

19 RÉPUBLIQUE FRANÇAISE
INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE
COURBEVOIE

11 N° de publication :
(à n'utiliser que pour les
commandes de reproduction)

3 141 830

21 N° d'enregistrement national : 22 11654

51 Int Cl⁸ : H 04 L 67/12 (2023.01), G 07 C 5/08

12

DEMANDE DE BREVET D'INVENTION

A1

22 Date de dépôt : 09.11.22.

30 Priorité :

43 Date de mise à la disposition du public de la demande : 10.05.24 Bulletin 24/19.

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

60 Références à d'autres documents nationaux apparentés :

○ Demande(s) d'extension :

71 Demandeur(s) : ACTIA Group Société Anonyme à Conseil d'Administration — FR.

72 Inventeur(s) : BERTRAND Antoine, RAVAYROL Christophe, DUREY Sylvain, NACEUR Firas et JMII Walid.

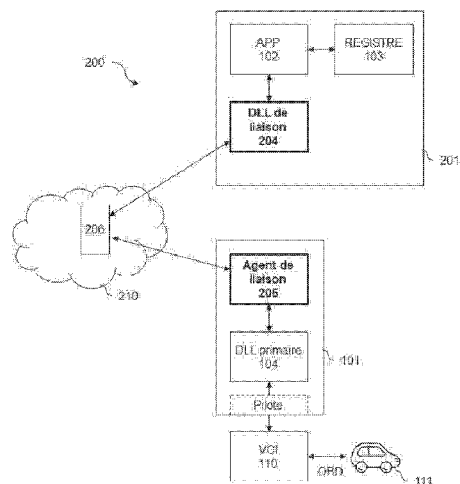
73 Titulaire(s) : ACTIA Group Société Anonyme à Conseil d'Administration.

74 Mandataire(s) : SANTARELLI.

54 DIAGNOSTIC DE VEHICULE A DISTANCE A L'AIDE D'APPLICATIONS DE DIAGNOSTIC STANDARDISEES NON MODIFIEES.

57 Dans le domaine du diagnostic automobile, l'invention propose des équipements et procédés de diagnostic véhicule à distance à l'aide d'applications de diagnostic véhicule conforme à la norme SAE J2534 ou ISO 22900-2:2022 D-PDU ou TMC RP1210. Une application de diagnostic conforme à SAE J2534 équipe un dispositif distant (201) alors que la DLL SAE J2534 primaire est préinstallée dans un dispositif (101) local au véhicule (111) inspecté. Pour conserver l'application et la DLL primaire sans modification par rapport à leur usage standardisé en local, l'invention prévoit de modifier l'entrée du registre Windows du dispositif distant associée normalement à la DLL primaire pour identifier une nouvelle DLL, dite de liaison, configurée pour router, via le réseau de communication (210) et un agent logiciel de liaison (205) du dispositif local (101), un flux de données entre ladite application de diagnostic et la DLL SAE J2534 de l'interface VCI (110) reliée au véhicule.

Figure pour l'abrégié : Fig.2



FR 3 141 830 - A1



Description

Titre de l'invention : DIAGNOSTIC DE VEHICULE A DISTANCE A L'AIDE D'APPLICATIONS DE DIAGNOSTIC STAN- DARDISEES NON MODIFIEES

DOMAINE DE L'INVENTION

[0001] L'invention concerne le domaine du diagnostic et de la maintenance automobile et en particulier le diagnostic à distance à l'aide d'applications de diagnostic véhicule conformes à certaines normes, telles la norme SAE J2534 de décembre 2004 aussi connue sous l'appellation de norme « Passthru ».

TECHNIQUES ANTERIEURES

[0002] Nombre de fonctions d'un véhicule sont pilotées par des Unités de Contrôle Electroniques (ECU) embarquées, autrement appelées « calculateurs ». Ceux-ci communiquent entre eux via un bus de communication embarqué. Les protocoles de communication, dont le CAN – Controller Area Network – est un exemple, varient d'un constructeur à l'autre.

[0003] Les calculateurs ECU peuvent être analysés, configurés, programmés ou re-programmés lors d'opérations de réparation et d'entretien des véhicules (RMI – Repair and Maintenance Information). Chaque constructeur a ainsi développé son propre logiciel ou « application de diagnostic véhicule » qui communique avec un boîtier relié au véhicule (via sa prise OBD - On Board Diagnostic), connu sous l'appellation d'interface de communication des véhicules (VCI – Vehicle Communication Interface), pour permettre un diagnostic de ses véhicules.

[0004] Néanmoins, afin que tous les opérateurs indépendants, tels les revendeurs agréés, réparateurs et garagistes, puissent intervenir sur ces calculateurs, des normes ont été érigées qui définissent les interfaces de programmation d'application (API) que ces applications de diagnostic véhicule et interfaces VCI doivent respecter pour communiquer entre elles. Les principales normes comprennent la SAE J2534 (notamment spécification J2534-1 de décembre 2004), ISO 22900-2:2022 D-PDU et TMC RP1210 (septembre 2006).

[0005] Par ailleurs, de nombreux organismes, tels la Commission Européenne, ont légiféré pour obliger les constructeurs à mettre à disposition des opérateurs indépendants une version de leur application logicielle de diagnostic véhicule. On connaît notamment le règlement (CE) n° 715/2007 (Euro 5/6) et ses règlements d'application n°692/2008 et n°566/2011 (ensemble « norme Euro 5 ») qui ont défini ces contraintes faites aux constructeurs automobiles à fournir les informations techniques indispensables au diagnostic, à la réparation, à l'entretien et à la maintenance de leurs véhicules, y

compris la possibilité de programmer ou reprogrammer les calculateurs.

- [0006] De fait, les opérateurs indépendants multimarques peuvent utiliser les applications de diagnostic véhicule de multiples constructeurs, généralement à titre onéreux pour des périodes ou utilisations limitées. Néanmoins, cette situation présente de nombreuses difficultés, dont la nécessité d'ouvrir et de gérer autant de compte client que de constructeurs, mais également le besoin d'utiliser plusieurs ordinateurs lorsque ces applications de diagnostic véhicule ne sont pas compatibles entre elles. En outre, il est un besoin de se former à toutes ces applications utilisées, ainsi que de veiller à leur mise à jour régulière. Il est alors classique que ces opérateurs indépendants se tournent vers des applications de diagnostic véhicule multimarques dont les services sont réduits ou dégradés comparativement aux applications constructeurs, voire sous-traitent l'opération de diagnostic chez un réparateur officiel du constructeur.
- [0007] Devant ce constat, il a été proposé des systèmes de diagnostic à distance de véhicule comprenant un centre d'assistance (hotline) tiers qui contrôle et dirige, via un réseau de communication, une opération spécifique de diagnostic à distance. La publication EP3690579 illustre un tel système.
- [0008] Ces systèmes permettent avantageusement à l'opérateur indépendant de conserver le véhicule à réparer dans son atelier, tout en réduisant les coûts et la gestion grâce à la mutualisation entre plusieurs opérateurs indépendants souscrivant au système.
- [0009] Ces systèmes requièrent cependant le développement d'applications de diagnostic véhicule à distance spécifiques, par exemple en modifiant le code des applications conformes aux normes ci-dessus afin d'y intégrer les fonctionnalités additionnelles telles celles assurant les communications via le réseau de communication, typiquement l'Internet. Cela garantirait aux opérateurs indépendants que l'application officielle de diagnostic véhicule du constructeur est utilisée pour la réalisation de l'opération de diagnostic.
- [0010] Cette situation n'est néanmoins pas entièrement satisfaisante car il est nécessaire d'attendre que chaque constructeur accepte d'implémenter de telles modifications. Il serait en revanche préférable de permettre la réutilisation des applications conformes aux normes et des interfaces VCI correspondantes également conformes aux normes, sans avoir à modifier leur code.

Exposé de l'invention

- [0011] Les inventeurs ont constaté que ces normes requièrent l'installation, sur le même ordinateur que l'application de diagnostic véhicule conforme à une des normes, d'une bibliothèque de liens dynamiques primaire (ou DLL pour « Dynamic Link Library »), elle-même conforme à la norme, qui est dédiée à l'interface VCI. L'application de diagnostic véhicule et la bibliothèque de liens dynamiques primaire communiquent

ainsi via l'API de la norme mise en place.

[0012] Ces normes requièrent également la mise à jour de la base de registre Windows (nom commercial) de l'ordinateur pour y indiquer l'emplacement local de cette bibliothèque de liens dynamiques primaire permettant son chargement par l'application et ainsi la communication entre l'application de diagnostic véhicule et l'interface VCI, soit directement soit via un pilote de l'interface VCI également installé sur l'ordinateur local au véhicule inspecté.

[0013] La présente invention vise à résoudre l'inconvénient susmentionné en permettant de contourner ces limitations des normes mentionnées, afin de permettre une utilisation distante des applications de diagnostic véhicule et interfaces VCI standardisées déjà existantes sans modification du code binaire des composants existants.

[0014] Dans ce dessein, l'invention concerne un dispositif de diagnostic à distance de véhicule, comportant :

- une interface de communication avec un réseau de communication, et
- une application de diagnostic véhicule conforme à l'un des standards SAE J2534, ISO 22900-2 D-PDU et TMC RP1210 et apte à communiquer avec une interface de communication véhicule, VCI, par l'utilisation d'une bibliothèque de liens dynamiques, DLL, primaire associée à l'interface VCI et conforme audit standard, ladite application de diagnostic étant configurée pour utiliser une entrée de registre de système d'exploitation (OS) afin d'identifier la DLL primaire,

[0015] caractérisé en ce que le dispositif de diagnostic à distance comporte une DLL de liaison apte à router, via le réseau de communication, un flux de données (e.g. un message) entre ladite application de diagnostic véhicule et une DLL primaire distante configurée pour communiquer localement avec une interface VCI reliée au véhicule, ladite entrée de registre étant configurée pour identifier la DLL de liaison en lieu et place d'une DLL primaire associée à l'interface VCI.

[0016] L'invention prévoit donc de détourner l'usage du registre Windows utilisé par l'application de diagnostic véhicule pour y indiquer une DLL réalisant les opérations de communication, et non plus la DLL primaire conventionnelle. Ainsi, l'application de diagnostic véhicule peut être déportée sur un dispositif distant sans modification ni de celle-ci, ni de la DLL primaire.

[0017] L'invention concerne également un système de diagnostic à distance de véhicule, comportant :

- un dispositif de diagnostic à distance connecté à un réseau de communication et comprenant une application de diagnostic véhicule conforme à l'un des standards SAE J2534, ISO 22900-2 D-PDU et TMC RP1210 et apte à communiquer avec une interface de communication véhicule, VCI, par

- l'utilisation d'une bibliothèque de liens dynamiques, DLL, primaire associée à l'interface VCI et conforme audit standard, ladite application de diagnostic étant configurée pour utiliser une entrée de registre de système d'exploitation (OS) afin d'identifier la DLL primaire, et
- un dispositif local au véhicule, connecté au réseau de communication et comprenant une DLL primaire configurée pour communiquer localement avec une interface VCI reliée à un véhicule à diagnostiquer,
- [0018] caractérisé en ce que le dispositif de diagnostic à distance comporte une DLL de liaison apte à router, via le réseau de communication et un agent logiciel de liaison du dispositif local, un flux de données entre ladite application de diagnostic véhicule et la DLL primaire de l'interface VCI reliée au véhicule, ladite entrée de registre étant configurée pour identifier la DLL de liaison en lieu et place d'une DLL primaire associée à l'interface VCI.
- [0019] Pour sa part, l'ajout, au niveau du dispositif local, d'un agent logiciel de liaison réalisant les opérations inverses de la bibliothèque de liaison permet également de conserver la DLL primaire de l'interface VCI sans modification aucune.
- [0020] L'invention concerne également un procédé de diagnostic à distance d'un véhicule, comportant, au niveau d'un dispositif de diagnostic à distance d'un dispositif local audit véhicule, les étapes suivantes :
- obtenir une application de diagnostic véhicule conforme à l'un des standards SAE J2534, ISO 22900-2 D-PDU et TMC RP1210 et apte à communiquer avec une interface de communication véhicule, VCI, par l'utilisation d'une bibliothèque de liens dynamiques, DLL, primaire associée à l'interface VCI et conforme audit standard, ladite application de diagnostic étant configurée pour utiliser une entrée de registre de système d'exploitation (OS) afin d'identifier la DLL primaire,
 - configurer l'entrée de registre pour identifier, en lieu et place d'une DLL primaire associée à l'interface VCI, une DLL de liaison apte à router, via un réseau de communication, un flux de données entre ladite application de diagnostic véhicule et une DLL primaire distante configurée pour communiquer localement avec l'interface VCI reliée au véhicule, et
 - charger la DLL de liaison lors d'une exécution de l'application de diagnostic véhicule de sorte qu'un message de diagnostic généré par l'application de diagnostic véhicule soit routé sur le réseau de communication vers la DLL primaire distante et l'interface VCI reliée au véhicule.
- [0021] Des caractéristiques facultatives des modes de réalisation de l'invention sont définies dans les revendications annexées. Certaines de ces caractéristiques sont expliquées ci-dessous en référence à un dispositif ou à un procédé, tandis qu'elles peuvent être

transposées en caractéristiques de procédé ou de dispositif respectivement.

- [0022] Dans un mode de réalisation, la DLL de liaison et/ou l'agent logiciel de liaison comprend un convertisseur apte à encapsuler le flux de données en une ou plusieurs trames de données structurées, par exemple JSON (JavaScript Object Notation) ou de balisage tel XML (eXtensible Markup Language) ou HTML (HyperText Markup Language), ou à désencapsuler une ou plusieurs trames de données structurées en ledit flux de données, de sorte à échanger des trames de données structurées sur le réseau de communication, typiquement l'Internet. Ce format simplifié, type JSON, permet de réduire le volume de données échangées, et ainsi accélérer le temps de traitement des opérations de diagnostic compte tenu de la distance.
- [0023] Ainsi, le procédé peut comprendre une étape de conversion du message de diagnostic par encapsulation en une ou plusieurs trames de données structurées avant routage vers l'interface VCI via le réseau de communication.
- [0024] Dans un autre mode de réalisation, un identifiant d'interface VCI est ajouté dans la trame de données structurées de sorte à router ladite trame vers une DLL primaire (via un agent de liaison dédié ou commun à plusieurs DLL, et par conséquent router la trame vers une interface VCI souhaitée) correspondant audit identifiant parmi plusieurs DLL primaires accessibles sur le réseau de communication. Cela permet d'assurer le bon contrôle à distance d'une VCI parmi un parc potentiel de plusieurs VCIs (car plusieurs utilisateurs).
- [0025] Dans un autre mode de réalisation, la DLL de liaison et l'agent logiciel de liaison sont configurés pour établir un canal de communication de type web socket entre eux, sur le réseau de communication. Cette disposition permet d'accélérer le temps de traitement des opérations de diagnostic compte tenu de la distance.
- [0026] Dans un autre mode de réalisation, la DLL de liaison et/ou l'agent logiciel de liaison comprend un filtre de routage configuré pour déterminer, en fonction d'un critère de filtrage, si un message reçu pour routage est à supprimer sans le router. De la sorte, certaines trames protocolaires inutiles peuvent être supprimées, réduisant les temps de latence des échanges entre l'application et la DLL primaire distante.
- [0027] Ainsi, le procédé peut comprendre une étape de filtrage de messages reçus pour routage comprenant la détermination, en fonction d'un critère de filtrage, si un message reçu est à supprimer et dans l'affirmative, la suppression dudit message sans le router.
- [0028] Selon une configuration statique, le critère de filtrage supprime un message de type VBAT relatif à une interrogation sur le niveau de batterie du véhicule, de type FwVersion relatif à une interrogation sur la version de firmware embarqué dans l'interface VCI, de type DllVersion relatif à une interrogation sur la version de la DLL primaire et/ou de type ApiVersion relatif à une interrogation sur la version d'API mise

en œuvre par la DLL primaire.

- [0029] Selon une configuration dynamique, le filtre de routage est configuré pour déterminer un type de message routé de façon récurrente et, en réponse, pour configurer ledit critère de filtrage de sorte à supprimer tout message ultérieur dudit type sur au moins une période prédéfinie. Par « de façon récurrente », il faut comprendre un message dont la fréquence moyenne de transmission est supérieure à une valeur seuil, par exemple plus de dix transmissions du message de même type par minute (fenêtre temporelle glissante).
- [0030] Dans un mode de réalisation, l'interface VCI reliée au véhicule forme un dispositif distinct du dispositif local. Cette disposition permet aux opérateurs indépendants de conserver leur architecture informatique actuelle, impliquant la présence d'un ordinateur auquel l'interface VCI est reliée (généralement par câble). Dans un mode de réalisation particulier, le dispositif local comporte un pilote associé à l'interface VCI pour router les données entre la DLL primaire et l'interface VCI externe au dispositif local. Le pilote permet notamment de s'adapter à différents jeux de configuration mis en œuvre dans le Firmware embarqué sur la VCI.
- [0031] Dans un autre mode de réalisation, le dispositif local forme l'interface VCI reliée au véhicule. Cette disposition permet aux opérateurs indépendants de s'affranchir de l'utilisation d'un ordinateur additionnel.
- [0032] La présente invention vise également un programme informatique comportant des instructions pour la mise en œuvre du procédé ci-dessus, lorsque ce programme est exécuté par un processeur. Ce programme peut utiliser n'importe quel langage de programmation (par exemple, un langage objet ou autre), et être sous la forme d'un code source interprétable, d'un code partiellement compilé ou d'un code totalement compilé.
- [0033] L'invention vise également un support d'enregistrement non transitoire lisible par un ordinateur sur lequel est enregistré un programme pour la mise en œuvre du procédé ci-dessus, lorsque ce programme est exécuté par un processeur.
- [0034] Au moins une partie des procédés selon l'invention peut être mise en œuvre par ordinateur. En conséquence, la présente invention peut prendre la forme d'un mode de réalisation entièrement matériel, d'un mode de réalisation entièrement logiciel (comportant les microprogrammes, les logiciels résidents, les microcodes, etc.) ou d'un mode de réalisation combinant des aspects logiciels et matériels qui peuvent tous être globalement appelés ici "circuit", "module" ou "système". De plus, la présente invention peut prendre la forme d'un produit de programme informatique incorporé dans tout support d'expression tangible disposant d'un code de programme utilisable par ordinateur incorporé dans le support.

Breve description des dessins

- [0035] D'autres particularités et avantages de l'invention apparaîtront encore dans la description ci-après, illustrée par les figures ci-jointes qui en illustrent des exemples de réalisation dépourvus de tout caractère limitatif.
- [0036] La [Fig.1] illustre schématiquement l'architecture d'un système de diagnostic véhicule selon des standards connus.
- [0037] La [Fig.1A] illustre des entrées de la base de registre Windows conformément à la norme SAE J2534.
- [0038] La [Fig.1B] illustre un procédé de chargement et déchargement d'une DLL SAE J2534.
- [0039] La [Fig.2] illustre schématiquement l'architecture d'un système de diagnostic à distance de véhicule selon des modes de réalisation de l'invention.
- [0040] La [Fig.3] illustre schématiquement une mise en œuvre de la liaison entre l'application de diagnostic véhicule et la DLL SAE J2534 selon des modes de réalisation de l'invention.
- [0041] La [Fig.4] illustre schématiquement la gestion des liaisons en cas de pluralité d'applications de diagnostic véhicule sur le dispositif distant selon des modes de réalisation de l'invention.
- [0042] La [Fig.5] illustre de façon schématique une structure matérielle de dispositif pour une mise en œuvre de l'invention.

DESCRIPTION DETAILLEE

- [0043] Dans le domaine du diagnostic et de la maintenance automobile, la présente invention propose des équipements et procédés de diagnostic véhicule à distance à l'aide d'applications de diagnostic véhicule conforme à certains standards/normes dédiés à un diagnostic *in situ*, tels les standards SAE J2534 (notamment spécification J2534-1 de décembre 2004), ISO 22900-2:2022 D-PDU et TMC RP1210 (septembre 2006).
- [0044] La [Fig.1] illustre schématiquement l'architecture d'un système 100 de diagnostic véhicule selon ces standards. Les exemples donnés ci-dessous s'appuient à titre principal sur la norme SAE J2534, qui est reprise à l'identique dans la norme ISO 22900-2:2022 D-PDU. La norme TMC RP1210 présente des mécanismes similaires.
- [0045] Le système 100 est typiquement mis en œuvre dans un garage pour véhicules légers.
- [0046] Il comporte un ordinateur ou « système hôte » 101 pour un opérateur, typiquement technicien en atelier, et un boîtier VCI dit « interface VCI » 110 ou « pass-thru device » dans la norme SAE.
- [0047] L'interface VCI 110 est reliée au véhicule 111 à inspecter par un câble connecté à une prise OSB, typiquement SAE J1962, du véhicule. Les protocoles de connexion et

communication avec les systèmes embarqués ECUs du véhicule 111 sont définis par le constructeur, et ne sont donc pas plus décrits.

- [0048] De l'autre côté, l'interface VCI 110 est reliée au système hôte 101 par un autre câble (RS-232, USB, Ethernet, IEEE1394) voire sans fil (Bluetooth – nom commercial, WiFi – nom commercial). Le système hôte 101 héberge une ou plusieurs applications de diagnostic véhicule constructeur 102, par exemple une application dite Euro5.
- [0049] L'interface VCI 110 interface ainsi le système hôte 101 avec les systèmes embarqués du véhicule 111, notamment effectue une traduction ou conversion, sans interprétation de message ou de trame de données, entre les protocoles de communication utilisés par les systèmes embarqués et le protocole de communication utilisé par le système hôte (l'application de diagnostic) hors du véhicule.
- [0050] Les normes susvisées définissent plus particulièrement les interfaces de programmation d'application (API) entre les applications de diagnostic véhicule 102 et les interfaces VCI 110.
- [0051] « norme » et « standard » sont utilisés ici comme synonymes. Par « conforme à la norme » ou « conforme au standard », il faut comprendre qu'un équipement ou module logiciel met en œuvre l'API de la norme, incluant donc les fonctions définies dans la norme, pour communiquer avec un autre équipement ou module logiciel également conforme à la norme. L'API contient un ensemble de routines ou fonctions qui peuvent être utilisées pour communiquer avec un autre composant logiciel.
- [0052] Une application de diagnostic véhicule 102 conforme à la norme utilise l'API commune de la norme, e.g. SAE J2534, ce qui lui permet de communiquer avec l'interface VCI 110 (reliée au véhicule), elle-même conforme à la norme, et ainsi contrôler les communications entre l'interface VCI 110 et le véhicule 111.
- [0053] En pratique, la conformité de l'interface VCI 110 à la norme est obtenue par l'installation et l'utilisation d'une bibliothèque de liens dynamiques ou « DLL » (Dynamic Link Library) mettant en œuvre l'API de la norme. Pour la suite, cette DLL est dénommée « DLL primaire » et est référencée 104 sur la Figure.
- [0054] La norme, telle SAE J2534, prévoit en effet que le constructeur d'une interface VCI 110 fournisse un pilote logiciel (« pilote primaire » par la suite) dédié à l'interface VCI permettant sa prise en charge matérielle par le système hôte et fournisse un programme d'installation et de configuration du système d'exploitation OS (par exemple Windows – nom commercial) de l'ordinateur 101, lequel programme installe la DLL primaire et d'autres fichiers requis, et met également à jour la base de registres 103 de l'OS. A cet effet, le programme d'installation fourni avec l'interface VCI crée une entrée de registre spécifique pour l'interface VCI, comme illustré dans l'exemple de la [Fig.1A] spécifique à la norme SAE J2534.
- [0055] Sur cette Figure, la base de registre comporte plusieurs entrées de registre 103-i qui

comprennent chacune un ensemble d'informations relatives aux interfaces VCI pré-installées sur le système hôte 101. Notamment, chaque entrée 103-i spécifique à une interface VCI 110 comporte un champ, « FunctionLibrary » dans la norme SAE, identifiant la DLL primaire correspondante, par exemple en y indiquant le chemin local d'accès à celle-ci.

- [0056] En utilisation, comme illustré par la [Fig.1B], l'application de diagnostic véhicule 102 utilise le registre 103 pour présenter à l'opérateur local la liste des interfaces VCI disponibles pour une utilisation à partir de l'application. Une fois qu'une interface VCI a été sélectionnée par l'opérateur, l'entrée 103-i correspondante du registre est utilisée pour récupérer toutes les informations concernant l'interface VCI afin que la DLL primaire 104 appropriée soit chargée pour être utilisée.
- [0057] Au démarrage de l'application de diagnostic véhicule 102 (étape 10), il est déterminé si une interface VCI est sélectionnée (étape 11). C'est le cas par exemple s'il n'y a qu'une seule interface VCI préinstallée sur l'ordinateur 101. Dans la négative, les entrées de registre 103-i sont scannées pour présenter à l'opérateur la liste des interfaces VCI disponibles (étape 12). Les noms des interfaces VCI, champ « Name » de la [Fig.1A], peuvent par exemple être présentés à l'opérateur sur un écran, pour sélection. A l'étape 13, ce dernier réalise la sélection d'une des interfaces VCI (à l'aide d'une interface utilisateur sur l'ordinateur 101, non illustrée). La localisation de la DLL primaire associée est récupérée à l'étape 14 grâce au champ « FunctionLibrary » (dans la norme SAE) de l'entrée 103-i correspondant à la sélection. La DLL primaire est alors chargée à l'étape 15, par exemple à l'aide de la fonction Windows « LoadLibrary ». Une correspondance dynamique est alors effectuée à l'étape 16 entre les fonctions/routines de l'API standardisée et les fonctions de la DLL primaire ainsi chargée.
- [0058] L'étape 17 consiste à réaliser des opérations de diagnostic sur le véhicule à l'aide de l'application 102. Les opérations de diagnostic peuvent comprendre des opérations d'interrogation de calculateurs embarqués, mais également des opérations de configuration, programmation ou reprogrammation des calculateurs.
- [0059] En pratique, l'application de diagnostic véhicule 102 exécutée établit et gère une connexion avec l'interface VCI 110, via la DLL primaire chargée 104 et l'éventuel pilote primaire, en appelant des fonctions API dédiées de la norme, par exemple les commandes PassThruOpen, PassThruClose, PassThruConnect, PassThruDisconnect, etc. dans la norme SAE.
- [0060] Par exemple, l'opérateur local décide d'actions de diagnostic à mener via une interface utilisateur permettant une interaction avec l'application de diagnostic véhicule 102 exécutée. En réponse à la sélection d'une action par l'opérateur, l'application de diagnostic véhicule 102 utilise l'API de la norme pour générer des

messages de commande/diagnostic constructeur à la DLL primaire. Cette dernière est apte, i.e. configurée, à transmettre cette commande à l'interface VCI (via le pilote primaire si prévu). Par exemple, l'application 102 appelle la fonction PassThruWriteMsgs de l'API (dans la norme SAE) pour envoyer un nouveau message de commande, lequel est reçu par la DLL primaire 104 qui le transmet à l'interface VCI qui lui est associée.

- [0061] Si une réponse est générée, celle-ci emprunte le chemin retour : l'application de diagnostic véhicule 102 qui doit récupérer le message de réponse de l'interface VCI appelle la fonction PassThruReadMsgs de l'API proposée par la DLL primaire 104 pour lire ce message de réponse. Le message de réponse peut ainsi être affiché à l'opérateur.
- [0062] L'étape 18 de déchargement de la DLL primaire 104 se produit en fin d'opération de diagnostic, lors de la fermeture de l'application 102 ou d'un changement d'interface VCI 110.
- [0063] La [Fig.2] illustre schématiquement l'architecture d'un système 200 de diagnostic à distance de véhicule selon des modes de réalisation de l'invention. Cette architecture reprend les composants ci-dessus mais dans une configuration distante, à savoir :
- [0064] au niveau d'un dispositif de diagnostic à distance 201 connecté à un réseau de communication 210, l'application de diagnostic véhicule 102 inchangée, conforme au standard SAE J2534, ISO 22900-2 D-PDU ou TMC RP1210 mis en œuvre, ainsi que la base de registre 103 listant les informations relatives aux interfaces VCI pré-installées sur l'équipement distant 201, et
- [0065] au niveau de l'ordinateur 101 de l'opérateur local, toujours la DLL primaire 104 inchangée, conforme au standard, configurée pour communiquer localement avec l'interface VCI 110 ciblée qui est reliée au véhicule 111 à diagnostiquer.
- [0066] Dans un scénario, le site local au véhicule est un garage indépendant alors que le site distant est un centre d'assistance (hotline) partagé entre plusieurs garages ou un centre d'un constructeur. Cela permet aux garagistes indépendants de déporter la gestion des multiples applications constructeur 102, tout en bénéficiant des services de diagnostic constructeur complets.
- [0067] Dans un autre scénario, le site local au véhicule est un domicile de particulier ou un lieu dans lequel le véhicule en panne est immobilisé alors que le site distant est un centre d'expert automobile, par exemple un garage. Un garagiste indépendant peut ainsi intervenir à distance auprès de ses clients, évitant une occupation de son garage et améliorant sa réactivité.
- [0068] L'invention permet donc d'effectuer des opérations de diagnostic avec n'importe quelle application de diagnostic véhicule conforme à la norme choisie qui contraint ces applications à fonctionner localement uniquement et non à distance du véhicule, mais

également avec n'importe quelle interface VCI conforme à la norme choisie, c'est-à-dire fournie avec un éventuel pilote primaire et une DLL primaire implémentant l'API de la norme. Aucune modification de ces composants logiciels n'est nécessaire. Cela est rendu possible par l'introduction de nouveaux composants intermédiaires comme décrits ci-après, dont l'un, de même nature DLL que la DLL primaire, trompe l'application de diagnostic véhicule pour permettre une communication intermédiaire via le réseau de communication.

- [0069] Le réseau de communication peut être tout type de réseau de données, filaire ou sans fil, incluant un réseau Wifi (nom commercial), le réseau Internet, un réseau local d'entreprise LAN ou WLAN, un réseau de téléphonie mobile (4G, 5G, etc.).
- [0070] Conformément à l'invention, le dispositif de diagnostic à distance 201 comporte une DLL de liaison 204 en lieu et place de la DLL primaire qui est, elle, conservée à distance sur l'ordinateur local 101. A cet effet, l'entrée de registre 103-i est donc configurée pour identifier la DLL de liaison en lieu et place d'une DLL primaire associée à l'interface VCI telle que requise par la norme, en changeant notamment le chemin « FunctionLibrary » pour y indiquer le chemin local d'accès à la DLL de liaison 204.
- [0071] Pour rendre l'interface VCI disponible dans l'application 102, celle-ci est installée sur le dispositif distant 201 selon la procédure classique décrite plus haut. Une entrée 103-i de registre dédiée est donc créée dont le chemin « FunctionLibrary » localise la DLL primaire installée. Cependant, cette dernière ne sera pas utilisée localement et peut donc être supprimée. En revanche, c'est la valeur dans l'entrée du registre qui est modifiée selon l'invention pour désormais indiquer la DLL de liaison 204.
- [0072] La DLL de liaison 204 peut être installée de façon classique par un script ou programme d'installation. Le script ou programme d'installation peut en outre procéder à la modification de l'entrée de registre 103-i évoquée ci-dessus.
- [0073] Aux fins de l'invention, la DLL de liaison 204 permet la communication avec le dispositif local 101. Aussi, la DLL de liaison est apte à (configurée pour) router, via le réseau de communication 210, un flux de données entre ladite application de diagnostic véhicule et la DLL primaire 104 distante. Le flux de données est formé des messages conformes à l'API de la norme tels qu'échangés, typiquement des commandes de diagnostic générées par l'application 102 ou des réponses générées par les calculateurs du véhicule 111 inspecté et remontées à la DLL primaire 104.
- [0074] Comme décrit ci-dessous en lien avec les **Figures 3 et 4**, la DLL de liaison 204 présente une double fonction aux fins de réaliser le routage des données.
- [0075] Dans le sens émission, elle est configurée pour découper le flux de données reçu de l'application 102, en trames de données exportables sur un protocole réseau IP (Internet Protocol), notamment selon un langage structuré, type JSON. Elle encapsule

ainsi les données (données brutes telles que reçues) dans des trames prévues par le protocole réseau (typiquement de tailles limitées). Elle a donc une première fonction de conversion et d'encapsulation. Elle est en outre configurée pour transmettre ces trames de données du dispositif distant 201 à l'ordinateur local 101, vers un agent dédié décrit ci-dessous. Elle présente donc une deuxième fonction de communication avec l'agent dédié.

- [0076] La communication des trames sur le réseau de communication 210 peut être réalisée avec tout protocole de communication, incluant la suite des protocoles internet, et notamment http (Hypertext Transfer Protocol), HTTPs, etc.
- [0077] Dans le sens réception, elle opère les fonctions réciproques : elle communique avec l'agent dédié et reçoit des trames de données, notamment selon un langage structuré, type JSON ; puis elle reconstitue un flux de données (en désencapsulant les données de ces trames reçues. Un message de réponse véhicule conforme à l'API est ainsi reformé.
- [0078] Comme illustré sur la Figure, l'ordinateur local 101 comporte un agent de communication réciproque de la DLL de liaison 104, pour établir une communication entre ces deux entités et opérer les fonctions réciproques. Cet agent est nommé « agent logiciel de liaison » par la suite et porte la référence 205. De par son rôle central dans l'ordinateur local 101 qui peut être relié (successivement) à plusieurs interfaces VCI différentes, l'agent logiciel de liaison 205 peut également être vu comme un serveur VCI.
- [0079] De façon générale, dans le sens réception, l'agent logiciel de liaison 205 reconstitue le flux de données initiales (c'est-à-dire les messages de diagnostic conformes à l'API de la norme) en désencapsulant les données des trames qu'il reçoit de la DLL de liaison 204, puis transmet ce flux reconstitué à la DLL primaire 104 (via l'API de la norme) pour communication au pilote de l'interface VCI ou directement à l'interface de la VCI comme dans l'art antérieur. Aussi, la DLL primaire 104 est récupérée puis chargée lors de l'exécution de l'agent logiciel de liaison 205, permettant ainsi d'effectuer une correspondance dynamique des fonctions de la DLL primaire 104 avec celles de l'agent logiciel de liaison 205. Dans le sens émission, l'agent logiciel de liaison 205 découpe un flux de données (message de réponse conforme à l'API de la norme) reçu de la DLL primaire 204 (initialement de l'interface VCI 110) en trames de données exportables sur le protocole de liaison, puis transmet ces trames de données vers la DLL de liaison 204 du dispositif distant 201.
- [0080] Dans une variante d'architecture qui s'affranchit de l'utilisation d'un ordinateur sur le site local, l'agent de liaison 205 et l'éventuel pilote primaire peuvent être embarqués dans l'interface VCI 110. Ainsi le « dispositif local » est formé de la seule interface VCI. Cette dernière comprend, dans ce cas, une interface de communication avec le réseau 210.

- [0081] La [Fig.2] illustre un réseau de communication 210 comprenant un ou plusieurs équipements 206 pouvant assurer une ou plusieurs fonctions relatives à la communication des trames de données entre le dispositif distant 201 et le dispositif local 101.
- [0082] Un équipement 206 de type routeur permet de router ces trames au sein d'un réseau, par exemple d'entreprise, auquel appartiennent les deux dispositifs.
- [0083] Un équipement 206 de type passerelle est utilisé lorsque les deux dispositifs appartiennent à des réseaux distincts et assure l'interconnexion entre ces réseaux et le routage des trames d'un réseau à l'autre. Une telle passerelle peut en outre assurer des services de sécurisation des communications, par exemple inclure un serveur OAuth2 (nom commercial) pour une authentification des dispositifs distants souhaitant accéder à l'agent de liaison (serveur VCI) 205.
- [0084] Le site local, par exemple l'ordinateur 201, peut en outre être doté d'un pare-feu configuré pour limiter l'accès à l'agent de liaison (serveur VCI) 205 par un unique port, par exemple le port 443 pour http.
- [0085] La [Fig.3] illustre schématiquement une mise en œuvre de la liaison entre l'application de diagnostic véhicule 102 et la DLL primaire 104. Dans cette configuration, la DLL de liaison 204 et l'agent logiciel de liaison 205 établissent un canal de communication de type web socket, sur le réseau de communication 210.
- [0086] Sur la Figure, la DLL de liaison 204 est représenté comme deux modules, l'un 204-1 de conversion de flux de données et l'autre 204-2 de communication sur le réseau de communication 210. De façon similaire, l'agent de liaison 205 comprend un premier module 205-2 de communication sur le réseau de communication 210 et un second module 205-1 de conversion de flux de données.
- [0087] Comme exposé plus haut, les modules 204-1 et 205-1 réalisent une conversion entre un flux de données conforme à l'API de la norme et des trames de données structurées par exemple au format JSON (soit dans le sens émission, soit dans le sens réception).
- [0088] Dans un mode de réalisation, ces modules 204-1 et 205-1 comprennent un filtre de routage prévu pour déterminer, en fonction d'un critère de filtrage, si un message reçu pour routage est à supprimer sans le router. En effet, l'utilisation à distance de l'application de diagnostic véhicule 102 initialement conçue pour être directement connectée à l'interface VCI peut poser des soucis de performance. En particulier, le protocole des normes susvisées est un protocole synchrone dans lequel l'application 102 attend de recevoir une réponse avant d'émettre un message suivant. Les échanges sont donc ralentis et peuvent entrer en conflit avec des temporisations prédéfinies au niveau des calculateurs embarqués, lesquelles sont souvent inférieures à 100ms.
- [0089] Le filtrage prévu dans les modules 204-1 et 205-1 permet ainsi de supprimer des trames protocolaires non indispensables.
- [0090] Le filtrage peut être statique. Par exemple, certains messages sont identifiables par un

format ou une valeur de champ spécifique. Une liste noire des types de message à supprimer sous routage (ou en variante une liste blanche des types de message à router sans suppression, les autres étant à supprimer) peut être prévue. Le module compare donc simplement le type d'un nouveau message reçu aux types de la liste, et en cas de message non autorisé, le supprime. A titre illustratif, les messages de type VBAT relatifs à une interrogation sur le niveau de batterie du véhicule inspecté, de type FwVersion relatifs à une interrogation sur la version de firmware (« logiciel embarqué ») embarqué dans l'interface VCI, de type DllVersion relatifs à une interrogation sur la version de la DLL primaire et/ou de type ApiVersion relatifs à une interrogation sur la version d'API mise en œuvre par la DLL primaire peuvent être supprimés systématiquement ou leur fréquence de routage peut être réduite notamment en laissant passer un message par période prédéfinie (par exemple un message par minute au lieu de dix messages sans filtrage). En effet, la fréquence de ces messages peut atteindre plusieurs dizaines par minute et ainsi réduit considérablement les performances d'un mode d'utilisation à distance de l'application de diagnostic véhicule 102.

- [0091] Ces messages peuvent être identifiés dans la signalisation d'appel de fonctions de l'API standardisée. A titre illustratif, la commande PassThruIoctl du standard SAE comprend un paramètre IoctlID dont la valeur identifie un de ces types de messages, notamment IoctlID=READ_VBATT pour lire l'état de batterie. De même, la commande PassThruReadVersion du standard SAE retourne les versions du firmware, du pilote et de l'API via les trois paramètres pFirmwareVersion, pDllVersion et pApiVersion respectivement.
- [0092] Le filtrage peut aussi être dynamique. Par exemple, le module détermine dans un premier temps un type de message routé de façon trop récurrente, qui peut être l'un ou plusieurs de ceux identifiés ci-dessus. A titre illustratif, un type de message est trop récurrent si sa fréquence de transmission est supérieure à une valeur seuil, par exemple dix transmissions par minute. En cas de détermination positive, tout message ultérieur ayant ce type de message est supprimé sans routage ou en variante, la fréquence de routage est réduite en laissant par exemple passer un seul message par unité de temps prédéfinie.
- [0093] De façon préférentielle, la réponse obtenue du véhicule pour la seule interrogation transmise à celui-ci est sauvegardée en cache de la DLL de liaison 204, de sorte à fournir une copie de cette réponse à tout nouveau message de même type qui est supprimé. Ainsi, l'application de diagnostic véhicule 102, non modifiée, n'est pas confrontée à des temporisations bloquantes (interrogation sans réponse).
- [0094] Les messages non supprimés sont communiqués au module 204-2 ou 205-2 de communication.

- [0095] Comme indiqué ci-dessus, la DLL de liaison 204 et l'agent logiciel de liaison 205, via ces deux modules 204-2 et 205-2, établissent un canal de communication de type web socket, sur le réseau de communication 210.
- [0096] Compte tenu du schéma client-serveur de la technologie web socket, le dispositif distant 201 peut être doté d'un serveur web socket 304. Le serveur web socket 304 peut être intégré à la DLL de liaison 204 ou être avantageusement un module indépendant (disponible sur étagère, par exemple une librairie Open Source) que la DLL de liaison 204 charge à la demande. En variante, le serveur web socket, référencé 305, est hébergé sur le dispositif local 101.
- [0097] Lorsque l'opérateur indépendant fait appel au service distant, le module 205-2 de communication de l'agent 205 se connecte au dispositif distant 201 en établissant une connexion Web Socket avec le serveur 304. De son côté, le module 204-2 de la DLL de liaison 204 établit également une connexion Web Socket avec le serveur 304 (de façon interne dans le dispositif distant 201) si celui-ci n'est pas directement intégré à la DLL de liaison. Un canal de communication entre les deux modules 204-2 et 205-2 est ainsi établi par web sockets.
- [0098] L'utilisation de l'application de diagnostic véhicule 102 est toujours conforme au procédé de la [Fig.1B]. Les aménagements selon l'invention sont transparents pour cette application et pour la DLL primaire 104.
- [0099] Les étapes 10, 11, 12, 13 et 14 demeurent inchangées.
- [0100] Compte tenu de la modification du champ « FunctionLibrary » pour y indiquer la DLL de liaison 204, l'étape 15 permet de récupérer et de charger (fonction Windows « LoadLibrary ») ladite DLL de liaison 204 et non plus la DLL primaire 104. La correspondance dynamique de l'étape 16 est alors effectuée avec les fonctions de la DLL de liaison 204 ainsi chargée.
- [0101] Cela conduit à ce que les opérations de diagnostic sur le véhicule, toujours générées à l'aide de l'application 102, (étape 17) se passent comme suit.
- [0102] L'opérateur indépendant faisant appel au service distant, son module 205-2 de communication a établi une connexion Web Socket avec le serveur web socket 304. La connexion entre l'interface VCI 110 et la DLL primaire 104 est, quant à elle, gérée de façon classique.
- [0103] Lorsque la DLL de liaison 204 est chargée suite à la sélection sur le dispositif distant 201 de l'interface VCI utilisée par l'opérateur indépendant, le module 204-2 de communication établit également une connexion Web Socket avec le serveur 304. Le canal de communication entre la DLL de liaison 204 et l'agent de liaison 205 est ainsi établi.
- [0104] Lorsqu'un message de diagnostic est généré par l'application de diagnostic véhicule 102, celui-ci est pris en charge par la DLL de liaison 204 conformément à la correspondance dynamique de l'étape 16. Le module 204-1 filtre (supprime) optionnellement

le message s'il est d'un type interdit, sinon le convertit en trames de données structurées, type JSON.

- [0105] Chaque trame comporte, typiquement dans un entête, un identifiant de la VCI cible (et donc de l'agent de liaison 205 associé) tel qu'un numéro de série de cette VCI. En variante, un identifiant de DLL primaire 204 ou un identifiant d'agent de liaison 205 pourrait être utilisé. Cet identifiant peut avoir été saisi ou sélectionné manuellement par l'opérateur distant lors de l'initialisation de la procédure. Par exemple, la liste des identifiants VCI disponibles pour l'opérateur distant est stockée dans un serveur du réseau de communication (serveur Internet) et est mise à jour par les opérateurs locaux à chaque nouvelle VCI installée sur leurs dispositifs locaux.
- [0106] Ces trames sont passées au module 204-2 qui les transmet au dispositif local 101 dont l'adresse (et le port) est associée, dans une table de routage par exemple au sein du serveur du réseau de communication, à l'identifiant de VCI. Cette transmission est réalisée à l'aide du protocole http ou HTTPS à l'intérieur de la connexion web socket. Dans ce cas, le module 205-2 peut prendre la forme d'un serveur http avec lequel le module 204-2 communique.
- [0107] Les trames reçues sont passées au module 205-1 grâce à l'identifiant de VCI présent dans chaque trame. Le module 205-1 désencapsule alors ces trames de sorte à reformer le message de diagnostic.
- [0108] Ce dernier est transmis par l'agent de liaison 205 à la DLL primaire 104. Cette transmission repose sur la correspondance dynamique effectuée entre l'agent de liaison 205 et la DLL primaire 104 lors du chargement de cette dernière.
- [0109] La suite de la transmission du message de diagnostic vers l'interface VCI est classique.
- [0110] La transmission des messages de réponse (depuis les calculateurs vers l'application 102) suit les mêmes mécanismes, dans le sens inverse.
- [0111] La [Fig.4] illustre schématiquement la gestion des liaisons en cas de pluralité d'applications de diagnostic véhicule 102 sur le dispositif distant 201.
- [0112] Un opérateur ou expert distant, utilisateur du dispositif distant 201, peut mener en parallèle ou dans le temps plusieurs opérations de diagnostic pour le compte d'un même garagiste (comme sur la [Fig.4]) ou pour le compte de plusieurs garagistes (auquel cas plusieurs dispositifs locaux 101 sont mis en œuvre, reliés au dispositif distant 201 par le réseau 210).
- [0113] L'opérateur distant dispose de plusieurs applications constructeur 102 pouvant provenir du même constructeur ou de plusieurs constructeurs différents, ici trois applications APP_A, APP_B et APP_C. Pour chaque application, une interface VCI est sélectionnée (étape 13) permettant le chargement d'une DLL de liaison respective 204. Dans l'exemple illustré, l'opérateur sélectionne l'interface VCI1 « service 1 » via

l'application APP_A. En conséquence de quoi, la DLL de liaison « service 1 » est chargée par APP_A. La Figure n'illustre que les DLL de liaison chargées et non celles préinstallées (possiblement plus nombreuses) qui ne sont pas utilisées. De même, l'opérateur sélectionne l'interface VCI2 « service 2 » via l'application APP_B, en conséquence de quoi, la DLL de liaison « service 2 » est chargée par APP_B. Egalement, l'opérateur sélectionne l'interface VCI3 « service 3 » via l'application APP_C, en conséquence de quoi, la DLL de liaison « service 3 » est chargée par APP_C.

- [0114] Du côté de l'opérateur local ou des opérateurs locaux faisant appel aux services de l'opérateur distant, le ou les agents de liaison 205 sont lancés. Un choix des interfaces VCI à utiliser peut être proposé à l'opérateur local de façon similaire au choix dans l'application de diagnostic véhicule 102, de sorte à permettre le chargement de la DLL primaire 104 idoine mais également du module de conversion 205-1 adapté à la DLL de liaison 205.
- [0115] Dans l'exemple de la [Fig.4], l'opérateur local lance trois agents de liaison 205 pour gérer trois interfaces VCI, VCI1, VCI2 et VCI3 à des moments différents. En effet, à un instant donné, l'ordinateur 101 est relié à une unique interface VCI. Les DLLs primaires et modules 205-1 « service 1 », « service 2 », « service 3 » sont donc chargés à trois moments différents. Bien entendu, ceux-ci peuvent être chargés simultanément mais sur plusieurs ordinateurs hôtes 101 (sur le même site ou sur des sites distincts).
- [0116] Dans un mode de réalisation, le module 205-2 de communication est commun à plusieurs modules de conversion 205-1 lorsque ceux-ci sont instanciés. Aussi, le module 205-2 est apte à router les trames de données reçues vers le module de conversion 205-1 instancié.
- [0117] En variante, chaque agent logiciel de liaison 205 a son propre module de communication 205-2.
- [0118] La [Fig.5] illustre de façon schématique une structure matérielle d'un dispositif 500 du système de diagnostic à distance 200 selon l'invention. Un tel dispositif peut être le dispositif distant 201, le dispositif local 101 et/ou l'interface VCI 110.
- [0119] Le dispositif 500 peut être un dispositif portatif tel un téléphone intelligent (smartphone), une tablette numérique, un ordinateur portable, un assistant personnel, ou être un dispositif fixe tel un ordinateur de bureau ou une borne interactive. Plus généralement, tout dispositif informatique aptes à la mise en œuvre des traitements évoqués ci-dessus peut être utilisé.
- [0120] Le dispositif 500 comprend un bus de communication 501 auquel sont préférentiellement connectés :
- [0121] - une ou plusieurs unités centrales de traitement 502, telles qu'un ou des processeurs CPU et/ou un ou des processeurs ou cartes graphiques GPU et/ou un ou des micropro-

cesseurs ;

- [0122] - une mémoire de stockage 503, de type ROM et/ou disque dur et/ou mémoire flash, pour le stockage de programmes informatiques 509 destinés à mettre en œuvre l'invention mais également de toute donnée nécessaire à l'exécution des programmes, incluant les DLL;
- [0123] - une mémoire vive 504, de type RAM voire vidéo RAM (VRAM), pour le stockage du code exécutable des programmes informatiques ainsi que les registres adaptés pour enregistrer des variables et des paramètres nécessaires à leur exécution ;
- [0124] - une ou plusieurs interfaces de communication 505 connectée à un réseau externe de communication NET, typiquement le réseau de communication 210, et/ou à des équipements, par exemple câble série, RS-232, USB, Ethernet, IEEE1394, Bluetooth ;
- [0125] - une ou plusieurs entrées/sorties I/O 506 permettant à un utilisateur d'interagir directement avec les programmes 509 de l'invention en cours d'exécution. Typiquement, les entrées/sorties peuvent inclure un écran servant d'interface graphique avec l'utilisateur et/ou un clavier ou tout autre moyen de pointage permettant à l'utilisateur de lancer par exemple l'exécution des programmes 509 et la sélection d'options et/ou un haut-parleur.
- [0126] De préférence, le bus de communication assure la communication et l'interopérabilité entre les différents éléments inclus dans le dispositif ou connectés à celui-ci. La représentation du bus n'est pas limitative et, en particulier, l'unité centrale est utilisable pour communiquer des instructions à tout élément du dispositif directement ou au moyen d'un autre élément du dispositif informatique.
- [0127] Le code exécutable stocké en mémoire peut être reçu au moyen du réseau de communication NET, via l'interface, afin d'y être stocké avant exécution. En variante, le code exécutable n'est pas stocké en mémoire non volatile mais peut être chargé en mémoire volatile depuis un serveur distant via le réseau de communication NET pour exécution directement. C'est le cas notamment des applications web (web apps).
- [0128] L'unité centrale est de préférence adaptée pour contrôler et diriger l'exécution des instructions ou des parties de code logiciel du ou des programmes selon l'invention. À la mise sous tension, le ou les programmes qui sont stockés en mémoire non volatile ou sur le serveur distant sont transférés/chargés dans la mémoire vive, qui contient alors le code exécutable du ou des programmes, ainsi que des registres pour le stockage des variables et des paramètres nécessaires à la mise en œuvre de l'invention.
- [0129] Dans un mode de réalisation, les traitements selon l'invention sont réalisés localement par le dispositif utilisateur, de préférence en temps réel ou quasi-temps réel. Dans ce cas, les programmes en mémoire mettent en œuvre l'ensemble des traitements décrits ci-après.
- [0130] Le réseau de communication NET peut être tout réseau informatique filaire ou non-

filaire ou un réseau de téléphonie mobile permettant une connexion à un réseau informatique tel que l'Internet.

[0131] Les exemples qui précèdent ne sont que des modes de réalisation de l'invention qui ne s'y limite pas.

Revendications

[Revendication 1]

Dispositif (201) de diagnostic à distance de véhicule, comportant :

- une interface de communication avec un réseau de communication (210), et
- une application de diagnostic véhicule (102) conforme à l'un des standards SAE J2534, ISO 22900-2 D-PDU et TMC RP1210 et apte à communiquer avec une interface de communication véhicule, VCI, par l'utilisation d'une bibliothèque de liens dynamiques, DLL, primaire associée à l'interface VCI et conforme audit standard, ladite application de diagnostic étant configurée pour utiliser une entrée (103-i) de registre (103) de système d'exploitation (OS) afin d'identifier la DLL primaire,

caractérisé en ce que le dispositif de diagnostic à distance (201) comporte une DLL de liaison (204) apte à router, via le réseau de communication, un flux de données entre ladite application de diagnostic véhicule et une DLL primaire distante (104) configurée pour communiquer localement avec une interface VCI (110) reliée au véhicule (111), ladite entrée de registre étant configurée pour identifier la DLL de liaison en lieu et place d'une DLL primaire associée à l'interface VCI.

[Revendication 2]

Système (200) de diagnostic à distance de véhicule, comportant :

- un dispositif (201) de diagnostic à distance connecté à un réseau de communication (210) et comprenant une application de diagnostic véhicule (102) conforme à l'un des standards SAE J2534, ISO 22900-2 D-PDU et TMC RP1210 et apte à communiquer avec une interface de communication véhicule, VCI, par l'utilisation d'une bibliothèque de liens dynamiques, DLL, primaire associée à l'interface VCI et conforme audit standard, ladite application de diagnostic étant configurée pour utiliser une entrée (103-i) de registre (103) de système d'exploitation (OS) afin d'identifier la DLL primaire, et
- un dispositif (101) local au véhicule, connecté au réseau de communication et comprenant une DLL primaire (104) configurée pour communiquer localement avec une interface VCI (110) reliée à un véhicule (111) à diagnostiquer,

caractérisé en ce que le dispositif de diagnostic à distance (201) comporte une DLL de liaison (204) apte à router, via le réseau de communication et un agent logiciel de liaison (205) du dispositif local (101), un flux de données entre ladite application de diagnostic véhicule et la DLL primaire de l'interface VCI reliée au véhicule, ladite entrée de registre étant configurée pour identifier la DLL de liaison en lieu et place d'une DLL primaire associée à l'interface VCI.

- [Revendication 3] Système (200) selon la revendication 2, dans lequel la DLL de liaison (204) et/ou l'agent logiciel de liaison (205) comprend un convertisseur (204-1, 205-1) apte à encapsuler le flux de données en une ou plusieurs trames de données structurées ou à désencapsuler une ou plusieurs trames de données structurées en ledit flux de données, de sorte à échanger des trames de données structurées sur le réseau de communication.
- [Revendication 4] Système (200) selon la revendication 3, dans lequel le convertisseur encapsule le flux de données dans une structure au format JavaScript Object Notation, JSON.
- [Revendication 5] Système (200) selon la revendication 3, dans lequel un identifiant d'interface VCI est ajouté dans la trame de données structurées de sorte à router ladite trame vers une DLL primaire correspondant audit identifiant parmi plusieurs DLLs primaires accessibles sur le réseau de communication.
- [Revendication 6] Système (200) selon l'une des revendications 2 à 5, dans lequel la DLL de liaison (204) et l'agent logiciel de liaison (205) sont configurées pour établir un canal de communication de type web socket entre eux, sur le réseau de communication (210).
- [Revendication 7] Système (200) selon l'une des revendications 2 à 6, dans lequel la DLL de liaison (204) et/ou l'agent logiciel de liaison (205) comprend un filtre de routage (204-1, 205-1) configuré pour déterminer, en fonction d'un critère de filtrage, si un message reçu pour routage est à supprimer sans le router.
- [Revendication 8] Système (200) selon la revendication 7, dans lequel le critère de filtrage supprime un message de type VBAT relatif à une interrogation sur le niveau de batterie du véhicule, de type FwVersion relatif à une interrogation sur la version de firmware embarqué dans l'interface VCI, de type DllVersion relatif à une interrogation sur la version de la DLL primaire et/ou de type ApiVersion relatif à une interrogation sur la version d'API mise en œuvre par la DLL primaire.

- [Revendication 9] Système (200) selon la revendication 7, dans lequel le filtre de routage est configuré pour déterminer un type de message routé de façon récurrente et, en réponse, pour configurer ledit critère de filtrage de sorte à supprimer tout message ultérieur dudit type sur au moins une période prédéfinie.
- [Revendication 10] Système (200) selon l'une des revendications 2 à 9, dans lequel l'interface VCI (110) reliée au véhicule (111) forme un dispositif distinct du dispositif local (101).
- [Revendication 11] Système (200) selon la revendication 10, dans lequel le dispositif local (101) comporte un pilote associé à l'interface VCI (110) pour router les données entre la DLL primaire (204) et l'interface VCI (110) externe au dispositif local.
- [Revendication 12] Système (200) selon l'une des revendications 2 à 9, dans lequel le dispositif local forme l'interface VCI (110) reliée au véhicule.
- [Revendication 13] Procédé de diagnostic à distance d'un véhicule, comportant, au niveau d'un dispositif (201) de diagnostic à distance d'un dispositif local (101) audit véhicule (111), les étapes suivantes :
- obtenir une application de diagnostic véhicule (102) conforme à l'un des standards SAE J2534, ISO 22900-2 D-PDU et TMC RP1210 et apte à communiquer avec une interface de communication véhicule, VCI, par l'utilisation d'une bibliothèque de liens dynamiques, DLL, primaire associée à l'interface VCI et conforme audit standard, ladite application de diagnostic étant configurée pour utiliser une entrée (103-i) de registre (103) de système d'exploitation (OS) afin d'identifier la DLL primaire,
 - configurer l'entrée de registre pour identifier, en lieu et place d'une DLL primaire associée à l'interface VCI, une DLL de liaison (204) apte à router, via un réseau de communication (210), un flux de données entre ladite application de diagnostic véhicule et une DLL primaire (104) distante configurée pour communiquer localement avec l'interface VCI (110) reliée au véhicule (111), et
 - charger (15) la DLL de liaison lors d'une exécution de l'application de diagnostic véhicule de sorte qu'un message de diagnostic généré par l'application de diagnostic véhicule soit routé sur le réseau de communication vers la DLL primaire distante et l'interface VCI reliée au véhicule.

[Fig. 1]

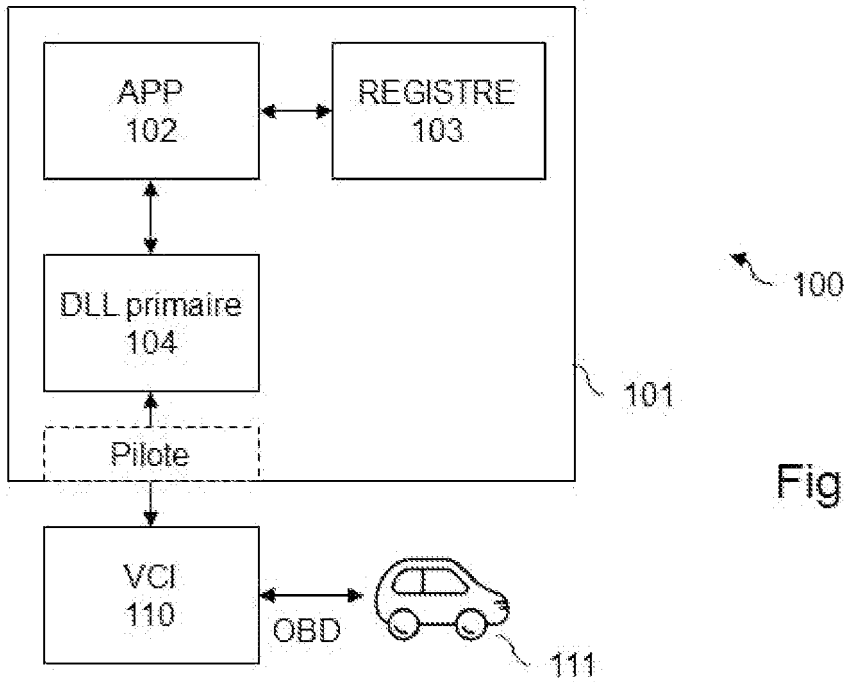


Figure 1

[Fig. 1A]

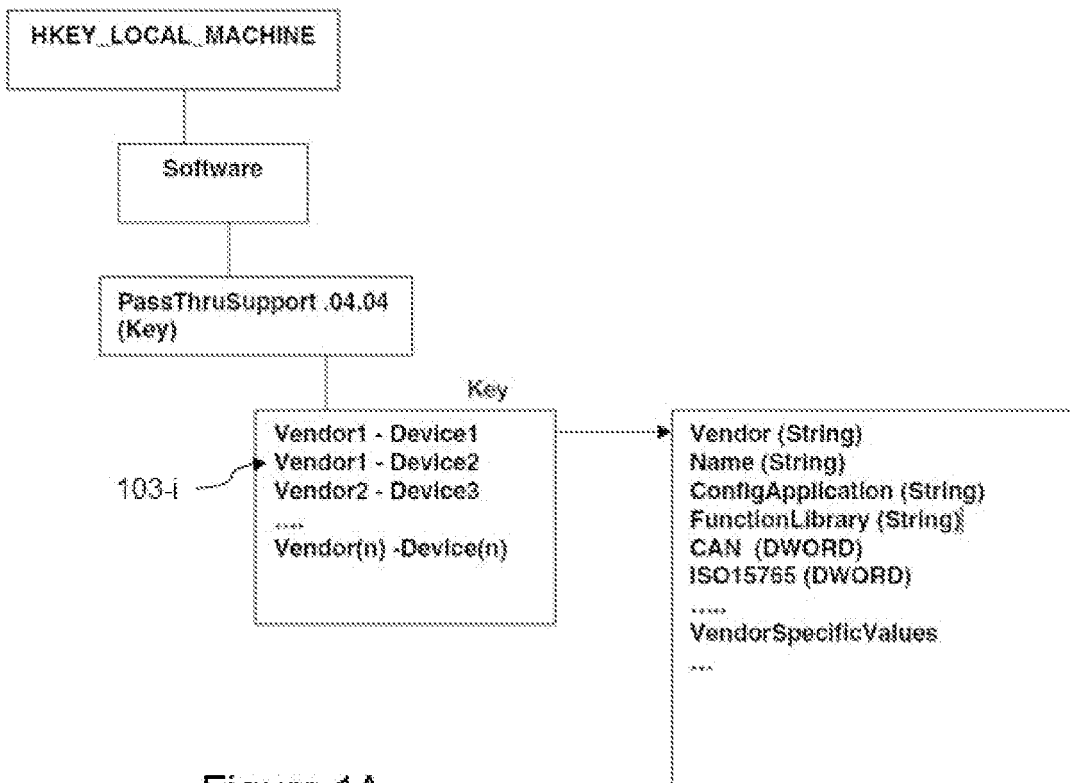


Figure 1A

[Fig. 1B]

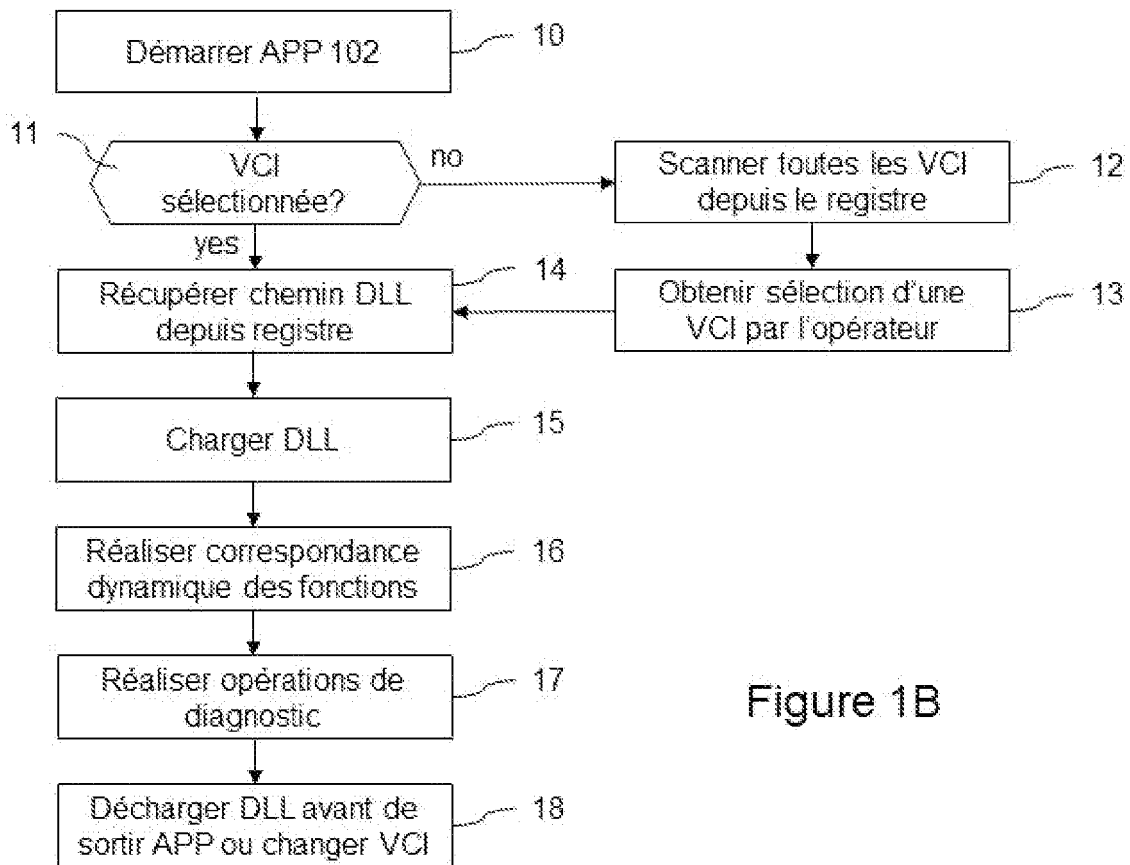


Figure 1B

[Fig. 2]

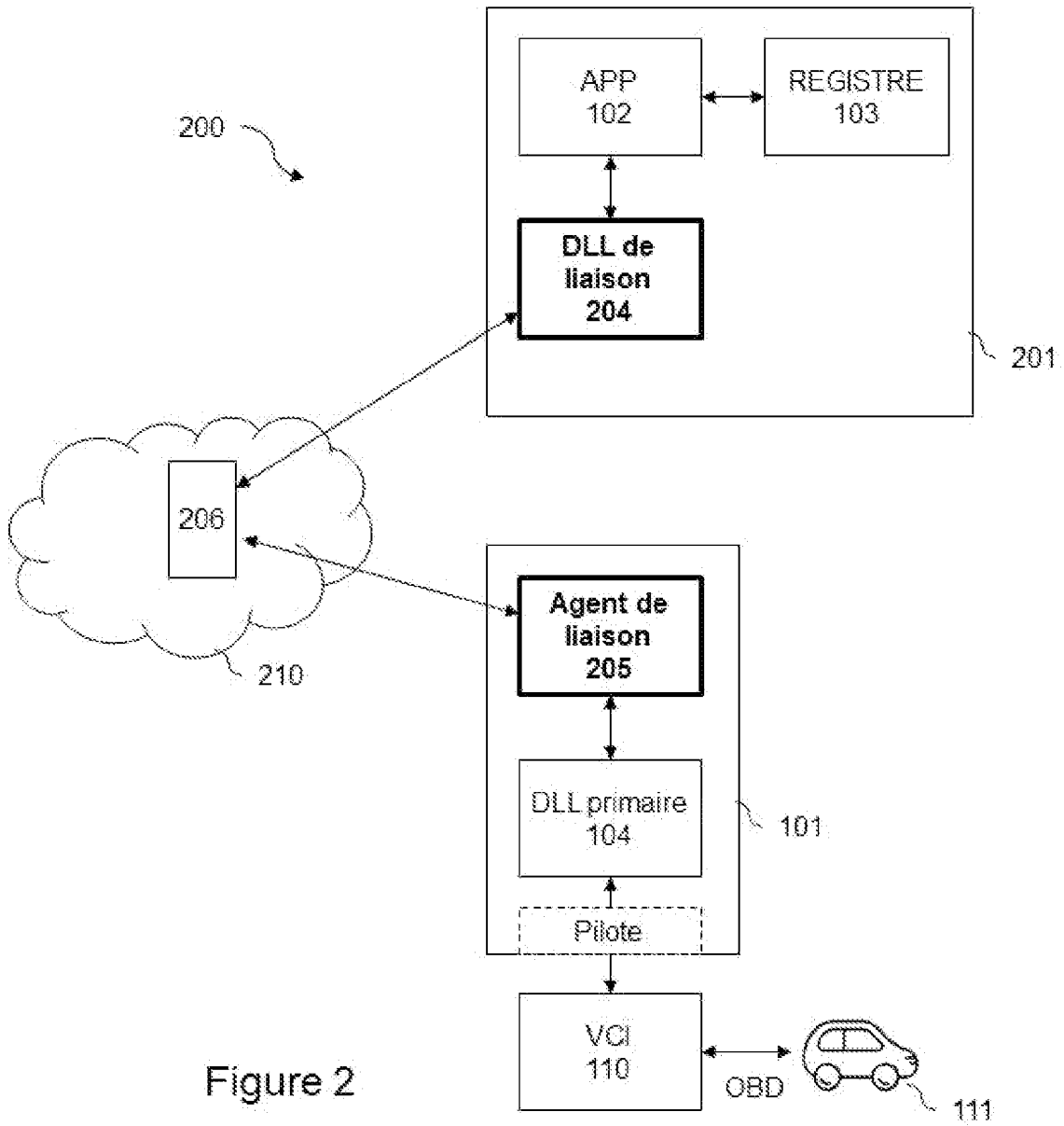


Figure 2

[Fig. 3]

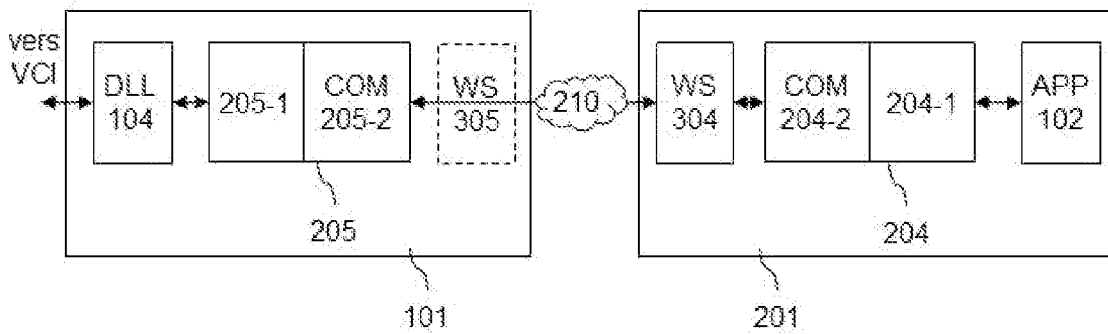


Figure 3

[Fig. 4]

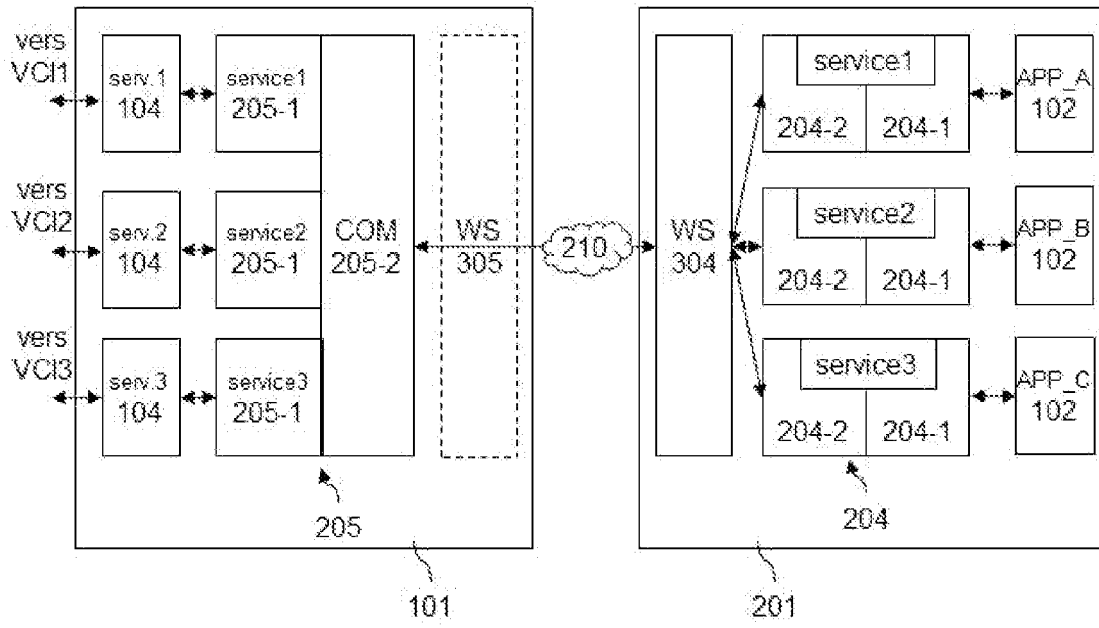


Figure 4

[Fig. 5]

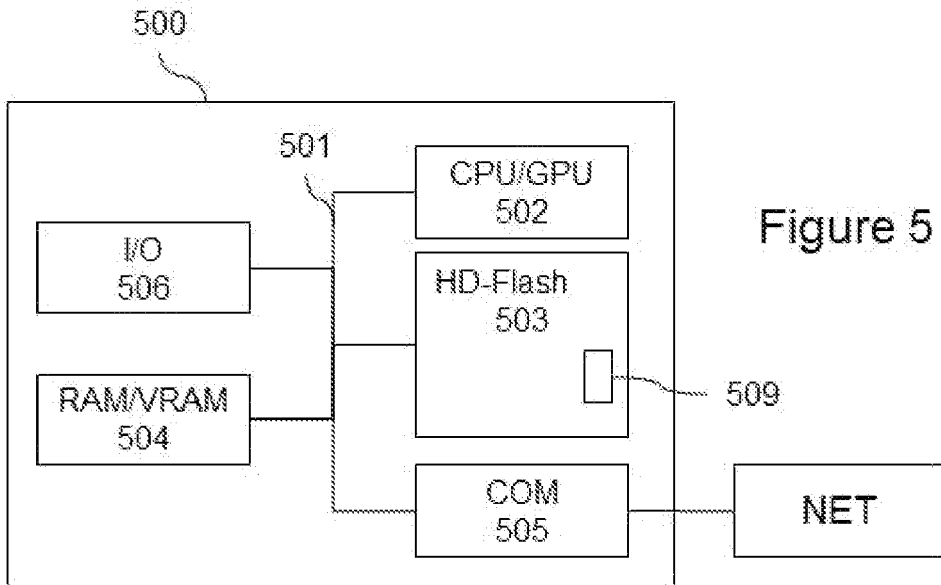


Figure 5



**RAPPORT DE RECHERCHE
PRÉLIMINAIRE**

N° d'enregistrement
national

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

FA 912173
FR 2211654

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	<p>JOHANSON MATHIAS ET AL: "Remote Vehicle Diagnostics over the Internet using the DoIP Protocol", THE SIXTH INTERNATIONAL CONFERENCE ON SYSTEMS AND NETWORKS COMMUNICATIONS, 23 octobre 2011 (2011-10-23), XP055788039, Extrait de l'Internet: URL:http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=0FCD6298AAB7C2778232ABB77C939E7B?doi=10.1.1.418.5332&rep=rep1&type=pdf> * section V *</p> <p style="text-align: center;">-----</p>	1-13	H04L67/12 G07C5/08
			DOMAINES TECHNIQUES RECHERCHÉS (IPC)
			G07C G06F H04W
Date d'achèvement de la recherche		Examineur	
12 mai 2023		Cuevas Casado, A	
CATÉGORIE DES DOCUMENTS CITÉS		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	
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			