Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An agency of Industry Canada

CA 2769239 A1 2011/02/24

(21) 2 769 239

(12) DEMANDE DE BREVET CANADIEN
CANADIAN PATENT APPLICATION

(13) A1

(86) Date de dépôt PCT/PCT Filing Date: 2010/07/28
(87) Date publication PCT/PCT Publication Date: 2011/02/24
(85) Entrée phase nationale/National Entry: 2012/01/26
(86) N° demande PCT/PCT Application No.: FR 2010/000552
(87) N° publication PCT/PCT Publication No.: 2011/020954
(30) Priorité/Priority: 2009/07/28 (FR0955281)

(51) Cl.Int./Int.Cl. *G06F 9/455* (2006.01), *G06F 21/00* (2006.01), *G06F 21/24* (2006.01), *H04W 12/06* (2009.01)

(71) Demandeur/Applicant:
AIRBUS, FR

(72) Inventeurs/Inventors:
VERMANDE, SEVERINE, FR;
BIONDI, PHILIPPE, FR

(74) Agent: ROBIC

(54) Titre : COMPOSANT LOGICIEL ET DISPOSITIF POUR LE TRAITEMENT AUTOMATISE DE DONNEES MULTI-USAGES, METTANT EN OEUVRE DES FONCTIONS AYANT BESOIN DE DIFFERENTS NIVEAUX DE SURETE OU LIMITES DE RESPONSABILITE

(54) Title: AUTOMATED PROCESSING OF MULTI-USAGE DATA, IMPLEMENTING FUNCTIONS REQUIRING VARIOUS LEVELS OF SECURITY OR LIMITS OF RESPONSIBILITY

(57) Abrégé/Abstract:
L'invention a notamment pour objet un composant logiciel pour le traitement automatisé de données multi-usages, mettant en oeuvre des fonctions ayant besoin de différents niveaux de sûreté ou limites de responsabilité. Le composant logiciel selon l'invention comprend une pluralité de machines virtuelles (215), chaque machine virtuelle étant adaptée à exécuter au moins une fonction ayant besoin d'un niveau de sûreté ou d'une limite de responsabilité prédéterminé et un hyperviseur (210) adapté à contrôler l'exécution de ladite pluralité de machines virtuelles.

(12) DEMANDE INTERNATIONALE PUBLIÉE EN VERTU DU TRAITÉ DE COOPÉRATION EN MATIÈRE DE BREVETS (PCT)

(19) Organisation Mondiale de la Propriété Intellectuelle
Bureau international

(43) Date de la publication internationale
24 février 2011 (24.02.2011)

PCT

(10) Numéro de publication internationale
WO 2011/020954 A3

(51) Classification internationale des brevets :
G06F 9/455 (2006.01) *G06F 21/00* (2006.01)

(21) Numéro de la demande internationale :
PCT/FR2010/000552

(22) Date de dépôt international :
28 juillet 2010 (28.07.2010)

(25) Langue de dépôt : français

(26) Langue de publication : français

(30) Données relatives à la priorité :
0955281 28 juillet 2009 (28.07.2009) FR

(71) Déposants *(pour tous les États désignés sauf US)* : **AIRBUS** [FR/FR]; 1, Rond Point Maurice Bellonte, F-31700 Blagnac (FR). **EUROPEAN AERONAUTIC DEFENCE AND SPACE COMPANY EADS FRANCE** [FR/FR]; 37Bd. de Montmorency, F-75016 Paris (FR).

(72) Inventeurs; et

(75) Inventeurs/Déposants *(pour US seulement)* : **VERMANDE, Severine** [FR/FR]; 16, Résidence des Chênes, F-31620 Cepet (FR). **BIONDI, Philippe** [FR/FR]; 26, rue d'Alésia, F-75014 Paris (FR).

(74) Mandataire : **SANTARELLI**; 14, Avenue de la Grande Armée, F-75017 Paris (FR).

(81) États désignés *(sauf indication contraire, pour tout titre de protection nationale disponible)* : AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) États désignés *(sauf indication contraire, pour tout titre de protection régionale disponible)* : ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), eurasien (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), européen (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Déclarations en vertu de la règle 4.17 :

— *relative à la qualité d'inventeur (règle 4.17.iv))*

Publiée :

— *avec rapport de recherche internationale (Art. 21(3))*

— *avant l'expiration du délai prévu pour la modification des revendications, sera republiée si des modifications sont reçues (règle 48.2.h))*

(88) Date de publication du rapport de recherche internationale :
14 avril 2011

WO 2011/020954 A3

(54) Title : AUTOMATED PROCESSING OF MULTI-USAGE DATA, IMPLEMENTING FUNCTIONS REQUIRING VARIOUS LEVELS OF SECURITY OR LIMITS OF RESPONSIBILITY

(54) Titre : TRAITEMENT AUTOMATISE DE DONNEES MULTI-USAGES, METTANT EN OEUVRE DES FONCTIONS AYANT BESOIN DE DIFFERENTS NIVEAUX DE SURETE OU LIMITES DE RESPONSABILITE

(57) Abstract : The subject of the invention is in particular a software component for the automated processing of multi-usage data, implementing functions requiring various levels of security or limits of responsibility. The software component according to the invention comprises a plurality of virtual machines (215), each virtual machine being adapted for executing at least one function requiring a level of security or a limit of responsibility which is predetermined and a hypervisor (210) adapted for controlling the execution of said plurality of virtual machines.

(57) Abrégé : L'invention a notamment pour objet un composant logiciel pour le traitement automatisé de données multi-usages, mettant en œuvre des fonctions ayant besoin de différents niveaux de sûreté ou limites de responsabilité. Le composant logiciel selon l'invention comprend une pluralité de machines virtuelles (215), chaque machine virtuelle étant adaptée à exécuter au moins une fonction ayant besoin d'un niveau de sûreté ou d'une limite de responsabilité prédéterminé et un hyperviseur (210) adapté à contrôler l'exécution de ladite pluralité de machines virtuelles.

Composant logiciel et dispositif pour le traitement automatisé de données multi-usages, mettant en œuvre des fonctions ayant besoin de différents niveaux de sûreté ou limites de responsabilité

La présente invention concerne les systèmes de traitement de données, notamment de données de systèmes d'information d'aéronefs, et plus particulièrement un composant logiciel et un dispositif pour le traitement automatisé de données multi-usages, mettant en œuvre des fonctions ayant besoin de différents niveaux de sûreté ou limites de responsabilité.

Depuis les événements tragiques du 11 septembre 2001 liés aux crashs d'avions commerciaux, la sûreté représente désormais une problématique essentielle de l'aéronautique. Pour répondre à celle-ci, les constructeurs et les compagnies aériennes ont développé et intégré des fonctions visant à améliorer la sûreté à bord des aéronefs.

Ainsi, à titre d'illustration, des portes de cockpit renforcées et des systèmes de vidéo surveillance interne ont été développés. De même, les systèmes d'information embarqués sont désormais protégés contre des malveillances.

Par ailleurs, les compagnies aériennes ont l'obligation règlementaire de mettre en œuvre des moyens techniques et organisationnels pour maintenir le niveau de sûreté des éléments d'un aéronef tel que déterminé à la livraison de celui-ci. Cette obligation règlementaire ne couvre que la sûreté physique et non la sûreté logique.

Cependant, du fait de cette obligation règlementaire, certaines compagnies aériennes demandent aux constructeurs d'aéronefs de permettre l'intégration de fonctions de sûreté dans les processus opérationnels des compagnies aériennes. En outre, certaines compagnies aériennes demandent aux constructeurs d'aéronefs que les fonctions opérationnelles et les fonctions de sûreté associées soient compatibles avec des matériels et des logiciels du commerce non spécifiques à l'aéronautique.

De façon générale, les systèmes de traitement automatisé de données, aussi appelés STADs (acronyme de Système de Traitement Automatisé de Données), peuvent être utilisés, dans un environnement aéronautique, pour héberger des applications logicielles opérationnelles et/ou de communication, c'est-à-dire comme des mallettes à outils, pour permettre au personnel opérationnel, par exemple le pilote et son second, aux techniciens et aux équipes de maintenance, de remplir certaines tâches de leur mission. Ces mallettes à outils peuvent également être ouvertes à d'autres usages. En particulier, une compagnie aérienne peut décider d'y installer ses propres applications métiers ou bureautiques. Les mallettes à outils ne sont pas des fonctions de sûreté, c'est-à-dire que leur rôle n'est pas d'assurer la sûreté, mais de permettre de réaliser des tâches opérationnelles.

Les données manipulées par les applications logicielles opérationnelles mises en œuvre dans les STADs peuvent être téléchargées, calculées, affichées et/ou transmises. En raison des contraintes de sûreté évoquées précédemment, il existe des besoins forts de sûreté en termes de confidentialité, d'intégrité et/ou de disponibilité de celles-ci.

Cependant, il est difficile de faire cohabiter des fonctions et des données sensibles avec des fonctionnalités pouvant communiquer vers l'extérieur et qui reposent sur des logiciels et des matériels du commerce lorsqu'un objectif de sûreté doit être respecté.

L'invention permet de résoudre au moins un des problèmes exposés précédemment.

L'invention a ainsi pour objet un composant logiciel pour ordinateur adapté au traitement automatique de données multi-usages, le composant logiciel mettant en œuvre des fonctions ayant besoin de différents niveaux de sûreté ou limites de responsabilité et comprenant,

- une pluralité de machines virtuelles, chaque machine virtuelle étant adaptée à exécuter au moins une fonction ayant besoin d'un niveau de sûreté ou d'une limite de responsabilité prédéterminé ; et,
- un hyperviseur adapté à contrôler l'exécution de ladite pluralité de machines virtuelles.

Le composant logiciel selon l'invention permet ainsi de mettre en œuvre des fonctions ayant des niveaux de sûreté ou des limites de responsabilité différentes dans une même machine, indépendamment de la plateforme matérielle et de l'architecture du système d'information utilisées à bord des aéronefs. Les éditeurs des applications logicielles mises en œuvre ne sont donc plus dépendants de l'évolution des systèmes d'exploitation et maîtrisent le cycle de vie de ces applications.

Le composant logiciel peut ainsi être mis en œuvre sur un STAD mobile du marché, selon une liste de compatibilité matérielle. Les limites de responsabilité étant clairement identifiées, il peut recevoir des applications logicielles de fournisseurs et de l'utilisateur. Un tel STAD peut être attaché à un aéronef ou à un utilisateur.

L'utilisation du composant logiciel selon l'invention n'augmente pas les besoins de maintenance par rapport à un équipement mono-usage, mobile ou non. Il assure un bon niveau de ségrégation ainsi qu'un bon niveau de sûreté dont l'intégrité des données opérationnelles. Il permet de contrôler le partage des ressources entre les différentes fonctions tout en étant relativement indépendant vis-à-vis des manques de fiabilité des produits du commerce mis en œuvre.

De façon avantageuse, ledit hyperviseur comprend des moyens d'authentification pour authentifier au moins une machine virtuelle de ladite pluralité de machine virtuelle afin, notamment, de contrôler la validité de données transmises. De même, lesdits moyens d'authentification sont, de préférence, adaptés à vérifier l'intégrité de ladite au moins une machine virtuelle authentifiée.

En outre, lesdits moyens d'authentification sont avantageusement adaptés à vérifier le niveau d'isolation de ladite au moins une machine virtuelle authentifiée par rapport à au moins une autre machine virtuelle de ladite pluralité de machines virtuelles afin, notamment, de contrôler la validité de données transmises au regard d'autres machines virtuelles.

Selon un mode de réalisation particulier, le composant logiciel comprend en outre des moyens de mémorisation de données traitées par au

moins une machine virtuelle de ladite pluralité de machine virtuelle, lesdits moyens de mémorisation étant adaptés à mémoriser lesdites données traitées dans une mémoire amovible dudit ordinateur. Le composant logiciel selon l'invention permet ainsi de mémoriser des données dont le niveau de confiance n'est pas sûr sans le compromettre. De tels moyens de mémorisation de données sont, de préférence, mis en œuvre par des machines virtuelles de ladite pluralité de machines virtuelles dont le niveau de sûreté est inférieur à un seuil prédéterminé.

Toujours selon un mode de réalisation particulier, le composant logiciel comprend en outre des moyens de contrôle d'un niveau de confiance d'au moins une donnée traitée par au moins une machine virtuelle de ladite pluralité de machines virtuelles, ladite au moins une donnée traitée ne pouvant être mémorisée localement dans ledit ordinateur qu'après qu'elle ait été contrôlée. Le composant logiciel selon l'invention permet ainsi de ne mémoriser localement que des données dont le niveau de confiance est sûr afin de ne pas le compromettre.

Toujours selon un mode de réalisation particulier, le composant logiciel comprend en outre des moyens de transfert de données entre une première et une seconde machines virtuelles de ladite pluralité de machines virtuelles, lesdits moyens de transfert étant adaptés à filtrer des données transférées si le niveau de sûreté de ladite seconde machine virtuelle est supérieur au niveau de sûreté de ladite première machine virtuelle afin de valider les données échangées, notamment selon leur type ou le besoin d'accès à ces données.

Toujours selon un mode de réalisation particulier, des données de configuration utilisées pour démarrer au moins une machine virtuelle de ladite pluralité de machine virtuelle ne sont pas modifiées au cours de l'exécution de ladite au moins une machine virtuelle démarrée afin de faciliter la maintenance du composant logiciel et permettre son redémarrage à partir d'un état stable et validé.

L'invention a également pour objet un dispositif comprenant des moyens adaptés à la mise en œuvre de chacun des éléments du composant

logiciel décrit précédemment dont les avantages sont similaires à ceux évoqués précédemment.

D'autres avantages, buts et caractéristiques de la présente invention ressortent de la description détaillée qui suit, faite à titre d'exemple non limitatif, au regard des dessins annexés dans lesquels :

- la figure 1 représente schématiquement un exemple d'environnement dans lequel peut être utilisé un système de traitement automatisé de données multi-usages mettant en œuvre l'invention ;
- la figure 2 illustre un exemple d'architecture d'un système de traitement automatisé de données multi-usages selon l'invention ;
- la figure 3 illustre schématiquement un exemple d'adaptation de certaines fonctions exécutées dans des machines ;
- la figure 4 illustre schématiquement certaines étapes mises en œuvre pour analyser les risques associés aux fonctions qui doivent être exécutées sur un même STAD ;
- la figure 5 illustre schématiquement un exemple d'algorithme pour répartir des applications logicielles mises en œuvre dans un STAD dans des machines virtuelles selon les fonctions auxquelles elles font appel; et,
- la figure 6 montre un exemple de dispositif permettant d'implémenter au moins partiellement l'invention.

L'invention permet notamment de remplacer les systèmes de traitement automatisé de données (STAD) mono-usages, mobiles ou fixes, utilisés aujourd'hui pour la maintenance et la mission, par un STAD unique sécurisé, de préférence mobile.

La figure 1 représente schématiquement un exemple d'environnement 100 dans lequel peut être utilisé un système de traitement de données automatique multi-usages mettant en œuvre l'invention. Selon cet exemple, un STAD 105 peut être utilisé par un membre d'équipage dans un aéronef 110 par exemple pour exécuter des applications logicielles de gestion de vol.

Le même STAD 105, ou un STAD similaire 115, peut être utilisé par une équipe de maintenance pour accéder aux données de maintenance de

l'aéronef 110 et/ou pour mettre à jour des données ou des applications logicielles de l'aéronef.

Par ailleurs, le même STAD 105, ou un STAD similaire 120, peut être utilisé dans les bureaux 125 de la compagnie aérienne, par exemple pour la préparation d'un vol. De façon similaire, le même STAD 105, ou un STAD similaire 130, peut être utilisé par son possesseur pour accéder à des applications bureautiques et à sa messagerie électronique à partir, par exemple, d'un accès réseau d'un hôtel 135.

Il convient de remarquer ici que les exemples illustrés sur la figure 1 ne sont donnés qu'à titre illustratif. Ils ne sont pas limitatifs.

Pour permettre la mise en œuvre de fonctions ayant des besoins de niveaux de sûreté différents sur un STAD unique, sans compromettre la sûreté de chacune de ces fonctions, plusieurs techniques sont combinées.

Les applications opérationnelles, les applications bureautiques et les applications personnelles d'un STAD, plus généralement toutes les fonctions mises en œuvre dans un STAD, sont ainsi hébergées dans plusieurs machines virtuelles mises en œuvre dans le STAD, selon les besoins de niveaux de sûreté et, de préférence, par responsabilité.

Il est rappelé ici qu'une machine virtuelle offre un environnement d'exécution ayant ses propres caractéristiques de configuration. En d'autres termes, deux machines virtuelles peuvent être considérées comme deux machines physiques indépendantes. Chaque machine virtuelle s'exécute avec son système d'exploitation, ses pilotes (appelés *drivers* en terminologie anglo-saxonne), ses applications logicielles et sa configuration de gestion et d'échange de données.

Un mécanisme de virtualisation permet notamment l'exécution de plusieurs machines virtuelles sur une machine réelle à l'aide d'un hyperviseur.

L'hyperviseur est responsable du partage des ressources de la machine réelle et de l'application des règles de contrôle d'accès aux ressources. Les ressources partagées entre les machines virtuelles sont, par exemple, la puissance de calcul CPU (sigle de *Central Processing Unit* en terminologie anglo-saxonne), les canaux de communication, les interruptions

matérielles et logicielles, les ports d'entrées/sorties, la mémoire, les horloges, les bus systèmes, les contrôleurs et/ou la mémoire de masse. L'invention est basée sur l'utilisation d'un hyperviseur standard, personnalisé pour gérer les machines virtuelles selon des règles prédéterminées.

La virtualisation mise en œuvre ici est une virtualisation matérielle, par exemple une virtualisation complète selon laquelle l'hyperviseur gère toutes les requêtes des machines virtuelles ou une paravirtualisation selon laquelle certaines requêtes sont gérées directement par les machines virtuelles.

Selon un mode de réalisation particulier, des outils logiciels de virtualisation adaptés au temps réel sont utilisés pour leurs bénéfices en terme de performances et de niveau de sûreté.

Par ailleurs, il est observé que la virtualisation permet, pour des systèmes embarqués, d'optimiser le poids du matériel informatique mis en œuvre comprenant les serveurs, les commutateurs et le câblage, mais aussi une réduction de consommation électrique ainsi qu'une simplification des procédures de déploiement et de maintenance, ce qui est particulièrement avantageux dans un environnement aéronautique.

La figure 2 illustre un exemple d'architecture d'un STAD selon l'invention, suffisamment sûr pour être utilisé dans des environnements ayant des niveaux de sûreté différents.

Comme représenté, le STAD 200 comprend ici une couche matérielle 205. Celle-ci correspond, par exemple, à celle d'un ordinateur personnel portable, aussi appelé *laptop* ou *notebook* en terminologie anglo-saxonne, d'un assistant personnel, aussi appelé PDA (sigle de *Personal Digital Assistant* en terminologie anglo-saxonne), ou d'un *smartphone*.

La couche matérielle devant être de confiance, elle peut consister en une plateforme ouverte de type PC dont le niveau de confiance est amélioré par l'utilisation d'un module d'authentification appelé TPM (sigle de *Trusted Platform Module* en terminologie anglo-saxonne). Il s'agit d'un composant matériel cryptographique défini par le *Trusted Computing Group*.

La couche matérielle 205 permet l'exécution d'une couche logicielle 210 comprenant l'hyperviseur. Elle permet d'exécuter plusieurs machines

virtuelles distinctes, par exemple les machines virtuelles MV1, MV2, MV3 et MVx, référencées 215-1, 215-2, 215-3 et 215-x, respectivement.

Une première machine virtuelle, ici la machine virtuelle 215-1, a un rôle et des droits particuliers. C'est la machine d'administration qui sert d'accès pour la maintenance et la configuration de la plateforme. Elle utilise ici un système d'exploitation OS1. Toujours selon cet exemple, la machine virtuelle 215-1 comprend un espace de stockage, ou mémoire de masse, et une interface de communication, notée I/O (sigle d'*Input/Output* en terminologie anglo-saxonne).

Une deuxième machine virtuelle, ici la machine virtuelle 215-2, permet au STAD de se connecter au système d'information le plus sensible, c'est-à-dire celui de l'aéronef. L'affichage lié à cette machine virtuelle peut être dirigé vers l'écran du STAD ou des écrans spécifiques de l'aéronef selon, par exemple, une interface utilisateur graphique standard de type client/serveur.

La machine virtuelle 215-2 utilise ici le système d'exploitation OS2 et une interface d'entrée/sortie permettant d'échanger des données avec le système d'information de l'aéronef via un lien de type station d'accueil cockpit. La machine virtuelle 215-2 permet d'exécuter des applications opérationnelles. Il convient de définir précisément les périphériques qui sont disponibles dans cet environnement ainsi que les droits d'administration pour assurer le niveau de sûreté requis.

Une troisième machine virtuelle, ici la machine virtuelle 215-3, permet au STAD de se connecter à des systèmes d'information peu sensibles, ici des systèmes distincts du système d'information de l'aéronef. A titre d'illustration, la machine virtuelle 215-3 permet au STAD d'accéder à un réseau interne de l'entreprise exploitant le STAD ou au réseau Internet, par exemple à une borne d'accès WiFi d'un l'hôtel.

Les risques associés à la machine virtuelle 215-3 sont plus élevés que ceux des machines virtuelles 215-1 et 215-2 car elle est ouverte vers des environnements susceptibles d'être source de compromission. Cette machine virtuelle peut donc être une cible pour des logiciels malveillants, appelés *malware* en terminologie anglo-saxonne. En outre, les applications logicielles et

les pilotes mis en œuvre dans cette machine virtuelle étant a priori des logiciels standard, elles représentent des failles potentiellement connues.

Par ailleurs, le concepteur et/ou la société exploitant le STAD peuvent décider de créer une ou plusieurs autres machines virtuelles 215-x pour répondre à des besoins spécifiques. A titre d'illustration, une machine virtuelle 215-x peut être utilisée pour l'exécution d'applications logicielles ayant besoin d'un niveau de sûreté équivalent à celui de la machine virtuelle 215-2 mais dont le fournisseur est différent de celui des applications exécutées dans la machine virtuelle 215-2. Ainsi, en dissociant les environnements d'exécution des applications selon les niveaux de sûreté requis et les responsabilités, il est possible d'utiliser un STAD unique.

De façon avantageuse, le niveau d'exécution est contraint dans l'espace utilisateur au démarrage du STAD de telle sorte que les droits d'administration ne soient pas accessibles. De plus, la bannière de connexion est, de préférence, désactivée pour empêcher un utilisateur de sortir de la couche de virtualisation. Il n'accède ainsi qu'aux machines virtuelles. Toujours de façon avantageuse, la séquence de touches de démarrage est désactivée. Par ailleurs, si des fonctionnalités de virtualisation matérielle sont implémentées dans le processeur du STAD, elles sont configurées de manière à ne pas dégrader le niveau de sûreté attendue.

Au démarrage du STAD ou lors de son activation, par exemple après une mise en veille, l'utilisateur est authentifié. Une telle authentification est ici réalisée par l'hyperviseur. L'accès aux machines virtuelles dépend de cette authentification. Ainsi, par exemple, un pilote et un copilote pourront accéder à toutes les machines virtuelles implémentées sur un STAD, à l'exception de la machine virtuelle d'administration utilisée pour configurer le STAD, tandis qu'un technicien de maintenance pourra accéder à cette dernière. Les machines virtuelles peuvent être lancées automatiquement à la mise en route du STAD ou à la requête de l'utilisateur. Chaque machine virtuelle peut être démarrée et stoppée indépendamment, en fonction des besoins de l'utilisateur. Celui-ci peut passer d'une machine virtuelle à une autre selon un mécanisme standard, par exemple via une interface graphique.

Selon un mode de réalisation particulier, les fonctions mises en œuvre dans les machines virtuelles sont adaptées en fonction de certains paramètres tels que le niveau de sûreté de la machine virtuelle dans laquelle elles sont exécutées.

La figure 3 illustre schématiquement un exemple d'adaptation de certaines fonctions exécutées dans des machines virtuelles selon ces paramètres, en fonction du type de fonction (étape 300).

Ainsi, par exemple, pour une fonction de communication permettant de transmettre des données, les machines virtuelles, ou certaines d'entre elles, passent par une phase d'authentification pour se connecter à certains systèmes externes au STAD (étape 305), par exemple à un système d'information d'un aéronef. Cette phase d'authentification, mise en œuvre par l'hyperviseur lors du lancement de la machine virtuelle et/ou d'échange de données, comprend les étapes suivantes,

- authentification de la machine virtuelle selon un mécanisme standard d'authentification (étape 310) ;
- vérification de l'intégrité de la machine virtuelle en contrôlant, par exemple, plusieurs critères de sûreté tels que le système d'exploitation et la date de dernière mise à jour (étape 315) ; et,
- vérification du niveau d'isolation de la machine virtuelle par rapport aux autres machines virtuelles selon les fonctions mises en œuvre (étape 320). Cette étape permet de vérifier qu'aucune autre machine virtuelle ne peut interagir avec celle dont l'authentification est demandée, par exemple via l'interface utilisateur ou le réseau, lorsque la machine virtuelle est connectée avec un système d'un aéronef. Alternativement, cette étape peut consister à vérifier le respect de règles de communication prédéterminées.

Ainsi, les données ne sont effectivement transmises (étape 325) qu'après authentification de la machine virtuelle afin qu'elle ne puisse transmettre de données erronées vers un système externe.

De même, le transfert de données entre machines virtuelles est contrôlé pour garantir les niveaux de sûreté requis. En effet, qu'elles aient été isolées pour des besoins de limite de responsabilité ou bien de niveaux de

sûreté différents, certaines fonctions peuvent nécessiter un transfert de données entre machines virtuelles et donc le paramétrage d'un canal de communication.

Ainsi, une fonction d'import est créée pour permettre de valider les données transitant vers une machine virtuelle ayant un niveau de sûreté plus élevé que celui de la machine virtuelle source (notée MV* sur la figure 3). Le principe de cette fonction est notamment de filtrer les données selon leur type (étape 330) et d'assurer que seules les données attendues (étape 335) transitent par le canal de communication ouvert (étape 325). Cependant, l'utilisateur reste maître de la validation, c'est-à-dire du transfert effectif des données d'une machine virtuelle vers une autre ayant un niveau de sûreté plus élevé.

Pour améliorer les opérations de maintenance des STADs, un mécanisme de photographies instantanées, appelées *snapshots* en terminologie anglo-saxonne, est, de préférence, mis en œuvre.

La fonction de photographies instantanées permet de mémoriser toutes les données relatives à une machine virtuelle, à un instant donné, de telle sorte que la machine virtuelle puisse être ultérieurement relancée et reconfigurée afin d'être remise dans l'état dans lequel elle se trouvait lorsque les données ont été mémorisées sous forme de photographie. Cette fonction peut être activée par un utilisateur ou l'être automatiquement selon une périodicité prédéterminée, par exemple toutes les semaines ou tous les mois, ou en réponse à des événements particuliers.

En outre, pour garantir un niveau de sûreté des STADs, un mécanisme particulier de gestion d'écriture de données est mis en œuvre dans chaque machine virtuelle. Ce mécanisme est ici basé sur les fonctions de copie d'images lors de l'écriture, appelées *copy-on-write* en terminologie anglo-saxonne, ainsi que d'interdiction d'écriture de certaines données dans la mémoire de masse du STAD.

Selon la fonction d'images copiées lors de l'écriture, toutes les données utilisées par une machine virtuelle lors de son démarrage sont copiées et seule cette copie est utilisée par la machine virtuelle. Ainsi, à chaque

démarrage suivant de la machine virtuelle les mêmes données que celles utilisées pour le démarrage précédent sont utilisées même si ces dernières ont été modifiées par la suite.

La combinaison de ces fonctions permet de garantir les niveaux de sûreté tout en évitant que les STADs aient besoin de davantage de maintenance que des équipements standard. En effet, si une machine virtuelle est compromise, il suffit de la restaurer à partir d'une photographie instantanée réalisée à partir d'un état stable pour retrouver un système fonctionnel.

De façon avantageuse, dans le but de ne pas corrompre la plateforme, seules les opérations nécessaires sont autorisées à écrire des données dans la mémoire de masse du STAD, en fonction du niveau de sûreté de la machine virtuelle (étape 340). Par exemple, seules les données opérationnelles et les données de l'utilisateur expressément identifiées peuvent être stockées dans le STAD. En outre, si le niveau de sûreté de la machine virtuelle est élevé, seules les données dont le niveau de confiance est vérifié peuvent être mémorisées dans le STAD (étapes 345 et 350). Une telle vérification consiste par exemple à valider l'intégrité ou l'origine des données ou à vérifier l'environnement dans lequel elles ont été produites. Néanmoins, même si de telles données peuvent être mémorisées dans le STAD, elles sont de préférence mémorisées dans un support amovible, telle qu'une carte SD (sigle de *Digital Secure* en terminologie anglo-saxonne) ou de type clé USB, pour faciliter le remplacement d'un STAD en évitant l'étape de récupération des données. Les données dont le niveau de confiance n'est pas vérifié peuvent être mémorisées dans un tel support amovible (étape 355).

Un traitement particulier est effectué pour les données que l'utilisateur acquiert à partir d'une machine virtuelle dont le niveau de sûreté est faible, c'est-à-dire dont le niveau de sûreté est inférieur à un seuil prédéterminé, par exemple lorsqu'une machine virtuelle peut accéder à Internet et donc recevoir des messages, des applications et des témoins (couramment appelés *cookies* en terminologie anglo-saxonne) ou lorsque des périphériques tels que des périphériques de stockage peuvent être reliés au STAD. Dans ce cas, pour des raisons opérationnelles et de sûreté, ces données ne peuvent être stockées

que sur un support amovible telle qu'une carte SD ou de type clé USB (étape 355).

Comme indiqué précédemment, chaque machine virtuelle correspond ici à un besoin de niveau de sûreté prédéterminé, les applications, par exemple les applications opérationnelles, les applications bureautiques et les applications personnelles de l'utilisateur étant réparties dans plusieurs machines virtuelles par besoins de niveaux de sûreté selon les fonctions auxquelles elles font appel. Ainsi, pour obtenir un niveau de sécurité général satisfaisant, il est important de configurer avec finesse l'hyperviseur et les machines virtuelles en tenant notamment compte des bons usages, des *drivers* strictement nécessaires et des moyens de communication supportant les fonctions du STAD afin de réduire autant que possible la surface d'attaque et ainsi ne pas dégrader le niveau de sûreté.

La figure 4 illustre schématiquement certaines étapes mises en œuvre pour analyser les risques associés aux fonctions qui doivent être exécutées sur un même STAD. Ces risques permettent de définir les machines virtuelles devant être implémentées dans le STAD et la répartition de ces fonctions dans les machines virtuelles utilisées.

Cette analyse consiste notamment à déterminer les paramètres d'exécution des fonctions afin de déterminer, en particulier, les interfaces de communication pouvant être utilisées, le système d'exploitation utilisé et la provenance des données traitées. Ces paramètres permettent de caractériser les paramètres de la machine virtuelle apte à mettre en œuvre la fonction et de définir un niveau de sûreté.

Il convient de remarquer qu'un niveau de sûreté peut également être directement associé à une fonction, par exemple s'il est imposé.

Une première étape (étape 400) a pour objet l'analyse du contexte afin de déterminer les paramètres permettant d'évaluer le risque de chaque fonction.

Les risques étant ici évalués dans le cas d'une perte de confidentialité, d'intégrité, de disponibilité ou d'authenticité, les paramètres

pouvant être pris en compte pour évaluer les risques peuvent notamment être les suivants,

- la valeur stratégique des informations par rapport à l'activité commerciale ;
- la criticité des informations par rapport à la sécurité des biens et des personnes ;
- les obligations réglementaires et contractuelles ;
- l'importance opérationnelle de la confidentialité, de l'intégrité, de l'authenticité et de la disponibilité des informations traitées ; et,
- les attentes et perception des parties prenantes ainsi que les conséquences sur l'image de marque.

Ces paramètres sont utilisés pour élaborer une grille d'évaluation qui définit des zones pour chaque risque identifié. A titre d'illustration, cette grille peut comprendre trois zones correspondant à des risques faibles, moyens et élevés. Le traitement associé à chaque zone, permettant notamment d'identifier les risques des fonctions devant être mises en œuvre, est de préférence prédéterminé pour permettre la comparaison et la reproductibilité des résultats. En d'autres termes, l'étape 300 permet d'établir le contexte de l'analyse des fonctions devant être mises en œuvre dans un STAD pour répondre à des besoins particuliers.

Dans une seconde étape (étape 405), les risques sont identifiés. Après avoir ici défini la liste des biens à protéger et identifié les responsabilités, les menaces potentielles ainsi que les vecteurs de ces menaces et les moyens de mitigation déjà implémentés sont identifiés. Il convient de remarquer que les listes utilisées doivent être suffisamment détaillées pour permettre une prise de décision quant aux besoins des niveaux de sûreté.

Dans une étape suivante (étape 410), les risques sont estimés en croisant les informations obtenues précédemment avec les vulnérabilités connues ainsi que le type et le niveau de conséquences en cas d'exploitation d'une vulnérabilité et la probabilité d'une attaque. Cette étape permet en particulier d'évaluer les conséquences possibles pour un risque donné selon le

contexte opérationnel et d'établir une liste des risques auxquels peut être soumis une fonction.

Enfin, dans une étape suivante (étape 415), le niveau de risque est évalué pour chaque fonction, en utilisant la grille établie précédemment et les traitements associés. En fonction du résultat obtenu, il peut être décidé de reprendre le processus d'analyse des risques (répétition des étapes 400 à 415) avec une variante d'implémentation afin, par exemple, de réduire le risque associé à une fonction en imposant des contraintes supplémentaires.

Les étapes décrites en référence à la figure 4, appliquées aux fonctions qui doivent être hébergées dans le STAD, permettent de définir les besoins de niveau de sûreté de chaque fonction. Ces résultats, éventuellement combinés avec un critère de responsabilité, permettent de définir les fonctions regroupées dans une même machine virtuelle. Comme indiqué précédemment, l'utilisation d'un critère de responsabilité permet de définir des limites de responsabilité entre les différents intervenants dont les fonctions ou les applications sont implémentées dans le STAD. Par exemple, il est possible de définir une machine virtuelle pour chaque intervenant afin que chacun soit responsable de sa partie.

Ainsi, à titre d'illustration, quatre machines virtuelles peuvent être mises en œuvre selon le schéma suivant,

- machine virtuelle F1 pour l'exécution d'applications ayant un besoin d'un niveau élevé de sûreté. Cette machine virtuelle est isolée des autres pour permettre de suivre la responsabilité du fournisseur 1 ;
- machine virtuelle F2 pour l'exécution d'applications ayant un besoin d'un niveau élevé de sûreté. Cette machine virtuelle est isolée des autres pour permettre de suivre la responsabilité du fournisseur 2 ;
- machine virtuelle F3 pour l'exécution d'applications ayant un besoin d'un niveau moyen de sûreté ; et,
- machine virtuelle F4 pour l'exécution d'applications ayant un besoin d'un faible niveau de sûreté.

Il convient de remarquer que la machine virtuelle d'administration utilisée pour la gestion des autres machines virtuelles, créée par l'hyperviseur

lors de l'installation, a un usage purement technique et non opérationnel. A partir de cette machine virtuelle sont en particulier gérés les droits utilisateurs d'accès aux autres machines.

La figure 5 illustre schématiquement un exemple d'algorithme pour répartir des applications logicielles mises en œuvre dans un STAD dans des machines virtuelles selon les fonctions auxquelles elles font appel.

Une variable *i*, représentant l'index des applications logicielles mises en œuvre dans le STAD, et une variable *j*, représentant l'index des fonctions appelées par l'application ayant l'index *i*, sont initialisées à la valeur un (étape 500). Les fonctions auxquelles fait appel l'application ayant l'index *i* sont déterminées (étape 505). Elles sont ici mémorisées sous forme de table dans la base de données 510.

Le besoin en niveau de sécurité de l'application ayant pour index *i*, appelé BNS(*i*), est alors défini comme étant le besoin en niveau de sécurité de la fonction ayant pour index *j*, appelé BNS(*j*) (étape 515).

Un test est ensuite effectué (étape 520) pour déterminer si la valeur de l'index *j* correspond au nombre de fonctions auxquelles fait appel la fonction ayant pour index *i*. Dans l'affirmative, l'index *j* est incrémenté de un (étape 525) et un autre test est effectué (étape 530) pour déterminer si le besoin en niveau de sécurité de la fonction ayant pour index *j* (BNS(*j*)) est supérieur au besoin en niveau de sécurité de l'application ayant pour index *i* (BNS(*i*)). Dans l'affirmative, le besoin en niveau de sécurité de l'application ayant pour index *i* (BNS(*i*)) est défini comme étant le besoin en niveau de sécurité de la fonction ayant pour index *j* (BNS(*j*)) (étape 535). L'algorithme se poursuit alors à l'étape 520.

Si la valeur de l'index *j* correspond au nombre de fonctions auxquelles fait appel la fonction ayant pour index *i*, l'application ayant pour index *i* est associé à la machine virtuelle dont le niveau de sécurité correspond à BNS(*i*) (étape 540). Cette information est ici mémorisée dans la base de données 545.

La valeur de l'index *i* est ensuite comparée au nombre d'applications mises en œuvre par le STAD pour déterminer si toutes les applications ont été

associées à une machine virtuelle (étape 550). Dans l'affirmative, le processus prend fin. Dans le cas contraire, l'index *i* est incrémenté de un et les étapes précédentes (étapes 505 à 550) sont répétées.

Comme décrit précédemment, il est possible de créer des machines virtuelles indépendamment les unes des autres afin de permettre leur intégration au sein d'un hyperviseur selon des besoins spécifiques. En outre, un tel mode de réalisation permet à différents intervenants d'intégrer leurs applications selon leurs besoins. Etant responsable d'une ou de plusieurs machines virtuelles, chaque intervenant peut ainsi offrir une garantie en terme de sûreté de fonctionnement.

Ainsi, en utilisant l'architecture du STAD et les algorithmes décrits précédemment, un fournisseur d'applications peut livrer une ou plusieurs machines virtuelles intégrant ses applications et communiquer le mode opératoire d'installation de la plateforme logicielle recommandée (à base de virtualisation) sur le STAD. Ce dernier peut aussi être fourni avec l'hyperviseur et la ou les machines virtuelles pré-installées.

De plus, comme indiqué précédemment, des clients peuvent utiliser une clé USB ou une carte SD pour stocker le profil et les données utilisateur tels qu'un identifiant de messagerie, des témoins et des favoris d'un navigateur Internet ainsi que des scripts de connexion. De cette façon, les STADs sont interchangeables et peuvent être gérés en flotte.

De façon avantageuse, une base logicielle commune est installée sur une flotte de STADs, pour tous les usages et tous les types d'aéronefs auxquels les STADs sont susceptibles d'être connectés. La ou les machines virtuelles correspondant aux types d'utilisation et/ou d'aéronefs sont ensuite installées. A la fin de la configuration, une ou plusieurs images de références sont créées pour permettre la configuration du STAD lors de son utilisation. Ces images de référence peuvent en outre être restaurées par l'utilisateur sans recourir à un technicien.

A titre d'illustration, la mallette d'outils connue sous le nom d'*Electronic Flight Bag* (EFB) est ici considérée. Il s'agit d'une mallette d'outils à

partir desquels un pilote et son second préparent une mission à effectuer. L'EFB n'est pas une application ayant un niveau de sûreté élevé.

Il existe essentiellement trois classes d'EFB définies de la façon suivante,

- classe 1 : les données de mission sont chargées dans un STAD portable depuis le sol. Le STAD est emporté à bord de l'aéronef mais il n'est pas connecté au système d'information de bord. Les données sont échangées avec le sol uniquement après l'atterrissage ;

- classe 2 : il s'agit d'un STAD portable standard, attaché à un pilote. Le STAD échange des données avec le sol, notamment la compagnie aérienne et l'aéroport, et avec le système d'information de l'aéronef pendant toutes les phases de vol ; et,

- classe 3 : le STAD est solidaire au cockpit et peut accéder aux systèmes critiques qui requièrent un haut niveau de certification.

Les EFBs de classe 2 peuvent donc être gérés par un STAD conforme à l'invention. Dans ce cas, le fournisseur d'EFBs livre les applications opérationnelles ou une partie d'entre elles au client. Il lui communique la liste des systèmes compatibles et la configuration minimale de l'hyperviseur ou livre le STAD avec l'hyperviseur et les machines virtuelles pré-installées.

Durant une phase de préparation initiale, le client prépare les machines virtuelles selon les applications opérationnelles et les données de configuration reçues. Il prépare également les machines virtuelles permettant de mettre en œuvre les applications spécifiques de l'entreprise, par exemple des applications de messagerie.

Ainsi, en utilisant les machines virtuelles créées, un technicien informatique peut installer l'hyperviseur du STAD qui permet alors d'utiliser les moyens informatiques de l'entreprise, les fonctionnalités de l'EFB et, éventuellement, d'autres.

Pour que les STADs soient interchangeables, les données des utilisateurs sont de préférence stockées sur une carte mémoire amovible, par exemple une carte de type SD et non sur le disque dur interne des STADs.

En utilisation, lorsqu'une compromission est détectée sur une machine virtuelle, par exemple une machine virtuelle qui s'est connectée à un site Internet, l'utilisateur peut redémarrer la machine virtuelle en utilisant l'image de référence associée la plus récente sans recourir à un technicien.

De même, en cas de panne matérielle d'un STAD, le technicien récupère la carte mémoire de l'utilisateur et l'insère dans un autre STAD de la flotte. Il n'a pas besoin de personnaliser ce STAD ni de copier des données de l'utilisateur.

La figure 6 illustre un exemple d'architecture matérielle adaptée à mettre en œuvre l'invention. Le dispositif 600 comporte ici un bus de communication 605 auquel sont reliés :

- une unité centrale de traitement ou microprocesseur 610 (CPU, sigle de *Central Processing Unit* en terminologie anglo-saxonne) ;
- une mémoire morte 615 (ROM, acronyme de *Read Only Memory* en terminologie anglo-saxonne) pouvant comporter les programmes nécessaires à la mise en œuvre de l'invention ;
- une mémoire vive ou mémoire cache 620 (RAM, acronyme de *Random Access Memory* en terminologie anglo-saxonne) comportant des registres adaptés à enregistrer des variables et paramètres créés et modifiés au cours de l'exécution des programmes précités ; et
- une interface de communication 650 adaptée à transmettre et à recevoir des données.

Le dispositif 600 dispose également, de préférence, des éléments suivants :

- un écran 625 permettant de visualiser des données telles que des représentations des commandes et de servir d'interface graphique avec l'utilisateur qui pourra interagir avec les programmes selon l'invention, à l'aide d'un clavier et d'une souris 630 ou d'un autre dispositif de pointage tel qu'un écran tactile ou une télécommande ;
- d'un disque dur 635 pouvant comporter les programmes précités et des données traitées ou à traiter selon l'invention ; et

- d'un lecteur de cartes mémoires 640 adapté à recevoir une carte mémoire 645 et à y lire ou à y écrire des données traitées ou à traiter selon l'invention.

Le bus de communication permet la communication et l'interopérabilité entre les différents éléments inclus dans le dispositif 600 ou reliés à lui. La représentation du bus n'est pas limitative et, notamment, l'unité centrale est susceptible de communiquer des instructions à tout élément du dispositif 600 directement ou par l'intermédiaire d'un autre élément du dispositif 600.

Le code exécutable de chaque programme permettant au dispositif programmable de mettre en œuvre les processus selon l'invention, peut être stocké, par exemple, dans le disque dur 635 ou en mémoire morte 615.

Selon une variante, la carte mémoire 645 peut contenir des données, notamment une table de correspondance entre les évènements détectés et les commandes pouvant être sollicitées, ainsi que le code exécutable des programmes précités qui, une fois lu par le dispositif 600, est stocké dans le disque dur 635.

Selon une autre variante, le code exécutable des programmes pourra être reçu, au moins partiellement, par l'intermédiaire de l'interface 650, pour être stocké de façon identique à celle décrite précédemment.

De manière plus générale, le ou les programmes pourront être chargés dans un des moyens de stockage du dispositif 600 avant d'être exécutés.

L'unité centrale 610 va commander et diriger l'exécution des instructions ou portions de code logiciel du ou des programmes selon l'invention, instructions qui sont stockées dans le disque dur 635 ou dans la mémoire morte 615 ou bien dans les autres éléments de stockage précités. Lors de la mise sous tension, le ou les programmes qui sont stockés dans une mémoire non volatile, par exemple le disque dur 635 ou la mémoire morte 615, sont transférés dans la mémoire vive 620 qui contient alors le code exécutable du ou des programmes selon l'invention, ainsi que des registres pour

mémoriser les variables et paramètres nécessaires à la mise en œuvre de l'invention.

Naturellement, pour satisfaire des besoins spécifiques, une personne compétente dans le domaine de l'invention pourra appliquer des modifications dans la description précédente.

REVENDICATIONS

1. Composant logiciel pour ordinateur adapté au traitement automatique de données multi-usages, le composant logiciel étant caractérisé en ce qu'il met en œuvre des fonctions ayant besoin de différents niveaux de sûreté ou limites de responsabilité et en ce qu'il comprend,

- une pluralité de machines virtuelles (215), chaque machine virtuelle étant adaptée à exécuter au moins une fonction ayant besoin d'un niveau de sûreté ou d'une limite de responsabilité prédéterminé, au moins une desdites fonctions étant adaptée selon des paramètres de la machine virtuelle dans laquelle elle est exécutée ; et,

- un hyperviseur (210) adapté à contrôler l'exécution de ladite pluralité de machines virtuelles.

2. Composant logiciel selon la revendication précédente selon laquelle ledit hyperviseur comprend des moyens d'authentification pour authentifier (310) au moins une machine virtuelle de ladite pluralité de machine virtuelle.

3. Composant logiciel selon la revendication précédente selon lequel lesdits moyens d'authentification sont adaptés à vérifier l'intégrité (315) de ladite au moins une machine virtuelle authentifiée.

4. Composant logiciel selon l'une quelconque des revendications 2 et 3 selon lequel lesdits moyens d'authentification sont adaptés à vérifier le niveau d'isolation (320) de ladite au moins une machine virtuelle authentifiée par rapport à au moins une autre machine virtuelle de ladite pluralité de machines virtuelles.

5. Composant logiciel selon l'une quelconque des revendications précédentes comprenant en outre des moyens de mémorisation de données traitées par au moins une machine virtuelle de ladite pluralité de machine virtuelle, lesdits moyens de mémorisation étant adaptés à mémoriser lesdites données traitées dans une mémoire amovible dudit ordinateur (355).

6. Composant logiciel selon la revendication précédente selon lequel lesdits moyens de mémorisation de données sont mis en œuvre par des machines virtuelles de ladite pluralité de machines virtuelles dont le niveau de sûreté est inférieur à un seuil prédéterminé (340).

7. Composant logiciel selon l'une quelconque des revendications précédentes comprenant en outre des moyens de contrôle d'un niveau de confiance (345) d'au moins une donnée traitée par au moins une machine virtuelle de ladite pluralité de machines virtuelles, ladite au moins une donnée traitée ne pouvant être mémorisée localement (350) dans ledit ordinateur qu'après qu'elle ait été contrôlée.

8. Composant logiciel selon l'une quelconque des revendications précédentes comprenant en outre des moyens de transfert de données entre une première et une seconde machines virtuelles de ladite pluralité de machines virtuelles, lesdits moyens de transfert étant adaptés à filtrer (330, 335) des données transférées si le niveau de sûreté de ladite seconde machine virtuelle est supérieur au niveau de sûreté de ladite première machine virtuelle.

9. Composant logiciel selon l'une quelconque des revendications précédentes selon lequel des données de configuration utilisées pour démarrer au moins une machine virtuelle de ladite pluralité de machine virtuelle ne sont pas modifiées au cours de l'exécution de ladite au moins une machine virtuelle démarrée.

10. Dispositif comprenant des moyens adaptés à la mise en œuvre de chacun des éléments du composant logiciel selon l'une quelconque des revendications précédentes.

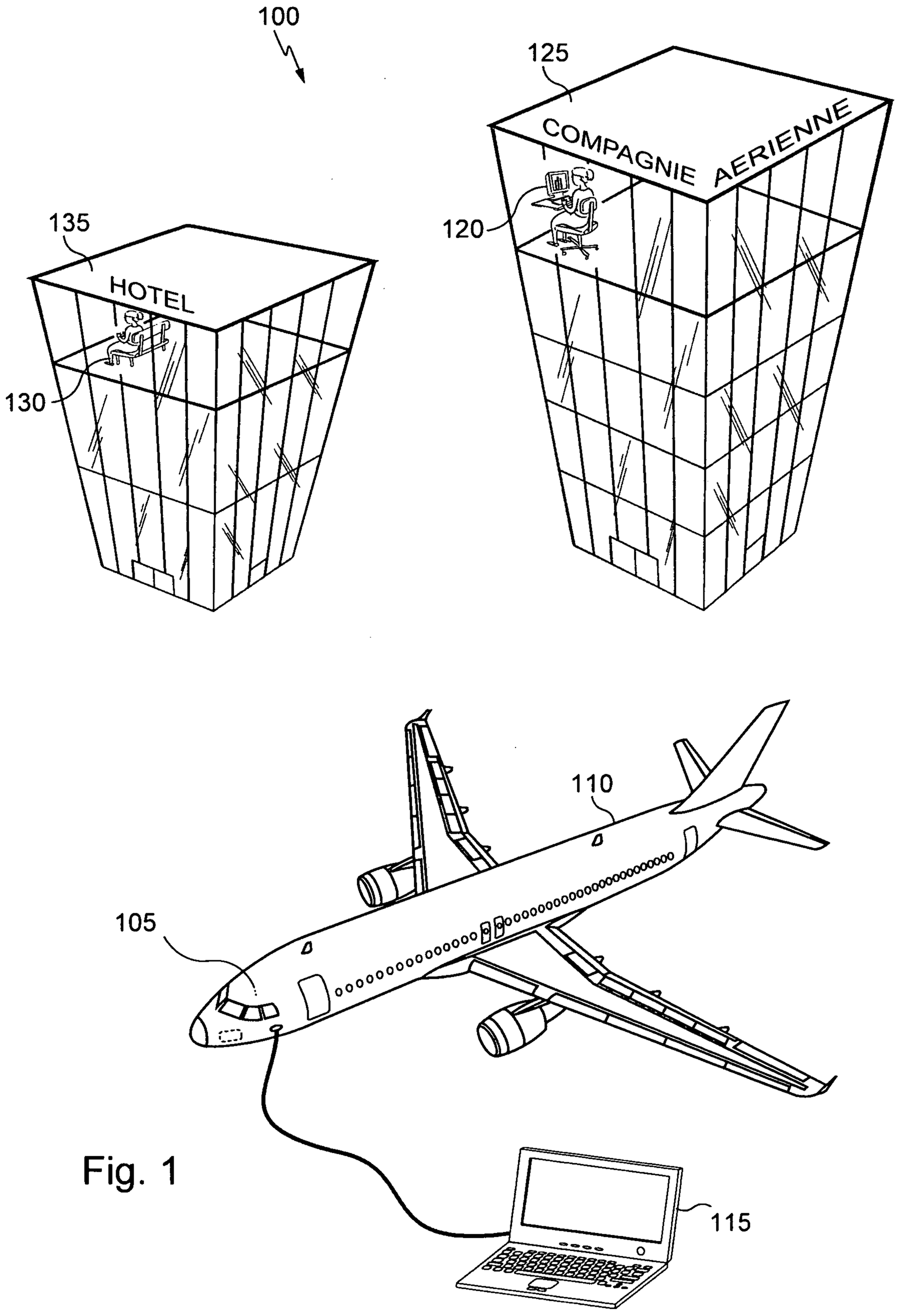

Fig. 1

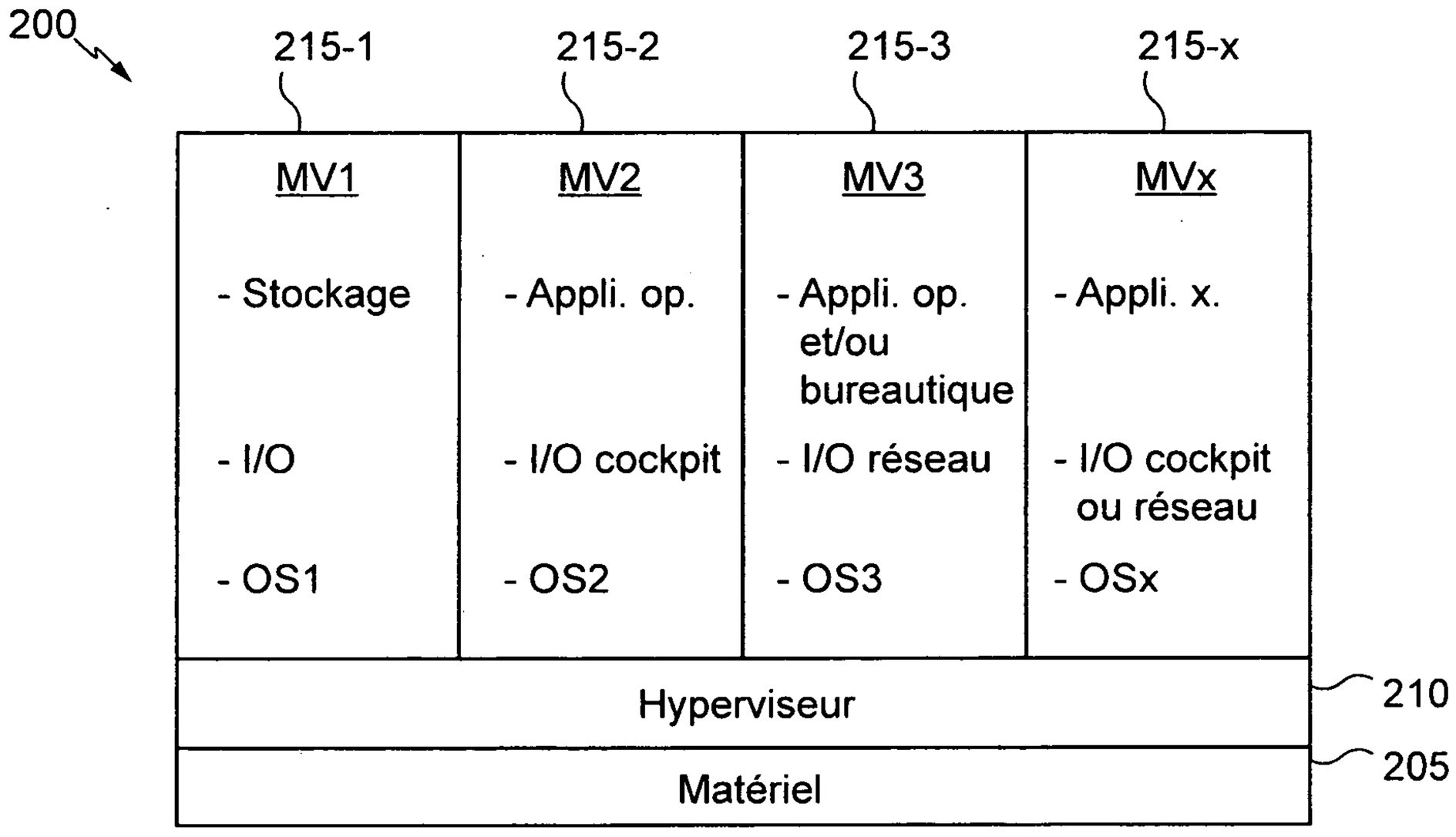
200
215-1
215-2
215-3
215-x
MV1
MV2
MV3
MVx
- Stockage
- Appli. op.
- Appli. op. et/ou bureautique
- Appli. x.
- I/O
- I/O cockpit
- I/O réseau
- I/O cockpit ou réseau
- OS1
- OS2
- OS3
- OSx
Hyperviseur
210
Matériel
205

Fig. 2

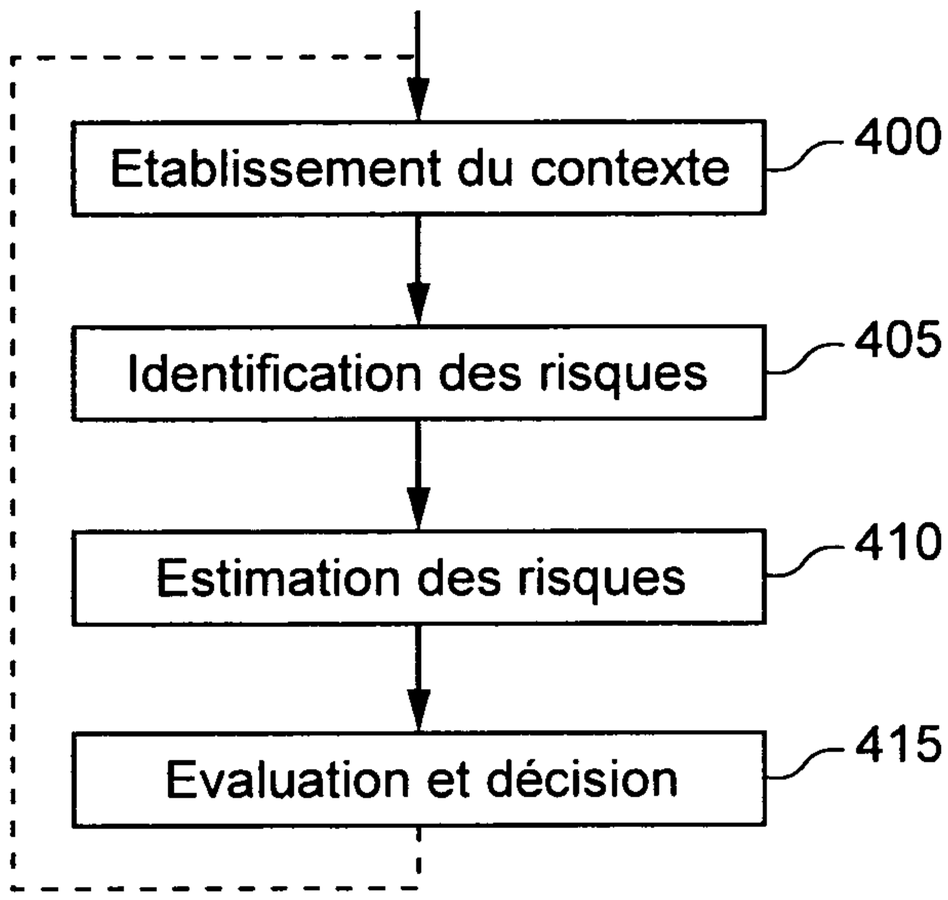
Etablissement du contexte
400
Identification des risques
405
Estimation des risques
410
Evaluation et décision
415

Fig. 4

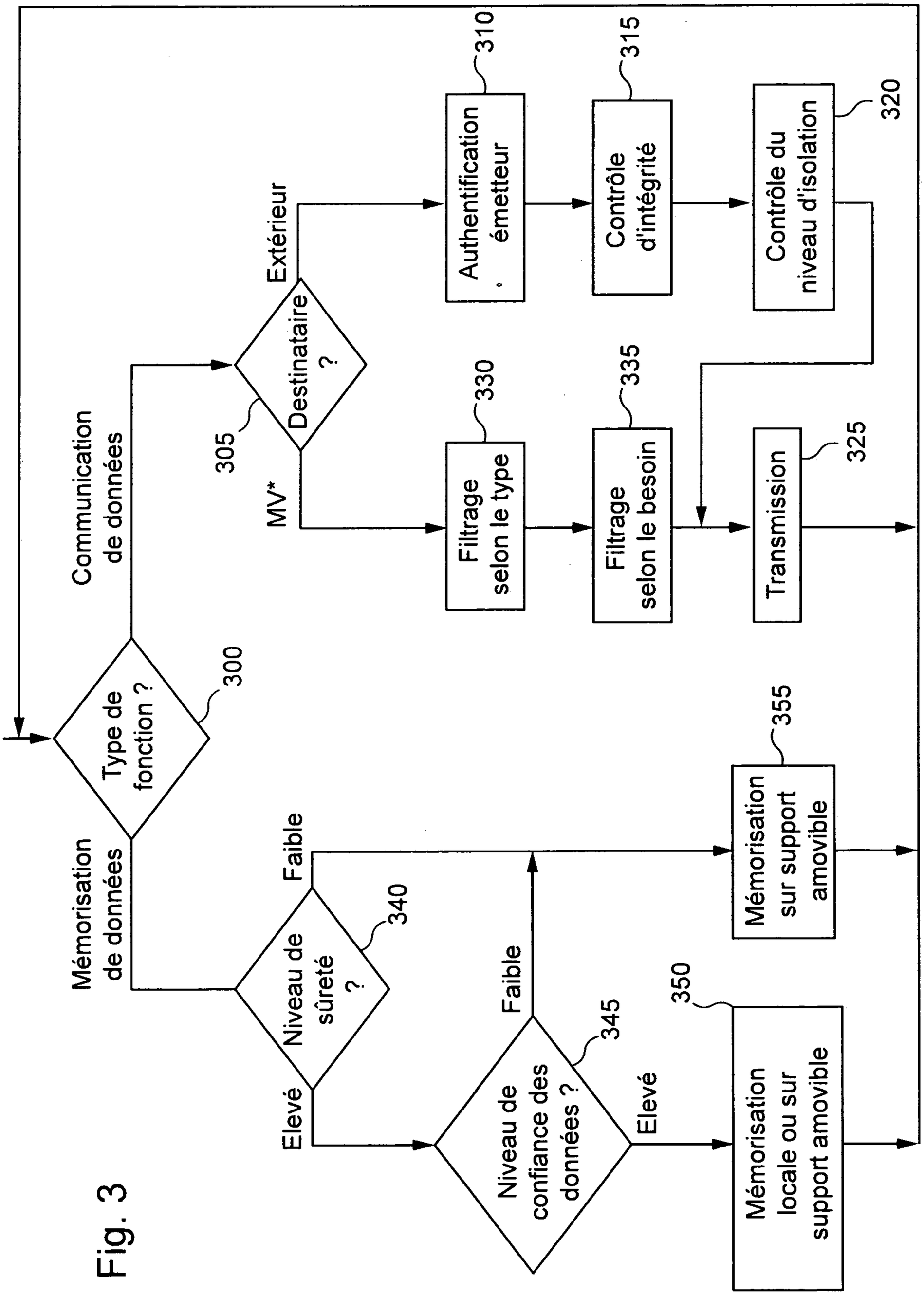
Communication de données
Mémorisation de données
Type de fonction ?
300
Destinataire ?
305
Extérieur
MV*
Authentification émetteur
310
Contrôle d'intégrité
315
Contrôle du niveau d'isolation
320
Filtrage selon le type
330
Filtrage selon le besoin
335
Transmission
325
Niveau de sûreté ?
340
Faible
Elevé
Mémorisation sur support amovible
355
Niveau de confiance des données ?
345
Faible
Elevé
Mémorisation locale ou sur support amovible
350

Fig. 3

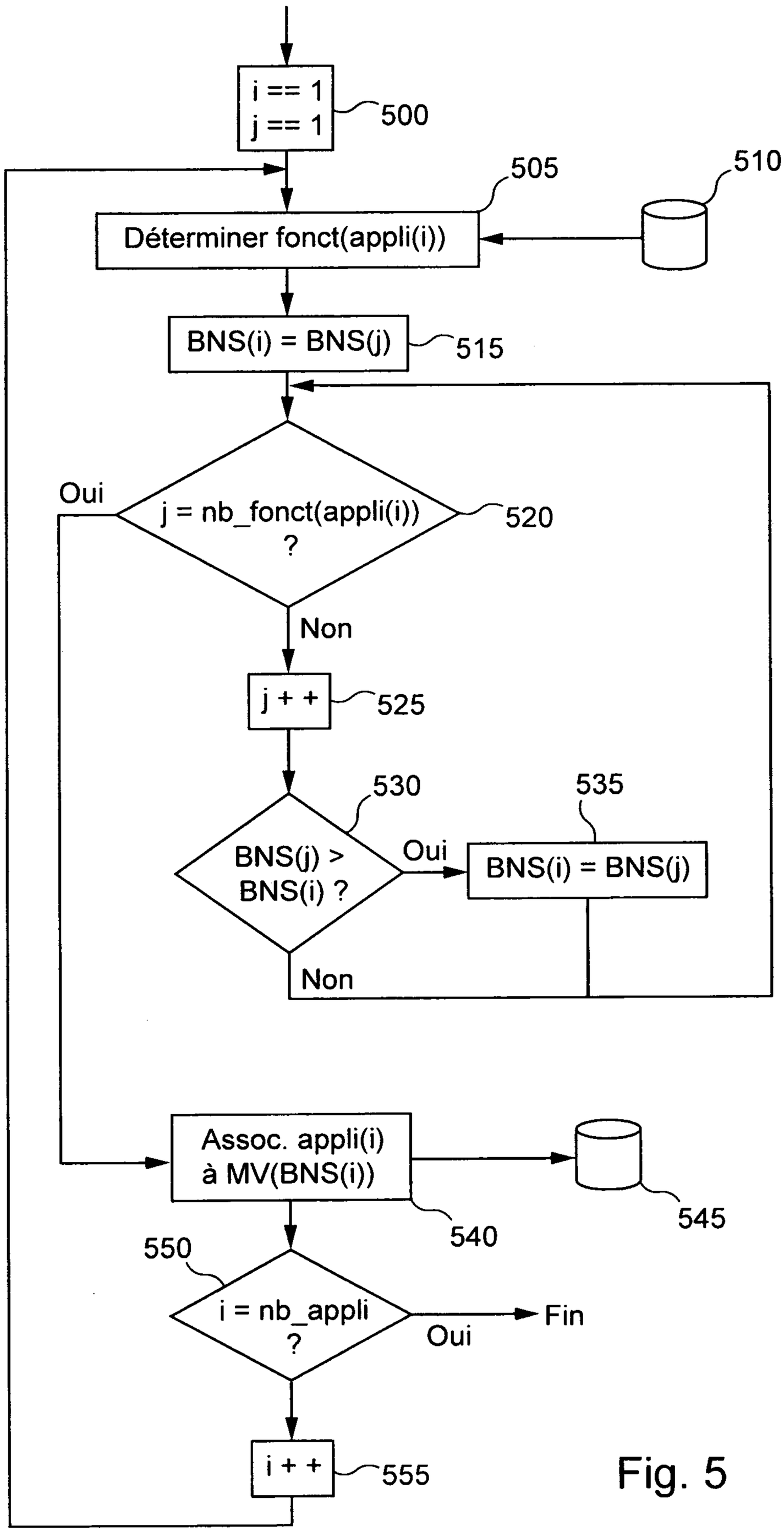
i == 1
j == 1
500
505
510
Déterminer fonct(appli(i))
BNS(i) = BNS(j)
515
Oui
j = nb_fonct(appli(i))
?
520
Non
j + +
525
530
BNS(j) >
BNS(i) ?
Oui
535
BNS(i) = BNS(j)
Non
Assoc. appli(i)
à MV(BNS(i))
540
545
550
i = nb_appli
?
Oui
Fin
i + +
555

Fig. 5

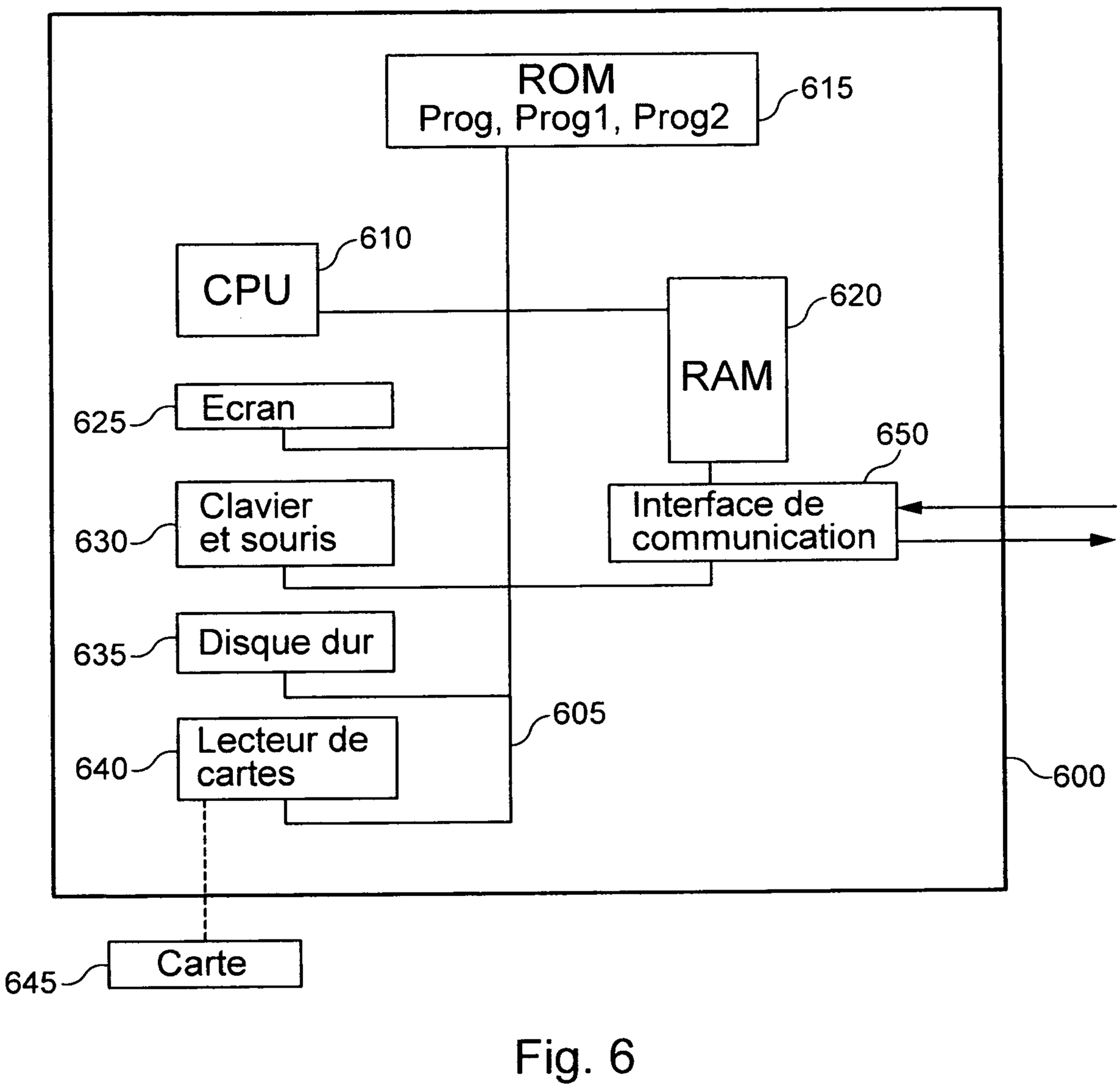
ROM
Prog, Prog1, Prog2
615
CPU
610
RAM
620
Ecran
625
650
Clavier
et souris
630
Interface de
communication
Disque dur
635
605
Lecteur de
cartes
640
600
Carte
645

Fig. 6