FAQ

- Projet / plateforme ESUP-SGC
  - A quoi correspond ESUP-SGC, que fait-il, quelle est sa couverture fonctionnelle ?
  - Nous hésitons à passer sur ESUP-SGC ...
  - De quel matériel ai-je besoin ?
  - L'impression des cartes nécessite de faire deux passages de carte ?
  - Est-ce qu'ESUP-SGC est pensé pour être "multi-établissements" ?
  - Est-ce qu'ESUP-SGC a été pensé pour respecter le RGPD ?
- ESUP-SGC
  - A quoi correspondent les rôles dans ESUP-SGC ?
    - ROLE_ADMIN
    - ROLE_SWITCH_USER
    - ROLE_SUPER_MANAGER
    - ROLE_MANAGER_XYZ
    - ROLE_LIVREUR
    - ROLE_UPDATER
    - ROLE_CONSULT
    - ROLE_CONSULT_XYZ
    - ROLE_VERSO
    - ROLE_USER
    - ROLE_USER_NO_EDITABLE
    - ROLE_USER_RENEWAL_PAYED
  - Dans l'application ESUP-SGC, peut-on utiliser une base de données externe Oracle pour récupérer des données utilisateurs ?
  - Je n'ai pas de groupes dans ldap, est-ce que je peux plutôt utiliser des filtres pour affecter les rôles dans l'application ESUP-SGC ?
  - Est-ce que ldap est obligatoire ?
  - L'eppn (eduPersonPrincipalName) est utilisé comme clef métier, il doit donc figurer dans le ldap de l'établissement ?
  - Les serveurs ESUP-SGC et ESUP-NFC-TAG sont shibbolethisés, leur déclaration en tant que Service Provider dans la fédération d'identités Renater est donc obligatoire ?
  - Qu'est-ce qu'un thème dans ESUP-SGC ?
  - Comment ajouter d'autres champs 'recto' dans un thème de carte ?
  - Comment faire une demande de carte en utilisant le webservice proposé par ESUP-SGC ?
  - De quelles données utilisateur issues du SI esup-sgc a besoin ?
  - Comment sont synchronisées les données utilisateur ?
  - Comment obtenir des identifiants pour utiliser l'API CNOUS lescrous ou encore l'API ESC (Eropean Student Card) ?
  - Peut-on encoder l'application CROUS quand on utilise des cartes vierges ?
  - Utilisation de cartes pré-encodées
  - Quelles informations sont échangées entre ESUP-SGC et l'API CROUS ?
  - A quoi correspondent les Apps disponibles depuis le menu de l'interface web ESUP-SGC ?
    - Encodeur
    - Encodeur - robot ZXP3
    - Application Android
    - Application Java
  - Dans le cadre d'une migration sur ESUP-SGC, comment réimporter les cartes éditées/encodées par l'ancien SGC ?
  - Comment repartir sur une base propre et vide dans esup-sgc, suffit-il de mettre "create" au lieu de "update" au niveau du fichier persistence.xml ?
  - Comment récupérer les photos par script ?
  - Comment récupérer par script les données et cartes d'un ou plusieurs utilisateurs ?
  - Quelle version de Java puis-je utiliser ?
  - Quelles optimisations serveur sont possibles ?
- ESUP-NFC-TAG
  - Je n'ai pas de groupes dans ldap, est-ce que je peux plutôt utiliser des filtres pour affecter les rôles dans l'application ESUP-NFC-TAG ?
  - Est-ce que ldap est obligatoire ?
  - Nous souhaitons mettre en place l'encodage des cartes pour du contrôle d'accès basé sur Desfire
  - J'ai besoin de connaître la master key de ma carte ?
  - Quelle est la master key de ma carte ?
  - Comment avoir la master-key sur des cartes pré-encodées CROUS ?
  - Quels types de cartes ESUP-NFC-TAG est-il capable d'encoder ?
  - Peut-on utiliser esup-nfc-tag pour modifier la master key d'une carte
  - Quelle est la différence entre Mifare Desfire EV1 et Mifare Desfire EV2 ?
  - Est-ce que ESUP-SGC peut coder la DEUInfo de la care étudiante européenne ?
  - Est-ce qu'on peut demander à un prestataire de se charger de coder la DEUInfo ?
  - Est-ce qu'esup-sgc peut positionner des DAM keys sur les cartes supportant Mifare DesFire EV2 ?
  - Peut-on "virtualiser" la carte multi-services ?
  - Est-ce qu'esup-nfc-tag peut mettre à jour électroniquement une carte Mifare Desfire ?
  - Est-ce qu'esup-nfc-tag peut ré-encoder complètement une carte Mifare Desfire ?
  - Quel est le taux de perte de cartes lors de l'édition dans ESUP-SGC ?
    - Problème d'impression
    - Problème d'encodage

Projet / plateforme ESUP-SGC

A quoi correspond ESUP-SGC, que fait-il, quelle est sa couverture fonctionnelle ?

En parcourant les différentes documentations, vous devriez vous faire une idée de la couverture fonctionnelle assez large d'ESUP-SGC.

Celui-ci est et a été présenté à diverses occasions et des vidéos, articles et présentations sont disponibles en ligne : ESUP-SGC#SGC-Documents,sp%C3%A9cifications,pr%C3%A9sentations,...

Nous vous invitons par exemple à visionner la dernière présentation actuellement en date : la présentation "ESUP-SGC, Système de Gestion de Cartes sur-mesure pour l'ESR" proposée aux JRES 2019 à Dijon **Vidéo / Diaporama / Article.**

Nous hésitons à passer sur ESUP-SGC ...

Si vous mettez en œuvre un projet de carte étudiante / professionnelle multi-services sur technologie Desfire dans votre établissement de l'ESR, la question du passage à ESUP-SGC peut effectivement se poser.

Les raisons qui peuvent vous pousser à mettre en place ESUP-SGC au lieu d'un autre SGC, notamment un SGC propriétaire fourni par un prestataire, sont en effet nombreuses :

- Une non satisfaction du produit actuellement utilisé, mal ou peu intégré dans votre Système d'Information (c'est ce qui nous a amené à développer ESUP-SG, voir à ce propos sgc-v3.pdf).
- Une meilleure maîtrise du SGC, de votre projet, de votre carte.
- Une indépendance vis à vis d'un prestataire et d'un logiciel propriétaire, dont la pérennité ne peut être garantie (ESUP-SGC est un logiciel libre qui vous appartient sans restriction).
- Une meilleure intégration du SGC dans votre Système d'Information, avec des interactions fortes et synchrones
    - avec vos briques du SI, source de données :
        - authentification/identification shibboleth
        - annuaire supann/ldap
        - bases de données sql
    - avec les services de votre SI, consommateurs de la carte :
        - CROUS/IZLY : via l' **API CROUS** - grâce à l'usage de cette API (en lieu et place de InfoCarteCROUS) ESUP-SGC vous y apporte une maitrise des échanges, une compréhension et possibilité de résoudre les problèmes de synchronisation/activation de comptes et carte Izly, ce en temps réel ; les avantages sont nombreux et qualitatifs pour l'usager final qui utilise les services CROUS/IZLY (étudiant, personnel, ...).
        - Contrôle d'accès : la synchronisation temps réel de vos cartes avec les contrôles d'accès que peut vous permettre de mettre en place ESUP-SGC renforce indéniablement la sécurité de vos accès.
        - Bibliothèques
        - Impression
        - Initiative de la Carte Etudiante Européenne (ESC) - voir la page de documentation de l'intégration du projet Carte étudiante européenne dans ESUP-SGC pour les détails très techniques opérationnels.
        - Outils institutionnels divers
- Une interface web dédiée à chacun, dont les utilisateurs finaux
- Une possibilité d'utiliser le matériel que vous souhaitez, en terme d'impression notamment ; toute la chaine d'édition pouvant utiliser uniquement des protocoles standards (et pas les API/SDK des imprimantes), cela vous évite les problèmes récurrents de compatibilité des imprimantes vis à vis des version d'OS, de fin de vie de modèles d'imprimantes, d'instabilité matériel ... notamment au cours du temps.
- ...

Un certain nombre de raisons peuvent a contrario vous pousser à faire un autre choix :

- Le logiciel que vous utilisez actuellement vous convient et vous en êtes satisfait.
- Vous utilisez un logiciel qui n'est pas juste un SGC mais un gestionnaire d'identités dont les fonctionnalités spécifiques vous sont précieuses (ESUP-SGC n'est pas un gestionnaire d'identités, c'est un SGC qui a vocation à s'intégrer dans un Système d'Information déjà établi).
- Vous n'avez pas les compétences et les moyens humains disponibles et motivés pour mettre en place un tel logiciel
- Vous ressentez le besoin d'avoir un prestataire qui vous accompagne de bout en bout pour assurer la mise en œuvre de votre projet, ce avec un logiciel maitrisé par le prestataire.
- Vous pensiez passer sur ESUP-SGC en lieu et place de votre solution actuelle pour réaliser des économies de prestation et de licence : vous n'envisagiez pas cependant de changer/modifier/améliorer votre organisation, les fonctionnalités proposées, etc.
- Le fait qu'ESUP-SGC impose une édition en 2 temps (impression puis encodage) vous parait rédhibitoire.
- ...

Bref, le choix n'est pas évident, et en tant que simple communauté et développeurs d'ESUP-SGC, nous n'avons aucun intérêt à vous convaincre de passer à ESUP-SGC si vous n'avez tout simplement pas l'envie, la motivation, la volonté d'adhérer à ce projet ; au contraire même, on serait très embêtés (en plus d'être étonnés tout de même 😅) qu'ESUP-SGC vous donne moins de satisfaction que votre ancienne solution !

De quel matériel ai-je besoin ?

En plus des applications web à installer sur un serveur, il vous faut pour éditer des cartes :

- des cartes Mifare DesfireEV1 ou EV2
- un PC sous linux ou windows (l'encodeur ne fonctionne actuellement pas sous MAC OS)
- une imprimante à carte plastique ps/pcl (exemples : evolis primacy, zebra zxp7, ...)
- une webcam
- un lecteur de carte pc/sc (exemple : Identiv UTrust 3700 F)

L'impression des cartes nécessite de faire deux passages de carte ?

Effectivement, contrairement aux autres Systèmes de Gestion de Cartes, ESUP-SGC ne permet pas d'imprimer et encoder la carte en 1 seul passage dans une imprimante à cartes disposant d'un lecteur/encodeur NFC.
Dans ESUP-SGC, une carte est d'abord imprimée par une imprimante à carte, puis celle-ci est encodée dans un second temps (voir à ce propos la vidéo E SUP-SGC - demande de carte, impression, encodage et activation).

C'est un choix qui a été fait dès l'étude préliminaire, en toute conscience, avant même la conception du SGC.
A ce sujet, voir le document "ESUP-SGC : UN SGC LIBRE", Juin 2017.

La mise en oeuvre des API propriétaires des imprimantes à cartes pose en effet quelques problèmes : instabilité des drivers, imprimante spécifique imposée, code/librairie non évolutif lié à un environnement logiciel donné, nombre de pannes/erreurs potentielles élevée, nombre de rejets élevé, code fermé, coûts plus élevés, maintenance plus compliquée, temps de mise en oeuvre et de manutention plus important à l'usage. Pour les SGC propriétaires du marché, ces difficultés de mise en oeuvre peuvent en partie être traitées par les prestataires/éditeurs au travers de différentes prestations : achat d'imprimantes spécifiques au travers du prestataire (vente liée), contrat de maintenance matérielle, contrat de maintenance logicielle, formation d'installation, dépannage ponctuelle, montée de version logicielle, migration de codes sur une nouvelle version d'imprimantes, etc. Ces prestations font partie intégrante du modèle économique et ne posent de fait pas de problème aux éditeurs. Pour un SGC libre, la situation est toute autre ; pour que le modèle de développement et de mutualisation autour du logiciel libre tienne, on minimise au maximum les problèmes côté des établissements pour faire en sorte que leurs installations fonctionnent avec un minimum de support (stabilité, autonomie, indépendance notamment matérielle), il en découle un certain nombre de choix très pragmatiques, notamment celui de fonctionner au maximum sur des standards. Ainsi pour l'impression des cartes on propose par défaut de se baser uniquement sur ps/pcl et pour l'encodage on utilise pc/sc ; ce qui implique d'imprimer et d'encoder la carte en 2 actions bien distinctes.

CF cette même FAQ, nous sommes conscients cependant que cet aspect puisse questionner les établissements qui hésitent à passer à ESUP-SGC.
Sur le papier, cette spécificité peut en effet paraître moins intéressante que l'impression+encodage en 1 passe ; à nouveau, notre expérience en tant qu'utilisateurs de SGC nous fait cependant dire le contraire !

Pour les établissements qui éditent une quantité conséquente de cartes à imprimer/encoder tous les ans (l'Université de Rouen Normandie qui a développé ESUP-SGC accueille 35.000 étudiants), en plus d'utiliser des imprimantes à cartes performantes pour l'impression (le fait de n'être pas lié à une API propriétaire spécifique permet de choisir l'imprimante que vous souhaitez), on propose d'utiliser une imprimante (avec lecteur nfc permettant l'encodage) pour procéder à l'encodage dans un second temps en utilisant les possibilités de chargement et d'encodage de l'imprimante par lot.
Actuellement on propose un tel "robot d'encodage" via un code spécifique à l'imprimante zxp3 de zebra. Dans le cadre de l'utilisation d'un autre type d'imprimante pour mettre en place cet encodage par lot, il faudra(it) porter le code sur l'API propriétaire spécifique à l'image que ce qui a été fait pour la zxp3 : https://github.com/EsupPortail/esup-sgc-client/tree/univ-rouen-robot-zxp3

Est-ce qu'ESUP-SGC est pensé pour être "multi-établissements" ?

Oui, une page dédiée à cette question est proposée sur ce WIKI.

Est-ce qu'ESUP-SGC a été pensé pour respecter le RGPD ?

Oui, une page dédiée à cette question est proposée sur ce WIKI.

ESUP-SGC

A quoi correspondent les rôles dans ESUP-SGC ?

ROLE_ADMIN

Permet d'avoir la vue "Admin" de l'interface, et donc accès aux paramètres de configuration, aux imports CSV, aux logs, etc.
Le rôle Admin permet également de disposer de la fonction SU (Switch User).

ROLE_SWITCH_USER

Permet de disposer de la fonction SU (Switch User) ; à donner éventuellement à un gestionnaire.
Attention avec ce rôle on peut faire un Switch User sur un administrateur, donc ce rôle peut finalement permettre de devenir administrateur simplement.

ROLE_SUPER_MANAGER

Permet d'avoir la vue "Manager" de l'interface. Un manager peut valider/refuser une demande, imprimer les cartes et les activer.
Le lien pour l'application java d'encodage (disponible depuis le menu Apps) est présent, cette application java étant à la fois cliente d'ESUP-SGC et ESUP-NFC-TAG.

ROLE_MANAGER_XYZ

Rôle particulier et dynamique, XYZ étant à changer par un userType, comme P par exemple (dans les configurations données par défaut, P est un userType qui désigne les personnels) : MANAGER_P.

Si l'utilisateur a le rôle MANAGER_P il ne pourra rechercher (et éditer, etc.) que les cartes dont les utilisateurs sont de userType P.

ROLE_LIVREUR

La vue "Manager" est disponible en lecture seule.

Il est possible de noter une carte comme livrée :

- via l'interface web
- en utilisant une application cliente esup-nfc-tag (disponible depuis le menu Apps) pour smartphone (android) ou de bureau (java) et en badgeant la carte qu'on livre

ROLE_UPDATER

Il est possible de mettre à jour électroniquement une carte en utilisant une application cliente esup-nfc-tag (disponible depuis le menu Apps) pour smartphone (android) ou de bureau (java) et en badgeant la carte.

Cette mise à jour électronique est configurée dans esup-nfc-tag (ajout d'applications, de fichiers et clefs ...)

ROLE_CONSULT

La vue "Manager" est disponible en lecture seule. Il n'est pas possible d'imprimer la carte ni de l'encoder, mais les liens vers les applications permettant d'utiliser le lecteur NFC sont présents, ils permettent de rechercher une carte via badgeage de la carte sur smartphone ou ordinateur.

Après badgeage d'une carte dans la 'salle recherche' de l'application cliente, à la validation la fiche de la carte est automatiquement affichée dans le navigateur de l'utilisateur connecté (dernière session en date) avec le même identifiant que sur l'application cliente (si connecté == si session en cours).

ROLE_CONSULT_XYZ

Même principe que pour ROLE_MANAGER_XYZ, permet de donner des droits de consultation restreints à certains userType uniquement (XYZ étant à remplacer par un userType configuré par ailleurs dans applicationContext-services.xml au travers des "userInfo").

ROLE_VERSO

Les utilisateurs ayant ce rôle peuvent voir le 'verso dématérialisé' d'une carte au travers d'un badgeage depuis esup-nfc-tag-droid ou esup-nfc-tag-desktop.

Notez que cette possibilité est offerte également aux utilisateurs ayant le rôle ROLE_CONSULT (ou/et ROLE_MANAGER, ROLE_SUPER_MANAGER, ROLE_ADMIN)

ROLE_USER

Pour pouvoir faire une demande de carte, l'utilisateur doit avoir ce rôle.

ROLE_USER_NO_EDITABLE

Si un utilisateur dispose de ce rôle, alors sa carte ne peut pas être éditée (par ex: problème sur dossier), même si celui-ci a pu effectuer la demande.

Plutôt que d'utiliser ce rôle, vous pouvez utiliser le champ 'userInfo' **editable** de l'utilisateur cf le tableau donné dans Configurations ESUP-SGC et ESUP-NFC-TAG-SERVER#SGCetESUP-NFC-TAG-SERVER-applicationContext-services.xml.

ROLE_USER_RENEWAL_PAYED

Si un utilisateur dispose de ce rôle, celui-ci doit payer avant de pouvoir demander un renouvellement de carte.

Plutôt que d'utiliser ce rôle, vous pouvez utiliser le champ 'userInfo' **requestFree** de l'utilisateur cf le tableau donné dans Configurations ESUP-SGC et ESUP-NFC-TAG-SERVER#SGCetESUP-NFC-TAG-SERVER-applicationContext-services.xml.

Dans l'application ESUP-SGC, peut-on utiliser une base de données externe Oracle pour récupérer des données utilisateurs ?

Oui.

Le driver oracle n'étant pas dans le maven central, une modification du pom.xml ne suffit pas. Il faudra en plus l'ajouter dans votre 'repository local' ainsi (après l'avoir téléchargé depuis https://www.oracle.com/technetwork/database/features/jdbc/default-2280470.html) :

```
mvn install:install-file -Dfile=/tmp/ojdbc7.jar -DgroupId=com.oracle -DartifactId=ojdbc7 -Dversion=12.1.0.2 -
Dpackaging=jar -DgeneratePom=true
```

puis ajout dans le pom.xml de la dépendance :

```
<dependency>
  <groupId>com.oracle</groupId>
  <artifactId>ojdbc7</artifactId>
  <version>12.1.0.2</version>
</dependency>
```

La configuration se fait ensuite dans applicationContext-services.xml avec un dataSource adéquat ...

```
    <bean id="dataSourceOracle" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close">
        <property name="driverClassName" value="oracle.jdbc.driver.OracleDriver"/>
        <property name="url" value="jdbc:oracle:thin:@oracle.devcake.co.uk:1521:INTL"/>
        <property name="username" value="sa"/>
        <property name="password" value=""/>
    </bean>
```

Je n'ai pas de groupes dans ldap, est-ce que je peux plutôt utiliser des filtres pour affecter les rôles dans l'application ESUP-SGC ?

Même si l'usage de groupes ldap, notamment via l'usage de Grouper, est conseillé (c'est ce que propose la configuration par défaut et c'est ce qui est utilisé dans la VM de démonstration), il est effectivement possible d'utiliser en lieu et place des filtres ldap.

Pour ce faire, dans applicationContext-services.xml on modifiera

```
        <bean id="groupService" class="org.esupportail.sgc.services.ldap.LdapGroupService">
                <property name="ldapTemplate" ref="ldapTemplate"/>
                <property name="groupSearchBase" value="ou=groups" />
                <property name="groupSearchFilter" value="member={0}"/>
                <property name="memberSearchBase" value="ou=people"/>
                <property name="memberSearchFilter" value="memberOf={0}"/>
        </bean>
```

par quelque chose du type :

```
        <bean id="groupService" class="org.esupportail.sgc.services.ldap.LdapFilterGroupService">
                <property name="ldapTemplate" ref="ldapTemplate"/>
                <property name="ldapFiltersGroups">
                        <map>
                                <entry key="(|(eduPersonAffiliation=student)(eduPersonAffiliation=employee))"
value="esup-sgc-users"/>
                                <entry key="eduPersonPrincipalName=joe@univ-ville.fr" value="esup-sgc-admins"/>
                                <entry key="eduPersonPrincipalName=jack@univ-ville.fr" value="esup-sgc-managers"
/>
                        </map>
                </property>
        </bean>
```

C'est ensuite ces noms de groupes ainsi déifinis ('esup-sgc-users', esup-sgc-admins', ...) qui peuvent être utilisés dans applicationContext-security.xml au niveau du bean sgcMappingGroupesRoles pour définir les rôles de chacun :

```
        <util:map id="sgcMappingGroupesRoles">
                <beans:entry key="esup-sgc-admins" value="ROLE_ADMIN" />
                <beans:entry key="esup-sgc-managers" value="ROLE_MANAGER" />
                <beans:entry key="esup-sgc-users" value="ROLE_USER" />
        </util:map>
```

Est-ce que ldap est obligatoire ?

L'identification par shibboleth est obligatoire. Ldap était aussi obligatoire au début du projet.

- L'idée est de s'appuyer au mieux sur supann (et schémas ldap associés) pour les attributs utilisateurs et donc de privilégier la récupération de ces attributs utilisateurs via LDAP.
- Ldap est aussi proposé pour gérer les rôles des utilisateurs via les groupes ldap.

Il est possible de fonctionner sans ldap, le premier cas d'usage ici a été le montage de site de démo esup-sgc-demo.univ-rouen.fr (voir la page wiki à propos de celle-ci, et notez notamment les limitations d'une telle intégration) :

- on récupère les attributs utilisateurs de shibboleth, et éventuellement d'une BD sql
- les groupes/rôles sont construits via des règles sur les attributs utilisateurs directement

L'eppn (eduPersonPrincipalName) est utilisé comme clef métier, il doit donc figurer dans le ldap de l'établissement ?

esup-sgc comme esup-nfc-tag utilise comme clef métier de l'utilisateur final l'eduPersonPrincipalName (l'eppn) ; pour la carte, c'est le CSN (Card Serial Number) qui est utilisé.

Ainsi dans la collecte d'informations réalisée sur les différentes sources de données possibles (principalement shibboleth, ldap, sql) l'eppn est la clef qui permet de synchroniser les champs/données utilisateurs (userinfos) individiellement.

Le paramétrage des requêtes sql ont en paramètre **?** l'eppn ; on aura alors des requêtes en *where eppn=?*

Pour le LDAP un filtre *eduPersonPrincipalName eq ?* est appliqué.

Aussi, pour pouvoir récupérer des champs utilisateurs au travers du ldap, **l'attribut eduPersonPrincipalName est requis dans le ldap**, et **il doit être indexé** (en eq) : une recherche ldap par filtre equals sur l'eppn est très régulièrement effectuée (à chaque synchronisation).
(*cn doit aussi être indexé pour permettre à esup-sgc d'effectuer une recherche ldap par filtre like sur cn (common name) lors du test ou dans l'outil de recherche ldap*)

Pour la partie SQL, le paramétrage de la requpête SQL complète rend la chose plus souple, on peut par exemple faire un *where uid = replace(?, '@univ-ville.fr', '')* ; la aussi, **il faut indexer la colonne sql sur laquelle le where est effectué**.

Les serveurs ESUP-SGC et ESUP-NFC-TAG sont shibbolethisés, leur déclaration en tant que Service Provider dans la fédération d'identités Renater est donc obligatoire ?

Non, rien n'oblige à déclarer vos serveurs ESUP-SGC et ESUP-NFC-TAG en tant que service provider dans la fédération d'identités (de production comme de test) Renater - https://services.renater.fr/federation/

Pour des raisons de tests, de développements ... si vous souhaitez notamment restreindre l'accès à ces serveurs, celà peut être plus simple/pratique /rapide de ne pas le faire en se contentant d'une approbation interne entre votre SP ESUP-SGC / ESUP-NFC-TAG et votre IdP (qui peut tout à fait être votre ldp de production).

Techniquement c'est par exemple ce qui est fait dans la VM de démonstration, VM embarquant un IdP shibboleth ; vous y retrouverez donc un exemple de telles configurations : VM ESUP-SGC

Si vous souhaitez par contre proposer un SGC multi-établissements de l'ESR, en permettant à des extérieurs issus d'autres établissements de demander des cartes avec leurs propres comptes d'établissements ; ou encore si vous voulez permettre l'intégration / importation de cartes d'SGC d'autres établissements dans votre propre SGC, déclarer votre SGC dans la fédération d'identités Renater est une bonne option.

Qu'est-ce qu'un thème dans ESUP-SGC ?

L'impression d'une carte par ESUP-SGC correspond à imprimer une page HTML dans un format "carte" (Cr80, soit une dimension de 85,7 x 54,03 mm). La mise en oeuvre de la charte graphique de la carte revient alors à une mise en forme d'un HTML via une feuille CSS, technologie maitrisée dans nos établissements.

Dans ESUP-SGC, un thème correspond ainsi notamment à une CSS ainsi qu'au logo de l'établissement tous deux utilisés lors de l'impression de la carte. Ce même CSS permet la prévisualisation (pour l'usager et pour le manager) de la carte, aussi dans ce contexte un css spécifique à la vue mobile est présent ainsi qu'un masque (de carte) et un qrcode.

On peut avoir plusieurs thèmes et une même 'clef' de thème peut également être utilisée plusieurs fois avec des versions différentes.

L'idée est ainsi de permettre d'avoir des thèmes différents suivant les individus et éventuellement des versions de thèmes différentes si le thème évolue dans le temps (changement de look de la carte d'un établissement).

L'affectation d'un thème à un utilisateur se fait au travers du peuplement d'un nouveau 'userInfo' nommé 'template'. La dernière version du thème correspondant à cette clef étant utilisée lors de l'impression de la carte.

Enfin notez que les thèmes sont gérés depuis l'interface web d'esup-sgc et l'outil permettant de les créer ou les modifier permet ainsi au passage d'éditer la CSS avec rendu immédiat synchronisé dans le navigateur (~ live edit du css) !

Comment ajouter d'autres champs 'recto' dans un thème de carte ?

7 champs utilisateurs rectox (de 1 à 7) sont proposés par ESUP-SGC pour être affichés et imprimés dans une carte.

Ces 7 champs sont construits et récupérés depuis le Système d'Information via la configuration d'ESUP-SGC.

Ces champs peuvent correspondre directement à des données utilisateurs atomiques comme le prénom, nom, date de naissance, etc.

Mais ils peuvent aussi correspondre à des blocs HTML complets permettant ainsi de proposer dans un même rectox plusieurs informations et de contourner si besoin cette limitation de **7** éléments.
On peut ainsi par exemple faire en sorte de proposer en recto4 un champ correspondant à

```
Née le 17/12/1999<br/>N° INE : 18100XXXXX
```

ou encore

```
<p class="card-birthday">Née le 17/12/1999</p><p class="card-ine">N° INE : 18100XXXXX</p>
```

Comment faire une demande de carte en utilisant le webservice proposé par ESUP-SGC ?

La demande de carte peut être faite par l'appel d'un webservice (API). Cet appel ressemble à ceci:

```
curl -F "eppn=username@univ-ville.fr" -F "difPhotoTransient=true" -F "crousTransient=true" -F
"europeanTransient=true" -F "PhotoFile.file=@/path/to/image.png"  https://esup-sgc.univ-ville.fr/wsrest/api
```

La liste des clients autorisés à utiliser ce webservice est définie dans la variable accessRestrictionWSRestAPI du fichier security.properties. Par exemple, pour autoriser certaines adresses IP (notez que la valeur de cette variable n'est pas entourée de guillemets):

```
accessRestrictionWSRestApi=hasIpAddress('127.0.0.1') or hasIpAddress('192.168.1.39') or hasIpAddress
('192.168.22.0/24')
```

De quelles données utilisateur issues du SI esup-sgc a besoin ?

ESUP-SGC a besoin de récupérer un certain nombre de champs/donnnées utilisateurs depuis le SI, il lui faut par exemple le nom et prénom pour l'imprimer sur la carte.

L'ensemble de ces champs sont listés dans le tableau de cette page wiki. On privilégie l'usage de champs utilisateurs 'standardisés'. En ce sens on privilégie notamment l'usage de champs supann.

Suivant votre usage, certains champs sont +/- obligatoires.

Comment sont synchronisées les données utilisateur ?

La synchronisation des données/champs utilisateurs depuis le Système d'Information vers ESUP-SGC se fait en fonction des "userInfoServices" que vous aurez configuré dans applicationContext-services.xml

En fonction des données synchronisées, cette synchronisation peut ensuite engendrer une synchronisation des services de contrôles d'accès, ldap, crous, esc-r (si la date de fin est dépassée et que la carte devient caduque, par exemple).

La synchronisation si-> esup-sgc est déclenchée par plusieurs moyens, en fonction de vos configurations et des actions des utilisateurs, c'est à dire :

- à chaque authentification de l'utilisateur et lors de la demande d'une carte
- lorsqu'un un gestionnaire clique sur le bouton "synchroniser" sur la fiche de l'utilisateur
- régulièrement en fonction de la configuration de votre fichier applicationTasksContext.xml
- si un appel web service (type *curl https://esup-sgc.univ-ville.fr/wsrest/api/sync?eppn=toto@univ-ville.fr*) est lancé - ce dernier moyen avancé peut vous permettre d'obtenir quelque chose de quasi synchrone en plaçant par exemple cette commande dans un trigger de votre base métier SI.

La page Synchronisations sur ce même wiki donne des informations techniques supplémentaires.

Comment obtenir des identifiants pour utiliser l'API CNOUS lescrous ou encore l'API ESC (Eropean Student Card) ?

Pour synchroniser les données utilisateurs et cartes avec l'API CNOUS ou encore l'API ESC vous avez besoin d'identifiants sur les plateformes de pré-production puis de production de l'api CNOUS et l'api ESC - cf Configurations API CROUS / ESCR.

Pour les obtenir, et en tant que membre de la DSI (ou référent technique dans votre établissement) vous pouvez contacter departement-vem@cnous.fr en mettant également en copie Vincent.Bonamy@univ-rouen.fr

Si vous êtes dans cette démarche, abonnez-vous également en premier lieu à la liste privée esup-sgc-devel.

Peut-on encoder l'application CROUS quand on utilise des cartes vierges ?

Dans le cadre d'Esup-SGC il est conseillé d'utiliser des cartes pré-encodées avec l'application CROUS/IZLY. Il est tout de même possible d'activer l'encodage CROUS lors de l'encodage de cartes vierges avec Esup-Sgc-Client. Pour cela il faut:

- Utiliser un PC sous un windows 64 bits et se procurer l'application cnousApi auprès de la liste esup-sgc-devel@esup-portail.org
- Obtenir une plage d'identifiants auprès du CNOUS ainsi qu'une DLL, un fichier clé CNOUS ZDC et une clé matérielle SAM.
- Installer l'application sous c:\cnousApi, le dossier devra contenir les fichier suivants:
  - cnous_fournisseur_carte.dll (fourni sur demande par le cnous)
  - CreationCarteCrous.exe (correspond à https://github.com/EsupPortail/esup-crous-client)
  - CreationCarteCrous.exe.config (correspond à https://github.com/EsupPortail/esup-crous-client)
  - key.txt (Clé CNOUS ZDC fourni sur demande par le cnous)
  - libeay32.dll (openSSL)
  - libssl32.dll (openSSL)
  - pcsc_desfire.dll (springcard)
- Brancher la clé USB SAM (fourni sur demande par le cnous)
- Lancer CreationCarteCrous.exe permet de s'assurer que l'application fonctionne correctement

  - "CreationCarteCrous.exe -t" doit afficher true
  - "CreationCarteCrous.exe -l" permet de lire l'application CROUS d'une carte

- Modifier la configuration du sgc dans applicationContext-services.xml

```
<bean class="org.esupportail.sgc.services.cardid.CnousCardIdService">
        <property name="appName" value="crous"/>
        <property name="idCounterBegin" value="<numero de debut de plage CNOUS>"/>
        <property name="postgresqlSequence" value="crous_smart_card_sequence"/>
        <property name="crousEncodeEnabled" value="true"/>
</bean>
```

- Insérer le premier identifiant de votre plage au niveau du idCounterBegin et mettre crousEncodeEnabled à true.

Lors du lancement de l'application Esup-Sgc-Client depuis le SGC, un contrôle de l'application cnousApi sera effectué.

La première ligne de log doit indiquer "dll cnous : OK"

Utilisation de cartes pré-encodées

En utilisant des cartes pré-encodées CROUS, vous n'aurez pas besoin de mettre en œuvre l'encodage CROUS (avec pc windows, dll crous, clef sam, génération d'identifiants izly ...).

Il vous suffira en effet simplement d'importer le fichier CSV reçu avec les cartes pré-encodées dans le SGC : onglet **Admin** > **Cartes Crous**.

Cela fonctionnera de fait si vous demandez à votre fournisseur de cartes un CSV reprenant le formattage proposé nativement par la DLL CROUS/CNOUS !

Lorsque vous commanderez votre pré-encodées avec l'application Desfire, pensez à demander de positionner une master-key spécifique sur l'ensemble de vos cartes pour avoir la maîtrise de celle-ci !
(voir à ce propos la q/r "Comment avoir la master-key sur des cartes pré-encodées CROUS ?")

Quelles informations sont échangées entre ESUP-SGC et l'API CROUS ?

ESUP-SGC échange avec l'API CROUS

- des données sur l'ayant droit (right holder) - cf RightHolder.java :
  identifier, firstName, lastName, email, dueDate, idCompanyRate, idRate, birthDate, ine, rneOrgCode, accountStatus, blockingStatus
- des données sur les cartes (smart card) - cf CrousSmartCard.java ; ces données correspondent aux données de sortie de la DLL CNOUS utilisée pour (pré)encoder les cartes avec l'application desfire crous/izly :
  idTransmitter, idMapping, idZdc, zdcCreationDate, pixSs, pixNn, appl, uid, rid
  (uid et rid sont tous deux valués avec le CSN)

A quoi correspondent les Apps disponibles depuis le menu de l'interface web ESUP-SGC ?

Ces applications sont des applications clientes permettant de badger la carte en lecture ou/et écriture.

Ces liens se configurent par un administrateur depuis le menu Admin > NavBarApp.

Pour une mise en oeuvre simplifiée, cf les documentations sur ce wiki, vous pouvez :

- vous appuyer sur l'application Android officielle sur le playstore Google d'EsupPortail
- et générer les clients pour PC depuis https://esup-sgc-client-web-installer.univ-rouen.fr/

Encodeur

Lien esup-sgc de esup-sgc-client.jnlp qui lance le jar esupsgcclient-XXXX.jar
Le code source est sur https://github.com/EsupPortail/esup-sgc-client - branche master
C'est l'encodeur par défaut à utiliser avec esup-sgc, il requiert une webcam et un lecteur usb nfc. Cela permet l'encodage de carte une à une.
Le jar est fourni par défaut compilé dans esup-sgc, il est signé par l'Université de Rouen.

Encodeur - robot ZXP3

Lien esup-sgc de esup-sgc-client-r2d2.jnlp qui lance le jar esupsgcclient-r2d2.jar
Le code source est sur https://github.com/EsupPortail/esup-sgc-client - branche univ-rouen-robot-zxp3
C'est un encodeur compatible esup-sgc, il requiert une webcam et une imprimante Zebra ZXP3 sous Windows. Cela permet l'encodage de plusieurs cartes via le chargeur de la ZXP3.
Le jar n'est par fourni par défaut dans esup-sgc, il faut le compiler soi-même et le signer (contrainte java web start).

Application Android

Lien esup-nfc-tag-server de esupnfctagdroid.apk
Le code source est sur https://github.com/EsupPortail/esup-nfc-tag-droid
C'est l'application cliente esup-nfc sous Android. Elle permet d'intégrer de manière générique le badgeage dans des applications institutionnelles. Pour ce faire elle propose le badgeage dans les 'salles' esup-nfc, salles elles-mêmes récupérées depuis d'autres applications. Esup-SGC fait partie de ces applications et propose des salles de recherche, livraison, verso dématérialisé, mise à jour.
L'apk n'est par fourni par défaut dans esup-nfc-tag-server, il faut configurer les sources (lien sur le serveur esup-nfc-tag-server) et le compiler soi-même.

Application Java

Lien esup-nfc-tag-server de esupnfctagdesktop.jar
Le code source est sur https://github.com/EsupPortail/esup-nfc-tag-desktop
C'est l'application cliente esup-nfc pour PC (Desktop : client java).
Elle permet d'intégrer de manière générique le badgeage dans des applications institutionnelles. Pour ce faire elle propose le badgeage dans les 'salles' esup-nfc, salles elles-mêmes récupérées depuis d'autres applications. Esup-SGC fait partie de ces applications et propose des salles de recherche, livraison, verso dématérialisé, mise à jour.
Le jar n'est par fourni par défaut dans esup-nfc-tag-server, il faut configurer les sources (lien sur le serveur esup-nfc-tag-server) et le compiler soi-même.

Dans le cadre d'une migration sur ESUP-SGC, comment réimporter les cartes éditées/encodées par l'ancien SGC ?

Voir Importation de 'cartes' dans ESUP-SGC / Migration des données.

Comment repartir sur une base propre et vide dans esup-sgc, suffit-il de mettre "create" au lieu de "update" au niveau du fichier persistence.xml ?

Oui, en mettant create en place de update au niveau du fichier persistence.xml la base est alors écrasée et recréée au prochain redémarrage du SGC.

Pour être complet, il faut noter ici que le trigger appelé lors de la suppression d'un 'fichier' (d'une photo ici surtout) n'est pas appelé en supprimant/recréant la base aussi directement. De fait les blobs (lo postgresql pour large object) correspondants aux fichiers/photos stockés en base ne sont en fait pas supprimés de la base postgresql et se retrouvent 'orphelins' car plus référencés dans la base esup-sgc qui a été réinitialisée.
Ils "trainent" donc dans la base postgresql et ne dérangent pas le moins du monde, à part qu'ils utilisent de la place ...
Aussi ici pour 'nettoyer' la base et supprimer effectivement ces blobs, on peut alors utiliser l'utilitaire vacuumlo sur la base (juste après avoir relancé esup-sgc avec create donc) :
postgres@debian-i7:~$ vacuumlo esupsgc

Plus d'info ici :
https://www.postgresql.org/docs/9.3/static/vacuumlo.html

Comment récupérer les photos par script ?

Esup-sgc propose une API permettant de récupérer les photos.

Pour pouvoir l'utiliser, l'IP du client doit être référencé dans accessRestrictionWSRestPhoto, fichier security.properties

- Récupérer la dernière photo en date non rejetée d'un utilisateur :

```
wget 'http://localhost:8080/wsrest/photo/joe@univ-ville.fr/photo'
```

- Récupérer la dernière photo en date dans un état donné d'un utilisateur :

```
wget 'http://localhost:8080/wsrest/photo/joe@univ-ville.fr/photo?cardEtat=ENABLED'
```

(état activé ici)
- Récupérer la dernière photo en date dans un état donné d'un utilisateur après une date précisée (renvoie un code http 304 si la photo en question n'a pas été modifiée depuis cette date) :

```
wget 'http://localhost:8080/wsrest/photo/joe@univ-ville.fr/photo?cardEtat=ENABLED&dateEtatAfter=2018-06-19'
```

Pour prendre en compte le choix de l'utilisateur de diffuser ou non sa photo, on peut faire les mêmes requêtes avec /restrictedPhoto en lieu et place de /photo :

```
wget 'http://localhost:8080/wsrest/photo/joe@univ-ville.fr/restrictedPhoto?cardEtat=ENABLED'
```

Comment récupérer par script les données et cartes d'un ou plusieurs utilisateurs ?

Esup-sgc propose une API permettant de récupérer les données et carte d'un ou plusieurs utilisateurs.

Pour pouvoir l'utiliser, l'IP du client doit être référencé dans accessRestrictionWSRestApi, fichier security.properties

```
wget 'http://localhost:8080/wsrest/api/get?eppn=toto@univ-ville.fr&eppn=titi@univ-ville.fr'
```

Quelle version de Java puis-je utiliser ?

Initialement, le projet fonctionnait de part et d'autre avec le JDK 8 d'Oracle.
Puis, suite au changement de license d'Oracle, à l'abandon prochain de Java Web Start, on a fait évoluer esup-sgc et esup-nfc-tag.

Désormais on propose

- côté serveur, pour esup-sgc et esup-nfc-tag, vous pouvez utiliser openjdk 11 (fourni par votre distribution) ; la version 8 est encore également supportée.
- côté client,
  - pour esup-sgc-client, esup-nfc-tag-desktop, esup-nfc-keyboard, vous pouvez utiliser openjdk11 avec openjfx11, c'est ce que vous propose et embarque l'installateur windows que vous pouvez générer depuis https://esup-sgc-client-web-installer.univ-rouen.fr/
  - si vous utilisez la version 'robot' d'esup-sgc-client utilisant une zxp3 pour encoder en série les cartes, cf la documentation à ce sujet vous devez rester sur une **version 8 du JDK disposant de JFX** (JavaFX) sur windows (le sdk zebra ne supportant pas les versions java ultérieurs), vous pouvez alors vous tourner sur la version de la communauté zulu du jdk+jfx en version 8 ; cf la documentation à ce sujet donc à nouveau.

Quelles optimisations serveur sont possibles ?

Octobre 2021, le esup-sgc de l'Université de Rouen Normandie gère 100.000 cartes pour plus de 50.000 utilisateurs (étudiants de l'année n et n-1, personnels, invités ...).

L'ensemble des briques serveur esup-sgc (esup-sgc, esup-nfc-tag, apache en frontal et base de données Postgresql) tourne sur une seule VM qui dispose de 2 CPUS et 8 GB de RAM.
La base de données prend un peu moins de 40GB sur disque.

Le service est stable et peu gourmand ; complexifier l'architecture en installant des mécansimes logiciels de failover ou load-balancing n'est pas conseillé.

1 GB de RAM alloué au tomcat d'esup-sgc (-Xms1024M -Xmx1024M) est suffisant.

L'AJP est utilisé entre le tomcat et le apache avec un proxypass ainsi fait :
ProxyPass / ajp://localhost:8009/ ttl=10 timeout=3600 retry=1

Pour 'monitorer' les tomcats, psi-probe, une sorte de webapp manager amélioré, peut être utile : https://github.com/psi-probe/psi-probe/

Un munin avec les plugins suivants peut également être utile par exemple :
cpu df_inode http_loadtime irqstats memory open_files postgres_autovacuum postgres_connections_ALL processes swap uptime df diskstats interrupts load netstat open_inodes postgres_checkpoints postgres_size_ALL proc_pri threads users

On conseille de paramétrer du cache http/apache et la compression côté des frontaux.

Côté PostgreSQL, on conseille de procéder à quelques paramétrages élémentaires permettant de mettre à profit les capacités matérielles du serveur.

Les synchronisations des utilisateurs s'opèrent grâce à de très nombreuses requêtes ldap/sql sur les LDAP et bases de données SQL du SI.
Ces requêtes sont issues des configurations données dans applicationContext-services.xml.
Aussi les ldap et bases de données doivent présenter des index permettant de répondre au mieux à ces requêtes (l'eppn est généralement utilisé comme clef).

ESUP-NFC-TAG

Je n'ai pas de groupes dans ldap, est-ce que je peux plutôt utiliser des filtres pour affecter les rôles dans l'application ESUP-NFC-TAG ?

Même si l'usage de groupes ldap, notamment via l'usage de Grouper, est conseillé (c'est ce que propose la configuration par défaut et c'est ce qui est utilisé dans la VM de démonstration), il est effectivement possible d'utiliser en lieu et place des filtres ldap.

Pour ce faire, dans applicationContext-security.xml on modifiera

```
        <beans:bean id="groupService" class="org.esupportail.nfctag.security.LdapGroupService">
                <beans:property name="ldapTemplate" ref="ldapTemplate"/>
                <beans:property name="groupSearchBase" value="ou=groups" />
                <beans:property name="groupSearchFilter" value="member={0}"/>
                <beans:property name="memberSearchBase" value="ou=people"/>
                <beans:property name="memberSearchFilter" value="memberOf={0}"/>
        </beans:bean>
```

par quelque chose du type :

```
        <beans:bean id="groupService" class="org.esupportail.nfctag.security.LdapFilterGroupService">
                <beans:property name="ldapTemplate" ref="ldapTemplate"/>
                <beans:property name="ldapFiltersGroups">
                        <util:map>
                                <beans:entry key="eduPersonPrincipalName=joe@univ-ville.fr" value="esup-nfc-
admins"/>
                                <beans:entry key="eduPersonPrincipalName=jack@univ-ville.fr" value="esup-nfc-
supervisors"/>
                        </util:map>
                </beans:property>
        </beans:bean>
```

C'est ensuite ces noms de groupes ainsi déifinis ('esup-sgc-admins', 'esup-nfc-supervisors') qui peuvent être utilisés au niveau du bean nfcMappingGroupesRoles pour définir les rôles de chacun :

```
        <util:map id="nfcMappingGroupesRoles">
                        <beans:entry key="esup-nfc-admins" value="ROLE_ADMIN" />
                        <beans:entry key="esup-nfc-supervisors" value="ROLE_SUPERVISOR" />
        </util:map>
```

Est-ce que ldap est obligatoire ?

L'identification par shibboleth est obligatoire. Ldap était aussi obligatoire au début du projet.

- Ldap est utilis pour gérer les rôles des utilisateurs via les groupes ldap.

Il est possible de fonctionner sans ldap, le premier cas d'usage ici a été le montage de site de démo esup-sgc-demo.univ-rouen.fr (voir la page wiki à propos de celle-ci, et notez notamment les limitations d'une telle intégration) :

- les groupes/rôles sont construits via des règles sur l'eppn uniquement ...

Nous souhaitons mettre en place l'encodage des cartes pour du contrôle d'accès basé sur Desfire

ESUP-SGC et ESUP-NFC-TAG permettent cette mise en oeuvre de l'encodage Desfire pour des usages très sécurisés comme le contrôle d'accès.

C'est la partie la plus délicate de l'installation d'une telle plateforme dans un établissement ou un groupement d'établissements : Desfire est assez éloignée de nos coeurs de métiers dans nos SI !

Pour une telle mise en oeuvre, nous vous conseilleurs de prendre connaissance et lire attentivement les documentations suivantes sur les espaces ESUP-SGC / ESUP-NFC-TAG :

- Tags NFC - getting started
- Contrôle d'accès à l'Université de Rouen Normandie
- Configuration Desfire avancée
- Configuration spécifique COMUE Normandie Université

Avant toute chose cependant, assurez-vous d'être en possession de la "master key" de vos cartes !

Sans cette clef, vous n'aurez pas la possibilité d'ajouter des "applications Mifare Desfire" sur vos cartes.

J'ai besoin de connaître la master key de ma carte ?

La "master key" de la carte Mifare Desfire est une clef (DES ou AES) qui permet d'ajouter des "applications Desfire" pour du contrôle d'accès ou tout autre type d'usage.

Ces applications Desfire peuvent être positionnées par esup-nfc-tag si vous avez en votre possession la master key de la carte.

Pour plus d'information sur NFC et Mifare Desfire, vous pouvez consulter la page wiki Tags NFC - getting started.

Vous avez donc besoin de connaitre effectivement la master key de votre carte uniquement si vous souhaitez ajouter des applications Mifare Desfire dans vos cartes.

Cependant nous vous conseillons fortement d'avoir la main sur vos propres cartes, c'est à dire de faire en sorte d'avoir en votre possession la master key de vos cartes.

Quelle est la master key de ma carte ?

Les cartes Mifare Desfire vierges ont une master key en DES à 0, sa représentation en hexa est 0000000000000000.

Dès qu'on encode une carte, il est d'usage de prendre la main sur la carte et de modifier cette master-key par défaut par une clef différente, usuellement en AES (car plus sécurisé).

Si vos cartes sont pré-encodées ou ont été encodées par un sous-traitant, **demandez leur de positionner une master-key unique pour l'ensemble de vos cartes (et d'en avoir connaissance)**.

Comment avoir la master-key sur des cartes pré-encodées CROUS ?

On conseille de faire **pré-encoder vos cartes avec l'application CROUS/IZLY**, cela vous évite de manipuler des clefs sam, dll windows, etc.

Au moment du pré-encodage, pour des questions de sécurité, une master-key est positionnée par l'entreprise qui encode vos cartes.

Par défaut cette master-key est diversifiée, non unique pour l'ensemble de vos cartes, et fonction de la clef SAM crous ; en d'autres termes cette master key vous est inconnue et vous ne pouvez pas alors ajouter de nouvelles applications avec votre esup-sgc / esup-nfc-tag.

Aussi, **on vous recommande fortement de demander à positionner une master-key sur vos cartes**. Pour ce faire, vous devez transmettre cette clef en suivant la procédure **"MasterKeyGenerator : Génération & communication de clés"** mise au point par le CROUS : **téléchargement ici.**

Cette procédure, accompagnée d'un programme java (jar) vous permet d'obtenir votre master-key à positionner et 3 fichiers supplémentaires, qui assemblés, permettent au prestataire de positionner la master-key sur vos cartes (sans pour autant en avoir connaissance).

Ces 3 fichiers doivent être envoyés par mail chiffrés (c'est le prestataire qui vous donne les 3 mails et 3 clefs de chiffrement respectives à utiliser pour ce faire).

Quels types de cartes ESUP-NFC-TAG est-il capable d'encoder ?

ESUP-NFC-TAG propose l'encodage (et décodage/lecture sécurisée) des cartes Mifare Desfire ; c'est à dire les cartes actuellement les plus utilisées dans nos établissements car considérées comme les plus sécurisées et permettant ainsi de proposer en confiance un moyen de paiement et de contrôle d'accès.

De fait ESUP-NFC-TAG supporte à la fois les cartes Mifare Desfire EV1 et Mifare Desfire EV2.

Peut-on utiliser esup-nfc-tag pour modifier la master key d'une carte

Dans certains cas (par exemple l'utilisation de carte vièrge dans le cadre du SGC) il peut être nécessaire de positionner une cle master sur une carte avant de pouvoir l'encoder normalament (ceci sans passer par les différents contrôles proposés par esup-nfc-tag). En effet esup-nfc-tag propose, dans son fonctionnement le plus courant, de contrôler l'existence de la carte puis de vérifier sa validité dans le système d'information (tagIdCheck et validateTag).

L'idée ici est de shunter ces deux étapes en utilisant les implémentations Dummy proposées dans esup-nfc-tag-server. Ces implémentations valident systématiquement les tags en retournant "true" quelle que soit la carte badgée.

Il est donc possible de créer un bean AppliExtApi reprenant l'implémantation AppliExtDummy en précisant un nom de salle (location) personnalisé.

De plus il faut déclarer un DesfireWriteConfig et un DesfireTag qui décrivent les actions à opérer sur la carte.

Dans ce cas il faut donc ajouter la configuration suivante:

applicationContext-desfire.xml (pour modifier la master key par defaut par une clé AES)

```
<bean id="desfireChangeMasterKeyTagEsupSgc" class="org.esupportail.nfctag.beans.DesfireTag" p:
formatBeforeWrite="false" p:keyStart="0000000000000000" p:keyTypeStart="DES" p:keyFinish="
12345678901234567890123456789012" p:keyTypeFinish="AES" p:keyVersionFinish="01">
</bean>

<bean id="desfireAuthConfigChangeKeyMasterEsupSgc" class="org.esupportail.nfctag.service.api.impl.
DesfireWriteConfig">
    <property name="desfireTag" ref="desfireChangeMasterKeyTagEsupSgc" />
    <property name="description" value="Changement de la Master Key"/>
</bean>
```

applicationContext-custom.xml (ajout d'une application dummy dediée)

```
<bean id="dummyChangeMasterKeyExtApi" class="org.esupportail.nfctag.service.api.impl.AppliExtDummy">
          <property name="description" value="Changement Master Key"/>
          <property name="locationsNames">
                  <util:list>
                          <value>Change Master Key</value>
                  </util:list>
          </property>
</bean>
```

Puis il faut créer une application esup-nfc-tag au niveau de l'IHM en utilisant les paramètres suivants :

- Nom : Change Master Key
- Configuration NFC : Changement de la Master Key
- Application externe : Changement Master Key
- Contrôle du tagId : TagIdCheckDummy

Quelle est la différence entre Mifare Desfire EV1 et Mifare Desfire EV2 ?

Mifare Desfire EV2 est la dernière génération de Mifare Desfire, elle fait suite à Mifare Desfire EV1.

EV2 est présentée comme plus sécurisée que EV1.

Il faut cependant distinguer la carte du protocole utilisé.

Aussi retenons que :

- EV2 est donc à la fois des cartes EV2 et un nouveau protocole EV2.
- Les cartes Desfire EV2 supportent le protocole EV1 et le protocole EV2.
- Les cartes Desfire EV1 ne supportent que le protocole EV1
- Le protocole EV2 est plus sécurisé (et plus complexe du coup) que le protocole EV1.
- Les nouvelles possibilités offertes par les cartes EV2 (application déléguée et libération de la mémoire)
    - ne sont pas supportées par les cartes EV1
    - mais peuvent être codées avec le protocole EV1.
- ESUP-SGC au travers d'ESUP-NFC-TAG utilise le protocole EV1 (et est donc compatible avec les cartes EV1 et EV2).

Codées avec ESUP-SGC / ESUP-NFC-TAG (via le protocole EV1), l'usage des cartes EV2 avec le protocole EV2 (par exemple au travers du contrôle d'accès) apporteraient de fait une plus grande sécurité que son usage au travers du protocole EV1.
Cf la documentation NXP https://www.nxp.com/docs/en/fact-sheet/MIFARE-DESFIRE-EV2-FS.pdf : "Proximity Check protects against relay attacks".

ESUP-SGC / ESUP-NFC-TAG, dans le cadre du projet de Carte Etudiante Européenne devrait également prochainement (en cours d'implémentation) supporter au niveau de l'encodage les nouvelles possibilités offertes par EV2, à savoir le support des applications déléguées et de la libération de la mémoire.

Est-ce que ESUP-SGC peut coder la DEUInfo de la care étudiante européenne ?

Oui, esup-sgc peut être configuré pour coder l'application DEUInfo dans un carte Mifare Desfire.

La documentation pour ce faire est donnée ici : Carte étudiante européenne

Est-ce qu'on peut demander à un prestataire de se charger de coder la DEUInfo ?

Non, l'implémentation d'esup-sgc fait qu'on ne peut pas déléguer cette partie à un prestataire.

La DEUInfo (Data European University Info) consiste notamment à coder l'ESCN (European Student Card Number) dans la carte.

Le QRCode utilisé par esup-sgc correspond à l'ESCN inclu dans une url en esc.gg

Le fonctionnement d'esup-sgc (qui imprime et encode en 2 temps) rattache ce qui est imprimé à la partie électronique en utilisant le QR Code d'une part et le CSN d'autre part.

Du coup l'association escn/csn ne peut pas se faire en dehors d'esup-sgc, et donc les établissements utilisant esup-sgc ne peuvent pas demander à un prestataire de pré-encoder la DEUInfo pour eux.

Est-ce qu'esup-sgc peut positionner des DAM keys sur les cartes supportant Mifare DesFire EV2 ?

Cf la documentation sur l'implémentation de la Carte étudiante européenne, esup-sgc peut charger les clefs DAM (Delegated Application Management) sur la carte lors de son encodage.

Ces clefs sont diversifiées pour chaque carte et esup-sgc les stocke en base de données pour pouvoir les utiliser lors d'éventuels échanges permettant à un établissement partenaire de créer une de ses application Mifare Desfire sur une carte issue d'esup-sgc.

Peut-on "virtualiser" la carte multi-services ?

Pour simplifier la gestion de cartes, pour des problèmes d'approvisionnement, pour des raisons éventuellement économiques et écologiques, on peut se demander si on peut "virtualiser" la carte multi-services.

Comme les téléphones proposent, pour une partie d'entre eux du moins, la technologie de sans contact (NFC), l'idée est en effet séduisante de pouvoir utiliser un téléphone au même titre qu'une carte Mifare Desfire pour badger sur les lecteurs NFC des différents services : contrôles d'accès, imprimantes, bibliothèques, restaurant universitaire ...

Concernant le NFC, et confère la page Tags NFC - getting started sur ce même wiki, il faut distinguer 2 grands usages des cartes Mifare Desfire dans nos établissements.

L'usage du simple numéro de série de la carte (CSN).
C'est l'usage le plus simple, non sécurisé (que l'on peut estimer comme acceptable dans certains cas d'usages ; certainement pas pour le contrôle d'accès ou le paiement cependant), non spécifique à Mifare Desfire, on peut penser sa retranscription / émulation aisée au travers d'un téléphone NFC. Il n'en est rien. Pour ne prendre que le système le plus répandu du marché, un Android officiel (non modifié/rooté) propose un CSN aléatoire (qui change a priori à chaque redémarrage de l'OS). De fait, il ne peut pas être utilisé comme identifiant au même titre qu'un CSN (fixe) d'une carte. On notera qu'en rootant un Android, selon la puce NFC du téléphone, on pourra fixer le CSN en le choisissant soi-même (ce qui permet par ailleurs d'usurper une carte pour un badgeage se basant sur le CSN fixe).

L'usage du protocole Mifare Desfire (et en excluant l'usage conjoint d'un CSN fixe ... ça exclue d'ailleurs de fait l'application Desfire de la Carte Etudiante Européenne qui propose la diversification de clef via le CSN ainsi que la signature via le CSN).
*Théoriquement** on peut estimer pouvoir émuler une carte Mifare Desfire avec un Android. ESUP-NFC-TAG lui-même propose avec esup-nfc-tag-droid de jouer des APDU NFC Mifare Desfire au travers d'un téléphone pour interagir avec une carte Mifare Desfire. Jouer les APDU de la carte plutôt que ceux du lecteur n'est donc *théoriquement** qu'une affaire d'implémentation des calculs des APDU, en se basant en plus sur les mêmes algorithmes de chiffrement déjà en place dans esup-nfc-tag-server. Si l'on souhaite ne pas donner les clefs des applications Mifare Desfire au téléphone (à proscrire pour des raisons de sécurité : les clefs manipulées directement par un logiciel dans un téléphone ne pouvant pas être considérées comme sécurisées - une personne malveillante étant susceptible de pouvoir les récupérer via différents scénarios), on peut en effet imaginer reprendre/étendre esup-nfc-tag : les apdu desfire sont calculés par le serveur esup-nfc-tag-server et esup-nfc-tag-droid ne fait que relais (proxy) avec le lecteur NFC de contrôle d'accès (sans jamais avoir connaissance des clefs notamment).
Cela présuppose que le téléphone ait une connexion internet pour dialoguer avec le serveur esup-nfc-tag-server au moment du badgeage de la carte. Le problème que cela pose également est qu'on se place alors dans un scénario qui ressemble à une attaque de type Man-in-the-Middle pour du contrôle d'accès sans contact : on se retrouve en effet à utiliser un téléphone pour faire proxy avec un mécanisme sécurisé (dans notre cas le serveur et non la carte d'une victime) permettant d'ouvrir une porte. C'est ce type de scénario et donc d'usage que NXP tente d'endiguer avec les évolutions de Mifare Desfire (EV1, EV2 puis maintenant EV3). En plus du problème technique que cela pourrait poser (avec des systèmes sophistiqués visant justement à empêcher ce type de pratique pour raisons de sécurité), on peut ainsi se demander si l'émulation d'une carte NXP Mifare Desfire est accepté/toléré par NXP, les solutions de contrôles d'accès, voire l'ANSSI.

** Notez bien que dans le parapgraphe ci-dessus, nous disons qu'on peut éventuellement émuler une carte Desfire depuis un Android **en théorie :** les possibilités de host-based card emulation (HCE) fourni nativement et officiellement par Google sur Android seraient si contraintes qu'elles ne permettraient en fait pas de proposer une émulation complète (cf par exemple cette question/réponse stackoverflow) et donc par exemple compatible avec les systèmes de controle d'accès du marché.*

Implémenter l'émulation Desfire (sur Android comme sur iOS) fait donc face à plusieurs problèmes : contraintes imposées par Apple/Google, problèmes de conformité (les cartes Mifare Desfire sont certifiées par l'ANSSI), de sécurité, de légalité (vis-à-vis de NXP).
En lien direct avec Google, NXP propose**rait** une émulation Desfire ? sur Android (via Google Pay) au travers de la solution MIFARE 2GO ; cette solution permet déjà d'utiliser son Android pour utiliser les transports public dans quelques villes (San Francisco, Whashington, Melbourne ...).

L'usage du protocole ISO/IEC 7816-4
Si l'émulation complète de Mifare Desfire est délicate, l'implémentation de commandes ISO/IEC 7816-4 semble plus à portée et plus en cohérence avec les possibilités d'HCE proposé par Android.
Aussi on peut imaginer proposer des services accessibles à l'utilisateur au travers d'une application Android émulant une "carte", notamment si on opère à la fois l'application cliente (Android) et serveur (lecteur NFC rattaché au service).
En ce sens l'émulation de la mise à disposition du fichier ESCN de la DEUInfo en ISO/IEC 7816-4 pourrait par exemple être envisagé (non étudié ni implémenté).

Au vu de ces considérations techniques, il parait donc préférable de recentrer le problème sur le besoin (fonctionnel) de départ et donc de reformuler par exemple la question en : **comment peut-on se passer de la carte ?**

Suivant les services considérés, on trouve des solutions déjà en place.

- Le CROUS propose une application Android spécifique permettant de régler un repas via QR-Code.
- esup-emargement propose la possibilité d'émarger avec un QR-Code (proposée par l'interface web d'esup-emargement, sans application telephone supplémentaire à installer) ; l'interface web permet aussi au gestionnaire de cocher une simple case pour palier l'oubli d'une carte par exemple.
- Les copieurs permettent une authentification par login/password (en plus de l'authentification/identification par badgeage).
- Dans les BUs, les documentalistes peuvent retrouver une personne via le nom/prénom.
- Pour le contrôle d'accès, on note que les portes peuvent être déverrouillées par clef via une serrure classique (en plus du badgeage donc) ; des contrôles d'accès permettent de déverrouiller des portes à distance également (via l'interface web, et donc sans carte).
- ...

Est-ce qu'esup-nfc-tag peut mettre à jour électroniquement une carte Mifare Desfire ?

Oui, ESUP-NFC-TAG est prévu pour ajouter une application Desfire sur des cartes déjà en circulation.

La configuration dans esup-nfc-tag-server correspondra quasiment à l'identifique à la configuration d'une édition de carte (structure Desfoire de la carte).

Cette configuration sera à utiliser au travers non plus d'esup-sgc-client mais d'esup-nfc-tag-droid ou esup-nfc-tag-desktop ; l'enrôlement via reconnaissance du qr-code n'étant ici pas nécessaire.

Aussi, pratiquement, dans ESUP-SGC, l'ajout d'une application de contrôle d'accès peut revenir alors à un simple badgeage de la carte avec un téléphone Android.

Est-ce qu'esup-nfc-tag peut ré-encoder complètement une carte Mifare Desfire ?

Techniquement, esup-sgc peut "réencoder" une carte en la formattant avant, c'est une option à spécifier dans la configuration de la structure de la carte dans esup-nfc-tag.

Si vous achetez des cartes pré-encodées avec crous/izly (cf Q/R "Utilisation de cartes pré-encodées" dans cette même FAQ) cependant, cela veut dire que l'application crous/izly n'est pas encodée par esup-sgc et que donc si esup-sgc formatte et réencode la carte, l'application crous/izly ne sera plus présente sur la carte.

Encoder soit-même la partie crous/izly via esup-sgc présente donc cet intérêt de pouvoir relancer un encodage en cas de problème. Si vous utilisez actuellement un SGC qui présente une instabilité lors de l'édition des cartes et notamment lors de leur encodage, pouvoir ré-encoder complètement une carte semble séduisant. L'encodage dans ESUP-SGC est cependant fiable (cf question ci-dessous sur le taux de perte lors de l'édition des cartes) ; aussi, les avantages qu'on peut retirer d'acheter des cartes pré-encodées crous/izly l'emportent (malgré le surcoût d'achât des cartes pour cette option de pré-encodage) sur cette possibilité : maintenance simplifiée, pas d'adhérence à la dll crous (et donc à windows), distribution logicielle facilitée, pas de gestion des clefs sam, ...

... ajoutons que la mise en oeuvre d'un pré-encodage crous/izly par un seul prestataire offre un niveau de sécurité plus élevé pour l'ensemble des établissements, c'est ainsi la recommandation du CNOUS.

Quel est le taux de perte de cartes lors de l'édition dans ESUP-SGC ?

Les SGC du marché proposent usuellement une édition comprenant l'impression et l'encodage de la carte en 1 seul passage dans une imprimante (faisant office également d'encodeur).

Cf la Q/R "L'impression des cartes nécessite de faire deux passages de carte ?", cela est possible en utilisant les APIs d'un modèle d'imprimante à carte spécifique (avec encodeur intégré). Dans les faits, outre les problèmes de maintenance (logiciel testé/validé uniquement pour une version/type d'OS, ... parfois plus maintenu, ...), les établissements utilisant ce type de Système de Gestion de Cartes constatent des taux d'échec/perte de cartes importants lors de l'édition (jusqu'à 30% de perte pour certains).

Avec esup-sgc / esup-nfc-tag (qui ne fait que du standard / basique lors de l'édition des cartes : impression ps/pcsl et encodage nfc/pcsc) **le taux de perte attendu est proche de 0, tant sur l'impression que sur l'encodage**.
(Notons que l'usage d'une ZXP3 pour encoder en masse les cartes, malgré l'usage de l'API propriétaire Zebra [usage la plus modérée possible cependant], se révèle également très fiable via ESUP-SGC).

Si, avec esup-sgc, vous constatez un taux de perte de l'ordre de 5% par exemple, ce n'est pas normal.

Problème d'impression

Si le problème se pose au niveau de l'impression, l'imprimante ou les rubans sont sans doute en cause (ou/et drivers et configurations associées).

Problème d'encodage

Si le problème se pose lors de l'encodage, ça peut être un problème humain (encodage avorté en 'arrachant' la carte du lecteur NFC alors qu'elle est en train d'être encodée ; on retrouve alors l'erreur 0x4d3).

Si le problème n'est pas humain mais technique, il est intéressant de retrouver l'erreur 'source'.

Une erreur 91DE DUPLICATE_ERROR correspond à une tentative d'encodage d'un élément déjà encodé : lorsque vous avez cette erreur c'est que l'encodage de la carte avait déjà été effectuée partiellement (partiellement car sinon le SGC ne détecterait plus la carte comme carte à encoder puisque marquée comme 'encodé' dans le SGC), et cet encodage n'avait pas abouti et l'erreur source (à prendre en compte) est donc donnée lors de cette première tentative d'encodage.

Si le problème technique n'est pas aléatoire mais systématique pour une carte/personne donnée, ça peut être un problème de configuration de la structure de la carte (erreur type 'boundary error' car on tente d'encoder plus de données que la taille du fichier par exemple ...)

Si le problème technique est "aléatoire", la piste la plus probable est un encodeur (ou/et driver associé) pas très stable. Certains modèles sont en effet plus fiables que d'autres (avec des drivers bien supportés sur windows comme sur linux également) ; ça peut également être lié à un poste et système d'exploitation à bout de souffle, ports USB HS, etc.