

SPECIFICATION TECHNIQUE DE BESOIN

Version : 0.1 (Non définitive)
Date : 03/11/2016
Rédigé par : L'équipe SmartLogger
Relu par : L'équipe SmartLogger
Approuvé par : ---

Signature du superviseur :

Objectif : Ce document est destiné à traduire les besoins des utilisateurs du logiciel et à établir une référence pour sa validation. Il doit être élaboré en coopération avec le demandeur puis approuvé par ce dernier. Son but est de recenser les principales exigences que l'équipe de développement s'engage à satisfaire dans le cadre du projet.

HISTORIQUE DE LA DOCUMENTATION

Version	Date	Modifications réalisées
0.1	02/11/06	Création
0.2		Ajouts de cas d'utilisations et correctifs

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.

L'objectif est de mettre un place un système d'alarme, basé sur ses propres capacités d'apprentissage. Sa tâche sera de prioriser les différentes données à traiter, selon leur niveau de criticité, et d'adopter une stratégie de prévention adaptée.

Dans cette optique, le client a recommandé l'emploi des technologies suivantes :

- Spring Boot : afin de réaliser la partie applicative.
- Angular 2 : pour la réalisation de l'interface graphique.
- Spark : pour la mise en place des algorithmes.

Ce dernier souhaite également que le produit respecte les contraintes suivantes :

- Il doit analyser des flux de données de type Shinken, Logstash et Apache Kafka.
- Il doit transmettre ses alertes via SMS, Mail, ou par l'application Slack.
- Il doit pouvoir s'adapter à d'autres flux de données et à d'autres méthodes d'alerte.

Le produit final devra ainsi être capable de fournir des résultats satisfaisants sur une longue durée d'exécution.

2. Documents applicables et de référence

- Le document de présentation du projet : 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'études 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 d'organiser un (très) grand nombre de données diverses.

Shinken : Application permettant la surveillance système et réseau. Elle surveille les hôtes et services spécifiés, alertant lorsque les systèmes vont mal et quand ils vont mieux.

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 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. Exigences fonctionnelles

4.1. Présentation de la mission du produit logiciel

Le logiciel final, bien que scindé en plusieurs parties distinctes sera déployé sur un serveur externe appartenant au client, du simple fait qu'il nécessite divers flux de données issus de systèmes externes. Il agira en tant qu'intermédiaire entre les logs et les opérateurs de maintenance, prenant ainsi position dans les procédures de maintenance des applicatifs du client.

Ce système sera employé par 3 types d'utilisateurs différents :

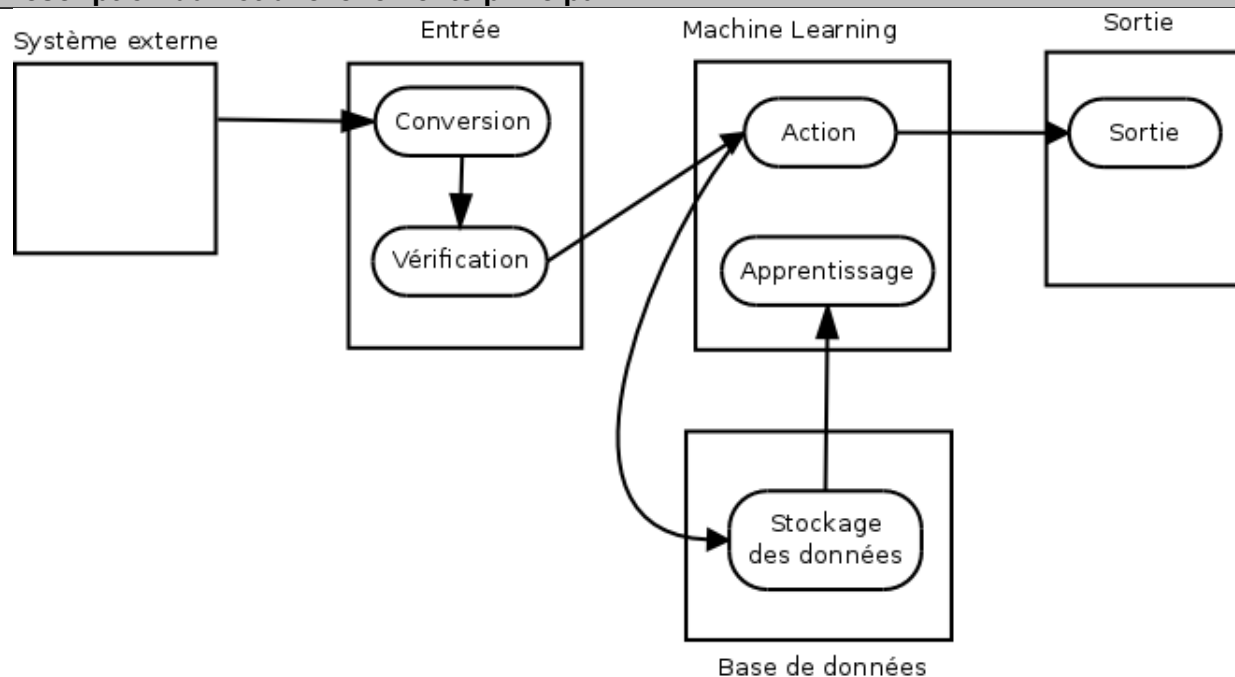
- Des **systèmes informatiques**, principaux fournisseurs de données du système.
- Des **entraîneurs**, fournissant à la machine la matière première pour lui fournir un apprentissage de base permettant le traitement des futures données reçues.
- Des **testeurs**, corrigeant les potentielles erreurs de comportement du système, et pouvant affiner les réglages de sa capacité d'apprentissage.

Ainsi, le système dispose de cinq cas d'utilisation.

4.2. Cas d'utilisation n°1 : Exécution périodique

Nom :	Exécution périodique
Acteurs concernés	Un système externe, le machine learning et la base de données.
Description	<p>Cas d'utilisation de la machine lors d'une exécution périodique.</p> <p>La machine reçoit des données d'un système externe, qui sont converties et vérifiées par un module interne. La machine reçoit ensuite ces données, effectue le stockage des données dans la base de données. Elle utilise ensuite ces données afin d'apprendre et d'améliorer son fonctionnement. On effectue ensuite potentiellement une sortie dans le cas de données à traiter.</p>
Pré-conditions	<p>a) Le système est opérationnel.</p> <p>b) Le système dispose de données à traiter.</p> <p>c) La machine a déjà « appris ».</p>
Événements déclenchants	a) Réception de données d'un système externe.
Conditions d'arrêt	<p>a) La machine a fini d'apprendre, stocker et traiter les données.</p> <p>b) Une sortie est potentiellement envoyée suite au traitement des données.</p>

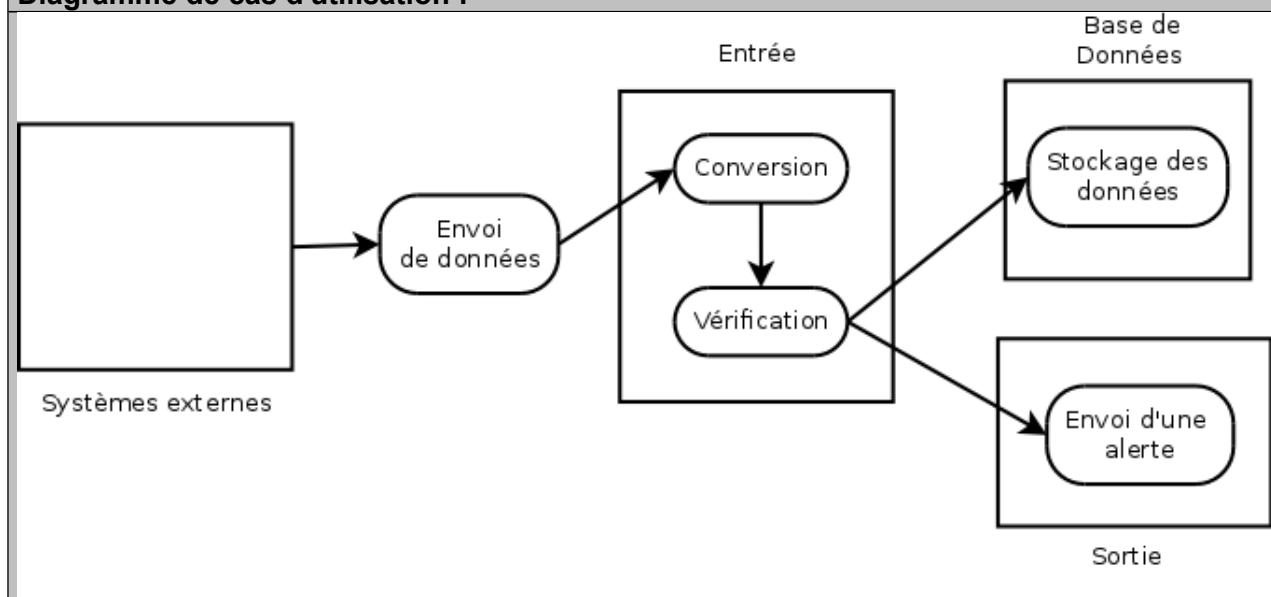
Description du flot d'événements principal :



4.3. Cas d'utilisation n°2 : Cas de stockage de données invalides dans la base de données.

Nom :	Stockage de données erronées
Acteurs concernés	Systèmes externes, le système et la base de données.
Description	Cas d'utilisation dans le cas où le système reçoit des données erronées. Le système reçoit des données erronées. Il tente de les convertir puis de les vérifier. Dans le cas où celles-ci sont erronées, on les stocke dans la base de données et on envoie une alerte.
Pré-conditions	a) Le système est opérationnel. b) Le système dispose de données à traiter. c) La machine a déjà « appris »
Événements déclenchants	a) La conversion des données a échoué. b) La vérification de la validité des données a échoué.
Conditions d'arrêt	a) Les données sont stockées dans la base de données. b) Une alerte est envoyée.

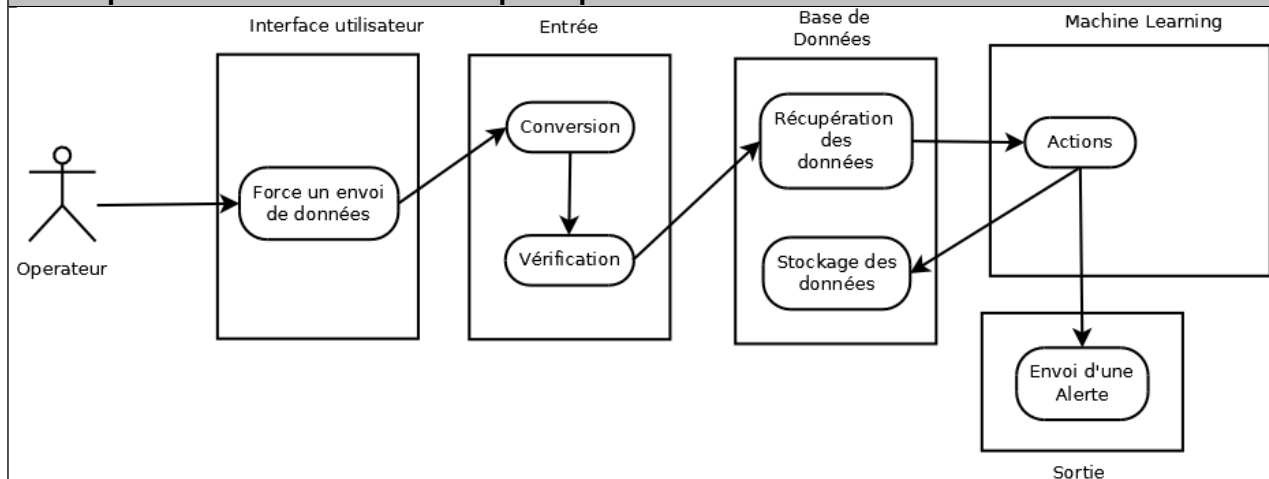
Diagramme de cas d'utilisation :



4.4. Cas d'utilisation n°3 : Utilisation de données erronées

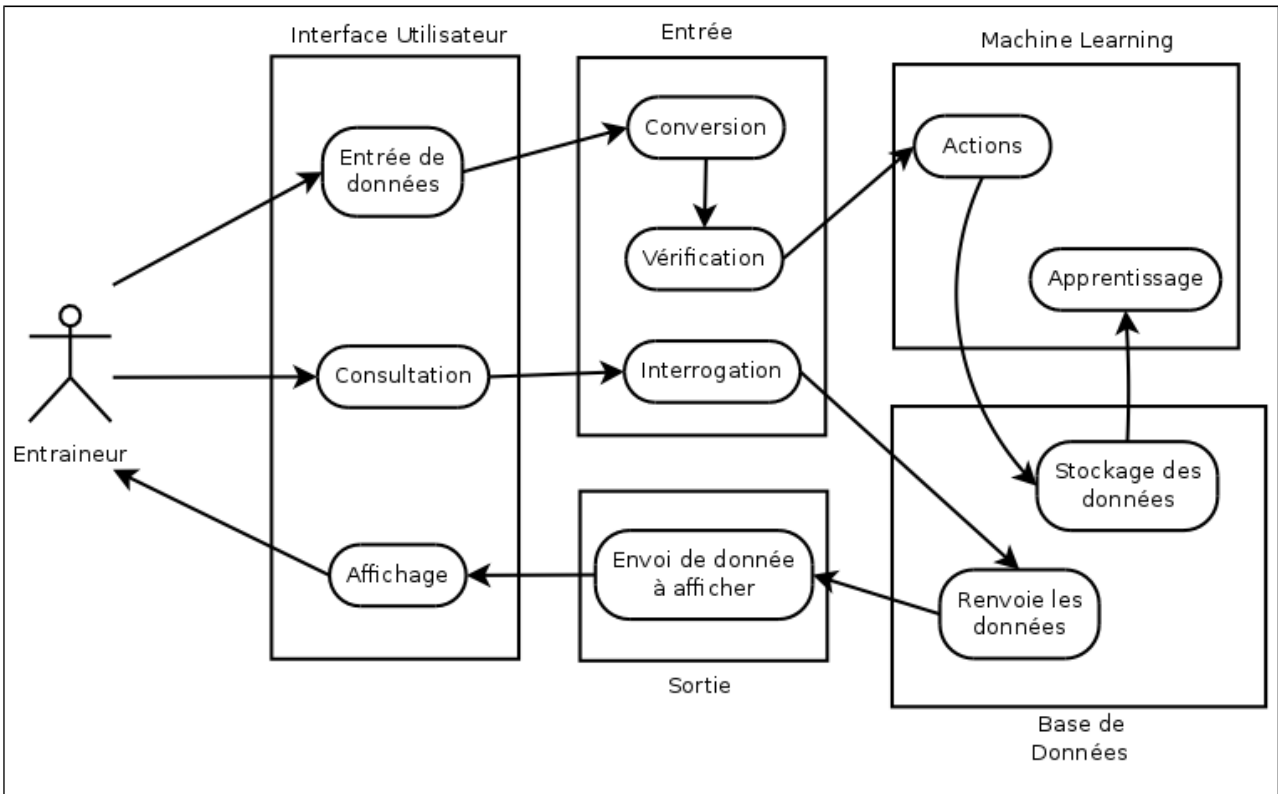
Nom :	Utilisation de données erronées
Acteurs concernés	Un opérateur, l'interface graphique, la base de données et la Machine Learning.
Description	<p>Cas d'utilisation dans le cas ou le système reçoit des données erronées.</p> <p>Le système reçoit de force des données invalides envoyées par un opérateur. Le système récupère les données, qui sont envoyées à la Machine Learning qui effectue son travail, puis elle effectue le stockage des données dans la base de données et envoie une alerte.</p>
Pré-conditions	<p>a) Le système est opérationnel.</p> <p>b) Le système dispose de données à traiter.</p> <p>c) La machine a déjà « appris »</p>
Événements déclenchants	<p>a) Envoi de force de données invalide.</p> <p>b) La conversion des données a échoué.</p> <p>c) La vérification de la validité des données a échoué.</p>
Conditions d'arrêt	<p>a) Les données sont stockées dans la base de données.</p> <p>b) Une alerte est envoyée.</p>

Description du flot d'événements principal :



4.5. Cas d'utilisation n°4 : Entrainement de la Machine Learning.

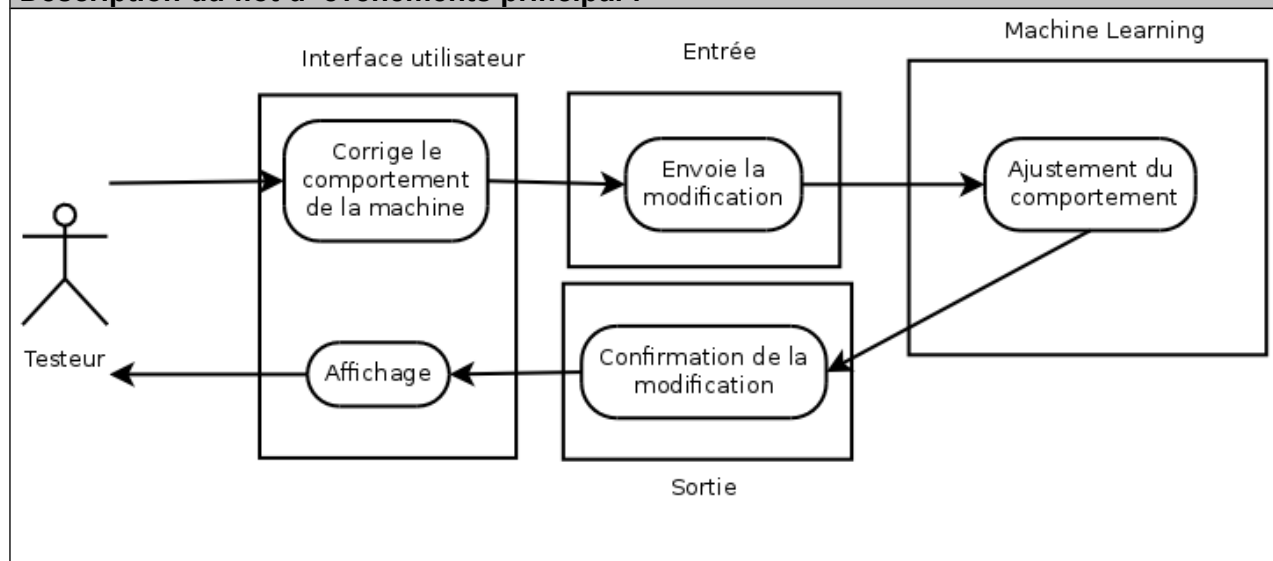
Nom :	Entrainement de la Machine.
Acteurs concernés	<i>Un entraîneur, l'interface utilisateur, la machine Learning et la base de données.</i>
Description	<p><i>Cas d'utilisation dans le cas de l'entrainement de la Machine Learning. Pour cela, un utilisateur humain (l'entraîneur) entre des données. Ces données sont converties par notre module de conversion et de vérification des données. Ces données sont ensuite envoyées à la Machine qui effectue le stockage de ces données, puis qui utilise toute la base de données afin de s'entraîner.</i></p> <p><i>L'entraîneur peut aussi consulter la base de données afin de vérifier le résultat de l'apprentissage de la Machine.</i></p>
Pré-conditions	<p><i>a) Le système est opérationnel.</i></p> <p><i>b) Le système dispose de données à traiter.</i></p> <p><i>c) La base de données possède des données affichables.</i></p>
Événements déclenchants	<p><i>a) L'entraîneur envoie des données que la machine peut traiter afin d'apprendre.</i></p> <p><i>b) Les données entrées sont valides.</i></p> <p><i>c) La conversion des données est effectuée.</i></p> <p><i>d) La machine stocke les données dans la base de données.</i></p> <p><i>e) La machine utilise ces données afin de s'améliorer.</i></p> <p><i>f) L'utilisateur consulte la base de données.</i></p>
Conditions d'arrêt	<i>a) La machine a fini d'apprendre et attends le prochain envoi de données.</i>
Description du flot d'événements principal :	



4.6. Cas d'utilisation n°5 : Ajustement du comportement de la machine.

Nom :	Ajustement du comportement de la machine
Acteurs concernés	Un testeur, une interface utilisateur, la machine learning.
Description	Cas d'utilisation d'ajustement de la machine. Un testeur envoie un correctif de comportement à la machine. La machine reçoit la modification et ajuste son comportement. Elle renvoie ensuite une confirmation de modification de son comportement.
Pré-conditions	a) Le système est opérationnel. b) La machine a déjà « appris ».
Événements déclenchants	a) Envoi d'un correctif comportemental d'un testeur humain à la machine. b) La machine ajuste son comportement en fonction des paramètres fournis par le testeur.
Conditions d'arrêt	a) La machine renvoie un message de confirmation de la réussite de la confirmation.

Description du flot d'événements principal :



5. Exigences opérationnelles

Référence	Désignation
OP-01	Le programme doit fonctionner sans interruption
OP-02	Les résultats déjà traités seront envoyés de manière périodique

6. Exigences d'interface

Référence	Désignation
IN-01	L'interface d'entrée est compatible d'emblée, avec un certain nombre de différents flux de données. (cf RE-01)
IN-02	L'interface d'entrée doit s'adapter à de nouveaux types de données à traiter.
IN-03	L'interface de sortie doit renvoyer ses résultats sous plusieurs formes différentes. (cf RE-02)
IN-04	L'interface de sortie peut s'adapter à des formats de sorties supplémentaires.

7. Exigences de qualité

Référence	Désignation
QL-01	Le code source fourni doit être propre et commenté.
QL-02	Les noms de variables et fonctions utilisés sont significatifs par rapport à leur utilité.
QL-03	Les différents identificateurs utilisés seront écrits en anglais.

8. Exigences de réalisation

Référence	Désignation
RE-01	Le système doit pouvoir accepter des flux d'entrée, de type Shinken, Logstash et Apache Kafka.
RE-02	Le système doit prédéfinir des flux de sortie, de type SMS, Slack et Mail.
RE-03	Le système sélectionne une sortie préférée, en fonction de la priorité associée à l'alerte.
RE-04	Créer une interface utilisateur ergonomique.
RE-05	Créer une base de données de type NoSQL, enregistrant les traitements effectués et leurs données.
RE-06	Le système prévoit et détecte les erreurs en se basant sur un algorithme de Machine Learning.