



ECOLE POLYTECHNIQUE DE L'UNIVERSITÉ FRANÇOIS RABELAIS DE TOURS
Département Informatique
64 avenue Jean Portalis
37200 Tours, France
Tél. +33 (0)2 47 36 14 14
polytech.univ-tours.fr

Projet Recherche & Développement 2016-2017

Tuteur académique
Barthélemy SERRES

Étudiant
Pierre DUCHENE (DI5)

2 avril 2018



Liste des intervenants

Nom	Email	Qualité
Pierre DUCHENE	pierre.duchene@etu.univ-tours.fr	Étudiant DI5
Barthélemy SERRES	barthelemy.serres@univ-tours.fr	Tuteur académique, Département Informatique



Avertissement

Ce document a été rédigé par Pierre DUCHENE susnommé l'auteur.

L'Ecole Polytechnique de l'Université François Rabelais de Tours est représentée par Barthélemy Serres susnommé le tuteur académique.

Par l'utilisation de ce modèle de document, l'ensemble des intervenants du projet acceptent les conditions définies ci-après.

L'auteur reconnaît assumer l'entière responsabilité du contenu du document ainsi que toutes suites judiciaires qui pourraient en découler du fait du non respect des lois ou des droits d'auteur.

L'auteur atteste que les propos du document sont sincères et assument l'entière responsabilité de la véracité des propos.

L'auteur atteste ne pas s'approprier le travail d'autrui et que le document ne contient aucun plagiat.

L'auteur atteste que le document ne contient aucun propos diffamatoire ou condamnable devant la loi.

L'auteur reconnaît qu'il ne peut diffuser ce document en partie ou en intégralité sous quelque forme que ce soit sans l'accord préalable du tuteur académique et de l'entreprise.

L'auteur autorise l'école polytechnique de l'université François Rabelais de Tours à diffuser tout ou partie de ce document, sous quelque forme que ce soit, y compris après transformation en citant la source. Cette diffusion devra se faire gracieusement et être accompagnée du présent avertissement.



Pour citer ce document

Pierre DUCHENE, , Projet Recherche & Développement, Ecole Polytechnique de l'Université François Rabelais de Tours, Tours, France, 2016-2017.

```
@mastersthesis{
  author={DUCHENE, Pierre},
  title={},
  type={Projet Recherche \& Développement},
  school={Ecole Polytechnique de l'Université François Rabelais de Tours},
  address={Tours, France},
  year={2016-2017}
}
```

Table des matières

Liste des intervenants	a
Avertissement	b
Pour citer ce document	c
Table des matières	i
Table des figures	iv
Introduction	1
1 Contexte de la réalisation.....	1
1.1 Objectifs	1
1.2 Hypothèses.....	1
1.3 bases méthodologiques.....	2
I Etat de l’art	3
2 Nagios	3
2.1 Fonctionnalités	3
2.2 Principe de fonctionnement	4
2.3 Fun fact	5
3 SNMP	5
3.1 Agents SNMP	6
3.1.1 SpeedFan.....	6
3.1.2 SNMPD	6
4 WMI	6

5	NSClient++	7
II	Cahier de spécification	8
6	Description générale du projet.....	8
6.1	Environnement du projet	8
6.2	Caractéristiques des utilisateurs	8
6.3	Fonctionnalités du système.....	9
6.4	Structure générale du système	9
6.5	Maintenance et évolution du système	9
7	Description des interfaces externes du logiciel	10
7.1	Interfaces matériel/logiciel	10
7.2	Interface Homme/Machine	11
7.3	Interface Logiciel/Logiciel	11
8	Spécifications fonctionnelles.....	11
8.1	Authentification.....	11
8.2	Ajouter/retirer/modifier des systèmes à l'interface de gestion.....	11
8.3	Wake on Lan	13
8.4	Exécution de scripts à distance	13
8.5	Monitoring des systèmes	14
8.5.1	Récupération et stockage des données.....	14
8.5.2	Présentation des données	15
9	Spécifications non fonctionnelles.....	15
9.1	Contraintes de développement et conception.....	15
9.1.1	Contraintes matérielles	15
9.1.2	Contraintes logicielles	16
9.2	Capacités.....	16
9.3	Résistance du système	16
III	Technologies	17
10	PHP.....	17
11	Framework PHP	18
11.1	Laravel.....	18
11.2	CodeIgniter	18
11.3	Symfony	18
12	Fonctionnement de Symfony.....	19
13	REST	20
IV	Environnement de test	21
14	Matériel.....	21

V	Analyse	22
15	Monitoring.....	23
15.1	Metrics	23
16	Wake On Lan	23
17	Exécution de scripts.....	24
18	Login	24
VI	gestion de projet	25
19	Définition des objectifs.....	25
20	Comptes rendu hebdomadaires	25
VII	Réalisation	26
21	Serveur web	26
22	Interface utilisateur	26
23	Tests	26
23.1	Scénarios API	26
23.2	Tests des pages web	27
23.2.1	Séquence ON/OFF.....	27
VIII	Conclusion	28
	Annexes	29
A	Trello	30
B	Technologies	35
C	Structure du projet	37
D	Stockage des scripts	45
E	Computers API documentation	47
F	ComputerUsers API documentation	52
G	MonitoringLoockup API documentation	55
H	Manuel utilisateur	57
I	Rapport précédent	65

Table des figures

Table des figures

1	Fonctionnement de Nagios core, par Diglinks (wikipedia).....	4
2	Abstraction par les plugins	4
3	Code PHP récupérant et affichant l'arbre OID de l'ordinateur ayant l'IP 169.254.254.70 en dessous du nœud 1.3.6.1.2.1.25	6
4	Symfony architecture	10
5	Architecture des contrôleurs REST	11
6	Base de donnée	12
7	structure machine/logiciel	12
8	Fenêtre de gestion principale vue admin.....	13
9	Fenêtre de gestion principale vue user	14
10	Arborescence des scripts	15
11	monitoring panel	16
12	Répartition des langages de programmation server side en octobre 2017 <i>selon w3techs.com</i>	17
13	exemple de code PHP imbriqué dans du HTML	18
14	Exemple de controller symfony	19
15	Environnement de test	21
16	Implémentation du module monitoring	23
17	Implémentation de metrics	24
18	Implémentation de wake on lan provider	24

Introduction

Ce projet de recherche et développement (PRD) anciennement projet de fin d'étude s'inscrit dans la formation dispensée à l'école d'ingénieur polytech Tours. Ce projet consiste à mener à bien un projet conséquent, avec pour objectif de donner une vraie expérience en terme de développement, gestion de projet et recherche à l'étudiant.

Ce PRD ayant déjà été traité l'année dernière, le but de mon travail est de reprendre ce qui a été fait, le rapport produit par mon prédécesseur est en annexe [I](#).

Le PRD est orienté vers une conception logicielle où la MOA est représenté par Mr Barthélémy Serres membre du laboratoire d'informatique de l'Université de Tours.

1 Contexte de la réalisation

Le projet s'inscrit dans un partenariat entre le centre d'étude de la renaissance de Tours et l'école d'ingénieur Polytech Tours. Ce partenariat vise à développer des outils numériques pour l'étude et la présentation des œuvres et des les mettre à disposition des chercheurs et du public lors de l'exposition avec le musée des beaux arts de Tours. C'est pour cette exposition que ce logiciel va être développé. Site du projet : [sculpture 3d](#)

1.1 Objectifs

L'objectif de ce projet est donc de mettre à disposition des équipes techniques de l'exposition un logiciel permettant d'allumer et d'éteindre à distance les outils informatiques utilisés pour cette exposition. Le logiciel devra permettre aussi de générer et présenter des rapports d'état des outils informatique (surveillance de la température, utilisation CPU, etc) et d'effectuer des tâches de maintenance (extinction/allumage, exécution de scripts, shell à distance, etc) puis dans un second temps de notifier les administrateurs en cas de problème.

1.2 Hypothèses

Hypothèse : Windows, UNIX et autre fournissent tous les outils nécessaires pour réaliser les objectifs. Si cette hypothèse se révèle exacte, le seul développement d'un outil centralisé sera nécessaire. Dans le cas contraire, le développement en parallèle de clients multi-plateforme devra être envisagé, ou une castration des fonctionnalités dans les cas où c'est impossible pour les vidéo-projecteurs par exemple.

1.3 bases méthodologiques

Pour le développement je vais utiliser le langage PHP couplé au framework Symfony, ainsi que le moteur de template Twig pour la génération des pages. Pour l'organisation du projet, je vais suivre le guide des bonnes pratiques de développement proposées par symfony.com à [cette adresse](#). Pour la modélisation des différentes parties du logiciel, je vais utiliser le format UML. Pour la modélisation de la base de la base de donnée, utilisant une méthode ORM, je vais me contenter de décrire les différentes entités avec UML. Les sources du projet ainsi que du rapport au format latex seront versionnées en utilisant le logiciel git avec l'interface de gestion gitlab qui permet entre autre de faire de l'intégration continue. Le serveur gitlab sera installé sur une machine de l'école à laquelle je pourrai me connecter en utilisant le VPN fourni par l'école. Ce choix à été motivé par ma connaissance de ce système et de la forte intégration de git dans mon IDE.

Première partie

Etat de l'art

Mon prédécesseur a déjà fait un état de l'art satisfaisant des outils permettant d'utiliser la technologie Wake On Lan dans son rapport (Annexe I), ici l'état de l'art portera principalement sur le monitoring.

2 Nagios

Nagios est une application client/serveur libre et open source. Cette application permet de surveiller l'état de service, switchs, serveurs et d'application. Il envoie une notification en cas de problème et quand il est résolu. Ce programme a été créé pour fonctionner sur les systèmes de type UNIX, puis a été porté sur windows.

La marque nagios a décliné son offre en plusieurs produits :

- Nagios Core : le produit original de monitoring.
- Nagios Log Server : Ce produit est un agrégateur de log centralisé.
- Nagios Fusion : Cet outil est prévu pour les grosses infrastructures, et réduit la charge de la node centrale en cas d'agrandissement de l'infrastructure.
- Nagios XI : C'est le produit phare de la marque, il apporte un grand nombre de fonctionnalités à Nagios Core mais n'est plus un logiciel libre mais open source et devient payant [Online demo](#)
- Nagios Network Analyser : Ce produit permet le monitoring de réseaux.

Pour la suite, nous allons nous concentrer sur Nagios Core qui se rapproche le plus du sujet.

2.1 Fonctionnalités

Le logiciel Nagios Core implémente les fonctionnalités suivante mais sans s'y limiter. (Source Documentation + Wikipedia Page)

- Surveillance de services réseaux
- Monitoring de machine, charge CPU, mémoire, usage disque, etc
- Interface SNMP
- Supervision à distance (tunneling ssh par exemple)
- Gestion utilisateur
- Notification par mail/sms/browser/etc via différents plugins, entièrement configurable
- La création de plugin personnalisé est possible

pour plus d'informations, voir le [site internet](#) de nagios

2.2 Principe de fonctionnement

Les figures 1 et 2 résument le mode de fonctionnement du logiciel, le cœur de nagios s'appuie sur des plugins qui récupèrent ou envoient des données de monitoring. Nagios core fait le lien entre les différents plugins.

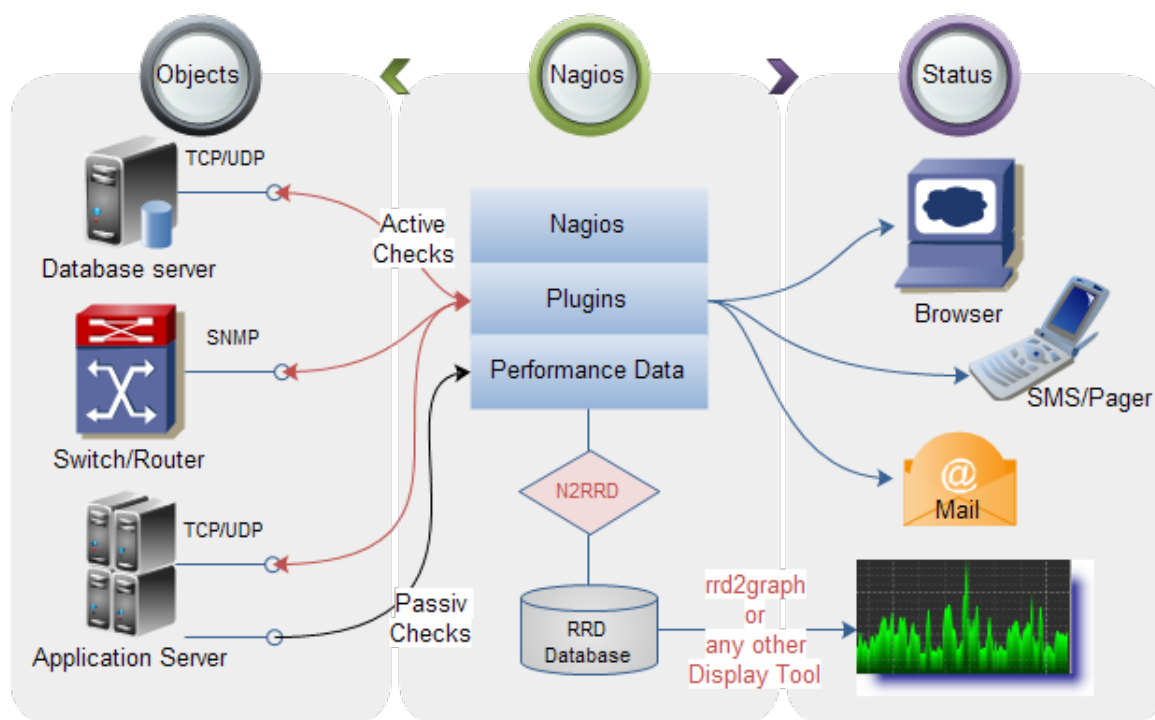


Figure 1 – Fonctionnement de Nagios core, par Diglinks (wikipedia)

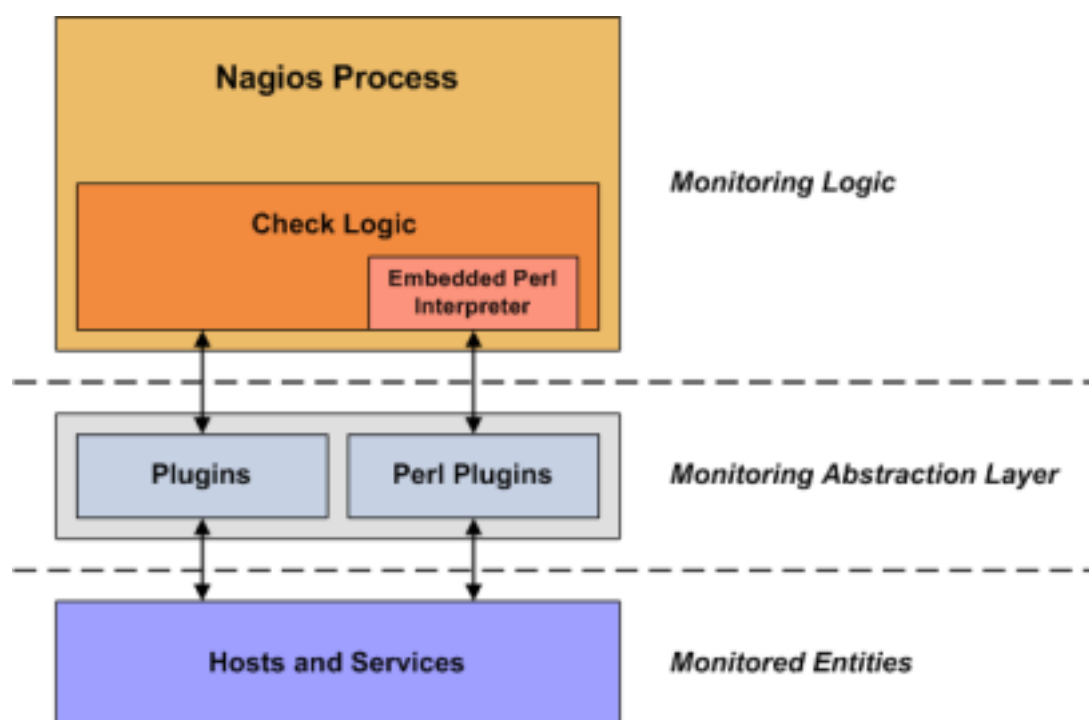


Figure 2 – Abstraction par les plugins

Nagios fournit un nombre de plugins conséquent qui permettent l'interaction avec un grand nombre de système, allant du monitoring de service réseau à des machines hôtes en passant par des envois de SMS d'alerte.

2.3 Fun fact

Le nom NAGIOS est en fait un acronyme récursif "Nagios Ain't Gonna Insist On Sainthood", "Sainthood" fait référence au nom original du logiciel : NetSaint

3 SNMP

SNMP (Simple Network Management Protocol) est un protocole de communication situé dans la couche applicative du modèle OSI, il est transporté via des datagrammes UDP. Il fait partie des "Internet Standards" définie par l'IETF Ce protocole permet de lire et modifier dans certains cas des variables situés dans un MIB (Management Information Base) qui décrit le système.

Il existe 3 versions de ce protocole 1,2c et 3. La version 3 est la plus récente ajoute une couche de sécurité via un chiffrement des données transportées et une amélioration de la connexion.

Le protocole snmp est implémenté dans la plupart des systèmes informatiques en réseau et dans la quasi totalité des équipements réseaux.

L'avantage de ce protocole est qu'il définit un standard décrit dans la [RFC 2790](#) qui est sensé être exposé par tous les systèmes "host" (entendre ici machine de travail comme windows ou unix).

Les implémentations de SNMP utilisent des OID (Object Identifiers) pour référencer les ressources accessibles via le protocole. Un OID représente un nœud dans l'arbre des OID qui sont définis par plusieurs organisations. Les arbres commençant par :

- 0 sont définis par l'[IUT-T](#)
- 1 sont définis par l'[ISO](#)
- 2 est un arbre joint des deux organisations précédentes.

L'arbre qui va nous intéresser est celui de l'ISO. Chaque nœud de l'arbre est assigné par une organisation. Prenons un exemple que je vais beaucoup utiliser: L'arbre suivant : **1.3.6.1.4.1.311**

- en dessous de 1 est assigné par ISO
- en dessous de 1.3.6 est assigné par le département de la défense américaine
- en dessous de 1.3.6.1.4.1 est assigné par l'[IANA](#)
- en dessous de 1.3.6.1.4.1.311 est assigné par Microsoft

Donc tous les nœuds en dessous de 1.3.6.1.4.1.311. sont assignés par Microsoft.

Regardons maintenant la signification de ces nombres :

- 1 => ISO
- 1.3 => Organisations
- 1.3.6 => Département de la défense américaine
- 1.3.6.1 => Internet
- 1.3.6.1.4 => Projets privés
- 1.3.6.1.4.1 => IANA
- 1.3.6.1.4.1.311 => Microsoft

Pour s'y retrouver dans tous ces arbres, je me suis référé aux différentes RFC sur le sujet et au site internet suivant . La RFC définissant tout cela est [ici](#) mais elle est assez indigeste.

```

1  <?php
2
3  /**
4   * Created by IntelliJ IDEA.
5   * User: Pierre
6   * Date: 19/10/2017
7   * Time: 09:48
8   */
9
10 $session = new SNMP( version: SNMP::VERSION_2c, hostname: '169.254.254.70', community: 'public');
11
12 /** @var string[] $a */
13 $a=$session->walk( 'objid: 1.1.3.6.1.2.1.25');
14 foreach ($a as $key=>$val) {
15     echo $key . ' => ' . $val . '<br>';
16 }

```

Figure 3 – Code PHP récupérant et affichant l'arbre OID de l'ordinateur ayant l'IP 169.254.254.70 en dessous du nœud 1.3.6.1.2.1.25

3.1 Agents SNMP

Les agents SNMP sont des petits logiciels qui remplissent les MIB avec des données. Ce qui permet donc à des logiciels distants d'y accéder.

3.1.1 SpeedFan

SpeedFan est un logiciel de monitoring qui permet de récupérer des valeurs comme la température et la vitesse du système de refroidissement. Grâce à un plug-in, il peut se comporter comme un agent SNMP, ce qui va nous permettre de récupérer ces valeurs et les traiter dans l'application.

3.1.2 SNMPD

Snmpd est le nom de l'agent pour la plupart des distributions linux, pour l'installer il suffit de lancer la commande

```
# apt-get install snmpd
```

Pour configurer l'agent, il suffit d'éditer le fichier "/etc/init.d/snmpd.conf".

4 WMI

Windows Management Instrumentation (WMI) est l'implémentation windows d'un standard (WEBM) développé au départ par un groupe d'industriels du secteur. Dans le but d'avoir un système de management unifié. Il est basé sur un CIM (Common Information Model) qui définit comment représenter un environnement en un ensemble d'objets que l'on peut lire et modifier.

La solution WMI n'est pas intéressante pour ce projet, ce standard est plus lourd que le standard SNMP, plus dur à utiliser et moins implémenté sur des outils léger tels que les routeurs et vidéoprojecteurs.

5 NSClient++

NSClient++ (nscp) est un client de monitoring. Ce logiciel a pour but de collecter le plus d'information sur la machine et de les transmettre, à un serveur ou tout simplement via un client web.

Ce client peut utiliser plusieurs protocoles, leur site web en fournit une liste comparative :

Protocole	Mode	Secure	Drawbacks
NRPE	Actif	Peut être sécurisé	Charge utile par défaut de 1024
NSCA	Passif	Sécurisé	Charge utile par défaut de 512
NRDP	Passif	Peut être sécurisé	
REST	Multiple	Sécurisé	
check_nt	Actif	Pas de chiffrement	Très limité
check_mk	Actif	Pas sécurisé	Lent et peu efficace
Graphite	Graphiques	Pas sécurisé	Uniquement pour des graphiques
SMTP	Passif	Pas sécurisé (avec NSClient++)	C'est un joujou
Syslog	Passif	Pas sécurisé	Uniquement pour envoyer des logs

Le mode Actif veut dire que le client envoie les informations au serveur, tandis que le mode passif implique que le serveur va demander au client les informations. La méthode REST (13) est la méthode recommandée par l'éditeur (natif).

NSClient++ permet aussi l'exécution de scripts. Ce qui fait de cet agent une bonne méthode pour l'interface de monitoring et d'exécution des scripts au cas où SNMP et l'ouverture de shell à distance ne suffisent pas.

Deuxième partie

Cahier de spécification

6 Description générale du projet

6.1 Environnement du projet

Le projet sera situé dans un musée , donc dans un réseau local avec un parc de machine aux systèmes d'exploitation hétérogène. Selon les évolutions possible, le système peut être amené à interagir avec les différents projets présents sur ces machines.

6.2 Caractéristiques des utilisateurs

L'ensemble des utilisateurs se résume à trois catégories :

- Personnel du musée : pas ou peu de connaissances en informatique, accèdera aux pages sans connections uniquement. Il aura accès à un nombre limité de fonctionnalités.
- Utilisateur : possède des connaissances en informatique, aura accès à l'ensemble des fonctionnalités du système sauf la gestion utilisateurs.
- Super-Utilisateur : possède des connaissances en informatique, aura accès à l'ensemble des fonctionnalités incluant les fonctions de gestion d'utilisateurs.

Au vu des différents types d'utilisateurs, un système d'authentification devra être mis en place.

6.3 Fonctionnalités du système

Les fonctionnalités principales du système sont les suivantes :

- Authentification : Les utilisateurs non autorisés ne doivent pas avoir accès aux éléments pouvant perturber le fonctionnement des systèmes gérés par l'application. En fonctionnement normal, ces personnels devront juste pouvoir surveiller les états, allumer et éteindre les appareils.
- Ajouter/Retirer/Modifier des systèmes à l'interface de gestion.
- Wake on lan : Pouvoir allumer ou éteindre un système depuis l'interface de gestion.
- Monitoring des systèmes : Pouvoir surveiller l'état des systèmes.
- Exécution à distance de scripts / ouverture de shell à distance.

Les fonctionnalités à implémenter si le temps le permet sont les suivantes :

- La possibilité d'accéder au logiciel depuis internet.
- Agréger les log des applications qui sont lancées sur les différents systèmes.

Ces différentes fonctionnalités sont détaillées plus loin dans le cahier de spécification fonctionnel

6.4 Structure générale du système

Le projet sera une application web utilisant le framework php symfony 4 avec le moteur de template Twig.

Le modèle présenté dans la figure 4 sera utilisé pour la génération des pages. Le contenu des pages par contre sera généré dynamiquement via des requêtes REST générées par des scripts javascript (technologie ajax).

La structure utilisée pour gérer les requêtes REST est décrite dans la figure 5. En fonction de la requête, le script de routing envoie la requête au bon contrôleur qui se charge de la traiter. Il y a trois controllers principaux :

- Monitoring Controller : Le controller de monitoring, il va aller interroger une interface avec la bonne implémentation en fonction du type de monitoring et de système d'exploitation de la machine à interroger.
- Action Controller : le contrôleur de contrôle du parc, va s'occuper du Wake On Lan et des scripts.
- Login Controller : Ce controller va utiliser une implémentation de OAuth2.

La structure de la base de donnée sera la suivante (excluant les éventuelles implémentations du Monitoring handler, script handler et OAuth2) Figure 6, l'idée de cette architecture est que pour l'ajout de fonctionnalité, ces objets ne sont pas modifiés, mais de nouvelles tables sont créées.

6.5 Maintenance et évolution du système

En parallèle du développement, une documentation technique et utilisateur sera construite. Cette documentation permettra de simplifier une maintenance du logiciel. Pour cela j'utiliserai le standard PHPDoc qui permet de générer une documentation via des mots clefs dans les commentaires des méthodes et des classes. Il me faudra aussi produire des documents décrivant

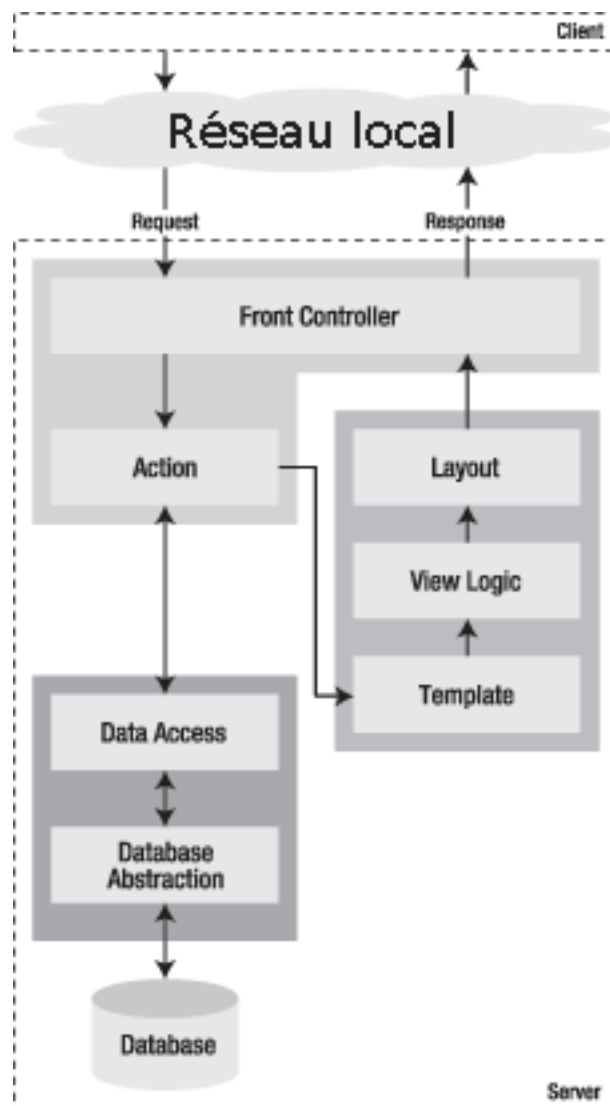


Figure 4 – Symfony architecture

comment déployer et configurer la solution. Durant le développement je garderai en tête les possibilités d'évolution du système et adapterai la structure de la solution pour rendre ces améliorations faciles à développer.

7 Description des interfaces externes du logiciel

7.1 Interfaces matériel/logiciel

Le projet a pour but de surveiller et contrôler les machines d'un réseau local. Le logiciel va donc devoir communiquer avec le réseau via l'interface Ethernet de la machine hôte.

On pourra imaginer utiliser un tunnel ssh pour se connecter à l'application depuis l'extérieur du réseau (Figure 7).

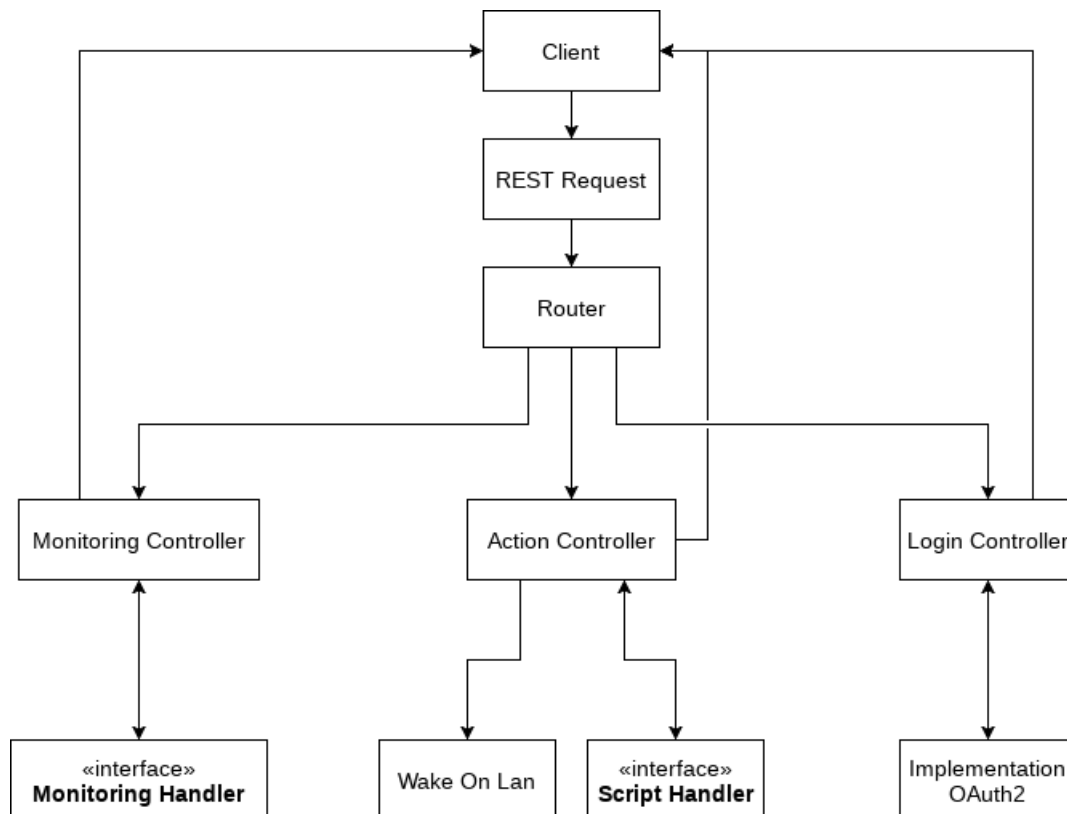


Figure 5 – Architecture des contrôleurs REST

7.2 Interface Homme/Machine

L'interface Homme/Machine se fera via un navigateur internet (Chrome, Opéra, Firefox, etc)

7.3 Interface Logiciel/Logiciel

Le logiciel aura besoin de s'interfacer avec plusieurs autres logiciels :

- Un SGBD, pour assurer la persistance des données
- Des systèmes d'exploitations via le réseau.

8 Spécifications fonctionnelles

8.1 Authentification

La fonctionnalité authentification permet aux utilisateurs avancés du système de se connecter pour accéder aux fonctionnalités qui leurs sont réservées. Pour cela, l'utilisateur se verra recevoir un mot de passe et un identifiant. Si le couple mot de passe / identifiant est correct, l'utilisateur aura alors accès aux fonctionnalités restreintes.

8.2 Ajouter/retirer/modifier des systèmes à l'interface de gestion

Le logiciel a besoin d'informations sur les systèmes à gérer. Pour cela, cette fonctionnalité va avoir besoin de stocker ces données dans une base de donnée, de les extraire et de les modifier.

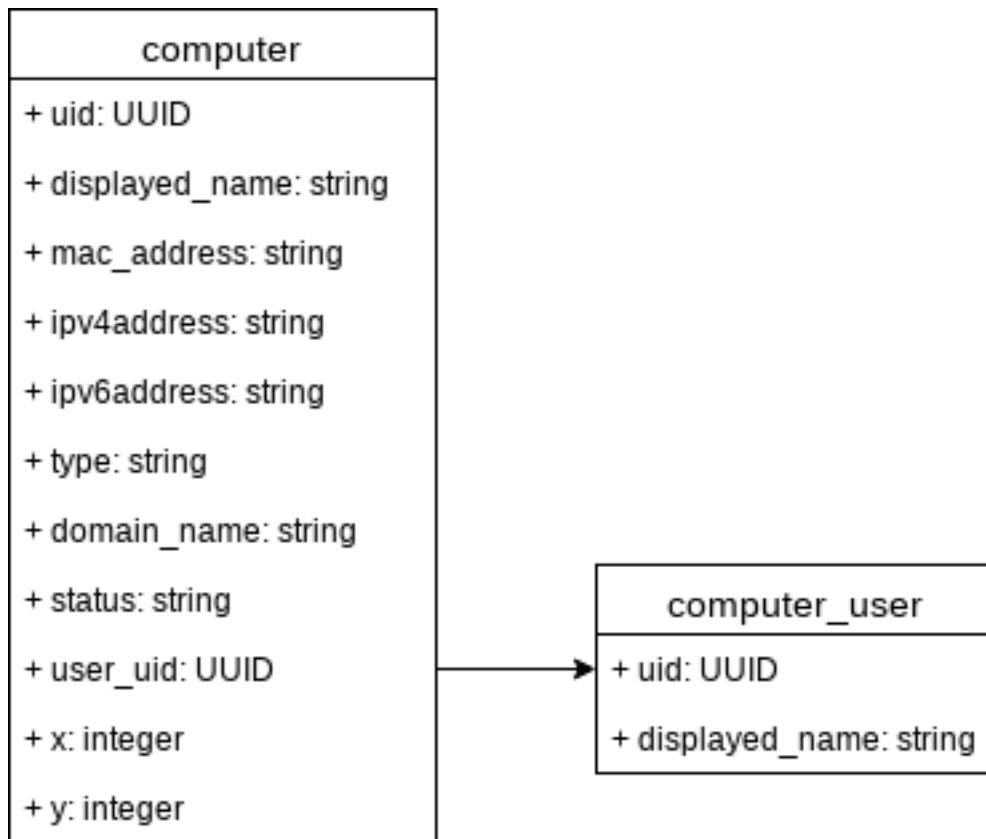


Figure 6 – Base de donnée

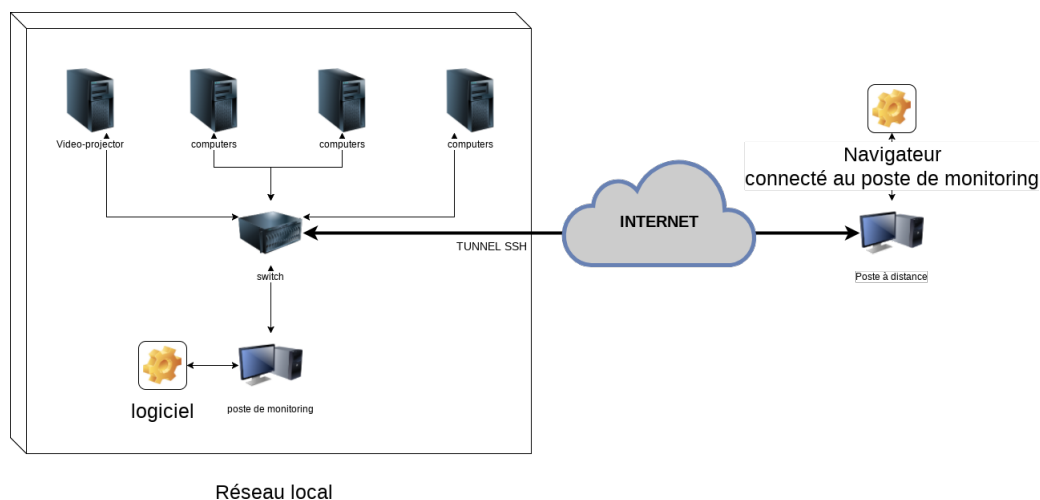


Figure 7 – structure machine/logiciel

Un système sera défini par plusieurs champs :

- Un identifiant unique (donné par le système)
- Comment le joindre sur le réseau (ndd ou ip)
- Système d'exploitation
- Son adresse MAC
- L'ID de son utilisateur qui contient ses informations de connections.
- D'autres champs pourront être ajouté en fonction des besoins.

L'interface de gestion (figure 8) devra aussi les présenter de manière claire, et exposer l'interface de modification uniquement aux utilisateurs connectés. Comme on peut le voir dans le mockup

ci dessus, la fenêtre principale sera simplement composée d'un simple tableau à 4 colonnes :

1. Le nom de l'ordinateur dans le logiciel
2. Le statut qui peut être entre autre Ok, Off, Rebooting ou Pending
3. Les actions que l'on peut effectuer, comme accéder à la vue détaillée de la machine, ou éditer la machine.
4. Les scripts à exécuter.

Name	Status	Actions	Execute
Computer 1	ok	Inspect Edit	Shutdown ▼
Computer 2	ok	Inspect Edit	Shutdown ▼
Computer 3	Overheat	Inspect Edit	Shutdown ▼
Computer 4	Off	Inspect Edit	Wake up ▼
Computer 5	ok	Inspect Edit	Shutdown ▼
Computer 6	Rebooting	Inspect Edit	▼
Computer 7	ok	Inspect Edit	Shutdown ▼
Computer 8	Pending	Inspect Edit	▼

Figure 8 – Fenêtre de gestion principale vue admin

Le panneau "user" (figure 9) est une version simplifiée du panel admin, le dropdown utilisé pour afficher la liste des scripts est remplacée par deux boutons : turn Off/On et Restart les boutons d'action sont grisés et non utilisables car l'utilisateur n'a pas les droits nécessaires. L'objectif étant de pouvoir se faire rapidement une idée des systèmes à surveiller et des actions à exécuter dans ces cas là, le code couleur rouge/vert/gris est simple à saisir et permet au technicien de rapidement décider quelle action entreprendre.

8.3 Wake on Lan

Cette fonctionnalité devra permettre de démarrer un des système inscrit dans la base de donnée à distance. Elle sera aisément visible et facile a utiliser car du personnel peu compétent en informatique sera amené à l'utiliser. Cette fonctionnalité devra être accessible pour tous les utilisateurs du logiciel.

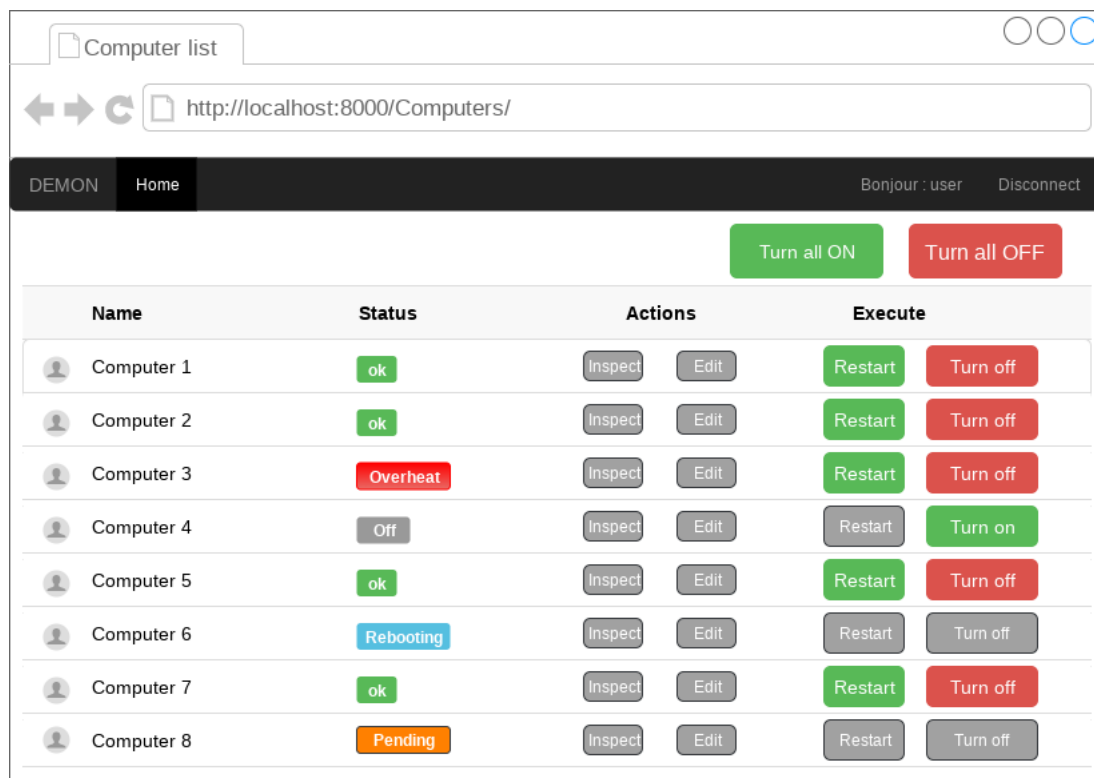


Figure 9 – Fenêtre de gestion principale vue user

8.4 Exécution de scripts à distance

Le logiciel doit être capable d'exécuter des scripts à distance sur les machines surveillées, les scripts pourront être stockés dans le système de fichier pour être chargés dans le logiciel automatiquement ou manuellement.

Les actions basiques à pouvoir exécuter sur les machines distantes (type Host) seront, éteindre et redémarrer. Pour le chargement automatique des scripts, on pourra imaginer une arborescence comme montré dans la figure 10

Les scripts "default_*" contiendront les scripts correspondant aux actions par défaut liés aux systèmes d'exploitation concernés. Ces actions peuvent être override par des scripts contenus dans un répertoire dont le nom est le même que l'ID de l'ordinateur en question.

8.5 Monitoring des systèmes

Cette fonctionnalité permet automatiquement de se connecter régulièrement puis de récupérer des informations sur les systèmes à gérer, de stocker ces données et de les présenter à l'utilisateur de manière à ce qu'il se fasse aisément une idée de la bonne ou mauvaise santé du système qu'il veut surveiller.

Cette fonctionnalité peut se découper en deux sous fonctionnalité récupération et présentation.

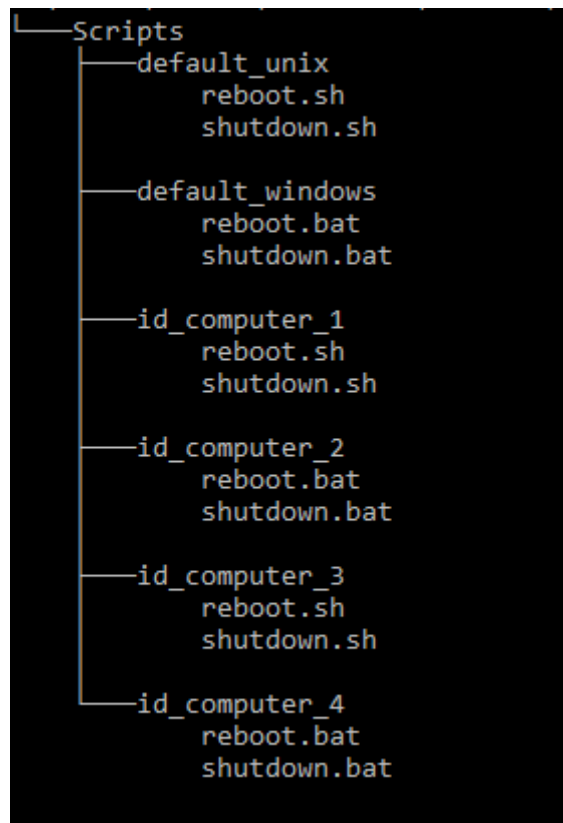


Figure 10 – Arborescence des scripts

8.5.1 Récupération et stockage des données

Dans cette première partie, on devra récupérer les données via différentes méthodes en fonction du système d'exploitation utilisées par ces systèmes. Une fois ces données récupérées, elles doivent être correctement stockées et historiées de manière à pouvoir être correctement affichées, puis supprimées en fonction de leur ancienneté pour ne pas saturer la RAM ou la base de données en fonction du mode de stockage choisi.

Ces données peuvent être de différents types, le système d'archivage doit donc pouvoir être suffisamment souple pour correspondre à tout type de donnée. On peut en imaginer plusieurs implémentations différentes.

8.5.2 Présentation des données

Cette deuxième partie consiste à la mise en forme des données de manière à ce que l'utilisateur du logiciel puisse les comprendre facilement via des graphiques par exemple (figure 11).

Le menu à gauche est composé d'un set d'outils permettant d'exécuter des scripts ou d'ajouter de nouvelles données à surveiller.

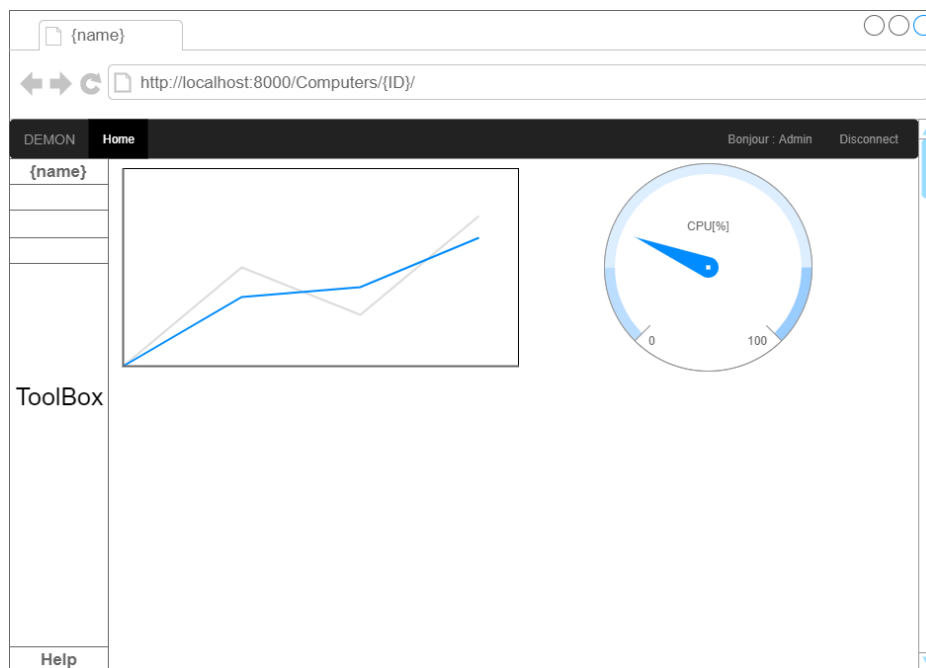


Figure 11 – monitoring panel

9 Spécifications non fonctionnelles

9.1 Contraintes de développement et conception

9.1.1 Contraintes matérielles

Le logiciel devra mobiliser le moins de ressources possible sur les machines qu'il devra surveiller, car elles sont déjà mobilisées par des programmes gourmands en ressources.

9.1.2 Contraintes logicielles

Le logiciel devra s'interfacer avec plusieurs OS différents et plusieurs types de systèmes allant de vidéo-projecteurs à des ordinateurs de bureau.

9.2 Capacités

La capacité maximale du logiciel sera déterminée par la capacité de la machine qui va héberger l'application (disque, CPU, RAM).

9.3 Résistance du système

Le système devra pouvoir être résilient. La résistance aux bugs devra être au cœur de la conception. Les fonctionnalités devront être indépendantes entre elles pour pouvoir fonctionner en mode diminué si certaines parties du logiciel ne sont pas en état de fonctionnement. En cas de grave problème de fonctionnement, le logiciel devra pouvoir se relancer automatiquement.

Troisième partie

Technologies

10 PHP

Le langage PHP est un langage de scripting crée pour le côté serveur applications. Il existe depuis 1994. Il est actuellement dans sa version 7.1. C'est cette version que je vais utiliser pour ce projet, elle sera normalement maintenue jusqu'à fin 2019. Ce langage est très populaire pour des technologies Web, il représente en Octobre 2017 près de 83 % des sites connus. Ce qui veut dire qu'il y a une très grosse communauté de développeurs et un nombre de ressources en ligne disponible très important.

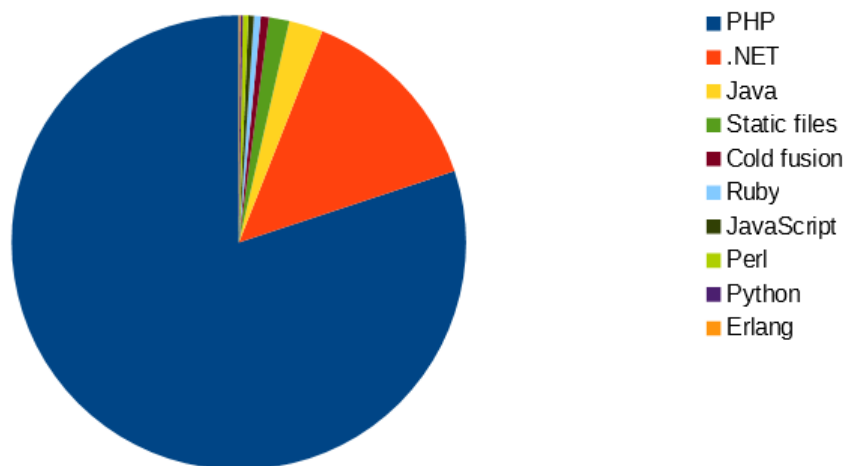


Figure 12 – Répartition des langages de programmation server side en octobre 2017 selon *w3techs.com*

PHP est d'abord conçu pour être imbriqué dans des pages HTML et être compilé à la volée par le serveur Web lors de l'envoi de la page au client.

Il est désormais utilisé comme langage objet dans des applications web complexes. Il a l'avantage d'être facile à déployer et à maintenir tout en répondant à des problématiques variées.

```
<!DOCTYPE html>
<html>
  <head>
    <title>PHP Test</title>
  </head>
  <body>
    <?php echo '<p>Hello World</p>'; ?>
  </body>
</html>
```

Figure 13 – exemple de code PHP imbriqué dans du HTML

11 Framework PHP

J’ai testé plusieurs frameworks différents qui pourraient être utiles pour ce projet

- CodeIgniter
- Symfony
- Laravel

11.1 Laravel

Laravel est un framework PHP basé sur le même cœur que Symfony. Après quelques tests et avoir set-up un environnement de développement. J’ai ensuite eu des difficultés à déployer la solution, la documentation a été compliquée à trouver et à consulter. C’est sûrement dû au fait que Laravel est assez récent et la communauté pas encore assez mûre.

11.2 CodeIgniter

Ce framework a l’avantage d’être extrêmement léger, mais je n’ai malheureusement pas réussi à le faire fonctionner. Je suis donc passé rapidement à autre chose.

11.3 Symfony

Symfony est un framework assez mûr, il existe depuis 2005. La communauté est très active ce qui veut dire que ressources et documentations sont disponibles en grande quantité et de qualité. Symfony est aussi un très bon compromis entre les deux frameworks cités précédemment Laravel étant trop gros pour le projet et codeIgniter très léger mais assez difficile à prendre en main.

C’est donc ce dernier framework que j’ai choisi pour ce projet, dans sa version 3. Il est facile à installer et à prendre en main. J’ai donc rapidement pu générer des pages webs et créer une petite base de donnée Postgres et interagir avec, en utilisant Doctrine, le gestionnaire d’ORM intégré à l’installation de base de Symfony.

12 Fonctionnement de Symfony

Tout d'abord qu'est ce qu'un framework ? Retournons à l'étymologie du mot, il est en premier lieu utilisé pour les structures en bois, comme pour une maison. En informatique, le framework est donc une sorte de structure permettant de construire une application en s'appuyant dessus. Le framework va fournir tout un ensemble d'outils, permettant de construire l'application, ainsi qu'une méthodologie pour y arriver. Symfony se définit comme étant un ensemble d'outils et une méthodologie de développement, ainsi qu'une philosophie et une communauté.

Qu'est ce que symfony fourni comme outils (non exhaustif):

- Un outil d'injection de dépendance. Cet outil permet d'accéder à des objets initialisés ailleurs. Par exemple, d'accéder à Doctrine depuis n'importe quel controller alors que l'objet est initialisé au démarrage. Symfony définit le module dependency injection de cette manière (traduit de l'anglais) : Le composant injection de dépendance permet de centraliser et de standardiser la manière dont les objets sont construits dans votre application.
- Un gestionnaire ORM (object relational mapping) qui va permettre de définir des objets comme étant des « entités » qui vont pouvoir être rendu persistants et ainsi sauvegardés en base de donnée, le moteur ORM intégré dans symfony est Doctrine.
- Symfony gère le routage et le lancement du serveur web via des outils en ligne de commande.
- Le moteur de template Twig qui permet de générer facilement des pages en fonction d'arguments passés en paramètre.
- Des outils développés par d'autres acteurs sont également disponibles et facilement intégrables.
- Liste des **composants** proposés par Symfony

Le principe d'un framework et donc de symfony est de permettre l'organisation du code. En développant une application avec Symfony, le développeur doit permettre de faire le liens entre un requête de l'utilisateur et la ressource associée (Json, page Web, etc). Le code permettant d'effectuer cela est situé dans des classes PHP appelées controller, et les méthodes qui les composent sont appelées actions mais on y réfère souvent par le terme controller c'est ce que je ferai ici. La configuration de routage va faire les liens entre les controllers et les requêtes. Les pages sont générées par des templates. Prenons un exemple, quand on accède à l'adresse suivante : `http://example.com/homepage/page` le système de routage de symfony va lancer l'exécution de ce controller, le paramètre `$request` contient la requête (évidement) et le paramètre `$parameter` va contenir 'page'.

```
1 <?php
2
3 namespace AppBundle\Controller;
4
5 use ...
6
7
8
9 class DefaultController extends Controller
10 {
11     /**
12      * @Route("/homepage/{parameter}")
13      */
14     public function exampleAction(Request $request, $parameter){
15         // Do stuff
16         return $this->render('myTemplate'[$parameter]);
17     }
18 }
```

Figure 14 – Exemple de controller symfony

13 REST

Les API REST (**R**epresentational **S**tate **T**ransfer) sont des API suivant un standard. Ces API doivent suivre un certain nombre de contrainte.

- Le système doit être client/serveur.
- Un serveur *stateless* chaque requête contient toutes les informations client, il n'y a pas de sessions utilisateurs au niveau du serveur.
- Utilisation d'un cache pour les requêtes.
- Chaque ressource possède un identifiant.
- messages auto-descriptifs les messages expliquent leur contenu, si le corps est du JSON avec un encodage UTF-8, cela doit être décrit dans le message.
- Architecture en couche.

Ces API permettent de déléguer un certain nombre de traitement des données aux clients. Elles utilisent des requêtes HTTP(S) et utilisent les verbes HTTP pour la manipulation des ressources.

- GET: Ce verbe permet de récupérer des ressources.
- POST: Ce verbe permet de créer des ressources.
- PUT: Ce verbe permet de modifier des ressources.
- DELETE: Ce verbe permet de supprimer des ressources.

Prenons un exemple pour y voir plus clair

URL	Verbe HTTP	Body	Result
http://example.com/api/books	GET	Vide	Retourne tous les objets "books"
http://example.com/api/books	POST	JSON	Crée une entrée et retourne l'objet (et son ID)
http://example.com/api/books/ID	GET	Vide	Retourne une entrée unique correspondant à l'ID indiqué dans l'URL
http://example.com/api/books/ID	PUT	JSON	Met à jours l'entrée correspondante à l'ID dans l'URL et retourne l'objet
http://example.com/api/books/ID	DELETE	Vide	Supprime l'entrée correspondante à l'ID précisé dans l'URL et retourne l'objet

Quatrième partie

Environnement de test

Pour la réalisation du logiciel, je vais devoir mettre en place un environnement de test le plus proche possible de l'environnement ciblé.

14 Matériel

Pour construire l'environnement de test, je vais utiliser trois machines, un vidéo-projecteur, un switch et mon ordinateur personnel comme indiqué sur la Figure 15. L'ensemble forme un réseau dont les adresses sont distribuées par le switch en DHCP, les machines se verront donner un nom de domaine.

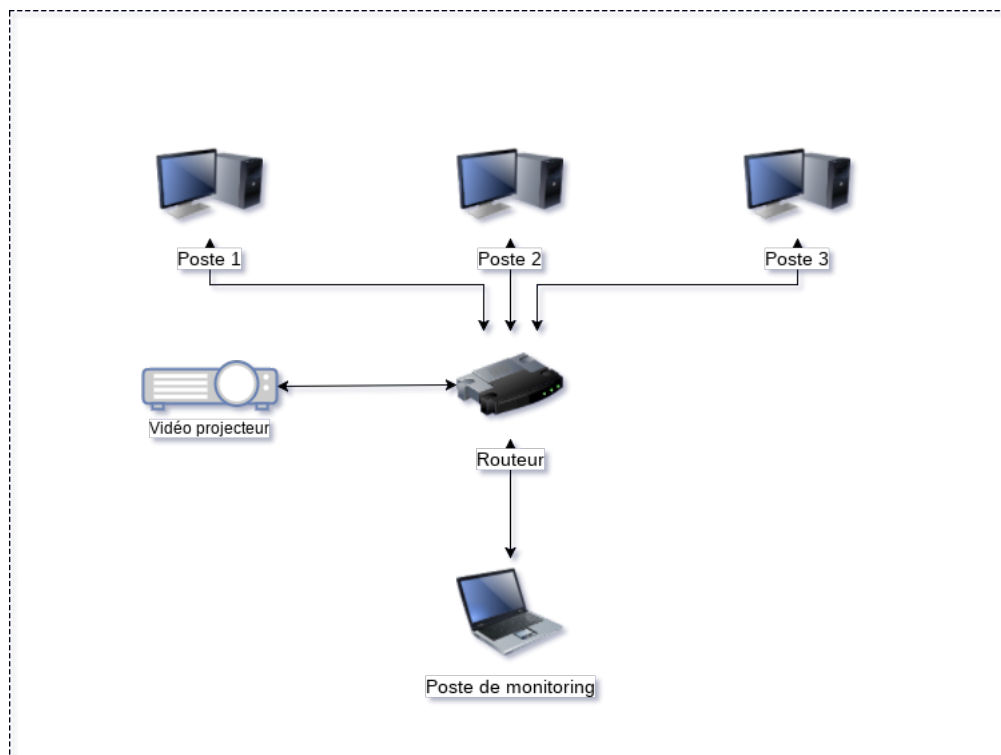


Figure 15 – Environnement de test

Deux des machines tourneront sur un système Windows, tandis que la dernière utilisera un système UNIX (debian ou ubuntu). Cet environnement peut être étendu en ajoutant des machines sur le réseau local ou en ajoutant des périphériques sur les machines, comme une Kinect par exemple.

Cinquième partie

Analyse

L'application sera composée de 4 modules comme indiqué sur la figure 5 et d'un canvas fourni par Symfony pour la création des pages (Twig).

15 Monitoring

Le service monitoring pourra utiliser plusieurs implémentations en fonction de l'entité concernée ou de son type. Pour l'instant, deux implémentations sont prévues, SNMP et Telnet avec le protocole RS-232. L'implémentation du module sera comme décrit dans la figure 16

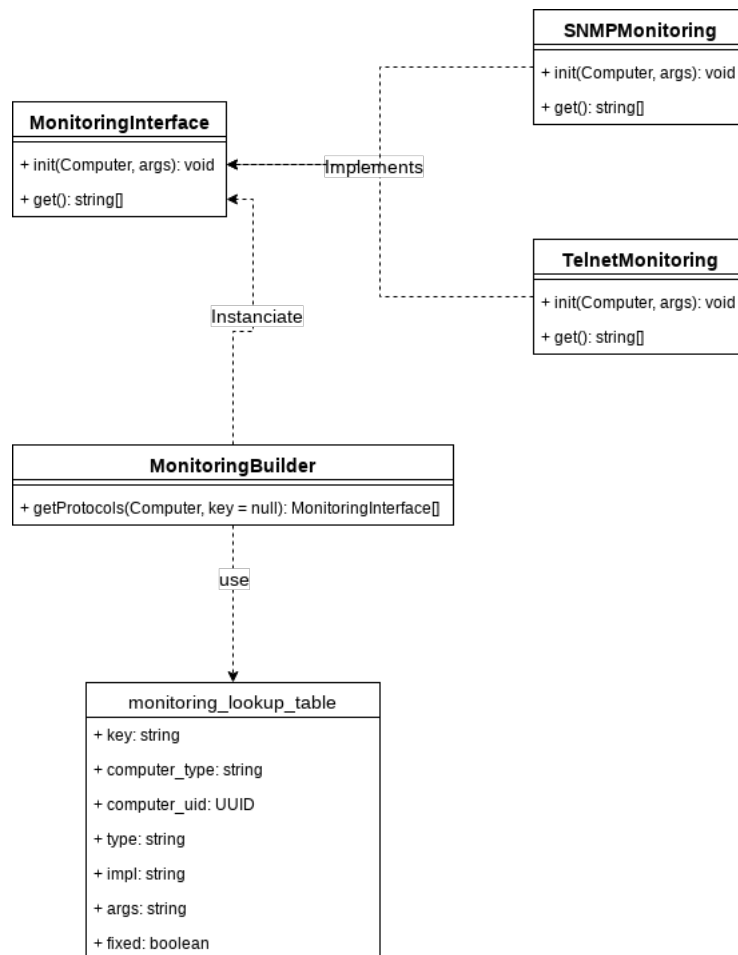


Figure 16 – Implémentation du module monitoring

L'entité `monitoring_lookup_table` sert en quelques sortes de guide d'instanciation au `MonitoringBuilder` lui indiquant quels instances de `MonitoringInterface` à utiliser et pour quelles valeurs.

15.1 Metrics

Les données récupérées par le module de monitoring sont stockées simplement stockés dans une table comme indiqué dans la figure 17.

Une métrique est donc un simple couple clef/valeur relié à un ordinateur et possédant une date de création.

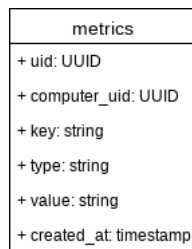


Figure 17 – Implémentation de metrics

16 Wake On Lan

La classe WOLProvider contient une méthode `wakeUp` qui prends en entrée un objet de la classe `Computer`, et avec ces informations, envoie le "magic paquet" sur le réseau. Cette méthode met à jour le status de l'objet `Computer`.

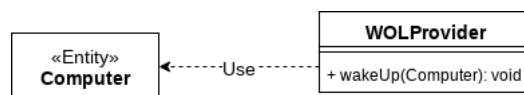


Figure 18 – Implémentation de wake on lan provider

17 Exécution de scripts

Le module d'exécution de scripts à distance, l'utilisation du protocole SSH sera utilisée grâce à la méthode PHP `ssh2_exec()`.

18 Login

Le module login est en partie fourni par Symfony comme décrit dans la [documentation](#).

Sixième partie

gestion de projet

La gestion de ce projet fut assez simple car étant le seul développeur, certaines procédures ne s'appliquent pas.

19 Définition des objectifs

Pour gérer mon temps, je me suis appuyé sur un Trello, cet outil permet de faciliter une gestion de projet agile type scrum. J'ai donc défini un certain nombre de tâches à effectuer puis les ai découpé en 2 versions majeures en fonction de leurs dépendances et importance. L'ensemble de ce trello est disponible en annexe [A](#). Cette manière de fonctionner me correspond particulièrement bien, j'ai tendance à vouloir faire tout en même temps. Travailler de cette manière me permet de canaliser cette tendance et de travailler sur plusieurs choses en les effectuant dans le bon ordre de priorité. De plus cela me permet de voir combien de tâches sont actuellement en cours et donc si il y en a trop, de ne pas en commencer une nouvelle.

20 Comptes rendu hebdomadaires

J'ai, dans le cadre de ce projet fait des compte rendu hebdomadaire assez informels avec mon encadrant chaque jeudi matin. Lui expliquant qu'est ce que j'avais fait depuis la dernière fois, ou j'en étais et qu'est ce que je prévoyais pour la prochaine fois. De plus, mon encadrant a accès au Trello et peut ainsi suivre en temps réel l'état d'avancement du projet.

Septième partie

Réalisation

La réalisation du projet se sépare en deux parties, le serveur web en php et l'affichage avec les technologies web classiques : HTML, CSS et Javascript. Une liste des technologies utilisées ainsi que les liens vers leur documentations officielles disponible en annexe [B](#)

21 Serveur web

Durant la programmation, j'ai mis à jour un wiki disponible sur le serveur gitlab ou est hébergé mon code. Ce wiki est disponible en Annexe, il détaille assez bien l'architecture et les fonctionnalités de la partie serveur. Concernant cette partie, je vous renvoie donc aux annexes [C](#), [D](#), [E](#), [F](#) et [G](#).

22 Interface utilisateur

Un guide de l'utilisateur est aussi disponible sur le wiki, je vous renvoie donc à l'annexe [H](#)

23 Tests

De par le fait que ce projet est un projet web et dépendant de son parc de machine en réseau. J'ai fait le choix de ne pas faire de tests unitaires, ni de tests fonctionnels automatiques. Car la mise en place de ces tests est beaucoup trop coûteux en temps. Pour la réalisation des tests, j'ai du effectuer des tests fonctionnels de l'api et de l'interface manuellement suivant des scénario prévu à l'avance.

23.1 Scénarios API

Pour toutes les API effectuant des opérations de type CRUD, différents scénario utilisant ces services ont été effectué, permettant de découvrir quelques bug mineurs qui ont été résolu dans la foulée.

Les opérations de type CRUD de l'API ont été vérifiées lors de leur conception, pour le reste des "endpoints", la vérification de leur bon fonctionnement à été effectuée dans le cadre des tests de l'interface utilisateur. Quelques bugs mineurs ont été découverts résolu dans la foulée.

23.2 Tests des pages web

Les différentes pages web ont été testée dans leur cas limite et ont permis de résoudre un grand nombre de bugs, en voici un exemple:

23.2.1 Séquence ON/OFF

Etat initial : la pool d'ordinateur est éteinte.

Séquence 1:

- Appui bouton ON
- Pas de temps d'attente
- Appui bouton OFF

Constatations : Après 1 minute 0-1 ou 2 PC éteints, le reste étant allumé.

Explication : Le script d'extinction ne peut être envoyé qu'une fois que le service sshd est lancé et en état d'accepter des connexions, le cas échéant, le script n'est pas exécuté et le serveur l'ignore.

Conclusion : C'est le comportement souhaité, RAS

Séquence 2:

- Appui bouton ON
- Attendre 2 minutes
- Appui bouton OFF

Constatations : Après 1 minute, l'ensemble des PC sont éteints.

Conclusion : C'est le comportement souhaité, RAS

Huitième partie

Conclusion

Au final, le projet est fonctionnel même si je n'ai pas réussi à remplir tout les objectifs, les fonctionnalités principales sont implémentées.

Avec un peu de recul, il est vrai que je ne m'y prendrais pas de la même manière pour un certain nombre de chose. Par exemple, je mettrais en place le même design pattern (Factory) qui a été utilisé pour réaliser le service de monitoring, dans un certain nombre d'autres services comme le service de contrôle (SSH) ou bien de wake on lan. J'aurais bien aimé aussi pouvoir utiliser plus en profondeur le moteur d'ORM avec des fonctionnalités que j'ai découvert trop tard dans l'année pour les utiliser dans ce projet.

De plus ce projet ne m'a pas permis d'apprendre beaucoup, en dehors des technologies assez marginales telles que SNMP et Telnet. Les technologies utilisées sont assez communes et je les avais déjà en partie mises en pratique dans d'autres projets. En revanche la réalisation de cette application m'a permis de consolider ma confiance en moi et en mes capacités.

Annexes

A

Trello

prd ☆ Personnel 🔒 Privé

A faire V2

- IHM V2

Modale plan info

☰
- API V2

module importation
- logger

A faire V1

En cours

- API V2

Telnet RS 232 - API

Done v1

- TESTS V1

Tests

☰ 💬 1
- API V1

Documentation des API
- IHM V1

Guide utilisateur
- V1

Script d'installation host
- API V1

Gestion utilisateur

💬 1
- IHM API V1

Bouton ON

IHM API V1

Bouton OFF

IHM V1

Ping display

IHM API V1

Entité Computer

IHM V1

Gestion profil connection IHM

API V1

Standalone tasks Symfony

IHM V1

Fenêtre simple amélioration

**IHM V1**

Connection/déconnexion bouton

IHM V1

Gestion Login/Mot de passe IHM

API V1

Gestion profil connection API

API V1

Gestion SSH

IHM V1

Scripting cas par cas

IHM API V1

Plan avec pastilles

**API V1**

Gestion Login/Mot de passe API

API V1

Gestion connection par clef

IHM **V1**

Gestion connection par clef

API **V1**

Wake on lan

IHM **V1**

Wake on lan

API **V1**

Ping API

Done v2

API **V2**

SNMP - API

V2

script update crontab

API **V2**

shortcut Scripts

API **V2**

Shutdown improvment

**IHM** **V2**

Monitoring - IHM

V2

Script configuration host windows

API **V2**

Monitoring - Controllers

idées

Script de génération de config server
pour un ordinateur

Recherches vidéoprojecteurs

1

scenarios de test



B

Technologies

Technologies

In this page, I will provide a list of the technologies used in this project and a link to each of their documentations.

Server side technologies

PHP/Symfony

The server is coded in PHP 7.1 and use the framework Symfony 4

- Documentation PHP : <http://php.net/docs.php>
- Documentation Symfony : <https://symfony.com/doc/current/index.html>

Doctrine

The database SQL request are handled by Doctrine 2.

- Documentation : <http://docs.doctrine-project.org/en/latest/>

Front side

HTML Templating

To build Html pages, I chose to use the TWIG PHP template engine, this engine is activated by default in Symfony 4 and works well with the PHP framework. *
Documentation : <https://twig.symfony.com/doc/2.x/>

JavaScript

I use a Javascript framework called VueJS in addition to the timeless JQuery

- Documentation VueJS: <https://vuejs.org/v2/api/>
- Documentation JQuery : <http://api.jquery.com/>

CSS

I also use a CSS framework called MaterializeCSS

- Documentation : <http://materializecss.com/about.html/>

C

Structure du projet

Project file structure

The project follow the Symfony best practices. All the server code of the application is located in the `src/` folder and the configuration files in the `config/` folder. The front-end code is located in the `template/` folder for the twig templates and the javascript code is in the `public/assets/js/views` folder.

Server code organisation

This part of the application also follow the Symfony best practices, the following folders are the most important : * `src/Controller`, this folder contains the code of the controllers, the entry point of the application. * `src/Entity`, this folder contains the Doctrine's ORM entities used in the application and some enumeration. * `src/Services`, this folder contains the different modules that actually do something.

Controllers

There is two types of controllers in this application, the “template” ones and the “api” ones. The first ones are the `ComputerController` and the `SecurityController`, the second ones are stored in the `src/Controller/api/` folder there is the `ComputerAPIController` and the `UserComputerAPIController`.

The templates controllers

The main function of theses controllers is to render the TWIG templates (learn more about TWIG templates). Theses controllers sets the routes and the security level needed for accessing the webpage.

The API controllers

There is only two API controllers and their functions are documented in the following wiki pages : * `ComputerAPIController` documentation * `ComputerUserAPIController` documentation

Services

The services provided by this application are the following are detailed below.

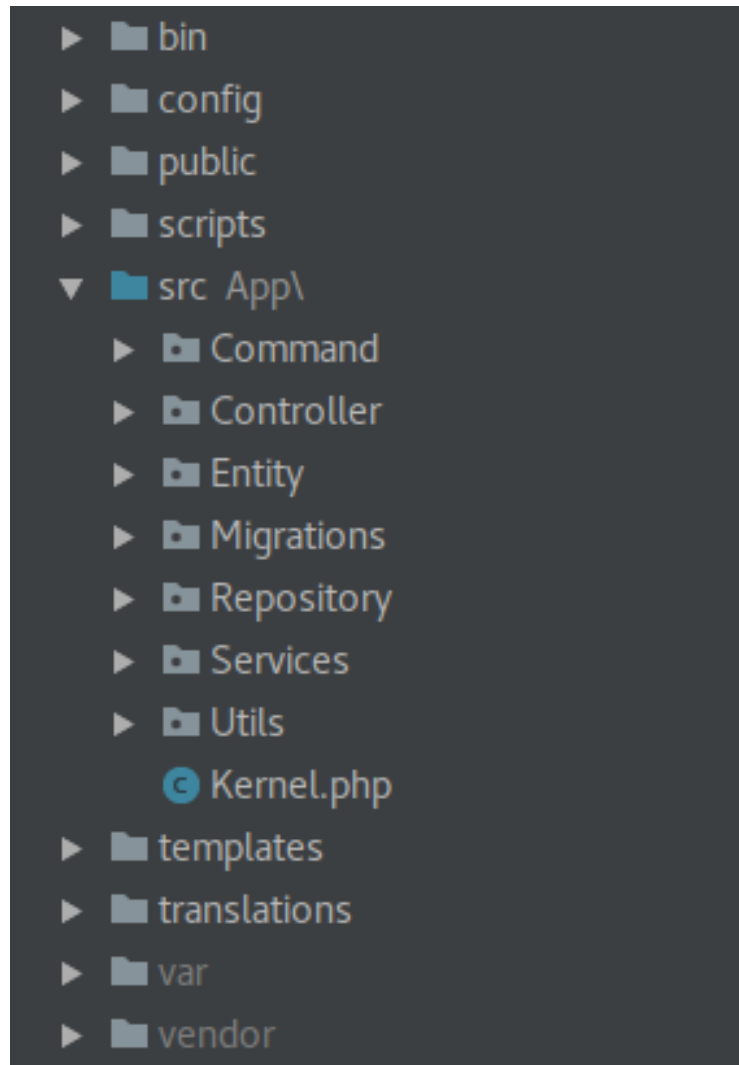


Figure 1: project architecture

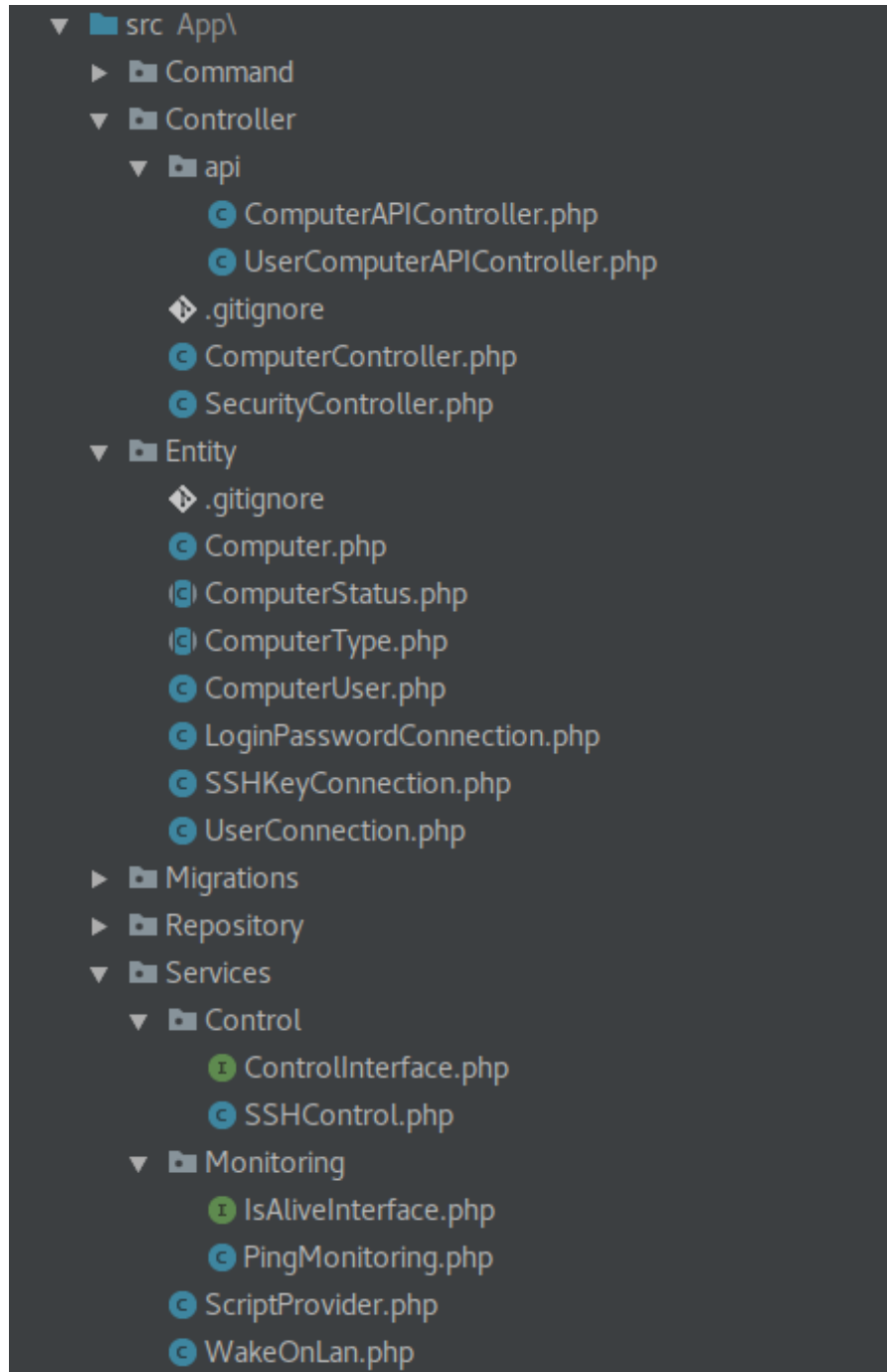


Figure 2: project server file organisation

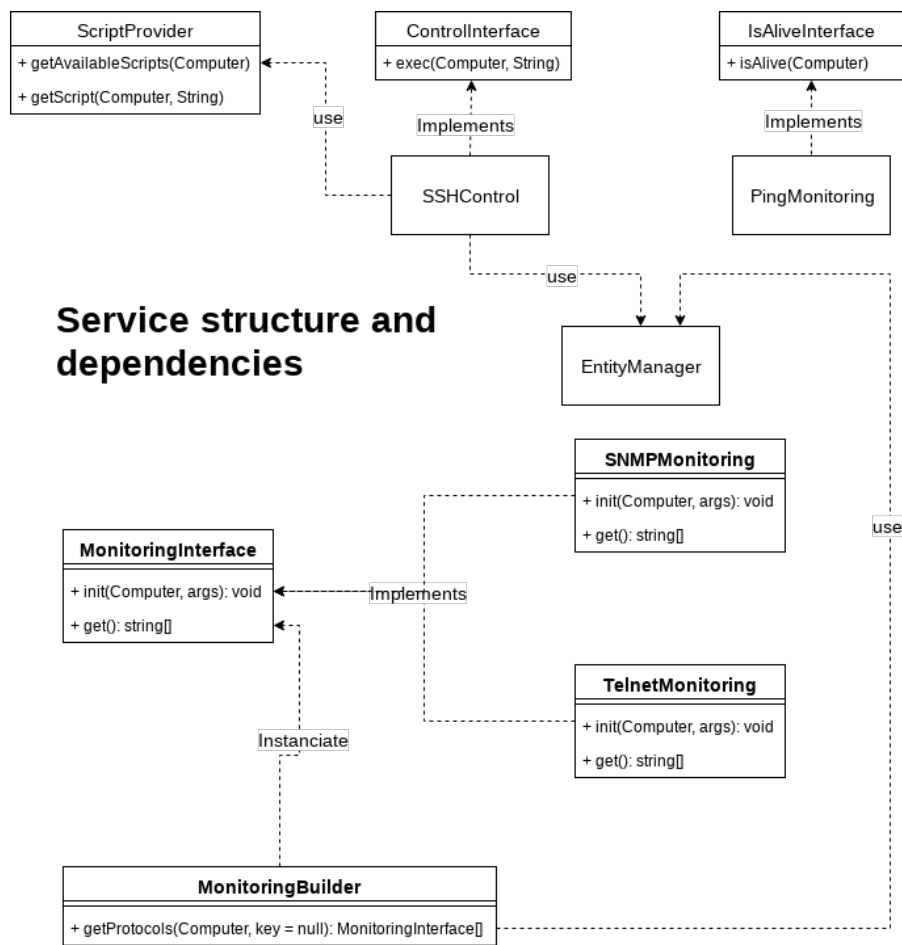


Figure 3: Service diagram

Script provider

This service (named ScriptProvider) provide the list of scripts that can be sent to the computers and the absolute path of theses scripts by accessing the filesystem. The scripts has to be stored in the filesystem following this convention.

Example :

```
use App\Services\ScriptProvider;

class SomeClass{

    public function someMethod(Computer $computer, ScriptProvider $scriptProvider){
        // Allow you to fetch all the available scripts.
        $scripts = $scriptProvider->getAvailableScripts($computer);

        // Allow you to fetch the absolute path of the script.
        $scriptPath = $scriptProvider->getScript('shutdown');
    }
}
```

Wake On Lan

This service (named WakeOnLan) provide a method that can wake the computer put in the parameter if the Mac address has been set. For this service to work, the wol.broadcast parameter has to be set.

Example :

```
use App\Services\WakeOnLan;

class SomeClass{

    public function someMethod(Computer $computer, WakeOnLan $wol){
        // Send the magic packet
        $wol->wake($computer);
    }
}
```

The IsAlive interface

This interface provides methods to check if a computer can be reach in the network, this interface is implemented using a ping on the ip address or domain name property of the object.

Example :

```

use App\Services\Monitoring\IsAliveInterface;

class SomeClass{

    public function someMethod(Computer $computer, IsAliveInterface $isAlive){
        // Update the computer's status and return a boolean
        if($isAlive->isAlive($computer)){
            return 'He\'s alive';
        }
        return 'I think he\'s dead ...';
    }
}

```

The ControlInterface

This interface provides a methods allowing you to send and execute scripts on the computers. This service use the Script provider service so the requirements of this service shall be met.

Example :

```

use App\Services\Control\ControlInterface;

class SomeClass{

    public function someMethod(Computer $computer, ControlInterface $control){
        // send and execute the shutdown script to the $computer.
        $control->exec($computer, 'shutdown');
    }
}

```

The MonitoringInterface

This interface provides a method allowing you to fetch information on the computers. The implementation of this interface should be instantiated using the Monitoring Builder class that extract the correct implementation from the database in the monitoring_loockup_table.

Example :

```

use App\Services\Monitoring\Protocols\MonitoringBuilder;
use App\Services\Monitoring\Protocols\MonitoringInterface;

class SomeClass{

    public function someMethod(Computer $computer, MonitoringBuilder $builder, $key){

```

```

        $protocols = $builder->getProtocols($computer, $key);

        for($protocols as $protocol){
            $value = $protocol->get();
        }
    }
}

```

Database

The database model is represented in the following figure.

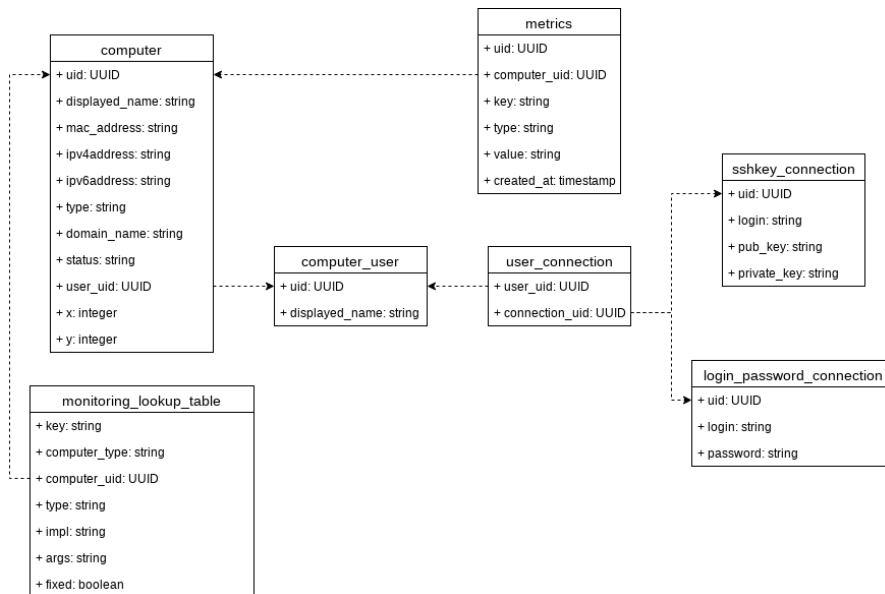


Figure 4: bdd

D

Stockage des scripts

How scripts are stored

This application provides an extensive and fully modular way of storing the scripts that will be sent to the computers. The scripts are stored in the filesystem of the server in the directory “scripts/”.

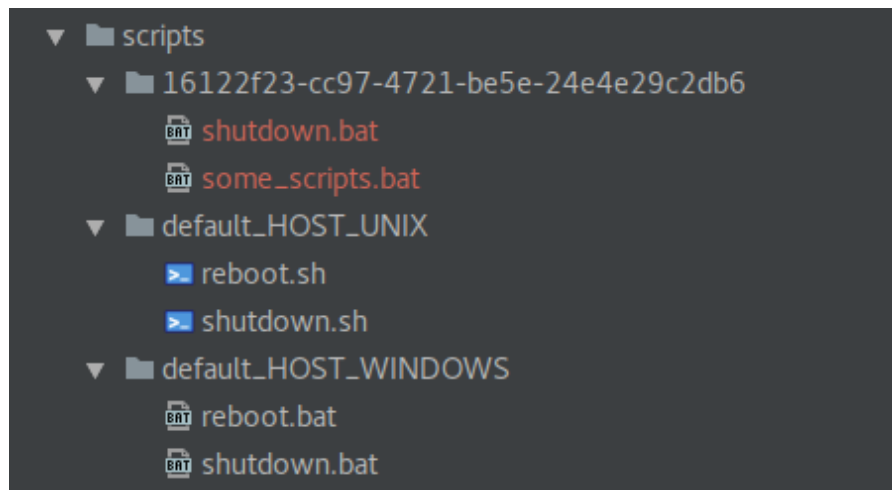


Figure 1: script tree

There is two types of directories in there, the ones with the “default_*” and the others. The first ones store the defaults scripts for the ComputerType after the first underscore. In the other hand you can override theses default scripts by creating a directory with the UUID of the computer as name to store the scripts for this computer.

E

Computers API documentation

Computer API documentation

A computer object is represented like the following :

```
{
  "uid": "db10716f-a084-4648-815c-caeb5ee9810a", // Object unique identifier
  "x": 53, // Horizontal coordinate of the Computer
  "y": 31, // Vertical coordinate of the Computer
  "displayName": "Computer 2", // Name of the object
  "macAddress": "64:31:50:3F:64:D6", // Mac address, this is mainly used by
  "ipv4Address": "169.254.72.2", // IP address
  "ipv6Address": null, // IPv6, the IPv6 support is not implemented
  "type": "HOST_WINDOWS", // Type of the Computer
  // for more information, see the type attribute
  "domainName": null, // Domain name
  "status": "off", // Status, for more information see the status attribute
  "userId": "efb8c7f2-0b5e-4c12-9144-111ec3e64e69" // Identifier of the user used for the Computer
}
```

Type attribute

The type attribute represent the type of the computer, the different types are the following : * 'HOST_WINDOWS' => Computer with a Windows exploitation system * 'HOST_UNIX' => Computer with an Unix exploitation system * 'UNDEFINED' => Undefined (obviously), this is the default value.

Status attribute

The status attribute represent the state of the computer, this value is updated every 60 seconds. This attribute can take the following values : * 'off' => The computer does not respond to the requests. * 'on' => The computer is responding to the requests.

CRUD

The API provide a set of operation allowing to manage the computers stored in the database. ### Fetch the list of computer

GET /api/v1/Computers/

The response will contains a list of computer objects.

Fetch one specific computer

GET /api/v1/Computers/{uid}/

The {uid} element is the identifier of the computer, the response will contain a computer object.

Creating a computer

POST /api/v1/Computers/

The body of the request has to contains a computer object, all the attributes can be null except the displayName attribute, the uid field should be set to null or not been set at all.

Updating a computer

PUT /api/v1/Computers/

Same as the creating computer API endpoint but the uid field should be set to the identifier of the computer that we want to update.

Deleting a computer

DELETE /api/v1/Computers/{uid}/

The {uid} element should be set to the identifier of the computer to be deleted

Wake on LAN

The wake on LAN features are exposed on the following endpoints :

GET /api/v1/Computers/wake

GET /api/v1/Computers/{uid}/wake

The first one wake every computers in the database with a mac address not set to null, the second one will wake the computer with the {uid} identifier.

Scripts

Fetching scripts

The application can run scripts on available computers.

GET /api/v1/Computers/{uid}/Scripts/

This endpoint can fetch the script list for the specified computer and return a list of string containing the script names. ### Running scripts The application can run the script with the following endpoint.

```
GET /api/v1/Computers/{uid}/{script_name}/
```

This endpoint will return a JSON response containing “ok” if everything’s fine and a 500 error code with a JSON response containing “{”error“:”script_execution_failed“}” in the case of a failure.

Metrics

The metrics endpoint are used to fetch the metrics objects represented as follow :

```
{
  "uid": "UUID" ,
  "computerUid": "UUID",
  "key": "name_of_the_metric",
  "type": "see_the_type_section_below",
  "value": "value",
  "createdAt": "<- explicit_name"
}
```

Type

The type field is used for display (mainly string and numeric values) and is also used in the monitoring process. For more information in the monitoring process and the importance of the type field, see the documentation of the MonitoringLookup API.

Get for all computers endpoint

The get all metrics endpoint is bound to this url :

```
GET /api/v1/Computers/Metrics?since={since}
```

The since parameter is used to specify the period of monitoring we want to fetch. the default value is 1 hour.

Get for one computer endpoint

This endpoint allow you to fetch all the metrics since {since} for the computer with the uid {uid}

```
GET /api/v1/Computers/{uid}/Metrics?since={since}
```

Shortcut endpoint

In order to save a lot of processing time, the application provide a set of shortcut endpoint. ### Status We can fetch all the status with the following endpoint

GET /api/v1/Computers/Status/

The response should be an object with the computer's uid as keys and the status as value. Here's an example :

```
{
  "eb90946e-dd05-46e7-a886-b0978c521b41": "off",
  "fb094ba0-a491-4d40-b47c-098c0493ae6d": "off",
  "db10716f-a084-4648-815c-caeb5ee9810a": "off",
  "2005e9c0-e4ec-40e8-abc3-132e15d09bd5": "ok",
  "9d530881-6fcc-4d9d-84f0-ccf36f5a0166": "off"
}
```

Scripts

We can fetch all the scripts with the following endpoint

GET /api/v1/Computers/Scripts/

The response should be an object with the computer's uid as keys and the list of script as value. Here's an example :

```
{
  "eb90946e-dd05-46e7-a886-b0978c521b41": ["shutdown","reboot"],
  "fb094ba0-a491-4d40-b47c-098c0493ae6d": ["shutdown","reboot"],
  "db10716f-a084-4648-815c-caeb5ee9810a": ["shutdown","reboot"],
  "2005e9c0-e4ec-40e8-abc3-132e15d09bd5": ["shutdown","reboot"],
  "9d530881-6fcc-4d9d-84f0-ccf36f5a0166": ["shutdown","reboot"]
}
```

F

ComputerUsers
documentation

API

ComputerUser API documentation

The ComputerUser can be shared between multiple computers, this object contains the information required in order to connect to the computer.

An ComputerUser object is represented like the following :

```
{
  "uid": "efb8c7f2-0b5e-4c12-9144-111ec3e64e69", // Unique identifier of the object
  "displayName": "user",                       // The displayed name of the object
  "login": "username",                         // The username used to log in the the c
  "password": "123456"                        // The password associated to the usernam
}
```

CRUD

The API provide a set of operation allowing to manage the ComputerUsers stored in the database. ### Fetch the list of ComputerUsers

GET /api/v1/Users/

The response will contains a list of objects containing two attributes : * uid * displayName

Fetch one specific ComputerUser

GET /api/v1/Users/{uid}/

The {uid} element is the identifier of the ComputerUser, the response will contain a ComputerUser object with only two attributes, the uid and the name.

Creating a ComputerUser

POST /api/v1/Users/

The body of the request has to contains a ComputerUser object, all the attributes can be null except the displayName attribute, the uid field should be set to null or not been set at all.

Updating a ComputerUser

PUT /api/v1/Users/

Same as the creating ComputerUser API endpoint but the uid field should be set to the identifier of the computer that we want to update.

Deleting a ComputerUser

DELETE /api/v1/Users/{uid}/

The {uid} element should be set to the identifier of the ComputerUser to be deleted

G

**MonitoringLoockup
documentation**

API

Monitoring lookups api documentation

This API provide a set of tools allowing you to perform CRUD operations on the Monitoring lookup entities.

CRUD

The Monitoring lookup entity are represented by the following JSON object :

```
{
  "key": "The name of the variable",
  "computerType": "Type of the computer to watch",
  "computerUid": "Uid of the computer to watch",
  "type": "Type of the variable",
  "impl": "The implementation of the monitoring",
  "args": "Arguments",
  "fixed": "Fixed variable"
}
```

Get all endpoint

This endpoint allow you tu fetch all the entities.

GET /api/v1/MonitoringLookups/

Get one endpoint

This endpoint allow you to fetch one entity

GET /api/v1/MonitoringLookups/{key}/

Upsert endpoint

This endpoint allow you to insert or update an entity

POST/PUT /api/v1/MonitoringLookups/

Delete

This endpoint allow you to delete an entity

DELETE /api/v1/MonitoringLookups/{key}/

A logo consisting of a blue square containing a white capital letter 'H'.

Manuel utilisateur

The main page

The first thing you see when you start the application is this page :

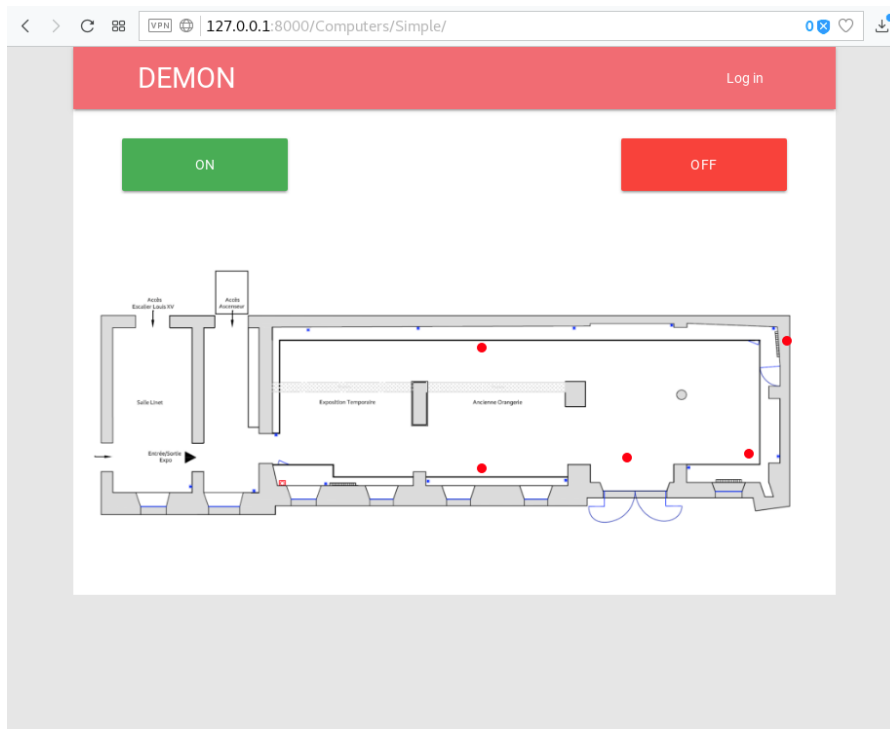


Figure 1: simple view

The “ON” button will send the wake signal to every known computer, and the “OFF” button will send a shutdown signal to every known computer. See, it’s not rocket science.

In the map in the bottom of the page, you will find a green chip if the computer in this location is powered on and a red one if not.

The navigation bar

At the top of every page of the application there is the navigation bar.

By clicking on the “DEMON” logo in the left side, you will be redirected to the main page.

The right side of the bar change if you’re logged in or not :

Not logged in

If you're not logged in, you will see a "log in" button that will take you to the login page.



Figure 2: navbar not logged in

Logged in

If you're logged in, you will see a welcome message with the name of the user logged in, a logout button and finally a Home button that will take you to the Computer list page



Figure 3: navbar logged in

The computer list page

To see this page, you need to be logged in as an admin.

This page display an abstract view of each known computer in the application. The list is composed of 4 columns : * Name: the name of the computer. * Status: the status of the computer. * Action: the actions that can be applied to the computer, there is two options : * access the monitoring page of the computer * editing the object * Execute: a dropdown that allow you to send scripts to the computer.

The "+" button send you to the create computer form.

The computer form page

The computer form is the same for creating and modifying the computers :

You can choose an existing user for the computer or create one by clicking the "create new user" button.

The screenshot shows a web browser window with the URL `127.0.0.1:8000/Computers/`. The page has a red header with the title 'DEMON' and navigation links 'Welcome admin!', 'Log out', and 'Home'. Below the header is a table with the following data:

Name	Status	Actions	Execute
Computer 2	off		reboot
Computer 7	off		shutdown
Computer 3	off		EXECUTE
Computer 5	off		EXECUTE
Status	off		EXECUTE
Computer 4	off		EXECUTE

Below the table is a red circular button with a white plus sign. The browser's address bar shows the URL `127.0.0.1:8000/Computers/` and the status bar shows the path `127.0.0.1:8000/Computers/`.

Figure 4: computer list

The screenshot shows a web form titled 'DEMON' with a red header. The header contains the text 'Welcome admin!' and links 'Log out' and 'Home'. The form has the following fields and buttons:

- Name:** A text input field containing 'Computer 3'.
- IPv4:** A text input field containing '169.254.72.3'.
- IPv6:** A text input field containing '0:0:0:0:0:0:0:0'.
- Domain name:** A text input field containing 'domain name'.
- Mac address:** A text input field containing '08:2E:5F:05:58:4C'.
- User:** A text input field containing 'passw'.
- CREATE NEW USER:** A green button located to the right of the 'User' field.
- PLACE COMPUTER:** A green button located at the bottom left of the form.
- SUBMIT >:** A green button located at the bottom right of the form.

Figure 5: computer form

The screenshot shows a web application interface. At the top, there is a dark red header bar with the word "DEMON" on the left and "Welcome admin ! Log out Home" on the right. Below the header, there is a light gray sidebar on the left with the word "Name" at the top. The main content area is white and contains a form titled "Create new user". The form has three input fields: "Name" with the placeholder text "name", "Login" with the placeholder text "login", and "password" with the placeholder text "password". At the bottom right of the form, there are two buttons: "CLOSE" and "SUBMIT".

Figure 6: computer form new user

And finally you can choose the position of the computer on the map by clicking the “place computer” button. And dragging the chip on the map.

The monitoring page

The monitoring page shows the metrics associated to the computer, there is two way of displaying the metrics, as string and as numeric values :

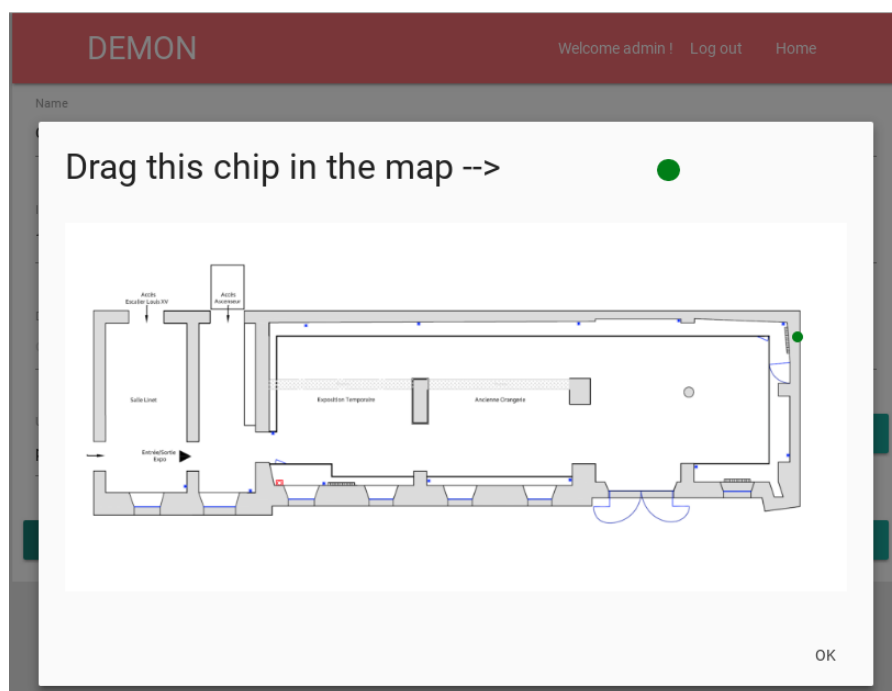
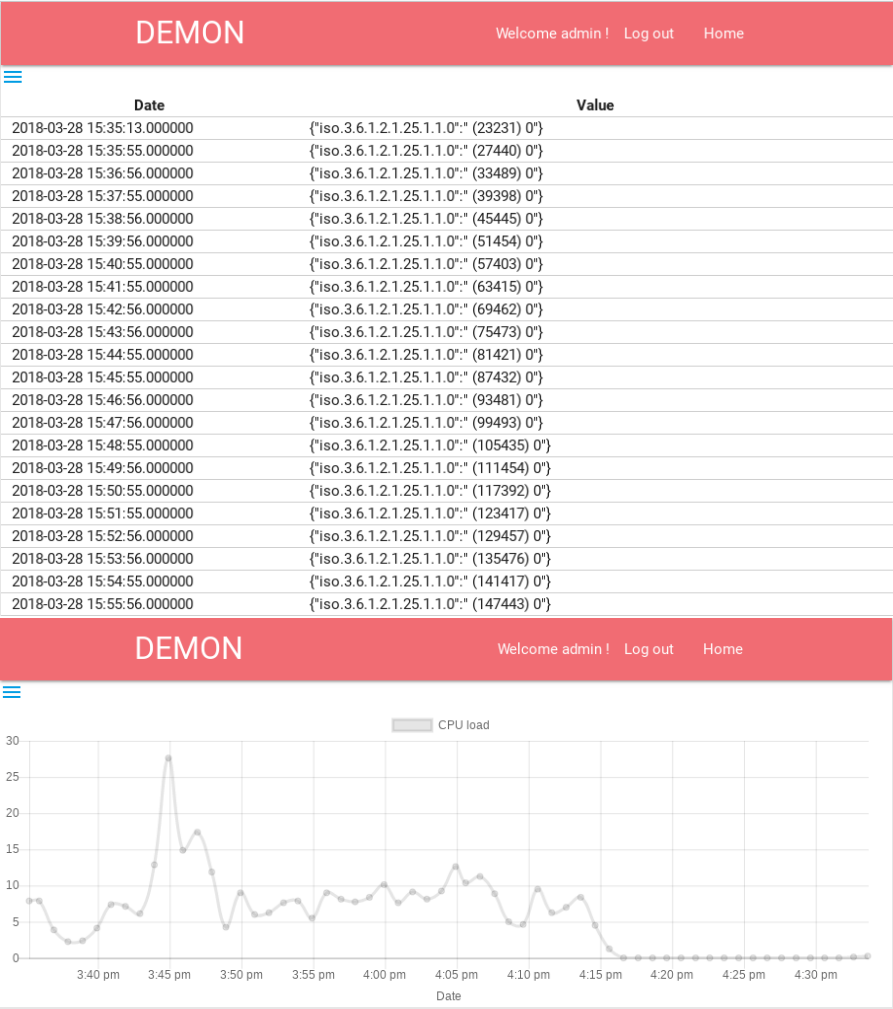
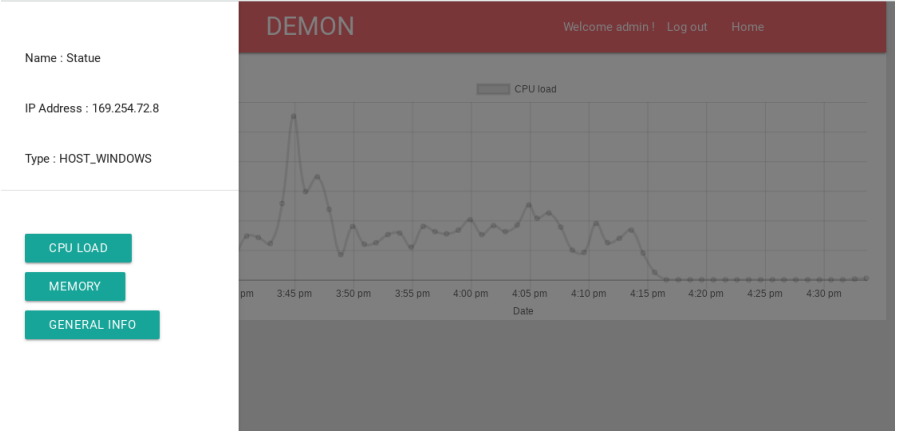


Figure 7: computer form place computer



The page display only one metric at a time, to view other metrics, there is a menu available on the side and displayed when the top left button is pressed.



I

Rapport précédent

1

Contexte de la réalisation

1 Acteurs, enjeux et contexte

Le projet de Recherche Sculpture 3D s'attache à faire découvrir au grand public des sculptures du patrimoine culturel local par le biais d'une exposition interactive digitale qui aura lieu au musée des beaux-arts de Tours. De nombreux autres ateliers sont en cours de développement pour cette exposition. Donc mon projet est pour concevoir une architecture réseau à gérer le démarrage et l'extinction des machines et les autres matérielles (vidéo projecteur, kinect) ainsi que permettre d'exécuter les scripts à la cible et afficher les états de la cible.

2 Objectifs

L'objectif du projet est de créer une architecture (plateforme web) qui peut allumer l'ordinateur et éteindre l'ordinateur à distance ainsi que permettre de surveiller l'état de l'ordinateur. De plus, un ordinateur et un vidéo projecteur sont séparés par groupe et il s'agit d'allumer et d'éteindre les matériels ensembles et il est obligatoire qu'il puisse exécuter les scripts sur l'ordinateur distant. L'interface est réalisé par le navigateur et il est facilement pour les personnes à utiliser et surveiller les états des matérielles dans la musée.

3 Hypothèse

Si le système de l'ordinateur est windows, j'utiliserai le API de windows. Si le système de l'ordinateur est Linux, j'utiliserai le API de Linux.

4 Bases méthodologiques

Dans le cas du développement d'une plateforme web, j'ai décidé d'utiliser les langages liés à ce type d'application (HTML, CSS, JavaScript, PHP ...). Pour réaliser allumer l'ordinateur à distance, j'utilise le technologie « Wake-On-Lan » et pour obtenir les états de la cible, j'utilise la technologie WMI pour obtenir les états de la cible. Pour la modélisation des fonctionnalités du système, j'ai choisi d'utiliser le langage UML.

2

Description générale

1 Environnement du projet

Mon projet est utilisé dans le musée pour gérer les matériels, je construirai un réseau privé dans le musée et j'utiliserai un ordinateur pour contrôler et surveiller les autres équipements. Dans la plateforme certains matériels sont séparés par groupe, par exemple, un ordinateur avec un vidéo projecteur, les gardiens peuvent allumer et éteindre les matériels ensembles en utilisant un ordinateur. Et il permet de surveiller les états des matériels et exécuter les scripts sur l'ordinateur distant par le biais du navigateur.

2 Caractéristiques des utilisateurs

Au cours de ce projet, il y a deux types d'acteurs qui auront chacun des autorisations différentes sur la plateforme web :

- les personnes externes : ils n'ont pas accès à la plateforme web
- les employés : ils ont un accès avec le mot de passe et effectuent les opérations, par exemple, allumer ou éteindre.

3 Fonctionnalités du système

Après avoir passé une authentification, l'interface de Web que je devrai réaliser est comme l'image suivante. L'écran dans la fenêtre est pour afficher les ordinateurs dans le réseau privé, on peut ajouter, supprimer et modifier un ordinateur en fonction des boutons à droite pour manipuler les machines. Chaque ordinateur va afficher les informations par ligne.

Quand on fait un clic droit, on peut voir la liste de manipulation pour l'ordinateur sélectionné. On peut éteindre et allumer l'ordinateur à distance selon le choix dans la liste. On peut choisir le document d'un script avec le bouton « Ouvrir Script » et l'exécuter lors d'appuyer le bouton « Exécuter Script ». La fenêtre en bas est utilisée pour afficher des informations de l'état de chaque machine, par exemple, le système, le nom d'utilisateur, les matériels.

L'état on/off est utilisé pour afficher si l'ordinateur est allumé ou pas, « on » représente que l'ordinateur est allumé, « off » représente que l'ordinateur est éteint. On peut voir un carré

	Nom de l'ordinateur	L'adresse de IP	Masque	Port	L'adresse MAC	Login	Etat on/off
<input type="checkbox"/>							
<input type="checkbox"/>							

Ajouter PC
Modifier PC
Supprimer PC

Ouvrir Script
ExécuterScript

Quitter

Des informations sur PC sélectionné

Figure 1 – l'interface de logiciel

devant chaque ligne et il est utilisé pour sélectionner les plusieurs ordinateurs. Après avoir sélectionné les plusieurs objets, on peut réaliser les fonctionnalités, par exemple, allumer, éteindre et exécuter un script ensemble.

4 Structure générale du système

Lors de ce projet, il est nécessaire de construire un réseau privé dans le musée. La structure générale est comme l'image ci-dessous. Il y a deux parties dans cette image, une est la partie de réseau privé, l'autre est la partie Internet (réseau externe).

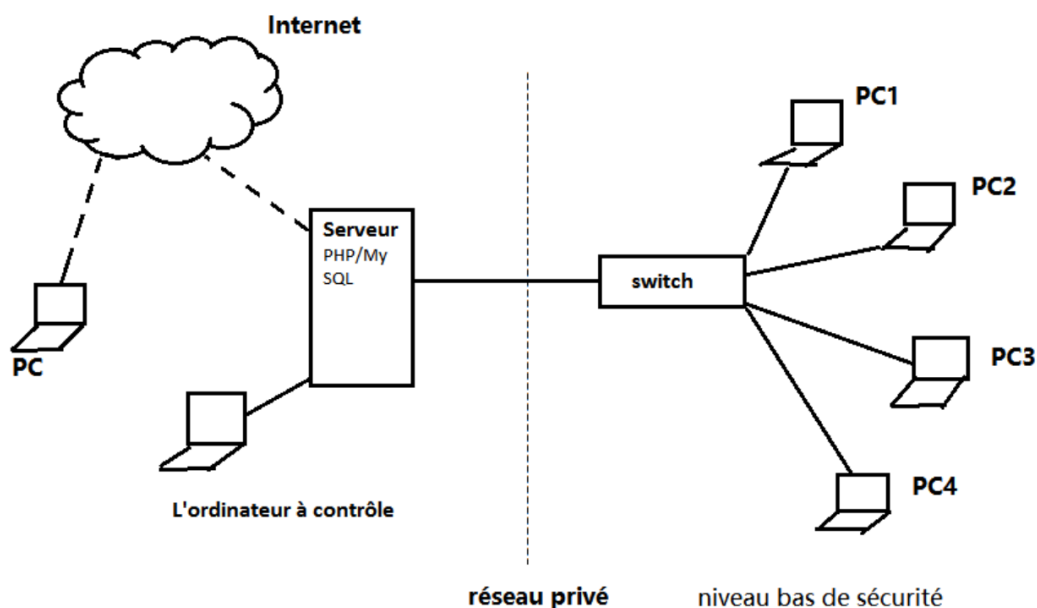


Figure 2 – le structure général de logiciel

Dans le réseau privé, on installe les documents que l'on réalisera dans le serveur pour le

projet, puis on entre la plateforme en passant une authentification sur un navigateur. Pour les composants de réseau, on met toutes les machines au niveau bas de sécurité afin d'utiliser la technologie « Wake-On-Lan » et on les paramètre pour réaliser les fonctionnalités que l'on a besoin dans le projet. Après on connecte les composants avec un switch qui donne une adresse IP à chaque cible en DHCP. On peut manipuler les cibles par biais de la plateforme, par exemple, allumer et éteindre l'ordinateur à distance, envoyer et exécuter un script à la cible, obtenir les informations de la cible, etc.

Pour la partie Internet, on peut connecter le serveur par Internet. Après on peut entrer la plateforme par biais de Internet dans une ordinateur à distance. C'est plus facile à manipuler les équipements à distance mais il envisagera les risques de sécurité. C'est compliqué mais c'est une partie à choisir dans mon projet. Donc je le ferais quand je finissais la partie de réseau privé vite.

Maintenant, j'introduis la façon de communication entre l'ordinateur à contrôle et le serveur. Le diagramme est comme l'image suivante.

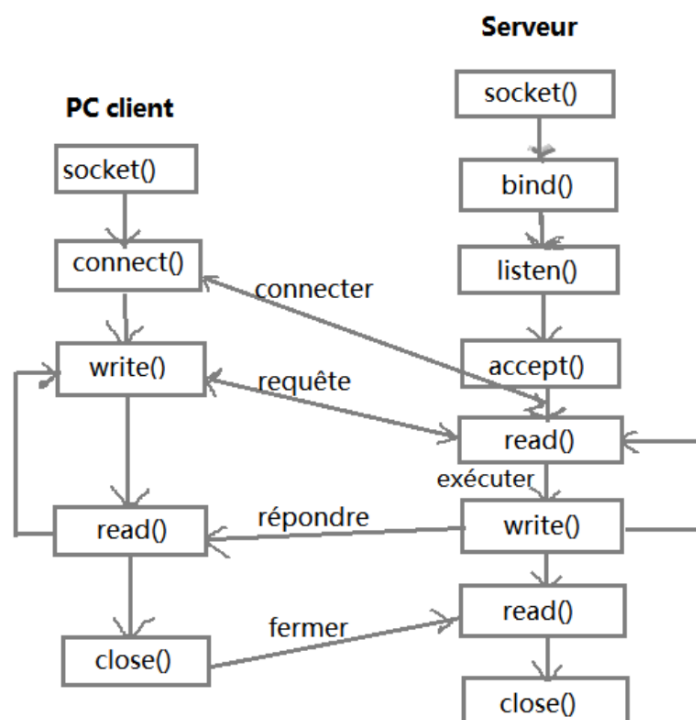


Figure 3 – la communication entre le serveur et la machine

Parce qu'il est dans le réseau privé, j'utilise la technologie « php socket » avec le protocole UDP afin de réaliser la communication. Après avoir construit la connexion, je peux envoyer un script qui contient les requêtes à la cible (par exemple le script pour étendre l'ordinateur, le script pour détecter les états de la machine destinataire, etc) ou les informations de paquet magique à la destination en utilisant la méthode « socket sendto » [5]. Dans le serveur, je mets des documents PHP pour détecter et recevoir le script, de plus, je mets des documents PHP et Mysql et HTML pour construire la plateforme. Après, le serveur va envoyer le message reçu ou le résultat pour répondre les demandes qu'elle a reçu avant. Enfin, l'ordinateur fermera la connexion entre la machine et le serveur.

3

Etat de l'art

1 Wake On Lan

WoL(Wake On Lan)est un standard des réseaux Ethernet qui permet à un ordinateur d'être allumé à distance. La technologie est réalisé en voyant un paquet magique sur la carte réseau de l'ordinateur cible. Attention, toutes les cartes réseaux et tous les BIOS ne sont pas compatible avec la réception de paquet magique. Maintenant, il est possible de lancer un WoL à travers internet, il faut que la machine soit située derrière un routeur NAT, ainsi le routeur pourra envoyer le paquet magique en UDP à son réseau privé, la machine ayant l'adresse MAC correspondante se mettra sous tension. En effet, la couche TCP/IP étant logicielle et l'ordinateur étant éteint, il ne possède pas d'adresse IP, seul un paquet UDP avec son adresse physique peut lui parvenir. Donc WoL existe en trois « version » [7] :

- Le Wake-On-Lan (WOL) désigne l'allumage à distance d'une machine via son interface Ethernet
- Le Wake-On-Wan (WOW) désigne l'allumage à distance via internet (une trame UDP/IP encapsule le "magic packet" pour le router vers la carte Ethernet)
- Le Wireless Wake-On-Lan (WoWLAN) désigne l'allumage à distance d'une machine via son interface WIFI

1.1 Théorie

On démontre comment cette technologie est réalisée par une image suivante [6].

Un petit programme exécuté sur le premier ordinateur envoie un paquet de données, ce paquet contient les octets suivants : FF FF FF FF FF FF suivis de l'adresse MAC de la carte réseau de destination répétée 16 fois. Ces données sont envoyées sur le port 7 ou sur n'importe quel port. Quand la carte réseau de destination reçoit cette suite d'octets (aussi appelée trame), elle envoie un signal à la carte mère de l'ordinateur. La carte mère allume alors l'ordinateur

1.2 Paramétrage Wake On Lan

Actuellement, la plupart des carte-mères récentes intègrent la fonctionnalité comme WoL. Avant d'utiliser cette technologie, il est nécessaire de paramétrer la configuration. Il contient des deux

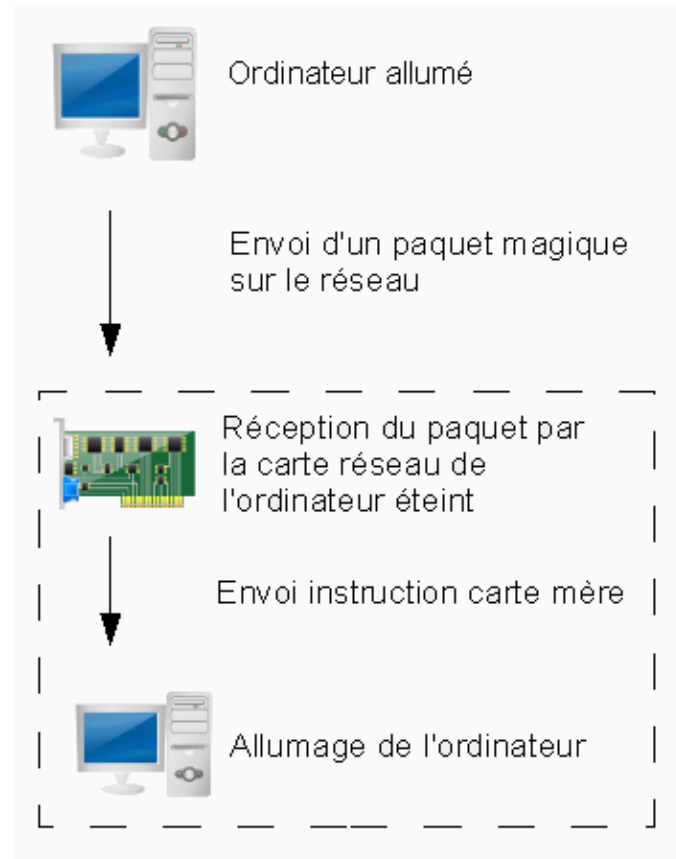


Figure 1 – Les processus de Wake On Lan

parties, une est la configuration du BIOS de l'ordinateur à distance et l'autre est la configuration du système d'exploitation.

Configuration du BIOS

Il faut entrer l'interface du BIOS quand l'ordinateur est démarré. Après on doit entrer le menu avancé dans la configuration du gestionnaire d'alimentation. On cherche quelques chose ressemblant à Wake on LAN, Wake on PCI, etc. Puis on met cette option en Enabled.

Configuration du Windows

Pour activer le Wake on Lan, il faut aller dans Démarrer > Connexions > Afficher toutes les connexions > La connexion que vous voulez configurer > Propriétés (clic droit) > Configurer > Avancé > Wake Up Capabilities > sélectionnez Magic frame dans la liste [1].

La dernière chose à faire est de récupérer l'adresse MAC de la carte réseau. On peut obtenir l'adresse MAC avec le commande `ipconfig/all` dans la console Windows.

2 WMI

Windows Management Instrumentation (WMI) est un système de gestion interne de Windows qui prend en charge la surveillance et le contrôle de ressources systèmes via un ensemble d'interfaces [2]. Il fournit un modèle cohérent et organisé logiquement des états de Windows. Il permet à des scripts de gérer Windows localement ou à distance. C'est grâce à WMI que le composant Propriétés système de Windows peut afficher les propriétés du système sur un ordinateur distant ou local.

2.1 Configuration pour WMI

Vérifier l'installation WMI

Sur une machine Windows hôte, WMI doit être installé pour fournir le CIM (Common Information Model) qui peut encapsuler les informations d'une manière orientée objet. Par défaut, tout système Windows plus récent que Windows XP doit avoir WMI installé et activé.

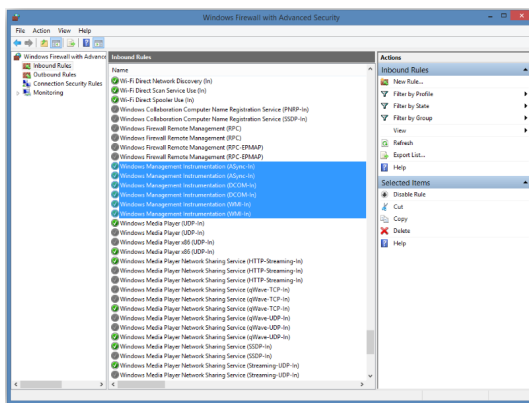
En fait, nous pouvons vérifier si WMI est installé à travers les deux étapes suivantes [4] :

- Lancez «Gestion de l'ordinateur» dans votre machine Windows et vérifiez si le service appelé «Windows Management Instrumentation» est en cours d'exécution. Sinon, démarrez ce service.
- Lancez "wbemtest" dans la console Windows. Une boîte de dialogue intitulée «Windows Management Instrumentation Test» s'affiche. Beaucoup de boutons dans cette boîte de dialogue sont actuellement désactivés, mais nous pouvons cliquer sur le bouton "Connecter ..." pour appeler une nouvelle boîte de dialogue. Puis nous cliquons simplement sur le bouton "Connecter" dans cette boîte de dialogue. Il nous ramènera à la fenêtre précédente avec tous les boutons activés.

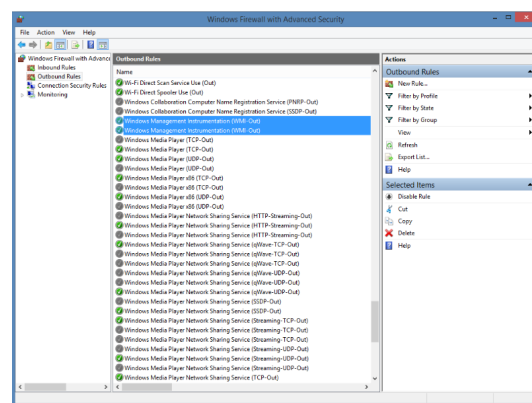
Vérifier le Pare-feu Windows

Le fait de pouvoir se connecter à l'interface WMI d'une machine est l'une des conditions préalables. Nous devons également assurer que le Pare-feu Windows permettra aux appels WMI de passer.

Dans le Pare-feu Windows, choisissez "Paramètres avancés" puis activez les règles d'accès entrantes et sortantes pour les entrées liées à WMI. Veuillez consulter les captures d'écran ci-dessous.



(a) Les règles d'accès entrantes dans le Pare-feu



(b) Les règles d'accès sortantes dans le Pare-feu

PHP extension pour WMI

Pour utiliser WMI en PHP, nous devons activer `php_com_dotnet.dll` en ajoutant une ligne au `php.ini` comme `extension=php_com_dotnet.dll` et nous démarrons le serveur.

2.2 Exemple de la programmation WMI en PHP

L'extrait de code ci-dessous présente quelques informations sur les disques d'une machine distante sur l'IP 192.168.1.4 :


```

1 <?php
2 $pc = "192.168.1.4"; //IP of the PC to manage
3
4 $WbemLocator = new COM ("WbemScripting.SWbemLocator");
5 $WbemServices = $WbemLocator->ConnectServer($pc, 'root\\cimv2', 'your account', 'your
    password');
6 $WbemServices->Security_->ImpersonationLevel = 3;
7
8 $disks = $WbemServices->ExecQuery("Select * from Win32_LogicalDisk");
9
10 foreach ($disks as $d)
11 {
12     $str=sprintf("%s (%s) %s bytes, %4.1f%% free\n", $d-
        >Name, $d->VolumeName, number_format($d->Size,0,'.',','), $d-
        >FreeSpace/$d->Size*100.0);
13
14     echo \$str;
15 }
16 ?>

```

Le résultat est comme suivant :

```

1 C: (System) 104,864,059,392 bytes, 60.4% free
2 D: (Data) 209,719,963,648 bytes, 84.3% free
3 E: (Misc) 185,521,188,864 bytes, 95.3% free

```

C'est un exemple simple, mais il définit la structure fondamentale d'un programme PHP WMI et nous pouvons connaître comment programmer WMI en PHP.

3 Les Logiciels Existants

Afin d'avoir une conception pour créer une interface et des fonctionnalités, j'ai recherché certains logiciels sur Internet qui ont déjà réalisé la fonctionnalité « WakeOnLan ». Je vais tester les logiciels et repérer les fonctionnalités qui sont intéressantes par le biais de la comparaison des logiciels, après je vais créer mon logiciel « WakeOnLan ».

Voici la liste des logiciels utilisés dans ce comparatif :

Logiciels	Version	URL
WakeOnLan	v3.9.3	http://www.dipisoft.com/articles.php?lng=fr&pg=90
WakeOnLAN	v2.11.12	http://wol.aquilatech.com/index.html
WakeOnLine		https://www.depicus.com/wake-on-lan/woli

3.1 WakeOnLan

Ce WakeOnLan avec lequel vous allez pouvoir tester l'état des machines (grâce à des Ping très rapides) que vous aurez préalablement déclarées et redémarrer celles que vous souhaitez en un seul clic. Le réveil à distance n'est pas la seule corde que cet outil a à son arc puisqu'il permet aussi de redémarrer, d'éteindre, de mettre en veille prolongée, de fermer/verrouiller la session sur une ou plusieurs machines distantes. Il faut toutefois que la configuration de la machine 'cible' le permette : le service 'serveur' doit notamment être démarré et vous (l'utilisateur qui souhaite éteindre la machine à distance) devez avoir des privilèges suffisants. L'interface de logiciel est comme l'image suivante,

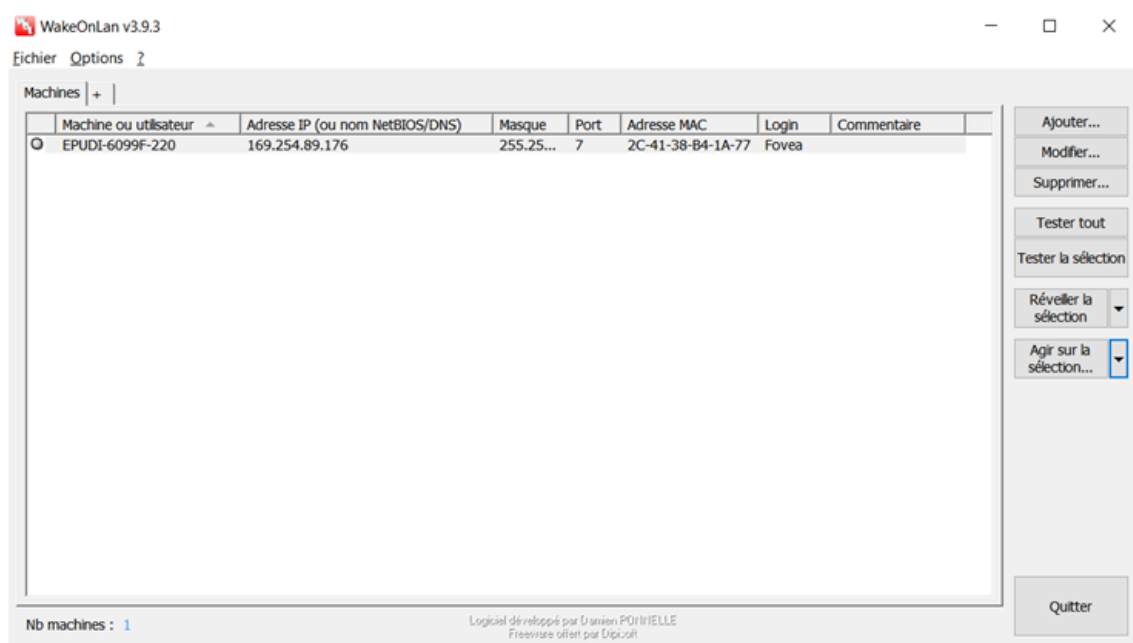
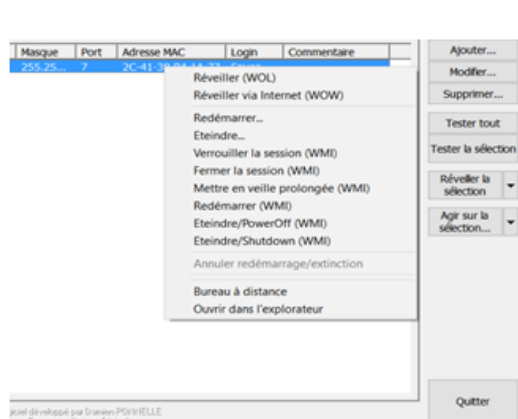


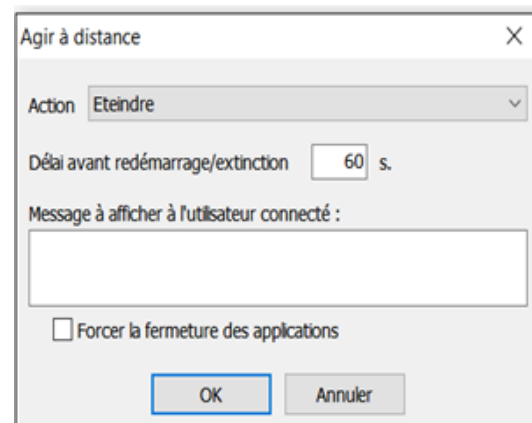
Figure 2 – L'interface WakeOnLan

Dans l'image ci-dessus, on peut voir les états de chaque machine qui est enregistré dans le logiciel, par exemple, le nom d'utilisateur, l'adresse IP, l'adresse MAC, etc. Si on fait un clic en droit sur la cible qui affiche dans le logiciel, il y a un menu pour réaliser les fonctionnalités comme l'image a affiche, par exemple, Réveiller(WOL), Réveiller via Internet(WOW), Eteindre, etc. De plus, dans le menu à droite, on peut ajouter, modifier et supprimer une machine qui existe dans le réseau privé.

Dans ce logiciel, on peut voir tous les états de chaque machine dans le réseau privé très claire. Et il permet de redémarrer et éteindre la machine à distance selon les paramètres que l'on saisit dans la fenêtre (le menu est comme l'image b affichant). On peut éteindre et rallumer l'ordinateur à distance avec un message affichant dans la cible dans le temps que l'on définit d'avance. Mais il y a des inconvénients dans le logiciel, par exemple, on ne peut pas détecter les machines dans le réseau privé automatique et on doit entrer les machines avec son IP et MAC à la main. Et ce logiciel ne peut pas exécuter les scripts à distance. Donc il ne suffit pas de ce que l'on a besoin dans mon projet.



(a) Le menu en clic à droit



(b) Le menu pour éteindre à distance

3.2 WakeOnLAN

WakeOnLAN est un logiciel qui intègre les fonctionnalités, par exemple, WOL, ping, shutdown et GUI. Il permet de faire les tâches suivantes :

- Rallumer un ordinateur à distance (wol)
- Eteindre un ordinateur à distance
- Exécuter le commande « ping » à distance afin d'afficher les états de la cible
- Rallumer un ordinateur via Internet (wow)
- Connecter au serveur distant via Remote Desktop
- Surveiller les paquets magiques pour faciliter de trouver le dépannage

3.2.1 Fenêtre principale

La fenêtre principale est divisée en trois parties. A gauche c'est les groupes et on peut activer ou désactiver par le bouton « Foldes ». Dans la partie du milieu, il affiche les états de tous les ordinateurs dans le réseau privé. On peut modifier la façon d'affichage en fonction de bouton « Views ». Les ordinateurs sont déterminés s'il est on-line ou off-line par le bouton « Pinger ». La partie à droite met les boutons que l'on utilise le plus souvent, le bouton « Start All » et le bouton « Emergency Shutdown ». Il est comme l'image suivante.

On peut obtenir un petit menu si on fait un clic-droit sur l'ordinateur dans le réseau privé.

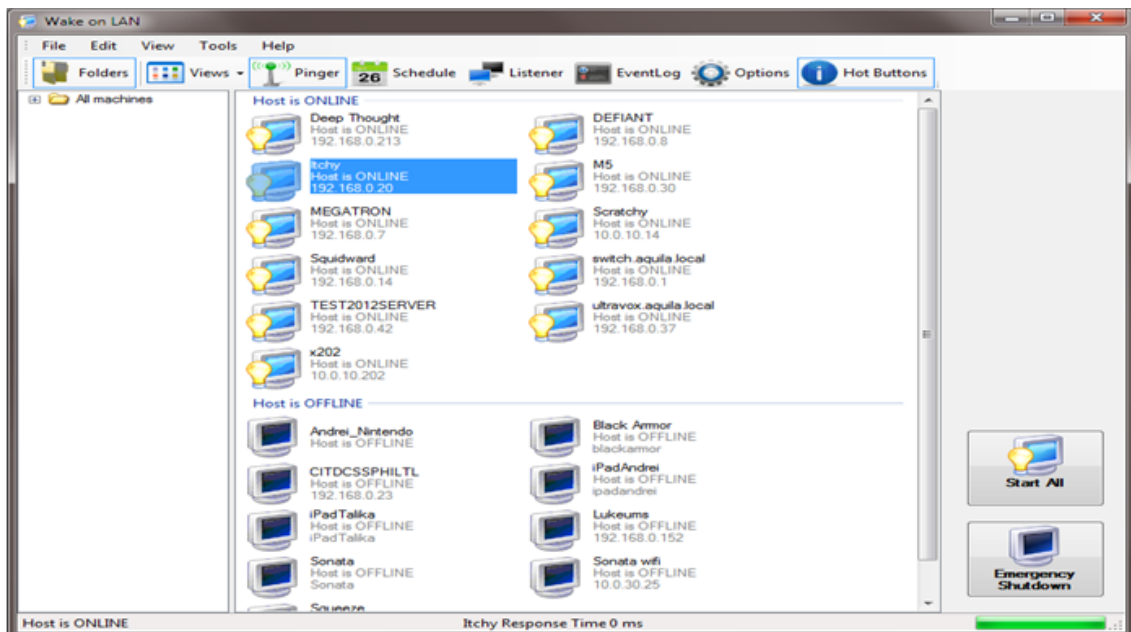
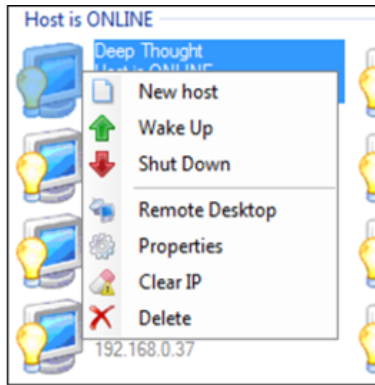


Figure 3 – La fenêtre principale dans le logiciel

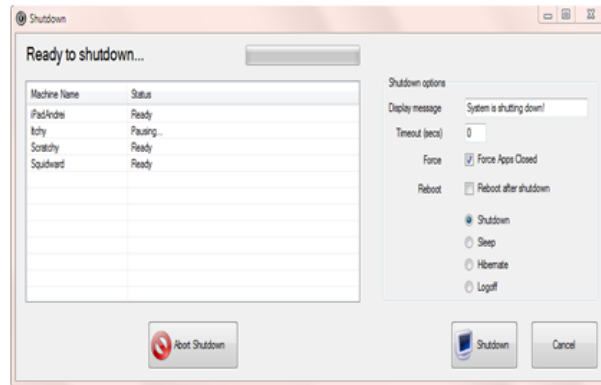
Le menu est comme l'image a suivante, on peut ajouter une nouvelle machine dans le réseau, redémarrer l'ordinateur à distance, éteindre l'ordinateur à distance etc.

Quand les icônes sont allumés dans la fenêtre principale, wol va utiliser la commande « ping » afin de vérifier si cette machine est online ou offline. Quand la machine change son état, un son de notification va lancer. Quand on choisit « wake up » dans le menu, l'ordinateur va allumer par les paquets magiques (wol).

Si on choisit la fonctionnalité « shut down » dans le menu, une fenêtre pour éteindre à distance apparaîtra comme l'image b. On peut saisir un message, le temps pour éteindre et la façon de l'extinction.



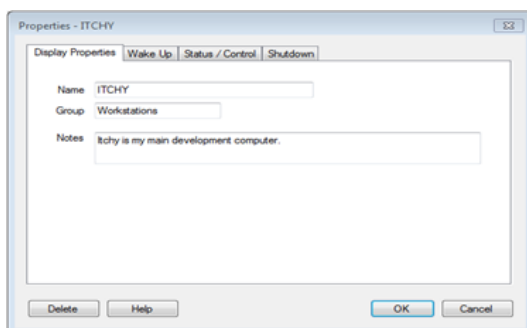
(a) Le menu de clic-droit



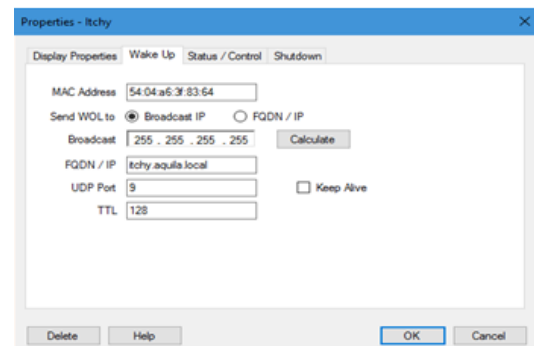
(b) L'interface pour éteindre à distance

Quand on ajoute une nouvelle machine dans le logiciel, on doit compléter les données dans les « General Host Properties », « Wake-Up Properties », « Status Properties » et « Shutdown Properties ». Ils sont comme les images suivantes, dans l'interface de « General Host Properties », on complète les informations générales pour la machine, par exemple, le nom de la machine et le groupe. Dans l'interface de « Wake-Up Properties », on complète les informations afin de réaliser la fonctionnalité de Wake-On-Lan. Dans l'interface de « Status Properties », il est pour réaliser les fonctionnalités « ping » et « remote desktop ».

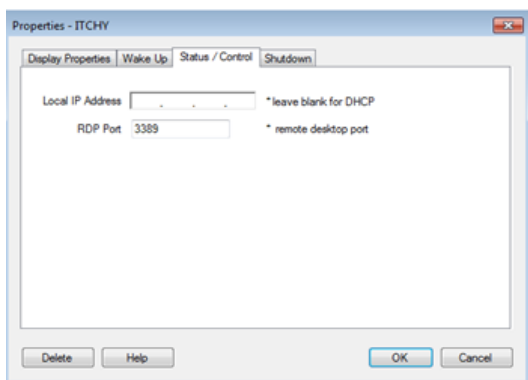
Dans l'interface de « Shutdown Properties », on doit saisir le nom d'utilisateur et le mot de passe afin d'éteindre la machine à distance.



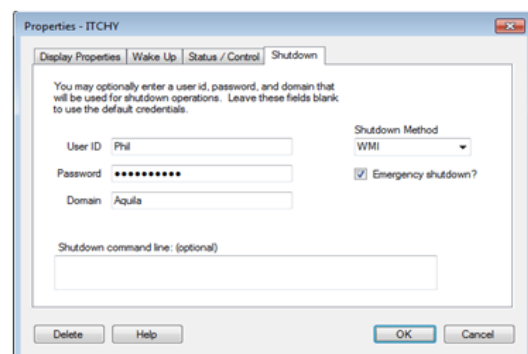
(a) General Host Properties



(b) Wake-Up Properties



(c) Status Properties



(d) Shutdown Properties

3.2.2 Les fonctionnalités spéciales

Search for machine

Dans le logiciel, on peut trouver la fonctionnalité « search » quand on clique sur Tools->Search for machines, l'interface est comme l'image suivante. On peut rechercher tous les ordinateurs dans le réseau privé à partir de l'adresse originale de l'adresse terminée en utilisant cette fonctionnalité. Dans l'écran, il affiche des informations principales de la machine, par exemple, le système que l'ordinateur utilise, l'adresse IP, l'adresse Mac, etc. Si on double-clique sur la ligne que l'on a trouvé, il affichera les informations détails de la machine qui analyse si cet ordinateur peut utiliser la technologie « Wake-On-Lan ». Parce que cette technologie a besoin des supports de la carte-mère et elle a besoin de configurer avant d'utiliser sur chaque machine. Donc on peut vérifier si on a déjà configuré chaque machine pour réaliser la technologie « Wake-On-Lan ».

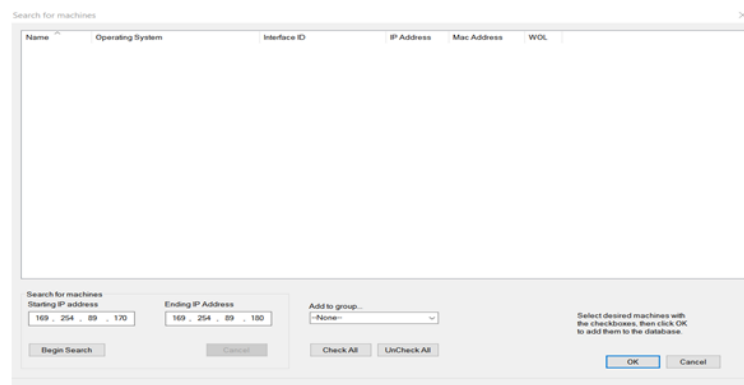


Figure 4 – L'interface search for machine

Troubleshooting

Comme on a dit précédemment, la technologie « Wake-On-Lan » a besoin des configurations de BIOS, de la carte-mère et de système d'exploitation. De plus, il n'est pas très clair si le paquet magique soit reçu et quel paquet magique la cible a reçu. Donc la technologie « troubleshooting » est développée pour analyser les paquets magiques et déterminer si le paquet magique soit reçu correctement. Pour réaliser cette fonctionnalité, c'est obligatoire d'installer ce logiciel dans l'ordinateur sur quel vous voulez analyser les paquets magiques. Après avoir installé, vous devez choisir Tools->Listen, comme l'image suivante

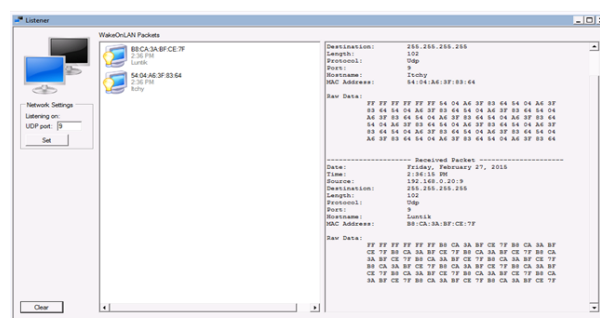


Figure 5 – L'interface troubleshooting

Scheduler

Cette fonctionnalité nous permet d'allumer et d'éteindre l'ordinateur à distance dans un intervalle de temps que l'on a paramètre. L'interface est comme l'image suivante.

On peut sélectionner les tâches qui réaliseront quand on crée une nouvelle tâche. D'abord, on



Figure 6 – L'interface scheduler

doit saisir le nom d'utilisateur qui a un privilège comme un administrateur et le mot de passe. Dans la fenêtre « Triggers », on peut choisir le temps quand la tâche va exécuter, de plus, il permet d'exécuter les tâches dans le temps fixé chaque jour ou chaque semaine. Dans la fenêtre « Action », on peut déterminer quelle tâche va exécuter, elle inclut beaucoup de tâches, par exemple, allumer l'ordinateur à distance, éteindre l'ordinateur, envoyer un message ou un email aux cibles.

Event Log

La fonctionnalité « Event Log » enregistre les traces d'allumer la machine, d'éteindre la machine et des exceptions quand le logiciel est en train d'exécuter. L'interface « Event Log » est comme l'image suivante. On peut cliquer des différents boutons afin d'afficher des informations des messages, erreurs et « warning ». on peut aussi trouver les événements que l'on veut par la fonctionnalité « Find ».

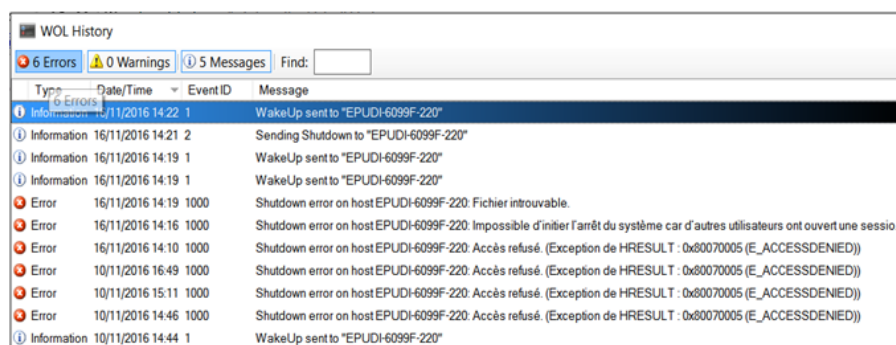


Figure 7 – L'interface Event Log

3.3 WakeOnLine

« Wake On Line » est une application web afin de réaliser la fonctionnalité « Wake-On-Lan » par Internet. Il permet de transmettre le paquet magique par Internet à la cible, c'est facile à utiliser cet outil dans le navigateur. L'interface de site web est comme l'image suivante, on peut saisir l'adresse IP et l'adresse MAC de la cible. La machine va démarrer lorsque l'on appuie le bouton « wake up ». Cet outil est créé justement pour la fonctionnalité « wake-on-lan » et il n'y a pas de l'autre fonction. Mais il me donne l'idée pour réaliser mon projet.

Figure 8 – L'interface wake on line

4 Des avantages et des limites

Dans mon projet, la plateforme web que je réaliserai a des fonctionnalités suivantes :

- Affichage les états on/ off de chaque machine dans le réseau privé.
- Allumer et éteindre les ordinateurs à distance.
- Envoyer et exécuter un script à la cible.
- Afficher les informations de la cible.

4.1 WakeOnLan

Les avantages sont comme suivants :

- L'Interface complète et clair qui peut afficher les états des cibles
- Réaliser les fonctionnalités principales dont j'ai besoin, par exemple, allumer et éteindre les ordinateurs à distance
- Permet de plusieurs fonctionnalités supplémentaires, par exemple, wake-on-wan
- Gratuit et relativement simple à installer

Mais il y a aussi des limites dans ce logiciel. Il doit être installé dans la machine avant d'utiliser. Il est gratuit mais il n'est pas opensource, donc ce n'est pas possible à modifier le logiciel afin d'adapter mon projet. Il ne peut pas envoyer et exécuter un script à la machine.

4.2 WakeOnLAN

Les avantages de logiciel « WakeOnLAN » sont suivants :

- L'interface plus complète et jolie que le logiciel « WakeOnLan ».
- Son ergonomie est complète et facile à utiliser.
- Réaliser des fonctionnalités principales dont mon projet a besoin.
- Permettre des autres fonctions, par exemple, Event Log, Scheduler.
- Gratuit et opensource, simple à installer
- Permettre d'allumer et d'éteindre les ordinateurs sous le système Linux et Windows.

Mais il y a des mêmes problèmes avec le logiciel « WakeOnLan », il n'y a pas de fonction qui peut envoyer et exécuter un script et il a besoin d'être installé dans chaque ordinateur, en particulier, on veut utiliser la fonction « Troubleshooting » pour vérifier les paquets magiques dans la cible.

4.3 WakeOnLine

Les avantages de plateforme « WakeOnLine » sont suivants :

- L'interface simple et claire.
- Il est une plateforme web, donc il n'a pas besoin d'installer et très facile à utiliser.

Les inconvénients de logiciel sont évidents, il a justement une fonction « wake-on-lan », ce n'est pas suffi comme une plateforme. Mais il me donne une conception pour réaliser mon projet.

5 Conclusion

Par la comparaison avec les logiciels que j'ai trouvés sur Internet, je connais mieux des avantages et des inconvénients dans mon projet. De plus, elle me donne une idée pour designer la plateforme web qui a des meilleures ergonomies et des interfaces. Je pense que je peux mettre en œuvre les fonctions supplémentaires à l'avenir pour améliorer et compléter mon projet. Par exemple, la fonction « troubleshooting », à mon avis, il est très utile pour analyser les paquets magiques et vérifier si la cible reçoit le paquet magique.

Mais dans mon projet, il existe des verrous technologies. Afin de réaliser la technologie, on doit mettre toutes les machines dans une protection basse. Pour entrer le système avec succès lors de démarrer à distance, il est mieux que l'on ne mette pas d'utilisateur et de mot de passe dans le système. Autrement, on va rester dans l'interface d'authentification de système lors de démarrer une machine.

4

Analyse et conception

1 Diagramme de classe

Dans la plateforme, on peut voir tous les ordinateurs dans la page index après avoir passé l'authentification. Puis, on peut ajouter, modifier et supprimer les machines dans le réseau privé en fonction des boutons dans la fenêtre principale ou le menu clic-droit. Pour chaque machine, on peut sélectionner les fonctionnalités, par exemple, allumer à distance, éteindre à distance et exécuter un script. On peut voir les informations de la cible quand on clique une ordinateur dans la fenêtre. On peut connaître les classes que j'utiliserai pour programmer et les relations entre les classes selon le diagramme de classe suivant.

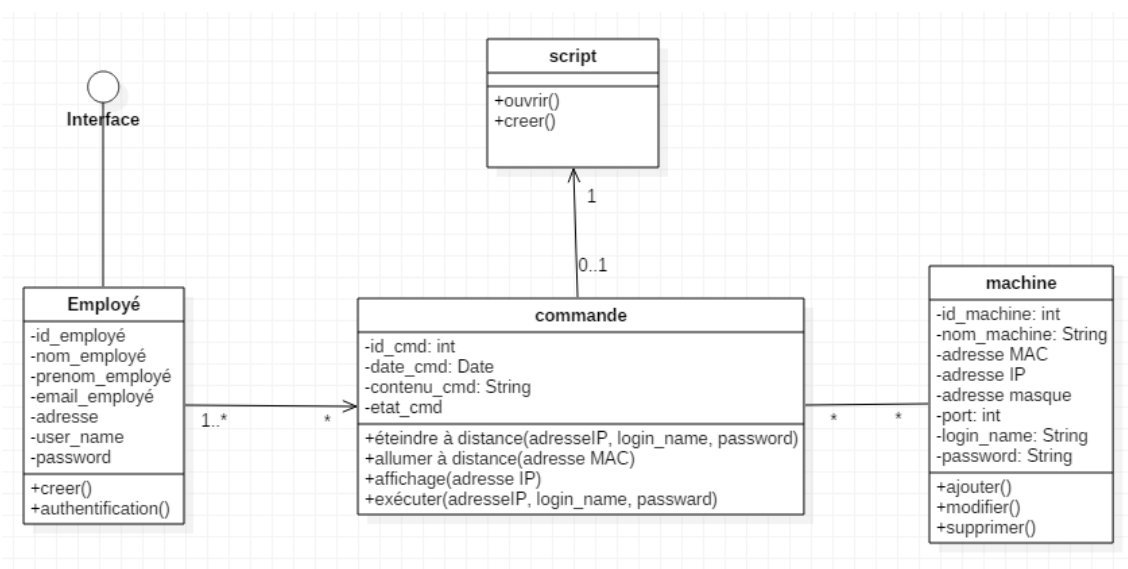


Figure 1 – Le diagramme de classe

Pierre DUCHENE

Encadrement : Barthélemy Serres

Contexte

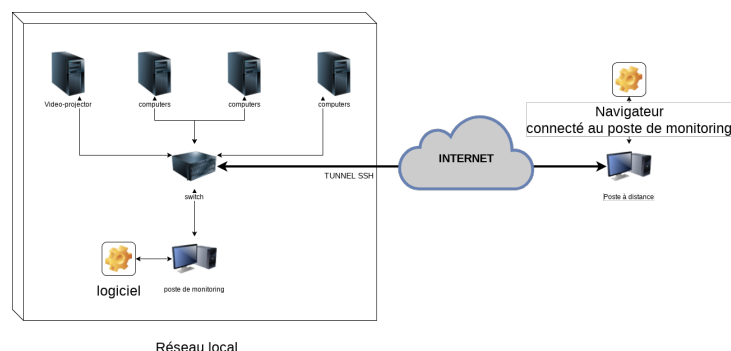
Polytech Tours travaille en collaboration avec le musée des beaux arts pour la création d'une **exposition numérique**. Mon projet de recherche et développement s'inscrit dans ce cadre, mais ne s'y limite pas.



Modélisation du réseau

Objectifs

L'objectif de ce projet est de créer une application de **monitoring** permettant de gérer entre autre le parc de machine de l'exposition.



Résultats attendus

Le logiciels devra être capable de présenter rapidement l'ensemble des statut des ordinateurs devant être surveillés.

Name	Status	Actions	Execute
Computer 1	ok	Inspect Edit	Shutdown
Computer 2	ok	Inspect Edit	Shutdown
Computer 3	Overheat	Inspect Edit	Shutdown
Computer 4	Off	Inspect Edit	Wake up
Computer 5	ok	Inspect Edit	Shutdown
Computer 6	Rebooting	Inspect Edit	
Computer 7	ok	Inspect Edit	Shutdown
Computer 8	Pending	Inspect Edit	

Mockup de l'interface admin

Contexte

Polytech Tours travaille en collaboration avec le musée des beaux arts pour la création d'une **exposition numérique**. Mon projet de recherche et développement s'inscrit dans ce cadre, mais ne s'y limite pas.

Objectifs

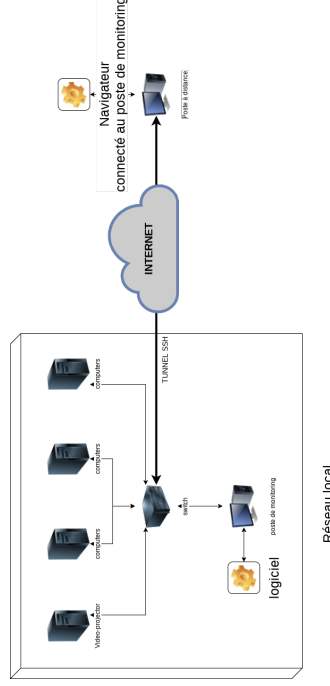
L'objectif de ce projet est de créer une application de **monitoring** permettant de gérer entre autre le parc de machine de l'exposition.

Résultats attendus

Le logiciels devra être capable de présenter rapidement l'ensemble des statut des ordinateurs devant être surveillés.



Modélisation du réseau



Mockup de l'interface admin

Résumé

Ce projet à pour but de créer un outil de monitoring capable de gérer un parc de machine hétérogène

Mots-clés

Monitoring, SNMP, Symfony, php, Musée

Abstract

This project aim to create an monitoring tool for a pool of computer

Keywords

Monitoring, SNMP, Symfony, php, Museum