Déploiement d'une Javacard à l'IMAG

Jean-François Desnos, Gérard Forestier

Déploiement d'une Java Card à l'Université de Grenoble - UFR IMAG

Jean-François Desnos
DSI Grenoble Universités
Université Joseph Fourier, BP 53, 38041 Grenoble cedex 9

Gérard Forestier
UFR Informatique et Mathématiques Appliquées (IMAG)
Université Joseph Fourier, BP 53, 38041 Grenoble cedex 9

Résumé

L'IMAG (Informatique et Mathématiques Appliquées de Grenoble), UFR de l'université Joseph Fourier, a déployé à la rentrée universitaire 2009-2010 un projet de carte multiservice distribuée aux étudiants, enseignants et personnels de l'UFR. Cette carte est devenue la carte d'identification visuelle et électronique de l'établissement, en remplacement des anciens supports. Il s'agit d'une carte à microprocesseur embarquant une plateforme OS Java, rendant possible le téléchargement dynamique d'applications après la phase de personnalisation de la carte. Le visuel de la carte comporte une photo du porteur ainsi que des informations d'identification. Les services accessibles par cette carte sont dans un premier temps :

- contrôle d'accès physique aux locaux du bâtiment principal de l'IMAG ;

- contrôle d'accès logique au réseau informatique de l'IMAG ;

- accès aux services de documentation.

D'autres services compléteront par la suite le catalogue des usages de cette carte, mais d'ores et déjà il est prévu, durant cette prochaine année universitaire, d'installer à titre expérimental une application de transport (système Calypso)ainsi que Moneo.

Mots clefs

Carte multiservice, Java Card, Calypso, Moneo, JVM, java, Sun Ray, PKI, ISO, Kerberos, X509

1 Historique

Soutenu financièrement par la région Rhône-Alpes, le projet de carte multiservice de Grenoble Universités est entré dans sa phase opérationnelle en septembre 2006. Il est commun aux universités de Grenoble et au CROUS de Grenoble.
À terme, le projet vise à équiper d'une carte multiservice les 80 000 étudiants et personnels des universités de Grenoble et de Savoie.
Cependant, la date du déploiement dépendra d'accords à trouver avec des partenaires extérieurs pour diminuer le coût de la carte pour les établissements universitaires, faute de quoi le projet risque d'être abandonné. Aujourd'hui, le projet en est donc toujours au stade « pilote », et l'on peut l'expliquer par un historique en trois phases :

Site pilote PHELMA (Grenoble-INP)

PHELMA est une école de Grenoble-INP (PHysique, ELectronique, MAtériaux). Depuis la rentrée universitaire de septembre 2006, il y a été distribué à ses 1 500 étudiants et personnels une carte Moneo BMS[1]2 permettant l'identification (carte d'étudiant, carte professionnelle, carte visiteur), l'accès aux locaux et aux bibliothèques, et le paiement de la restauration CROUS. La BMS2 comporte deux interfaces : avec et sans contact.

À partir de 2008, de nouvelles cartes ont été distribuées à tous les étudiants et personnels de PHELMA et remplacent les précédentes : il s'agit de nouvelles BMS2 comportant l'application de transport normalisée Calypso.

[1]BMS : La société BMS – Billettique Monétique Services – est chargée de la conception, du développement et de l'exploitation du porte-monnaie électronique français (« MONEO »)

Recherche de partenaires

En 2007, Grenoble Universités estime que le coût de la carte Moneo BMS2, environ 7 €, rend impossible le déploiement aux 80 000 étudiants et personnels des universités de Grenoble et de Savoie. Grenoble Universités suspend donc le projet, dans l'attente de partenaires ou sponsors participant financièrement à l'achat des cartes, pour laisser de manière pérenne à l'université un coût participatif de 1 à 2 €.

En 2008, un groupe de travail comportant des représentants de la Région Rhône-Alpes, du Conseil Général de l'Isère, de BMS Moneo, de la communauté d'agglomération de Grenoble, du CROUS de Grenoble et de Grenoble Universités suggère pour base de réflexion la répartition de coût de carte suivante : 56% pour les universités, 23% pour le CROUS, 21% pour les transports en commun.

Le service global serait amélioré par l'utilisation d'un support unique pour l'ensemble des services, et l'économie sur l'achat de cartes multiples calculée sur 5 ans serait proche de 500 000 € (cas d'une BMS2 à 7 € pièce, ce qui est notre cas).

Mais le projet de carte commune CROUS-Université-Transports publics bute sur la nécessité d'utiliser une carte émise par une banque, Moneo étant pour le CROUS un service *sine qua non*.

En septembre 2008, le CROUS décide alors de déployer pour son propre compte des cartes BMS1 (à contact, moins chères) avec pour seul service la restauration CROUS.

Site pilote IMAG de l'université Joseph Fourier

Afin de pouvoir mettre en place une carte multiservice réellement évolutive et à un prix supportable pour les universités, il est initié en 2009 un nouveau projet pilote, proposé par l'UFR IMAG, avec la technologie *Java Card*. Spécialisée dans l'enseignement et la recherche en informatique et en mathématiques appliquées, l'UFR IMAG se lance en mars 2009.

Connectés aux bases de données de l'université (Apogée pour les étudiants et Harpège pour les personnels), les trois postes de gestion des cartes installés à l'UFR IMAG ont édité depuis le mois de septembre dernier près de 2000 Java Card. Ces cartes d'étudiant ou professionnelles permettent désormais un contrôle d'accès physique au bâtiment de l'UFR, un contrôle d'accès logique aux postes informatiques et aux réseaux de l'UFR (authentification *Sun Ray*[2]) et un accès aux services de documentation.

Parallèlement, les discussions sont relancées avec BMS et les transporteurs, pour ajouter les logiciels (« applets ») Moneo et Calypso (transports publics) sur les cartes délivrées cette année.

L'accord de BMS pour une expérimentation Moneo sur la Java Card, et l'accord d'importants industriels pour une expérimentation Calypso, étant d'ores et déjà obtenus, ces « pilotes » Moneo et Calypso seront testés en usage réel sur environ 500 cartes au premier semestre 2010.

2 Objectif du projet pilote UFR IMAG

Depuis 2007, l'UFR avait comme objectif de mettre en place une carte unique pour des fonctions de contrôle d'accès physique pour l'accès aux bâtiments et d'accès logique pour accéder au Système d'Information, avec des évolutions possibles vers une authentification forte X509[3].
La carte BMS2 ne permet pas de mettre en place ces fonctionnalités : il n'est pas possible pour l'université de charger de nouvelles applications, car elle comporte seulement une zone mémoire (600 octets) « non bancaire » disponible. Après un appel d'offre début 2009, nous avons alors retenu une Java Card comportant les deux interfaces avec et sans contact. Nous avions besoin de l'interface contact en raison de l'existence de cent cinquante Sun Ray comportant un lecteur à contact.
La technologie Java Card présente une sécurité importante, et la possibilité (grâce à Java) de la mise en œuvre d'applications de façon standardisée à travers des applets. Pour nous, l'esprit d'un projet pilote, de plus financé par la région Rhône Alpes, est de proposer une solution ambitieuse et pérenne pouvant servir d'alternative aux solutions déployées actuellement, qui présentent des limitations fortes sur l'aspect cohabitation d'applications diverses sur la même carte.

[2]Sun Ray: solution client léger deSUN :http://*www.**sun**.com/software/**sunray**/index.jsp*
[3]X509 : norme de cryptographie de l'Union Internationale des télécommunications pour les infrastructures à clés publiques (PKI).

3 Présentation de la Java Card

Une Java Card est une carte à microprocesseur (ou puce) qui est capable d'exécuter des programmes (applets) écrits dans le langage Java Card (sous ensemble du langage Java). Une machine virtuelle Java est embarquée sur la carte. Une application carte à puce (Java Card compris) est toujours constituée d'une partie cliente s'exécutant sur l'ordinateur hôte et une partie serveur s'exécutant sur la carte avec des échanges au format APDU[4]. Dans ce cas là, le serveur est en Java.
Un avantage majeur de la Java Card est d'être indépendante de la plateforme de développement et de permettre de décharger ou de mettre à jour dynamiquement les applications sur la carte, dans un un environnement standardisé. Cela permet la mise à jour des applets.
Une application pourra donc être développée puis installée sur n'importe quelle plate-forme. Par ailleurs, Java est un langage objet de haut niveau exécutable sur n'importe quel système d'exploitation.
Contrairement aux autres cartes où la programmation de la carte se fait avec des langages bas niveau « physique », la Java Card permet une programmation « logique » de la carte.
Les opérations de gestion de l'application ne sont pas définies par Java, mais par la spécification Global Platform[5][2] qui définit une architecture pour gérer et installer de manière sécurisée des programmes sur des cartes multi-applications (Java Card et autres technologies).

Historique :

- premières applications en 1996 avec apparition V1.0 de la spécification Java Card ;
- 1997 : version 2.0 de la spécification ;
- 1999 : version 2.1 des spécifications (spécification du format de fichier sur la carte, définition explicite de la machine virtuelle Java Card) ;
- 2002 : version 2.2 des spécifications (possibilité de communiquer avec plusieurs applets, intégration RMI[6]) ;
- Avril 2008 : version 3.0 des spécifications (encore très peu utilisée).

4 La technologie Java Card [3]

Glossaire

Nous commençons par expliquer les sigles les plus courants :

- **applet (contexte Java card)** : une application écrite dans un sous-ensemble du langage Java qui est chargée et exécutée sur une carte Java. Chaque applet embarquée sur la carte est isolée des autres par un pare-feu et dispose de son propre contexte ;
- **AID** : les applets sont identifiées par un AID[7]: id du fournisseur et les suivants identifient l'application ;
- **APDU** : les échanges entre carte et terminal se font au moyen d'APDUs (suite d'octets respectant le format défini dans le standard ISO7816[8]-4) ;

Command APDU:

Mandatory header				Optional body		
CLA	INS	P1	P2	Lc	Data	Le

- **CLA**: 1 byte to identify the application
- **INS**: 1 byte for the instruction code
- **P1-P2**: Instruction parameters
- **Lc**: Length of the data field
- **Le**: Maximum length of the data field of the response

Response APDU:

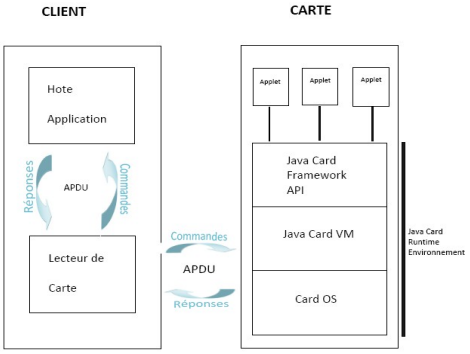
Optional body	Mandatory code	
Data	SW1	SW2

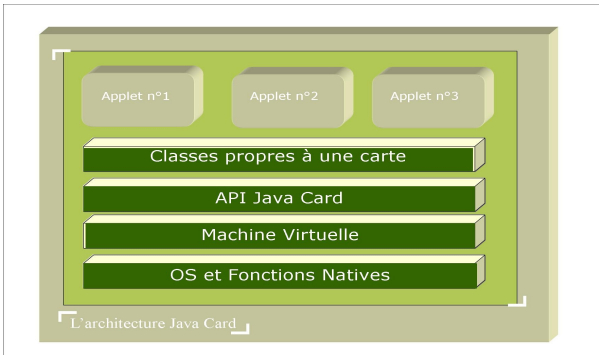
SW1-SW2: Command execution feedback

[4] APDU : Application Protocol Data Unit
[5]Spécifications Global Platform : http://www.globalplatform.org/
[6]RMI : Remote Method Invocation
[7]AID : Application IDentifier
[8]ISO 7816 est la norme acceptée au niveau international pour les cartes à puce, ISO 7816-1 pour les caractéristiques physiques de la carte, ISO 7816-2 pour les dimensions et locations des contacts, ISO 7816-3 pour les signaux et protocoles de transmission, ISO 7816-4 pour les différentes commandes de base de la carte

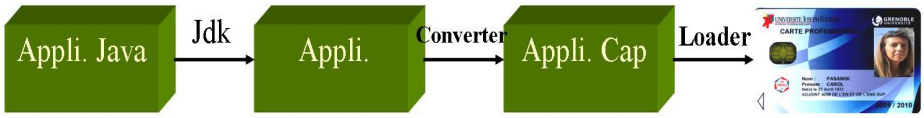
- ***JCVM*** (Java Card Virtual Machine) : nom de la machine virtuelle dans la technologie Java Card, découpée en 2 parties, la première contenant l'interpréteur de bytecode situé sur la carte, et la deuxième, qui contient les autres fonctionnalités classiques d'une machine virtuelle (convertisseur, chargement de classes, vérification du bytecode) ;
- ***JCRE (***Java Card Runtime Environment) : fournit des mécanismes de sécurité (séparation système de la carte et applications). Il gère les ressources de la carte (processeur et données), l'exécution et la sécurité des applets. Celles-ci interagissent avec le JCRE par des APIs spécifiques. Plusieurs applets pourront cohabiter sur une carte, même si elles appartiennent à des fabricants différents. Les minima requis pour faire tourner un programme Java Card sont 24 ko de ROM, 16 ko d'EEPROM et 1 ko de RAM.

Architecture





Chargement d'une applet sur la carte



Les applets sont chargées dans la carte par l'intermédiaire d'un fichier CAP[9]. Ce fichier est généré en deux étapes. Premièrement, les classes Java sont compilées de façon classique (avec javac, la seule différence étant l'utilisation de l'API Java Card et non celle du Java classique) vers des fichiers .class. Ensuite, un outil de conversion (converter), fourni par le kit de développement Java Card, regroupe l'ensemble des classes d'un même package au sein d'un seul fichier CAP. Il effectue également un certain nombre de vérifications sur le code (vérification de la taille des entiers manipulés par exemple) ainsi que l'édition des liens entre les classes.
Concernant le chargement d'une applet sur la carte, il est nécessaire de suivre un schéma d'authentification dont le point d'entrée est la connaissance de la master key. Le schéma utilisé sur la carte IMAG est celui des cartes VISA.

Limitation de la Java Card:

- le framework Java Card 2.x est un sous-ensemble du langage Java ;

[9]*Fichier CAP*. Le *fichier CAP* est le format standard de *fichier* pour la compatibilité binaire de la plate-forme *Java* Card.

– la Java Card supporte les types booléen, byte, short, les exceptions, l'allocation dynamique, méthodes virtuelles, interfaces, tableaux à une dimension, packages, object ;
– la Java Card ne supporte pas : le chargement dynamique des classes, les threads et leur synchronisation, le clonage des objets, la fonction finalize, les tableaux à n dimensions, le ramasse miettes, les types de données float, double, long, char et string.

Sécurité Java Card [4] [5]

La sécurité Java Card repose sur les protections des données et programmes vis à vis d'intrusions extérieures (signature, session avec code PIN, chiffrement des données), mais aussi sur le cloisonnement entre les différentes applications (zones programmes et données) : chaque applet s'exécute dans un espace qui lui est propre, donc elle ne peut avoir d'action sur une autre applet (sécurisation assuré par la JVCM). Il existe aussi un mécanisme de partage permettant de manière explicite d'accéder à des services offerts par une autre applet.

Les API permettent des écritures et lectures cohérentes (commit, rollback).

Quelques algorithmes de cryptage utilisés dans notre Java Card : SHA (Secure Hash Algorithm), DES (Data Encryption Standard), RSA (Rivest, Shamir and Adleman Asymmetric Cipher algorithm).

5 Le projet UFR IMAG : déploiement « applet université »

Déploiement

Après un appel d'offre au printemps, nous avons déployé la solution carte à puce à l'UFR IMAG pendant l'été 2009 avec les fonctionnalités de contrôle d'accès et de personnalisation (visuelle et électrique) de la carte. À la rentrée de septembre 2009, nous avons fait les cartes des étudiants et du personnel sur cette Java Card. C'est la société ARD, spécialiste des solutions de cartes à puce, qui a déployé sa solution suivant notre cahier des charges.

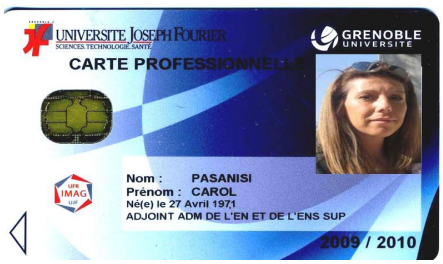
Carte Optelio Contactless D72 R2

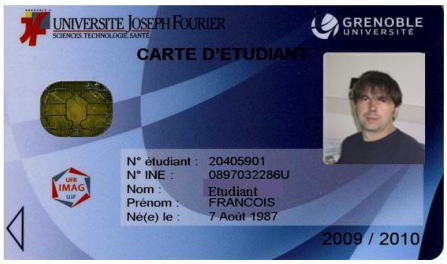
La carte que nous utilisons est une carte : Optelio Contactless D72 R2 fournie par Gemalto:

EEPROM : 68Kbytes, Java Card Release : 2.1.1 Global Plateform release : 2.1.1	Type : Dual interface ISO 14443-1 compliant Contact interface : ISO 7816 Contactless interface : ISO 14443 –2, -3, -4, Type A or Type B

Visuel

Nous avons deux types de cartes, une carte d'étudiant pour les étudiants (avec connexion Apogée), et une carte professionnelle pour le personnel (avec connexion Harpège). L'intégration de ces données et la personnalisation de la carte est réalisée par le logiciel IdConcept, de la société ARD.





Mapping de la carte (personnalisation électrique)

Le mécanisme de sécurité de l'applet Université comporte un cryptage des données dans la carte, des clés et conditions d'accès. Les données sont stockées cryptées (ci-dessous PUBLIC2 et PRIVE) ou en clair (ci-dessous ACCES-S), et leur accès peut être

public (par les gestionnaires, sur login + mdp) ou privé (par le titulaire de la carte, avec code PIN). Voici un extrait du mapping actuel de la carte (qui pourra évoluer rapidement) :

L'APPLET UNIVERSITE : structuration en blocs de données

ACCES	N°BLOC	COMMENTAIRES	Protection	Crypt	sign	pin	DONNEES/COMMANDES
PUBLIC2	0x01	Bloc format TLV (*) : données publiques. Les données sont libres en lecture, transmises cryptées et garanties authentiques par signature. Nota : l'encryption des données est automatique à l'écriture. WRITE PROTECTED & AUTO CRYPTED, READ FREE (NOT DECRYPTED)	w	✓	✓		Données si Etudiant 0x31: Affection component code (5) 0x32: Affectation component label (12) 0x33: Stage code (8) 0x34: Student's INE (13) Données si personnel ou enseignant 0x51: "Indice Nouveau Majoré" (6) 0x52: Employee's CROUS rate code (not used) (7) 0x53: "En poste/Hébergé" (3) 0x54: "Corps" (22) 0x55: Affection component code (7) 0x56: Affectation component Label (22) 0x57: employee number (Harpege Code) (10) Commandes : idem Public1
PRIVE	0x02	Bloc format TLV (*) données privées. Les données sont libres en lecture sous session pin, transmises cryptées et garanties authentiques par signature. Nota : l'encryption des données est automatique à l'écriture WRITE PROTECTED & AUTO CRYPTED, READ PIN-PROTECTED (NOT DECRYPTED)	w	✓	✓	✓	Données si Etudiant 0x81: Grant Holder (3) Données si personnel ou enseignant none Commandes : 0x00 0xB2: Read Record 0x00 0xD2: Write Record 0x00 0x84: Get Challenge (authentified sessions)
ACCES-S	0x80	Blocs signature : données utiles aux signatures des blocs ACCES, PUBLIC et PRIVE (MD5) INTERNAL MANAGEMENT (PROTECTED CUSTOM COMMAND)	w				Commandes: 0x80 0x50: Compute Signature (on given data bloc)
PUBLIC-S	0x81		w				
PRIVE-S	0x82		w				
PIN	0xA0	Données utiles à la gestion du code pin.				✓	Données :
CLE	0xF0	Le pin est "0000" à l'issue du processus	w				(*) TLV : Tag Length Value

Contrôle d'accès logique

Liminaire : dans un premier temps, nous avons choisi de ne déployer l'authentification logique que pour nos terminaux Sun Ray, car ils intègrent facilement cette fonctionnalité (mais nous pourrons l'étendre à tout poste informatique équipé d'un lecteur de carte avec ou sans contact). Cette fonctionnalité utilise l'interface contact de la carte à puce. Avec les Sun Ray, il est possible d'intégrer l'authentification par carte à différents niveaux d'usages.

Plusieurs niveaux d'usages sont donc possibles avec les clients SunRay :

- **1er niveau** : gestion du token (Id) de la carte pour mobilité de session, identification sans authentification. Dans la technologie Sun Ray il est possible de gérer la mobilité de session permettant à un usager de retrouver son environnement d'un terminal à l'autre. Dans ce cas la session est associé à l'ID[10] de la carte avec gestion de l'insertion/suppression de la carte pour redirection du flux graphique (environnement X). Lecture de la carte par commandes APDU ;
- **2ème niveau** : identification du porteur de carte par association (Id carte, nom utilisateur) soit dans la base de Sun Ray serveur (LDAP) ou sinon dans une base de données ou un annuaire externe capable de faire l'association entre l'identifiant de la carte (ID) et le nom de la personne (login). Ce mode qui demande le login (lecture dans la carte) et l'ID de la carte authentifie la carte, mais pas le porteur de cette carte ;
- **3ème niveau** : authentification avec association d'un mécanisme d'authentification :
  - simple : mot de passe, demandé en plus de l'insertion de la carte. C'est le mode retenu (login, ID carte, mot de passe) pour le déploiement du projet pilote à l'UFR IMAG ,
  - forte : jetons (Kerberos[11]) ou certificats (X509) avec code PIN[12], cette fonctionnalité, capable d'authentifier fortement le porteur, est une solution complexe à mettre en œuvre (déploiement d'une PKI[13]).
- **4ème niveau** : authentification applicative, les applications (si elles sont prévues pour cela) peuvent aller lire les données dans la carte à tout moment.

[10]ID unique de la carte, pas obligatoirement le numéro de série mais un identifiant unique
[11]Kerberos est un protocole d'authentification réseau créé au Massachusetts Institute of Technology (MIT). Kerberos utilise un système de tickets au lieu de mots de passe en texte clair.
[12]Code PIN : *Personal Identification Number*
[13]Une **Infrastructure à clés publiques** (ICP) ou **Infrastructure de Gestion de Clefs** (IGC) ou encore **Public Key Infrastructure** (PKI),

Pour l'accès à la carte, le logiciel Sun Ray Serveur utilise une implémentation de PC/SC[14], qui est l'API (bibliothèque) standard pour l'accès à des cartes à puces sous Windows. Une implémentation libre de PC/SC, appelée PC/SC Lite, est disponible sous GNU/Linux. La distribution utilisée par Sun Ray Serveur est une implémentation de PC/SC-lite API dérivée du projet Open Source the Open Source MUSCLE [6].

Utilisation des cartes avec les différents systèmes :

- **pour les postes Windows [7]** : par défaut, avec le framework de base fourni par Windows, il faut utiliser les mécanismes d'authentification forte. Si l'on ne veut pas utiliser les PKI, ce qui est pour le moment notre cas, il faut étendre le module d'authentification de Windows.
  Dans les systèmes NT4, 2000 XP, 2003 serveur, le module qui gère la boîte de dialogue de Login s'appelle GINA (Graphical Identification ans Network Authentication) et se présente sous forme de librairie (dll) chargée par Winlogon. Ce composant est remplaçable, c'est à dire qu'un éditeur de logiciel ou un constructeur de matériel (carte à puce, authentification par emprunte digitale) peut développer son propre module GINA et remplacer celui livré par défaut pour enrichir les fonctionnalités d'authentification. Il ne peut y avoir qu'un seul composant GINA installé et reconnu par le système. Du point de vue éditeur, l'écriture d'un module Gina est compliquée car il doit implémenter toutes les fonctionnalités disponibles comme l'authentification par carte à puces, mais aussi la gestion du bureau à distance ou le double facteur d'authentification. Depuis Windows Vista et Windows Serveur 2008, il y a remplacement du module GINA. Winlogon s'appuie maintenant sur un module appelé LogonUI permettant l'enregistrement de plusieurs modules d'authentification permettant ainsi d'étendre le système (authentification par carte à puces) sans réécrire les fonctions de bases (module authentification initial). Pour modifier le système d'authentification, il faut maintenant créer un fournisseur d'informations d'identification (credential provider).
- **pour les postes Unix/Linux : utilisation des PAM**[15] :
  PAM est un système permettant de gérer individuellement l'authentification des applications. Un des principaux avantages de PAM est que l'authentification est centralisée pour toutes les applications d'un système. Ainsi, il n'est plus nécessaire de réécrire à chaque fois tous les programmes lorsqu'une nouvelle méthode d'authentification apparaît. Dans le cas des cartes à puce, PAM sert d'interface d'authentification entre la carte et le système ou l'application.

Contrôle d'accès physique

Chaque fois qu'il est délivré ou mis à jour une carte, que ce soit pour un étudiant ou un personnel, la base de données du système d'accès aux locaux est mise à jour, avec affectation d'un profil à l'utilisateur (actuellement une dizaine de profils). Chaque profil donne des autorisations d'accès, que ce soit pour des locaux ou des plages horaires. Actuellement, il y a 16 portes (d'entrées, de salles de TP, etc...) qui sont munies de lecteurs de cartes Java Card, avec une généralisation prévue cette année. Les lecteurs lisent les numéros logiques des cartes avec des commandes APDU (même mécanisme que pour les Sun Ray).

6 Bilan et perspectives

Ce qu'a permis la Java Card

La Java Card a permis que la technologie ne soit plus un frein pour faire cohabiter les différentes applications des différents partenaires comme le transport, les banques, les collectivités, les universités, le CROUS. Jusqu'à présent, la plupart des cartes à puce ne permettaient pas une cohabitation simple de plusieurs applications sur une carte puisqu'il fallait que l'usage soit anticipé avant la conception de la carte (BMS2 : Moneo + Calypso) et excluant ainsi intrinsèquement toute possibilité d'ajout de nouvelles applications. L'arrivée d'une carte avec une technologie adaptée à l'aspect « multiservice » a permis de relancer des discussions entres ces différents partenaires.

Les futures applications

Les usages futurs de la carte multiservice Java Card :

[14]PC/SC : Personal computer/Smart Card (ou *PC/SC*) est une bibliothèque logicielle pour l'accès à des cartes à puce sous Microsoft Windows
[15]Pluggable Authentication Module

- transport : intégration de l'applet Calypso sur la carte UFR IMAG, des études sont en cours pour la faisabilité de ce projet en partenariat avec la région Rhône Alpes ;
- Moneo : intégration de l'applet Moneo sur la carte UFR IMAG, des études sont en cours pour la faisabilité de ce projet en partenariat avec BMS ;
- émargement aux examens : permettre aux étudiants d'émarger directement avec leur carte aux examens. L'étudiant utilisera sa carte et son code PIN en début et fin d'examen pour l'authentification. Lors de l'inscription et de la délivrance des cartes, une attention particulière a été apportée aux étudiants sur ce code confidentiel et sur les usages qu'il pourrait avoir dans l'année ;
- bornes pour services interactifs : afin de donner plus d'autonomie aux utilisateurs, nous souhaitons mettre en place des bornes interactives permettant l'accès à des services (potentiels) tels que : consultation de soldes, remise à jour sur carte / réinscription, notion de fidélité, certificat de scolarité, relevés de notes ;
- certification X509 pour mise en place d'une certification forte, mais elle nécessite la mise en place d'une PKI.

Quelques recommandations et ....conclusion

Le choix de la carte est important : il faut une mémoire suffisante pour faire cohabiter les applications et offrir une certaine pérennité à la carte. Voici à titre d'exemple des estimations de taille pour les applets chargées sur la carte et celles en prévision :

- applet université : 10ko ;
- applet Moneo : 9ko ;
- applet Calypso : 13ko.

En raison de la complexité du domaine des cartes à puce, dans un milieu fermé, une Assistance à Maitrise d'Ouvrage spécialisée dans le domaine est nécessaire. Enfin il peut être utile de retenir un interlocuteur unique pour les aspects fourniture de la carte, Système de Gestion de Cartes (SGC), et aussi déploiements, travaux infrastructures de contrôle d'accès.

Conclusion : ce projet nous a permis de mettre en œuvre une technologie novatrice qui a atteint un niveau de maturité technique et permet d'envisager des déploiements à plus grande échelle dans les années à venir. Ce choix nous a permis de mettre en place dans la première applet « université » les usages que nous avions envisagés depuis un certain temps. Malgré la jeunesse de cette technologie, le déploiement des fonctions de contrôle d'accès et de personnalisation de cartes s'est très bien déroulé. Le potentiel de cette carte avec les possibilités d'intégration de nouvelles applications offre de nombreuses perspectives intéressantes dont certaines sont déjà en cours.

7 Bibliographie

[1] Sun Microsystems, Inc. Java Card Technology, http://java.sun.com/products/javacard/

[2] Mike Hendry, *Multi- application Smart Cards*, Chap 9 Java card and GlobalPlateform, Cambridge University Press, 2007

[3] Samia Bouzefran et Gilles Grimaud, *La, programmation Java Card*, Misc,hors-Série n°2 Novembre/Décembre 2008

[4] Damien Sauveron, http://damien.sauveron.free.fr/publications/index.html

[5] Samia Bouzefran , http://cedric.cnam.fr/~bouzefra/

[6] Ludovic Rousseau et David Corcoran, Le projet muscle ou la bibliothèque PC/SC Lite, Linux Magazine n°39 Novembre/Décembre 2008

[7] Emmanuel Dreux, *La sécurité sous Windows Vista*, Editions ENI, 2009