

Université Cadi Ayyad
Ecole Nationale des Sciences Appliquées de Safi
Département Informatique, Réseaux et Télécommunications (IRT)

MEMOIRE

De

PROJET DE FIN D'ETUDES

Pour l'obtention du diplôme :
INGENIEUR D'ETAT
En Génie des Réseaux et Télécoms

Mise en Place d'une solution de Supervision Zabbix



Réalisé Par
TITOUR Rami

Effectué à
GOMOBILE SARL - CASABLANCA

Encadré à l'ENSAS par :
Prof. Anouar DALLI

Encadré à GOMOBILE par :
M.BENAABAD Mehdi
M.ELFAKIR Sami

Soutenu le07/2019 devant le jury composé de :

Mr., Professeur à l'ENSA de Safi
Mr., Professeur à l'ENSA de Safi
Mr., Professeur à l'ENSA de Safi

Année Universitaire : 2018/2019

Appréciation et Signature de l'encadrant

Dédicaces

A mes chers parents, je vous présente tous mes salutations, gratitude et amour pour leurs soutiens pendant toute ma carrière, mon échec avant mon succès. Je vous remercie gravement pour votre patience envers mes fautes commises et pour les efforts que vous avez consentis pour mon instruction et mon bien être. Que ce travail soit l'exaucement de vos vœux tant formulés, le fruit de vos innombrables sacrifices.

A ma fiancée qui a bien été à mes côtés tout au long de ce parcours qui m'a fourni l'effort et le soutien demandé pour que je sois aujourd'hui devant vous.

A ma famille pour vos encouragements et l'amour que vous nous portez.

A mes chers amis Yassine, Adnane, Saïd, Omar, en souvenir de nos sincère et profonde amitié et des moments agréables que nous avons passé ensemble, je vous souhaite courage et paix et une vie pleine de succès.

TITOUR Rami

Remerciements

La réalisation de ce mémoire a été possible grâce au concours de plusieurs personnes à qui je voudrais témoigner toute ma gratitude.

Je voudrais tout d'abord adresser toute ma reconnaissance à mon encadrant externe qui m'a guidé tout au long de ma période de PFE Monsieur BENAABAD Mehdi, pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter ma réflexion.

Je remercie aussi gravement le Président Directeur Générale de GOMOBILE Monsieur ELFAKIR Sami qui m'a donné l'opportunité de m'intégrer dans sa société et pour sa confiance dans mes compétences professionnelles.

Je tiens à remercier spécialement l'équipe GOMOBILE, M. OUBALLA Youssef, M. RAHMAOUI Abdelhamid, Mme.FOUIRINA Hanane et Mlle. MILAD Kamilia qui m'ont facilité l'intégration à l'équipe et qui m'ont aidé professionnellement et personnellement dans mon sujet.

Je désire aussi remercier les professeurs de l'école nationale des sciences appliqués, qui m'ont fourni les outils nécessaires à la réussite de mes études universitaires.

Je voudrais exprimer ma reconnaissance envers les amis et collègues qui m'ont apporté leur soutien moral et intellectuel tout au long de ma démarche.

Un grand merci à mon encadrant interne Monsieur DALLI Anouar pour ses conseils, ses contributions et son soutien tout au long de ma période de stage, ce qui a grandement facilité mon travail.

Résumé

Le présent document constitue une synthèse de mon travail réalisé dans le cadre de mon projet de fin d'études, au sein de l'organisme GOMOBILE. Ce projet a pour objectif d'optimiser, superviser et automatiser la détection des pannes systèmes et réseaux dans l'architecture de GOMOBILE.

Le présent projet avait pour mission la mise en place d'un système de supervision afin d'être préventif et l'identification des sources ciblées des pannes du systèmes de GOMOBILE.

En outre, le projet a pour deuxième et principale mission de préparer un système de supervision adéquat et avancer convenable aux expansions de l'architecture de GOMOBILE qui vise à augmenter leur capacité d'émission des appels sans avoir des futures contraintes. Face à ses objectifs les tâches consistent à :

- Etude préliminaire de l'architecture
- Conception de la solution
- Emulation de l'architecture et de la solution
- Mise en place de la solution

Abstract

The following document is a detailed conclusion for the work done during my end of study project at GOMOBILE, which had as main objectif setting up a sophisticated monitoring system capable of the automatic detection of all the system and network problems which can make the troubleshooting more organized and targeted.

Also the project had another mission which was controlling all the components and knowing their states at every seconde which will be facilitating the addition of any other materiel in the near futur and it will help GOMOBILE in the expansion procedure and to extend their capacity of calls per hour. The project was devised into four main steps :

- Studying and understanding GOMOBILE's architecture
- Setting up the solution's conception
- Emulating the architecture and testing the solution
- Setting up the solution

Liste des acronymes

IP : Internet Protocol

UDP : User Datagram Protocol

TCP : Transmission control Protocol

IAX : Inter-Asterisk eXchange

SIP : Session Initiation Protocol

IPBX : Internet Protocol Private Branch eXchange

RTP : Real-Time Transport Protocol

IETF : Internet Engineering Task Force

IMS : IP Multimedia Subsystem

CUBE : Cisco Unified Border Element

JMX : Java Management Extensions

SNMP : Simple Network Management Protocol

MIB : Management Information Base

KPI : Key Performance Indicators

Liste des tableaux

Tableau 1 : Comparaison des prérequis matériel	39
Tableau 2 : Méthode de calcul d'espace	39

Liste des figures

Figure 1 : Partenaire de GOMOBILE	14
Figure 2 : Architecture GOMOBILE	19
Figure 3 : Fonctionnement de Zabbix.....	27
Figure 4 : Requête envoyer par le Zabbix serveur au Zabbix agent	28
Figure 5 : Récupération de la liste des items.....	29
Figure 6 : Envoie des données collectées.....	30
Figure 7 : Principe du protocole SNMP	31
Figure 8 : L'authentification du SNMP.....	33
Figure 9 : Conception de la solution.....	41

Table des matières

Partie 1 - Etude préliminaire	Erreur ! Signet non défini.
I. Introduction de l'entreprise	13
A. Présentation générale.....	13
B. Organigramme de GOMOBILE	15
II. Etude de l'existant	16
A. Définition des composantes.....	16
B. Description de l'architecture	18
C. Critique de l'existant et solution proposée	20
III. Système de supervision	21
A. Introduction	21
B. Différents systèmes de supervision	22
1. PRTG.....	22
2. Nagios.....	23
3. Zabbix.....	24
C. Conclusion.....	24
IV. Analyse des besoins.....	25
A. Fonctionnement de Zabbix	25
1. Structure de Zabbix	25
2. Gestion des flux.....	27
3. SNMP et Zabbix.....	30
B. Etude des KPI's	34
1. Système :	34
2. Base de données :	36
3. Java :	37
4. Serveur Web :	37

C. Prérequis Matériels.....	38
1. Prérequis pour la mémoire.....	38
2. Prérequis CPU	38
3. Taille de la base de données	39

Partie 1

Etude préliminaire

I. Introduction de l'entreprise

A. Présentation générale

GOMOBILE est une startup technologique opérant dans le secteur de télécom fondée en 2013 après avoir remporté le prix startup world cup en 2016. GOMOBILE offre aux entreprises un produit innovant leur permettant de diffuser des appels à des milliers des contacts à travers des messages vocaux personnalisés (à la demande du client) interactif, ainsi que le suivie et reporting des campagnes effectués (taux de décrochement, minutes consommés...).

GOMOBILE a émis plus de 2 millions d'appels depuis juillet 2017 pour le compte des grands groupes leaders dans le secteur d'activité avec une capacité d'effectuer jusqu'à 4000 appels par heure. GOMOBILE donne à ses clients accès à la plateforme OSIX qui permet de gérer les messages vocaux à transmettre ainsi que leurs listes des numéros, en plus de cela générer des rapports de suivies des appels, ou bien à l'aide d'une API pour déclencher des appels à partir de système d'information externes. Parmi les cas d'usages des services de la société on trouve :

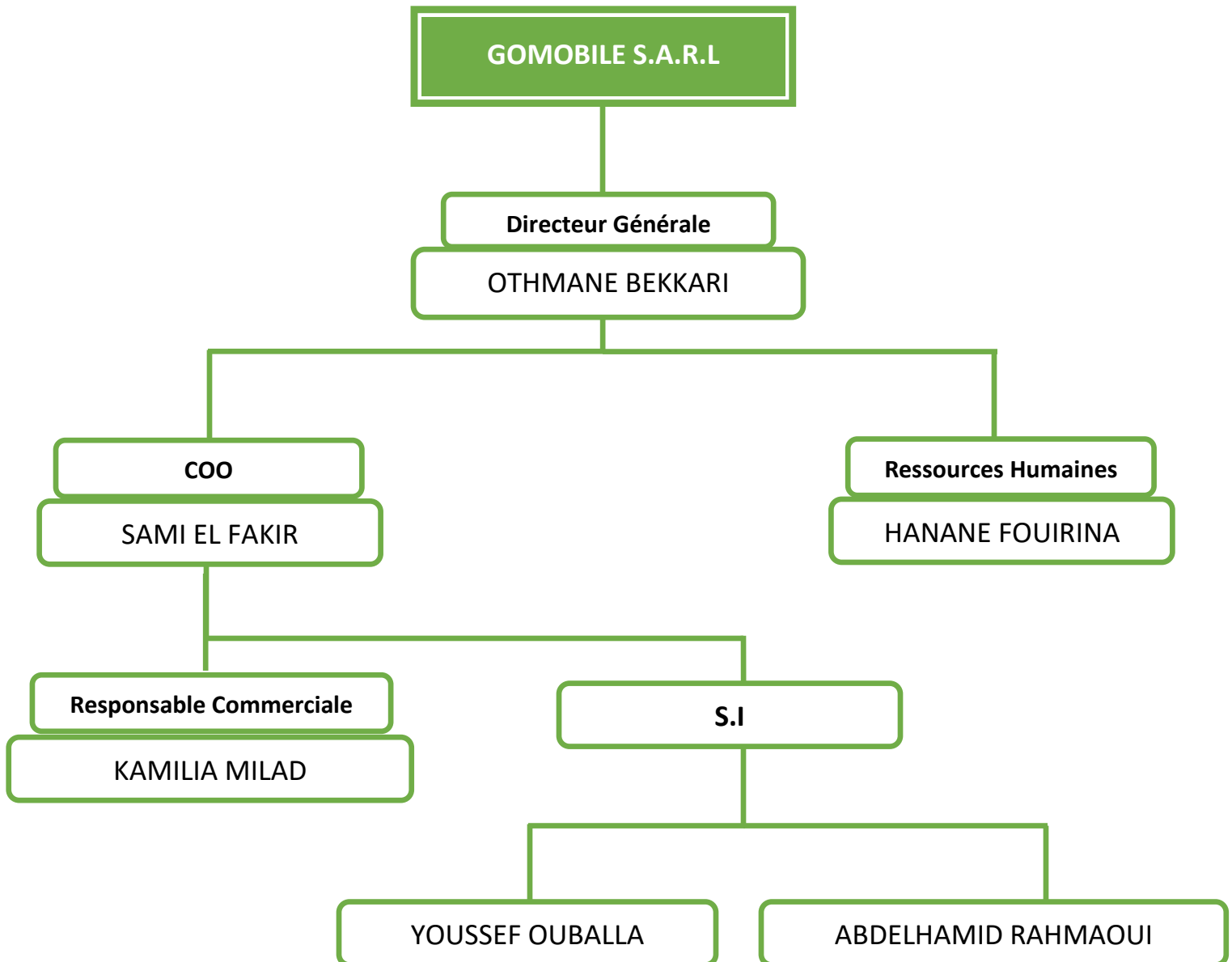
- Recouvrement de créances
- Rappel d'échéances
- Fidélisation de clients
- Qualification commerciale
- Enquêtes de satisfaction
- Notifications d'urgence
- Formation & Coaching
- Campagnes promotionnelles

Parmi les partenaires de Gomobile on trouve :



Figure 1 : Partenaire de GOMOBILE

B. Organigramme de GOMOBILE



II. Etude de l'existant

A. Définition des composantes

1. La VoIP et Asterisk :

La voix sur IP est le transport de la voix sur les réseaux IP sous forme des paquets en utilisant le protocole UDP car il procure de meilleurs délais d'envoi des paquets que TCP, car il n'utilise pas de contrôle de réception. Pour établir la VoIP on a besoin d'Asterisk qui jouera le rôle d'un IPBX, ce dernier est un logiciel libre publié et créé par Mark Spencer de la société Digium en 1999. Il fonctionne sous Linux, BSD, Mac OS X.

Asterisk utilise différents protocoles afin de faire de la téléphonie, tels que SIP ou encore H323 et IAX. Les protocoles SIP (Session Initiation Protocol) et H323 sont des protocoles qui permettent d'établir, modifier et terminer des sessions multimédia.

SIP est un protocole normalisé et standardisé par l'IETF. Il a été conçu pour établir, modifier et terminer des sessions multimédia. Il se charge de l'authentification et de la localisation des multiples participants mais également de la négociation sur les types de média utilisables par les différents participants. SIP ne transporte pas les données échangées durant la session comme la voix ou la vidéo. SIP étant indépendant de la transmission des données, tout type de données et des protocoles peut être utilisé pour cet échange. Cependant le protocole RTP (Real-time Transport Protocol) assure le plus souvent les sessions audios et vidéo.

2. MariaDB

C'est un système de gestion de base de données relationnel (SGBDR) disponible sous licence GPL. Il s'agit d'un nouveau logiciel crée du code source de MySQL. Il est programmé en C, C++, Perl, Bourne-Again shell par les programmeurs de MySQL afin de le rendre libre. Quand aux administrateurs qui utilise des serveurs CentOS vont automatiquement utiliser MariaDB.

Il supporte deux langages de programmation requête SQL et le SQL/PSM, permet de combiner des requêtes SQL et des instructions procédurales (boucles, conditions...), dans le but de créer des traitement complexes destinées à être stockés sur le serveur de base de données.

MariaDB utilise deux moteurs principaux MyISAM et InnoDB, dont InnoDB supporte les transactions et l'intégrité automatique des tables alors que MyISAM ne les supporte pas.

Le couple PHP/MySQL est très utilisé par les sites web et proposés par la majorité des hébergeurs web. Plus de la moitié des sites web fonctionnent sous Apache, qui est le plus souvent utilisé avec PHP et MySQL

3. PHP

C'est un langage de programmation libre, principalement utilisé dans la production des pages Web dynamiques à travers un serveur HTTP, c'est un langage impératif orienté objet et il peut même être utilisé comme langage de script en ligne de commande sans utiliser de serveur HTTP ni navigateur. Il est parmi les langages principaux utilisés par la majorité des développeurs des sites Web.

4. JAVA

La technologie Java définit à la fois un langage de programmation et une plateforme informatique. Créée par l'entreprise Sun Microsystems, et reprise par la société Oracle, la

technologie Java est indissociable du domaine de l'informatique et du web. On la retrouve donc sur les ordinateurs, mais aussi sur les téléphones mobiles, consoles de jeux, etc.

Elle généralement basé sur la syntaxe du langage C++, et un langage de programmation informatique orienté objet Java permet de développer des applications client-serveur. Les applications développées en java peuvent fonctionner sur différents systèmes d'exploitation. La popularité de l'environnement d'exécution Java dans les navigateurs, et le fait que Java dans les navigateurs soit indépendant de l'OS, ont rendu ce langage attrayant pour les pirates informatiques.

B. Description de l'architecture

GOMOBILE est en cours de migration vers la nouvelle architecture qui est plus structuré et dispose des matériaux adéquats et qui répandent aux besoins fonctionnels de GOMOBILE. L'ancienne architecture de GOMOOBILE possède moins de composant que d'autre société mais la complexité de l'architecture se résume dans le fonctionnement du système et les opérations exécuter pour détailler plus l'architecture de GOMOBILE on procédera à la définition détaillée des composantes et de leur fonctionnement :

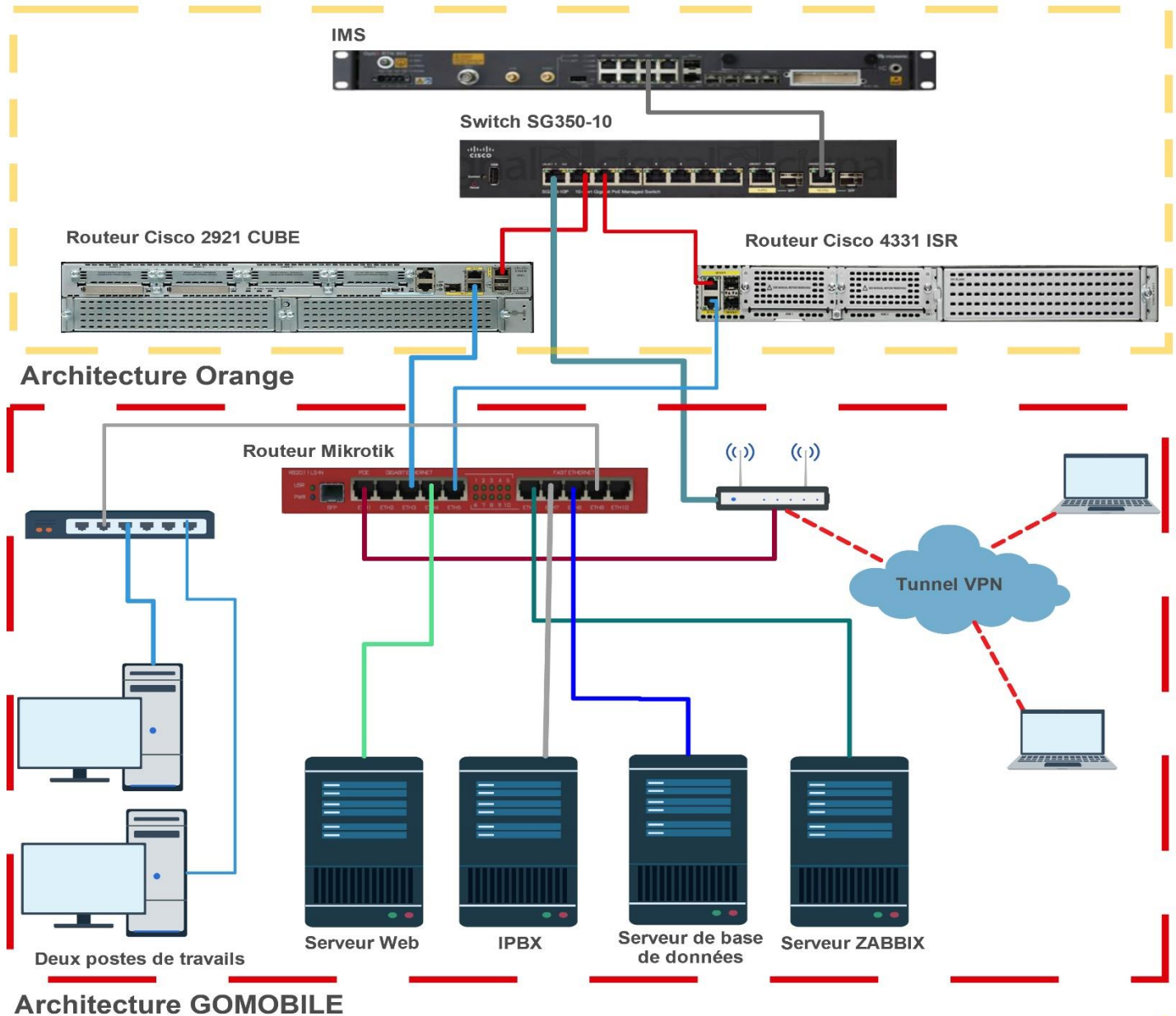


Figure 2 : Architecture GOMOBILE

- **Systèmes de gestion de base de données :** réservé au stockage de toutes les données et de manipuler ces données à travers le langage de requête SQL et les opérations exécuter ou bien à exécuter c'est l'une des composantes principales à GOMOBILE pour conserver les informations

des clients et génération des rapports des campagnes, nécessaire pour garantir la connexion des clients au serveur web et dont la communication entre le serveur de VoIP et le serveur web s'établit via un programme Java ce qui permet au client de lancer les campagnes d'appels facilement.

- **Serveur Web :** GOMOBILE utilise ce serveur pour déployer la plateforme OSIX qui est considéré comme une application web de lancement des campagnes et de poursuite des appels ainsi la génération des rapports contenant toutes les informations sur le déroulement des campagnes.
- **Serveur de Téléphonie sur IP :** C'est un serveur permettant la transmission de voix sur des réseaux IP, il existe évidemment plusieurs solutions et logiciel qui utilise ce concept, dans notre cas GOMOBILE utilise l'IPBX Asterisk ce dernier qui permet la génération des appels téléphoniques depuis la plateforme OSIX lors de l'exécution du programme java qui apporte les informations sur la campagne depuis la base de données.

La nouvelle architecture a pour but l'optimisation du système de GOMOBILE qui cible à augmenter le taux des appels émis par heure

C. Critique de l'existant et solution proposée

Les systèmes d'informations sont plus exposés aux problèmes ce qui rend l'identification de ces derniers très difficile et parfois presque impossible pour les administrateurs systèmes. GOMOBILE souffre souvent du problème de dysfonctionnement des appels dynamique lors des campagnes à cause de l'augmentation de l'utilisation de la RAM ou bien du CPU à cause de

l'exécution de plusieurs processus et de la grande quantité des requêtes entre la base de données, le serveur web et le programme Java dans la période du surcharge comme étant lancer des compagnes simultanément et préparer plusieurs scénarios à se mettre dans la base de données ce qui nécessite mettre en pause toutes les compagnes et un redémarrage manuel du système contenant le programme Java et le serveur de VoIP.

Toutes pannes ou arrêt du système coûte très chers pour la société GOMOBILE, alors ils ont décidé de changer la stratégie du traitement et d'automatiser leurs détection des problèmes afin de garantir la disponibilité et la fiabilité des services offerts, la stabilité du système est l'une des services les plus demandés au marché, donc G ont choisi une approche d'être préventif et de connaître l'état des équipements et services réseaux à tout moment pour améliorer la résolution des pannes et là rendre momentanée et plus facile pour les administrateurs du systèmes.

Alors dans ce cadre vient l'idée de la mise en place d'un système de supervision pour optimiser la gestion des risques du système et permettre la détection automatique des pannes et se notifier de l'état du système à chaque instant.

III. Système de supervision

A. Introduction

C'est un système qui permet la mesure du fonctionnement des équipements périodiquement et fait le poursuit des états de ces systèmes soit qu'ils sont des équipements ou bien des composantes réseaux. Quand on est en train de faire une comparaison entre des outils de supervision il faut se baser sur les points forts et les inconvénients du système en termes de la supervision des réseaux, serveurs et infrastructure mais avant d'entamer la comparaison il faut

spécifier les critères qui feront la différence entre nos systèmes de supervision parmi ceux les plus importants sont :

- Support
- Facilité d configuration et l'intuitivité (aptitude à comprendre très rapidement et envisager le problème le plus rapidement possible)
- Évolutivité
- Découverte automatique du réseau et cartographie
- Prise en charge étendue des fournisseurs de matériel
- Gestion des alertes, rapports et notifications (Email/SMS)
- Interface utilisateur facile à utiliser

B. Différents systèmes de supervision

Il existe plusieurs logiciels de supervision dans cette partie on fera une petite comparaison sur les services fournissent et en conclure le système de supervision adopter en fin de cette partie :

1. PRTG

PRTG est aussi les systèmes de supervision les plus utilisés au monde entier par les grandes entreprises car il offre plusieurs services et fonctionnalités qui sont prêt à être utilisés et développé par les fournisseurs de services, ce qui le rend payant et près à l'utilisation dès

l'abonnement avec le minimum des modifications à effectuer, ces caractéristiques et ces fonctionnalités offertes sont comme suit :

- Payant
- L'Analyse de la bande passante et le flux est complément supporter
- L'intuitivité de son interface est disponible contrairement à Zabbix qui offrent ce qui est basic
- Cartographie disponible
- VoIP et QoS support est disponible contrairement à Zabbix qui utilise des scripts personnalisés
- Possède une application mobile

2. Nagios

Nagios est un logiciel ordonnanceur de supervision d'infrastructure informatique, les systèmes et réseaux avec un système d'alerte. Néanmoins la version libre offre des services limités par rapport à celle payante pourtant il reste des logiciels de supervision les plus utilisés au monde entier parmi. On fera une comparaison des services entre la version payante et la version libre :

- Nagios XI offre des fonctionnalités plus avancées dans l'interface des utilisateurs
- La génération des rapports
- La version libre ne supporte pas l'envoi et la réception des Traps SNMP et l'intégration des solutions

- La possibilité de personnalisation de la configuration est hors porté dans la version libre de Nagios
- Le mappage du réseau est très basic

3. Zabbix

Zabbix est un logiciel de supervision complètement libre il offre tous ces services à tous ces utilisateurs avec la possibilité de personnalisation et le choix d'adopter n'importe quelle architecture voulu selon le besoin, c'est un logiciel adéquat et simple à l'utiliser avec moins de problèmes fréquenter lors de l'installation ou bien la configuration, ces caractéristiques sont :

- La personnalisation de l'interface de gestion
- Support du VoIP et QoS a besoin des scripts pour être fournit
- Pour la personnalisation des services on a besoin de développer nos propres modèles
- La découverte automatique n'est pas fonctionnelle à 100% en comparaison avec le PRTG

C. Conclusion

D'après la demande du client GOMOBILE, on s'est mis d'accord à utiliser un système de supervision qui respecte plusieurs conditions les plus importantes sont la gratuité, l'évolutivité et une qualité supérieure des services fournis, et le seul à garantir toutes ces fonctionnalités ainsi vues que la société a déjà des connaissances sur Zabbix et la présence d'une documentation large sur son fonctionnement encourage le choix de Zabbix comme l'outil de supervision idéale pour GOMOBILE.

IV. Analyse des besoins

A. Fonctionnement de Zabbix

1. Structure de Zabbix

Zabbix est un logiciel libre créé en 1998 par Alexei Vladishev et activement développé et soutenu par ZABBIX SIA qui permet la supervision des divers services réseaux, systèmes d'exploitation, matériels et applications, le serveur Zabbix est considéré polyvalent c'est-à-dire peut superviser et vérifier les statuts d'une multitude de services réseaux.

Zabbix est constitué généralement des composantes suivantes : interface de gestion, serveur de traitement, base de données, Zabbix Agent, Zabbix Proxy.

- **Le serveur de traitement** : est la composante la plus importante dans Zabbix c'est celui qui effectue toutes les opérations qui ont relations avec Trap & Poll des informations et même il permet de calculer les déclencheurs et envoie des notifications aux utilisateurs. La collection des données depuis les matériaux superviser à l'aide des agents Zabbix ou des proxy Zabbix avant de les stocker sur sa base de données, il contient aussi toutes les configurations nécessaires.
- **L'interface graphique** : permet de visualiser toutes les informations et configurations essentielles ainsi qu'il facilite la gestion des composantes surveiller ces informations qui sont actualisés à tous moments spécifier et de même pour les notifications l'interface graphique permet de visualiser les problèmes déclencher et trouver la cause de ses problèmes facilement.

- **La base de données :** est pour stocker toutes les informations à propos des hôtes surveiller accumulé par le serveur de données ou par le proxy.
- **Zabbix Agent :** est nécessaire pour établir la communication entre le serveur et l'équipement à surveiller car il est l'outil de collection des informations et les transmettent au serveur Zabbix afin de les traiter ultérieurement.
- **Zabbix Proxy :** est utilisé dans la supervision centralisés des sites distants et peut aussi être utilisé pour réduire les charges sur le serveur.
- **Zabbix JavaGateway :** est daemon Zabbix et un support natif programmer en java pour monitorer les applications JMX et collecter les différentes valeurs des programmes sur une hôte, le serveur Zabbix envoie des requêtes au Zabbix JavaGateway qui utilise l'API de gestion JMX pour surveiller les hôtes à distance, le JavaGateway accepte les connexions provenant du serveur Zabbix ou proxy.
- **SNMP Agent :** Il est généralement utilisé pour la supervision des imprimantes, commutateurs et routeurs qui ont le SNMP activer, cet agent exécute ces contrôles seulement sur le protocole UDP.

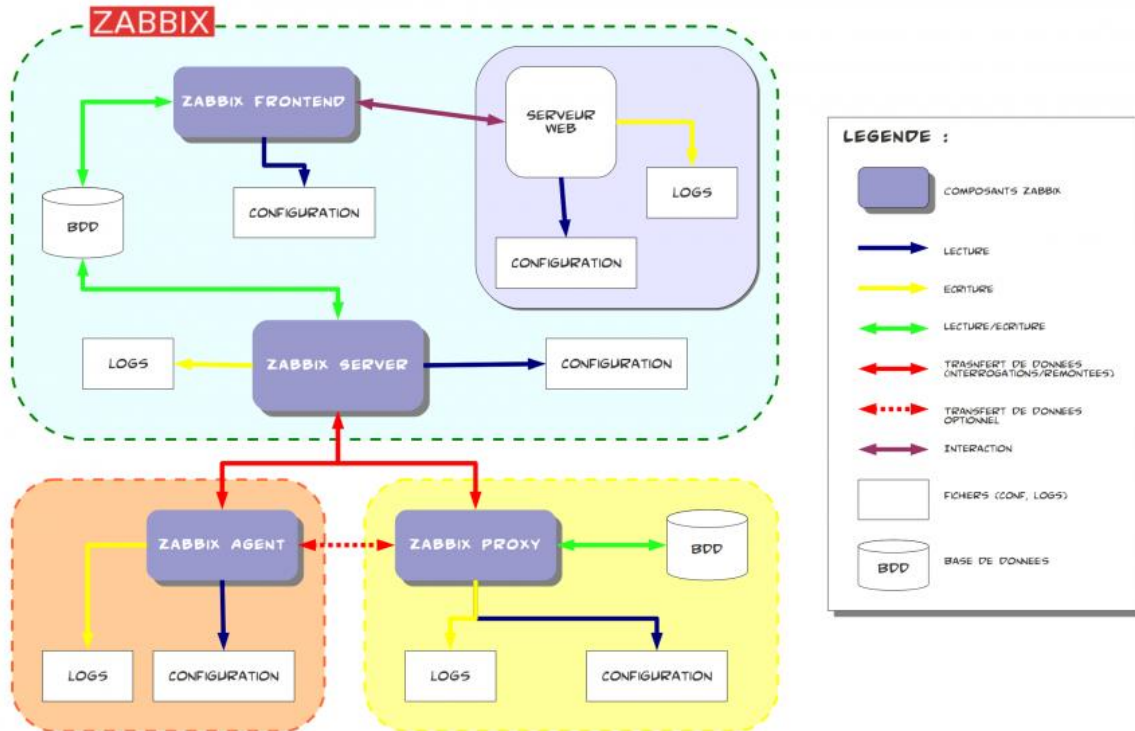


Figure 3 : Fonctionnement de Zabbix

Ces trois parties constitue le cœur de Zabbix et sans eux il ne peut pas fonctionner chacun d'eux à un rôle majeur dans le fonctionnement de Zabbix on peut même conclure que ce sont des éléments complémentaires comme le montre la figure au-dessus.

2. Gestion des flux

Zabbix utilise le protocole JSON pour communiquer avec les Zabbix agents pour exécuter des checks passifs, le serveur Zabbix lance des requêtes vers l'agent et l'agent envoie sa réponse

par une mesure qui est ensuite stocker dans la base de données pour illustrer cette procédure en détail voici une figure qui montre plus sur cette communication :

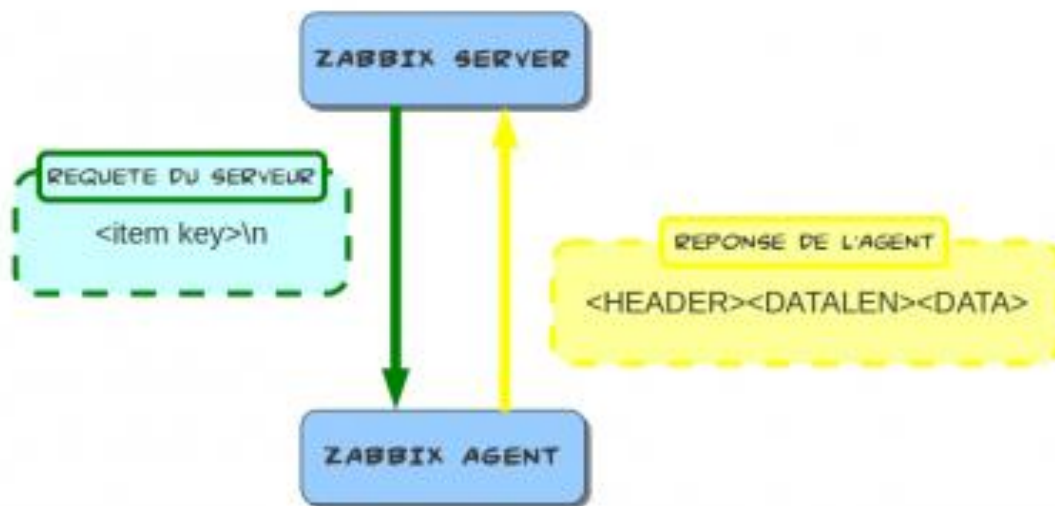


Figure 4 : Requête envoyer par le Zabbix serveur au Zabbix agent

Le fonctionnement des checks actives est un peu différent à celui des checks passifs, les checks actifs n'attendent pas la requête du serveur pour transmettre les informations car ils effectuent des tests périodiques et les transmettent au serveur, le processus de fonctionnement est composé de deux parties :

- **Récupération de la liste des items** : chaque agent démarré en mode actif effectue une requête au serveur pour avoir la liste des items définis pour son hôte et commencer la collection des informations.

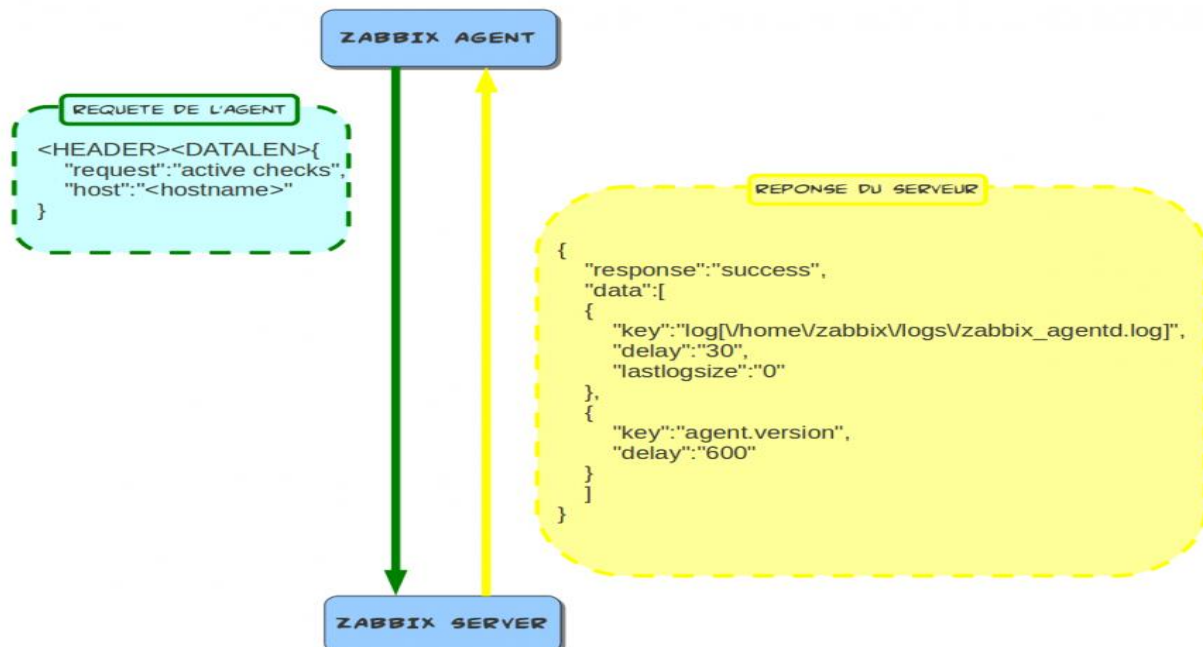


Figure 5 : Récupération de la liste des items

- **Envoi des données collectées :** une fois la communication entre le serveur et l'agent est établit ce dernier procède à la collection des informations à propos de tous les items surveiller et les transmettent au serveur pour les traiter et les stocker sur sa base de données ainsi que les afficher sur son interface graphique aux utilisateurs, en fin de compte le serveur répond l'agent par une requête indiquant la bonne réception des informations. La voilà une figure au-dessous montre un exemple des échanges entre le serveur et l'agent.

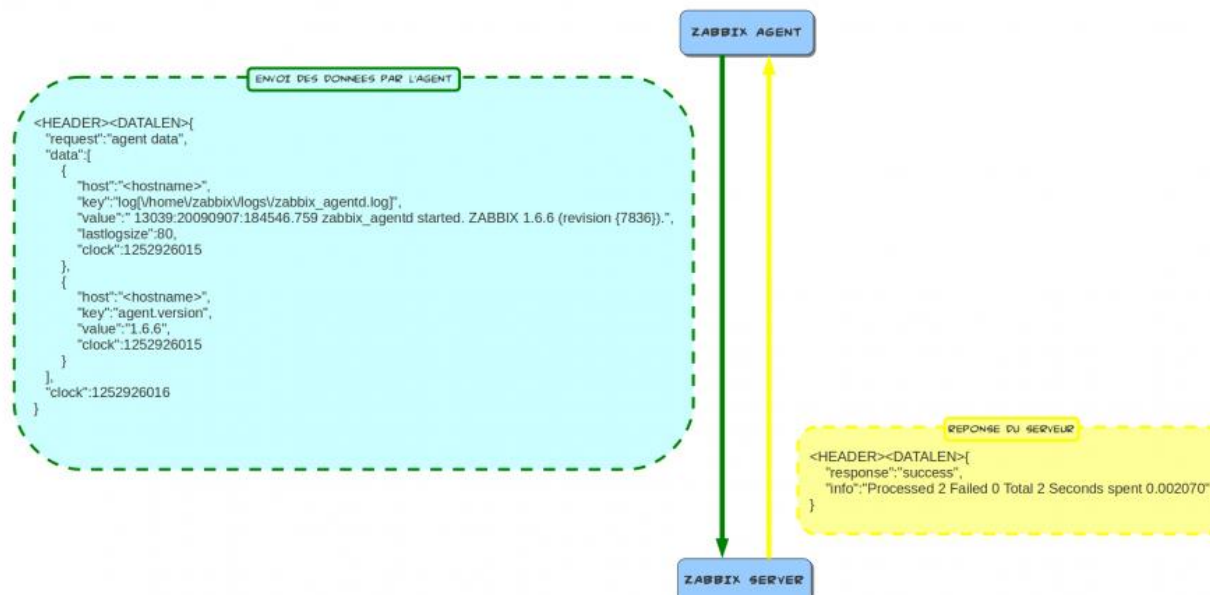


Figure 6 : Envoie des données collectées

3. SNMP et Zabbix

Signifie Protocole simple de gestion de réseau c'est un protocole de communications qui permet aux administrateurs réseau de gérer, diagnostiquer et superviser des problèmes réseaux et matériels distant Ce protocole est principalement utilisé par Zabbix pour monitorer les matériaux réseaux.

SNMP : Protocole

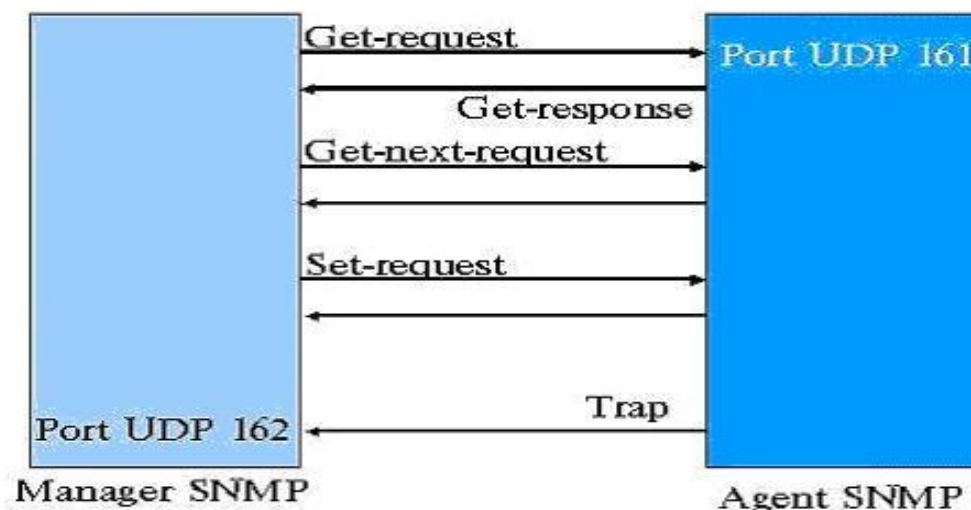


Figure 7 : Principe du protocole SNMP

L'environnement de gestion SNMP est constitué de plusieurs composantes : la station de supervision, les éléments actifs du réseau, les variables MIB et un protocole.

Le système de gestion de réseau est basé sur deux éléments principaux : un superviseur et des agents. Le superviseur est la console qui permet à l'administrateur réseau d'exécuter des requêtes de management. Les agents sont des entités qui se trouvent au niveau de chaque interface connectant l'équipement managé au réseau et permettant de récupérer des informations sur différents objets. SNMP permet la communication entre le superviseur et les agents afin de collecter les objets souhaités dans la MIB, qui est une base de données (Management Information Base) réservé pour classer les objets collecter. Et le protocole qui permet à la station de supervision d'aller chercher les informations sur les éléments de réseaux et de recevoir des alertes provenant de ces mêmes éléments. Le protocole SNMP est basé su un fonctionnement asymétrique. Il est constitué d'un ensemble de requêtes, de réponses et d'un nombre limité

d'alertes. Le manager envoie des requêtes à l'agent, lequel retourne des réponses. Lorsqu'un événement anormal surgit sur l'élément réseau, l'agent envoie une alerte (trap) au manager. SNMP utilise le protocole UDP. Le port 161 est utilisé par l'agent pour recevoir les requêtes de la station de gestion, et le port 162 est réservé pour la station de gestion pour recevoir les alertes des agents.

Il existe plusieurs versions de SNMP les fonctionnalités de chacune différents voici une petite présentation des versions et leurs avantages et inconvénients :

- **SNMPv1 (ancien standard) :** La sécurité de cette version est triviale, car la seule vérification exécuter et celle de la chaîne de caractère « Community ». SNMPsec : la version qui ajoute au protocole SNMPv1 la sécurité qui sera basé sur les groupes.
- **SNMPv2 :** cette version possède plusieurs sous-versions expérimental comme étant SNMPv2p, SNMPv2c, SNMPv2u et SNMPv* elles sont créées principalement au début pour mettre à jour les opérations et la sécurité du SNMPv1 en rendant l'utilisation du SNMPsec nécessaire c'est-à-dire rendre la sécurité basée sur les groupes. Puis depuis cette version une autre s'est créée SNMPv2u dans cette version la sécurité est basée sur les usagers, puis la version SNMPv2* qui combine entre SNPv2p et SNMPv2u.
- **SNMPv3 (standard actuel) :** Cette version vise principalement à inclure la sécurité des transactions alors qu'elle comprend l'identification des parties qui communiquent et l'assurance que la conversation soit privée, même si elle passe par un réseau public, et elle est basée sur deux concepts USM et VACM

Parmi ces versions, la plus utilisé est la version 1, alors que plusieurs version 2 ont été proposés mais sans qu'aucune d'eux ne soit adopter comme standard, Les trames dans la version 1 sont encodés en ASN.1 alors que les requêtes et les réponses ont les mêmes formats.

Pour mieux expliquer le fonctionnement et la sécurité de la version 3 on vise à définir USM et VACM et pour lesquelles on réservera la partie suivante :

Commençant par l'USM dans ce modèle de sécurité trois mécanismes sont utilisés pour empêcher tous les types d'attaques.

- **L'authentification** : empêche l'attaqueur de changer le paquet SNMPv3 en cours de transmission et de valider le mot de passe de la personne qui transmet la requête.

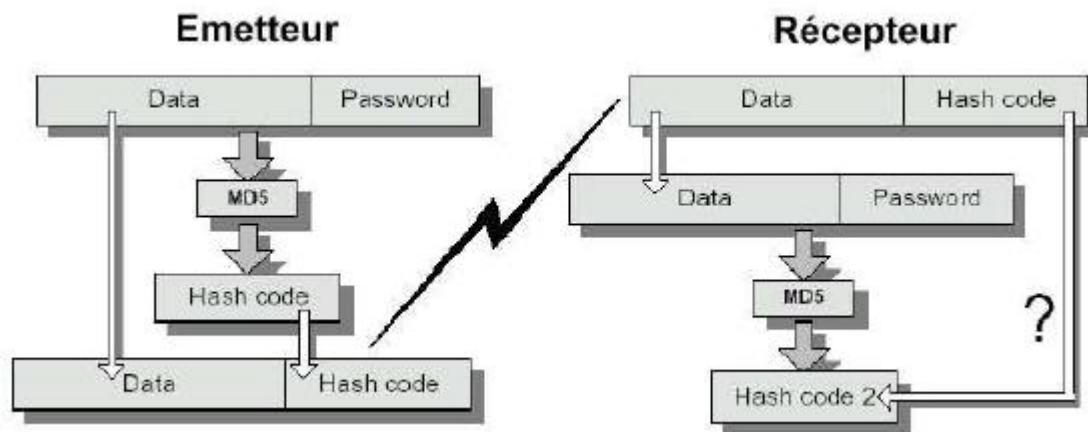


Figure 8 : L'authentification du SNMP

Le serveur Zabbix interroge les périphériques SNMP pour obtenir plusieurs valeurs dans une seule requête, ceci affecte plusieurs types d'éléments SNMP ou y'en a 3 types :

- Les éléments SNMP standard
- Eléments SNMP avec index dynamiques
- Règles de découverte de bas niveau SNMP

Tous les éléments SNMP d'une même interface avec des paramètres identiques doivent être interrogés en même temps. Les deux premiers types d'éléments sont pris par les enquêteurs par lots de 128 éléments au plus, alors que les règles de découverte de bas niveau sont traitées individuellement. Dans le bas niveau, il existe deux types d'opérations effectuées pour interroger des valeurs :

- Obtenir plusieurs objets spécifiés
- Parcourir une arborescence OID

Pour « Getting », GetRequest-PDU est utilisée avec au plus 128 liaisons variables. Et pour le « walking » on utilise GetNextRequest-PDU pour SNMPv1 et GetBulkRequest avec le champ max-répétitions avec au plus 128 est utilisé avec SNMPv2 et SNMPv3

B. Etude des KPI's

Les KPI ou autrement dit les indicateurs clés de performance ce sont des indicateurs qui identifie le fonctionnement des processus systèmes et des applications pour superviser une machine il est essentiel de les identifier afin de poursuivre le fonctionnement de nos composantes à GOMOBILE

1. Système :

- **CPU nice time** : Nice est un programme utilisé pour invoquer une utilité ou bien un script Shell avec une priorité particulière, pour contrôler le temps du CPU allouer au processus, alors le nice time est le temps. Temps passé par le processeur à exécuter les processus des utilisateurs qui ont été configurés
- **CPU User time** : le temps passé par le processeur à exécuter les processus des utilisateurs qui n'ont pas été configurés.
- **CPU idle time** : le temps passé par le processeur en exécutant aucune tâche, on peut monitorer l'utilisation du processeur en général ou bien de chaque core
- **CPU iowait time** : le temps passé par le processeur en attendant I/O pour finir.
- **CPU softirq time** : Durée pendant laquelle le processeur le processeur traite les interruptions logicielles
- **CPU system time** : Durée d'exécution du noyau et ses processus par le processeur
- **Interrupts per seconde** : Les interruptions du processeur par seconde
- **Processor load** : le calcul du load du processeur divisé par le nombre de core.
- **CPU system time** : le temps passé par le processeur en exécutant le kernel et ses processus.
- **CPU utilisation** : l'utilisation en cours du CPU
- Le nombre des processus en cours : résume le nombre de processus en cours d'utilisation par le processeur
- Nombre totale de processus : le nombre total de processus en marche
- Mémoire utilisé
- Mémoire disponible
- L'espace du swap
- L'application qui utilise la mémoire le plus
- Alerte en cas d'utilisation haute de la Ram
- Monitorer l'espace des fichiers systèmes /, /home, /boot
- Les inodes libres sur / et /boot

- Monitorer le flux entrant
- Le flux sortant des données
- Nombre d'utilisateur connecter

2. Base de données :

- Le statut de Mysql
- Les requêtes lentes de Mysql
- Nombre de requêtes perdues
- Nombre de byte envoyer et reçus par seconde
- Monitorer toutes les opérations de mysql comme Select, delete, update ...
- Nombre de modifications faites sur une base de données
- Nombre de requête exécuter par seconde
- Nombre de connections
- Nombre de users connecter
- CPU time
- Les lignes d'une table de données spécifiques (calls, Scénario, Scénario_users, Compagnes ...)
- Nombre de requêtes vides exécuter
- Last query cost
- Max_statement_time_exceeded nombre de requête qui dépasse la durée d'exécution
- Max_used_connexions le nombre maximum de connection en même temps
- Memory_used la mémoire utilisée

3. Java :

- Mémoire utilisé
- Nombre de processus utilisé
- Informations du garbage collector
- Heap
- Temps de réponse
- Throughput de la requête
- Les requêtes SQL avec la java
- Et le nombre des outils qui chauffe l'API
- Nombre de thread en marche : User threads, daemon threads
- Logs + erreurs

4. Serveur Web :

- La moyenne de la vitesse de download par seconde pour toutes les étapes du scénario
- Nombre des étapes échoués
- Le dernier message d'erreur
- Supervision du code référant au statut du Site
- Supervision des variables, headers, champ Post et les requêtes exécuter
- Vitesse du download par seconde du scénario
- Vitesse de download de chaque chemin monitorer dans le site web
- Obtenir le temps de réponse de chaque chemin du site et leurs statuts
- Le code de réponse de chaque étape

C. Prérequis Matériels

1. Prérequis pour la mémoire

Zabbix a besoin à la fois de la mémoire physique et de la mémoire disque qui dépend principalement du nombre d'hôtes et des paramètres à superviser, et si on a besoin de garder des historiques de longue durée ou bien ceux d'une petite durée, le démon Zabbix nécessite plusieurs connexions au serveur de base de données donc la mémoire allouée contrôle la vitesse de stockage ou bien d'extraction des données par le moteur de base de données utilisé.

En minimum on peut même faire une allocation de mémoire de taille 128 Mb pour la mémoire physique et pour la mémoire disque 256 Mb.

2. Prérequis CPU

Le CPU joue un rôle important dans la vitesse de traitement de données collecter entre la base de données le serveur Zabbix et ses démons ainsi que cela contrôle d'une façon directe le nombre des hôtes et paramètres à superviser.

Voilà un tableau qui montre plusieurs configurations possibles selon le nombre d'hôtes et les items à monitorer

Nom	Plateforme	CPU/Mémoire	Base de données	Nombre D'hôtes
<i>Small</i>	CentOS	Virtual Appliance	MySQL InnoDB	100
<i>Medium</i>	CentOS	2 CPU cores/2GB	MySQL InnoDB	500

<i>Large</i>	RedHat Enterprise Linux	4 CPU cores/8GB	RAID10 MySQL InnoDB or PostgreSQL	>1000
<i>Very large</i>	RedHat Enterprise Linux	8 CPU cores/16GB	Fast RAID10 MySQL InnoDB or PostgreSQL	>10000

Tableau 1 : Comparaison des prérequis matériel

La minimum configuration du CPU nécessaire pour le fonctionnement de Zabbix avec sa minimum résolution est une virtuel Appliance

3. Taille de la base de données

Pour calculer l'espace nécessaire pour la base de données Zabbix il faut calculer la mémoire et déterminer la durée des stockages des informations du serveur des exemples de calcule qu'on va faire sont :

Paramètre	Formule pour l'espace disque requis (en octets)
<i>Configuration de Zabbix</i>	Taille fixe. Normalement 10MB ou moins.
<i>Historique</i>	Jours*(items/taux de rafraichissement)*24*3600*octets bytes : nombre d'octets nécessaire pour conserver une seule valeur, dépend du moteur de base de données, normalement ~90 bytes.
<i>Tendances</i>	Jours*(items/3600)*24*3600*octets octets : normalement~90 bytes.
<i>Evénements</i>	Jours*événement*24*3600*bytes octets : normalement ~170 bytes.

Tableau 2 : Méthode de calcul d'espace

Donc l'espace calculé ne sera pas utilisé automatiquement après l'installation de Zabbix mais il augmentera jusqu'à un point qui dépend des paramètres Housekeeper, alors l'espace total du disque requis est calculé selon la formule suivante :

Configuration + Historique + Tendances + Evénements

Partie 2 :

Conception & Mise

en place

I. Conception de la solution

Pour mettre en clair la conception utilisée on a essayé de faire une émulation du système de GOMOBILE afin d'éviter toutes les ambiguïtés et effectuer des tests avant de mettre en place la solution dans l'architecture réel, pour résumé notre conception de la solution on a mis le schéma suivant :

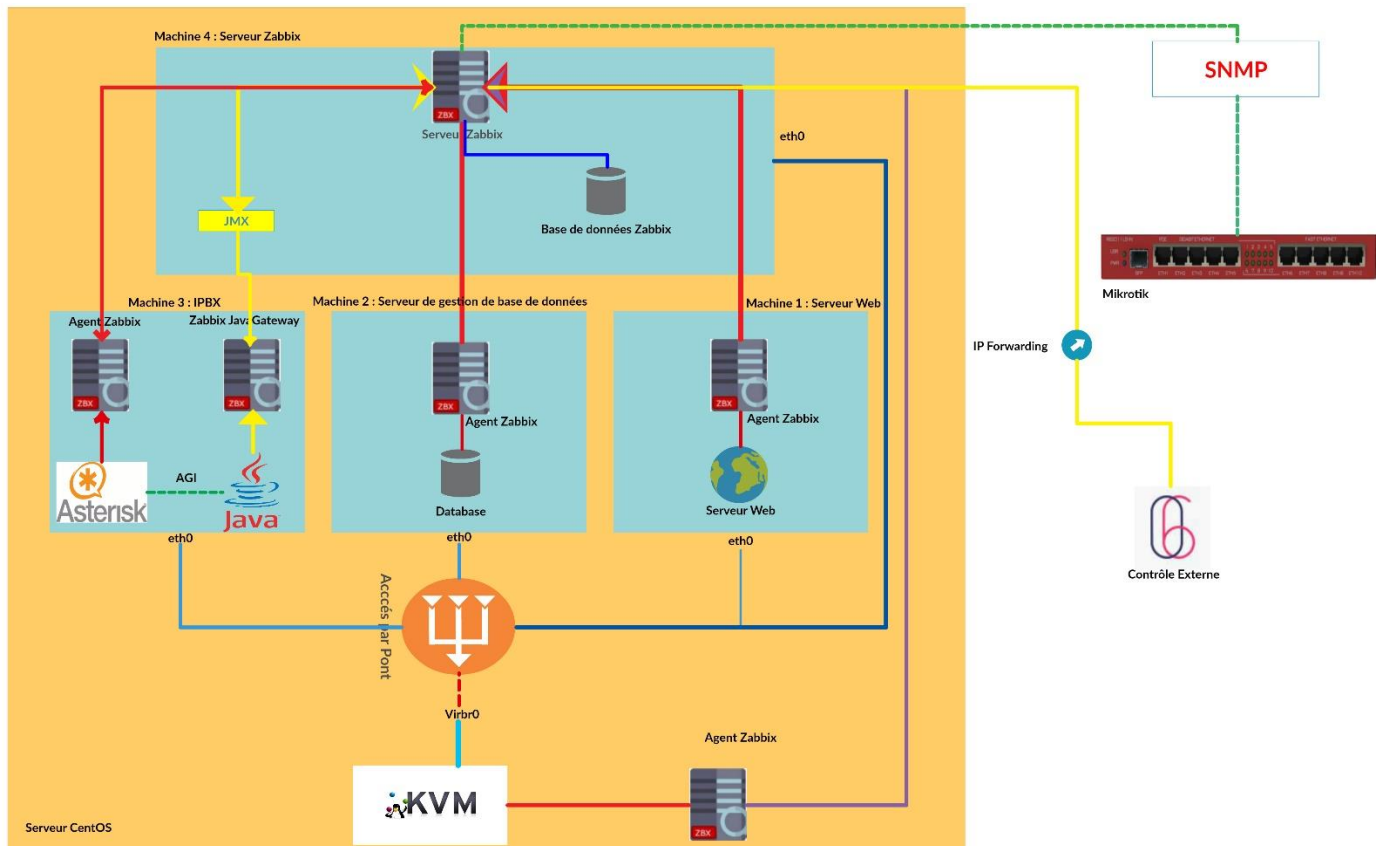


Figure 9 : Conception de la solution

La figure ci-dessus représente un schéma de la solution réalisé pour gérer l'architecture de GOMOBILE en détaillant la démarche de supervision pour chacune des composantes de l'architecture GOMOBILE :

1. Serveur CentOS

C'est le serveur principal contenant des machines virtuelles déployer avec l'outil KVM Linux qui transforme le noyau Linux en Hyperviseur de type 1 (HVM et PV avec virtio). Il traduit les instructions des vCPU via les instructions VT AMD et intel. KVM est aussi un émulateur de matériel qui utilise qemu et les pilotes virtio. Chaque VM est considéré comme un processus et chaque vCPU est considéré comme un thread de processeur, où il existe 3 types de virtualisation :

- Virtualisation totale (Full virtualization) : qemu
- Virtualisation Hardware-Assisted : qemu+KVM
- Paravirtualisation : qemu+KVM+virtio, Xen

Cette machine où on a installé l'agent Zabbix pour superviser les machines virtuelles et collecter les informations sur leur états et l'espace occupé par chaque machine ainsi que les domaines actives, le trafic réseau KVM et KVM pools en utilisant des scripts injectés dans la machine chacun au rôle précédemment indiqué ainsi qu'une plateforme en xml développé spécifiquement pour interagir avec les scripts KVM. Cette machine est le cœur de toute les machines car c'est elle qui fournit les ressources matérielles pour que les autres machines fonctionnent indépendamment l'une à l'autre.

2. Machine virtuelle du Serveur Zabbix

Le serveur Zabbix nécessite une base de données interne pour sauvegarder les données collecter et les tests effectués pour une date limitée établie dans l'interface web, ici le serveur web sert à injecter des plateformes personnalisées pour effectuer des tests personnalisés sur des logiciels non supportés sur les plateformes disponibles. Notre serveur réalise spécifiquement les checks actifs plus que les tests passifs. Cette machine contrôle la machine virtuelle principale, l'IPBX et JAVA, le serveur de gestion de base de données, le serveur web et la plateforme web.

3. Machine virtuelle d'IPBX et JAVA

Cette machine virtuelle contient Asterisk comme serveur de téléphonie sur IP et qui utilise le protocole SIP pour les communications le Trunking avec les Trunk de l'opérateur Orange est nécessaire pour effectuer des appels à l'externe du réseau local. GOMOBILE possède 2 trunks Cisco pour augmenter la capacité d'émission des appels et la vitesse de traitement des compagnes, Asterisk est trunker avec les deux pour rendre le traitement dynamique c'est-à-dire choisir le cube dont il va nous servir dans de chaque compagne pour diminuer la surcharge sur un Cisco qui est déjà occupé par une compagne. Comme cité avant GOMOBILE utilise le protocole SIP dans la communication et le fichier extensions.conf pour gérer les communications par exemple y'en a des clients qui demandent que les appels effectuer soit enregistrer dans la base de données, Asterisk possède des fonctions qui facilite l'usage comme étant :

- Dial() pour appeler,
- ChanSpy() & Monitor() pour écouter les appels en cours,
- Record() pour enregistrer les appels
- PlayBack() pour lire un audio enregistrer dans la base de données.

La communication s'établit avec la plateforme web selon l'AGI d'Asterisk qui donne la possibilité d'avoir des informations depuis la base de données de la plateforme via un programme JAVA qui s'exécute au cours de la campagne sa fonction est la liaison entre les informations dans la base de données et Asterisk pour que ce dernier puisse avoir les listes des numéros les scénarios préparer ainsi qu'elle permet de récupérer les informations enregistrer durant l'appel comme les interactions des appelants pour avoir leur avis et les enregistrer dans la base de données pour les afficher dans la plateforme et pouvoir générer des rapports de suivi des campagnes.

// On passe maintenant à la supervision qui est l'étape qui nous intéresse le plus il est très important de superviser l'IPBX pour avoir les informations concernant l'état d'Asterisk et des Trunks ainsi le nombre d'appels exécuter dans un intervalle de temps, aussi on cible avoir l'état

4. Machine virtuelle serveur de gestion de base de données

Cette machine contient un système de gestion de base de données MariaDB qui permet le stockage de toutes les informations de la plateforme et d'Asterisk, elle est organisée selon des tables chacune à pour rôle spécifique y'en a celle dédié aux scénarios, scénariosUser, Calls, Campagne le rôle de chacune se résume dans ce qui suit :

- Scénarios : contient la liste des audios à diffuser et le type des audios (interactifs, dynamique, statique)
- ScénariosUser : contient les listes des numéros à appeler durant la campagne, le montant, le nom et prénom

- Calls : c'est une table dynamique qui se remplit lors de la compagne elle contient les listes des numéros en cours de diffusions des appels sa capacité de remplissage est de 150 lignes, elle fait passer 150 appels à la fois
- Compagne : contient les informations sur la compagne telle que l'identifiant, le nom de la compagne, la date, durée de compagne.

Donc cette machine a un rôle organisationnel et une mémoire de toutes les opérations exécuter sur la plateforme et Asterisk ce qui la rend la machine la plus importante dans cette architecture car c'est elle qui nous garantie la vitesse de traitement des données et l'enregistrement de nos données avec toute sécurité et fiabilité.