

⑫

BREVET D'INVENTION

B1

⑤④ Procédé de gestion de droits associés à un objet.

②② Date de dépôt : 14.03.22.

③③ Priorité :

⑥③ Références à d'autres documents nationaux
apparentés :

☐ Demande(s) d'extension :

⑦① Demandeur(s) : *Bulgari Horlogerie S.A Société
Anonyme* — CH.

④③ Date de mise à la disposition du public
de la demande : 15.09.23 Bulletin 23/37.

④⑤ Date de la mise à disposition du public du
brevet d'invention : 28.06.24 Bulletin 24/26.

⑤⑥ Liste des documents cités dans le rapport de
recherche :

Se reporter à la fin du présent fascicule

⑦② Inventeur(s) : *PICCIRILLO Ennio et BALDI Davide.*

⑦③ Titulaire(s) : *Bulgari Horlogerie S.A Société
Anonyme.*

⑦④ Mandataire(s) : *NOVAIMO.*



Description

Titre de l'invention : Procédé de gestion de droits associés à un objet.

Domaine Technique de l'invention

[0001] L'invention concerne un procédé de gestion de droits associés à un objet. L'invention concerne encore l'utilisation d'un tel objet dans un procédé de gestion des droits attachés à cet objet. L'invention concerne aussi un dispositif de gestion de droits associés à un tel objet. L'invention concerne aussi un produit programme de gestion de droits associés à un tel objet. L'invention concerne aussi un support d'enregistrement de données dédié à la mise en œuvre du procédé de gestion de droits associés à un tel objet. L'invention concerne enfin un signal d'un support de données, portant un tel produit programme d'ordinateur

Etat de la technique antérieure

[0002] Dans certains domaines, notamment dans le domaine des produits de luxe, il existe un besoin de tenir un registre des produits manufacturés et vendus. Il peut aussi exister un intérêt de suivre la chaîne de propriété tout au long de la vie des produits. Or, un tel suivi ne peut être garanti une fois le produit mis sur le marché. Un registre ne peut être tenu que si les différents propriétaires du produit se font connaître auprès du teneur du registre, notamment auprès de la manufacture tenant le registre.

Présentation de l'invention

[0003] Le but de l'invention est de fournir un procédé de gestion de droits associés à un objet remédiant aux inconvénients évoqués ci-dessus et améliorant les procédés de gestion connus de l'art antérieur. En particulier, l'invention propose un procédé de gestion de droits associés à un objet qui permette de garantir une traçabilité et une authenticité tout en permettant de gérer en outre des droits associés à des produits et/ou services annexes à l'objet.

Résumé de l'invention

[0004] Selon l'invention, un procédé permet de gérer des droits associés à un objet, tel qu'une pièce d'horlogerie ou de joaillerie. Le procédé comprend les étapes suivantes :

- création d'un premier jeton numérique associé à un premier droit, notamment au droit de propriété, sur l'objet, le premier jeton numérique étant stocké et authentifié grâce à un premier protocole de chaîne de blocs,
- création d'un deuxième jeton numérique, notamment un jeton numérique non fongible, associé à un deuxième droit, notamment un droit d'accès à un contenu numérique, en particulier un contenu multimédia, associé à l'objet, le deuxième jeton numérique étant stocké et authentifié grâce à un deuxième protocole de chaîne de

blocs,

- création d'un contrat intelligent liant les droits sur les premier et deuxième jetons numériques de sorte que le droit de propriété sur l'un des premier et deuxième jetons numériques ne peut pas être transféré sans le droit de propriété sur l'autre des premier et deuxième jetons numériques.

[0005] Le premier protocole de chaîne de blocs et le deuxième protocole de chaîne de blocs sont avantageusement d'un même type ou le premier protocole de chaîne de blocs et le deuxième protocole de chaîne de blocs sont avantageusement un même protocole de chaîne de blocs.

[0006] Le contrat intelligent peut impliquer la création, grâce à un protocole de chaîne de blocs, d'un troisième jeton numérique incluant des informations comprenant :

- au moins une information contenue dans le premier jeton numérique, tel que notamment une information de l'objet et/ou un numéro de série de l'objet et/ou un identifiant de la transaction de création du premier jeton numérique et/ou la date d'activation du certificat d'authenticité et/ou la date de la vente du produit et/ou un identifiant du premier jeton numérique, et

- au moins une information contenue dans le deuxième jeton numérique, tel que notamment un identifiant du deuxième jeton numérique.

[0007] De préférence, le contrat intelligent est configuré de sorte que le transfert du droit de propriété sur le premier jeton numérique implique le transfert du droit de propriété sur le deuxième jeton numérique et/ou le transfert du droit de propriété sur le troisième jeton numérique, et/ou

le transfert du droit de propriété sur le deuxième jeton numérique implique le transfert du droit de propriété sur le premier jeton numérique et/ou le transfert du droit de propriété sur le troisième jeton numérique, et/ou

le transfert du droit de propriété sur le troisième jeton numérique implique le transfert du droit de propriété sur le premier jeton numérique et/ou le transfert du droit de propriété sur le deuxième jeton numérique.

[0008] Le premier jeton numérique et/ou le deuxième jeton numérique peuvent être associés à des portefeuilles numériques de crypto-actifs, notamment à un même portefeuille numérique de crypto-actifs.

[0009] Le portefeuille numérique de crypto-actifs ou l'un des portefeuilles numériques de crypto-actifs peut être associé à :

- une clé publique, ou

- un code associé à la clé publique,

la clé publique ou le code étant inscrit sur :

- l'objet, notamment sur un composant d'une pièce d'horlogerie, en particulier sur un barillet ou sur un rochet de barillet d'une pièce d'horlogerie, ou

- sur un support matériel vendu avec l'objet, ou
- dans un document numérique, comme un fichier informatique ou un message, notamment un MMS (de l'anglais Multimedia Messaging Service) ou un SMS (de l'anglais Short Message).

- [0010] La clé publique ou le code associé à la clé publique peut être inscrit par impression ou par gravure ou par collage d'une étiquette.
- [0011] Le code associé à la clé publique peut être un code lisible par une machine, comme une étiquette NFC ou comme un code-barres, en particulier comme un code matriciel ou un code QR.
- [0012] Le portefeuille numérique de crypto-actifs ou l'un des portefeuilles numériques de crypto-actifs peut être associé à :
- une clé privée, ou
 - un code associé à la clé privée,
- la clé privée ou le code associé à la clé privée étant inscrit :
- sur un support matériel annexé à l'objet, notamment sur une carte, sur une étiquette ou sur un document physique, et/ou
 - dans un document numérique, comme un fichier informatique ou un message, notamment un MMS ou un SMS.
- [0013] Le code associé à la clé privée peut être un code lisible par une machine, comme une étiquette NFC ou comme un code-barres, en particulier comme un code matriciel ou un code QR.
- [0014] L'invention porte aussi sur une utilisation d'un objet ou d'un support matériel annexé à l'objet et/ou d'un accessoire, notamment d'une pièce d'horlogerie, en particulier d'un barillet ou d'un rochet de barillet d'une pièce d'horlogerie comme support d'une clé publique ou d'un code associé à une clé publique d'un portefeuille numérique de crypto-actifs.
- [0015] Selon l'invention, un dispositif, notamment une architecture informatique distribuée, permet de gérer des droits associés à un objet, le dispositif comprenant des éléments matériels et/ou logiciels mettant en œuvre le procédé défini précédemment, notamment des éléments matériels et/ou logiciels conçus pour mettre en œuvre le procédé défini précédemment.
- [0016] Selon l'invention, un dispositif comprend des moyens de mettre en œuvre des étapes du procédé défini précédemment.
- [0017] Selon l'invention, un produit programme d'ordinateur comprend des instructions de code de programme enregistrées sur un support lisible par ordinateur pour mettre en œuvre les étapes du procédé défini précédemment lorsque ledit programme fonctionne sur un ordinateur.
- [0018] L'invention porte aussi sur un produit programme d'ordinateur téléchargeable depuis

un réseau de communication et/ou enregistré sur un support de données lisible par un ordinateur et/ou exécutable par un ordinateur. Le produit programme comprend des instructions qui, lorsque le programme est exécuté par l'ordinateur, conduisent celui-ci à mettre en œuvre des étapes du procédé défini précédemment.

[0019] L'invention porte encore sur un support d'enregistrement de données, lisible par un ordinateur, sur lequel est enregistré un programme d'ordinateur comprenant des instructions de code de programme de mise en œuvre du procédé défini précédemment.

[0020] L'invention porte encore sur un support d'enregistrement lisible par ordinateur comprenant des instructions qui, lorsqu'elles sont exécutées par un ordinateur, conduisent celui-ci à mettre en œuvre des étapes du procédé défini précédemment.

[0021] L'invention porte encore sur un signal d'un support de données, portant le produit programme d'ordinateur défini précédemment.

Présentation des figures

[0022] Les objets, caractéristiques et avantages de la présente invention sont exposés en détail dans la description suivante d'un mode de réalisation particulier fait à titre non-limitatif en relation avec les figures jointes.

[0023] La [Fig.1] est une vue d'une pièce d'horlogerie dont les droits sont gérés selon un mode d'exécution du procédé selon l'invention.

Description détaillée

[0024] Le mode d'exécution du procédé de gestion décrit par la suite est explicité en application à un objet 1 étant une pièce d'horlogerie 1. Toutefois, ce procédé peut être appliqué à tout type d'objet physique ou immatériel, produit ou service. Il peut notamment aussi être appliqué à une pièce de joaillerie.

[0025] L'objet est l'élément principal dont les droits sont gérés selon le procédé. L'objet peut être commercialisé avec un élément annexe ou un élément auxiliaire qui peut aussi être de tout type comme un objet physique ou immatériel, produit ou service.

[0026] Dans le mode de réalisation décrit, alors que l'élément principal est la pièce d'horlogerie, l'élément auxiliaire comprend un service d'accès à des données, notamment à des données multimédia pouvant contenir des images, des contenus sonores, des vidéos. Ces données peuvent par exemple retracer l'histoire du projet créatif ayant donné lieu à la mise de l'objet sur le marché. Ces données peuvent par exemple contenir des témoignages des personnes impliquées dans le projet comme le témoignage d'un directeur artistique ou comme le témoignage d'un directeur technique s'adressant directement au propriétaire de la pièce d'horlogerie.

[0027] L'élément auxiliaire peut aussi contenir une œuvre d'art numérique ou physique vendue avec la pièce d'horlogerie et produite par exemple par un artiste ayant collaboré avec la manufacture, en particulier ayant collaboré avec la manufacture dans

le processus de création de la pièce d'horlogerie. L'élément auxiliaire peut aussi être constitué par un objet accessoire comme par exemple un écrin pour la pièce d'horlogerie.

- [0028] Une fois l'élément principal (par exemple, l'objet constitué par la pièce d'horlogerie) et l'élément auxiliaire (par exemple, le service d'accès à des données) produits par la manufacture, un mode d'exécution du procédé objet de l'invention est mis en œuvre.
- [0029] Le procédé permet de gérer des droits sur l'élément principal. En particulier, il est possible de gérer le droit de propriété sur l'élément principal. Ce droit de propriété peut être démembré en nue-propriété et usufruit. La gestion de tout autre droit associé à l'élément principal peut être envisagée.
- [0030] Le procédé permet aussi de gérer des droits sur l'élément auxiliaire. En particulier, il est possible de gérer le droit de propriété sur l'élément auxiliaire. Ce droit de propriété peut être démembré en nue-propriété et usufruit. La gestion de tout autre droit associé à l'élément auxiliaire peut être envisagée. Ces droits sur l'élément auxiliaire peuvent être vus comme des droits associés à l'élément principal, l'élément auxiliaire étant associé à l'élément principal.
- [0031] L'association entre l'élément auxiliaire et l'élément principal peut être de toute nature comme notamment fonctionnelle, commerciale, conceptuelle, ou artistique. L'association peut en outre être seulement la conséquence de la mise en œuvre du procédé de gestion des droits.
- [0032] Dans une première étape du procédé de gestion, on crée ou on frappe (en anglais « to mint ») un premier jeton numérique (en anglais « token ») associé à un premier droit, notamment au droit de propriété, sur la pièce d'horlogerie. Le premier jeton numérique est par exemple du type non-fongible (en anglais « NFT » pour non-fongible token). Le premier jeton numérique est stocké et authentifié grâce à un premier protocole de chaîne de blocs (en anglais « blockchain »). Le premier protocole est par exemple le protocole de la chaîne de blocs AURA.
- [0033] Dans une deuxième étape du procédé de gestion, on crée ou on frappe un deuxième jeton numérique associé à un deuxième droit, notamment un droit d'accès à un contenu numérique, en particulier un contenu multimédia, associé à la pièce d'horlogerie. Comme vu précédemment, le deuxième droit peut être un droit sur un élément auxiliaire. Le deuxième jeton numérique est par exemple du type non-fongible. Le deuxième jeton numérique est stocké et authentifié grâce à un deuxième protocole de chaîne de blocs. Le deuxième protocole est par exemple le protocole de la chaîne de blocs POLYGON.
- [0034] Par exemple, le deuxième droit peut être un droit d'accès à un contenu numérique se trouvant sur un site internet qui appartient à la manufacture. Le deuxième droit peut donc être un droit d'usufruit sur le site internet alors que la nue-propriété du site

internet demeure détenue par la manufacture.

- [0035] Le premier protocole de chaîne de blocs et le deuxième protocole de chaîne de blocs peuvent être d'un même type, voire même le premier protocole de chaîne de blocs et le deuxième protocole de chaîne de blocs peuvent être un même protocole de chaîne de blocs.
- [0036] La deuxième étape peut être mise en œuvre de différentes manières.
- [0037] Par exemple, dans un premier mode d'exécution, la manufacture déclenche l'opération de création du deuxième jeton numérique dès que la pièce d'horlogerie est terminée ou lors de la vente de la pièce d'horlogerie de la manufacture au premier client ou premier propriétaire.
- [0038] Par exemple, dans un deuxième mode d'exécution, c'est le premier client ou le premier propriétaire qui déclenche l'opération de création du deuxième jeton numérique après qu'il a pris possession de la pièce d'horlogerie.
- [0039] Afin de mettre en œuvre ce deuxième mode d'exécution :
 - une clé publique, ou
 - un code associé à la clé publique,
 est inscrit sur la pièce d'horlogerie 1, notamment sur un composant 2 de la pièce d'horlogerie, en particulier sur un barillet 2 ou sur un rochet 3 de barillet 2 de la pièce d'horlogerie 1.
- [0040] Alternativement, la pièce d'horlogerie peut être fournie avec un support matériel comme un document, une étiquette, un certificat, un écrin ou tout accessoire sur lequel est inscrit la clé publique ou le code associé à la clé publique. Alternativement encore, la pièce d'horlogerie peut être fournie avec un document numérique, comme un fichier informatique ou un message, notamment un MMS ou un SMS, contenant la clé publique ou le code associé à la clé publique. Ce document numérique peut être remis ou envoyé lors de l'achat de la pièce d'horlogerie.
- [0041] En complément, lors de la vente de la pièce d'horlogerie,
- [0042]
 - une clé privée, ou
 - un code associé à la clé privée,
 peut être fourni avec un support matériel comme un document, une étiquette, un certificat, un écrin ou tout accessoire sur lequel est inscrit la clé privée ou le code associé à la clé privée. Alternativement, la pièce d'horlogerie peut être fournie avec un document numérique, comme un fichier informatique ou un message, notamment un MMS ou un SMS, contenant la clé privée ou le code associé à la clé privée. Ce document numérique peut être remis ou envoyé lors de l'achat de la pièce d'horlogerie. De préférence, la délivrance de la clé privée ou du code associé à la clé privée est sécurisée. Par exemple, lorsque la clé privée ou le code associé à la clé privée est apposé sur un document ou plus généralement sur tout support physique, il est

recouvert d'un revêtement le masquant de sorte que seul le premier propriétaire pourra le voir après retrait du revêtement.

- [0043] L'une et/ou l'autre des clés et/ou l'un et/ou l'autre des codes associés aux clés peuvent être inscrits par impression ou par gravure ou par collage d'une étiquette.
- [0044] De préférence, l'un et/ou l'autre des codes associés aux clés est un code lisible par radiofréquences ou optiquement par une machine, comme une étiquette NFC (en anglais « near field contact »), un code-barres, en particulier comme un code matriciel ou un code QR.
- [0045] Ainsi, après l'achat de la pièce d'horlogerie, le premier propriétaire de la pièce d'horlogerie peut en ayant accès à la clé publique et à la clé privée déclencher la création du deuxième jeton numérique. Pour ce faire, il peut se rendre sur un site internet dédié et entrer les clés publique et privée pour lancer la création du deuxième jeton numérique.
- [0046] Dès leur création, les premier et deuxième jetons numériques sont associés à un ou deux portefeuilles numériques de crypto-actifs (en anglais « wallet »). Ils sont dès leur création associés à un portefeuille numérique du premier propriétaire ou transférés d'un portefeuille numérique de la manufacture à un portefeuille numérique du premier propriétaire lors de la vente de la pièce d'horlogerie. Le premier jeton numérique et le deuxième jeton numérique peuvent être associés à des portefeuilles numériques de crypto-actifs différents. Avantageusement, le premier jeton numérique et le deuxième jeton numérique peuvent être associés à un même portefeuille numérique de crypto-actifs.
- [0047] Les clés publique et privée mentionnées précédemment sont avantageusement les clés publique et privée du portefeuille numérique de crypto-actifs du premier propriétaire et contenant, après vente de la pièce d'horlogerie de la manufacture au premier propriétaire, le premier jeton numérique et le deuxième jeton numérique.
- [0048] Avantageusement, dans une troisième étape du procédé de gestion, on crée un contrat intelligent (en anglais « smart contract ») liant les droits sur les premier et deuxième jetons numériques de sorte que le droit de propriété sur l'un des premier et deuxième jetons numériques ne peut pas être transféré sans le droit de propriété sur l'autre des premier et deuxième jetons numériques. Autrement dit, les premier et deuxième jetons numériques ne peuvent pas être transférés l'un sans l'autre. De préférence encore, en supposant que les premier et deuxième jetons numériques sont à l'origine associés à un premier portefeuille numérique, ils ne peuvent être transférés qu'ensemble pour être associés à un deuxième portefeuille numérique suite à un transfert.
- [0049] De préférence, le contrat intelligent implique la création ou la frappe, grâce à un troisième protocole de chaîne de blocs, d'un troisième jeton numérique. Le troisième protocole de chaîne de blocs est de préférence le même protocole que le premier

protocole de chaîne de blocs ou que le deuxième protocole de chaîne de blocs

[0050] Le troisième jeton numérique contient avantageusement des informations comprenant :

- au moins une information contenue dans le premier jeton numérique, tel que notamment une information de la pièce d'horlogerie et/ou un numéro de série de la pièce d'horlogerie et/ou un identifiant de la transaction de création du premier jeton numérique et/ou la date d'activation du certificat d'authenticité et/ou la date de la vente de la pièce d'horlogerie et/ou un identifiant du premier jeton numérique, et
- au moins une information contenue dans le deuxième jeton numérique, tel que notamment un identifiant du deuxième jeton numérique.

[0051] En alternative, le premier jeton numérique et le deuxième jeton numérique (et éventuellement le troisième jeton numérique) peuvent être créés simultanément ou quasi simultanément. Dans un tel cas, le premier jeton numérique peut être créé en incluant une donnée relative au deuxième jeton numérique, de même que le deuxième jeton numérique peut être créé en incluant une donnée relative au premier jeton numérique. Ainsi, les premier et deuxième jetons peuvent être liés dès leur création.

[0052] De manière préférée, le contrat intelligent est configuré de sorte que :

- le transfert du droit de propriété sur le premier jeton numérique implique le transfert du droit de propriété sur le deuxième jeton numérique et/ou le transfert du droit de propriété sur le troisième jeton numérique, et/ou
- le transfert du droit de propriété sur le deuxième jeton numérique implique le transfert du droit de propriété sur le premier jeton numérique et/ou le transfert du droit de propriété sur le troisième jeton numérique, et/ou
- le transfert du droit de propriété sur le troisième jeton numérique implique le transfert du droit de propriété sur le premier jeton numérique et/ou le transfert du droit de propriété sur le deuxième jeton numérique.

[0053] En pratique, lorsqu'on déclenche le transfert d'un des jetons numériques d'un portefeuille numérique de crypto-actifs à un autre portefeuille numérique de crypto-actifs, on déclenche automatiquement le transfert de l'autre des jetons numériques ou des autres jetons numériques. Avantageusement, dans une application logicielle permettant ce transfert, lorsqu'on ordonne le transfert de l'un des jetons numériques, l'application émet un message alertant des conséquences du transfert sur les autres jetons numériques.

[0054] De manière avantageuse encore, le contrat intelligent peut être configuré pour limiter le transfert du ou des jetons numériques, par exemple pour :

- conditionner le transfert du ou des jetons numériques à l'autorisation d'un tiers, tel que par exemple la manufacture,
- limiter la fréquence de transfert du ou des jetons numériques, par exemple limiter le

transfert des jetons numériques à un transfert par an,
 - interdire le transfert du ou des jetons numériques.

[0055] Notamment, l'une ou l'autre de ces limitations peuvent être mises en œuvre en cas de perte ou de vol de la pièce d'horlogerie.

[0056] Dans le cas d'interdiction du transfert du ou des jetons numériques, il peut être prévu que ce soit l'ensemble du portefeuille numérique de crypto-actifs contenant les jetons numériques qui puisse être transféré d'un propriétaire à un autre.

[0057] Grâce aux procédés décrits plus haut, il est possible de gérer les droits sur un élément principal tel qu'une pièce d'horlogerie. En particulier, il est possible d'authentifier et de tracer les différentes transactions passées sur les protocoles de chaîne de blocs.

[0058] Avantageusement, toutes les transactions sur les protocoles de chaîne de blocs peuvent être sauvegardées par la manufacture. A cette fin notamment, la manufacture peut gérer des nœuds de réseau de l'architecture permettant de réaliser des traitements de constructions des chaînes de blocs impliquées dans le stockage et l'authentification des jetons numériques.

[0059] En remarque, le code associé à la clé publique et apposé sur la pièce d'horlogerie elle-même est avantageusement utilisé par le propriétaire pour se connecter à un site internet proposant le contenu numérique déjà évoqué précédemment. Par exemple, une lecture, à l'aide d'un téléphone intelligent (en anglais « smartphone »), du code QR apposé sur la pièce d'horlogerie, permet de connecter automatiquement le téléphone au site internet proposant le contenu.

[0060] L'invention porte aussi sur l'utilisation d'un objet et/ou d'un support matériel annexé à l'objet et/ou d'un accessoire, notamment de la pièce d'horlogerie 1, en particulier du barillet 2 ou du rochet 3 de barillet 2 de la pièce d'horlogerie 1, comme support de la clé publique ou du code 4 associé à la clé publique du portefeuille numérique de crypto-actifs.

[0061] Un mode de réalisation d'un dispositif, notamment d'une architecture informatique distribuée, permet de gérer les droits associés à l'objet. Ce mode de réalisation comprend des éléments matériels et/ou logiciels mettant en œuvre le procédé décrit précédemment.

[0062] Un mode de réalisation d'un produit programme d'ordinateur comprend des instructions de code de programme enregistrées sur un support lisible par ordinateur pour mettre en œuvre des étapes du procédé décrit précédemment lorsque ledit programme fonctionne sur un ordinateur.

[0063] Un exemple de code de produit programme utilisé pour mettre en œuvre le procédé de gestion décrit est détaillé ci-après.

[0064] *pragma solidity ^0.8.0;*

[0065] *interface IERC721Metadata {*

```

[0066] function name() external view returns (string memory _name);
[0067] function symbol() external view returns (string memory _symbol);
[0068] function tokenURI(uint256 _tokenId) external view returns (string memory);
[0069] }
[0070] interface IERC721 {
[0071] event Transfer(
[0072] address indexed _from,
[0073] address indexed _to,
[0074] uint256 indexed _tokenId
[0075] );
[0076] event Approval(
[0077] address indexed _owner,
[0078] address indexed _approved,
[0079] uint256 indexed _tokenId
[0080] );
[0081] event ApprovalForAll(
[0082] address indexed _owner,
[0083] address indexed _operator,
[0084] bool _approved
[0085] );
[0086] function balanceOf(address _owner) external view returns (uint256);
[0087] function ownerOf(uint256 _tokenId) external view returns (address);
[0088] function safeTransferFrom(
[0089] address _from,
[0090] address _to,
[0091] uint256 _tokenId,
[0092] bytes calldata data
[0093] ) external;
[0094] function safeTransferFrom(
[0095] address _from,
[0096] address _to,
[0097] uint256 _tokenId
[0098] ) external;
[0099] function transferFrom(
[0100] address _from,
[0101] address _to,
[0102] uint256 _tokenId
[0103] ) external;

```

```

[0104] function approve(address _approved, uint256 _tokenId) external;
[0105] function setApprovalForAll(address _operator, bool _approved) external;
[0106] function getApproved(uint256 _tokenId) external view returns (address);
[0107] function isApprovedForAll(address _owner, address _operator)
[0108] external
[0109] view
[0110] returns (bool);
[0111] }
[0112] interface IERC721Receiver {
[0113] function onERC721Received(
[0114] address _operator,
[0115] address _from,
[0116] uint256 _tokenId,
[0117] bytes calldata _data
[0118] ) external returns (bytes4);
[0119] }
[0120] interface ILuxochainNFT {
[0121] event OwnershipTransferred(address indexed, address indexed);
[0122] function owner() external view returns (address);
[0123] function exists(uint256 tokenId) external view returns (bool);
[0124] function getStorageType() external view returns (uint256);
[0125] function isLuxochainNFT() external view returns (bool);
[0126] function transferOwnership(address newIssuer) external;
[0127] function totalSupply() external view returns (uint256);
[0128] function count() external view returns (uint256);
[0129] event TokenFrozen(uint256 tokenId, address unfreezableAddress);
[0130] event TokenUnfrozen(uint256 tokenId, address newOwner);
[0131] function safeMint(
[0132] uint256 tokenId,
[0133] address to,
[0134] string calldata tokenMetadataURI
[0135] ) external;
[0136] function multipleSafeMint(
[0137] uint256[] calldata _tokensIds,
[0138] address _to,
[0139] string[] calldata _tokenMetadataURIs
[0140] ) external;
[0141] function freezeToken(uint256 tokenId, address unfreezeAddress) external;

```

```

[0142]     function unfreezeToken(uint256 tokenId, address newOwner) external;
[0143]     function isTokenFrozen(uint256 tokenId) external view returns (bool);
[0144]     function burn(uint256 tokenId) external;
[0145]     }
[0146]     contract LUXO_BLG_NFT is IERC721, IERC721Metadata, ILuxochainNFT {
[0147]         bytes4 private constant _INTERFACE_ID_ERC721_METADATA = 0x5b5e139f;
[0148]         bytes4 private constant _INTERFACE_ID_ERC721 = 0x80ac58cd;
[0149]         bytes4 private constant _INTERFACE_ID_ERC165 = 0x01ffc9a7;
[0150]         uint256 private _totalSupply;
[0151]         uint256 private _maxTotalSupply;
[0152]         uint256 private _count;
[0153]         string private _name;
[0154]         string private _symbol;
[0155]         address private _issuer;
[0156]         mapping(address => mapping(address => bool)) private _operatorApprovals;
[0157]         mapping(uint256 => address) private _tokenApprovals;
[0158]         mapping(uint256 => address) private _owners;
[0159]         mapping(address => uint256) private _balances;
[0160]         mapping(uint256 => string) private _metadataURIs;
[0161]         mapping(uint256 => bool) private _frozen;
[0162]         mapping(uint256 => address) private _unfreezeAddresses;
[0163]         bool private _isFreeMintable;
[0164]         constructor(
[0165]             string memory name_,
[0166]             string memory symbol_,
[0167]             bool isFreeMintable_,
[0168]             uint256 maxTotalSupply_
[0169]         ) {
[0170]             _name = name_;
[0171]             _symbol = symbol_;
[0172]             _isFreeMintable = isFreeMintable_;
[0173]             _issuer = msg.sender;
[0174]             _maxTotalSupply = maxTotalSupply_;
[0175]         }
[0176]         function isLuxochainNFT() public view virtual override returns (bool) {
[0177]             return true;
[0178]         }
[0179]         function getStorageType() public view virtual override returns (uint256) {

```

```

[0180]     return I;
[0181] }
[0182] function owner() public view virtual override returns (address) {
[0183]     return _issuer;
[0184] }
[0185] function transferOwnership(address newIssuer) public virtual override {
[0186]     require(msg.sender == _issuer, "Not issuer");
[0187]     require(newIssuer != address(0), "New issuer is the zero address");
[0188]     _issuer = newIssuer;
[0189]     emit OwnershipTransferred(msg.sender, newIssuer);
[0190] }
[0191] function supportsInterface(bytes4 interfaceId)
[0192]     public
[0193]     view
[0194]     virtual
[0195]     returns (bool)
[0196] {
[0197]     return
[0198]         interfaceId == _INTERFACE_ID_ERC721_METADATA ||
[0199]         interfaceId == _INTERFACE_ID_ERC721 ||
[0200]         interfaceId == _INTERFACE_ID_ERC165;
[0201] }
[0202] function balanceOf(address __owner)
[0203]     public
[0204]     view
[0205]     virtual
[0206]     override
[0207]     returns (uint256)
[0208] {
[0209]     return _balances[__owner];
[0210] }
[0211] function ownerOf(uint256 tokenId)
[0212]     public
[0213]     view
[0214]     virtual
[0215]     override
[0216]     returns (address)
[0217] {

```

```

[0218]     return _owners[tokenId];
[0219] }
[0220] function name() public view virtual override returns (string memory) {
[0221]     return _name;
[0222] }
[0223] function symbol() public view virtual override returns (string memory) {
[0224]     return _symbol;
[0225] }
[0226] function totalSupply() external view override returns (uint256) {
[0227]     return _totalSupply;
[0228] }
[0229] function count() external view override returns (uint256) {
[0230]     return _count;
[0231] }
[0232] function tokenURI(uint256 tokenId)
[0233]     public
[0234]     view
[0235]     virtual
[0236]     override
[0237]     returns (string memory)
[0238]     {
[0239]         return _metadataURIs[tokenId];
[0240]     }
[0241] function transferFrom(
[0242]     address from,
[0243]     address to,
[0244]     uint256 tokenId
[0245] ) public virtual override {
[0246]     require(
[0247]         _isApprovedOrOwner(msg.sender, tokenId),
[0248]         "Transfer caller is not owner nor approved"
[0249]     );
[0250]     _transfer(from, to, tokenId);
[0251] }
[0252] function safeTransferFrom(
[0253]     address from,
[0254]     address to,
[0255]     uint256 tokenId

```

```

[0256] ) public virtual override {
[0257]     safeTransferFrom(from, to, tokenId, "");
[0258] }
[0259] function safeTransferFrom(
[0260]     address from,
[0261]     address to,
[0262]     uint256 tokenId,
[0263]     bytes memory _data
[0264] ) public virtual override {
[0265]     _safeTransfer(from, to, tokenId, _data);
[0266] }
[0267] function _safeTransfer(
[0268]     address from,
[0269]     address to,
[0270]     uint256 tokenId,
[0271]     bytes memory _data
[0272] ) internal virtual {
[0273]     _transfer(from, to, tokenId);
[0274]     require(
[0275]         _checkOnERC721Received(from, to, tokenId, _data),
[0276]         "Transfer to non ERC721Receiver implementer"
[0277]     );
[0278] }
[0279] function _transfer(
[0280]     address from,
[0281]     address to,
[0282]     uint256 tokenId
[0283] ) internal virtual {
[0284]     require(to != address(0), "Transfer to the zero address");
[0285]     require(!_frozen[tokenId], "Token is frozen");
[0286]     require(
[0287]         _isApprovedOrOwner(msg.sender, tokenId),
[0288]         "Sender cannot transfer token"
[0289]     );
[0290]     _balances[from] -= 1;
[0291]     _balances[to] += 1;
[0292]     _owners[tokenId] = to;
[0293]     _approve(address(0), tokenId);

```

```

[0294]     emit Transfer(from, to, tokenId);
[0295] }
[0296] function exists(uint256 tokenId)
[0297] public
[0298] view
[0299] virtual
[0300] override
[0301] returns (bool)
[0302] {
[0303]     return _owners[tokenId] != address(0);
[0304] }
[0305] function _safeMint(
[0306]     uint256 tokenId,
[0307]     address to,
[0308]     string calldata tokenMetadataURI
[0309] ) internal virtual {
[0310]     _owners[tokenId] = to;
[0311]     _metadataURIs[tokenId] = tokenMetadataURI;
[0312]     emit Transfer(address(0), to, tokenId);
[0313] }
[0314] function safeMint(
[0315]     uint256 tokenId,
[0316]     address to,
[0317]     string calldata tokenMetadataURI
[0318] ) external override {
[0319]     require(
[0320]         msg.sender == _issuer || _isFreeMintable,
[0321]         "Not Issuer nor free mint"
[0322]     );
[0323]     require(to != address(0), "Mint to the zero address");
[0324]     require(!exists(tokenId), "Token already minted");
[0325]     _totalSupply += 1;
[0326]     require(_maxTotalSupply >= _totalSupply, "Max total supply reached");
[0327]     _count += 1;
[0328]     _balances[to] += 1;
[0329]     _safeMint(tokenId, to, tokenMetadataURI);
[0330] }
[0331] function multipleSafeMint(

```

```

[0332]   uint256[] calldata tokensIds,
[0333]   address to,
[0334]   string[] calldata tokenMetadataURIs
[0335] ) external override {
[0336]   require(
[0337]     msg.sender == _issuer || _isFreeMintable,
[0338]     "Not Issuer nor free mint"
[0339]   );
[0340]   require(
[0341]     tokensIds.length == tokenMetadataURIs.length,
[0342]     "Different number of tokens and metadata provided"
[0343]   );
[0344]   require(to != address(0), "Mint to the zero address");
[0345]   require(
[0346]     _maxTotalSupply >= _totalSupply + tokensIds.length,
[0347]     "Max total supply reached"
[0348]   );
[0349]   for (uint256 i = 0; i < tokensIds.length; i++) {
[0350]     uint256 tokenId = tokensIds[i];
[0351]     require(!exists(tokenId), "Token already minted");
[0352]     _safeMint(tokenId, to, tokenMetadataURIs[i]);
[0353]   }
[0354]   _count += tokensIds.length;
[0355]   _totalSupply += tokensIds.length;
[0356]   _balances[to] += tokensIds.length;
[0357] }
[0358] function freezeToken(uint256 tokenId, address unfreezeAddress)
[0359]   external
[0360]   override
[0361]   {
[0362]     require(
[0363]       _isApprovedOrOwner(msg.sender, tokenId),
[0364]       "Cannot freeze token: permission denied"
[0365]     );
[0366]     require(unfreezeAddress != address(0), "Invalid unfreeze address");
[0367]     _frozen[tokenId] = true;
[0368]     _unfreezeAddresses[tokenId] = unfreezeAddress;
[0369]     emit TokenFrozen(tokenId, unfreezeAddress);

```

```

[0370]     }
[0371]     function unfreezeToken(uint256 tokenId, address newOwner)
[0372]         external
[0373]         override
[0374]         {
[0375]             require(
[0376]                 _unfreezeAddresses[tokenId] == msg.sender,
[0377]                 "You're not allowed to unfreeze this token"
[0378]             );
[0379]             require(newOwner != address(0), "Invalid new owner address");
[0380]             delete _frozen[tokenId];
[0381]             delete _unfreezeAddresses[tokenId];
[0382]             _owners[tokenId] = newOwner;
[0383]             emit TokenUnfrozen(tokenId, newOwner);
[0384]         }
[0385]     function isTokenFrozen(uint256 tokenId)
[0386]         public
[0387]         view
[0388]         virtual
[0389]         override
[0390]         returns (bool)
[0391]         {
[0392]             return _frozen[tokenId];
[0393]         }
[0394]     function burn(uint256 tokenId) external override {
[0395]         require(!_frozen[tokenId], "Token is frozen");
[0396]         address __owner = ownerOf(tokenId);
[0397]         require(_isApprovedOrOwner(msg.sender, tokenId), "Permission denied");
[0398]         _totalSupply -= 1;
[0399]         _balances[__owner] -= 1;
[0400]         delete _owners[tokenId];
[0401]         _approve(address(0), tokenId);
[0402]         emit Transfer(__owner, address(0), tokenId);
[0403]     }
[0404]     function isContract(address account) internal view returns (bool) {
[0405]         return account.code.length > 0;
[0406]     }
[0407]     function _checkOnERC721Received(

```

```

[0408]     address from,
[0409]     address to,
[0410]     uint256 tokenId,
[0411]     bytes memory _data
[0412] ) private returns (bool) {
[0413]     if (!isContract(to)) return false;
[0414]     try
[0415]         IERC721Receiver(to).onERC721Received(
[0416]             msg.sender,
[0417]             from,
[0418]             tokenId,
[0419]             _data
[0420]         )
[0421]     returns (bytes4 retval) {
[0422]         return retval == IERC721Receiver.onERC721Received.selector;
[0423]     } catch (bytes memory reason) {
[0424]         if (reason.length == 0) {
[0425]             revert("Transfer to non ERC721Receiver implementer");
[0426]         } else {
[0427]             assembly {
[0428]                 revert(add(32, reason), mload(reason))
[0429]             }
[0430]         }
[0431]     }
[0432] }

[0433] function _isApprovedOrOwner(address spender, uint256 tokenId)
[0434]     internal
[0435]     view
[0436]     virtual
[0437]     returns (bool)
[0438] {
[0439]     require(exists(tokenId), "Operator query for nonexistent token");
[0440]     address __owner = ownerOf(tokenId);
[0441]     return (spender == __owner ||
[0442]         getApproved(tokenId) == spender ||
[0443]         isApprovedForAll(__owner, spender));
[0444] }
[0445] function _setApprovalForAll(

```

```

[0446]     address __owner,
[0447]     address operator,
[0448]     bool approved
[0449] ) internal virtual {
[0450]     require(__owner != operator, "Approve to caller");
[0451]     _operatorApprovals[__owner][operator] = approved;
[0452]     emit ApprovalForAll(__owner, operator, approved);
[0453] }
[0454] function _approve(address to, uint256 tokenId) internal virtual {
[0455]     _tokenApprovals[tokenId] = to;
[0456]     emit Approval(ownerOf(tokenId), to, tokenId);
[0457] }
[0458] function approve(address to, uint256 tokenId) public virtual override {
[0459]     address __owner = ownerOf(tokenId);
[0460]     require(to != __owner, "Approval to current owner");
[0461]     require(
[0462]         msg.sender == __owner || isApprovedForAll(__owner, msg.sender),
[0463]         "Approve caller is not owner nor approved for all"
[0464]     );
[0465]     _approve(to, tokenId);
[0466] }
[0467] function getApproved(uint256 tokenId)
[0468]     public
[0469]     view
[0470]     virtual
[0471]     override
[0472]     returns (address)
[0473] {
[0474]     require(exists(tokenId), "Approved query for nonexistent token");
[0475]     return _tokenApprovals[tokenId];
[0476] }
[0477] function setApprovalForAll(address operator, bool approved)
[0478]     public
[0479]     virtual
[0480]     override
[0481] {
[0482]     _setApprovalForAll(msg.sender, operator, approved);
[0483] }

```

[0484] *function isApprovedForAll(address __owner, address operator)*

[0485] *public*

[0486] *view*

[0487] *virtual*

[0488] *override*

[0489] *returns (bool)*

[0490] *{*

[0491] *return _operatorApprovals[__owner][operator];*

[0492] *}*

[0493] *}*

[0494] L'invention porte encore sur un support d'enregistrement de données, lisible par un ordinateur, sur lequel est enregistré un programme d'ordinateur comprenant des instructions de code de programme de mise en œuvre du procédé décrit précédemment.

[0495] L'invention porte encore sur un signal d'un support de données, portant le produit programme d'ordinateur décrit précédemment.

Revendications

- [Revendication 1] Procédé de gestion de droits associés à un objet (1), tel qu'une pièce d'horlogerie (1) ou de joaillerie, le procédé comprenant les étapes suivantes :
- création d'un premier jeton numérique associé à un premier droit, notamment au droit de propriété, sur l'objet, le premier jeton numérique étant stocké et authentifié grâce à un premier protocole de chaîne de blocs,
 - création d'un deuxième jeton numérique, notamment un jeton numérique non fongible, associé à un deuxième droit, notamment un droit d'accès à un contenu numérique, en particulier un contenu multimédia, associé à l'objet, le deuxième jeton numérique étant stocké et authentifié grâce à un deuxième protocole de chaîne de blocs,
 - création d'un contrat intelligent liant les droits sur les premier et deuxième jetons numériques de sorte que le droit de propriété sur l'un des premier et deuxième jetons numériques ne peut pas être transféré sans le droit de propriété sur l'autre des premier et deuxième jetons numériques.
- [Revendication 2] Procédé de gestion selon la revendication précédente, caractérisé en ce que le premier protocole de chaîne de blocs et le deuxième protocole de chaîne de blocs sont d'un même type ou le premier protocole de chaîne de blocs et le deuxième protocole de chaîne de blocs sont un même protocole de chaîne de blocs.
- [Revendication 3] Procédé de gestion selon la revendication 1 ou 2, caractérisé en ce que le contrat intelligent implique la création, grâce à un protocole de chaîne de bloc, d'un troisième jeton numérique incluant des informations comprenant :
- au moins une information contenue dans le premier jeton numérique, tel que notamment une information de l'objet et/ou un numéro de série de l'objet et/ou un identifiant de la transaction de création du premier jeton numérique et/ou la date d'activation du certificat d'authenticité et/ou la date de la vente du produit et/ou un identifiant du premier jeton numérique, et
 - au moins une information contenue dans le deuxième jeton numérique, tel que notamment un identifiant du deuxième jeton numérique.
- [Revendication 4] Procédé de gestion selon l'une des revendications précédentes, caractérisé en ce que le contrat intelligent est configuré de sorte que :

- le transfert du droit de propriété sur le premier jeton numérique implique le transfert du droit de propriété sur le deuxième jeton numérique et/ou le transfert du droit de propriété sur le troisième jeton numérique, et/ou
- le transfert du droit de propriété sur le deuxième jeton numérique implique le transfert du droit de propriété sur le premier jeton numérique et/ou le transfert du droit de propriété sur le troisième jeton numérique, et/ou
- le transfert du droit de propriété sur le troisième jeton numérique implique le transfert du droit de propriété sur le premier jeton numérique et/ou le transfert du droit de propriété sur le deuxième jeton numérique.

[Revendication 5] Procédé de gestion selon l'une des revendications précédentes, caractérisé en ce que le premier jeton numérique et/ou le deuxième jeton numérique sont associés à des portefeuilles numériques de crypto-actifs, notamment à un même portefeuille numérique de crypto-actifs.

[Revendication 6] Procédé de gestion selon la revendication précédente, caractérisé en ce que le portefeuille numérique de crypto-actifs ou l'un des portefeuilles numériques de crypto-actifs est associé à :

- une clé publique, ou
- un code (4) associé à la clé publique,

la clé publique ou le code étant inscrit sur :

- l'objet, notamment sur un composant (2) d'une pièce d'horlogerie (1), en particulier sur un barillet (2) ou sur un rochet (3) de barillet (2) d'une pièce d'horlogerie (1), ou
- sur un support matériel vendu avec l'objet, ou
- dans un document numérique, comme un fichier informatique ou un message, notamment un MMS ou un SMS.

[Revendication 7] Procédé de gestion selon la revendication précédente, caractérisé en ce que la clé publique ou le code associé à la clé publique est inscrit par impression ou par gravure ou par collage d'une étiquette.

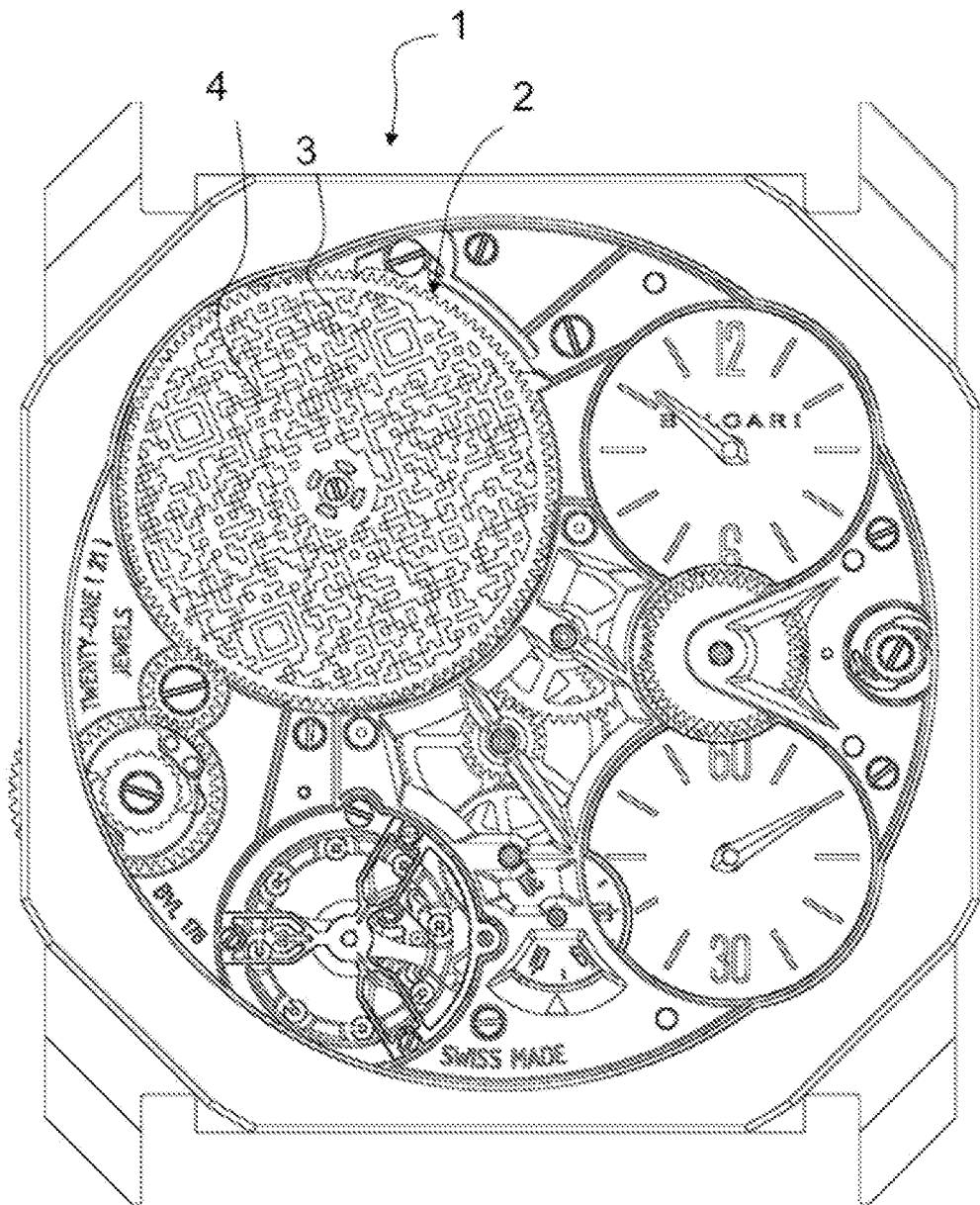
[Revendication 8] Procédé de gestion selon la revendication 6 ou 7, caractérisé en ce que le code associé à la clé publique est un code lisible par une machine, comme une étiquette NFC ou comme un code-barres, en particulier comme un code matriciel ou un code QR.

[Revendication 9] Procédé de gestion selon l'une des revendications précédentes, caractérisé en ce que le portefeuille numérique de crypto-actifs ou l'un des portefeuilles numériques de crypto-actifs est associé à :

- une clé privée, ou
 - un code associé à la clé privée,
- la clé privée ou le code associé à la clé privée étant inscrit :
- sur un support matériel annexé à l'objet, notamment sur une carte, sur une étiquette ou sur un document physique, et/ou
 - dans un document numérique, comme un fichier informatique ou un message, notamment un MMS ou un SMS.

- [Revendication 10] Procédé de gestion selon la revendication 9, caractérisé en ce que le code associé à la clé privée est un code lisible par une machine, comme un code-barres, en particulier comme une étiquette NFC ou comme un code matriciel ou un code QR.
- [Revendication 11] Dispositif, notamment architecture informatique distribuée, permettant de gérer des droits associés à un objet (1), le dispositif comprenant des éléments matériels et/ou logiciels mettant en œuvre le procédé selon l'une des revendications 1 à 10, notamment des éléments matériels et/ou logiciels conçus pour mettre en œuvre le procédé selon l'une des revendications 1 à 10.
- [Revendication 12] Produit programme d'ordinateur comprenant des instructions de code de programme enregistrées sur un support lisible par ordinateur pour mettre en œuvre les étapes du procédé selon l'une quelconque des revendications 1 à 10 lorsque ledit programme fonctionne sur un ordinateur.
- [Revendication 13] Support d'enregistrement de données, lisible par un ordinateur, sur lequel est enregistré un programme d'ordinateur comprenant des instructions de code de programme de mise en œuvre du procédé selon l'une des revendications 1 à 10.

[Fig. 1]



RAPPORT DE RECHERCHE

articles L.612-14, L.612-53 à 69 du code de la propriété intellectuelle

OBJET DU RAPPORT DE RECHERCHE

L'I.N.P.I. annexe à chaque brevet un "RAPPORT DE RECHERCHE" citant les éléments de l'état de la technique qui peuvent être pris en considération pour apprécier la brevetabilité de l'invention, au sens des articles L. 611-11 (nouveau) et L. 611-14 (activité inventive) du code de la propriété intellectuelle. Ce rapport porte sur les revendications du brevet qui définissent l'objet de l'invention et délimitent l'étendue de la protection.

Après délivrance, l'I.N.P.I. peut, à la requête de toute personne intéressée, formuler un "AVIS DOCUMENTAIRE" sur la base des documents cités dans ce rapport de recherche et de tout autre document que le requérant souhaite voir prendre en considération.

CONDITIONS D'ETABLISSEMENT DU PRESENT RAPPORT DE RECHERCHE

☒ Le demandeur a présenté des observations en réponse au rapport de recherche préliminaire.

☐ Le demandeur a maintenu les revendications.

☒ Le demandeur a modifié les revendications.

☐ Le demandeur a modifié la description pour en éliminer les éléments qui n'étaient plus en concordance avec les nouvelles revendications.

☐ Les tiers ont présenté des observations après publication du rapport de recherche préliminaire.

☐ Un rapport de recherche préliminaire complémentaire a été établi.

DOCUMENTS CITES DANS LE PRESENT RAPPORT DE RECHERCHE

La répartition des documents entre les rubriques 1, 2 et 3 tient compte, le cas échéant, des revendications déposées en dernier lieu et/ou des observations présentées.

☐ Les documents énumérés à la rubrique 1 ci-après sont susceptibles d'être pris en considération pour apprécier la brevetabilité de l'invention.

☒ Les documents énumérés à la rubrique 2 ci-après illustrent l'arrière-plan technologique général.

☐ Les documents énumérés à la rubrique 3 ci-après ont été cités en cours de procédure, mais leur pertinence dépend de la validité des priorités revendiquées.

☐ Aucun document n'a été cité en cours de procédure.

1. ELEMENTS DE L'ETAT DE LA TECHNIQUE SUSCEPTIBLES D'ETRE PRIS EN CONSIDERATION POUR APPRECIER LA BREVETABILITE DE L'INVENTION

NEANT

2. ELEMENTS DE L'ETAT DE LA TECHNIQUE ILLUSTRANT L'ARRIERE-PLAN TECHNOLOGIQUE GENERAL

WO 2020/092900 A2 (VERONA HOLDINGS SEZC;
YANTIS JONATHAN [US] ET AL.)
7 mai 2020 (2020-05-07)

US 2021/103938 A1 (BULAWSKI LUKASZ JACEK
[CH] ET AL) 8 avril 2021 (2021-04-08)

US 11 144 978 B1 (STOYANOV STOYAN
NIKODIMOV [US])
12 octobre 2021 (2021-10-12)

US 2019/366475 A1 (SCARSELLI BRUNO [US] ET
AL) 5 décembre 2019 (2019-12-05)

WO 2019/081667 A1 (COINPLUS SA [LU])
2 mai 2019 (2019-05-02)

3. ELEMENTS DE L'ETAT DE LA TECHNIQUE DONT LA PERTINENCE DEPEND DE LA VALIDITE DES PRIORITES

NEANT