Configuration Desfire avancée

- applicationContext-desfire.xml
- applicationContext-custom.xml
    - Ajout d'une configuration NFC pour une authentification en DESFIRE (NfcAuthConfig)
    - Ajout d'une application d' affichage du verso de la carte (AppliExtApi)
- Mise à jour électronique de cartes
    - Mise à jour classique
    - Mise à jour par essais successifs

esup-nfc-tag est capable de lancer les commandes Desfire EV1, ces commandes (APDU) sont calculées par esup-nfc-tag-server et transmises une à une jusqu'au lecteur NFC grâce aux différents clients qui ne font que transmettre les commandes et retourner les réponses à ces commandes.

Le code qui permet cela se situe dans la classe DESFireEV1Service.java d'esup-nfc-tag-server

Cette implémentation Desfire EV1 permet ainsi de lire, écrire ou mettre à jour des cartes en relation avec esup-sgc notamment : les informations à écrire /mettre à jour dans les fichiers Desfire pouvant être récupérées/interprétées par/depuis esup-sgc.

Vous verrez ci-dessous que les configurations XML à la Spring reprenent pour l'écriture **et/ou** la mise à jour des cartes Desfire la structure d'une carte Desfire (master key, applications, fichiers et clefs) avec des configurations typiques de Desfire (amks pour application master key settings, nok, etc.).

On y explique au passage quelques élements de ces configurations bien sûr, et les intégrateurs ayant une bonne connaissance de Desfire s'y retrouveront sans problème.

Cependant, Desfire EV1 reste une technologie NXP relativement fermée, même si des librairies et documentations sont librement accessibles sur internet aujourd'hui (et depuis près de 20 ans maintenant).

Reste donc que pour avoir une vision et une compréhension exacte de la structuration Desfire EV1, un document comme le "datasheet" complet de "mifare DESFire Contactless Multi-Application IC with DES and 3DES Security MF3 IC D40" peut s'avérer utile à consulter. Ce document disponible depuis le nom de fichier M075031_desfire.pdf est un document d'Avril 2004 qui est noté comme confidentiel et qui peut normalement être consulté uniquement après signature d'un accord de non divulgation (NDA), comme le relève ce thread dans le forum de mifare.net : https://www.mifare.net/support/forum/topic/desfire-ev1-aes-2k3des-clrc663-examples/

Note : cf la question "Quelle est la différence entre Mifare Desfire EV1, Mifare Desfire EV2 ou encore Mifare Desfire EV3 ? dans la FAQ", esup-nfc-tag utilise Desfire EV1 comme protocole lors de l'écriture ce qui lui permet notamment de pouvoir écrire sur les EV1, EV2 ou EV3 ; les donnéés écrites peuvent ensuite êtreutilisées par n'importe quel autre protocole Desfire (EV1, EV2 ou EV3) en fonction des possibilités de la carte : une compatibilité totale est donc assurée avec l'ensemble des cartes de la famile Desfire.

applicationContext-desfire.xml

Dans le ficher applicationContext-desfire.xml on trouve la structure et les données qui seront écrites sur la carte lors de l'encodage.

En premier lieu on déclare un bean de type DesfireWriteConfig qui ajoutera une application externe (AppliExtApi) d'écriture. Cette "api" sera disponible dans l'interface graphique pour permettre de configurer une application d'encodage comme décrit ici :

https://www.esup-portail.org/wiki/display/SGC/Configurations+ESUP-SGC+et+ESUP-NFC-TAG-SERVER#ConfigurationsESUP-SGCetESUP-NFC-TAG-SERVER-ESUP-NFC-TAG-SERVER

On lui passe en argument un objet DesfireTag qui décrit le format de la carte en elle même (structure et données) :

- p:formatBeforeWrite précise si la carte doit être formatée avant l'encodage
- p:keyStart spécifie la PICC Master Key qui sera utilisée pour s'authentifier sur la carte **avant** l'encodage
- p:keyTypeStart est le type de la clé (DES ou AES) pré-positionnée sur la carte **avant** l'encodage
- p:keyFinish spécifie la PICC Master Key qui sera positionnée sur la carte **après** l'encodage (si pas de changement de clé, il faut saisir la même valeur de pour p:keyStart)
- p:keyTypeFinish est le type de la clé (DES ou AES) positionnée sur la carte **après** encodage (si pas de changement de clé, il faut saisir la même valeur de pour p:keyTypeStart)
- p:keyVersionFinish est le numéro de version attribué à la nouvelle clé en fin d'encodage (n'a aucune incidence sur le fonctionnement de la carte mais peut permettre de s'assurer que l'encodage est bien complet)

Voici un exemple complet :

```
<bean id="desfireAuthConfigComueWriteEsupSgc" class="org.esupportail.nfctag.service.api.impl.DesfireWriteConfig"
>
        <property name="desfireTag" ref="desfireComueTagEsupSgc" />
        <property name="description" value="Ecriture ESUP SGC"/>
</bean>


<bean id="desfireComueTagEsupSgc" class="org.esupportail.nfctag.beans.DesfireTag" p:formatBeforeWrite="false" p:
keyStart="0000000000000000" p:keyTypeStart="DES" p:keyFinish="0000000000000000" p:keyTypeFinish="DES" p:
keyVersionFinish="00">
 <property name="applications">
  <util:list>
   <bean class="org.esupportail.nfctag.beans.DesfireApplication" p:desfireAppId="AB0123" p:amks="0B" p:nok="84">
    <property name="files">
     <util:list>
      <bean class="org.esupportail.nfctag.beans.DesfireFile" p:fileNumber="00" p:communicationSettings="03" p:
accessRights="1223" p:fileSize="1F0000" p:tagWriteApi-ref="idp2sTagWriteEsupSgc"/>
     </util:list>
    </property>
    <property name="keys">
     <util:list>
      <bean class="org.esupportail.nfctag.beans.DesfireKey" p:keyNo="00" p:keyVer="01" p:key="
00000000000000000000000000000000"/>
      <bean class="org.esupportail.nfctag.beans.DesfireKey" p:keyNo="01" p:keyVer="01" p:key="
00000000000000000000000000000000"/>
      <bean class="org.esupportail.nfctag.beans.DesfireKey" p:keyNo="02" p:keyVer="01" p:key="
00000000000000000000000000000000"/>
      <bean class="org.esupportail.nfctag.beans.DesfireKey" p:keyNo="03" p:keyVer="01" p:key="
00000000000000000000000000000000"/>
     </util:list>
    </property>
   </bean>
  </util:list>
 </property>
</bean>
```

Une clé de type p:keyStart="0000000000000000" p:keyTypeStart="DES" est une clé PICC par défaut pour une carte Desfire. Avec cette clé la carte reste formatable facilement à l'aide d'un smartphone par exemple.

Si l'on détail, l'objet DesfireTag peut contenir une liste de DesfireApplication.

Voici la structure d'une application DesfireApplication:

```
<bean class="org.esupportail.nfctag.beans.DesfireApplication" p:desfireAppId="AB0123" p:amks="0B" p:nok="84">
 <property name="files">
  <util:list>
   <bean class="org.esupportail.nfctag.beans.DesfireFile" p:fileNumber="00" p:communicationSettings="03" p:
accessRights="1223" p:fileSize="1F0000" p:tagWriteApi-ref="idp2sTagWriteEsupSgc"/>
  </util:list>
 </property>
 <property name="keys">
  <util:list>
   <bean class="org.esupportail.nfctag.beans.DesfireKey" p:keyNo="00" p:keyVer="01" p:key="
00000000000000000000000000000000"/>
   <bean class="org.esupportail.nfctag.beans.DesfireKey" p:keyNo="01" p:keyVer="01" p:key="
00000000000000000000000000000000"/>
   <bean class="org.esupportail.nfctag.beans.DesfireKey" p:keyNo="02" p:keyVer="01" p:key="
00000000000000000000000000000000"/>
   <bean class="org.esupportail.nfctag.beans.DesfireKey" p:keyNo="03" p:keyVer="01" p:key="
00000000000000000000000000000000"/>
  </util:list>
 </property>
</bean>
```

Dans cet exemple, l'application a pour identifiant p:desfireAppId="AB0123", elle est configurée avec **p:nok** à 84 qui indique 4 clés AES

p:amks="0B" correspond à "application master key settings", la gestion des droits sur l'application (0B indique que la configuration de l'application est modifiable à condition de s'authentifier avec la clé 0 de l'application, qu'on appelle aussi clef master/maitre de l'application Desfire)

L'application peut contenir des fichiers de type DesfireFile et des clés DesfireKey

Dans notre cas l'application possède un fichier numéro "00" (p:fileNumber="00")

La taille du fichier est 1F0000, soit 31 octets (on lit pair par pair de droite à gauche : 00001F, soit 31).

Le ficher est paramétré avec les autorisations suivantes **p:accessRights**="1223" :

- clé 1 pour la lecture
- clé 2 pour l'écriture
- clé 2 pour lecture et écriture
- clé 3 pour le changement d'autorisations

La clé 0 est la Master Key de l'application, dans l'exemple toutes les clés sont à zéro et insérées en tant que version "01" (p:keyVer="01")

Les types de clé sont déterminés par la valeur de l'attribut **p:nok** de la DesfireApplication :

- le premier chiffre représente de type (0 : DES, 8 : AES)
- le deuxième est le nombre de clés

La valeur à écrire dans le fichier est définie par p:tagWriteApi-ref

Il faut configurer l'url qui permet de récupérer cette valeur (ici c'est esup-sgc qui fournit la donnée) :

```
<bean id="idp2sTagWriteEsupSgc" class="org.esupportail.nfctag.service.api.impl.TagWriteRestWs">
 <property name="idFromEppnInitUrl" value="https://esup-sgc-test.univ-rouen.fr/wsrest/nfc/idFromEppnInit"/>
</bean>
```

applicationContext-custom.xml

Ajout d'une configuration NFC pour une authentification en DESFIRE (NfcAuthConfig)

```
<bean id="desfireAuthSiServices" class="org.esupportail.nfctag.service.api.impl.DesfireReadConfig">
        <property name="desfireKeyNumber" value="01"/>
        <property name="desfireFileNumber" value="00"/>
        <property name="desfireAppId" value="AB0123"/>
        <property name="desfireAppName" value="si-service"/>
        <property name="desfireFileOffset" value="000000"/>
        <property name="desfireKey" value="00000000000000000000000000000000"/>
        <property name="description" value="Authentification SI Service COMUE"/>
</bean>
```

Avec cet exemple le client ira lire le fichier desfireFileNumber "00" de l'application desfire desfireAppId : AB0123 présente sur la carte à l'aide de la clé desfireKey : 00000000000000000000000000000000.
Le desfireFileOffset permet de préciser à partir de quel octet on démarre la lecture.
Esup-nfc-tag-server enverra le contenu du fichier desfire ainsi que le desfireAppName : si-service (dans notre cas) à l'application TagIdCheck paramétrée dans esup-nfc-tag-server. Je web-service tagIdCheck associé à cette application doit contôler si une carte correspond effectivement à l'identifiant contenu dans le fichier defire et doit renvoyer un tagLog (voir Implémentation du Web Service TagIdCheck)

Ajout d'une application d' affichage du verso de la carte (AppliExtApi)

```
    <bean id="esupSgcVersoExtApi" class="org.esupportail.nfctag.service.api.impl.AppliExtRestWs">
            <property name="isTagableUrl" value="https://esup-sgc-test.univ-ville.fr/wsrest/nfc/isTagable"/>
            <property name="validateTagUrl" value="https://esup-sgc-test.univ-ville.fr/wsrest/nfc/validateTag"/>
            <property name="getLocationsUrl" value="https://esup-sgc-test.univ-ville.fr/wsrest/nfc
/locationsVerso"/>
            <property name="displayUrl" value="https://esup-sgc-test.univ-ville.fr/wsrest/nfc/verso"/>
            <property name="description" value="Web Service Verso"/>
    </bean>
```

La nouvelle propriété displayUrl permet de définir l'emplacement des données qui seront affichées suite à la validation d'un tag. Esup-sgc peut retourner le verso en fonction du csn via cette adresse https://esup-sgc-test.univ-ville.fr/wsrest/nfc/verso.

Mise à jour électronique de cartes

Mise à jour classique

Esup-nfc-tag, en plus d'encoder les cartes, notamment en lien avec esup-sgc, peut permettre de mettre à jour vos cartes Desfire : il peut vous permettre d'ajouter des applications Desfire sur des cartes qui ont été prélablement éditées, ce si vous avez connaissance de la master-key.

Dans l'exemple ci-dessous, on montrer comment on peut déclencher des ajouts d'applications en fonction de dernières dates d'encodage (ou mises à jour élecroniques) de cartes.

Ici les dates de mises à jour électroniques sont récupérées depuis esup-sgc et les données à écrire dans le fichier également.

Notez qu'on met que des clefs AES 0000... pour les clefs des applications. Et on a gardé la master-key par défaut des cartes Desfire : clef DES 0000....

```
    <bean id="idAcTagWriteEsupSgc" class="org.esupportail.nfctag.service.api.impl.TagWriteRestWs">
                <property name="idFromCsnUrlTemplate" value="https://esup-sgc.univ-ville.fr/wsrest/nfc
/idFromCsn?csn={0}&appName=access-control"/>
    </bean>

    <bean id="csnDomainTagWriteEsupSgc" class="org.esupportail.nfctag.service.api.impl.TagWriteRestWs">
                <property name="idFromCsnUrlTemplate" value="https://esup-sgc.univ-ville.fr/wsrest/nfc
/idFromCsn?csn={0}&appName=csn-domain"/>
    </bean>


    <bean id="dateLastUpdateEsupSgc" class="org.esupportail.nfctag.service.api.impl.TagLastUpdateRestWs">
                <property name="wsUrl" value="https://esup-sgc.univ-ville.fr/wsrest/nfc/lastUpdateFromCsn"/>
    </bean>

<bean id="desfireComueTagUpdateEsupSgc" class="org.esupportail.nfctag.beans.DesfireTag" p:formatBeforeWrite="
false" p:keyStart="0000000000000000" p:keyTypeStart="DES" p:keyFinish="0000000000000000" p:keyTypeFinish="DES"
p:keyVersionFinish="00">
         <property name="applications">
                <util:list>
                        <bean class="org.esupportail.nfctag.beans.DesfireApplication"
                            p:desfireAppId="F585C1" p:amks="0B" p:nok="85" p:updateDate="2016-03-01 12:00" p:
tagLastUpdateRestWs-ref="dateLastUpdateEsupSgc">
                            <property name="files">
                                    <util:list>
                                            <bean class="org.esupportail.nfctag.beans.DesfireFile"
                                                  p:fileNumber="00" p:communicationSettings="03" p:
accessRights="1444" p:fileSize="1F0000" p:tagWriteApi-ref="idAcTagWriteEsupSgc"/>
                                            <bean class="org.esupportail.nfctag.beans.DesfireFile"
                                                  p:fileNumber="01" p:communicationSettings="03" p:
accessRights="2444" p:fileSize="1F0000" p:tagWriteApi-ref="idAcTagWriteEsupSgc"/>
                                            <bean class="org.esupportail.nfctag.beans.DesfireFile"
                                                  p:fileNumber="02" p:communicationSettings="03" p:
accessRights="3444" p:fileSize="1F0000" p:tagWriteApi-ref="idAcTagWriteEsupSgc"/>
                                    </util:list>
                            </property>
                            <property name="keys">
                                    <util:list>
                                            <bean class="org.esupportail.nfctag.beans.DesfireKey"
```

```
                                                    p:keyNo="00" p:keyVer="01" p:key="
00000000000000000000000000000000"/>
                                            <bean class="org.esupportail.nfctag.beans.DesfireKey"
                                                    p:keyNo="01" p:keyVer="01" p:key="
00000000000000000000000000000000"/>
                                            <bean class="org.esupportail.nfctag.beans.DesfireKey"
                                                    p:keyNo="02" p:keyVer="01" p:key="
00000000000000000000000000000000"/>
                                            <bean class="org.esupportail.nfctag.beans.DesfireKey"
                                                    p:keyNo="03" p:keyVer="01" p:key="
00000000000000000000000000000000"/>
                                            <bean class="org.esupportail.nfctag.beans.DesfireKey"
                                                    p:keyNo="04" p:keyVer="01" p:key="
00000000000000000000000000000000"/>
                                    </util:list>
                            </property>
                        </bean>
                           <bean class="org.esupportail.nfctag.beans.DesfireApplication"
                            p:desfireAppId="F585C2" p:amks="0B" p:nok="86" p:updateDate="2018-04-24 12:00" p:
tagLastUpdateRestWs-ref="dateLastUpdateEsupSgc">
                            <property name="files">
                                    <util:list>
                                            <bean class="org.esupportail.nfctag.beans.DesfireFile"
                                                  p:fileNumber="00" p:communicationSettings="03" p:
accessRights="1FFF" p:tagWriteApi-ref="csnDomainTagWriteEsupSgc"/>
                                            <bean class="org.esupportail.nfctag.beans.DesfireFile"
                                                  p:fileNumber="01" p:communicationSettings="03" p:
accessRights="2FFF" p:tagWriteApi-ref="csnDomainTagWriteEsupSgc"/>
                                            <bean class="org.esupportail.nfctag.beans.DesfireFile"
                                                  p:fileNumber="02" p:communicationSettings="03" p:
accessRights="3FFF" p:tagWriteApi-ref="csnDomainTagWriteEsupSgc"/>
                                            <bean class="org.esupportail.nfctag.beans.DesfireFile"
                                                  p:fileNumber="03" p:communicationSettings="03" p:
accessRights="4FFF" p:tagWriteApi-ref="csnDomainTagWriteEsupSgc"/>
                                            <bean class="org.esupportail.nfctag.beans.DesfireFile"
                                                  p:fileNumber="04" p:communicationSettings="03" p:
accessRights="5FFF" p:tagWriteApi-ref="csnDomainTagWriteEsupSgc"/>
                                    </util:list>
                            </property>
                            <property name="keys">
                                    <util:list>
                                            <bean class="org.esupportail.nfctag.beans.DesfireKey"
                                                  p:keyNo="00" p:keyVer="01" p:key="
00000000000000000000000000000000"/>
                                            <bean class="org.esupportail.nfctag.beans.DesfireKey"
                                                  p:keyNo="01" p:keyVer="01" p:key="
00000000000000000000000000000000"/>
                                            <bean class="org.esupportail.nfctag.beans.DesfireKey"
                                                  p:keyNo="02" p:keyVer="01" p:key="
00000000000000000000000000000000"/>
                                            <bean class="org.esupportail.nfctag.beans.DesfireKey"
                                                  p:keyNo="03" p:keyVer="01" p:key="
00000000000000000000000000000000"/>
                                            <bean class="org.esupportail.nfctag.beans.DesfireKey"
                                                  p:keyNo="04" p:keyVer="01" p:key="
00000000000000000000000000000000"/>
                                            <bean class="org.esupportail.nfctag.beans.DesfireKey"
                                                  p:keyNo="05" p:keyVer="01" p:key="
00000000000000000000000000000000"/>
                                    </util:list>
                            </property>
                        </bean>

                </util:list>
        </property>
    </bean>

    <bean id="desfireAuthConfigComueUpdateEsupSgc" class="org.esupportail.nfctag.service.api.impl.
DesfireUpdateConfig">
                <property name="desfireTag" ref="desfireComueTagUpdateEsupSgc" />
                <property name="description" value="Mise à jour ESUP SGC"/>
```

```
    </bean>
```

Mise à jour par essais successifs

Suivant l'historique de votre gestion de cartes (SGC différents, erreurs de la part de prestataires, misconfigurations, fusions d'établissements), vous pouvez potentiellement vous retrouver avec un parc de cartes en activité dans des états différents et sans même avoir la possibilité d'identifier l'état de chacune d'elles.

On a pu notamment être confrontés à des cartes qui avaient des master-key restées par défaut, ou qui ont pu être positionnées différemment d'une année sur l'autre ...

esup-nfc-tag est capable de proposer via une seule et unique salle de badgeage de faire des tentatives de mises à jour successives. Il joue ces tentatives les unes à la suite des autres, jusqu'à la dernière. Seul le "résultat" de la dernière tentative est pris en compte comme résultat global du badgeage. En configurant une telle procédure finement, cela peut vous permettre de proposer une seule et unique application de badgeage à vos usagers qui va en un seul badgeage permettre de remettre en règle (en 'conformité) la carte **quelque soit son état d'origine (== parmi tous les états d'origine connus /possibles)**.

Ces tentatives sont en fait autant de configurations de DesfireTag que voulu.

Voici un exemple d'une mise à jour par essais successifs - on se place dans un cas où on a des cartes qui peuvent avoir ou non une master-key en AES à 11111111111111111111111111111111 ou 22222222222222222222222222222222.
On souhaite ajouter une application F22221 avec fichier vide et des clefs (2 clefs DES ici 82). On souhaite positionner une master-key en AES à 33333333333333333333333333333333 et que ça badge valide ci effectivement on a finalement une master-key AES à 33333333333333333333333333333333

```
    <bean id="desfireComueTagUpdateEsupSgc" class="org.esupportail.nfctag.beans.DesfireTag" p:keyStart="
11111111111111111111111111111111" p:keyTypeStart="AES" p:keyFinish="33333333333333333333333333333333" p:
keyTypeFinish="AES" p:keyVersionFinish="00">
                    <property name="applications">
                            <util:list>
                    <bean class="org.esupportail.nfctag.beans.DesfireApplication"
                            p:desfireAppId="F22221" p:amks="0B" p:nok="82">
                            <property name="files">
                                    <util:list>
                                            <bean class="org.esupportail.nfctag.beans.DesfireFile"
                                                  p:fileNumber="00" p:communicationSettings="03" p:
accessRights="1111" p:fileSize="1F0000"/>
                                    </util:list>
                            </property>
                            <property name="keys">
                                    <util:list>
                                            <bean class="org.esupportail.nfctag.beans.DesfireKey"
                                                    p:keyNo="00" p:keyVer="01" p:key="
00000000000000000000000000000000"/>
                                            <bean class="org.esupportail.nfctag.beans.DesfireKey"
                                                    p:keyNo="01" p:keyVer="01" p:key="
00000000000000000000000000000000"/>
                                    </util:list>
                            </property>
                        </bean>
                </util:list>
                </property>
    </bean>

        <bean id="desfireComueTagUpdateEsupSgc2" class="org.esupportail.nfctag.beans.DesfireTag" p:keyStart="
22222222222222222222222222222222" p:keyTypeStart="AES" p:keyFinish="33333333333333333333333333333333" p:
keyTypeFinish="AES" p:keyVersionFinish="00">
                <property name="applications">
                        <util:list>
                                <bean class="org.esupportail.nfctag.beans.DesfireApplication"
                                          p:desfireAppId="F22221" p:amks="0B" p:nok="82">
                                    <property name="files">
                                            <util:list>
                                                    <bean class="org.esupportail.nfctag.beans.DesfireFile"
                                                              p:fileNumber="00" p:communicationSettings="
03" p:accessRights="1111" p:fileSize="1F0000"/>
                                            </util:list>
                                    </property>
```

```
                                        <property name="keys">
                                                <util:list>
                                                        <bean class="org.esupportail.nfctag.beans.DesfireKey"
                                                                p:keyNo="00" p:keyVer="01" p:key="
00000000000000000000000000000000"/>
                                                        <bean class="org.esupportail.nfctag.beans.DesfireKey"
                                                                p:keyNo="01" p:keyVer="01" p:key="
00000000000000000000000000000000"/>
                                                </util:list>
                                        </property>
                                </bean>
                        </util:list>
                </property>
        </bean>

        <bean id="desfireComueTagUpdateEsupSgcFinal" class="org.esupportail.nfctag.beans.DesfireTag" p:
keyStart="33333333333333333333333333333333" p:keyTypeStart="AES" p:keyFinish="33333333333333333333333333333333"
p:keyTypeFinish="AES" p:keyVersionFinish="00"/>


        <bean id="desfireAuthConfigComueUpdateEsupSgc" class="org.esupportail.nfctag.service.api.impl.
DesfireUpdateConfig">
            <property name="desfireTags">
                        <util:list>
                                <ref bean="desfireComueTagUpdateEsupSgc"/>
                                <ref bean="desfireComueTagUpdateEsupSgc2"/>
                                <ref bean="desfireComueTagUpdateEsupSgcFinal"/>
                        </util:list>
                </property>
        <property name="description" value="Mise à jour par essais successifs"/>
    </bean>
```