

31/12/2025

SAE3.02 Développer des applications communicantes

Fiche d'installation et d'utilisation

Table des matières

1. Prérequis.....	1
1.1 Systèmes et environnement	1
1.2 Configuration réseau des machines	2
1.3 Paquets à installer sur les VM Linux.....	2
1.3.1 Installation de Python et outils nécessaires	2
1.3.2 Installation des dépendances de la bibliothèque python PyQt5 (Sur toute les machine Linux).....	2
1.4 Bibliothèques Python requises.....	3
1.4.1 MySQL Connector (MariaDB)	3
1.4.2 PyQt5 (interfaces graphiques).....	3
1.5 Base de données MariaDB.....	3
2. Installation du projet	4
3. Lancement et utilisation	4
3.1 Lancement du Master	5
3.2 Lancement d'un Routeur	5
3.3 Lancement d'un Client	6
4. Limitations connues.....	7
5. Sécurité.....	7

1. Prérequis

1.1 Systèmes et environnement

- Système d'exploitation :
 - Linux (Ubuntu ou Debian)
 - Windows possible
- Python 3.10 minimum
- Accès réseau entre les machines (LAN ou machines virtuelles)

Vérification de la version Python :

```
python3 –version
```

1.2 Configuration réseau des machines

Il faut au moins 4 machines (physique ou virtuelle).

Par exemple :

- Une VM Linux pour le Master.
- Une VM Linux pour les Routeurs (qui héberge tous les routeurs).
- Une VM Linux pour un Client (récepteur du message).
- L'hôte Windows pour un deuxième Client (émetteur du message).

Il faut toujours deux cartes réseaux pour les VM :

- Une interface NAT pour avoir la connexion à Internet.
- Une interface en Host-only (réseau privé-hôte) pour que les machines puissent communiquer entre elles dans le même réseau.

Ces deux interfaces sont à configurer en DHCP sur toutes les machines.

Exemple sur linux :

Ouvrir le fichier des interfaces réseaux

```
nano /etc/network/interfaces
```

Vérifier que l'interface choisie pour le mode Host-only soit en DHCP (on avoir ceci)

```
auto <id_interface>  
iface <id_interface> inet dhcp
```

Si la modification de ce fichier est nécessaire, redémarrer le service networking

```
Systemctl restart networking
```

1.3 Paquets à installer sur les VM Linux

Mise à jour du système

```
sudo apt update
```

1.3.1 Installation de Python et outils nécessaires

```
sudo apt install -y python3 python3-pip python3-venv
```

1.3.2 Installation des dépendances de la bibliothèque python PyQt5 (Sur toute les machine Linux)

```
sudo apt install -y \
```

```
libxcb-xinerama0 \
```

```
libxkbcommon-x11-0 \
```

```
libxcb-cursor0 \
```

```
libxcb-icccm4 \
```

```
libxcb-image0 \
```

```
libxcb-keysyms1 \
```

```
libxcb-render-util0 \
```

```
libxcb-shape0 \
```

```
libxcb-xfixes0
```

1.4 Bibliothèques Python requises

1.4.1 MySQL Connector (MariaDB)

Utilisé par le master pour accéder à la base de données :

```
pip install mysql-connector-python
```

1.4.2 PyQt5 (interfaces graphiques)

Utilisé par toutes les machines pour :

- l'interface graphique du master,
- l'interface graphique des routeurs,
- l'interface graphique des clients.

```
pip install PyQt5
```

1.5 Base de données MariaDB

Installation de MariaDB (machine master)

```
sudo apt install -y mariadb-server
```

```
sudo systemctl start mariadb
```

```
sudo systemctl enable mariadb
```

Création de la base de données

```
mariadb
```

```
CREATE DATABASE SAE302;
```

```
USE SAE302;
```

```
CREATE TABLE routeurs (
    id_routeur VARCHAR(50) PRIMARY KEY,
    ip_routeur VARCHAR(50),
    port INT NOT NULL,
    cle_publique TEXT
);
```

Création de l'utilisateur de la base

```
CREATE USER 'toto'@'localhost' IDENTIFIED BY 'toto';
GRANT ALL PRIVILEGES ON SAE302.* TO 'toto'@'localhost';
FLUSH PRIVILEGES;
```

2. Installation du projet

Organisation des fichiers

- Sur la machine master :
 - master.py
- Sur la machine routeur :
 - router.py
- Sur les machines clientes :
 - client.py

Aucune modification du code n'est nécessaire pour adapter les adresses IP : tout est fourni via les interfaces graphiques ou les paramètres des commandes.

3. Lancement et utilisation

3.1 Lancement du Master

Connaitre l'adresse IP de cette machine dans le réseau

`ip a`

Puis trouver l'adresse IP de l'interface host-only.

Démarrage du master en graphique

`python3 master.py`

Renseigner les champs ip du master (ip de l'interface host only de la machine) et port du master (au choix) puis démarrer le master.

Démarrage du master en ligne de commande

`python3 master.py <ip_master> <port_master>`

exemple:

`python3 master.py 192.168.178.3 5000`

Fonctionnement :

- écoute sur toutes les interfaces réseau,
- connexion à la base MariaDB,
- réception des déclarations des routeurs,
- réponse aux requêtes des clients,
- affichage graphique des routeurs enregistrés.

3.2 Lancement d'un Routeur

Démarrage des routeurs en graphique

`python3 router.py`

Une interface graphique s'ouvre.

Informations à renseigner :

- ID du routeur
- IP de la machine hébergeant le routeur (s'obtient avec `ip a`)
- Port d'écoute du routeur
- IP du master
- Port du master

Fonctionnement :

- le routeur se déclare automatiquement au master,

- ses informations sont stockées en base,
- il écoute les messages entrants,
- il affiche le message reçu et le message relayé.

Démarrage en ligne de commande des routeurs (à faire pour chaque routeur)

```
python3 router.py <ip_VRouter> <ID_router> <port_router> <ip_master> <port_master>
```

3.3 Lancement d'un Client

Démarrage du client en graphique

```
python3 client.py
```

Une interface graphique s'ouvre.

Création du client

Champs obligatoires :

- IP du client
- Port d'écoute du client

Connexion au master

Champs :

- IP du master
- Port du master

Fonction :

- récupération de la liste des routeurs disponibles depuis la base de données.

Sélection des routeurs

- la liste des routeurs s'affiche,
- l'utilisateur sélectionne au minimum 3 routeurs,
- l'ordre de sélection définit le chemin.

Envoi d'un message

Champs :

- IP du destinataire
- Port du destinataire

- Message à envoyer

Fonction :

- encapsulation du message,
 - envoi au premier routeur,
 - transit via les routeurs sélectionnés,
 - réception par le client final.
-

4. Limitations connues

- Les routeurs arrêtés brutalement peuvent rester présents dans la base
 - Le chiffrement RSA est partiellement implémenté
 - Les clés publiques sont stockées mais non exploitées pleinement
-

5. Sécurité

- Aucun identifiant n'est codé en dur côté client ou routeur
- La base est isolée sur le master
- Le chiffrement a été étudié et testé (RSA), mais non finalisé
- Les traces de tests cryptographiques sont conservées dans le code