

①⑨ RÉPUBLIQUE FRANÇAISE
INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE
PARIS

①⑪ N° de publication :

2 874 146

(à n'utiliser que pour les
commandes de reproduction)

②① N° d'enregistrement national :

04 51779

⑤① Int Cl⁸ : H 04 N 5/44 (2006.01), G 06 F 9/445

①②

DEMANDE DE BREVET D'INVENTION

A1

②② Date de dépôt : 04.08.04.

③⑦ Priorité :

④③ Date de mise à la disposition du public de la
demande : 10.02.06 Bulletin 06/06.

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

⑥⑦ Références à d'autres documents nationaux
apparentés :

⑦① Demandeur(s) : THOMSON LICENSING S.A. Société
anonyme — FR.

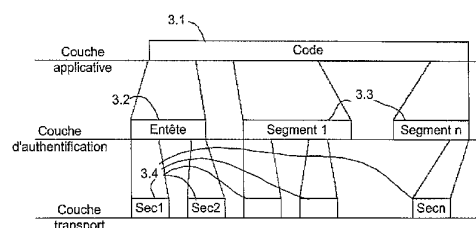
⑦② Inventeur(s) : ALBERT FREDERIC, BOUSSAAD
HAMID, GAUTIER ERIC, JUMPERTZ JEAN LUC et
RABOISSON STEPHANE.

⑦③ Titulaire(s) :

⑦④ Mandataire(s) : THOMSON.

⑤④ METHODE DE MISE-A-JOUR DU LOGICIEL RESIDENT DANS UN APPAREIL ET APPAREIL ADAPTE A SE
METTRE A JOUR PAR LA METHODE.

⑤⑦ L'invention permet de mélanger dans un même flux
différentes versions de logiciel à destination de différents
décodeurs. Le principe est de définir un format de données
unique qui permet de mettre à jour un décodeur à partir d'un
flux transportant les segments binaires de plusieurs versions
de logiciel résident, ces segments binaires étant composés
de segments communs à plusieurs versions de
logiciel et de segments spécifiques à chacun des versions
de logiciel.



FR 2 874 146 - A1



**Méthode de mise-à-jour du logiciel résident dans un appareil et appareil
adapté à se mettre à jour par la méthode.**

La présente invention concerne la mise-à-jour du logiciel résident
5 contenu dans des appareils. Le contexte est plus particulièrement la mise-à-
jour par distribution dans un flux du logiciel de gestion de l'appareil.

Il est connu, en particulier dans le domaine des appareils de
réception de télévision numérique couramment appelés décodeurs numériques
10 de télévision, de distribuer dans des flux de type flux de transport MPEG
(MPEG-2 System : ISO/IEC, 1994. Generic Coding of Moving Pictures and
Associated Audio : Systems, (MPEG-2 Systems Specification), November,
ISO/IEC 13818-1) des versions de mise-à-jour du logiciel résident dans le
décodeur. On entend par logiciel résident l'ensemble logiciel embarqué dans
15 un appareil et servant à le faire fonctionner. Dans ce domaine, une version
donnée de logiciel résident s'applique généralement à un décodeur en fonction,
entre autres, de la version matérielle de ce décodeur, de l'opérateur gérant le
décodeur voire de l'usage prévu pour ce décodeur. Il est habituel de construire
un flux MPEG, appelé flux de téléchargement, contenant une version du logiciel
20 résident destiné à la mise-à-jour d'un type bien défini de décodeur.

A l'origine du déploiement des décodeurs numériques, chaque flux
de mise-à-jour transportait l'ensemble des segments constituant le logiciel
fonctionnel et était adressé à un type de plates-formes particulier. Au fur et à
25 mesure de l'extension du parc d'abonnés les opérateurs ont souvent fait appel
à de nouveaux industriels pour diversifier leurs fournisseurs de décodeurs. Ces
derniers ont eux-mêmes cherchés à réduire le coût des équipements en
fournissant de nouveaux décodeurs basés sur de nouvelles versions
matérielles. Le parc matériel est alors devenu hétérogène et le besoin en
30 bande passante allouée aux flux de téléchargement est devenu plus important.

En même temps, pour les différentes versions matérielles de
décodeurs qu'un industriel fournit à un opérateur, les versions logiciel
correspondantes partagent souvent des segments de code communs. Ces
35 segments sont en général plus particulièrement situés dans les couches hautes
de l'architecture logiciel (piles de protocole, applications, interface...).

On rencontre un problème identique chez les industriels qui utilisent le téléchargement de logiciel dans les phases de production. En effet ils peuvent être amenés à fabriquer à un moment donné sur une même chaîne de production et à partir d'une même version matérielle de matériel, des 5 décodeurs pour différents opérateurs, ces décodeurs étant téléchargés avec des versions logiciel différentes, ces versions de logiciel partageant des segments de code communs. Ces segments sont en général plus particulièrement situés dans les couches basses de l'architecture logiciel 10 (système d'exploitation, pilotes...).

Le problème technique consiste donc à diminuer les besoins en bande passante allouée au flux de téléchargement et ceci aussi bien pour l'opérateur que l'industriel. Ce problème peut s'énoncer de la manière 15 suivante :

Comment un opérateur peut diminuer le débit alloué aux flux transportant différentes versions de logiciel dédiées à des plates-formes différentes, quand ces différentes versions de logiciel partagent des segments 20 de code communs (piles de protocole, applications, interface...) ?

Comment un industriel peut améliorer son processus de fabrication en diminuant le débit alloué aux flux de téléchargement diffusés dans ses usines de production, quand ceux ci sont utilisés pour télécharger sur une 25 même plate-forme, différentes versions de logiciel dédiées à différents opérateurs, ces différentes versions de logiciel présentant des segments de code communs (système d'exploitation, pilotes...) ?

En corollaire à ces problèmes on peut ajouter comment diminuer le 30 temps de téléchargement pour améliorer l'expérience de l'utilisateur?

L'invention permet de mélanger dans un même flux différentes versions de logiciel à destination de différents décodeurs. Le principe est de définir un format de données unique qui permet de mettre à jour un décodeur à 35 partir d'un flux transportant les segments binaires de plusieurs versions de logiciel résident, ces segments binaires étant composés de segments

communs à plusieurs versions de logiciel et de segments spécifiques à chacun des versions de logiciel.

5 En corollaire la procédure mise en œuvre par le loader pour récupérer les segments de code dédiés à une version matérielle de décodeur est plus complexe. Néanmoins cette procédure peut être améliorée pour faire en sorte que le loader ne cherche à récupérer que les segments de code qui doivent être mis à jour.

10 L'invention concerne une méthode de construction d'un support de distribution contenant une pluralité de versions de logiciel résident à destination d'une pluralité d'appareils qui comporte une étape d'identification au sein des différentes versions de logiciel résident composant la pluralité des parties communes à au moins deux versions de logiciel résident, ainsi qu'une étape de
15 découpage des différentes versions de logiciel résident composant la pluralité en segments, ainsi que l'assemblage de ces segments au sein d'un support de distribution, les segments composant une partie commune à au moins deux versions se retrouvant une seule fois au sein du support de distribution.

20 Selon un mode particulier de réalisation l'invention comporte en outre une étape de marquage de chaque segment par un moyen d'identification des appareils visés par la ou les versions de logiciel résident dont le segment fait partie.

25 Selon un mode particulier de réalisation de l'invention la pluralité d'appareils est une pluralité d'appareils de réception de télévision numériques.

Selon un mode particulier de réalisation de l'invention le support de distribution est un flux de transport MPEG.

30

L'invention concerne également une méthode d'acquisition de tout ou partie d'une version de logiciel résident par un appareil possédant des moyens d'accès à un support de distribution distribué par un réseau de distribution, ledit support comportant une pluralité de segments composant une
35 pluralité de versions de logiciel résident, la méthode comporte une étape d'identification au sein du support de distribution d'au moins un segment destiné audit appareil, ainsi qu'une étape de téléchargement dudit segment

identifié au sein de l'appareil depuis ledit support de distribution et le stockage en mémoire dudit segment.

5 Selon un mode particulier de réalisation de l'invention l'étape d'identification au sein du support de distribution comporte une étape de comparaison d'un jeu d'identificateurs identifiant au sein de chaque segment les appareils visés par la version de logiciel résident auquel le segment appartient et un jeu d'identificateurs résident sur l'appareil et codant son type et son utilisation.

10

Selon un mode particulier de réalisation de l'invention le jeu de d'identificateurs contient au moins un identificateur de plate-forme identifiant le type précis de version matérielle de l'appareil, un identificateur de produit identifiant l'opérateur opérant l'appareil et un identificateur d'usage identifiant l'usage prévu de l'appareil.

15

Selon un mode particulier de réalisation de l'invention l'appareil est un récepteur de télévision numérique et le support de distribution un flux de transport MPEG.

20

L'invention concerne également un support de distribution d'une pluralité de versions de logiciel résident à destination d'une pluralité d'appareils, lesdites versions de logiciel étant découpées en segment où ledit support de distribution rassemble les segments de la pluralité de versions de logiciel résident, les segments communs à au moins deux versions de logiciel n'étant présent qu'une seule fois sur le support.

25

Selon un mode particulier de réalisation de l'invention chaque segment possède un jeu d'identificateurs identifiant au sein de chaque segment les appareils visés par la version de logiciel résident auquel le segment appartient.

30

Selon un mode particulier de réalisation de l'invention le jeu de d'identificateurs contient au moins un identificateur de plate-forme identifiant le type précis de version matérielle de l'appareil, un identificateur de produit identifiant l'opérateur opérant l'appareil et un identificateur d'usage identifiant l'usage prévu de l'appareil.

35

Selon un mode particulier de réalisation de l'invention le support a la forme d'un flux de transport MPEG.

5 L'invention concerne également un appareil possédant des moyens d'accès à un support de distribution contenant une pluralité de versions de logiciel résident à destination d'une pluralité d'appareils, ces versions étant découpées en une pluralité de segments au sein du support de distribution, ledit appareil possédant des moyens de télécharger des segments présents au
10 sein du support de description et des moyens de stocker ces segments et possédant des moyens de déterminer les segments faisant partie d'une version de logiciel résident lui étant destinés parmi la pluralité de segments présents au sein du support de distribution.

15 Selon un mode particulier de réalisation de l'invention l'appareil possède un jeu d'identificateurs codant son type et son utilisation et des moyens de comparaison de ce jeu d'identificateurs avec un jeu d'identificateurs présent au sein de chaque segment présent au sein du support de distribution et identifiant les appareils visés par la version de logiciel résident auquel le
20 segment appartient.

Selon un mode particulier de réalisation de l'invention les moyens d'accès à un support de distribution sont des moyens d'accès à un flux de transport MPEG.

25

L'invention sera mieux comprise, et d'autres particularités et avantages apparaîtront à la lecture de la description qui va suivre, la description faisant référence aux dessins annexés parmi lesquels :

30 La figure 1 montre l'architecture connue d'un système de diffusion de logiciel de téléchargement sur un parc de décodeurs.

La figure 2 montre un exemple d'architecture logicielle d'un décodeur.

35 La figure 3 montre les différents niveaux de découpages du logiciel résident dans un exemple de réalisation de l'invention.

La figure 4 montre le segment d'entête dans l'exemple de réalisation de l'invention.

La figure 5 montre le format de la section MPEG dans l'exemple de réalisation de l'invention.

La figure 6 montre les étapes de la méthode utilisée par le décodeur pour retrouver les segments qu'il doit télécharger dans l'exemple de réalisation de l'invention.

La figure 7 montre l'architecture d'un décodeur selon l'exemple de réalisation de l'invention.

Un exemple de réalisation de l'invention va maintenant être décrit.

10 Cet exemple se situe dans le domaine des décodeurs de télévision numérique, mais l'invention peut s'appliquer à tout appareil possédant un logiciel résident susceptible d'être mis à jour au travers d'un moyen de diffusion général. On peut citer par exemple des lecteurs de DVD mis à jour par un DVD unique contenant des versions de logiciel destinées à différentes plate-formes ou

15 autre.

La figure 7 illustre l'architecture d'un décodeur référencé 7.1 selon un exemple de réalisation de l'invention. Ce décodeur est constitué d'un module de réception référencé 7.7 (« Tuner » en anglais) permettant la

20 réception des flux contenant les services audio et vidéo. Ce module peut être un module satellite, câble, terrestre voire une interface de connexion réseau de type IP. Les flux issus de ce module de réception seront ensuite traités par le démultiplexeur/décodeur MPEG référencé 7.6. Ce module va être en charge de la vérification des droits et du décryptage du flux si nécessaire. Ensuite le

25 service désiré sera isolé dans le flux, en effet un même flux peut contenir un multiplexage de plusieurs services audio et vidéo. Une fois le service désiré isolé, les différents flux élémentaires en sont extraits. Ces flux se composent généralement au minimum d'un flux vidéo et d'un flux audio compressés, généralement aujourd'hui, selon la norme MPEG-2 mais l'invention fonctionne

30 quel que soit le format des services, comme MPEG-4 ou autre. Ces flux sont donc décompressés puis convertis en analogique pour fournir une sortie vidéo et une sortie audio. Toutes ces opérations sont pilotées par un logiciel résident stocké de manière répartie entre la mémoire morte référencée 7.4 et la mémoire permanente réinscriptible référencée 7.3. La mémoire vive

35 référencée 7.2 va servir, quant à elle, de mémoire de travail pour l'exécution du logiciel résident.

La figure 2 illustre schématiquement les différents blocs entrant dans la composition du logiciel résident présent sur l'exemple de réalisation de l'invention. Ce logiciel résident fonctionne sur le matériel, référencé 2.8, que nous avons décrit plus haut. Il se compose de deux parties, une première, 5 référencée 2.5, consiste en un chargeur référencé 2.6 et d'un ensemble minimal de pilotes, référencé 27, permettant à ce chargeur de fonctionner. Typiquement ce chargeur et ses pilotes sont stockés en mémoire permanente non réinscriptible de façon à garantir leur intégrité au cours du temps. En effet ce chargeur a pour fonction de permettre la mise-à-jour du logiciel du décodeur 10 et doit pouvoir fonctionner en tout état de cause quel que soit l'état du logiciel résident stocké dans la mémoire réinscriptible, sa corruption ne doit pas être possible.

L'autre partie du logiciel résident, référencée 2.1, consiste en un jeu 15 complet de pilotes, référencés 2.4, permettant la gestion du matériel. Au-dessus de ces pilotes se trouve un système d'exploitation, référencé 2.3, offrant un ensemble de fonctionnalités génériques reposant sur les pilotes. L'application, référencée 2.2, comporte le logiciel de haut niveau qui va apporter à l'utilisateur les fonctionnalités de son décodeur. Il est composé d'une 20 interface homme-machine et implémente les fonctionnalités telles que le changement de service, la gestion du moteur d'interactivité éventuel, des applications interactives et autre. Il est courant que cette partie comporte aussi un chargeur, référencé 2.9, permettant la mise-à-jour de cet ensemble logiciel référencé 2.1. Ce chargeur, peut lui-même être mis à jour. En effet, lorsqu'il 25 télécharge une nouvelle version du logiciel système, celui-ci peut contenir une nouvelle version de ce chargeur qui va être mise en mémoire en lieu et place du chargeur référencé 2.9. Si un problème se produit lors de cette mise-à-jour, le chargeur référencé 2.6 stocké en mémoire permanente sera toujours en mesure de permettre une nouvelle mise-à-jour du logiciel permettant de 30 résoudre le problème et de retrouver une version opérationnelle de la partie logicielle référencée 2.1 du logiciel et donc du chargeur en faisant partie référencé 2.9.

Le besoin d'effectuer un téléchargement de logiciel dans un 35 décodeur peut avoir plusieurs causes. La première cause est la corruption du logiciel installé. En effet, la partie logicielle référencée 2.1 étant stockée en mémoire réinscriptible, il est toujours possible qu'elle se retrouve corrompue.

Dans ce cas, la corruption est détectée au démarrage suivant de l'appareil qui va vérifier par un système de CRC (« Cyclic Redondancy Check ») l'intégrité des modules composant le logiciel. Si une corruption est détectée dans un module autre que le chargeur, celui-ci va être utilisé pour télécharger une nouvelle version intègre du logiciel. Il est ici possible, soit de mettre à jour une nouvelle version complète du logiciel système ou simplement des modules détectés comme corrompus. Lorsque le chargeur lui-même est détecté comme corrompu la mise-à-jour est confiée au chargeur permanent référencé 2.6.

Une autre cause de téléchargement est la disponibilité d'une nouvelle version du logiciel résident. Cette nouvelle version pouvant corriger des erreurs présentes dans la version courante et/ou ajouter de nouvelles fonctionnalités à ce logiciel résident. Dans ce cas la détection par le décodeur, lors de sa phase de démarrage, d'une nouvelle version va provoquer, éventuellement sous certaines conditions, le téléchargement de cette nouvelle version par le chargeur référencé 2.9 et son installation dans le décodeur.

Il est également généralement possible de provoquer le téléchargement et l'installation d'une version donnée de logiciel. Dans ce cas, l'utilisateur, où le technicien agréé après déverrouillage d'une éventuelle protection, va pouvoir préciser la version de logiciel résident qu'il désire installer sur l'appareil et provoquer son téléchargement. Ceci peut être fait lors d'opérations de maintenance ou lors de la préparation des appareils en usine avant livraison. De cette manière il est toujours possible de configurer un appareil avec une version choisie de logiciel système en fonction de l'usage auquel on destine l'appareil.

L'infrastructure typique de téléchargement est décrite dans la figure 1. L'ensemble d'une version de logiciel destiné à un appareil est constitué de différents modules formant entre autres les pilotes, le système d'exploitation et la couche applicative. Ces modules référencés 1.1, 1.2 et 1.3 vont être assemblés et mélangés dans un flux de transport MPEG sur un mélangeur référencé 1.4. Ce flux va être émis par une station d'émission référencée 1.5 vers un satellite référencé 1.6 qui va le diffuser à destination des décodeurs référencés 1.7 à 1.10. L'infrastructure ici décrite est celle de la diffusion par satellite, mais le schéma est le même pour un autre type de diffusion qui peut être le câble, le terrestre où même un réseau interne au sein d'une usine de

fabrication. Dans ce cas le flux calculé par le mélangeur référencé 1.4 sera émis par un serveur chargé de fabriquer le signal à destination du décodeur.

De manière classique, le flux de téléchargement construit va contenir une

version du logiciel à destination d'un type donné de décodeur mais dans le

- 5 cadre de l'invention il sera possible de mélanger des modules appartenant à plusieurs versions de logiciel dans ce flux.

Un certain nombre d'identificateurs sont utilisés pour marquer les versions de logiciel résident et déterminer les décodeurs concernés. On va
10 trouver principalement les identificateurs suivants qui vont jouer un rôle déterminant dans le processus de téléchargement :

- L'identificateur de plate-forme, composé en fait de l'agrégation de deux identificateurs, PT indiquant le type de plate-forme ce qui correspond au modèle de décodeur et PV permettant
15 d'indiquer le numéro de version dans le type de plate-forme. Cet identificateur permet donc d'identifier précisément un modèle matériel de décodeur.
- L'identificateur de produit, PR, qui va identifier généralement l'opérateur utilisant le décodeur. En effet, un même décodeur
20 matériel ne va pas utiliser le même logiciel en fonction de l'opérateur qui va le déployer.
- L'identificateur d'usage, US, qui va permettre pour une même plate-forme et un même produit de différencier l'usage qui sera
25 fait de la plate-forme. On peut ainsi différencier, par exemple, une plate-forme destinée à un abonné d'une plate-forme destinée au test ou autre.

Un jeu de ces identificateurs est mémorisé dans le décodeur permettant de connaître le modèle exact de plate-forme, l'opérateur et l'usage
30 auquel est destiné ce décodeur. Les flux de téléchargement contiennent également un jeu de ces identificateurs permettant de connaître les décodeurs pour lesquels le logiciel contenu est créé. Dans le cadre de l'invention chaque segment possédera son propre jeu de paramètre.

35 De manière connue lorsqu'un décodeur doit se mettre à jour avec une nouvelle version de logiciel pour les raisons décrites précédemment, il va utiliser des paramètres internes lui permettant d'identifier le flux contenant la

version de logiciel qui lui est destiné. Il se connecte alors sur ce flux et vérifie que les identificateurs de plate-forme, de produit et d'usage contenu dans le flux correspondent bien à ceux qu'il possède. Dans ce cas il va télécharger le logiciel contenu dans le flux et le mémoriser.

5

Dans le cadre de l'exemple de réalisation de l'invention le mécanisme de mise-à-jour est basé sur la transmission de segments binaires référencés 3.3. Ces segments correspondent à un découpage des images mémoires, référencées 3.1, des différentes versions de logiciel résident que l'on veut mélanger dans le flux. Ces segments peuvent correspondre directement aux différents modules composant les versions de logiciel résident ou être le résultat d'un autre mode de découpage. Ces segments de code sont précédés d'un segment entête référencé 3.2 qui décrit l'ensemble des segments de code diffusés. L'ensemble des segments est transporté à travers des sections MPEG référencées 3.4. En effet les flux de transports MPEG sont composés d'une succession de sections de 188 octets.

Le segment « entête » illustré figure 4 est lui-même composé d'un « entête » puis d'une boucle décrivant chacun des segments de code inclus dans le flux et d'un champ signature qui permet d'authentifier ce segment d'entête et son contenu.

L'entête du segment « entête » comprend les champs suivants :

- « OUI » spécifié par DVB (« Digital Video Broadcast », le consortium chargé de définir les standards de diffusion numérique) permet d'identifier de manière unique chaque industriel et donc chaque segment décrit par l'entête
- « Key Index » nécessaire à l'authentification du flux dans le cas où on utiliserait un algorithme basé sur des clés publiques. Il permet d'identifier la clé publique à utiliser pour authentifier le flux
- « FlowVersionId » permet de connaître une évolution du contenu du flux. Cette information peut éventuellement être diffusée dans la signalisation et permettre alors au décodeur de déclencher un téléchargement à chaque fois qu'elle évolue
- « EncryptedSeed, InitialValue » informations nécessaires au déchiffrement des segments

Pour chacun des segments on trouve les informations suivantes :

- 5

 - « PlatformType, PlatformVersion » permettent d'identifier la plate-forme matérielle dans laquelle devra être chargé le segment. Un segment commun à toutes les plate-formes sera identifié de manière unique par un couple « (PlatformType, PlatformVersion) = any »
 - « ProductId » permet d'identifier le produit ou le client. Un segment commun à tous les produits sera identifié de manière unique par un « ProductId = any »
- 10

 - « UsageId » permet d'identifier l'usage du décodeur dans le parc (abonné, test...). Un segment commun à tous les usages sera identifié de manière unique par un « UsageId = any »
 - « SegmentVersionId » identifie la version du segment. Il est utilisé pour savoir si un segment est déjà présent dans le
- 15

 - décodeur et ne nécessite pas d'être téléchargé.
 - « SegmentType, SegmentId » correspondent respectivement au type de segment (système d'exploitation, pilotes, interface, ...) et à son identifiant dans le type.
 - « CrcMemoryType » : Type de CRC (« Cyclic Redundancy Check ») calculé sur le segment stocké en mémoire.
- 20

 - « HashMemoryType » : Type de hachage calculé sur le segment stocké en mémoire. Nécessaire pour vérifier l'authentification du flux.
 - « SegStorageAdress » : Adresse de stockage du segment
- 25

 - en mémoire
 - « SegSizeMemory » : Taille du segment en mémoire (segment non compressé)
 - « CrcFlowType » : Type de CRC calculé sur le segment transporté dans le flux. Nécessaire pour vérifier l'intégrité des
- 30

 - segments lors de la réception
 - « CompressionType » : Type de compression sur le segment transporté dans le flux. Permet de savoir si le segment est compressé ou non et quel algorithme de compression a été
- 35

 - utilisé.
 - « EncryptionType » : Type de chiffrement sur le segment transporté dans le flux. Permet de savoir si le segment est chiffré ou non et quel algorithme de chiffrement a été utilisé.

- « CrcFlow » : CRC du segment transporté dans le flux

On trouve enfin :

« Signature » : information permettant d'authentifier le flux

5

Le format de transport est basé sur le format de transport en sections MPEG illustré figure 5. On y retrouve les informations « OUI », « SegmentNb », « FlowVersionId » décrites ci-dessus. Ces informations seront utilisées pour récupérer les données transmises dans le flux.

10

De cette façon il est possible de mélanger dans le même flux les segments appartenant à différentes images mémoires correspondant à des versions différentes de logiciel résident. Chaque segment possédant son propre jeu de paramètres PT, PV, PR et US, la destination de chaque segment est définie. D'autre part, le fait de donner à certains de ces paramètres la valeur « any » permet de ne pas dupliquer les segments partager entre différentes versions de logiciel. En effet un segment partagé entre les différentes versions de logiciel d'un produit donné destinées à toutes les plateformes matérielles va être marqué avec PR égal à la valeur du produit et (PT, PV) égal à « any » et de même pour les autres paramètres. Pour ce faire, on va commencer par identifier dans les différentes versions de logiciel résident à assembler dans le flux les parties communes à au moins deux de ces versions. Chaque module est en effet marqué par configuration, ou autre, avec un jeu de paramètres PT, PV, PR et US indiquant sa destination. L'analyse de ces paramètres permet donc d'identifier les modules communs à plusieurs versions. Ensuite, lors du découpage de ces modules en segments, chaque segment va être marqué avec le jeu de paramètre PT, PV, PR et US lui correspondant, les segments communs n'étant pas dupliqués lors de la construction du support de distribution, ici le flux MPEG.

20
25
30

Les étapes principales de téléchargement sont illustrées figure 6.

Une première étape, référencée E1, consiste à acquérir le segment entête par un filtrage section sur les champs « TableId, SegmentNb=0, OUI, SectionSyntaxVersion, FlowVersionId ».

35

Une seconde étape référencée E2 consiste à vérifier l'authentification de ce segment entête grâce à sa signature.

5 Une troisième étape référencée E3 consiste à construire la liste des segments correspondant au décodeur effectuant le téléchargement en extrayant successivement (voir table ci-dessous):

1. les segments dédiés à la plate-forme (PT, PV), au produit PR et à l'usage US du décodeur effectuant le téléchargement.
2. les segments dédiés à la plate-forme, au produit, avec US=any.
- 10 3. les segments dédiés à la plate-forme avec (PT, PV)=any. Dans ce cas la valeur de US est ignorée
4. les segments dédiés au produit, à l'usage avec (PT, PV)=any.
5. les segments dédiés au produit avec (PT, PV)=any, US=any.
- 15 6. les segments génériques, c'est à dire où (PT, PV) = any, PR=any et quel que soit la valeur de US.

Plate-forme	Produit	Usage	
PT, PV ? any	PR ? any	US ? any	1
		US = any	2
	PR = any	US non pris en compte	3
PT, PV = any	PR ? any	US ? any	4
		US = any	5
	PR = any	US non pris en compte	6

20 Une quatrième étape, référencée E4, consiste, dans le cas d'une mise-à-jour due à la disponibilité d'une nouvelle version de logiciel, à ne récupérer dans cette liste que les segments dont le «SegVersionId» est différent du «SegVersionId» stocké en mémoire et correspondant à la précédente mise-à-jour. Dans le cas où le téléchargement serait du à une

25 corruption de certains segments dans le décodeur, la version récupérée est la même que celle que l'on possède déjà et l'on va récupérer les segments identifiés comme corrompus.

30 Une cinquième étape consiste à vérifier l'intégrité de chacun des segments récupérés grâce au CRC de chaque segment.

Une sixième étape consiste à stocker en mémoire les segments téléchargés.

- 5 Bien que l'exemple de réalisation de l'invention se situe dans le cadre des décodeurs de télévision numérique, il apparaîtra à l'homme du métier que l'invention peut s'appliquer à tous les appareils possédant des capacités de mise-à-jour de leur logiciel résident via un réseau de distribution de données.

REVENDEICATIONS

1. Méthode de construction d'un support de distribution contenant une pluralité de versions de logiciel résident à destination d'une pluralité
5 d'appareils

caractérisée en ce qu'elle comporte au moins les étapes suivantes :

- 10 - l'identification au sein des différentes versions de logiciel résident composant la pluralité des parties communes à au moins deux versions de logiciel résident ;
- le découpage des différentes versions de logiciel résident composant la pluralité en segments ;
- 15 - l'assemblage de ces segments au sein d'un support de distribution, les segments composant une partie commune à au moins deux versions se retrouvant une seule fois au sein du support de distribution.

2. Méthode selon la revendication 1 comportant en outre une étape de marquage de chaque segment par un moyen d'identification des
20 appareils visés par la ou les versions de logiciel résident dont le segment fait partie.

3. Méthode selon la revendication 2 où la pluralité d'appareils est une pluralité d'appareils de réception de télévision numériques.

25

4. Méthode selon la revendication 3 où le support de distribution est un flux de transport MPEG.

5. Méthode d'acquisition de tout ou partie d'une version de logiciel
30 résident par un appareil possédant des moyens d'accès à un support de distribution distribué par un réseau de distribution

caractérisé en ce que, ledit support comportant une pluralité de segments composant une pluralité de versions de logiciel résident, la méthode comporte au moins les étapes suivantes :

- 35 - l'identification au sein du support de distribution d'au moins un segment destiné audit appareil ;

- le téléchargement dudit segment identifié au sein de l'appareil depuis ledit support de distribution ;
- le stockage en mémoire dudit segment.

5 6. Méthode selon la revendication 5 où l'étape d'identification au sein du support de distribution comporte une étape de comparaison d'un jeu d'identificateurs identifiant au sein de chaque segment les appareils visés par la version de logiciel résident auquel le segment appartient et un jeu d'identificateurs résident sur l'appareil et codant son type et son utilisation.

10

7. Méthode selon la revendication 6 où le jeu de d'identificateurs contient au moins un identificateur de plate-forme identifiant le type précis de version matérielle de l'appareil, un identificateur de produit identifiant l'opérateur opérant l'appareil et un identificateur d'usage identifiant l'usage
15 prévu de l'appareil.

8. Méthode selon la revendication 7 où l'appareil est un récepteur de télévision numérique et le support de distribution un flux de transport MPEG.

20

9. Support de distribution d'une pluralité de versions de logiciel résident à destination d'une pluralité d'appareils, lesdites versions de logiciel étant découpées en segment

caractérisé en ce que ledit support de distribution rassemble les
25 segments de la pluralité de versions de logiciel résident, les segments communs à au moins deux versions de logiciel n'étant présent qu'une seule fois sur le support.

10. Support selon la revendication 9 où chaque segment possède
30 un jeu d'identificateurs identifiant au sein de chaque segment les appareils visés par la version de logiciel résident auquel le segment appartient.

11. Support selon la revendication 10 où le jeu de d'identificateurs contient au moins un identificateur de plate-forme identifiant le type précis de
35 version matérielle de l'appareil, un identificateur de produit identifiant l'opérateur opérant l'appareil et un identificateur d'usage identifiant l'usage prévu de l'appareil.

12. Support selon la revendication 11 ayant la forme d'un flux de transport MPEG.

5 13. Appareil possédant des moyens d'accès à un support de distribution contenant une pluralité de versions de logiciel résident à destination d'une pluralité d'appareils, ces versions étant découpées en une pluralité de segments au sein du support de distribution, ledit appareil possédant des moyens de télécharger des segments présents au sein du support de description et des moyens de stocker ces segments

10 caractérisé en ce qu'il possède des moyens de déterminer les segments faisant partie d'une version de logiciel résident lui étant destinés parmi la pluralité de segments présents au sein du support de distribution.

15 14. Appareil selon la revendication 13 où l'appareil possède un jeu d'identificateurs codant son type et son utilisation et des moyens de comparaison de ce jeu d'identificateurs avec un jeu d'identificateurs présent au sein de chaque segment présent au sein du support de distribution et identifiant les appareils visés par la version de logiciel résident auquel le

20 segment appartient.

 15. Appareil selon la revendication 14 où les moyens d'accès à un support de distribution sont des moyens d'accès à un flux de transport MPEG.

25

FIGURES

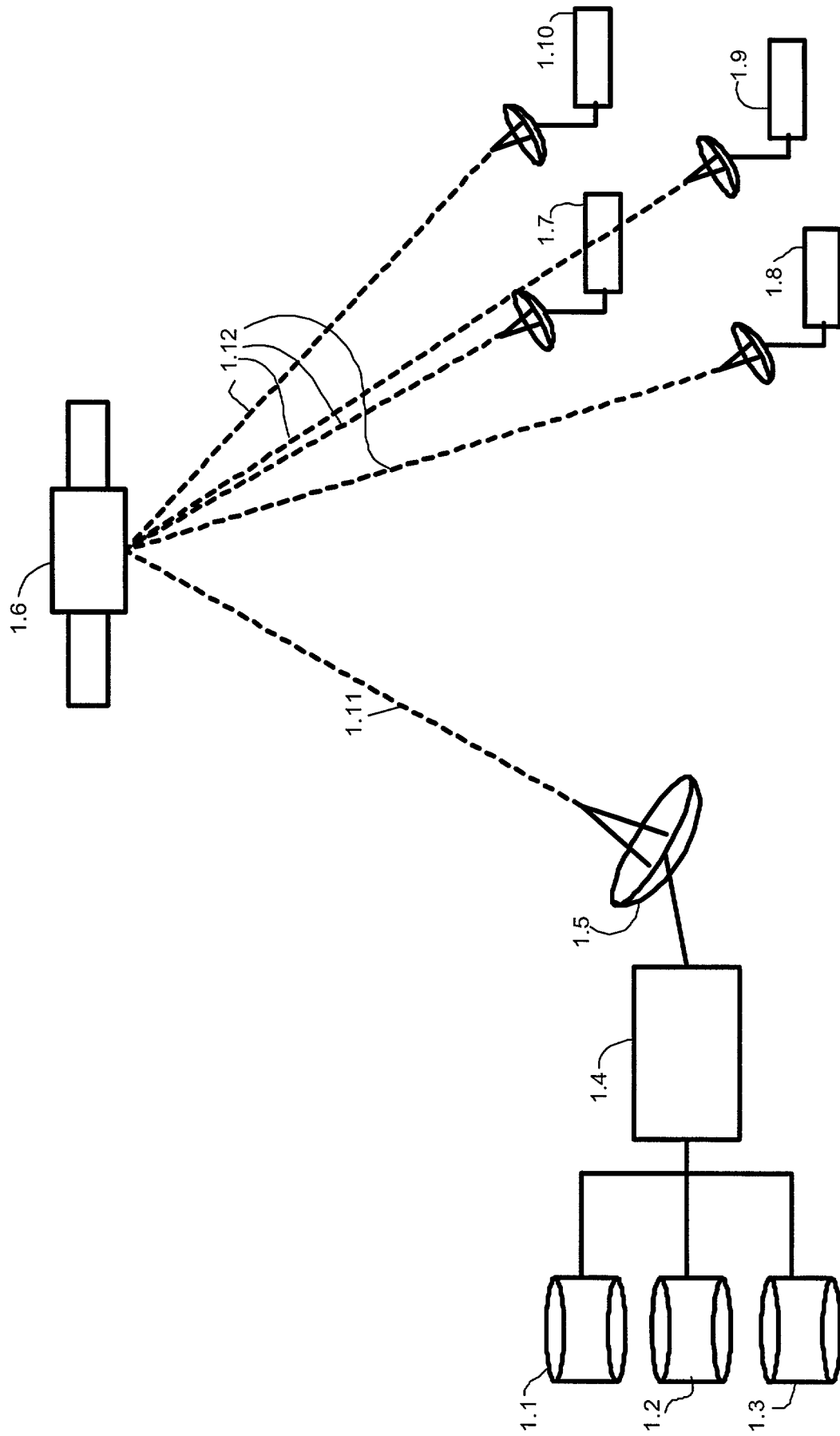
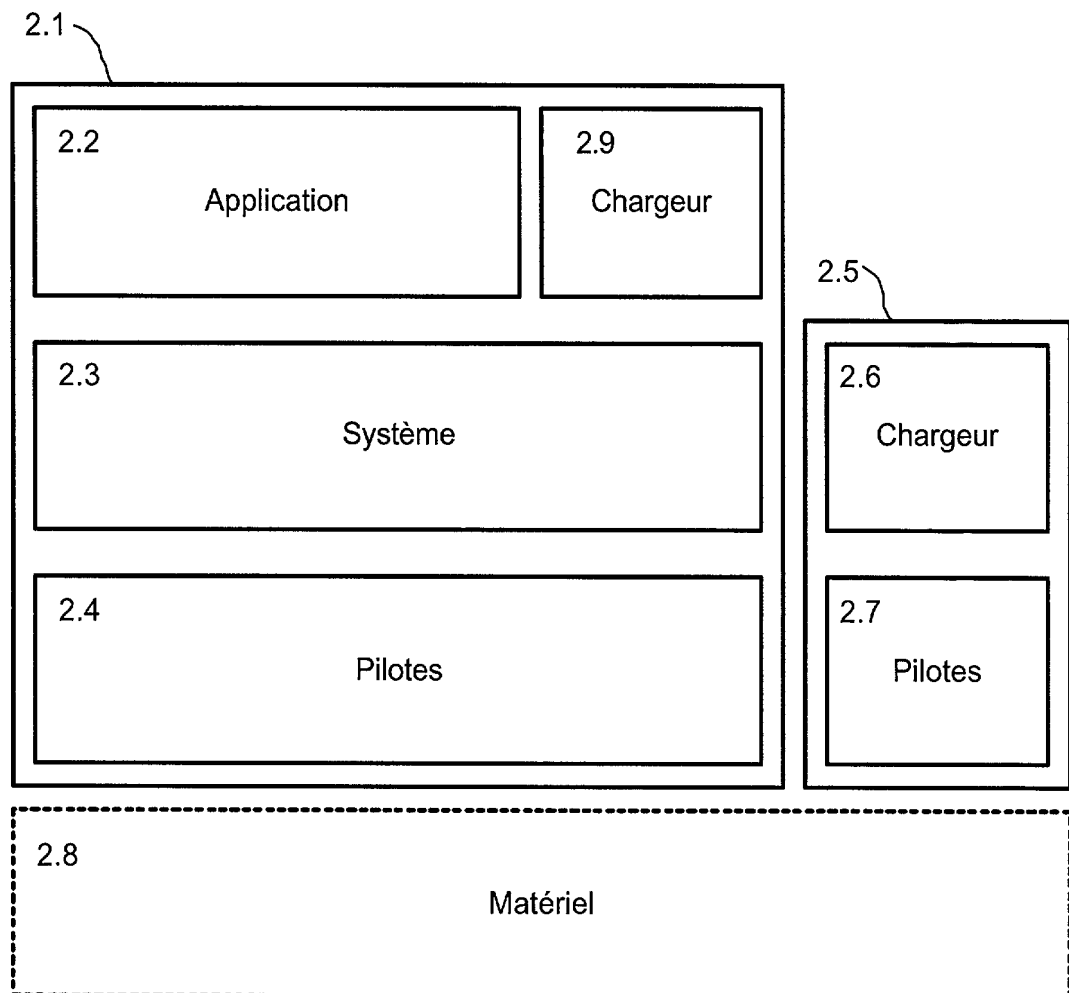
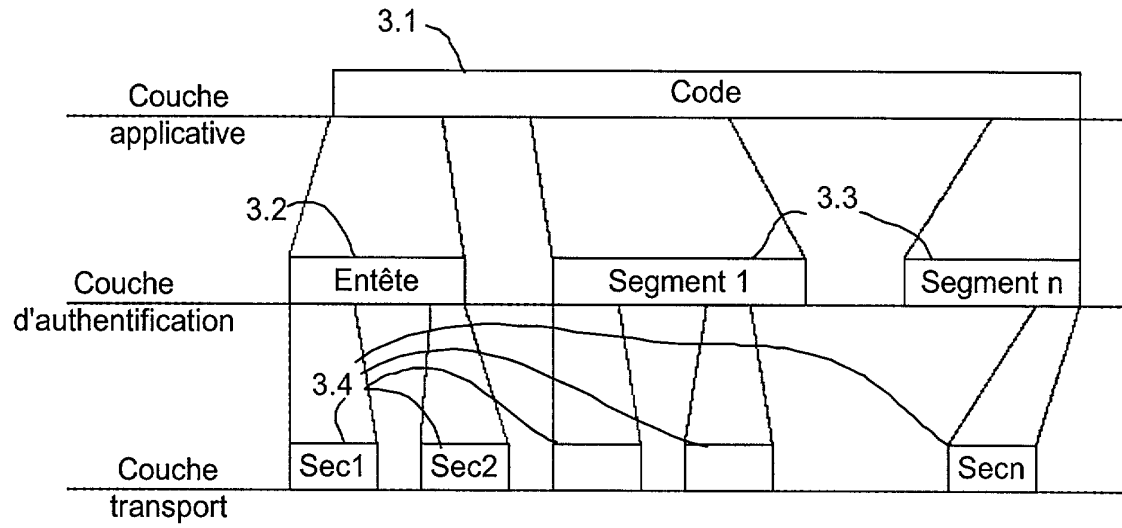


Fig. 1

**Fig. 2**

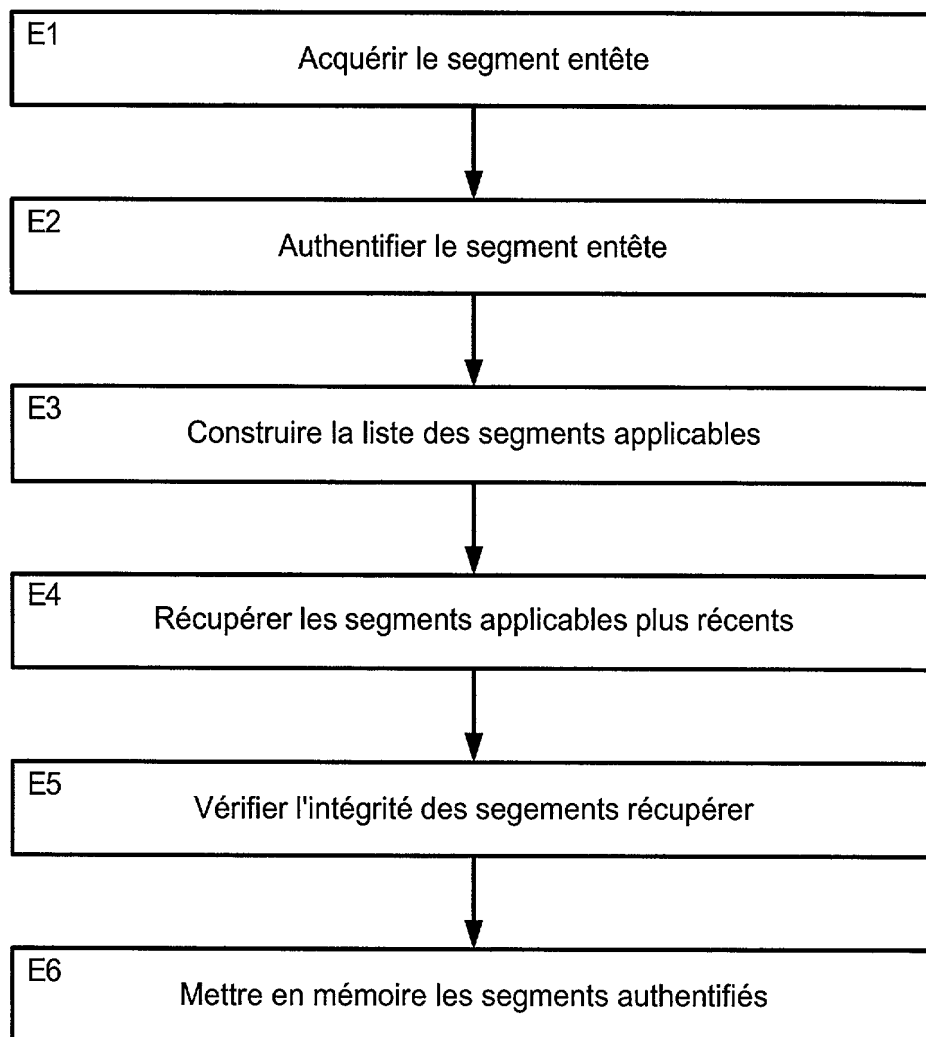
**Fig. 3**

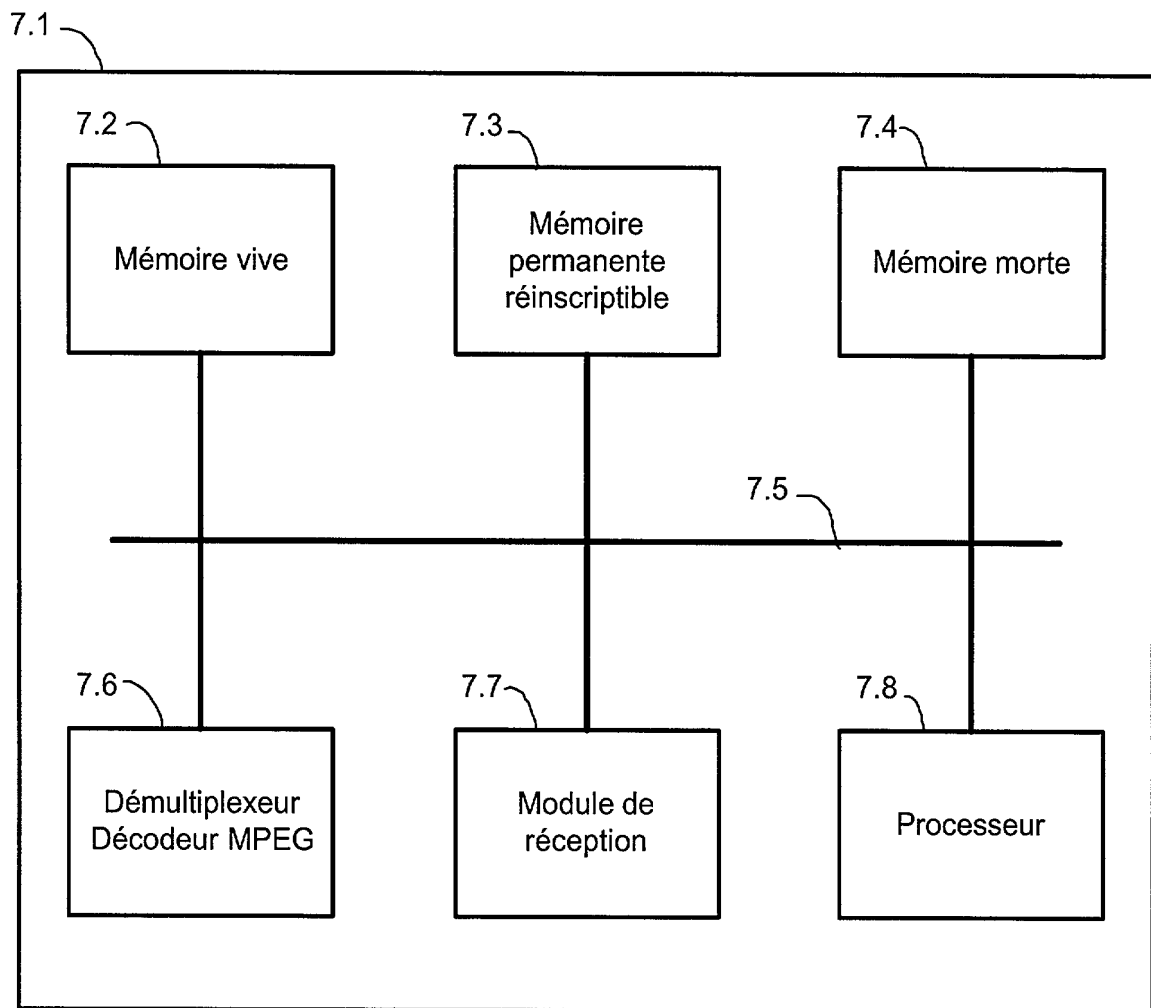
Entête		
Champ	Taille octets	Description
MagicNumber	4	
HeaderSize	4	Taille de l'entête
OUI	3	Identifiant unique d'Organisation (DVB)
Unused	1	
KeyIndex	2	Index de la clef public de vérification de la signature
FlowVersionId	2	Information de version du flux
EncryptedSeed	16	nécessaire au déchiffrement AES
InitialValue	16	nécessaire au déchiffrement AES
Unused	4	
NumberOfSegments	4	Nombre de segments
For(s=0;s<NumberOfSegments;s++){		
SegmentNb	2	Numéro de segment
PlatformType (PT)	2	Type de Plates-formes d'un point de vue matériel
PlatformVersion (PV)	2	Version Matériel (masque de bits)
ProductId (PR)	2	Caractérise le produit (ou le client)
UsageId (US)	1	Usage du segment
SegmentVersionId	1	Numéro de version du segment
SegmentType	1	Type de segment
SegmentId	1	Identifiant du segment
CrcMemoryType	1	Type de CRC sur le segment stocké en mémoire 0x0 = Pas de CRC 0x2 = CRC32St20 0x3 = CRC32Mpeg2
HashMemoryType	1	Type de Hash sur le segment stocké en mémoire 0x0 = Pas de Hash 0x1 = SHA
Unused	2	
SegStorageAddress	4	Adresse de stockage du segment en mémoire flash.
SegSizeMemory	4	Taille du segment en mémoire (non compressé)
CrcMemory	4	CRC du segment stocké en mémoire
HashMemory	20	Hash du segment stocké en mémoire (non compressé)
SegSizeFlow	4	Taille du segment dans le flux (réellement transmis après éventuellement compression et chiffrement)
CrcFlowType	1	Type de CRC sur segment transporté dans le flux (voir CrcMemoryType)
CompressionType	1	Type de compression : 0x0 = pas de compression, 0x1 = compression modifiée issue de gzip
EncryptionType	1	Type de chiffrement 0x0 = pas de chiffrement 0x1 = chiffrement AES
Unused	1	
CrcFlow	4	CRC du segment transporté dans le flux
Unused	4	
}		
Signature	128	Information d'authentification calculée sur l'ensemble des champs précédents

Fig. 4

<i>MPEG Section</i>		
<i>Field</i>	<i>Size in Bits</i>	<i>Description</i>
TableId	8	
SectionSyntaxIndicator	1	Inutilisé
'0'	1	Inutilisé
<i>Reserved</i>	2	
SectionLength	12	Voir DVB
SegmentNb	16	Numéro de segment
<i>Reserved</i>	2	
VersionNumber	5	Inutilisé
CurrentNextIndicator	1	Inutilisé
SectionNumber	8	= 0x00
LastSectionNumber	8	= 0x00
OUI	24	Identifiant d'organisation unique
SectionSyntaxVersion	8	Version de la syntaxe de la de section
FlowVersionId	16	Information de version du flux
Padding	16	
SegmentOffset	32	Indique l'adresse relative of du champ de données dans le flux
For(n=0;n< N;n++){		
Data	N*8	
}		
Crc32	32	

Fig. 5

**Fig. 6**

**Fig. 7**



RAPPORT DE RECHERCHE PRÉLIMINAIRE

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

N° d'enregistrement
national

FA 656310
FR 0451779

DOCUMENTS CONSIDÉRÉS COMME PERTINENTS		Revendication(s) concernée(s)	Classement attribué à l'invention par l'INPI
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes		
X	US 6 453 470 B1 (GAZDA ET AL) 17 septembre 2002 (2002-09-17) * colonne 2, ligne 20 - ligne 62 * * colonne 1, ligne 28 - ligne 30 * * colonne 3, ligne 7 - ligne 57 * * colonne 4, ligne 45 - ligne 67 * * colonne 5, ligne 9 - ligne 15 * * figures 1,2 *	1-15	H04N5/44 G06F9/445
X	----- EP 0 828 386 A (SONY CORPORATION) 11 mars 1998 (1998-03-11) * colonne 1, ligne 3 - ligne 47 * * colonne 4, ligne 41 - colonne 6, ligne 4 * * * colonne 7, ligne 15 - ligne 39 * * colonne 8, ligne 40 - colonne 10, ligne 7 * * colonne 11, ligne 5 - ligne 34 * * colonne 12, ligne 6 - colonne 14, ligne 10 * * figures 1,2,8,9a-9d,12 *	1-15	
X	----- EP 1 437 893 A (MICROSOFT CORPORATION) 14 juillet 2004 (2004-07-14) * alinéa [0005] - alinéa [0016] * * alinéa [0058] * * alinéa [0070] * * alinéa [0082] - alinéa [0097] * * alinéa [0115] - alinéa [0117] * * alinéa [0126] - alinéa [0128] * * alinéa [0131] - alinéa [0156] * * figures 1-10 * ----- -/-	1-3,5-7, 9-11,13, 14	
A		8	H04N G06F
		DOMAINES TECHNIQUES RECHERCHÉS (Int.CL.7)	
Date d'achèvement de la recherche		Examineur	
17 février 2005		Fragua, M	
CATÉGORIE DES DOCUMENTS CITÉS X : particulièrement pertinent à lui seul Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie A : arrière-plan technologique O : divulgation non-écrite P : document intercalaire 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			



RAPPORT DE RECHERCHE PRÉLIMINAIRE

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

N° d'enregistrement
national

FA 656310
FR 0451779

DOCUMENTS CONSIDÉRÉS COMME PERTINENTS		Revendication(s) concernée(s)	Classement attribué à l'invention par l'INPI
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes		
A	US 2003/028899 A1 (MACINNIS) 6 février 2003 (2003-02-06) * alinéa [0011] - alinéa [0012] * * alinéa [0022] - alinéa [0028] * * alinéa [0031] - alinéa [0032] * * alinéa [0036] * * alinéa [0040] - alinéa [0041] * * alinéa [0049] - alinéa [0050] * * figures 1-5 * -----	1-3,5-7, 9-11,13, 14	DOMAINES TECHNIQUES RECHERCHÉS (Int.CL.7)
Date d'achèvement de la recherche		Examineur	
17 février 2005		Fragua, M	
CATÉGORIE DES DOCUMENTS CITÉS X : particulièrement pertinent à lui seul Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie A : arrière-plan technologique O : divulgation non-écrite P : document intercalaire 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			

ANNEXE AU RAPPORT DE RECHERCHE PRÉLIMINAIRE**RELATIF A LA DEMANDE DE BREVET FRANÇAIS NO. FR 0451779 FA 656310**

La présente annexe indique les membres de la famille de brevets relatifs aux documents brevets cités dans le rapport de recherche préliminaire visé ci-dessus.

Les dits membres sont contenus au fichier informatique de l'Office européen des brevets à la date du 17-02-2005

Les renseignements fournis sont donnés à titre indicatif et n'engagent pas la responsabilité de l'Office européen des brevets, ni de l'Administration française

Document brevet cité au rapport de recherche		Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
US 6453470	B1	17-09-2002	AU 1818701 A	30-04-2001
			CA 2392240 A1	05-04-2001
			CN 1391670 T	15-01-2003
			DE 60006415 D1	11-12-2003
			DE 60006415 T2	09-09-2004
			EP 1222534 A2	17-07-2002
			JP 2003510927 T	18-03-2003
			TW 476046 B	11-02-2002
			WO 0124000 A2	05-04-2001

EP 0828386	A	11-03-1998	JP 10084540 A	31-03-1998
			CA 2214900 A1	10-03-1998
			CN 1179672 A ,C	22-04-1998
			DE 69726669 D1	22-01-2004
			DE 69726669 T2	07-10-2004
			EP 0828386 A2	11-03-1998
			ES 2213196 T3	16-08-2004
			ID 19369 A	02-07-1998
			US 6263506 B1	17-07-2001

EP 1437893	A	14-07-2004	US 2004131076 A1	08-07-2004
			EP 1437893 A1	14-07-2004
			JP 2004215284 A	29-07-2004

US 2003028899	A1	06-02-2003	US 6487723 B1	26-11-2002
			US 5951639 A	14-09-1999
			AU 1693597 A	02-09-1997
			DE 69712320 D1	06-06-2002
			DE 69712320 T2	07-11-2002
			EP 0880857 A1	02-12-1998
			WO 9730549 A1	21-08-1997
