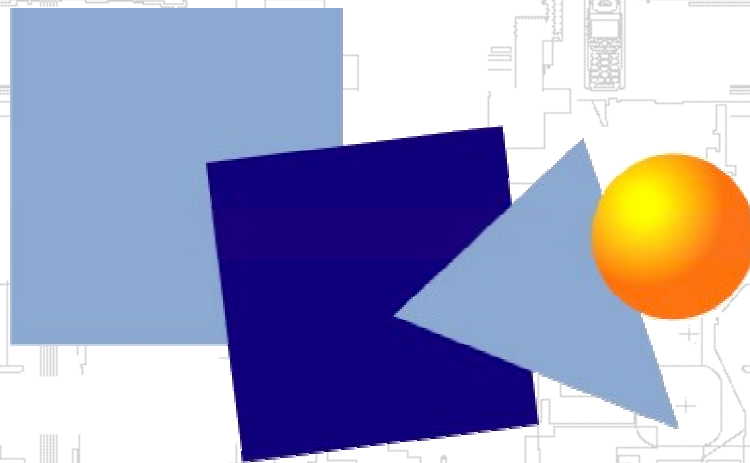


Industrialiser une application de téléchargement et d'administration d'applets sur carte SIM



RAPSODIA
Software & Telecommunications

Rapport de Projet de Fin d'Etude

Présenté par : Bardes Eric
Créé le : 2000/06/02
Responsable : Frick Nicolas
Tuteur : Duda Andrzej



**RAPPORT DE PROJET DE FIN D'ETUDE
INDUSTRIALISER UNE APPLICATION
DE TELECHARGEMENT ET
D'ADMINISTRATION D'APPLETS SUR
CARTE SIM**



<i>Document reference</i>	<i>ALM/SRV/CR</i>
<i>Version</i>	<i>1.0</i>
<i>Date</i>	<i>2000/06/02</i>
<i>Author</i>	<i>Bardes Eric</i>
<i>Number of pages</i>	<i>19</i>



**INDUSTRIALISER UNE APPLICATION DE
TELECHARGEMENT ET D'ADMINISTRATION D'APPLETS
SUR CARTE SIM**

EDITIONS

Issue Date	Version N°	Page	Subject	Authors
2000-06-02	1.0	19	Creation	Bardes Eric



TABLE DES MATIERES

1 RAPSODIA : L'ENTREPRISE.....	6
2 NOTIONS UTILES	7
2.1 La Carte SIM (Subscriber Identify Module)	7
2.2 Le SMS (Short Message Service)	7
2.3 L'ASP (Application Server Platform)	7
2.4 L'ALM (Application Loader and Manager).....	8
3 LES OBJECTIFS.....	9
4 ARCHITECTURE CHOISIE.....	10
4.1 L'Interface graphique.....	10
4.2 Le serveur HTTP.....	11
4.3 Le module d'administration.....	11
4.4 L'ALM	12
4.5 L'ASP	12
4.6 Le pilote du lecteur de carte	12
5 UN POINT TECHNIQUE : LE ROUTAGE DANS L'ASP.....	14
6 LE PLANNING.....	15
7 CONCLUSION	16
8 BIBLIOGRAPHIE.....	17
8.1 La documentation interne	17
8.2 Les Normes	17
9 L'ENVIRONNEMENT	18
9.1 Le Matériel.....	18
9.2 Les logiciels	18
9.2.1 Le développement	18
9.2.2 Les tests	18
10 GLOSSAIRE	19



***INDUSTRIALISER UNE APPLICATION DE
TELECHARGEMENT ET D'ADMINISTRATION D'APPLETS
SUR CARTE SIM***

1 RAPSODIA : L'ENTREPRISE

RAPSODIA est une START-UP localisée sur Boulogne. Elle a été fondée en juin 1999 par Olivier LEROUX. Issue d'un partenariat de deux entreprises : Prosodie et Oberthur, elle bénéficie d'un solide savoir-faire dans le monde des cartes à puces et des opérateurs de télécommunication.

Elle a pour vocation de fournir des logiciels modulables à des réseaux d'intégrateurs tels que Prosodie. Ceux-ci permettent de développer des solutions SIM Toolkit adaptées aux besoins spécifiques de chaque opérateur GSM ou fournisseur d'information.

Le SIM Toolkit est une API Java présente sur la carte SIM offrant la possibilité d'accéder, sur les téléphones mobiles compatibles SIM Toolkit, aux ressources de celui-ci.

Constituée d'une équipe commerciale de six personnes et d'une équipe technique de dix huit personnes, elle est très présente sur le marché international. Principalement active en Asie où le monde GSM subit une expansion considérable, elle commence à se positionner sur le marché européen avec très récemment un contrat signé en Italie et des projets pilotes lancés en Espagne et en Angleterre.

Son activité s'est élargi et propose également :

- du développement de logiciels serveurs et applicatifs destinés aux services à valeur ajoutée mobiles
- du conseil
- de la formation
- de la maintenance
- du support



2 NOTIONS UTILES

2.1 La Carte SIM (Subscriber Identify Module)

La carte SIM est une carte à puce disposant d'un système d'exploitation et d'un système de fichiers. Les fichiers permettent de contenir un certain nombre d'informations propres à l'opérateur et des informations propres à l'abonné.

La nouvelle génération de carte dites SIM ToolKit PHASE2+, disposent d'une machine virtuelle Java. De plus la capacité de stockage a été améliorée (jusqu'à 32ko de mémoire). On peut alors y placer des applets java que le système d'exploitation de la carte peut exécuter.

Les téléphones mobiles compatibles SIM ToolKit permettent à la carte d'utiliser leurs ressources (par exemple le clavier ou l'écran).

Les cartes SIM ToolKit communiquent avec l'extérieur, à travers du mobile, avec le SMS.

2.2 Le SMS (Short Message Service)

Le SMS fournit aux réseaux GSM la possibilité d'échanger des messages courts (SM : Short Messages). Ceux-ci sont utilisés de manière standard pour la signalisation et le contrôle du réseau (localisation, authentification, notification...). Les téléphones mobiles peuvent émettre et recevoir des SM dès qu'ils sont allumés.

Les messages courts peuvent contenir jusqu'à 160 caractères alphanumériques (figure 1).

Les SM reçus par le mobile peuvent être transmis à la carte pour être enregistrés ou traités.

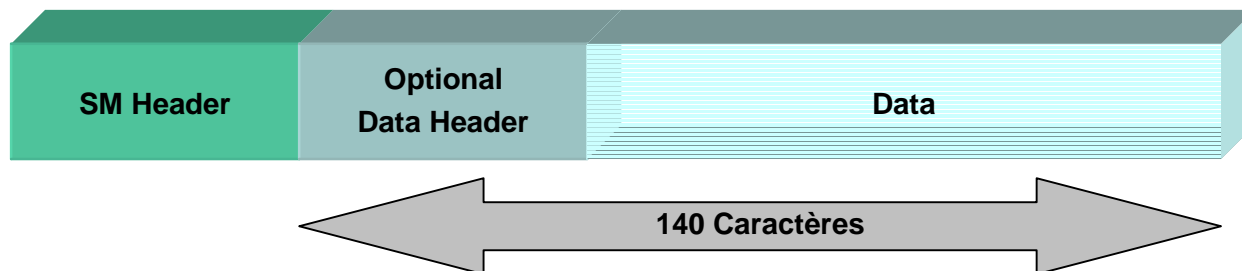


Figure 1 : format général d'un message court

2.3 L'ASP (Application Server Platform)

Pour offrir des services à valeur ajoutée, on peut développer des applications GSM en utilisant le SIM ToolKit (figure 2).

L'ASP offre à ces applications une interface unifiée avec le réseau GSM. Connecté à différents SM SC (Short Message Service Center), il se charge de formater les données et de les envoyer aux applets par SMS. D'autre part, il permet de transmettre les données reçues d'une applet à l'application concernée.

2.4 L'ALM (Application Loader and Manager)

L'ALM est une application GSM qui s'appuie sur l'ASP. Cette application permet aux opérateurs GSM d'administrer les cartes de leurs abonnés «Over The Air».

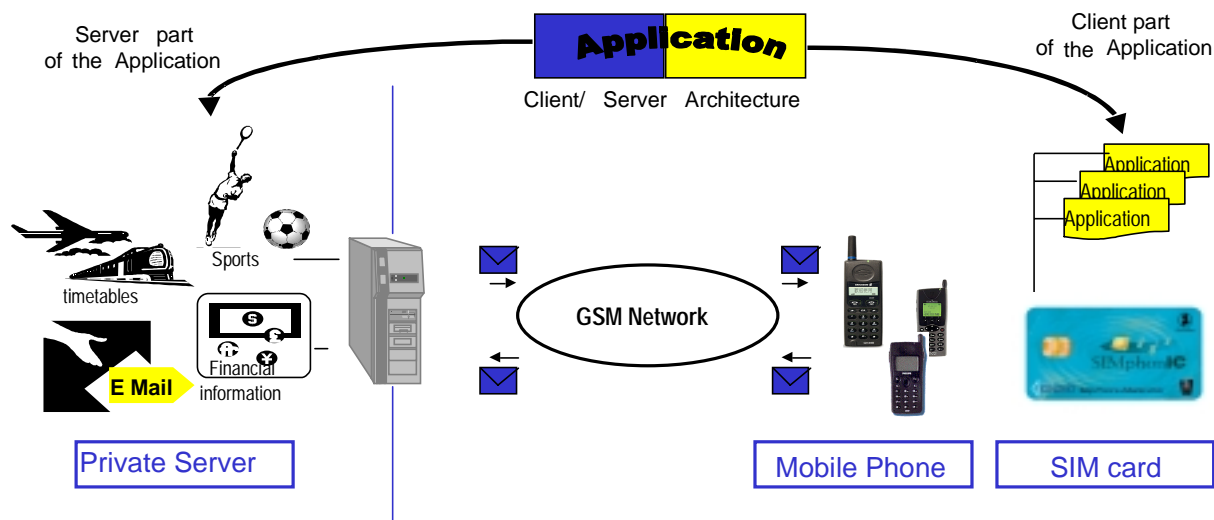


Figure 2 : Architecture générale d'une application GSM

Les fonctionnalités de cette application sont :

- téléchargement/effacement de fichiers GSM
- blocage/déblocage de fichiers GSM
- mise à jour de fichiers GSM
- changement de la taille allouée a des fichiers GSM

Avec ce mécanisme, l'opérateur peut également charger des applets qui sont stockées sous forme d'ensemble de fichiers GSM.

Cette application utilise des informations provenant de différentes sources :

- la base de donnée de l'opérateur pour les informations sur les abonnés
- des données provenant des constructeurs de carte pour les informations sur les fonctionnalités particulières des différentes cartes ainsi que les différents protocoles supportés
- des données provenant des différents fournisseurs de service pour les informations propres aux applets.



3 LES OBJECTIFS

Avant le début du projet, l'ALM n'était encore qu'un projet pilote. Comme il l'a été présenté lors de la pré-soutenance, je devais intégrer une équipe pour travailler sur l'industrialisation de ce produit. Il avait, alors été présenté un planning prévisionnel des tâches de l'équipe.

Parmi les différents points de cette industrialisation, je me suis occupé principalement d'ajoutée une nouvelle fonctionnalité au produit : le «self provisioning» internet.

L'objectif étant de permettre aux opérateurs d'offrir à leurs abonnés la possibilité de charger à leur demande les applications. Pour cela les opérateurs proposeront à leurs clients des packs contenant un lecteur de carte.

Les abonnés pourront alors de chez eux, avec leur carte dans un lecteur, commander à partir d'une page Web le chargement de l'application choisie.

Pour les clients ne disposant pas d'ordinateurs ou de connexion internet, il sera possible de se rendre dans un point de vente de l'opérateur pour demander ce téléchargement.

Initialement, l'application destinée à piloter le lecteur de carte pour les téléchargements devait être développée par notre partenaire Oberthur, fournisseur des lecteurs.

Compte tenu des délais annoncés pour réaliser cette application, il a été décidé de prendre en charge le développement.

Cette nouvelle fonctionnalité implique également des modifications sur l'ASP. En effet, contrairement aux applications classiques, les données ne seront pas uniquement véhiculées sur le réseau GSM, mais pourront aussi être émises via internet.

La difficulté étant d'avoir un mécanisme de chargement similaire par internet et par le réseau GSM, de manière à ce que cela soit transparent pour l'application, et que cela ne modifie pas toute l'architecture de l'ASP.

4 ARCHITECTURE CHOISIE

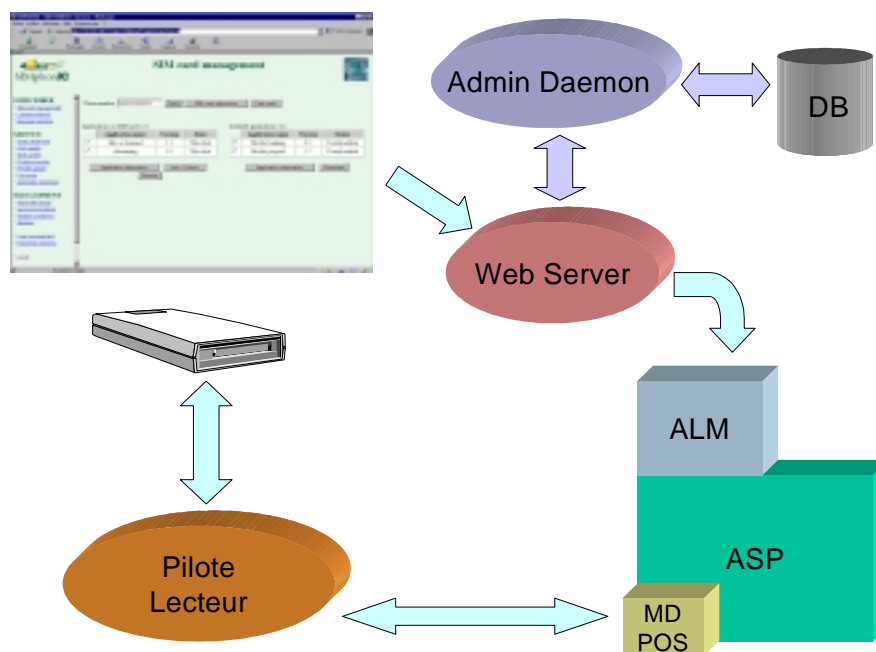


Figure 3 : Architecture "Self Provisioning" Internet

Depuis une interface graphique, l'utilisateur s'authentifie auprès du serveur. Une fois la connexion validée par un module d'administration, il lui est possible de choisir des applets parmi celles qui lui sont autorisées.

Lorsqu'un téléchargement est demandé, la requête est transmise à l'ALM. Celui-ci, par le biais de l'ASP envoie alors l'applet sous forme de SM au pilote du lecteur de carte qui les envoie à la carte de la même manière qu'un téléphone mobile.

4.1 L'Interface graphique

L'interface graphique utilisateur est fournie sous forme d'une page web. Reposant sur des scripts cgi en perl et du javascript, elle bénéficie d'une grande souplesse d'intégration car il est aisé de modifier le contenu.

De plus les pages HTML permettent aux opérateurs d'effectuer des modifications dynamiquement sans avoir à fournir à tous les abonnés une nouvelle version du logiciel.

Enfin, l'avantage majeur de cette solution, est qu'elle s'adapte à tous les environnements possibles (Mac, PC Windows/Linux/OS2 ...). En effet, des navigateurs internet gratuits sont disponibles pour toutes les configurations. Cela évite donc de porter sur chaque configuration une interface propriétaire.



Il aurait été possible de développer une interface graphique en Java, mais les applications java sont encore trop dépendantes des machines virtuelles installées.

4.2 Le serveur HTTP

N'importe quel serveur capable d'exécuter des scripts cgi écrits en perl peut être utilisé !

Pour les versions de démonstration, le serveur libre de droit Apache est proposé. Il est relativement simple à configurer, et fonctionne sur la plus part des machines UNIX.

Pour accroître la confidentialité des informations échangées, peut être utilisé un serveur web supportant le protocole HTTPS permettant des échanges cryptés.

4.3 Le module d'administration

Le «Self Provisioning» internet exige du système une grande robustesse. En effet, ouvert sur web, il est susceptible de subir de lourdes charges. De plus, intervient tous les problèmes de confidentialité liés au réseau.

Le module d'administration renforce le serveur web, il permet d'attribuer à chaque utilisateur authentifié une session.

Pour réaliser cela, le module attribue à chaque nouveau connecté valide un numéro de session unique. Celui-ci est conservé par l'utilisateur sous la forme d'un cookie avec le numéro d'identification de l'utilisateur.

A chaque chargement de page, le module d'administration compare le contenu du cookie et l'adresse IP d'origine avec les données qu'il conserve pour chaque session.

Le module d'administration stocke localement toutes les informations sur les droits de l'utilisateur au début de chaque session et les conserve jusqu'à ce qu'elle se termine. Ainsi, la base de donnée n'est pas consultée pour chaque nouvelle page chargée.

L'autre tâche du module d'administration, est d'effectuer toutes les requêtes vers la base de donnée. Elle n'est pas consultée directement par les scripts cgi. De cette manière, il y a une garantie que seules les données accessibles sont consultées, et également de protection la base de donnée contre les surcharges éventuelles. En effet, celle-ci doit rester hautement disponible pour l'ALM. Pour cela, le module est configuré pour accepter un nombre déterminé de sessions simultanées.

Un système de notification d'événements permet de notifier à l'administrateur les saturations encourues. Ainsi, celui-ci peut redimensionner le système pour mieux satisfaire les abonnés.

La communication entre les scripts cgi et le module d'administration passe par IP avec l'utilisation de sockets. Il en est de même pour l'interrogation de la base de donnée. Cela permet d'installer le serveur http, la base de donnée et le module d'administration sur des machines distinctes. Ce dernier est écrit en C ANSI et bénéficie donc d'une grande portabilité.

L'API cliente est également développée en C ANSI. Le langage XS permet d'interfacier celle-ci dans des scripts perl en fournissant un nouveau module perl.



4.4 L'ALM

A ce niveau, le fonctionnement est le même que pour un téléchargement «over The Air». Aucune modification n'a donc dû être apportée à l'application.

4.5 L'ASP

Conçu de manière très modulaire, les modifications nécessaires au téléchargement par internet ont été mineures. Pour limiter les changements, il a été décidé de conserver le mécanisme de base.

Comme il l'a été mentionné précédemment, l'ASP interagit avec le réseau GSM grâce à des connexions à des SM SC. L'architecture de l'ASP permet l'adaptation aux différences introduites par les différents constructeurs de SM SC.

En effet, les données fournies par les applications, sont d'abord encapsulées dans des SM formatés conformément à la norme GSM 03.40. Ensuite, un module de l'ASP : un MD (Media Driver) est dédié à chaque SM SC connecté, se charge d'adapter ce SM standard pour satisfaire le protocole du SM SC.

Pour le «self provisioning » Internet, il a donc suffit de spécifier un protocole de communication avec le pilote du lecteur de carte en se basant sur les SM. Puis de développer un MD intégrant ce protocole.

Ce MD, nommé MD POS en rapport avec le nom du pilote du lecteur de carte : le module point de vente (Point of Sale), présente des différences notoires avec les autres MD :

il est du type serveur (la connexion est à l'initiative du pilote) alors que les autres sont de type client

Il doit gérer plusieurs connexions simultanément (plusieurs abonnés peuvent se connecter au même moment) alors que les autres n'ont qu'un seul interlocuteur.

Ces différences induisent des difficultés supplémentaires. Entre autre, la connexion de différents clients impose un routage au cœur du MD POS. Ce routage sera décrit plus en détail dans les points techniques.

4.6 Le pilote du lecteur de carte

Ce logiciel est destiné à ouvrir une connexion vers le MD POS présenté plus haut via IP, puis à transférer les SM reçus en suivant le protocole dédié à cette application, à la carte conformément à la norme GSM 11.11 (spécifications en annexe).

Le téléchargement ne peut s'effectuer qu'une fois la connexion établie. De ce fait, il est nécessaire de synchroniser la connexion avec la demande de téléchargement.

Les navigateurs internet offrent différentes solutions pour réaliser cela :

- Utiliser un applet Java
- Utiliser un plug'in
- Utiliser un «application helper»

L'applet Java tourne dans un environnement protégé non autorisé à accéder aux ressources système comme les périphériques à moins de satisfaire un certain nombre de conditions contraignantes (applet signée,...).



**INDUSTRIALISER UNE APPLICATION DE
TELECHARGEMENT ET D'ADMINISTRATION D'APPLETS
SUR CARTE SIM**

Le plug'in (module d'extension d'un navigateur) est trop dépendant de celui-ci. Cela implique une limitation inutile car il faudrait alors proposer une version pour chaque navigateur existant, ce qui augmenterait la charge ultérieure du support technique.

L' «application helper» est une application capable de lire des fichiers. Lorsqu'un fichier, de type MIME correspondant à celui attribué à l'application, doit être chargé par le navigateur et qu'il ne sait pas l'interpréter, il fait appel à l'application pour le traiter.

Cette dernière solution a été retenue. Elle présente en plus l'avantage de pouvoir lui fournir un fichier que nous utiliserons comme un fichier de configuration.

Ceci est particulièrement intéressant car cela permet à l'opérateur de changer sa configuration de manière transparente pour l'utilisateur.

Pour des raisons de taille, le module POS est développé en C++. Le module doit être le plus léger possible pour pouvoir être facilement téléchargeable sur internet dans des délais raisonnables.

Actuellement les APIs des pilotes des lecteurs de carte ne sont disponibles que pour des machines Intel sous Windows. C'est pourquoi le problème de portabilité du module POS n'est pas encore posé.

Dédié aux architectures Windows, il est développé avec en utilisant les MFC (Microsoft Foundation Class) et l'API Win32.

Les APIs pour les lecteurs n'offrent pas la possibilité d'utiliser les interruptions associées au lecteur. Pour cette raison, la gestion du lecteur doit se faire de manière synchrone. Comme les transferts avec la carte sont des opérations lentes, pour ne pas saturer le flux de données provenant de l'ALM, le module POS a une architecture multi-threads. Une thread gère les échanges avec le serveur et une autre les échanges avec la carte.



5 UN POINT TECHNIQUE : LE ROUTAGE DANS L'ASP

Traditionnellement, dans les applications GSM, les cartes sont désignées par un MSISDN (Mobile Station international ISDN number), plus communément appelé le numéro de téléphone. Ce MSISDN est affecté par le réseau lorsque le mobile s'est connecté : la carte SIM n'en a pas connaissance.

Chaque SM SC du réseau peut adresser une plage prédéfini de MSISDN. Toutes les plages des différents SM SC d'un même opérateur sont disjointes. De plus chaque opérateur dispose de plage de MSISDN disjointes pour leurs abonnés.

Ainsi, pour envoyer un SM vers un MSISDN donné, il suffit de trouver la plage de numéro le contenant parmi celles des différents SM SC connectés et de l'envoyer à celui concerné.

L'ASP dispose d'un router qui connait les différentes plages des différents SM SC connectés.

Lorsqu'un abonné demande un téléchargement «Over The Air», son MSISDN est récupéré dans la base de donnée, et le router indique ensuite le SM SC visé.

Pour le «self provisioning internet», la carte n'est plus dans un mobile relié au réseau mais dans un lecteur de carte.

On a donc deux problèmes à résoudre, sachant que l'utilisateur ne doit pas avoir à saisir son MSISDN :

- Comment diriger les informations vers le bon MD POS, alors que le router désigne un autre MD
- Comment une fois le MD POS trouvé, celui-ci va-t-il savoir vers quelle connexion faut-il faire suivre les informations.

La première idée a été, en constatant que la connexion au MD POS et la requête web provenaient de la même adresse IP, de concaténer celle-ci au MSISDN obtenu dans la base de donnée puis d'envoyer les informations directement vers le MD POS. Là le MD POS parcourait le numéro de téléphone pour en extraire l'adresse IP et retrouver la bonne connexion.

Cette solution n'est pas satisfaisante car c'est un «bricolage» pour contourner le fonctionnement normal de l'ASP. De plus, cette solution restreint l'utilisation à un unique MD POS. Celle qui a été adopté est la suivante :

Lorsque l'utilisateur commande le téléchargement, un fichier de type MIME propre au module POS est généré contenant le MSISDN obtenu dans la base de donnée. Le module POS initialise alors sa connexion avec une trame particulière contenant le MSISDN.

Le MD POS, à la réception de cette trame, met à jours une table d'association MSISDN/numéro de connexion.

Enfin le router a été modifié pour permettre un routage dynamique. Le MD POS déclare auprès du router ce nouveau MSISDN. Lors d'une interrogation, celui-ci parcourt sa table dynamique à la recherche du numéro et s'il n'y figure pas, il détermine le SM SC d'après les plages de numéro.

6 LE PLANNING

Voici le planning détaillé du déroulement du projet :

		fev.		mars				avril				mai				juin			
		3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Etude bibliographique et expérimentation de l'ALM																			
Spécification du module POS																			
MD POS	Dev & Test API IP Asynchrone																		
	Spécification																		
	Développement																		
	Modification du routeur																		
	Tests																		
Administration Daemon	Analyse & Spécification																		
	Développement Admin daemon																		
	Développement API Client + module perf																		
	Tests + intégration dans les scripts cgi																		
module POS																			

Figure 4 : Planning du projet

Celui-ci n'a pas du tout la même physionomie que celui présenté lors de la présoutenance. En effet, à cette occasion, il avait été donné un planning prévisionnel de l'activité de l'équipe. Suite à celle-ci, nous avons cherché à définir un sous-ensemble cohérent d'activité qui pourrait faire un projet plus structuré.

De plus, au moment de la présoutenance, les deux tâches : Administration daemon et module POS n'étaient pas encore connues. Celles-ci sont issues d'une phase classique de l'industrialisation : l'adaptation aux besoins du client.

Par ailleurs, comme il l'a déjà été mentionné, le développement du module POS devait être pris en charge par Oberthur. Après une réunion technique sur les spécifications de celui-ci, la charge a été évaluée par notre partenaire à deux mois et demi de travail. Comme RAPSODIA souhaitait conserver la maîtrise de cette application de manière à pouvoir la faire évoluer, il a été décidé de le prendre en charge.

Le planning montre un retour d'activité sur la tâche liée au MD POS. Ceci s'explique par le simple fait que l'activité a dû être arrêtée car nous ne disposions pas encore des outils permettant de le tester, le module POS n'étant pas disponible. Or il existe un logiciel déjà mentionné : le Composer qui intègre les mêmes fonctionnalités que le module POS bien que plus complet. Cet outil dispose d'un MD qui lui est attribué lui permettant de se connecter à l'ASP pour effectuer des tests. Le protocole est le même que celui utilisé par le module POS excepté la trame d'initialisation qui n'existe pas avec le Composer. Nous avons donc demandé à Oberthur (les créateurs du Composer) d'ajouter cette trame pour que nous puissions tester notre MD POS avec les modifications apportées au routeur. Il a donc fallu attendre que cela soit fait avant d'achever les tests.



7 CONCLUSION

A la fin de ces quatre mois de projet, les principaux objectifs sont atteints. Le media driver IP multi-point dédié au module POS est intégré dans l'ASP. Le module d'administration est fonctionnel, il ne lui manque plus qu'à implémenter certaines fonctions d'interrogation de la base de donnée. Mais cela ne pose aucune difficulté car il ne reste plus qu'à formater celles-ci. Par contre, le module POS n'est qu'à moitié développé. L'architecture est déterminée et les choix d'implémentation sont faits. Il aurait pu être achevé si des informations techniques n'avaient pas tardées. En effet, Oberthur devait fournir des documents, mais en raison de leur déménagement, il a fallu attendre une semaine avant de les obtenir.

Malgré cela, on peut dès à présent effectuer le téléchargement par internet avec un lecteur de carte en utilisant le Composer : logiciel développé par Oberthur simulant un mobile.

Ce fut un projet plongé au cœur de la réalité du monde professionnel car une solution industrielle de l'ALM doit être livrée fin septembre en Italie. Il a donc fallu aménager le planning pour s'adapter aux priorités de l'entreprise et ne pas risquer de bloquer les autres membres de l'équipe. Cette situation est très stimulante. De plus cette expérience a été très enrichissante, puisque j'ai eu l'opportunité de prendre part aux discussions concernant le module POS avec les sous-traitants en tant qu'expert technique.

La solution de «self provisioning» internet étant à peine développée que des évolutions sont prévues. Il est envisagé d'utiliser le même procédé pour charger des applications sur les cartes «Galactic» d'Oberthur. Ces cartes sont des cartes bancaires disposant elles aussi d'une machine virtuelle Java. Le téléchargement ne se ferait plus conformément à la norme GSM 11.11 mais suivant le standard VOP (VISA Open Platform).

D'un point de vue plus générale, l'arrivée du GPRS (General Packet Radio Service) et ultérieurement de l'UMTS (Universal Mobile Telecommunications System), risque de changer le contexte car ces deux technologies sont sensées fournir un SMS avec des débits beaucoup plus élevé qu'avec le support actuel permettant alors des échanges plus volumineux.



8 BIBLIOGRAPHIE

8.1 La documentation interne

Les documents internes concernant l'ASP et l'ALM :

- Application Server Platform – Administration Guide
- Application Server Platform – Reference Guide
- SIMphonIC Application Loader and Manager - Functional Spécifications
- Application Loader & Manager – User Guide

8.2 Les Normes

Les documents provenant de l'ETSI (European Telecommunications Standards Institute) permettent de comprendre les mécanismes du GSM ; en voici la liste :

- GSM 03.40 : « Digital cellular telecommunications system (Phase 2+) ; Technical realization of the Short Message Service (SMS) Point-to-Point (PP) »
- GSM 11.11 : « Digital Cellular telecommunications system (Phase 2+) ; Specification of the Suscriber Identity Module – Mobile Equipment (SIM – ME) interface »
- GSM 11.14 : « Digital Cellular telecommunications system (Phase 2+) ; Specification of the SIM Application Toolkit for the Suscriber Identity Module – Mobile Equipment (SIM – ME) interface »



9 L'ENVIRONNEMENT

9.1 Le Matériel

Quatre types de machines sous environnement UNIX sont à disposition pour effectuer les portages nécessaires et les tests d'intégration :

- Des PC intel X86 sous Linux RedHat 6.0
- Une station Sun Ultra 5 sous Solaris
- Une station HP sous HP-UX
- Une station IBM sous AIX

D'autre part, pour les tests applicatifs, nous disposons d'un certain nombre de mobiles de divers constructeurs.

9.2 Les logiciels

La suite Office de Microsoft est utilisée pour la documentation et les présentations.

9.2.1 Le développement

Tout le développement se fait en C ANSI pour favoriser la portabilité.

Les compilots utilisés sont ceux fournis par les constructeurs pour les stations et le compilot gcc de GNU pour les machines sous Linux.

La gestion de versions est faite avec le logiciel en ligne de commandes CVS.

Pour le module POS, le développement se fait Microsoft Visual Studio 6.0

9.2.2 Les tests

SIMphonIC Composer pour simuler l'environnement GSM jusqu'au mobile.

Le serveur WEB Apache sur Linux.

Les serveurs SQL Postgres, SYBASE et Oracle sur Linux.



10 GLOSSAIRE

- **ALM** : **Application Loader and Manager**
- **ASP** : **Application Server Platform**
- **ETSI** : **European Telecommunications Standards Institute**
- **GPRS** : **General Packet Radio Service**
- **GSM** : **Global System for Mobile communication**
- **MD** : **Media Driver**
- **MSISDN** : **Mobile Station ISDN number**
- **POS** : **Point Of Sale**
- **SC** : **Service Center**
- **SIM** : **Subscriber Identity Modul**
- **SM** : **Short Message ou message court**
- **SMS** : **Short Message Service**
- **UMTS** : **Universal Mobile Telecommunications System**
- **VOP** : **Visa Open platform**