



École Nationale Supérieure d'Informatique pour l'Industrie et l'Entreprise

## Rapport de stage de première année

« Mise en place d'une application de gestion des stocks, de la maintenance et du parc automobile »

*Élève* : Loïc DUBARD

*Tuteur académique à l'ENSIIE* : Ivan AUGÉ



*Entreprise* : Centre de Gestion de la Fonction Publique et Territoriale de la Réunion (Saint-Pierre, 974)

*Maître de stage* : Gwenaël BILLOUDET

Stage du lundi 01 juillet 2019 au vendredi 30 août 2019

# REMERCIEMENTS

Je tiens à remercier Monsieur Gwenaël BILLOUDET, pour m'avoir donné l'opportunité de réaliser un tel projet, et pour la confiance qu'il m'a accordée tout au long de mon stage. Projet grâce auquel j'ai pu mettre en pratique ce que j'ai acquis, qui m'a permis d'apprendre beaucoup, en informatique ainsi qu'en entreprise.

# Liste des sigles et abréviations

Voici les sigles et abréviations utilisés dans ce rapport :

- **CDG** : Centre De Gestion.
- **ERP** : Enterprise Resource Planning (en français : progiciel de gestion intégré ou PGI).
- **RADIUS** : Remote Authentication Dial-In User Service.
- **VM** : Virtual Machine (en français : machine virtuelle)
- **AD** : Active Directory
- **DNS** : Domain Name Server
- **DHCP** : Dynamic Host Configuration Protocol
- **LDAP** : Lightweight Directory Access Protocol
- **NPS** : Network Policy Server
- **IP** : adresse IP (avec IP pour Internet Protocol)
- **MVC** : Model View Controller

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Contexte . . . . .	4
1.2	Présentation de l'entreprise . . . . .	4
<b>2</b>	<b>Missions &amp; objectifs du stage</b>	<b>6</b>
2.1	Mission principale . . . . .	6
2.2	Missions secondaires . . . . .	6
<b>3</b>	<b>Travaux, Démarche générale &amp; choix techniques</b>	<b>6</b>
3.1	Odoo . . . . .	6
3.1.1	Installation du serveur Odoo . . . . .	6
3.1.2	Configuration du nom de domaine . . . . .	7
3.1.3	Intégration Active Directory . . . . .	9
3.1.4	Reverse proxy nginx . . . . .	10
3.1.5	Création d'un cronjob pour relancer le serveur s'il y a un crash . . . . .	11
3.1.6	Parc automobile . . . . .	11
3.1.7	Inventaire & demandes de matériel . . . . .	20
3.1.8	Équipements et maintenance . . . . .	25
3.2	Installation d'un serveur RADIUS . . . . .	27
3.3	Création de plugin sur Centreon . . . . .	27
<b>4</b>	<b>Conclusion</b>	<b>29</b>
<b>A</b>	<b>Annexe : Développement Durable et Responsabilité Sociale dans l'entreprise</b>	<b>i</b>
<b>B</b>	<b>Annexe : code de la partie demande de matériel du module Inventaire</b>	<b>i</b>
<b>C</b>	<b>Annexe : Installation d'un serveur RADIUS pour authentification WPA2 Entreprise sur SOPHOS XG Firewall</b>	<b>xii</b>
<b>D</b>	<b>Annexe : Code du plugin centreon pour les capteurs de températures de la salle serveur</b>	<b>xxii</b>
<b>E</b>	<b>Annexe : Code du plugin centreon pour le vCenter Server</b>	<b>xxiv</b>

# 1 Introduction

## 1.1 Contexte

Dans le cadre de ma formation en école d'ingénieur, je dois réaliser un stage en entreprise d'une durée de 8 semaines minimum, afin de mettre en pratique les connaissances que j'ai acquises et valider ma première année.

Lors de ma recherche de stage et après quelques contacts par courriel, j'ai été mis en relation avec M. BILLOUDET, Ingénieur Territorial en Systèmes d'Information, Directeur du Pôle Ressources Techniques et Directeur des Systèmes d'Information du Centre de Gestion de la Fonction Publique et Territorial de la Réunion. Après un échange par mail, celui-ci m'a fait part d'une liste de projets qu'il avait, pouvant m'intéresser, mais m'a prévenu que les démarches pour avoir un stage dans la fonction publique risquaient d'être longues. J'ai donc attendu que ses relances au service des Ressources Humaines portent leurs fruits.

Finalement, m'y étant pris suffisamment tôt, je fus accepté en tant que stagiaire, pour une période de 9 semaines.

L'objectif de ce stage fût de créer une solution informatique à destination du Service Logistique. Cette solution a pour but de répondre à des besoins spécifiques, tels que la gestion du stock et de l'inventaire et du parc automobile ainsi que des demandes de maintenances et de matériels.

## 1.2 Présentation de l'entreprise

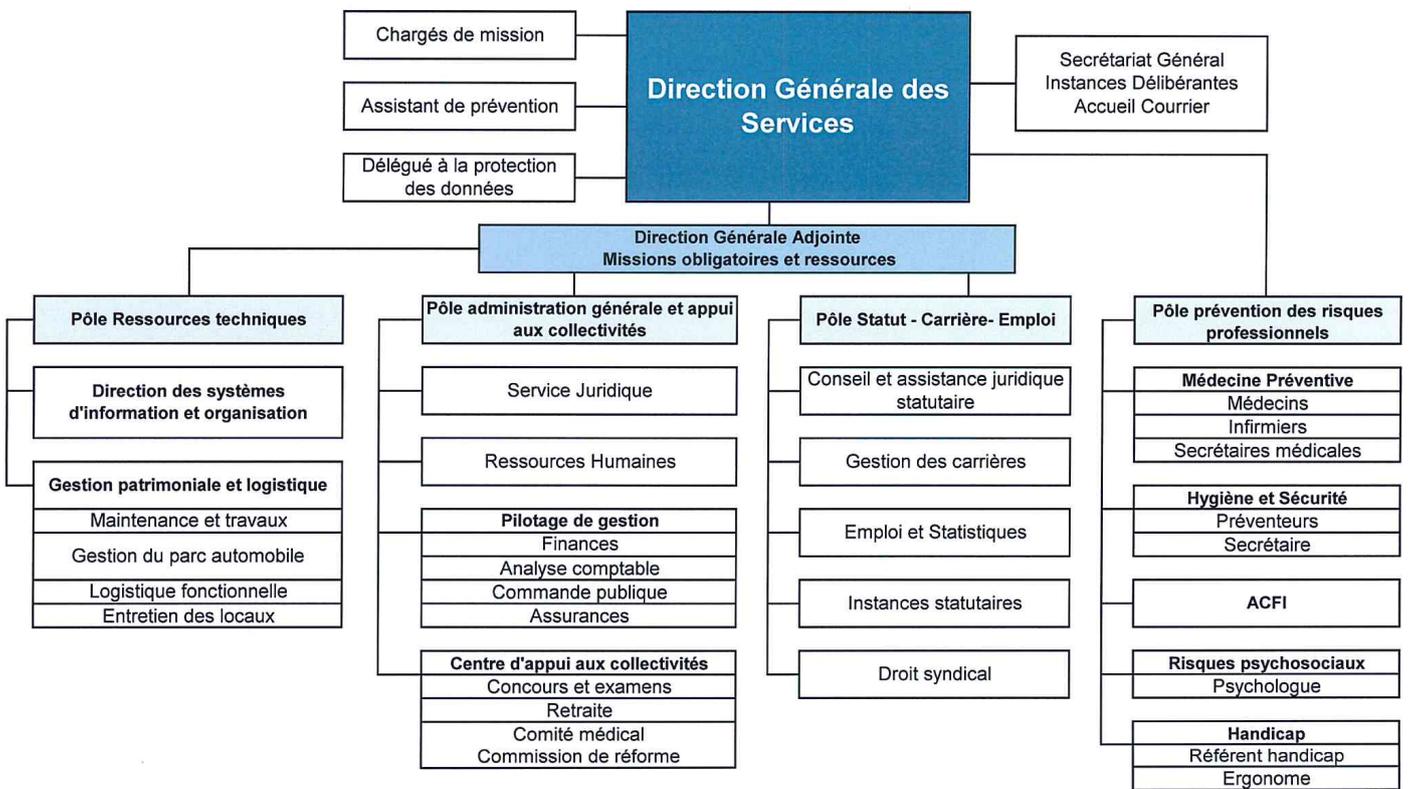
Le CDG est une collectivité publique locale à caractère administratif qui a pour but la gestion du personnel territorial.

Les missions du CDG sont les suivantes :

- l'organisation des concours et examens professionnels
- la gestion des carrières des agents des collectivités
- la gestion de la Bourse de l'Emploi
- l'organisation des Commission Administratives Paritaires, Comités techniques et Conseils de discipline
- le remboursement des décharges syndicales
- le suivi des dossiers CNRACL
- le secrétariat du Comité Médical et de la Commission de Réforme
- la prévention des risques professionnels
- la prévention des risques psycho-sociaux
- la mise à disposition d'un service de médecine professionnelle pour les agents des collectivités
- l'accompagnement des collectivité dans l'insertion et le maintient dans l'emploi de personnes en situation de handicap

La page suivante présente l'organigramme du Centre de gestion de la Réunion.

**Conseil d'administration du 29 mars 2019**  
**Affaire n°CA/19-03-29/13**



Le Président,  
**Léonus THEMOT**

## 2 Missions & objectifs du stage

### 2.1 Mission principale

Ma mission principale pour ce stage est la mise en œuvre d'un logiciel dédié au Service Logistique permettant :

- la gestion du stock et inventaire multi-sites
- la gestion des demandes, planification et interventions techniques
- la gestion du parc automobile avec demande de réservation de véhicule
- l'adaptation et développement pour les besoins fonctionnels et ergonomiques du CDG

### 2.2 Missions secondaires

En plus de ma mission principale, l'entreprise m'a confié des tâches annexes :

La mise en place d'un serveur RADIUS pour monitorer l'activité des utilisateurs du réseau wifi du Centre de Gestion et augmenter la sécurité du réseau interne. En effet la solution qui est utilisée actuellement est un accès basique par une même clé WPA constituée d'une simple phrase, pour tous les utilisateurs.

La mise à niveau du serveur de supervision sous centreon et la configuration de la surveillance de nouveaux hôtes et services.

## 3 Travaux, Démarche générale & choix techniques

### 3.1 Odoo

Le logiciel demandé étant en fait un cas spécifique d'utilisation d'un ERP ("Entreprise Ressource Planning") ou PGI en français ("Progiciel de Gestion Intégré"), mon maître de stage m'a proposé de commencer sur la base du projet complet, gratuit et open source : Odoo.

A la différence des autres ERP Open Source comme GLPI, Aria ou OpenGST, Odoo est modulaire et offre la possibilité de développer et intégrer facilement ses propres modules grâce au langage Python et XML. Il propose pour cela une variété de tutoriels et documents pour accompagner les développeurs dans cette tâche.

Les sources du projet Odoo sont accessibles sur la page : <https://github.com/odoo/odoo>

Résumé des étapes :

- Création d'une VM de développement sous Debian 9 sur l'ordinateur personnel sur VMWare Workstation
- installation des prérequis (nodejs, python3, git, vim)
- Clonage du dépôt github du projet open source Odoo et tests du programme
- Création de la même VM sur les serveurs du CDG via VMWare VSphere
- Création des accès ssh en local pour un transfert facile et sécurisé des modifications
- Installation et tests de l'application et de modules sur le réseau local
- Attribution d'un nom d'hôte à l'adresse ip fixe de la VM et utilisation d'un reverse proxy nginx pour un accès en interne à l'application ("odoo.cdg.lan")
- Visuel de l'application
- Modification du code des modules et création de nouveaux modules si besoin
- Tests et prise en compte du retour de quelques utilisateurs
- Rédaction du rapport

#### 3.1.1 Installation du serveur Odoo

La VM que j'ai installée sur vSphere possède les caractéristiques suivantes :

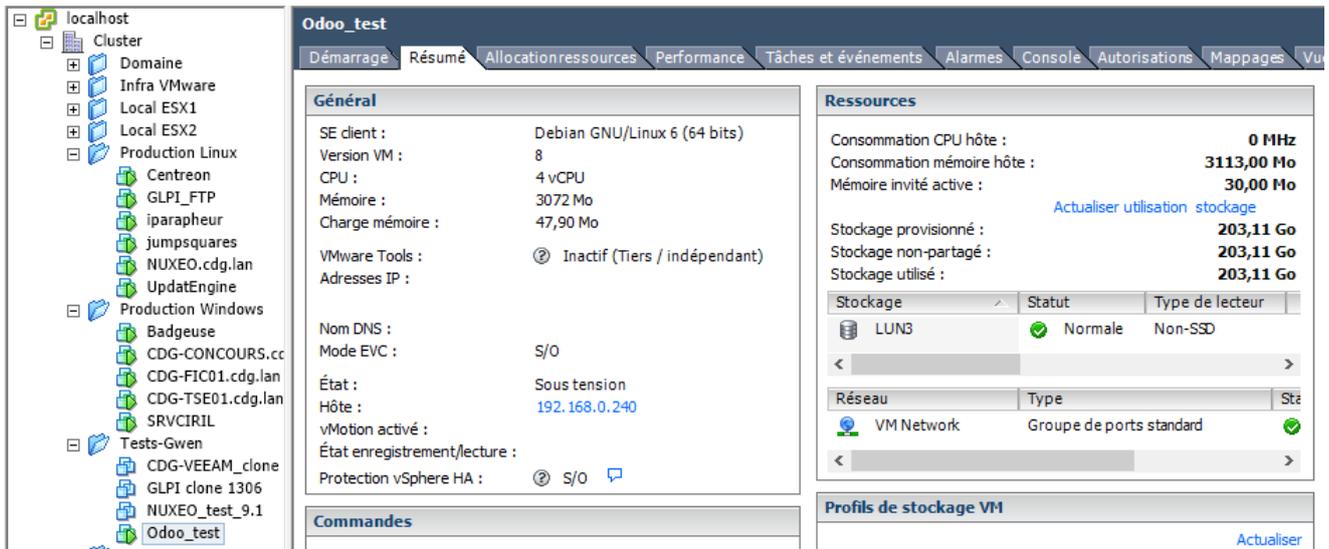


FIGURE 1 – Capture de la configuration de la VM sur VSphere

J'installe les paquets requis au bon fonctionnement de odoo :

```
$ sudo apt install sudo git vim screen python3 python3-pip curl postgresql-all libldap2-dev libsasl2-dev
$ curl -sL https://deb.nodesource.com/setup_8.x | sudo bash -
$ sudo apt install build-essential nodejs
$ sudo su - postgres -c "createusers -s odoo"
$ sudo pip3 install pyldap
$ git clone -b 12.0 --single-branch git://github.com/odoo/odoo.git
$ cd odoo/; sudo pip3 install -r requirements.txt
```

J'ai créé ensuite un script exécutable (j'ai fait un `chmod +x launch`) à l'emplacement `/home/odoo/odoo/launch` qui permet de lancer l'application dans un screen ou d'avoir accès au screen si elle est déjà lancée. (On peut échapper un screen sans tuer le processus par CTRL+A,D).

Voici le contenu de ce script :

```
#!/bin/bash
screen -S odoo-app -RDU ./odoo-bin --addons-path=addons --log-handler=werkzeug :WARNING 2>
/var/log/odoo/odoo12-server.log
```

Les modules sont situés dans le dossier `/home/odoo/odoo/addons`.

Comme premier test, j'ai téléchargé un add-on gratuit qui change le thème graphique de base et permet une meilleure configuration du visuel et des couleurs de l'application. Il faut ensuite décompresser l'archive du module dans le dossier `addons` et aller dans l'onglet application puis taper le nom du module dans la barre de recherche et enfin cliquer sur installer.

### 3.1.2 Configuration du nom de domaine

Jusque là, l'application n'était accessible que par l'ip dynamique de la VM sur le réseau. Le but étant de pouvoir y accéder toujours avec la même adresse, j'ai donc demandé à mon directeur de stage, M. Billouet de configurer le serveur DHCP en conséquence. Il m'a donc donné les accès administrateur et j'ai pu réserver une adresse fixe et un nom de domaine à ma VM :

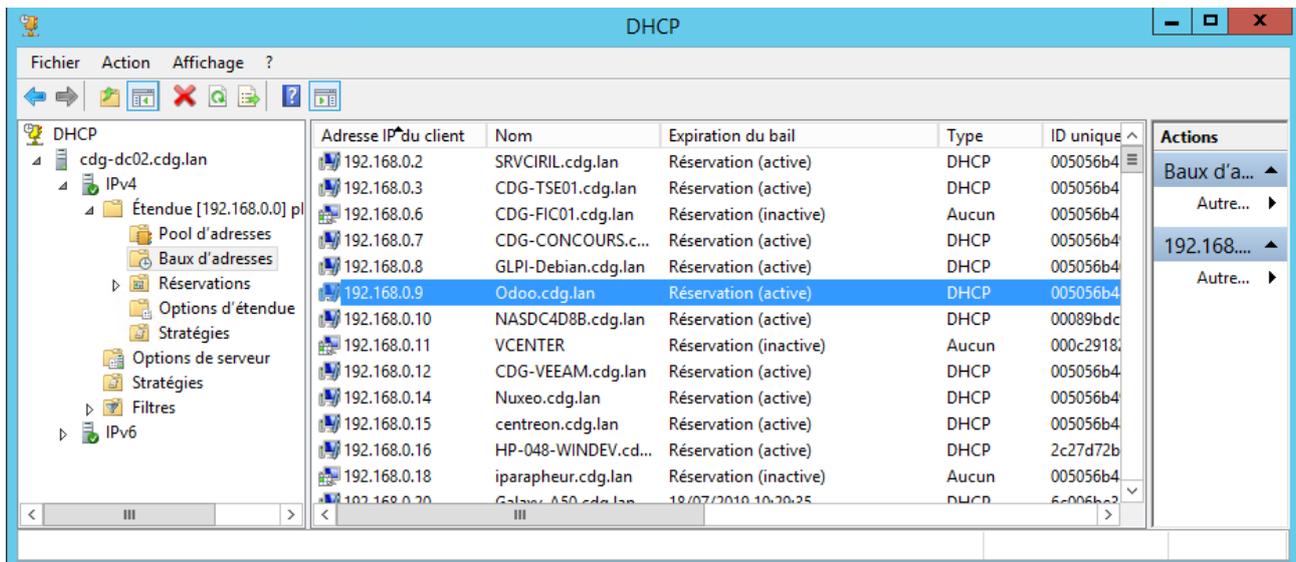


FIGURE 2 – configuration des baux d’adresse sur le serveur DHCP

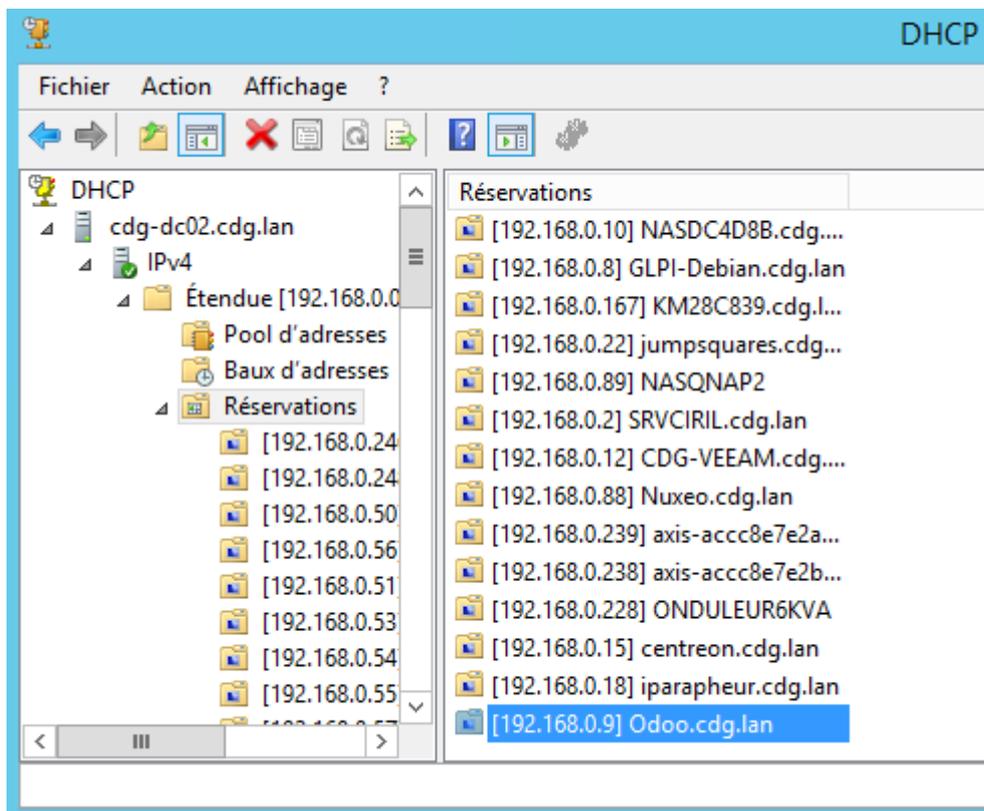


FIGURE 3 – configuration des réservations d’adresses IP sur le serveur DHCP

J’ai au préalable récupéré l’adresse mac de la VM avec la commande :

```
$ ip a
```

Après le changement d’ip, il faut relancer le client dhcp sur la VM, par un simple :

```
$ dhclient
```

L'application est maintenant accessible à l'adresse `odoo.cdg.lan:8069`.

### 3.1.3 Intégration Active Directory

Le compte d'accès à l'AD pour l'application a été créé sous cette architecture :

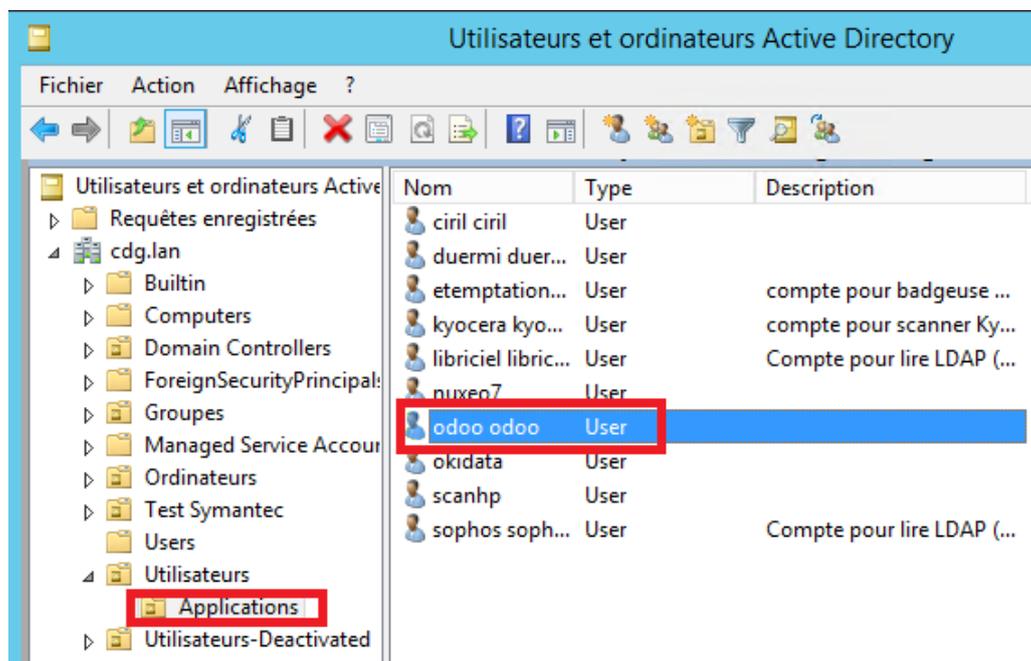


FIGURE 4 – Configuration de l'Active Directory

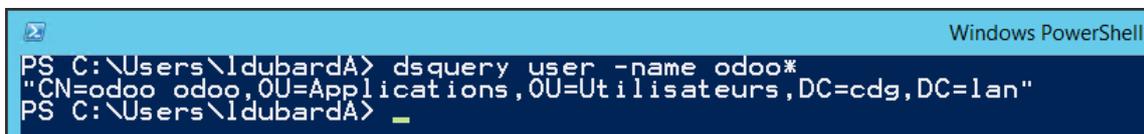


FIGURE 5 – Commande permettant de récupérer la ligne LDAP binddn pour la connexion à l'AD

J'ai donc configuré l'application de cette façon :

<b>Modifier</b>	<b>Créer</b>	<b>Action</b> ▾	1 / 1 < >
CDG Réunion, Siège			
<b>Information du serveur</b>		<b>Information de connexion</b>	
Adresse du serveur LDAP	cdg.lan	LDAP binddn	CN=odoo odoo, OU=Applications, OU=Utilisateurs, DC=cdg, DC=lan
Port du serveur LDAP	3 268	Mot de passe LDAP	██████████
Utiliser TLS	<input type="checkbox"/>		
<b>Paramètre du processus</b>		<b>Information utilisateur</b>	
Base LDAP	DC=cdg, DC=lan	Créer un utilisateur	<input checked="" type="checkbox"/>
Filtre LDAP	sAMAccountName=%s	Utilisateur modèle	template
Séquence	10		

FIGURE 6 – Configuration du client LDAP sur l'application Odoo

### 3.1.4 Reverse proxy nginx

NGINX est un serveur HTTP et reverse proxy gratuit et open-source ainsi qu'un serveur IMAP/POP3.

Il est connu pour ses grandes performances, sa stabilité, la richesse de ses fonctionnalités, sa simplicité de configuration, et sa basse consommation de ressources.

Je l'ai donc utilisé ici comme reverse proxy afin d'accéder à l'application en passant par le port http de base (port 80) au lieu de passer par le port 8069.

J'ai commencé par l'installer :

```
$ sudo apt install nginx
```

Ensuite j'ai modifié le fichier principal de configuration :

```
$ sudo vi /etc/nginx/sites-enabled/default
```

je rajoute la nouvelle configuration dans location :

```
location / {
    proxy_pass http://127.0.0.1:8069;
    proxy_redirect off;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
}
```

et j'ai commenté le code de location déjà existant.

Enfin j'ai redémarré nginx :

```
$ sudo systemctl restart nginx
```

L'application est maintenant accessible à l'adresse `odoo.cdg.lan`.

### 3.1.5 Création d'un cronjob pour relancer le serveur s'il y a un crash

**cron** est un programme qui permet aux utilisateurs des systèmes Unix d'exécuter automatiquement des scripts, des commandes ou des logiciels à une date et une heure spécifiée à l'avance, ou selon un cycle défini à l'avance.

Dans le cas où le serveur subit un crash, j'ai créé une routine cron (ou *cronjob*) qui lance toutes les minutes un script qui vérifie si le processus n'est plus présent et le relance le cas échéant de cette manière :

```
$ crontab -e
```

Puis avec l'éditeur de texte par défaut qui s'ouvre je rajoute à la fin du fichier :

```
* * * * * /home/odoo/cron_odoo.sh
```

Je sauvegarde et je quitte, ce qui recharge la configuration de crontab automatiquement. J'écris maintenant le fameux script `/home/odoo/cron_odoo.sh`

```
$ cat << 'EOF' >/home/odoo/cron_odoo.sh
#!/bin/bash
if! [[ $(screen -ls | grep "odoo_app") ]]; then
cd /home/odoo/odoo
script /dev/null
screen -S odoo_app -d -m ./odoo-bin --addons-path=addons --log-handler=werkzeug :WARNING 2>
/var/log/odoo/odoo12-server.log
fi
EOF
$ chmod +x cron_odoo.sh
```

### 3.1.6 Parc automobile

Voici le diagramme UML simplifié du Modèle logique de données recherché :

## Extension du module en rajoutant un onglet calendrier de réservation fonctionnel

C'est dans cette partie que cela commence à devenir compliqué, puisque pour modifier les sources, il faut comprendre comment est organisé le code.

Le code des modules est organisé selon le modèle MVC (Model View Controller) et l'on retrouve toujours cette architecture :

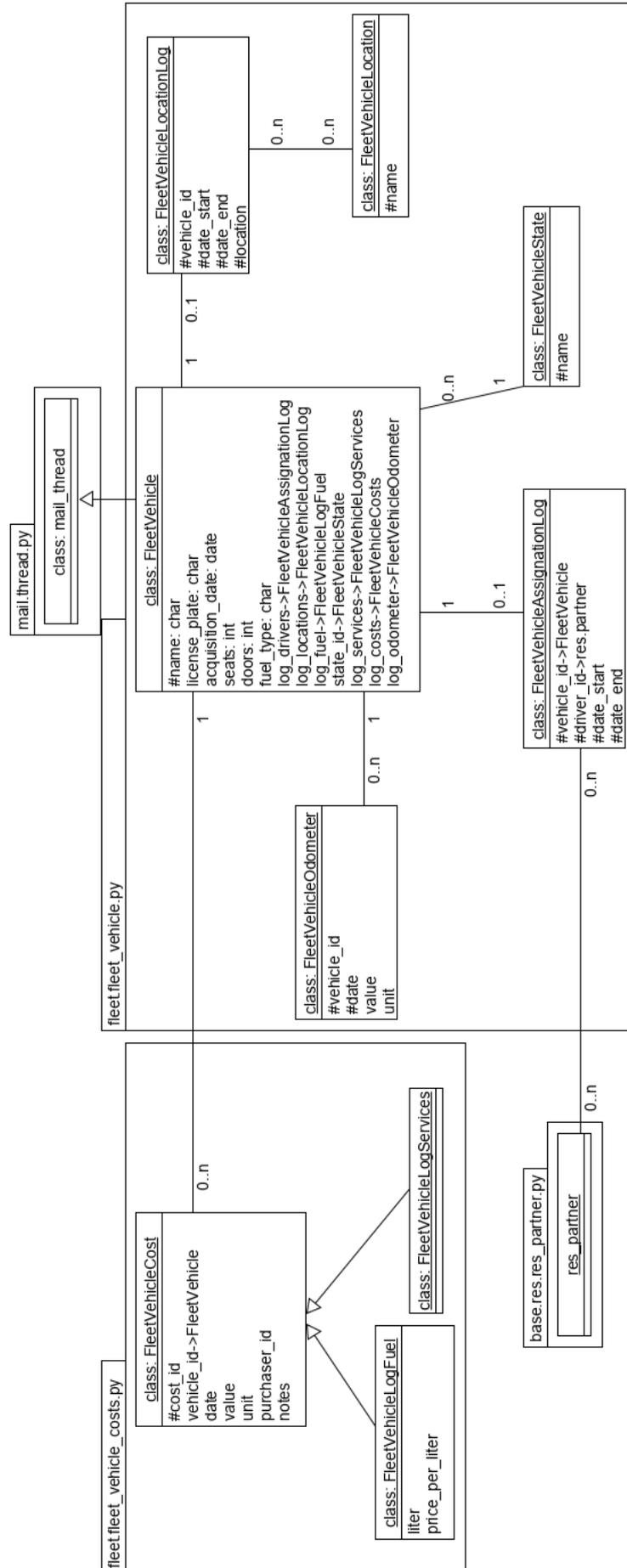


FIGURE 7 – MLD allégé pour le module Parc auto.

```

+ newaddon/
|   __init__.py
|   __manifest__.py
|   + data/
|       |       newaddon_demo.xml
|       |       newaddon_data.xml
|       |       ...                               #d'autres fichiers .xml
|   + i18n/
|       |       fr.po
|       |       ...                               #d'autres fichiers de traduction en .po
|   + models/
|       |       __init__.py
|       |       newaddon_functionalite1.py
|       |       newaddon_functionalite2.py
|       |       + __pycache__/                    #cache de python
|       |           |       __init__.cpython-35.pyc
|       |           |       newaddon_functionalite1.cpython-35.pyc
|       |           |       newaddon_functionalite2.cpython-35.pyc
|       |           |       ...                     #autres fichiers .cpython-35.pyc
|       |           |       ...                     #autres fichiers .py
|   + views/
|       |       newaddon_menu1_views.xml
|       |       newaddon_menu2_views.xml
|       |       ...                               #autres fichiers .xml
|   + security/
|       |       newaddon_security.xml             #règles d'enregistrements
|       |       ir.model.access.csv              #droits d'accès
|   + static/
|       |       + description/
|       |           |       icon.png
|       |           |       icon.svg
|       |       + img/
|       |           |       ...                     #images .png

```

Je m'intéresse donc d'abord à la vue et au fichier `addons/fleet/views/fleet_vehicle_views.xml` dans lequel je crée un nouveau menu et la vue associée.

Puis je modifie les droits d'accès en lecture/écriture/création/suppression pour les utilisateurs et gestionnaires dans le fichier : `addons/fleet/security/ir.model.access.csv`.

Enfin je modifie plus précisément ces droits par des règles d'enregistrements dans l'application (Les flèches désignent les lignes rajoutées).

Application		Parc automobile		Nom		Utilisateur	
Groupe de partage		<input type="checkbox"/>					
Utilisateurs		Hérité		Menus		Vues	
Droits d'accès		Règles sur les enregistrements		Notes			
Nom	Objet	Domaine	Appliquer pour la lecture	Appliquer à l'écriture	Appliquer à la création	Appliquer à la suppression	
User can only see his/her vehicle's odometer	Suivi du kilométrage pour le véhicule	[("vehicle_id.driver_id",'=','user.partner_id.id)]	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
User can only see his/her vehicle's fuel log	Suivi de la consommation pour les véhicules	[("cost_id.vehicle_id.driver_id",'=','user.partner_id.id)]	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
User can only see "Disponible in the list of status	Statut du véhicule	[("name",'=','Disponible')]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	←
User can only see his/her vehicle's services	Services associés aux véhicules	[("cost_id.vehicle_id.driver_id",'=','user.partner_id.id)]	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
User can only see his/her own assignation logs	Historique de conducteurs sur un véhicule	[("driver_id",'=','user.partner_id.id)]	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	←
User can only see his/her costs	Coût associé à un véhicule	[("vehicle_id.driver_id",'=','user.partner_id.id)]	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
User can only see his/her vehicle state restrict	Véhicule	[("state_id.name",'=','Disponible')]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	←
User can only see his/her contracts	Information du contrat pour un véhicule	[("cost_id.vehicle_id.driver_id",'=','user.partner_id.id)]	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

FIGURE 8 – Vue dans l'application de la configuration des règles d'enregistrements pour les **utilisateurs** du parc auto

Application		Parc automobile		Nom		Gestionnaire	
Groupe de partage		<input type="checkbox"/>					
Utilisateurs		Hérité		Menus		Vues	
Droits d'accès		Règles sur les enregistrements		Notes			
Nom	Objet	Domaine	Appliquer pour la lecture	Appliquer à l'écriture	Appliquer à la création	Appliquer à la suppression	
Manager has all right on location history	location history on a vehicle		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Manager has all right on the list of possible location	Vehicle Location		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Manager has all rights on vehicle's vehicle's odometer	Suivi du kilométrage pour le véhicule		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Manager has all rights on vehicle's fuel log	Suivi de la consommation pour les véhicules		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Manager can see all status	Statut du véhicule		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	←
Manager has all rights on vehicle's services	Services associés aux véhicules		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
manager has all right on list of drivers	Historique de conducteurs sur un véhicule		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	←
Manager has all rights on vehicle's costs	Coût associé à un véhicule		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Manager has all rights on vehicle	Véhicule		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Manager has all rights on vehicle's contracts	Information du contrat pour un véhicule		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

FIGURE 9 – Vue dans l'application de la configuration des règles d'enregistrements pour les **gestionnaires** du parc auto

Il faut bien faire attention à mettre le bon filtre de domaine pour l'utilisateur et aucun filtre de domaine pour le gestionnaire.

Enfin dans un soucis d'ergonomie je déplace les menus et sous menus inutiles pour l'utilisateur mais utile pour les gestionnaires dans un nouvel onglet "Gestion" uniquement accessible aux gestionnaires du parc auto. Ces sous menus sont les suivants :

- Suivi des interventions
- Suivi du kilométrage des véhicules
- Suivi du niveau d'essence.

Un simple utilisateur du parc automobile n'a maintenant accès qu'à la liste des véhicules avec le statut disponible et au calendrier de réservation.

Il ne peut que créer un nouvel enregistrement dans le calendrier, pour un véhicule dont le statut est disponible uniquement et à son nom uniquement. Il ne pourra plus supprimer ou modifier cet enregistrement.

## Création d'un historique de lieux

Je commence par créer un onglet de modification des différents Lieux possibles que je place dans le menu Configuration.

→ dans `fleet/models/fleet_vehicle.py` :

```
class FleetVehicleLocation(models.Model):
    _name = 'fleet.vehicle.location'
    _order = 'name asc'
    _description = 'Vehicle Location'

    name = fields.Char(required=True, translate=True)

    _sql_constraints = [('fleet_location_name_unique', 'unique(name)', 'Location name
        ↪ already exists')]
```

→ dans `fleet/views/fleet_vehicle_views.xml` :

```
<record id="fleet_vehicle_location_action" model="ir.actions.act_window">
  <field name="name">Lieux</field>
  <field name="res_model">fleet.vehicle.location</field>
  <field name="view_type">form</field>
  <field name="view_mode">form,tree</field>
  <field name="help" type="html">
    <p class="o_view_nocontent_smiling_face">
      view the list of locations
    </p>
  </field>
</record>

<menuitem id="fleet_vehicle_location_menu" parent="fleet_configuration" action="
  ↪ fleet_vehicle_location_action" sequence="89"/>
```

Ensuite je crée la classe historique des lieux pour un véhicule :

→ dans `fleet/models/fleet_vehicle.py` :

```
class FleetVehicleLocationLog(models.Model):
    _name = "fleet.vehicle.location.log"
    _description = "location history on a vehicle"
    _order = "date_start"

    vehicle_id = fields.Many2one('fleet.vehicle', string="Vehicle", required=True)
    location = fields.Many2one('fleet.vehicle.location', string="Location", required=True
        ↪ )
    date_start = fields.Datetime(string="Start Date")
    date_end = fields.Datetime(string="End Date")
```

→ dans `fleet/views/fleet_vehicle_views.xml` :

```

<record id="fleet_vehicle_location_log_view_list" model="ir.ui.view">
  <field name="name">fleet.vehicle.location.log.tree</field>
  <field name="model">fleet.vehicle.location.log</field>
  <field name="arch" type="xml">
    <tree string="Locations_Logs" editable="bottom">
      <field name="vehicle_id"/>
      <field name="location"/>
      <field name="date_start"/>
      <field name="date_end"/>
    </tree>
  </field>
</record>

<record id="fleet_vehicle_location_log_calendar" model="ir.ui.view">
  <field name="name">fleet.vehicle.location.log.calendar</field>
  <field name="model">fleet.vehicle.location.log</field>
  <field name="arch" type="xml">
    <calendar date_start="date_start" date_stop="date_end" color="location">
      <field name="location"/>
      <field name="vehicle_id"/>
    </calendar>
  </field>
</record>
<record id="fleet_vehicle_location_log_action" model="ir.actions.act_window">
  <field name="name">Calendrier des lieux</field>
  <field name="res_model">fleet.vehicle.location.log</field>
  <field name="view_type">form</field>
  <field name="view_mode">calendar,form</field>
  <field name="help" type="html">
    <p class="o_view_nocontent_smiling_face">
      view the location calendar
    </p>
  </field>
</record>

<menuitem id="fleet_vehicle_location_log_menu" parent="fleet_vehicles" action="
  ↪ fleet_vehicle_location_log_action" sequence="89"/>

```

Ensuite dans la classe FleetVehicle je modifie le champ location de base (qui était un simple texte) et je rajoute les méthodes de création, d'écriture et d'ouverture de cet historique à l'exemple de l'historique des conducteurs.

Et je rajoute un bouton sur la fiche de description d'un vehicule dans fleet/views/fleet\_vehicle\_views.xml.

Enfin je n'oublie de rajouter les droits d'accès à cette nouvelle fonctionnalité pour les groupes utilisateurs et gestionnaires .

## Envoi automatique de mail lors de la réservation d'un véhicule

J'ai dû d'abord installer le module "Règles d'actions automatisées" puis j'ai créé un nouveau modèle de courriel :

The image shows the Odoo 12.0 Configuration interface. The top navigation bar includes 'Configuration', 'Tableau de bord', 'Utilisateurs et sociétés', 'Traductions', 'Paramètres Généraux', and 'Technique' (highlighted with a red box and the number 3). The main content area is divided into several sections:

- 6 Applications installées**: Includes a 'Parcourir les applications' button and links for 'Magasin d'applications', 'Achats intégrés', and 'Magasin de thèmes'.
- 6 Utilisateurs actifs**: Includes a 'Gérer les droits d'accès' button and an 'Inviter de nouveaux utilisateurs' section with an email input field and an 'Inviter' button. Below it, 'Invitations en attente' lists 'Idubarda' and 'template'.
- Partagez l'engouement**: Includes social media icons for Twitter, Facebook, and LinkedIn.
- Traductions**: Includes a 'Charger une traduction' button.
- Centre de Gestion de la Fonction Publique Territoriale de la Réunion**: Includes a 'Configurer' button.
- Odoo 12.0 (Community Edition)**: Includes a red box around the 'Activer le mode développeur' button (with the number 2) and a 'Désactiver le mode développeur' button.

A red box with the number 1 highlights the gear icon in the left sidebar. A second red box with the number 2 highlights the 'Activer le mode développeur' button.

Below the main configuration page, a dropdown menu is shown for the 'Technique' section. The menu items are: Courriel, Sous-types, Valeurs du suivi, Messages, Courriels, Types d'activités, Modération du canal, Blacklist Mass Mail, Serveurs de courriels entrant, Serveurs de courriel sortant, **Modèles** (highlighted with a red box), Lettres, and Abonnés.

The bottom part of the image shows the 'Gestion des réservations du Parc Auto...' configuration page. It includes a 'S'applique à' dropdown set to 'Historique de conducteurs sur un véhicule', tabs for 'Contenu', 'Configuration de courriels', 'Paramètres avancés', and 'Générateur de placeholder dynamique', and a 'Sujet [ODOO] Gestion des réservations du Parc Automobile'.

The email template content is as follows:

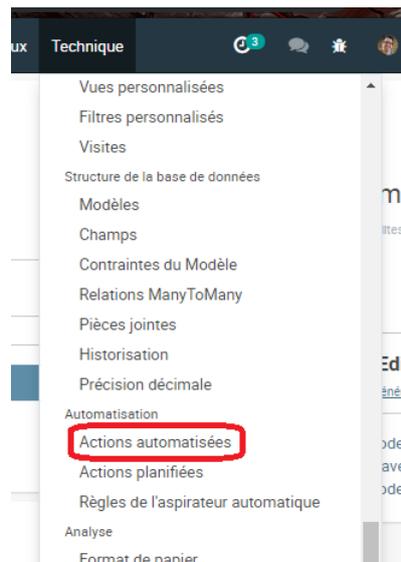
Cher(e) Gestionnaire du parc auto,

Une nouvelle réservation de véhicule à été faite pour \${object.driver\_id.name} sur le véhicule : \${object.vehicle\_id.name} du \${object.date\_start} au \${object.date\_end}.

Ceci est un email généré automatiquement,  
merci de ne pas y répondre.

FIGURE 10 – Procédure pour configurer les modèles de mail dans l'application

ensuite j'ai créé l'action automatique :



Actions automatisées / Notifier les gestionnaires du parc auto de la création d'une nouvelle réservation de véhicule

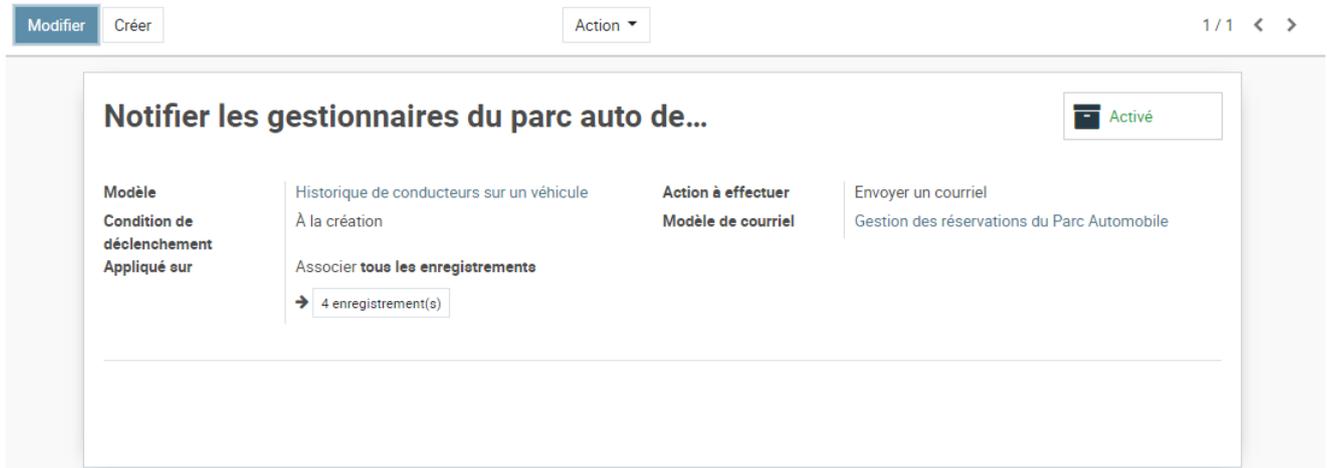


FIGURE 11 – Procédure pour configurer l'envoi de mail automatique dans l'application

Enfin j'ai donné le droit d'accès à la configuration du mail pour les gestionnaires du parc auto. Voici le rendu final de vue d'ensemble (pour les gestionnaires) :

**Parc automobile** Véhicules Gestion Analyse Configuration 3 Admin (test1)

Véhicules Statut x Recherche...

Créer Importer Filtres Regrouper par Favoris

Dispo / St-Pierre	Dispo / Etang-Salé	Dispo / Ste-Clotilde	Remisage domicile
6	2	1	
<ul style="list-style-type: none"> <li>BA-059-EJ : Renault/CLIO Logistique</li> <li>BM-731-DL : Renault/KANGOO Logistique</li> <li>CN-303-FS : Renault/CLIO Logistique</li> <li>CW-255-NQ : Peugeot/208 Logistique</li> <li>CW-703-MY : Renault/KANGOO Logistique</li> <li>EQ-240-JW : Renault/CLIO DR CALUZIO</li> </ul>	<ul style="list-style-type: none"> <li>DF-008-HD : Renault/TWINGO Pole Santé</li> <li>DF-153-HD : Renault/TWINGO Pole Santé</li> </ul>	<ul style="list-style-type: none"> <li>CN-337-FS : Renault/TWINGO St Clotilde</li> </ul>	<ul style="list-style-type: none"> <li>BA-674-FS : Renault/CLIO DR AZEMA</li> <li>CN-332-FS : Renault/TWINGO DR ZIEGLE</li> <li>CW-171-NQ : Peugeot/208 DR GRIZEAU</li> <li>CW-229-NQ : Peugeot/208 DR BADIA</li> <li>CW-268-NQ : Peugeot/208 DR ROLLIN</li> <li>DM-885-PH : Seat/LEON Président</li> <li>EQ-037-JW : Renault/CLIO Ivan HOAREAU</li> <li>EQ-170-JW : Renault/CLIO Guy-Noel SEVRIN</li> </ul>

FIGURE 12 – Vue finale du module Parc Automobile

Et voici la vue finale de calendrier de demande de réservation :

**Parc automobile** Véhicules Gestion Analyse Configuration 3 Admin (test1)

Calendrier des réservations (août 2019) Recherche...

← Aujourd'hui → Jour Semaine Mois Filtres Favoris

W	lun.	mar.	mer.	jeu.	ven.	sam.	dim.
31		29	30	31	1	2	3
		Admin Renault/CLIO/BA-059-EJ	Admin Renault/CLIO/BA-059-EJ				
32	5	6	7	8	9	10	11
33	12	13	14	15	16	17	18
		Admin Renault/CLIO/BA-059-EJ	Admin Renault/CLIO/BA-059-EJ				
34	19	20	21	22	23	24	25

août 2019

lun.	mar.	mer.	jeu.	ven.	sam.	dim.
1	2	3	4			
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

**Conducteur**

Admin

FIGURE 13 – Onglet du calendrier de réservation de véhicules

## Interdire le chevauchement de deux réservations d'un même véhicule

Avant mon apport, il était possible de créer et de modifier une réservation pour la faire chevaucher sur une autre réservation du même véhicule. J'ai donc dû réécrire entièrement les méthodes `create()` et `write()` de la classe `FleetVehicleAssignmentLog()` :

```
class FleetVehicleAssignmentLog(models.Model):
    _name = "fleet.vehicle.assignment.log"
    _inherit = ['mail.thread', 'mail.activity.mixin']
    _description = "Drivers history on a vehicle"
    _order = "date_start"

    vehicle_id = fields.Many2one('fleet.vehicle', string="Vehicule", required=True)
    driver_id = fields.Many2one('res.partner', string="Driver", required=True)
    date_start = fields.Datetime(string="Start Date", required=True)
    date_end = fields.Datetime(string="End Date", required=True)

    @api.model
    def create(self, vals):
        can_create=True
        print("[TESTING] values to add in assignation log :",vals)
        for record in self.env['fleet.vehicle.assignment.log'].search([( 'vehicle_id', '=
        ↪ ',vals[ 'vehicle_id' ])]):
            print("[TESTING] test de la valeur de record.datestart :",record.date_start)
            if record.date_start <= datetime.strptime(vals[ 'date_start' ], '%Y-%m-%d %H:%M
            ↪ :%S'):
                # d1<d2 equivaut a d1 plus vieux que d2
                if record.date_end >= datetime.strptime(vals[ 'date_start' ], '%Y-%m-%d %H
                ↪ :%M:%S'):
                    can_create = False ; break
            elif record.date_start <= datetime.strptime(vals[ 'date_end' ], '%Y-%m-%d %H:%M
            ↪ :%S'):
                can_create = False ; break
        if can_create:
            res = super(FleetVehicleAssignmentLog, self).create(vals)
            return res
        else:
            raise exceptions.ValidationError('Veuillez ne pas se faire chevaucher les
            ↪ reservations pour un meme vehicule !')

    @api.multi
    def write(self, vals):
        can_write=True
        print("[TESTING] ",self,self.vehicle_id)
        for record in self.env['fleet.vehicle.assignment.log'].search([( 'vehicle_id', '=
        ↪ ',self.vehicle_id.id)]):
            #on teste uniquement si c'est le même véhicule
            print("[TESTING] record_id:",record.id,"self_id:",self.id)
            if record.id != self.id: # car il ne faut pas interdire en testant la ré
            ↪ servation sur elle même !
                if 'date_start' in vals and record.date_start <= datetime.strptime(vals[
                ↪ 'date_start' ], '%Y-%m-%d %H:%M:%S'):
                    if record.date_end >= datetime.strptime(vals[ 'date_start' ], '%Y-%m-%d
                    ↪ %H:%M:%S'):
                        can_write = False ; break
                elif 'date_end' in vals and record.date_start <= datetime.strptime(vals[
                ↪ 'date_end' ], '%Y-%m-%d %H:%M:%S'):
                    can_write = False ; break
        if can_write:
            res = super(FleetVehicleAssignmentLog, self).write(vals)
            return res
        else:
            raise exceptions.ValidationError('Veuillez ne pas se faire chevaucher les
            ↪ reservations pour un meme vehicule !')
```

### 3.1.7 Inventaire & demandes de matériel

L'idée est ici de recoder le module "Inventaire" déjà proposé par Odoo pour le simplifier au maximum et intégrer à la manière du module "Maintenance", un onglet utilisateur "Demande de matériel".

Le code final, disponible dans l'annexe B donne la vue suivante :

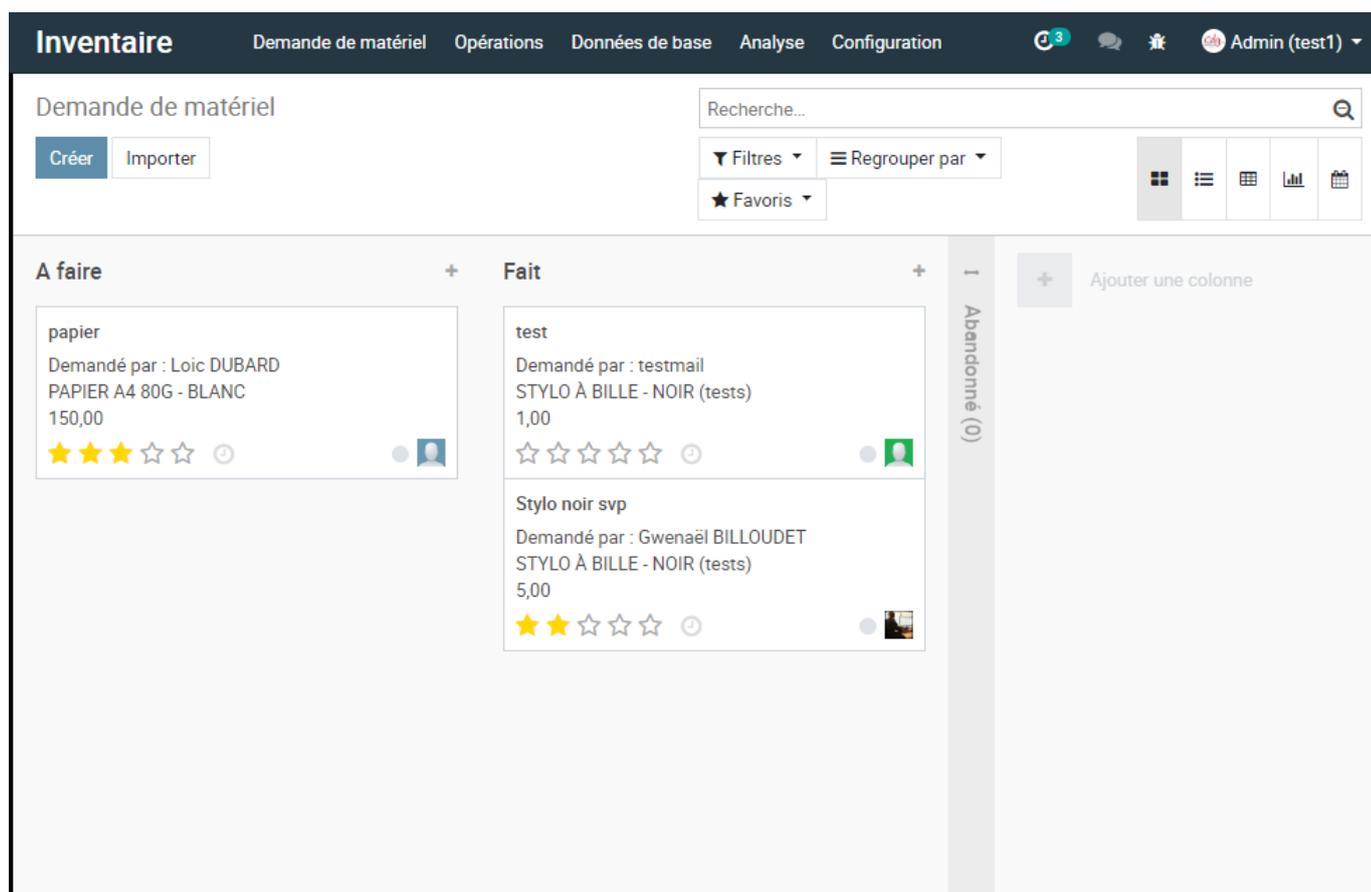


FIGURE 14 – Vue de la page demandes de matériel

Le fait que l'on voit absolument toutes les demandes de matériels faites depuis le début de la mise en production risque de nuire au côté simple et pratique de l'application. J'ai donc décidé de rajouter un filtre de base, pour n'afficher que les demandes *non archivées*.

Pour ce faire, j'ai activé les options développeurs, rajouter un filtre dans la *vue de recherche*, puis modifier l'action pour rajouter la valeur par défaut de ce filtre :



FIGURE 15 – accès vers la configuration de la vue de recherche et l'action en mode développeur

✦ Modifier la vue de recherche
✕

**Nom de Vue**

**Type de Vue**

**Modèle**

**Séquence**

**Actif**

**Champ enfant**

**Vue héritée**

**Mode héritage de Vue**

**Modèle de Données**

**ID Externe**

Architecture

Droits d'accès

Vues héritées

[Éditer les Traductions](#)

```

<?xml version="1.0"?>
<search string="stock Request Search">
  <field name="name"/>
  <field name="product_id"/>
  <field name="product_qty"/>
  <field name="owner_user_id"/>
  <field name="stage_id"/>
  <filter string="My Requests" name="my_requests" domain="[('owner_user_id', '=', uid)]"/>
  <separator/>
  <filter string="To do" domain="[('stage_id.done', '=', False)]" name="todo"/>
  <filter string="Blocked" name="kanban_state_block" domain="[('kanban_state', '=', 'blocked')]/>
  <filter string="In Progress" name="progress" domain="[('kanban_state', '=', 'normal')]/>
  <filter string="Done" name="done" domain="[('kanban_state', '=', 'done')]/>
  <separator/>
  <filter string="Unscheduled request" domain="[('schedule_date', '=', False)]" name="unscheduled"/>
  <separator/>
  <filter string="My Activities" name="activities_my" domain="[('activity_ids.user_id', '=', uid)]"/>
  <separator/>
  <filter string="Late Activities" name="activities_overdue" domain="[('activity_ids.date_deadline', '&lt;', context_today().strf
  <filter string="Today Activities" name="activities_today" domain="[('activity_ids.date_deadline', '=', context_today().strftime
  <filter string="Future Activities" name="activities_upcoming_all" domain="[('activity_ids.date_deadline', '&gt;', context_today
  <separator/>
  <filter string="Unread Messages" name="message_needaction" domain="[('message_needaction', '=', True)]"/>
  <separator/>
  <filter string="Archived" name="inactive" domain="[('archive', '=', True)]"/>
  <filter string="Active" name="active" domain="[('archive', '=', False)]"/>
  <group expand="0" string="Group by...">
    <filter string="Stages" name="stages" domain="[]" context="{ 'group_by' : 'stage_id' }"/>
    <filter string="Created By" name="created_by" domain="[]" context="{ 'group_by': 'owner_user_id' }"/>
  </group>
</search>

```

Sauvegarder

Annuler

FIGURE 16 – Modification de la vue de recherche pour la page demande de matériel

✦ Modifier l'action
✕

---

**Nom de l'action**

**ID Externe**

**Objet**

**Objet source**

**Utilisation de l'action**

**Type d'action**

**Fenêtre cible**

Paramètres Généraux
Sécurité

**Vues**

**Mode de Vue**

**Ref. de Vue**

**Réf. de vue de recherche**

**Filtres**

**Valeur de Domaine**

**Valeur du contexte**

**Limite**

**Recherche automatique**

**Filtrer**

**Restreindre aux listes**

**Aide**

↻ B I U ↵ 16 A ✎ ☰ ☱ ☲ ☳ ☴ ☵ ☶ ☷ 🔗 📄 </> ↺ ↻

Add a new stock request

Follow the process of the request and communicate with the collaborator.

Sauvegarder
Annuler

FIGURE 17 – Modification de l'action pour la page demande de matériel

Les utilisateurs de base ont aussi accès à la liste des articles mais rien de plus :

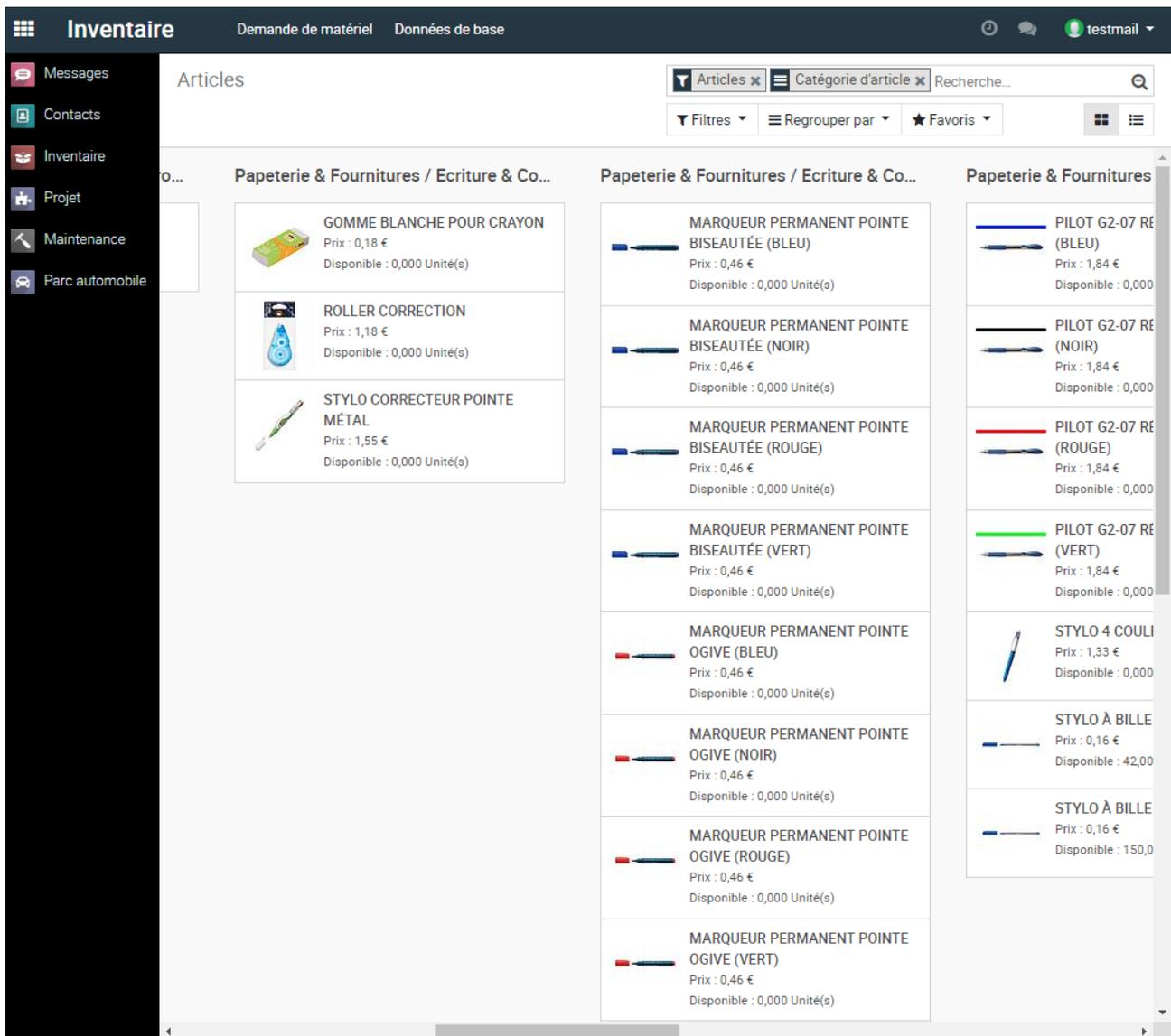


FIGURE 18 – Vue utilisateur de la page "Liste des articles"

Le principe d'utilisation est simple, on renseigne les quantités disponibles dans le stock en faisant un 'inventaire' dans 'Opérations' -> 'Ajustement de stocks'.

Puis lorsqu'une demande de matériel est faite, un mail est envoyé aux gestionnaires, qui se chargent de la traiter. Une fois traitée, il suffit de glisser la demande dans la colonne 'fait' et les quantités sont automatiquement enlevées du stock. Enfin, pour épurer la vue, il faut archiver les vieilles demandes en cliquant sur modifier puis sur *Archiver* et non pas les supprimer.

Un gestionnaire peut visualiser les mouvements de produit par : 'Analyse' -> 'Mouvements de produits'.

### 3.1.8 Équipements et maintenance

Je ne vais pas détailler cette partie car le raisonnement est similaire à celui effectué pour le parc automobile et la vue est quasiment la même que celle de l'inventaire et demande de matériel. Je vais cependant donner le diagramme UML simplifié du Modèle de données :

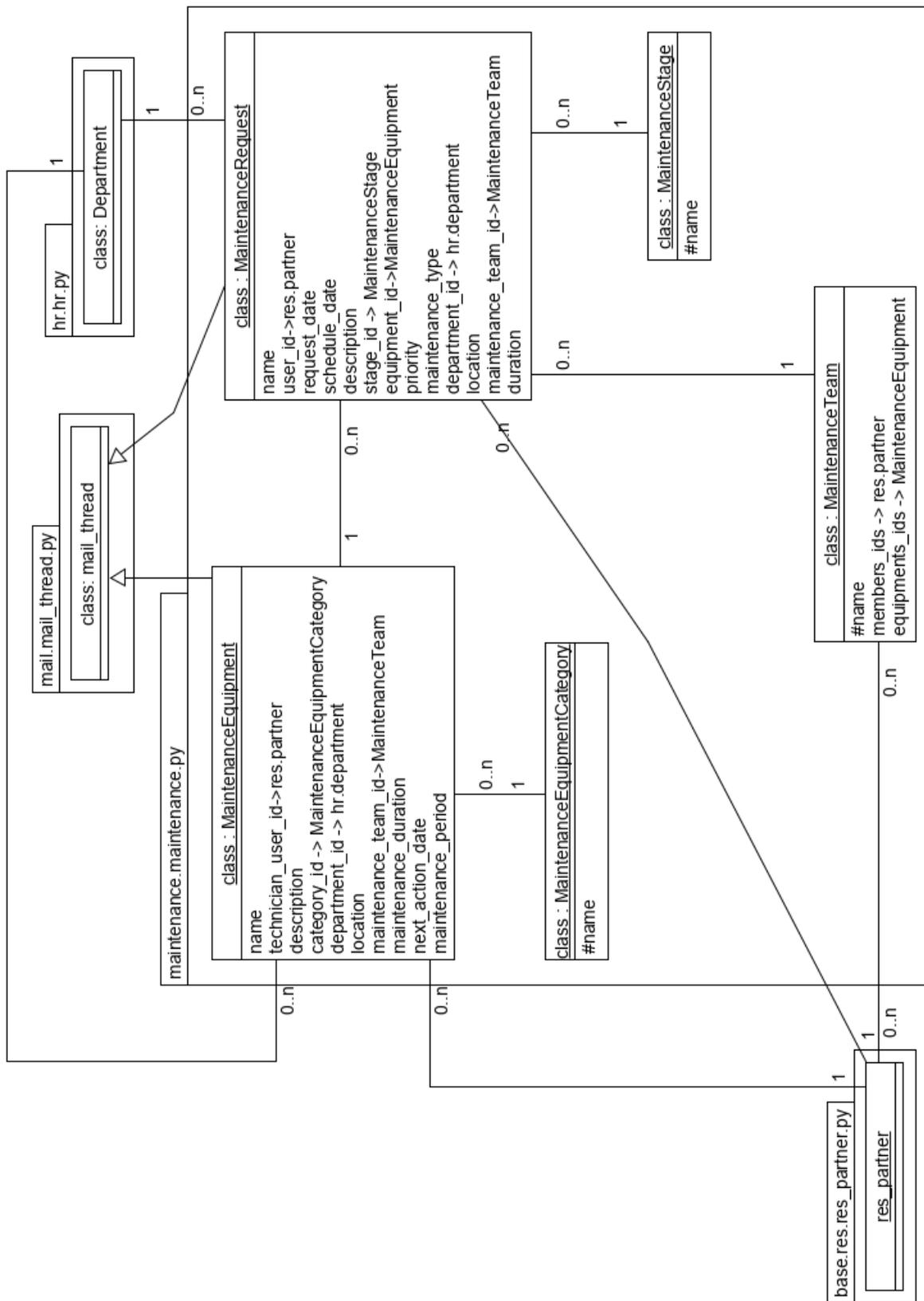


FIGURE 19 – MLD allégé pour le module Maintenance.

## 3.2 Installation d'un serveur RADIUS

Pendant les moments de flottement, durant lesquels mes développements ne pouvaient plus avancer sans un avis utilisateur et de nouvelles demandes de fonctionnalités, j'ai pu m'amuser sur la création d'un serveur d'authentification à distance, destiné à augmenter la sécurité des accès wifi du CDG.

En effet, avant cela, se posait un problème de sécurité : il n'y avait pas d'accès par utilisateur, juste une simple authentification par clé WPA.

Voici un schéma de l'infrastructure WiFi obtenue :

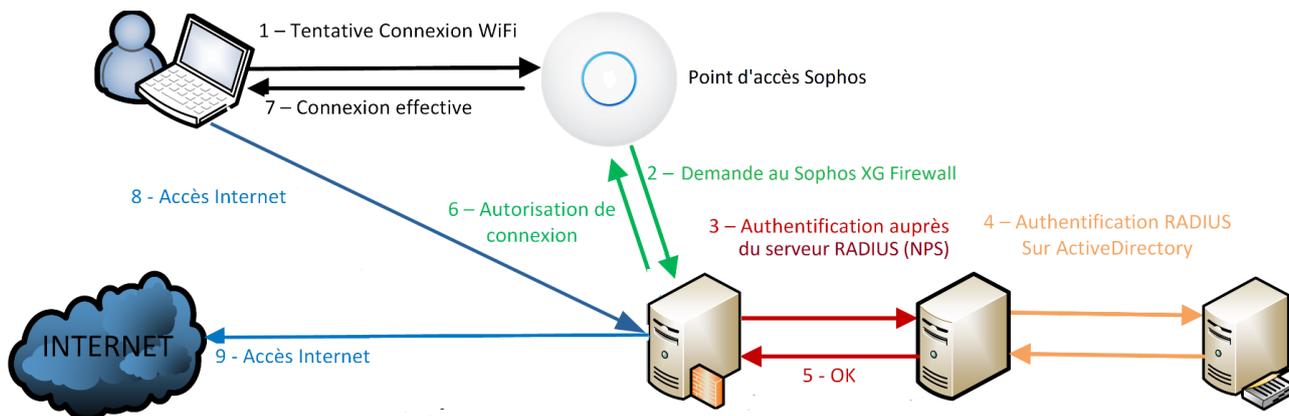


FIGURE 20 – Schéma de l'infrastructure WIFI obtenue

Je décris brièvement en annexe la procédure que j'ai appliquée, en utilisant le service 'NPS' de windows server 2012 et en configurant correctement le Pare feu Sophos déjà en place.

## 3.3 Création de plugin sur Centreon

Centreon est un outil de supervision informatique permettant de surveiller les services, les applications, les hôtes, systèmes et réseaux, basé sur concept de Nagios. Le cœur de Centreon est Open Source, distribué sous licence GPL v2 et gratuit. Des modules complémentaires gratuits et open sources sont développés par des contributeurs sur GitHub.

Je me suis d'abord occupé d'effectuer la migration de centreon vers la version 18.10, ce qui ne fut pas chose facile. Puis j'ai codé un plugin pour les capteurs de températures "serverscheck", ainsi qu'un autre plugin pour le VMware vCenter server et les ESXi.

Les deux plugins ont été codés en python, le premier en utilisant le module requests qui va chercher les informations voulues (températures et humidité) sur l'interface web des capteurs. Le second plugin va directement envoyer en ssh (après création et échange de clés) les commandes permettant d'extraire les informations de CPU, RAM, swap et disques.

Les plugins s'occupent ensuite d'afficher les données extraites dans une chaîne de caractère dans le bon format et de renvoyer le bon code de fin d'exécution.

les messages de sorties sont de cette forme :

```
TEXT OUTPUT | PERFDATA LINE 1 PERFDATA LINE 2 ... PERFDATA LINE N
```

par exemple lors de l'exécution de mon plugin :

```

[/usr/lib/centreon/plugins]# ./check_vmware_vcenter_server.py 192.168.0.11 root vmware mem 0 0
0 0; echo $?
CRITICAL :total :8.19G used :7.91G(96.5%) free :0.28G shared :0.0G buffers :0.5G cached :2.8G
|used=7909788B;;;0;8194248 used_prc=96.52854050792702%;0 :80;0 :90;0;100
2

```

Le code d'erreur permet à centreon de différencier l'état des services et des hôtes de cette manière :

code	service state	host state
0	OK	UP
1	WARNING	UP
2	CRITICAL	DOWN/UNREACHABLE
3	UNKNOWN	DOWN/UNREACHABLE

FIGURE 21 – Tableau des différents codes de sortie d'une commande et leurs significations

Enfin, les données de performances sont automatiquement traitées par centreon et misent dans des graphiques. Pour cela, les données doivent être envoyées sous cette forme :

```
name=data<unit>;warn_min :warn_max ;crit_min :crit_max ;min ;max
```

Dans l'onglet configuration de centreon : il ne reste plus qu'à :

- ajouter les commandes d'exécutions du plugin,
- ajouter les hôtes, les associer à une commande de ping et définir la période de contrôle et de notification (sur 24/7 ici)
- ajouter les différents services appliqués à chaque hôte (CPU, MEM, PING, DISK, SWAP, TEMP, HUM) et définir la période de contrôle et de notification
- redémarrer le collecteur

## 4 Conclusion

Ce premier stage dans le domaine du développement informatique m'a beaucoup intéressé et j'ai beaucoup appris.

- J'ai, premièrement, eu l'occasion de mettre en pratique ce que j'avais appris par le passé, lors de mes études, et au sein d'un vrai projet. J'ai aussi pris l'habitude de me débrouiller seul à l'aide des différentes documentations accessibles par internet. Le dialogue fût aussi un élément important car nous n'utilisons pas toujours les mêmes termes pour désigner les mêmes choses ou encore les besoins réels sont différents de ma vision des choses, j'ai donc pris soin de poser les questions appropriées pour être certain de bien se comprendre.
- J'ai apprécié être confronté à un "vrai" projet, destiné à de "vrais" clients, de découvrir par là une certaine réalité de l'entreprise, souvent faite de contraintes, d'apprendre à adapter une solution informatique au besoins métier.
- Enfin, j'ai trouvé très gratifiant le fait de réussir à remplir la mission qui m'a été confiée en réalisant les différentes fonctionnalités qu'ils m'étaient demandées de faire.



# A Annexe : Développement Durable et Responsabilité Sociale dans l'entreprise

Le CDG Réunion intervient également dans les domaines de la **médecine préventive**, de l'**hygiène et sécurité**, de la **prévention des risques psycho-sociaux**, du **handicap** et de la **qualité de vie au travail**.

Tout d'abord, le service "Prévention des Risques Professionnels", composé de deux ingénieurs et de quatre conseillers en prévention, développe une démarche globale de prévention et accompagne les collectivités dans la mise en œuvre des règles d'hygiène et de sécurité des agents territoriaux.

Ensuite, le Service d'Accompagnement des risques psycho-sociaux, composé d'une psychologue du travail et d'une équipe de psychologues vacataires, réalise un travail de prévention concernant les risques psychosociaux dans une démarche d'écoute individuelle et/ou collective des agents, en vue d'améliorer et de préserver leur qualité de vie au travail.

De plus, le service de médecine professionnelle : composé de 11 médecins du travail et de 10 assistants, ce service permet aux collectivités, dans le cadre de la mutualisation des moyens, de répondre à leurs obligations de visites médicales annuelles en direction de leurs agents. Ces médecins doivent également consacrer le tiers de leur temps à des visites de postes.

Au travers de la Commission de Réforme et du Comité Médical, le Centre de Gestion propose également de gérer les inaptitudes temporaires ou définitives des agents, qu'il s'agisse de maladie, de handicap professionnel ou non.

# B Annexe : code de la partie demande de matériel du module Inventaire

Voici le code :

```

from collections import namedtuple
from datetime import datetime
from dateutil import relativedelta

from odoo import api, fields, models, SUPERUSER_ID, _
from odoo.addons import decimal_precision as dp
from odoo.exceptions import UserError, ValidationError
from odoo.tools import DEFAULT_SERVER_DATETIME_FORMAT

import logging

_logger = logging.getLogger(__name__)

class ProductRequestStage(models.Model):
    """ Model for case stages. This models the main stages of a product Request
    ↪ management flow. """
    _name = 'stock.request.stage'
    _description = 'Product Request Stage'
    _order = 'sequence, id'

    name = fields.Char('Name', required=True, translate=True)
    sequence = fields.Integer('Sequence', default=20)
    fold = fields.Boolean('Folded in Stock Product Request Pipe')
    done = fields.Boolean('Request Done')

class ProductRequest(models.Model):
    _name = 'stock.request'
    _inherit = ['mail.thread', 'mail.activity.mixin']
    _description = 'Demande de matériel'
    _order = "id desc"

    @api.returns('self')
    def _default_stage(self):
        return self.env['stock.request.stage'].search([], limit=1)

    @api.multi
    def _track_subtype(self, init_values):
        self.ensure_one()
        if 'stage_id' in init_values and self.stage_id.sequence <= 1:
            return 'stock.mt_req_created'
        elif 'stage_id' in init_values and self.stage_id.sequence > 1:
            return 'stock.mt_req_status'
        return super(ProductRequest, self)._track_subtype(init_values)

    name = fields.Char('Sujet', required=True)
    description = fields.Text('Description')
    company_id = fields.Many2one('res.company', string='Company', default=lambda self:
        ↪ self.env.user.company_id)
    request_date = fields.Date('Date de la demande', track_visibility='onchange',
        ↪ default=fields.Date.context_today, help="Date où la demande doit être
        ↪ satisfaite")
    owner_user_id = fields.Many2one('res.users', string='Créé par l\\utilisateur',
        ↪ default=lambda s: s.env.uid, required=True)
    product_id = fields.Many2one('product.product', string='Produit', ondelete='restrict',
        ↪ index=True, required=True)
    product_qty = fields.Float('Quantité', digits=0, store=True, help='Quantity in the
        ↪ default UoM of the product', required=True)
    priority = fields.Selection([( '0', 'Very Low'), ('1', 'Low'), ('2', 'Normal'), ('3',
        ↪ 'High'), ('4', 'Very High'), ('5', 'Absolute')], string='Priority')
    color = fields.Integer('Color Index')
    archive = fields.Boolean(default=False, help="Set archive to true to hide the stock
        ↪ product request without deleting it.")
    schedule_date = fields.Datetime('Date prévue', help="Date when it's planned to
        ↪ deliver the product. It should not differ much from the Request Date. ")
    department_id = fields.Many2one('hr.department', string='Service/Pôle')
    stage_id = fields.Many2one('stock.request.stage', string='Stage', ondelete='restrict',
        ↪ track_visibility='onchange', group_expand='_read_group_stage_ids', default
        ↪ =_default_stage)
    kanban_state = fields.Selection([( 'normal', 'In Progress'), ('blocked', 'Blocked'),
        ↪ ('done', 'Ready for next stage')], string='Kanban State', required=True,
        ↪ default='normal', track_visibility='onchange')

```

```

@api.multi
def archive_product_request(self):
    self.write({'archive': True})

@api.multi
def reset_product_request(self):
    """ Reinsert the stock request into the maintenance pipe in the first stage """
    first_stage_obj = self.env['stock.request.stage'].search([], order="sequence asc
    ↪ ", limit=1)
    # self.write({'active': True, 'stage_id': first_stage_obj.id})
    self.write({'archive': False})

@api.model
def create(self, vals):
    # if stage done else create a new stock move
    if 'stage_id' in vals:
        if vals['stage_id'] == self.env['stock.request.stage'].search([('name', '
        ↪ ilike', 'Fait')]).id:
            move = self.env['stock.move'].create({
                'name': 'Demande de matériel satisfaite',
                'location_id': 1,
                'location_dest_id': 9,
                'product_id': vals['product_id'],
                'product_uom': self.env['product.product'].search([('id', '=', vals['
                ↪ product_id'])]).uom_id.id,
                'product_uom_qty': vals['product_qty'],
                'date_expected': datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
                'date': datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
                'company_id': 1,
            })
            move._action_confirm()
            move._action_assign()
            # This creates a stock.move.line record.
            # You could also do it manually using self.env['stock.move.line'].create
            ↪ ({...})
            move.move_line_ids.write({'x_user_id': vals['owner_user_id'], 'qty_done':
            ↪ : vals['product_qty']})
            move._action_done()
            print("[DEBUG] on créé un mouvement de produit pour une demande de maté
            ↪ riel satisfaite !")
    # context: no_log, because subtype already handle this
    self = self.with_context(mail_create_nolog=True)
    request = super(ProductRequest, self).create(vals)
    if request.owner_user_id or request.user_id:
        request._add_followers()
    request.activity_update()
    return request

@api.multi
def write(self, vals):
    # Overriden to reset the kanban_state to normal whenever
    # if stage done else create a new stock move
    if 'stage_id' in vals:
        if vals['stage_id'] == self.env['stock.request.stage'].search([('name', '
        ↪ ilike', 'Fait')]).id:
            move = self.env['stock.move'].create({
                'name': 'Demande de matériel satisfaite',
                'location_id': 1,
                'location_dest_id': 9,
                'product_id': self.product_id.id,
                'product_uom': self.product_id.uom_id.id,
                'product_uom_qty': self.product_qty,
                'date_expected': datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
                'date': datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
                'company_id': 1,
            })
            move._action_confirm()
            move._action_assign()
            # This creates a stock.move.line record.
            # You could also do it manually using self.env['stock.move.line'].create
            ↪ ({...})
            move.move_line_ids.write({'x_user_id': self.owner_user_id.id, 'qty_done':
            ↪ self.product_qty})
            move._action_done()
            print("[DEBUG] on créé un mouvement de produit pour une demande de maté
            ↪ riel satisfaite !")

```

```

# if stage from done to not done then delete stock move
if vals['stage_id'] != self.env['stock.request.stage'].search([( 'name', '
↳ ilike', 'Fait')]).id and self.stage_id.id == self.env['stock.request.
↳ stage'].search([( 'name', 'ilike', 'Fait')]).id:
    move = self.env['stock.move'].create({
        'name': 'Demande de matériel remise à non faite',
        'location_id': 9,
        'location_dest_id': 15,
        'product_id': self.product_id.id,
        'product_uom': self.product_id.uom_id.id,
        'product_uom_qty': self.product_qty,
        'date_expected': datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
        'date': datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
        'company_id': 1,
    })
    move._action_confirm()
    move._action_assign()
    # This creates a stock.move.line record.
    # You could also do it manually using self.env['stock.move.line'].create
    ↳ ({...})
    move.move_line_ids.write({'x_user_id': self.owner_user_id.id, 'qty_done':
    ↳ self.product_qty})
    move._action_done()
    print("[DEBUG] on créé un mouvement de produit pour une demande de maté
    ↳ riel remise à non satisfaite !")
# the stage (stage_id) of the Maintenance Request changes.
if vals and 'kanban_state' not in vals and 'stage_id' in vals:
    vals['kanban_state'] = 'normal'
res = super(ProductRequest, self).write(vals)
if vals.get('owner_user_id') or vals.get('user_id'):
    self._add_followers()
    #self.filtered(lambda m: m.stage_id.done).write({'close_date': fields.Date.
    ↳ today()})
    self.activity_feedback(['stock.mail_act_product_request'])
if vals.get('user_id') or vals.get('schedule_date'):
    self.activity_update()
if vals.get('product_id'):
    # need to change description of activity also so unlink old and create new
    ↳ activity
    self.activity_unlink(['stock.mail_act_product_request'])
    self.activity_update()
return res

def _add_followers(self):
    for request in self:
        partner_ids = (request.owner_user_id.partner_id).ids
        request.message_subscribe(partner_ids=partner_ids)

@api.model
def _read_group_stage_ids(self, stages, domain, order):
    """ Read group customization in order to display all the stages in the
    kanban view, even if they are empty
    """
    stage_ids = stages._search([], order=order, access_rights_uid=SUPERUSER_ID)
    return stages.browse(stage_ids)

def activity_update(self):
    """ Update requests activities based on current record set state.
    It reschedule, unlink or create stock product request activities. """
    self.filtered(lambda request: not request.schedule_date).activity_unlink(['stock
    ↳ .mail_act_stock_request'])
    for request in self.filtered(lambda request: request.schedule_date):
        date_dl = fields.Datetime.from_string(request.schedule_date).date()
        updated = request.activity_reschedule(
            ['stock.mail_act_stock_request'],
            date_deadline=date_dl,
            new_user_id=request.user_id.id or request.owner_user_id.id or self.
            ↳ env.uid)
        if not updated:
            if request.product_id:
                note = _('Request planned for <a href="#" data-oe-model="%s" data-oe
                ↳ -id="%s">%s</a>') % (
                    request.product_id._name, request.product_id.id, request.
                    ↳ product_id.display_name)

```

```

else:
    note = False
    request.activity_schedule(
        'stock.mail_act_stock_request',
        fields.Datetime.from_string(request.schedule_date).date(),
        note=note, user_id=request.user_id.id or request.owner_user_id.
        ↪ id or self.env.uid)

@api.multi
def unlink(self):
    if self.stage_id.id == self.env['stock.request.stage'].search([( 'name', 'ilike', '
    ↪ Fait')]).id:
        move = self.env['stock.move'].create({
            'name': 'Demande de matériel supprimée',
            'location_id': 9,
            'location_dest_id': 15,
            'product_id': self.product_id.id,
            'product_uom': self.product_id.uom_id.id,
            'product_uom_qty': self.product_qty,
            'date_expected': datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
            'date': datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
            'company_id': 1,
        })
        move._action_confirm()
        move._action_assign()
        # This creates a stock.move.line record.
        # You could also do it manually using self.env['stock.move.line'].create
        ↪ ({...})
        move.move_line_ids.write({'x_user_id': self.owner_user_id.id, 'qty_done':
        ↪ self.product_qty})
        move._action_done()
        print("[DEBUG] on créé un mouvement de produit pour une demande de matériel
        ↪ faite mais supprimée !")
    return super(ProductRequest, self).unlink()

```

Et voici la vue :

```

<!-- stock.request : views -->
<record id="product_request_view_search" model="ir.ui.view">
  <field name="name">stock.request.search</field>
  <field name="model">stock.request</field>
  <field name="arch" type="xml">
    <search string="stock_request_search">
      <field name="name"/>
      <field name="product_id"/>
      <field name="product_qty"/>
      <field name="owner_user_id"/>
      <field name="stage_id"/>
      <filter string="MyRequests" name="my_requests" domain="[('owner_user_id
        ↪ ',,uid)]"/>
      <separator/>
      <filter string="ToDo" domain="[('stage_id.done',,False)]" name="
        ↪ todo"/>
      <filter string="Blocked" name="kanban_state_block" domain="[('
        ↪ kanban_state',,blocked)]"/>
      <filter string="InProgress" name="progress" domain="[('kanban_state',,
        ↪ ',normal)]"/>
      <filter string="Done" name="done" domain="[('kanban_state',,done)
        ↪ ]"/>
      <separator/>
      <filter string="Unscheduled_request" domain="[('schedule_date',,
        ↪ False)]" name="unscheduled"/>
      <separator/>
      <filter string="MyActivities" name="activities_my"
        domain="[('activity_ids.user_id',,uid)]"/>
      <separator/>
      <filter string="LateActivities" name="activities_overdue"
        domain="[('activity_ids.date_deadline',,&lt;' ',context_today()
        ↪ strftime('%Y-%m-%d'))]"
        help="Show all records which has next action date is before today"/>
      <filter string="TodayActivities" name="activities_today"
        domain="[('activity_ids.date_deadline',,context_today()
        ↪ strftime('%Y-%m-%d'))]" />
      <filter string="FutureActivities" name="activities_upcoming_all"
        domain="[('activity_ids.date_deadline',,&gt;' ',context_today()
        ↪ strftime('%Y-%m-%d'))]" />
      <separator/>
      <filter string="UnreadMessages" name="message_needaction" domain="[('
        ↪ message_needaction',,True)]"/>
      <separator/>
      <filter string="Archived" name="inactive" domain="[('archive',,True)]
        ↪ "/>
      <group expand='0' string='Group by... '>
        <filter string='Stages' name='stages' domain='[]' context='{
          ↪ group_by':,stage_id}' />
        <filter string='CreatedBy' name='created_by' domain='[]' context='
          ↪ {'group_by':,owner_user_id}' />
      </group>
    </search>
  </field>
</record>

```

```

<record id="product_request_view_form" model="ir.ui.view">
  <field name="name">stock.request.form</field>
  <field name="model">stock.request</field>
  <field name="arch" type="xml">
    <form string="stock_Request">
      <header>
        <button string="Cancel" name="archive_product_request" type="object"
          ↪ attrs="{ 'invisible ':_[( 'archive ',_='_',_True)]}" />
        <button string="Reopen_Request" name="reset_product_request" type="
          ↪ object" attrs="{ 'invisible ':_[( 'archive ',_='_',_False)]}" />
        <field name="stage_id" widget="statusbar" options="{ 'clickable ':_
          ↪ '1'}" attrs="{ 'invisible ':_[( 'archive ',_='_',_True)]}" />
      </header>
      <sheet>
        <div attrs="{ 'invisible ':_[( 'archive ',_='_',_False)]}">
          <span class="badge_badge-warning_float-right">Canceled</span>
        </div>
        <div class="oe_right">
          <field name="kanban_state" class="oe_inline" widget="
            ↪ state_selection" />
        </div>
        <div class="oe_title">
          <label for="name" class="oe_edit_only" string="Name" />
          <h1>
            <field name="name" placeholder="Demande_de_matériel" />
          </h1>
        </div>
        <group>
          <group>
            <field name="owner_user_id" string="Demandé_par" />
            <field name="product_id" />
            <field name="product_qty" string="Quantité_demandée" />
            <field name="request_date" readonly="True" />
            <field name="archive" invisible="1" />
            <field name="priority" widget="priority" />
          </group>
          <group>
            <field name="department_id" />
            <field name="schedule_date" />
          </group>
        </group>
        <field name="description" placeholder="Internal_Note_....." />
      </sheet>
      <div class="oe_chatter">
        <field name="message_follower_ids" widget="mail_followers" />
        <field name="activity_ids" widget="mail_activity" />
        <field name="message_ids" widget="mail_thread" />
      </div>
    </form>
  </field>
</record>

```

```

<record id="hr_product_request_view_kanban" model="ir.ui.view">
  <field name="name">stock.request.kanban</field>
  <field name="model">stock.request</field>
  <field name="arch" type="xml">
    <kanban default_group_by="stage_id">
      <field name="stage_id" />
      <field name="color" />
      <field name="priority" />
      <field name="product_id" />
      <field name="product_qty" />
      <field name="owner_user_id" />
      <field name="kanban_state" />
      <field name="message_needaction_counter" />
      <field name="activity_ids" />
      <field name="activity_state" />
      <templates>
        <t t-name="kanban-tooltip">
          </t>
        <t t-name="kanban-box">
          <div t-attf-class="oe_kanban_color_{kanban_getcolor(record.
            ↪ color.raw_value)}_oe_kanban_card_oe_kanban_global_click_
            ↪ oe_semantic_html_override">
            <div class="o_dropdown_kanban_dropdown">
              <a role="button" class="dropdown-toggle_o-no-caret_btn"
                ↪ data-toggle="dropdown" href="#" aria-label="
                ↪ Dropdown_menu" title="Dropdown_menu">
                <span class="fa_fa-ellipsis-v" />
              </a>
              <div class="dropdown-menu" role="menu">
                <t t-if="widget.editable"><a role="menuitem" type="
                  ↪ edit" class="dropdown-item">Edit...</a></t>
                <t t-if="widget.deletable"><a role="menuitem" type="
                  ↪ delete" class="dropdown-item">Delete</a></t>
                <ul class="oe_kanban_colorpicker" data-field="color"
                  ↪ />
              </div>
            </div>
          </div>
          <div class="oe_kanban_content" tooltip="kanban-tooltip">
            <div class="o_kanban_record_top">
              <b class="o_kanban_record_title"><field name="name" />
              </b>
            </div>
            <div class="o_kanban_record_body">
              <span name="owner_user_id" t-if="record.
                ↪ owner_user_id.raw_value">Demandé par : <field
                ↪ name="owner_user_id" /><br/></span>
              <span class="oe_grey" t-if="record.product_id.
                ↪ raw_value"><field name="product_id" /><br/></span>
              <span class="oe_grey" t-if="record.product_qty.
                ↪ raw_value"><field name="product_qty" /><br/></span>
            </div>
            <div class="o_kanban_record_bottom">
              <div class="oe_kanban_bottom_left">
                <field name="priority" widget="priority" />
                <t t-if="record.message_needaction_counter.
                  ↪ raw_value">
                  <span role="alert" class="oe_kanban_mail_new
                    ↪ ' title='Unread_Messages'<i class='
                    ↪ fa_fa-comments' role="img" aria-label
                    ↪ ='Unread_Messages' /><t t-raw="record.
                    ↪ message_needaction_counter.raw_value"
                    ↪ /></span>
                </t>
                <div class="o_kanban_inline_block_ml4_mr4">
                  <field name="activity_ids" widget="
                    ↪ kanban_activity" />
                </div>
              </div>
            </div>
          </div>
        </t>
      </templates>
    </kanban>
  </field>
</record>

```

```

        <div class="oe_kanban_bottom_right">
            <field name="kanban_state" widget="
                ↪ state_selection"/>
            
        </div>
    </div>
</div>
<div class="oe_clear"></div>
</div>
</t>
</templates>
</kanban>
</field>
</record>

<record id="hr_product_request_view_tree" model="ir.ui.view">
    <field name="name">stock.request.tree</field>
    <field name="model">stock.request</field>
    <field name="arch" type="xml">
        <tree string="stock␣Request" decoration-bf="message_needaction==True">
            <field name="message_needaction" invisible="1"/>
            <field name="name"/>
            <field name="request_date" groups="base.group_no_one"/>
            <field name="owner_user_id"/>
            <field name="product_id"/>
            <field name="product_qty"/>
            <field name="stage_id"/>
        </tree>
    </field>
</record>

<record id="hr_product_request_view_graph" model="ir.ui.view">
    <field name="name">stock.request.graph</field>
    <field name="model">stock.request</field>
    <field name="arch" type="xml">
        <graph string="stock␣Request">
            <field name="owner_user_id"/>
            <field name="stage_id"/>
        </graph>
    </field>
</record>

<record id="hr_product_request_view_pivot" model="ir.ui.view">
    <field name="name">stock.request.pivot</field>
    <field name="model">stock.request</field>
    <field name="arch" type="xml">
        <pivot string="stock␣Request">
            <field name="owner_user_id"/>
            <field name="stage_id"/>
        </pivot>
    </field>
</record>

<record id="hr_product_view_calendar" model="ir.ui.view">
    <field name="name">stock.request.calendar</field>
    <field name="model">stock.request</field>
    <field name="arch" type="xml">
        <calendar date_start="schedule_date" color="owner_user_id">
            <field name="name"/>
            <field name="owner_user_id"/>
        </calendar>
    </field>
</record>

```

```

<!-- stock.request : actions -->
<record id="hr_product_request_action" model="ir.actions.act_window">
  <field name="name">Demande de matériel</field>
  <field name="res_model">stock.request</field>
  <field name="view_mode">kanban , tree , form , pivot , graph , calendar</field>
  <field name="view_id" ref="hr_product_request_view_kanban" />
  <field name="context">{'default_user_id': uid}</field>
  <field name="help" type="html">
    <p class="o_view_nocontent_smiling_face">
      Add a new stock request
    </p><p>
      Follow the process of the request and communicate with the collaborator.
    </p>
  </field>
</record>

<record id="hr_product_todo_request_action_from_dashboard" model="ir.actions.
  ↪ act_window">
  <field name="name">stock Requests</field>
  <field name="res_model">stock.request</field>
  <field name="view_mode">kanban , tree , form , pivot , graph , calendar</field>
  <field name="context">{
    'default_stock_stage_id': active_id ,
  }</field>
  <field name="help" type="html">
    <p class="o_view_nocontent_smiling_face">
      Add a new stock request
    </p><p>
      Follow the process of the request and communicate with the collaborator.
    </p>
  </field>
</record>

<record id="hr_product_request_action_cal" model="ir.actions.act_window">
  <field name="name">Stock Requests</field>
  <field name="res_model">stock.request</field>
  <field name="view_mode">calendar , kanban , tree , form , pivot , graph</field>
  <field name="view_id" ref="hr_product_view_calendar" />
  <field name="help" type="html">
    <p class="o_view_nocontent_smiling_face">
      Add a new stock request
    </p><p>
      Follow the process of the request and communicate with the collaborator.
    </p>
  </field>
</record>

<record id="stock_request_action_reports" model="ir.actions.act_window">
  <field name="name">Stock Requests</field>
  <field name="res_model">stock.request</field>
  <field name="view_mode">graph , pivot , kanban , tree , form , calendar</field>
  <field name="help" type="html">
    <p class="o_view_nocontent_smiling_face">
      Add a new stock request
    </p><p>
      Follow the process of the request and communicate with the collaborator.
    </p>
  </field>
</record>

```

```

<!-- stock.stage : views -->
<record id="hr_product_stage_view_search" model="ir.ui.view">
  <field name="name">stock.request.stage.search</field>
  <field name="model">stock.request.stage</field>
  <field name="arch" type="xml">
    <search string="Stock␣Request␣Stages">
      <field name="name" string="Stock␣Request␣Stages"/>
    </search>
  </field>
</record>

<record id="hr_product_stage_view_tree" model="ir.ui.view">
  <field name="name">stock.request.stage.tree</field>
  <field name="model">stock.request.stage</field>
  <field name="arch" type="xml">
    <tree string="Maintenance␣Request␣Stage" editable="top">
      <field name="sequence" widget="handle"/>
      <field name="name"/>
      <field name="fold"/>
      <field name="done"/>
    </tree>
  </field>
</record>

<record id="hr_product_stage_view_kanban" model="ir.ui.view">
  <field name="name">stock.request.stage.kanban</field>
  <field name="model">stock.request.stage</field>
  <field name="arch" type="xml">
    <kanban class="o_kanban_mobile">
      <templates>
        <t t-name="kanban-box">
          <div t-attf-class="oe_kanban_global_click">
            <div>
              <strong>×field name="name"/</strong>
            </div>
          </div>
        </t>
      </templates>
    </kanban>
  </field>
</record>

<!-- stock.stages : actions -->
<record id="hr_product_stage_action" model="ir.actions.act_window">
  <field name="name">Stages</field>
  <field name="res_model">stock.request.stage</field>
  <field name="view_mode">tree,kanban,form</field>
  <field name="view_type">form</field>
  <field name="help" type="html">
    <p class="o_view_nocontent_smiling_face">
      Add a new stage in the stock request
    </p>
  </field>
</record>

<!-- Menu items hierachy -->
<menuitem
  id="menu_s_request_form"
  name="Demande␣de␣matériel"
  parent="menu_stock_root"
  action="hr_product_request_action"
  sequence="1"/>

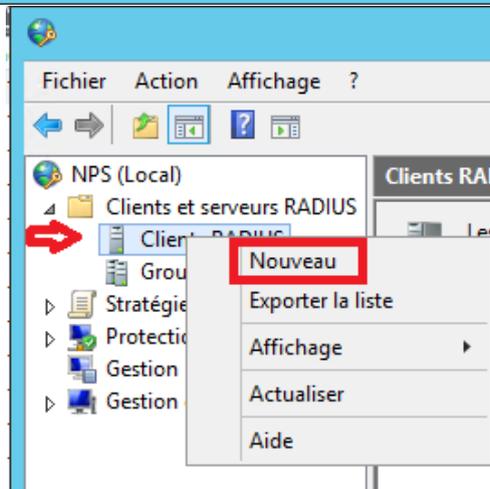
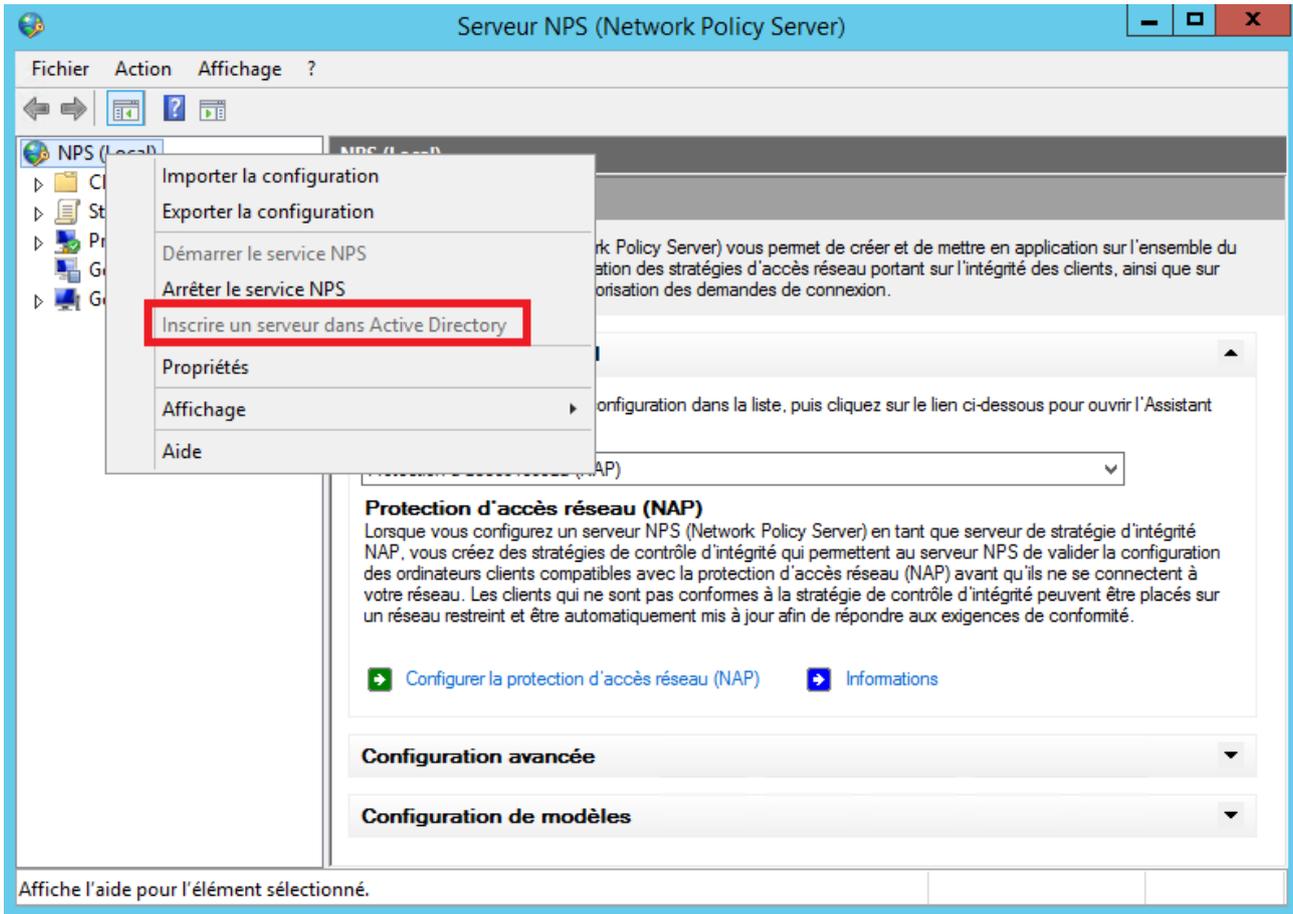
<menuitem
  id="menu_stock_stage_configuration"
  name="Stock␣requests␣Stages"
  parent="menu_stock_config_settings"
  action="hr_product_stage_action"
  sequence="3"/>

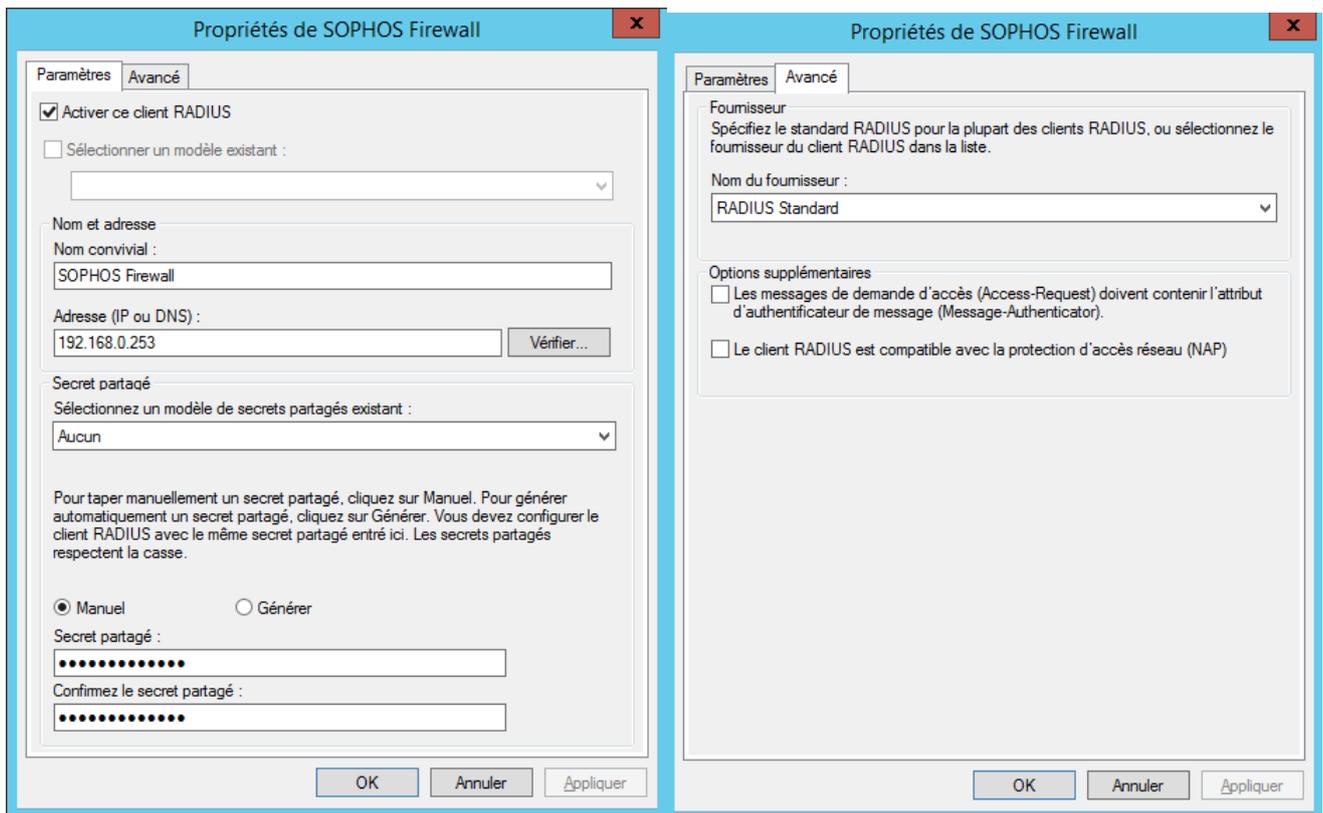
```



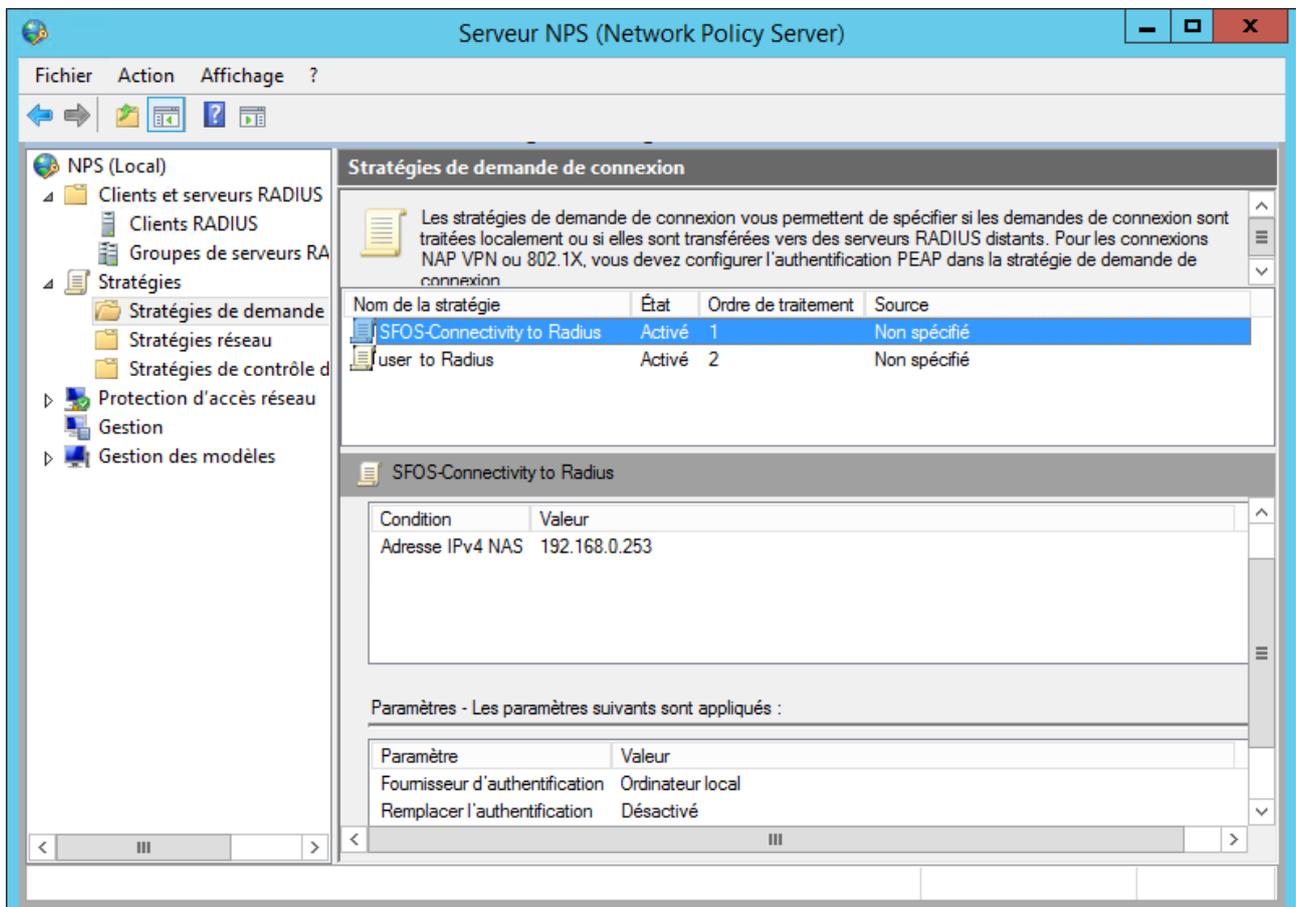
Il faut maintenant configurer le serveur NPS qui servira de RADIUS (Remote Authentication Dial-In User Service).

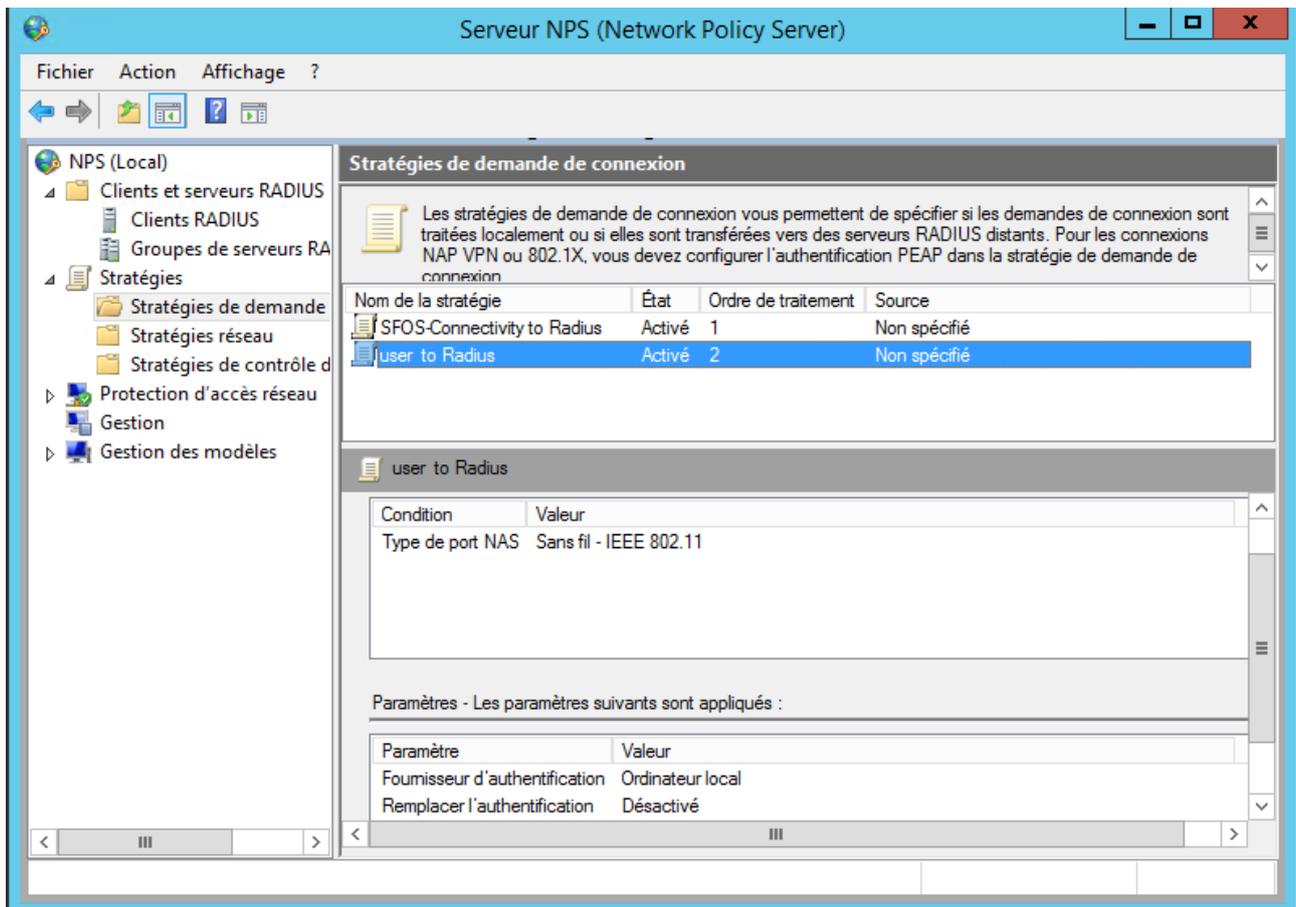
Je commence par inscrire le serveur dans l'AD, et on renseigne les informations sur le client RADIUS : le pare-feu SOPHOS.





Ensuite, je configure les stratégies de demande de connexion : une pour le test de connexion et une autre pour l'utilisation en pratique. (En effet, dans le journal des événements j'ai pu remarquer que les informations envoyées lors de ces deux type d'échanges ne sont pas les mêmes).





On peut remarquer que le paramètre "Remplacer l'authentification" est à la valeur "Désactivé". Si l'on tente de l'activer comme le montre dans la documentation officielle de sophos (<https://community.sophos.com/kb/en-us/132912>), le serveur NPS n'arrivera plus à traiter l'authentification PEAP. Cela m'a demandé plusieurs heures de tests et recherches sur les certificats d'authentification.

Je configure ensuite les stratégies réseau, en créant deux profils : un pour le testing et un autre pour l'utilisation normale :

- NPS (Local)
  - Clients et serveurs RADIUS
    - Clients RADIUS
    - Groupes de serveurs RA
  - Stratégies
    - Stratégies de demande
    - Stratégies réseau
    - Stratégies de contrôle d
  - Protection d'accès réseau
    - Gestion
  - Gestion des modèles

### Stratégies réseau

Les stratégies réseau vous permettent d'autoriser les connexions au réseau de manière sélective, et d'indiquer les circonstances dans lesquelles ces connexions peuvent s'effectuer ou non.

Nom de la stratégie	État	Ordre de traitement	Type d'accès	Source
Secure Wireless connections	Activé	1	Accorder l'accès	Non spécifié
SFOS Connectivity testing to Radius	Activé	2	Accorder l'accès	Non spécifié

#### Secure Wireless connections

Conditions - Si les conditions suivantes sont réunies :

Condition	Valeur
Groupes Windows	CDG\WiFi
Type d'authentification	EAP

Paramètres - Les paramètres suivants sont appliqués :

Paramètre	Valeur
Configuration du protocole EAP (Extensible Authentication Protocol)	Configuré
Autorisation d'accès	Accorder l'accès
Méthode EAP (Extensible Authentication Protocol)	Microsoft: PEAP (Protected EAP)
Type de port NAS	Sans fil - IEEE 802.11
Méthode d'authentification	Protocole EAP OU MS-CHAP v1 OU MS-CHAP v1 (l'utilisateur peut modifier l...
Contrainte de mise en conformité NAP	Autoriser un accès réseau complet
Mettre à jour les clients non conformes	Vrai
Framed-Protocol	PPP
Service-Type	Framed

- NPS (Local)
  - Clients et serveurs RADIUS
    - Clients RADIUS
    - Groupes de serveurs RA
  - Stratégies
    - Stratégies de demande
    - Stratégies réseau
    - Stratégies de contrôle d
  - Protection d'accès réseau
    - Gestion
  - Gestion des modèles

### Stratégies réseau

Les stratégies réseau vous permettent d'autoriser les connexions au réseau de manière sélective, et d'indiquer les circonstances dans lesquelles ces connexions peuvent s'effectuer ou non.

Nom de la stratégie	État	Ordre de traitement	Type d'accès	Source
Secure Wireless connections	Activé	1	Accorder l'accès	Non spécifié
SFOS Connectivity testing to Radius	Activé	2	Accorder l'accès	Non spécifié

#### SFOS Connectivity testing to Radius

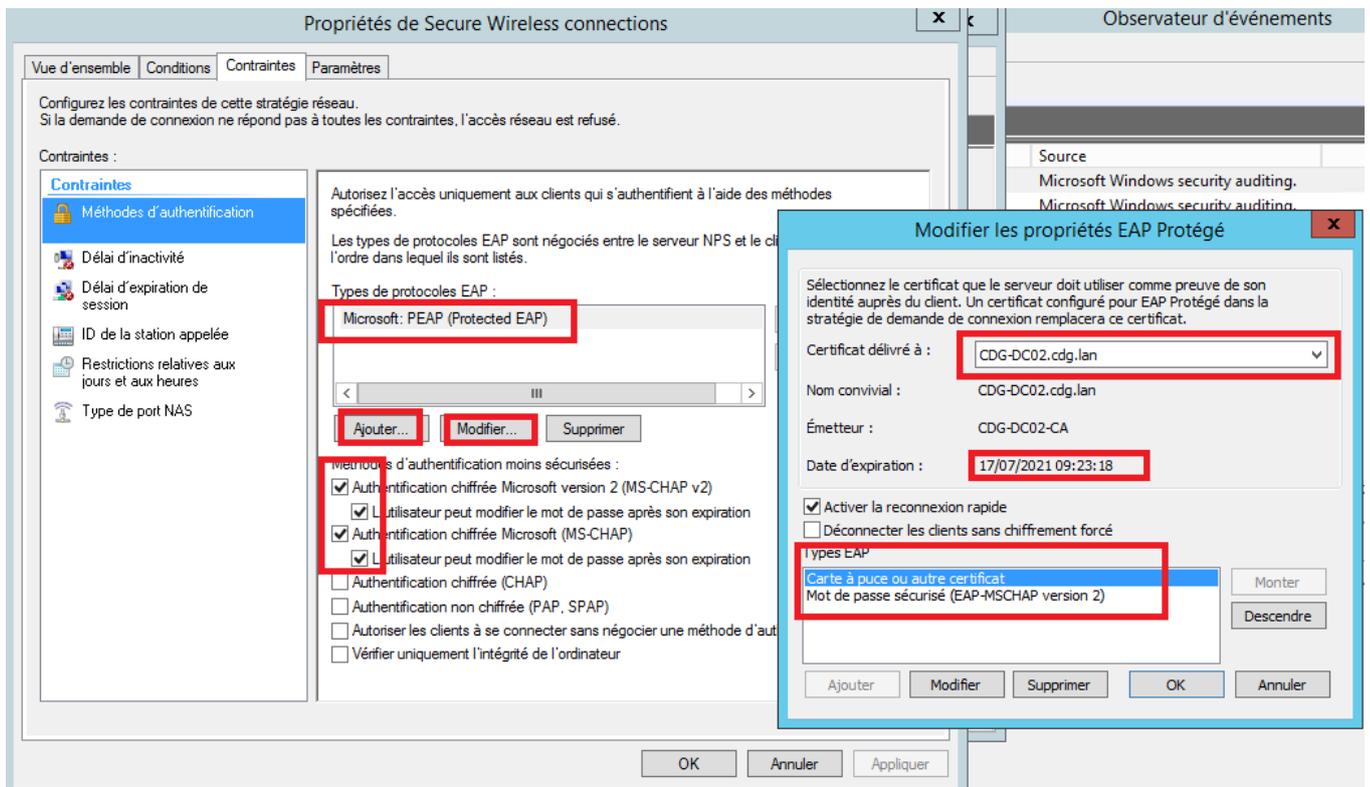
Conditions - Si les conditions suivantes sont réunies :

Condition	Valeur
Adresse IPv4 NAS	192.168.0.253
Groupes Windows	CDG\WiFi
Type d'authentification	PAP

Paramètres - Les paramètres suivants sont appliqués :

Paramètre	Valeur
Autorisation d'accès	Accorder l'accès
Méthode d'authentification	Authentification non chiffrée (PAP, SPAP)
Contrainte de mise en conformité NAP	Autoriser un accès réseau complet
Mettre à jour les clients non conformes	Vrai
Framed-Protocol	PPP
Service-Type	Framed

Pour la stratégie "Secure Wireless connections" j'ai configuré le protocole PEAP de cette manière (j'ai recréé un certificat de type serveur RAS et IAS mais ce n'est pas forcément nécessaire comme il y en avait déjà un pour le contrôleur de domaine) :



Enfin je configure le firewall.  
Je créer le serveur RADIUS :

Je crée un nouveau réseau sans fil avec le mode de sécurité WPA2 Entreprise et je rajoute ce nouveau réseau au groupe qui contient les points d'accès utiles :

**SOPHOS** XG Firewall

MONITOR & ANALYZE

- Control center
- Current activities
- Reports
- Diagnostics

PROTECT

- Firewall
- Intrusion prevention
- Web
- Applications
- Wireless**
- Email
- Web server
- Advanced threat
- Central synchronization

## Edit group

Wireless client list | Wireless networks | Access points | **Access point groups** | Mesh networks

Name \*

Wireless networks

wlnet1	 
wlnet2	 



Add new item

VLAN tagging  Enable

Access points

AP15C Etage	
AP15C Couloir Rdc	

Add new item

Enfin je lance une connexion et je vérifie le journal des évènements :

Le serveur NPS a accordé l'accès total à un utilisateur car l'hôte répond aux critères définis par la stratégie d'intégrité.

Utilisateur :  
ID de sécurité : CDG\ldubardA  
Nom de compte : ldubardA  
Domaine du compte : CDG  
Nom de compte complet : cdg.lan/Utilisateurs/Loic DUBARD

Ordinateur client :  
ID de sécurité : NULL SID  
Nom de compte : -  
Nom de compte complet : -  
Version du système d'exploitation : -  
Identificateur de la station appelée : 00-1A-8C-A4-0C-B5:CDG\_RADIUS\_TEST  
Identificateur de la station appelante : 94-B8-6D-48-43-CE

Serveur NAS :  
Adresse IPv4 du serveur NAS : -  
Adresse IPv6 du serveur NAS : -  
Identificateur du serveur NAS : CDG\_RADIUS\_TEST  
Type de port du serveur NAS : Sans fil - IEEE 802.11  
Port du serveur NAS : 1

Client RADIUS :  
Nom convivial du client : SOPHOS Firewall  
Adresse IP du client : 192.168.0.253

Informations détaillées de l'authentification :  
Nom de la stratégie de demande de connexion : user to Radius  
Nom de la stratégie réseau : Secure Wireless connections  
Fournisseur d'authentification : Windows  
Serveur d'authentification : CDG-DC02.cdg.lan  
Type d'authentification : PEAP  
Type EAP : Microsoft: Mot de passe sécurisé (EAP-MSCHAP version 2)  
Identificateur de la session du compte : 35443246303032312D3030303030303134

Informations de quarantaine :  
Résultat : Accès complet

Journal : Sécurité

Source : Microsoft Windows security    Connecté : 17/07/2019 11:31:39

Événement : 6278    Catégorie : Serveur NPS

Niveau : Information    Mots-clés : Succès de l'audit

Utilisateur : N/A    Ordinateur : CDG-DC02.cdg.lan

Opcode : Informations

J'ai eu affaire aussi à ce problème, lors de l'essai de connexion à partir d'un smartphone sous android :

Informations détaillées sur l'authentification :  
Nom de la stratégie de demande de connexion : user to Radius  
Nom de la stratégie réseau : Secure Wireless connections  
Fournisseur d'authentification : Windows  
Serveur d'authentification : CDG-DC02.cdg.lan  
Type d'authentification : PEAP  
Type EAP : -  
Identificateur de la session du compte : 35443246303032312D3030303030303143  
Résultats de la journalisation : Les informations de suivi ont été inscrites dans le fichier journal local.  
Code raison : 265  
Raison : La chaîne de certificats a été fournie par une autorité qui n'est pas approuvée.

La solution est de configurer le réseau sur "ne pas valider le certificat" côté smartphone :

## Add network

Network name

CDG\_RADIUS\_TEST

---

Security

802.1x EAP



EAP method

PEAP



Phase 2 authentication

MSCHAPV2



CA certificate

Do not validate



Identity

ldubardA

---

Anonymous identity

Anonymous

---

Password

\*\*\*\*\*

---

[Cancel](#) [Save](#)

# D Annexe : Code du plugin centreon pour les capteurs de températures de la salle serveur

```
import requests , sys , re

help="Usage : "+sys.argv[0]+" [IP-ADDR] "+" [up|internal|external|humidity]\n"
data={"Authorization": "Basic YWRtaW46YWRtaW4="}

if sys.argv[1] in ["help", "-h", "--help"]:
    print(help)
    exit(0)

url=sys.argv[1]
for i in url.split("."):
    try:
        assert type(int(i))==int, "UNKNOWN: bad ip address format"
    except:
        print("UNKNOWN: bad ip address")
        exit(3)
if len(sys.argv)<2:
    print("UNKNOWN: less than 2 arguments given")
    exit(3)
session=requests.session()
try:
    response=session.get("http://"+url+"/retallvalue.xml", headers=data).content.decode("
    ↳ UTF-8").split("\r\n")
except:
    print("UNKNOWN: no connection to the sensor. Perhaps verify authorization headers in
    ↳ the plugin script ? Or maybe you've just entered a bad ip ?")
    exit(3)

def number_of_value(list):
    """return the number of sensors"""
    i=0
    for elt in list:
        if 'ssvalue' in elt:
            i+=1
    return i
def search(list, text):
    """return the index of the element of list where we can find text"""
    i=0
    for elt in list:
        if text in elt:
            break
        i+=1
    return i

def get_value(response, text):
    return re.split("<>", response[search(response, text)])[2]

def final(code, message):
    status=['OK: ', 'WARNING: ', 'CRITICAL: ', 'UNKNOWN: ']
    print(status[code]+message)
    exit(code)

def compute(name, matrix, status):
    message=name+": "+matrix[name]['currentvalue']+{"Temp.": "deg;", "Humid.": "%;", "Power":
    ↳ ";"}[matrix[name]['type']]
    try:
        if not float(matrix[name]['downrangemin']) < float(matrix[name]['currentvalue'])
        ↳ < float(matrix[name]['downrangemax']):
            message+="";
            status=max(status, 2)
        elif not float(matrix[name]['warningrangemin']) < float(matrix[name]['
        ↳ currentvalue']) < float(matrix[name]['warningrangemax']):
            message+="";
            status=max(status, 1)
    except:
        try:
            if matrix[name]['currentvalue'] not in ['PWR OK', '?']:
                status=max(status, 2)
        except:
            status=3
    return status, message
```

```

matrix={}
message=""
status=0
stop_at=-1
if (len(sys.argv)==3):
    stop_at=['internal','external','humidity','up'].index(sys.argv[2])

for i in range(number_of_value(response)):
    name=get_value(response,'name'+str(i))
    matrix[name]={
        'state':0,
        'id':get_value(response,'ssid'+str(i)),
        'name':name,
        'currentvalue':get_value(response,'ssvalue'+str(i)),
        'warningrangemin':get_value(response,'rsafe'+str(i)+"m"),
        'warningrangemax':get_value(response,"rsafe"+str(i)+"x"),
        'downrangemin':get_value(response,'rwarn'+str(i)+"m"),
        'downrangemax':get_value(response,"rwarn"+str(i)+"x"),
        'type':get_value(response,'sstype'+str(i))
    }
    status_i,message_i=compute(name,matrix,0)
    if stop_at==i:
        final(status_i,message_i+" | temp="+matrix[name]['currentvalue']+" warnmin="+
            ↪ matrix[name]['warningrangemin']+" warnmax="+matrix[name]['warningrangemax']
            ↪ +" critmin="+matrix[name]['downrangemin']+" critmax="+matrix[name]['
            ↪ downrangemax'])
        status=max(status,status_i)
        message+=message_i

if stop_at==3:
    status=0#if up then give infos but always OK.
final(status,message)

```

# E Annexe : Code du plugin centreon pour le vCenter Server

```
#!/opt/rh/rh-python36/root/bin/python
import os, subprocess, sys
from subprocess import Popen, PIPE

def error():
    print("UNKNOWN: WRONG COMMAND ARGUMENTS !")
    exit(3)

if len(sys.argv)==1 or sys.argv[1] in ['-h', '--help']:
    print('Usage: '+sys.argv[0]+' [host] [user] [password] [cpu|temp|mem|disk|swap|
    ↪ traffic] [warningmin] [warningmax] [criticalmin] [criticalmax]')
    exit(3)
if len(sys.argv) != 9:
    error()

cmd, host, user, password, option, warnmin, warnmax, critmin, critmax=sys.argv
try:cmd={'cpu': 'mpstat -P ALL', 'temp': '', 'mem': 'free', 'swap': 'free', 'traffic': 'ifstat', '
    ↪ disk': 'df'}[option]
except: error()
try:
    output, err = Popen(['sshpass', '-p', password, 'ssh', '-i', '~/.ssh/id_rsa', '-o', '
    ↪ StrictHostKeyChecking=no', user+'@'+host, cmd], stdin=PIPE, stdout=PIPE,
    ↪ stderr=PIPE).communicate()
    if critmax!='0':
        print(output.decode())
except Exception as e:
    print(e)
    exit(3)

def traitement(output):
    perfs=output.decode('utf-8').split('\n')
    perfs_new=[] for line in perfs]
    tmp=[]
    i=0
    for line in perfs:
        tmp=line.split(" ")
        for elt in tmp:
            if elt != "":
                perfs_new[i].append(elt)
        i+=1
    return perfs_new

def mem(output, swap=False):
    perfs=traitement(output)
    message=""
    data=0

    used=int(perfs[1 if not swap else 3][2])
    total=int(perfs[1 if not swap else 3][1])
    used_prc=used*100/total

    for i in range(len(perfs[0]) if not swap else 3):
        data=int(perfs[1 if not swap else 3][i+1])
        message+="{:}{:}G{ } ".format(perfs[0][i], round(data/1000000,2), "({}%)" .
            ↪ format(round(used_prc,1)) if i == 1 else '')

    message+="|"+'used '+"+str(used)+"B;;;0;"+str(total)+" used_prc="+str(used_prc)
    ↪ +"%;0:80;0:90;0:100"
    if used_prc>90:code=2
    elif used_prc>80:code=1
    else: code=0
    return code, message
```

```

def cpu(output):
    perfs=traitement(output)[3:]
    perfs.remove([])
    cpu_idle=[float(line[-1]) for line in perfs]
    cpu_used_usr=[float(line[2]) for line in perfs]
    cpu_used_nice=[float(line[3]) for line in perfs]
    cpu_used_sys=[float(line[4]) for line in perfs]
    cpu_used_iowait=[float(line[5]) for line in perfs]
    cpu_used_irq=[float(line[6]) for line in perfs]
    cpu_used_soft=[float(line[7]) for line in perfs]
    cpu_used_steal=[float(line[8]) for line in perfs]
    cpu_used_guest=[float(line[9]) for line in perfs]
    cpu_names=[line[1] for line in perfs]
    message=""
    for i in range(len(cpu_used_usr)-1,-1,-1):
        message="usr_"+cpu_names[i]+": "+str(cpu_used_usr[i])+"% nice_"+cpu_names
        ↪ [i]+": "+str(cpu_used_nice[i])+"% sys_"+cpu_names[i]+": "+str(
        ↪ cpu_used_sys[i])+"% idle_"+cpu_names[i]+": "+str(cpu_idle[i])+"%\n
        ↪ "+message
        message+="cpu_"+cpu_names[i]+"="+str(round(100-cpu_idle[i],2))+
        ↪ "%;0:80;0:90;0:100 "
    if cpu_idle[0]<10:code=2
    elif cpu_idle[0]<20:code=1
    else:
        code=0
    return code,message

def disk(output):
    perfs=traitement(output)
    code,message,total,used,free=0,"",0,0,0
    for i in range(1,len(perfs)-1):
        total,used,free=int(perfs[i][1]),int(perfs[i][2]),int(perfs[i][3])
        message="Storage \'{ }\' Usage Total: { }G Used: { }G ({}%) Free: { }G ({}%
        ↪ \n".format(perfs[i][-1],total/1024/1000,used/1024/1000,round(used
        ↪ /total*100,2),free/1024/1000,round(free/total*100,2))+message #
        ↪ corps du message
        message+= "used_{ }={ }B;0:{ };0:{ };0:{ } ".format(perfs[i][-1],used,0.80*
        ↪ total,0.90*total,total) #données de performance
        if int(perfs[i][4].replace('%',''))>90:
            code=max(2,code)
        elif int(perfs[i][4].replace('%',''))>80:
            code=max(1,code)
    if code==0:
        message="All storages are ok.\n"+message
    elif code==1:
        message="At least one storage is above 80% used\n"+message
    else:
        message="At least one storage is above 90% used\n"+message
    return code,message

def final(code,message):
    print(['OK:', 'WARNING:', 'CRITICAL:', 'UNKNOWN:'][code]+message)
    exit(code)

if option=='mem':
    code,message=mem(output)
elif option=='swap':
    code,message=mem(output,swap=True)
elif option=='cpu':
    code,message=cpu(output)
elif option=='disk':
    code,message=disk(output)
final(code,message)

```

# Bibliographie

- Site du CDG :<https://www.cdgreunion.fr>
- Documentation de sophos pour l'installation d'un serveur RADIUS sur windows server 2012 : <https://community.sophos.com/kb/en-us/132912#what%20to%20do>
- migration de centreon vers la 18.10 : <https://pixelabs.fr/mise-a-jour-centreon-2-8-26-vers-18-10-iso-18-10-5/>
- documentation sur l'utilisation de plugins avec NAGIOS <https://pixelabs.fr/nagios-plugins-monitoring/>
  
- mémoire de fin d'étude pour l'obtention du diplôme de Master en Informatique sur le développement d'une solution ERP pour la gestion de maintenance assistée par ordinateur : <http://dspace.univ-tlemcen.dz/bitstream/112/12141/1/Developpement-dune-solution-ERP.pdf>
  
- créer un mouvement de stock code python <https://stackoverflow.com/questions/47739027/how-to-create-stock-move-in-odoo-from-custom-module>