

ARCHITECTURE DU LOGICIEL

Version : 0.1

Date : 26/11/12

Rédigé par : Emmanuel Mocquet

Relu par :

Approuvé par :

MISES A JOUR

Version	Date	Modifications réalisées
0.1	26/11/12	Création

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 étudier les solutions cryptographiques pouvant permettre l'authentification ou la signature via carte à puce, puis à développer une ou plusieurs applications établissant la communication entre ce support et un PC ; ceci permettra le chiffrement et/ou l'authentification de données.

La seconde partie consiste à obfusquer tout contenu déposable sur le site internet « Facebook », tel que des images, videos... Pour ce faire, **TODO**

Nous ne décrivons ici que la partie architecturale.

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 « Canevas STB.pdf » ;
- De même, la seconde partie est disponible dans le document « ... »
- Et la STB associée dans « ... »

Le sujet propose différents 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

- *Abréviations*
- *Formalisme utilisé*
- *Légendes et conventions de représentation*

4. Architecture physique du matériel utilisé

Pour la première partie du projet :

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 sur cartes se fera via l'API fournie par un plugin d'Eclipse : JCDE. Leur gestion se fera grâce à l'outil GlobalPlatform, suivant lui-même le standard globalplatform.

Décrire les caractéristiques de la plate-forme cible

- *performances du calculateur*
- *périphériques et matériels spécifiques*
- *système d'exploitation (type et version)*
- *liste ou tableau des produits logiciels nécessaires (noms, origines, types et versions)*

5. Architecture statique du logiciel

5.1. Structure logique

PC -----| Interface PCSC |----- Lecteur de cartes ----- Carte

Identifier les principaux constituants du logiciel à développer et présenter la structure statique dans laquelle ils s'intègrent sous la forme d'une « carte de l'architecture » (schéma descriptif faisant apparaître les constituants et leurs interdépendances)

Pour chaque constituant identifié

5.2. Description du constituant « X »

- *Rôle*
- *Propriétés et attributs de caractérisation*
- *Services offerts (interfaces)*
- *Dépendances avec d'autres constituants (services utilisés, composants « sur étagère » utilisés)*
- *Langage de programmation*
- *Procédé de développement (techniques, méthodes et/ou outils)*
- *Taille et complexité.*

5.3. Justifications techniques

Fournir une rapide justification des choix technique originaux.

6. Fonctionnement dynamique

Pour les principaux scénarii identifiés dans la spécification technique de besoin.

- *Liste des composants mis en jeu*
- *Description du processus de mise en œuvre sous la forme d'une séquence d'appels aux services offerts par les différents composants et en faisant clairement apparaître*
 - *la logique et le flux des événements traités*
 - *les interactions du logiciel avec les acteurs*
 - *les interactions entre les composants au travers de leurs interfaces*

7. Traçabilité

Récapitulatif des liens de dépendance entre les constituants du logiciel et les exigences de la STB (par exemple sous forme de tableau).