**DOSSIER D’ARCHITECTURE TECHNIQUE**

**DAT nomappli**

Table des matières

[1. Contexte et architecture globale 3](#_Toc186797172)

[1.1. Contexte 3](#_Toc186797173)

[1.2. Architecture globale 3](#_Toc186797174)

[2. Architecture logique 3](#_Toc186797175)

[2.1. Couches applicatives 3](#_Toc186797176)

[2.1.1. Couche présentation 3](#_Toc186797177)

[2.1.2. Couche coordination 3](#_Toc186797178)

[2.1.3. Couche application 3](#_Toc186797179)

[2.2. Fichiers et clés de configuration 3](#_Toc186797180)

[2.3. Liste des langages et composants logiciels utilisés 3](#_Toc186797181)

[3. Architecture technique 4](#_Toc186797182)

[3.1. Dimensionnement et volumétrie 4](#_Toc186797183)

[3.1.1. Volumétrie des données 4](#_Toc186797184)

[3.1.2. Typologie des utilisateurs 4](#_Toc186797185)

[3.1.3. Dimensionnement des ressources 4](#_Toc186797186)

[3.2. Schéma d’architecture technique 4](#_Toc186797187)

[3.3. Matrice de flux 5](#_Toc186797188)

[3.4. Description des inferfaces externes 5](#_Toc186797189)

[3.4.1. Envoi de mail 5](#_Toc186797190)

[3.4.2. Stockage S3 5](#_Toc186797191)

[3.4.3. Appel API xxx 5](#_Toc186797192)

[3.5. Environnements d’exécution 6](#_Toc186797193)

[3.5.1. staging 6](#_Toc186797194)

[3.5.2. preproduction 6](#_Toc186797195)

[3.5.3. production 6](#_Toc186797196)

# 1. Contexte et architecture globale

## 1.1. Contexte

Description de l’application et à quoi elle sert

## 1.2. Architecture globale

Décrire rapidement les différents composants applicatifs avec un schéma fonctionnel

# 2. Architecture logique

## 2.1. Couches applicatives

Description des différentes couches logicielles (présentation, coordination, application, persistance, modèle, sécurité, services techniques divers, gestions des logs, authentification, ..) en précisant les patterns d’architecture principaux (MVC, ..)

### 2.1.1. Couche présentation

### 2.1.2. Couche coordination

### 2.1.3. Couche application

…

## 2.2. Fichiers et clés de configuration

Décrire l’emplacement des fichiers de configuration et les clés de configuration principales présentent dans ces fichiers (il n’est pas nécessaire de détailler dans ce document toutes les clés de configuration mais chaque clé de configuration devra être commentée au sein de chaque fichier pour savoir à quoi elle sert)

## 2.3. Liste des langages et composants logiciels utilisés

Lister les principaux langages, frameworks et librairies utilisées en précisant l’utilité et le type de licence open source (toute utilisation de librairie non open source doit être explicitement approuvée par l’IRSN)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Brique logicielle** | **Version** | **Site internet** | **Usage** | **Licence** |
| **python** | 3.9 | python.org | Langage de l’application |  |
| **python flask** | 2.2 | flask.palletsprojects.com | Framework web | Licence BSD |
| **bootstrap** | 5.3 | getbootstrap.com | Framework css/html | Licence MIT |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 3. Architecture technique

## 3.1. Dimensionnement et volumétrie

### 3.1.1. Volumétrie des données

Indiquer la taille et le dimensionnement des fichiers de données (nombre de fichiers, volumétrie en Go approximative) ou bases de données (nombre de lignes de tables les plus importantes, nombre de tables approximatives), ainsi que les perspectives de croissances annuelles.

### 3.1.2. Typologie des utilisateurs

Lister les différents profils utilisateurs, leur nombre et leur usage

|  |  |  |  |
| --- | --- | --- | --- |
| **Profil** | **Fonctionnalités accédées** | **Nombre** | **Période / fréquence d’utilisation** |
| Utilisateur non authentifié | Page d’accueil | Plusieurs dizaines de connexions simultannées possibles | Le site doit fonctionner en heures ouvrées |
| Utilisateur authentifié | Application | ~ 10 | Heures ouvrées, outil de travail utilisé quotidiennement |
| Administrateur | Application + Fonctionnalités d’aministration | ~ 4 | Heures ouvrées, utilisation ponctuelle |
|  |  |  |  |

### 3.1.3. Dimensionnement des ressources

Lister les besoins en termes de ressources pour les différents composants

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Environnement (production, preproduction, ..)** | **Composant** | **RAM (Go)** | **CPU (nombre)** | **Disque (Go ou To)** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## 3.2. Schéma d’architecture technique

Réaliser un schéma en précisant les flux réseaux avec les URL et port et les protocoles. Fournir le fichier source utilisé pour réaliser le schéma (visio, drawio)

Pour les interfaces externes indiquer les url de production

Exemple de schéma réalisé avec draw.io.

Une image contenant texte, capture d’écran, diagramme, nombre

Description générée automatiquement

## 3.3. Matrice de flux

Inclure tous les flux applicatifs y compris les flux externes

|  |  |  |  |
| --- | --- | --- | --- |
| **Composant source** | **Composant destination** | **Protocole / port** | **Description de flux (à quoi il sert)** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## 3.4. Description des inferfaces externes

Décrire toutes les interfaces applicatives avec des services externes

### 3.4.1. Envoi de mail

### 3.4.2. Stockage S3

### 3.4.3. Appel API xxx

## 3.5. Environnements d’exécution

Décrire les différents environnements, à quoi il servent, les données utilisées et les URL externes concernées (les accès au site et également les interfaces externes) par l’application et le cas échéant pour les interfaces externes. Décrire les mécanismes éventuels de restriction sur les environnements de test ou de preproduction (white list IP).

### 3.5.1. staging

L’application est accessible via l’URL xxx.irsn.fr (whitelist IP pour l’IRSN et les développeurs)

Les données sont des données fictives

### 3.5.2. preproduction

L’application est accessible via l’URL xxx.preprod.ul2i.fr (whitelist IP pour l’IRSN et les développeurs)

Les données sont des données réelles copiées de l’environnement de production

### 3.5.3. production