Documentation de mise en œuvre ESUP-SGC / ESUP-NFC-TAG

Introduction

Cette application permet de gérer le cycle de vie des cartes NFC de votre établissement, de la demande à sa désactivation en passant par son impression, encodage et activation dans votre système d'information.

Cette application fonctionne via une authentification/identification Shibboleth et en lien avec une instance esup-nfc-tag-server

Une Machine Virtuelle est à votre disposition pour pouvoir manipuler ESUP-SGC et ainsi avoir à disposition un exemple complet d'installation : VM ESUP-SGC.

Concepts

Afin de mettre en œuvre le système de gestion de carte ESUP-SGC, vous devez installer ESUP-SGC mais également ESUP-NFC-TAG.

ESUP-SGC

- Gère le cycle de vie de vos cartes (workflow allant de la demande à désactivation de la carte)
- Authentifie les utilisateurs via une authentification Shibboleth
- Récupère les informations (nom, prénom, email, date de naissance, etc.) des utilisateurs depuis l'identification shibboleth ou/et un annuaire LDAP ou/et une base de données relationnelle
- Synchronise les informations utilisateurs et de cartes avec l'API du CNOUS (CROUS, IZLY) en temps réel.
- Synchronise les informations utilisateurs étudiants et de cartes avec l'API ESC - European Student Card : http://europeanstudentcard.eu
- Synchronisation avec différentes solutions de contrôles d'accès (P2S, TIL, Synchronic)
- Reversement des informations de cartes (csn, 'identifiant desfire', photo) dans un annuaire LDAP
- Permet l'impression des cartes (en mode PS/PCL)

esup-sgc ne sait pas encoder les cartes, la partie technique propre au NFC est propre à ESUP-NFC-TAG

ESUP-NFC-TAG

ESUP-NFC-TAG (https://www.esup-portail.org/wiki/display/ESUPNFC) se décompose d'une partie serveur (implémentant toute la logique métier, la gestion des périphériques et applications clientes ainsi que les algorithmes de chiffrement Mifare Desfire) nommé ESUP-NFC-TAG-SERVER qui :

- Permet la lecture et l'écriture des cartes en NFC - technologie **Mifare Desfire**
- Gère des applications clientes (de lecture notamment mais aussi écriture ou/et mise à jour pour ESUP-SGC)
- Gère les périphériques sans contact (lecteur PC/SC, smartphone, …)

C'est esup-nfc-tag-server qui connait la structure de la carte Mifare Desfire. Il communique avec esup-sgc pour récupérer le contenu à écrire dans les fichiers (ex : code d'identification pour le contrôle d'accès)

Pour lire ou encoder une carte, un "client lourd" est nécessaire ; aussi les applications clientes suivantes sont disponibles :

- esup-nfc-tag-desktop : application java qui permet d'utiliser un encodeur PC/SC USB depuis une machine linux/windows/mac disposant de Java
- esup-nfc-tag-droid : application Android qui utilise le lecteur NFC du smartphone
- esup-nfc-tag-keyboard : application java qui permet d'émuler des frappes clavier permettant ainsi de saisir l'identifiant du porteur de carte par exemple

ESUP-SGC-CLIENT

ESUP-SGC inclut également son propre client pour l'encodage des cartes (esup-sgc-client) qui est en fait également client d'esup-nfc-tag-server.

L'encodage via esup-sgc-client se passe de cette manière :

- Edition en 2 temps :
  - lecture du QR code présent sur la carte (identifiant l'individu à encoder) ; la carte ayant été imprimée depuis l'interface web d'esup-sgc via un navigateur web
  - association de la carte avec l'individu dans le SGC
  - encodage des différentes applications DESFIRE telles que décrites dans le fichier de configuration d'ESUP-NFC-TAG-SERVER
  - ... et optionnellement (on recommande plutôt d'acheter des cartes pré-encodés IZLY) encodage de l'application IZLY (CROUS) via ESUP-NFC-TAG-SERVER en utilisant la clé SAM du CNOUS et des DLL Windows
- Edition en 1 temps :
  - sélection de la carte à imprimer et encoder depuis l'interface web d'esup-sgc via un navigateur web
  - impression de la carte
  - association de la carte avec l'individu dans le SGC et encodage des différentes applications DESFIRE telles que décrites dans le fichier de configuration d'ESUP-NFC-TAG-SERVER
  - ... et optionnellement (on recommande plutôt d'acheter des cartes pré-encodés IZLY) encodage de l'application IZLY (CROUS) via ESUP-NFC-TAG-SERVER en utilisant la clé SAM du CNOUS et des DLL Windows

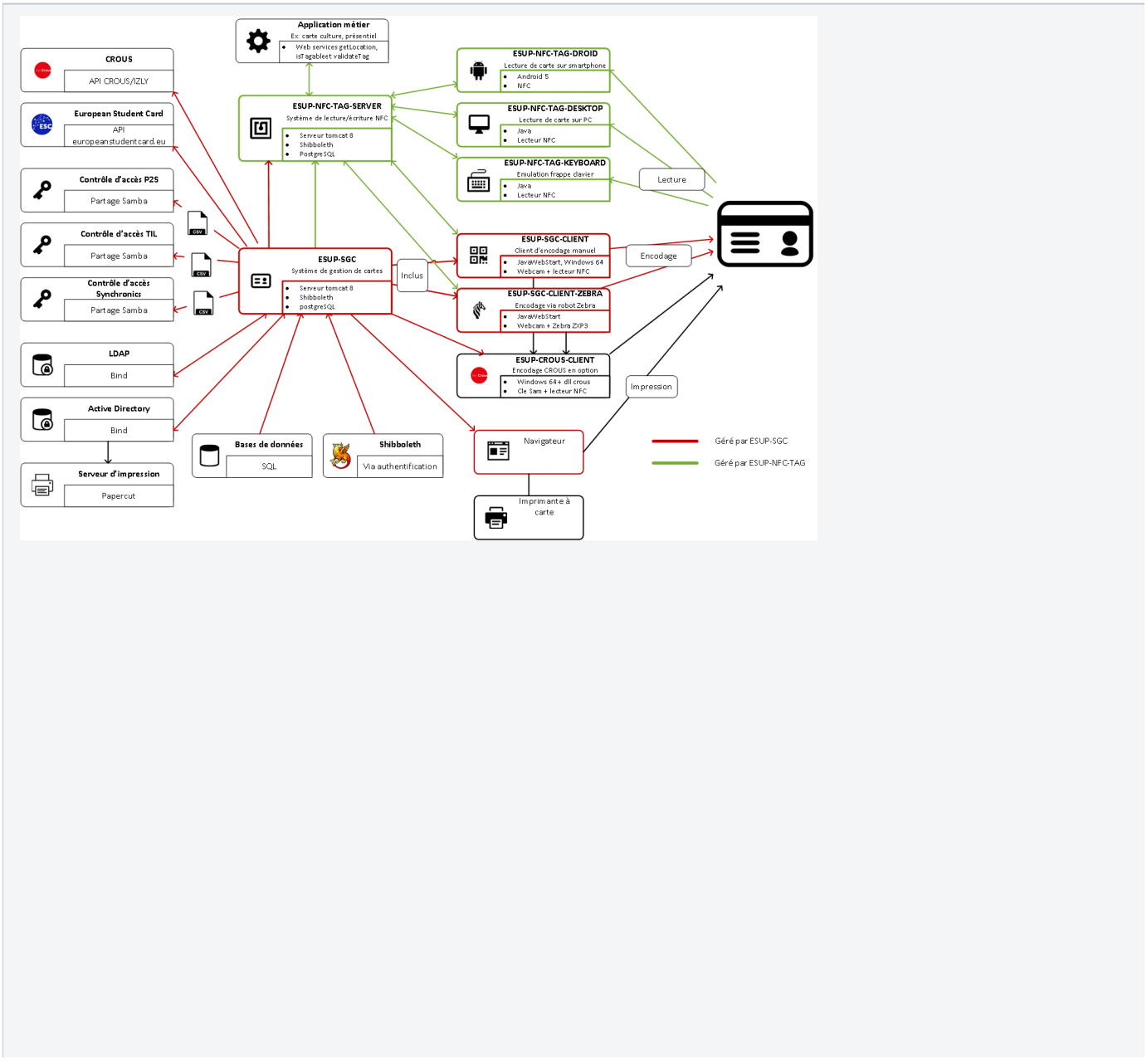


Schéma d'architecture de la solution

Pre-requis

Logiciels

- Java : nous recommandons l'usage d'openjdk
    - pour la partie serveur les versions 8 ou 11 ou **17** (généralement présentes dans les distributions) conviennent,
    - pour la partie cliente vous pouvez utiliser openjdk et openjfx en version 17.
- Maven (dernière version 3.x) : le mieux est de l'installer via le système de paquets de votre linux - http://maven.apache.org/download.cgi
- Postgresql : le mieux est de l'installer via le système de paquets de votre linux.
- Tomcat 8.5 ou **9** (Tomcat 10 n'est **pas** supporté) : https://tomcat.apache.org/download-90.cgi
- Apache + libapache2-mod-shib2
- Git

Les deux applications serveurs doivent être « shibbolethisées » - voir la documentation renater : https://services.renater.fr/federation/docs/installation/sp

Matériel

Voici le matériel minimal requis pour pouvoir mettre en place un Système de Gestion de cartes via l'environnement ESUP-SGC.

- Serveur : 2 CPU, RAM > 2 Go, Disque > 20 Go
- Cartes Mifare Desfire EV1 ou EV2
- Edition 2 temps :
    - Un lecteur RFID USB Compatible PC/SC pour encodage
    - Une webcam
    - Une imprimante à carte.
- Edition 1 temps :
    - Une imprimante à carte evolis (primacy) / zebra (zc300) avec lecteur NFC

A cela, un Smartphone Android > 5 avec lecteur NFC peut également s'avérer utile par exemple.

Installation des pré-requis

Note:
Les utilisateurs, chemins d'installation, ports utilisés ci-dessous ne sont qu'une suggestion.
Les exemples de configuration système sont basés sur Debian.

Les deux services seront installés sur le même serveur, l'un répondant avec le nom esup-sgc.univ-ville.fr et l'autre avec le nom esup-nfc-tag.univ-ville.fr. Ces VirtualHosts seront configurés sous Apache.

Installer les paquets nécessaires

```
apt-get install wget apache2 libapache2-mod-shib git
apt-get install postgresql postgresql-contrib
```

Il est également nécessaire d'avoir un JDK d'installé (**OpenJDK** [installation par paquet]).

De même il vous faudra maven, que vous pouvez soit installer par paquet (apt-get install maven), soit manuellement depuis http://maven.apache.org/download.cgi

Installation des instances Tomcat

Création de l'utilisateur esup:

```
groupadd esup
useradd -g esup -m esup
```

Installer les deux instances de Tomcat. L'une sera utilisée pour ESUP-SGC, l'autre pour ESUP-NFC-TAG.

```
cd /opt/
wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.58/bin/apache-tomcat-9.0.58.tar.gz
tar xzvf apache-tomcat-9.0.58.tar.gz
mv apache-tomcat-9.0.58 apache-tomcat-9.0.58-esup-nfc-tag

tar xzvf apache-tomcat-9.0.58.tar.gz
mv apache-tomcat-9.0.58 apache-tomcat-9.0.58-esup-sgc

ln -s apache-tomcat-9.0.58-esup-nfc-tag tomcat-esup-nfc-tag
ln -s apache-tomcat-9.0.58-esup-sgc tomcat-esup-sgc

rm -Rf /opt/tomcat-esup-sgc/webapps/*
mkdir /opt/tomcat-esup-sgc/webapps/ROOT/
rm -Rf /opt/tomcat-esup-nfc-tag/webapps/*
mkdir /opt/tomcat-esup-nfc-tag/webapps/ROOT/

chown -R esup:esup /opt/apache-tomcat-9.0.58-esup-sgc/
chown -R esup:esup /opt/apache-tomcat-9.0.58-esup-nfc-tag/
```

Note : Le contenu du dossier webapps est impérativement à supprimer, si vous ne le faites pas l'interface d'administration d'esup-sgc ne s'affichera pas. l'URL /manager rentrera en conflit avec la webapp manager de tomcat livrée par défaut lors du déploiement du serveur tomcat.

Les options Java étant les mêmes pour les deux instances, on peut créer un fichier commun qui sera lu lors du démarrage des tomcat.

```
cat > /opt/esup-env <<EOF
#!/bin/sh
JAVA_HOME=/opt/jdk
#GRADLE_HOME=/usr/local/gradle-2.14.1
#ANDROID_HOME=/usr/local/android-sdk
JAVA_OPTS="-Xms256m -Xmx512m"
EOF
```

Note: si vous devez utiliser des certificats dont les autorités de certifications ne sont pas reconnus par le trustore par défaut de votre JVM (certificats autosignés par exemple), vous pouvez à ce niveau préciser dans JAVA_OPTS un trustore spécifique (que vous créez vous même) intégrant les autorités de certificats utilisés par les VirtualHosts Apache (ou encore serveurs ldaps).

```
JAVA_OPTS="-Xms256m -Xmx512m -Djavax.net.ssl.trustStore=/opt/esup.univ-ville.jks -Djavax.net.ssl.
trustStorePassword=esupesup"
```

Ces deux instances seront démarrées via Systemd. On peut donc créer et activer ces deux services:

Pour Esup-SGC:

```
cat > /etc/systemd/system/tomcat-esup-sgc.service <<EOF
# Systemd unit file for tomcat
[Unit]
Description=Apache Tomcat Web Application Container
After=syslog.target network.target
[Service]
Type=forking
EnvironmentFile=/opt/esup-env
Environment=CATALINA_PID=/opt/tomcat-esup-sgc/temp/tomcat.pid
Environment=CATALINA_HOME=/opt/tomcat-esup-sgc
ExecStart=/opt/tomcat-esup-sgc/bin/startup.sh
ExecStop=/bin/kill -15 $MAINPID
User=esup
Group=esup
[Install]
WantedBy=multi-user.target
EOF

systemctl enable tomcat-esup-sgc.service
```

Pour Esup-NFC-TAG:

```
cat > /etc/systemd/system/tomcat-esup-nfc-tag.service <<EOF
# Systemd unit file for tomcat
[Unit]
Description=Apache Tomcat Web Application Container
After=syslog.target network.target
[Service]
Type=forking
EnvironmentFile=/opt/esup-env
Environment=CATALINA_PID=/opt/tomcat-esup-nfc-tag/temp/tomcat.pid
Environment=CATALINA_HOME=/opt/tomcat-esup-nfc-tag
ExecStart=/opt/tomcat-esup-nfc-tag/bin/startup.sh
ExecStop=/bin/kill -15 $MAINPID
User=esup
Group=esup
[Install]
WantedBy=multi-user.target
EOF

systemctl enable tomcat-esup-nfc-tag.service
```

Configuration des Tomcat

Pour l'instance d'Esup-SGC, éditer `/opt/tomcat-esup-sgc/conf/server.xml` afin de configurer le port 8205 et un connecteur AJP sur le port 8209.

Les connecteurs HTTP et HTTPS (ports 8080 et 8443 par défaut) doivent être commentés (ou être configurés sur un autre port pour éviter les conflits avec la seconde instance Tomcat)

```
<Server port="8205" shutdown="SHUTDOWN">
<!-- (...) -->
<Connector port="8209" protocol="AJP/1.3" redirectPort="8443" asyncTimeout="1200000" tomcatAuthentication="
false" scheme="https" secure="true" URIEncoding="UTF-8" secretRequired="false"/>
```

Idem pour l'instance d'Esup-NFC-TAG qui utilisera les ports 8305 et 8309.

```
<Server port="8305" shutdown="SHUTDOWN">
<!-- (...) -->
<Connector port="8309" protocol="AJP/1.3" redirectPort="8443" asyncTimeout="1200000" tomcatAuthentication="
false" scheme="https" secure="true" URIEncoding="UTF-8" secretRequired="false"/>
```

- Le tomcatAuthentication à false est indispensable pour que l'authentification shibboleth via le frontal apache et mod_shib fonctionne.
- L'attribut asyncTimeout est donné en milli-secondes (1200000 pour 20 minutes) - esup-sgc et esup-nfc-tag utilisent la technique de long polling lors du badgeage de cartes ; c'est aussi ce qui explique le fait que l'on mette un timeout de 3600 secondes au niveau des ProxyPass Apache (les timeout des ProxyPass Apache doivent être plus grands que les asyncTimeout des tomcat).

Configuration d'Apache

Activer les modules suivants:

```
a2enmod rewrite
a2enmod ssl
a2enmod proxy_ajp
a2enmod proxy_http
a2enmod shib
```

Créer un fichier de configuration pour le VirtualHost **esup-sgc.univ-ville.fr** `/etc/apache2/sites-available/esup-sgc.univ-ville.fr.conf`

```
<VirtualHost *:80>
    ServerName esup-sgc.univ-ville.fr
    ServerAdmin webmaster@univ-ville.fr
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    RewriteEngine On
    RewriteCond %{REQUEST_METHOD} ^(TRACE|TRACK)
    RewriteRule .* - [F]
    RewriteRule ^/(.*)$   https://esup-sgc.univ-ville.fr/$1   [L,R]
</VirtualHost>

<VirtualHost *:443>
    ServerName esup-sgc.univ-ville.fr
    ServerAdmin webmaster@univ-ville.fr
    DocumentRoot /var/www/html
    ErrorLog  ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    SSLEngine on
    SSLCertificateFile    /etc/letsencrypt/live/esup-sgc.univ-ville.fr/cert.pem
    SSLCertificateKeyFile /etc/letsencrypt/live/esup-sgc.univ-ville.fr/privkey.pem
    SSLCertificateChainFile  /etc/letsencrypt/live/esup-sgc.univ-ville.fr/chain.pem

    ProxyPass /Shibboleth.sso !
    ProxyPass /secure !
    ScriptAlias /secure /var/www/printenv.pl
    ShibCompatValidUser Off

    <Location /Shibboleth.sso>
        SetHandler shib
        AuthType None
        Require all granted
    </Location>
    <Location /shibboleth-sp>
        AuthType None
        Require all granted
    </Location>
    Alias /shibboleth-sp/main.css /usr/share/shibboleth/main.css
    <Location /secure>
        AuthType shibboleth
        ShibRequestSetting requireSession 1
        require shib-session
        ShibUseHeaders On
        ShibRequestSetting applicationId default
    </Location>
    <Location />
        AuthType shibboleth
        ShibRequestSetting requireSession 1
        require shib-session
        ShibUseHeaders On
        ShibRequestSetting applicationId default
    </Location>
   <Location "/resources">
        Require all granted
        ShibRequireSession Off
   </Location>
   <Location "/wsrest">
       Require all granted
       ShibRequireSession Off
   </Location>
   <Location "/payboxcallback">
       Require all granted
       ShibRequireSession Off
   </Location>
```

```
   ProxyPass / ajp://localhost:8209/ ttl=10 timeout=3600 retry=1

   AddOutputFilterByType DEFLATE text/plain text/html text/css text/javascript application/x-javascript
application/javascript application/json image/svg+xml


</VirtualHost>
```

Idem pour le VirtualHost **esup-nfc-tag.univ-ville.fr** dans `/etc/apache2/sites-available/esup-nfc-tag.univ-ville.fr.conf`

```
<VirtualHost *:80>
    ServerName esup-nfc-tag.univ-ville.fr
    ServerAdmin webmaster@univ-ville.fr
    DocumentRoot /var/www/html
    ErrorLog  ${APACHE_LOG_DIR}/error_esup-nfc-tag.log
    CustomLog ${APACHE_LOG_DIR}/access_esup-nfc-tag.log combined
    RewriteEngine On
    RewriteCond %{REQUEST_METHOD} ^(TRACE|TRACK)
    RewriteRule .* - [F]
    RewriteRule ^/(.*)$   https://esup-nfc-tag.univ-ville.fr/$1   [L,R]
</VirtualHost>

<VirtualHost *:443>
    ServerName esup-nfc-tag.univ-ville.fr
    ServerAdmin webmaster@univ-ville.fr
    DocumentRoot /var/www/html
    ErrorLog  ${APACHE_LOG_DIR}/error_esup-nfc-tag.log
    CustomLog ${APACHE_LOG_DIR}/access_esup-nfc-tag.log combined
    SSLEngine on
    SSLCertificateFile    /etc/letsencrypt/live/esup-nfc-tag.univ-ville.fr/cert.pem
    SSLCertificateKeyFile /etc/letsencrypt/live/esup-nfc-tag.univ-ville.fr/privkey.pem
    SSLCertificateChainFile  /etc/letsencrypt/live/esup-nfc-tag.univ-ville.fr/chain.pem

    ProxyPass /Shibboleth.sso !
    ProxyPass /secure !
    ScriptAlias /secure /var/www/printenv.pl
    ShibCompatValidUser Off
    <Location /Shibboleth.sso>
        SetHandler shib
        AuthType None
        Require all granted
    </Location>
    <Location /shibboleth-sp>
        AuthType None
        Require all granted
    </Location>
    Alias /shibboleth-sp/main.css /usr/share/shibboleth/main.css
    <Location /secure>
        AuthType shibboleth
        ShibRequestSetting requireSession 1
        require shib-session
        ShibUseHeaders On
        ShibRequestSetting applicationId esup-nfc-tag
    </Location>
    <Location /manager>
        AuthType shibboleth
        ShibRequestSetting requireSession 1
        require shib-session
        ShibUseHeaders On
        ShibRequestSetting applicationId esup-nfc-tag
    </Location>
    <Location /admin>
        AuthType shibboleth
        ShibRequestSetting requireSession 1
        require shib-session
        ShibUseHeaders On
        ShibRequestSetting applicationId esup-nfc-tag
    </Location>
```

```
    <Location /nfc>
        AuthType shibboleth
        ShibRequestSetting requireSession 1
        require shib-session
        ShibUseHeaders On
        ShibRequestSetting applicationId esup-nfc-tag
    </Location>

    ProxyPass / ajp://localhost:8309/ ttl=10 timeout=3600 retry=1

    AddOutputFilterByType DEFLATE text/plain text/html text/css text/javascript application/x-javascript
application/javascript application/json image/svg+xml

</VirtualHost>
```

A noter que l'**`applicationId`** du **`ShibRequestSetting`** diffère selon les VirtualHosts.
De plus, dans cet exemple, chaque VirtualHost dispose de son propre certificat. Il est tout à fait possible d'utiliser le même sous-réserve que les noms des deux VirtualHosts y soient indiqués (SAN).

En présentant la chaîne correctement et avec des certificats tels que proposés par Renater/Sectigo (les chemins ci-dessus correspondent à des exemples de chemins de certificats récupérés via le protocole ACME par certbot ... depuis Sectigo) le keystore par défaut de votre JVM suffira et **vous n'avez pas besoin alors d'importer unitairement vos certificats dans le keystore java**.

Si jamais vous avez un certificat issu d'une autorité de certification non reconnue par votre JVM (à proscrire) vous pouvez, en dernier recours, intégrer ce certificat au keystore ainsi par exemple:

```
# on copie le cacerts initial pour conserver la confiance dans les autorités de certification racines par défaut
cp /etc/ssl/certs/java/cacerts /opt/esup.univ-ville.jks
# par défaut le password est changeit,on le modifie
keytool -storepasswd -keystore cacerts
# utile si on doit supprimer un ancien certificat expiré pour en mettre un nouveau
keytool -delete -alias mon_cert -keystore /opt/esup.univ-ville.jks
# on importe le certificat
keytool -import -file /etc/apache2/certs/esup-sgc.crt -alias sgc -trustcacerts -keystore /opt/esup.univ-ville.
jks
```

À nouveau, normalement, avec un certificat valide et bien présenté, vous n'avez pas besoin de réaliser cette opération sur le keystore java et vous n'avez pas besoin de fait d'avoir ce fichier keystore /opt/esup.univ-ville.jks

Activer les sites:

```
a2dissite 000-default.conf
a2ensite  esup-sgc.univ-ville.fr
a2ensite  esup-nfc-tag.univ-ville.fr
```

Installation du SP Shibboleth

Générer une nouvelle clé:

```
shib-keygen
```

Cette commande permet de générer les fichiers `sp-key.pem` et `sp-cert.pem` dans `/etc/shibboleth/`

Editer `/etc/shibboleth/shibboleth2.xml`

Avant la balise **ApplicationDefaults**, ajouter un **RequestMap** avec le nom des deux virtualhost:

```
    <RequestMapper type="Native">
        <RequestMap applicationId="default">
            <Host name="esup-nfc-tag.univ-ville.fr" applicationId="esup-nfc-tag" authType="shibboleth"
requireSession="false"/>
            <Host name="esup-sgc.univ-ville.fr" applicationId="esup-sgc" authType="shibboleth" requireSession="
false"/>
        </RequestMap>
    </RequestMapper>
```

Configurer la balise **SSO** pour utiliser le WAYF de Renater (ou un Idp par défaut sinon, cf variante en commentaires) :

```
    <ApplicationDefaults entityID="https://esup-sgc.univ-ville.fr" ...>
        <Sessions ...>
                        <!--
            <SSO entityID="https://idp.univ-ville.fr/idp/shibboleth">
              SAML2 SAML1
            </SSO>
                        -->
                        <SSO location="/"
                discoveryProtocol="SAMLDS" discoveryURL="https://discovery.renater.fr/renater"
>

              SAML2
SAML1

            </SSO>
```

Penser à modifier le contact du support:

```
<Errors supportContact="sysadmin@univ-ville.fr"
```

Votre fournisseur de Metadata Renater (ou directement l'IDP - variante donnée en commentaires ; si vous n'utilisez pas la fédération Renater ...) - notez ici l'usage du "Whitelist" sur la fédération Renater :
Attention si vous avez un sp en version 3, uri="https://xx" est ignoré et il faut mettre url= !!! L'erreur étant peu évidente à comprendre, c'est 1/2 journée perdue

```
    <!--
        <MetadataProvider type="XML" validate="true"
              url="https://idp.univ-ville.fr/idp/shibboleth"
              backingFilePath="idp.univ-ville.fr-metadata.xml">
        </MetadataProvider>
        -->
 <MetadataProvider type="XML" url="https://metadata.federation.renater.fr/renater/main/main-idps-renater-
metadata.xml" backingFilePath="/etc/shibboleth/metadatas/main-idps-renater-metadata.xml" reloadInterval="7200">
               <MetadataFilter type="RequireValidUntil" maxValidityInterval="2419200"/>
               <MetadataFilter type="Signature" certificate="renater-metadata-signing-cert-2016.pem"/>
               <MetadataFilter type="Whitelist">
                 <Include>urn:mace:cru.fr:federation:univ-rouen.fr</Include>
                 <Include>https://shibboleth.insa-rouen.fr/idp/shibboleth</Include>
               </MetadataFilter>
 </MetadataProvider>
```

Et enfin, avant la fermeture de la balise ApplicationDefaults, rajoutez un ApplicationOverride. Dans notre cas, le VirtualHost esup-sgc.univ-ville.fr utilisera le default, on utilisera un id spécifique pour esup-nfc-tag:

```
        <ApplicationOverride id="esup-nfc-tag" entityID="https://esup-nfc-tag.univ-ville.fr/shibboleth"/>
```

Les Metadata doivent à présent être téléchargeables à ces adresses:

https://esup-sgc.univ-ville.fr/Shibboleth.sso/Metadata
https://esup-nfc-tag.univ-ville.fr/Shibboleth.sso/Metadata

Il reste donc à les intégrer à l'IDP, soit directement, soit en passant par la fédération d'identité.

Rotation des logs

Il est possible de mettre en place une rotation journalière des logs. La plupart des distributions fournissent logrotate. Il est donc possible de créer un fichier de config `/etc/logrotate.d/esup-sgc` avec, par exemple, le contenu suivant:

```
/opt/tomcat-esup-sgc/logs/catalina.out {
        copytruncate
        daily
        missingok
        rotate 30
        compress
        delaycompress
}
/opt/tomcat-esup-nfc-tag/logs/catalina.out {
        copytruncate
        daily
        missingok
        rotate 30
        compress
        delaycompress
}
```

Sous Debian, il est également possible d'éditer le fichier `/var/lib/logrotate/status` pour déterminer plus précisément la date de rotation (ceci est utile dans le cas d'une utilisation de lvm2 par exemple).

Installation

Éléments requis

Pour avoir un système de gestion de cartes fonctionnel, l'installation minimale consiste à installer :

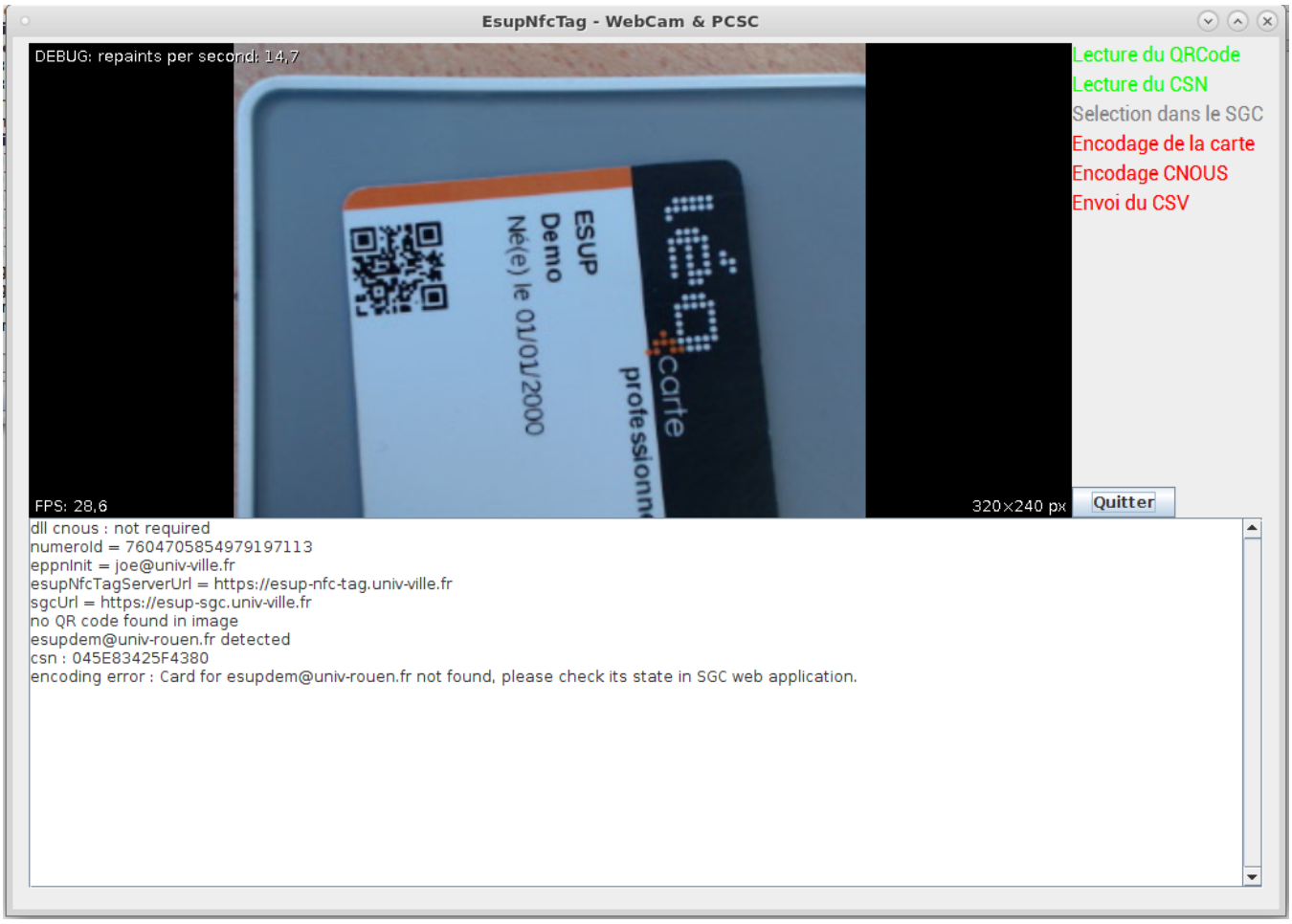
- un serveur tomcat avec l'application web esup-sgc : Installation ESUP-SGC - Configurations ESUP-SGC et ESUP-NFC-TAG-SERVER
- un serveur tomcat avec l'application web esup-nfc-tag-server : ESUP-NFC-TAG-SERVER

L'ordre d'installation n'a pas d'importance. Toutefois, les deux applications étant dépendentes, la documentation d'installation d'ESUP-SGC comporte des configurations d'ESUP-NFC-TAG-SERVER

Pour enrôler et éventuellement encoder vos cartes vous avez également besoin d'un client esup-sgc-client : voir la documentation [archivé]

Installation materielle

Dans l'application esup-sgc-client, pour l'édition en 2 temps, l'encodage des cartes s'effectue à l'aide de la webcam et du lecteur de carte NFC. La webcam est disposée au dessus du lecteur de cartes de manière à le filmer. Lorsque qu'une carte est placée sur le lecteur la webcam lit le qrcode puis lance l'encodage.



Pour automatiser l'encodage des cartes il est possible d'utiliser une imprimante Zebra ZXP3. voir : Tuto robot encodeur basé sur une Zebra ZXP3

Pour plus d'informations sur les différentes possibilités d'édition, voir la page ESUP-SGC-Client et édition des cartes

Éléments optionnels

Vous pouvez également mettre en place les applications Esup-nfc-tag-desktop et Esup-nfc-tag-droid :

- Esup-nfc-tag-desktop : ESUP-NFC-TAG-DESKTOP
- Esup-nfc-tag-droid : ESUP-NFC-TAG-DROID

Alliées à ESUP-SGC, ces applications pourront vous permettre

- de rechercher une carte en la badgeant
- de marquer une carte comme "livrée" en la badgeant