès à une fonction,...

Ces traces enregistrées dans la base de données ne sont pas accessibles par l'IHM du FNAEG. Ces informations ne peuvent être extraites qu'en requêtant directement la base de données.

### **6.1.1.3 Interfaces**

Les interfaces avec les autres systèmes d'information se divisent en cinq catégories.

- Les échanges télématiques :
  - réception des profils envoyés par les laboratoires
  - réception de demandes de données administratives
  - émission des rapports de rapprochement
- Les exports de base de données :
  - alimentation de la base CODIS
  - alimentation de la base PRÜM (génération du fichier pour les article 4 de PRÜM)
- Les imports depuis un support amovible :
  - réception de profils génétiques envoyés par les laboratoires
- Interactions avec Tigre :
  - Ajout d'indexations
  - Suppression d'indexations
  - Lancement de rapprochements
  - Mise à jour de l'identification de traces
  - Récupération des résultats de rapprochements
- Interactions avec Vault
  - Récupération des informations d'identification de la base de données

### **6.1.1.4 Échanges Télématiques**

Les échanges télématiques avec le FNAEG utilisent des courriels au format S/MIME. Le FNAEG utilise l'adresse source pour identifier l'émetteur. Les courriels sont chiffrés et signés numériquement par le FNAEG et les laboratoires ou le SCPPB.

#### **Format des messages**

Les messages sont caractérisés par :

- un corps de message toujours vide,

## CCTP FNAEG-NG ANNEXE 2 ARCHITECTURE TECHNIQUE

- une pièce jointe en format zip (le fichier zip contient 1 ou n fichiers. Chaque fichier contient un (et un seul) arbre XML) ou un (et un seul) fichier XML.
- Principes de sécurité

Les messages sont sécurisés par l'utilisation du standard S/MIME. Ce standard est défini dans les

RFC suivantes : 2632, 2633, 2634, 3369, 3370

Les entités émettrices et destinataires disposent de certificats produits par le ST(SI)<sup>2</sup>.

Pour interagir avec la messagerie électronique, l'application FNAEG utilise le projet INES qui scrute la boîte mail et transmet les informations nécessaires au traitement du mail dans une file JMS. L'application FNAEG n'est donc jamais directement connectée à la boîte mail, mais à des files JMS contenant les informations.

### ***Réception des profils envoyés par les laboratoires***

Cette interface réalise le transfert des profils génétiques par les laboratoires et la transmission de l'acquittement par l'application FNAEG.

L'échange se déroule en deux étapes :

- Le laboratoire transmet au FNAEG un à n profils génétiques venant du même expert.

Le FNAEG envoie un acquittement ou un message d'exception.

Si le FNAEG ne reçoit pas les données administratives dans les délais impartis, celui-ci peut émettre un troisième type de message :

- Le FNAEG informe le laboratoire de la suppression d'un profil génétique.

### ***Demandes de données administratives***

Cette interface réalise la demande des données administratives par les laboratoires ou le SCPPB et la transmission de ces données par l'application FNAEG.

L'échange se déroule en 2 étapes :

- Le laboratoire ou le SCPPB envoie une demande de données administratives.
- Le FNAEG répond avec les données administratives ou un message d'exception.
- Le laboratoire ou le SCPPB envoie un message d'acquittement.

### ***Émission des rapports de rapprochement***



Le FNAEG utilise aussi le projet INES pour envoyer les rapports de rapprochement aux requérants.

#### **6.1.1.5 Exports de la base de données**

##### ***Alimentation de la base CODIS***

L'alimentation de la base CODIS est une opération manuelle exécutée quotidiennement. L'application FNAEG permet la génération d'un fichier au format XML CODIS (CODIS CMF 3.2 Interface specification). Ce fichier est exploité par l'application CODIS.

##### ***Alimentation de la base PRÜM***

La base de données PRÜM est mise à jour via webservice ou batch à chaque envoi de questions par la France, ainsi que par les réponses et questions posées par les pays membres.

Les données transmises vers la base PRÜM sont :

- les profils génétiques ;
- les codes-barres échantillon ;
- la nature de l'échantillon (Individu ou Trace).

##### ***Import de profils génétiques***

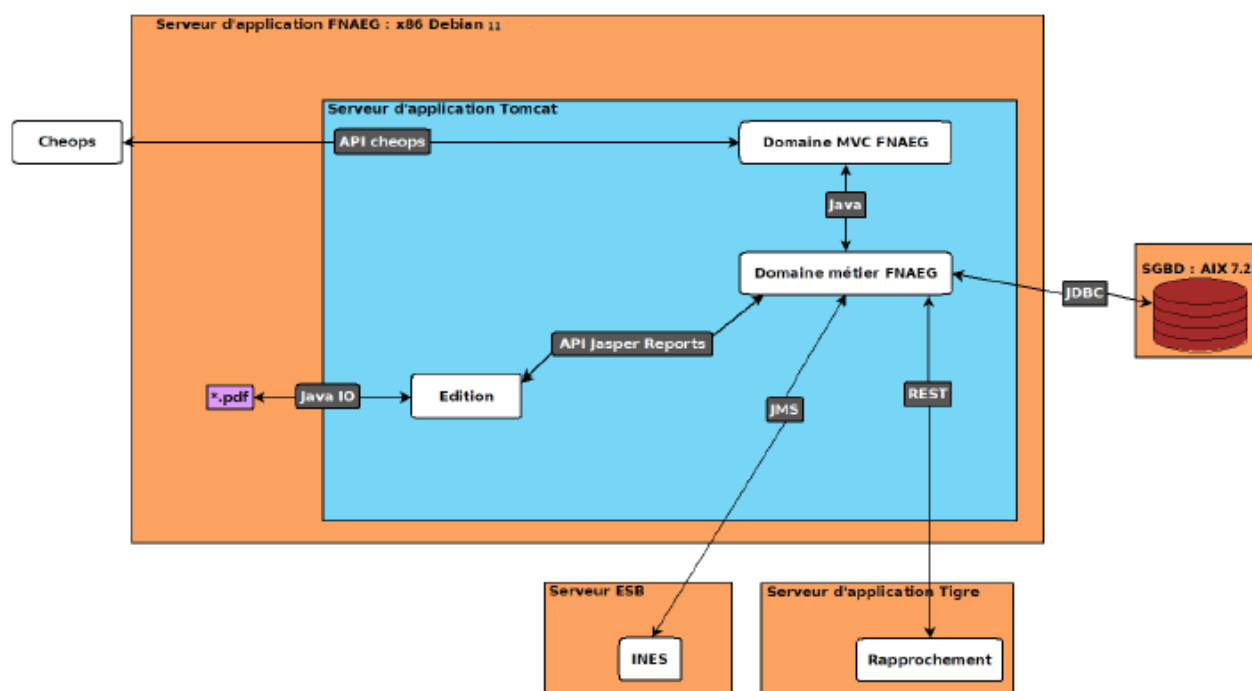
L'import de profils génétiques depuis un support amovible est une opération manuelle. L'opérateur qui effectue cet import reçoit un support (CD-ROM) contenant une ou plusieurs archives ZIP contenant des fichiers XML. Chaque fichier contient un ou plusieurs profils.

Pour chaque fichier reçu, l'opérateur utilise la fonction d'importation (upload) qui est mise à sa disposition dans l'interface de gestion.

### **6.1.2 Architecture Logicielle**

#### **6.1.2.1 Schéma d'architecture logicielle**

Le schéma ci-dessous présente la communication entre les principaux modules techniques de l'application FNAEG.



### 6.1.2.2 Tableau des Principaux Frameworks Java

Nom	Version	Description
JasperReport	6.19.0	Génération des rapports
Spring	5.3.18	Framework transverse (Pattern IOC)
slf4j	1.7.36 (base on Log4j 2.17.0)	Gestion des logs
XercesImpl	2.8.0	Traitement des flux XML
Hibernate	5.6.72.5.29	Persistance en base de données
Struts	2.5.29	Framework MVC
Oracle ojdbc10	19.13.0.0.1	Driver de connexion oracle

## 7 PRÜM EG

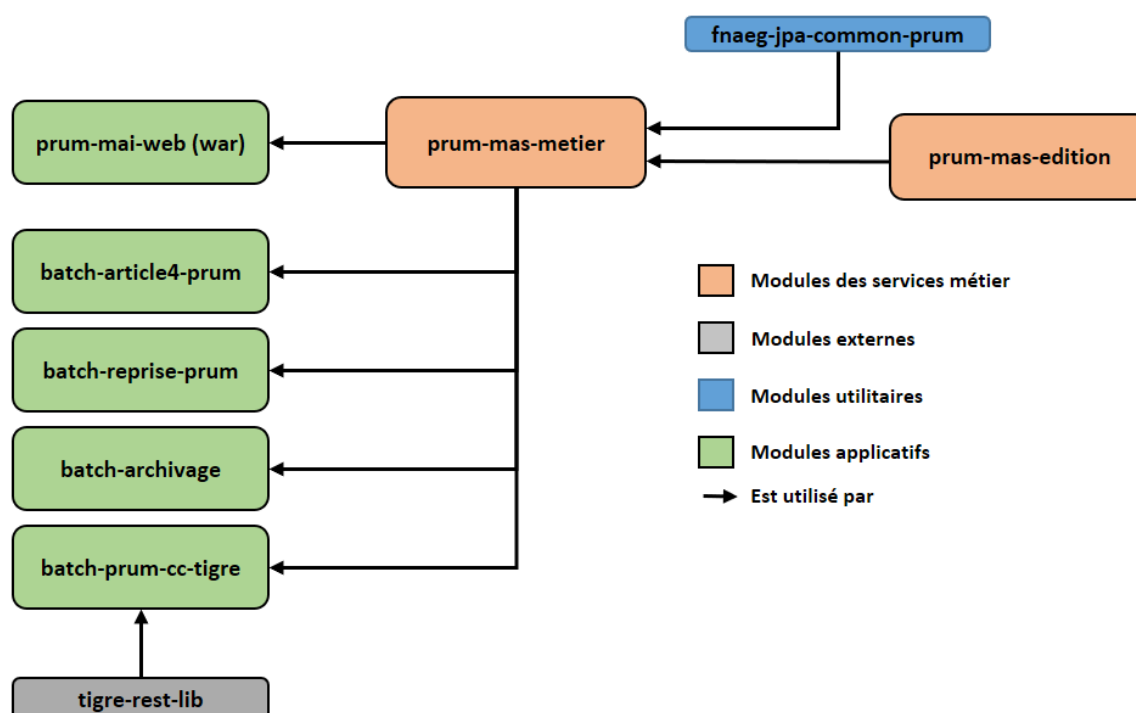
### 7.1 Architecture Logique

#### 7.1.1 Architecture Applicative

##### 7.1.1.1 Conception des modules applicatifs

PRUM			
prum-conf	Génération des scripts (lancement, arrêt, ..)		
prum-db	Scripts sql patch		
prum-mab	batch-parent	batch-prum-cc-tigre	Batch de communication CC/TIGRE/PRUM
		batch-article4-prum	Batch d'importatin des articles 4 PRUM
		batch-reprise-prum	Batch de reprise des données PRUM
		batch-archivage	Batch d'archivage des résultats « no-hits »
pruma-mai	prum-mai-web	Interface Web Prum	
prum-mas	prum-mas-edition	Rapport Jasper	
	prum-mas-metier	Services métiers PRUM	

Les modules appelés lors de l'exécution :



#### **7.1.1.1.1 Module «prum-mai-web »**

Le module « prum-mai-web » implémente le modèle MVC contenant les vues et contrôleurs pour la gestion de l'interaction avec l'application. Il constitue le packaging Web de l'application PRÜM.

#### **7.1.1.1.2 Module « prum-mas-metier »**

Le module « prum-mas-metier » contient les services métier de l'application PRÜM. Ces services sont utilisés par les modules « prum-mai-web » ainsi que les batchs.

#### **7.1.1.1.3 Module « fnaeg-jpa-common-prum »**

Le module « fnaeg-jpa-common-prum » est un module offrant des services génériques de gestion de la persistance des données selon le standard Java JPA.

#### **7.1.1.1.4 Module « prum-mas-edition »**

Le module « prum-mas-edition » contient les templates JasperReports des différents rapports de l'application. Les rapports sont bâtis avec l'outil Jasper Report. Ils ne sont pas stockés sur le serveur mais simplement proposés au téléchargement via l'interface web.

#### **7.1.1.1.5 Module « tigre-rest-lib »**

Le module « tigre-rest-lib » contient les services et objets permettant l'interaction avec les services TIGRE.

### **7.1.1.2 Interfaces**

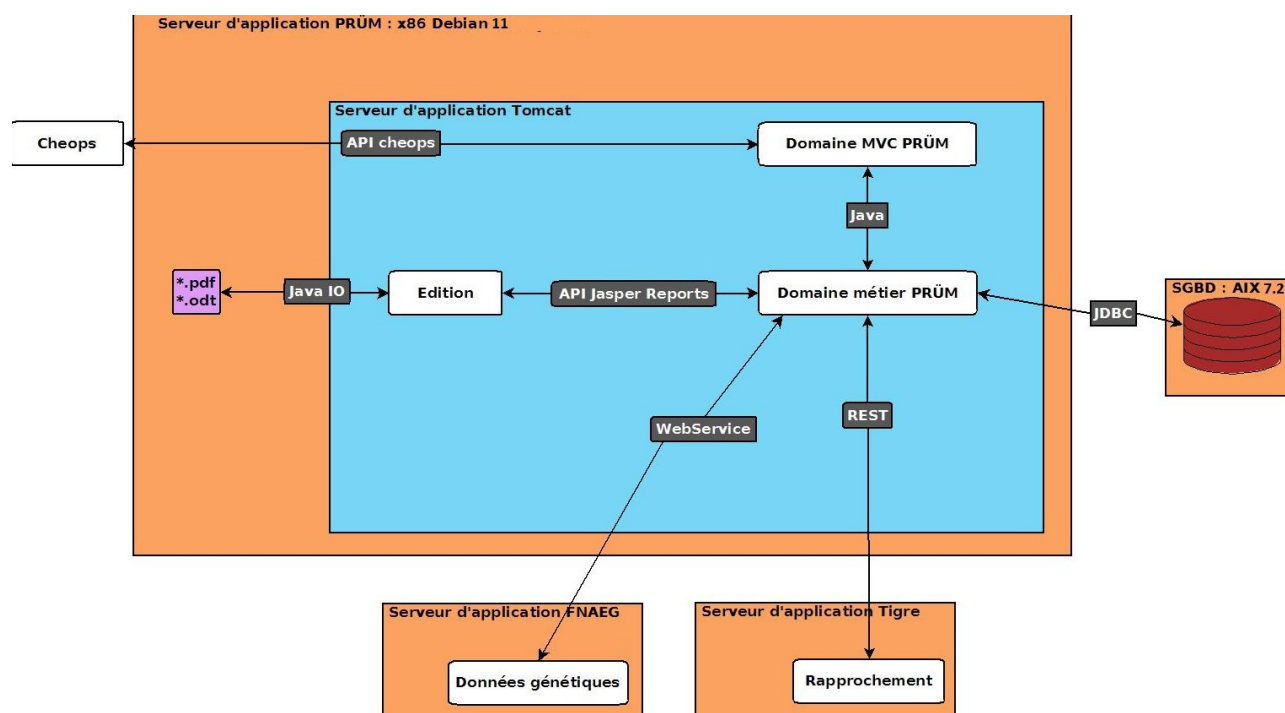
Les interfaces avec les autres systèmes d'information sont au nombre de quatre et se divisent en trois catégories.

- Les exports de base de données :
  - Alimentation de la base du CC
- Les imports de base de données :
  - Alimentation de nouveaux profils par FNAEG
  - Alimentation de nouveaux profils par les questions et réponses des pays membres
- Interactions avec Tigre.

## 7.1.2 Architecture Logicielle

### 7.1.2.1 Schéma d'architecture logicielle

Le schéma ci-dessous présente la communication entre les principaux modules techniques de l'application PRÜM



### 7.1.2.2 Tableau des Principaux Framework Java

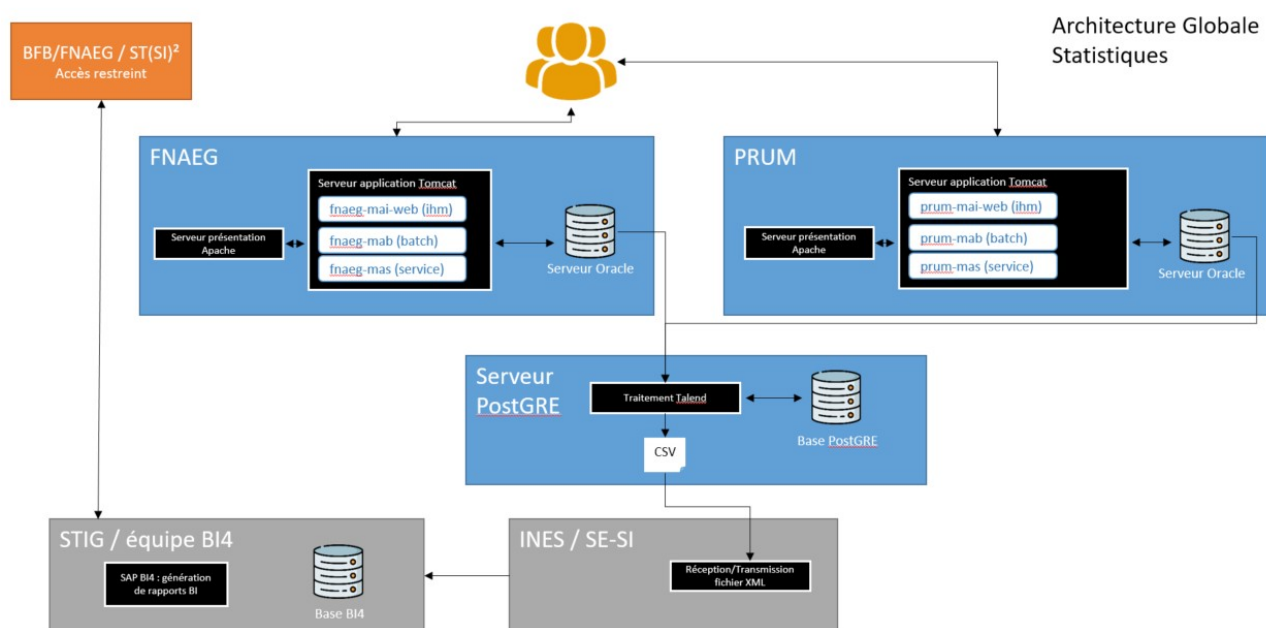
Nom	Version	Description
JasperReport	6.19.0	Génération des rapports
Spring	5.3.18	Framework transverse (Pattern IOC)
slf4j	1.7.36 (base on Log4j 2.17.0)	Gestion des logs
Hibernate	5.6.72.5.29	Persistance en base de données
Struts	2.5.29	Framework MVC
Oracle ojdbc8	19.13.0.0.1	Driver de connexion oracle

**7.1.2.3 Batch applicatifs Java intégrés dans PRUM**

Nom	Description
Archivage	Permet d'archiver les résultats no-hits dans des tables dédiées.
Article4	Lecture du fichier fourni par FNAEG et insertion dans PRUM
PRUM-CC-TIGRE	Permet de transmettre les questions et réponses entre les différents composants (PRUM, CC, TIGRE)

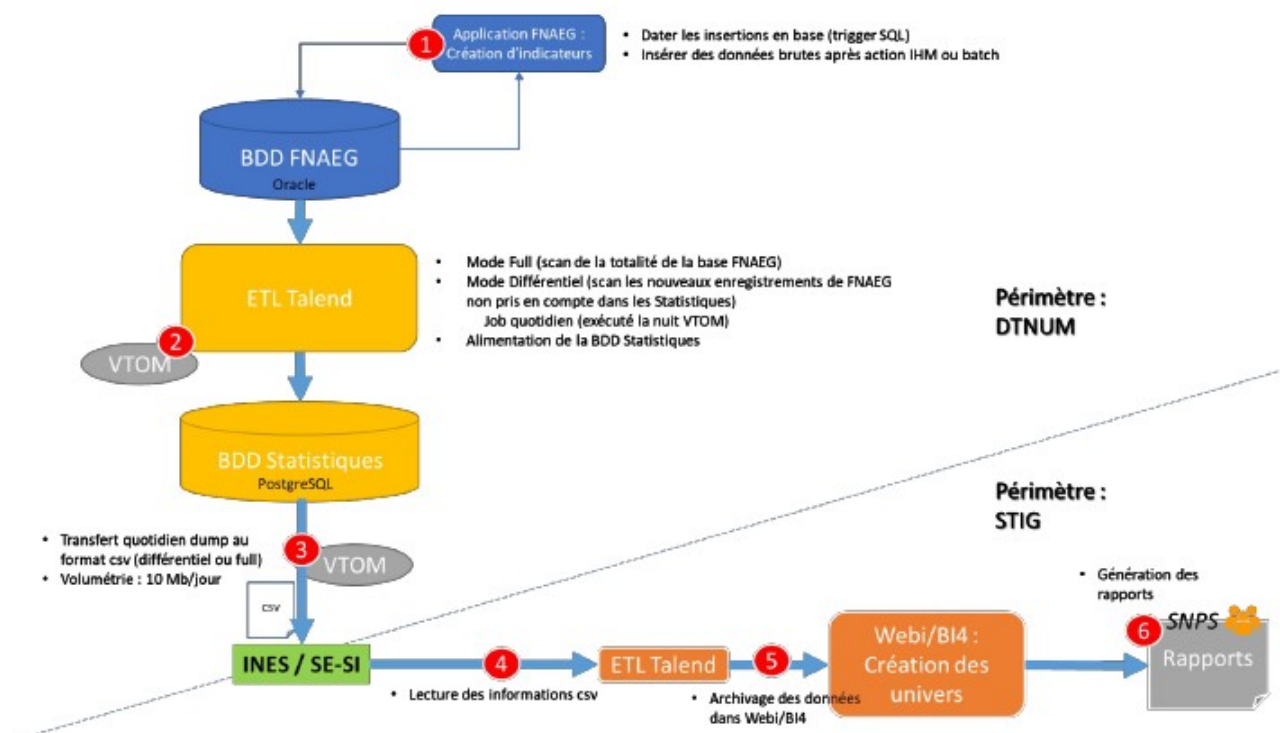
## 8 Statistiques

### 8.1 Architecture Globale

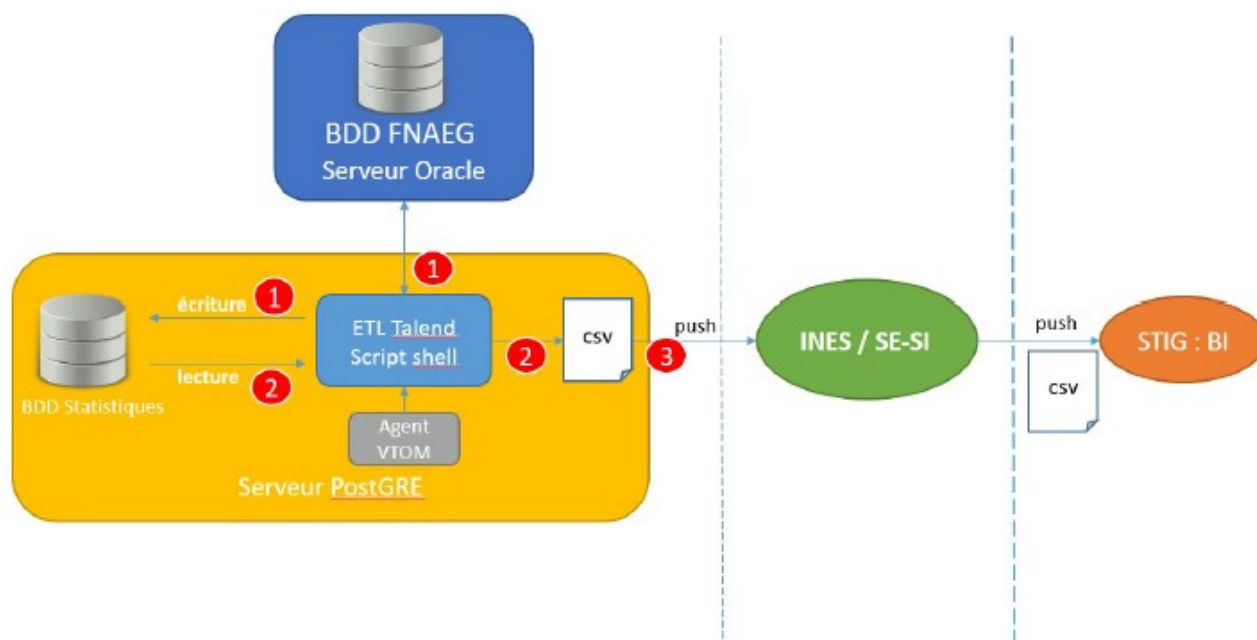


### 8.2 Statistiques FNAEG

### 8.2.1 Architecture applicative générale



### 8.2.2 Enchaînement des traitements





## CCTP FNAEG-NG ANNEXE 2 ARCHITECTURE TECHNIQUE

Ordre	Traitement effectué
1	Un Job Talend sélectionne les enregistrements de FNAEG à stocker dans la base statistiques (les enregistrements sélectionnés varient selon le mode full ou différentiel défini pour les jobs) Les jobs filtrent ensuite ces données pour ne récupérer que les éléments nécessaires à la génération des rapports Statistiques sur BI 4. Les jobs insèrent ensuite les données filtrées dans PostGRE
2	Les Jobs Talend récupèrent les informations de la base Statistique et les inscrivent dans des fichiers au format csv
3	Un script shell envoie les fichiers csv au sein d'une archive tar.gz via sftp au serveur d'application INES

### 8.2.3 Dimensionnement Statistiques FNAEG

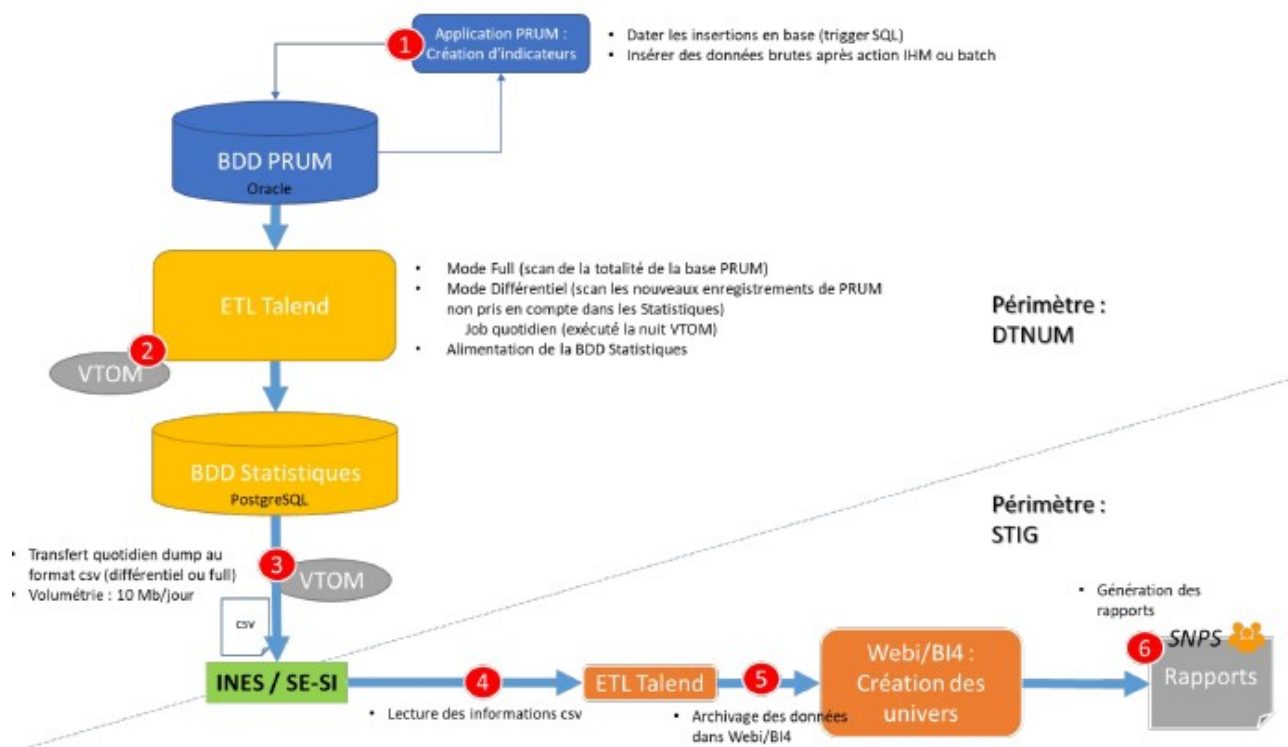
Le dimensionnement physique du serveur applicatif FNAEG ainsi que la base de données Oracle reste inchangé avec l'ajout du module Statistique.

Dimensionnement du serveur PostGRE SQL :

Élément	OS	vCPU	RAM	Disque	Quantité
Serveur Base de données PostGRE SQL 9.6	Linux 64bits (Debian)	4	48 Go	185 Go	1

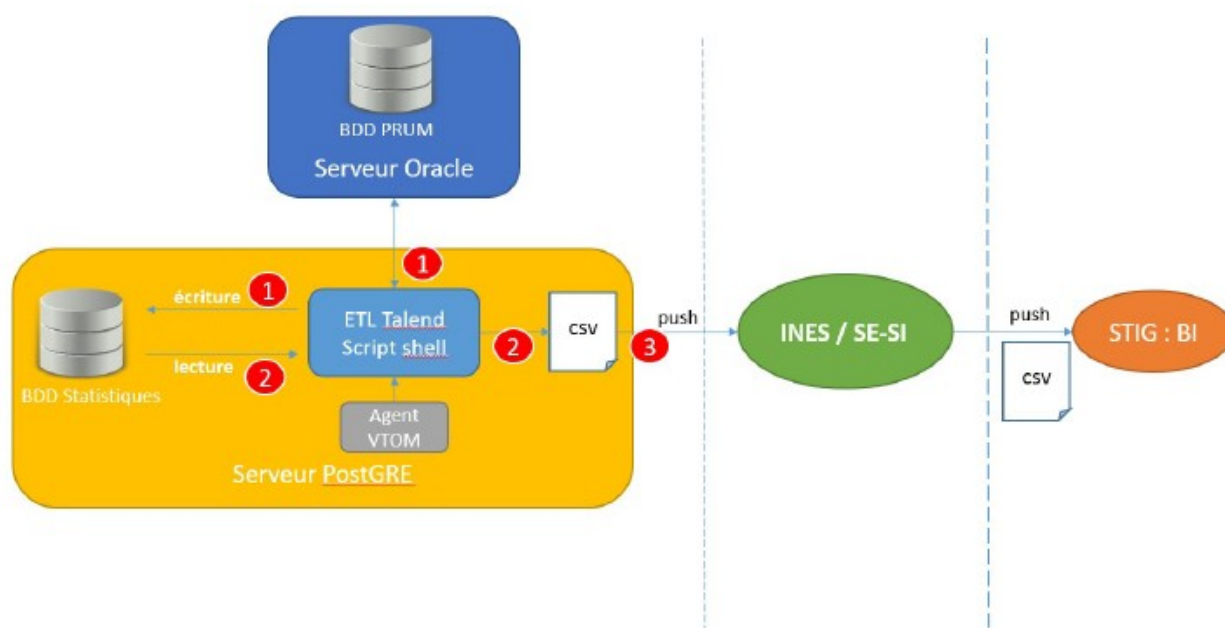
## 8.3 Statistiques PRUM

### 8.3.1 Architecture Applicative Générale



### 8.3.2 Enchaînement des Traitements

## CCTP FNAEG-NG ANNEXE 2 ARCHITECTURE TECHNIQUE



Ordre	Traitement effectué
1	Un Job Talend sélectionne les enregistrements de PRUM à stocker dans la base statistique (les enregistrements sélectionnés varient selon le mode full ou différentiel défini pour le job) Le job filtre ensuite ces données pour ne récupérer que les éléments nécessaires à la génération des rapports Statistiques sur BI 4. Le job insère ensuite les données filtrées dans PostGRE
2	Un Job Talend récupère les informations de la base Statistique et les inscrit dans un fichier au format csv
3	Un script shell envoie le fichier csv via sftp au serveur d'application INES

### 8.3.3 Dimensionnement Statistiques PRUM

Le dimensionnement physique du serveur applicatif PRUM ainsi que la base de données Oracle reste inchangé avec l'ajout du module Statistique.

Dimensionnement du serveur PostGRE SQL :

Élément	OS	vCPU	RAM	Disque	Quantité
Serveur Base de données PostGRE SQL 9.6	Linux 64bits (Debian)	4	48 Go	435 Go	1

## 9 TIGRE

### 9.1 Introduction

Le tableau ci-dessous présente les définitions des termes utilisés dans ce chapitre.

Termes	Définitions
Terminologie du domaine informatique	
Moteur de recherche	Serveur permettant de stocker des données dans un format non relationnel.
Apache Tomcat	Serveur d'application opensource de la fondation Apache.org
Apache Httpd	Serveur web opensource de la fondation Apache.org
Apache SolR	Moteur de recherche opensource de la fondation Apache.org
Lucene	API de recherche opensource de la fondation Apache, cœur du moteur de recherches SolR

### 9.2 Présentation technique Générale de l'Application

#### 9.2.1 Éléments Structurant de l'application

##### 9.2.1.1 *Contrainte Technique*

Les contraintes techniques sont :

Sécurité : l'application n'est pas exposée directement à des utilisateurs, la sécurité sera assurée par le réseau et la configuration des FireWalls.

##### 9.2.1.2 *Une application REST*

L'application étant un moteur dont les interfaces sont exposées pour d'autres systèmes (PRÜM EG ou FNAEG), l'exposition de services REST a été retenue.

L'exposition en webservice aurait été une alternative tout à fait acceptable mais le REST est beaucoup plus souple et permet de s'y interfacer d'une façon plus facile, les contrats d'interfaces n'étant pas nécessaires.

Enfin, une application cliente à base de librairies Javascript peut être interfacée à cette application pour son administration et supervision.

### 9.2.1.3 Un moteur de recherche central

Les bases de données relationnelles permettent la modélisation de tous les cas métiers possibles mais ne sont pas nécessairement les plus performantes. Leurs scalabilités par ailleurs sont les plus souvent limitées à une scalabilité cellulaire (augmentation de la performance du serveur).

Le moteur de recherche quant à lui permet selon les modélisations d'avoir une approche plus pertinente des recherches. De plus, les moteurs de recherche par leurs conceptions sans jointures entre objets sont scalables horizontalement et verticalement.

La complexité de la recherche des séquences ADN réside dans l'autorisation d'une erreur d'analyse. Cette tolérance oblige la base relationnelle à avoir une requête du type :

```
ALLELE1_A in (2.2, 2.1, 2.3, 1.2, 3.2)
AND ALLELE1_B in (5.3, 5.2, 5.X, 4.3, 6.3)
AND ... (x39)
```

Si cette stratégie peut fonctionner pour les recherches simples entre une trace et un individu. Elle devient beaucoup plus laborieuse (au niveau temps de réponse) dans les cas de recherche par parentalité où la moitié des allèles est permise :

```
(ALLELE1_A in (2.2, 2.1, 2.3, 1.2, 3.2) OR ALLELE1_B in (5.3, 5.2, 5.X, 4.3, 6.3))
AND (ALLELE2_A in (2.2, 2.1, 2.3, 1.2, 3.2) OR ALLELE2_B in (5.3, 5.2, 5.X, 4.3,
6.3))
AND ... (x38)
```

Les moteurs de recherche intègrent un calcul de pertinence permettant des recherches du type : « Recherche les individus qui devrait contenir les séquences ... » mais trié par pertinence.

On peut aussi intégrer un seuil : « il doit y avoir au moins 45 résultats corrects »

La recherche simple pourrait être représentée ainsi :

```
{ "bool" : {
  "should" : [ {
    "terms" : { "D1S1656_A" : [ "9" ] }
  }, { "terms" : { "D1S1656_B" : [ "20.3" ] }
  }, { "terms" : { "D2S441_A" : [ "8" ] }
  }, { "terms" : { "D2S441_B" : [ "17" ] } } ],
  "boost" : 1.0,
  "minimum_should_match" : "45"
}}
```

La recherche de parentalité serait possible en modifiant le seuil à 24.

La recherche des traces mélangées serait possible avec la requête :

```
{ "bool" : {
  "should" : [ {
    "terms" : { "D1S1656_A" : [ "9", "11", "13.3", "11.2", "11", "11.1" ] }
  }, { "terms" : { "D1S1656_B" : [ "20.3", "14.2", "20.1", "12.1", "17.3" ] }
  }, { "terms" : { "D2S441_A" : [ "8", "12.1", "12.2", "13.2", "10.3" ] }
  }, { "terms" : { "D2S441_B" : [ "17", "24.1", "18.2", "17.2", "12.2" ] } } ],
```

```
"boost" : 1.0,  
"minimum_should_match" : "20"  
}}
```

Pour plus de détail, le document technique de TIGRE (SFD\_Moteur\_TIGRE) contient les spécificités de ce fonctionnement.

#### **9.2.1.4 Une application non transactionnelle**

L'un des éléments structurant de la solution est donc le moteur de recherche. Seulement, si les moteurs de recherche sont rapides et scalables ils ne sont pas transactionnels.

Le moteur de rapprochement intègre donc une chaîne d'alimentation et un suivi de celle-ci.

Pour des raisons de pertinence, il convient de prioriser l'indexation à la recherche. Ceci afin de maximiser les chances de succès lors d'un rapprochement. En effet, l'absence d'une trace dans la base au moment de la recherche peut être préjudiciable à l'avancée d'une enquête.

#### **9.2.1.5 Le cas d'utilisation de la base de données**

La base de données dans le cadre du FNAEG est utilisée pour stocker les travaux d'indexation et de recherche des séquences.

Les tables utilisées comme files d'attente répliquera le principe de FIFO et permet de prioriser les indexations aux recherches individuelles.

En complément, la base de données pourra contenir les séquences elles-mêmes pour des nécessités de réindexation par exemple ou de simplification de reprise de données.

#### **9.2.1.6 L'interface d'administration**

Le moteur exposant ses services en REST l'interface d'administration minimale a été développée sous le modèle « Application Web Monopage » avec un framework Javascript. Ainsi, l'application JEE est cohérente et n'expose que des services REST.

Les services REST sont consommés par l'application FNAEG, les batchs de PRÜM, l'interface d'administration mais aussi les sondes de supervision.

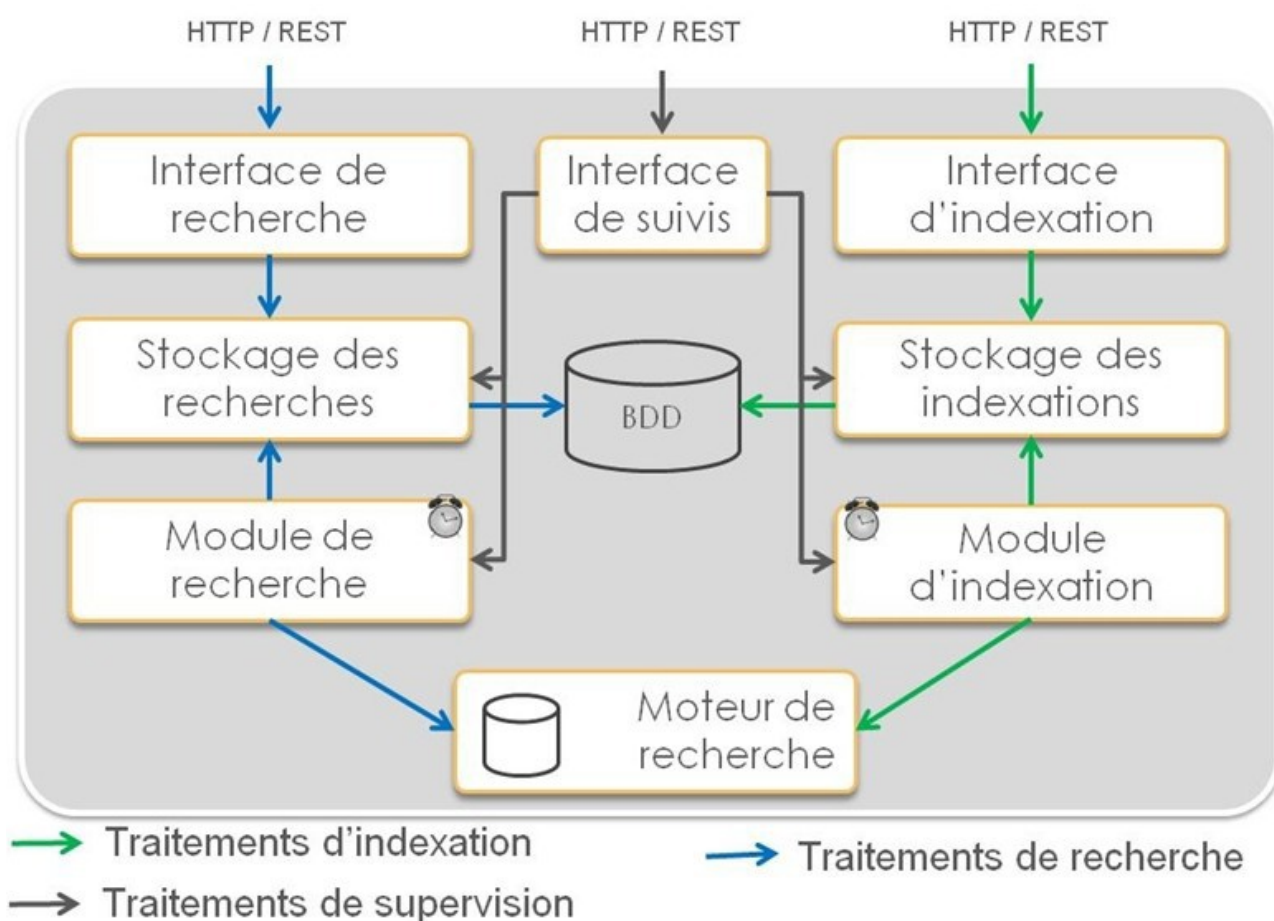
Les frameworks utilisés dans cette application utilisent les composants :

- requirejs,
- jquery,
- backbonejs,
- twitter bootstrap,
- lodash.

## 9.2.2 Architecture de la solution

### 9.2.2.1 Architecture logique

Le schéma suivant présente l'architecture logique de la solution :



Les points essentiels de cette architecture sont :

La gestion asynchrone des demandes et des indexations. Elle permet la priorisation des demandes et surtout éviter l'engorgement de l'application lors des traitements par lots de PRÜM EG.

L'usage d'une base de données pour stocker ces traitements en attente.

L'exposition en REST de l'ensemble des fonctionnalités.

L'interface de suivi sous le modèle « Application Web Monopage » permettant d'avoir une information immédiate sur les traitements en cours.

Des batchs intégrés à l'application continuellement en fonctionnement et vérifiant les files d'attentes de travail.

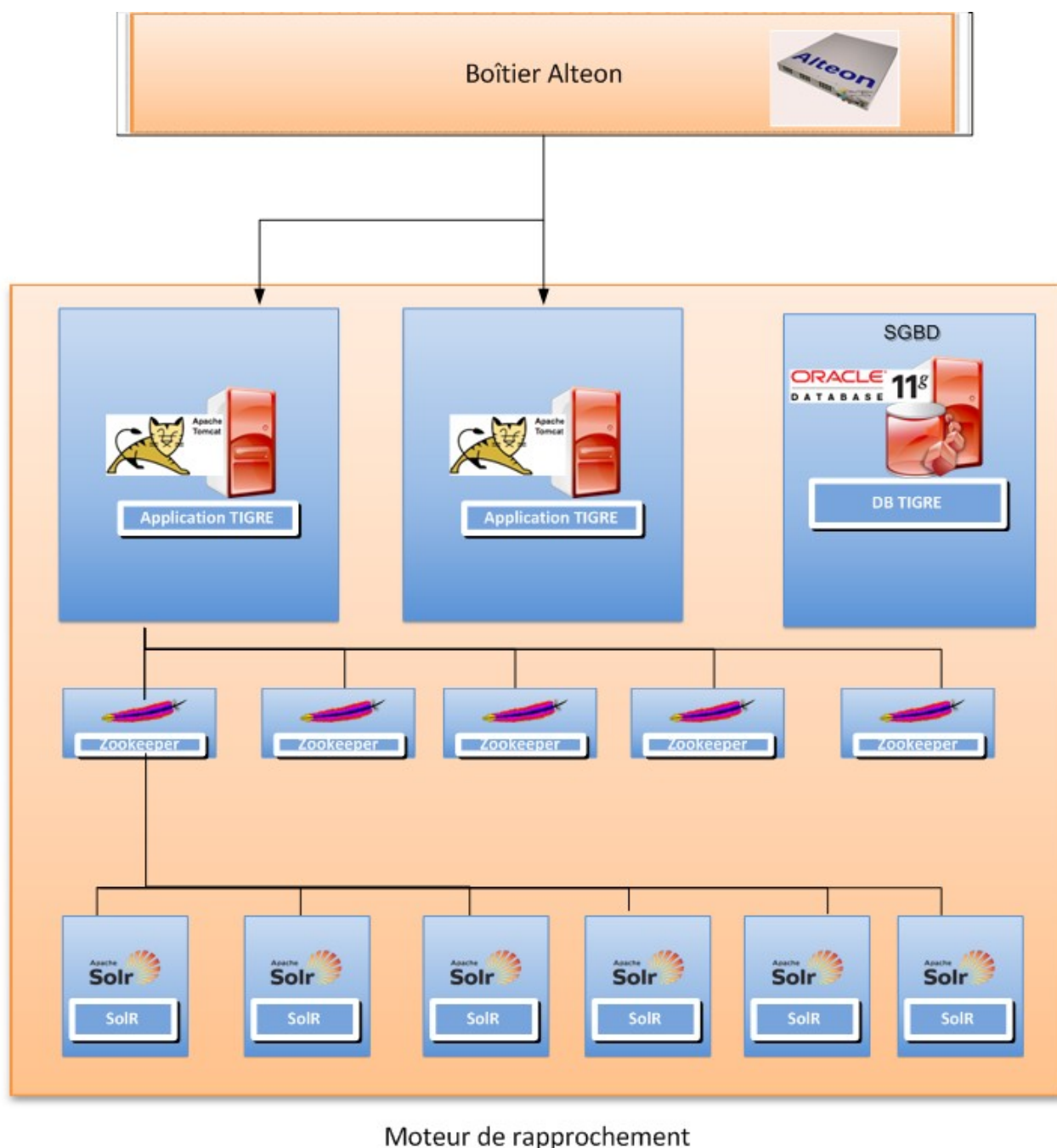
### **9.2.2.2 Architecture physique**

#### **9.2.2.2.1 Architecture physique du moteur de rapprochement**

L'architecture technique est composée de 3 briques :

- le serveur d'application ;
- la base de données ;
- le moteur de recherche.





La plate-forme est accédée à travers d'un répartiteur de charge qui distribue les requêtes entre les instances Tomcat. Les serveurs Web Httpd sont facultatifs étant donné que l'application ne propose pas de page html à des utilisateurs (seulement aux administrateurs).

### 9.2.2.3 Description du flux réseau

Serveurs émetteurs	Serveurs récepteurs	Protocoles	Flux
--------------------	---------------------	------------	------

## CCTP FNAEG-NG ANNEXE 2 ARCHITECTURE TECHNIQUE

Serveurs applicatifs PRÜM EG	Load Balanceur	HTTP/REST	Flux externe au moteur de rapprochement
Serveurs applicatifs FNAEG	Load Balanceur	HTTP/REST	Flux externe au moteur de rapprochement
Load Balanceur Alteon	Serveurs d'applications	HTTP/REST	Flux externe au moteur de rapprochement
Serveurs d'applications	Serveurs de moteurs de recherches	HTTP	Flux applicatif sur moteur de recherche
Serveurs moteurs de recherches	Serveurs moteurs de recherches	HTTP	Flux de synchronisation entre nœuds SolR
Base de données	Serveurs bases de données	SQL	Flux de requêtes base de données

### 9.2.3 Dimensionnement

#### 9.2.3.1 Environnement de Production

Le tableau ci-dessous présente le dimensionnement estimé de l'environnement de production :

Élément	OS	vCPU	RAM	Disque	Quantité
Serveur d'application TIGRE	Linux 64bits (Debian 11)	2	2 Go	50 Go	2
Serveur Moteur de recherche	Linux 64bits (Debian 11)	4	8 Go	70 Go	6
Base de données	AIX 7.2	2	4 Go	200 Go	1

#### 9.2.3.2 Environnement de Développement

Élément	OS	vCPU	RAM	Disque	Quantité
Serveur	Linux 64bits (Debian 11)	4	8 Go	100 Go	1

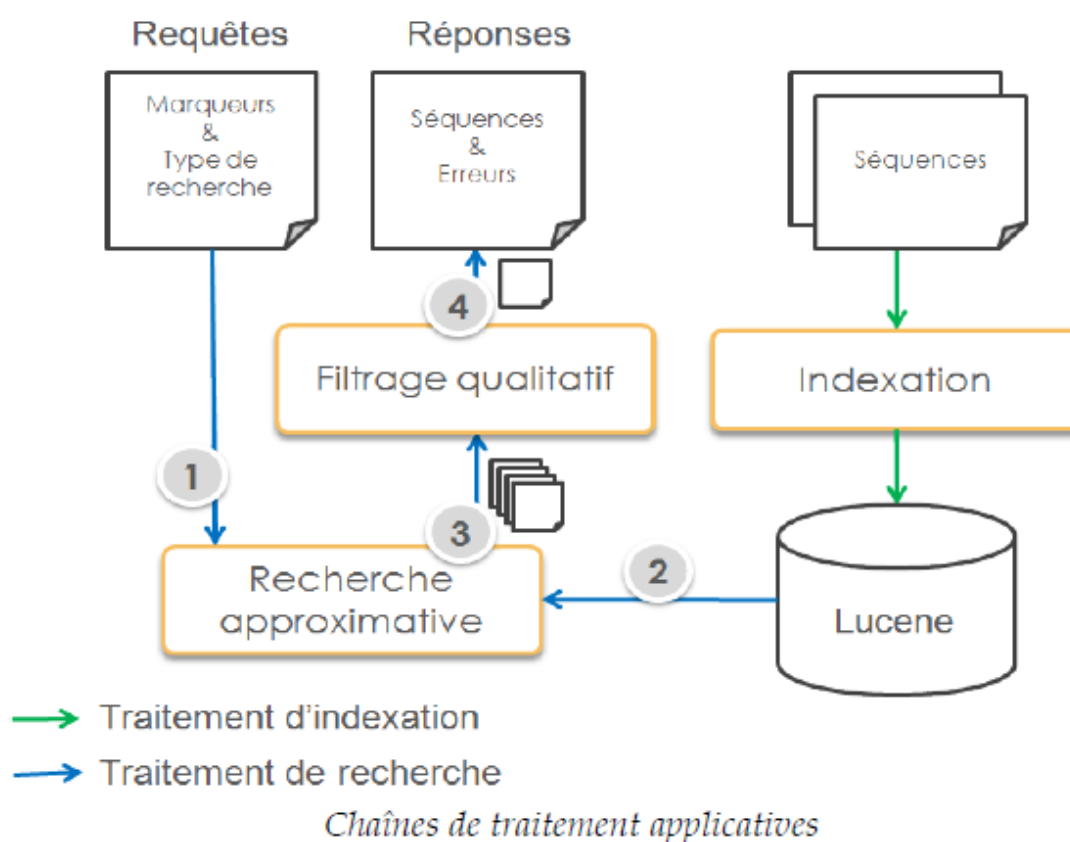
## 9.2.4 Supervision

L'application offre en complément des sondes de supervision habituelles (CPU, Mémoire, Log) des interfaces REST permettant de réaliser des transactions de maintenance ou des contrôles sur le fonctionnement des batchs ou encore les tailles des files d'attentes.

## 9.3 Architecture détaillée

### 9.3.1 Architecture applicative

Le schéma ci-dessous présente les chaînes de traitement pour l'indexation (en vert) et pour la recherche (en bleu)



### 9.3.2 Composants de l'application et version

#### 9.3.2.1 Composant de l'application

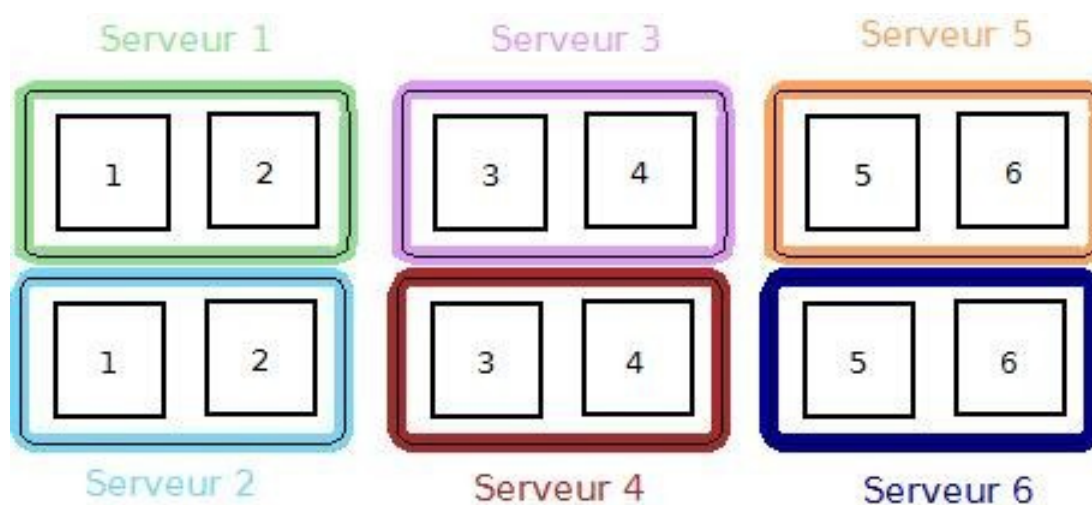
Composant	Version
Framework JAVA	Spring 5.3.18

Tests Unitaires	JUnit.jupiter 5.8.2
Mapper JSON	Jackson-mapper 2.13.2
Calculs statistiques	Yammer.metrics 2.2.0
Compilation	Maven 3.7.0
Framework application d'administration	Backbonejs, requirejs, jquery, lodash
Style application d'administration	Twitter Bootstrap

### 9.3.3 Architecture applicative

#### 9.3.3.1 Moteur de recherche

Les moteurs de recherche permettent la répartition des données sur plusieurs instances. La répartition envisagée est constituée de 6 serveurs, chacun hébergeant 2 noeuds. Les 12 noeuds sont répartis sur 6 shards de 2 réplicats.



*Répartition de l'index sur plusieurs noeuds (6 shards et 2 réplicats)*

#### 9.3.3.2 Serveur d'application

Le serveur d'application nécessite uniquement un serveur servlet. Aussi, le serveur Apache Tomcat apporte pleinement satisfaction.

Le serveur d'application Apache Tomcat est donc retenu dans la solution.

#### 9.3.3.3 L'interface d'administration

## CCTP FNAEG-NG ANNEXE 2 ARCHITECTURE TECHNIQUE

L'usage exclusivement d'exploitation de cette interface permet les simplifications suivantes :

- Pas d'authentification,
- Pas de gestion de droits sur l'application,
- Pas de SSO avec un système externe.

Cette interface est réalisée avec des frameworks javascript tels que les suivants :

- requirejs,
- backbonejs,
- jquery,
- lodash,
- twitter bootstrap (partie CSS).

Cette technologie dit « application web monopage » permet d'utiliser directement les interfaces REST et de soulager le serveur d'application des traitements de page html. En effet, dans le cas des applications web mono page, c'est le navigateur qui travail et qui réalise le rendu des pages html.

## 10 Communication Center

### 10.1 Introduction

#### 10.1.1 Terminologie

Le tableau ci-dessous présente les définitions des termes utilisés dans ce chapitre :

Termes	Définition
<i>Terminologie du domaine informatique</i>	
NIFI	Apache NiFi est une plateforme de logistique de données intégrées en temps réel capable de traiter et d'automatiser les mouvements entre données de différentes sources et systèmes divers.
API	« interface de programmation d'application » est une interface logicielle qui permet de « connecter » un logiciel ou un service à un autre logiciel ou service afin d'échanger des données et des fonctionnalités.
REST	« representational state transfert » est un style d'architecture logicielle définissant un ensemble de contraintes à utiliser pour créer des services web.
XSD	Les fichiers XML Schéma Définition (XSD) permettent de décrire la structure d'un document XML. Le grand intérêt de ce fichier est de servir à la validation du document XML en définissant des règles.
INES	« Infrastructure Nationale d'Échanges Sécurisés » Ce service propose un ensemble de sous-services en lien avec des besoins d'échange d'informations entre applications.

## **10.2 Présentation technique générale de la solution**

### **10.2.1 Les éléments structurants de la situation**

#### ***10.2.1.1 Contrainte techniques***

Les contraintes techniques sont :

- Sécurité : l'application n'est pas exposée directement à des utilisateurs, la sécurité sera assurée par le réseau et la configuration des FireWalls.
- Répondre aux Articles 3 d'un pays demandeur en moins d'un quart d'heure.

#### ***10.2.1.2 Un système de traitement de données NIFI***

Le Communication Center est un serveur de traitement et de transfert de données, qui doit récupérer des fichiers, les traiter et les archiver. C'est une tâche redondante qui doit être faite sur un rythme soutenu et rapide.

NIFI est un système basé sur la configuration de tâche répétitif simple et robuste d'utilisation. Cela nous permet aussi de décorréliser le traitement des données (qui sera modifié ou mis à jour très rarement) de son utilisation par une application tiers (API REST qui est vouée à évoluer plus souvent) et donc d'avoir une modularité plus accrue (mettre à jour, redémarrer, déplacer... sans tout arrêter).

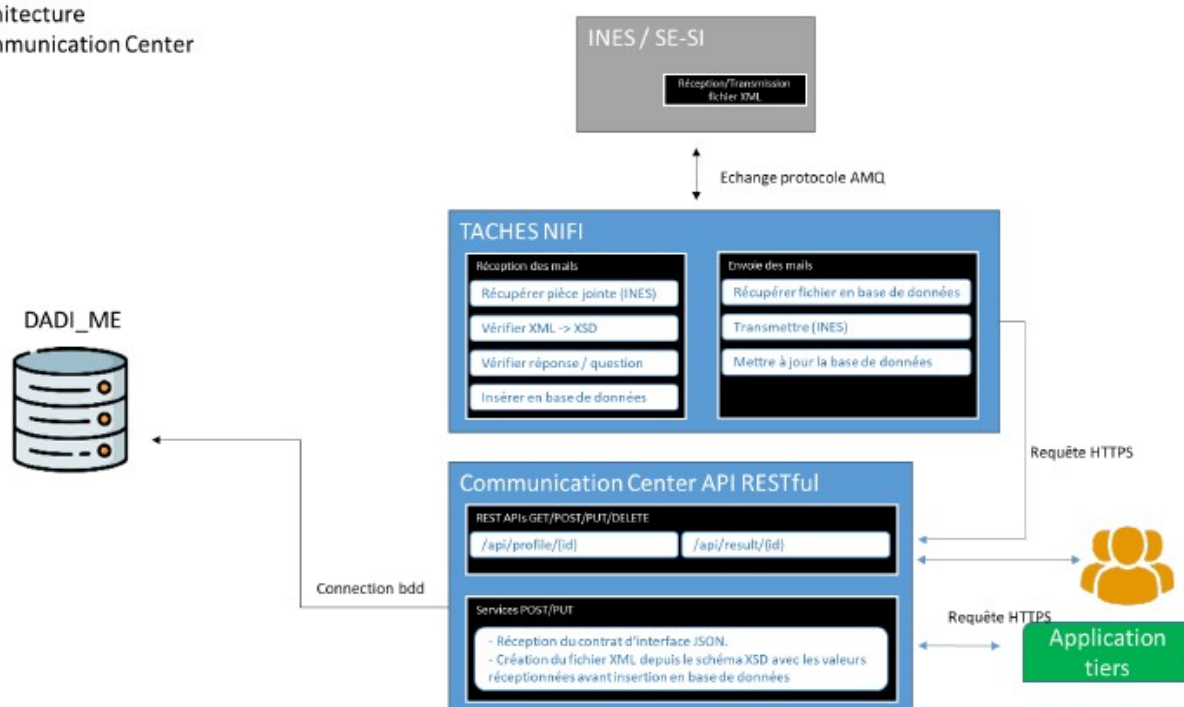
#### ***10.2.1.3 Une application API REST***

Le Communication Center, n'a pas vocation à servir directement une interface utilisateur. Néanmoins il peut y avoir un intérêt à interagir et manipuler ses données. Pour cela une API REST est mis à disposition pour alimenter une interface décorrélée pour recueillir ou peupler les données du Communication Center.

### **10.2.2 Architecture Globale**

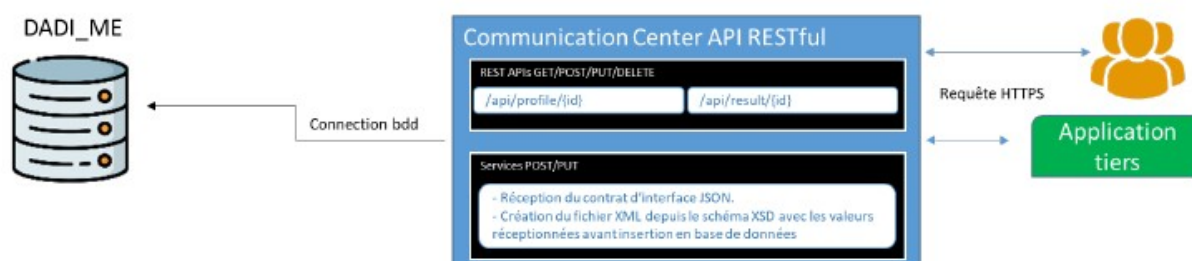
## CCTP FNAEG-NG ANNEXE 2 ARCHITECTURE TECHNIQUE

Architecture  
Communication Center



### 10.3 Architecture Détaillée

#### 10.3.1 Architecture Communication Center API RESTfull





### 10.3.2 Composant Communication Center API RESTfull

Composant	Version
Frameworks JAVA	Spring boot 3.2.1
Tests Unitaires	Spring restdocs 3.0.1
Mapper XML	Jakarta 4.0.1
Compilation	Maven 3.8.1

#### 10.3.2.1 End-points exposés

Les end-points sont des adresses url qui permettent d'interagir avec les données du Communication Center via le protocole HTTPS.

C'est de cette façon que nous pouvons récupérer des données à afficher ou à manipuler.

Ces end-points sont construits sur le schéma de la base de données et peuvent avoir une terminaison variable et des paramètres de requête

exemple: <https://nomdedomaine.fr/api/profile/{id}?param1=1&param2=2>

Les end-points principaux en POST/GET/PUT/DELETE sont :

- /api/profile/{id} Tous les « → profile » ou un profile id = 1 (question(s)) de la table « profile »
- /api/result/{id} Tous les « → result » ou un result id = 2 (réponse(s)) de la table « result »

**Remarque** : les end-points cités ci-dessus sont des points de départ. Les API REST sont amenées à évoluer suivant les besoins des consommateurs.

exemple de recherche des questions appartenant à une référence de requête :

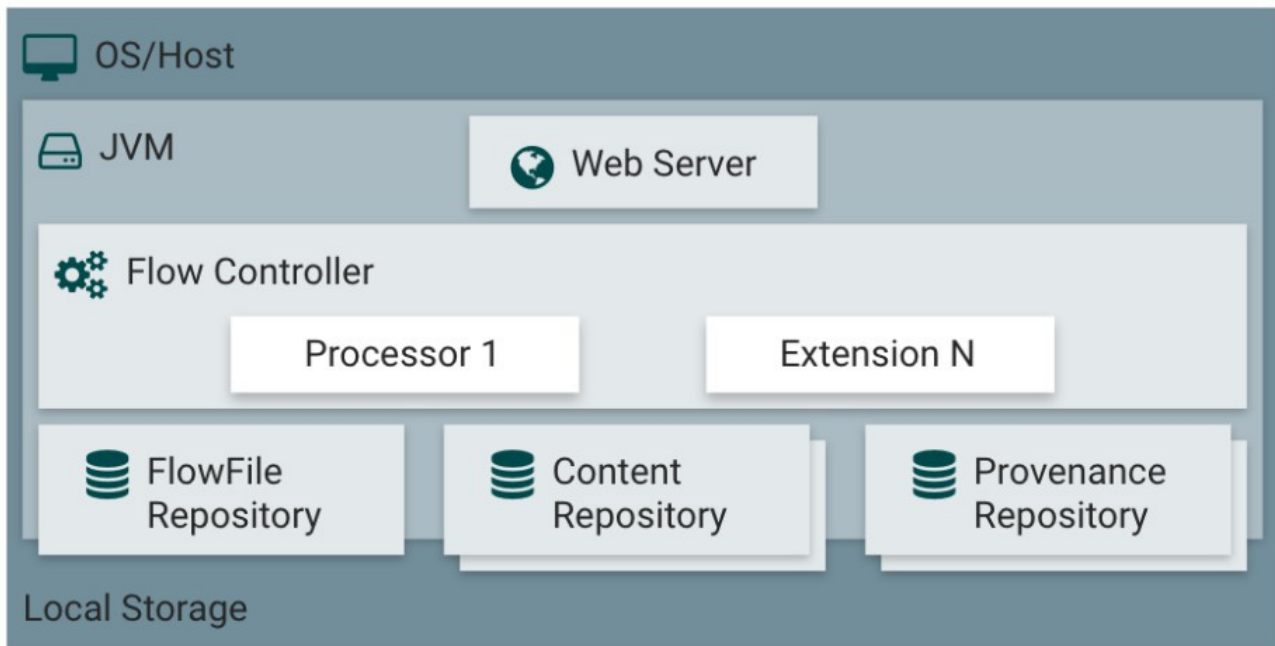
/api/profile/groupid/{id}.

#### 10.3.2.2 Service XML

Ce service permet la construction des fichiers d'échange XML qui seront sauveés en base afin que NIFI puisse les traiter.

Le fichier XML est construit sur la base d'un fichier XSD, alimenté par les données d'un contrat d'interface JSON, transmis par une application tiers.

### 10.3.3 Architecture NIFI



#### 10.3.3.1 Installation NIFI

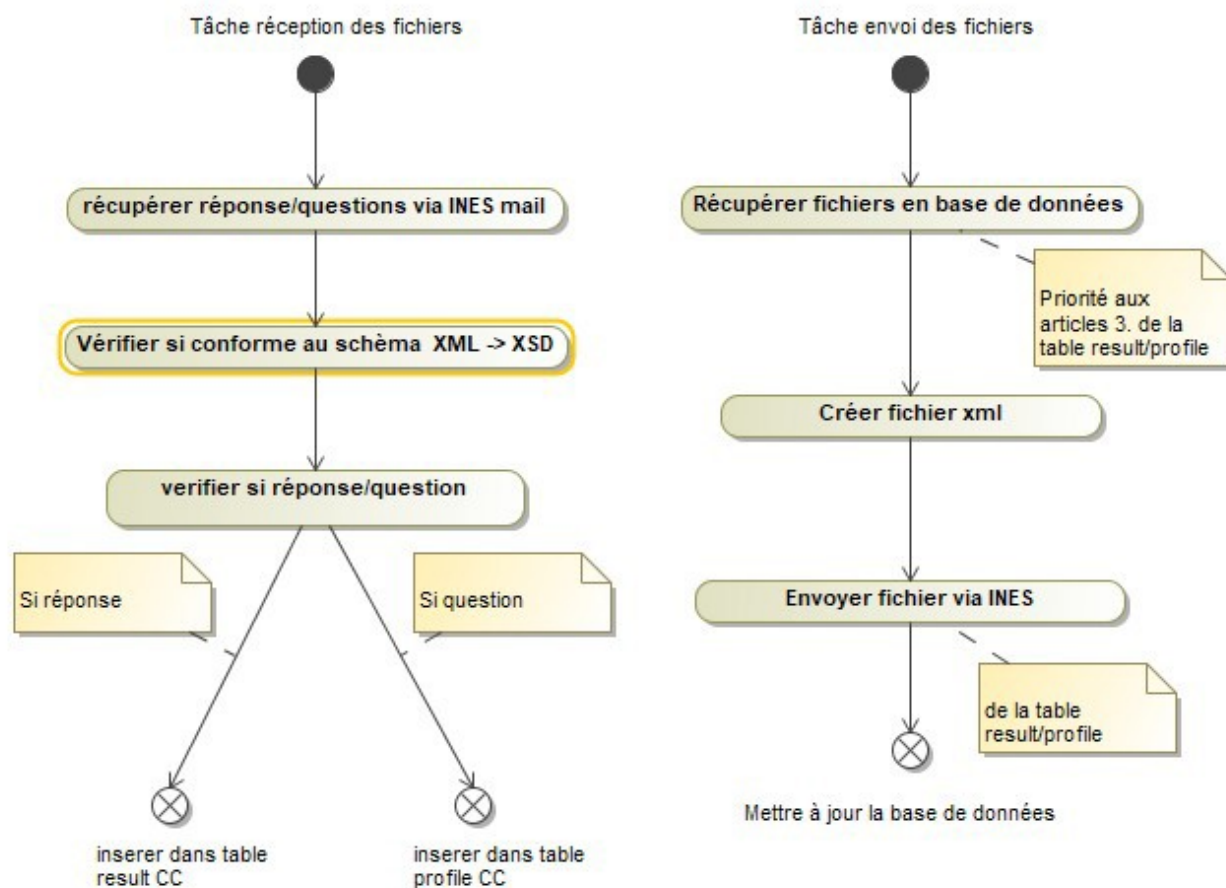
NIFI est une application JAVA multi système lancée dans une JVM (Java Virtual Machine).

[Documentation for 1.25.0 - Apache NiFi](#)

#### 10.3.3.2 Interface de configuration

NIFI est un système de tâches configurable, et propose un serveur web qui délivre une Interface web pour la configuration des tâches et processus de chaque tâche (Il existe aussi un système de commande pour paramétrer les tâches et processus).

### 10.3.4 Tâches et Processus Communication Center NIFI



## 11 Complexité de l'application FNAEG-NG et indicateurs technique

### 11.1 Le code

L'application FNAEG-NG comporte pas moins de 200 000 lignes de code, ventilées comme suit :

FNAEG : 186 397 lignes,

TIGRE : 14 000 lignes,

Prüm : 23 908 lignes,

Les batch (dont ceux des statistiques) : 600 lignes.

Le code utilise que des standards « Java » connus et maîtrisés par tous.

Plusieurs « framework », améliorant l'industrialisation et la centralisation du code, sont utilisés (JasperReports, Spring, Log4j, Xmlbeans, XercesImpl, Hibernate, Struts et Oracle ojdbc10 et 8).

De plus dans le cadre des statistiques, l'outil « Talend Open Studio for Data Integration » est utilisé et génère automatiquement du code « Java » (lancé par des scripts Shell).

## **11.2 FNAEG**

### **11.2.1 Analyse base de données**

#### ***11.2.1.1 Structure***

La base de données contient :

- + 150 tables ;
- 0 procédure stockée ;
- 25 déclencheurs (trigger).

#### ***11.2.1.2 Volumétrie des tables (Avril 2025)***

Le tableau suivant présente les plus grosses tables en terme de volumétrie de manière décroissante.

La volumétrie est pilotée par le nombre d'empreintes génétiques.

Sur un total de plus de 150 tables, il y a

5 tables à plus de 100 millions de lignes ;

20 tables entre 1 à 41 millions de lignes ;

Ces 25 tables représente 99 % de la volumétrie totale de la base de données.

CCTP FNAEG-NG ANNEXE 2 ARCHITECTURE TECHNIQUE

Nom des tables	Occurrences
VALEUR_PARAMETRE	565 047 663
PROFIL_GENETIQUE	307 269 538
RAPPROCHEMENT_INDIVIDU_ALLELE	250 059 417
ACTION	170 442 833
RAPPROCHEMENT_INDIVIDU_MARQUEUR	122 392 362

Activité FNAEG

Ci dessous le nombre de connexion sur l’application sur 1 mois.

