



**IT LINK**  
Accélérateur d'innovation

Réf : OPVIGIL - DAT v3.00

Edité le : 14/11/2024

# OPVigil :

## Document d'architecture technique



A l'attention de :

Equipe projet SCHAPI

Rédigé par :

Benjamin Leclerc, Ingénieur d'études, IT link

Tom Spreters, Ingénieur d'études, IT link

Approuvé par :

Emmanuel CAMUS, Responsable pôle d'expertise SI/Web, IT link

Préambule

Ce document constitue l'architecture technique de la plateforme « OPVigil ».

## Enregistrement des évolutions

Version	Date	Auteur	§ : description
1.00	16/02/2021	B. LECLERC T. SPRETERS	Version initiale
1.01	01/03/2021	B. LECLERC T. SPRETERS	Mise à jour suite à l'atelier technique n°1 : schémas d'architecture, détails des services, modes hors ligne et authentification Cerbère.
1.02	05/03/2021	B. LECLERC T. SPRETERS	Mise à jour suite à l'atelier technique n°2 : schémas d'architecture, détails des services, modes hors ligne et authentification Cerbère.
1.03	23/03/2021	B. LECLERC T. SPRETERS	Prise en compte des retours du SCHAPI et des échanges lors de l'atelier technique n°3.
1.04	02/04/2021	Y. AUFFRET	Prise en compte des retours du SCHAPI après nouvelle itération (compléments Luc MULLER-RIPALDA)
1.05	08/04/2021	Y. AUFFRET	Corrections pour validation SCHAPI du document
1.06	28/09/2021	Y. AUFFRET	Mise à jour de la version de MapServer
1.07	25/02/2022	T. SPRETERS	Mise à jour des versions des applications suite à la montée de version des VM sur Debian 11. Mise à jour pour la version RELEASE1
2.00	16/02/2023	T. SPRETERS	Mise à jour pour la V2 d'OPVigil : Schéma d'architecture globale, description des flux de publication, archivage et nettoyage des données, ajout de la VM d'archivage Prise en compte des remarques lors de l'atelier de conception V2 n°3
2.01	17/04/2023	T. SPRETERS	Mise à jour pour intégrer le composant « API historique »
3.00	14/11/2024	Y. AUFFRET	Mise à jour pour la V3 d'OPVigil : MapServer sur la VM d'archivage, utilisation de la VM Historique pour le calcul des indicateurs et nouvelle bibliothèque de graphiques ApexCharts

## Table des matières

<b>1. Description des exigences attendues .....</b>	<b>5</b>
1.1. Contexte et enjeux .....	5
1.2. Intégration dans l'infrastructure SCHAPI.....	5
1.3. Haute disponibilité.....	6
1.4. Utilisation en mode hors ligne .....	6
1.5. Authentification Cerbère.....	6
<b>2. Architecture.....</b>	<b>7</b>
2.1. Chaîne de production <b>antérieure</b> .....	7
2.2. Vue globale de l'architecture OPVigil .....	9
2.3. Description des services .....	9
2.4. Description des flux.....	11
2.5. Infrastructure technique .....	16
2.6. Fonctionnements en mode dégradé .....	17
<b>3. Composants logiciels.....</b>	<b>19</b>
3.1. Système d'exploitation .....	19
3.2. Liste des applicatifs.....	19
3.3. Frameworks et bibliothèques logicielles .....	20

# 1. Description des exigences attendues

## 1.1. Contexte et enjeux

Le SCHAPI établit et diffuse, en coordination avec les services de prévision des crues (SPC), une information continue de vigilance « crues », publiée sur le site « Vigicrues » et par mails. Il intervient en appui des 19 SPC et de la CVH de Guyane. À ce titre, il exerce une mission d'organisation, d'animation, d'assistance, de conseil et de formation auprès des services et des établissements intervenant dans le domaine de la prévision de crues, et plus généralement, de l'hydrologie et de l'hydrométrie.

Les outils actuels de production de la vigilance, Vigilance.exe et ProdVigil, rencontrent des limites et seront remplacés par une application web : OPVigil.

Les objectifs du projet OPVigil sont :

- Faciliter les conditions de production de la vigilance ;
- Permettre la diffusion de la vigilance ;
- Garantir la continuité du service.

Ce document décrit l'architecture d'OPVigil mise en place pour répondre à ces objectifs.

## 1.2. Intégration dans l'infrastructure SCHAPI

OPVigil s'intègre dans un environnement technique existant et doit donc être déployé en respectant les contraintes de cette plateforme.

L'ensemble des composants OPVigil sont installés **sur deux machines virtuelles (VM)** afin de simplifier le déploiement et la gestion des environnements.

Cette VM sera basée sur une VM de référence fournie par le SCHAPI déjà utilisée par les autres applicatifs déployés sur la même plateforme de virtualisation.

OPVigil sera déployé dans plusieurs environnements. Les environnements identifiés sont :

- Opérationnel
- Intégration (alimentant Vigicrues Intégration)
- Test (alimentant Vigicrues Test)
- Formation (alimentant une instance de Vigicrues à définir)

De manière générale, OPVigil sera construit de façon à déployer un environnement opérationnel et autant d'environnements non-opérationnels que nécessaires aux besoins du SCHAPI. Ces deux modes comportent des différences visuelles pour les distinguer. Les interfaces sont paramétrées dans la configuration des instances.

**Les VM OPVigil sont identiques** selon les environnements (opérationnel ou non) et une configuration permet de lancer l'application dans un de ces modes.

### 1.3. Haute disponibilité

La haute disponibilité de l'application est gérée au niveau des VM par VMWare dans l'infrastructure de virtualisation du SCHAPI. Il n'est aujourd'hui pas nécessaire de prévoir de mécanisme HA supplémentaire dans OPVigil.

### 1.4. Utilisation en mode hors ligne

Le SCHAPI et les SPC doivent être en capacité de produire la vigilance crue quelle que soit la situation, 24h/24h, 7j/7, y compris lorsque les accès réseau sont indisponibles.

Pour répondre à ce besoin, l'application OPVigil est développée avec un mode hors ligne mettant en œuvre les Service Workers et en utilisant les composants navigateur et fournis par Angular. Les données et services associés à certaines fonctionnalités de production locale sont mis en cache dans le navigateur et permettent de continuer à utiliser l'application sans connexion au backend PHP Symfony. En mode dégradé, les prévisionnistes peuvent donc rédiger leurs bulletins sans connexion au RIE et les transmettre au SCHAPI pour intégration.

Dans le cas où le cache du navigateur est absent et donc ce mode dégradé est indisponible, l'application OPVigil peut être téléchargée sous forme d'archive. Ce fonctionnement est décrit dans le paragraphe 2.6.4.

### 1.5. Authentification Cerbère

La connexion des utilisateurs OPVigil est faite par le portail d'authentification du ministère : Cerbère.

Cerbère définit la liste des utilisateurs ayant accès à OPVigil. Les droits précis de ces utilisateurs sont définis dans OPVigil.

En cas d'indisponibilité de Cerbère l'application doit proposer un mode d'authentification alternatif tout en restant connecté au backend OPVigil. Il doit permettre de maintenir la gestion des droits utilisateurs et la production de la vigilance. Ce fonctionnement est décrit dans le paragraphe 2.6.1.

## 2. Architecture

### 2.1. Chaîne de production **antérieure**

#### 2.1.1. Production en mode normal

Etapas de la production :

- Production dans Vigilance.exe en utilisant les référentiels en local
- Dépôt de la production sur un FTP RIE (et sur le réseau Météo-France pour le SPC ME)
- Ingestion de la production par ProdVigil
- Cas nominal (1) :
  - o Production nationale dans ProdVigil (cartes, bulletins et PDF associés)
  - o Publication Météo France
  - o Exécution du script de publication : Vigicrues, mails et historisation
- Cas des CVH ou actualisations sans changement de couleur (2) :
  - o Exécution du script de publication : Vigicrues, mails et historisation

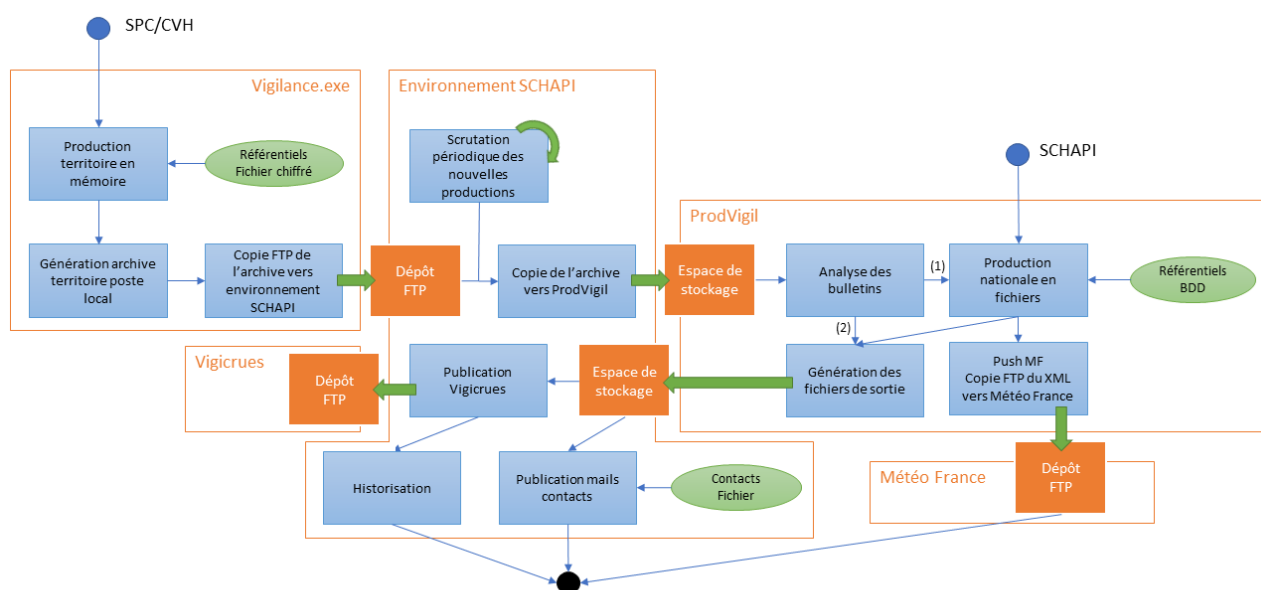


Figure 1 : Chaîne de production entre Vigilance.exe et ProdVigil

### 2.1.2. Production en mode dégradé : défaut du FTP entre SPC et SCHAPI

Dans ce mode dégradé, une étape supplémentaire est ajoutée pour le transfert par mail de la production de Vigilancia.exe vers ProdVigil. Le reste de la chaîne est identique.

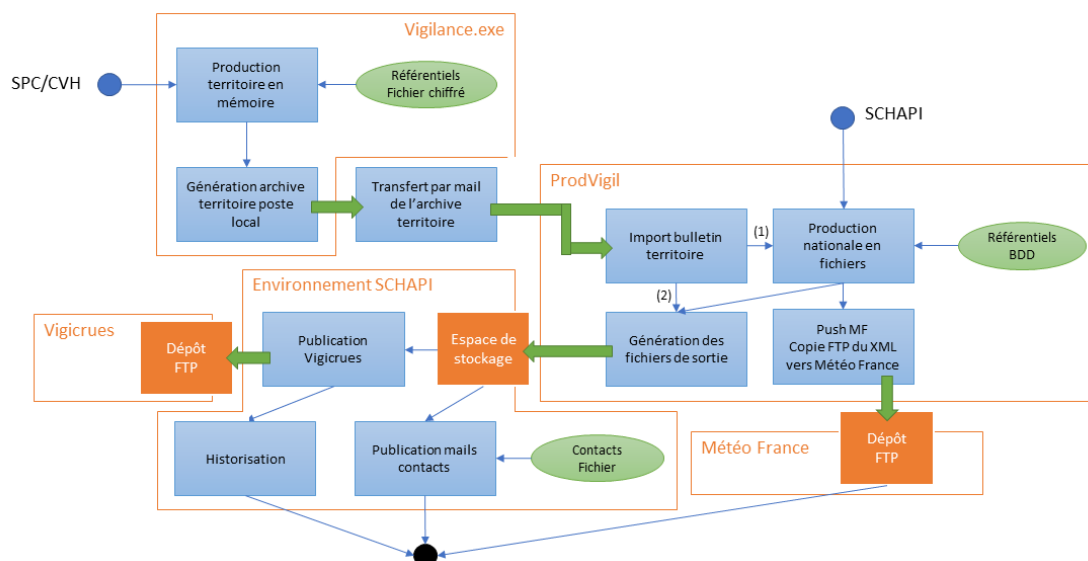


Figure 2 : Chaîne de production en cas de coupure du FTP entre SPC et SCHAPI

### 2.1.3. Production en mode dégradé : défaut de la connexion entre ProdVigil et Vigicrues

En cas de coupure de la connexion RIE entre ProdVigil et Vigicrues, la production est déposée sous forme d'archive au format tar.gz sur le système de ticketing de l'hébergeur de Vigicrues.

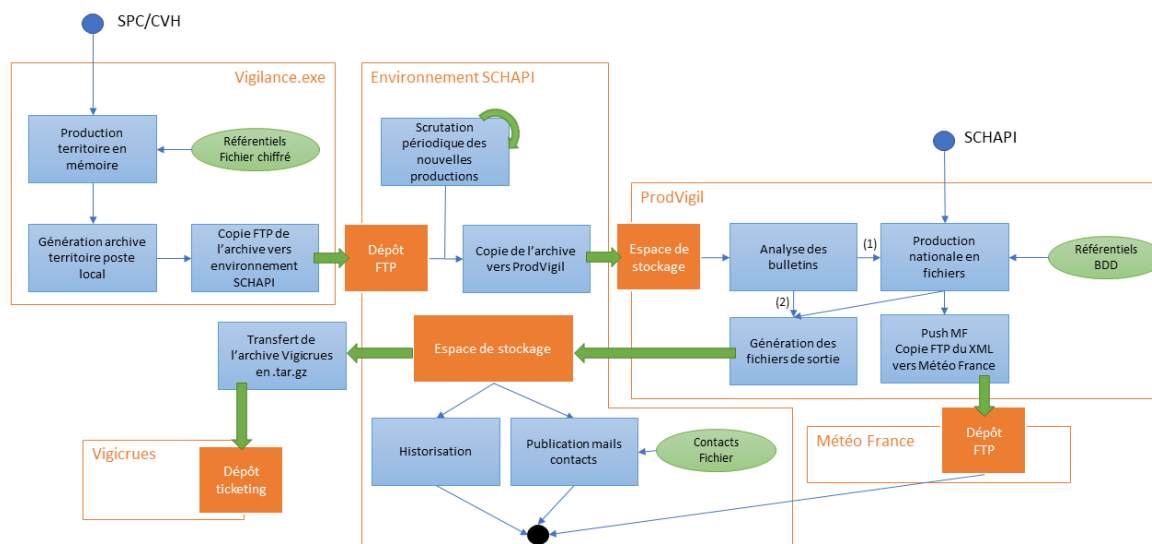


Figure 3 : Chaîne de production en cas de coupure du RIE entre ProdVigil et Vigicrues



## 2.2. Vue globale de l'architecture OPVigil

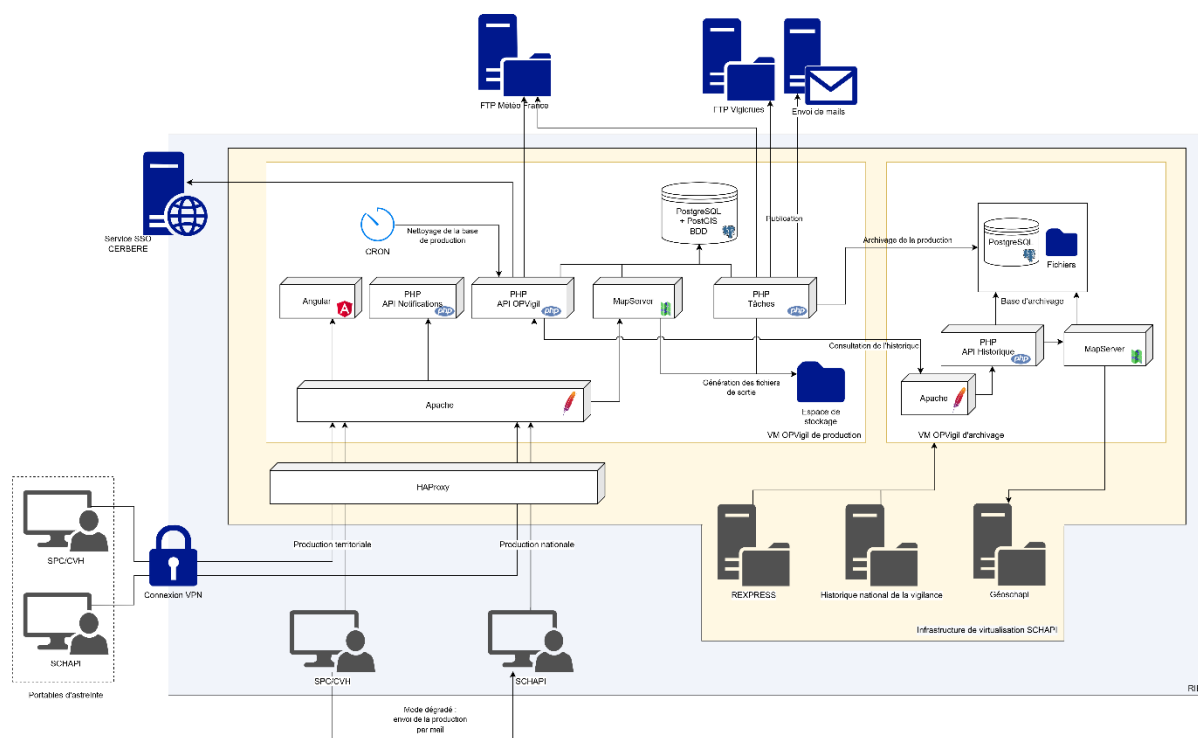


Figure 4 : Schéma de l'architecture globale OPVigil

## 2.3. Description des services

### 2.3.1. HAProxy

HAProxy est le reverse proxy en amont de la VM OPVigil. Il est géré par le SCHAPI et est commun avec les autres applications déployées dans son infrastructure de virtualisation.

Les URL d'accès aux différents environnements seront préfixées par le FQDN du proxy et du nom de l'environnement :

- <https://services.schapi.e2.rie.gouv.fr/opvigil/> pour l'environnement opérationnel ;
- <https://services.schapi.e2.rie.gouv.fr/opvigil-integration/> pour l'environnement d'intégration ;
- <https://services.schapi.e2.rie.gouv.fr/opvigil-test/> pour l'environnement de test ;
- <https://services.schapi.e2.rie.gouv.fr/opvigil-formation/> pour l'environnement de formation.

### 2.3.2. Apache

Apache est le serveur HTTP en amont des autres services. Il distribue les fichiers statiques du frontend Angular et exécute le code PHP des APIs.

### 2.3.3. Frontend

Le frontend Angular est l'application s'exécutant dans les navigateurs des utilisateurs. Elle est développée en utilisant les services workers et la mise en cache pour pouvoir fonctionner sans connexion aux services backend.

### 2.3.4. API

L'API REST OPVigil est le backend PHP de l'application. Il gère notamment l'authentification, les droits utilisateurs, la production territoriale et nationale ainsi que la production des fichiers de sorties.

### 2.3.5. API Notifications

L'API de Notifications, également en PHP, est le service gérant les flux par WebSocket dans OPVigil. Ces flux sont les communications de type PUSH depuis le serveur vers les navigateurs des utilisateurs comme la messagerie entre prévisionnistes et les notifications pour forcer la sauvegarde et reprendre la main sur un bulletin.

### 2.3.6. API Historique

L'API Historique, également en PHP, est installée sur la VM OPVigil d'archivage. Elle permet de déporter les calculs liés à la consultation de l'historique **et aux indicateurs** sur une autre VM, afin de ne pas perturber la production. Toutes les communications entre le frontend Angular et l'API Historiques passent par l'API REST OPVigil (§2.3.4), qui reste responsable de la vérification des droits et rôles utilisateurs.

### 2.3.7. Service de tâches – génération et publication

Le service de tâches est un module PHP en charge des tâches de fond OPVigil : génération des areamaps, génération des fichiers de production, publication des productions, archivage des productions, nettoyage de la base de production ...

### 2.3.8. MapServer

MapServer est utilisé pour la génération des cartes et area maps de vigilances. Ce service utilise les fichiers de configuration et templates pour leur génération.

### 2.3.9. Base de données

La base de données PostgreSQL contient les données des comptes utilisateurs, droits, référentiels et bulletins territoriaux et nationaux.

### 2.3.10. Espace de stockage

Cet espace de stockage est utilisé pour l'alimentation des scripts de publications avec les fichiers contacts, cartes générées par MapServer et autres fichiers générés par la production de la vigilance.

### 2.3.11. Scripts de publications

Les scripts de publications sont chargés de l'envoi des productions le FTP Vigicrues, l'envoi des mails ainsi que l'archivage de la production. Dans la version 1 d'OPVigil, ils sont repris depuis ProdVigil.

Les protocoles FTP et SFTP doivent être pris en charge pour la version 2.

## 2.4. Description des flux

### 2.4.1. Accès utilisateur depuis le RIE

Pour accéder à OPVigil, le cas nominal est depuis un poste connecté un RIE. L'utilisateur accède à l'application à travers la HAProxy Schapi et Apache qui sert les fichiers Angular.

L'application Angular est alors dans le navigateur et communique avec l'API Rest PHP.

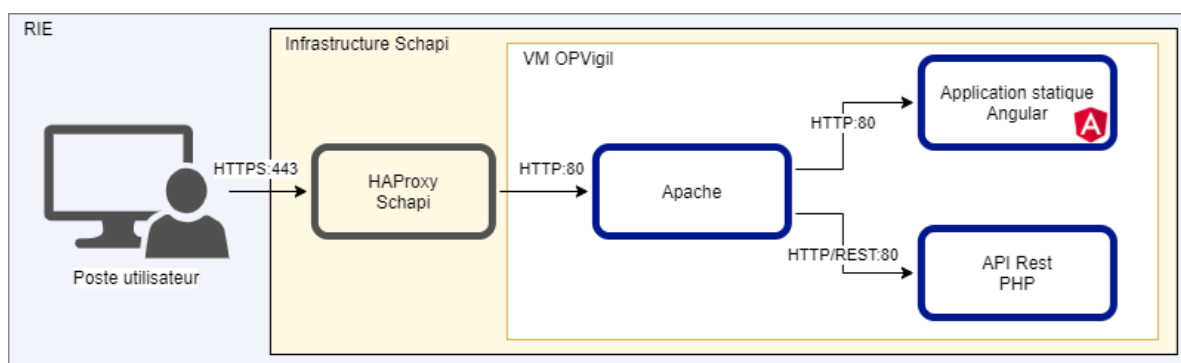


Figure 5 : Flux - Accès utilisateur depuis le RIE

### 2.4.2. Accès utilisateur en connexion VPN (mode nomade)

Si le poste utilisateur n'est pas dans le RIE, il peut s'y connecter en utilisant son VPN.

Pour d'OPVigil cette configuration est identique à une connexion directe depuis le RIE et le flux de connexion est identique avec l'ajout du VPN en intermédiaire.

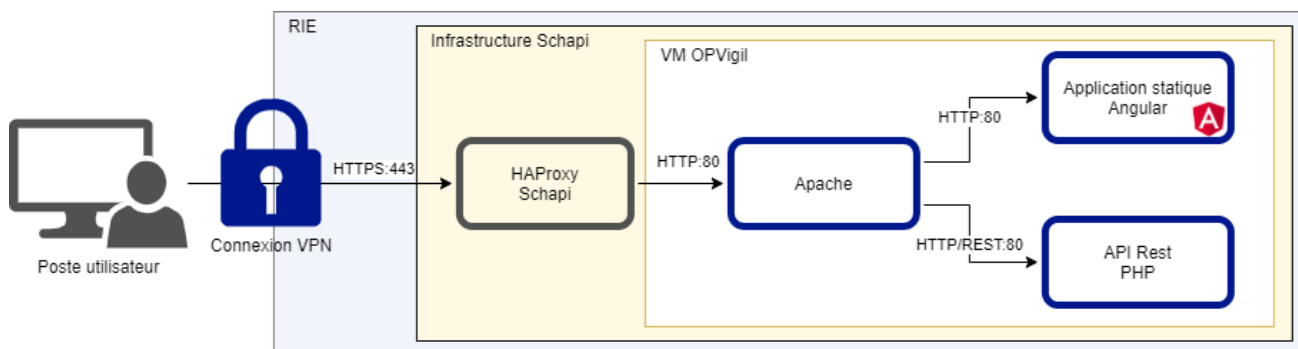


Figure 6 : Flux - Accès utilisateur en connexion VPN

### 2.4.3. Connexion SSO avec Cerbère

Cerbère est utilisé comme service d'authentification CAS. L'application Angular redirige l'utilisateur sur Cerbère où il s'authentifie puis le backend PHP OPVigil vérifie la validation du token auprès de Cerbère.

En cas d'indisponibilité de Cerbère, la connexion se fait directement avec le backend PHP.

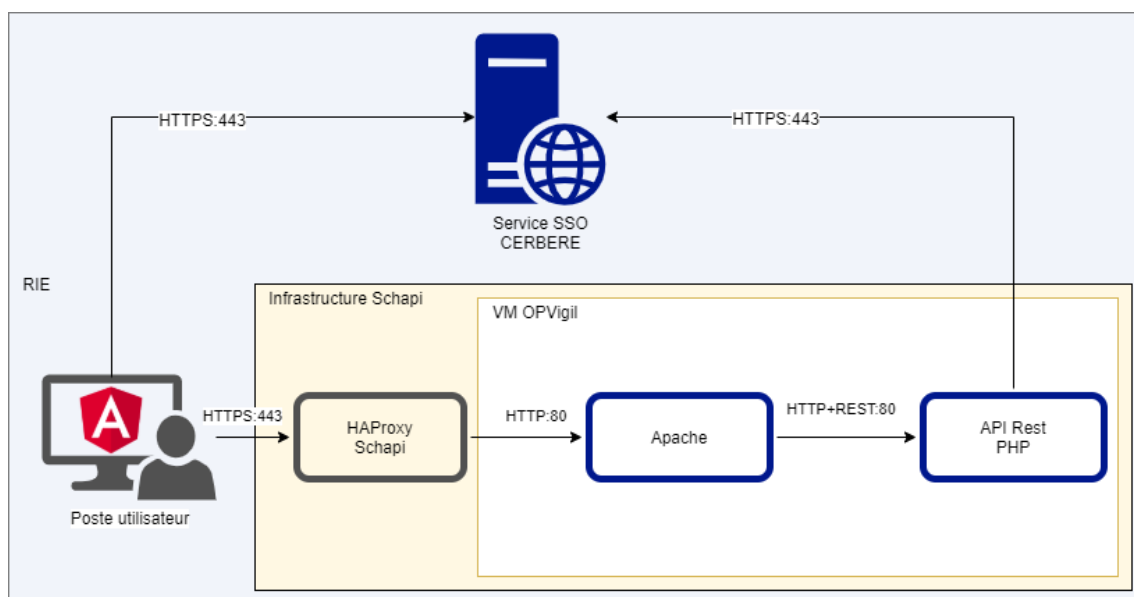


Figure 7 : Flux - Connexion SSO avec Cerbère

#### 2.4.4. Messagerie instantanée et notifications PUSH

Un autre type de flux est présent entre le backend et l'application Angular. Basé sur le protocole WebSocket il permet l'échange de données de type PUSH (du backend vers le navigateur). Ces notifications sont produites par un service PHP distinct. Apache est en charge de la répartition du trafic entre les deux services PHP.

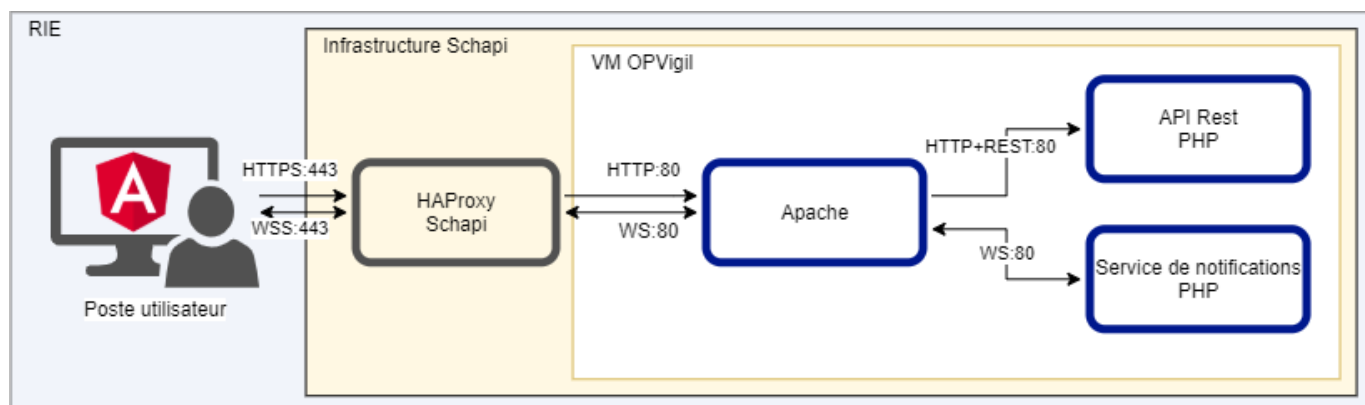


Figure 8 : Flux - Messagerie et notifications PUSH

#### 2.4.5. Production territoriale

Le prévisionniste territorial crée des bulletins qui sont sauvegardés dans la base de données PostgreSQL. Les fichiers PDF uploadés (documents externes ou générés associées aux tronçons dans le bulletin) sont sauvegardés dans l'espace de stockage. Ces données sont gérées dans le backend par l'API REST PHP.

La vigilance est produite en utilisant les référentiels présents dans la base de données.

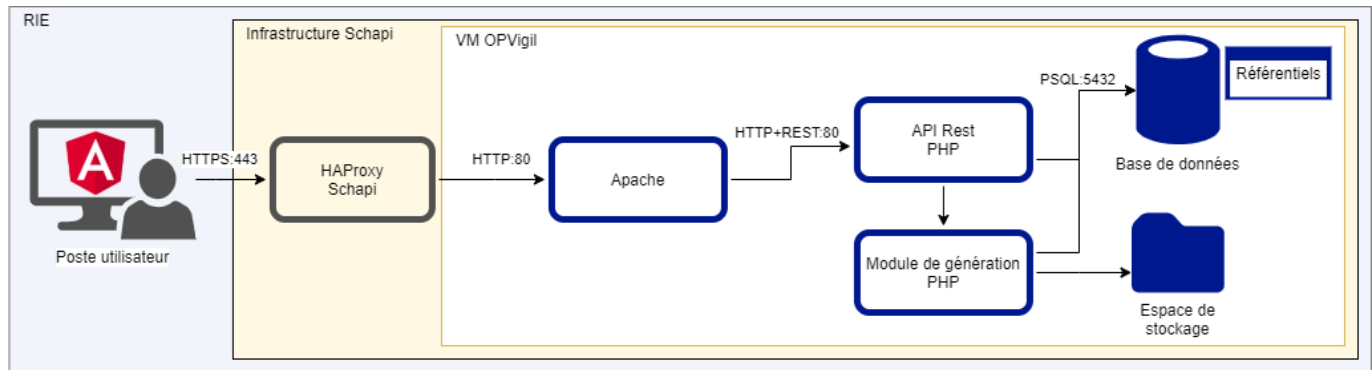


Figure 9 : Flux - Production territoriale

#### 2.4.6. Production territoriale en mode dégradé (cache disponible)

L'application web OPVigil passe en mode dégradé lorsque le poste utilisateur n'est plus connecté au RIE et donc lorsque le backend PHP Symfony n'est pas accessible.

Dans ce mode, la vigilance est produite en local en utilisant les référentiels mis en cache dans le navigateur. Elle est ensuite exportée au format ZIP, transmise par email et réintégrée dans la chaîne de production depuis un poste connecté au RIE.

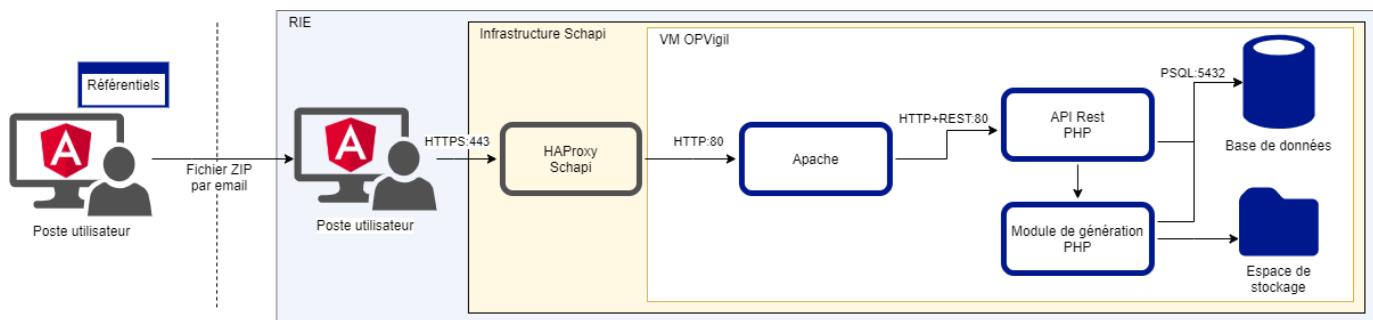


Figure 10 : Flux - Production territoriale en mode dégradé

#### 2.4.7. Production territoriale en mode dégradé (sans cache)

Si le cache du navigateur ne contient pas l'application OPVigil, le prévisionniste territorial peut utiliser l'archive de secours pour produire la vigilance.

Cette archive de secours contient une version portable d'OPVigil et des référentiels (détails du fonctionnement en 2.6.4). Elle est générée à partir des fichiers ressources inclus dans le code source de l'application et des référentiels présents en base de données.

En cas de défaut du cache du navigateur et d'indisponibilité du RIE, l'archive de secours est transmise au prévisionniste territorial qui peut alors produire avec cette version locale d'OPVigil. La production est ensuite transmise par email comme dans le mode dégradé avec cache.

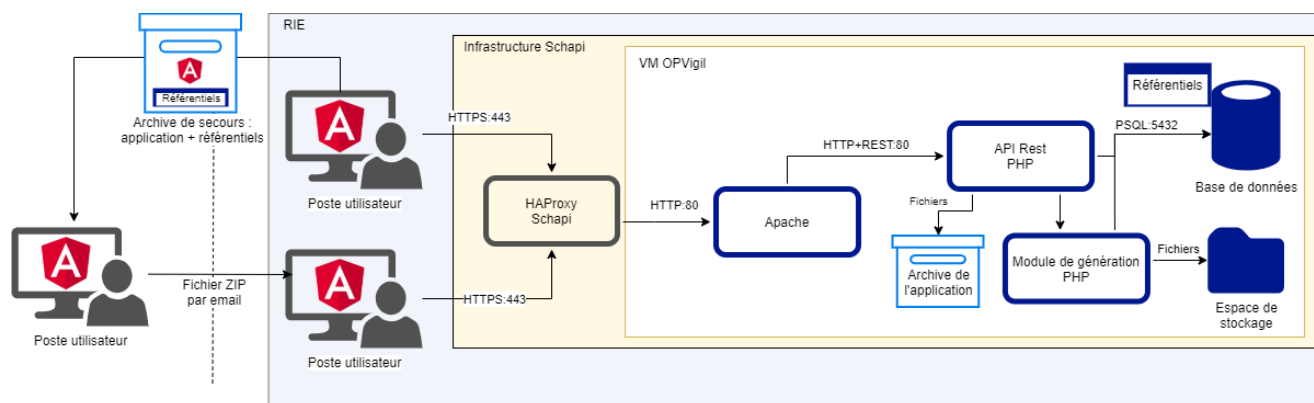


Figure 11: Flux - Production territoriale avec l'archive de secours

#### 2.4.8. Production nationale

La production nationale se fait toujours en mode connecté. Les bulletins sont sauvegardés dans la base de données.

Une fois la production terminée, le module PHP de génération est en charge de la génération des fichiers de sortie :

- Cartes de vigilances
- Fichier XML pour Météo France
- Bulletins au format PDF
- Fichiers de contacts
- ... (liste exhaustive dans le document l'annexe « Tableau des fichiers à produire pour la publication de la vigilance crues » des spécifications fonctionnelles détaillées).

Ces fichiers sont placés dans l'espace de stockage local de la VM OPVigil de production. Ils sont ensuite envoyés par le module de publication sur les serveurs FTP de Vigicrues. Ce module est aussi en charge de l'envoi des mails et de l'archivage sur la VM OPVigil d'archivage (§2.4.9).

L'API REST effectue l'envoi direct à Météo France avec les identifiants présents dans les fichiers de configuration.

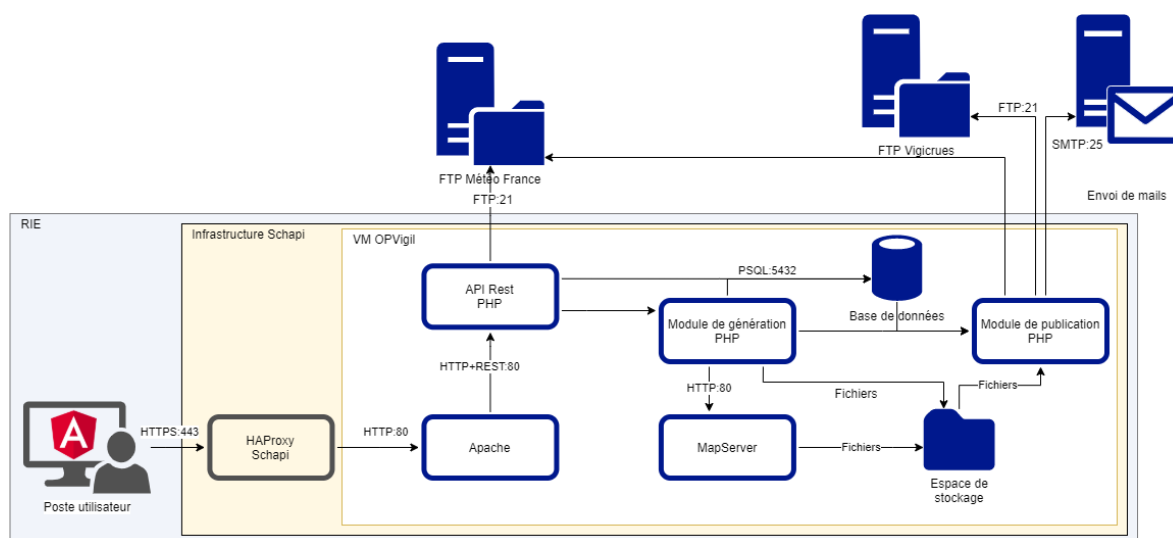


Figure 12 : Flux - Production nationale

**CONFIDENTIEL**

## 2.4.9. Archivage de la production

L'archivage des productions historiques est réalisé par le module de publication :

- Insertion dans la base de données d'archivage
- Copie en SFTP des fichiers de production vers l'espace de stockage de la VM OPVigil d'archivage.

L'archivage est effectué immédiatement après la publication, par la même tâche du « worker » Symfony.

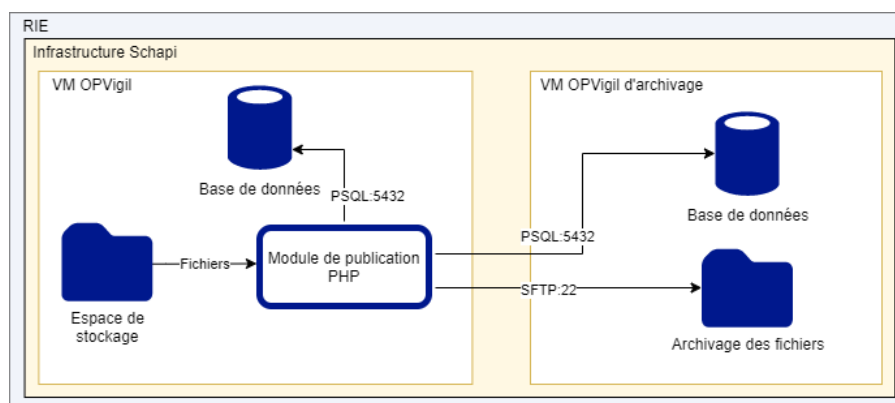


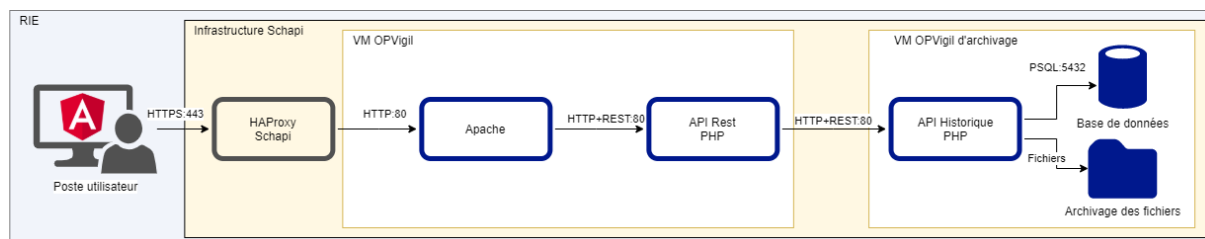
Figure 13 : Flux - Archivage de la production

## 2.4.10. Consultation de l'historique et des indicateurs

Les calculs liés à la consultation des données historiques et des indicateurs sont séparés sur l'API Historique, déployée sur la VM OPVigil d'archivage.

Aucun traitement n'est effectué par l'API Rest : les requêtes et réponses sont transmises directement, au format JSON, entre le frontend Angular et l'API Historique.

L'API Rest effectue les vérifications de permissions par rapport au rôle de l'utilisateur connecté. Une authentification basique est mise en place entre l'API Rest et l'API Historique.



## 2.4.11. Nettoyage des données de production

Une tâche périodique (par défaut tous les mois, paramétrable dans la configuration OPVigil) réalise le nettoyage des données de production. Cette tâche supprime de la base de production les bulletins datant de plus de 30 jours (valeur par défaut, configurable dans les variables d'environnement de l'application) ainsi que les fichiers de production associés sur l'espace de stockage local.

Une sauvegarde SQL (« dump » de la base de données) est réalisée avant la suppression. Ces sauvegardes sont stockées sur le système de fichier de la VM de production. En cas d'erreur de sauvegarde, le nettoyage n'est pas effectué.

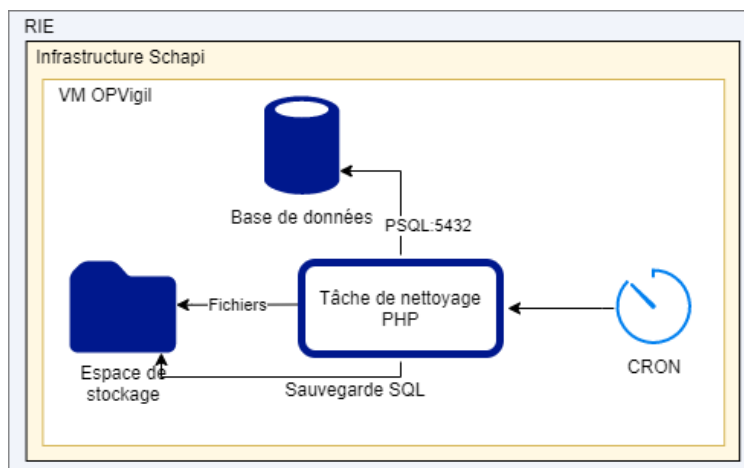


Figure 14: Flux - Nettoyage des données de production

## 2.5. Infrastructure technique

### 2.5.1. Machines virtuelles

Les machines virtuelles sont dimensionnées selon les caractéristiques suivantes :

#### VM Opvigil de production

- Système d'exploitation : Debian 11
- Processeur : 4 cœurs
- RAM : 4Go
- Espace disque : 150Go

#### VM Opvigil d'archivage

- Système d'exploitation : Debian 11
- Processeur : 4 cœurs
- RAM : 4Go
- Espace disque : 500Go
  - La VM doit être configurée avec un grand nombre d'inodes disponibles pour pouvoir stocker tous les fichiers de l'historique de la vigilance : 22 634 496 inodes

### 2.5.2. Certificats TLS

Le certificat TLS est géré au niveau du reverse proxy du SCHAPI.

Les communications entre le reverse proxy du SCHAPI et la VM OPVigil ainsi que les communications à l'intérieur de la VM sont en http.

### 2.5.3. Haute disponibilité et scalabilité

OPVigil est installé dans une machine virtuelle de production et une machine d'archivage. Chaque service est exécuté en une instance. Néanmoins, l'architecture présentée dans ce document et l'application sont spécifiées pour permettre la scalabilité d'un maximum de services si le besoin se présente.



## 2.6. Fonctionnements en mode dégradé

### 2.6.1. Indisponibilité de Cerbère

Comme la gestion des droits utilisateurs a lieu dans OPVigil, l'application doit maintenir une base des comptes utilisateurs à partir des informations retournées par Cerbère.

En cas d'indisponibilité de Cerbère, cette base locale sera utilisée directement :

- OPVigil détecte l'indisponibilité de Cerbère (ou un administrateur système déclare une panne Cerbère, pertinence des possibilités à évaluer en phase de développement) ;
- Connexion avec le nom d'utilisateur Cerbère et un mot de passe de secours ;
- En cas d'oubli, ou de première utilisation, le mot de passe de secours est réinitialisé par email ;
- Le mot de passe de secours est sauvegardé, salé et hashé, dans la base locale ;
- L'utilisateur est connecté avec les droits définis dans la base locale.

### 2.6.2. Indisponibilité de Cerbère et des boîtes mail

Dans le cas où les boîtes mail sont également indisponibles en même temps que Cerbère, la procédure de réinitialisation de mot de passe décrite dans le point précédent ne peut pas être utilisée.

Pour pallier cette situation, un script sera à disposition du Pôle Système d'Information (PSI) du SCHAPI permettant de réinitialiser le mot de passe d'un utilisateur en connexion directe à la base de données.

Ainsi, si un utilisateur ne se souvient plus de son mot de passe, ou ne l'a jamais défini, et que les boîtes mail ne sont pas disponibles, un administrateur du PSI SCHAPI pourra réinitialiser son mot de passe de secours et le transmettre à l'utilisateur.

### 2.6.3. Indisponibilité du RIE : mode hors ligne

Plusieurs cas d'indisponibilité du RIE sont à prendre en compte :

- Perte de connexion dans un SPC et VPN indisponible ;
- Perte de RIE entre le backend OPVigil et Vigicrues.

Pour les prévisionnistes et administrateurs SCHAPI, l'application sera toujours accessible depuis le réseau interne SCHAPI, ce cas n'est donc pas à prendre en compte dans le mode hors ligne.

#### Perte de connexion dans un SPC et VPN indisponible

Lorsque l'application détecte une perte de connexion, c'est-à-dire l'indisponibilité du backend PHP Symfony, elle passe en mode hors ligne.

Dans ce mode, il n'y a plus d'authentification, et tout utilisateur a accès aux outils de production de la vigilance territoriale en local. L'application utilise les données mises en cache dans le navigateur pour le référentiel géographique.

Les bulletins sont produits et sauvegardés dans le cache local. Une option permet de télécharger une archive de la production. Cette archive est ensuite à envoyer par email au SCHAPI.

Une autre option, accessible par les utilisateurs SCHAPI, permet d'intégrer la production territoriale reçue par email dans OPVigil pour utilisation dans la production nationale.

Note : Si l'application web n'a jamais été chargée dans le navigateur ou si le cache du navigateur a été vidé, elle ne pourra pas démarrer dans ce mode hors ligne. Dans cette situation un autre mode de secours est disponible sous forme d'une archive téléchargeable de l'application (2.6.4).

#### Perte de connexion entre le backend OPVigil et Vigicrues

En cas de coupure de la connexion entre le backend OPVigil et Vigicrues, les productions ne peuvent pas être envoyées par les scripts de publications.

Dans cette situation le backend PHP est accessible par le frontend Angular OPVigil qui reste en mode en ligne. Les fonctionnalités d'authentification, gestion des droits et de production sont disponibles.

L'application permet de télécharger une archive de la production pour la transmettre à l'hébergeur de Vigicrues.

#### 2.6.4. Archive téléchargeable de l'application

Dans le cas où l'application OPVigil n'a jamais été utilisée sur un poste ou qu'elle n'est pas présente dans le cache du navigateur, une solution doit permettre aux utilisateurs de récupérer l'application localement afin de pouvoir tout de même produire.

Pour cela, une version portable de l'application est produite sous forme d'une archive de secours. Cette archive contient les fichiers statiques de l'application Angular, les données des référentiels et un script permettant d'ouvrir l'application avec un serveur web local sans installation.

L'archive est construite à la demande par OPVigil à partir des données de référentiel géographique en base de données. Elle peut être téléchargée par un administrateur.

L'archive peut être téléchargée après chaque mise à jour du référentiel ou à la demande pour être transmise à un utilisateur qui en a besoin.

L'archive et les bulletins générés doivent posséder un numéro de version. Ce numéro de version sera validé à l'import des bulletins afin d'éviter le cas où un bulletin serait généré avec une ancienne version de l'application et des référentiels.

## 3. Composants logiciels

### 3.1. Système d'exploitation

Les VMs utilisent Debian 11 (Bullseye) qui est l'OS des VMs de références SCHAPI.

### 3.2. Liste des applicatifs

Les composants suivants seront installés sur les VM OPVigil. Les versions LTS et disponibles dans les dépôts Debian 11 sont privilégiées.

#### 3.2.1. Apache

Apache est un serveur web par lequel l'ensemble du trafic HTTP sera servi. Il sera également utilisé pour servir les fichiers statiques du frontend Angular OPVigil.  
La version à installer est la 2.4, version fournie sur Debian 11.

#### 3.2.2. PHP

PHP est le langage du backend OPVigil, dans sa version 7.4.

#### 3.2.3. MapServer

MapServer est un serveur cartographique permettant de produire des cartes à partir de fichiers de configuration au format « .map » et une source de données comme une base PostgreSQL avec l'extension PostGIS.

La version de MapServer visée est la 7.6.

Images cliquables – « AreaMap »

Différentes technologies existent pour définir des zones cliquables sur une image de carte :

- Images et AreaMap :  
Solution utilisée aujourd'hui sur ProdVigil, les AreaMaps définissent les zones cliquables par liste de point formant un polygone. Ces zones sont incluses dans le code HTML de la page web et appliquées sur une image produite séparément. Les AreaMaps nécessitent cependant l'injection de code javascript pour corriger la responsivité et intégrer les liens et infobulles.
- Images au format SVG :  
Le format d'image SVG permet d'inclure des zones cliquables. Leur définition se fait de la même façon que pour les AreaMaps et les liens sont embarqués dans l'image. Le SVG a également un meilleur support de la responsivité que les AreaMaps.
- Mapfile côté serveur :  
Cette option déporte la configuration des zones cliquables côté serveur. Au clic, le navigateur envoie les coordonnées au serveur qui répond avec l'URL de la page à suivre. Cette solution n'est pas adaptée à OPVigil car elle nécessite la mise en place d'un serveur accessible publiquement.

Ces trois solutions utilisent la définition en polygones des zones cliquables utilisée aujourd'hui par ProdVigil et l'intègrent différemment sur une image.

Afin de maintenir la compatibilité des fichiers de sorties de la production envoyés à Vigicrues, la première version d'OPVigil conservera la solution actuelle à base d'images et AreaMaps. Un travail sera mené au niveau de leur création pour optimiser l'utilisation des templates dans MapServer.

Dans un second temps, une évolution vers des SVG sera envisagée en synchronisation avec Vigicrues pour améliorer la responsivité des cartes.

#### 3.2.4. PostgreSQL / PostGIS

Le serveur de base de données PostgreSQL contient les référentiels et production en cours. PostGIS est une extension à PostgreSQL pour la gestion de données géoréférencées.

PostgreSQL est installé en version 13 et PostGIS en 3.

### 3.3. Frameworks et bibliothèques logicielles

#### 3.3.1. Angular

Angular est un framework MVC Javascript utilisé pour le développement du frontend OPVigil. Il permet la mise en place d'une application Web dynamiques et fluidifie l'expérience utilisateur en évitant les chargements de pages à chaque nouvelle action. Il permet également la mise en place de Service Worker pour l'exécution d'applications en mode non connecté.

La version utilisée est la dernière LTS : 11.0.

#### 3.3.2. Bootstrap

Bootstrap est un framework CSS permettant un cadre de base pour les composants HTML et pour la responsivité de l'application.

La version utilisée est la dernière version stable compatible avec Angular 11.

#### 3.3.3. Symfony

Symfony est le framework PHP utilisé pour développer le backend OPVigil.

La version utilisée est la dernière LTS : 4.4

Les bundles Symfony (extensions) seront pris dans leurs dernières versions compatibles avec la 4.4 LTS.

#### 3.3.4. Doctrine

Doctrine est un ORM permettant d'appliquer une couche d'abstraction à la base de données pour PHP.

Les mises à jour du schéma de la base de données sont gérées avec doctrine-migration.

#### 3.3.5. DomPDF

DomPDF est une bibliothèque PHP, compatible avec Symfony, utilisée pour la génération de fichiers PDF. Elle utilise des templates HTML et supporte la mise en page CSS.

### 3.3.6. PDF.js - PDFAssembler

PDFAssembler est une bibliothèque javascript permettant la concaténation de fichiers PDF. Elle se base sur PDF.js, un outil de Mozilla compatible avec tous les navigateurs pour l'affichage de PDF.

### 3.3.7. jsPDF

jsPDF est une bibliothèque javascript permettant la génération de fichiers PDF en local dans le navigateur. Elle supporte l'intégration d'images, la mise en page et peut utiliser des templates HTML.

### 3.3.8. ApexCharts

ApexCharts est une bibliothèque javascript permettant la génération de graphiques divers en local dans le navigateur. Elle propose différents types de rendus correspondants aux indicateurs attendus par le SCHAPI ainsi que les fonctions de zoom et d'export.