(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL DE LA PROPRIÉTÉ INDUSTRIELLE

COURBEVOIE

(11) **N° de publication :** ***3 145 627***

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

(21) **N° d'enregistrement national :** **23 01077**

(51) Int Cl[8] : ***G 06 F 21/64*** *(2023.01), G 06 F 21/60*

(12) DEMANDE DE BREVET D'INVENTION A1

(22) **Date de dépôt :** 06.02.23.

(30) **Priorité :**

(43) **Date de mise à la disposition du public de la demande :** 09.08.24 Bulletin 24/32.

(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) :** *TJC Société anonyme* — FR.

(72) **Inventeur(s) :** JULIEN Thierry.

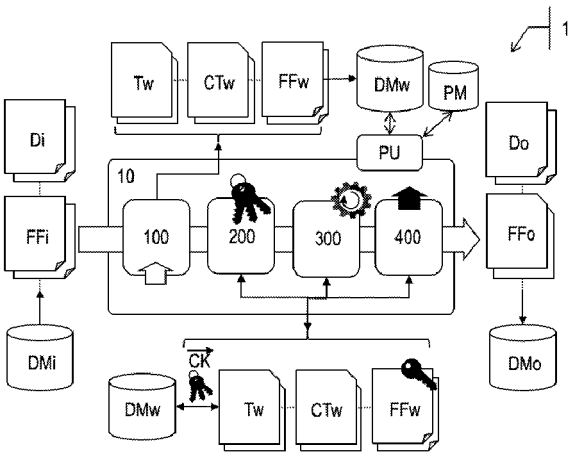
(73) **Titulaire(s) :** TJC Société anonyme.

(74) **Mandataire(s) :** MED'INVENT CONSULTING.

(54) Procédé de gestion de données concernées par des obligations contradictoires de conservation et de mise à jour.

(57) L'invention concerne un procédé de gestion de données (10) mis en œuvre par un système informatique 1. Un tel procédé 10 comprend un traitement d'importation (100) desdites données stockées dans une ou plusieurs archives unitaires (FFi) pour créer des conteneurs de données (Tw) et des tables associatives (CTw) ainsi que des archives unitaires (FFw) de travail. Ledit procédé (10) comporte un traitement (200) de chiffrement desdites archives unitaires de travail (FFw) préalablement à toute première mise à jour des données. Il comporte en outre un traitement (300) de toute consigne en mise à jour desdites données agencé de sorte à gérer des données concernées par des obligations de conservation et de mise à jour. Un tel procédé de gestion de données (10) peut comporter également un traitement (400) d'exportation desdites données gérées.

Figure à publier avec l'abrégé : Fig 1



FR 3 145 627 - A1

Description

Titre de l'invention : Procédé de gestion de données concernées par des obligations contradictoires de conservation et de mise à jour

[0001] L'invention concerne un procédé de gestion, c'est-à-dire d'importation, de mise à jour, voire d'exportation, de données concernées par des obligations contradictoires, d'une part, de conservation et, d'autre part, de mise à jour ou de suppression. A titre d'exemple préféré, de telles données peuvent comporter des données personnelles importées, créées, modifiées ou supprimées, voire exportées par un logiciel de gestion de données, également connu sous l'appellation anglo-saxonne d'ERP pour « *Enterprise Resource Planning* ».

[0002] De telles données sont exploitées notamment en Entreprise dans le cadre d'applications informatiques multiples, telles que la comptabilité, la gestion des ressources humaines, la vente, la production, etc. Ces applications informatiques sont soumises à des obligations d'audit par des organismes divers (états, organisations internationales, organismes dédiés) en application de codes des impôts, de règlements de police, etc. De tels audits imposent une conservation de données et de documents pendant des périodes définies.

[0003] Ainsi, selon la législation française, des pièces comptables (ou données) doivent être conservées durant un minimum de trois années à partir de la clôture de l'exercice comptable concerné. Des documents fiscaux nécessitent une conservation de six ans ou encore les contrats commerciaux et autres correspondances associées doivent être conservés durant cinq ans à compter de leurs conclusions respectives. De la même manière, les bulletins de salaires de tout collaborateur, y compris sous leurs formes numériques, doivent être conservés durant cinq années suivant leurs émissions. Nous pouvons également mentionner que tout document en lien avec un titre de propriété intellectuelle, tel qu'un brevet d'invention, doit être conservé durant cinq années à partir de l'extinction dudit titre.

[0004] De façon alternative, un archivage fiscal sert à conserver de façon pérenne l'ensemble des informations, données et traitements informatiques. Il permet de répondre aux requêtes qu'une administration fiscale peut exercer dans le cadre ou en dehors d'un contrôle fiscal. Une archive fiscale est définie dans le présent document comme le résultat d'un processus qui permet de garantir qu'un contenu d'une telle archive est identique aux informations initiales. L'intégrité des archives produites dans le temps doit être garantie, c'est-à-dire depuis leurs créations jusqu'à leurs éventuelles suppressions. Une archive fiscale doit en outre être exploitable, c'est-à-dire lisible, y compris lorsque l'application logicielle qui l'a produite n'est plus disponible.

L'utilisation d'un format dit « ouvert » est donc généralement recommandé. Une archive fiscale se présente donc généralement sous la forme d'un fichier texte ou plat, voire sous la forme d'un document électronique ou d'une image.

[0005] Les archives potentiellement soumises à des contrôles ou audits réalisés à l'initiative coercitive de tiers, tels que le fisc ou les douanes, sont appelées « archives externes ». Les archives qui n'ont qu'une utilité informationnelle sont nommées « archives internes ». Par abus de langage, nous nommerons archives fiscales de telles archives externes. Les archives internes se doivent de respecter les obligations de protection de données personnelles.

[0006] Dans le même temps, de multiples lois et règlements imposent aux entreprises une destruction, une anonymisation ou une pseudoanonymisation de données personnelles. C'est par exemple le cas du Règlement Général sur la Protection des Données, également connu sous l'acronyme RGPD (ou GPDR pour « *General Data Protection Regulation* » selon une terminologie anglo-saxonne), qui contraint chaque détenteur de telles données à ne conserver celles-ci qu'en présence de consentements explicites des personnes ressortissantes de l'Union Européenne concernées par lesdites données, pour des durées limitées et déterminées ou encore si et seulement si une finalité ou un objectif l'exige.

[0007] Les entreprises sont donc soumises à des obligations contradictoires (obligation de conservation, obligation de suppression) dans leurs applications de gestion et également dans leurs archives.

[0008] Par ailleurs, les applications informatiques sont mises en œuvre par des ordinateurs personnels, ou plus largement par tous dispositifs électroniques adaptés (téléphones mobiles intelligents, tablettes tactiles, etc.), par des serveurs informatiques distants et coopérant avec des stations locales ou des ordinateurs personnels au travers de réseaux informatiques de communication filaire ou sans fil ou encore par des solutions à base d'informatique dite « sans serveur » ou « *serverless* », selon une terminologie anglo-saxonne, de conteneurs ou d'autres technologies adaptées. D'un point de vue technique, un tel système informatique mettant en œuvre une application de gestion de données s'apparente à un ordinateur comportant une unité de traitement consistant en un ou plusieurs microprocesseurs ou microcontrôleurs mettant en œuvre des instructions de programmes d'un système d'exploitation de ressources matérielles (mémoires de données ou de programmes, périphériques, etc.) et des applications logicielles hébergées, telles qu'un ERP. Lesdites instructions de programme sont chargées dans une ou plusieurs mémoires dudit système informatique en communication avec ladite unité de traitement. On entend par « mémoire ou moyen de mémorisation » toute mémoire informatique que celle-ci soit volatile ou non. Une mémoire non volatile est une mémoire informatique dont la technologie permet de conserver ses

données en l'absence d'une alimentation en énergie électrique. Elle peut contenir des données résultant de saisies, de calculs, de mesures et/ou des instructions de programmes. Les principales mémoires non volatiles actuellement disponibles sont de type inscriptible électriquement, telles que l'EPROM (« *Erasable Programmable Read-Only Memory* », selon une terminologie anglo-saxonne) ou encore inscriptibles et effaçables électriquement, telle que l'EEPROM (« *Electrically-Erasable Programmable Read-Only Memory* »), flash, SSD (« *Solid-State Drive* », selon une terminologie anglo-saxonne), etc. Les mémoires non volatiles se distinguent des mémoires dites « volatiles » dont les données sont perdues en l'absence d'une alimentation électrique. Les principales mémoires volatiles actuellement disponibles sont de type RAM (« *Random Access Memory* » selon une terminologie anglo-saxonne ou encore nommée « mémoire vive »), DRAM (mémoire vive dynamique, nécessitant une réactualisation régulière), SRAM (mémoire vive statique nécessitant une telle réactualisation lors d'une sous-alimentation électrique), DPRAM, VRAM, etc.

[0009] L'obligation de conservation des données d'une entreprise est également confrontée à l'obsolescence des logiciels d'ERP. Se pose ainsi régulièrement la question pour une entreprises de devoir arrêter des versions devenues obsolètes de logiciels de gestion de données au profit d'une nouvelle version ou d'un nouveau système. Or, un tel processus de décommissionnement (ou « *decommissioning* » selon une terminologie anglo-saxonne) d'une application obsolète et de mise en œuvre d'une migration desdites données et autres documents apparaît complexe et coûteux. Pour cette raison, des systèmes dits « systèmes hérités » demeurent trop souvent exploités, alors même que des solutions plus modernes et efficaces sont disponibles, par crainte des risques et des coûts qu'engendrerait une telle migration des données vers un nouveau système de gestion.

[0010] Il n'existe pas à cette heure de solution satisfaisante pour répondre à ces besoins souvent contradictoires. En effet, les obligations de conservation priment dans de nombreux pays au détriment des contraintes liées à la protection des données personnelles. De ce fait, les acteurs proposant des solutions pour satisfaire au Règlement Général sur la Protection des Données excluent le volet « conservation ». La réciproque est également et généralement vraie. Une solution de stockage « non modifiable » garantissant la non-destruction des informations pour une durée prédéfinie, contrevient aux obligations évolutives de destruction de données.

[0011] Généralement, les obligations de conservation, par exemple les obligations fiscales, priment sur les obligations de destructions, par application du RGPD par exemple. Nous pourrions donc considérer qu'il suffit de conserver les archives fiscales pendant la durée légale, puis de les supprimer afin de respecter l'ensemble des obligations. En pratique, pour des applications logicielles couvrant un spectre multiple d'entités de

contrôles (par exemple les applications multi-sociétés et multi-pays), multiplier les archives fiscales devient extrêmement complexe et identifier les règles de conservation quasi impossible. Le besoin existe de pouvoir constituer une archive qui soit à la fois une archive fiscale et une archive interne.

[0012] La présente invention a donc pour objet de proposer un procédé de gestion de données concernées par des obligations contradictoires de conservation et de mise à jour (y compris la suppression) pour être préférentiellement mis en œuvre dans le cadre d'une application logicielle dite « LSA », acronyme anglo-saxon de « *Legacy System Application* », c'est-à-dire une application permettant de gérer les données d'un « système hérité » également connu sous l'expression anglo-saxonne « *legacy system* ».

[0013] L'invention répond aux inconvénients soulevés par l'état de la technique en permettant notamment de constituer une archive, par exemple une archive fiscale, d'en garantir son statut d'archive fiscale, tout en satisfaisant aux règles de protection, voire de destruction, des données personnelles.

[0014] Parmi les nombreux avantages procurés par l'invention, nous pouvons mentionner plus particulièrement que l'invention permet de :

- simplifier et permettre une migration (c'est-à-dire un processus de décommissionnement ou « *decommissioning* » selon une terminologie anglo-saxonne) des données et de documents depuis une application de gestion de données et/ou de documents obsolète vers une nouvelle application et ainsi répondre à la problématique des systèmes dit « systèmes hérités » qui demeurent trop souvent en ligne, alors même que des solutions plus modernes et efficaces sont disponibles ;
- répondre aux besoins contradictoires de conservation et de mise à jour de données personnelles ;
- prévenir toute duplication de données (par exemple en multipliant les archives fiscales par pays) selon leurs fonction et utilisation et ainsi réduire les coûts des infrastructures assurant une telle gestion de données, en termes d'interopérabilité, de volumes de stockage, d'émission de CO2, etc. ;
- constituer des archives permettant de répondre aux attentes d'un contrôle externe, comme émanant du fisc par exemple ;
- conserver des archives pour les informations possiblement pertinentes qui ne le sont pas suffisamment pour être intégrées au système cible.

[0015] A cette fin, l'invention prévoit un procédé de gestion de données conçu pour être mis en œuvre par une unité de traitement d'un système informatique comportant également une première mémoire de données accessible en lecture et en écriture par ladite unité de traitement. Pour répondre aux inconvénients soulevés par l'état de la technique, un

tel procédé de gestion de données est agencé de sorte que :

- lesdites données sont stockées en clair dans une première archive unitaire au sein de la première mémoire de données dudit système informatique, ladite première archive unitaire comportant une ou plusieurs archives élémentaires comprenant chacune un ou plusieurs champs ;
- ledit procédé comporte :
  - une étape de création, dans ladite première mémoire de données, d'un conteneur de données dont les enregistrements décrivent respectivement les archives élémentaires de ladite première archive unitaire et pour lesquels un attribut d'enregistrement correspond à un champ d'archive élémentaire, chaque enregistrement dudit conteneur de données comportant en outre un attribut supplémentaire dont la valeur consiste en une clé unique désignant l'information contenue dans un enregistrement dudit conteneur de données ;
  - une étape de création, dans ladite première mémoire de données, d'un conteneur associatif comportant un enregistrement associé à chaque enregistrement du conteneur de données, chaque enregistrement dudit conteneur associatif comportant un attribut pour contenir la clé unique désignant l'enregistrement du conteneur de données qui lui est ainsi associé, des attributs pour respectivement désigner la première archive unitaire et l'archive élémentaire de cette dernière à partir desquels l'enregistrement du conteneur de données a été initialisé ;
  - une étape :
    - de création, dans la première mémoire de données :
      - d'autant de clés de chiffrement/déchiffrement qu'il existe d'enregistrements dans le conteneur de données ;
      - d'une deuxième archive unitaire à partir de la première archive unitaire et desdites clés de chiffrement/déchiffrement de sorte que chaque champ d'une même archive élémentaire de ladite deuxième archive unitaire résulte d'une opération de chiffrement du champ correspondant de la même archive élémentaire de ladite première archive unitaire par l'une desdites clés de chiffrement/déchiffrement ;
    - de mise à jour d'un attribut dans l'enregistrement du

conteneur associatif associé à l'enregistrement du conteneur de données de sorte que ledit attribut désigne la clé de chiffrement/déchiffrement utilisée pour chiffrer les champs de l'archive élémentaire de ladite première archive unitaire ;

- une étape de suppression de la première archive unitaire dans la première mémoire de données.

[0016] Pour initialiser la mise en œuvre d'un tel procédé, par exemple à la suite d'un processus de décommissionnement d'une application tierce, les données gérées peuvent être initialement stockées en clair dans une archive unitaire tierce, dite « archive unitaire d'entrée » au sein d'une deuxième mémoire de données dudit système informatique, ladite archive unitaire tierce comportant une ou plusieurs archives élémentaires comprenant chacune un ou plusieurs attributs. Dans ce cas, un procédé selon l'invention peut comporter une première étape de lecture de l'archive unitaire d'entrée dans la deuxième mémoire de données et de création dans la première mémoire de données de la première archive unitaire à partir de ladite archive unitaire d'entrée de sorte que lesdites première archive unitaire et l'archive unitaire d'entrée soient identiques.

[0017] Lorsqu'un procédé de gestion de données selon l'invention comporte un traitement de mise à jour consistant en une suppression d'un enregistrement du conteneur de données décrivant une archive élémentaire, ledit traitement peut comporter une étape d'effacement de l'attribut désignant la clé de chiffrement/déchiffrement dans l'enregistrement du conteneur associatif correspondant audit enregistrement du conteneur de données.

[0018] En variante, lorsqu'un procédé de gestion de données selon l'invention comporte un traitement de mise à jour consistant en une modification d'un attribut d'un enregistrement du conteneur de données décrivant une archive élémentaire, ledit traitement peut comporter :

- une étape de mise à jour dudit attribut dans le conteneur de données ;
- une étape de création d'une clé de chiffrement/déchiffrement et d'une troisième archive unitaire dont une archive élémentaire correspond à l'enregistrement du conteneur de données dont l'attribut a été modifié et pour laquelle chaque champ d'archive élémentaire correspond à chaque attribut dudit enregistrement chiffré par la clé de chiffrement/déchiffrement préalablement créée ;
- une étape :
  - d'effacement de l'attribut désignant la clé de chiffrement/déchiffrement dans l'enregistrement du conteneur associatif correspondant à l'enregistrement mis à jour du conteneur de données ;

- de création, dans ledit conteneur associatif, d'un nouvel enregistrement associé à l'enregistrement mis à jour dans le conteneur de données, le nouvel enregistrement dudit conteneur associatif comportant :
  - un attribut pour contenir la clé unique désignant ledit enregistrement du conteneur de données qui a été mis à jour ;
  - des attributs pour respectivement désigner la troisième archive unitaire et l'archive élémentaire de cette dernière qui vient d'être initialisée ;
  - un attribut désignant la clé de chiffrement/déchiffrement utilisée pour chiffrer les champs de l'archive élémentaire de ladite troisième archive.

[0019] Pour terminer la mise en œuvre d'un tel procédé de gestion de données conforme à l'invention en vue d'une exploitation tierce desdites données, celui-ci peut comporter un traitement d'exportation des données comprenant :

- une étape de lecture de tout enregistrement du conteneur associatif comportant un attribut désignant une clé de chiffrement/déchiffrement ;
- une étape de :
  - lecture de l'archive élémentaire de l'archive unitaire conjointement désignées par ledit enregistrement lu au sein du conteneur associatif ;
  - déchiffrement des champs de ladite archive élémentaire lue à partir de la clé de chiffrement/déchiffrement désignée également par l'enregistrement lu dans le conteneur associatif ;
  - création d'une archive élémentaire dans une quatrième archive unitaire, dite « archive unitaire de sortie » dont les champs correspondent respectivement auxdits champs déchiffrés de l'archive élémentaire lue.

[0020] De manière avantageuse, un procédé de gestion de données selon l'invention peut comporter une étape d'enregistrement dans une troisième mémoire de données du système informatique de ladite archive unitaire de sortie.

[0021] Selon un mode de réalisation avantageux, un tel procédé de gestion de données conforme à l'invention peut comporter une étape de vérification de la pertinence de ladite archive unitaire de sortie, l'étape d'enregistrement dans une troisième mémoire de données n'étant mise en œuvre que si ladite étape de vérification atteste que ladite archive unitaire de sortie est pertinente.

[0022] De la même manière, l'invention prévoit qu'un tel procédé conforme à l'invention puisse comporter une étape de vérification de la pertinence de la première archive unitaire créée, préalable à la mise en œuvre de l'étape de création du conteneur de

données et/ou de l'étape de création du conteneur associatif, au moins l'une de ces deux étapes de création n'étant mise en œuvre que si ladite étape de vérification atteste que ladite première archive unitaire est conforme à l'archive unitaire d'entrée, ladite première archive unitaire étant supprimée de la première mémoire de données et l'étape de création dans ladite première mémoire de données d'une première archive unitaire à partir de l'archive unitaire d'entrée étant de nouveau mise en œuvre dans le cas contraire.

[0023] Selon ce dernier mode de réalisation avantageux, ladite étape de vérification de la pertinence de la première archive unitaire peut consister en la comparaison de codes de redondances respectivement et préalablement calculés à partir de l'archive unitaire d'entrée et de la première archive unitaire.

[0024] A l'instar de la précédente étape de vérification de la pertinence de la première archive unitaire, un procédé de gestion de données conforme à l'invention peut comporter une étape de vérification de la pertinence de la deuxième archive unitaire créée au regard de la première archive unitaire préalablement à la mise en œuvre de l'étape de suppression de la première archive unitaire dans ladite première mémoire de données, ladite étape de vérification consistant à :

- déchiffrer les champs de chaque archive élémentaire de la deuxième archive unitaire à l'aide de la clé de chiffrement/déchiffrement désignée dans l'enregistrement qui est associé à ladite archive élémentaire dans le conteneur associatif et créer une archive élémentaire de champs clairs dans une cinquième archive unitaire ;
- évaluer la conformité de ladite cinquième archive unitaire à ladite première archive unitaire ;

l'étape de suppression de la première archive unitaire dans la première mémoire de données, n'étant mise en œuvre que si ladite étape de vérification de la pertinence de ladite deuxième archive unitaire atteste que ladite cinquième archive unitaire est conforme à la première archive unitaire, ledit procédé comportant, dans le cas contraire, une étape de suppression dans la première mémoire de données, de ladite deuxième archive unitaire et de ladite cinquième archive unitaire provoquant une nouvelle mise en œuvre de l'étape de création d'une deuxième archive unitaire à partir de la première archive unitaire et de clés de chiffrement/déchiffrement.

[0025] Une telle étape de vérification de la pertinence de la deuxième archive unitaire créée au regard de la première archive unitaire peut consister en la comparaison de codes de redondances respectivement et préalablement calculés à partir des première et cinquième archives unitaire.

[0026] Pour ne pas alourdir inutilement la mise en œuvre d'un procédé de gestion de données selon l'invention et préserver ainsi les ressources matérielles de l'unité de

traitement mettant en œuvre ledit procédé et la première mémoire de données, l'étape de création d'une deuxième archive unitaire peut n'être mise en œuvre que si des données contenues dans la première archive unitaire sont concernées par des obligations contradictoires de conservation et de mise à jour.

[0027] Selon un mode de réalisation avantageux, chaque archive unitaire peut consister en un fichier plat, dont chaque archive élémentaire peut consister en une ligne dudit fichier plat.

[0028] Selon un mode de réalisation avantageux, un conteneur de données peut consister en une table courante et un conteneur associatif peut consister en une table associative.

[0029] Selon un deuxième objet, l'invention concerne un produit programme d'ordinateur comportant une ou plusieurs instructions de programme exécutables par l'unité de traitement d'un système informatique, lesdites instructions de programme étant chargeables dans une mémoire non volatile dudit système informatique et dont l'exécution par ladite unité de traitement provoque la mise en œuvre d'un procédé de gestion de données conforme à l'invention.

[0030] Selon un troisième objet, l'invention concerne un support de mémorisation lisible par un ordinateur comportant les instructions d'un tel produit programme d'ordinateur.

[0031] Selon un quatrième objet, l'invention concerne un système informatique comprenant une unité de traitement, une mémoire comprenant les instructions de programme d'un produit programme conforme à l'invention.

[0032] D'autres caractéristiques et avantages apparaîtront plus clairement à la lecture de la description qui suit et à l'examen des figures qui l'accompagnent parmi lesquelles :

[0033] [Fig.1] illustre l'architecture simplifiée d'un système informatique mettant en œuvre un procédé de gestion de données conforme à l'invention pour importer, mettre à jour et exporter de telles données concernées par des obligations de conservation et de mise à jour ;

[0034] [Fig.2] illustre un traitement d'importation de données d'un tel procédé de gestion de données conforme à l'invention ;

[0035] [Fig.3] illustre un exemple d'une table courante de données exploitée par un procédé de gestion de données conforme à l'invention, une telle table comportant pour tout enregistrement un attribut clé prévu pour identifier de manière unique une information unitaire répartie dans les autres attributs de chaque enregistrement ;

[0036] [Fig.3A] illustre un exemple particulier d'une telle table courante de données exploitée par un procédé de gestion de données conforme à l'invention, une telle table comprenant deux enregistrements dont les attributs caractérisent respectivement des clients d'un fournisseur exploitant une application de gestion de données mettant en œuvre un procédé de gestion de données conforme à l'invention ;

[0037] [Fig.4] illustre un exemple d'une table associative exploitée par un procédé de

gestion de données conforme à l'invention, une telle table associative comportant pour tout enregistrement des attributs associant une information unitaire traduite dans au moins une table courante de données et un fichier plat traduisant ladite information unitaire à des fins de conservation ;

[0038] [Fig.4A] illustre un exemple particulier d'une telle table associative en lien avec l'exemple de la table courante illustrée par la [Fig.3A] ;

[0039] [Fig.5] illustre un traitement préalable à une première mise à jour de données d'un procédé de gestion de données conforme à l'invention ;

[0040] [Fig.6] illustre l'exemple particulier de la table courante de données et de la table associative qui lui est associée, tables respectivement illustrées par les figures 3A et 4A, ladite table associative étant modifiée, préalablement à toute mise à jour de ladite table courante, par la mise en œuvre d'un tel traitement préalable à une première mise à jour de données conforme à l'invention et illustré par la [Fig.5] ;

[0041] [Fig.7] illustre un traitement de mise à jour de données d'un procédé de gestion de données conforme à l'invention ;

[0042] [Fig.8] illustre l'exemple particulier de la table courante de données et de la table associative qui lui est associée, tables respectivement illustrées par les figures 3A, 4A et 6, ladite table associative étant modifiée conjointement à la suppression d'un enregistrement de ladite table courante, par la mise en œuvre d'un tel traitement de mise à jour de données conforme à l'invention et illustré par la [Fig.7] ;

[0043] [Fig.9] illustre l'exemple particulier de la table courante de données et de la table associative qui lui est associée, tables respectivement illustrées par les figures 3, 3A, 6 et 8, ladite table associative étant modifiée conjointement à la modification d'un attribut d'un enregistrement de ladite table courante, par la mise en œuvre d'un tel traitement de mise à jour de données conforme à l'invention et illustré par la [Fig.7] ;

[0044] [Fig.10] illustre un traitement d'exportation de données d'un procédé de gestion de données conforme à l'invention ;

[0045] [Fig.11] illustre un exemple de fichier exporté résultant de la mise en œuvre d'un traitement d'exportation de données d'un procédé de gestion de données conforme à l'invention.

[0046] Les données dont on souhaite assurer une gestion peuvent prendre des formes plus ou moins complexes. Des formes simples traduisent par exemple des préférences d'utilisateurs pour spécifier une ou plusieurs langues dans laquelle certaines informations sont restituées via une interface homme-machine de sortie, la charte graphique à exploiter pour présenter de telles informations, etc. D'autres données sont plus complexes et interreliées sur la base desquelles un logiciel de gestion de données assure des fonctions d'ajout, de modification, de suppression ou de consultation. D'autres données enfin peuvent identifier ou désigner des documents qui leurs sont

associés.

[0047] Il existe plusieurs formats de sauvegarde ou de stockage de telles données que l'on englobe généralement sous le terme de « base de données ». Parmi de tels formats, nous pouvons citer les fichiers plats (ou « flat files » selon une terminologie anglo-saxonne) et les tables d'une base de données relationnelles, structurant davantage les données que les fichiers plats. Dans un cas comme dans l'autre, les données sont organisées selon certains critères en vue de permettre leur exploitation par un logiciel ou une application de gestion.

[0048] On entend par « fichier plat », un fichier texte, non compilé, dans lequel sont stockées des données. Chaque ligne d'un tel fichier plat comporte un ou plusieurs champs généralement séparés par un délimiteur ou par une césure définie par un nombre de caractères déterminé pour chacun desdits champs. A titre d'exemple, la virgule, le point-virgule ou encore les deux points sont souvent exploités en tant que délimiteurs. Les fichiers plats sont largement utilisés dans les projets de sauvegarde de données ou d'importation/exportation de données en raison de la facilité avec laquelle ils véhiculent des données entre deux serveurs informatiques par exemple. Ils sont donc généralement exploités pour constituer des archives dont on peut assurer l'intégrité et l'exploitabilité. Les fichiers plats ne contiennent toutefois aucune relation ou lien entre différentes lignes ou différents champs au sein d'un même fichier plat ou à destination d'autres lignes et/ou champs de fichiers plats distincts. Un fichier plat est ainsi un conteneur de données simple selon lequel les champs d'une même ligne constitue un ensemble de données indépendantes. Un fichier plat peut néanmoins contenir des métadonnées, par exemple définies dans les premières lignes du fichier. Nous pouvons citer différents formats de fichiers plats parmi les plus connus dont :

- le fichier « TXT » qui est un fichier texte non formaté et exploitable par toute application logicielle ;
- le fichier « CSV » (abréviation anglo-saxonne de « *Comma Separated Values* » pour lequel chaque champ est dépourvu de formatage et est séparé du suivant par une virgule ;
- le fichier « XML » (abréviation anglo-saxonne de « *Extensible Markup Language* » pour lequel chaque champ est associé à des balises pour décrire et structurer les données ;
- le fichier « AIS » (abréviation anglo-saxonne de « Audit Information System » format d'audit SAP (marque déposée), ce fichier contenant des métadonnées.

[0049] Lorsqu'une base de données nécessite l'exploitation de données plus structurées et interreliées, celle-ci comporte des tables en lieu et place de simples fichiers plats. On entend dans ce document par « table » un conteneur de données comportant des in-

formations sur les relations entre les différentes données, telles qu'un type de produit ou les caractéristiques d'une personne. Chaque ensemble d'informations associées à un produit ou à une personne spécifique par exemple constitue un enregistrement de la table et les colonnes énumèrent des attributs qui se rapportent à ce produit ou à cette personne. Les bases de données relationnelles sont composées d'un ensemble de tables qui peuvent être accessibles et reconstruites de différentes manières. Pour accéder à l'information, il est requis d'utiliser des requêtes pour interroger de façon interactive les données contenues dans la base de données relationnelle et pour collecter les données dans le cadre de rapports. Une clé est un groupe d'attributs qui permet d'identifier de façon univoque une ligne dans une relation comme les attributs « prénom » et « nom » d'une personne. On distingue généralement deux types de tables. Le premier concerne les tables dites « tables courantes » de données qui comportent des enregistrements dont les attributs caractérisent conjointement un produit ou une personne pour reprendre l'exemple précédent. Le deuxième type concerne des tables dites « tables associatives » qui assurent des liaisons entre deux tables courantes par exemple. Par extension, dans ce document, on parlera de :

- « conteneurs de données » pour décrire tous conteneurs (tables courantes, fichiers plats ou équivalent) qui caractérisent, par exemple, des produits, services, sujets, etc. ;
- « conteneur de données associatif », ou de « conteneur associatif » par souci de concision, pour décrire tout conteneur (table associative, fichier plat ou équivalent) qui assure des liaisons entre conteneurs de données (tables courantes ou fichiers plats par exemple).

[0050] La [Fig.1] illustre un exemple d'un système informatique 1 de gestion de données. Un tel système informatique peut consister en un ordinateur, un serveur informatique ou tout autre dispositif électronique ou ensemble de dispositifs électroniques. De manière simplifiée, la [Fig.1] illustre un système informatique comportant une unité de traitement PU, sous la forme d'un ou plusieurs microprocesseurs ou microcontrôleurs coopérant avec une ou plusieurs mémoires de données DMi, DMw, DMo. Ainsi, par exemple, DMi caractérise une application « *legacy* », DMw caractérise l'application actuelle, est DMo caractérise une application qui prendra la relève de l'application actuelle. Lesdites mémoires de données DMi, DMw, DMo peuvent être des entités ou des ensembles d'entités physiquement dissociés les uns des autres ou bien être confondus pour certains au sein d'une même entité physique. En outre, ces mêmes mémoires de données DMi, DMw, DMo peuvent coopérer directement, par voie filaire, c'est-à-dire via un bus de communication, avec ladite unité de traitement PU ou bien être distantes de ladite unité de traitement et hébergées par des systèmes informatiques tiers. L'unité de traitement PU coopère alors avec une ou plusieurs mémoires

de données via un réseau de communication avec ou sans fil, tel qu'Internet. A titre d'exemple illustratif, considérons en lien avec la [Fig.1], que le système informatique 1 comporte un serveur informatique dont l'unité de traitement PU est agencée pour mettre en œuvre un procédé de gestion de données 10 exploitant une mémoire de données DMw locale et qui coopère, en mode SaaS (acronyme anglo-saxon de « *Software-as-a-Service* ») ou équivalent, avec des ordinateurs distants comportant respectivement les mémoires de données DMi et DMo. Le SaaS est un modèle d'exploitation commerciale de logiciels selon lequel lesdits logiciels sont installés, maintenus et exploités sur des serveurs distants chez un tiers plutôt que sur le système informatique de l'utilisateur final. Ce dernier exploite un service en ligne généralement moyennant le paiement d'un abonnement auprès du tiers.

[0051] Pour adapter le fonctionnement d'un système informatique 1, un ou plusieurs programmes d'ordinateur comportant des instructions exécutables par l'unité de traitement PU dudit système informatique 1, sont chargées dans une mémoire de programmes PM, lesdites instructions de programme provoquant la mise en œuvre de procédés idoines par ladite unité de traitement PU.

[0052] La [Fig.1] illustre un exemple de description fonctionnelle très simplifiée d'un procédé 10 de gestion de données conforme à l'invention et mis en œuvre par un système informatique 1. Principalement, un tel procédé 10 consiste en un premier traitement 100 visant à importer des données d'archive depuis une mémoire de données DMi. Le traitement 100 peut être une copie d'une archive fiscale déjà constituée, ou de façon plus complète l'extraction d'une archive fiscale. Ces données, dont certaines peuvent être concernées par une double obligation de conservation et de mise à jour, constituent des archives élémentaires, éventuellement groupées sous la forme d'archives unitaires. A titre d'exemple préféré mais non limitatif, de telles archives unitaires consistent en un ou plusieurs fichiers plats FFi dont les lignes, voire des champs de telles lignes, forment respectivement des archives élémentaires. D'autres formats d'archives unitaires pourraient être retenus, comme des tables de données, en lieu et place de fichiers plats. Il suffit que lesdits formats respectent les critères de traçabilité et d'exploitabilité évoqués précédemment en lien avec la notion d'archives internes et externes, par exemple.

[0053] Des données ont pu également servir à constituer ou identifier un ensemble de documents Di, par exemple, des contrats, des feuilles de salaire, des factures. Pour constituer une telle forme de stockage sous la forme d'un ou plusieurs fichiers plats FFi, ces données ont pu faire l'objet d'un processus d'extraction de données depuis une base de données relationnelle devenue obsolète. Les archives sous la forme de fichiers plats FFi peuvent être respectivement associées à un ensemble de codes de redondance ou d'empreintes Hi destinés à prouver leur intégrité. Un tel code de redondance peut

par exemple résulter d'une mise en œuvre d'une fonction cryptographique de hachage (également connue sous la terminologie anglo-saxonne « *Secure Hash Function* » ou par l'acronyme SHA). Il existe de nombreux algorithmes, parmi lesquels nous pouvons citer SHA-1, SHA-256, SHA-512, MD5, pour réaliser une telle fonction de hachage. La fonction cryptographique de hachage est principalement agencée de sorte qu'une altération du fichier aussi mineure soit-elle, provoque une claire modification dudit code de redondance si celui-ci est recalculé à l'issue de ladite altération. L'invention ne saurait être limitée par ces seuls exemples de production de codes de redondance ni même par leur exploitation ou existence. Le traitement d'importation 100 consiste principalement à traduire les données d'archive contenues dans le ou les fichiers plats FFi, pour que les informations contenues se retrouvent sous la forme de fichiers plats ou d'archives unitaires de travail FFw, d'une ou plusieurs tables courantes de données Tw, ou plus généralement de tous conteneurs de données, et d'une ou plusieurs tables associatives CTw d'une base de données pour en faciliter la gestion. Ces nouvelles structures de données, conteneurs et archives unitaires, sont enregistrées dans la mémoire de données locales DMw. Un tel traitement d'importation 100 sera décrit plus en détail en lien avec les figures 2, 3 et 4 sous une forme générale et en en lien avec les figures 2, 3A et 4A au travers d'un exemple d'application concret et simplifié.

[0054] Comme l'indique la figure 1, un procédé de gestion de données 10 selon l'invention comporte un processus de mise à jour 300 desdites données pour mettre à jours certaines informations, c'est-à-dire modifier et/ou supprimer ces dernières. Préalablement à la mise en œuvre d'un tel traitement de mise à jour 300, un procédé 10 comporte un pré-traitement 200 visant à chiffrer les fichiers de travail FFw à partir de clés de chiffrement/déchiffrement $\overrightarrow{CK}$. Enfin, pour mettre un terme à la gestion des données en vue d'une migration subséquente, un tel procédé 10 peut comporter un traitement 400 de *decommissioning,* selon une terminologie anglo-saxonne, visant à exporter les données sous la forme de fichiers plats FFo à destination d'une mémoire de données de sortie DMo. La gestion des documents d'entrée Di, copiés sous la forme de documents de travail Dw dont certains pourront être copiés pour devenir des documents de sortie Do ne font pas directement partie de l'invention. En revanche, l'invention prévoit que des données de désignation, par exemple des indexes, désignant de tels documents Di, Dw, Do puissent être associées à ces derniers. Ces données de désignation peuvent être traitées comme des archives élémentaires ou unitaires. Ainsi, la gestion de tels documents Di, Dw, Do peut être considérée comme adressée, indirectement, par un procédé 10 conforme à l'invention.

[0055] Examinons plus en détail, en lien avec les figures 2, 3 et 4 un traitement d'importation de données 100 d'un procédé de gestion de données 10 conforme à l'invention.

[0056] Considérons que les données d'archive sont initialement stockées en clair dans un fichier plat FFi au sein d'une mémoire de données d'entrée DMi du système informatique 1. Un tel fichier plat FFi comporte une ou plusieurs lignes, décrivant respectivement des archives élémentaires, comprenant chacune un ou plusieurs champs pour respectivement stocker des informations distinctes comme l'indique la [Fig.3].

[0057] Un traitement d'importation 100 peut consister tout d'abord en une étape 101 de lecture de l'archive unitaire, par exemple, sous la forme du fichier plat FFi et de création 102 dans la mémoire de données de travail DMw d'une deuxième archive unitaire, par exemple, sous la forme d'un deuxième fichier plat FFw-1 à partir dudit premier fichier plat FFi de sorte que lesdits premier et deuxième fichiers plats FFi et FFw-1 soient identiques. Comme l'indique la [Fig.3], un fichier plat FFw-1 consiste ainsi en une pluralité de m lignes FFw-1R1 à FFw-1Rm, chacune comportant n champs TFd1, TFd2 à TFdn. Selon l'exemple illustré par la [Fig.3], ces derniers sont séparés par un délimiteur d, en l'espèce le point-virgule. L'invention ne saurait être limitée par ce seul choix de délimiteur et pourrait exploiter ':' ou '/' en lieu et place dudit point-virgule.

[0058] Afin de s'assurer que le fichier FFw-1 est bien conforme au fichier plat FFi et/ou que ce dernier n'a pas été altéré préalablement ou durant l'étape 101, un traitement 100 peut avantageusement comporter une étape 103 de vérification de la pertinence du fichier plat FFw-1. Une telle étape peut consister en la comparaison de codes de redondances Hi et H1 respectivement et préalablement calculés à partir des fichiers plats FFi et FFw-1. Si l'étape 103 atteste que lesdits codes de redondances Hi et H1 sont différents, le fichier plat FFw-1 est jugé non conforme, en l'espèce non identique, au fichier plat d'entrée FFi. Dans ce cas, situation illustrée par le lien 103-n sur la [Fig.2], ledit fichier plat FFw-1 est supprimé, en une étape 104, de la mémoire de données DMw et l'étape 102 de création d'un nouveau fichier plat FFw-1 à partir dudit fichier plat d'entrée FFi est de nouveau mise en œuvre. Dans le cas contraire, situation illustrée par le lien 103-y en [Fig.2], le traitement 100 peut suivre son cours.

[0059] Quelles que soient les modalités de constitution de l'archive unitaire FFw-1, un traitement 100 comporte une étape 105 de création, dans la première mémoire de données DMw, d'un conteneur de données Tw, sous la forme avantageuse d'une table courante, dont les enregistrements TwR1, TwR2, TwR3, ..., TwRm , désignent chacun une archive élémentaire et correspondent respectivement aux lignes FFw-1R1, FFw-1R2, FFw-1R3, ..., FFw-1Rm du fichier plat FFw-1 constituant une archive unitaire. Une telle table courante Tw est telle qu'un attribut TwRxAn d'enregistrement correspond à un champ FFw-1RxCn de ligne.

[0060] La table courante Tw est créée à l'étape 105 de telle sorte que chaque enregistrement comporte en outre un attribut supplémentaire TwRxAUK dont la valeur UK1, UK2,

UK3, UKm consiste en une clé unique désignant de manière univoque l'information contenue dans un enregistrement déterminé de ladite table courante Tw et par conséquent l'information (archive élémentaire) contenue dans une ligne correspondante du fichier plat FFw-1 (archive unitaire). Une telle clé unique UK1, UK2, UK3, UKm peut être élaborée selon différentes techniques, parmi lesquelles nous pouvons citer la mise en œuvre d'une fonction cryptographique de hachage de tout ou partie des valeurs des attributs d'un enregistrement de la table Tw à l'exception de l'attribut TwRxAUK, le tirage (sans doublons) d'un nombre aléatoire, l'incrémentation d'un nombre, etc.

[0061] En lien avec les figures 2 et 4, l'étape 105 consiste en outre à créer, dans la mémoire de données DMw, un conteneur associatif CTw, sous la forme avantageuse d'une table associative, comportant un enregistrement CTwR1, CTwR2, CTwR3, …, CTwRk associé à chaque enregistrement TwR1, TwR2, TwR3, …,TwRm de la table courante Tw. Chaque enregistrement de ladite table associative CTw comporte un attribut UK pour contenir la clé unique désignant l'enregistrement de la table courante qui lui est ainsi associé. Chaque enregistrement de ladite table associative CTw comporte en outre des attributs FN, RN pour respectivement désigner le fichier plat FFw-1 ainsi que la ligne de ce dernier à partir desquels l'enregistrement de la table courante Tw qui lui est associé a été initialisé. La clé unique UK1, UK2, UK3, UKm permet ainsi d'associer, via la table associative CTw, une ligne FFw-1R1, FFw-1R2, FFw-1R3, …, FFw-1Rm du fichier FFw-1 à un enregistrement TwR1, TwR2, TwR3, …, TwRm de la table Tw. A titre optionnel et avantageux, chaque enregistrement CTwR1, CTwR2, CTwR3, …, CTwRk de la table associative CTw peut comporter des attributs accessoires CD, CID pour désigner respectivement l'horodatage CD et l'identifiant CID de la personne ayant déclenché le traitement d'importation 100. Il est ainsi possible de connaître, par exemple à des fins d'audits, l'instant d'importation des données et la personne qui en a eu la charge. Un enregistrement CTwR1, CTwR2, CTwR3, …, CTwRk peut en outre comporter d'autres attributs, comme les attributs référencés CK, ED et EID exploités par le traitement 200. Ces derniers seront décrits ultérieurement en lien avec la [Fig.5]. Les attributs que l'on peut qualifier de « désignation » comme UK, FN, RN, pour ne citer qu'eux, peuvent s'entendre comme une valeur, un index vers une table ou un conteneur de données d'indirection, un pointeur, une adresse non ambigüe ou tout autre moyen permettant de désigner un contenu, en l'espèce une archive unitaire, une archive élémentaire ou toute structure de données décrivant de telles archives ou contenus.

[0062] Les figures 3A et 4A illustrent, au travers d'un exemple d'application concret et simplifié, la mise en œuvre d'un traitement d'importation 100 tel que décrit précédemment. Les données d'entrée sont stockées dans une unique archive unitaire sous

la forme avantageuse d'un unique fichier plat FFi1 comportant deux informations décrivant des fournisseurs et correspondant à deux lignes (ou plus généralement deux archives élémentaires) au sein dudit fichier FFi1 (ou archive unitaire). Ces dernières comportent chacune quatre champs pour désigner respectivement un code fournisseur, le pays dudit fournisseur, le nom de la personne référente et une adresse postale. A l'issue de la mise en œuvre de l'étape 101, un fichier plat de travail FFw1-1 est créé à l'identique du fichier plat FFi1. Comme l'indique la [Fig.3A], le fichier plat FFw1-1 comporte :

- la ligne FFw1-1R1 qui correspond à un premier fournisseur français et comprend les champs « 00001 », « FR », « Dupond » et « 14 rue de l'océan » séparés deux à deux par le délimiteur « ; » ;
- la ligne FFw1-1R2 qui comprend les champs « 00002 », « DE », « Meyer » et « Zentrale Straße » également séparés deux à deux par le délimiteur « ; ».

[0063] A l'issue de la mise en œuvre de l'étape 105, un conteneur de données sous la forme d'une table courante Tw1 est créé ainsi qu'un conteneur associatif CTw1 sous la forme avantageuse d'une table associative. Nous pouvons constater que ladite table courante Tw1 comporte deux enregistrements Tw1R1 et Tw1R2 comprenant chacun cinq attributs. Les quatre premiers attributs PID, PC, PN et PA correspondent respectivement aux champs d'une ligne (archive élémentaire) du fichier plat FFw1-1. Le cinquième attribut UK est agencé pour comporter une clé unique qui désigne sans ambiguïté l'information d'un fournisseur contenue dans l'enregistrement. Ainsi, la [Fig.3A] décrit une table courante Tw1 comprenant :

- l'enregistrement Tw1R1 correspondant à la première ligne FFw1-1R1 du fichier plat FFw1-1 dont l'attribut PID comporte la valeur « 00001 », l'attribut PC comporte la valeur « FR », l'attribut PN comporte la valeur « Dupond » et l'attribut PA comporte la valeur « 14 rue de l'océan » ; ledit enregistrement comprend en outre l'attribut UK qui comporte la valeur unique « 12345678 » désignant de manière univoque l'information contenue dans ledit enregistrement Tw1R1 ;
- l'enregistrement Tw1R2 correspondant à la deuxième ligne FFw1-1R2 du fichier plat FFw1-1 dont l'attribut PID comporte la valeur « 00002 », l'attribut PC comporte la valeur « DE », l'attribut PN comporte la valeur « Meyer » et l'attribut PA comporte la valeur « Zentrale Straße » ; ledit enregistrement comprend en outre l'attribut UK qui comporte la valeur unique « 98765432 » désignant de manière univoque l'information contenue dans ledit enregistrement Tw1R2.

[0064] Précisons que si les archives FFi1 et/ou FFw1 sont constituées d'une façon telle que les champs PID sont naturellement uniques, alors les champs UK et PID pourraient

être confondus dans la table TW1.

[0065] L'étape 105 a également engendré la création de la table associative CTw1. Celle-ci comporte deux enregistrements CTw1R1 et CTw1R2 respectivement associés aux enregistrements Tw1R1 et Tw1R2 de la table courante Tw1. En effet, l'enregistrement CTw1R1 comporte un premier attribut UK qui comprend la valeur unique « 12345678 » désignant de manière univoque l'archive élémentaire décrivant l'information de fournisseur contenue dans l'enregistrement Tw1R1. Ledit enregistrement CTw1R1 comprend en outre un attribut FN désignant le fichier plat FFw1-1 et un attribut RN désignant la ligne 1, soit la ligne FFw1-1R1 dudit fichier. A titre optionnel, ledit enregistrement CTw1R1 comporte un attribut CD pour mémoriser l'horodatage (jour et/ou heure) ts1 de la création dudit enregistrement, donc de l'importation des données, et un attribut CID pour désigner un identifiant U012 d'un utilisateur en charge de réaliser ladite importation.

[0066] L'enregistrement CTw1R2 comporte quant à lui un premier attribut UK qui comprend la valeur unique « 98765432 » désignant de manière univoque l'archive élémentaire décrivant l'information de fournisseur contenue dans l'enregistrement Tw1R2 de la table courante Tw1. Ledit enregistrement CTw1R2 comprend en outre un attribut FN désignant le fichier plat FFw1-1 et un attribut RN désignant la ligne 2, soit la ligne FFw1-1R2 dudit fichier. A titre optionnel, ledit enregistrement CTw1R2 comporte un attribut CD pour mémoriser l'horodatage ts2 de la création dudit enregistrement, donc de l'importation des données, et un attribut CID pour désigner un identifiant U012 d'un utilisateur en charge de réaliser ladite importation.

[0067] Nous pouvons constater que les attributs CK, ED et EID illustrés par la [Fig.4A] ne sont pas initialisés à ce stade de la mise en œuvre du procédé 10. En revanche, la table associative CTw1 assure clairement un lien entre les enregistrements de la table courante Tw1 et les lignes (archives élémentaires) du fichier plat FFw1-1 à partir desquelles lesdits enregistrements de la table Tw1 ont été initialisés. Ainsi, une forme de « triptyque » a été constitué entre une information contenue dans une archive unitaire (fichier plat FFw1-1), la même information contenue dans un conteneur de données (table courante Tw1) et un lien entre lesdites informations par le biais du conteneur associatif (table associative CTw1). Le fichier plat FFw1-1 est non modifiable, la table Tw1 est modifiable et la table associative CTW1 assure le lien entre les deux.

[0068] La [Fig.5] illustre un exemple de description fonctionnelle d'un traitement 200 préalable à toute mise en œuvre subséquente de l'un des enregistrements d'une table courante Tw ou Tw1 créée par un traitement d'importation 100 tel que celui décrit en lien avec la [Fig.2] et la [Fig.3] ou 3A.

[0069] Un tel traitement 200 est essentiel pour se conformer aux contraintes du RGPD ou de

toute autre obligation visant à protéger les données personnelles par exemple. Il consiste principalement à chiffrer toute archive unitaire avec une clé de chiffrement/déchiffrement dédiée à chaque archive élémentaire de ladite archive unitaire. Une preuve de conservation des archives unitaires et élémentaires peut ainsi être garantie car toute archive élémentaire ultérieurement modifiée ne sera plus déchiffrable en utilisant ladite clé de chiffrement/déchiffrement originelle. La mise en œuvre d'un tel traitement 200 peut être automatiquement déclenchée en fin de mise en œuvre d'un traitement 100 d'importation ou, en variante, déclenchée à la suite d'une consigne d'un utilisateur du système mettant en œuvre un procédé selon l'invention. Selon un mode de réalisation alternatif, lesdits traitements successifs 100 et 200 pourraient ne constituer qu'un seul et même traitement déclenché à l'importation des données. Selon un autre mode de réalisation alternatif, le traitement 200 peut être lancé automatiquement, manuellement, sur un périmètre complet ou partiel, en une ou plusieurs fois.

[0070] Un tel traitement 200 consiste tout d'abord en une étape 201 pour lire, dans la mémoire de données de travail DMw, un fichier plat FFw-1 associé éventuellement à un code de redondance H1 caractérisant ce dernier, si un tel code H1 a été conjointement enregistré dans ladite mémoire de travail DMw, par exemple à l'issue de la mise en œuvre d'un traitement d'importation 100. Ledit traitement 200 comporte une étape 203 de création, dans la mémoire de données DMw, d'autant de clés de chiffrement/déchiffrement $\overrightarrow{CK}$ qu'il existe de lignes dans le fichier plat FFw-1 qui a été créé conjointement avec une table courante Tw à l'issue de la mise en œuvre du traitement d'importation 100. De telles clés de chiffrement/déchiffrement peuvent être créées selon toute technique connue. Elle peuvent par exemple être dérivées depuis une clé mère commune ou allouée par archive unitaire. Une telle dérivation peut être réalisée à partir de ladite clé mère et d'une graine aléatoire ou pseudoaléatoire. La longueur, par exemple égale à 128 ou 256 bits, desdites clés de chiffrement/déchiffrement est déterminée en fonction de l'algorithme de chiffrement/déchiffrement choisi. Un tel algorithme de chiffrement/déchiffrement peut être choisi parmi une multitude d'algorithmes de chiffrement symétrique connus tels que, de manière non limitative, le DES, Triple DES, l'AES, Twofish, Rrindjael, Blowfish, Serpent, RC6 ou Mars. Toutefois, l'invention ne serait être limitée par ces choix d'algorithmes.

[0071] Les clés de chiffrement/déchiffrement utilisées pour les archives unitaires présentent également quelques avantages complémentaires. La sécurité des systèmes d'information est assurée en partie grâce aux chiffrements. Les chiffrements se font en général au niveau d'une archive unitaire (fichier). Or, les modèles de données des applications de gestion les plus utilisés sont connus. Par exemple, une table dans le logiciel SAP commence par un champ à trois chiffres, quasiment toujours identique, suivi d'un numéro d'organisation d'achat, suivi d'un numéro de commande dont le

format commence quasi systématiquement par 450000. De telles similitudes dans les archives élémentaires et la grande quantité d'archives élémentaires dans une archive unitaire favorisent grandement la possibilité de déchiffrement par un tiers non autorisé. Le chiffrement de chaque archive élémentaire (qui peut être réalisé en complément des chiffrements classiques), rend le chiffrement plus sûr en ce sens qu'il porte sur moins de données. Le type de chiffrement n'a donc pas nécessairement à être très fort, puisqu'il existera des millions de clés différentes. Un chiffrage faible est également d'implémentation plus simple en termes de temps de traitement.

[0072] L'invention est décrite au travers d'une exploitation préférée d'algorithmes de chiffrement symétrique. Ainsi, une même clé $\overrightarrow{CK}$ peut être utilisée pour chiffrer et déchiffrer un contenu. En variante, l'invention pourrait exploiter un système de chiffrement/déchiffrement asymétrique notamment si la conservation des archives chiffrées et des clés doivent être détenues par des entreprises différentes.

[0073] Les clés de chiffrement/déchiffrement $\overrightarrow{CK}$ sont exploitées par ladite étape 203 pour créer un nouveau fichier plat FFw-2 à partir dudit fichier plat FFw-1. Selon l'invention, chaque champ d'une même ligne dudit nouveau fichier plat FFw-2 résulte d'une opération de chiffrement du champ correspondant de la même ligne dudit fichier plat FFw-1 par l'une desdites clés de chiffrement/déchiffrement $\overrightarrow{CK}$. Ainsi, le nouveau fichier plat FFw-2 est en quelque sorte une copie chiffrée, ligne par ligne - c'est-à-dire archive élémentaire par archive élémentaire, de l'archive unitaire sous la forme du fichier plat FFw-1, les attributs de chaque ligne ayant tous été chiffrés par l'une desdites clés de chiffrement/déchiffrement $\overrightarrow{CK}$. L'invention prévoit que lesdites lignes ou archives élémentaires soient respectivement chiffrées par des clés $\overrightarrow{CK}$ différentes.

[0074] L'invention prévoit de nombreuses alternatives : chiffrer l'ensemble d'une archive élémentaire à l'aide d'une clé $\overrightarrow{CK}$, ou chiffer les champs d'une telle archive élémentaires respectivement par des clés différentes ou dérivées, voire toute autre combinaison. Il est aussi possible de réutiliser les clés de chiffrement/déchiffrement $\overrightarrow{CK}$ identiques pour des archives unitaires différentes. Rappelons que la logique de l'invention est de créer une nouvelle archive unitaire chiffrée chaînée à une archive unitaire initiale et qui viendra en remplacement de cette dernière. Cette nouvelle archive unitaire chiffrée pourra, sans jamais être modifiée, perdre l'information concernant une archive élémentaire, sans modification, par la perte de la connaissance de la clé de chiffrement/déchiffrement $\overrightarrow{CK}$ associée permettant le déchiffrement de l'archive élémentaire.

[0075] Pour chaque ligne ainsi chiffrée, ladite étape 203 consiste en outre à mettre à jour un attribut CK, dans l'enregistrement CTwR1, ... CTwRk de la table associative CTw qui désigne la ligne qui a été chiffrée du fichier FFw-1, de sorte que ledit attribut CK

désigne la clé de chiffrement/déchiffrement $\overrightarrow{CK}$ utilisée pour chiffrer les champs de ladite ligne dudit fichier plat FFw-1 pour créer la ligne chiffrée du nouveau fichier plat FFw-2. Un tel attribut CK peut comporter la clé de chiffrement/déchiffrement $\overrightarrow{CK}$ en tant que telle ou toute valeur, éventuellement plurielle, nécessaire à la désignation ou la génération de ladite clé de chiffrement/déchiffrement $\overrightarrow{CK}$. L'invention prévoit, à ce stade, différentes variantes pour lier l'enregistrement correspondant à la ligne du fichier plat FFw-1 à la ligne chiffrée du nouveau fichier plat FFw-2 ainsi constitué. Selon une première variante, les fichiers plats FFw-1 et FFw-2 partagent un radical de nommage commun, seule une extension permettant de distinguer la version claire de la version chiffrée. Alternativement, l'enregistrement de la table associative CTw peut comporter un attribut supplémentaire pour désigner le nouveau fichier plat FFw-2 ainsi nouvellement créé. Dans ce cas, les deux fichiers peuvent ne pas suivre une règle de nommage imposée.

[0076] Quelle que soit la technique choisie, à l'issue de la mise en œuvre de l'étape 203, ladite table associative CTw lie, directement ou indirectement :

- une clé de chiffrement/déchiffrement choisie parmi l'ensemble $\overrightarrow{CK}$ qui a été utilisée pour chiffrer une archive élémentaire de l'archive unitaire claire FFw-1 (en l'espèce une ligne d'un fichier plat selon l'exemple illustré par les figures 3 et 4),
- ladite archive élémentaire de ladite archive unitaire FFw-1,
- l'archive élémentaire au sein de la version chiffrée FFw-2 de ladite archive unitaire FFw-1 et,
- l'enregistrement du conteneur de données Tw (en l'espèce un enregistrement d'une table courante) selon l'exemple illustré par les figures 3 et 4) initialisé à partir de ladite archive élémentaire de l'archive unitaire claire FFw-1 lors du traitement d'importation 100.

[0077] Comme l'indique la figure 4, l'enregistrement de la table associative ainsi mise à jour pour désigner la clé de chiffrement/déchiffrement $\overrightarrow{CK}$ exploitée peut en outre comporter un attribut complémentaire ED pour enregistrer un horodatage dudit chiffrement et un attribut complémentaire EID pour stocker un identifiant de la personne en charge du déclenchement dudit processus de création de l'archive unitaire FFw-2 sous la forme d'un fichier plat.

[0078] Un traitement 200 comporte en outre une étape 206 de suppression de l'archive unitaire claire FFw-1 dans la première mémoire de données DMw. Seule prospère la version chiffrée FFw-2 de celle-ci à l'issue de la mise en œuvre dudit traitement 200. La version chiffrée FFw-2 n'a de sens que si la suppression de l'archive unitaire claire est réalisée.

[0079] Une mise en œuvre d'un tel traitement 200 est illustrée par la [Fig.6] au travers de l'exemple simplifié des données dont l'importation 100 a été précédemment décrite en lien avec les figures 3A et 4A.

[0080] Ladite [Fig.6] présente ainsi une table Tw1 identique à celle décrite en lien avec la [Fig.3A]. En effet, le traitement 200 ne modifie pas ladite table courante Tw1. En revanche, ladite [Fig.6] illustre, à l'issue de la mise en œuvre du traitement 200, la suppression du fichier FFw1-1 dans la mémoire de donnée DMw. Cette suppression est symbolisée par la barre oblique barrant la représentation graphique dudit fichier plat FFw1-1. Ladite [Fig.6] décrit également la création du fichier plat FFw1-2 qui correspond à la copie chiffrée dudit fichier plat FFw1-1. On retrouve ainsi dans les lignes FFw1-2R1 et FFw1-2R2 dudit fichier FFw1-2, les quatre champs chiffrés décrivant des fournisseurs. Nous pouvons également constater que la clé de chiffrement/déchiffrement CK1 a été exploitée pour chiffrer la ligne FFw1-2R1 et que la clé CK2 a été exploitée pour chiffrer la ligne FFw1-2R2.

[0081] Cette exploitation desdites clés CK1 et CK2 se traduit par la mise à jour de la table associative CTw1. Ainsi, l'enregistrement CTw1R1 associé à la première ligne des fichiers FFw-1 et FFw-2 est modifié de sorte que l'attribut CK désigne la clé de chiffrement/déchiffrement exploitée, en l'espèce la clé CK1. A titre optionnel, ledit l'enregistrement CTw1R1 est également mis à jour pour horodater le chiffrement du fichier plat FFw1-1 et mémoriser l'identifiant de l'opérateur ayant déclenché ledit chiffrement. Ainsi, l'attribut ED comporte l'horodatage ts3 et l'attribut EID comporte l'identifiant U020.

[0082] L'enregistrement CTw1R2, associé quant à lui à la deuxième ligne des fichiers FFw1-1 et FFw1-2, est modifié de sorte que l'attribut CK désigne la clé de chiffrement/déchiffrement exploitée, en l'espèce la clé CK2. A titre optionnel, ledit enregistrement CTw1R2 est également mis à jour pour horodater le chiffrement de la deuxième ligne du fichier plat FFw1-1 et mémoriser l'identifiant de l'opérateur ayant déclenché ledit chiffrement. Ainsi, l'attribut ED comporte l'horodatage ts4 et l'attribut EID comporte l'identifiant U020.

[0083] La [Fig.5] illustre en outre un exemple de traitement 200 préalable à toute modification ou suppression d'une archive élémentaire au sein d'un conteneur de données Tw (par exemple une table courante), comportant une étape optionnelle de vérification de la pertinence de l'archive unitaire chiffrée FFw-2 au regard de l'archive unitaire claire FFw-1, ces dernières pouvant se présenter sous une forme avantageuse de fichiers plats. En effet, l'archive unitaire claire FFw-1 ayant vocation à être supprimée à l'issue de la mise en œuvre de l'étape 206, il peut être avantageux de s'assurer que le processus de chiffrement 203 s'est correctement déroulé.

[0084] A titre d'exemple de réalisation préférée, une telle vérification peut consister en une

première étape itérative de déchiffrement 204 des champs de chaque ligne du fichier plat chiffré FFw-2 (créé à l'étape 203) à l'aide de la clé de chiffrement/déchiffrement désignée dans l'enregistrement qui est associé à ladite ligne dans la table associative CTw et de création d'une ligne de champs clairs dans un nouveau fichier plat FFw-2'. A l'issue de la création d'une version déchiffrée FFw-2' dudit fichier plat chiffré FFw-2, une étape 205 d'évaluation de la conformité dudit nouveau fichier plat déchiffré FFw-2' au regard du fichier plat clair initial FFw-1 est mise en œuvre. L'invention prévoit alors qu'avantageusement, l'étape 206 n'est mise en œuvre que si le test de conformité 205 est positif (situation symbolisée par le lien 205-y sur la [Fig.5]). L'étape 206 peut inclure la constitution d'une « trace » pour prouver que le test 205 a été réalisé avec succès. Cette trace peut se traduire par une conservation d'un code de redondance ou un haché du fichier déchiffré FFw-2' dans une table de l'application mettant en œuvre un procédé de gestion de données 10 selon l'invention. En revanche, si ledit test 205 échoue (situation symbolisée par le lien 205-n sur la [Fig.5]), le fichier plat clair FFw-2' et le fichier plat chiffré FFw-2 sont supprimés de la mémoire de données DMw. L'étape de création 203 d'une archive unitaire chiffrée FFw-2, sous la forme avantageuse d'un fichier plat, est alors de nouveau mise en œuvre et soumise à l'approbation d'une nouvelle vérification de conformité 204-205. Nous pouvons noter que l'archive unitaire claire FFw-2' est supprimée dans la mémoire de données DMw, quel que soit le « verdict » 205-y ou 205-n du test de conformité 205 précédemment évoqué.

[0085] A titre d'exemple de réalisation avantageux, l'étape 205 d'évaluation de la conformité d'un nouveau fichier plat déchiffré FFw-2' au regard d'un fichier plat clair initial FFw-1 peut consister en la comparaison de codes de redondances H1, H2' respectivement et préalablement calculés à partir desdits fichiers plats FFw-1 et FFw-2'. De tels codes de redondances peuvent par exemple résulter d'une mise en œuvre d'une fonction cryptographique de hachage. Un haché est le résultat d'un calcul appliqué à l'archive unitaire, d'une taille très inférieure à celle de ladite archive unitaire, et dont le calcul a la particularité qu'un changement mineur dans l'archive unitaire (par exemple la modification d'un caractère), entraîne un changement important de la valeur dudit haché. En conséquence seuls deux hachés identiques témoignent que leurs calculs respectifs ont porté sur deux fichiers identiques.

[0086] La [Fig.5] présente en outre un exemple avantageux pour lequel la création 203 d'une archive unitaire chiffrée FFw-2 et la suppression 206 de l'archivage unitaire initial FFw-1 ne sont mises en œuvre que si (situation illustrée par le lien 202-y dans la [Fig.5]) les données contenues dans lesdites archives unitaires comportent des données concernées par des obligations contradictoires de conservation et de mise à jour, comme ce peut être le cas de données personnelles au regard du RGPD par exemple.

Selon cet exemple avantageux, un traitement 200 peut comporter avantageusement un test 202 visant à conditionner les mises en œuvre desdites étapes 203 à 206 (situations illustrées dans la [Fig.5] respectivement par le lien 202-y, dans l'affirmative, et par le lien 202-n, dans la négative). Un tel test 202 peut consister en la lecture d'une méta-information contenue dans l'archive unitaire FFw-1 (héritée éventuellement de l'archive unitaire d'entrée FFi ou créée par la mise en œuvre d'un traitement d'importation 100 idoine) et qui caractérise de telles obligations. Un tel test 202 pourrait également mettre en œuvre toute autre analyse portant sur le type des données contenues dans une telle archive unitaire FFw-1. De façon alternative ou complémentaire, c'est le processus 200 tout entier qui pourrait être mis en œuvre préalablement à un traitement 300 sur le seul périmètre concerné par le traitement 200 considéré. Dans les deux cas, le traitement 200 pourra s'appliquer à toutes les données d'un conteneur de données (une ou plusieurs archives unitaire) ou à une partie uniquement de ces archives unitaires.

[0087] L'étape 202 peut également constituer un processus de retraitement des archives unitaires en deux catégories d'archives unitaires : d'une part des archives unitaire pour lesquelles aucun enregistrement n'est soumis à des contraintes de destruction/modification, et d'autre part des archives unitaires pour lesquelles chaque enregistrement est soumis à de telles contraintes.

[0088] L'invention a été décrite selon un exemple de réalisation préféré, selon lequel un conteneur associatif CTw, sous la forme avantageuse d'une table associative, est créé dans la première mémoire de données DMw, avec un conteneur de données Tw, sous la forme avantageuse d'une table courante, en une même étape 105 d'un traitement d'importation 100. En variante, l'invention prévoit que la création d'un tel conteneur associatif CTw puisse être créé de manière dissociée de la création d'un conteneur de données Tw. Il suffit qu'une telle étape de création d'un conteneur associatif CTw soit mise en œuvre préalablement à la mise en œuvre de l'étape 203 décrite précédemment. Dans ce cas, une telle étape de création d'un conteneur associatif CTw peut faire partie d'un traitement 200 préalable à toute mise en œuvre subséquente de l'un des enregistrements d'une table courante Tw.

[0089] Comme l'indique la [Fig.1], un procédé de gestion de données 10 conforme à l'invention peut comporter en outre un traitement 300 de mise à jour des données gérées. La [Fig.7] illustre un tel traitement 300 en réponse à une consigne (référencée USP en [Fig.7]) de modification du contenu d'un conteneur de données Tw émanant d'une consigne d'un utilisateur ou d'une règle de gestion, appliquée via une interface ou un algorithme. La consigne USP, réceptionnée ou détectée en une étape 310, s'applique à l'archive élémentaire et à son équivalent dans le conteneur de données Tw mais elle ne peut, rappelons-le, s'appliquer à l'archive unitaire en tant que telle. Selon

que ladite consigne USP concerne une suppression d'une archive élémentaire ou d'une mise à jour, le traitement 300 se subdivise en deux principaux sous-traitements référencés 300-D et 300-U sur la [Fig.7].

[0090] Lorsque que ledit traitement de mise à jour 300 consiste en la traduction 310 d'une consigne de suppression (situation illustrée par le lien 310-D sur la figure 7) d'un enregistrement d'un conteneur de données Tw (par exemple, sous la forme avantageuse d'une table courante) décrivant une archive élémentaire, ledit traitement 300 consiste en le sous-traitement 300-D. Celui-ci comporte une étape 321 de suppression dudit enregistrement objet de ladite consigne USP en suppression dans ledit conteneur de données Tw, et d'effacement de l'attribut CK désignant la clé de chiffrement/déchiffrement dans l'enregistrement du conteneur associatif CTw (par exemple, sous la forme avantageuse d'une table associative) correspondant à l'enregistrement supprimé dans le conteneur de données Tw. Ainsi, supprimer des archives élémentaires d'une même archive unitaire en perdant la connaissance des clés de chiffrement/déchiffrement $\overrightarrow{CK}$ respectivement associées auxdites archives élémentaires, permet de ne pas modifier l'archive unitaire et répond ainsi aux besoins contradictoires de conservation et de mise à jour de telles archives. Lorsque les données concernées par une telle requête USP en suppression portent sur des données de désignation d'un document de travail Dw, ladite étape 321 peut en outre détruire ou effacer dans la mémoire de travail ledit document Dw désigné par lesdites données de désignation.

[0091] Une mise en œuvre d'un tel sous-traitement 300-D est illustrée par la [Fig.8] au travers de l'exemple simplifié des données dont l'importation 100 a été précédemment décrite en lien avec les figures 3A et 4A et dont le traitement préalable 200 a été décrit en lien avec la [Fig.6].

[0092] Ladite [Fig.8] présente un conteneur de données Tw1 identique à la table Tw1 illustrée par la [Fig.6], exception faite de l'enregistrement Tw1R1 qui a fait l'objet d'une consigne USP en suppression. Les attributs PID, PC, PN, PA et UK de celle-ci apparaissent vierges sur la [Fig.8]. Nous pouvons constater que le fichier plat (archive unitaire) FFw1-2 demeure inchangé à l'issue du traitement 300, 300-D. Il conserve donc un statut d'archive fiscale et peut servir de base à une présentation de données à des organismes de contrôles coercitifs. Ladite [Fig.8] présente en outre un conteneur associatif CTw1 similaire à la table associative CTw1 d'ores et déjà décrite en lien avec la [Fig.6]. Toutefois, à l'issue de la mise en œuvre du sous-traitement 300-D, l'enregistrement CTw1R1 dudit conteneur associatif CTw1 (c'est-à-dire l'enregistrement qui est associé à l'enregistrement Tw1R1 supprimé puisque les deux enregistrements CTw1R1 et Tw1R1 - avant la suppression de l'enregistrement Tw1R1 - comportent des attributs CK de valeurs « 12345678 » identiques) a été modifié. Cette modification porte sur l'attribut CTw1R1A2 (qui apparaît ceinturée par une ligne dis-

continue sur la [Fig.8]) qui désignait jusqu'alors la clé de chiffrement/déchiffrement CK1 (cf. [Fig.6]). Ledit attribut CTw1R1A2 est à présent effacé. Ainsi, bien que le fichier FFw1-2 soit conservé en l'état, il n'est plus possible d'accéder – via la table associative CTw1 - à l'information claire en lien avec la ligne FFw1-2R1 de ce dernier puisque l'enregistrement CTw1R1 ne désigne plus la clé de chiffrement/déchiffrement nécessaire au déchiffrement de ladite ligne FFw1-2R1.

[0093] La [Fig.7] décrit en outre un sous-traitement 300-U (du traitement 300) déclenché à la suite de la réception d'une consigne USP en modification de tout ou partie d'un enregistrement d'un conteneur de données Tw traduisant une archive élémentaire. Une telle consigne USP peut consister par exemple en la modification de la valeur contenue par un attribut ou en l'effacement d'un tel attribut.

[0094] Un tel sous-traitement 300-U comporte une étape 331 de mise à jour de l'enregistrement dans le conteneur de données Tw (par exemple sous la forme avantageuse d'une table courante) qui est l'objet de ladite consigne en modification USP, pour modifier ou effacer la valeur contenue dans un ou plusieurs attributs de celui-ci.

[0095] Ledit sous-traitement 300-U comporte alors une étape de création 332 d'une nouvelle clé de chiffrement/déchiffrement, complétant ainsi l'ensemble $\widetilde{CK}$ précédemment constitué notamment à l'issue du traitement 200. Ladite étape 332 consiste en outre en la création d'une nouvelle archive unitaire, en l'espèce un nouveau fichier plat, FFw-3 dont une ligne (archive élémentaire) correspond à l'enregistrement de la table courante Tw dont au moins un attribut a été modifié à l'étape 331. Chaque champ de la ligne créée au sein du nouveau fichier plat FFw-3 correspond à chaque attribut (exception faite de l'attribut UK) dudit enregistrement modifié, la valeur de chacun desdits attributs étant chiffrée par la nouvelle clé de chiffrement/déchiffrement créée. Si une archive unitaire FFw-3 existe déjà, et peut être complétée, alors l'étape 332 consiste en l'ajout d'une archive élémentaire à l'archive unitaire existante (par exemple jusqu'à l'obtention d'un nombre de lignes et ou d'un volume prédéfini). Il peut sembler surprenant de s'autoriser ainsi à modifier une archive unitaire alors que l'invention vise à ne pas modifier les archives unitaires. En fait, une telle archive ayant été constituée par l'application mettant en œuvre un procédé de gestion de données 10 selon l'invention, les requêtes formulées par des contrôles coercitifs ne nécessitent pas l'utilisation d'une archive fiscale. Cette archive fiscale ne serait requise que si l'application mettant en œuvre un procédé de gestion de données 10 selon l'invention venait à être décommissionnée. En conséquence, la création de FFw-3 n'est réellement requis qu'au décommissionnement de l'application mettant en œuvre un procédé de gestion de données 10 selon l'invention. Le stockage des informations sous la forme d'une « archive fiscale » FFw-3 simplifie le développement de l'application mettant en

œuvre un procédé de gestion de données 10 selon l'invention dans la mesure où la donnée ainsi créée peut également faire l'objet d'un traitement 300 ultérieur.

[0096] Un tel sous-traitement 300-U comporte également une étape 333 de mise à jour du conteneur associatif (par exemple, sous la forme avantageuse d'une table associative) CTw associé au conteneur de données Tw (par exemple, sous la forme avantageuse d'une table courante). Cette mise à jour de ladite table associative CTw consiste plus précisément en l'effacement de l'attribut CK, désignant la clé de chiffrement/déchiffrement précédemment exploitée pour créer l'archive unitaire (par exemple, sous la forme avantageuse d'un fichier plat) FFw-2, dans l'enregistrement de ladite table associative CTw qui correspond ou qui est associé à l'enregistrement mis à jour dans le conteneur de données Tw en réponse à la consigne USP.

[0097] Ladite étape 333 consiste en outre à créer, dans ledit conteneur associatif CTw, un nouvel enregistrement associé audit enregistrement mis à jour dans le conteneur de données Tw. Ce nouvel enregistrement dudit conteneur associatif CTw comporte principalement :

- un attribut UK pour contenir la clé unique désignant ledit enregistrement du conteneur de données Tw qui a été mis à jour ;
- des attributs pour respectivement désigner la nouvelle archive unitaire FFw-3 et l'archive élémentaire de cette dernière qui a été initialisée à l'étape 332 ;
- un attribut désignant la clé de chiffrement/déchiffrement utilisée pour chiffrer les champs de l'archive élémentaire dans l'archive unitaire FFw-3 (c'est-à-dire les champs d'une ligne d'un fichier plat lorsque l'archive unitaire FFw-3 consiste en un tel fichier plat). Ce chiffrement peut ne pas être obligatoire. Il est avantageusement mis en œuvre, afin que le processus 300 s'applique également à la modification de données contenues dans l'archive unitaire FFw-3, des solutions alternatives d'implémentation pourraient être envisagées sans la nécessité du chiffrement des archives élémentaires dans l'archive unitaire FFw-3.

[0098] Lorsque, à l'instar de l'exemple décrit par la [Fig.4], le conteneur associatif CTw (par exemple, sous la forme avantageuse d'une table associative) comporte des attributs accessoires CD, ED pour inscrire les horodatages respectifs de la création de l'entrée dans ladite table CTw et de la création de l'archive élémentaire chiffrée dans l'archive unitaire FFw-3, voire encore des attributs accessoires CID et EID pour inscrire les identifiants des personnes ayant été à l'origine desdites créations, l'étape 333 de création, dans ladite table associative CTw, d'un nouvel enregistrement associé audit enregistrement mis à jour dans le conteneur de données Tw, consiste en outre, par cohérence, à initialiser lesdits attributs accessoires CD, CID, ED, EID en lien avec la création et le chiffrement de l'archive élémentaire de l'archive unitaire FFw-3. Ainsi,

de tels attributs accessoires CD et ED, ainsi que CID et EID, sont identiques deux à deux et désignent respectivement l'instant de création de l'archive élémentaire dans ladite archive unitaire FFw-3 (ou de la création de l'enregistrement dans la table associative CTw) et la personne à l'origine de la consigne USP de mise à jour de l'enregistrement dans le conteneur de données Tw.

[0099] Le conteneur associatif CTw ainsi modifié à l'étape 333, lie directement ou indirectement, la nouvelle clé de chiffrement/déchiffrement qui a été utilisée pour créer ladite archive élémentaire au sein de l'archive unitaire FFw-3, l'archive élémentaire au sein de cette dernière et l'enregistrement du conteneur de données Tw mis à jour. Ainsi, une modification d'une archive élémentaire en la supprimant, puis en stockant l'information modifiée, en vue de constituer dans l'instant ou ultérieurement, une nouvelle archive élémentaire sans altérer l'archive unitaire originelle, répond également aux besoins contradictoires de conservation et mise à jour.

[0100] Une mise en œuvre d'un tel sous-traitement 300-U est illustrée par la [Fig.9] au travers de l'exemple simplifié des données dont l'importation 100 a été précédemment décrite en lien avec les figures 3A et 4A et dont le traitement préalable 200 a été décrit en lien avec la [Fig.6]. L'agencement dudit sous-traitement 310-U en trois étapes séquentielles 331, 332, et 333 est arbitraire dans la mesure où les étapes peuvent être réalisées en parallèle ou dans un ordre différent.

[0101] Ladite [Fig.9] présente un conteneur de données sous la forme d'une table Tw1 identique à la table courante Tw1 illustrée par la [Fig.6], exception faite de l'attribut Tw1R1A4 de l'enregistrement Tw1R1 qui a fait l'objet d'une consigne USP en modification. En effet, selon cet exemple, le siège social d'un fournisseur français a déménagé depuis le « 14, rue de l'océan » vers la « Place centrale ». Cet exemple peut surprendre car il ne consiste pas à anonymiser une information, mais il peut se comprendre comme une pseudo anonymisation, permettant de rendre un jeu de données utilisable (par exemple, à des fins de recherche ou de marketing) sans pour autant contenir d'information personnelle utilisable. Nous pouvons constater que l'archive unitaire, sous la forme avantageuse du fichier plat FFw1-2, demeure inchangée à l'issue du traitement 300, 300-U. Ladite [Fig.9] présente toutefois la création d'une nouvelle archive unitaire sous la forme d'un nouveau fichier plat FFw1-3 qui comporte une archive élémentaire, en l'espèce une ligne FFw1-3R1, comprenant, à l'instar d'une archive élémentaire sous la forme d'une ligne du fichier plat FFw1-2, quatre champs qui correspondent respectivement aux attributs PID, PC, PN et PA de l'enregistrement Tw1R1 modifié, lesdits attributs étant chiffrés au moyen d'une nouvelle clé de chiffrement/déchiffrement CK3 pour constituer les champs de ladite archive élémentaire FFw1-3R1.

[0102] La [Fig.9] illustre en outre un conteneur associatif sous la forme avantageuse d'une

table associative CTw1 similaire à celle d'ores et déjà décrite en lien avec la [Fig.6]. Toutefois, à l'issue de la mise en œuvre du sous-traitement 300-U, l'enregistrement CTw1R1 de la table associative CTw1, associé à l'enregistrement Tw1R1 qui est l'objet de la consigne USP en modification, a été modifié. Cette modification consiste à effacer la valeur de l'attribut CTw1R1A2 (apparaissant ceinturé par une ligne discontinue sur la [Fig.9]). Cet attribut désignait jusqu'alors la clé de chiffrement/déchiffrement CK1. Ainsi, bien que l'archive unitaire (sous la forme avantageuse d'un fichier plat) FFw1-2 soit conservée en l'état, il n'est plus possible d'accéder – via la table associative CTw1 - à l'information claire en lien avec l'archive élémentaire FFw1-2R1 de ladite archive unitaire puisque l'enregistrement CTw1R1 ne désigne plus la clé de chiffrement/déchiffrement nécessaire au déchiffrement de ladite archive élémentaire FFw1-2R1. En revanche, ladite table associative CTw1 comporte un enregistrement CTw1R3 supplémentaire au regard de celle décrite en lien avec la [Fig.6].

[0103] Le nouvel enregistrement CTw1R3 de ladite table associative CTw1 comporte :

- un attribut UK, référencé CTw1R3A1 sur la [Fig.9], pour contenir la clé unique « 12345678 » désignant l'enregistrement Tw1R1 de la table courante Tw1, ou plus généralement du conteneur de données, qui a été mis à jour traduisant une archive élémentaire amendée ; le fait de conserver la même clé unique UK assure une meilleure traçabilité des modifications, mais, en variante le champ UK pourrait être recalculé ;
- des attributs CTw1R3A3 et CTw1R3A4 pour respectivement désigner la nouvelle archive unitaire FFw1-3 et la ligne ou archive élémentaire FFw1-3R1 de cette dernière ;
- un attribut CTw1R3A2 désignant la clé de chiffrement/déchiffrement CK3 utilisée pour chiffrer les champs de ladite archive élémentaire FFw1-3R1 de ladite archive unitaire FFw1-3.

[0104] Comme évoqué précédemment, un tel nouvel enregistrement CTw1R3 de la table associative CTw1, associé à l'enregistrement Tw1R1 mis à jour dans la table courante Tw1, peut comporter des attributs accessoires CD, CID, ED, EID en lien avec les archives élémentaires de ladite archive unitaire FFw-3, comme l'indique la [Fig.9]. En l'espèce, les attributs CD et ED désignent l'horodatage ts5 de la création dudit enregistrement CTw1R3 et les attributs ED et EID désignent (par l'identifiant U030) la personne à l'origine de la consigne de mise à jour de l'enregistrement Tw1R1 dans la table courante Tw1.

[0105] A l'issue de la mise en œuvre du sous-traitement 300-U, la table associative CTw1 assure le lien entre les enregistrements Tw1R1 et Tw1R2 de la table courante Tw1 et les fichiers de travail FFw1-2 et FFw1-3. Plus précisément, la première ligne FFw1-3R1 du fichier plat FFw1-3, dont les champs sont déchiffrables par

l'exploitation de la clé de chiffrement/déchiffrement CK3, est associée à l'enregistrement Tw1R1 de la table courante Tw1. La deuxième ligne FFw1-2R2 du fichier plat FFw1-2, dont les champs sont déchiffrables par l'exploitation de la clé de chiffrement/déchiffrement CK2, est associée à l'enregistrement Tw1R2 de la table courante Tw1. L'enregistrement CTw1R1 de la table associative CTw1 n'est plus utile qu'à des fins de traçabilité et pourrait être archivé.

[0106] La situation aurait été identique, si la consigne USP en modification d'un enregistrement d'un conteneur de données Tw1 avait porté sur la suppression du nom d'une personne référente, tel que désigné par l'attribut PN pour satisfaire aux règles du RGPD, par exemple.

[0107] Comme l'indique la [Fig.1], l'invention prévoit qu'un procédé de gestion de données 10 puisse comporter un traitement 400 visant à terminer l'exploitation dudit procédé 10 et exporter les données gérées vers une application tierce. Un exemple d'un tel traitement 400 est illustré par la [Fig.10].

[0108] L'objectif de ce traitement 400 peut consister à exploiter un conteneur associatif CTw associé à un conteneur de données Tw, pour constituer une archive unitaire (par exemple sous la forme d'un fichier plat) de sortie FFo à partir des archives unitaires de travail chiffrées FFw.

[0109] Un tel traitement 400 peut comporter une première étape itérative 401 de lecture de tout enregistrement d'un tel conteneur associatif CTw (par exemple, sous la forme avantageuse d'une table associative) comportant un attribut UK désignant une clé de chiffrement/déchiffrement CK. Lorsqu'une archive unitaire de travail FFw se présente sous la forme avantageuse d'un fichier plat de travail, pour chaque enregistrement ainsi sélectionné, une étape 402 consiste à lire la ligne du fichier plat de travail FFw désignée par les attributs FN et RN (voir figure 4) de l'enregistrement sélectionné et à déchiffrer les champs de ladite ligne lue en exploitant la clé de chiffrement/déchiffrement désignée (attribut CK) également par l'enregistrement sélectionné parmi les clés de chiffrement/déchiffrement $\overrightarrow{CK}$. Une telle étape 402 consiste alors à créer, pour chaque enregistrement sélectionné au sein de la table associative CTw, une ligne dans un nouveau fichier plat FFo dont les champs clairs correspondent respectivement auxdits champs déchiffrés. Un tel traitement 400 peut en outre comporter une étape 404 d'enregistrement dudit fichier plat de sortie FFo dans une mémoire de données DMo du système informatique, tel que le système informatique 1 selon la [Fig.1]. Un tel fichier plat FFo constitue une archive unitaire de sortie dont chaque ligne décrit une archive élémentaire. Une telle étape 404 peut en outre consister en l'effacement des fichiers ou archives unitaires de travail chiffrés FFw dans la mémoire de données DMw. Au préalable, l'invention prévoit, selon un mode de réalisation optionnel, qu'un tel traitement 400 comporte une étape de vérification 403 de la pertinence dudit fichier

plat de sortie FFo. Dans ce cas, l'étape 404 d'enregistrement dans une mémoire de données DMo, voire de suppression des fichiers de travail chiffrés FFw dans la mémoire de données DMw, n'est mise en œuvre que si ladite étape de vérification 403 atteste que ledit fichier plat FFo est a priori pertinent. Cette situation est symbolisée par le lien 403-y en [Fig.10]. Dans le cas contraire (situation symbolisée par le lien 403-n en [Fig.10]), ledit fichier plat FFo est supprimé et les étapes 401 et 402 sont de nouveau mises en œuvre pour créer une nouvelle archive unitaire de sortie, en l'espèce, selon cet exemple, un nouveau fichier plat de sortie FFo. Une telle étape de vérification 403 peut consister en diverses opérations portant sur le fichier plat FFo, les fichiers de travail FFw et/ou les tables courante Tw et associative CTw. A titre d'exemple, ladite vérification 403 peut consister à compter le nombre d'enregistrements dans la table associative CTw exploités pour déchiffrer les champs d'une ligne d'un fichier de travail FFw et constituer une ligne de champs clairs dans ledit fichier plat de sortie FFo puis à comparer ledit nombre d'enregistrements avec le nombre de lignes créées dans ledit fichier plat de sortie FFo. Il ne faut pas tenir compte des enregistrements de CTw correspondant à des enregistrements supprimés au sens de l'étape 300-D. Si ces deux nombres sont égaux alors, le fichier plat FFo est considéré (cf. le lien 403-y en [Fig.10]) comme pertinent par l'étape 403. Dans le cas contraire (cf. le lien 403-n en [Fig.10]), la vérification échoue. En variante et/ou en complément, ledit nombre de lignes créées dans ledit fichier plat de sortie FFo peut être comparé avec le nombre d'enregistrements non nuls présents dans la table courante (conteneur de données) Tw associée à la table associative CTw exploitée pour créer ledit fichier plat de sortie FFo. Si les deux nombres sont égaux, alors l'archive unitaire de sortie FFo peut être déclarée pertinente à l'étape 403.

[0110] La [Fig.11] illustre le résultat d'une mise en œuvre d'un tel traitement 400 sur l'exemple simplifié des données dont l'importation 100 a été précédemment décrite en lien avec les figures 3A et 4A, dont le traitement préalable 200 a été décrit en lien avec la [Fig.6] et dont la mise à jour 300-U d'un attribut d'un enregistrement a été illustrée par la [Fig.9].

[0111] En exploitant la table associative CTw1 décrite en lien avec ladite [Fig.9], seuls deux enregistrements CTw1R2 et CTw1R3 comportent un attribut CK désignant une clé de déchiffrement/déchiffrement, en l'espèce les clés CK2 et CK3. Le fichier plat de sortie FFo1 (archive unitaire de sortie) décrit en [Fig.11], résultant de la mise en œuvre d'un traitement d'exportation 400, comporte donc deux lignes FFo1R1 et FFo1R2 (archives élémentaires). La première ligne FFo1R1, issue de l'exploitation de l'enregistrement CTw1R2 de la table associative CTw1, correspond à la deuxième ligne du fichier FFw1-2 après déchiffrement des champs à partir de la clé de chiffrement/déchiffrement CK2. La seconde ligne FFo1R2, issue de l'exploitation de l'enregistrement CTw1R3

de la table associative CTw1, correspond, quant à elle, à la première ligne du fichier FFw1-3 après déchiffrement des champs grâce à la clé de chiffrement/déchiffrement CK3.

[0112] En variante, un tel traitement 400 d'exportation des données contenues dans une table courante Tw1 pourrait consister à parcourir les enregistrements de cette dernière et à créer une ligne dans ledit fichier plat de sortie FFo1 dont les champs sont ceux contenus dans les attributs desdits enregistrements de ladite table courante Tw1 à l'exception de l'attribut UK désignant une clé unique.

[0113] Plus généralement, un tel traitement 400 selon la [Fig.10] a ainsi pour vocation à remplacer les fichiers d'archives chiffrés par des fichiers d'archive en clair, également nommés « fichiers d'archive clairs ». Cette opération est un prérequis à un décommissionnement de l'application mettant en œuvre un procédé 10 puisqu'une archive unitaire n'est utilisable, en cas de contrôle coercitif, que si elle demeure lisible. Le traitement 400 illustré par la [Fig.10] ne présente pas l'intégralité des étapes nécessaires et suffisantes pour décommissionner une application mettant en œuvre un procédé de gestion de données 10 selon l'invention. En effet, les conteneurs de données gérés par une telle application feraient l'objet de création d'archives unitaires selon un processus auditable, les archives claires seraient reprises en l'état ainsi que tous documents associés à des index (données de désignation) lorsque ces documents n'auront pas été supprimés. Ainsi, un traitement 400 selon la [Fig.10] décrit principalement un processus de fourniture d'archives unitaires claires, postérieures aux dernières modifications appliquées par une ou plusieurs itérations d'un traitement 300 avant un arrêt ou décommissionnement de l'application mettant en œuvre un procédé 10 de gestion de données conforme à l'invention.

[0114] Les traitements 100, 200, 300 et 400 ont été décrits en lien avec les figures 2 à 11, au travers d'exemples portant sur des données dont l'importation ne constitue qu'un seul conteneur de données, en l'espèce une table courante, Tw et un seul conteneur associatif, en l'espèce une table associative CTw, par mesure de simplification. Lorsque lesdites données importées sont issues d'une pluralité d'archives unitaires d'entrée FFi, par exemple sous la forme d'une pluralité de fichiers plats d'entrée, une telle importation peut engendrer la création d'une pluralité de conteneurs de données ou de tables courantes Tw et donc de tables associatives CTw. Lesdits traitement 200, 300 et 400 portent alors itérativement sur lesdites pluralités de conteneurs de données Tw et conteneurs associatifs CTw. En d'autres termes, il peut y avoir plusieurs conteneurs de données (plusieurs tables courantes par exemple), et pour chaque conteneur de données, il peut y avoir plusieurs archives unitaires (sous la forme avantageuse respectivement de plusieurs fichiers plats FFi et FFw) selon des relations n-m et pas seulement par des relations 1-1 ou même 1-n. En variante, un seul conteneur associatif

CTw global pourrait être créé et exploité pour désigner tout enregistrement de tout conteneur de données Tw, Enfin, certains conteneurs de données Tw et associatifs CTw pourraient ne constituer qu'une seule et même entité dans la mémoire de données concernée DMw.

Revendications

[Revendication 1] Procédé (10) de gestion de données conçu pour être mis en œuvre par une unité de traitement (PU) d'un système informatique (1) comportant également une première mémoire de données (DMw) accessible en lecture et en écriture par ladite unité de traitement (PU), ledit procédé (10) étant caractérisé en ce que lesdites données sont stockées en clair dans une première archive unitaire (FFw-1) au sein de la première mémoire de données (DMw) dudit système informatique (1), ladite première archive unitaire (FFw-1) comportant une ou plusieurs archives élémentaires (FFw-1R1, FFw-1R2, FFw-1R3, …, FFw-1Rm) comprenant chacune un ou plusieurs champs (FFw-1RxCn) et en ce que ledit procédé (10) comporte :

- une étape (100, 105) de création, dans ladite première mémoire de données (DMw) d'un conteneur de données (Tw) dont les enregistrements (TwR1, TwR2, TwR3, …, TwRm) décrivent respectivement les archives élémentaires (FFw-1R1, FFw-1R2, FFw-1R3, …, FFw-1Rm) de ladite première archive unitaire (FFw-1) et pour lesquels un attribut (TwRxAn) d'enregistrement correspond à un champ (FFw-1RxCn) d'archive élémentaire, chaque enregistrement dudit conteneur de données (Tw) comportant en outre un attribut supplémentaire (TwRxAUK) dont la valeur (UK1, UK2, UK3, UKm) consiste en une clé unique identifiant l'information contenue dans un enregistrement dudit conteneur de données (Tw) ;
- une étape (105) de création, dans ladite première mémoire de données (DMw), d'un conteneur associatif (CTw) comportant un enregistrement (CTwR1, CTwR2, CTwR3, …, CTwRk) associé à chaque enregistrement (TwR1, TwR2, TwR3, …,TwRm) du conteneur de données (Tw), chaque enregistrement dudit conteneur associatif comportant un attribut (UK) pour contenir la clé unique désignant l'enregistrement du conteneur de données qui lui est ainsi associé, des attributs pour respectivement désigner la première archive unitaire (FFw-1) et l'archive élémentaire de cette dernière à partir desquels l'enregistrement du conteneur de données (Tw) a été initialisé ;

– une étape (203) :
  - de création, dans la première mémoire de données (DMw) :
    - d'autant de clés de chiffrement/déchiffrement ($\overrightarrow{CK}$) qu'il existe d'enregistrements dans le conteneur de données (Tw) ;
    - d'une deuxième archive unitaire (FFw-2, FFw1-2) à partir de la première archive unitaire (FFw-1, FFw1-1) et desdites clés de chiffrement/déchiffrement ($\overrightarrow{CK}$) de sorte que chaque champ (FFw1-2R1F1) d'une même archive élémentaire (FFw1-2R1) de ladite deuxième archive unitaire (FFw-2, FFw1-1) résulte d'une opération de chiffrement du champ (FFw1-2R1F1) correspondant de la même archive élémentaire (FFw1-1R1) de ladite première archive unitaire (FFw-1, FFw1-1) par l'une (CK1) desdites clés de chiffrement/déchiffrement ($\overrightarrow{CK}$) ;
  - de mise à jour d'un attribut (CK) dans l'enregistrement (CTwR1, … CTwRk) du conteneur associatif (CTw) associé à l'enregistrement (Tw1R1, …, TwRm) du conteneur de données (Tw) de sorte que ledit attribut (CK) désigne la clé (CK1) de chiffrement/déchiffrement utilisée pour chiffrer les champs de l'archive élémentaire de ladite première archive unitaire (FFw-1, FFw1-1) ;
– une étape (206) de suppression de la première archive unitaire (FFw-1, FFw1-1) dans la première mémoire de données (DMw).

[Revendication 2] Procédé (10) selon la revendication précédente, pour lequel :

– les données sont initialement stockées en clair dans une archive unitaire tierce, dite « archive unitaire d'entrée » (FFi) au sein d'une deuxième mémoire de données (DMi) dudit

système informatique (1), ladite archive unitaire tierce (FFi) comportant une ou plusieurs archives élémentaires comprenant chacune un ou plusieurs attributs ;

- le procédé (10) comporte une première étape (101) de lecture de l'archive unitaire d'entrée (FFi) dans la deuxième mémoire de données et de création (102) dans la première mémoire de données (DMw) de la première archive unitaire (FFw-1) à partir de ladite archive unitaire d'entrée (FFi) de sorte que lesdites première archive unitaire (FFw-1) et l'archive unitaire d'entrée (FFi) soient identiques.

[Revendication 3] Procédé (10) selon la revendication 1 ou 2, pour lequel, lorsque ledit procédé (10) comporte un traitement (300) de mise à jour consistant en une suppression (300-D) d'un enregistrement (Tw1R1) du conteneur de données (Tw) décrivant une archive élémentaire, ledit traitement (300, 300-D) comporte une étape (321) d'effacement de l'attribut (CK, CTw1R1A2) désignant la clé de chiffrement/déchiffrement dans l'enregistrement (CTw1R1) du conteneur associatif (CTw, CTw1) correspondant audit enregistrement (Tw1R1) du conteneur de données (Tw, Tw1).

[Revendication 4] Procédé (10) selon la revendication 1 ou 2, pour lequel, lorsque ledit procédé (10) comporte un traitement (300) de mise à jour consistant en une modification (300-U) d'un attribut (Tw1R1A4) d'un enregistrement (Tw1R1) du conteneur de données (Tw, Tw1) décrivant une archive élémentaire, ledit traitement (300, 300-U) comporte :

- une étape (331) de mise à jour dudit attribut (Tw1R1A4) dans le conteneur de données (Tw, Tw1) ;
- une étape de création (332) d'une clé de chiffrement/déchiffrement (CK3) et d'une troisième archive unitaire (FFw-3, FFw1-3) dont une archive élémentaire (FFw1-3R1) correspond à l'enregistrement (Tw1R1) du conteneur de données (Tw, Tw1) dont l'attribut (Tw1R1A4) a été modifié et pour laquelle chaque champ d'archive élémentaire correspond à chaque attribut dudit enregistrement (Tw1R1) chiffré par la clé de chiffrement/déchiffrement (CK3) préalablement créée ;
- une étape (333) :
  - d'effacement de l'attribut (CK, CTw1R1A2)

désignant la clé de chiffrement/déchiffrement dans l'enregistrement (CTw1R1) du conteneur associatif (CTw, CTw1) correspondant à l'enregistrement mis à jour du conteneur de données (Tw, Tw1) ;

- de création, dans ledit conteneur associatif, d'un nouvel enregistrement (CTw1R3) associé à l'enregistrement (Tw1R1) mis à jour dans le conteneur de données (Tw, Tw1), le nouvel enregistrement dudit conteneur associatif (CTw, CTw1) comportant :
  - un attribut (UK, CTw1R3A1) pour contenir la clé unique (UK) désignant ledit enregistrement (Tw1R1) du conteneur de données (Tw, Tw1) qui a été mis à jour,
  - des attributs (CTw1R3A3, CTw1R3A4) pour respectivement désigner la troisième archive unitaire (FFw-3, FFw1-3) et l'archive élémentaire (FFw1-3R1) de cette dernière qui vient d'être initialisée,
  - un attribut (CTw1R3A2) désignant la clé de chiffrement/déchiffrement (CK3) utilisée pour chiffrer les champs de l'archive élémentaire (FFw1-3R1) de ladite troisième archive unitaire (FFw-3, FFw1-3).

[Revendication 5] Procédé (10) selon l'une quelconque des revendications précédentes, comportant un traitement d'exportation des données (400) comportant :

- une étape (401) de lecture de tout enregistrement du conteneur associatif (CTw) comportant un attribut (UK) désignant une clé de chiffrement/déchiffrement (CK) ;
- une étape (402) de :
  - lecture de l'archive élémentaire de l'archive unitaire (FFw-2, FFw-3) conjointement désignées par ledit enregistrement lu au sein du conteneur associatif (CTw) ;
  - déchiffrement des champs de ladite archive élémentaire lue à partir de la clé de chiffrement/dé-

chiffrement désignée également par l'enregistrement lu dans le conteneur associatif (CTw) ;

- création d'une archive élémentaire (FFo1R1, FFo1R2) dans une quatrième archive unitaire, dite « archive unitaire de sortie » (FFo, FFo1) dont les champs correspondent respectivement auxdits champs déchiffrés de l'archive élémentaire (FFw-2, FFw-3) lue.

[Revendication 6] Procédé (10) selon la revendication précédente, comportant une étape (404) d'enregistrement dans une troisième mémoire de données (DMo) du système informatique (1) de ladite archive unitaire de sortie (FFo).

[Revendication 7] Procédé selon la revendication précédente comportant une étape de vérification (403) de la pertinence de ladite archive unitaire de sortie (FFo), l'étape d'enregistrement dans une troisième mémoire de données (DMo) n'étant mise en œuvre que si (403-y) ladite étape de vérification (403) atteste que ladite archive unitaire de sortie (FFo) est pertinente.

[Revendication 8] Procédé (10) selon la revendication 2, comportant une étape (103) de vérification de la pertinence de la première archive unitaire (FFw-1, FFw1-1) créée, préalable à la mise en œuvre de l'étape de création (105) du conteneur de données (Tw, Tw1) et/ou de l'étape (105) de création du conteneur associatif (CTw, CTw1), au moins l'une de ces deux étapes de création (105) n'étant mise en œuvre que si (103-y) ladite étape de vérification (103) atteste que ladite première archive unitaire (FFw-1, FFw1-1) est conforme à l'archive unitaire d'entrée (FFi), ladite première archive unitaire (FFw-1, FFw1-1) étant supprimée (104) de la première mémoire de données (DMw) et l'étape de création (102) dans ladite première mémoire de données (DMw) d'une première archive unitaire (FFw-1) à partir de l'archive unitaire d'entrée (FFi) étant de nouveau mise en œuvre dans le cas contraire (103-n).

[Revendication 9] Procédé (10) selon la revendication précédente, pour lequel l'étape (103) de vérification de la pertinence de la première archive unitaire (FFw-1, FFw1-1) consiste en la comparaison de codes de redondances (Hi, H1) respectivement et préalablement calculés à partir de l'archive unitaire d'entrée (FFi, FFi1) et de la première archive unitaire (FFw-1, FFw1-1).

[Revendication 10] Procédé (10) selon l'une quelconque des revendications précédentes,

comportant une étape de vérification (204, 205) de la pertinence de la deuxième archive unitaire (FFw-2, FFw1-2) créée au regard de la première archive unitaire (FFw-1, FFw1-1) préalablement à la mise en œuvre de l'étape (206) de suppression de la première archive unitaire (FFw-1, FFw1-1) dans ladite première mémoire de données (DMw), ladite étape de vérification (204, 205) consistant à :

- déchiffrer (204) les champs de chaque archive élémentaire de la deuxième archive unitaire (FFw-2, FFw1-2) à l'aide de la clé de chiffrement/déchiffrement désignée dans l'enregistrement qui est associé à ladite archive élémentaire dans le conteneur associatif (CTw, CTw1) et créer une archive élémentaire de champs clairs dans une cinquième archive unitaire (FFw-2') ;
- évaluer la conformité (205) de ladite cinquième archive unitaire (FFw-2') à ladite première archive unitaire (FFw-1, FFw1-1) ;

l'étape (206) de suppression de la première archive unitaire (FFw-1, FFw1-1) dans la première mémoire de données (DMw), n'étant mise en œuvre que si (205-y) ladite étape de vérification (205) de la pertinence de ladite deuxième archive unitaire (FFw-2, FFw1-2) atteste que ladite cinquième archive unitaire (FFw-2') est conforme à la première archive unitaire (FFw-1), ledit procédé (10) comportant, dans le cas contraire (205-n), une étape de suppression (207) dans la première mémoire de données (DMw), de ladite deuxième archive unitaire (FFw-2) et de ladite cinquième archive unitaire (FFw-2') provoquant une nouvelle mise en œuvre de l'étape de création (203) d'une deuxième archive unitaire (FFw-2) à partir de la première archive unitaire (FFw-1, FFw1-1) et de clés de chiffrement/déchiffrement ($\overrightarrow{CK}$).

[Revendication 11] Procédé (10) selon la revendication précédente, pour lequel l'étape (205) de vérification de la pertinence de la deuxième archive unitaire (FFw-2, FFw1-2) créée au regard de la première archive unitaire (FFw-1, FFw1-1) consiste en la comparaison de codes de redondances (H1, H2') respectivement et préalablement calculés à partir des première (FFw-1) et cinquième (FFw-2') archives unitaires.

[Revendication 12] Procédé (10) selon l'une quelconque des revendications précédentes, pour lequel l'étape (203) de création d'une deuxième archive unitaire

(FFw-2, FFw1-2) n'est mise en œuvre que si (202, 202-y) des données contenues dans la première archive unitaire (FFw-1) sont concernées par des obligations contradictoires de conservation et de mise à jour mise à jour.

[Revendication 13] Procédé selon l'une quelconque des revendications précédentes, pour lequel chaque archive unitaire consiste en un fichier plat, dont chaque archive élémentaire consiste en une ligne dudit fichier plat.

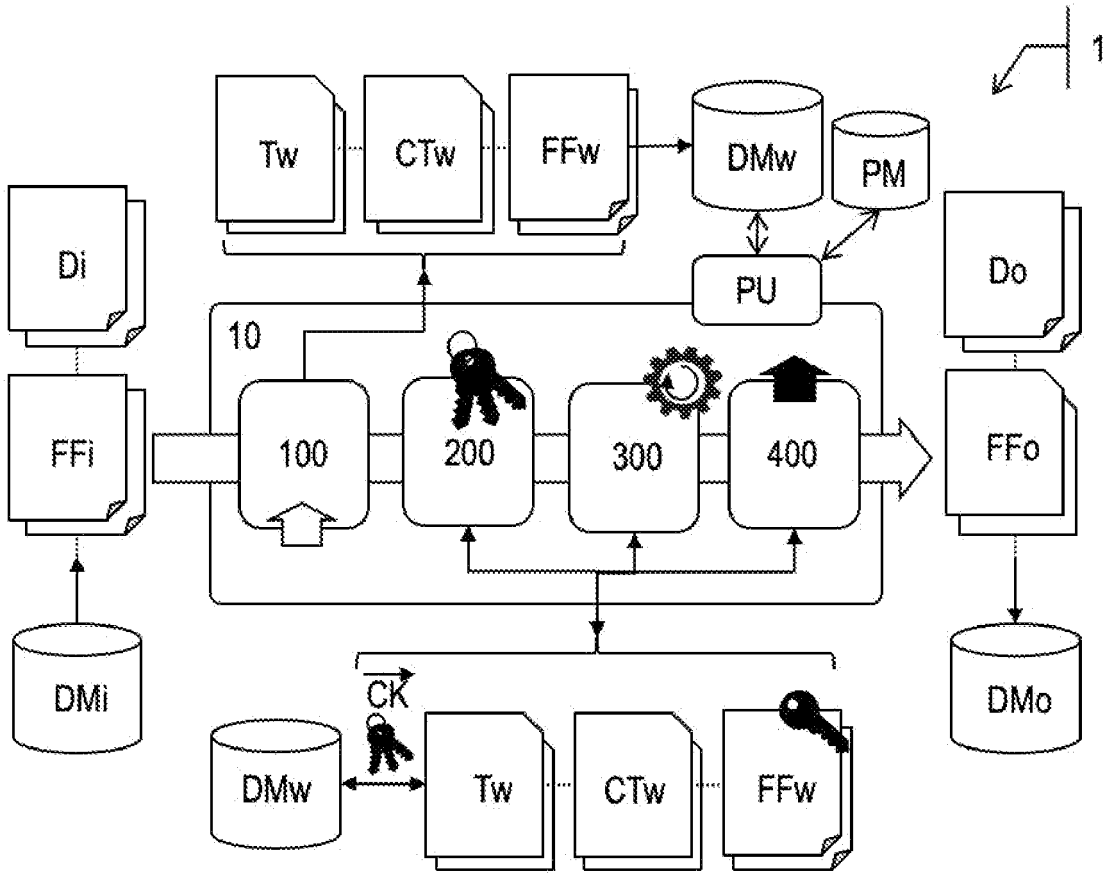
[Revendication 14] Procédé selon l'une quelconque des revendications précédentes, pour lequel le conteneur de données (Tw) consiste en une table courante et un conteneur associatif (CTw) consiste en une table associative.

[Revendication 15] Produit programme d'ordinateur comportant une ou plusieurs instructions de programme exécutables par l'unité de traitement (PU) d'un système informatique (1), lesdites instructions de programme étant chargeables dans une mémoire non volatile dudit système informatique (1) et dont l'exécution par ladite unité de traitement (PU) provoque la mise en œuvre d'un procédé (10) selon l'une quelconque des revendications précédentes.

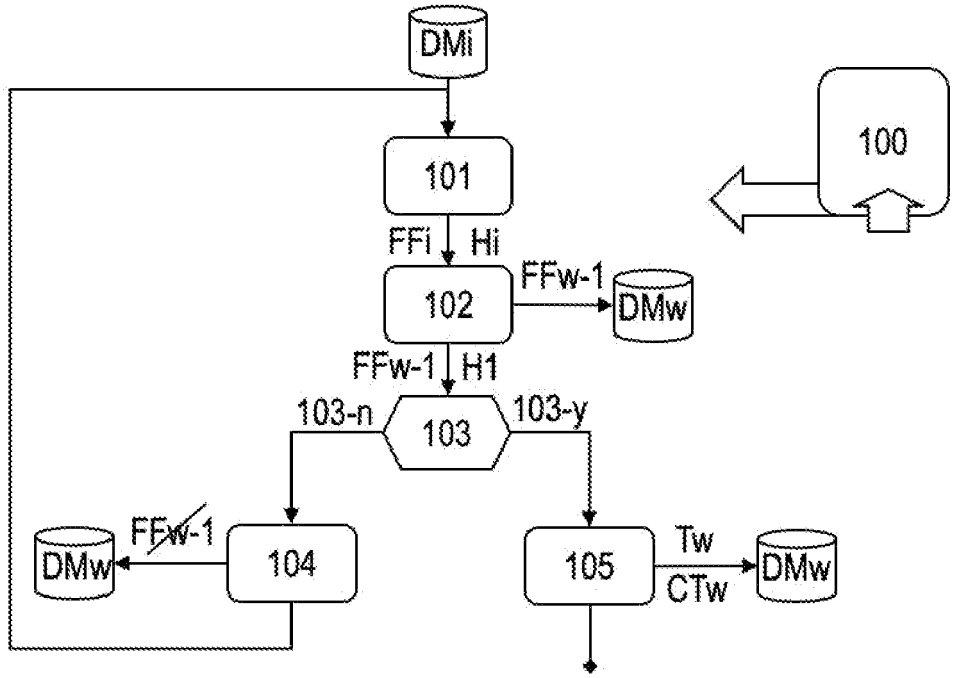
[Revendication 16] Support de mémorisation lisible par un ordinateur comportant les instructions d'un produit programme d'ordinateur selon la revendication précédente.

[Revendication 17] Système informatique (1) comprenant une unité de traitement (PU), une mémoire comprenant les instructions de programme d'un produit programme d'ordinateur selon la revendication 15.

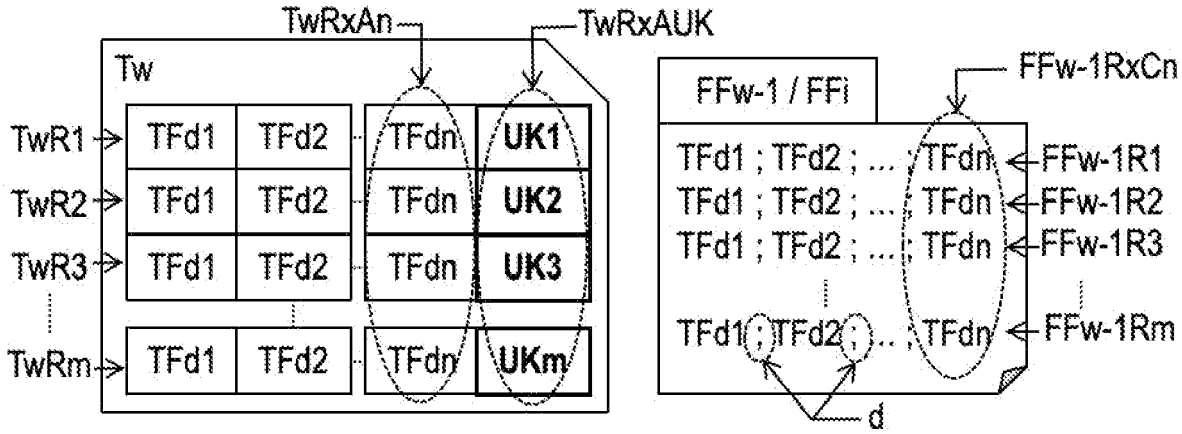
[Fig. 1]


1
Tw
CTw
FFw
DMw
PM
Di
PU
Do
10
FFi
100
200
300
400
FFo
DMi
CK
DMw
Tw
CTw
FFw
DMo

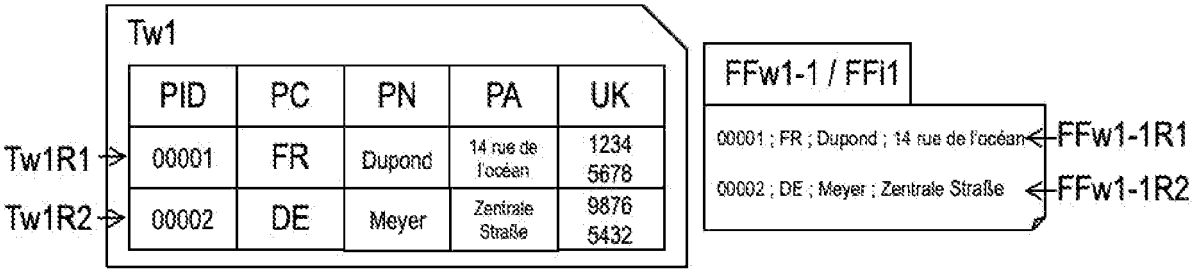
[Fig. 2]


DMi
100
101
FFi
Hi
102
FFw-1
DMw
FFw-1
H1
103-n
103
103-y
FFw-1
DMw
104
105
Tw
CTw
DMw

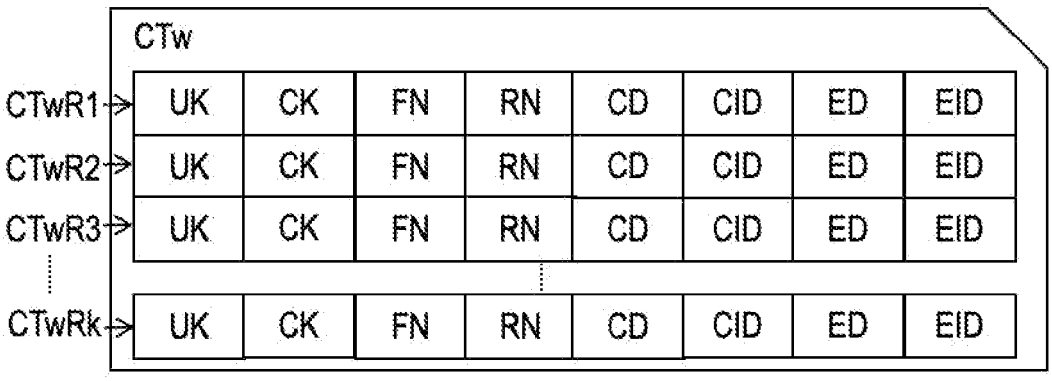
[Fig. 3]



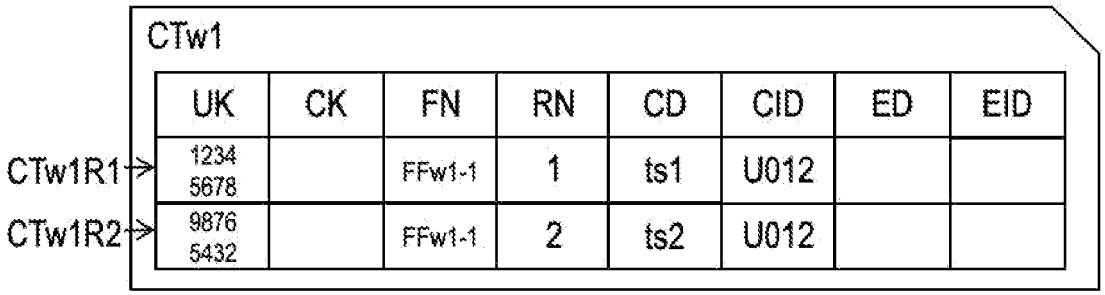
[Fig. 3A]



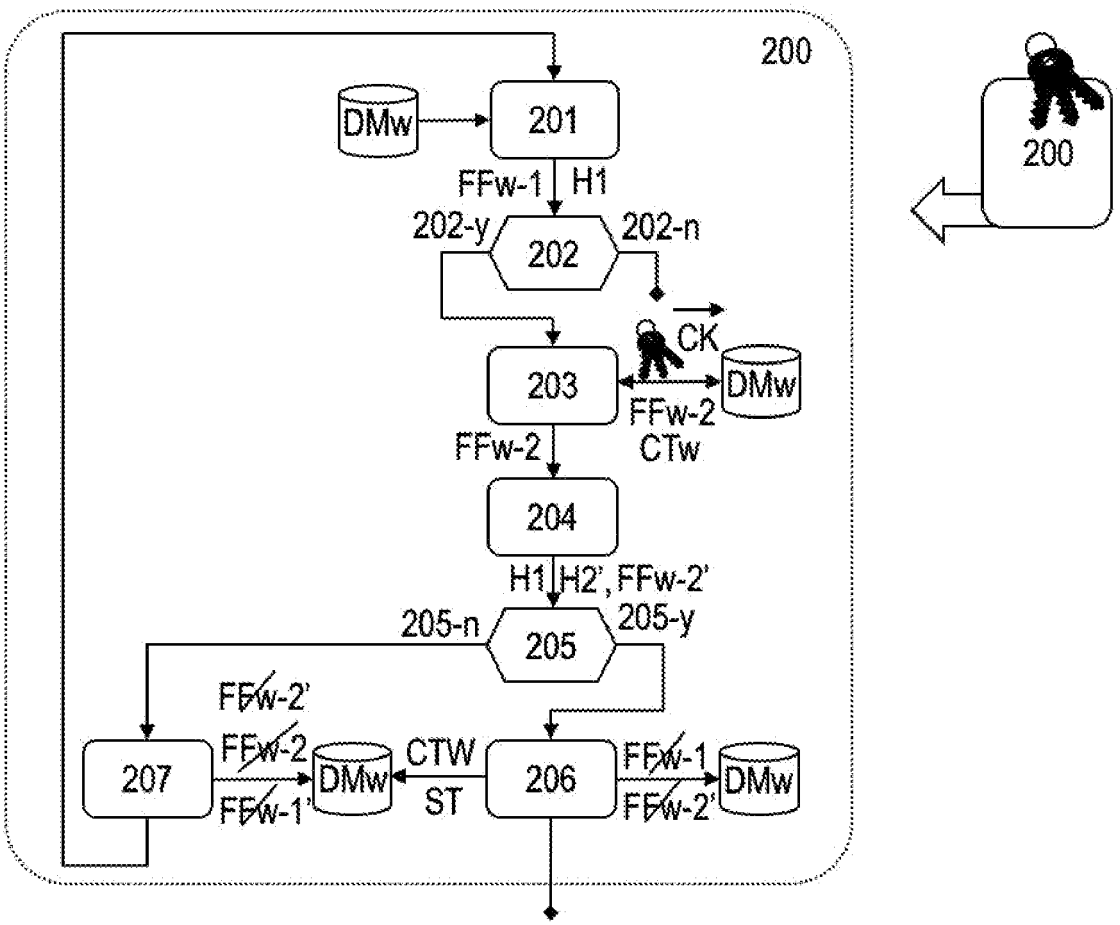
[Fig. 4]



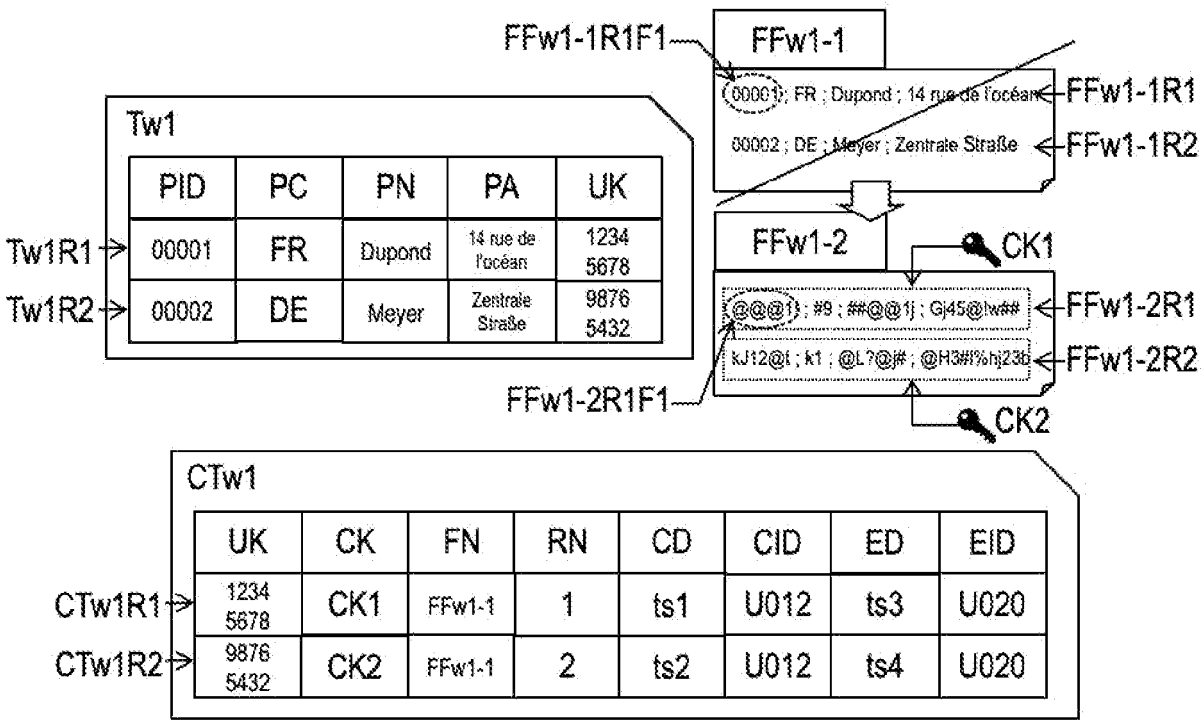
[Fig. 4A]



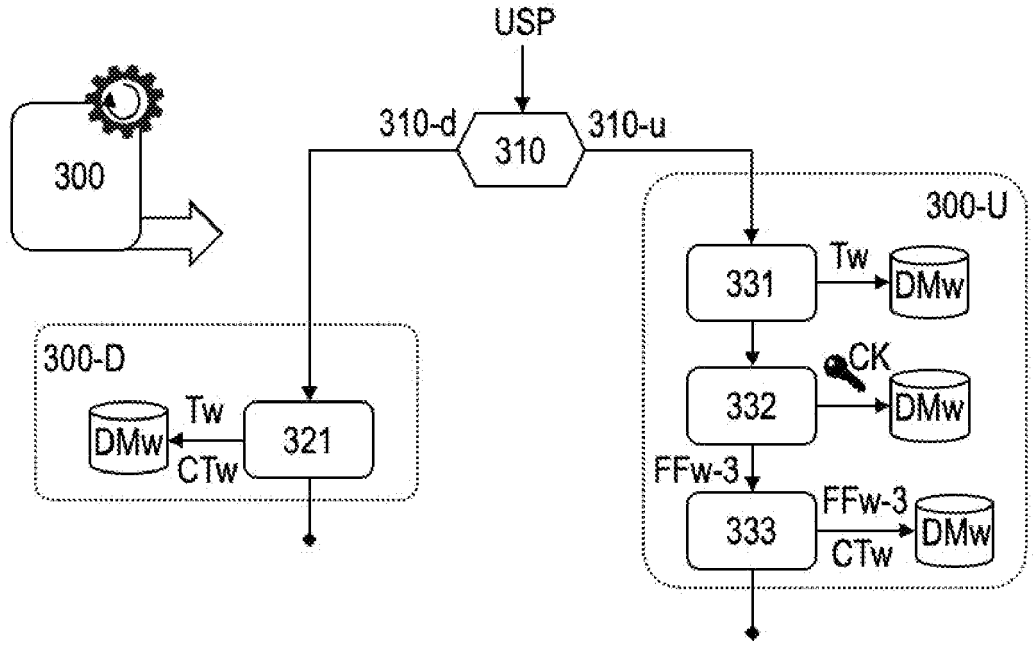
[Fig. 5]


200
DMw
201
FFw-1
H1
202-y
202
202-n
CK
203
DMw
FFw-2
CTw
FFw-2
204
H1
H2', FFw-2'
205-n
205
205-y
FFw-2'
207
FFw-2
FFw-1'
DMw
CTW
ST
206
FFw-1
FFw-2'
DMw
200

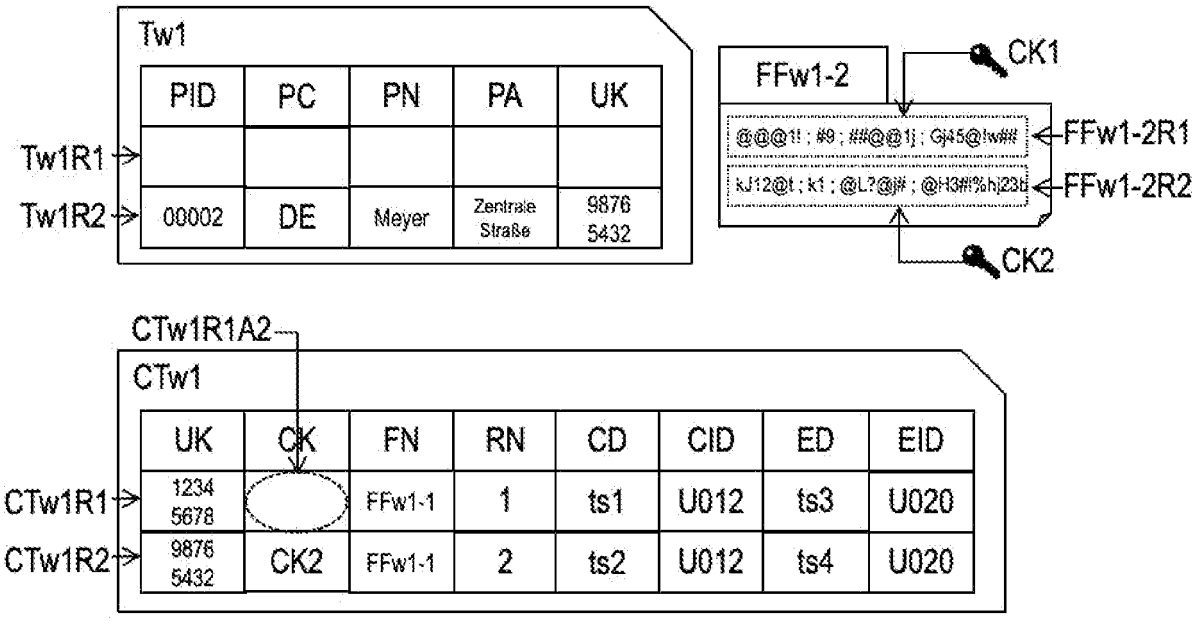
[Fig. 6]


FFw1-1R1F1
FFw1-1
00001; FR ; Dupond ; 14 rue de l'océan
FFw1-1R1
00002 ; DE ; Meyer ; Zentrale Straße
FFw1-1R2
Tw1
PID
PC
PN
PA
UK
Tw1R1
00001
FR
Dupond
14 rue de l'océan
1234 5678
Tw1R2
00002
DE
Meyer
Zentrale Straße
9876 5432
FFw1-2
CK1
@@@1 ; #9 ; ##@@1j ; Gj45@!w##
FFw1-2R1
kJ12@t ; k1 ; @L?@# ; @H3#!%hj23b
FFw1-2R2
FFw1-2R1F1
CK2
CTw1
UK
CK
FN
RN
CD
CID
ED
EID
CTw1R1
1234 5678
CK1
FFw1-1
1
ts1
U012
ts3
U020
CTw1R2
9876 5432
CK2
FFw1-1
2
ts2
U012
ts4
U020

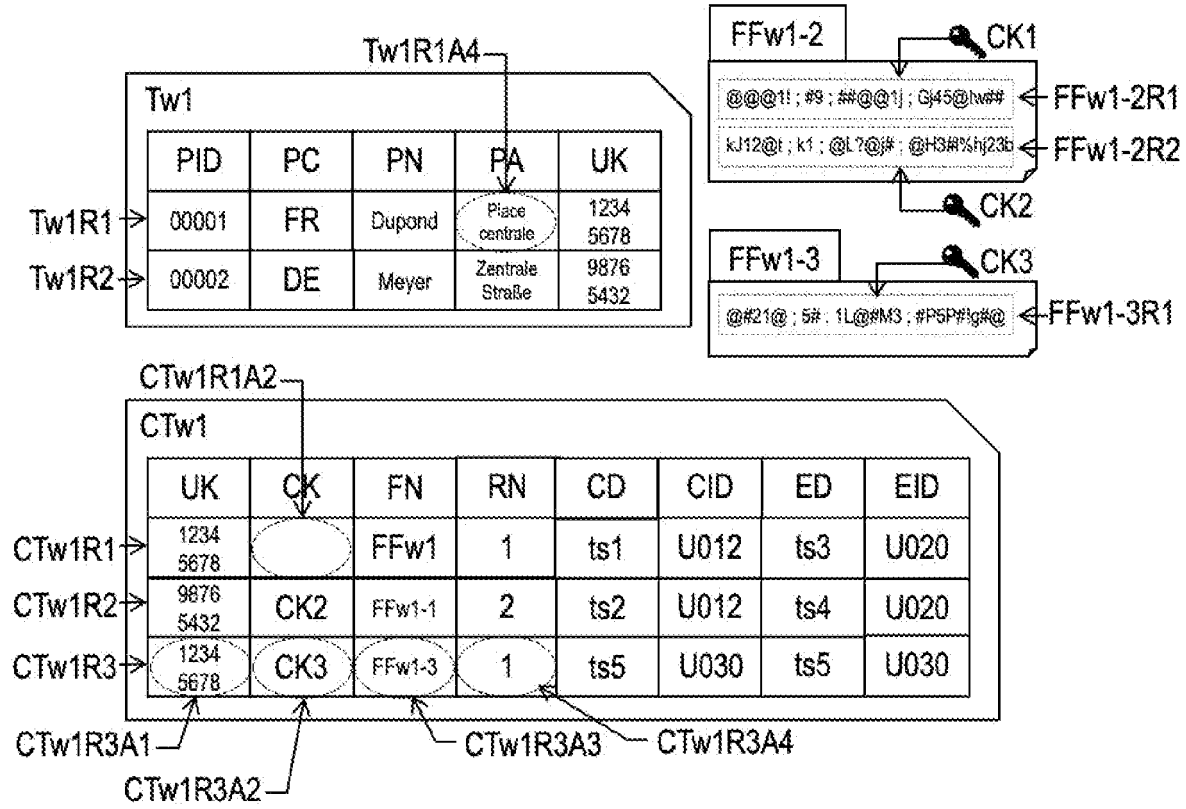
[Fig. 7]


USP
310-d
310
310-u
300
300-U
331
Tw
DMw
332
CK
DMw
FFw-3
333
FFw-3
CTw
DMw
300-D
DMw
Tw
CTw
321

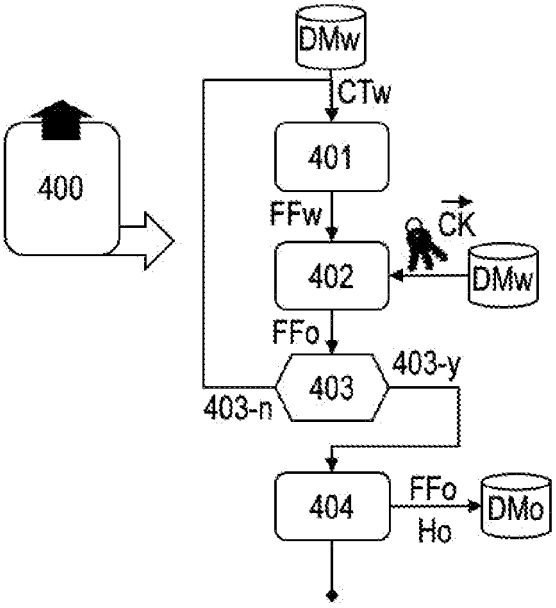
[Fig. 8]


Tw1
PID
PC
PN
PA
UK
Tw1R1
Tw1R2
00002
DE
Meyer
Zentrale Straße
9876 5432
FFw1-2
CK1
@@@1! ; #9 ; ##@@1j ; Gj45@!w##
kJ12@t ; k1 ; @L?@# ; @H3#%hj23t
FFw1-2R1
FFw1-2R2
CK2
CTw1R1A2
CTw1
UK
CK
FN
RN
CD
CID
ED
EID
CTw1R1
CTw1R2
1234 5678
FFw1-1
1
ts1
U012
ts3
U020
9876 5432
CK2
FFw1-1
2
ts2
U012
ts4
U020

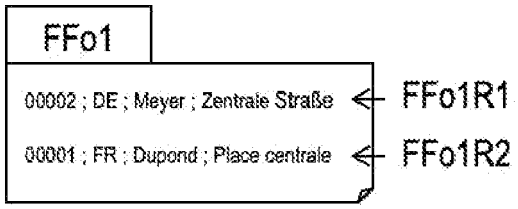
[Fig. 9]



[Fig. 10]



[Fig. 11]



RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL DE LA PROPRIÉTÉ INDUSTRIELLE

RAPPORT DE RECHERCHE PRÉLIMINAIRE

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

N° d'enregistrement national

FA 917135
FR 2301077

DOCUMENTS CONSIDÉRÉS COMME PERTINENTS		Revendication(s) concernée(s)	Classement attribué à l'invention par l'INPI
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes		
X	US 2020/026710 A1 (PRZADA ZBIGNIEW [CA] ET AL) 23 janvier 2020 (2020-01-23) * abrégé * * alinéa [0122] - alinéa [0182]; figures 1A-3 * * alinéa [0308] - alinéa [0328]; figures 17-27 * -----	1-17	G06F 21/60 G06F 21/64
X	US 2020/409914 A1 (NADIMPALLI VENKATA RAMANA [US] ET AL) 31 décembre 2020 (2020-12-31) * abrégé * * alinéa [0011] - alinéa [0023]; figures 1-2H * * alinéa [0032] - alinéa [0038]; figure 3 * * alinéa [0045] - alinéa [0047] * -----	1-17	
			DOMAINES TECHNIQUES RECHERCHÉS (IPC)
			G06F

Date d'achèvement de la recherche	Examinateur
8 septembre 2023	Barieux, Marc

CATÉGORIE DES DOCUMENTS CITÉS

X : particulièrement pertinent à lui seul
Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie
A : arrière-plan technologique
O : divulgation non-écrite
P : document intercalaire

T : théorie ou principe à la base de l'invention
E : document de brevet bénéficiant d'une date antérieure à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure.
D : cité dans la demande
L : cité pour d'autres raisons
.....
& : membre de la même famille, document correspondant

EPO FORM 1503 12.99 (P04C14)

ANNEXE AU RAPPORT DE RECHERCHE PRÉLIMINAIRE RELATIF A LA DEMANDE DE BREVET FRANÇAIS NO. FR 2301077 FA 917135

La présente annexe indique les membres de la famille de brevets relatifs aux documents brevets cités dans le rapport de recherche préliminaire visé ci-dessus.
Les dits membres sont contenus au fichier informatique de l'Office européen des brevets à la date du **08-09-2023**
Les renseignements fournis sont donnés à titre indicatif et n'engagent pas la responsabilité de l'Office européen des brevets, ni de l'Administration française

Document brevet cité au rapport de recherche	Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
US 2020026710 A1	23-01-2020	CA 3050220 A1	19-01-2020
		US 2020026710 A1	23-01-2020
		US 2023062655 A1	02-03-2023
US 2020409914 A1	31-12-2020	US 2020409914 A1	31-12-2020
		US 2022043785 A1	10-02-2022

EPO FORM P0465