

DOCUMENTATION TECHNIQUE

Gestionnaire de Mots de Passe Sécurisé

Architecture RPC avec Chiffrement AES-256 et TLS 1.3

Déploiement avec Docker

Développé par :



Hasna Daoui



Nana Diawara



Projet d'Application répartie



Résumé Exécutif

Ce document présente un **gestionnaire de mots de passe sécurisé** développé en Java, mettant en œuvre une architecture client-serveur basée sur RPC (Remote Procedure Call). L'application intègre plusieurs couches de sécurité incluant le chiffrement AES-256 pour le stockage des données, le hachage SHA-256 avec salage pour les authentifications, et une communication sécurisée via SSL/TLS 1.3. Le système est conteneurisé avec Docker pour une installation simplifiée et une isolation renforcée.

Table des matières

1	Introduction	3
1.1	Objectifs pédagogiques	3
1.2	Description générale du projet	3
2	Aspects de sécurité	3
3	Prérequis et Configuration Environnementale	3
3.1	Téléchargement des Bibliothèques Java	3
3.2	Création du Keystore PKCS12	4
4	Configuration Docker	5
4.1	Installation de Docker	5
4.1.1	Prérequis	5
4.1.2	Installation sur Linux (Ubuntu/Debian)	5
4.1.3	Vérification de l'installation	5
4.1.4	Ajout de l'utilisateur au groupe Docker (optionnel)	6
5	Structure du Projet	6
6	Configuration Docker Compose	7
6.1	Présentation du fichier <code>docker-compose.yml</code>	7
6.2	Analyse de la configuration	7
6.2.1	Service <code>server</code>	7
7	Dockerfile du Client	8
7.1	Présentation du fichier	8
8	Dockerfile du Serveur	8
9	Procédure d'Exécution	9
9.1	Prérequis	9
9.2	Commandes d'Installation	9
9.3	Configuration Requise	9
9.3.1	Pour l'affichage graphique	9
9.4	Arrêt Propre	10
10	Dépannage	10
11	Architecture Technique	10
11.1	Fonctionnalités du Serveur	10
11.2	Fonctionnalités du Client	11
12	Conclusion	11

1 Introduction

Ce projet a pour objectif de développer une application Java permettant de gérer des mots de passe personnels de manière sécurisée. L'application repose sur le modèle client-serveur utilisant la technologie RPC (Remote Procedure Call), avec une interface graphique conviviale développée en Java.

1.1 Objectifs pédagogiques

- Comprendre les principes des appels de procédures distantes (RPC) en Java
- Concevoir une interface graphique simple et intuitive à l'aide de JavaFX ou Swing
- Intégrer des mécanismes de sécurité de base : authentification, chiffrement, autorisation
- Appréhender les vulnérabilités possibles des communications réseau

1.2 Description générale du projet

L'application permet à l'utilisateur de se connecter à l'aide d'un nom d'utilisateur et d'un mot de passe, puis d'accéder à une interface pour consulter, ajouter, modifier ou rechercher des mots de passe associés à différents comptes (réseaux sociaux, emails, etc.). Le serveur distant se charge de stocker et gérer ces données.

2 Aspects de sécurité

Attention : La sécurité est un aspect critique de cette application. Tous les mécanismes suivants doivent être correctement configurés.

Le projet inclut des éléments fondamentaux de sécurité, tels que :

- Le hachage des mots de passe utilisateurs (utilisation d'un algorithme comme SHA-256)
- Une connexion simulée de manière sécurisée (type HTTPS basique)
- Une validation des données saisies par les utilisateurs
- Une gestion simple de sessions pour maintenir l'état de la connexion

3 Prérequis et Configuration Environnementale

3.1 Téléchargement des Bibliothèques Java

Pour assurer le bon fonctionnement de l'application, il est nécessaire de récupérer les fichiers JAR suivants :

- **GSON** : bibliothèque pour la manipulation JSON

```
curl -O https://repo1.maven.org/maven2/com/google/code/gson/gson/2.10.1/gson-2.10.1.jar
```

— **SQLite JDBC** : pilote JDBC pour la base de données SQLite

```
curl -O https://repo1.maven.org/maven2/org/xerial/sqlite-jdbc/3.42.0.0/sqlite-jdbc-3.42.0.0.jar
```

```
g1@ubuntu:~/rpc-docker/data$ sudo apt install curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libcurl4
The following NEW packages will be installed:
  curl libcurl4
0 upgraded, 2 newly installed, 0 to remove and 600 not upgraded.
Need to get 396 kB of archives.
After this operation, 1,129 kB of additional disk space will be used.
Do you want to continue? [y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 libcurl4 amd64 7.68.0-1ubuntu2.25 [235 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 curl amd64 7.68.0-1ubuntu2.25 [162 kB]
Fetched 396 kB in 2s (251 kB/s)
Selecting previously unselected package libcurl4:amd64.
(Reading database ... 156117 files and directories currently installed.)
Preparing to unpack .../libcurl4_7.68.0-1ubuntu2.25_amd64.deb ...
Unpacking libcurl4:amd64 (7.68.0-1ubuntu2.25) ...
Selecting previously unselected package curl.
Preparing to unpack .../curl_7.68.0-1ubuntu2.25_amd64.deb ...
Unpacking curl (7.68.0-1ubuntu2.25) ...
Setting up libcurl4:amd64 (7.68.0-1ubuntu2.25) ...
Setting up curl (7.68.0-1ubuntu2.25) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9) ...
g1@ubuntu:~/rpc-docker/data$ curl -O https://repo1.maven.org/maven2/com/google/gson/gson/2.10.1/gson-2.10.1.jar
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 276k 100 276k 0 741k 0 --:--:-- --:--:-- --:--:-- 741k
g1@ubuntu:~/rpc-docker/data$ curl -O https://repo1.maven.org/maven2/org/xerial/sqlite-jdbc/3.42.0.0/sqlite-jdbc-3.42.0.0.jar
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 12.4M 100 12.4M 0 7213k 0 0:00:01 0:00:01 --:--:-- 7209k
g1@ubuntu:~/rpc-docker/data$ openssl req -x509 -newkey rsa:2048 -keyout server.key -out server.crt -days 365 -nodes -subj "/C=MA/ST=Taroudant/L=Kchachda/O=CyberSecurity/OU=LocalDev/CN=172.16.101.128"
Generating a RSA private key
.....+++++
writing new private key to 'server.key'
.....
g1@ubuntu:~/rpc-docker/data$ openssl pkcs12 -export -inkey server.key -in server.crt -out keystore.p12 -name server
Enter Export Password:
Verifying - Enter Export Password:
```

FIGURE 1 – Installation des bibliothèques GSON et SQLite

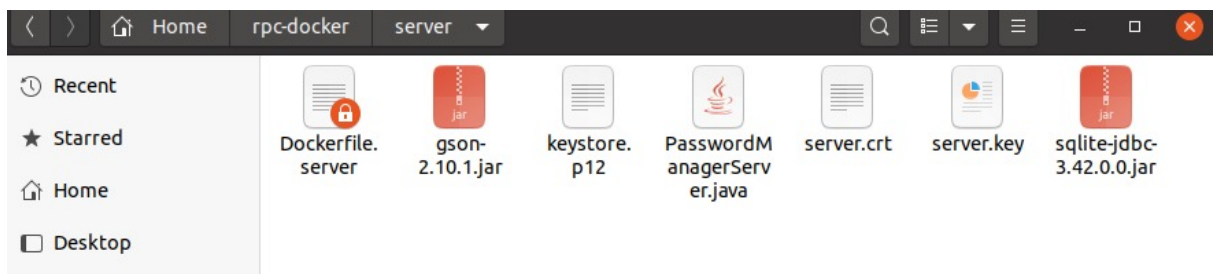


FIGURE 2 – Emplacement des fichiers

Note importante : Les fichiers JAR doivent être placés dans le même répertoire que le serveur pour qu'ils soient accessibles lors de l'exécution.

3.2 Création du Keystore PKCS12

Le keystore est indispensable pour configurer une connexion sécurisée via SSL/TLS. Il est créé à partir d'un certificat auto-signé généré avec OpenSSL.

1. Génération de la clé privée et du certificat auto-signé :

```
openssl req -x509 -newkey rsa:2048 -keyout server.key -out
server.crt \
-days 365 -nodes \
-subj "/C=MA/ST=Taroudant/L=Kchachda/O=CyberSecurity/OU=
LocalDev/CN=172.16.101.128"
```

2. Création du keystore au format PKCS12 à partir de la clé et du certificat :

```
openssl pkcs12 -export -inkey server.key -in server.crt \
-out keystore.p12 -name server
```

Lors de cette étape, un mot de passe d'export est demandé. Ce mot de passe protégera le keystore (exemple utilisé : **cyber**).

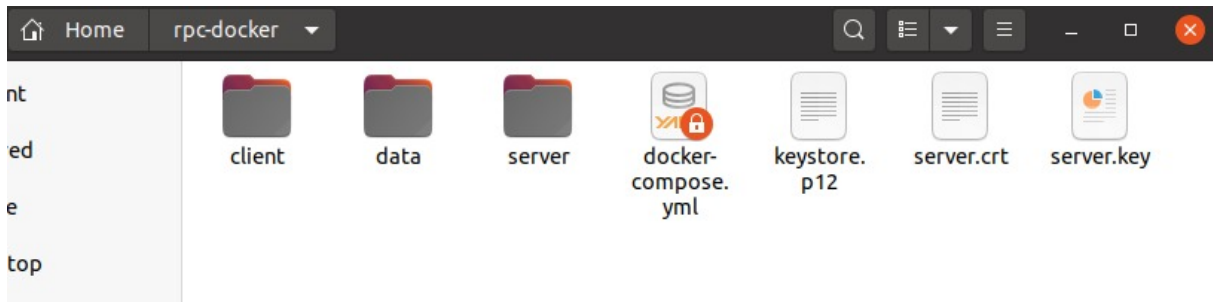


FIGURE 3 – Fichiers SSL générés

4 Configuration Docker

4.1 Installation de Docker

4.1.1 Prérequis

- Système Linux, Windows ou macOS
- Compte Docker Hub (facultatif pour les téléchargements publics)

4.1.2 Installation sur Linux (Ubuntu/Debian)

```
sudo apt update
sudo apt install docker.io -y
sudo systemctl enable docker
sudo systemctl start docker
sudo apt install docker-compose
```

4.1.3 Vérification de l'installation

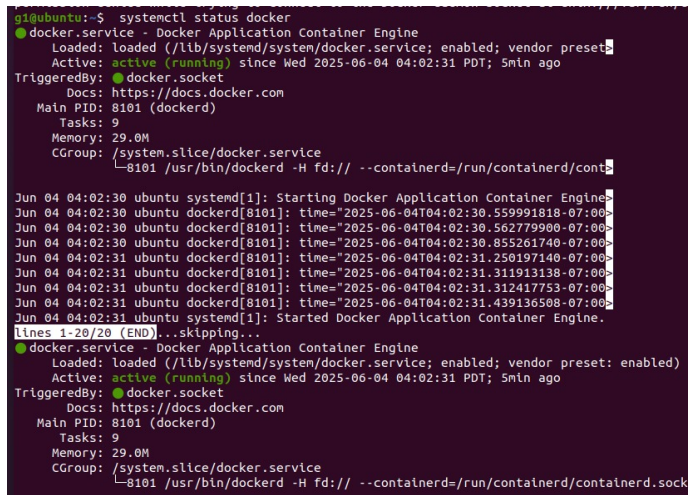
```
# Vérifier la version de Docker
docker version

# Vérifier l'état du moteur Docker
docker info

# Vérifier que Docker est actif
sudo systemctl status docker
```

4.1.4 Ajout de l'utilisateur au groupe Docker (optionnel)

```
sudo usermod -aG docker $USER
```



```
g1@ubuntu:~$ systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2025-06-04 04:02:31 PDT; 5min ago
     TriggeredBy: ● docker.socket
    Docs: https://docs.docker.com
   Main PID: 8101 (dockerd)
      Tasks: 9
     Memory: 29.0M
    CGroup: /system.slice/docker.service
            └─8101 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/cont

Jun 04 04:02:30 ubuntu systemd[1]: Starting Docker Application Container Engine:
Jun 04 04:02:30 ubuntu dockerd[8101]: time="2025-06-04T04:02:30.559991818-07:00"
Jun 04 04:02:30 ubuntu dockerd[8101]: time="2025-06-04T04:02:30.562779900-07:00"
Jun 04 04:02:30 ubuntu dockerd[8101]: time="2025-06-04T04:02:30.855261740-07:00"
Jun 04 04:02:31 ubuntu dockerd[8101]: time="2025-06-04T04:02:31.250197140-07:00"
Jun 04 04:02:31 ubuntu dockerd[8101]: time="2025-06-04T04:02:31.311913138-07:00"
Jun 04 04:02:31 ubuntu dockerd[8101]: time="2025-06-04T04:02:31.312417753-07:00"
Jun 04 04:02:31 ubuntu dockerd[8101]: time="2025-06-04T04:02:31.439136508-07:00"
Jun 04 04:02:31 ubuntu systemd[1]: Started Docker Application Container Engine.
lines 1-20/20 (END) ...skipping...
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2025-06-04 04:02:31 PDT; 5min ago
     TriggeredBy: ● docker.socket
    Docs: https://docs.docker.com
   Main PID: 8101 (dockerd)
      Tasks: 9
     Memory: 29.0M
    CGroup: /system.slice/docker.service
            └─8101 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/sock
```

FIGURE 4 – Statut de Docker

5 Structure du Projet

Le projet est organisé selon l'arborescence suivante :

```
password-manager/
├── client/
│   ├── PasswordManagerClient.java
│   ├── Dockerfile.client
│   ├── gson-2.10.1.jar
│   └── sqlite-jdbc-3.42.0.0.jar
├── server/
│   ├── PasswordManagerServer.java
│   ├── Dockerfile.server
│   ├── keystore.p12
│   ├── gson-2.10.1.jar
│   └── sqlite-jdbc-3.42.0.0.jar
├── data/
├── docker-compose.yml
└── README.md
```

6 Configuration Docker Compose

6.1 Présentation du fichier `docker-compose.yml`

Le fichier `docker-compose.yml` permet d'orchestrer le déploiement d'une application client-serveur à l'aide de Docker. Voici le code complet :

```
services:
  server:
    build:
      context: ./server
      dockerfile: Dockerfile.server
    ports:
      - "8443:8443"
    volumes:
      - ./data:/app/data
      - /tmp/.X11-unix:/tmp/.X11-unix
    environment:
      - DB_URL=jdbc:sqlite:/app/data/password_manager.db
      - DISPLAY=${DISPLAY}
    networks:
      - password-manager-net
    tty: true

  client:
    build:
      context: ./client
      dockerfile: Dockerfile.client
    depends_on:
      - server
    volumes:
      - /tmp/.X11-unix:/tmp/.X11-unix
    environment:
      - DISPLAY=${DISPLAY}
    networks:
      - password-manager-net
    tty: true

networks:
  password-manager-net:
    driver: bridge
```

6.2 Analyse de la configuration

6.2.1 Service `server`

- **Construction** : Utilise le Dockerfile situé dans `./server/Dockerfile.server`
- **Ports** : Exposition du port 8443 (généralement utilisé pour HTTPS alternatif)
- **Volumes** :
 - Persistance des données via `./data:/app/data`
 - Partage du socket X11 pour l'affichage graphique

- Environnement :
 - Configuration de la base de données SQLite
 - Variable DISPLAY pour l'interface graphique

7 Dockerfile du Client

7.1 Présentation du fichier

Le fichier `Dockerfile.client` constitue la configuration de construction de l'image Docker pour le client de notre application. Voici son contenu complet :

```
FROM openjdk:17-jdk-slim

WORKDIR /app

RUN apt-get update && apt-get install -y \
    libfreetype6 \
    libxext6 \
    libxrender1 \
    libxtst6 \
    libxi6 \
    libxrandr2 \
    libxinerama1 \
    libx11-6 \
    fontconfig \
    fonts-dejavu-core \
    && rm -rf /var/lib/apt/lists/*

COPY . /app

RUN javac -cp "gson-2.10.1.jar:sqlite-jdbc-3.42.0.0.jar"
    PasswordManagerClient.java

CMD ["java", "-cp", ".:gson-2.10.1.jar:sqlite-jdbc-3.42.0.0.jar",
    "PasswordManagerClient"]
```

8 Dockerfile du Serveur

```
FROM eclipse-temurin:17-jdk

# Install GUI libraries
RUN apt-get update && apt-get install -y \
    libxext6 \
    libxrender1 \
    libxtst6 \
    libxi6 \
    libxrandr2 \
    libxinerama1 \
    libx11-6 \
    && rm -rf /var/lib/apt/lists/*
```



```
WORKDIR /app

COPY PasswordManagerServer.java .
COPY gson-2.10.1.jar .
COPY sqlite-jdbc-3.42.0.0.jar .
COPY keystore.p12 /app/keystore.p12

# Compile the server GUI app
RUN javac -cp "gson-2.10.1.jar:sqlite-jdbc-3.42.0.0.jar"
    PasswordManagerServer.java

# Run the server GUI app
CMD ["java", "-cp", ".:gson-2.10.1.jar:sqlite-jdbc-3.42.0.0.jar",
    "PasswordManagerServer"]
```

9 Procédure d'Exécution

9.1 Prérequis

- Docker et Docker Compose installés
- Système Linux/X11 (pour l’affichage des interfaces graphiques)
- Au moins 2GB de RAM disponible

9.2 Commandes d'Installation

```
# 1. Construire les images
docker compose build

# 2. Démarrer les services
docker compose up
```

```

root@kali:~/rpm-builder# rpm-builder --help
Usage: rpm-builder [options]
Options:
  -h, --help            display this help message
  -c, --config FILENAME configuration file
  -d, --debug            enable debugging
  -e, --error FILENAME   error log file
  -f, --force            force installation
  -i, --install          install the package
  -l, --list             list the packages
  -n, --name NAME        package name
  -p, --path PATH        path to the package
  -r, --rpm-rpms RPMS    rpm files
  -s, --source SRCS       source files
  -t, --target TARGET     target architecture
  -v, --verbose          verbose output
  -w, --work-dir DIR     working directory
  -x, --x86_64           x86_64 architecture
  -y, --yes              assume yes to all questions
  -z, --zstd             use zstd for compression
  --rpm-arch ARCH        rpm architecture
  --rpm-flags FLAGS      rpm flags
  --rpm-headers HEADERS  rpm headers
  --rpm-libs LIBS        rpm libs
  --rpm-libs64 LIBS64    rpm libs64
  --rpm-libs70 LIBS70     rpm libs70
  --rpm-libs80 LIBS80     rpm libs80
  --rpm-libs90 LIBS90     rpm libs90
  --rpm-libs100 LIBS100   rpm libs100
  --rpm-libs110 LIBS110   rpm libs110
  --rpm-libs120 LIBS120   rpm libs120
  --rpm-libs130 LIBS130   rpm libs130
  --rpm-libs140 LIBS140   rpm libs140
  --rpm-libs150 LIBS150   rpm libs150
  --rpm-libs160 LIBS160   rpm libs160
  --rpm-libs170 LIBS170   rpm libs170
  --rpm-libs180 LIBS180   rpm libs180
  --rpm-libs190 LIBS190   rpm libs190
  --rpm-libs200 LIBS200   rpm libs200
  --rpm-libs210 LIBS210   rpm libs210
  --rpm-libs220 LIBS220   rpm libs220
  --rpm-libs230 LIBS230   rpm libs230
  --rpm-libs240 LIBS240   rpm libs240
  --rpm-libs250 LIBS250   rpm libs250
  --rpm-libs260 LIBS260   rpm libs260
  --rpm-libs270 LIBS270   rpm libs270
  --rpm-libs280 LIBS280   rpm libs280
  --rpm-libs290 LIBS290   rpm libs290
  --rpm-libs300 LIBS300   rpm libs300
  --rpm-libs310 LIBS310   rpm libs310
  --rpm-libs320 LIBS320   rpm libs320
  --rpm-libs330 LIBS330   rpm libs330
  --rpm-libs340 LIBS340   rpm libs340
  --rpm-libs350 LIBS350   rpm libs350
  --rpm-libs360 LIBS360   rpm libs360
  --rpm-libs370 LIBS370   rpm libs370
  --rpm-libs380 LIBS380   rpm libs380
  --rpm-libs390 LIBS390   rpm libs390
  --rpm-libs400 LIBS400   rpm libs400
  --rpm-libs410 LIBS410   rpm libs410
  --rpm-libs420 LIBS420   rpm libs420
  --rpm-libs430 LIBS430   rpm libs430
  --rpm-libs440 LIBS440   rpm libs440
  --rpm-libs450 LIBS450   rpm libs450
  --rpm-libs460 LIBS460   rpm libs460
  --rpm-libs470 LIBS470   rpm libs470
  --rpm-libs480 LIBS480   rpm libs480
  --rpm-libs490 LIBS490   rpm libs490
  --rpm-libs500 LIBS500   rpm libs500
  --rpm-libs510 LIBS510   rpm libs510
  --rpm-libs520 LIBS520   rpm libs520
  --rpm-libs530 LIBS530   rpm libs530
  --rpm-libs540 LIBS540   rpm libs540
  --rpm-libs550 LIBS550   rpm libs550
  --rpm-libs560 LIBS560   rpm libs560
  --rpm-libs570 LIBS570   rpm libs570
  --rpm-libs580 LIBS580   rpm libs580
  --rpm-libs590 LIBS590   rpm libs590
  --rpm-libs600 LIBS600   rpm libs600
  --rpm-libs610 LIBS610   rpm libs610
  --rpm-libs620 LIBS620   rpm libs620
  --rpm-libs630 LIBS630   rpm libs630
  --rpm-libs640 LIBS640   rpm libs640
  --rpm-libs650 LIBS650   rpm libs650
  --rpm-libs660 LIBS660   rpm libs660
  --rpm-libs670 LIBS670   rpm libs670
  --rpm-libs680 LIBS680   rpm libs680
  --rpm-libs690 LIBS690   rpm libs690
  --rpm-libs700 LIBS700   rpm libs700
  --rpm-libs710 LIBS710   rpm libs710
  --rpm-libs720 LIBS720   rpm libs720
  --rpm-libs730 LIBS730   rpm libs730
  --rpm-libs740 LIBS740   rpm libs740
  --rpm-libs750 LIBS750   rpm libs750
  --rpm-libs760 LIBS760   rpm libs760
  --rpm-libs770 LIBS770   rpm libs770
  --rpm-libs780 LIBS780   rpm libs780
  --rpm-libs790 LIBS790   rpm libs790
  --rpm-libs800 LIBS800   rpm libs800
  --rpm-libs810 LIBS810   rpm libs810
  --rpm-libs820 LIBS820   rpm libs820
  --rpm-libs830 LIBS830   rpm libs830
  --rpm-libs840 LIBS840   rpm libs840
  --rpm-libs850 LIBS850   rpm libs850
  --rpm-libs860 LIBS860   rpm libs860
  --rpm-libs870 LIBS870   rpm libs870
  --rpm-libs880 LIBS880   rpm libs880
  --rpm-libs890 LIBS890   rpm libs890
  --rpm-libs900 LIBS900   rpm libs900
  --rpm-libs910 LIBS910   rpm libs910
  --rpm-libs920 LIBS920   rpm libs920
  --rpm-libs930 LIBS930   rpm libs930
  --rpm-libs940 LIBS940   rpm libs940
  --rpm-libs950 LIBS950   rpm libs950
  --rpm-libs960 LIBS960   rpm libs960
  --rpm-libs970 LIBS970   rpm libs970
  --rpm-libs980 LIBS980   rpm libs980
  --rpm-libs990 LIBS990   rpm libs990
  --rpm-libs1000 LIBS1000  rpm libs1000
  --rpm-libs1010 LIBS1010 rpm libs1010
  --rpm-libs1020 LIBS1020 rpm libs1020
  --rpm-libs1030 LIBS1030 rpm libs1030
  --rpm-libs1040 LIBS1040 rpm libs1040
  --rpm-libs1050 LIBS1050 rpm libs1050
  --rpm-libs1060 LIBS1060 rpm libs1060
  --rpm-libs1070 LIBS1070 rpm libs1070
  --rpm-libs1080 LIBS1080 rpm libs1080
  --rpm-libs1090 LIBS1090 rpm libs1090
  --rpm-libs1100 LIBS1100 rpm libs1100
  --rpm-libs1110 LIBS1110 rpm libs1110
  --rpm-libs1120 LIBS1120 rpm libs1120
  --rpm-libs1130 LIBS1130 rpm libs1130
  --rpm-libs1140 LIBS1140 rpm libs1140
  --rpm-libs1150 LIBS1150 rpm libs1150
  --rpm-libs1160 LIBS1160 rpm libs1160
  --rpm-libs1170 LIBS1170 rpm libs1170
  --rpm-libs1180 LIBS1180 rpm libs1180
  --rpm-libs1190 LIBS1190 rpm libs1190
  --rpm-libs1200 LIBS1200 rpm libs1200
  --rpm-libs1210 LIBS1210 rpm libs1210
  --rpm-libs1220 LIBS1220 rpm libs1220
  --rpm-libs1230 LIBS1230 rpm libs1230
  --rpm-libs1240 LIBS1240 rpm libs1240
  --rpm-libs1250 LIBS1250 rpm libs1250
  --rpm-libs1260 LIBS1260 rpm libs1260
  --rpm-libs1270 LIBS1270 rpm libs1270
  --rpm-libs1280 LIBS1280 rpm libs1280
  --rpm-libs1290 LIBS1290 rpm libs1290
  --rpm-libs1300 LIBS1300 rpm libs1300
  --rpm-libs1310 LIBS1310 rpm libs1310
  --rpm-libs1320 LIBS1320 rpm libs1320
  --rpm-libs1330 LIBS1330 rpm libs1330
  --rpm-libs1340 LIBS1340 rpm libs1340
  --rpm-libs1350 LIBS1350 rpm libs1350
  --rpm-libs1360 LIBS1360 rpm libs1360
  --rpm-libs1370 LIBS1370 rpm libs1370
  --rpm-libs1380 LIBS1380 rpm libs1380
  --rpm-libs1390 LIBS1390 rpm libs1390
  --rpm-libs1400 LIBS1400 rpm libs1400
  --rpm-libs1410 LIBS1410 rpm libs1410
  --rpm-libs1420 LIBS1420 rpm libs1420
  --rpm-libs1430 LIBS1430 rpm libs1430
  --rpm-libs1440 LIBS1440 rpm libs1440
  --rpm-libs1450 LIBS1450 rpm libs1450
  --rpm-libs1460 LIBS1460 rpm libs1460
  --rpm-libs1470 LIBS1470 rpm libs1470
  --rpm-libs1480 LIBS1480 rpm libs1480
  --rpm-libs1490 LIBS1490 rpm libs1490
  --rpm-libs1500 LIBS1500 rpm libs1500
  --rpm-libs1510 LIBS1510 rpm libs1510
  --rpm-libs1520 LIBS1520 rpm libs1520
  --rpm-libs1530 LIBS1530 rpm libs1530
  --rpm-libs1540 LIBS1540 rpm libs1540
  --rpm-libs1550 LIBS1550 rpm libs1550
  --rpm-libs1560 LIBS1560 rpm libs1560
  --rpm-libs1570 LIBS1570 rpm libs1570
  --rpm-libs1580 LIBS1580 rpm libs1580
  --rpm-libs1590 LIBS1590 rpm libs1590
  --rpm-libs1600 LIBS1600 rpm libs1600
  --rpm-libs1610 LIBS1610 rpm libs1610
  --rpm-libs1620 LIBS1620 rpm libs1620
  --rpm-libs1630 LIBS1630 rpm libs1630
  --rpm-libs1640 LIBS1640 rpm libs1640
  --rpm-libs1650 LIBS1650 rpm libs1650
  --rpm-libs1660 LIBS1660 rpm libs1660
  --rpm-libs1670 LIBS1670 rpm libs1670
  --rpm-libs1680 LIBS1680 rpm libs1680
  --rpm-libs1690 LIBS1690 rpm libs1690
  --rpm-libs1700 LIBS1700 rpm libs1700
  --rpm-libs1710 LIBS1710 rpm libs1710
  --rpm-libs1720 LIBS1720 rpm libs1720
  --rpm-libs1730 LIBS1730 rpm libs1730
  --rpm-libs1740 LIBS1740 rpm libs1740
  --rpm-libs1750 LIBS1750 rpm libs1750
  --rpm-libs1760 LIBS1760 rpm libs1760
  --rpm-libs1770 LIBS1770 rpm libs1770
  --rpm-libs1780 LIBS1780 rpm libs1780
  --rpm-libs1790 LIBS1790 rpm libs1790
  --rpm-libs1800 LIBS1800 rpm libs1800
  --rpm-libs1810 LIBS1810 rpm libs1810
  --rpm-libs1820 LIBS1820 rpm libs1820
  --rpm-libs1830 LIBS1830 rpm libs1830
  --rpm-libs1840 LIBS1840 rpm libs1840
  --rpm-libs1850 LIBS1850 rpm libs1850
  --rpm-libs1860 LIBS1860 rpm libs1860
  --rpm-libs1870 LIBS1870 rpm libs1870
  --rpm-libs1880 LIBS18
```

FIGURE 5 – Processus de construction Docker

9.3 Configuration Requisite

9.3.1 Pour l’affichage graphique

Autoriser les connexions X11 depuis Docker :

```

gi@ubuntu:~/rpc-docker$ sudo docker-compose up
[+] Running 2/2
✓ Container rpc-docker-server-1 Recreated
✓ Container rpc-docker-client-1 Created
Attaching to client-1, server-1
server-1 | [SERVER] Registered: hasna
server-1 | [SERVER] Logged in: hasna
server-1 | [SERVER] [hasna] Listed accounts
server-1 | [SERVER] [hasna] Created account: linkedin
server-1 | [SERVER] [hasna] Listed accounts
server-1 | [SERVER] [hasna] Updated account ID 1
server-1 | [SERVER] [hasna] Listed accounts
server-1 | [SERVER] [hasna] Retrieved password for account ID 1
server-1 | [SERVER] [hasna] Listed accounts
server-1 | [SERVER] [hasna] Created account: hh
server-1 | [SERVER] [hasna] Listed accounts
server-1 | [SERVER] [hasna] Retrieved password for account ID 2

```

FIGURE 6 – Démarrage des services

```
xhost +local:docker
```

9.4 Arrêt Propre

Pour arrêter les services et nettoyer :

```

docker compose down
xhost -local:docker # Retire les permissions X11

```

10 Dépannage

Problème	Solution
Erreurs X11	Vérifier que <code>xhost +local:docker</code> a été exécuté
Port 8443 occupé	Modifier le port dans <code>docker-compose.yml</code>
Certificat manquant	Générer un keystore avec OpenSSL selon la procédure décrite
Problèmes de permissions	Ajouter l'utilisateur au groupe docker : <code>sudo usermod -aG docker \$USER</code>

11 Architecture Technique

11.1 Fonctionnalités du Serveur

- Écoute sur le port 8443 en SSL/TLS
- Stockage des données dans une base SQLite
- Interface d'administration via GUI Swing
- Gestion des connexions clientes via des threads séparés
- Journalisation des activités en temps réel

11.2 Fonctionnalités du Client

- Interface utilisateur moderne avec thème personnalisé
- Connexion SSL/TLS avec certificat auto-signé
- Authentification sécurisée (login/enregistrement)
- Gestion CRUD complète des comptes
- Affichage sécurisé des mots de passe

12 Conclusion

Ce projet constitue un exercice complet permettant de mettre en œuvre des connaissances en Java réseau, interfaces graphiques et sécurité informatique. Il offre une base concrète pour comprendre comment développer des applications distribuées tout en respectant les bonnes pratiques de sécurité.

L'utilisation de Docker facilite grandement le déploiement et garantit la reproductibilité de l'environnement d'exécution sur différents systèmes.