ARCHITECTURE DU LOGICIEL

Version :	0.1
Date :	17/11/2016
Rédigé par :	L'équipe SmartLogger
Relu par :	L'équipe SmartLogger
Approuvé par :	---

Objectif : Le but de ce document est de décrire les solutions techniques conçues pour répondre aux exigences définies dans la spécification technique de besoin. Il doit identifier et décrire les différents modules ou constituants du logiciel ainsi que leurs interfaces de telle sorte que chacun d'entre eux puisse être développé de façon autonome par un membre de l'équipe avant d'être intégré.

La lecture de ce document doit également permettre d'appréhender l'ensemble des paramètres techniques pris en considération par les auteurs pour définir et élaborer les stratégies et démarches de développement.

HISTORIQUE DE LA DOCUMENTATION

Version	Date	Modifications réalisées
0.1	17/11/2016	Création

1. Objet :

Le but du projet est de créer un système, permettant d'alerter l'utilisateur sur des données en provenance d'applicatifs défectueux dans l'optique de faciliter leurs correctifs. Une fois produit, ce système sera utilisé par l'entreprise cliente à leurs propres fins.
Dans ce but, l'entreprise cliente a fait part de ses exigences en termes de fonctionnalités minimales :

- Le système devra être capable d'analyser des flux de données de type Shinken, Logstash et Kafka.
- Il alertera les opérateurs en employant des moyens de communication utilisés par l'entreprise, tels que l'application Slack, l'envoi de mails ou par notification SMS.
- Il pourra s'adapter à des flux de données ou méthodes d'alerte non prédéfinies qui seront implantées dans le système par de futurs utilisateurs.
- Enfin, il devra fonctionner sur de longues périodes de fonctionnement et, dans l'idéal, être opérationnel indéfiniment.

A ce titre, l'exigence principale en terme de conception réside dans le découpage efficace de l'application à réaliser. Le souhait du client étant de récupérer un système facilement personnalisable. Il faudra s'assurer que certains composants-clés soient aisément interchangeables :

- Le type d'algorithme employé par l'analyseur du système
- Le type de format de données à traiter
- Le type d'alerte et de notifications

2. Documents applicables et de référence

- Le document de Spécification Technique du Besoin : STB.pdf
- Le document de présentation client : SmartLogger.pdf

3. Terminologie et sigles utilisés

Définitions et Notions :

Log : Un fichier log ou «log» est un fichier contenant l'historique des événements d'un processus en particulier. Ici, nous considérerons des logs issus d'applications WEB externes.

Événement : Un événement représente un changement au sein d'un processus. Il peut s'agir d'un changement en mémoire (ajout d'une donnée), un changement dans l'interface (clic de souris), etc.

Flux de données : Un flux de données représente une quantité potentiellement illimitée de données qui est manipulée sur un laps de temps défini afin d'en extraire une certaine quantité.

Traitement en temps réel : Un traitement est réalisé en temps réel, si le système qui l'effectue peut adapter sa vitesse à l'évolution de ce dernier.

Machine Learning (ML) : Champ d'étude de l'intelligence artificielle ayant pour but de faire évoluer, au cours du temps, la façon dont un système effectue un même traitement. Par extension, un algorithme de Machine Learning est un algorithme permettant d'implanter une telle capacité d'apprentissage à un système.

User Interface (UI) : Désigne une application pouvant être manipulée par un utilisateur dans le but d'utiliser un système.

Technologies employées :

SGBD : Système de gestion de base de données, permet de stocker, manipuler et organiser un (très) grand nombre de données diverses.

Shinken : Application permettant la surveillance de systèmes et de réseaux. Elle surveille les hôtes et services spécifiés, lançant une alerte lors d'éventuelles fluctuations de l'état actuel d'un système.

Logstash : Outil informatique permettant de gérer des événements et des logs.

Apache Kafka : Projet open-source visant à fournir un système unifié en temps réel à latence faible pour la manipulation de flux de données.

Slack : Logiciel de gestion de projets, principalement utilisé pour son système de communication entre membres d'équipes de projet.

MongoDB : SGBD orienté documents, il permet de manipuler des données sans avoir à concevoir la façon dont ces dernières seront gérées en interne.

NoSQL : Désigne une certaine famille de SGBDs pouvant manipuler de plus grands volumes de données en outrepassant d'anciennes règles pré-établies sur les autres types de SGBD.

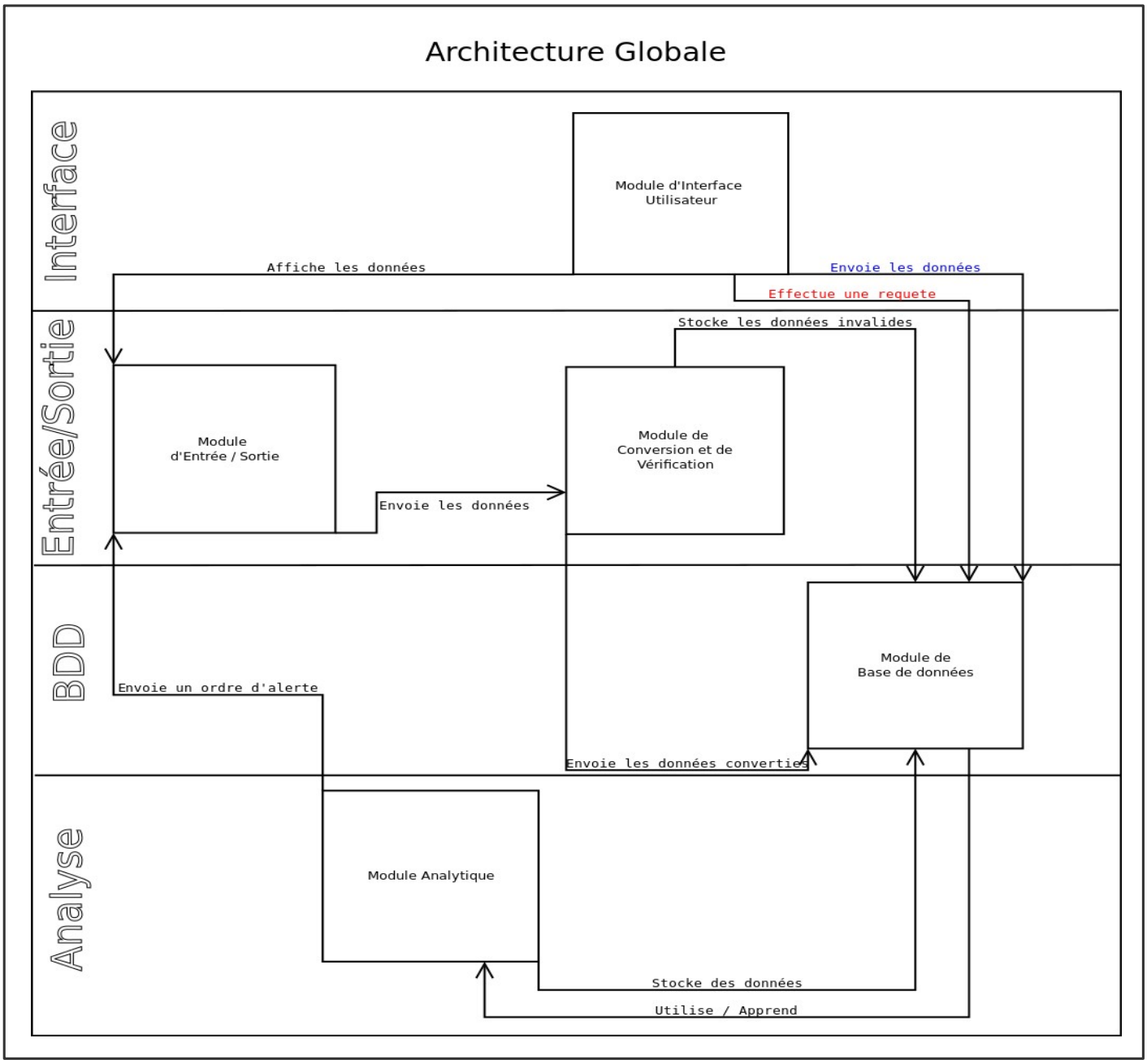
4. Configuration requise

- Périphérique hôte : Serveur local de l'entreprise cliente
- Système d'exploitation : OS de type Linux (version précise non définie)
- Produits et composants logiciels utilisés :

Nom	Origine	Type	Version
Shinken		Logiciel	2.4
Logstash		Logiciel	2.4
Kafka		Logiciel	0.9
Spark		Bibliothèque	2.0.1
MongoDB		SGBD	3.2.10
Angular 2		Framework	2.0.0
Spring Boot		Framework	1.4.2

5. Architecture statique

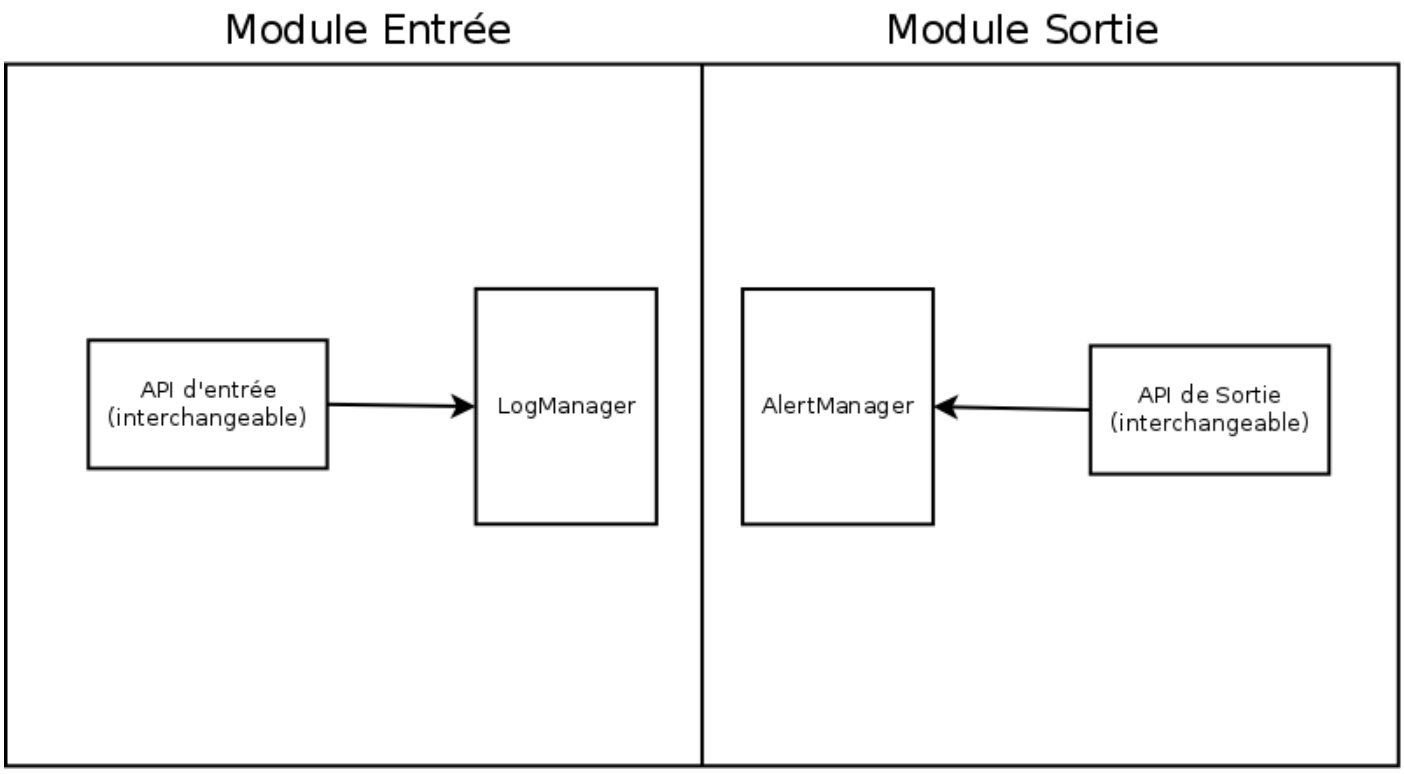
5.1. Structure principale du système



Le système s'organise en 5 modules distincts :

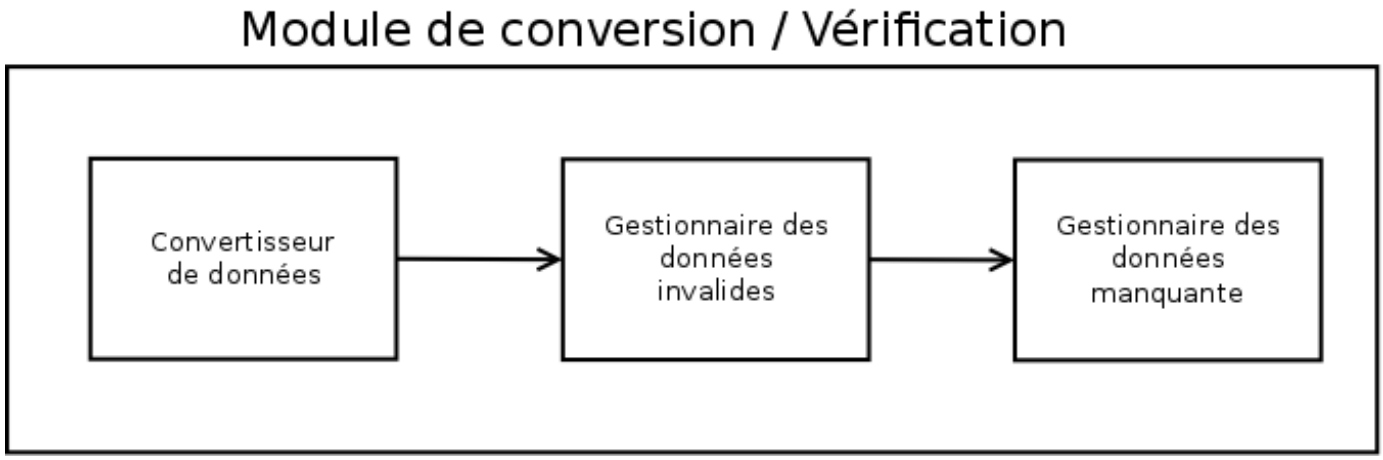
- Le module d'entrée/sortie (ES) : Permet toute interaction entre le système et tout acteur extérieur.
- Le module de conversion/vérification (CV) : Assure la validité des données avant leur exploitation par les autres modules du système.
- Le module d'interface utilisateur (UI) : Permet à un opérateur de communiquer avec le système : soit dans le but de visualiser les actions du système, soit pour émettre des ordres.
- Le module de base de données (DB) : Réalise le stockage permanent de toute donnée ayant été manipulée par le logiciel.
- Le module d'analyse (ML) : Constitue le modèle applicatif majeur du système, il réalise l'ensemble de ses analyses et prédictions sur les différents échantillons de données fournis.

5.2. Module ES – Entrée et Sortie du système



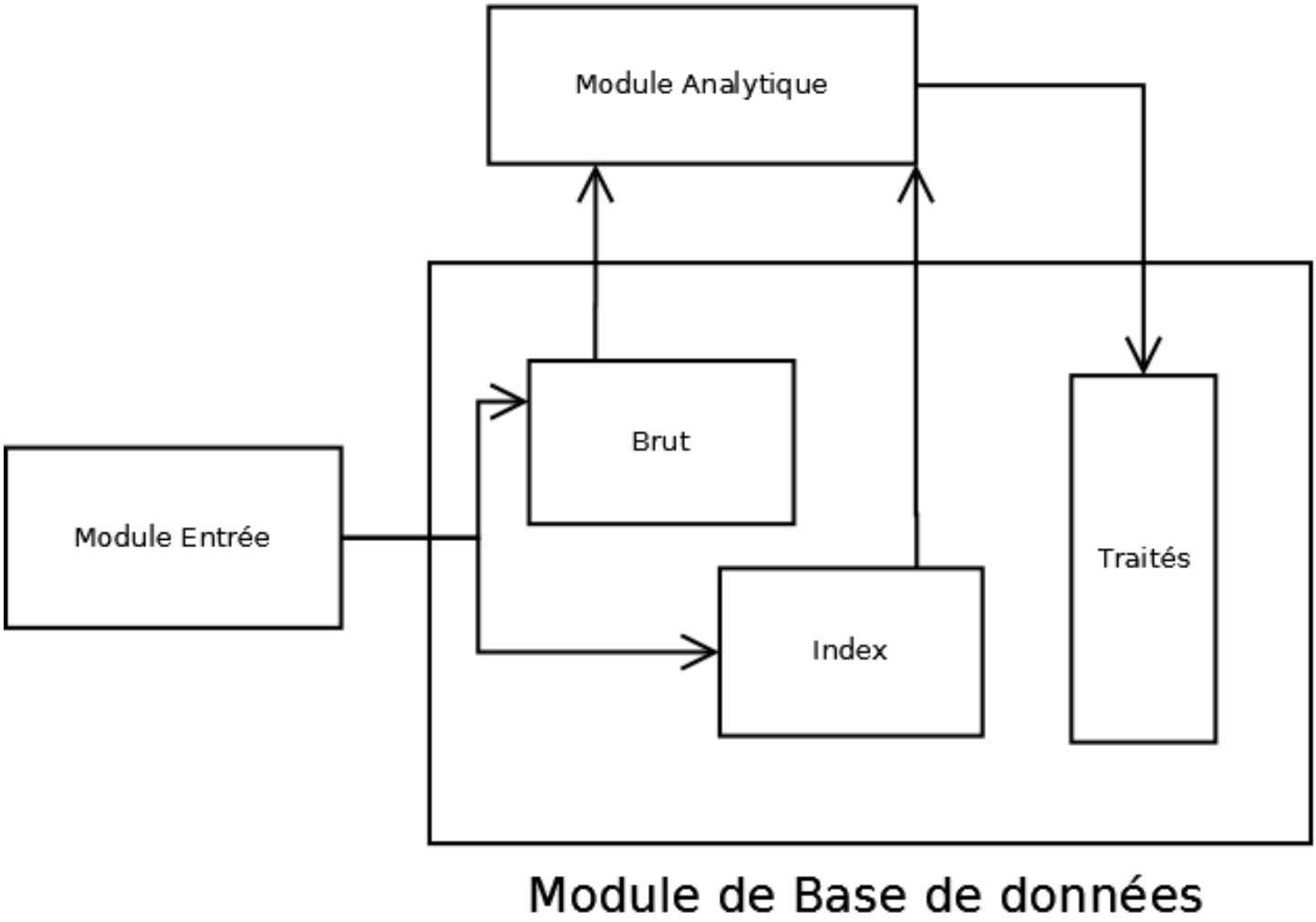
Rôle	Gestionnaire d'entrée / sortie
Propriétés	Modules interchangeables, Adaptabilité à diverses API
Attributs	Unique module capable de communiquer avec un acteur extérieur
Services offerts	Entrée : Permet aux fournisseurs de données de les faire transiter Sortie : Relaye les alertes lancées par le système
Dépendances	Aucune dépendance
Langage de programmation	Java
Procédé de développement	Utilisation du framework Spring Boot
Taille	Faible
Complexité	Moyenne

5.3. Module CV – Conversion et Vérification des données



Rôle	Assure l'utilisabilité des données pour le module d'analyse
Propriétés	Modules interchangeables
Attributs	Module permettant une vérification des données avant leurs insertions réelles dans le système
Services offerts	Conversion des données dans un format unique Vérifie la validité des données (format et données corrompues) Gère l'absence de certaines données majeures
Dépendances	Aucune dépendance
Langage de programmation	Java
Procédé de développement	Utilisation du framework Spring Boot
Taille	Faible
Complexité	Moyenne

5.4. Module DB – Base de Données



Rôle	Assure l'interaction avec la base de données
Propriétés	Unique module interagissant avec la base de données
Attributs	
Services offerts	Permet de requêter la base de données. Permet d'ajouter de nouvelles entrées
Dépendances	Module Entrée et du module Analytique
Langage de programmation	Java, NoSQL, JSON
Procédé de développement	Utilisation du SGBD MongoDB et du framework Spring Boot
Taille	Faible
Complexité	Faible