

12

DEMANDE DE BREVET D'INVENTION

A1

22 Date de dépôt : 19.02.99.

30 Priorité :

43 Date de mise à la disposition du public de la demande : 08.09.00 Bulletin 00/36.

56 Liste des documents cités dans le rapport de recherche préliminaire : *Se reporter à la fin du présent fascicule*

60 Références à d'autres documents nationaux apparentés :

71 Demandeur(s) : BULL CP8 Société anonyme — FR.

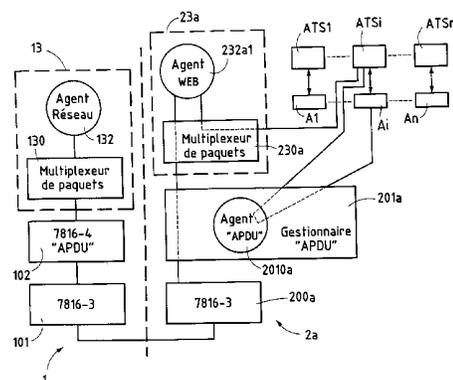
72 Inventeur(s) : BOUDOU ALAIN, SIEGELIN CHRISTOPH et URIEN PASCAL.

73 Titulaire(s) :

74 Mandataire(s) : BULL SA.

54 PROCÉDE D'ACTIVATION D'APPLICATIONS LOCALISEES DANS UNE CARTE A PUCE PAR UN NAVIGATEUR DU TYPE DIT "WEB".

57 L'invention concerne un procédé d'activation d'une (A_i) des applications (A₁-A_n) d'une carte à puce (2a) connectée à un terminal (1), muni d'un lecteur, par l'intermédiaire d'un navigateur de type "WEB". Le terminal (1) et la carte à puce (2a) comprennent chacun, outre des couches protocolaires conventionnelles (101-102), répondant aux normes ISO 7816, une couche spécifique (13, 23a). Cette dernière comprend des agents intelligents (132, 232a₁) pour l'établissement de sessions d'échanges bidirectionnels de données, ce qui permet à la carte à puce (2a) de présenter une fonctionnalité de serveur "WEB". En outre, la carte à puce (2a) comprend des agents intelligents (ATS₁-ATS_n), dits traducteurs de scripts, interagissant avec les applications (A₁-A_n), ce qui permet d'activer une application sélectionnée (A_i), à l'aide d'ordres conventionnels, répondant aux normes ISO 7816. Dans une variante de réalisation, on prévoit des dispositions pour une mise à jour dynamique de la carte à puce (2a), notamment à partir du navigateur.



L

PROCEDE D'ACTIVATION D'APPLICATIONS LOCALISEES DANS UNE CARTE A PUCE PAR UN NAVIGATEUR DU TYPE DIT "WEB"

La présente invention concerne un procédé
d'activation d'applications localisées dans une carte à puce
5 par un navigateur du type dit "WEB".

Le procédé selon l'invention est plus
particulièrement concerné par une station d'utilisateur
munie d'un lecteur de carte à "puce" et connectée au réseau
Internet.

10 Dans le cadre de l'invention, le terme "station
d'utilisateur" doit être compris dans un sens général. La
station précitée peut être notamment constituée par un
ordinateur personnel fonctionnant sous divers systèmes
d'exploitation, tels WINDOWS ou UNIX (tous deux étant des
15 marques déposées). Elle peut être aussi constituée par une
station de travail, un ordinateur portable ou un terminal de
carte, dit dédié.

De même, dans le cadre de l'invention, le terme
"réseau Internet" englobe, outre le réseau Internet
20 proprement dit, les réseaux privés d'entreprises ou
similaires, dits "intranet", et les réseaux les prolongeant
vers l'extérieur, dits "extranet", de façon général tout
réseau dans lequel les échanges de données s'effectuent
selon un protocole du type internet.

25 Dans ce qui suit, sans en limiter en quoi que ce
soit la portée, on se placera dans le cadre de l'application
préférée de l'invention, sauf mention contraire. On
considérera donc une station d'utilisateur, que l'on
appellera simplement "terminal", munie d'un lecteur de carte
30 à puce et connecté à un réseau de type Internet.

Un système d'application à base de carte à puce
comporte généralement les éléments principaux suivants :

2

- une carte à puce ;
- un système hôte constituant le terminal précité ;
- un réseau de communication, à savoir le réseau Internet dans l'application préférée ;
- 5 - et un serveur d'application connecté au réseau.

La figure 1A illustre schématiquement un exemple d'architecture de ce type. Le terminal 1, par exemple un ordinateur individuel, comporte un lecteur 3 de carte à puce 2. Ce lecteur 3 peut être ou non physiquement intégré dans le terminal 1. La carte à puce 2 comporte un circuit intégré 20 dont des connexions d'entrées-sorties affleurent en surface de son support pour autoriser une alimentation en énergie électrique et des communications avec le terminal 1. Ce dernier comprend des circuits d'accès à un réseau de transmissions de données RI. Ces circuits dépendent, notamment, de la nature du réseau RI et du terminal 1. A titre d'exemples, il peut s'agir d'une carte réseau pour un réseau de type local ou d'un modem pour se connecter à une ligne téléphonique commutée ou à un réseau numérique à intégration de services ("RNIS"), pour se connecter au réseau Internet, par exemple via un prestataire de services Internet ("Internet Service Provider" ou "ISP", selon la terminologie anglo-saxonne).

Le terminal 1 comprend naturellement tous les circuits et organes nécessaires à son bon fonctionnement, et qui n'ont pas été représentés dans un but de simplification du dessin : unité centrale, mémoires vive et fixe, mémoire de masse à disque magnétique, lecteur de disquette et/ou de CédéRom, etc.

Habituellement, le terminal 1 est aussi relié à des périphériques classiques, intégrés ou non, tels un écran de visualisation 5 et un clavier 6.

Le terminal 1 peut être mis en communication avec des serveurs connectés au réseau RI, dont un seul, 4, est

illustré sur la figure 1. Dans le cas de l'application préférée de l'invention, les circuits d'accès 11 mettent le terminal 1 en communication avec les serveurs 4 grâce à un logiciel particulier 10, appelé navigateur de type "WEB", ou "browser" selon la terminologie anglo-saxonne. Celui-ci permet d'accéder à diverses applications réparties sur l'ensemble du réseau RI, généralement selon un mode "client-serveur".

Habituellement, les communications sur les réseaux s'effectuent conformément à des protocoles répondant à des standards comprenant plusieurs couches logicielles superposés. Dans le cas d'un réseau RI de type Internet, les communications s'effectuent selon des protocoles spécifiques à ce type de communications, qui seront détaillés ci-après, mais qui comprennent également plusieurs couches logicielles. Le protocole de communication est choisi en fonction de l'application plus particulièrement visée : interrogation de pages "WEB", transferts de fichiers, courrier électronique (e-mel, ou "e-mail" selon la terminologie anglo-saxonne), forums ou "news", etc.

L'architecture logique du système comprenant un terminal, un lecteur de carte à puce et la carte à puce, est représentée schématiquement par la figure 1B. Elle est décrite par la norme ISO 7816, qui elle-même comportent plusieurs sous-ensembles :

- ISO 7816-1 et 7816-2, en ce qui concerne les dimensions et le marquage des cartes ;
- ISO 7816-3, en ce qui concerne le transfert de données entre le terminal et la carte à puce ; et
- ISO 7816-4, en ce qui concerne la structure du jeu d'ordres et le format des commandes.

Sur la figure 1B, du côté terminal 1, on n'a représenté que les couches répondant à la norme ISO 7816-3, référencées 101, et le gestionnaire d'ordres "APDU" (norme ISO 7816-4),

référéncé 201. Du côté carte à puce 2, les couches répondant à la norme ISO 7816-3 sont référéncées 200 et le gestionnaire d'ordres "ADPU" (norme ISO 7816-4) est référéncé 201. Les applications sont référéncées $A_1, \dots, A_i, \dots, A_n$; n étant le nombre maximum d'applications présentes sur la carte à puce 2.

Une application "cardlet", A_i , présente dans la carte à puce 2 (figure 1A), dialogue avec le terminal 1 au moyen d'un jeu d'ordres. Ce jeu présente typiquement des ordres d'écriture et des ordres de lecture. Le format des ordres est connu sous l'abréviation anglo-saxonne de "APDU" (pour "Application Protocol Data Unit"). Il est défini par la norme ISO 7816-4 précitée. Un "APDU" de commande est noté "APDU.command" et un "APDU" de réponse est noté "APDU.response". Les "APDU" sont échangés entre le lecteur de carte et la carte à puce au moyen d'un protocole spécifié par la norme ISO 7816-3 précitée (par exemple en mode caractère : $T=0$; ou en mode bloc : $T=1$).

Lorsque la carte à puce 2 inclue plusieurs applications distinctes, comme illustré sur la figure 1B, on parle de carte multi-applicative. Cependant, le terminal 1 dialogue avec une seule application à la fois. Une application A_i se présente, par exemple, sous la forme d'une pièce de logicielle, dite "applet", en langage "JAVA" (marque déposée), que l'on appellera ci-après "cardlet". La sélection d'un "cardlet" particulier A_i est obtenu à l'aide d'un "APDU" du type sélection ("SELECT"). Une fois ce choix effectué, les "APDU" qui le suivent sont acheminés vers ce "cardlet". Un "APDU SELECT" nouveau a pour effet d'abandonner l'application en cours et d'en choisir une autre. Le sous-ensemble logiciel gestionnaire des "APDU" 201 permet de choisir une application particulière A_i dans la carte à puce 2, de mémoriser l'application ainsi choisie, et de transmettre et/ou recevoir des "APDU" vers et depuis cette application.

En résumé de ce qui vient d'être décrit, la sélection d'une application A_j et le dialogue avec celle-ci s'effectue par échanges d'ordres "APDU". On suppose que les applications A_j sont des applications conventionnelles, que
5 l'on appellera ci-après "GCA" (pour "Generic Card Application" ou application de carte générique).

Dans un système d'applications à base de carte à puce, comme illustré par l'architecture de la figure 1B, cette dernière peut se voir dévolues diverses fonctions.

10 Cependant, il est à noter que la carte 3 ne peut communiquer directement avec les navigateurs du commerce, sauf à modifier le code de ces derniers. Les cartes à puce actuelles, qui par ailleurs sont conformes aux standards rappelés ci-dessus, ont une configuration matérielle et
15 logicielle qui ne permet pas non plus de communiquer directement avec le réseau Internet. En particulier, elles ne peuvent recevoir et transmettre des paquets de données, selon l'un ou l'autre des protocoles utilisés sur ce type de réseau. Il est donc nécessaire de prévoir une pièce de
20 logiciel additionnelle, implantée dans le terminal 1, généralement sous la forme de ce qui est appelé un "plug-in", selon la terminologie anglo-saxonne. Cette pièce de logiciel, qui porte la référence 12 sur la figure 1A, effectue l'interface entre le navigateur 10 et la carte 2,
25 plus précisément les circuits électroniques 20 de cette carte 2.

Dans l'état actuel de la technique, le système hôte associé au lecteur de carte 3, c'est-à-dire le terminal 1, est associé également à une application particulière. En
30 d'autres termes, il est nécessaire de prévoir un terminal spécifique, dit "dédié", pour chaque application particulière.

L'invention a donc notamment pour objet un procédé d'activation d'applications localisées dans une carte à puce
35 par un navigateur du type dit "WEB", permettant de palier

les inconvénients de l'art connu, dont certains viennent d'être rappelés.

Selon une caractéristique importante du procédé, la carte à puce présente au système hôte, c'est-à-dire le terminal, un modèle de terminal virtuel, par exemple sous la forme d'une page en langage "HTML" (pour "HyperText Markup Language"), ou plus généralement en langage hypertexte, ou encore sous la forme d'un "applet", en langage "JAVA", ce qui permet à l'utilisateur de choisir une application particulière parmi celles disponibles et proposées par la carte à puce. De ce fait, le terminal se trouve donc banalisé et supporte une pluralité d'applications. Le système hôte est vu comme un périphérique de la carte à puce, et il met à sa disposition des ressources matérielles, tels un écran de visualisation, un clavier, etc.

Pour ce faire, on prévoit une couche de logiciel de communication spécifique dans la carte à puce et son pendant dans le terminal. Le terme "spécifique" doit être entendu comme spécifique au procédé de l'invention. En effet, ces couches de communications, dites spécifiques, sont banalisées quelle que soit l'application considérée. Elles n'interviennent que dans le processus d'échange de données bidirectionnel entre la carte à puce et le terminal, d'une part, et la carte à puce et le réseau.

Les couches logicielles de communication spécifiques comprennent notamment des composants logiciels, dits "agents intelligents", permettant en particulier des conversions de protocoles. Il existe des agents appareillés dans les couches de communication spécifiques respectives associées au terminal et à la carte à puce. Selon le procédé de l'invention, il s'établit des sessions entre agents appareillés.

Selon une autre caractéristique importante, le procédé de l'invention rend possible l'activation d'applications de type conventionnel, c'est-à-dire du type

"CGA" précité, localisées dans une carte à puce, sans devoir les modifier en quoi que ce soit.

Pour ce faire, on prévoit un ou plusieurs agents intelligents dits traducteurs de script, qui reçoivent des requêtes d'un navigateur et les traduisent en ordres "APDU" 5
compréhensibles par l'application de type "CGA".

Selon une autre caractéristique importante, le procédé de l'invention rend possible la gestion d'applications d'un type non conventionnel, sans devoir 10
modifier l'architecture du système.

Selon une autre caractéristique importante, le procédé de l'invention permet un téléchargement dynamique de nouvelles applications dans la carte à puce, de type traditionnel ou non, ainsi que la mise à jour ou la 15
suppression d'une ou plusieurs applications.

L'invention a donc pour objet principal un procédé d'activation d'au moins une application localisée dans une carte à puce connectée à un terminal, dit local, muni d'un lecteur de ladite carte à puce, par l'intermédiaire d'un 20
navigateur de type "WEB", caractérisé en ce qu'il comprend au moins les phases suivantes :

a/ une première phase préliminaire consistant à implanter, dans ladite carte à puce, une première pièce de logiciel spécifique, formant interface avec au moins 25
lesdites applications, enregistrée dans la carte à puce ;

b/ une deuxième phase préliminaire consistant à implanter, dans le terminal, une seconde pièce de logiciel spécifique formant interface avec au moins 30
ledit navigateur "WEB" ;

- en ce que lesdites première et seconde pièces de logiciel spécifique comprennent en outre au moins une paire de premières entités logicielles appariées,

chacune desdites entités coopérant l'une avec l'autre de manière à permettre l'établissement d'une session d'échanges de données bidirectionnels entre ledit terminal et ladite carte à puce, de manière à ce que
5 ladite carte à puce offre la fonctionnalité d'un serveur "WEB" ;

- en ce qu'il est procédé à une troisième phase préliminaire consistant en l'implantation dans ladite carte à puce d'une deuxième entité logicielle apte à
10 interpréter une suite d'instructions et à la traduire en une suite d'ordres, ladite deuxième entité logicielle coopérant avec lesdites applications et ladite seconde pièce de logiciel spécifique ;

- et ce qu'il comprend au moins les étapes suivantes :

15 1/ établissement d'une session d'échanges de données entre ledit terminal et ladite carte à puce, par l'intermédiaire d'une desdites paires de premières entités logicielles appariées, de manière à transmettre une requête pour atteindre l'une desdites applications ;

20 2/ réception de ladite requête par l'une desdites deuxièmes entités logicielles, interprétation d'une desdites suites d'instructions, ladite suite d'instructions étant associée à ladite application à accéder, et transmission desdits ordres traduits à cette
25 application, de manière à l'activer ; et

3/ élaboration et transmission audit navigateur "WEB", par l'intermédiaire de ladite deuxième entité logicielle, et via lesdites première et seconde pièces de logiciel spécifique, d'une réponse à ladite requête.

30 L'invention sera mieux comprise et d'autres caractéristiques et avantages apparaîtront à la lecture de la description qui suit en référence aux figures annexées, parmi lesquelles :

- les figures 1A et 1B illustrent schématiquement les architectures matérielles et logiques, respectivement, d'un exemple de système d'application à base de carte à puce selon l'art connu, ;

5 - la figure 2 illustre schématiquement un exemple de système d'application à base de carte à puce selon l'invention, cette dernière agissant en tant que serveur "WEB" ;

10 - la figure 3 illustre un échange entre terminal et carte à puce sous forme d'une page de menu en langage "HTML" ;

- la figure 4 illustre de façon simplifiée l'architecture logique d'un système dans lequel la carte à puce comprend des agents intelligents ;

15 - les figures 5A à 5D illustrent des architectures de systèmes conformes à l'invention, selon plusieurs modes de réalisation, dans lesquels la carte à puce comprend des agents intelligents traducteurs de scripts ;

20 - la figure 6 illustre un exemple de formulaire transmis à la carte à puce par le terminal ; et

- la figure 7 illustre schématiquement les principales phase d'échanges entre un navigateur et une carte à puce selon le procédé de l'invention.

25 Avant de décrire le procédé d'activation d'applications localisées dans une carte à puce selon l'invention et de détailler une architecture pour sa mise en oeuvre, il apparaît tout d'abord intéressant de rappeler brièvement les caractéristiques principales des protocoles de communication sur les réseaux.

30 L'architecture des réseaux de communication est décrite par diverses couches. A titre d'exemple, le standard "OSI" ("Open System Interconnection"), défini par l' "ISO", comporte sept couches qui vont des couches dites basses (par exemple la couche dite "physique" qui concerne le support de

transmission physique) aux couches dites hautes (par exemple la couche dite d' "application"), en passant par des couches intermédiaire, notamment la couche dite de "transport". Une couche donnée offre ses services à la couche qui lui est
5 immédiatement supérieure et requiert de la couche qui lui immédiatement inférieure d'autres services, via des interfaces appropriées. Les couches communiquent à l'aide de primitives. Elles peuvent également communiquer avec des couches de même niveau. Dans certaines architectures, une ou
10 l'autre de ces couches peuvent être inexistantes.

Dans un environnement de type Internet, les couches sont au nombre de cinq, et de façon plus précise, en allant de la couche supérieure à la couche inférieure : la couche d'applications ("http", "ftp", "e-mail", etc.), la couche de
15 transport ("TCP"), la couche d'adressage de réseau ("IP"), la couche de liens de données ("PPP", "Slip", etc.) et la couche physique.

Ces rappels étant effectués, on va maintenant décrire une architecture de système d'application à base de
20 carte à puce autorisant celle-ci à agir comme un serveur "WEB". Un exemple d'une telle architecture est représenté schématiquement sur la figure 2. Les éléments communs à la figure 1 portent les mêmes références et ne seront re-décrits qu'en tant que de besoin. Pour simplifier le dessin,
25 on n'a pas représenté les divers périphériques connectés au terminal (figure 1 : écran 5 et clavier 6, par exemple).

A l'exception de couches logicielles de protocole de communication spécifiques, référencées 13 et 23a, respectivement implantées dans le terminal 1 et la carte à
30 puce 2a, les autres éléments, matériels ou logiciels, sont communs à l'art connu.

Le terminal 1 comprend des circuits 11 d'accès au réseau RI, constitués par exemple d'un modem pour le réseau Internet ou d'une carte réseau pour un réseau local. Ces
35 circuits regroupent les couches logicielles inférieures C1

et C₂, correspondant aux couches "physique" et de "lien de données".

On a également représenté les couches supérieures C₃ et C₄, correspondant aux couches "d'adressage de réseau" ("IP", dans le cas d'Internet) et de "transport" ("TCP"). La
5 couche supérieure d'application ("http", "ftp", "e-mail", etc.) n'a pas été représentée.

L'interface entre les couches inférieures, C₁ et C₂, et les couches supérieures, C₃ et C₄, est constituée par une
10 couche logicielle généralement appelée "driver couches basses". Les couches supérieures, C₃ et C₄, s'appuient sur cette interface et sont mises en oeuvre par l'intermédiaire de bibliothèques de fonctions spécifiques ou bibliothèques réseau 14, avec lesquelles elles correspondent. Dans le cas
15 du réseau Internet, "TCP/IP" est mis en oeuvre au moyen de bibliothèques dites de "sockets".

Cette organisation permet à un navigateur 10 (figure 1) de poser des requêtes vers un serveur 4 (figure 1), pour la consultation de pages "WEB" (protocole "HTTP"), pour le transfert de fichiers (protocole "FTP") ou
20 l'envoi de courrier électronique (protocole "e-mail"), ce de façon tout à fait classique.

Le terminal 1 comprend également un lecteur de carte 3, intégré ou non. Pour communiquer avec la carte à
25 puce 2a, le lecteur de carte englobe également deux couches basses, CC₁ (couche physique) et CC₂ (couche de lien de données), jouant un rôle similaire aux couches C₁ et C₂. Les interfaces logicielles avec les couches CC₁ et CC₂ sont décrites, par exemple, par la spécification "PC/SC"
30 ("part 6, service provider"). Les couches elles-mêmes, CC₁ et CC₂, sont notamment décrites par les normes ISO 7816-1 à 7816-4, comme il a été rappelé.

Une couche logicielle supplémentaire 16 forme interface entre les couches applicatives (non représentées)

et les couches inférieures, CC₁ et CC₂. La fonction principale dévolue à cette couche est une fonction de multiplexage/démultiplexage.

5 Les communications avec la carte à puce 2a s'effectuent selon un paradigme similaire à celui utilisé pour la manipulation de fichiers dans un système d'exploitation du type "UNIX" (marque déposée) : OUVRIER ("OPEN"), LIRE ("READ"), ECRIRE ("WRITE"), FERMER ("CLOSE"), etc.

10 Du côté de la carte à puce 2a, on retrouve une organisation similaire, à savoir la présence de deux couches basses, référencées CCA₁ (couche physique) et CCA₂ (couche de lien de données), ainsi qu'une couche d'interface 26a, tout à fait similaire à la couche 16.

15 Selon une première caractéristique importante, on prévoit, de part et d'autre, c'est-à-dire dans le terminal 1 et dans la carte à puce 2a, deux couches de protocoles spécifiques : 13 et 23a, respectivement.

20 Dans le terminal 1, la couche spécifique 13 s'interface aux "drivers couches basses" 15, aux bibliothèques 14 des couches réseau, C₃ et C₄, et aux couches protocolaires du lecteur de carte 3, c'est-à-dire les couches inférieures, CC₁ et CC₂, via la couche de multiplexage 16. La couche spécifique 13 permet le transfert
25 des paquets réseaux de et vers la carte à puce 2a. En outre, elle adapte les applications existantes telles le navigateur Internet 10 (figure 2), le courrier électronique, etc., pour des utilisations mettant en oeuvre la carte à puce 2a.

30 Du côté de la carte à puce 2a, on retrouve une organisation tout à fait similaire constituée par une instance supplémentaire de la couche spécifique, référencée 23a, pendant de la couche 13.

De façon plus précise, les couches spécifiques, 13 et 23a, sont subdivisées en trois éléments logiciels principaux :

- 5 - un module, 130 ou 230a, de transfert de blocs d'informations entre les couches 13 et 23a, via les couches conventionnelles CC₁, CC₂, CCA₁ et CCA₂ ;
- une ou plusieurs pièces de logiciel, dites "agents intelligents", 132 ou 232a, qui réalisent, par exemple, des fonctions de conversion de protocoles ;
- 10 - et un module de gestion de la configuration spécifique, 131 et 231a, respectivement ; module qui peut être assimilé à un agent intelligent particulier.

On retrouve donc, dans le terminal 1 et la carte à puce 2a, une pile protocolaire de communication entre les
15 deux entités.

Les couches de niveau deux (couches de lien de données), CC₂ et CCA₂, assurent l'échange entre la carte à puce 2a et le terminal 1. Ces couches sont responsables de la détection et l'éventuelle correction d'erreurs de
20 transmission. Différents protocoles sont utilisables, et à titre d'exemples non exhaustifs les suivants :

- la recommandation ETSI GSM 11.11 ;
- le protocole défini par la norme ISO 7816-3, en mode caractère T=0 ;
- 25 - le protocole défini par la norme ISO 7816-3, en mode bloc T=1 ;
- ou le protocole défini par la norme ISO 3309, en mode trame "HDLC" (pour "High-Level Data Link Control procedure" ou procédure de commande de liaison à haut
30 niveau).

Dans le cadre de l'invention, on utilisera de préférence le protocole ISO 7816-3, en mode bloc.

De façon connue en soi, à chaque couche de protocole il est associé un certain nombre de primitives qui permettent les échanges de données entre couches de même niveau et d'une couche à l'autre. A titre d'exemple, les primitives associées à la couche de niveau deux sont du type "demande de données" (*"Data.request"*) et "envoi de données" par la carte (*"Data.response"*), ainsi que "confirmation de données" (*"Data.confirm"*), etc.

De façon plus spécifique, les couches 13 et 23a sont chargées du dialogue entre la carte à puce 2a et l'hôte, c'est-à-dire le terminal 1. Ces couches permettent l'échange d'informations entre un utilisateur (non représenté) du terminal 1 et la carte à puce 2a, par exemple via des menus déroulants sous la forme d'hypertexte au format "HTML", comme il sera montré au regard de la figure 3. Elles permettent aussi la mise en place d'une configuration adaptée pour l'émission et/ou la réception de paquets de données.

Comme il a été indiqué ci-dessus, les couches comprennent trois entités distinctes.

La première couche, 130 ou 230a, est essentiellement constituée par un multiplexeur logiciel. Elle permet l'échange d'informations entre la carte à puce 2a et le terminal hôte 1, sous la forme d'unités de données de protocole. Elle joue un rôle similaire à celui d'un commutateur de paquets de données. Ces unités sont émises ou reçues via la couche de niveau 2 (couche de liens de données). Ce protocole particulier de communication permet de mettre en communications au moins une paire d' "agents intelligents". Le premier agent de chaque paire, 132, est situé dans la couche 13, côté terminal 1, le second, 232a, est situé dans la couche 23i, côté carte à puce 2a. Une liaison entre deux "agents intelligents" est associée à une session. Une session est un échange de données bidirectionnel entre ces deux agents.

Un agent intelligent peut réaliser tout ou partie de fonctions des couches de niveau trois et quatre, en fonction de la configuration mise en oeuvre par le terminal 1.

Un agent intelligent particulier est identifié
5 avantageusement par un nombre entier, par exemple sur 16 bits (nombre compris entre 0 et 6535). Cet identificateur est utilisé, par exemple, dans une unité de donnée de protocole constituant une référence de destination et une référence de source.

10 Il existe deux grandes catégories d'agents intelligents : les agents de type "serveur", qui sont identifiés par une référence fixe, et les agents de type "client", qui sont identifiés par une référence variable, délivrée par le module de gestion de configuration, 130 ou
15 230a.

Le processus d'ouverture d'une session est habituellement le suivant : un agent intelligent de type "client" ouvre la session vers un agent intelligent de type "serveur". Les couches 130 et 230a gèrent des tables (non
20 représentées) qui contiennent la liste des agents intelligents présents, côté terminal hôte 1 et carte à puce 2a.

Les agents intelligents sont associés à des propriétés ou attributs particuliers. Pour fixer les idées,
25 et à titre d'exemple non limitatif, les six propriétés suivantes sont associées aux agents intelligents :

- "hôte" : agent localisé dans le terminal ;
- "carte" : agent localisé dans la carte à puce ;
- "local" : agent ne communiquant pas avec le réseau ;
- 30 - "réseau" : agent communiquant avec le réseau ;
- "client" : agent qui initialise une session ;
- "serveur" : agent qui reçoit une demande de session.

Les agents intelligents permettent d'échanger des données (de l'hypertexte par exemple), mais également de déclencher des transactions réseau.

Les modules de gestion de configuration, 131
5 et 231a, respectivement, sont assimilables, comme il a été
indiqué, à des agents intelligents particuliers. Par
exemple, le module 131, côté terminal hôte 1, gère notamment
des informations relatives à la configuration de ce terminal
(modes de fonctionnement), liste des autres agents présents,
10 etc. Le module 231a, côté carte à puce 2a, a des fonctions
analogues. Ces deux agents peuvent être mis en communication
l'un avec l'autre pour établir une session.

Selon une caractéristique importante, la carte à
puce 2a propose au système hôte, c'est-à-dire au terminal 1,
15 un modèle de terminal virtuel. Pour ce faire, la carte à
puce 2a se comporte comme un serveur "WEB".

La carte à puce 2i est "adressée" par le
navigateur 10. Elle lui transmet alors une page de type
"WEB" en langage "HTML", un "applet" ou tout autre pièce de
20 logiciel. A titre d'exemple, la page "WEB" peut se présenter
sous la forme d'une page d'accueil donnant un choix
d'applications possibles et/ou d'hyperliens vers des
serveurs extérieurs.

De façon pratique, la carte à puce 2a est
25 avantageusement "adressée" par utilisation d'une adresse
"URL" (pour "Universal Resource Location") définissant un
rebouclage sur le terminal 1 lui-même, et non un pointage
sur un serveur externe. A titre d'exemple, la structure de
cet "URL" est habituellement la suivante :

30 `http://127.0.0.1:8080 (1),`

dans la quelle 127.0.0.1 est l'adresse "IP" de rebouclage
et 8080 est le numéro de port.

La figure 3 illustre schématiquement ce processus. On suppose sur cette figure, qu'en réponse à la requête du navigateur 10, la carte à puce 2a présente une page *P* en langage "HTML", page affichée par exemple sur l'organe de visualisation 5 du terminal 1. La page *P*, que l'on qualifiera d'accueil, peut afficher, de façon usuelle, différents éléments graphiques ou textes, mais comporte au moins un certain nombre d'hyperliens vers des serveurs externes, $H1_1$, $H1_2$, ..., $H1_i$, à $H1_n$, *i* et *n* étant des nombres arbitraires. *n* représente le nombre maximum de choix possibles. Il est fonction naturellement de la carte à puce 2a insérée dans le lecteur 3, et du nombre maximum *n* d'application A_i (figure 1B) présentes sur cette cartes. Les choix présentés peuvent dépendre des droits qui sont accordés au possesseur de la carte à puce 2a : abonnement souscrit à des services, niveau d'habilitation, etc. Le processus décrit utilise tout ou partie des couches de communication standards (non représentées), ainsi que les couches spécifiques, 13 et 23a.

Chaque hyperlien, par exemple l'hyperlien $H1_i$, pointe sur une ressource "URL" externe. A titre d'exemple, la structure de l' "URL" peut être la suivante :

`http://127.0.0.1:8081/www.NOM.com/index.html` (2),

dans laquelle 127.0.0.1 est l'adresse "IP" et 8081 est le numéro de port, "NOM.com" le nom d'un site Internet d'une société ou autre entité, conformément aux règles de nommage habituellement utilisées, et "index.html" la page d'accueil de ce site. En lieu et place du suffixe ".com", a priori utilisé pour des organisations à caractère commercial, il existe d'autres suffixes, tels ".fr", ".gov", etc., qui sont associés à la localisation du site Internet ou à la nature de l'organisme.

La figure 4 illustre de façon simplifiée l'architecture logique d'un système dont la carte à puce 2a comprend des agents intelligents, dont deux seulement ont

été représentés : un agent intelligent de type non
précisément défini 232a₁ et un agent intelligent 232a₁, de
type dit "WEB". La pile logique comprend, les couches de
protocole inférieures, référencées 200a, répondant aux
5 normes ISO 7816-3 (figure 2 : CCa₁ et CCa₂), le gestionnaire
de commandes "APDU" 201a₁, et le multiplexeur de
paquets 230a, ce dernier étant interfacé aux agents
intelligents, notamment l'agent intelligent "WEB" 232a₁.

Du côté terminal, il existe deux piles, l'une
10 communiquant avec le réseau, l'autre avec la carte à
puce 2a. La première pile comprend les organes 11
(figure 2 : C₁ et C₂) d'accès au réseau (normes OSI 1 et 2)
et les couches de protocole "TCP/IP" (figure 2 : C₃ et C₄),
référencées 100. Ces dernières couches sont interfacées avec
15 le navigateur "WEB" 10. L'autre pile comprend les couches de
protocole inférieures, référencées 101, répondant aux normes
ISO 7816-3 (figure 2 : C₁ et C₂), le gestionnaire 101
d'ordres "APDU" et le multiplexeur de paquets 102, ce
dernier étant interfacé avec des agents intelligents, dont
20 un seul 132, est représenté. Ce dernier, que l'on supposera
de "type réseau", peut en outre communiquer, d'une part avec
le navigateur 10, via les couches "TCP/IP" 100, d'autre part
avec le réseau Internet RI, via ces mêmes couches
"TCP/IP" 100 et l'organe 11, d'accès au réseau RI.

25 Le gestionnaire d'ordres "APDU" 201a est également
interfacé avec une ou plusieurs couches de niveau
applications, que l'on appellera simplement applications.
Ces applications sont, comme il a été indiqué, des
applications de type conventionnel, que l'on a appelé
30 "cardlet".

En résumé, la fonction "serveur WEB", fournie par la
carte à puce 2a, peut être réalisée par l'association de
l'agent intelligent "WEB" 232a₁ dans la carte à puce et de
l'agent réseau 132 dans le terminal 1.

La carte à puce 2a présente donc bien la fonctionnalité serveur "WEB". En outre, selon une caractéristique importante du procédé de l'invention, n'importe quelle application conventionnelle, A_1 à A_n , du type "CGA" précité, peut être activée au travers de ce serveur "WEB", soit par le navigateur "WEB" 10 présent dans le terminal 1, soit par un navigateur éloigné, localisé en un point quelconque du réseau Internet RI. Selon le procédé de l'invention, les applications, A_1 à A_n , ne nécessitent pas d'être ré-écrites et sont mises en oeuvre telles quelles.

Selon une autre caractéristique de l'invention, ces applications restent accessibles à un terminal de type conventionnel, c'est-à-dire conforme à l'art connu.

En d'autres termes, la compatibilité avec l'existant, en ce qui concerne le mode d'échanges avec les applications présentes sur la carte à puce, doit subsister. On doit toute fois remarquer que, dans ce cas, le terminal (conventionnel) n'est pas banalisé : il doit être adapté aux applications.

Pour répondre à ces exigences, la fonction serveur "WEB", offerte par la carte à puce, inclue un mécanisme similaire à la fonction dite "CGI" (pour "Common Gateway Interface") implantée dans les serveurs "WEB" classique.

Avant de décrire un exemple d'architecture conforme à l'invention, permettant de réaliser une fonction de ce type, au sein même de la carte à puce, il est utile de rappeler les principales caractéristiques d'un mode de fonctionnement "CGI".

Le "CGI" est une spécification de mise en oeuvre, depuis un serveur "WEB", d'applications écrites pour les systèmes d'exploitation "UNIX" (marque déposée), "DOS", ou "WINDOWS" (marque déposée). A titre d'exemple, pour le système d'exploitation "UNIX", la spécification est

"CGI 1.1" et pour le système d'exploitation "WINDOWS 95", la spécification est "CGI 1.3".

Toujours à titre d'exemple, une requête "HTTP" pour une adresse "URL", du type :

5 "http://www.host.com/cgi-bin/xxx.cgi" (3),

dans laquelle "host" se réfère à un système hôte (généralement éloigné), est interprétée par un serveur "WEB" comme l'exécution d'un script de commande, de type "CGI" nommé "xxx" et présent dans le répertoire "cgi-bin" de ce
10 système hôte. Bien que le nom du répertoire puisse être a priori quelconque, par convention, c'est le nom donné au répertoire stockant les scripts de type "CGI". Un script est une suite d'instructions du système d'exploitation du système hôte dont le résultat final est transmis au
15 navigateur "WEB" émetteur de la requête précitée. Différents langages peuvent être utilisés pour écrire ce script, par exemple le langage "PERL" (marque déposée).

De façon pratique, la requête est habituellement affichée sur un écran informatique sous la forme d'un
20 formulaire compris dans une page "HTLM". Le langage "HTLM" permet de traduire un formulaire en une adresse "URL". Le formulaire comporte un ou plusieurs champs, obligatoires ou non, qui sont remplis par un utilisateur à l'aide des moyens de saisie habituels : clavier pour le texte, souris pour les
25 cases à cocher ou les boutons dits "radio", etc. Le contenu du formulaire (ainsi qu'éventuellement des informations et instructions dites "cachées") est émis à destination du serveur "WEB". Le code "HTLM" de la page décrit la structure matérielle du formulaire (cadre, graphisme, couleur, et tout
30 autre attribut), ainsi que la structure des champs de données à saisir (nom, longueur, type de données, etc.).

La transmission peut s'effectuer selon deux types de formats principaux. Un premier format utilise la méthode dite "POST" et un second la méthode dite "GET". Une

information de type de format est présente dans le code de la page formulaire.

Ce mécanisme n'est cependant pas directement transposable à une carte à puce, même si celle-ci offre la
5 fonctionnalité serveur "WEB" conformément à l'une des caractéristiques de l'invention.

On va maintenant décrire un exemple d'architecture permettant d'activer une application quelconque, de type conventionnel, via un serveur "WEB" sur la carte à puce, par
10 référence à la figure 5A.

Lors d'une première étape, un utilisateur (non représenté) invoque depuis son navigateur "WEB" (figure 4 :10) une l'adresse "URL" qui peut se présenter de la façon suivante :

15 "http://@carte:8080/pme.html" (4),

dans laquelle "@carte" est une adresse IP de la carte à puce (par exemple l'adresse de rebouclage "127.0.01" décrite en regard de la figure 3 : voir formule (1)). Pour fixer les idées, et à titre d'exemple, "pme.html" est une page en
20 langage "HTML" relative à une application particulière offerte par la carte à puce, que l'on appellera ci-après "porte-monnaie électronique" ou "PME". Lors d'une deuxième étape, de la manière décrite précédemment, la carte à puce renvoie une page "HTML" qui peut se présenter sous la forme
25 illustrée par la figure 6.

Outre divers attributs graphiques (cadre, graphisme, etc.) et informations (titre, libellés, etc), le formulaire de l'exemple comporte deux champs numériques, Ch_1 et Ch_2 , intitulés "Certificat" et "Débit", ainsi que deux boutons
30 "radio", b_1 et b_2 , intitulés "Soumettre" et "RAZ", respectivement. Il comporte en outre, sur la droite de la figure, la fonction habituelle dite "d'ascenseur". Ce

formulaire s'affiche sur l'écran de visualisation (figure 1 : 5) du terminal 1.

Lors d'une troisième étape l'utilisateur renseigne les deux champs numériques, Ch_1 et Ch_2 . Dans cet exemple, les deux champs doivent être obligatoirement renseignés. En cliquant sur b_1 , l'utilisateur transmet le contenu du formulaire. En cliquant sur b_2 , il efface toutes les données affichées, soit pour les corriger, soit pour en saisir une nouvelle série.

Si l'utilisateur clique sur b_1 , les données sont émises et reçues par l'agent réseau 132. Il est rappelé que le navigateur "WEB" n'est pas obligatoirement situé dans le terminal local 1. Il peut être indifféremment dans celui-ci ou localisé dans un système quelconque connecté au réseau Internet RI (voir figure 4, par exemple). Les données traversent alors le multiplexeur de paquets 130 (qui constitue l'un des composants de la couche spécifique 13, côté terminal 1), le gestionnaire d'ordres "APDU" 102, les couches protocolaires 101, pour être transmises à la carte à puce 2a. Elles traversent ensuite les couches protocolaires 200a, le gestionnaire d'ordres "APDU" 201a, le multiplexeur de paquets 230a pour être reçues par l'agent "WEB" 232a₁. Il s'établit donc une session logique entre les deux agents intelligents, comme explicités précédemment.

Il convient de remarquer que les données adressées à l'agent "WEB" 232a₁ sont transportées, de façon conventionnelle en soi, sous formes d'ordres "APDU" destinés à l'application particulière "Multiplexeur de paquets". Le gestionnaire d'ordres "APDU" 201a sélectionne cette application de manière tout à fait similaire aux autres applications de type "CGA" présentes dans la carte à puce 2a, référencées A_1 à A_n . En d'autres termes, le multiplexeur de paquets 230a est vu par le gestionnaire d'ordres "APDU" 201a comme une application "CGA" ordinaire.

La requête "HTTP" est alors analysée par l'agent "WEB" 232a₁ qui détecte une référence à un répertoire particulier, que l'on appellera ci-après par convention "cgi-smart", d'une part, et à une application particulière, par exemple "pme" dans le cas de l'exemple décrit. Le chemin complet est donc, en l'occurrence "cgi-smart/pme".

Selon une caractéristique importante du procédé de l'invention, l'entité ci-dessus désigne un script particulier associé une application également particulière ("PME" en l'occurrence).

Lors d'une quatrième étape, le script est alors interprété par un agent intelligent dit "Agent traducteur de script", que l'on appellera "ATS" ci-après. Cette traduction peut être réalisée de différentes manières :

- a/ par l'agent "WEB" 232a₁ lui-même, qui est doté dans ce cas d'une double capacité ;
- b/ par un agent script unique capable de traduire l'ensemble des scripts présents dans la carte à puce 2a ;
- c/ par un agent de script dédié que l'on appellera "ATSD" ci-après (un par script) ; ou
- d/ par un agent "APDU" 2010a du gestionnaire d'ordres "APDU" 201a, qui est doté, dans ce cas, d'une double capacité.

L'agent "APDU" 2010a est une composante de la couche gestionnaire d'ordres "APDU" 201a. Cette dernière, comme il a été indiqué, est une couche capable de centraliser tous les ordres "APDU" émis et/ou reçus par le système, de sélectionner des applications, parmi A_1 à A_n , mais également d'offrir une interface de type agent intelligent. Elle est donc capable, selon l'une des caractéristiques du procédé de communiquer avec tous les agents intelligents du système (via des sessions), que ces agents soient localisés dans le terminal 1 ou la carte à puce 2a.

Dans le cas c/ ci-dessus, une session est ouverte entre l'agent "WEB" 232a₁ et l'un des agents "ATSD".

La figure 5A illustre un exemple d'architecture pour laquelle les agents traducteurs sont du type "ATSD". Ils sont référencés ATS_1 à ATS_n et associés aux applications A_1 à A_n . L'application sélectionnée étant supposée être l'application A_i , la session s'établit entre l'agent "WEB" 232a₁ et l'agent ATS_i .

Un agent traducteur de script génère une suite d'ordres "APDU". Une session est ouverte entre l'agent traducteur, par exemple l'agent ATS_i , et l'agent "APDU" 2010a. Les ordres sont alors émis vers l'agent "APDU" 2010a. Le gestionnaire d'ordres "APDU" 201a sélectionne l'application "CGA" A_i (par exemple l'application "PME") et lui transmet les ordres "APDU", ordres traduits et donc conventionnels, qu'elle est en mesure de comprendre. Cette application est donc correctement activée, sans avoir à la modifier ou à la réécrire.

Les réponses de l'application "CGA" A_i sont transmises au gestionnaire d'ordres "APDU" 201a, à l'agent "APDU" 2010a, puis de nouveau à l'agent ATS_i (et de façon plus générale à l'agent traducteur de script).

En fonction de la réussite ou de l'échec du déroulement du script, l'agent traducteur de script, par exemple l'agent ATS_i dans l'exemple de la figure 5A, élabore une page en langage "HTML" et la transmet via les différentes couches empruntées par la requête initiale, mais en sens inverse, ce pour être présentée sur l'écran de visualisation 5 (figure 1).

Les différents cheminements sont représentés symboliquement sur la figure 5A par des traits pleins reliant les blocs fonctionnels ou en pointillés à l'intérieur de ces blocs.

Selon un autre aspect du procédé de l'invention, la carte à puce 2a doit pouvoir continuer à être utilisée en conjonction avec un terminal conventionnel. La carte à puce ne joue plus alors le rôle de serveur "WEB" et les fonctionnalités "agents intelligents" ne sont pas utilisées. Il en est de même de la couche spécifique 23a (figure 2). Ces différents composants restent "transparents" vis-à-vis des échanges entre terminal et carte à puce. Ces échanges s'effectuent de façon purement conventionnelle, par échange d'ordres "ADPU" entre une application localisée dans le terminal et une application localisée dans la carte à puce, comme il a été explicité en regard des figures 1A et 1B.

La figure 7 résume de façon schématique les principales étapes du processus qui vient d'être décrit :

- a/ transmission via le réseau Internet RI (ou à partir du terminal local : dans les deux cas à l'aide d'un navigateur conventionnel 10) d'une requête "HTTP", référencée RQ ;
- b/ réponse du serveur "WEB" de la carte à puce 2a, sous la forme d'un formulaire, référencé FO ;
- c/ transmission du formulaire rempli, sous la forme d'une nouvelle requête RQ ; et
- d/ réponse sous la forme d'une page "HTLM", référencée PR.

La réponse pourrait d'ailleurs consister en la transmission d'un fichier, ou d'une pièce de logiciel ou "Applet".

Selon une autre caractéristique importante du procédé selon l'invention, et dans une variante de mise en oeuvre supplémentaire, le système peut accommoder des applications non conventionnelles. On entend par là des applications qui n'ont pas besoin d'échanges d'ordres "ADPU". Ce type d'application, ne répondant pas aux normes ISO précitées n'est pas accessible par des terminaux classiques, mais l'est, sans nécessiter aucune modification

supplémentaire de leur architecture, par des terminaux conformes à l'invention, c'est-à-dire comportant la couche spécifique 13.

5 La figure 5B illustre schématiquement un exemple d'une architecture conforme à cette variante supplémentaire. Les applications ont été référencées A'1 à A'n.

10 Les premières étapes du procédé selon cette variante supplémentaire sont similaires sinon identiques aux phases précédemment décrites en regard de la figure 5A. Par contre la traduction du script est réalisée par un agent intelligent spécifique que l'on appellera "Agent Traducteur de Script Dédié et Autonome" ou "ATSDA". On doit bien comprendre que, comme précédemment, cet agent peut être, soit l'agent "WEB" 232a1, doté d'une double capacité, soit
15 un agent "ATSDA", unique pour tout le système, soit l'agent "APDU" 2010a, doté d'une double capacité, soit encore, comme illustré sur la figure 5B, une pluralité d'agents "ATSDA", référencés ATS'1 à ATS'n. Dans ce cas, chaque agent "ATSDA" est associé à une seule application.

20 Comme il est aisé de constater sur la figure 5B, les échanges de données entre le terminal 1 et la carte à puce 2a s'effectuent directement entre les couches spécifiques 13 et 23a. il n'y a donc plus de liaisons entre les applications et un gestionnaire d'ordres "APDU".
25 Cependant, dans une variante supplémentaire, ce dernier peut subsister, ainsi que les couches protocolaires conventionnelles. Il en est de même en ce qui concerne le terminal 1. Ces composants ont été représentés en traits pointillés. Cette disposition présente plusieurs avantages.
30 D'une part, le terminal 1 ne subit aucune modification d'architecture et il peut communiquer indifféremment avec des cartes à puce comportant des applications non conventionnelles ou avec des cartes à puce entièrement conventionnelles. D'autre part, si la carte à puce 2a
35 comprend elle-même les couches protocolaires

conventionnelles, on peut y stocker, à la fois des applications conventionnelles, de type "CGA", accessibles selon le procédé explicité en regard de la figure 5A ou par un terminal conventionnel, et des applications non conventionnelles, accessibles uniquement par la variante du procédé qui vient d'être explicitée.

Selon une autre caractéristique importante du procédé selon l'invention, et dans une variante de mise en oeuvre supplémentaire, illustrée schématiquement par la figure 5C, il est possible de télécharger dynamiquement de nouvelles applications dans la carte à puce 2a. Il est également possible de mettre à jour les applications existantes ou d'en supprimer, ce toujours de façon dynamique, ce que ne permettent pas les cartes à puce de l'art connu. Pour ces dernières, il est nécessaire de modifier le terminal pour tenir compte des modifications effectuées dans la carte à puce.

Sur la figure 5C, on n'a représenté que les composants de la partie carte à puce 2a, étant entendu que le terminal est doté obligatoirement d'une architecture conforme à l'invention (voir figure 2, par exemple) pour pouvoir bénéficier de ces fonctionnalités avantageuses. Il est en effet nécessaire d'utiliser la fonction serveur "WEB" offerte par la carte à puce 2a, selon une des caractéristiques du procédé de l'invention. Pour ce faire, le terminal doit donc comporter la couche spécifique 13 (figure 2).

L'architecture générale de la carte à puce 2a est similaire à celle décrite en regard de la figure 5A, et les composants communs ne seront re-décrits qu'en tant que de besoin.

Le problème posé est de pouvoir télécharger et enregistrer dans la carte à puce 2a, une ou plusieurs applications supplémentaires. Pour fixer les idées, on supposera que l'on veuille ajouter l'application Aj, le

nombre total d'applications présentes dans la carte à puce 2a devenant égal à $n+1$. On suppose que les applications présentes, ainsi que la nouvelle application, sont de type conventionnel ("CGA"), mais des applications non conventionnelles pourraient tout aussi bien être téléchargées dans la carte à puce 2a par le même procédé.

Dans l'état de l'art, il est habituel de prévoir une pièce de logicielle 6, connue sous l'appellation anglo-saxonne "loader" et que l'on appellera ci-après "téléchargeur". La carte à puce 2a est placée dans un dispositif électronique externe approprié, que l'on a référencé *EXT*, qui permet de télécharger une nouvelle application et de l'enregistrer.

Selon le procédé de l'invention, on prévoit en outre une pièce de logiciel 7, dont la fonction est de surveiller les téléchargements. On doit bien comprendre que les fonctions "téléchargeur" 6 et "surveillance de téléchargement" 7 peuvent être confondues, en modifiant la pièce de logiciel 6.

La pièce de logiciel supplémentaire 7, "avertit" le gestionnaire d'ordres "APDU" lorsqu'elle détecte le téléchargement d'une nouvelle application, l'application A_j en l'occurrence. Celui-ci enregistre alors cette modification. Il pourra donc désormais adresser l'application A_j . De même la couche spécifique est mise à jour, notamment le composant 231a, gestionnaire de configuration de la carte à puce 2a. Il est en outre nécessaire de mettre à jour l'agent traducteur "ATS", "ATSD", ou "ATSDA", selon les cas, s'il s'agit d'un agent unique, ou d'en créer un nouveau ATS_j , associé à la nouvelle application A_j . Les informations nécessaires à cette mise à jour sont téléchargées en même temps que la nouvelle application A_j .

Les mises à jour ou les suppressions d'applications peuvent s'effectuer selon le même mécanisme.

Lorsque la carte à puce 2a est de nouveau connectée à un terminal conforme à l'invention, celui-ci pourra être averti des nouvelles possibilités de la carte à puce ou des modifications, par l'envoi d'une page "HTLM", de façon
5 similaire à ce qui a été décrit en regard de la figure 5A.

Dans une variante supplémentaire encore, illustrée schématiquement par la figure 5D, on prévoit une application spécifique dans la carte à puce 2, que l'on référencera par convention A_0 , associée à un traducteur de script dédié
10 ATS_0 . Cependant, la traduction peut également être effectuée de façon centralisée, comme précédemment indiquée (agent "WEB", etc.). A cette application A_0 , on attribue le rôle de téléchargement et de surveillance des applications téléchargées, modifiées ou supprimées. Lorsque la carte à
15 puce 2a est connectée à un terminal 1 selon l'invention, l'agent traducteur ATS_0 , interagissant avec l'application A_0 , génère une page $PMAJ$, que l'on appellera page de mise à jour. Celle-ci peut se présenter sous la forme d'un menu indiquant les applications présentes dans la
20 carte à puce 2a, avec éventuellement des informations sur leur version et/ou la date de téléchargement ou de modification, menu transmis de façon automatique ou sur requête du navigateur 10 du terminal 1.

Dans une variante préférée, le téléchargement, la
25 mise à jour ou la suppression d'applications peut s'effectuer directement à partir du navigateur 10, en utilisant la fonction serveur "WEB" de la carte à puce 2a. Les données nécessaires sont transmises par échanges entre le terminal 1 et de la carte à puce 2a, selon le processus
30 habituel. L'application A_0 se charge alors de mettre à jour les informations de configuration de la carte à puce 2a, notamment en interagissant sur le gestionnaire d'ordres "APDU" (non représenté sur la figure 5D), ainsi éventuellement que de mettre à jour ou de créer le
35 traducteur de script associé à une nouvelle application, ou

à une application modifiée. Le gestionnaire d'ordres "APDU" peut ensuite adresser la nouvelle application téléchargée.

L'application A_0 et son traducteur de script, centralisé ou non (ATS_0), constitue donc un service complet
5 de mise à jour des applications, service supplémentaire offert par la carte à puce 2a. Il doit être clair que ce service de mise à jour reste compatible avec l'existant. D'une part, les cartes à puce 2a peuvent continuer à être téléchargée à partir d'un dispositif extérieur *EXT*, d'autre
10 part, les applications téléchargées ou mises à jour par l'intermédiaire du navigateur 10, peuvent être d'un type conventionnel "CGA" ou non conventionnel. Dans le premier cas, elles restent compatibles avec les terminaux de l'art connu.

15 A la lecture de ce qui précède, on constate aisément que l'invention atteint bien les buts qu'elle s'est fixés.

Notamment, tout en restant compatible avec les applications et normes en vigueur, elle apporte de nombreuses possibilités supplémentaires. Elle permet
20 d'accepter également des applications d'un type nouveau, que l'on pourra appeler "natives", sans nécessiter de changements de l'architecture du terminal communiquant avec la carte à puce, à l'exception des modifications nécessaires pour qu'il devienne conforme à l'invention (couche
25 spécifique).

Il doit être clair cependant que l'invention n'est pas limitée aux seuls exemples de réalisations explicitement décrits, notamment en relation avec les figures 2 à 7.

L'invention s'applique notamment à un service porte-
30 monnaie électronique, mais ne saurait en aucun cas être limitée à cette seule application.

REVENDEICATIONS

1. Procédé d'activation d'au moins une application localisée dans une carte à puce connectée à un terminal, dit local, muni d'un lecteur de ladite carte à puce, par l'intermédiaire d'un navigateur de type "WEB", caractérisé en ce qu'il comprend au moins les phases suivantes :

a/ une première phase préliminaire consistant à implanter, dans ladite carte à puce (2a), une première pièce de logiciel spécifique (23a), formant interface avec au moins lesdites applications (A_1-A_n), enregistrée dans la carte à puce (2a) ;

b/ une deuxième phase préliminaire consistant à implanter, dans le terminal (1), une seconde pièce de logiciel spécifique (13) formant interface avec au moins ledit navigateur "WEB" (10) ;

- en ce que lesdites première et seconde pièces de logiciel spécifique (13, 23a) comprennent en outre au moins une paire de premières entités logicielles appariées (132, 232a), chacune desdites entités (132, 232a) coopérant l'une avec l'autre de manière à permettre l'établissement d'une session d'échanges de données bidirectionnels entre ledit terminal (1) et ladite carte à puce (2a), de manière à ce que ladite carte à puce (2a) offre la fonctionnalité d'un serveur "WEB" ;

- en ce qu'il est procédé à une troisième phase initiale consistant en l'implantation dans ladite carte à puce (2a) d'une deuxième entité logicielle (ATS_1-ATS_n), apte à interpréter une suite d'instructions et à la traduire en une suite d'ordres, ladite deuxième entité logicielle (ATS_1-ATS_n) coopérant avec lesdites applications (A_1-A_n) et ladite seconde pièce de logiciel spécifique (23a) ;

- et ce qu'il comprend au moins les étapes suivantes :

1/ établissement d'une session d'échanges de données entre ledit terminal (1) et ladite carte à puce (2a), par l'intermédiaire d'une desdites paires de premières entités logicielles appariées (132, 232a₁), de manière à transmettre une requête pour atteindre l'une desdites applications (A_i) ;

5 2/ réception de ladite requête par l'une desdites deuxièmes entités logicielles (ATS_i), interprétation d'une desdites suites d'instructions, ladite suite d'instructions étant associée à ladite application à accéder (A_i), et transmission desdits ordres traduits à cette application, de manière à l'activer ; et

10 3/ élaboration et transmission audit navigateur "WEB" (10), par l'intermédiaire de ladite deuxième entité logicielle (ATS_i), et via lesdites première (13) et seconde (23a) pièces de logiciel spécifique, d'une réponse à ladite requête.

15

2. Procédé selon la revendication 1, caractérisé en ce que ledit lecteur de carte à puce (3) et ladite carte à puce (2a) comprennent des première et deuxième piles protocolaires pour lesdites transmissions de données, chacune comprenant au moins des couches de communication logicielles (101, 200a), dites basses, de manière à

20 permettre lesdits échanges de données entre ladite carte à puce (2a) et ledit terminal (1), ces couches formant interface avec lesdites première (13) et seconde (23a) pièces de logiciel spécifique, respectivement, et en ce que ces pièces de logiciel spécifique (13, 23a)

25 comprennent chacune deux entités supplémentaires constituées d'un module de transfert de données (130, 230a), formant interface avec lesdites couches basses (101, 200a) des première et deuxième piles protocolaires, et d'un module de gestion (131, 231a), et

30 en ce que lesdites premières entités de chaque paire sont

35

constituées de modules logiciels, dits agents intelligents (132, 232a₁) établissant lesdites sessions.

5 3. Procédé selon la revendication 2, caractérisé en ce que, ladite suite d'instructions à interpréter étant constituée par un script, chacune desdites deuxièmes entités logicielles est constituée par un module logiciel dit agent intelligent traducteur de script (ATS_1-ATS_n).

10 4. Procédé selon la revendication 3, caractérisé en ce qu'il est prévu un seul agent traducteur de script, commun à toutes lesdites applications.

 5. Procédé selon la revendication 4, caractérisé en ce que ledit agent traducteur de script est confondu avec ledit agent intelligent (232a₁) établissant lesdites sessions.

15 6. Procédé selon la revendication 3, caractérisé en ce qu'il est prévu une pluralité d'agents traducteurs de script dits dédiés (ATS_1-ATS_n), chaque agent traducteur de script dédié étant associé à une seule desdites applications (A_1-A_n).

20 7. Procédé selon la revendication 3, caractérisé en ce que lesdites applications sont d'un type devant être activé à l'aide d'un jeu d'ordres dits "ADPU", définis par la norme ISO 7816, ledit terminal (1) et ladite carte à puce (2a) comprennent une entité logicielle supplémentaire (102, 201a), dite gestionnaire d'ordres "ADPU", en ce que
25 ladite interface entre lesdites couches basses (101, 200a) des première (101) et deuxième (200a) piles protocolaires, d'une part, et lesdites première (13) et seconde (23a) pièces de logiciel spécifique (23a), d'autre part,

s'effectue par l'intermédiaire desdits gestionnaires d'ordres (102, 200a).

5 8. Procédé selon la revendication 7, caractérisé en ce que ledit gestionnaire d'ordres (201a), présent dans la carte à puce (2a), enregistre notamment des données décrivant la configuration desdites applications (A_1-A_n), et en que le procédé comprend des étapes consistant, sur réception de ladite requête d'accès à une (A_i) des dites applications de la carte à puce (2a), en la sélection, par 10 ledit gestionnaire d'ordres (201a), à partir des dites données de configuration, de ladite application à accéder (A_i) et d'un desdits agents intelligents traducteurs de script (ATS_i), associé à cette application (A_i), en la réception desdits ordres traduits par l'agent intelligent traducteur de script (ATS_i) et à l'envoi de ces ordres à 15 ladite application sélectionnée (A_i).

20 9. Procédé selon la revendication 8, caractérisé en ce que ledit gestionnaire d'ordres (201a) de la carte à puce (2a) comprend en outre une entité logicielle du type agent intelligent (2010a) permettant l'établissement d'une session d'échanges de données bidirectionnels avec d'autres agents intelligents, cet agent (2010a) étant dit agent intelligent gestionnaire d'ordres, et en ce que ladite carte à puce (2a) comporte un seul agent 25 intelligent traducteur de script confondu avec ledit agent intelligent gestionnaire d'ordres (2010a).

30 10. Procédé selon la revendication 8, caractérisé en ce que, la carte à puce (2a) comprenant en outre une entité logicielle (6) permettant de télécharger et d'enregistrer dans celle-ci au moins une application supplémentaire (A_j) à partir d'un organe extérieur (EXT), il est prévu une entité logicielle (7) de détection et de

surveillance dudit téléchargement, et en ce que, lorsqu'un téléchargement d'une application supplémentaire (Aj) est détecté, il est procédé à une étape de mise à jour desdites données de configuration, et à la création ou à la mise à jour d'un agent intelligent de traduction de script (ATSj) associé à ladite application supplémentaire (Aj), et à une étape ultérieure de transmission audit terminal (1) de données reflétant la mise à jour de la carte à puce, par utilisation de ladite fonctionnalité serveur "WEB" de ladite carte à puce (2a).

11. Procédé selon la revendication 8, caractérisé en ce qu'il comprend une phase préliminaire supplémentaire consistant à implanter, dans ladite carte à puce (2a), une application spécifique (A0) et un agent intelligent traducteur de script (ATS0) qui lui est associé, destinés à offrir, via ladite fonctionnalité de serveur "WEB" un service dit de mise à jour de la carte à puce (2a), et en ce que le procédé comprend au moins les étapes suivantes :

1/ sous la commande dudit navigateur "WEB" (10), établissement d'une session d'échanges de données entre ledit terminal (1) et ladite carte à puce (2a), par l'intermédiaire d'une desdites paires d'agents intelligents (132, 232a1) formant les premières entités logicielles appariées, de manière à transmettre une requête d'accès à ladite application spécifique (A0) et l'envoi de données provoquant une action déterminée consistant dans le chargement d'une application supplémentaire, la mise à jour d'une desdites applications ou sa suppression ;

2/ réception de ladite requête par ledit agent intelligent traducteur (ATS0), interprétation d'une suite d'instructions associée à ladite application spécifique (A0) et transmission desdits ordres traduits à cette application (A0), de manière à

l'activer, à réaliser ladite action déterminée et à mettre à jour lesdites données de configuration ; et 3/ élaboration et transmission audit navigateur "WEB" (10), par l'intermédiaire dudit agent traducteur intelligent de script (ATS₀), et via lesdites première (13) et seconde (23a) pièces de logiciel spécifique, d'une réponse à ladite requête consistant en des données reflétant la nouvelle configuration de la carte à puce (2a).

10 **12.** Procédé selon la revendication 1, caractérisé en ce que ladite requête d'accès à une desdites applications est constituée par une page en langage "HTLM" comprenant un formulaire, en ce que ledit formulaire comporte au moins un champ de données (Ch₁, Ch₂) à remplir et/ou une
15 case à cocher par un utilisateur, via ledit navigateur "WEB" (10), de manière à ce que les données associées soient transmises à ladite carte à puce (2a) lorsque ledit utilisateur effectue une action de validation prédéterminée.

20 **13.** Procédé selon la revendication 1, caractérisé en ce que lesdites réponses consistent en l'envoi audit navigateur "WEB" (10) d'une page en langage "HTLM".

14. Procédé selon la revendication 1, caractérisé en ce que ledit navigateur "WEB" (10) est localisé dans ledit
25 terminal local (1).

15. Procédé selon la revendication 1, caractérisé en ce que ledit réseau de transmission de données est un réseau de type Internet (RI), en ce que réseau (RI) est décrit par une troisième pile protocolaire comprenant un
30 nombre déterminé de couches logicielles de communication (C₁-C₄), et en ce que ladite première pièce de logiciel

spécifique (13) forme interface avec lesdites couches basses (CC₂-CC₁) de la première pile protocolaire et avec des couches déterminées de ladite troisième pile protocolaire (C₂, C₃).

- 5 **16.** Procédé selon la revendication 15, caractérisé en ce que ledit navigateur "WEB" (10) est localisé dans un terminal éloigné (4), connecté audit réseau de type Internet (RI).

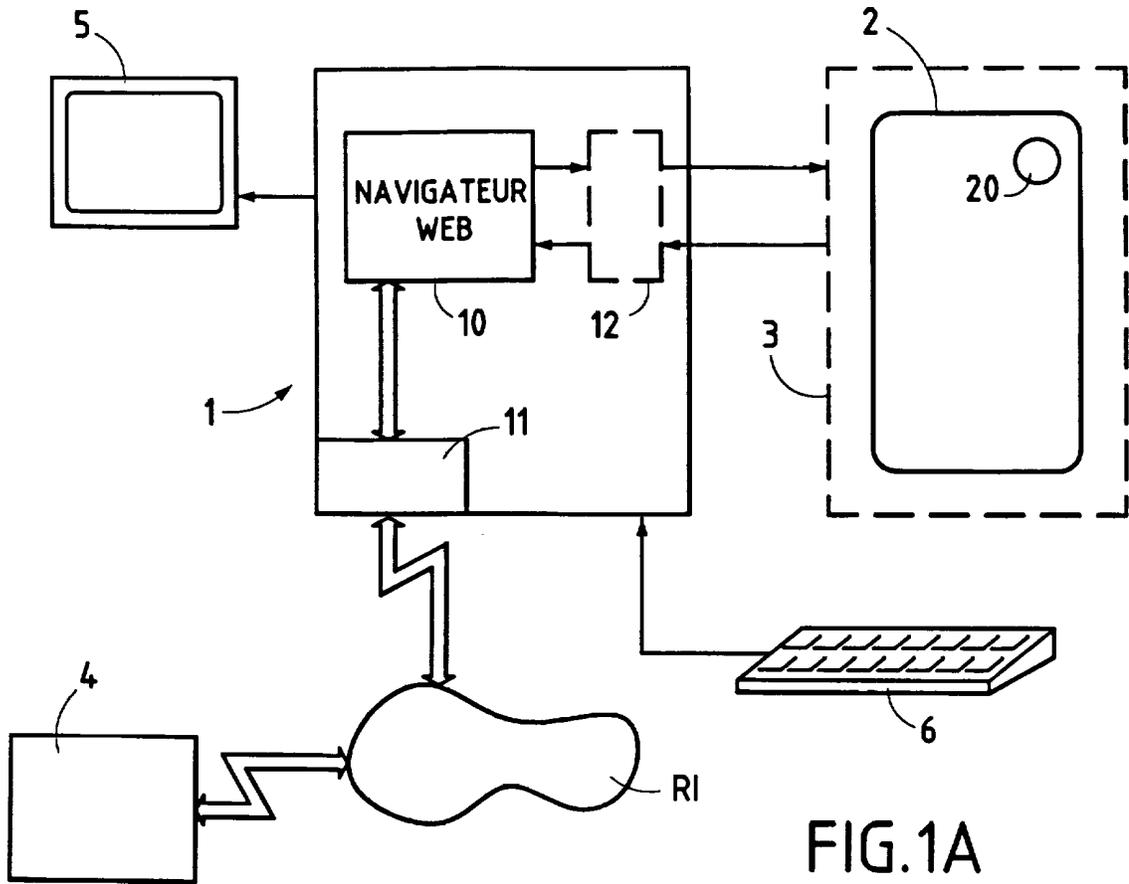


FIG. 1A

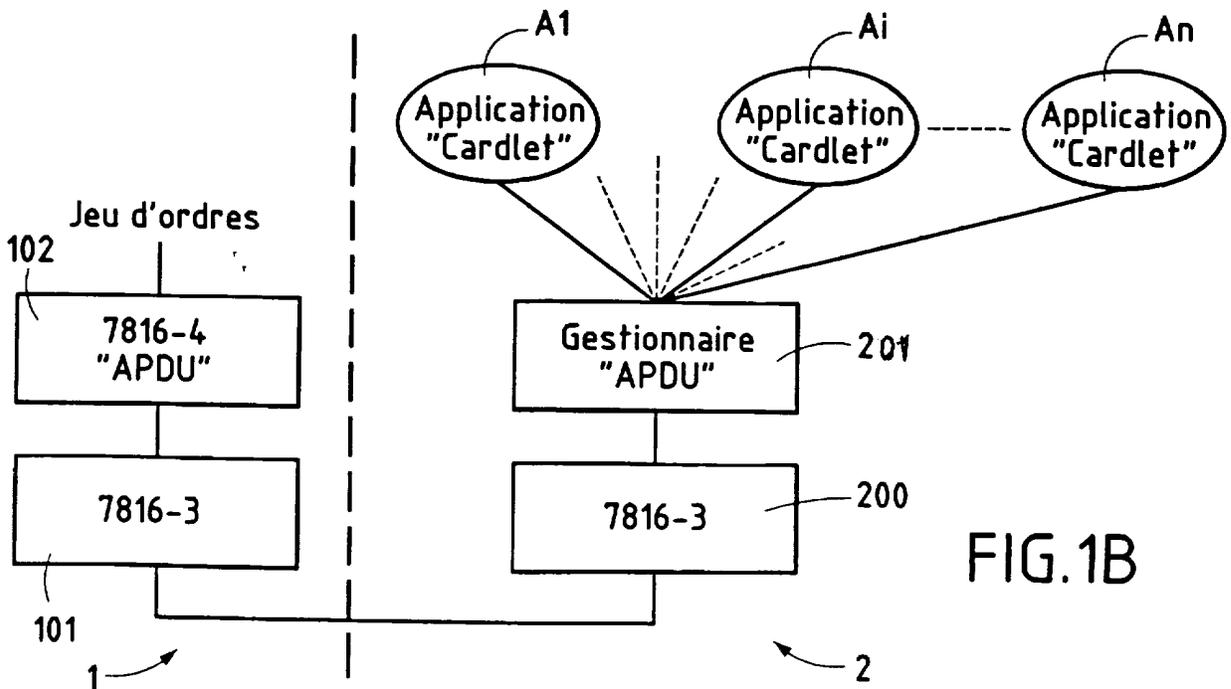


FIG. 1B

FIG. 2

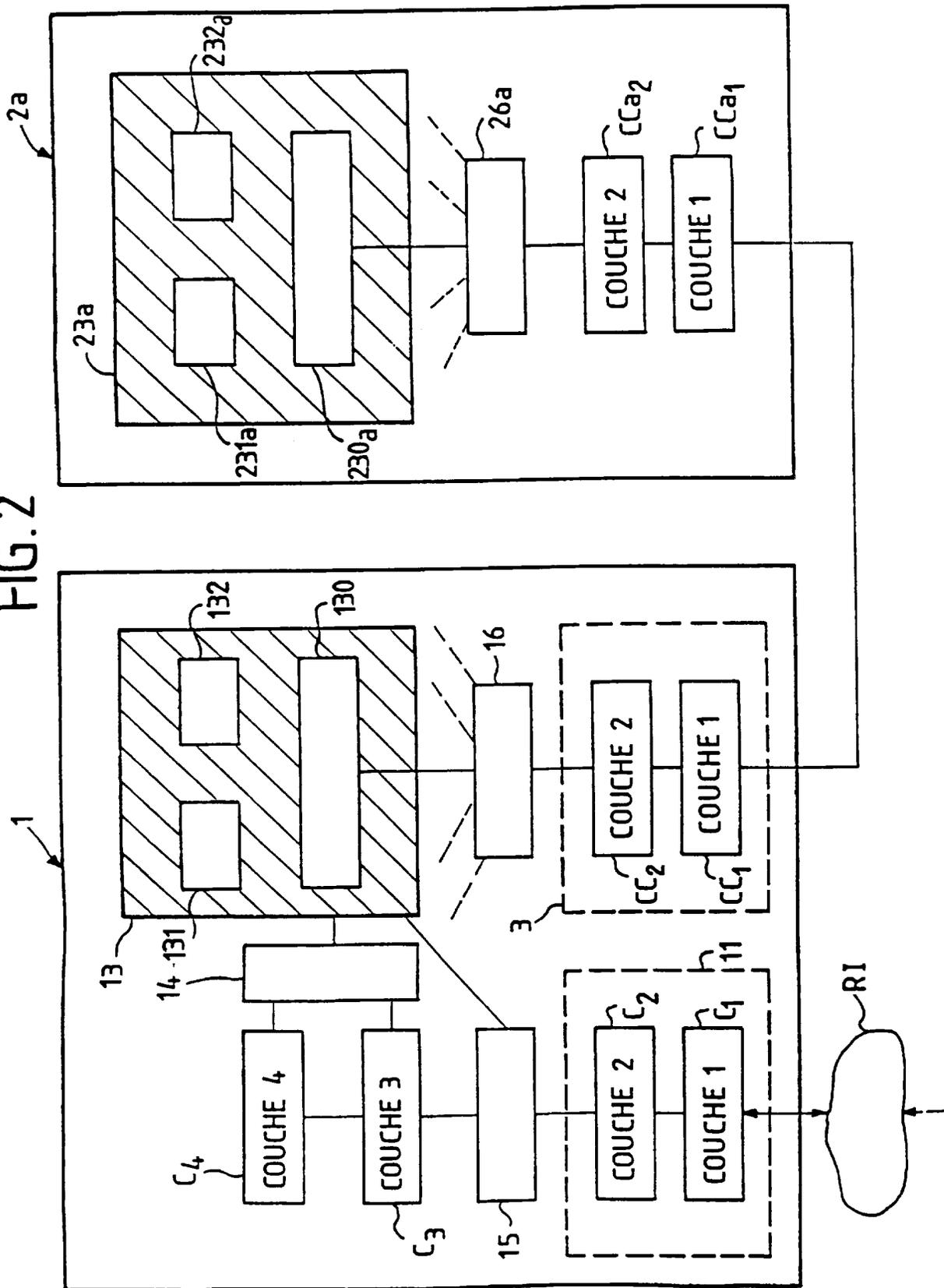


FIG.3

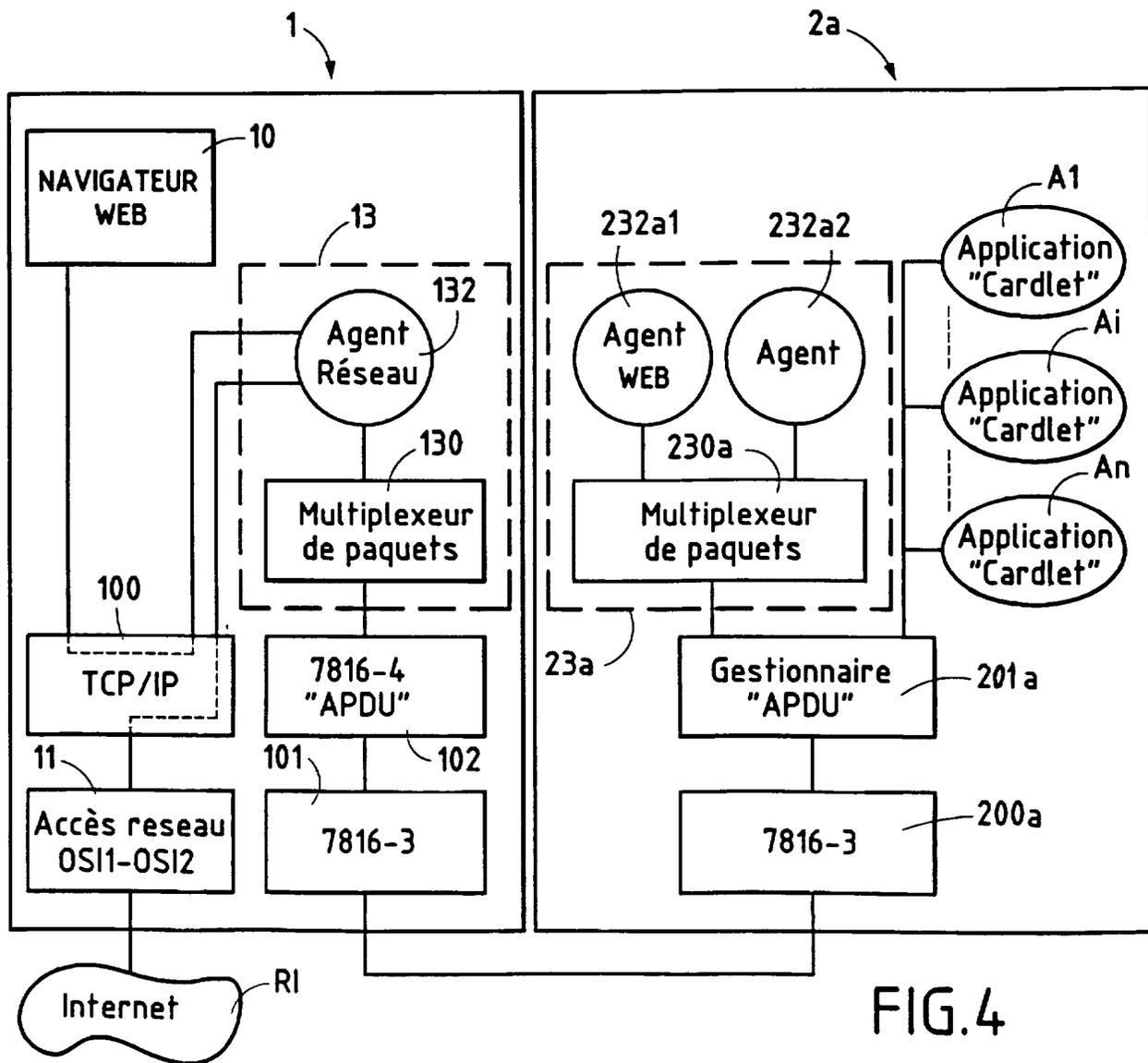
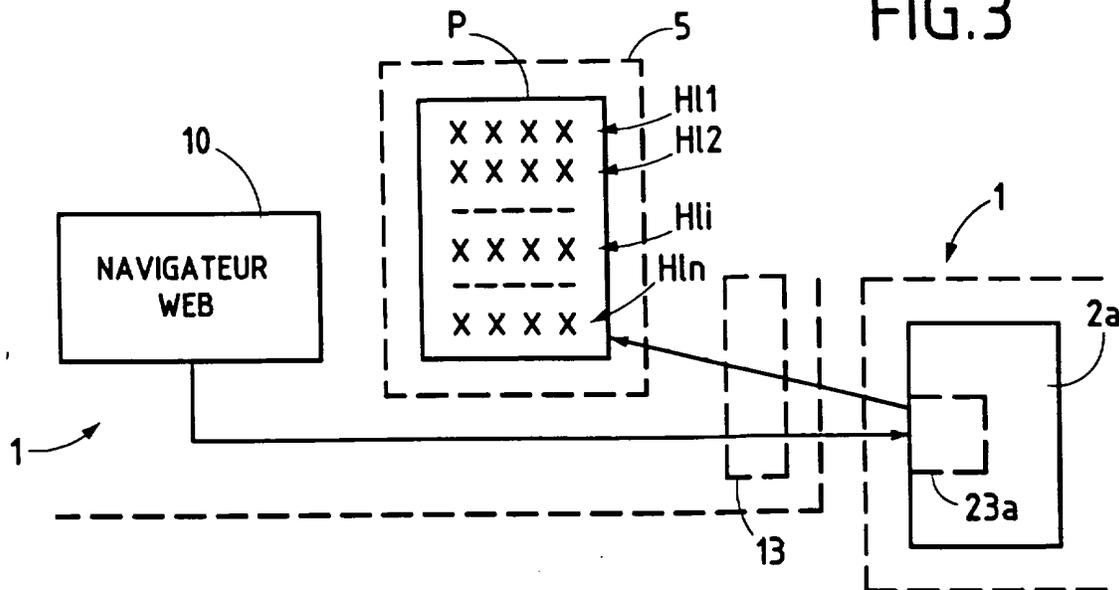


FIG.4

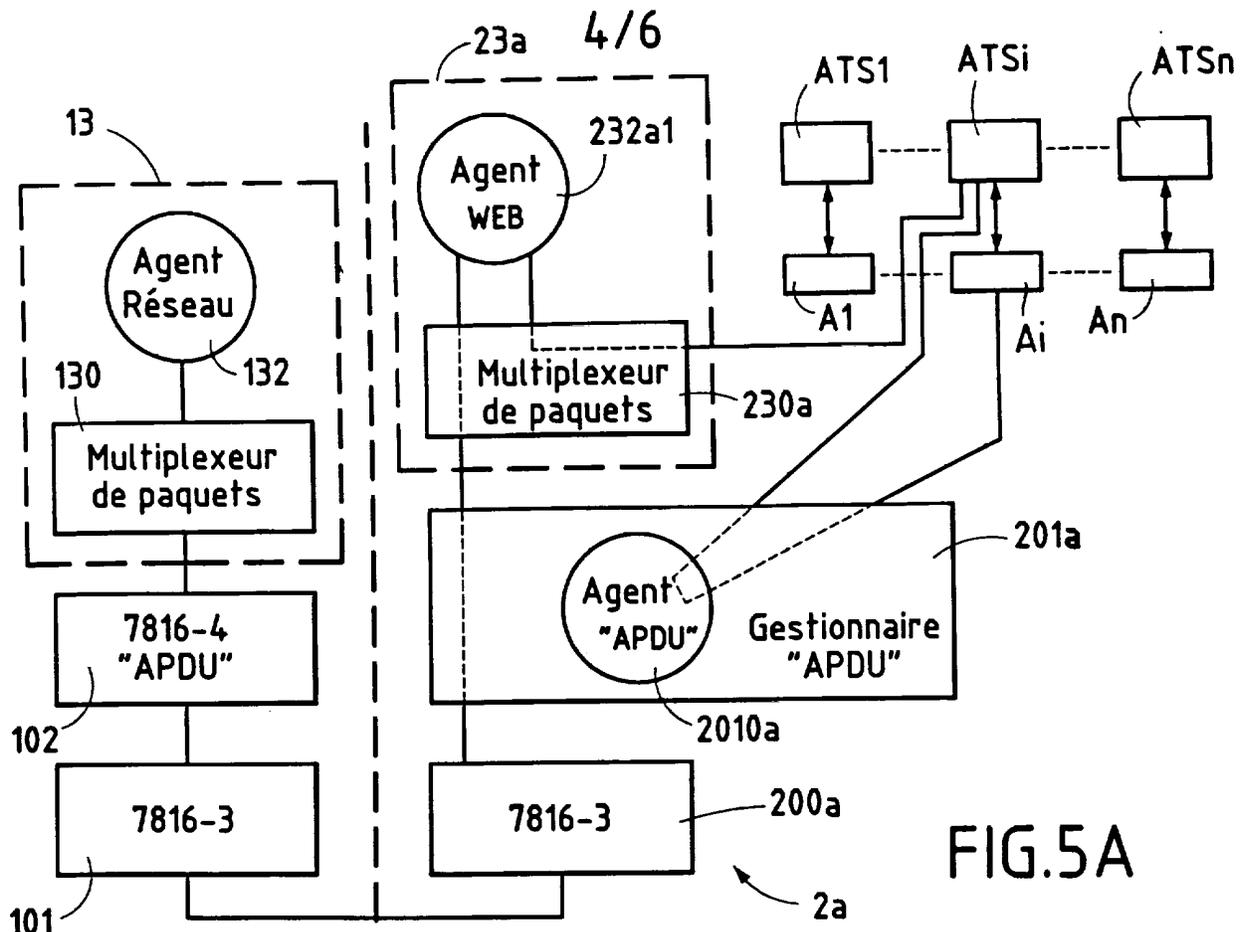


FIG. 5A

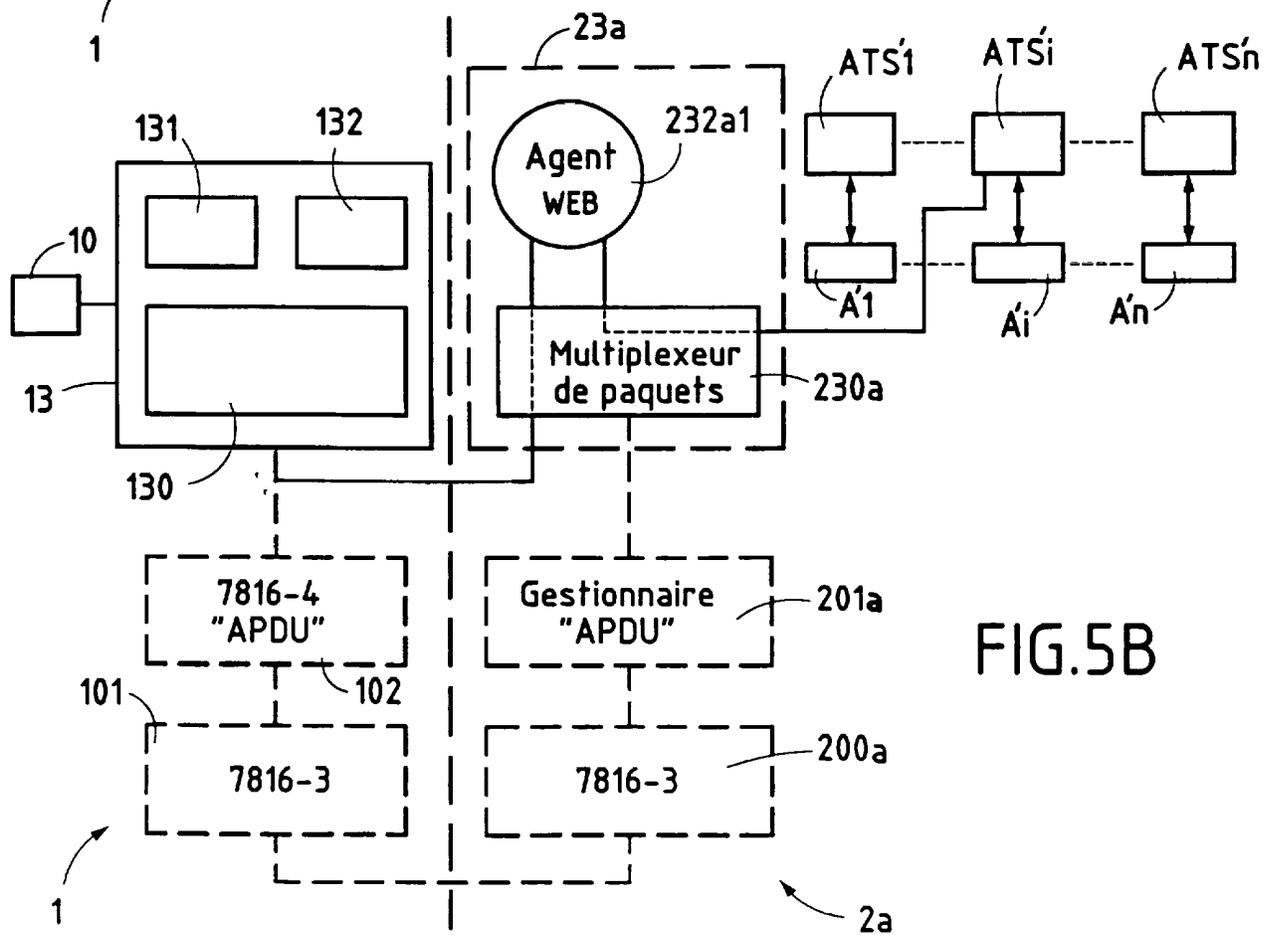


FIG. 5B

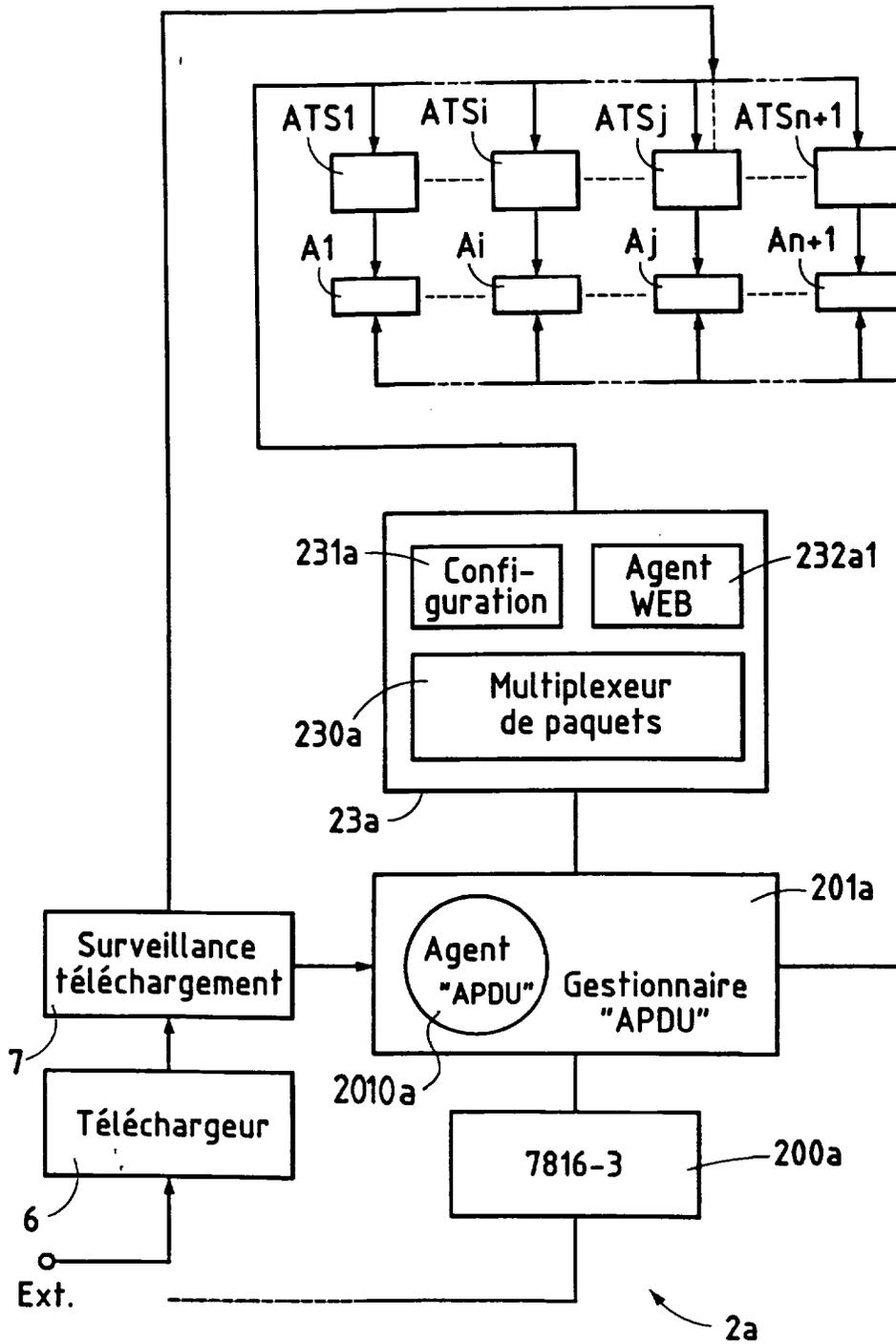
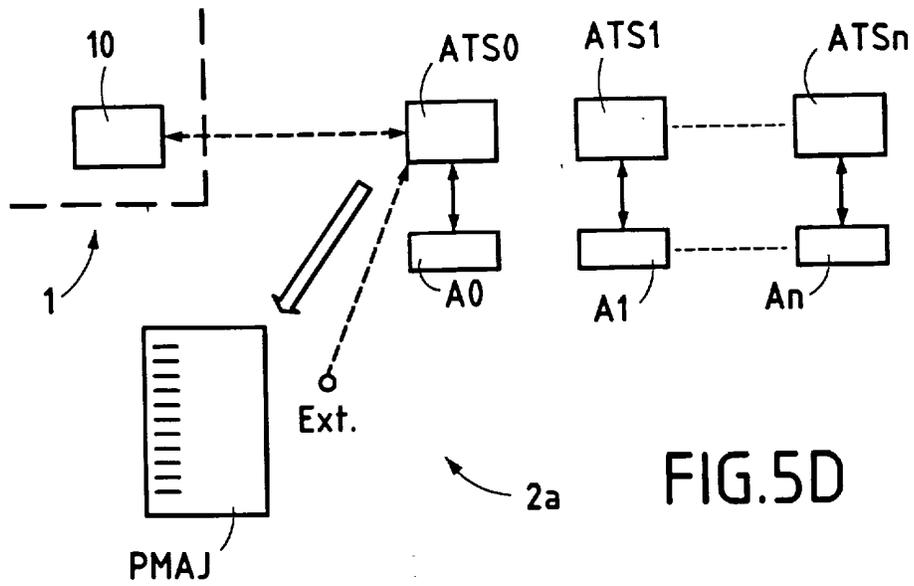


FIG.5C



P.M.E. ~ PAGE D'ACCUEIL

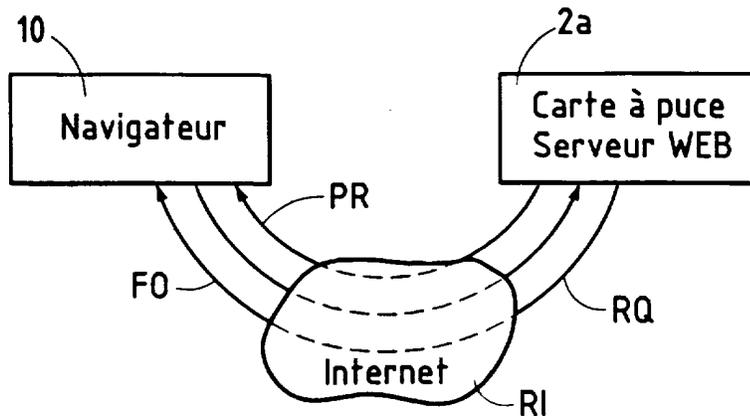
Formulaire de paiement P.M.E

Certificat Ch1

Débit Ch2

b1 b2

FIG. 6



DOCUMENTS CONSIDERES COMME PERTINENTS		Revendications concernées de la demande examinée
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes	
Y	GB 2 326 010 A (IBM) 9 décembre 1998 (1998-12-09)	1-3
A	* page 2, ligne 25 - page 3, ligne 3 * * page 3, ligne 16 - ligne 38; revendication 1; figures 1-6 * * abrégé *	4-16
Y	WO 98 19237 A (SCHLUMBERGER TECHNOLOGIES INC) 7 mai 1998 (1998-05-07)	1-3
A	* page 6, ligne 22 - page 7, ligne 19 * * page 7, ligne 24 - ligne 31 * * page 8, ligne 5 - page 9, ligne 24; revendications 1,3-5,8,11,22; figures 1,2,12,14 * * abrégé *	4-10
A	WO 97 50207 A (TELIA AB PUBL) 31 décembre 1997 (1997-12-31)	1-16
	* page 3, ligne 1 - ligne 4 * * page 4, ligne 2 - ligne 6 * * page 5, ligne 11 - ligne 15 * * page 6, ligne 6 - ligne 13 * * page 11, ligne 11 - ligne 13; revendication 1; figures 1,2 * * abrégé *	
A	WO 97 05582 A (KEYCORP LIMITED ;COOPER DEAN ANTHONY (AU); ACHELLES PETER (AU)) 13 février 1997 (1997-02-13)	1-16
	* colonne 2, ligne 32 - colonne 3, ligne 13 * * colonne 3, ligne 29 - ligne 51; revendication 1; figures 1,2 * * abrégé *	
	--- -/--	
Date d'achèvement de la recherche		Examineur
26 novembre 1999		Wauters, J
<p>CATEGORIE DES DOCUMENTS CITES</p> <p>X : particulièrement pertinent à lui seul Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie A : pertinent à l'encontre d'au moins une revendication ou arrière-plan technologique général O : divulgation non-écrite P : document intercalaire</p> <p>T : théorie ou principe à la base de l'invention E : document de brevet bénéficiant d'une date antérieure à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure. D : cité dans la demande L : cité pour d'autres raisons & : membre de la même famille, document correspondant</p>		

1

EPO FORM 1503 03.82 (P04C13)

INSTITUT NATIONAL
de la
PROPRIETE INDUSTRIELLE

RAPPORT DE RECHERCHE
PRELIMINAIRE

établi sur la base des dernières revendications
déposées avant le commencement de la recherche

N° d'enregistrement
national

FA 574869
FR 9902056

DOCUMENTS CONSIDERES COMME PERTINENTS		Revendications concernées de la demande examinée
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes	
A	EP 0 626 664 A (GEMPLUS CARD INT) 30 novembre 1994 (1994-11-30) * page 2, ligne 5 - ligne 17; revendication 1; figures 2,4,5,8,13 * * abrégé *	1-4,7-9, 11-16
A	WO 98 25239 A (STRATEGIC ANALYSIS INC) 11 juin 1998 (1998-06-11) * page 2, ligne 36 - page 3, ligne 13 * * page 3, ligne 32 - page 4, ligne 2 * * page 4, ligne 12 - ligne 26; figures 1-3 * * abrégé *	1-4
A	WO 98 43212 A (CHAN ALFRED ;WEISE JOEL M (US); WENTKER DAVID C (US); KEKICHEFF MA) 1 octobre 1998 (1998-10-01) * figures 1-4 * * abrégé *	10
A	FR 2 657 445 A (GEMPLUS CARD INT) 26 juillet 1991 (1991-07-26)	
		DOMAINES TECHNIQUES RECHERCHES (Int.CL.6)
Date d'achèvement de la recherche		Examineur
26 novembre 1999		Wauters, J
<p>CATEGORIE DES DOCUMENTS CITES</p> <p>X : particulièrement pertinent à lui seul Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie A : pertinent à l'encontre d'au moins une revendication ou arrière-plan technologique général O : divulgation non-écrite P : document intercalaire</p> <p>T : théorie ou principe à la base de l'invention E : document de brevet bénéficiant d'une date antérieure à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure. D : cité dans la demande L : cité pour d'autres raisons & : membre de la même famille, document correspondant</p>		

1

EPO FORM 1503 03.82 (P04C13)