Université Mohammed V - Rabat
École Nationale Supérieure d'Informatique
et d'Analyse des Systèmes

Mémoire de Projet de Fin d'études

Filière

Ingénierie du Web et Informatique Mobile

Option : Web Intelligence

Sujet :

Conception et développement d'un connecteur du switch monétique avec le système de monitoring Zabbix

---

Réalisé par :

Ayoub EL ASSARI

Devant le jury composé de :

M.Jamal EL HACHIMI (Président)

Mme.Laila CHEIKHI (Examinatrice)

M.Ahmed FAQIHI (Encadrant)-ENSIAS

M.Anas BENZIDIYA (Encadrant)-PayLogic

Année Universitaire 2022-2023

“

À mes chers parents, *pour leurs amours et leurs soutiens, aucune dédicace ne saurait être assez éloquente pour exprimer l’amour, l’estime, le dévouement et le respect que j’ai toujours eu pour vous. Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien-être. Ce travail est le fruit de vos sacrifices que vous avez consentis pour mon éducation et ma formation.*

À mes chers : Otman, Fatima *les mots ne suffisent pas pour exprimer l’amour et l’affection que je porte pour vous. Je vous dédie ce travail pour tout le soutien et la présence qui m’ont été d’un grand secours au long de ma vie.*

À toute la famille El assari, à tous mes amis : Ayoub Mansouri, Soufiane Mzizi, Abdlwahab Jbouri, Yasser Ouaziz, Ayoub Moussaoui, Hamza Elhnait, Adnane Drief, Aladin Cherkaoui, Oussame krab, Mohemed Boumaza *en qui j’avais raison de porter toute ma confiance, merci pour votre aide et soutien.*

Aux toutes mes professeurs, Aux familles Cje et Club Quran, aux bureaux ADEI 2020-2023, à tous les ensiastes, *je ne saurais vous exprimer en quelques mots, tous les sentiments de la gratitude et de l’amour que je vous porte.*

À mes grands parents décédés, *j’espère que vous apprécierez cette humble dédicace comme preuve de reconnaissance de la part d’un fils qui a toujours prié pour le salut de vos âmes. Puisse Allah, le tout puissant, vous avoir en sa sainte miséricorde.*

À tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail

Merci.

Remerciements

Avant de présenter plus en détail cette expérience professionnelle, il est important de commencer par exprimer ma gratitude envers ceux qui m'ont apporté de précieuses connaissances tout au long de ce stage et également remercier chaleureusement ceux qui ont contribué à rendre ce stage extrêmement bénéfique.

J'adresse mes sincères remerciements à ***M. Anas BENZIDIYA*** *le directeur téchnique de PayLogic et mon encadrant durant ce stage, pour son accueil convivial, son implication dans le suivi du déroulement du projet, ainsi que ses interventions propices émanant aux moments adéquats de confusion ou de doute, qui m'aidaient à chaque reprise à retrouver le meilleur chemin à suivre.*

Je tiens ensuite à exprimer ma profonde gratitude envers mon encadrant académique ***M. Ahmed FAQIHI*** *pour avoir accepté d'encadrer mon projet de fin d'études et avoir apporté tout l'aide nécessaire pour l'aboutissement du projet et la rédaction du présent PFE.*

Je remercie également toute l'équipe pédagogique de l'École Nationale Supérieure d'Informatique et d'Analyse des Systèmes (ENSIAS) pour avoir assuré une formation de haute qualité et, plus précisément, à notre chef de filière ***M. Ahmed ZELLOU*** *pour nous avoir aidé tout au long de notre parcours universitaire. Veuillez trouver ici le témoignage de notre respect le plus profond.*

Que tous ceux qui ont contribué de près ou de loin à l'aboutissement de ce projet trouvent l'expression de mes remerciements les plus sincères.

J'ai eu l'immense plaisir d'étudier au sein de l'École Nationale Supérieure d'Informatique et d'Analyse des Systèmes.

Résumé

Le présent document est le fruit du travail réalisé dans le cadre de mon projet de fin d'études effectué au sein de l'entreprise PayLogic à Casablanca, afin d'obtenir mon diplôme d'Ingénieur d'État en ingénierie du web et informatique mobile à l'École Nationale Supérieure d'Informatique et d'Analyse des Systèmes. Ce projet avait pour objectif la mise en place d'un connecteur de switch monétique avec les systèmes de monitoring externes.

Afin d'arriver à l'objectif de ce sujet on a commencé par une étude comparative entre les outils de surveillance et après le choix des outils on a les testé sur des machines virtuelles Linux et Windows.

Après le teste des outils on a passé vers l'environnement réel qui est le trésor public de Madagascar, on a surveillé les différents serveurs du trésor a savoir le serveur « Pay Way », le serveur « Payment(ONLINE) », le serveur « Proxy ».

pour tous ces serveurs on a surveillé la partie infrastructure(CPU, RAM, Disque dur) et la partie applicative(les fichiers jar les serveurs Tomcat, nombre des threads, nombre des classes ...), et finalement on a commencé le développement du API(connecteur) qui va permet de charger la configuration à partir des interfaces Pay Way.

Mots clés : *monétique, switch monétique, système de monitoring, connecteur .*

Abstract

This document is the result of work carried out as part of my final year project at PayLogic in Casablanca, in order to obtain my State Engineering Diploma in Web Engineering and Mobile Computing at the national school of computer science and systems analysiss. The aim of this project was to set up a connector between a payment switch and external monitoring systems.

In order to achieve the objective of this topic, we began with a comparative study of monitoring tools, and after selecting the tools we tested them on Linux and Windows virtual machines.

After testing the tools, we moved on to the real environment, which is Madagascar's public treasury. We monitored the various treasury servers, namely the "PayWay" server, the "Payment(ONLINE)" server and the "Proxy" server.

For all these servers, we monitored the infrastructure (CPU, RAM, hard disk) and the applications (jar files, Tomcat servers, number of threads, number of classes, etc.), and finally began development of the API(connector) that will allow configuration to be loaded from the Pay Way interfaces.

Keywords : electronic payment, electronic payment switch, monitoring system, connector .

Liste des abréviations

API	Application Programming Interfaces
ATM	Automated Teller Machines
GAB	Guichet Automatique Bancaire
GUI	Graphical User Interface
IDA	Integrated Development Environment
IP	Internet Protocol
IT	Information Technology
JAR	Java Archive
JMX	Java Management Extensions
JVM	Java Virtual Machine
NCPA	Nagios Cross-Platform Agent
RAM	Random Access Memory
SSL	Secure Sockets Layer
UML	Unified Modeling Language
WAR	Web Application Archive
XML	Extensible Markup Language

Liste des tableaux

Table des figures

Table des matières

Introduction générale

La monétique est un domaine transversal qui implique la collaboration de différents acteurs, tels que les banques, les opérateurs de téléphonie mobile, les fournisseurs de solutions de paiement, les commerçants et les consommateurs. L'objectif principal de la monétique est de fournir des moyens de paiement simples, sûrs et rapides pour les transactions financières électroniques. Ainsi, les consommateurs peuvent acheter des biens et des services sans utiliser de liquidités physiques, ce qui facilite le processus de paiement et améliore l'expérience d'achat.

La monétique est en constante évolution, car de nouvelles technologies et de nouveaux modes de paiement apparaissent régulièrement. Les technologies de paiement par carte sont devenues courantes depuis plusieurs décennies, mais avec la montée en puissance des smartphones et des applications mobiles, les paiements mobiles sont de plus en plus populaires.

La sécurité est une préoccupation majeure dans le domaine de la monétique. Les réglementations de sécurité sont de plus en plus strictes et les acteurs de l'industrie doivent constamment mettre à jour leurs systèmes de sécurité pour prévenir les fraudes et les piratages informatiques. Les solutions de paiement doivent être conformes à des normes internationales de sécurité.

La monétique est également un domaine économique important. Les transactions financières électroniques génèrent des revenus pour les acteurs de l'industrie et ont un impact sur l'économie mondiale. Les fournisseurs de solutions de paiement, les établissements financiers et les commerçants ont tous un intérêt financier dans le développement et l'adoption de nouvelles technologies de paiement.

La gestion des systèmes de paiement est un domaine critique pour les institutions financières et les entreprises qui gèrent les transactions financières électroniques. Le système de paiement est la principale activité financière et doit être surveillé, analysé et contrôlé pour garantir sa disponibilité, son efficacité, sa sécurité et sa conformité aux réglementations et normes en vigueur. La gestion du système de paiement électronique aide les acteurs du secteur financier à surveiller

toutes leurs transactions électroniques, y compris les transactions en ligne, les paiements mobiles, les paiements par carte de crédit ou de débit, etc. Cela permet également d'identifier tout problème ou violation et de prendre des mesures pour les résoudre. Les entreprises qui gèrent des transactions de monnaie électronique doivent s'assurer que leurs systèmes sont toujours opérationnels et fonctionnent efficacement. Tout problème ou interruption peut entraîner d'énormes pertes financières et nuire à leur réputation. Par conséquent, la gestion du système de paiement est très importante pour la continuité du programme financier.

La gestion des systèmes de paiement aide également les entreprises à protéger les données financières de leurs clients. Les transactions de paiement électronique sont des cibles de choix pour les cybercriminels et les fraudeurs qui cherchent à voler des informations personnelles et financières. Les outils de surveillance du système de paiement peuvent détecter les activités suspectes et les attaques potentielles, et aider les clients à agir rapidement pour protéger leurs systèmes et leurs clients. Les outils de suivi des processus de paiement sont de plus en plus sophistiqués et offrent de nombreuses fonctionnalités différentes. Par exemple, ils peuvent collecter et analyser des données en temps réel, identifier des niveaux et des seuils, créer des rapports et des indicateurs de performance, analyser les vulnérabilités de sécurité et gérer la configuration et les mises à jour. Ces outils permettent aux entreprises de mieux gérer leurs processus de paiement et de s'assurer qu'ils sont conformes aux réglementations et normes en vigueur

Ce rapport, qui est la synthèse du travail réalisé lors du projet, est organisé en quatre chapitres :

- *Le premier chapitre aborde le contexte général du projet. Il présente l'organisme d'accueil et une vue globale sur le projet.*
- *Le deuxième chapitre concerne l'étude et la comparaison des systèmes de monitoring*
- *Le troisième chapitre aborde la présentation des besoins fonctionnles et non fonctionnels ainsi l'analyse et la conception du projet .*
- *Enfin, le quatrième chapitre aborde la réalisation de notre travail.*

Chapitre 1

Contexte général du projet

Dans ce chapitre, nous allons nous intéresser tout d'abord à la présentation du cadre du projet, de l'organisme d'accueil . Nous exposerons ensuite le sujet du travail qui nous a été confié ainsi que l'environnement qui a servi à son achèvement et nous finirons par présenter la problématique.

1.1 Introduction

Cette étape consiste à faire l'étude globale du système envisagé. Pour cela, nous commencerons par décrire l'organisme d'accueil ensuite nous présenterons le cadre et le contexte du projet pour enfin expliquer la méthodologie et la conduite du projet que nous avons suivi.

1.2 Organisme d'accueil

1.2.1 PayLogic

Fondée en 2010, PayLogic est devenue un fournisseur de solutions de paiement électronique à croissance rapide, promouvant l'écosystème des paiements numériques. Grâce à sa culture avant gardiste et à ses pratiques de gestion fondées sur des valeurs fondamentales, l'entreprise est reconnue comme un facilitateur dynamique et innovant offrant des solutions de paiement transparentes, rationalisées et rentables dans de nombreux pays.[1]

Fig. 1.1 : Paylogic-Logo

Leurs actions quotidiennes sont guidées par l'engagement envers le client, l'intégrité, le travail d'équipe, le respect des personnes et la passion de fournir des solutions innovantes, technologiques, sécurisées et évolutives de grande valeur. Grâce à leur large couverture fonctionnelle, à leur architecture ouverte, à leur modularité et à leurs nombreux paramètres, leurs solutions sont utilisées par de nombreux acteurs - banques, opérateurs de télécommunications, commutateurs nationaux/régionaux et autres institutions - dans plus de 50 pays. PayLogic a développé une expertise dans l'ingénierie de la sécurité électronique et a également développé des partenariats avec les leaders mondiaux dans ce domaine, en particulier avec le groupe Thales dont il est le partenaire le plus actif (avec le plus grand nombre de références) en Afrique. elle fournit des solutions d'affaires globales avec

les packages d'ingénierie associés, y compris le développement de solutions intégrées, le déploiement, la personnalisation et la maintenance.

1.2.2 Partenaires mondiaux de Paylogic

PayLogic collabore avec des fournisseurs de matériel et de logiciel leaders dans leurs secteurs pour pouvoir intégrer les dernières technologies, surmonter les complexités, et aider les clients à moderniser ou compléter leur infrastructure afin d'atteindre de meilleures performances commerciales et de gestion.
Les partenariats de PayLogic mettent l'accent sur les ventes conjointes et le développement des affaires. les clients beneficient des expériences communes de l'entreprise dans differents environnements de business exigeants et complexes.
Les solutions de Paylogic sont renfocées par un écosystème composé de puissants partenariats et alliances mondiales ,et suivent une méthodologie de développement Agile, les solutions permettant à leurs clients d'adopter la technologie pour en faire un avantage compétitif sur leur marché[1].
Parmi les Partenaire de Paylogic on peut citer :

- ***Thalesgroup** : Thales est un groupe d'électronique français spécialisé dans l'aérospatiale, la défense, la sécurité et le transport terrestre dont le siège social se situe dans le quartier de La Défense à Paris.*
- ***FINACARD - FINATECH Group** : Finatech Group est un intégrateur technologique de référence ayant vu le jour en 2007 et issu de la volonté du groupe CAPITAL GROUP de se doter d'un bras technologique capable de déployer des solutions à haute valeur ajoutée auprès d'opérateurs et de donneurs d'ordres stratégiques au Maroc et en Afrique*
- ***ITGStore Consulting** : ITGStore est une société de services, d'intégration de services et d'infrastructures IT,il est basée au Cameroun et dispose des agences à Douala, Yaoundé, Niamey et Paris,et opère à la fois sur le marché Camerounais et de la sous-région (Tchad, RCA, Congo, Niger,…) mais également sur le marché des services en Outsourcing et Nearsourcing.*
- ***ORACLE** : Oracle (Oracle Corporation) est une entreprise américaine créée en 1977 par Larry Ellison. Ses produits phares sont le système de gestion de base de données Oracle Database, le serveur d'applications Oracle Weblogic Server, le progiciel de gestion intégré Oracle E-Business Suite et l'offre de cloud computing Oracle Cloud Infrastructure. En 2019, Oracle*

était la deuxième plus grande entreprise de logiciels en matière de chiffre d'affaires et de capitalisation boursière

Fig. 1.2 : Les partenariats de Paylogic

1.2.3 Les clients de PayLogic

Ces compagnies font confiance à Paylogic, des références de toutes tailles, des banques individuelles à de grands groupes bancaires en passant par de grandes institutions en Afrique, au Moyen-Orient[1] :

- *Airtel*
- *Al Hawafiz Computer Devices*
- *AttijariWafa Bank*
- *Bank of Africa – BOA*
- *Bank of Khartoum*
- *Bantek SA*
- *Banque Commerciale du Sahel*
- *Banque de Développement du Mali – BDM*
- *Banque de l'habitat de Côte d'Ivoire*
- *Banque Gabonaise et Française Internationale – BGFI*
- *Banque Marocaine du Commerce Extérieur - BMCE*
- *Banque Nationale du Développement Agricole – BNDA*
- *Barid Bank*
- *Centre Monétique Interbancaire – CMI*
- *CIH Bank*

- *Compagnie Bancaire de l'Afrique Occidentale – CBAO*
- *Ferlo*
- *Future Bank Al Moustaqbal Iraq*
- *Groupe Crédit Agricole du Maroc - CAM*
- *Groupement interbancaire monétique de l'Afrique centrale - GIMAC*
- *Infoset*
- *Invest Bank*
- *Moov*
- *Quds Bank*
- *Société Générale Marocaine de Banques - SGMA*

Fig. 1.3 : Les clients de PayLogic

1.2.4 Historique de Paylogic

2006 *: Paylogic S.A fut fondé sous le nom de Cap Payment System Company.*

2010 *: Cap Payment System Company fut renommé en Paylogic S.A.*

2014 *: En partenaire avec partenaire Thalès E-Security, Paylogic a organisé son premier séminaire informatique ayant pour but de couvrir les dernières innovations dans le domaine de sécurité de paiements électronique élargissant ainsi les réseaux des chefs d'entreprise. Dans la même année, l'entreprise a organisé un second séminaire sur la sécurité des paiements informatique où elle a le plaisir de rencontrer ses clients et partenaires.*

***2014** : Quds Bank signe un accord avec Paylogic pour l'achat et l'exploitation d'un système de guichets automatiques. Cela permet à la banque de pouvoir procurer à ses clients des cartes guichet*

***2014** : La société marocaine PAYLOGIC a remporté l'appel d'offres lancé par la Banque des États d'Afrique Centrale pour la région CEMAC concernant la mise en place du système central des incidents de paiement*

***2015** : Sur la 4ème édition du Prix de l'innovation pour l'Afrique, grâce à sa plateforme panafricaine de paiement mobile, Paylogic a remporté le deuxième prix du prix de l'innovation.*

***2019** : PayLogic a remporté le prix de l'innovation de la meilleure solution de paiement électronique à l'Africa Pay ID Expo 2019.*

1.2.5 Présentation de l'activité de l'entreprise

1.2.5.1 Domaine d'activité :

Grâce à sa grande expérience, Paylogic opère sur divers domaines d'activités dont un leader en monétique. Parmi ses activités, on cite :

- *E-sécurité*
- *Emission Instantanée*
- *Solution de Paiement*

1.2.5.2 Solutions Proposées :

Paylogic est un intégrateur et éditeur de logiciel monétique offrant à la fois des fonctionnalités et solutions aussi bien back-office que front-office.

- *des solutions de paiement électronique qui comprend une solution ATM / POS.*
- *des solutions d'émission de cartes de crédit et de débit.*
- *des solutions de cartes à puce entièrement conformes à la norme EMV.*
- *des solutions de gestion des alertes.*
- *des systèmes d'aide à la décision et à la conformité de cartes de crédit.*
- *L'organisation de services de PayLogic assure l'installation, la gestion de projet et l'assistance post-installation pour tous les produits PayLogic.*

1.2.6 Les produits de Paylogic :

en plus de 13 ans d'expérience, PayLogic a continuellement fourni des solutions de paiement électronique modernes qui ont renforcé l'infrastructure technique des entreprises et les ont aidées à être des leaders dans leur domaine d'activité. PayLogic propose quatre services principaux :

1.2.6.1 PAYWAY

PayLogic fournit une solution de paiement électronique unifiée, complète, sous le nom de PayWay. Construite sur une plateforme Oracle, avec l'aide de la technologie XML et couplée à une architecture multiserveurs distribuée, PayWay est conçue pour être exécutée par de gros clients ainsi que par des clients qui agissent dans des marchés très spécifiques. PayWay est implémenté en Java pour les raisons suivantes :

- ✓ *Java offre une forte robustesse aux applications d'entreprise grâce à des fonctionnalités linguistiques telles que la gestion des exceptions, la gestion de la mémoire et la protection contre les pointeurs de mémoire.*
- ✓ *Java fournit des fonctionnalités multithreading et désynchronisées, positionnant PayWay pour d'excellentes performances sur des systèmes multiprocesseurs.*
- ✓ *Java offre une syntaxe de language relativement simple comparée aux languages comme C++, ce qui est souhaitable lors de la construction d'applications volumineuses et complexes, offrant une possibilité de maintenance plus intéressante et moins couteuse.*
- ✓ *Java est un langage multiplateformes facilitant l'installation de migrations et de correctifs.*

Fig. 1.4 : PayWay

1.2.6.2 Architecture PayWay

PaySwitch est le pivot central de l'échange de transactions et de messages. L'élément principal qui permet à cette technologie de mener à bien les transactions est la fonction de routage, de commutation et de gestion des transactions ainsi que des autorisations passées entre les différentes applications. PaySwitch est conçu dans le but d'être robuste, performant, évolutif et sécurisé. Le module intègre :

- ✓ *Une Interface de Canaux (Paiement en ligne, Paiement par mobile)*
- ✓ *Un Conducteur de Terminal (ATM, POS, Kiosk)*
- ✓ *Une Application complète alliant les éléments ci-dessus.*

La business logique de PaySwitch est implémentée à un niveau compétitif via une modularité flexible et extensible, composée de PayCore, PayAdmin, PayCard, PayIssue, PayAcq, PaySecure, PayCSM…

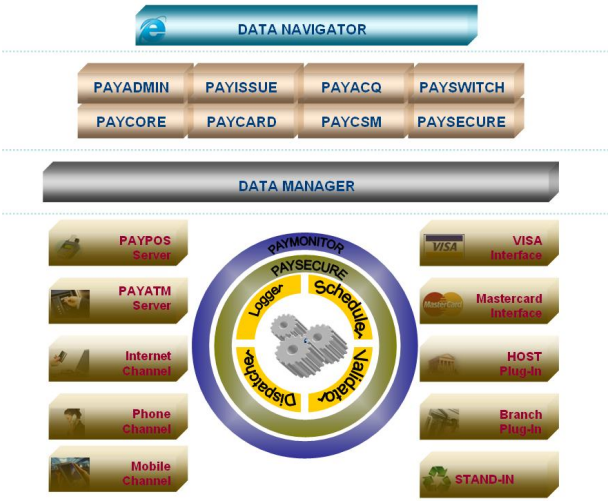

Fig. 1.5 : PayWay architecture

Caractéristiques clés de PaySwitch : *PaySwitch assure l'intégrité des transactions à l'aide de fonctionnalités de traitement transactionnelles et de conception d'application tolérantes aux pannes, garantissant ainsi un réseau fiable et efficace à la disposition des utilisateurs 24 heures sur 24, 7 jours sur 7. En fournissant un moteur à base de règles pour la commutation et le routage basé sur le type de carte, la source de la transaction, le type de transaction, les différents canaux disponibles et prenant en charge tous les types de demandes d'autorisation entrantes et sortantes.*
Définition de chemin de traitement dynamique Prise en charge des protocoles d'application réseau courants ainsi que tous les types de messages. Permet l'intégration de plusieurs systèmes (Visa, MasterCard, EuroPay, POS) sur une plate-forme stable. Fournit une plate-forme commune pour le développement d'applications et agissant en tant que principal contrôleur de traitements.
Entièrement intégré au module de traitement d'autorisation autonome PayWay avec ses fonctionnalités de stockage et de retransmission.
Offre un routage sophistiqué des messages inter procédés basé sur les protocoles ISO / TCP, des installations permettant d'étendre et d'améliorer facilement le réseau, la gestion des communications de données, la surveillance réseau avancée et des capacités complètes de gestion d'événements systèmes. Offre des fonctionnalités complètes de gestion et de journalisation de messages d'alertes ou d'évènements.
Fournit de divers services qui facilitent les opérations réseaux. Permet aux opérateurs de définir et de contrôler l'endroit où les messages d'alertes système des processus d'application sont routés, enregistrés et présentés Fournit des installations pour l'exécution et la gestion d'applications basées sur Java. Fournit une plate-forme avec tolérance de pannes ainsi que de multiples options de configuration et de capacités d'extension du système. Prend en charge l'autorisa-

tion/vérification des cartes de crédit, de débit, prépayées et à puce en utilisant une combinaison de contrôles négatif, de contrôles de vitesse d'utilisation, de contrôles d'identification, d'authentification positifs et de contrôles de solde de compte. Fournit les statistiques intégrées, requêtes et contrôles de surveillance.

1.2.7 Application mobile AMA

AMA est un projet financé par PayLogic qui a été lancé en novembre 2012 afin de développer une plateforme panafricaine instantanée, gratuite et sécurisée pour le paiement mobile.

AMA est une alliance entre les banques, les opérateurs financiers et les opérateurs de télécommunications. Cette plateforme permet de connecter les clients et les commerçants de tout le continent africain via leurs téléphones mobiles uniquement.

AMA vise à utiliser des technologies avancées pour fournir une solution aux problèmes d'inclusion financière en offrant une gamme de services bancaires de paiement à faible coût, plus sécurisés et accessibles à tous.

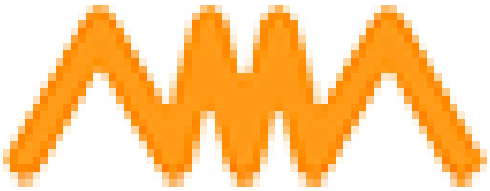

Fig. 1.6 : AMA

1.2.7.1 Ecosystème AMA

L'écosystème AMA comprend :

- ✓ *Les banques : Clients, Marchands, Agences, GAB*
- ✓ *Les opérateurs financiers : Clients, Agences*
- ✓ *Les opérateurs télecoms : Clients, Marchands, Points de vente, Agences, Kiosques*

1.2.7.2 Sécurité AMA

AMA garantit des mécanismes d'authentification puissants basés sur le protocole de sécurité OAuth, garantissant une sécurité lors de l'accès au compte d'utilisateur. AMA utilise des mécanismes de cryptage et de tokenization. Vos données sont stockées dans un environnement hautement sécurisé respectant la norme PCI-DSS. Vos transactions sont cryptées via HTTPS.

1.2.8 E-Security

PayLogic est le partenaire de Thales e-Security au Maroc et en Afrique pour les produits et services de Thales e-Security comprenant des solutions de sécurité de paiement et d'application. Thales est un leader mondial des systèmes d'information critiques et un acteur clé de l'assurance de la sécurité des citoyens, des infrastructures et des nations.

PayLogic est habilité à fournir tous les services de Thales e-Security et a obtenu les niveaux d'accréditation nécessaires pour développer et prendre en charge les solutions de Thales e-Security.

Fig. 1.7 : E-Security

1.2.8.1 Module de sécurité matérielle

Module de sécurité matérielle pour les applications de paiement :

- ✓ *PayShield9000*
- ✓ *Dispositif de gestion de clés*
- ✓ *Gestionnaire HSM distant*

1.2.8.2 Gestion de l'identité

La solution d'authentification et de sécurité concernant les transactions des utilisateurs d'Identity Management qu'elle utilise est :

SafeSign

- *Cette solution permet aux institutions d'authentifier en toute sécurité les identités des utilisateurs et de signer numériquement les transactions commerciales.*

- *SafeSign est conçu pour être interopérable avec les infrastructures existantes.*

1.2.8.3 Appareils de cryptage réseau

Il existe trois types de produits pour le cryptage réseau :

- ✓ *Datacryptor IP Network Encryption*
- ✓ *Datacryptor Link and Layer 2 Encryption*
- ✓ *Datacryptor High Assurance Government Encryption*

Datacryptor fournit une sécurité périmétrique forte, certifiée pour une large gamme d'architectures réseau. Datacryptor offre une résistance à la fraude à la fois robuste et qui fonctionne à des vitesses de réseau natives sans latence excessive. En outre, il s'intègre facilement avec les anciens ainsi que les nouveaux réseaux.

1.2.8.4 Gestion de la clé de cryptage

Systèmes de gestion de clés de cryptage : bandes physiques CryptoStor. Ce système de gestion de clé de chiffrement compresse et chiffre les données lors de leur enregistrement sur bandes physique. Il optimise la sécurité des données enregistrées et stockées sur bande en automatisant les clés de sécurité du système de gestion utilisées sans nuire aux performances du système existant.

1.2.8.5 Appareils d'horodatage

Les produits d'horodatage sont :

- ✓ *Horloge Principal - Heure Source*
- ✓ *Serveur d'horodatage*

Ils garantissent l'exactitude de la date et de l'heure de la création ou de la modification des documents numériques. Ils protègent les clés d'horodatage à l'aide d'un matériel inviolable indépendamment certifié, contrairement aux systèmes logiciels dans lesquels les administrateurs peuvent facilement manipuler les valeurs de temps.

1.2.9 Instant card issuing

L'Instant Card Issuing est le système d'émission de cartes de crédit à puce EMV dans un environnement distribué. L'Instant

Card Issuing permet à la banque de fournir à ses clients un service supplémentaire qui est l'émission instantanée des cartes en agence. Ces cartes peuvent être émises dans les agences bancaires, les centres commerciaux, les clubs sportifs, les universités et les établissements d'enseignement, les administrations, les bases militaires, etc.

Fig. 1.8 : Instant Card Issuing

1.2.9.1 Type de cartes émises

- *Visa*
  - ✓ *Débit : Electron*
  - ✓ *Crédit : Classic, Gold, Platinum, Business*
  - ✓ *Crédit : Classic, Gold, Platinum, Business*
- *MasterCard*
  - ✓ *Débit : Maestro*
  - ✓ *Crédit : Classic, Gold, Platinum, Business*
  - ✓ *Sans contact : PayPass, Mifare*
- *Cartes privées*

1.2.10 Les services de PayLogic

1.2.10.1 section Académie de formation de PayLogic

Les cours de l'Académie de formation PayLogic sont conçus comme des cours pratiques ou « boîte à outils » destinés aux praticiens débutants á expérimentés, ainsi qu'aux agents certifiés EFT souhaitant renforcer leurs compétences déja acquises. La PayLogic Academy offre une possibilité de formation spéciale aux praticiens qui souhaitent se familiariser avec les nouveaux outils EFT

ainsi que se refamiliariser avec les anciens. Les cours sont dispensés dans les mêmes conditions et apportent la même expérience que dans le monde réel, ce qui fait d'une personne qui a completé le PayLogic Training, un acteur de premier plan dans ce domaine. leurs cours comprennent :

- ✓ *Un Contenu Centré sur les Sujets Addressés*
- ✓ *Des Formateurs Expérimentés et Compétents*
- ✓ *Des Prix Compétitifs*
- ✓ *Un Environnement Optimal*
- ✓ *Une Grande Facilité d'Accès*
- ✓ *Un Matériel d'Apprentissage de Bonne Qualité*

1.2.10.2 Integration

PayLogic garantit à ses clients l'aide nécessaire pour ancrer leurs projets dans la réalité et rentabiliser rapidement leur investissement en offrant à chacun d'entre eux le meilleur processus d'intégration possible. Nous mettons á votre disposition une équipe solide combinant expertise et méthodologie dans la gestion de projets d'intégration. PayLogic assure aussi une formation hautement qualifiée ainsi qu'un transfert de compétences afin de permettre au client d'avoir á sa disposition une équipe autonome pour l'administration et le bon fonctionnement des solutions implémentées et suggérées.

1.2.10.3 Support 24/7

Chez PayLogic, nous nous sommes dévoués à vous fournir un programme de support technique qualifié. Nos ingénieurs et experts en support techniques sont prêts à répondre à vos demandes et á vous aider à résoudre vos problèmes. PayLogic Support vous fournit un accès illimité à nos équipes d'experts. Nos ingénieurs qualifiés, répondront à vos questions et vous assisteront 24 heurs sur 24 et 7 jours sur 7 via notre service d'assistance dédié au suivi de vos requêtes. Une assistance téléphonique et par mail est disponible de 8h30 á 18h30, du Lundi au Vendredi (Hors Jours Fériés Marocains)

1.2.11 Organigramme de Paylogic

Le sujet de stage de fin d'étude est une solution dont le département technique en est principalement responsable.Il a pour rôle de créer, suivre, monitorer ainsi qu'assurer sa réalisation. Afin, d'éclaircir plus la structuration de l'entreprise, on représente donc son organigramme sous le schéma suivant :

Fig. 1.9 : Organigramme de Paylogic .

1.3 Présentation du sujet

1.3.1 Introduction

Dans le domaine des transactions financières électroniques, la monétique joue un rôle crucial en assurant la gestion sécurisée et efficace des paiements. Les systèmes de paiement modernes reposent sur des infrastructures complexes qui facilitent des opérations rapides et fiables. Dans ce contexte, l'objectif du projet présenté est de concevoir et de développer un connecteur afin de lier le switch monètique avec les systèmes de monitoring externe.

Le switch monétique est un élément essentiel de l'architecture monétique, responsable de l'interconnexion entre différents acteurs tels que les commerçants, les banques et les prestataires de services de paiement. Il assure le routage sécurisé des données et facilite la communication entre ces parties prenantes, garantissant ainsi le bon déroulement des opérations financières.

L'innovation clé de ce projet réside dans l'intégration de systèmes de monitoring externes au switch monétique. Ces systèmes permettront de surveiller en temps réel les performances, la disponibilité et la sécurité .

Ce projet s'inscrit dans une perspective d'innovation technologique et de réponse aux besoins croissants du secteur des paiements électroniques. En combinant la puissance du switch monétique avec des systèmes de monitoring externes, il vise à améliorer la sécurité, l'efficacité et la confiance dans les transactions

financières.

1.3.2 Description du sujet et problématique

Les switchs monétiques sont des dispositifs critiques utilisés dans les environnements financiers pour le traitement sécurisé des transactions électroniques. Ils jouent un rôle essentiel en assurant la gestion et le contrôle des flux de paiement. Afin de garantir leur bon fonctionnement, il est nécessaire de surveiller en temps réel leurs performances, leur disponibilité et leur sécurité. Actuellement, les systèmes de monitoring tels que Nagios XI et Zabbix sont largement utilisés pour surveiller et gérer les infrastructures informatiques. Cependant, l'intégration de ces systèmes avec les switches monétiques peut représenter un défi technique. En effet, les switches monétiques des grands applications fonctionnant sur plusieurs serveur, rendant difficile leur intégration avec les outils de monitoring standard.

La conception et le développement d'un connecteur spécifique pour les switches monétiques permet de résoudre cette problématique. Ce connecteur utilise les configurations saisies au niveau des interfaces de switch monétique (Payway) pour les envoyer au systèmes de monitoring externes (nagios,zabbix....) De plus, en offrant la possibilité aux clients et a l'équipe PayLogic de configurer les paramètres de surveillance à travers le switch monètique, l'adoption et la gestion des switches monétiques seraient grandement simplifiées.
Les utilisateurs pourraient facilement configurer les hostes et les différents métrique à mésurer (CPU, RAM, disk dur ...).

Alors ***comment peut on centraliser la configuration des systèmes de monitoring à travers les interfaces du switch monétique ?***

1.3.3 Solution proposée

Comme nous l'avons vu dans la section précédente notre objectif est de centralise la configuration des systèmes de monitoring à travers les interfaces de switch monétique ,donc mon solution proposée est de faire la conception et le développement d'une API (connecteur) pour lier le switch monétique avec les systèmes de monitoring externes.

1.4 Méthodologie du projet

1.4.1 Présentation du développement en cascade

Le modèle de développement en cascade est l'un des modèles les plus anciens et les plus traditionnels du cycle de vie du développement logiciel. Il suit une approche séquentielle et linéaire, où chaque phase est réalisée dans un ordre prédéfini et les activités ne peuvent commencer que lorsque la phase précédente est terminée. Voici une description plus générale de chaque phase du modèle en cascade[2] :

- *Analyse des besoins : Dans cette phase initiale, les exigences du projet sont collectées en travaillant avec les parties prenantes, les utilisateurs finaux et les experts du domaine. L'objectif est de comprendre les besoins fonctionnels et non fonctionnels du système à développer.*
- *Conception : Une fois que les exigences sont bien comprises, la phase de conception commence. Elle consiste à concevoir l'architecture globale, à décomposer le système en sous-systèmes, à définir les interfaces et à déterminer les relations entre les différents modules. Cette phase comprend également la conception détaillée des différents composants du système.*
- *Développement : Après avoir achevé la phase de conception, l'équipe de développement passe à la phase de développement réel. Les programmeurs écrivent le code source du logiciel en utilisant les langages de programmation appropriés. Les modules ou les fonctionnalités sont développés et intégrés conformément à la conception établie.*
- *Tests : Une fois le développement terminé, la phase de tests commence. Elle vise à vérifier si le logiciel fonctionne conformément aux exigences spécifiées. Les tests incluent des tests unitaires pour vérifier chaque composant individuellement, des tests d'intégration pour vérifier l'interaction entre les composants, des tests système pour évaluer le fonctionnement global du système, et des tests de validation pour s'assurer que le logiciel répond aux besoins des utilisateurs finaux.*
- *Déploiement : Après avoir réussi les tests, le logiciel est prêt à être déployé dans l'environnement de production. Cette phase comprend l'installation du logiciel, la configuration du système, la migration des données et la formation des utilisateurs. Le logiciel est mis à la disposition des utilisateurs finaux pour une utilisation opérationnelle.*

- *Maintenance : Une fois le logiciel déployé, il entre dans la phase de maintenance. Cette phase comprend la gestion des problèmes, les correctifs de bugs, les mises à jour des fonctionnalités et les améliorations continues en fonction des besoins des utilisateurs et des évolutions technologiques. La maintenance vise à assurer le bon fonctionnement et la pérennité du logiciel.*

On a choisie le modèle en cascade car il est simple et facile à comprendre. Il convient particulièrement aux projets où les exigences sont clairement définies et stables, et les changements ultérieurs sont peu probables. Cependant, il peut être moins adapté aux projets complexes ou innovants, où les besoins peuvent évoluer rapidement et nécessitent une rétroaction continue entre les phases.

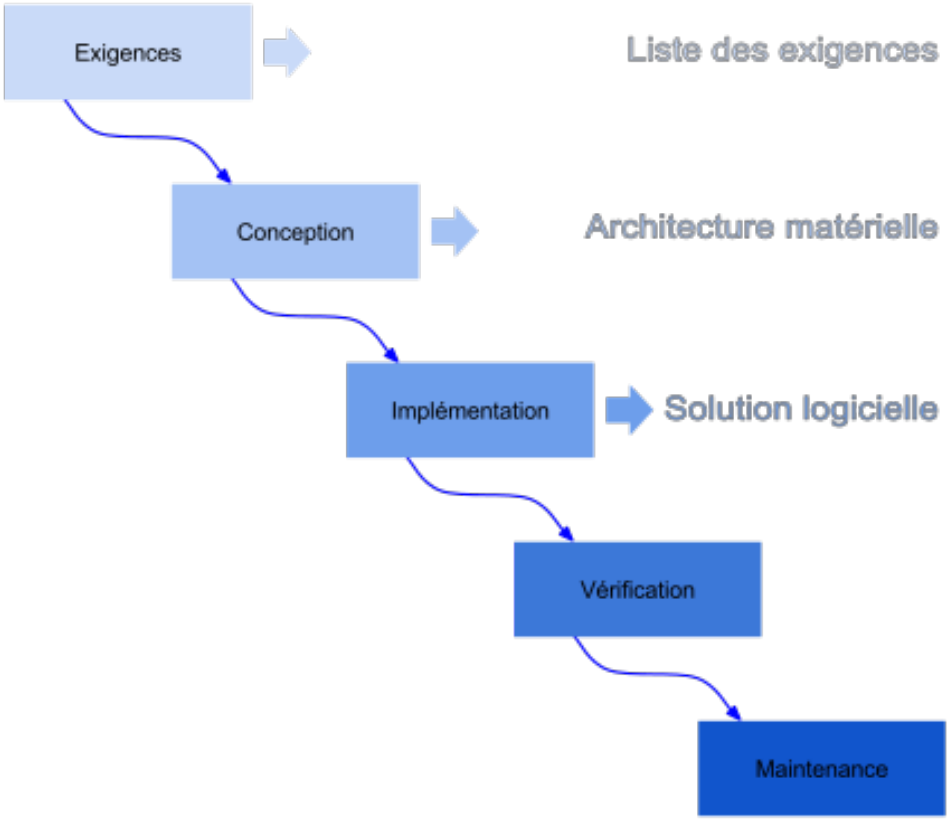

Fig. 1.10 : développement en cascade.

1.4.2 Planification du projet

La planification est une étape essentielle dans tout projet de développement, aussi importante que le processus de développement lui-même. Elle permet de définir les différentes phases du projet de manière séquentielle et de respecter les délais fixés.

Dans notre cas, la planification étant déjà terminée, il convient maintenant de traiter la question des coûts, c'est-à-dire le temps nécessaire pour chaque tâche. Le meilleur outil pour représenter cette situation est le diagramme de Gantt[3].

Le diagramme de Gantt est un outil graphique qui permet de visualiser la progression du projet en mettant en évidence les différentes tâches et leur positionnement temporel par rapport à l'ensemble du projet. Il facilite le suivi du projet et offre une vision claire de son avancement.

Entre la planification et la réalisation, entre la théorie et la pratique, le diagramme de PERT nous permet de définir des intervalles de réalisation et de progression du projet. Quand le diagramme de Gantt, il nous permet de suivre le temps réellement écoulé lors de la réalisation du projet.

Compte tenu du cycle de développement en cascade, le développement est représenté sous la forme de plusieurs étapes, réparties sur les périodes d'analyse, de conception et de réalisation. Sur le diagramme ci-dessous, vous pouvez observer les tâches réalisées en fonction de leur durée.

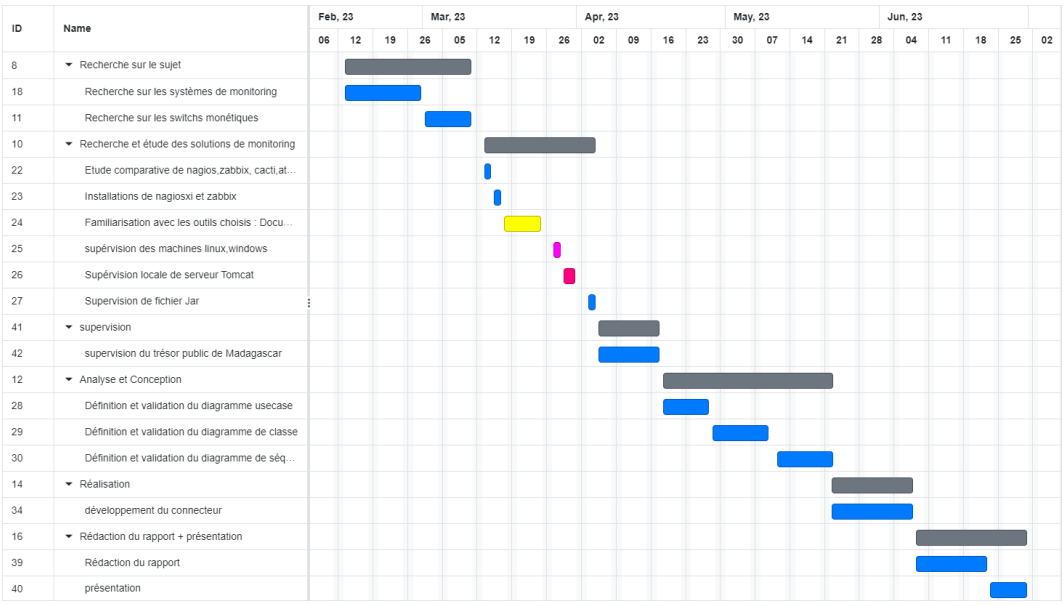

Fig. 1.11 : diagramme de gantt

1.5 Conclusion

Ce chapitre nous a permis de cerner le contexte général du projet. Après avoir présenté le groupe PayLogic, son secteur d'activités, ses produits et son approche, nous avons délimité notre projet en citant son contexte qui donne une idée générale sur le métier de la monétique. Nous avons également listé les différents outils de conduite de projet que nous avons utilisé et le processus du développement adopté. Le chapitre suivant sera consacré à l'étude des systèmes de monitoring .

Chapitre 2

Étude des systèmes de monitoring

Ce chapitre se concentre sur la définition précise du concept de supervision. Dans notre démarche, nous allons effectuer une étude comparative approfondie des différents outils de supervision disponibles. Nous examinerons leurs fonctionnalités, leurs avantages et leurs limitations, afin d'évaluer leur pertinence par rapport à notre contexte spécifique. Cette analyse comparative nous permettra d'identifier l'outil qui répond le mieux à nos besoins et à nos objectifs de supervision.

Une fois que nous aurons passé en revue les différentes options, nous serons en mesure de préciser notre choix final concernant l'outil de supervision retenu. Nous prendrons en compte des critères tels que la convivialité, la flexibilité, la compatibilité avec nos systèmes existants, ainsi que la disponibilité d'un support technique fiable. Cette décision sera fondamentale pour assurer le succès de notre projet de supervision et pour optimiser notre efficacité opérationnelle.

2.1 Introduction

Actuellement, les systèmes d'information jouent un rôle de plus en plus crucial dans les entreprises, mais ils deviennent également de plus en plus complexes. La maintenance et la gestion de ces systèmes sont devenues une priorité urgente. Afin de répondre à ce besoin, plusieurs logiciels de surveillance et de supervision de réseaux ont été développés. Leur objectif est de vérifier en temps réel l'état du réseau et de signaler rapidement tout incident éventuel. Ces outils permettent de réduire considérablement les délais d'intervention et de prendre immédiatement en charge les anomalies sans que les utilisateurs du réseau ne soient affectés ou remarquent des erreurs.

2.2 Principe de la supervision

2.2.1 Définition

La surveillance est définie comme une technique qui tire le meilleur parti des ressources informatiques pour obtenir des informations sur l'état des réseaux et de leurs composants. Ces données seront traitées et affichées pour faire la lumière sur les problèmes potentiels. La surveillance peut résoudre automatiquement les problèmes ou avertir les administrateurs via un système d'alerte (email ou SMS, par exemple).
Ce processus de supervision est effectué à différents niveaux dans un parc de machines : au niveau de l'interconnexion des machines (réseau), au niveau de chaque machine individuelle (système) et au niveau des services fournis par chaque machine (applications).[4]

- ***Supervision réseau :****Le terme "réseau" désigne ici l'aspect de la communication entre les machines. Le but de la supervision est de garantir le bon fonctionnement des communications et de mesurer les performances des liens (débit, latence, taux d'erreurs). Dans ce contexte, il est possible de vérifier si une adresse IP est toujours accessible, si un port spécifique est ouvert sur une machine donnée, ou encore d'analyser les statistiques de latence sur le lien réseau[18].*

- ***Supervision système :*** *Dans ce cas, la surveillance se concentre sur la machine elle-même, et en particulier sur ses ressources.*

Par exemple, il est possible de contrôler l'utilisation de la mémoire ou la charge du processeur sur le serveur, ou encore d'analyser les fichiers journaux système[18].

- ***Supervision applicative :*** *Cette technique est plus avancée, car elle permet de vérifier le bon fonctionnement d'une application lancée sur une machine. Par exemple, il est possible de tester la connexion sur le port de l'application pour vérifier si elle retourne ou demande les bonnes informations, ainsi que d'analyser les fichiers journaux d'application pour détecter d'éventuels problèmes[18].*

2.2.2 L'importance de la supervision

La supervision est un concept essentiel dans de nombreux domaines, allant de l'informatique à l'industrie en passant par la santé et les télécommunications. Elle se réfère à un processus de surveillance et de contrôle continu visant à garantir le bon fonctionnement d'un système, d'un processus ou d'une activité. Le principe fondamental de la supervision est de collecter des données en temps réel, de les analyser et de prendre des mesures appropriées pour maintenir ou améliorer les performances et la stabilité du système supervisé[4].

Le processus de supervision repose sur plusieurs éléments clés. Tout d'abord, il est crucial de définir les objectifs de supervision, c'est-à-dire les critères et les normes selon lesquels le système doit être évalué. Ces objectifs peuvent inclure des mesures de performance, de qualité, de disponibilité, de sécurité, etc. Ils fournissent un cadre pour évaluer l'état du système et prendre des décisions informées.

Ensuite, la supervision implique la collecte de données pertinentes à partir des hôtes, d'instruments ou d'autres sources de données. Ces données peuvent être des mesures physiques telles que la température, la pression, le débit, ou des données issues de systèmes informatiques tels que des journaux d'événements, des métriques de performance, etc[19].

Une fois les données collectées, elles sont analysées et interprétées pour évaluer l'état du système. Cela peut être fait de manière automatisée à l'aide d'algorithmes et de techniques d'apprentissage automatique, ou par des opérateurs humains spécialisés. L'analyse des données peut permettre de détecter des anomalies, des tendances ou des schémas qui pourraient indi-

quer des problèmes potentiels ou des opportunités d'amélioration.

En fonction des résultats de l'analyse, des actions sont prises pour corriger ou prévenir les problèmes identifiés. Ces actions peuvent être prises automatiquement par des systèmes de contrôle-commande ou manuellement par des opérateurs formés. Par exemple, en cas de détection d'une panne imminente, des alertes peuvent être générées pour avertir les responsables et des procédures de sauvegarde peuvent être enclenchées pour éviter une interruption de service[19].

2.3 Étude comparative des systèmes de monitoring

Dans cette partie, nous allons commencer une étude de certains outils de surveillance afin de décider et de choisir celui qui convient le mieux dans notre cas.

2.3.1 Cacti

Cacti est un outil de surveillance de réseau et de création de graphiques basés sur le web, à code source ouvert. Il a été conçu pour permettre aux utilisateurs d'interroger des services à des intervalles prédéterminés et de représenter graphiquement les données obtenues. L'application frontale de Cacti facilite la collecte des données et la création de graphiques, offrant ainsi une solution conviviale pour la surveillance des réseaux[5].

- ✓ ***Avantages***
  - *Configuration : Avec l'utilisation des templates pour les machines, les graphiques, et la récupération des données tout se configure aisément et entièrement via l'interface web[20]*
  - *Gestion des utilisateurs : permet de contrôler l'accès et les autorisations des différents utilisateurs de l'application.*
  - *facilité d'utilisation et son interface conviviale : Avec son interface basée sur le web, Cacti offre une expérience utilisateur intuitive, ce qui permet aux utilisateurs de configurer et de gérer facilement leur système de surveillance de réseau.*
- ✓ ***Inconvinients***

- *Support communautaire limité : Bien que Cacti ait une communauté d'utilisateurs active, le support peut parfois être limité par rapport à des solutions commerciales ou à des outils de surveillance de réseau plus largement adoptés.*
- *Dépendance aux plugins tiers : Bien que Cacti offre de nombreuses fonctionnalités de base, certaines fonctionnalités avancées peuvent nécessiter l'installation de plugins tiers.*
- *Cacti se dispose d'une API que vous pouvez utiliser dans votre projet mais la documentation de l'API n'est pas bien expliquée.*

2.3.2 Atera

Atera est un outil de surveillance disponible sous la forme de l'un des deux systèmes SaaS, l'automatisation des services professionnels (PSA) et la surveillance et la gestion à distance (RMM). Les fournisseurs de services gérés et les équipes informatiques utilisent la solution Atera basée sur des agents pour surveiller un nombre illimité de terminaux matériels et logiciels et résoudre les problèmes système. Les scripts RMM peuvent surveiller les connexions à distance, gérer les correctifs, installer des logiciels, orchestrer la réponse aux incidents, etc.[6]

✓ ***Avantages***

- *Surveillance et alertes en temps réel pour les ressources système, le comportement des utilisateurs, le flux de trafic réseau/IP,...*
- *Identification instantanée des appareils non gérés*
- *Patchs de mise à jour logicielle automatisés*
- *Tâches d'administration et de maintenance préconfigurées, ainsi que scripts personnalisés*

✓ ***Inconvinients***

- *Tableaux de bord, alertes, rapports et fonctionnalités d'automatisation difficiles à personnaliser.*
- *Atera offre une api mais les fonctionnalités que vous pouvez réaliser a travers l'api sont limités est n'est pas bien documenté.*

2.3.3 Zabbix

ZABBIX est un logiciel libre permettant de surveiller l'état de divers services réseau, serveurs et autres matériels réseau et pro-

duisant des graphiques dynamiques de consommation des ressources. C'est un logiciel créé par Alexei Vladishev[7].

✓ ***Avantages***

- *excellente flexibilité et évolutivité.*
- *surveillance facile de tout type de fournisseur ou d'appareil[20].*
- *alerte sur les « situations exceptionnelles » liées à un « problème spécifique ».*
- *très grande communauté.*
- *L'API de zabbix vous permet de gérer toutes les fonctionnalités offertent par l'interface zabbix.*

✓ ***Inconvinients***

- *Toutes les données de surveillance sont stockées dans une base de données, ce qui nécessite d'allouer des capacités informatiques supplémentaires pour gérer une base de données dans les grands réseaux.*

2.3.4 Nagios XI

Nagios est un outil de surveillance système vétéran : Nagios Core est l'option open source et Nagios XI est une solution d'entreprise commerciale. On peut utiliser Nagios pour surveiller de manière centralisée les réseaux, les serveurs, les applications et d'autres composants du système, en obtenant une visibilité unique sur l'état de l'environnement informatique via une interface Web[20].
Nagios propose également des plugins qui vous permettent d'étendre son architecture de base pour répondre à vos besoins uniques de surveillance du système. Son déploiement hautement disponible permet une surveillance continue du système[8].

✓ ***Avantages***

- *Capacités de création de rapports étendues et extensibles*
- *Détecte rapidement les pannes, émettant automatiquement des alertes*
- *Nagios XI offre une interface conviviale, facile à naviguer et à configurer, ce qui le rend adapté aux utilisateurs ayant différents niveaux d'expertise technique.*
- *Accès complet au code (open source)*

✓ ***Inconvinients***

- *La version open-source gratuite est limitée en fonctionnalités*

- *Plusieurs modules complémentaires requis pour des fonctionnalités complètes.*
- *Nagios xi depend des plugins mais la majorité des fonctionnalités ne nécessite pas des plugins sont implémenté dans Nagios serveur (mesure RAM, CPU...).*

2.4 Choix de l'outil de supervision

Il existe de nombreux systèmes de monitoring disponibles, mais après avoir soigneusement évalué nos options, nous avons décidé d'opter pour Nagios et Zabbix. Plusieurs raisons ont motivé notre choix. Tout d'abord, ces deux outils jouissent d'une solide réputation dans le domaine du monitoring. Ils ont été utilisés et testés par de nombreuses entreprises et organisations, ce qui témoigne de leur fiabilité et de leur efficacité.

En outre, Nagios et Zabbix bénéficient d'une vaste communauté d'utilisateurs. Cette communauté active et engagée offre un soutien précieux, que ce soit sous forme de documentation, de forums d'entraide ou de ressources supplémentaires. La possibilité de partager des connaissances et d'échanger des idées avec d'autres experts du monitoring est un avantage considérable.

En choisissant Nagios et Zabbix, nous avons également pris en compte leur extensibilité et leur flexibilité. Ces outils offrent des fonctionnalités avancées et une architecture modulaire, ce qui nous permettra de personnaliser notre système de monitoring en fonction de nos besoins spécifiques. De plus, leur capacité à surveiller une grande variété de plateformes et de dispositifs nous assure une visibilité complète sur notre infrastructure, qu'il s'agisse de serveurs, de réseaux ou d'applications.

2.5 Étude des systèmes choisis

2.5.1 Nagios xi

Nagios est un puissant outil de surveillance et de gestion des systèmes qui permet de surveiller en temps réel l'état et les performances de divers composants d'une infrastructure informatique. on'ai utilisé Nagios XI, la version commerciale de Nagios, avec une licence gratuite qui offre des fonctionnalités limitées . Nagios XI, par rapport à Nagios Core, offre une interface utilisateur graphique conviviale qui simplifie la configuration et la gestion de la surveillance. Avec Nagios XI, on'ai pu profiter de fonc-

tionnalités avancées telles que des tableaux de bord personnalisables, des graphiques, des rapports détaillés, la planification de la maintenance, la gestion des incidents et des utilisateurs, ainsi que des intégrations avec d'autres outils. Cette licence gratuite m'a permis d'explorer et d'exploiter certaines fonctionnalités de Nagios XI, offrant une meilleure expérience utilisateur par rapport à Nagios Core, tout en prenant en compte les limitations de la version gratuite[9].

2.5.1.1 Architecture de Nagios xi

NagiosXI fonctionne sur une architecture basée sur un modèle client/serveur. Cette architecture repose sur trois composantes principales : le scheduler, l'interface graphique utilisateur (GUI) et les plugins[9].

- ***Le scheduler** :*
  est une partie du serveur Nagios, joue un rôle essentiel. Il vérifie régulièrement les plug-ins à des intervalles définis et effectue des actions en fonction des résultats de ces vérifications. Il s'assure ainsi que les services et les ressources surveillés sont constamment évalués et que les actions appropriées sont prises en cas de problèmes détectés[21].
- ***La GUI** :*
  est l'interface de Nagios qui s'affiche dans une page web générée par l'interface Common Gateway Interface (CGI). Cette interface offre aux utilisateurs la possibilité de visualiser et de gérer les configurations, les alertes, les boutons d'état (vert pour OK, rouge pour erreur...), les graphiques , et bien d'autres éléments liés à la surveillance[21].
- ***Les plugins** :*
  constituent une partie essentielle de NagiosXI, et ils sont configurables par l'utilisateur. Ils sont chargés de vérifier différents services ou ressources spécifiques et de renvoyer les résultats au serveur Nagios. Ces résultats permettent au serveur de prendre les actions correspondantes, comme générer des alertes, envoyer des notifications ou effectuer des actions correctives[21].

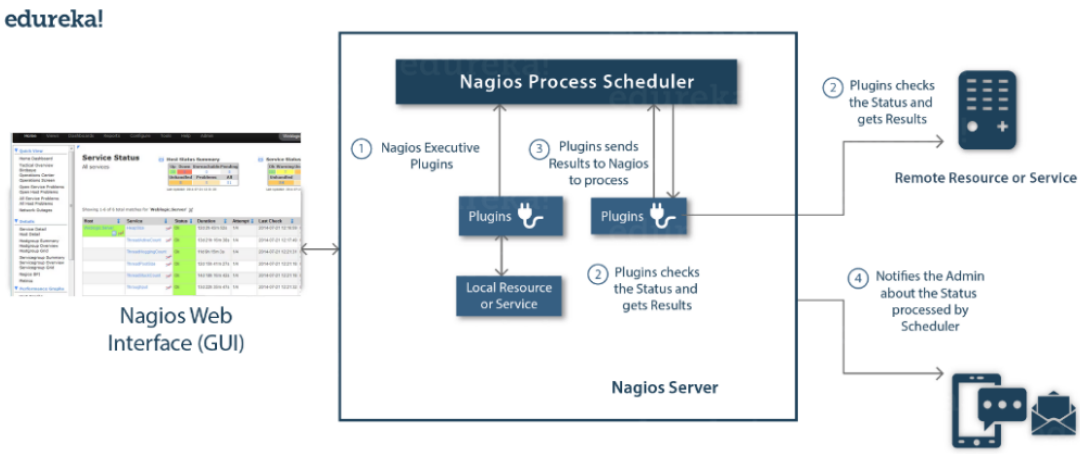

Fig. 2.1 : Architecture de Nagios

2.5.1.2 Les plugins actifs avec NCPA

Nagios offre différentes méthodes pour collecter des informations sur les machines du réseau. Il utilise à la fois une méthode active et une méthode passive, qui reposent sur l'exécution d'un daemon sur les machines surveillées. Ces deux méthodes sont généralement combinées pour assurer une surveillance efficace.

Il est important de souligner qu'un daemon est un programme informatique qui s'exécute en arrière-plan dans un système d'exploitation multitâche, sans être directement contrôlé par un utilisateur.

Le module démon NCPA (Nagios Cross-Platform Agent) fonctionne différemment des plugins locaux utilisés sur le serveur Nagios lui-même pour surveiller ses propres ressources. NCPA permet l'exécution de plugins actifs directement sur les machines à surveiller. Dans ce cas, c'est le serveur Nagios qui initie la demande d'exécution du plugin actif[10].

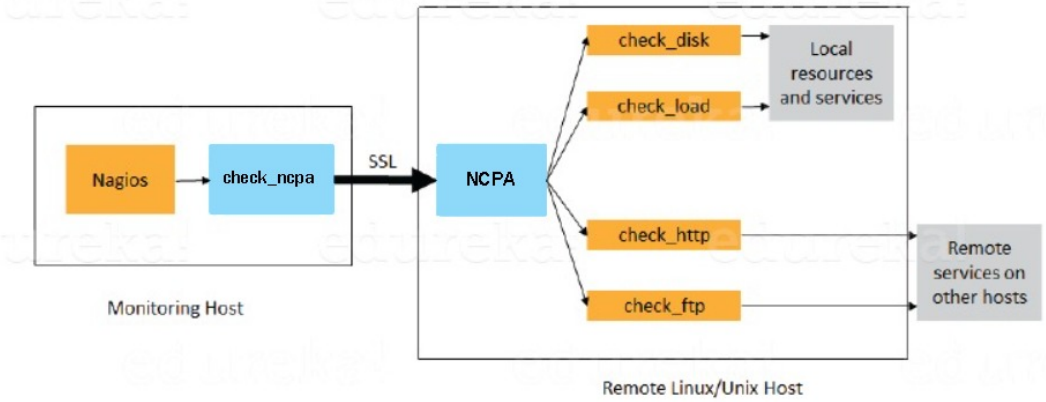

Fig. 2.2 : Ncpa

La procédure interne se déroule comme suit :
le serveur Nagios envoie une demande d'exécution du plugin P sur la machine M via le client NCPA. Le démon NCPA hébergé sur la machine M reçoit la demande d'exécution du plugin P et procède à son exécution. Une fois le plugin exécuté sur la machine M, le démon NCPA collecte les informations résultantes de l'exécution du plugin P et envoie le résultat au serveur Nagios. Enfin, le serveur Nagios interprète les résultats et prend les mesures appropriées.
Ce type de procédure permet d'assurer une surveillance distante des machines. Toutefois, il est nécessaire d'ouvrir un port de communication pour permettre au NCPA d'établir une communication avec son client et récupérer les informations d'état concernant les machines distantes[10][22].

2.5.2 Zabbix

Zabbix est une plateforme de surveillance et de gestion des performances qui permet de collecter, analyser et visualiser les données provenant de divers équipements, systèmes et applications informatiques. Elle offre une solution complète pour la surveillance de l'infrastructure informatique, la détection des problèmes, la génération d'alertes et la génération de rapports.

2.5.2.1 Architecture de Zabbix

L'architecture de Zabbix repose sur un modèle client/serveur. Zabbix se compose de plusieurs composants logiciels majeurs,

dont les responsabilités sont décrites ci-dessous[11].

- ***Serveur*** *:*
  Le serveur Zabbix est le processus central du logiciel Zabbix.le serveur effectue l'interrogation et la réception des données, il calcule les déclencheurs, envoie des notifications aux utilisateurs. C'est le composant central auquel les agents et les proxys de Zabbix rapportent des données sur la disponibilité et l'intégrité des systèmes[23].
- ***Stockage des données*** *:*
  les données sont stockées dans une base de données relationnelle, la base de données sert de référentiel central pour stocker toutes les informations collectées par Zabbix, y compris les données de surveillance, les configurations, les événements, les alertes, les journaux, etc.Chaque élément surveillé, tel qu'un hôte ou un service, est associé à des données spécifiques, telles que les valeurs des métriques, les horodatages, les états, les seuils, etc. Ces données sont enregistrées périodiquement dans la base de données, ce qui permet de les analyser, de générer des graphiques, des rapports et d'afficher l'historique des événements[23].
- ***Interface Web*** *:*
  L'interface Web est fournie pour permettre un accès facile à Zabbix de n'importe où et de n'importe quelle plate-forme. L'interface fait partie du serveur Zabbix et fonctionne généralement (mais pas nécessairement) sur la même machine physique que celle qui exécute le serveur.
- ***Proxy*** *:*
  Le proxy Zabbix est un processus qui peut collecter des données de surveillance à partir d'un ou plusieurs équipements surveillés et envoyer les informations au serveur Zabbix, en travaillant essentiellement pour le compte du serveur. Toutes les données collectées sont bufferisées localement puis transférées au serveur Zabbix auquel appartient le proxy[24].
- ***Agent*** *:*
  L'agent Zabbix est déployé sur une cible de surveillance pour superviser activement les ressources locales et les applications (disques durs, mémoire, statistiques de processeur, etc.).
  L'agent rassemble les informations opérationnelles localement et transmet les données au serveur Zabbix pour un traitement ultérieur. En cas d'échec (par exemple, un disque dur plein ou un service en panne), le serveur Zabbix peut alerter activement les administrateurs de la machine particulière qui a signalé la panne.

Les agents Zabbix sont extrêmement efficaces en raison de l'utilisation d'appels système natifs pour collecter des informations statistiques.

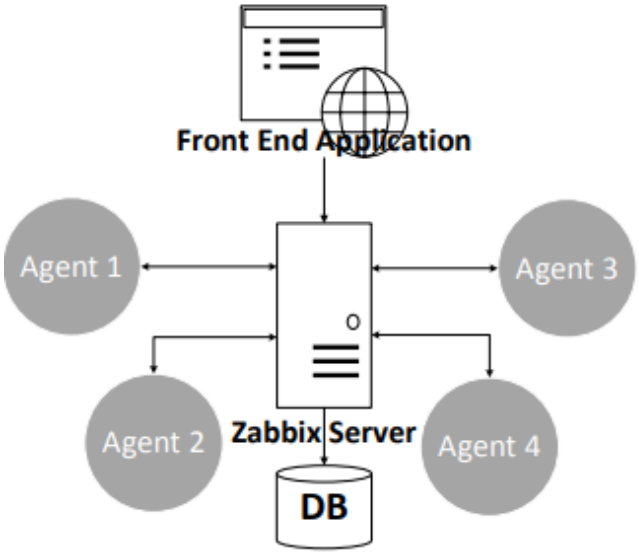

Fig. 2.3 : Architecture de Zabbix

2.5.2.2 Vérifications passives et actives

Les agents Zabbix peuvent effectuer des vérifications passives et actives.
Dans une vérification passive, l'agent répond à une demande de données. Le serveur Zabbix (ou le proxy) demande des données, par exemple, la charge du processeur, et l'agent Zabbix renvoie le résultat.
Les vérifications actives nécessitent un traitement plus complexe. L'agent doit d'abord récupérer une liste d'éléments du serveur Zabbix pour un traitement indépendant. Ensuite, il enverra périodiquement de nouvelles valeurs au serveur[29].

2.5.3 Supervision à l'aide JMX

SNMP est destiné aux périphériques réseau, tandis que JMX est destiné aux applications Java,donc JMX (Java Management Extensions) est une technologie de gestion et de surveillance pour les applications Java. Elle fournit un ensemble d'interfaces et de normes pour gérer, surveiller et contrôler des applications et des systèmes Java.
JMX permet aux développeurs et aux administrateurs système

de gérer dynamiquement les applications Java à travers les fonctionnalitées suivants[11] :

- *L'interface JMX permet d'exposer des métriques de performances pour les serveurs d'applications Java les plus populaires, tels que Tomcat, JBoss et WebLogic. Elle offre une solution pratique pour la surveillance, que ce soit avec ou sans agent, des serveurs d'applications[25].*
- *Grâce à JMX, il est possible de collecter des métriques essentielles sur les performances des serveurs d'applications Java. Ces métriques incluent des informations telles que l'utilisation de la mémoire, l'utilisation du CPU, le nombre de connexions, les temps de réponse, etc[25].*

Pour superviser une application Java à travers JMX, il est en effet nécessaire de configurer un port de communication pour permettre l'accès aux informations de gestion et de surveillance via JMX.
La configuration du port JMX peut être réalisée au niveau de l'application ou du serveur d'application lui-même.Cela implique de définir des paramètres spécifiques dans le fichier de configuration de l'application ou du serveur.
La configuration typique comprend la spécification du port JMX à utiliser, le mode de communication (local ou distant), les authentifications et les autorisations nécessaires pour accéder aux informations JMX.
Pour les serveurs d'applications Tomcat et les fichiers jar , j'ai fait la configuration du JMX dans le fichier de configuration (setenv.sh) spécifique au serveur. Ces fichiers contenir des propriétés et des directives spécifiant le port JMX à utiliser, ainsi que d'autres paramètres de sécurité et d'authentification.

2.6 Conclusion

En conclusion, ce chapitre a présenté une méthodologie d'étude comparative des outils de supervision. En évaluant leurs fonctionnalités, avantages et limitations, nous serons en mesure de choisir l'outil qui répondra le mieux à nos besoins spécifiques. Cette décision sera cruciale pour le succès de notre projet de supervision et l'optimisation de notre efficacité opérationnelle.

Chapitre 3

Analyse et Conception

Dans ce chapitre, nous abordons la phase d'analyse et de conception du projet, en présentant la solution proposée ainsi que sa mise en œuvre en termes de besoins fonctionnels et non fonctionnels. De plus, nous incluons les diagrammes de conception pertinents qui permettent de visualiser la structure du projet. Enfin, nous décrivons l'architecture technique de l'environnement réel . En examinant ces éléments, nous acquérons une compréhension approfondie des différents aspects du projet, nous sommes en mesure de le concevoir de manière adéquate, et nous augmentons les chances de sa réussite lors de la mise en œuvre ultérieure.

3.1 Spécification des besoins

Dans un projet, les besoins fonctionnels se réfèrent aux fonctionnalités pratiques et tangibles du produit, tandis que les besoins non fonctionnels servent de critères de qualité pour évaluer la mise en œuvre de ces fonctionnalités. Après une analyse approfondie du projet, nous rassemblons dans cette section tous les besoins fonctionnels et non fonctionnels de notre application.

3.1.1 Besoins fonctionnels

L'objectif fondamental d'un projet est de répondre à un besoin spécifique. Les besoins fonctionnels sont donc une manifestation de ce que le service ou le produit final doit accomplir pour satisfaire ce besoin. Ils décrivent les fonctionnalités, les actions et les tâches spécifiques que le projet doit mettre en œuvre pour atteindre ses objectifs.
Alors, Notre connecteur doit être capable de faire :

- ***Configurer les hosts*** *:Le connecteur doit être en mesure d'établir une communication bidirectionnelle avec Zabbix (Nagios) afin de pouvoir configurer l'host.*
- ***Configurer les items*** *: Le connecteur doit être capable d'établir une connexion avec le serveur afin de pouvoir configurer les items.*
- ***définir les valeurs limites de déclenchement des triggers*** *: Le connecteur doit permettre à l'utilisateur de définir les valeurs de déclenchemet des trigger qui engendre l'envoie des notifications aux utilisateurs*

3.1.2 Besoins non fonctionnels

Les besoins non fonctionnels sont tous les besoins qui ne sont pas en relation avec le métier mais en général, ils sont liés indirectement au système, en d'autres termes les besoins non fonctionnels sont les besoins de qualité.
Les besoins non fonctionnels de notre connecteur sont représentés comme suit :

- ***Performance*** *: Le connecteur doit être capable de gérer un grand volume de données et de maintenir des temps de réponse acceptables pour assurer une surveillance en temps réel.*

- ***Sécurité*** *: Le connecteur doit garantir la sécurité des données échangées entre le switch monétique et Zabbix (Nagios).*
- ***L'interopérabilité*** *: Le connecteur doit être en mesure de s'intégrer avec le système existant (Payway).*

3.2 Analyse conceptuelle

Pour notre projet, nous avons adopté la méthodologie UML afin de créer des modèles visuels et des diagrammes pour représenter l'architecture du système. Cette phase de modélisation revêt une importance particulière, car elle permet de capturer la vision "fonctionnelle" de l'architecture du système.
Ces diagrammes fonctionnels nous ont permis de visualiser et de comprendre la structure et le comportement du système de manière claire et précise. Ils ont servi de référence pour la conception et la mise en œuvre du système.

3.2.1 Identification des acteurs

Notre connceteur fait partie du module PayMonitor, donc les acteurs seront les clients qui ont acheté ce module en particulier ou l'application en générale, ou bien les membres de l'équipe support et intégration qui assiste les clients en cas de problème.

3.2.2 Diagramme de cas d'utilisations

Le diagramme de cas d'utilisation est un outil qui permet de définir les interactions entre le système et les acteurs, en identifiant toutes les fonctionnalités que le système doit fournir. Il représente l'interopérabilité entre le système et les utilisateurs, en décrivant les différents scénarios d'utilisation.Dans le cadre de notre projet, nous avons élaboré un diagramme de cas d'utilisation général qui répertorie toutes les actions principales que notre système doit prendre en charge.[13]

En présentant ce diagramme, nous donnons une visualisation claire et concise des différentes façons dont les utilisateurs interagiront avec notre système, ce qui facilite la compréhension des exigences fonctionnelles et oriente la conception et le développement ultérieurs.

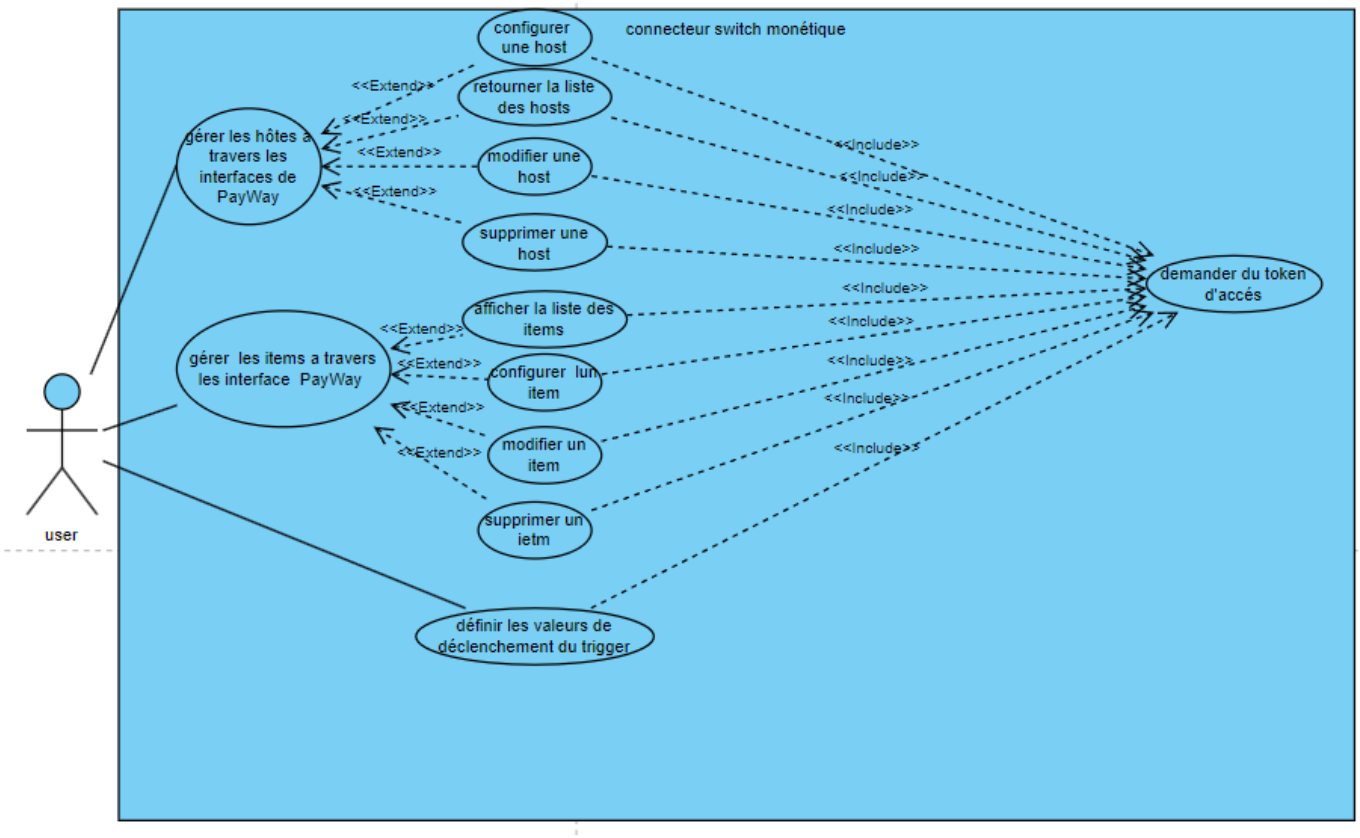

Fig. 3.1 : Diagramme de cas d'utilisation

3.2.2.1 Cas d'utilisation :Gestion des hôtes à travers les interfaces de Payway

Nom de cas d'utilisation	Gérer les hôtes à travers les interfaces Payway
Acteur	Client de Paylogic/ équipe support et intégration
Description	L'acteur peut gérer le hôte qui veut monitorer à traver les interfaces payway .
Scenarios possible	- L'utilisateur peut configurer un hôte . - L'utilisateur peut recevoir la liste des hôtes . - L'utilisateur peut modifier ou supprimer un hôte.

Tab. 3.1 : Gestion des hôtes à travers les interfaces de Payway

3.2.2.2 Cas d'utilisation :Gestion des items à travers les interfaces de Payway

Nom de cas d'utilisation	Gérer les items à travers les interfaces de Payway
Acteur	Client de Paylogic/ équipe support et intégration
Description	L'acteur peut gérer les items d'un hôte à traver les interfaces de payway .
Scenarios possible	- L'utilisateur peut configurer un item . - L'utilisateur peut afficher la liste des items . - L'utilisateur peut modifier ou supprimer un item.

Tab. 3.2 : Gestion des items à travers les interfaces de Payway

Remarque : *le terme "item" désigne les métriques qu'on veut monitorer (CPU, RAM, mémoire...)*

3.2.3 Diagrammes de séquences

Les diagrammes de séquence sont des outils graphiques utilisés dans la notation UML pour représenter de manière séquentielle et chronologique les interactions entre les acteurs et le système.

Ces diagrammes capturent visuellement les échanges de messages et les actions qui se déroulent entre les différents acteurs et le système, en mettant l'accent sur l'ordre dans lequel ces interactions se produisent. Ils permettent de modéliser le déroulement des scénarios d'utilisation et de visualiser comment les objets et les acteurs interagissent au fil du temps[13].

3.2.3.1 Diagramme de séquence : Chargement de configuration

Le diagrammes de séquence de chargement de configuration fournie une représentation claire et précise des étapes et des messages échangés entre les acteurs et le système, ce qui facilite la compréhension du comportement dynamique du système et aide à valider les fonctionnalités attendues.

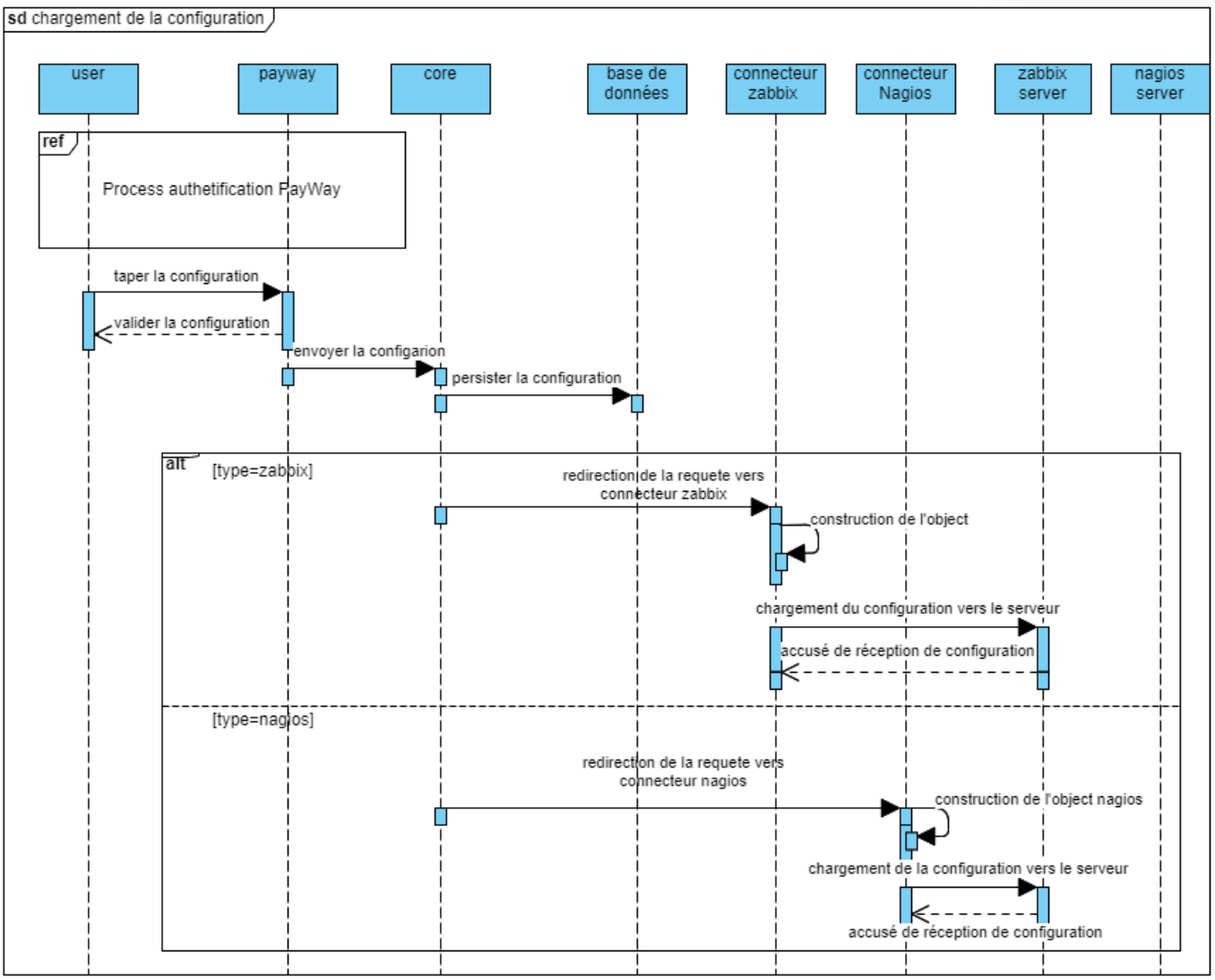

Fig. 3.2 : Diagramme de séquence : Chargement de configuration

3.2.3.2 Diagramme de séquence : Monitoring d'un serveur

Le diagrammes de séquence de monitoring de serveur représente un example concret de récupération des mesures du serveur à surveillé en montrant toutes les étapes et les messages échangés entre les acteurs et le système. Ils simplifient la compréhension du comportement dynamique du système et contribuent à la validation des fonctionnalités attendues.

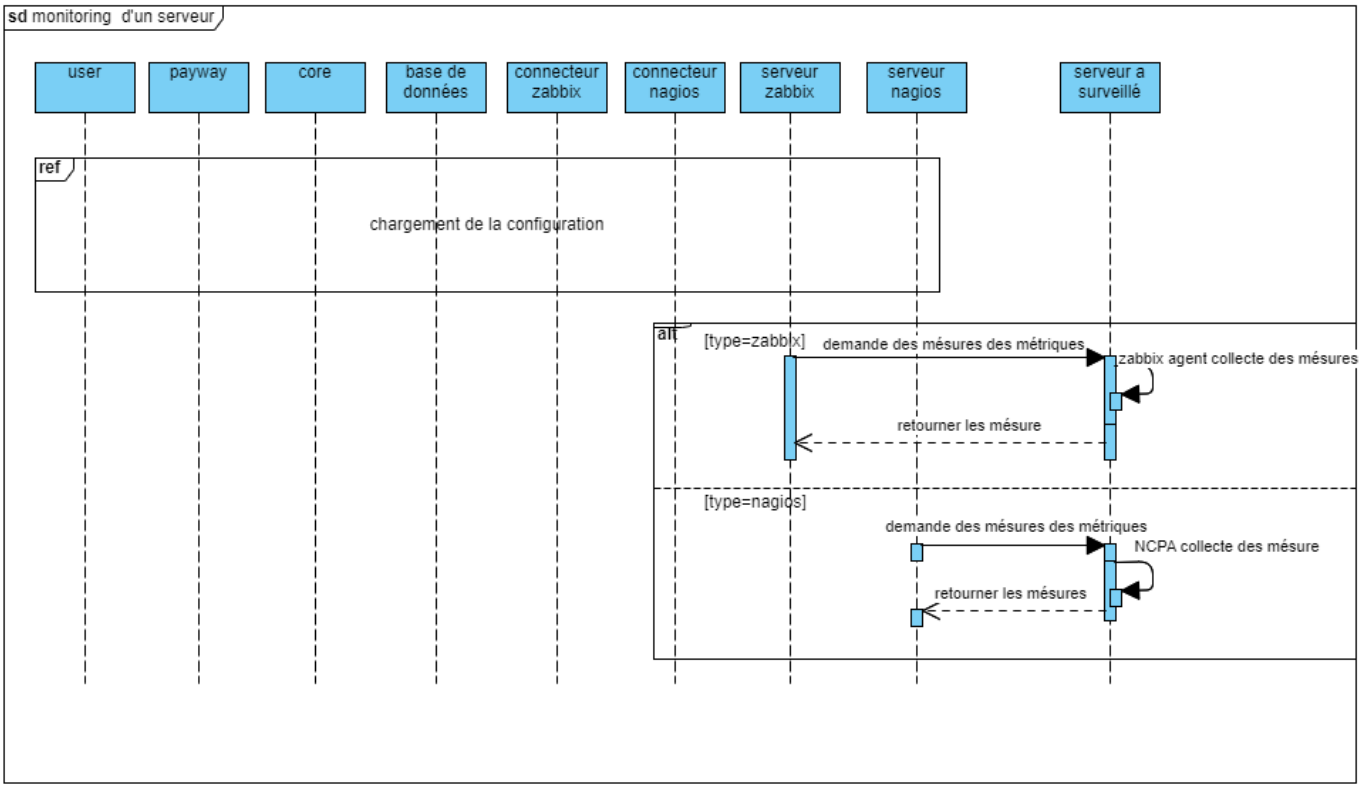

Fig. 3.3 : Diagramme de séquence : Monitoring d'un serveur

3.2.4 Diagramme de classe

Le diagramme de classe est considéré comme l'un des types de diagrammes UML les plus précieux, car il offre une description claire de la structure d'un système spécifique en modélisant ses classes, ses attributs, ses opérations et les relations entre ses objets. Alors que le diagramme de cas d'utilisation se concentre sur le système du point de vue des acteurs, le diagramme de classes se concentre plutôt sur la structure interne du système[30].

Pour établir une indépendance entre les interfaces de Payway et les outils de surveillance Zabbix et Nagios XI on était obligé de chercher les attributs en commun entre les deux outils de surveillance et les attributs qui sont spécifiques à chacun on a les modélisé sous forme de clé valeur avec le contenu de la clé c'est le nom du paramètre et le contenu de la valeur c'est la valeur du paramètre[13].

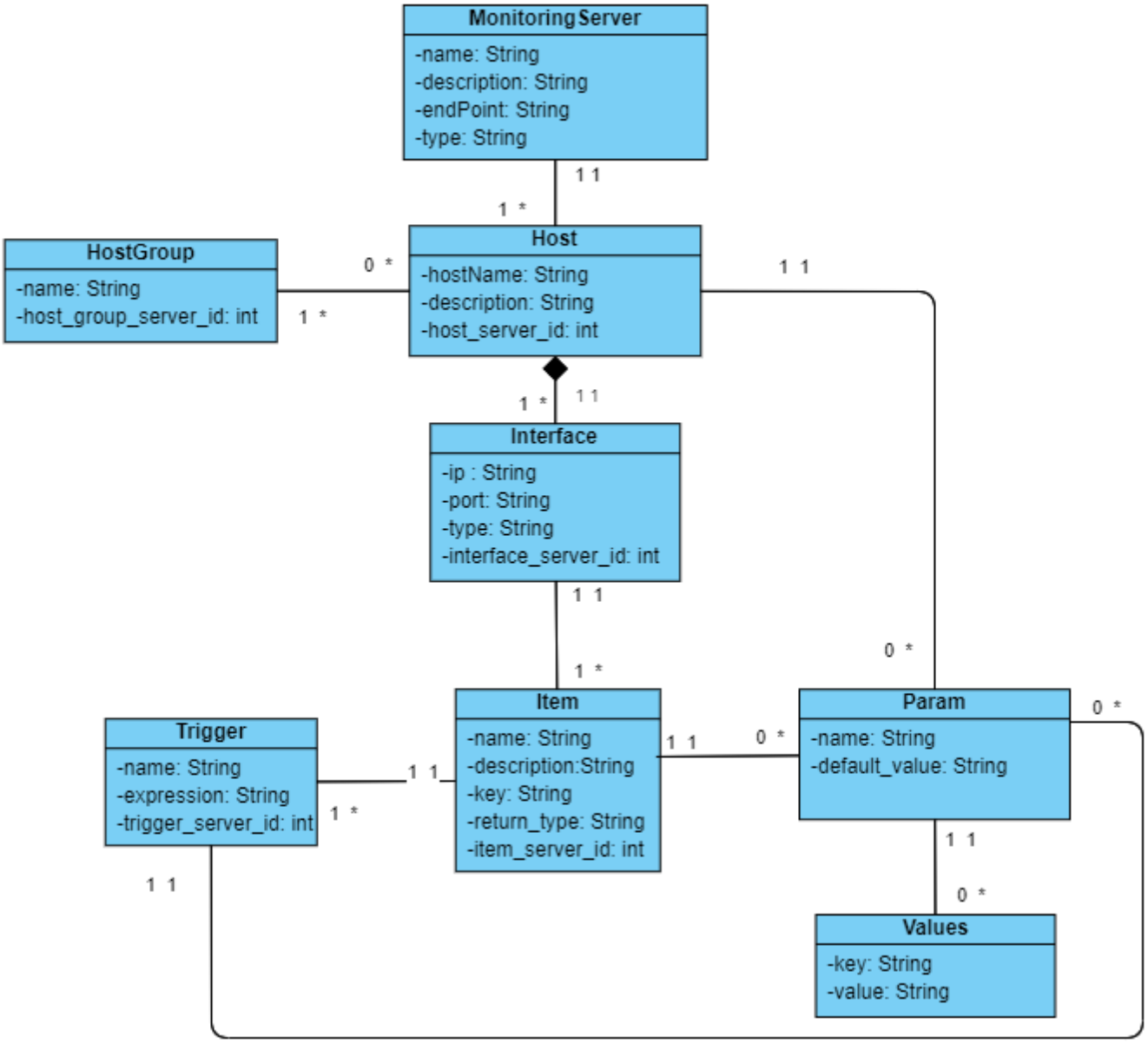

Fig. 3.4 : Diagramme de classes

3.3 Architecture Technique de l'environnement réel

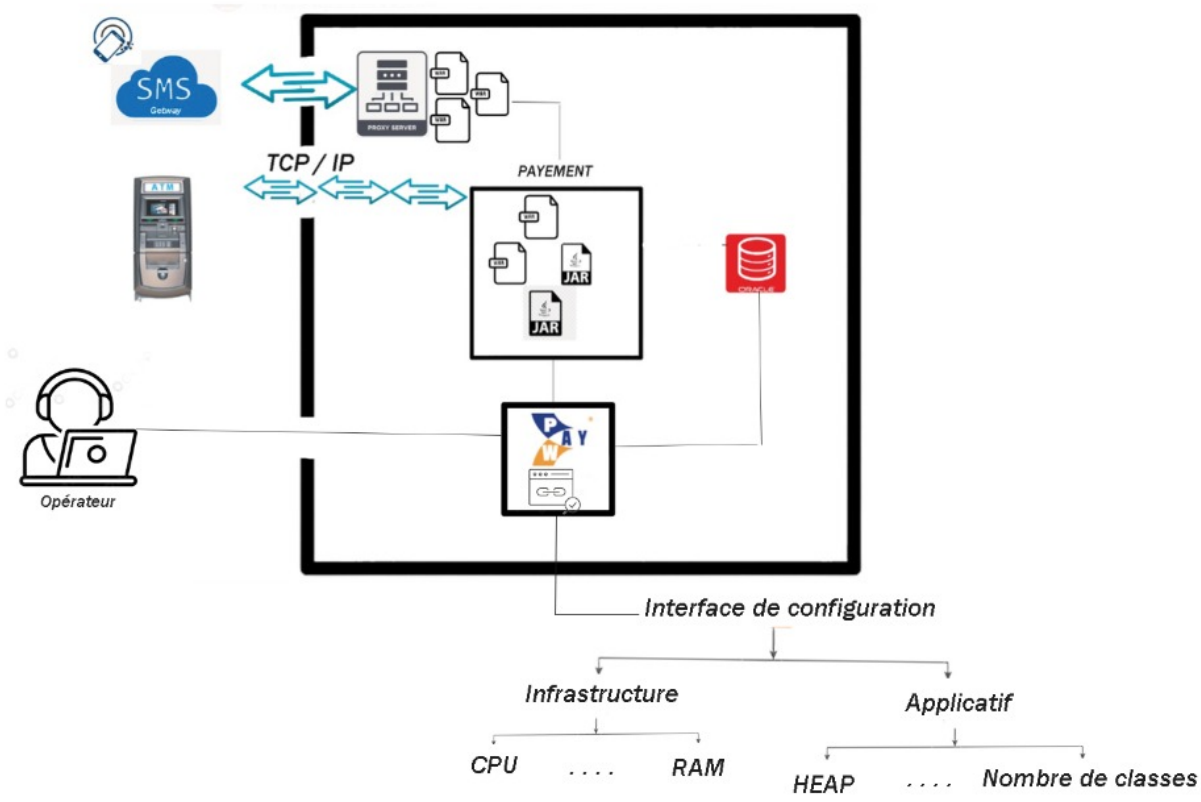

Fig. 3.5 : Architecture de l'environnement réel

L'architecture technique de notre projet, qui s'inscrit dans le cadre de l'Environnement Trésorie Public de Madagascar, comprend les éléments suivants :

- ***Serveur Payway** : Il s'agit d'un switch monétique qui joue un rôle central dans le traitement des transactions de paiement. Ce serveur est équipé de fichiers WAR , JAR et des serveurs Tomcat, qui constituent les composants logiciels nécessaires au fonctionnement du système.*
- ***Serveur Payment** : Ce serveur est également équipé de fichiers JAR et WAR et est responsable de la communication avec les GABs . Il utilise le protocole TCP/IP pour établir des connexions et échanger des données avec les GAB. Le serveur Payment gère les transactions financières et assure la sécurité des échanges d'informations sensibles.*

- ***Serveur Proxy*** *:Ce serveur agit comme une passerelle entre votre environnement de projet et l'extérieur. Il contient également des fichiers WAR et communique avec divers processus, tels que les services de messagerie mobile et SMS. Le serveur Proxy permet de gérer les communications externes et d'assurer la sécurité en filtrant les requêtes entrantes et sortantes.*
- ***Base de données(Oracle)*** *:Notre environnement de projet inclut une base de données qui stocke les informations relatives aux transactions, aux utilisateurs et à d'autres données pertinentes. La base de données joue un rôle essentiel dans la persistance des données et permet aux différents composants du système d'accéder et de manipuler ces données de manière sécurisée et cohérente.*

Le monitoring de l'environnement dans le cadre de Notre projet se divise en deux aspects principaux : l'infrastructure et l'applicatif. Chacun de ces aspects permet de surveiller différents éléments pour assurer le bon fonctionnement du système. Voici plus de détails sur ces deux composantes :

- ***Monitoring de l'infrastructure*** *:Cette partie du monitoring se concentre sur la surveillance des ressources matérielles et systèmes de l'infrastructure . Elle inclut des métriques telles que l'utilisation de la RAM, du CPU, de l'espace de stockage. L'objectif est de s'assurer que les ressources sont suffisantes pour répondre aux besoins du système et d'identifier tout problème d'étranglement ou toute surcharge potentielle. Les outils de monitoring Zabbix(Nagios) sont utilisés pour collecter ces données et générer des alertes en cas de dépassement de seuils prédéfinis.*

- ***Monitoring de l'applicatif*** *: Cette partie se concentre sur la surveillance des composants logiciels spécifiques à nos applicatinos. Elle peut inclure des métriques liées à la mémoire (heap), au nombre de threads, nombre classes chargées.Ces métriques permettent de comprendre les performances de l'application et d'identifier tous problème qui peut être généré.*

En surveillant à la fois l'infrastructure et l'applicatif, nous pouvons avoir une vue d'ensemble complète de l'état de notre environnement de projet. Cela nous permet de détecter et de résoudre rapidement les problèmes de performance, d'optimiser les ressources, d'anticiper les pannes potentielles et d'assurer une expérience utilisateur optimale. Les données collectées par les outils de monitoring nous aidons à prendre des décisions informées concernant l'optimisation et l'amélioration continue de

notre architecture technique.

3.4 Architecture Technique du module Pay-Monitor

L'applicatif core est le noyau du module Paymonitor, responsable de récupérer les données de configuration à partir des interfaces de Payway et de les stocker dans la base de données. Il traite ensuite les requêtes et détermine le connecteur approprié (connecteur Zabbix ou connecteur Nagios XI) chargé de les recevoir et de les envoyer vers le serveur de monitoring.
L'applicatif core agit comme un intermédiaire entre les interfaces de Payway, la base de données et les connecteurs de monitoring, en coordonnant la récupération des données de configuration et le routage approprié des requêtes vers le connecteur approprié.

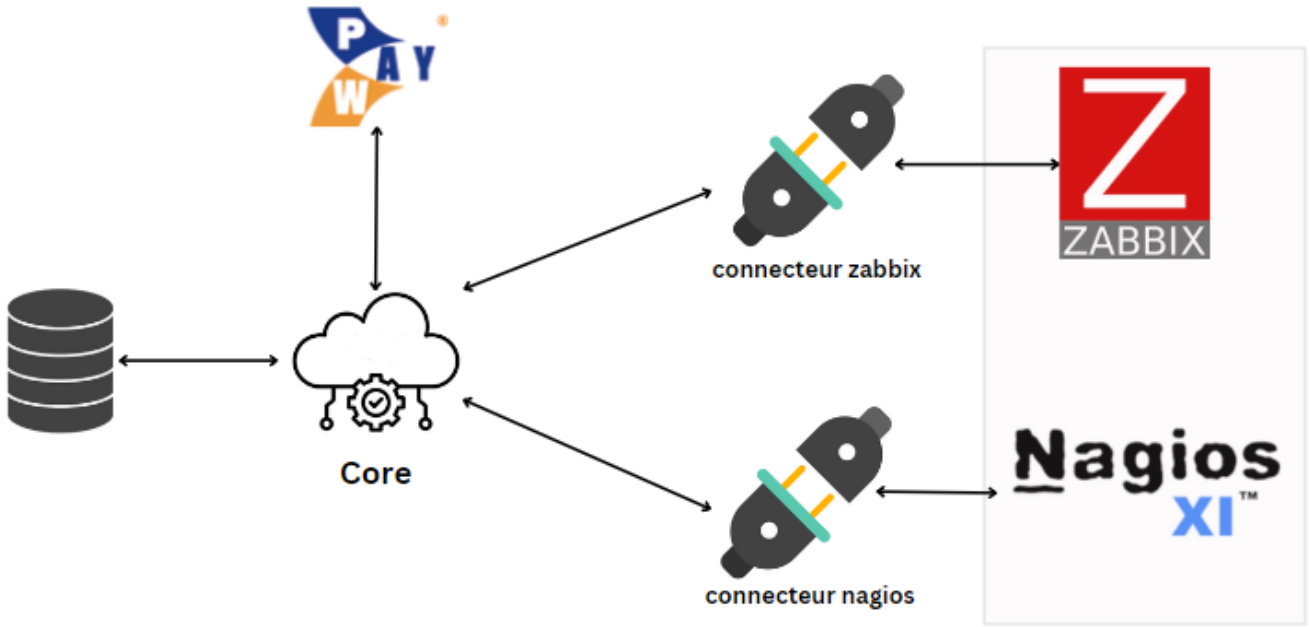

Fig. 3.6 : Architecture du module PayMonitor

3.5 Conclusion

Ce chapitre d'analyse et de conception du projet nous a permis de présenter la solution envisagée ainsi que sa mise en œuvre à travers les besoins fonctionnels et non fonctionnels identifiés.

Les diagrammes de conception ont fourni une représentation visuelle de la structure et des interactions du projet. De plus, la description de l'architecture technique de l'environnement réel a mis en place les bases nécessaires pour une mise en œuvre réussie.

Chapitre 4

Réalisation

4.1 Introduction

Dans ce chapitre, nous allons explorer les différents outils essentiels qui ont été utilisés tout au long du projet, ainsi que nous allons fournir des captures d'écran de test deu connecteur . Cette partie couvre à la fois la partie supervision de l'infrastructure et le développement de connecteur.

4.2 Les outils de réalisation

4.2.1 Outils de supervision

Dans ce projet, j'ai utilisé Nagios XI et Zabbix pour mettre en place le monitoring de mon infrastructure. J'ai également utilisé les agents correspondants, à savoir NCPA pour Nagios XI et Zabbix Agent pour Zabbix.

4.2.1.1 Nagios xi

Nagios XI fonctionne en tant que plateforme de supervision robuste qui collecte et analyse les données de surveillance pour différents composants de l'infrastructure. Pour ce faire, Nagios XI s'appuie sur NCPA , un agent léger installé sur les hôtes à superviser. NCPA est configuré avec des plugins spécifiques permettant de collecter les informations pertinentes sur les hôtes surveillés, comme les vérifications de disponibilité, les mesures de performances et les contrôles de service. Une fois installé, NCPA communique régulièrement avec le serveur Nagios XI, fournissant des mises à jour de statut, ainsi que des résultats de vérification et de collecte de données. Les données sont ensuite traitées par Nagios XI, qui génère des alertes en cas de dépassement des seuils définis ou de problèmes identifiés. L'interface de Nagios XI offre des fonctionnalités avancées pour visualiser les données de surveillance, créer des tableaux de bord, des graphiques, des rapports et des journaux d'événements, tout en permettant une configuration et une personnalisation flexibles de la supervision.[14]

4.2.1.2 Zabbix

Zabbix est une plateforme complète de supervision qui permet de collecter, stocker, analyser et visualiser les données de surveillance pour divers éléments de l'infrastructure. Le système repose sur Zabbix Agent, un agent installé sur les hôtes à superviser. Zabbix Agent est configuré avec des éléments de surveillance définissant les métriques à collecter, telles que l'utilisation du processeur, la disponibilité des services, la charge du système, etc. Une fois installé, Zabbix Agent communique avec le serveur Zabbix, envoyant régulièrement les données collectées. Le serveur Zabbix stocke les données dans une base de données et les analyse pour générer des graphiques, des alertes

et des rapports. L'interface utilisateur de Zabbix offre une expérience conviviale où les administrateurs peuvent configurer les éléments de surveillance, les seuils d'alerte, les graphiques et les tableaux de bord. Les alertes générées par Zabbix peuvent être transmises aux administrateurs système via différents canaux, tandis que l'interface permet une visualisation en temps réel des données de surveillance, une analyse des tendances historiques et la génération de rapports détaillés[15]

4.2.2 Outils de développement

4.2.2.1 Spring Boot

Le Java Spring Framework (Spring Framework) est une infrastructure d'entreprise open source largement utilisée pour développer des applications autonomes de production qui s'exécutent sur la machine virtuelle Java (JVM). Il offre une multitude de fonctionnalités et de bibliothèques pour faciliter le développement d'applications robustes et évolutives. Dans le contexte du

développement d'applications Web et de microservices, le Java Spring Boot (Spring Boot) est un outil qui simplifie et accélère ce processus en utilisant Spring Framework. Il repose sur trois principales fonctionnalités[24] :

- ***Configuration automatique*** *:Spring Boot analyse le code et les dépendances de l'application pour détecter automatiquement la configuration requise. Cela permet de réduire considérablement la quantité de configuration manuelle nécessaire, ce qui accélère le développement.*
- ***Approche directive de la configuration*** *:Spring Boot utilise une approche basée sur des conventions plutôt que sur des configurations explicites. Il fournit des annotations et des conventions pour définir rapidement les composants de l'application, tels que les contrôleurs, les services et les repositories. Cela facilite la mise en place rapide d'une architecture cohérente et bien structurée.*
- ***Possibilité de créer des applications autonomes*** *: : Spring Boot permet de créer des applications autonomes qui peuvent être exécutées indépendamment, sans nécessiter de serveur d'applications externe. Il embarque un serveur Web intégré, ce qui facilite le déploiement et la distribution de l'application.*

En combinant ces fonctionnalités, Spring Boot permet de configurer et de déployer rapidement des applications Spring avec une configuration et une installation minimales. Cela offre aux

développeurs un environnement de développement plus efficace et leur permet de se concentrer sur la logique métier de leurs applications.

4.2.2.2 Postman

Postman est un outil utilisé pour tester, déboguer et documenter les API (Application Programming Interfaces). Il fournit une interface conviviale permettant aux développeurs de créer, envoyer et recevoir des requêtes HTTP pour interagir avec des services web[16].
Parmi les fonctionnalités clés de Postman :

- *Envoi de requêtes HTTP : Postman permet d'envoyer facilement des requêtes HTTP telles que GET, POST, PUT, DELETE, etc. Il offre une interface intuitive où vous pouvez spécifier les paramètres de la requête, les headers, les corps de requête et les authentifications.*
- *Tests et validation : Postman permet d'écrire des scripts de test pour valider les réponses des API. Vous pouvez ajouter des assertions pour vérifier si les réponses correspondent aux résultats attendus. Cela facilite la validation de la conformité des API et la détection des erreurs.*

4.2.2.3 IntelliJ IDEA

IntelliJ IDEA est un environnement de développement intégré (IDE) populaire développé par JetBrains. Il est principalement utilisé pour le développement Java, mais prend également en charge d'autres langages de programmation tels que Kotlin, Groovy, Scala, et plus encore. IntelliJ IDEA offre un ensemble complet d'outils et de fonctionnalités pour améliorer la productivité et rationaliser le processus de développement. Les principales caractéristiques d'IntelliJ IDEA comprennent un éditeur de code intelligent, un débogueur avancé, une intégration de contrôle de version, des outils de test[17].

4.2.2.4 Vmware workstation

VMware Workstation est une gamme de produits d'hyperviseur de bureau qui permet aux utilisateurs d'exécuter des machines virtuelles, des conteneurs et des clusters Kubernetes. Quelles sont les différentes éditions de VMware Workstation ? La gamme de produits VMware Workstation se compose de deux produits : Workstation Pro et Workstation Player[34].

4.2.2.5 Git

Git est un logiciel de gestion de versions décentralisé. C'est un logiciel libre et gratuit, créé en 2005 par Linus Torvalds, auteur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2. Le principal contributeur actuel de Git, et ce depuis plus de 16 ans, est Junio C Hamano[34].

4.3 Surveillance de l'environnement réel (trésor public de madagascar)

Dans cette partie on va vous présente le résultat de la surveillance de l'environnement réelle qui est constitué de plusieurs serveurs Payway, Payment et serveur proxy.
pour pouvoir surveillé un serveur il faut le configurer comme vous le trouvez ci-dessous :

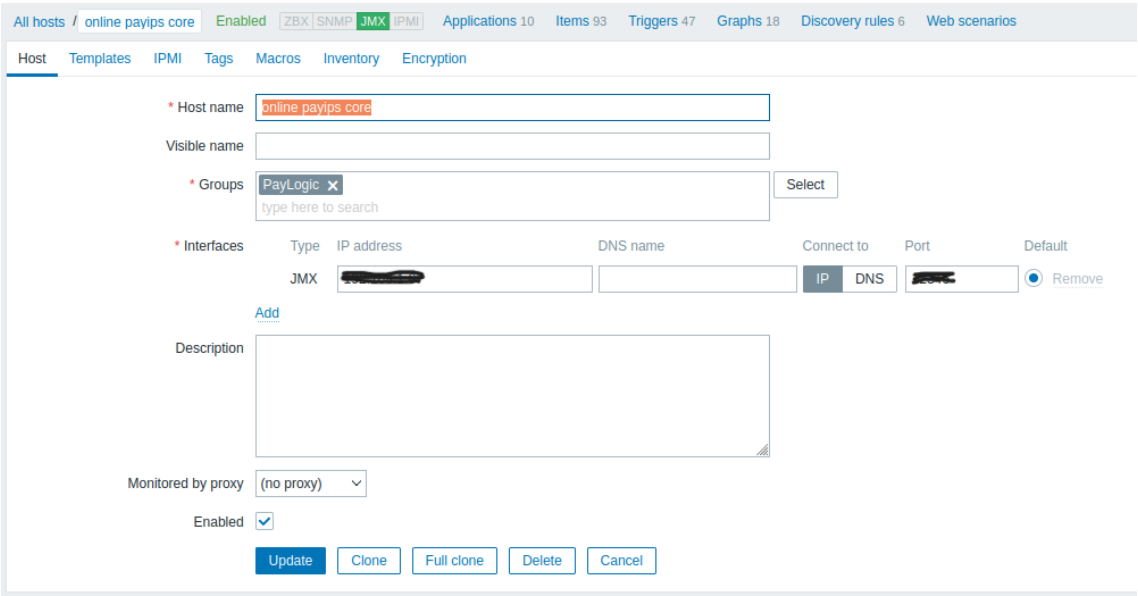

Fig. 4.1 : configurer un serveur

l'interface ci-dessous présente l'ensemble des serveurs qu'on a surveillé

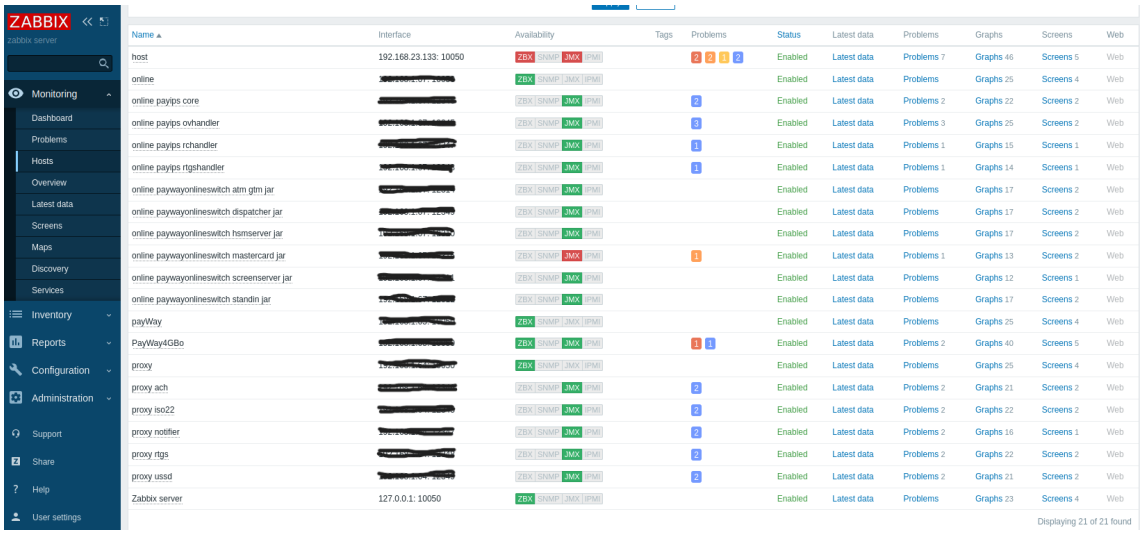

Fig. 4.2 : les serveurs surveillés

Après la configuration des hostes on a configuré des items , vous trouvez ci-dessous des exemples de configuration des items :

- *configuration CPU*

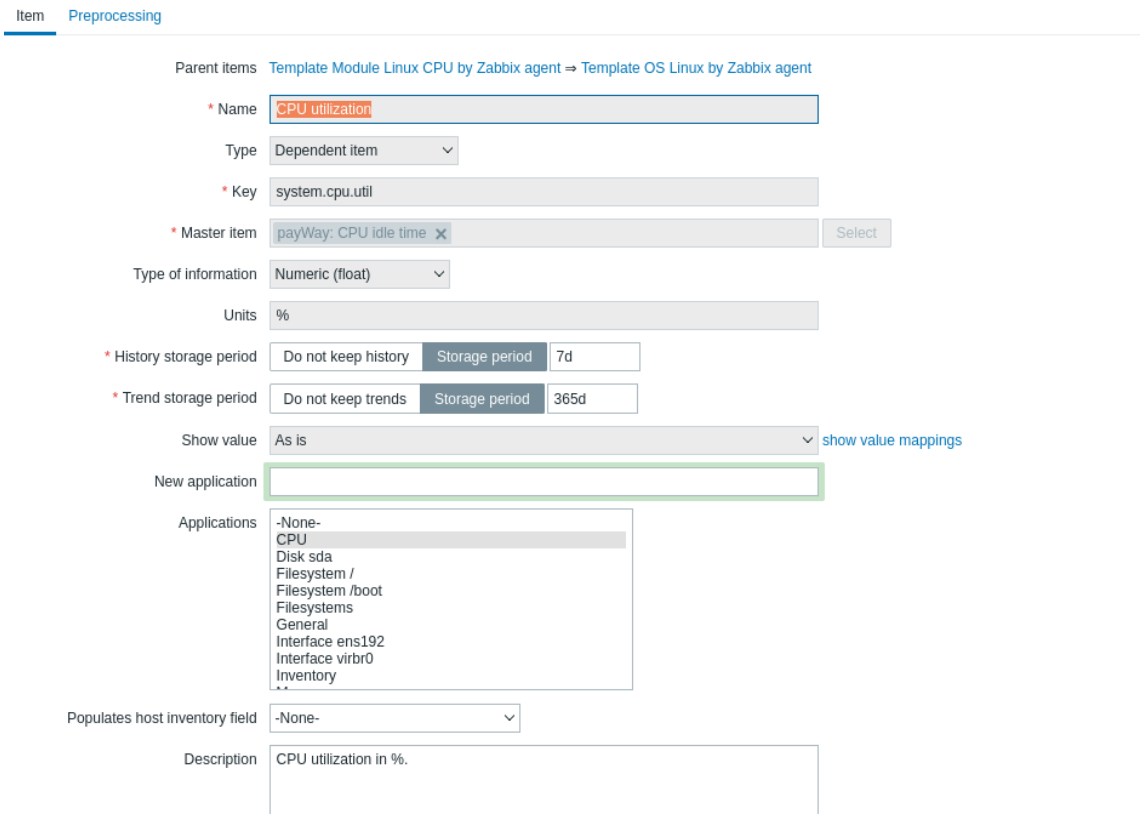

Fig. 4.3 : configuration CPU

- *configuration RAM*

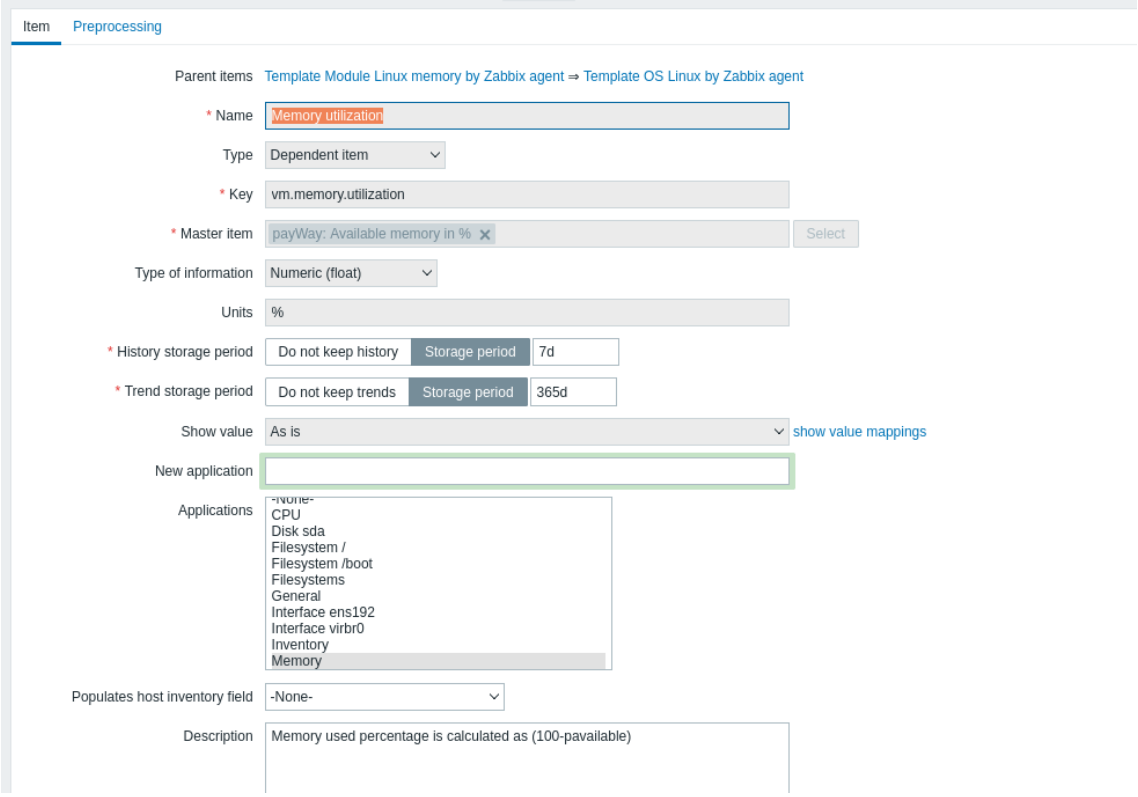

Fig. 4.4 : configuration RAM

- *configuration disque dur*

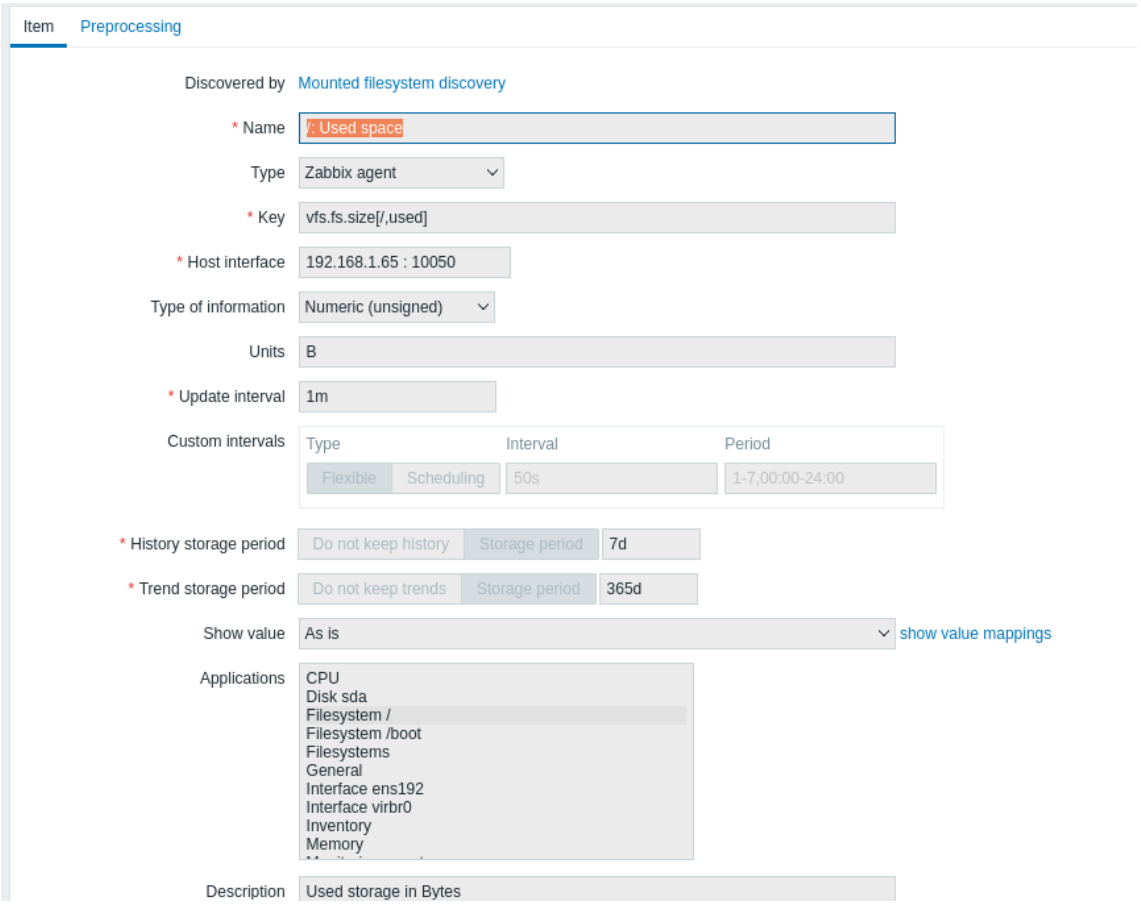

Fig. 4.5 : configuration disque dur

après avoir configuré les différents items souhaités on a configuré les triggers(alerts) qui sont des éléments déclencheurs qui permettent d'envoyer notifier en cas de dépassement des valeurs définies

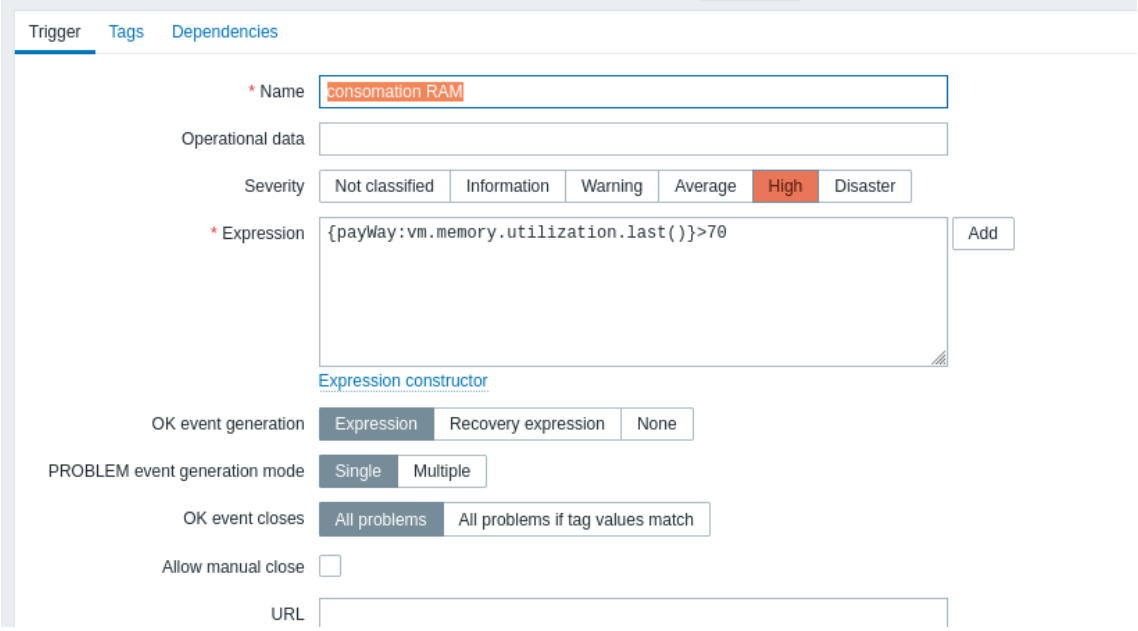

Fig. 4.6 : configuration d'un triggers

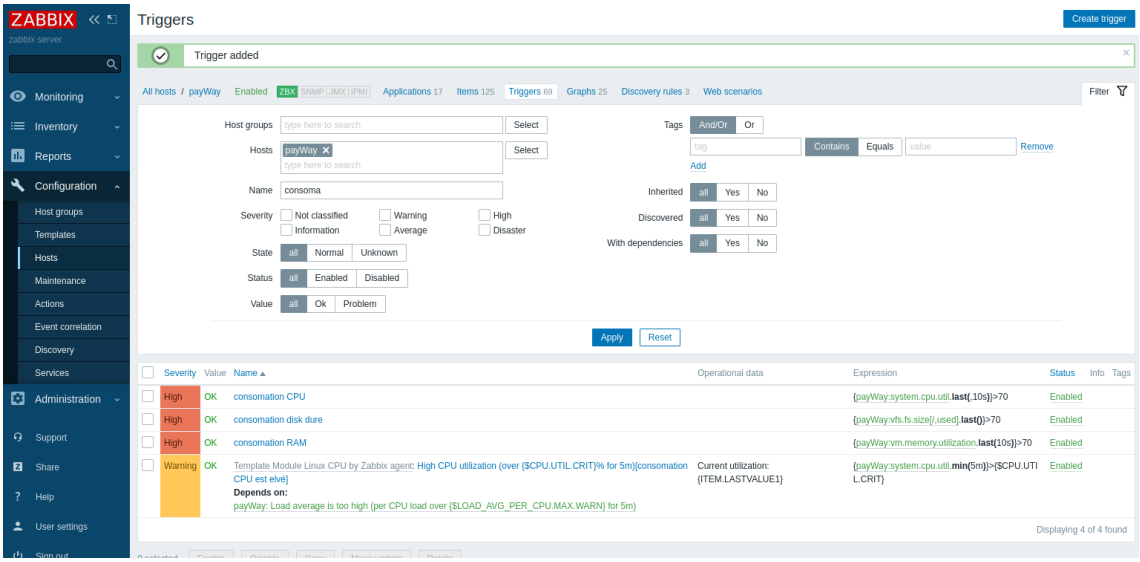

Fig. 4.7 : exemples des triggers

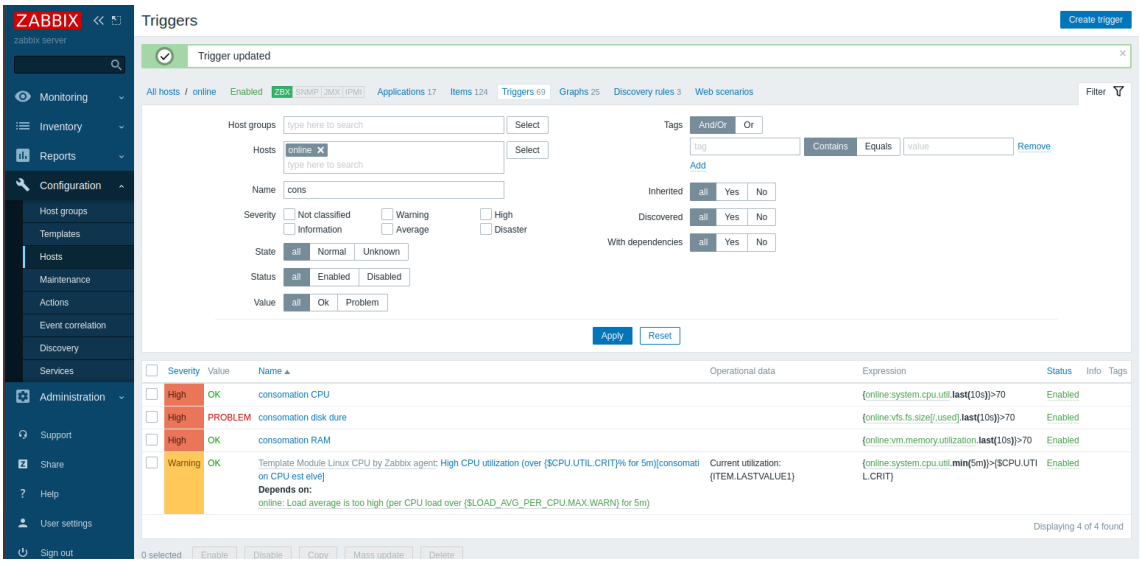

Fig. 4.8 : exemples des triggers : erreur

après le déclenchement du trigger un email sera envoyé a l'administrateur pour le prévenir.

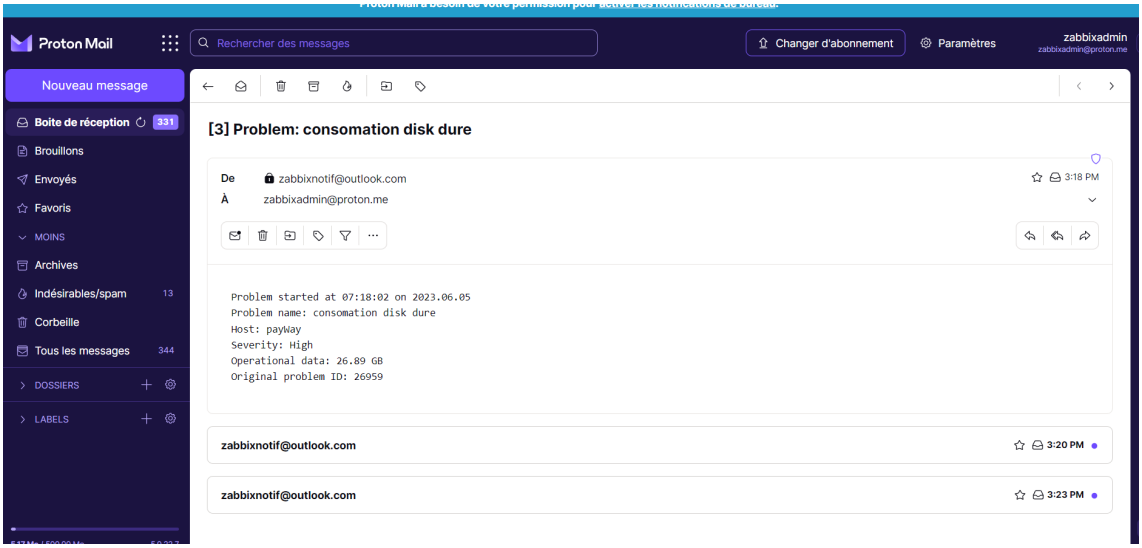

Fig. 4.9 : exemples des triggers : erreur

on a aussi surveillé les fichiers log et on a filtré les résultats pour ne présenter que les lignes contenant des erreurs.

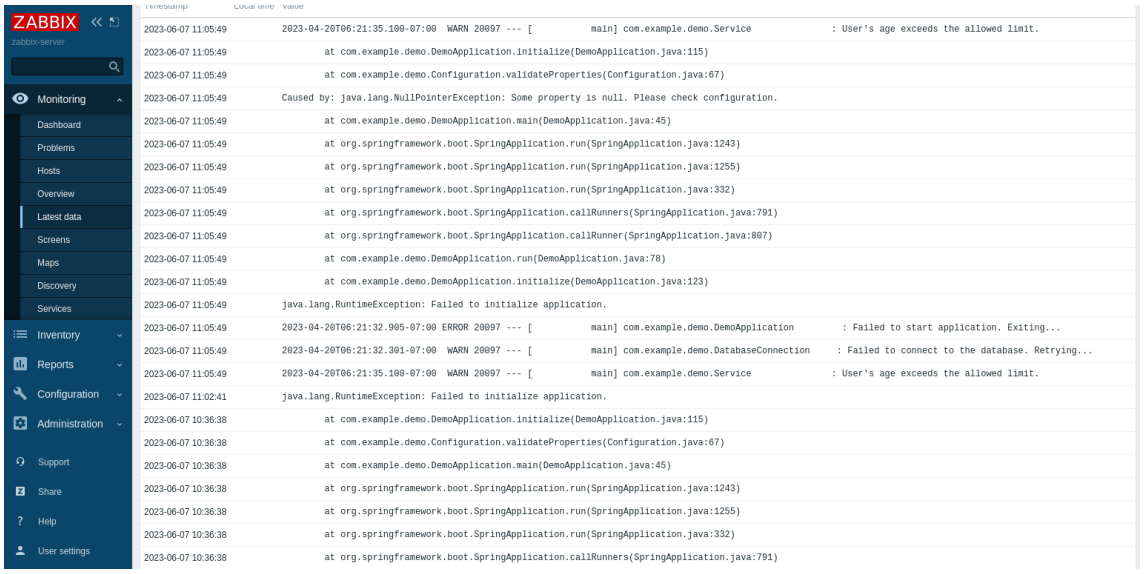

Fig. 4.10 : les fichiers logs

et pour faciliter la visualisation des données on a créé un tableau de bord qui regroupe les différents graphes des métriques désirées .

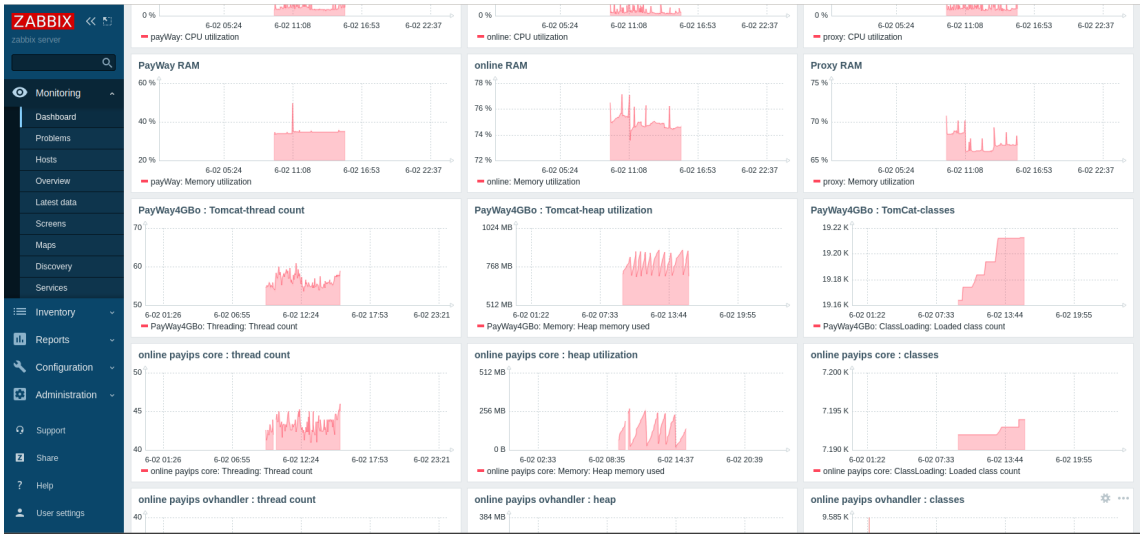

Fig. 4.11 : tableau de bord

4.4 Démonstration du connecteur

Dans cette partie je vais essayer de presenter le connecteur développé qui est API qui permet de communiquer avec le serveur de monitoring, a travers ce connecteur nous pouvons effectuer des opérations telles que l'ajout, la suppression ou la modification d'hôtes ou des items... avec la visualisation du résultat au niveau des serveurs. Avant de commencer la configuration il faut

s'authentifier au serveur zabbix à partir de postman.

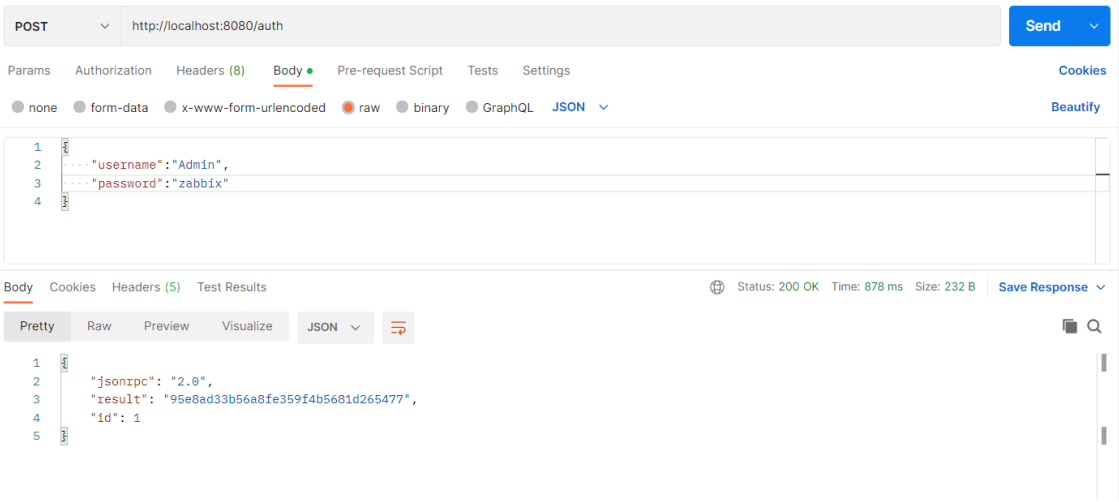

Fig. 4.12 : Autentification à zabbix

Dans l'example suivant j'ai procédé à la configuration de création d'un hôte (test Api agent) dans Zabbix pour le surveiller.

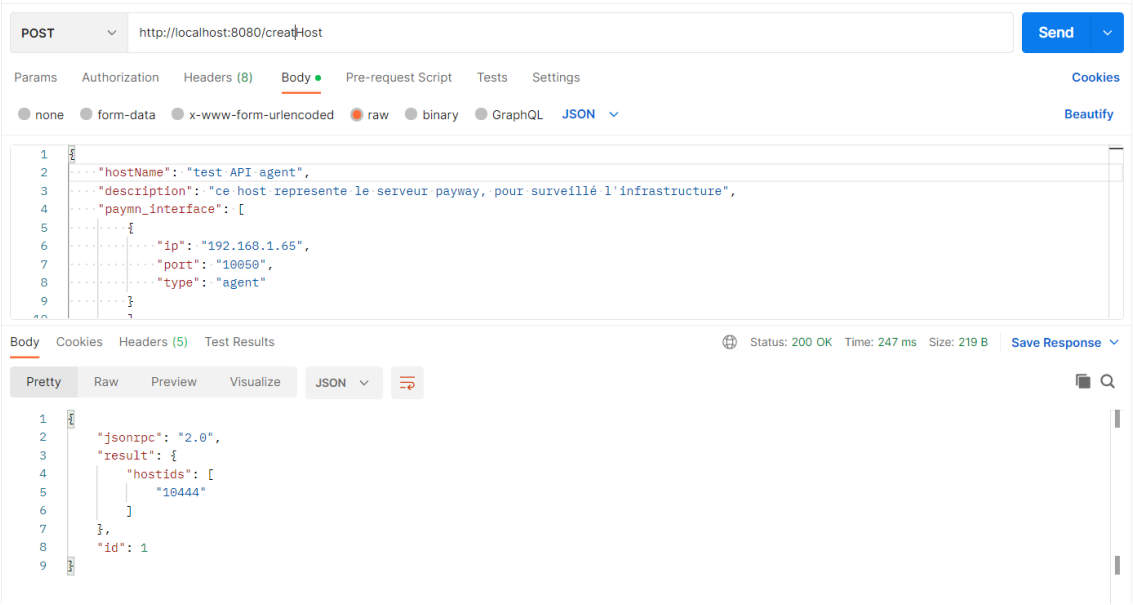

Fig. 4.13 : create host dans zabbix

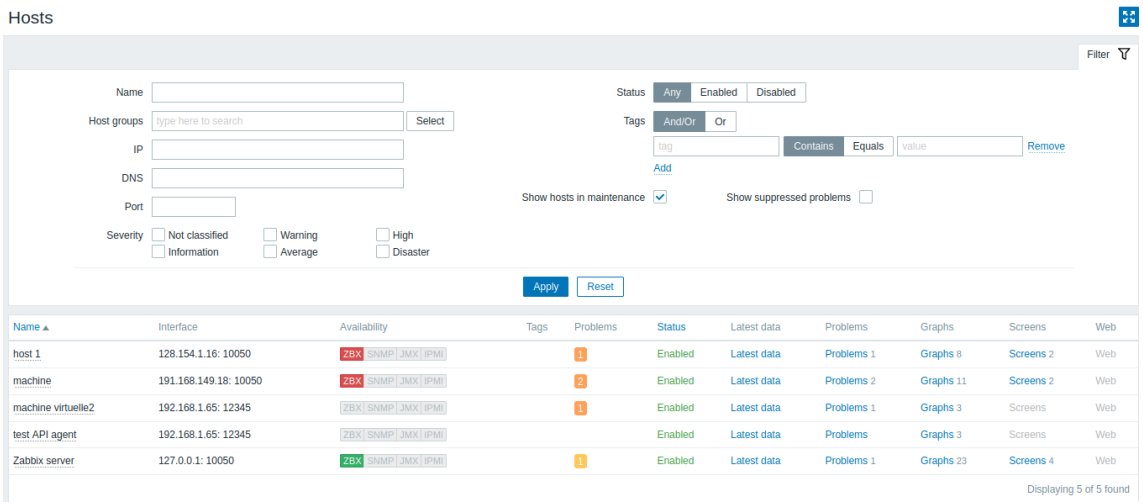

Fig. 4.14 : create host resultat

Si les données transmises sont incomplètes, cela génère une erreur.

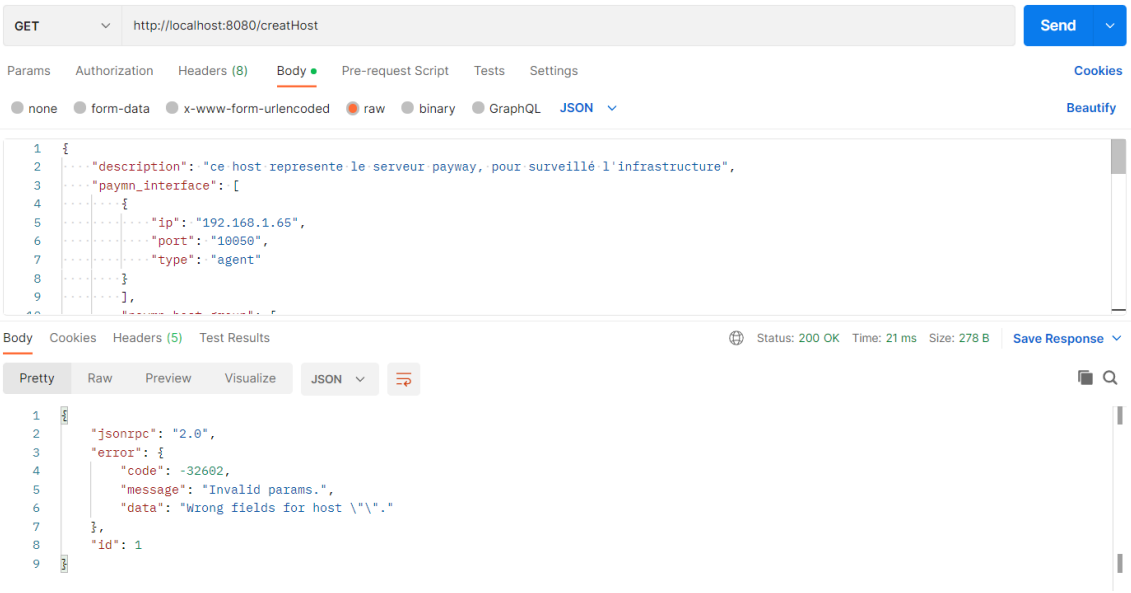

Fig. 4.15 : create host error

et on peut aussi récupérer la liste des hostes créent dans le serveur

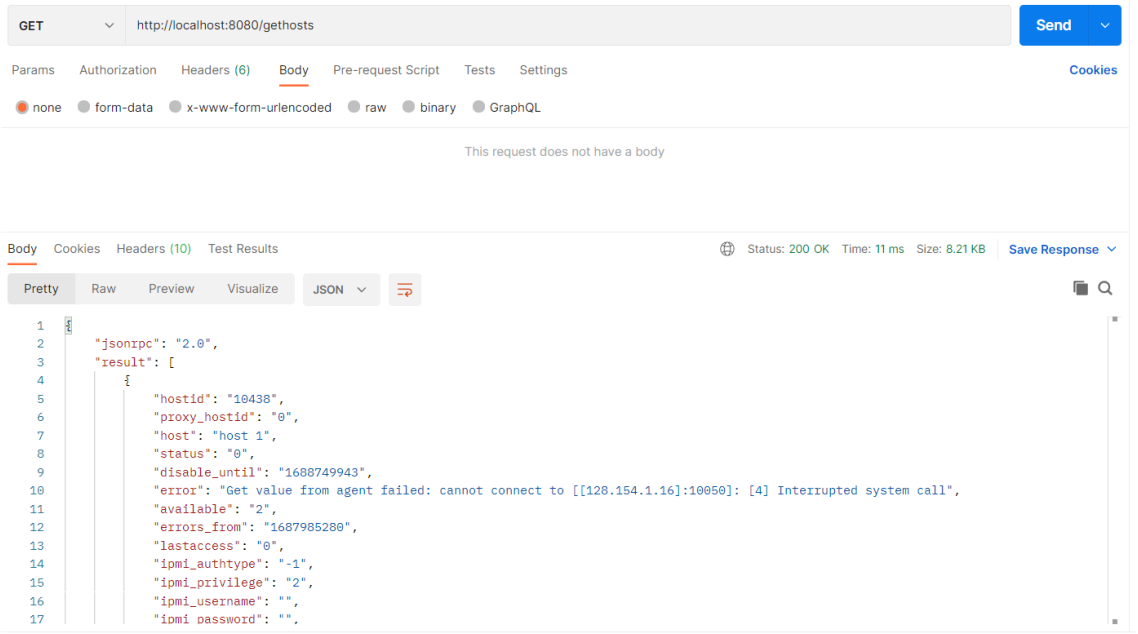

Fig. 4.16 : liste des hostes

on peut aussi supprimer un hoste par son ID

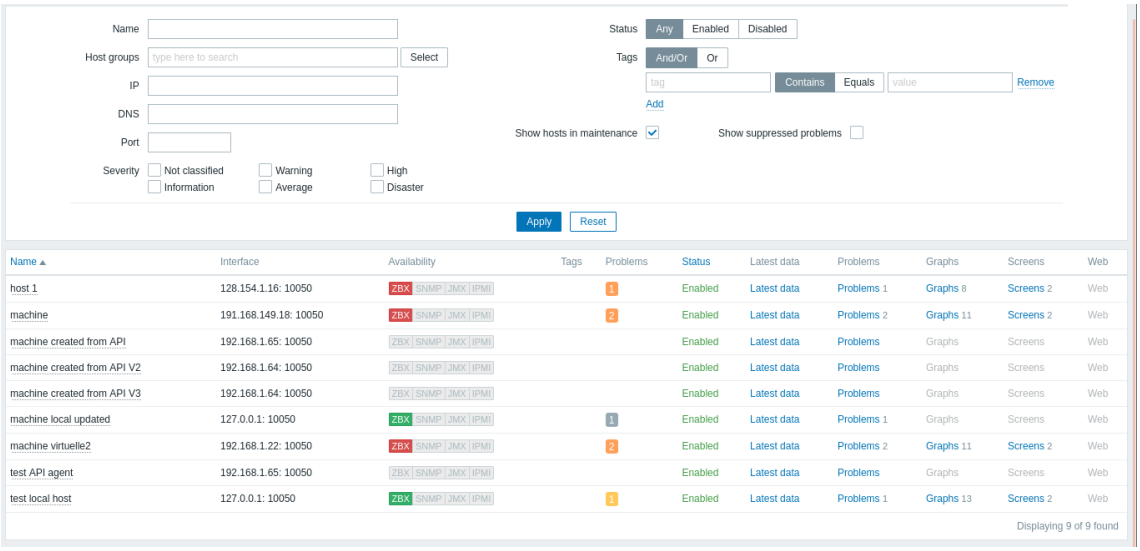

Fig. 4.17 : interface avant de supprimer l'host

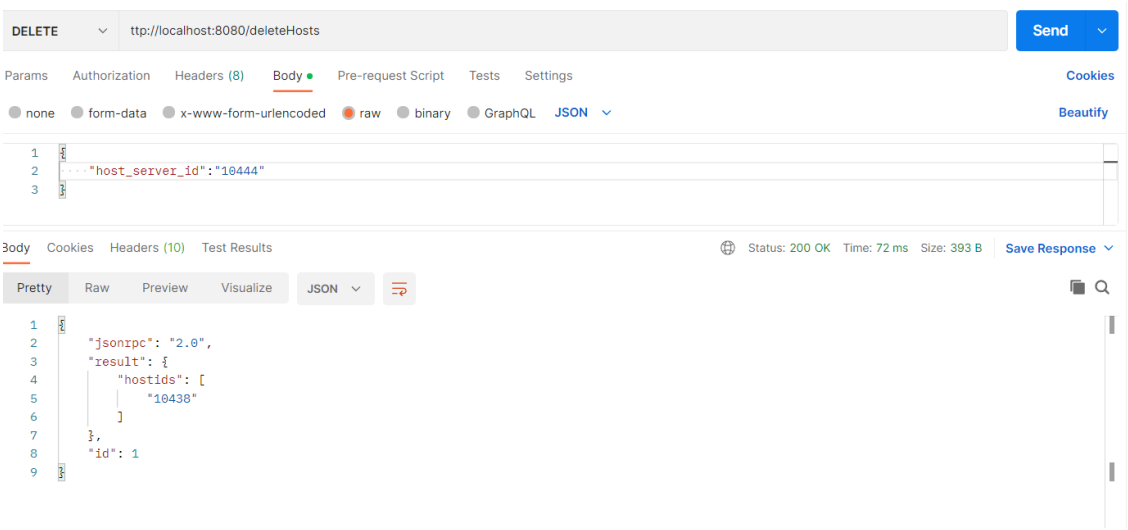

Fig. 4.18 : delete host

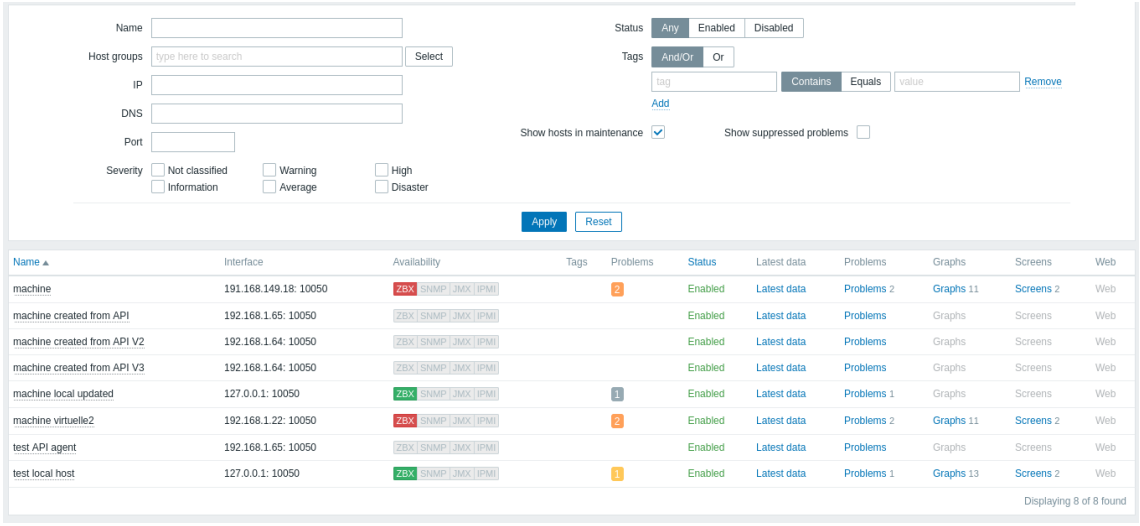

Fig. 4.19 : interface après la suppression du host

on peut aussi récupérer les informations d'un host par son nom.

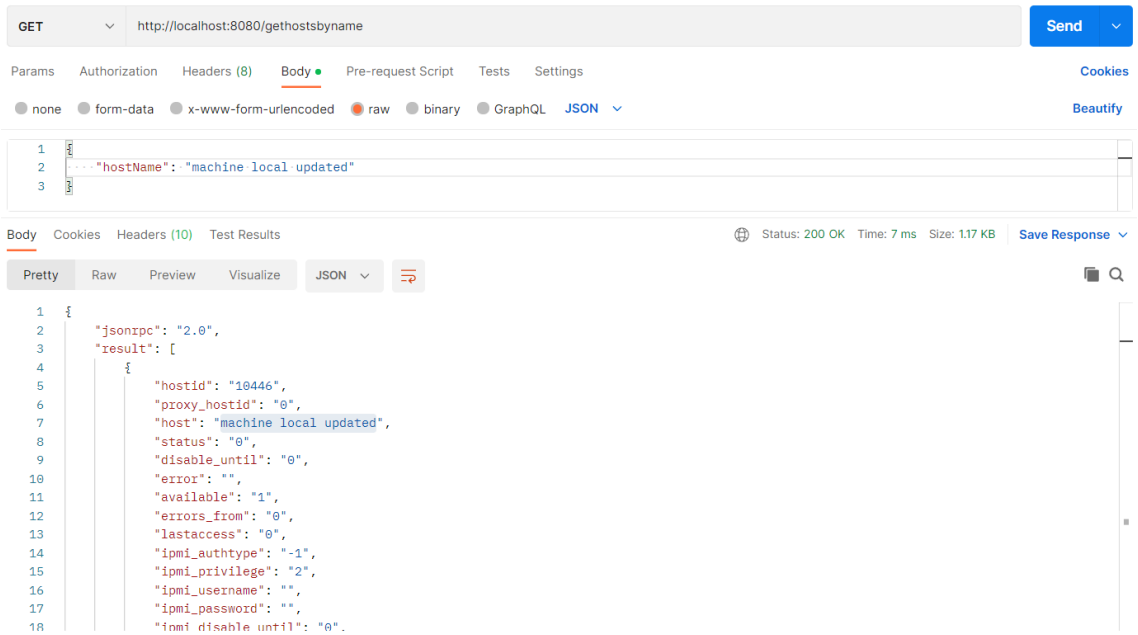

Fig. 4.20 : get host by name

l'endpoint suivant permet de retourner les hosts selon leurs gravités d'erreur

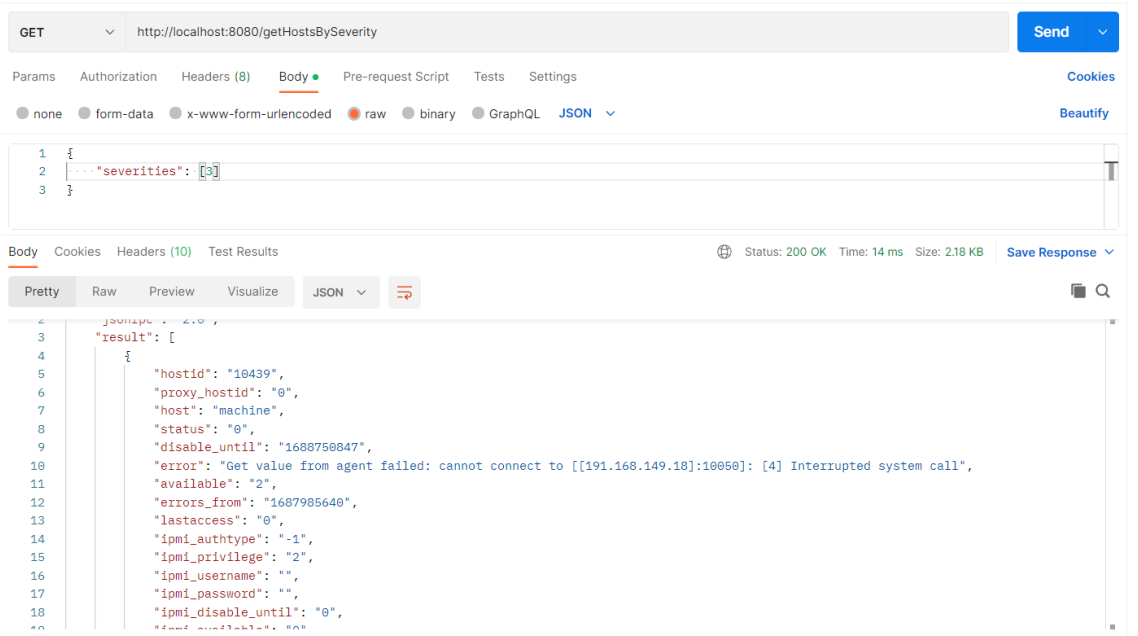

Fig. 4.21 : get host by severity

et on peut aussi modifier les informations d'un host

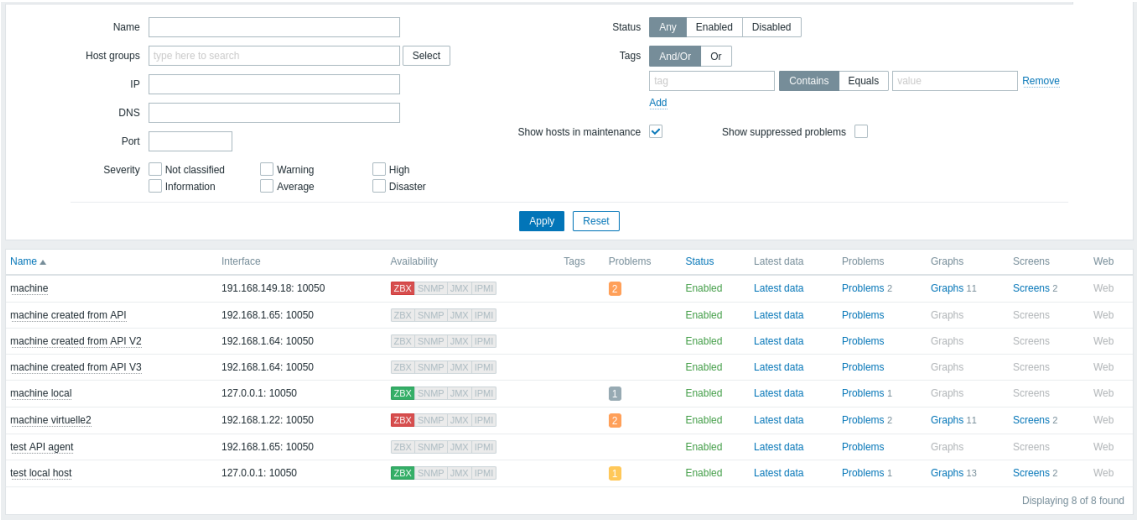

Fig. 4.22 : l'interface avant la modification

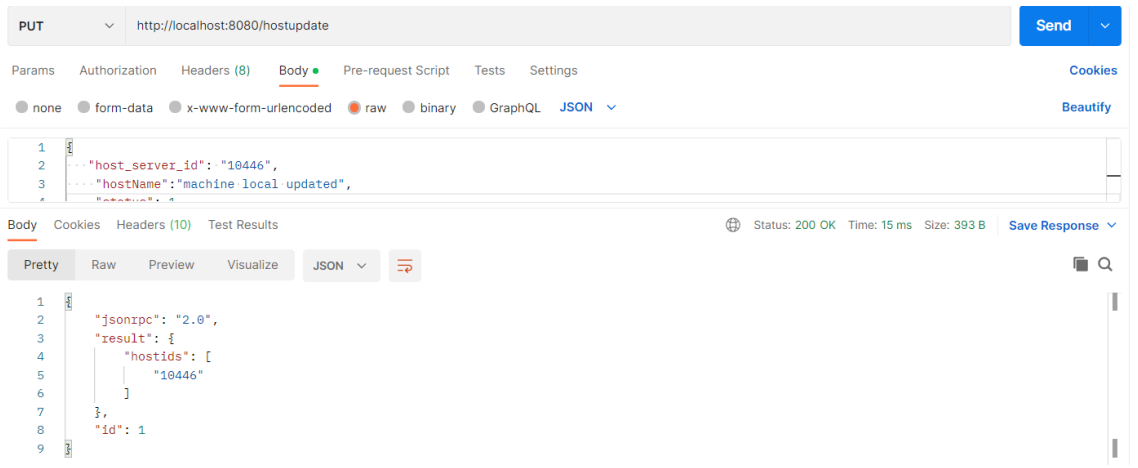

Fig. 4.23 : update host

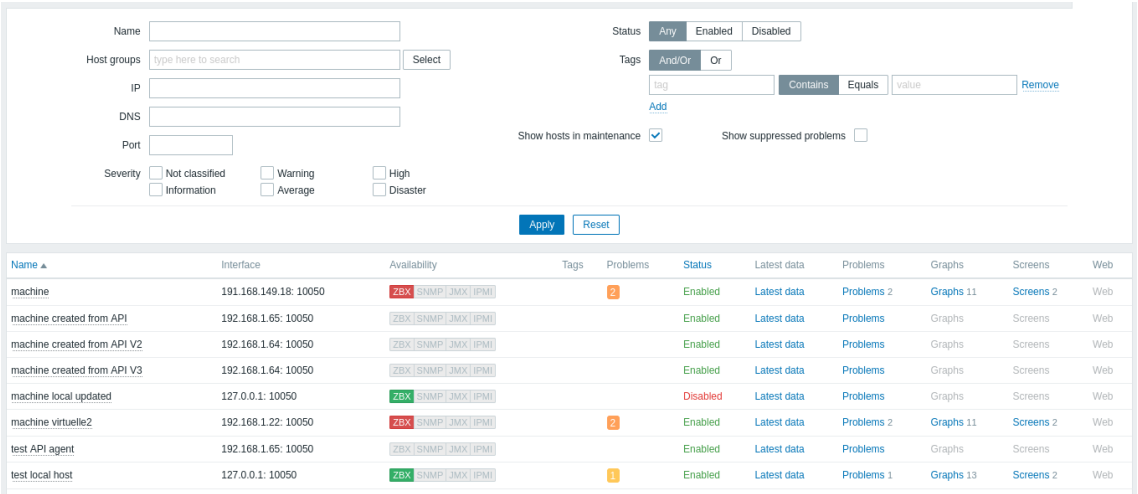

Fig. 4.24 : l'interface avant la modification

Dans le cas suivant,j'ai fait la configuration pour crée un item nomé "item pour mésuré l'éspace disque dur libre" dans la machine qui est déja créer (tast payway).

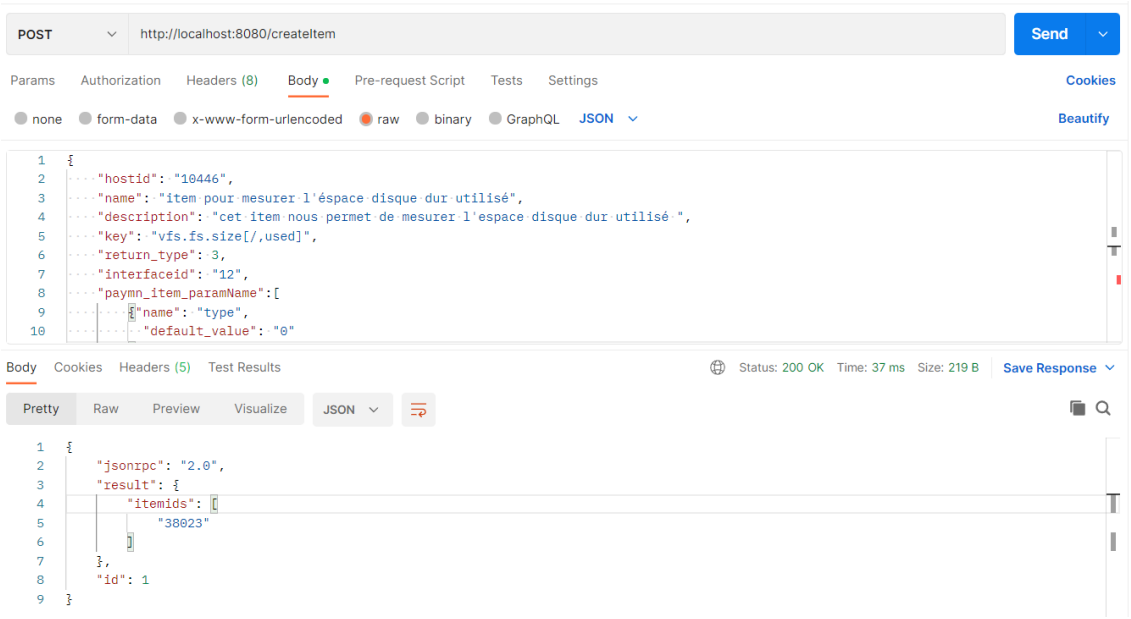

Fig. 4.25 : create item

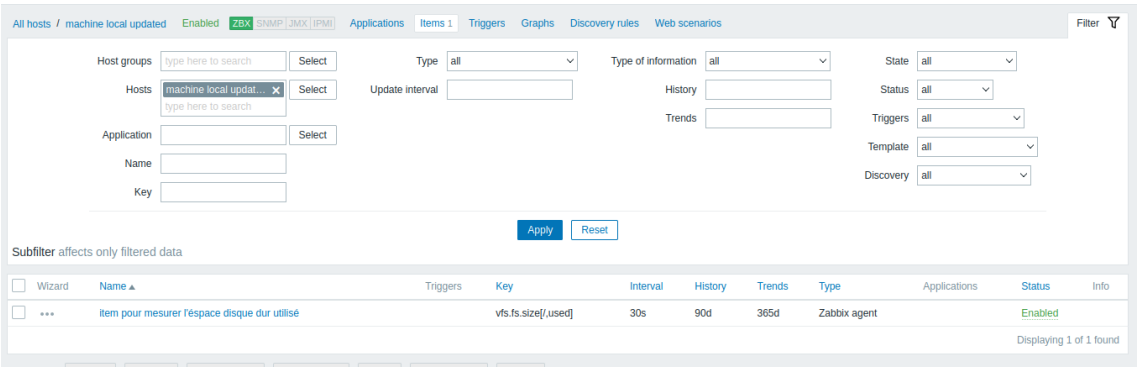

Fig. 4.26 : l'interface après la création de l'item

on peut aussi récupérer les informations de configuration d'un item par son clé.

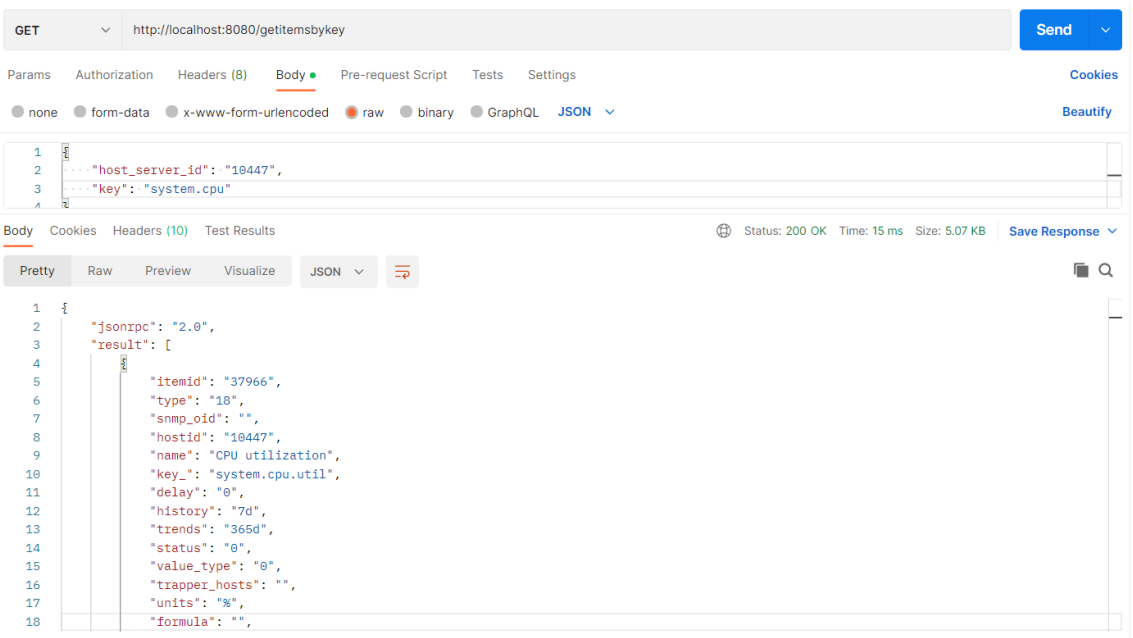

Fig. 4.27 : get item by key

et si on a plus besoin d'un item il est possible de le supprimer

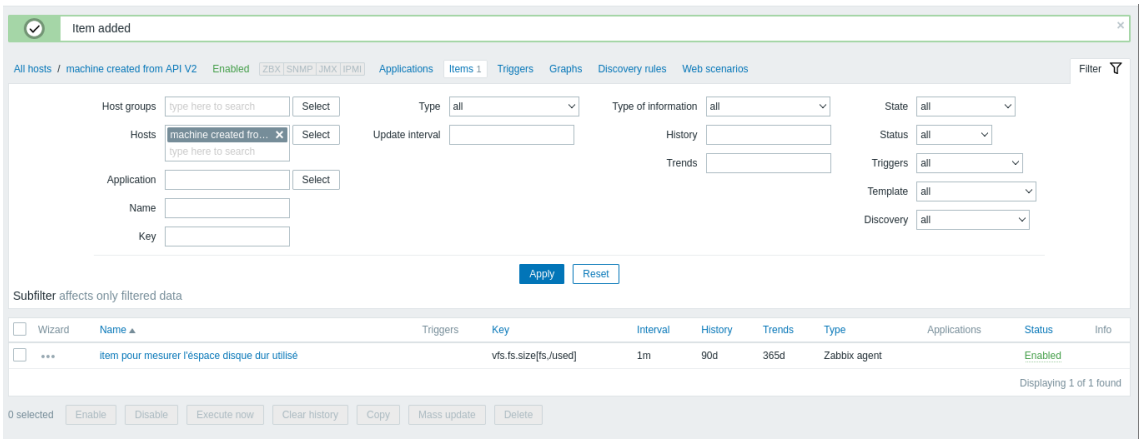

Fig. 4.28 : avant la suppression de l'item

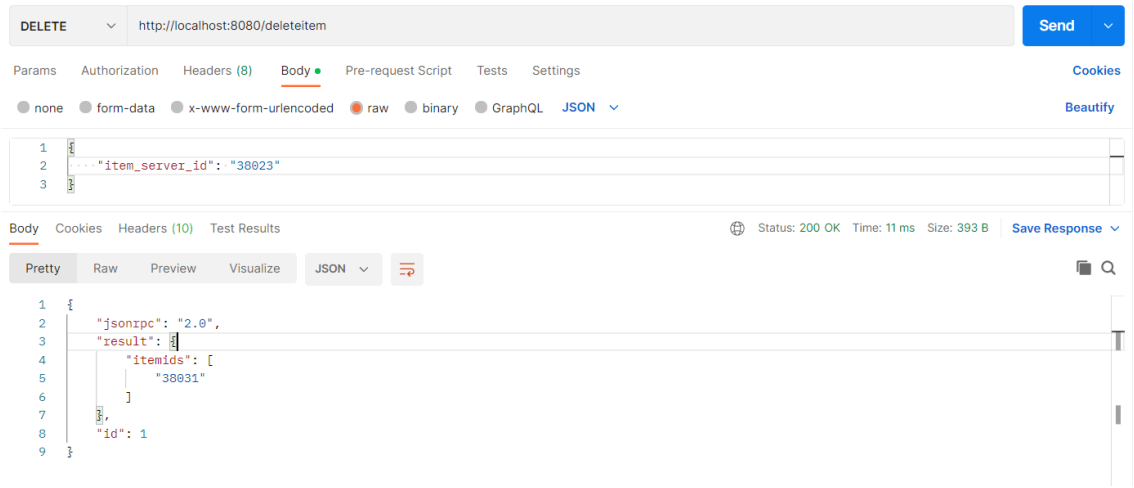

Fig. 4.29 : delete item

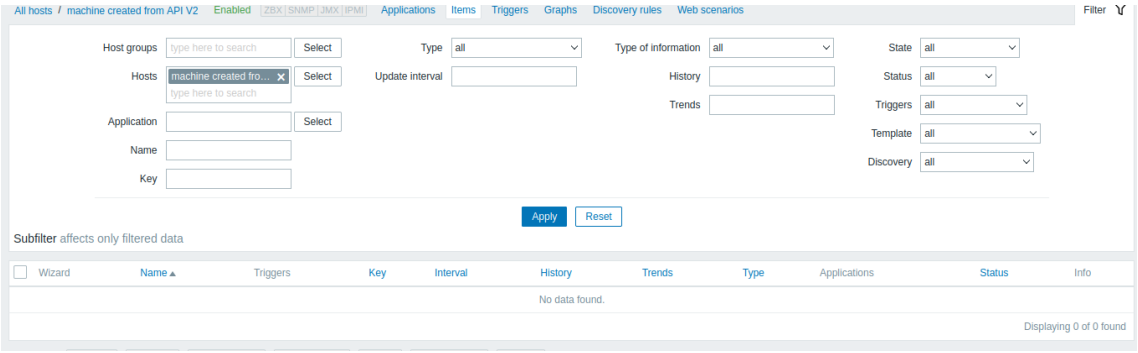

Fig. 4.30 : après la suppression de l'item

dans l'exemple suivant on a testé la creation d'un trigger nomé "consomation disk dur est élevé"

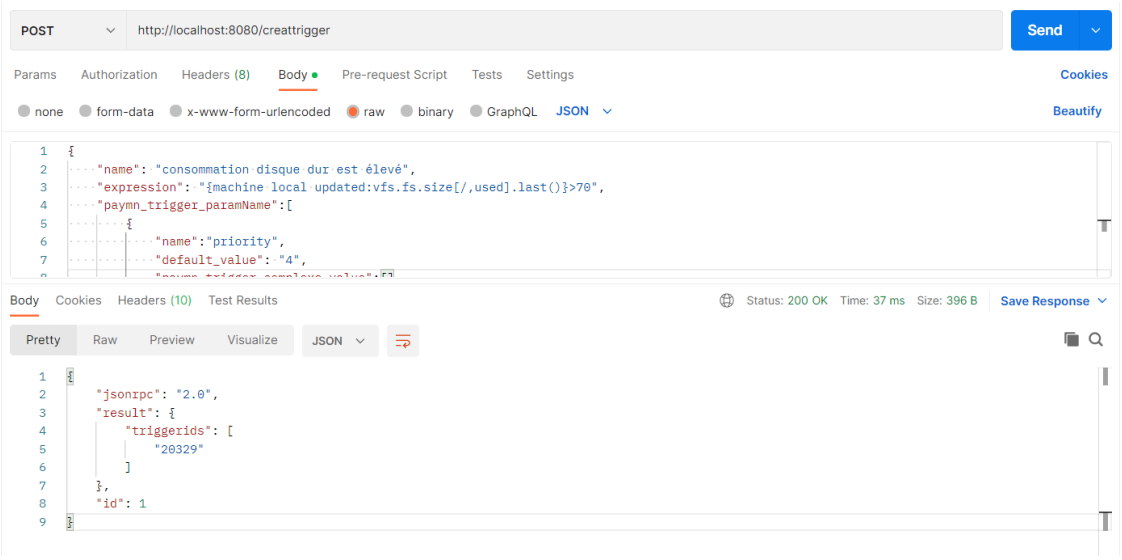

Fig. 4.31 : create trigger

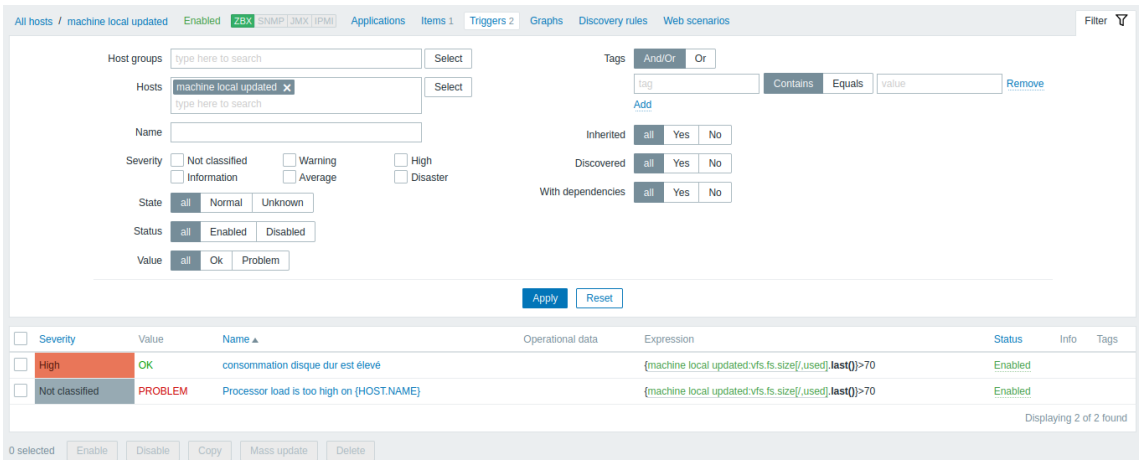

Fig. 4.32 : l'interface après la création du trigger

4.5 Conclusion

Dans ce chapitre nous avons travaillé sur l'environnement réel "trésor public de Madagascar", nous avons surveillé les trois serveurs composant le trésor public de Madagascar, et pour chaque serveur nous avons surveillé la partie infrastructure et applicatif en utilisant notre connecteur. et après avoir comprendre le fonctionnement de Zabbix on a développé notre conneteur qui permet de charger la configuration au niveau du serveur.

Conclusion générale et perspectives

Le projet visait à étudier les outils de surveillance existants, tester les outils choisis dans un environnement réel au sein du Trésor Public de Madagascar, et surveiller à la fois l'infrastructure (RAM, CPU, disque dur) et la partie applicative, notamment les fichiers JAR et le serveur Tomcat.

Au cours du projet, nous avons réalisé une recherche approfondie sur les outils de surveillance disponibles et avons sélectionné ceux qui répondaient le mieux aux besoins.

Grâce à ces outils, nous avons pu surveiller efficacement l'infrastructure en collectant des données sur l'utilisation de la RAM, du CPU et du disque dur. Cela nous a permis de détecter les problèmes de performance, d'identifier les goulets d'étranglement potentiels et de prendre des mesures préventives pour optimiser l'infrastructure.

Nous avons également surveillé la partie applicative en collectant des informations telles que le nombre de threads, le HEAP et le nombre de classes des fichiers JAR et du serveur Tomcat. Cette surveillance nous a aidés à évaluer les performances de l'application, à détecter les éventuelles fuites de mémoire ou les problèmes liés à la charge de travail.

Enfin, dans le cadre du projet, nous avons développé un connecteur personnalisé pour charger la configuration des outils de surveillance. Cela nous a permis de personnaliser les paramètres de surveillance en fonction des besoins des clients et d'assurer une intégration transparente entre les outils et le switch monétique.

L'une des perspectives que nous souhaitons explorer est l'ajout de fonctionnalités supplémentaires pour répondre aux besoins et aux attentes croissants des utilisateurs. Parmi ces fonctionnalités, deux aspects cruciaux se démarquent : la configuration du serveur d'e-mail et la gestion des utilisateurs

Bibliographie

[1] https://www.pay-logic.com/

[2] https://www.ionos.fr/digitalguide/sites-internet/developpement-web/modele-en-cascade/

[3] https://www.gantt.com/fr/

[4] IT Infrastructure Monitoring Tools A Complete Guide - 2021 Edition

[5] Network Monitoring Management Using Cacti https://nsrc.org/workshops/2020/ekiti-connect/netmgmt/en/cacti/cacti-from-packages.pdf

[6] https://crm.org/news/atera-review

[7] Zabbix Network Monitoring Essentials By Andrea Dalle Vacche , Andrea Dalle Vacche , Stefano Kewan Lee

[8] https://www.techtarget.com/searchitoperations/definition/Nagios

[9] Building a Monitoring Infrastructure with Nagios 1st Edition by David Josephsen

[10] https://www.nagios.org/ncpa/help.php

[11] Zabbix 5 IT Infrastructure Monitoring Cookbook : Explore the new features of Zabbix 5 for designing, building, and maintaining your Zabbix setup by Nathan Liefting (Author), Brian van Baekel (Author)

[12] https://docs.oracle.com/javase/8/docs/technotes/guides/management/agent.html

[13] Cours Mme Laila CHEIKHI - UML. département ingénieurie du web et informatique mobile.

[14] https://www.orsenna.fr/nagios-xi/#:~:text=Nagios%C2%AE%20XI%E2%84%A2%20est,tout%20en%20conservant%20son%20efficacit%C3%A9.

[15] Cours M.khalid nafil "Développement full stack". département ingénieurie du web et informatique mobile.

[16] https://openclassrooms.com/fr/courses/6573181-adoptez-les-api-rest-pour-vos-projets-web/7498761-utilisez-postman-pour-formuler-vos-requetes
https://fr.theastrologypage.com/intellij-idea

[17] https://www.purestorage.com/fr/knowledge/what-is-vmware.html

[18] IT Monitoring Gaps The Ultimate Step-By-Step Guide Paperback – November 30, 2021 by Gerardus Blokdyk (Author)

[19] Multimodal monitoring of Web servers https://ieeexplore.ieee.org/abstract/document/1022857

[20] Mastère Professionnel en Nouvelles Technologies des Télécommunications et Réseaux (N2TR)

[21] Instant Nagios Starter by Michael Guthrie

[22] Supervision avec Nagios https://www.ibisc.univ-evry.fr/~petit/Enseignement/Reseau/IPV6/nagios.pdf

[23] Zabbix System Overview https://www.initmax.cz/wp-content/uploads/2022/03/zabbix-overview-en.pdf

[24] Zabbix 7 IT Infrastructure Monitoring Cookbook : Explore the new features of Zabbix 7 for designing, building, and maintaining your Zabbix setup, 3rd Edition (English Edition)

[25] Oracle® Fusion Middleware Developing Manageable Applications Using JMX for Oracle WebLogic Server