

Outil d'installation d'applications
sur une grille de cartes à puce
de type Java à API respectant
GlobalPlatform



Plan

- I. Contexte
- II. Outils utilisés
- III. Description des besoins
- IV. Description des données
- V. Architecture de l'API
- VI. Gestion des exceptions et tests

I- Contexte

- La carte à puce
 - La technologie JavaCard
 - GlobalPlatform
-
- Un circuit électronique
 - Manipuler des informations en sécurité
 - Plusieurs types de cartes
 - Nous nous intéresserons aux cartes à microprocesseur et à contact

I- Contexte

- La carte à puce
- La technologie JavaCard
- GlobalPlatform

Applications

- Télécommunications
- Industrie bancaire et monétaire
- Santé
- Transport
- Authentification
-

I- Contexte

- La carte à puce
- La technologie JavaCard
- GlobalPlatform

JavaCard

- Faire fonctionner des applications écrites en Java sur une carte à puce
- Une plate-forme sécurisée pour carte à puce

I- Contexte

- La carte à puce
- La technologie JavaCard
- GlobalPlatform

Les avantages de la JavaCard

- La programmation orientée objets offerte par Java
- La possibilité d'utiliser les environnements de développement existants pour Java
- Plate-forme ouverte qui définit des APIs et un environnement d'exécution standardisé
- L'indépendance des applications par

I- Contexte

- La carte à puce
- La technologie JavaCard
- **GlobalPlatform**

GlobalPlatform

- Apporter un standard au monde des cartes
- Gérer les cartes de façon indépendante du matériel
- Spécifications flexibles pour des émetteurs de carte

I- Contexte

- La carte à puce
- La technologie JavaCard
- GlobalPlatform

Mécanismes de sécurité

Sécuriser les communications:

- S'assurer que les applications chargées sont officiellement signées
- Vérifier l'identité du porteur de carte;
- Par rapport aux échanges entre la carte et une entité extérieure:
 - Authentification mutuelle;

Outils

- Eclipse : plate-forme Java open source facilement étendue par des plug-ins
- JCOP : (JavaCard Open Platform)

Un plug-in qui offre :

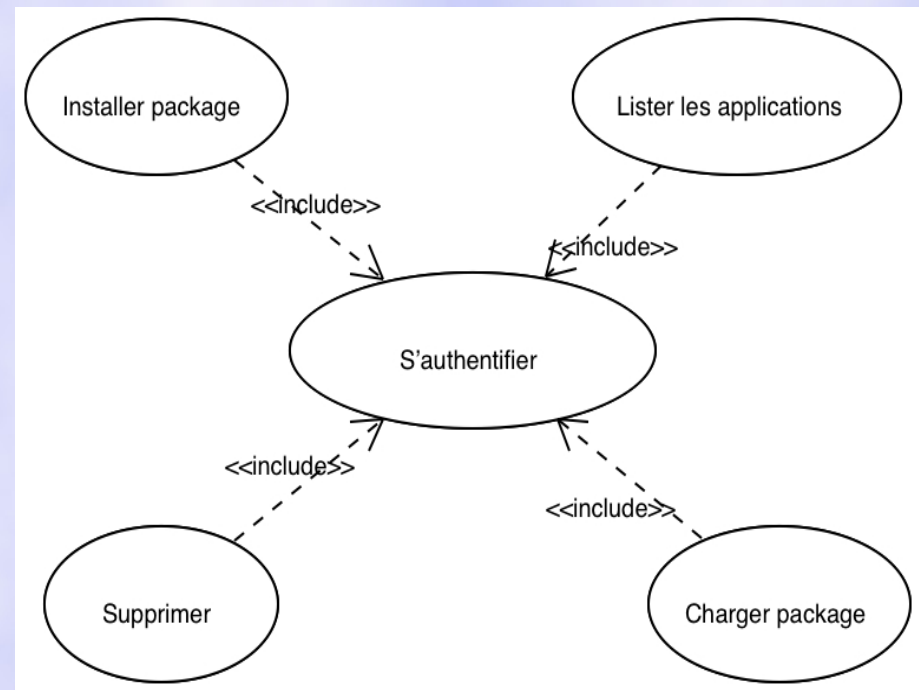
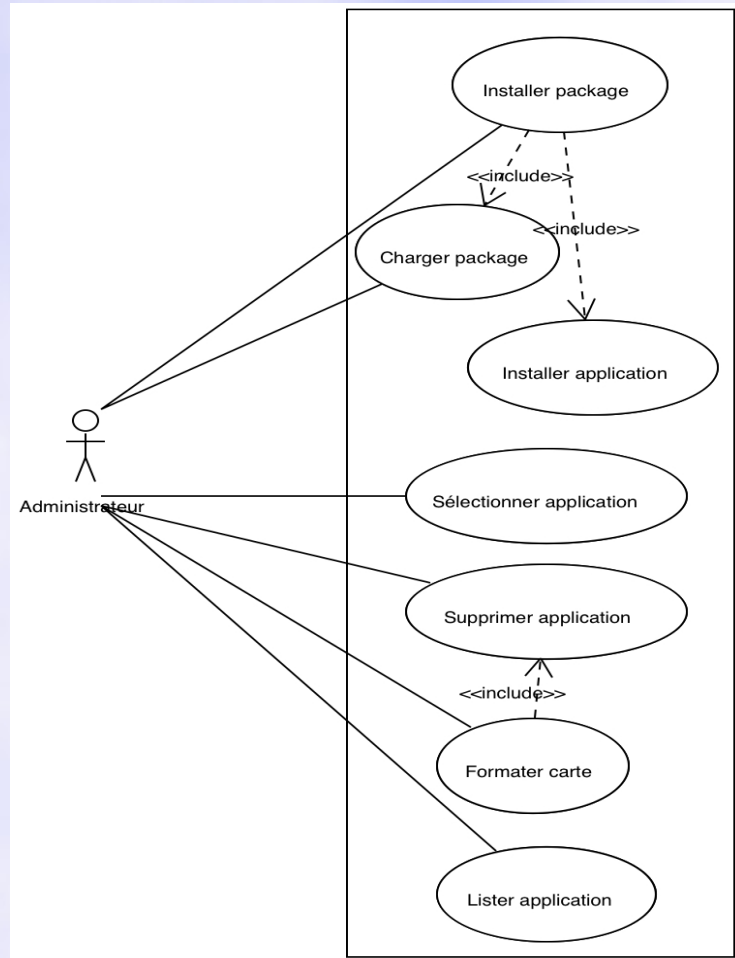
- Un convertisseur de byte code JavaCard
- Un simulateur de cartes JavaCard
- Le JCOP Shell
- Wiki en ligne : <http://pfa.fisoft1.com>
- CVS

III. Rappel des besoins - Objectif

- Notre objectif
 - Respecter les spécificité GlobalPlatform
 - Permettre la réalisation d'opérations élémentaires sur une carte.
- Les services offerts
 - Multiples et variés
 - Intialisation d'un canal de communication avec la carte
 - Installer, supprimer, sélectionner une application
 - Lister les applications disponibles sur la carte

Rappel des besoins – Services

- Diagramme des cas d'utilisation :



Rappel des besoins – « CU »

- Description des cas d'utilisation
 - Charger / Installer / Supprimer Package
 - Commandes installForLoad, installForInstall etc...
 - Sélectionner une application
 - Commande Sélectionner
 - S'authentifier
 - Algorithme de cryptage avec utilisation de clés.
- Scénarios d'utilisation

Description des données - APDU

- Standard APDU
 - Communication Carte/Monde extérieur
 - Format commande/réponse APDU

Couche application : Application Protocole Data Unit (APDU)
Couche transport : Transport Protocole Data Unit (TPDU)
Couche Physique

TAB. 5.1 Pile protocolaire

En-tête obligatoire				Corps optionnel		
CLA	INS	P1	P2	Lc	Champ données	Le

TAB. 5.2 – Format de la commande APDU

Champ optionnel	En-queue obligatoire	
Champ données	SW1	SW2

TAB. 5.3 – Format de la réponse APDU

Description des données - JCVM

- Technologie JavaCard : JCVM
 - Architecture JavaCard
 - Machine virtuelle JavaCard (JCVM)
 - Fichier CAP
 - Listing des composantes

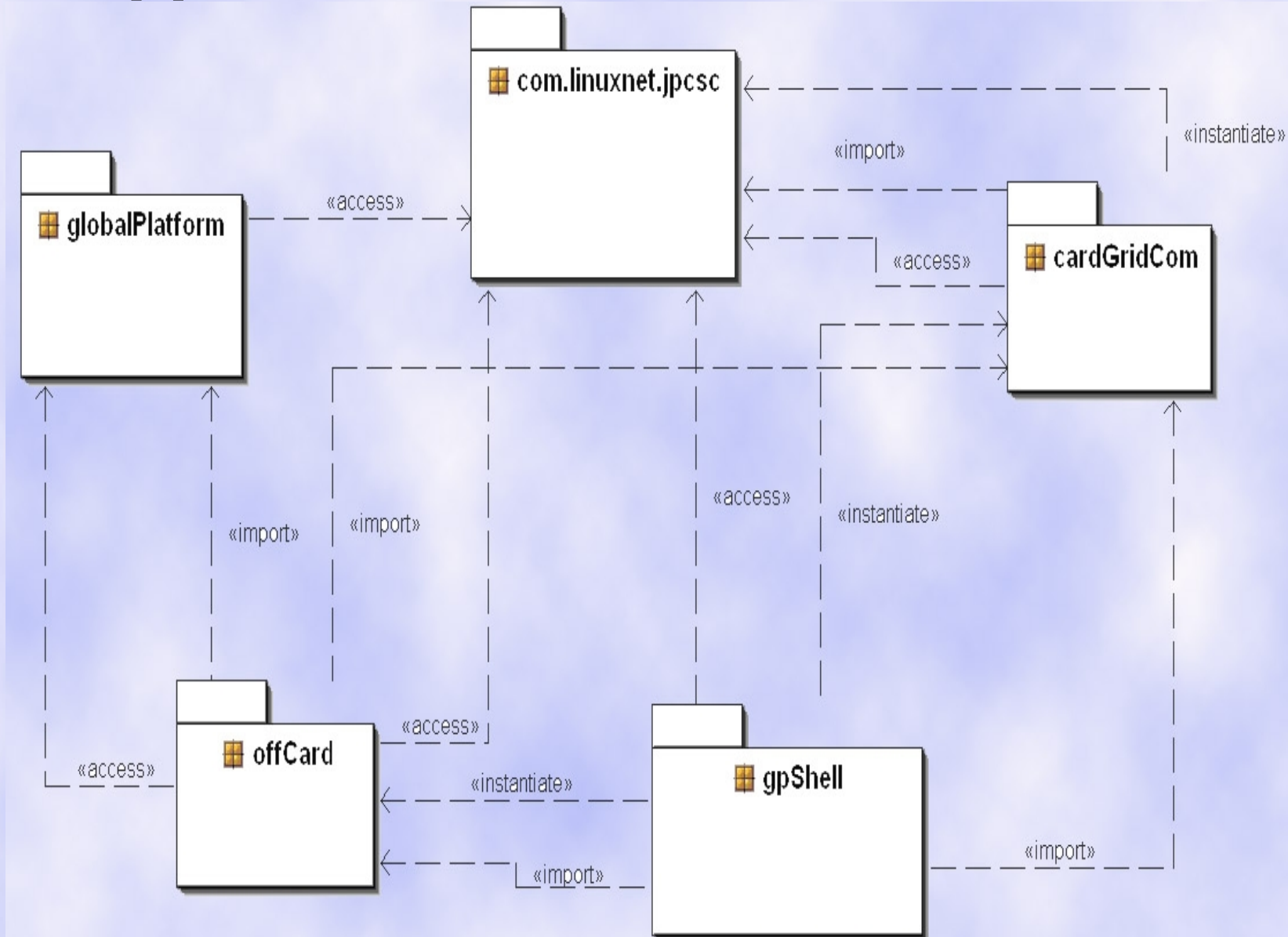
Applications
Java Card APIs : Présentes sur toutes les cartes
Java Card Virtual Machine
Interpréteur de bytecode
Hardware de la carte et son système natif

TAB. 5.4 – Architecture de la Java Card

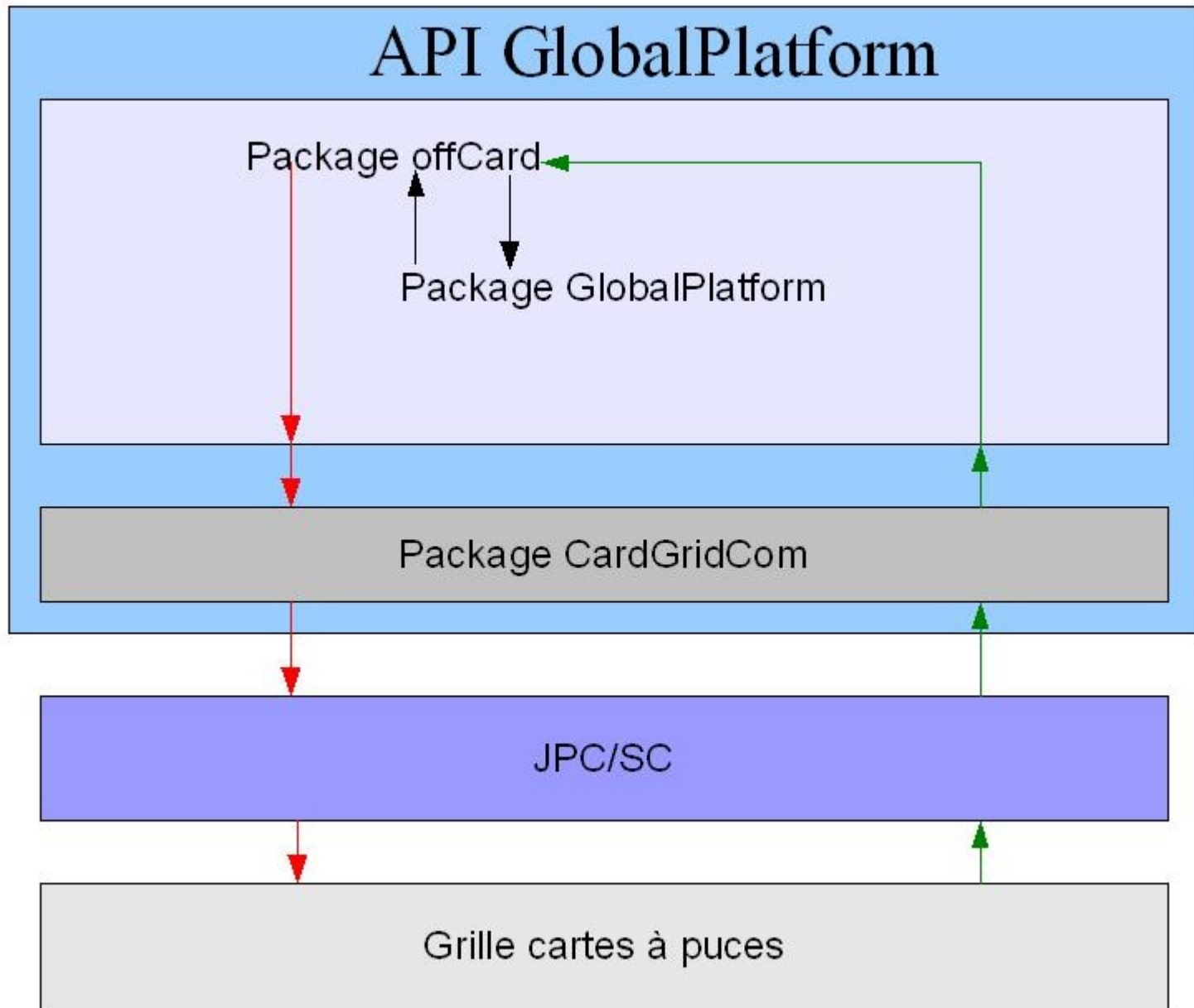
Étiquette	Taille	Données
-----------	--------	---------

TAB. 5.5 – Format d'un composant du package

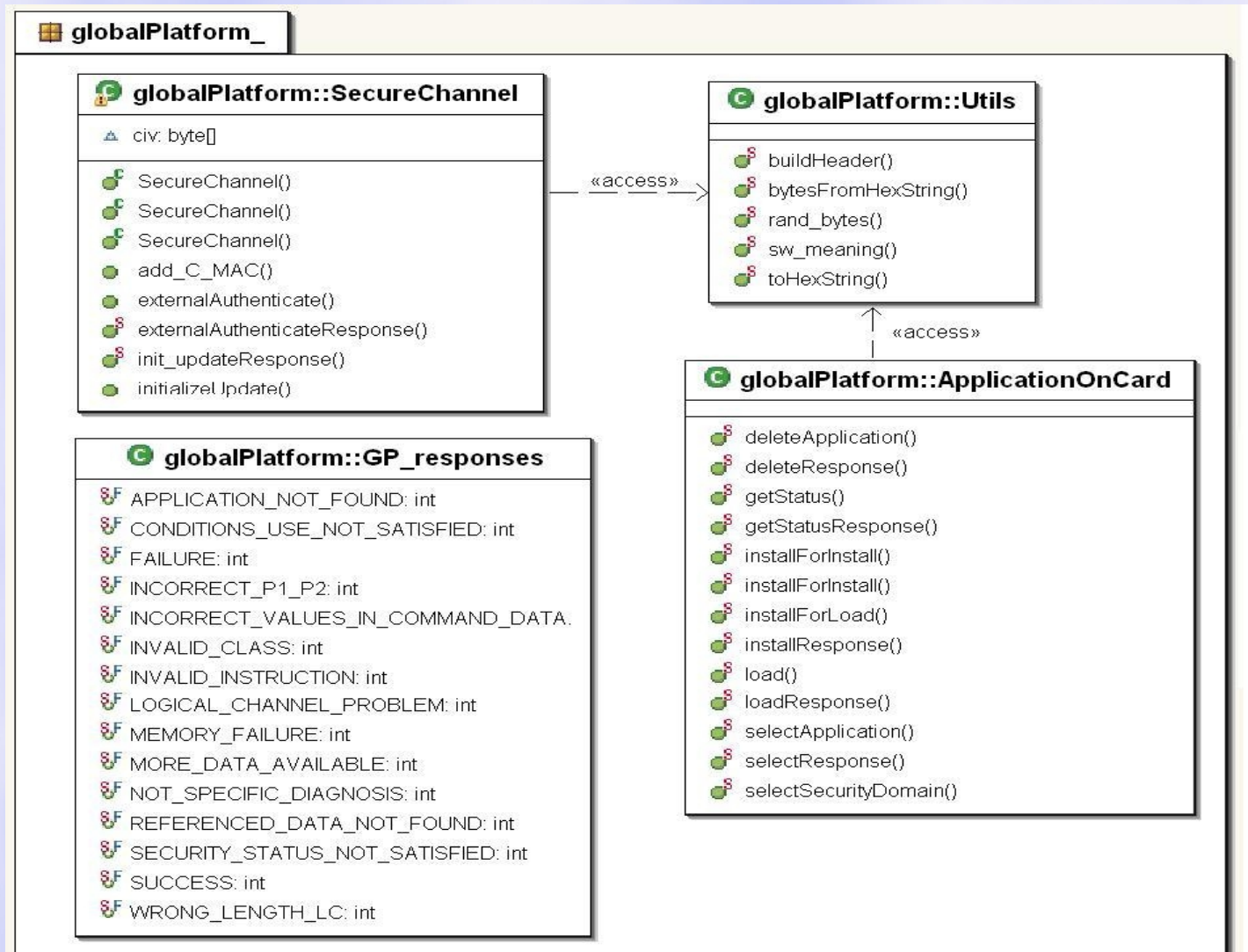
V- Architecture de l'application:
















Communication avec les cartes



V-1- Package globalPlatform



- Classe ApplicationOnCard:

C ApplicationOnCard	
	deleteApplication()
	deleteResponse()
	getStatus()
	getStatusResponse()
	installForInstall()
	installForInstall()
	installForLoad()
	installResponse()
	load()
	loadResponse()
	selectApplication()
	selectResponse()
	selectSecurityDomain()

- Construction des commandes APDU

=

En-tête obligatoire			
CLA	INS	P1	P2

Entête (méthode
Utils.builHeader)

+

Corps optionnel		
Lc	Champ données	Le

Corps (optionnel)

- Analyse des réponses APDU

ApplicationOnCard

Détails des méthodes

1. Envoie d'APDUs

- `deleteApplication(byte[], byte)`
- `getStatus(byte, byte, byte[])`
- `installForLoad(byte[], byte[], byte[], byte[], byte[])`
- `installForInstall(byte[], byte[], byte[], byte, byte[])`
- `installForInstall(byte[], byte[], byte[], byte, byte[], byte[])`
- `selectApplication(byte[], byte)`
- `selectSecurityDomain()`

2. Analyse de réponses

Ex: `selectResponse(byte[])`

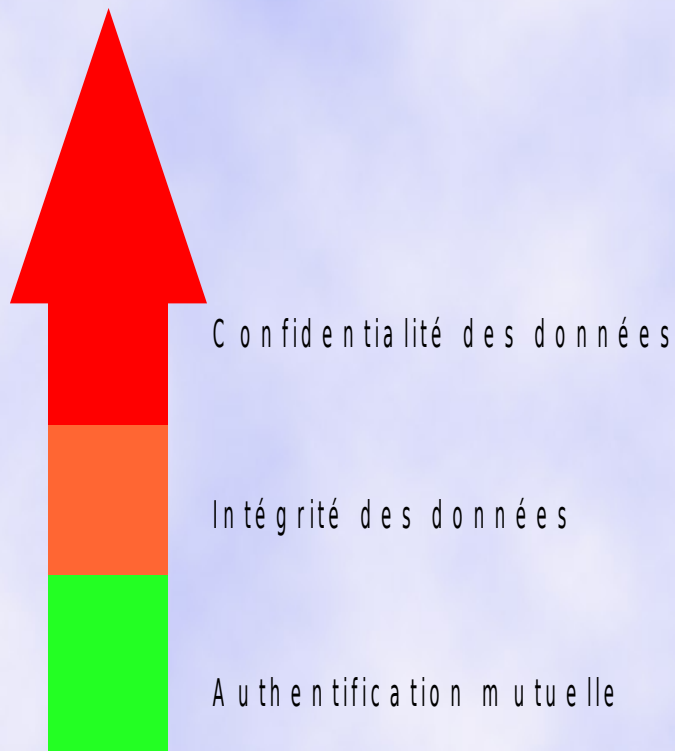
Un Exemple...

```
public static byte[] deleteApplication(byte[] aid, byte p2){
    int i=0;
    byte[]header = Utils.buildHeader((byte)0x80/* CLA */,
                                     (byte)0xE4 /* INS */,
                                     (byte)0x00 /* P1 */,
                                     p2/* P2 */,
    (byte) (aid.length+2)           /*lc: Data length= length(4F+ AID.length + AID)*/
    );

    /*Begin Data*/
    int len = aid.length + header.length + 3;
    byte[] cmd_buf = new byte[len];
    System.arraycopy(header, 0, cmd_buf, 0, header.length) ; /*
    Header */
    i=header.length;
    cmd_buf[i++]= (byte)0x4F ; /*Tag for an AID*/
    cmd_buf[i++]= (byte)aid.length ;
    System.arraycopy(aid, 0, cmd_buf, i, aid.length) ; /* Data */
    i=i + aid.length ;
    cmd_buf[i]= 0x00; /* Le */
    return cmd_buf ;
}
```

Canal sécurisé

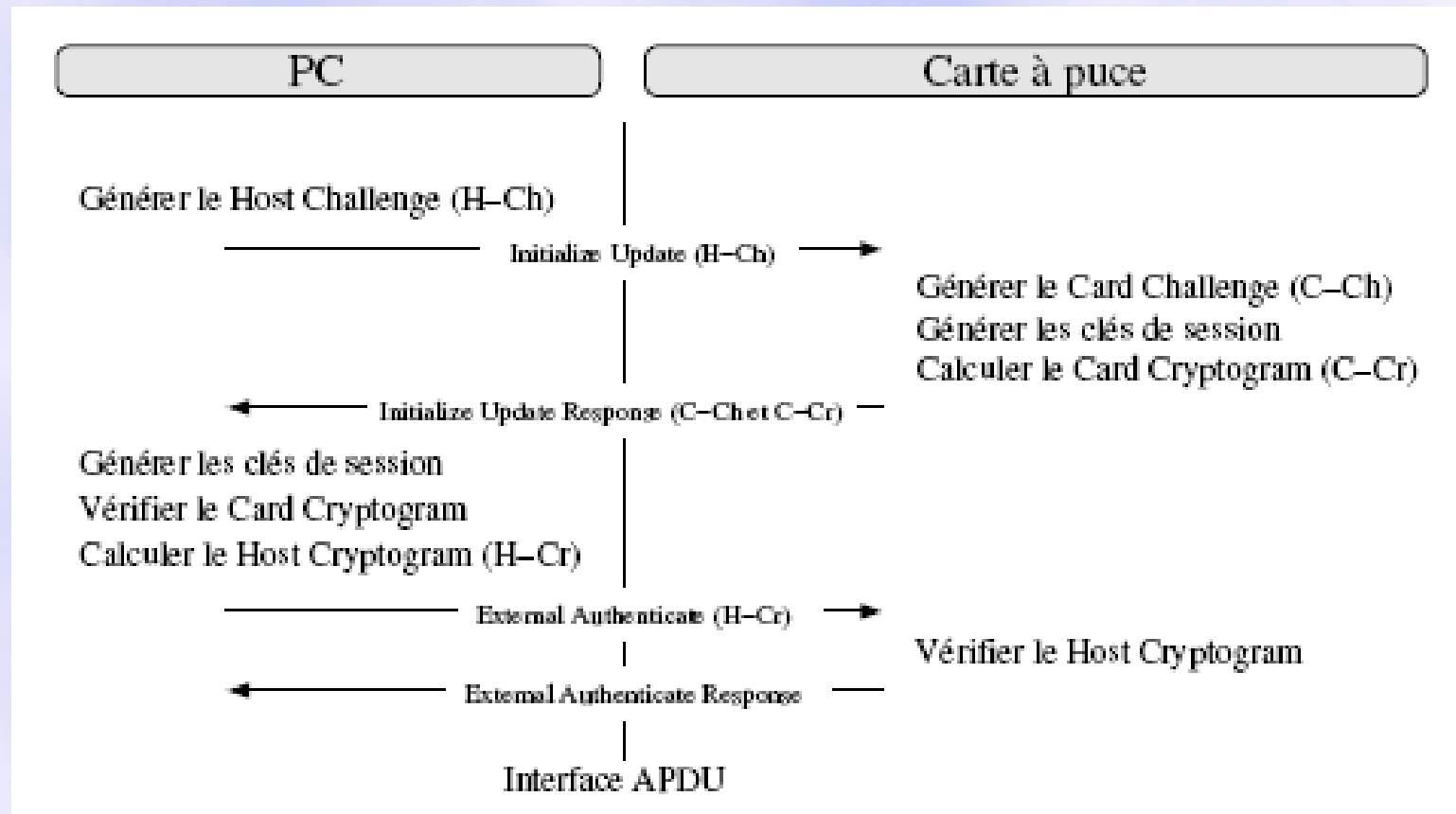
3 niveaux de
sécurité :



- Clé statique ENC (S-ENC)
- Clé statique MAC (S-MAC)
- Clé de cryptage (DEK)

Canal sécurisé

Authentification mutuelle



Canal sécurisé

Commande: initialize-update

A P D U i n i t - u p d a t e

	P 1 I n d e x d e s c l é s	P 2 V e r s i o n d e s c l é s		D a t a H o s t C h a l l e n g e	
--	---	---	--	---	--

R é p o n s e i n i t - u p d a t e

	C a r d c r y p t o g r a m	C a r d c h a l l e n g e	
--	--------------------------------------	-----------------------------	--

Canal sécurisé

Commande: external-authenticate

A P D U external-authenticate

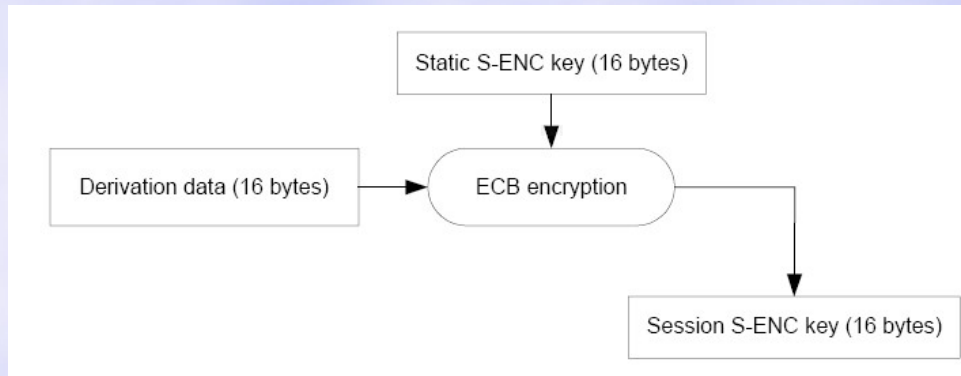
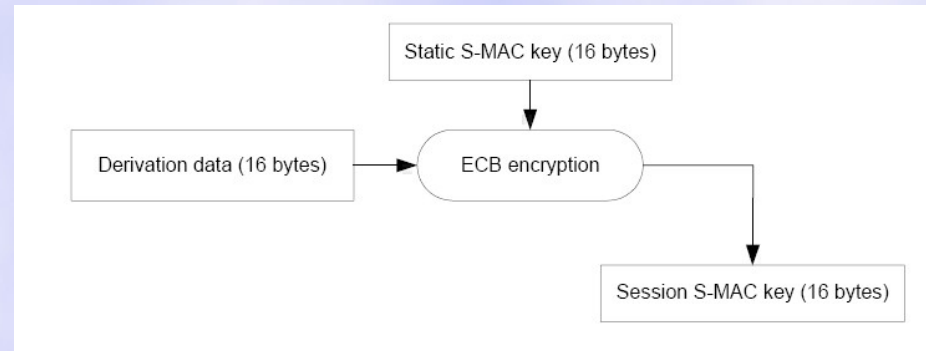
	P 1 S e c u r i t y l e v e l		D a t a <u>H o s t c r y p t o g r a m e t M A C</u>	
--	---	--	--	--

Canal sécurisé

Commande: external-authenticate

Clés de session

:

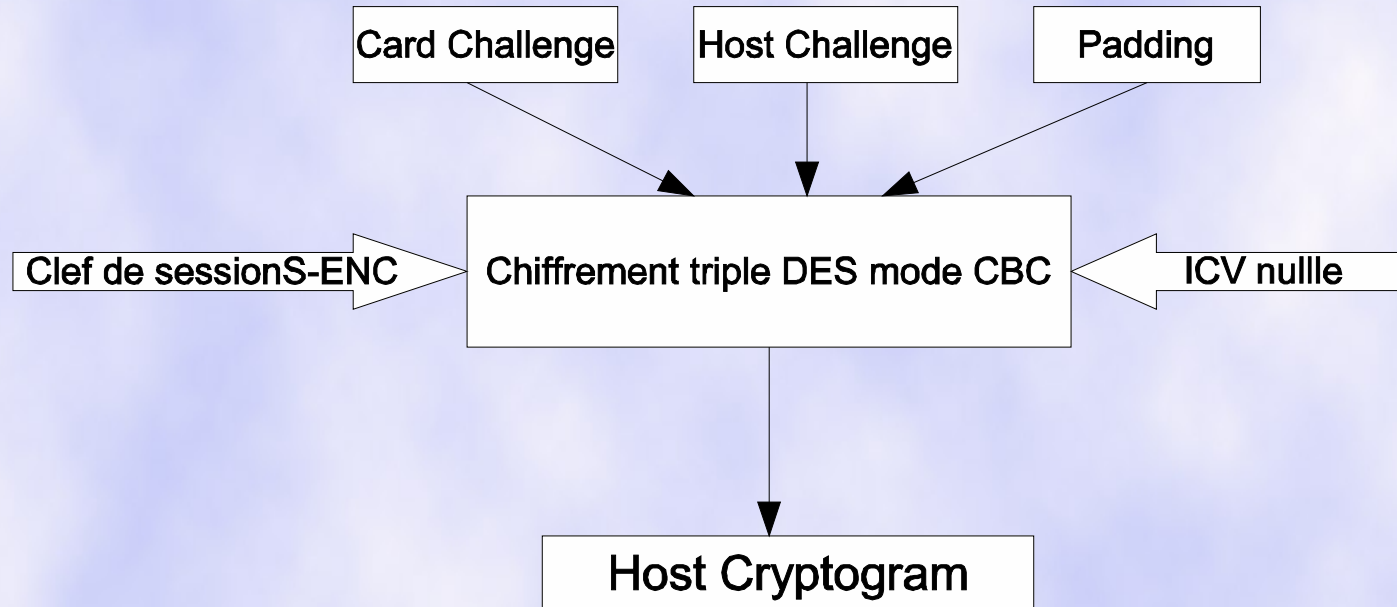


Canal sécurisé

Commande: external-authenticate

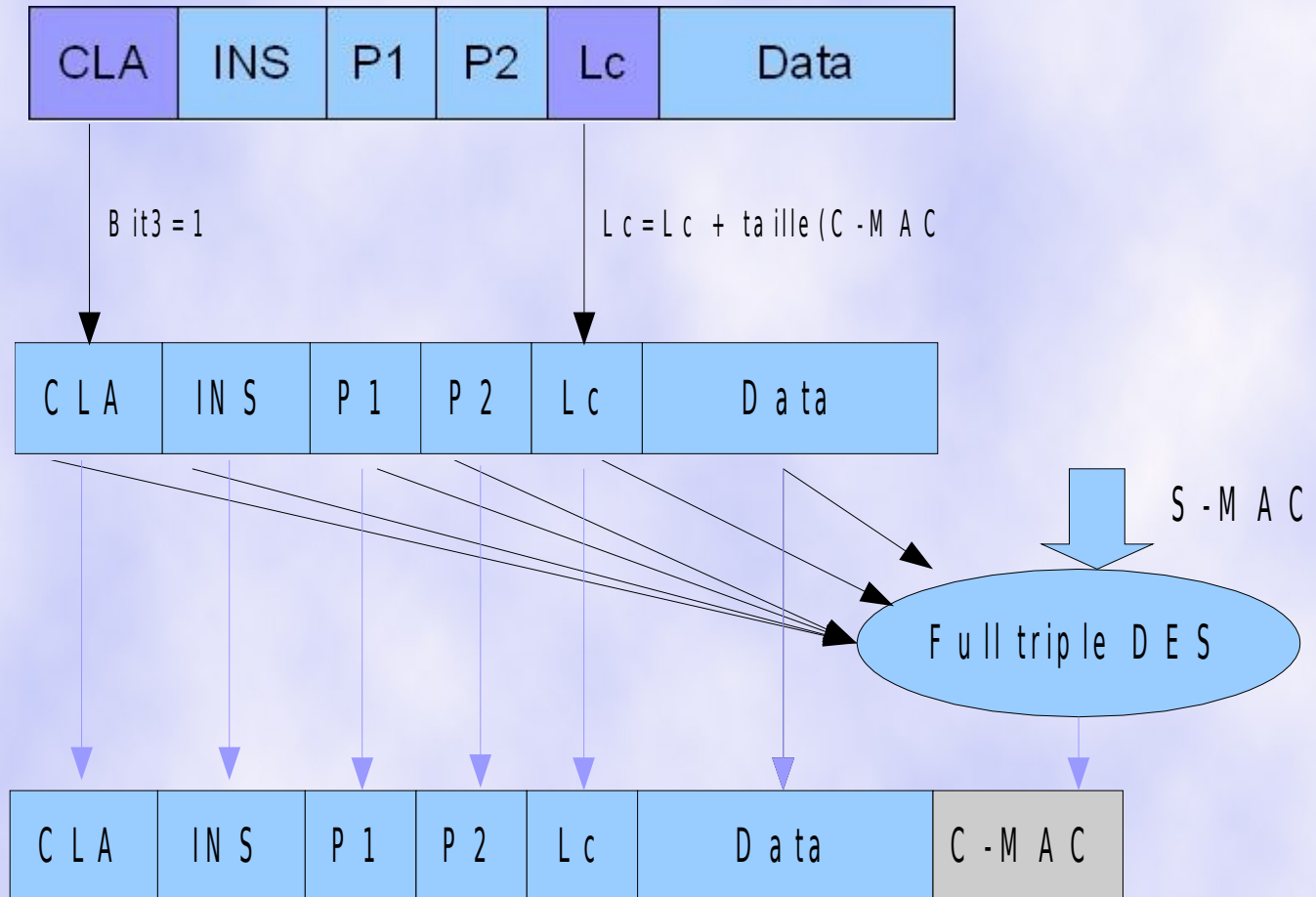
Host cryptogram

:



Canal sécurisé

Commande: external-authenticate



Canal sécurisé

Commande: external-authenticate

A P D U external-authenticate

	P 1 S e c u r i t y l e v e l		D a t a <u>H o s t c r y p t o g r a m e t M A C</u>	
--	---	--	--	--

R é p o n s e external-authenticate

A u t h e n t i f i c a t i o n v a l i d é e

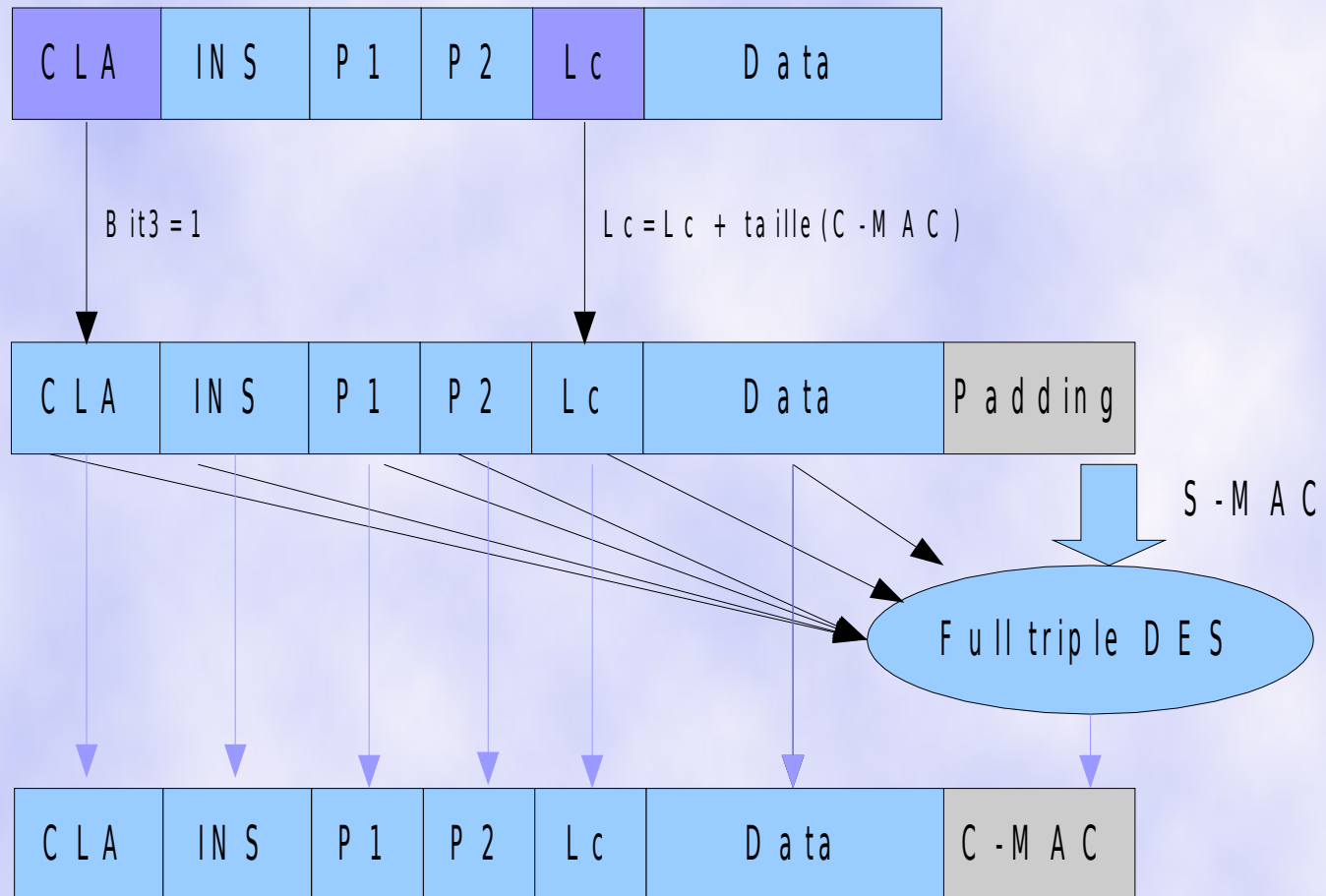
S W 1 9 0	S W 2 0 0
---------------------	---------------------

A u t h e n t i f i c a t i o n r e f u s é e

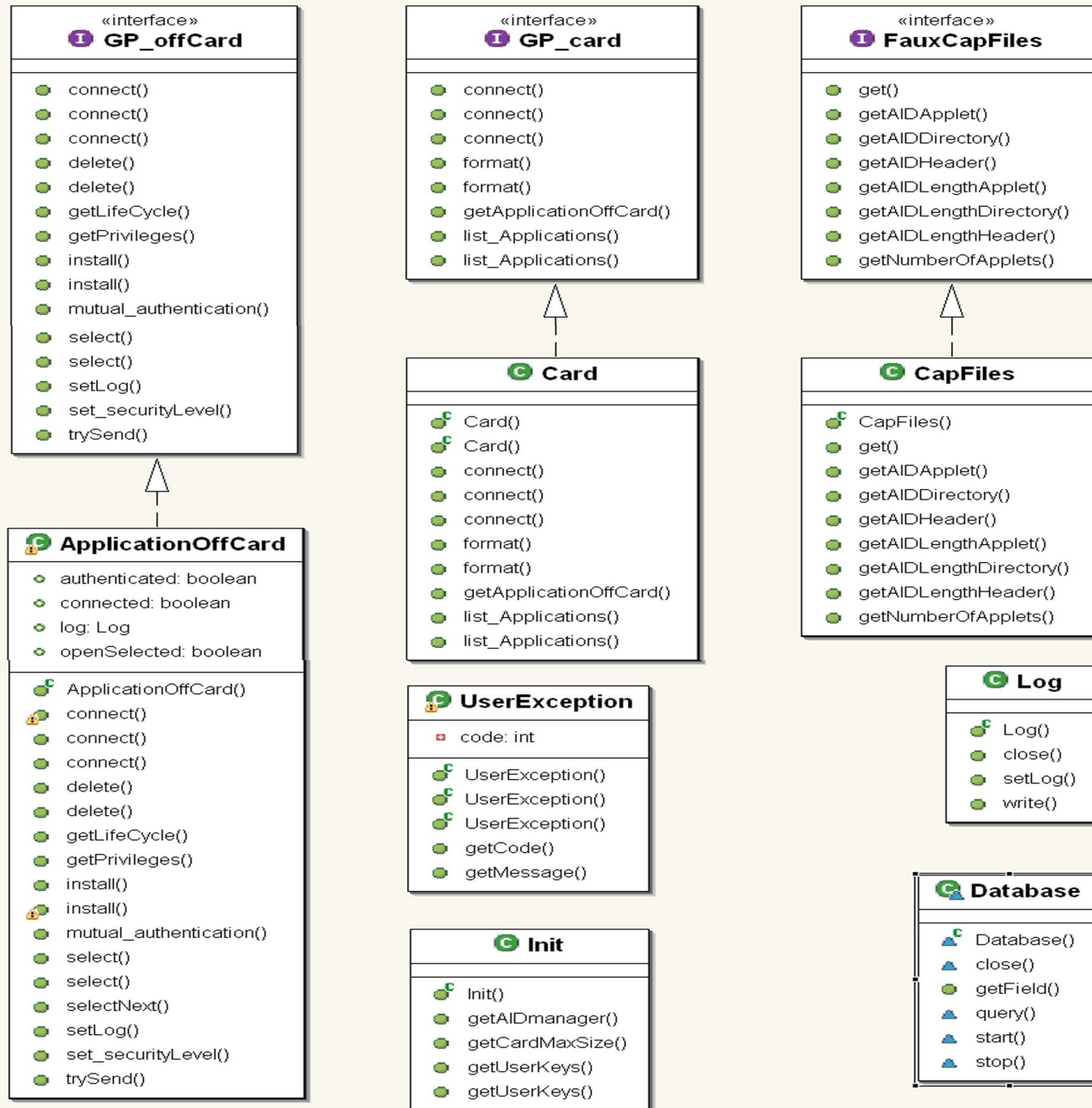
S W 1 6 3	S W 2 0 0
---------------------	---------------------

Canal sécurisé

Intégrité des données: L'ajout d'un C-MAC pour toutes les commandes



Package OffCard



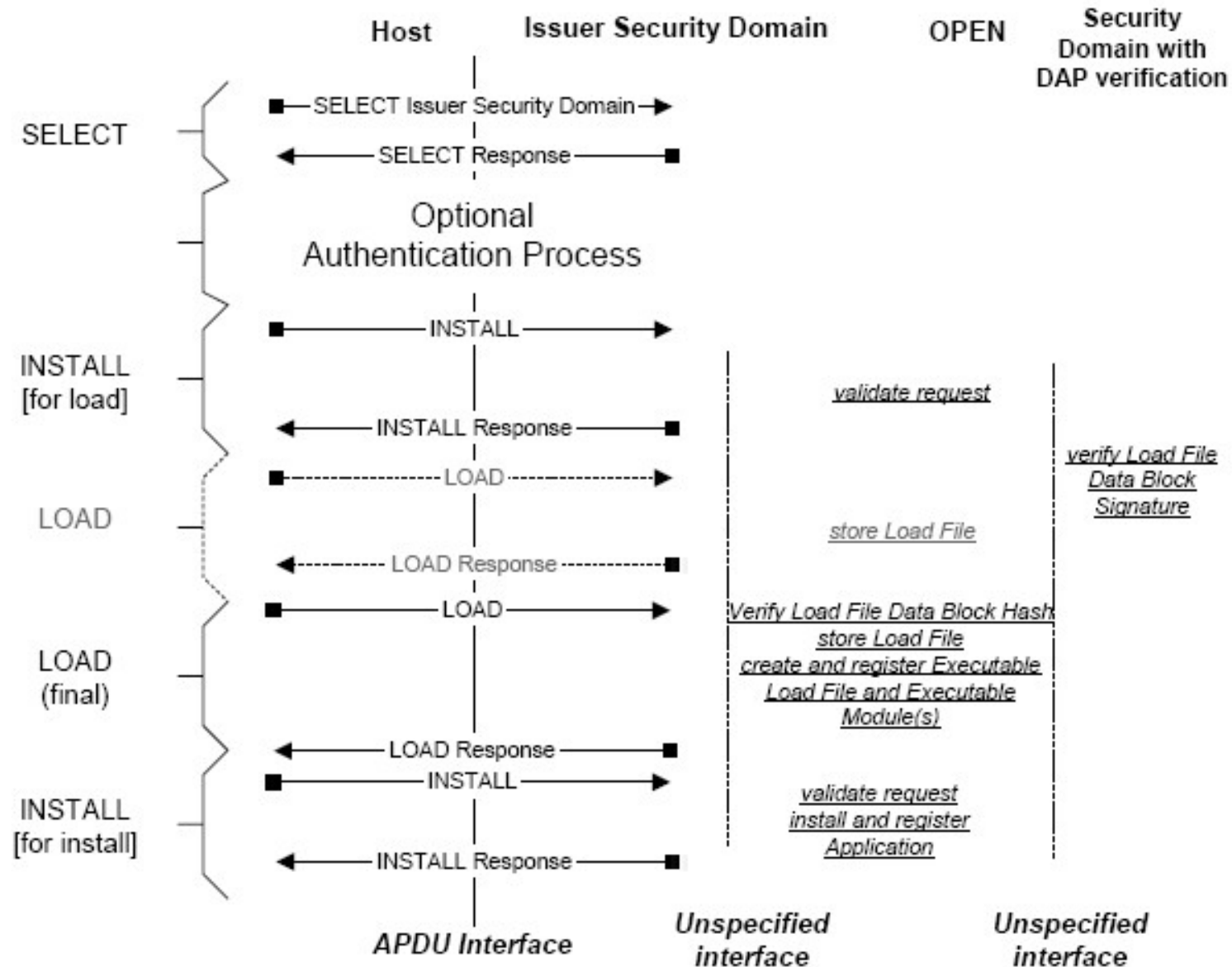
Classe ApplicationOffCard

- Initialisation de la communication avec la carte.
- Chargement et installation d'un package.
- Suppresion d'un package.
- Sélection d'une application.
- Privilèges et états d'une application sur la carte

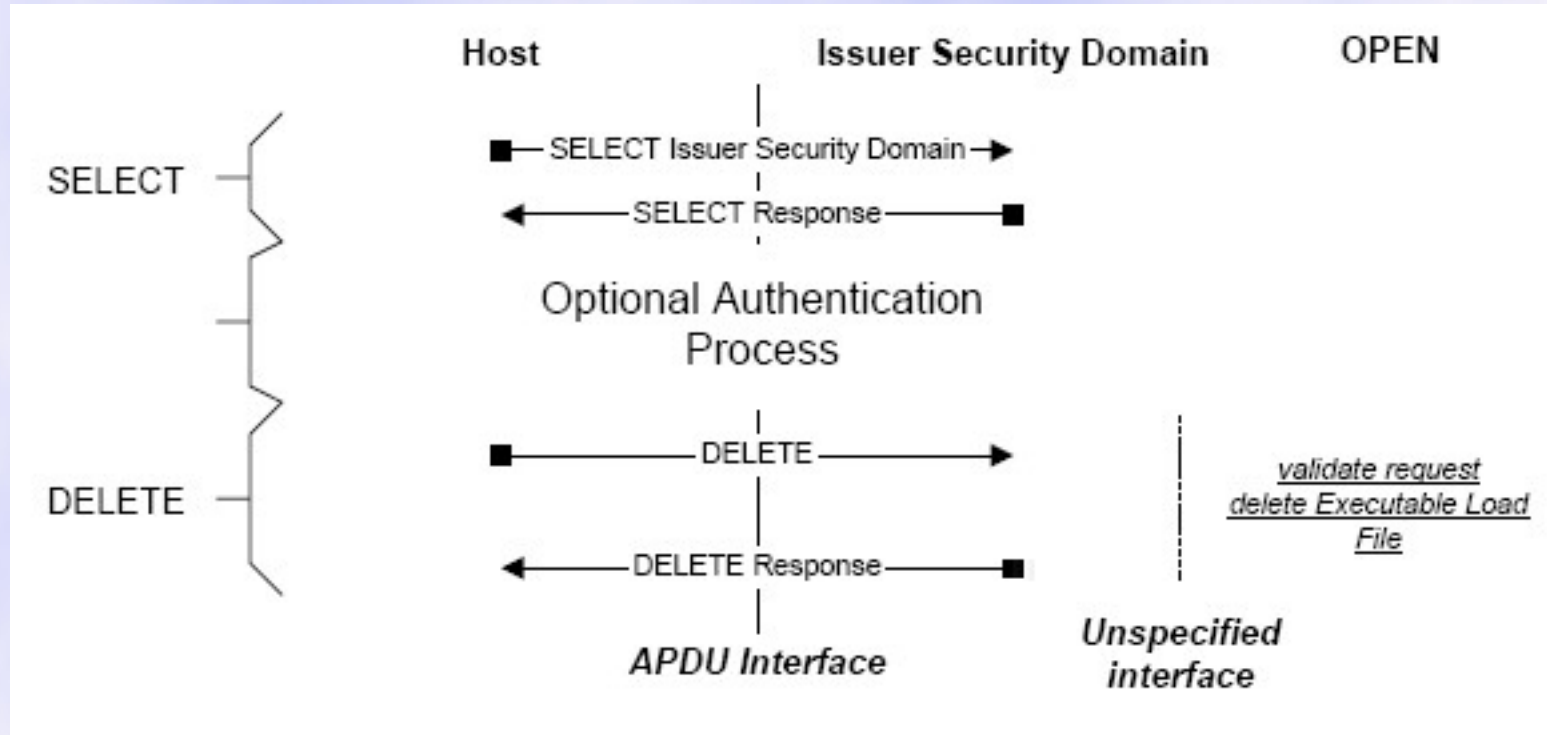
L'initialisation de la communication avec la carte se fait en trois étapes :

- La connexion au lecteur.
- La sélection du card manager.
- L'authentification mutuelle.

Chargement et installation d'un package



Suppression d'un package



Classe Card

- Listage des applications de la carte.

A12DF3615DC	SELECTABLE
C412EF3615D	SELECTABLE

B1E93C41F61	LOADED
F1D85C36C62	LOADED
4FE92C4EF47	LOADED

- Formtage de la carte.

VI- Gestion des exceptions

- UserException
 - Exceptions PC/SC
 - Exceptions GlobalPlatform
-
- Cette classe hérite de la classe Exception
 - Appartient au package offCard
 - Gérer les exception levée par l'API
 - 3 constructeurs :
 - Un par défaut
 - Un pour le message d'erreur
 - Un pour le code d'erreur

VI- Gestion des exceptions

- `UserException`
- **Exceptions PC/SC**
- `Exceptions GlobalPlatform`

- Exceptions lors de la connexion ou la communication
- Encapsulation de méthodes de la couche JPCSC dans blocs *try* et *catch*
- Une exception levée est capturée et envoyée sous forme `UserException`

VI- Gestion des exceptions

- UserException
 - Exceptions PC/SC
 - Exceptions GlobalPlatform
-
- Codes erreur définis par ce package est codés sur des champs SW1 et SW2
 - Module GP_responses définissant les constantes des erreurs possibles

Tests

- Tests Unitaires : interface Junit
- Tests avec JCOP
- Tests avec cartes
- Compatibilité et intégration avec la deuxième partie du projet

JCOP

- Permet de tester les fonctions développées sur des cartes
- Permet aussi de simuler une carte et faire ainsi des tests

Tests avec cartes

- Des tests avec des cartes mais sans JCOP effectués
- Communication directe avec la carte
- Un jeu de programmes avec plusieurs scénarios et différentes fonctions

Exemple de scénarios

- Installation puis sélection puis suppression et enfin réinitialisation d'un même package
- Charger deux fois successives le même package sur la même carte, une exception levée
- Supprimer un package non existant, une exception levée

Compatibilités et intégrations

- Problèmes de prototypes entre méthodes développées et appels
- Remédier en fournissant des interfaces pour les classes *ApplicationOffCard* et *Card*

Extension :GPShell

Install

➤ *Run install ?*

installInAll

Delete

install -help to see this help.

Format

install cardName packagePath [userKeys]

Or

formatAll

install cardName packagePath [keyVersion keyIndexNumber]

ls

Conclusion