Livre blanc technique

Version 1.0.1 - 28 mai 2021

Introduction

Veritise™ offre des services de vérification, d'identification, et de collecte et d'analyse de données aux entreprises comme aux particuliers. C'est grâce à la blockchain de Veritise™, véritable pierre angulaire du système, que ces solutions peuvent être proposées à l'échelle internationale.

La blockchain de Veritise™ est une technologie de pointe de prochaine génération, fondée sur l'architecture Symbol de NEM et dotée d'un réseau optimisé pour le traitement rapide, sécurisé et entièrement vérifiable des paiements et transactions. La richesse fonctionnelle de la blockchain de Veritise ouvre la voie à de puissantes capacités, comme les comptes à niveaux et signatures multiples, les actifs intelligents, les espaces de nom, les transactions agrégées, les contrôles de métadonnées et le stockage des données sur chaîne grâce à de robustes méthodes de chiffrement.

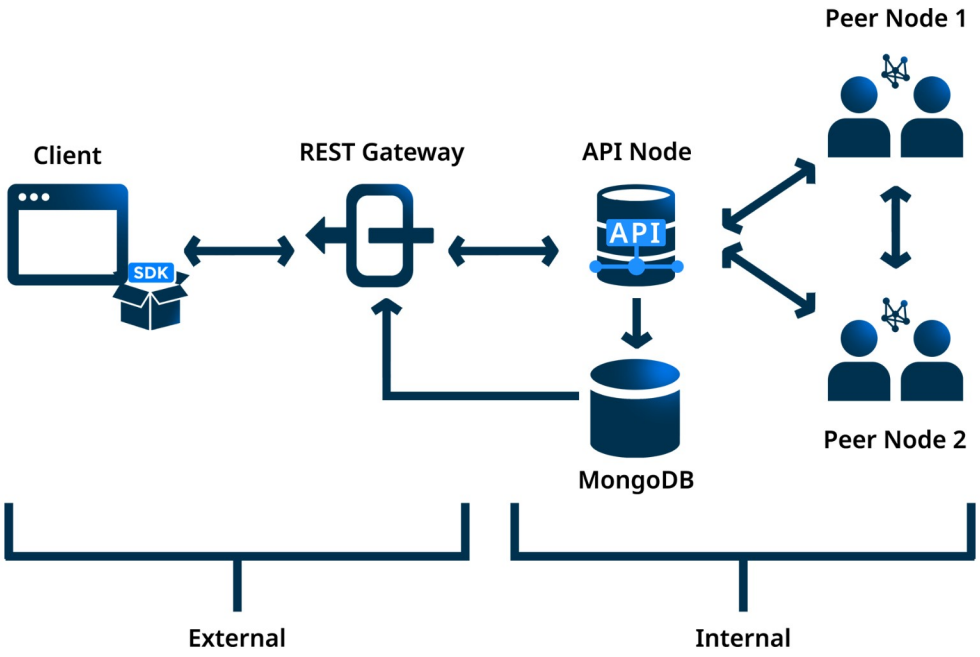
Les futurs cas d'utilisation, comme la tokenisation entièrement conforme des titres et autres instruments financiers, peuvent ainsi devenir réalité, pour plus de rapidité, une fraude réduite et des coûts minimisés.

L'architecture de la Blockchain de Veritise™ permet une customisation avancée tant au niveau du réseau que des nœuds individuels. Les paramètres à l'échelle d'un réseau, spécifiés sur celui-ci, doivent être les mêmes pour tous les nœuds qui le composent. À l'inverse, les paramètres propres à chaque nœud, situés à leur niveau, peuvent varier sur l'ensemble des nœuds d'un même réseau. Plutôt que de supporter des contrats intelligents Turing-complets, une démarche basée sur les plug-ins/extensions a été adoptée. En effet, si cette première approche apporte une plus grande flexibilité aux utilisateurs, elle est également plus sujette à erreurs. Notre modèle fondé sur les plug-ins limite les opérations possibles sur une blockchain et réduit de fait sa surface d'attaque.

Il est en outre largement plus simple d'optimiser la performance d'un ensemble discret d'opérations que celle d'un ensemble infini. Autant de caractéristiques qui permettent à la blockchain de Veritise™ d'atteindre le haut débit pour lequel elle a été conçue.

Topologie du réseau Veritise™

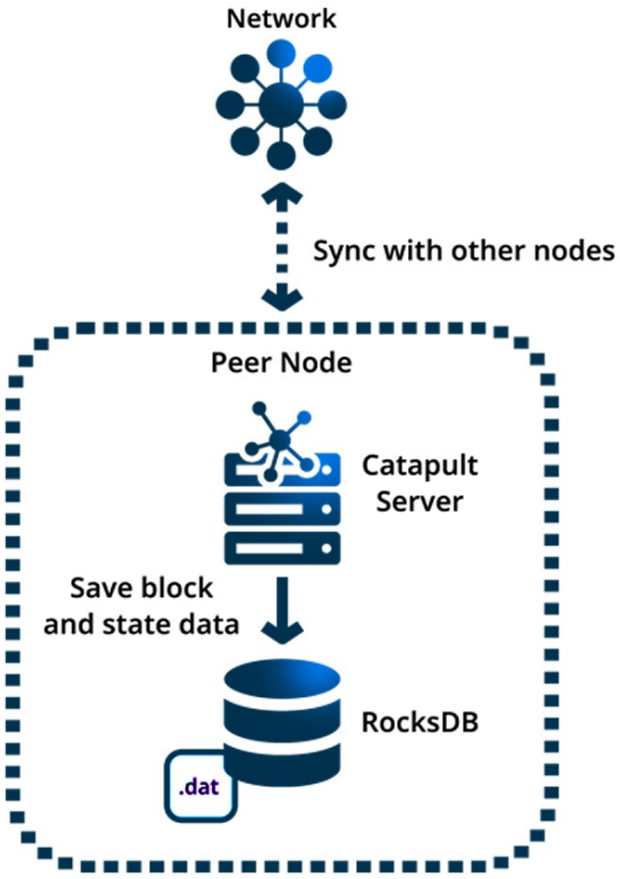
La blockchain de Veritise™ se fonde sur une infrastructure de **blockchain** publique dotée de nœuds de réseaux de trois catégories différentes : pairs, API ou doubles. La communication avec le réseau est assurée par une composante externe, la passerelle REST.



Infrastructure de la blockchain de Veritise™

Nœud pair :

Les nœuds pairs forment la base de la blockchain de Veritise™. Leur rôle consiste à vérifier les transactions et les blocs, exécuter l'algorithme de consensus, créer de nouveaux blocs et propager les changements sur l'ensemble du réseau.



Communication par nœud pair

Les nœuds API poussent les nouvelles transactions vers le réseau pair-à-pair, où elles sont soit incluses à un bloc, soient écartées. À l'issue du traitement du bloc, le nœud enregistre :

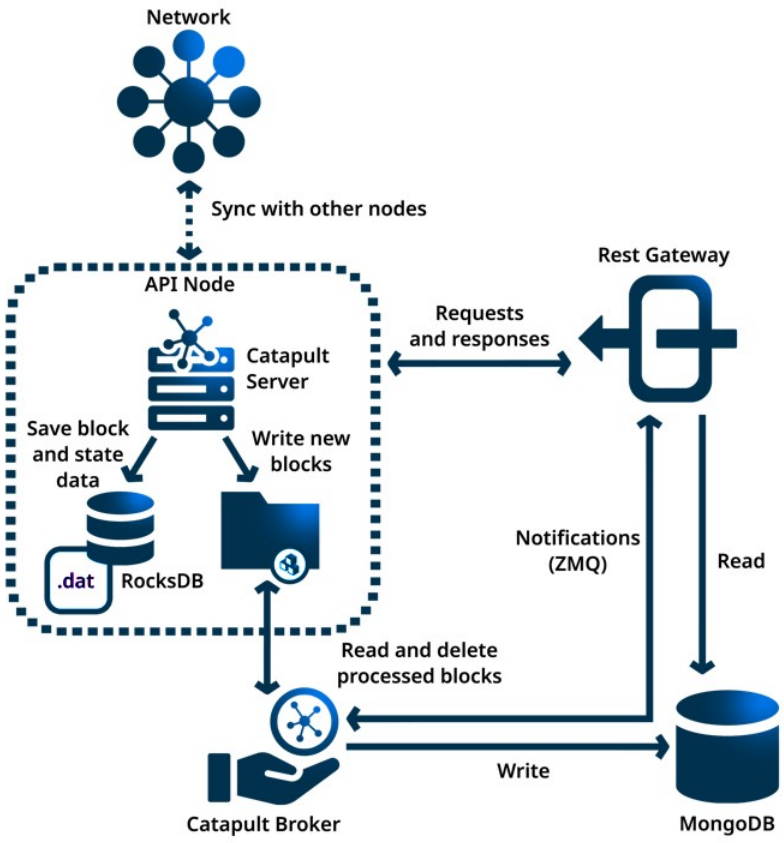
- Le binaire de chaque bloc sur disque en tant que fichier plat
- Le nouvel état de la chaîne en mémoire ou dans RocksDB (configurable)

Les nœuds pairs stockent l'état de la chaîne dans RocksDB. Les structures de données mises en cache sont sérialisées et conservées en tant que valeurs pour les clés

correspondantes. Par exemple, une colonne dans cette base de données relie les clés publiques aux adresses. Une autre les entrées d'état de compte en tant que valeurs pour les clés d'adresse correspondantes.

Nœud API :

La fonction première d'un nœud API est de stocker les données dans un format lisible dans MongoDB. Le logiciel du serveur Catapult permet de paramétrer des nœuds API autonomes ou avec capacités de pair (nœuds doubles).



Communication par nœud pair + API (double)

Plutôt que d'écrire les données directement dans MongoDB, les nœuds les ajoutent à une file d'attente provisoire sur fichier appelée « spool ». Un agent gestionnaire consomme les données du spool et met MongoDB à jour en conséquence. Une fois un bloc traité, l'agent envoie les changements aux instances Catapult-REST via ZMQ.

Les nœuds API ont également pour charge de recueillir les cosignatures des transactions agrégées cautionnées, qui ne sont traitées qu'une fois complètes. MongoDB stocke les blocs, les transactions et les états de la chaîne pour des requêtes haute performance.

L'agent gestionnaire met l'instance MongoDB liée à jour quand :

- Le traitement d'un nouveau bloc/d'un ensemble de blocs est terminé
- Le traitement de nouvelles transactions non confirmées est terminé

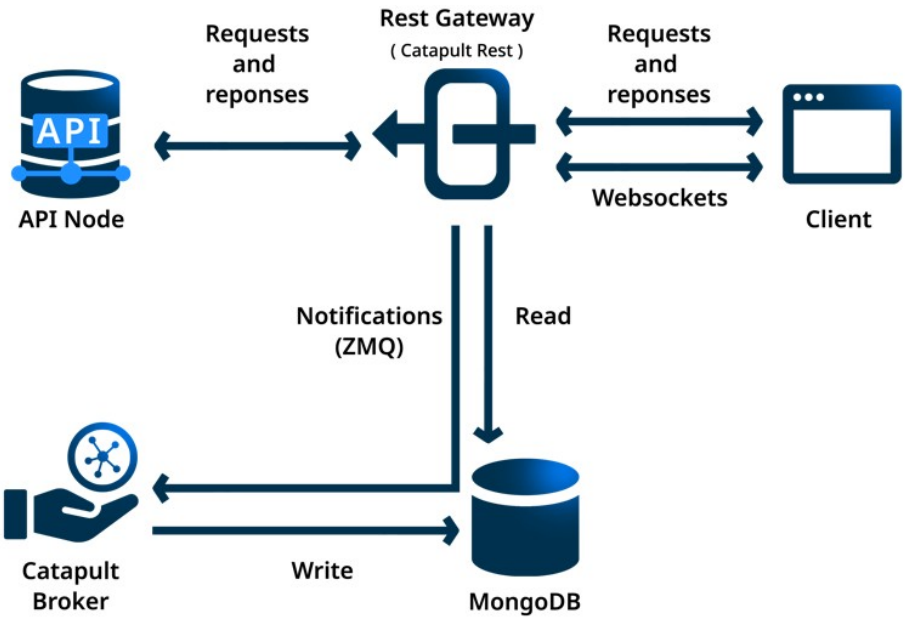
ZeroMQ (ZMQ) est une bibliothèque de messagerie asynchrone haute performance permettant les souscriptions en temps réel. Elle transporte les notifications du nœud API jusqu'au point d'extrémité ZMQ, où Catapult REST les réceptionne. C'est une alternative aux WebSockets REST, conçue pour les usages où la performance est essentielle.

Nœud double :

Les nœuds doubles sont tout bonnement un surensemble des nœuds pairs et API. Ils supportent l'ensemble des capacités des deux types de nœuds. Comme ils présentent toutes les capacités des nœuds API, ils nécessitent aussi un agent.

Passerelle REST :

Les passerelles REST traitent les requêtes du client API JSON. Elles lisent les données de MongoDB, formatent les réponses, et les renvoient au client. Cette composante a également pour fonction de renvoyer des évènements au client via les WebSockets.



Communication par passerelle REST

Chaque passerelle REST est connectée à une instance API pour envoyer les nouvelles requêtes de transaction déclenchées côté client, et recevoir les mises à jour en temps réel au moyen des sockets.

Nœuds d'amorçage

À son lancement, tout nouveau nœud est isolé, faute de connexions à d'autres pairs. Avant de pouvoir véritablement contribuer, par exemple valider ou produire des blocs, il doit rejoindre un réseau. Sur le réseau Veritise™, une liste de nœuds d'amorçage statiques est conservée dans un fichier de configuration des pairs.

Pour rejoindre un réseau, tout nouveau nœud doit d'abord se connecter à ceux-ci. Ces fichiers ne sont pas nécessairement identiques pour l'ensemble des nœuds d'un réseau.

Communication entre nœuds

L'abréviation TLS, pour *Transport Layer Security* soit « sécurité de la couche de transport », désigne un protocole sécurisé de communication entre serveurs. Toutes les connexions entre les nœuds Veritise™ sont effectuées conformément à sa dernière itération (TLS v1.3), selon une procédure de vérification sur mesure garantissant un niveau de confidentialité et de performance inégalé par rapport à ses précédentes versions.

Chaque nœud doit être certifié X509 au niveau de la racine et du nœud, au moyen d'une chaîne de deux certificats. Tous les certificats doivent être de type X25519. Veritise™ n'en accepte aucun autre.

Le certificat racine doit être autosigné au moyen de la clé privée de signature d'un compte. Ce compte est supposé correspondre au propriétaire unique d'un nœud. Pour des raisons de sécurité, il est fortement déconseillé de conserver cette clé privée sur un serveur en fonctionnement. Le certificat du nœud est signé par le certificat racine. Il peut contenir une paire de clés publique/privé aléatoire. Ce certificat sert à authentifier les sessions TLS et à dériver des clés de chiffrement partagées pour du chiffrement optionnel. Il peut faire l'objet de permutations aussi fréquentes que souhaitées. Cette procédure d'authentification est appliquée indépendamment par chaque nœud partenaire. Si le protocole échoue à l'un ou l'autre des nœuds, la connexion est immédiatement rompue.

Pour la communication réseau, Veritise™ utilise le protocole de contrôle de transmissions TCP au port spécifié dans les paramètres du nœud. Cette communication s'articule autour d'un modèle de paquet à plus haut niveau, qui s'ajoute aux paquets TCP. Chaque paquet commence par un en-tête de 8 octets spécifiant sa taille et son type. Une fois entièrement reçus, les paquets peuvent être traités.

Des connexions de courtes comme de longues durées sont utilisées. Ces dernières sont dédiées aux activités répétitives comme la synchronisation des blocs ou des transactions. Elles supportent tant les sémantiques *push* que de requêtes et réponses.

Le réseau est paramétré de façon à ce que les connexions puissent durer pendant 200 tours de sélection de blocs avant d'être recyclables. Les connexions qui dépassent ce seuil sont recyclées principalement dans le but de permettre des interactions directes avec d'autres nœuds partenaires, et dans une moindre mesure pour prévenir les connexions zombies. Les connexions de courte durée servent quant à elles aux interactions entre les nœuds plus complexes à plusieurs étapes, comme la détection des nœuds ou la synchronisation temporelle. Elles contribuent à éviter que les ressources de synchronisation viennent à manquer, ce qui peut arriver lorsque toutes les connexions de longue durée sont ouvertes et qu'aucun partenaire de synchronisation n'est disponible.

Cryptographie

Veritise™ repose sur les principes de la cryptographie sur les courbes elliptiques (*elliptic curve cryptography* ou ECC). Pour la sécurité et la rapidité, le choix de la courbe sous-jacente est crucial. Veritise™ a opté pour l'algorithme de signature numérique Ed25519, qui engendre de courtes signatures de 64 octets et permet la vérification rapide des signatures. Le traitement des blocs n'implique ni génération de clés ni signature, la vitesse de ces opérations n'a donc aucune importance.

Paire de clés publique/privée

La clé privée est un nombre entier aléatoire de 256 bits. C'est également le cas de la clé publique, laquelle est dérivée de la clé privée. La paire de clés correspond à un compte de la blockchain de Veritise™ associé à un état mutable stocké sur la blockchain.

Comptes et adresses

Les comptes (paires de clés publiques/privées) du réseau Veritise™ servent d'identités numériques permettant d'identifier les utilisateurs, entreprises, produits et objets uniques, de les suivre et de les tracer, de conserver et de maintenir des registres de provenance, et de garantir l'entière traçabilité des évènements qui leur sont attachés.

Chaque compte dispose d'une adresse unique dérivée de la clé publique. Typiquement, c'est cette adresse qui est partagée plutôt que la clé publique, car elle est plus courte et recueille des informations au sujet du réseau.

Déchiffrée, une adresse est une valeur de 24 octets composée des trois éléments suivants :

- Octet de réseau
- Empreinte de 160 bits de la clé de signature publique d'un compte
- Somme de contrôle de 3 octets

La somme de contrôle permet de détecter rapidement les adresses erronées. Des jetons peuvent être envoyés à toute adresse valide, même celles qui n'ont aucune transaction à leur actif. Si la clé privée du compte sur lequel les mosaïques sont envoyées n'appartient à personne, elles seront en toute probabilité perdues à jamais.

Les adresses peuvent être représentées soit au format hexadécimal (48 caractères), soit, plus habituellement, au format chiffré base32 (39 caractères).

Pour convertir une clé publique en adresse, il convient de procéder aux étapes suivantes :

1. Exécuter SHA3-256 sur la clé publique
2. Exécuter RIPEMD160 sur l'empreinte obtenue à l'étape 1
3. Exécuter l'octet de la version du réseau sur l'empreinte RIPEMD160
4. Exécuter SHA3-256 sur le résultat, prendre les trois premiers octets en tant que somme de contrôle
5. Concaténer le résultat de l'étape 3 et la somme de contrôle de l'étape 4
6. Encoder ce résultat avec Base32

Comptes multisignatures

Les comptes Veritise™ peuvent être convertis en comptes multisignatures. Les cosignataires (à savoir les autres comptes) en deviennent alors les gestionnaires.

À compter de ce moment-là, le compte multisignature ne peut pas annoncer de transactions individuellement. Un cosignataire doit proposer une transaction, et l'envelopper dans une **transaction agrégée**.

Pour qu'elle soit ajoutée au bloc, les autres cosignataires doivent donner leur accord.

Clés publiques

Un compte peut-être associé à zéro ou plusieurs clés publiques. Sont supportées :

1. ***Les clés de signature :*** ces clés publiques ED25519 sont utilisées pour signer et vérifier des données. Si n'importe quel compte peut recevoir des données, seuls ceux dotés d'une clé publique de ce type peuvent en envoyer. Cette clé publique est la seule utilisée pour le calcul de l'adresse d'un compte.
2. ***Les clés de lien :*** ces clés publiques relient un compte principal à un nœud de staking distant. Pour un compte principal, cette clé publique spécifie le compte du nœud de staking distant pouvant signer des blocs à sa place. Pour un compte distant, elle spécifie le compte principal au nom duquel il est autorisé à signer des blocs. Ces liens sont bidirectionnels et établis systématiquement par paires.
3. ***Les clés de nœud :*** ce type de clé publique est émis sur un compte principal. Il indique la clé publique du nœud sur lequel ce dernier peut déléguer le staking. Précision importante, ceci ne signifie pas que le distant stake activement sur le nœud, mais uniquement qu'il y est autorisé. Si le compte du nœud de staking ou le propriétaire du nœud est honnête, le compte sera limité pour que le staking ne puisse être délégué que sur un seul nœud à la fois. Un staker honnête ne devrait pour sa part envoyer sa clé privée de staking distant qu'à un seul nœud à la fois. Changer de distant invalidera l'ensemble des précédentes permissions de staking distant accordées à tous les autres nœuds (pour la confidentialité persistante des clés déléguées). Toute clé privée de staking distant antérieure deviendra alors invalide, et ne pourra plus être utilisée pour générer des blocs. Les propriétaires de nœuds honnêtes ne devraient effectuer de staking distant qu'au moyen d'une clé privée de staking distant actuellement liée à sa clé publique de nœud.
4. ***Les clés VRF :*** ces clés publiques ED25519 sont utilisées pour générer et vérifier des valeurs aléatoires. Pour qu'un compte soit éligible au staking, une clé publique VRF doit être établie sur le Compte principal. Une fonction aléatoire vérifiable (*Verifiable Random Function* ou VRF) utilise une paire de clés publique/privée pour générer

des valeurs pseudo-aléatoires. Seul le propriétaire de la clé publique peut générer une valeur de façon à ce qu'elle ne puisse pas être prédéterminée par un adversaire. Quiconque dispose de la clé publique peut vérifier si la valeur a bien été générée par la clé privée qui y est associée.

5. ***Les clés de vote :*** ces clés publiques distinctes sont utilisées pour signer et vérifier les messages de finalisation. Toutes les clés de vote sont temporaires et doivent être enregistrées accompagnées d'une époque de début et de fin. Pour qu'un compte soit autorisé à voter, une clé publique de ce type doit être établie sur le compte principal.
6. ***Les clés de transport :*** les paires de clés de ce type sont utilisées par les nœuds aux fins du transport sécurisé via TLS.

Transactions

Le réseau Veritise™ supporte deux types fondamentaux de transactions : les **transactions basiques** et **agrégées**.

Les **transactions basiques** représentent une opération unique, et nécessitent une seule signature. Elles se composent de données tant vérifiable que non vérifiable par chiffrement. Toutes les données vérifiables sont contigües et signées par le signataire de la transaction. Les données non vérifiables sont quant à elles soit ignorées (remplissage), soit calculables par déterminisme à partir de données vérifiables. Chaque transaction basique requiert la vérification d'une seule signature exactement.

Les **transactions agrégées** contiennent une ou plusieurs transactions pouvant nécessiter de multiples signatures. Elles permettent de combiner des transactions basiques pour former des opérations potentiellement complexes et les exécuter de façon atomique. Elles apportent une plus grande souplesse par rapport aux systèmes qui garantissent l'atomicité

des opérations individuelles uniquement, tout en limitant l'ensemble global des opérations permises à un périmètre fini.

Si la composition d'une transaction agrégée est plus complexe que celle d'une transaction basique, elles partagent quelques similarités. Une transaction agrégée présente par exemple le même en-tête non vérifiable qu'une transaction basique, lequel est traité de la même façon. Les transactions agrégées sont cependant dotées d'un pied de page de données non vérifiables suivies de transactions et cosignatures imbriquées.

Une transaction agrégée peut toujours être soumise au réseau avec toutes les cosignatures nécessaires. Dans ce cas, elle est dite « complète », et traitée comme toute autre transaction sans opération spécifique.

Les nœuds API peuvent également accepter les transactions agrégées cautionnées ne présentant pas toutes les cosignatures requises. Le soumissionnaire paie dans ce cas une caution, qui ne lui sera retournée que si et seulement si toutes les cosignatures nécessaires sont recueillies avant que la transaction n'expire. Sous réserve que cette caution soit versée en amont, un nœud API recueillera les cosignatures associées à la transaction en question jusqu'à ce qu'elle en compte le nombre nécessaire ou expire.

Messages de statut des transactions

Les sujets des messages de transaction sont constitués d'un marqueur de sujet et d'un filtre facultatif sur l'adresse non résolue. Lorsque ce dernier est fourni, seuls les messages impliquant l'adresse non résolue spécifiée seront remontés. Par exemple, un message sera généré pour une transaction de transfert uniquement si l'adresse non résolue indiquée correspond à son émetteur ou son destinataire. Si aucune adresse non résolue n'est précisée, les messages seront créés pour toutes les transactions.

Les messages suivants sont supportés :

- Transaction : une transaction a été confirmée, c'est à dire fait partie d'un bloc
- Ajout d'une transaction non confirmée : une transaction non confirmée a été ajoutée au cache des transactions non confirmées
- Suppression d'une transaction non confirmée : une transaction non confirmée a été supprimée du cache des transactions non confirmées
- Ajout d'une transaction partielle : une transaction partielle a été ajoutée au cache des transactions partielles
- Suppression d'une transaction partielle : une transaction partielle a été supprimée du cache des transactions partielles
- Statut de transaction : le statut d'une transaction a changé

Le sujet d'un message de cosignature est constitué d'un marqueur de sujet et d'un filtre facultatif sur l'adresse non résolue. Le message est envoyé aux clients souscripteurs lorsqu'une nouvelle cosignature est ajoutée au cache des transactions partielles dans le cadre d'une transaction agrégée. Lorsque le filtre sur l'adresse non résolue est fourni, les messages ne seront émis que pour les transactions agrégées impliquant l'adresse spécifiée. Sinon, des messages seront transmis pour l'ensemble des changements.

Validation des données

La blockchain de Veritise™ utilise des structures arborescentes pour stocker les grandes quantités de données associées à un bloc ne pouvant pas être extraites directement de son en-tête. Ceci permet aux clients légers de vérifier si un élément existe (tel qu'une transaction ou un avis de réception) sans demander l'ensemble de l'historique du registre.

Fonctionnalités de la blockchain de Veritise™

1. Systèmes de comptes à niveaux et signatures multiples

Comme nous l'avons déjà mentionné, les comptes de la blockchain de Veritise peuvent être convertis en comptes multisignatures fonctionnant à différents niveaux et profondeurs. Les cosignataires (à savoir les autres comptes) en deviennent alors les gestionnaires. À compter de ce moment-là, le compte multisignature ne peut pas annoncer de transactions individuellement. Un de ses cosignataires doit proposer une transaction, et l'envelopper dans une transaction agrégée. Pour qu'elle soit ajoutée au bloc, les autres cosignataires doivent donner leur accord.

La version actuelle des comptes multisignatures de Veritise™ repose sur un schéma « M-sur-N », où M est égal au nombre de cosignataires requis pour annoncer une transaction et N au total des cosignataires pour le compte en question. M peut ainsi correspondre à toute valeur égale ou inférieure à N (par ex. 1-sur-4, 2-sur-2, 4-sur-9, 9-sur-10, etc.).

Un compte multisignatures peut en outre être cosignataire d'un autre compte multisignatures, et ainsi de suite jusqu'à 4 niveaux. Ces comptes à plusieurs niveaux appliquent en sus une logique « ET/OU » aux transactions multisignatures, créant des processus et logiques d'implémentation multiniveaux, dans la mesure où un cosignataire peut correspondre à une application de processus métier donné, autrement dit vérifier certaines règles métier.

2. Jetons (mosaïques)

Outre les jetons VTS, pièces maîtresses de l'infrastructure de la blockchain de Veritise™, les utilisateurs/entreprises sont libres d'émettre des actifs intelligents sous

la forme de jetons, mais également de collections d'actifs plus spécialisés comme les points de fidélité, les signatures, les indicateurs de statut, les votes, les identifiants produit, voire d'autres monnaies.

Chaque actif dispose d'un identifiant unique représenté par un nombre entier non-signé de 64 bits et d'un ensemble de propriétés et de drapeaux paramétrables pouvant être définis à l'étape de création du jeton.

3. Espaces de nom

Les espaces de noms fonctionnent de façon similaire aux domaines Internet. Pour en créer un, la première étape consiste à choisir un nom que vous utiliserez pour désigner un compte ou un actif. Il doit être unique sur le réseau, et composé d'un maximum de 64 caractères parmi les suivants : {a, b, c, ..., z, 0, 1, 2, ..., 9, _ , -.}

Au moyen des transactions de type alias, Veritise™ pourra relier les espaces de nom aux comptes et actifs numériques (jetons). L'alias ou son actif correspondant peuvent être utilisés de façon interchangeable lors de l'envoi d'une transaction. Les alias rendent les adresses plus mémorables et les actifs (jetons) reconnaissables. Les reçus du bloc stockent la résolution de l'alias pour une transaction donnée.

La fonctionnalité d'espace de nom peut être utilisée aux fins d'attribuer des identifiants et/ou des étiquettes conviviales aux comptes Veritise™, de naviguer facilement dans le registre des transactions et d'initier des transactions au moyen de cette étiquette ou cet identifiant.

4. Restrictions sur les comptes

L'infrastructure de la blockchain de Veritise™ permet aux comptes de configurer un ensemble de règles intelligentes visant à bloquer l'annonce ou la réception de transactions en fonction d'une série de restrictions.

Le propriétaire d'un compte (ou les propriétaires dans le cas des comptes multisignatures) peut modifier les restrictions qui y sont attachées.

Il peut ainsi décider de recevoir uniquement les transactions émanant d'adresses autorisées. À l'inverse, il peut aussi établir une liste d'adresses bloquées.

Les restrictions sur les transactions entrantes sont utiles lorsque le compte n'est censé en recevoir que de certaines adresses connues, ou lorsque le compte souhaite bloquer toute transaction provenant d'expéditeurs inconnus.

Par défaut, lorsqu'aucune restriction n'est définie, tous les comptes du réseau peuvent annoncer des transactions au compte non restreint. Un compte peut également décider d'appliquer des restrictions sur les transactions sortantes, pour limiter les comptes destinataires autorisés.

5. Restrictions sur les opérations

Un compte peut être configuré de façon à autoriser/bloquer les annonces de transactions sortantes d'un type d'opération prédéfini. Il renforce ainsi sa sécurité en prévenant l'annonce par erreur de transactions non désirées.

6. Transactions agrégées

Les transactions agrégées fusionnent plusieurs transactions pour n'en former qu'une, permettant des échanges sans confiance (*trustless swaps*) et autres logiques avancées. Pour ce faire, la blockchain de Veritise™ génère un contrat intelligent jetable à usage unique.

Lorsque tous les comptes impliqués ont cosigné la transaction agrégée, toutes les transactions qu'elle contient sont exécutées au même moment.

Différents types de transactions agrégées sont prédéfinis sur Veritise™ :

- **Complète** - Une transaction agrégée est dite complète lorsque tous les participants requis l'ont signée. Les cosignataires peuvent s'y attacher sans utiliser la blockchain. Une fois toutes leurs signatures recueillies, l'un d'entre eux peut annoncer la transaction au réseau. Si sa configuration interne est valide et aucune erreur de validation n'est relevée, les transactions qui la composent seront exécutées en même temps. Les transactions agrégées complètes permettent d'ajouter davantage de transactions par bloc en regroupant en leur sein de multiples transactions.
- **Cautionnée** - une transaction agrégée est dite cautionnée lorsqu'elle nécessite la signature d'autres participants. Une fois cette transaction agrégée cautionnée annoncée, elle rentre dans un état partiel pour une durée allant jusqu'à 2 jours (paramétrable sur la blockchain de Veritise™) et notifie son statut via WebSockets ou appels API HTTP. Chaque fois que le cosignataire signe la transaction et annonce une cosignature agrégée cautionnée, le réseau vérifie si tous les autres cosignataires ont signé. Une fois toutes les signatures recueillies, la transaction devient non confirmée jusqu'à ce que le réseau l'intègre à un bloc.

  Les transactions agrégées sont utilisées par Veritise™ pour mettre en œuvre des logiques métier et l'automation des processus. Par exemple pour créer un compte, l'activer, et attribuer une étiquette/un identifiant en tant qu'alias d'espace de nom en une seule fois.

7. Contrôles de métadonnées

La blockchain de Veritise™ offre la possibilité d'associer des données customisées relatives à un compte, actif numérique (jeton) ou espace de nom à une transaction.

Veritise™ utilise les métadonnées suivantes :

- Attacher les informations pertinentes aux comptes (étiquetage de compte, catégorisation du compte, etc.).
- Valider la valeur attachée à un compte pour permettre à l'application d'arrière-guichet de la blockchain de Veritise™ d'effectuer une action hors chaîne (actions déclenchées, actions validées, etc.).

Les métadonnées sont identifiées de façon unique par l'uplet {signer, target-id, metadata-key}. L'inclusion d'un signataire à cet identifiant composite permet à différents comptes de spécifier les mêmes métadonnées sans créer de conflit.

La valeur liée à un identifiant est une chaîne allant jusqu'à 4096 caractères.

Les entrées de métadonnées sont stockées sur la blockchain de Veritise™, comme le message d'une transaction de transfert standard, mais également en tant qu'état clé-valeur. Cette fonctionnalité réduit le temps de lecture des applications client ; en effet, les métadonnées permettent d'accéder à l'information à l'aide de clés plutôt qu'en traitant l'ensemble de l'historique des transactions hors chaîne dans le but d'obtenir la valeur du dernier message de transaction.

8. Message

Sur le réseau de la blockchain de Veritise™, les transactions de transferts peuvent porter un message d'une longueur (paramétrable) allant jusqu'à 4096 caractères, permettant d'horodater définitivement les données sur la blockchain. Par défaut, les messages associés sont visibles pour tous les participants du réseau de la blockchain. Seuls les expéditeurs et destinataires peuvent en revanche accéder aux messages chiffrés. Le réseau de la Blockchain de Veritise™ utilise la fonction de chiffrement par bloc AES de Bouncy Castle en mode CBC pour chiffrer et déchiffrer les messages.

Consensus

Veritise™ a choisi la preuve d'enjeu (*Proof of Stake* ou PoS) comme méthode de consensus dans le but de récompenser l'utilisation plutôt que la thésaurisation. Elle vise à calculer l'importance d'un compte donné au moyen d'un score global, sans sacrifier la performance ou la mise à l'échelle.

L'algorithme prend en compte les facteurs suivants pour évaluer l'importance d'un compte, à savoir la mesure qui servira au final à sélectionner le prochain nœud de staking :

- **Enjeu :** le montant total des mosaïques de harvesting détenues, les propriétaires des soldes les plus importants ayant le plus intérêt à voir l'écosystème prospérer. Seuls les comptes de plus de 10 000 jetons VTS (grands comptes) sont éligibles au staking.
- **Transactions :** le montant total des frais payés par un compte. Ceci encourage l'activité sur le réseau.
- **Nœuds :** le nombre de fois où un compte a touché des frais collectés par un nœud. Le réseau favorise ainsi les comptes qui utilisent des nœuds.

Un score d'importance basé sur ces trois facteurs est périodiquement calculé pour tous les grands comptes. Celui-ci détermine la probabilité qu'un compte soit sélectionné pour produire le prochain bloc et soit rémunéré pour le consensus.

Scores partiels

Dans un premier temps, le réseau calcule les scores partiels suivants pour tous les grands comptes, à l'issue de chaque période d'importance (720 blocs, soit environ 3 heures) :

- **Score d'enjeu (S) :** le solde du compte divisé par le solde de tous les grands comptes à la fin de la période.
- **Score de transaction (T) :** le montant total des frais payés par le compte divisé par le montant total des frais payés par tous les grands comptes sur la période.
- **Score de nœud (N) :** Le nombre de fois où le compte a touché des frais d'un nœud divisé par le nombre de fois où tous les grands comptes ont touché des frais d'un nœud sur la période.
- **Score d'activité (A) :** la moyenne des scores T et N pondérés à 80 % et 20 % respectivement, divisée par le solde du compte. Cette opération bénéficie aux petits comptes, car leur score d'importance dépendra davantage de leur activité que de leur enjeu.

  Un score d'activité absolu (A') est d'abord calculé :

$$A' = \frac{10000}{Solde}(0.8\,T + 0.2\,N)$$

Puis le score d'activité réel (A) est calculé en divisant A' par la somme des scores d'activité absolus de tous les grands comptes.

Le score d'importance est ensuite calculé sur la base des scores partiels ci-dessus.

Score d'importance

Le score d'importance (I) correspond à la moyenne des scores S et A, pondérée par un facteur d'activité γ :

$$I = \gamma A + (1-\gamma) S$$

Sur le réseau de Veritise™, γ est égal à 0,05 (5 %).

Enfin, parmi tous les comptes éligibles au staking, la probabilité d'être sélectionné est proportionnelle au score d'importance effectif, défini comme le plus petit des deux scores d'importance précédents.

Caractère définitif du règlement

La technologie blockchain de Veritise™ intègre un mécanisme de finalisation des blocs déterministe au niveau du protocole permettant l'établissement de points de contrôle qu'aucune des parties ne peut supprimer. L'information du réseau est définitivement immuable par dessin.

Cette approche est inspirée de GRANDPA, mis en œuvre par Polkadot. GRANDPA est amplement influencé par le protocole CASPER, lui-même grandement nourri de PBFT. Traditionnellement, PBFT utilise trois types de messages : *pre-prepare* (pré-préparation), *prepare* (préparation) et *commit* (validation). Les messages de pré-préparation, utilisés pour lancer les tours, ne sont pas déployés par Veritise™, qui s'appuie à cette fin sur le temps écoulé. Les messages de préparation et de validation de PBFT correspondent grossièrement aux messages de pré-vote et de pré-validation sur le réseau Veritise™.

Frais de réseau

Par défaut les frais seront réglés dans la monnaie sous-jacente du réseau Veritise™, à savoir en jetons VTS. Tous les frais (création de jetons, location d'espaces de nom, transactions de transfert, etc.) sont configurables sur le réseau Veritise™.

Les frais associés à une transaction dépendent principalement de sa taille. Les frais effectifs déduits du compte expéditeur correspondent au produit de la taille de la transaction et d'un multiplicateur de frais fixé par le nœud qui produit le bloc.

$$\mathbf{effectiveFee = transaction::size * block::feeMultiplier}$$

Les propriétaires de nœuds sur Veritise™ seront libres de choisir n'importe quelle valeur positive en tant que multiplicateur, y compris zéro. Ce multiplicateur favorisera la concurrence entre les nœuds et permettra de maintenir une structure de frais en ligne avec l'économie de la blockchain et la demande de services. Le multiplicateur de frais (fee_multiplier) est stocké dans l'en-tête du bloc lorsque le nœud en produit un nouveau, déterminant les frais effectifs payés pour chaque transaction incluse.

Avant d'annoncer une transaction, l'expéditeur doit spécifier, en phase de définition, les frais maximums (max_fee) que le compte permet de dépenser pour la transaction en question.

Si les frais effectifs (`effective_fee`) sont inférieurs ou égaux aux frais maximums (`max_fee`), un harvester pourrait choisir d'inclure la transaction au bloc. Les nœuds sont libres d'établir leur stratégie d'inclusion :

- **Privilégier les plus anciennes (*prefer-oldest*) :** Une stratégie particulièrement adaptée aux réseaux avec de hautes exigences de débit de transactions. Consiste à inclure d'abord les transactions les plus anciennes.

- **Minimiser les frais (*minimize-fees*) :** pour les nœuds philanthropiques. Consiste à inclure d'abord les transactions que les autres nœuds ne souhaitent pas inclure.
- **Maximiser les frais (*maximize-fees*) :** la stratégie la plus commune sur les réseaux publics. Consiste à inclure d'abord les transactions avec les frais les plus élevés.

Pour s'assurer que la transaction soit incluse sans fixer de frais maximums inutilement hauts, l'expéditeur de la transaction peut demander à la passerelle REST quel était le multiplicateur médian, moyen, le plus haut et le plus bas du réseau sur les derniers N blocs.

L'expéditeur pourrait par exemple fixer comme suit les frais maximums :

$$\mathbf{maxFee = transaction::size * network::medianFeeMultiplier}$$

Récompenses du réseau

La configuration du réseau de la blockchain de Veritise™ prévoit un compte de collecte des frais de réseau sur lequel sera versé un pourcentage des gains de staking (frais de bloc et inflation). Fixé à 10 %, il sert à soutenir le développement du réseau et les différents programmes de récompense.

Une collecte prédéterminée s'applique également aux frais de location des espaces de nom et sous espaces de nom et aux frais d'émission de jetons, elle aussi dédiée au développement et aux activités de pénétration du marché de Veritise™.

Chaque opérateur de nœud peut en outre établir un compte bénéficiaire pour partager un pourcentage (jusqu'à 25 %) de ses gains de harvesting. Un moyen pour les opérateurs de créer des structures incitatives pour ceux qui soutiennent leur nœud.

Méthodes de staking

Différentes méthodes de staking sont proposées, en fonction de l'appartenance ou non du nœud au compte de staking et du niveau de sécurité souhaité : le staking local, distant ou délégué.

Staking local

La méthode la plus simple à mettre en place, et la moins sûre. Elle consiste à modifier la configuration d'un nœud de façon à ce qu'il ne soit plus disponible que pour les propriétaires de nœud. Il suffit pour ce faire de remplir les propriétés de staking applicables dans le fichier de configuration du nœud.

Visible, la clé privée du compte du participant au staking est stockée dans les propriétés du nœud, car elle est nécessaire à la validation des blocs créés. Une approche problématique en matière de sécurité, dans la mesure où ce compte contient des fonds et des acteurs non invités pourraient accéder au fichier de configuration si le nœud est compromis. Les clés privées des comptes approvisionnés devraient toujours être stockées hors ligne.

Pour cette raison, cette méthode est fortement déconseillée, en faveur du staking distant ou délégué.

Staking distant

Les propriétaires de nœuds peuvent utiliser un compte distant en qualité de mandataire pour valider les blocs nouvellement créés, tout en touchant leurs gains de staking sur leur compte principal. Ce compte distant n'étant pas approvisionné en fonds, le fait que sa clé privée soit exposée dans un fichier de configuration sur le nœud ne pose pas de problème. Le score d'importance se base quant à lui toujours sur le compte principal.

Dans ce système, le compte principal continue d'être désigné participant au staking par souci de simplification, tandis que le compte distant est appelé mandataire.

Le harvesting distant est activé exactement comme le harvesting local, mais au moyen de la clé privée du compte distant dans les propriétés du nœud, et en annonçant une transaction reliant le compte distant au compte principal.

Staking délégué

Les comptes éligibles ne détenant pas de nœud peuvent aussi bénéficier du harvesting en demandant à un nœud de staker et de participer pour eux au consensus. Le score d'importance du compte est utilisé et tous les frais collectés sont répartis entre le compte et le bénéficiaire du nœud (comme expliqué dans la section Récompenses). C'est un accord mutuellement bénéfique pour le compte et le nœud.

On dit alors que le compte délègue le staking au nœud, mais le compte est toujours considéré comme le participant au staking.

Le staking délégué s'active de façon similaire au staking distant, mais étant donné que le compte n'a pas accès à la configuration du nœud, il annonce une transaction de type PersistentDelegationRequest pour demander la délégation. Une fois la requête réceptionnée, le nœud peut y accéder ou non, selon sa configuration et le reste des demandes reçues.

Comme pour le staking distant, un compte mandataire distant est utilisé de façon à ce que la clé privée du compte principal ne soit jamais exposée à un quelconque risque.

Tokénomique

- Offre totale : 300 de jetons VTS

- Offre en circulation la 1re année : 75M à 200M maximum en fonction des ventes de jetons publics
- Inflation extrêmement basse (à partir de 0,2 % ou moins par an, diminué au fil du temps)
    - Pour maintenir l'offre et compenser les clés/portefeuilles/fonds perdus
    - Pour récompenser les nœuds qui maintiennent le réseau de la blockchain alors que le volume de transactions continue d'augmenter
    - Pour protéger la valeur pour les détenteurs de jetons
- Plus de 400 TPS (transactions par seconde) sur le réseau public et plus de 3 000 TPS sur le réseau privé
- Temps de génération de bloc ciblé : 15 secondes avec des temps de transaction confirmée de moins de 5 secondes
- Jusqu'à 4096 caractères de données sur-chaîne par transaction

- Fin du livre blanc technique -

- contenu susceptible d'évoluer -