ARCHITECTURE DU LOGICIEL

Version :	0.2
Date :	02/01/13
Rédigé par :	Emmanuel Mocquet
Relu par :	
Approuvé par :	

MISES A JOUR

Version	Date	Modifications réalisées
0.1	26/11/12	Création
0.2	02/01/13	Complétion

1. Objet :

Ce document a pour but de décrire les solutions techniques conçues pour répondre aux exigences définies dans la spécification technique de besoin. Il doit identifier et définir les différents modules ou constituants du logiciel ainsi que leur interfaces de telle sorte que chacun d'entre eux puisse être développé de façon autonome par un ou plusieurs membre de l'équipe, avant d'être intégré.

Ce projet se découpe en deux parties distinctes. La première consiste à développer une ou plusieurs applications établissant la communication entre ce support et un PC. Il est attendu qu'au final, il puisse y avoir chiffrement, authentification et/ou stockage de données.

La seconde partie consiste à permettre à un utilisateur de contrôler les données qu'il dépose sur « Facebook ». Les détails consacrés à cette partie sont disponibles dans le document associé.

Nous décrirons ici la partie architecturale du projet « SmartCards ».

2. Documents applicables et de référence

- Le sujet de la première partie projet et sa description sont disponibles dans le document « sujet-cartes-a-puce.pdf » ;
- Le document de spécification technique de besoin s'intitule « Smartcards STB.pdf » ;

Le sujet propose plusieurs documents de référence :

- *Cartes à puce. Administration et utilisation,* LINUX Magazine Hors-Série no 39. Diamond Editions, Nov./Déc. 2008 ;
- *Cartes à puce. Découvrez leurs fonctionnalités et leurs limites*, MISC Hors-Série no 2. Diamond Editions, Nov./Déc. 2008.

3. Terminologie et sigles utilisés

- SDK: un Software Development Kit est un ensemble d'outils, de documents et des exemples permettant ou facilitant le développement d'applications.
- API : une Application programming Interface fournit un certain nombre de structures (fonctions, objets...) et/ou spécifie la façon dont doit être exécutée une certaine action.
- APDU : « un Application Protocol Unit est un message échangé entre une carte à puce et un lecteur de carte à puce. Il est normalisé et décrit dans l'ISO 7816 partie 4 ». (Wikipedia)

4. Architecture physique du matériel utilisé

Par la nature du projet, des cartes à puces et leurs lecteurs sont nécessaires. Ces cartes seront des Java Card avec 40K de mémoire EEPROM et les lecteurs seront des Omnikey 3121.

Afin d'établir un dialogue entre le lecteur et le système, nous utiliserons la bibliothèque PC/SC-Lite.

Le développement d'applications se fera sur des machines de type GNU/Linux. Pour cette raison, le SDK JavaCard utilisé sera au maximum en version 2.2, la dernière disponible (3) étant uniquement compatible avec Windows. Aussi, pour des raisons de compatibilité, le SDK Java deva être en version 1.5.

Nous utiliserons ainsi l'API fournie par ce SDK et le plugin d'Eclipse JCDE pour automatiser certaines étapes durant la conception des applications.

Leur gestion se fera grâce à l'outil GlobalPlatform, suivant lui-même le standard globalplatform.

5. Architecture des logiciels

5.1. Structure logique

Concernant le fonctionnement final, c'est-à-dire l'intégration avec l'application pour Facebook (qu'on appelera ici « Proxy »), les composants seront les suivants :

5.2. Description du constituant « Proxy »

Le proxy se servira de la carte à puce pour vérifier les identifants de l'utilisateur.
La description complète de ce composant est disponible dans le document associé à la partie Facebook.

5.3. Description du constituant « SoftCard »

Le composant SoftCard fera office de passerelle entre le proxy en traduisant les requêtes émises par celui-ici et en les transmettant à la carte. Il devra auparavant s'être authentifié auprès de cette dernière.
Il devra être suffisamment générique ou modulable pour pouvoir être utilisé sans proxy, c'est-à-dire dans une situation d'authentification par exemple.
Il sera écrit en Java et possèdera un secret partagé avec la carte.

5.4. Description du constituant «SmartCard»

La carte à puce contiendra les identifiants (login et mot de passe) de l'utilisateur du proxy Facebook.
Seront aussi stockés le certificat, la chaîne d'autorités associée et la clef privée du porteur de la carte.
Le programme installé sur la carte et permettant la maniupulation de ces données sera écrit en Java et installé sur la carte grâce au SDK JavaCard 2.2. Il sera de taille relativement réduite et sera peu complexe.

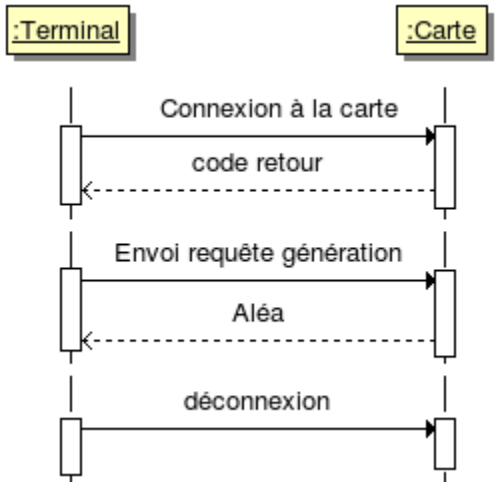
5.5. Justifications techniques

Le fait de stocker le certificat, sa chaîne de ceritications et la clef privée du porteur de la carte permettra dans un premier temps de vérifier l'identité du proxy et de pouvoir témoigner de l'identité de la carte (son porteur). Un dialogue pourra ensuite se faire avec le proxy, via un canal sécurisé.

6. Fonctionnement dynamique

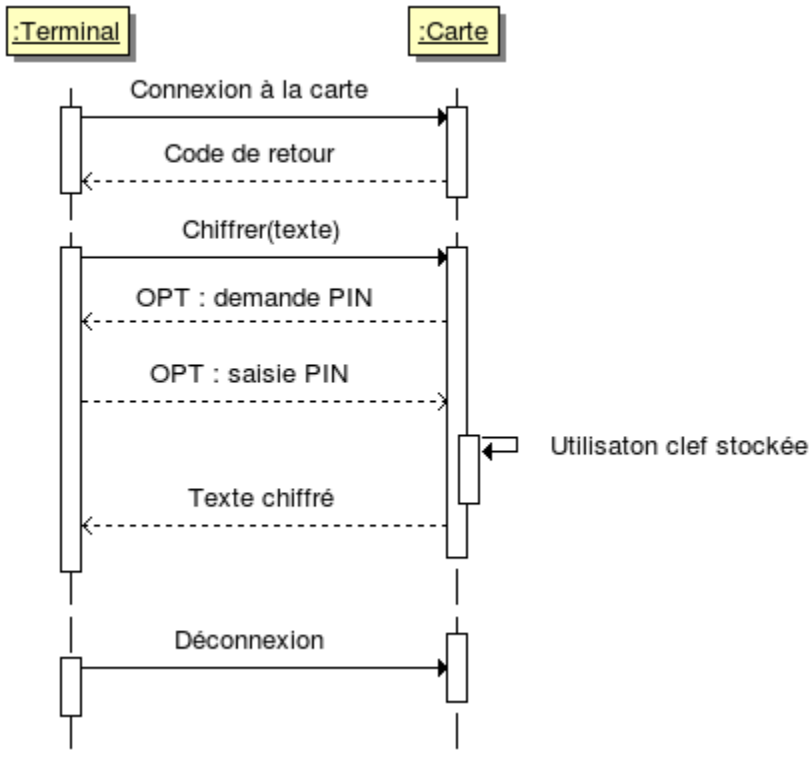
6.1. Génération d'un nombre pseudo-aléatoire

La génération d'un nombre pseudo-aléatoire est simple, puisqu'il s'agit de « sélectionner » l'applet via l'envoi d'un APDU. A la réception de celle-ci, une méthode est appelée (« process ») qui se chargera de générer et de renvoyer un nombre aléatoire.



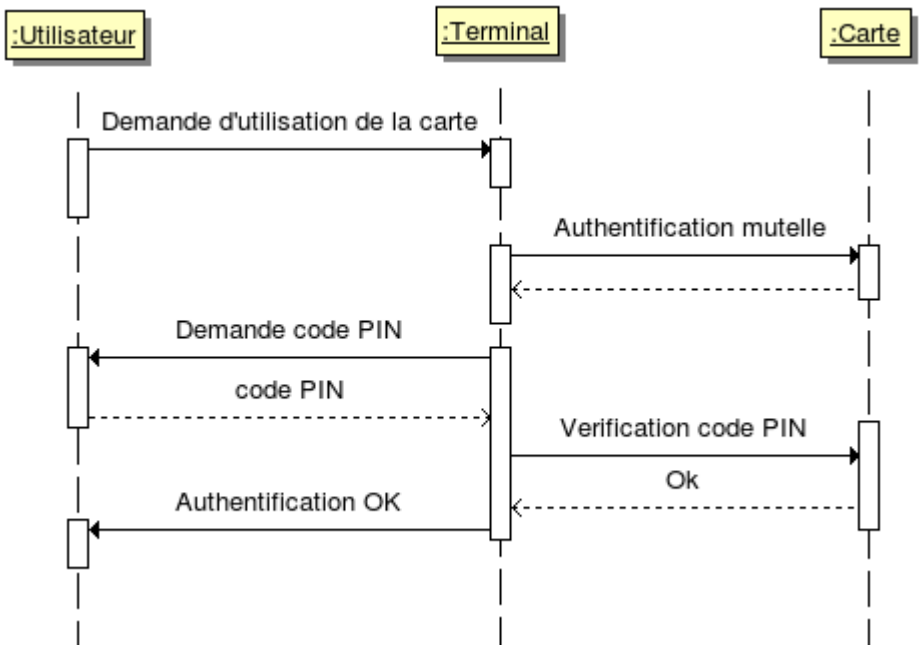
6.2. Chiffrement de données

Le schéma suivant représente le scénario durant lequel aucune erreur n'est produite. Sont mis en jeu le terminal (en l'occurence le lecteur de carte) et la carte à puce.
En cas de signature, le protocole sera sensiblement le même.



6.3. Authentification de l'utilisateur

Lorsque l'utilisateur insère la carte et qu'il cherche à s'y connecter, le scénario suivant s'applique, en supposant qu'aucune erreur ne survient.



7. Traçabilité

Exigences		F-FO-10	F-FO-20	F-FI-10	F-FI-20	F-FQ-10	F-FQ-20	F-FQ-30	F-FR-10
Composants	*Proxy*	X			X	X			
	SoftCard	X		X	X	X	X	X	X
	SmartCard	X	X	X		X	X	X	X