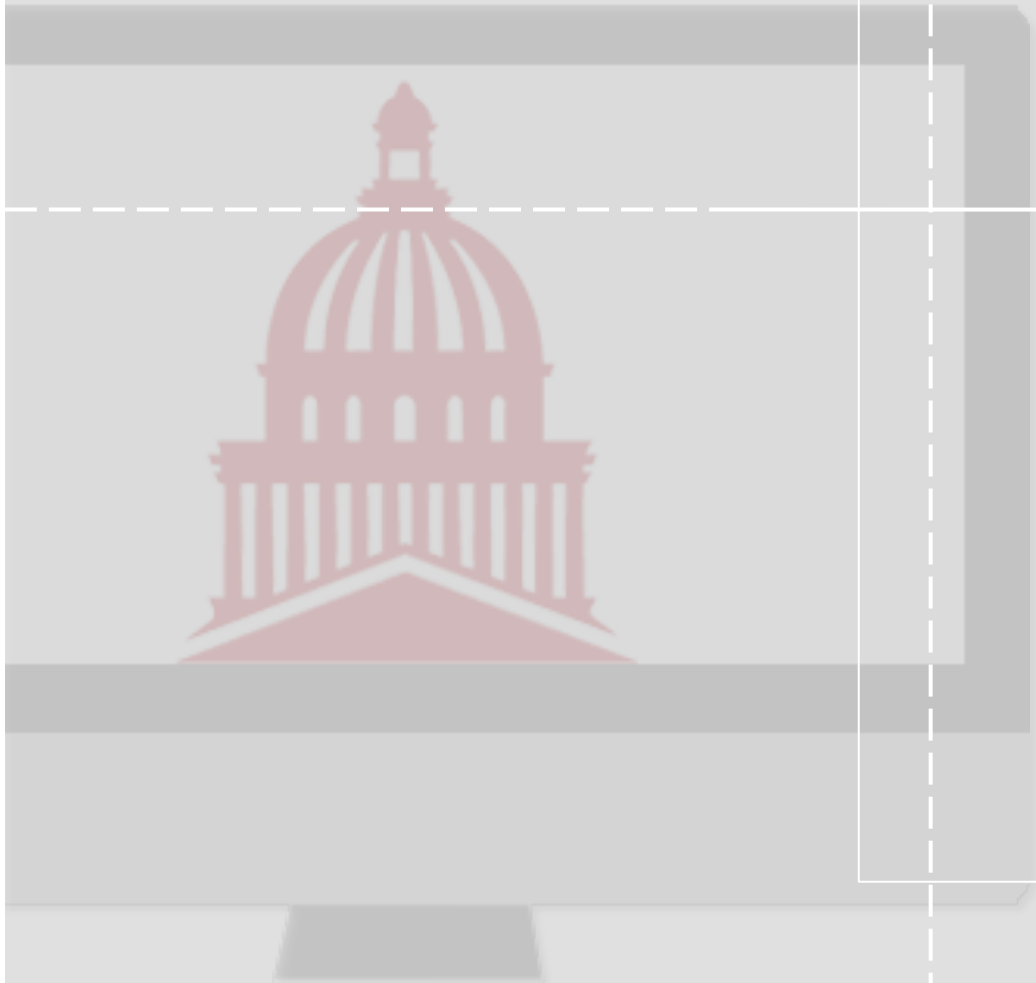
Le petit guide des smart contracts



Intro

Le petit guide des smart contracts vous est présenté par les départements Digital Contract et Legal Design de l'association Assas Legal Innovation. En effet, cette association a pour but de sensibiliser les étudiants et les professionnels à l'impact des nouvelles technologies sur le droit et à leur régulation. En proposant ce guide, l'association cherche donc à permettre aux juristes d'appréhender la notion de smart contract et à déterminer ses applications potentielles dans le domaine juridique. Pour faciliter la compréhension des notions, nous avons eu recours à des schémas illustrant notre propos. Ce guide vient notamment compléter le petit guide de la Blockchain qui avait été réalisé sur le même modèle par les pôles Blockchain-Fintech et Legal Design.

Le SMART CONTRACT est un programme qui, s'appuyant sur la technologie Blockchain, utilise les conditions et les données inscrites sur cette dernière dans le but qu'un contrat librement consenti, auquel est appliqué un code le rendant lisible par la machine, s'exécute automatiquement et soit infalsifiable. Le terme « smart » ne désigne pas ici la trace d'une intelligence artificielle, mais l'auto exécution du contrat.

Bien que la technologie de la chaîne de blocs soit considérée par beaucoup comme le symbole de l'avenir, le terme " contrat intelligent " est connu depuis des décennies.

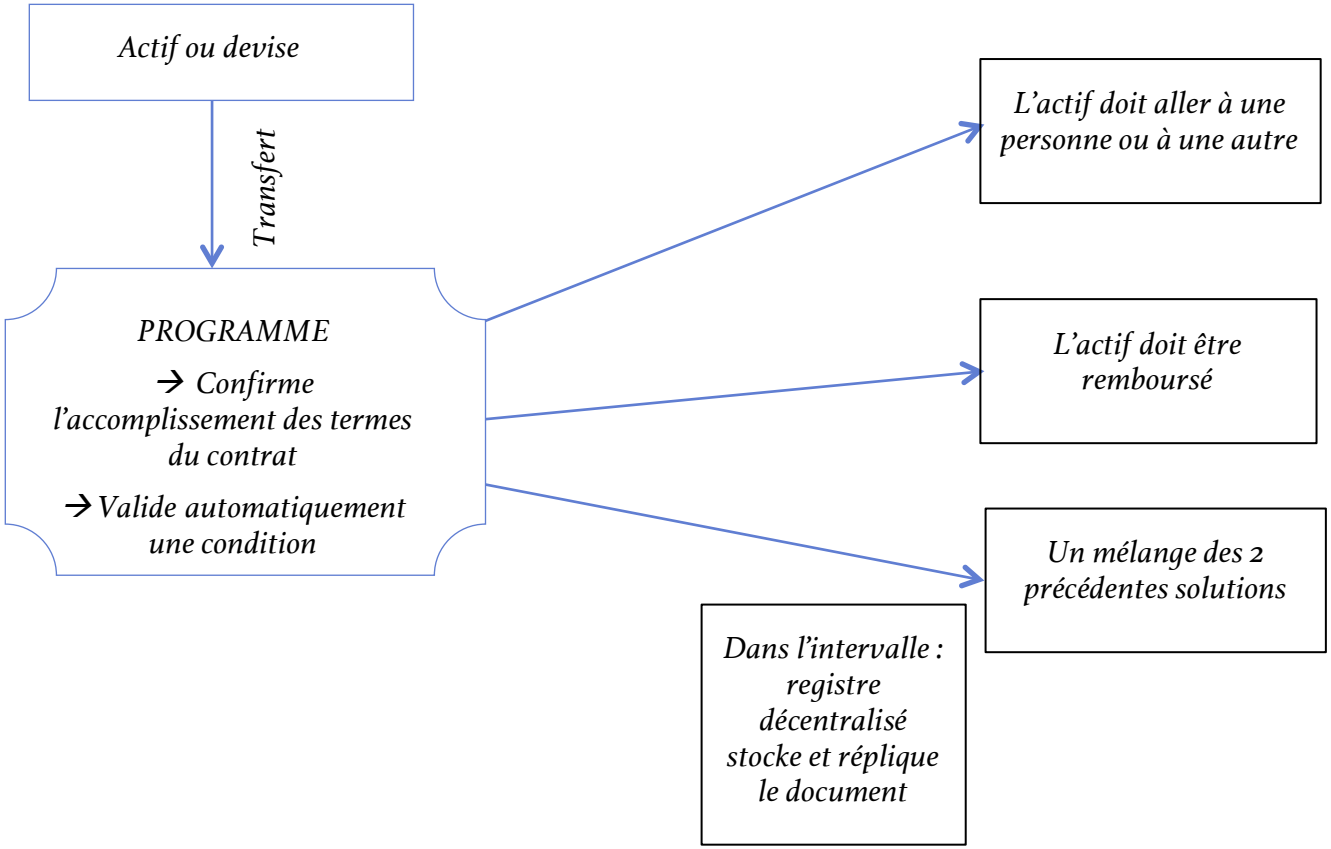
Un cryptographe et informaticien Nick Szabo a eu l'idée au milieu des années 1990. Selon ses propres termes, un contrat intelligent " est un protocole de transaction informatisé qui exécute les termes d'un contrat. Les objectifs généraux sont de satisfaire aux conditions contractuelles communes".

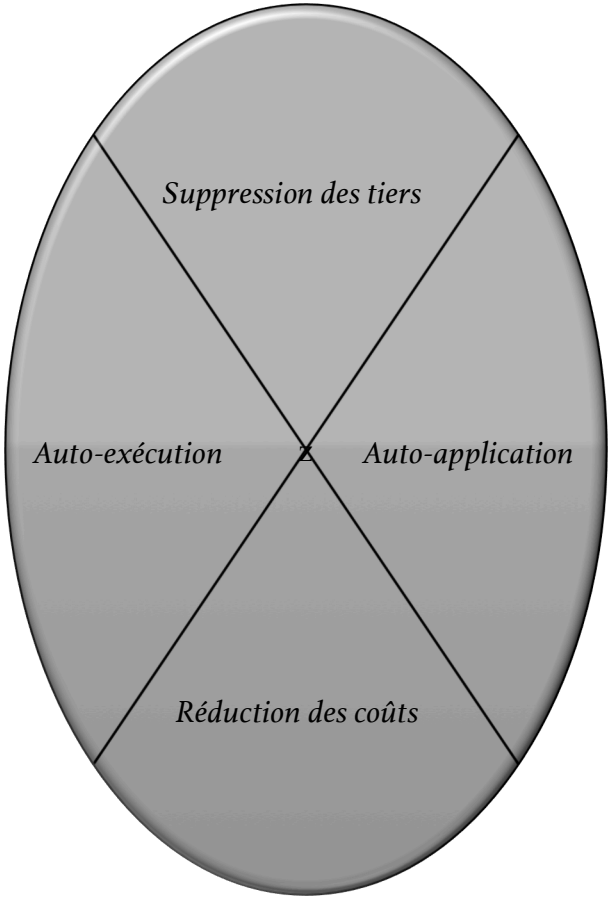
L'idée principale derrière les contrats intelligents est de déterminer les relations et les obligations entre les parties via un code informatique et les administrer automatiquement.

Pour définir les smart contracts de manière moins technique, il convient de parler de contrats intelligents permettent d'échanger de l'argent, des biens, des actions ou tout ce qui a de la valeur d'une manière transparente et non conflictuelle. En d'autres termes, les contrats intelligents apportent la confiance, ce qui est un facteur crucial pour un réseau décentralisé de chaînes de blocs où les parties restent anonymes.

Selon Vitalik Buterin, programmeur du projet Ethereum, lors de l'événement Blockchain à Washington DC en 2016: Grâce à l'utilisation d'un contrat intelligent, un actif ou une devise est transféré dans un programme qui contrôle sa conformité avec l'ensemble des conditions. A un moment donné, ce programme confirme l'accomplissement des termes du contrat et " il valide automatiquement une condition et détermine automatiquement si l'actif doit aller à une personne ou à l'autre, ou s'il doit être remboursé immédiatement à la personne qui l'a envoyé ou une combinaison des deux ". Dans l'intervalle, un registre décentralisé stocke et réplique également le document, ce qui lui confère une certaine sécurité et une immuabilité.

Le fonctionnement des smart contracts





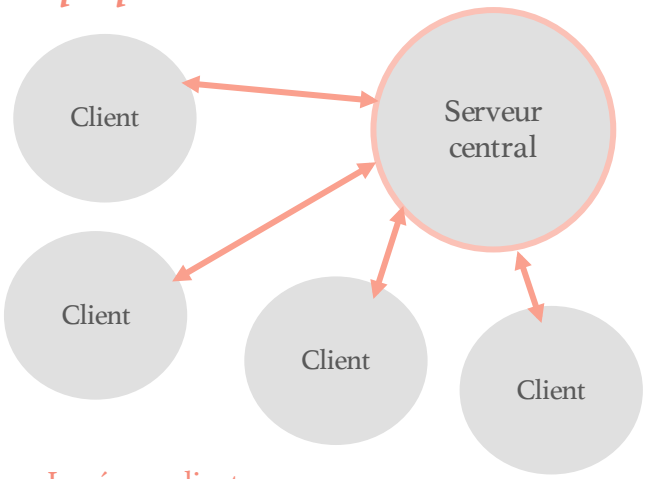
Caractéristiques des smart contracts

I. L'utilisation d'un réseau pair à pair (P2P)

Le pair à pair est un réseau de partage de données entre plusieurs ordinateurs.

*Il doit être distingué du **réseau client-serveur.***

Dans un réseau client-serveur, le serveur est la seule source des données. C'est une entité passive qui attend les requêtes des clients et leur envoie des données. Ce type de réseau est dit propriétaire et centralisé.

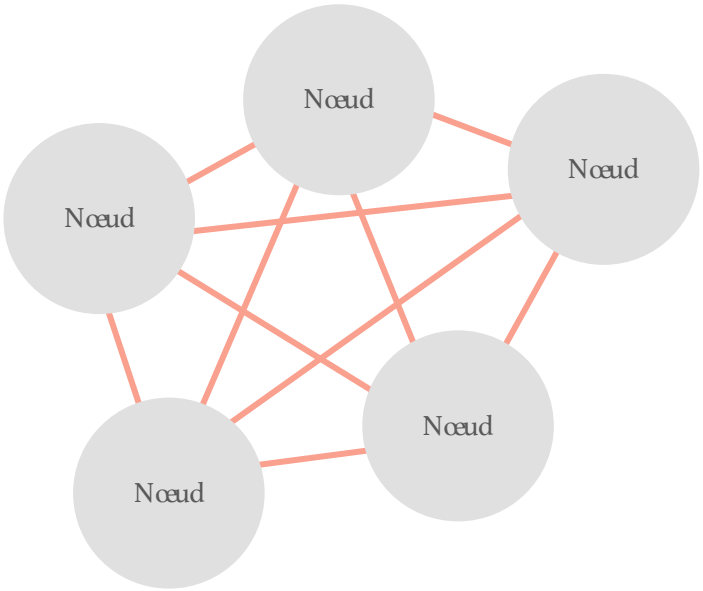


Le réseau client-serveur

Le problème est que si un nombre important de clients envoie une requête au serveur, celui-ci aura du mal à répondre à toutes les demandes.

*C'est le problème de la congestion.**

*Pour remédier à ce problème est apparu le modèle du **pair-à-pair** (peer-to-peer, égal à égal). C'est un système décentralisé, distribué.*



Le réseau peer-to-peer

Le P2P est un système de répartition de charge. Les entités sont à la fois clients et serveurs :

- *Clients car elles peuvent solliciter des données*
- *Serveurs car elles peuvent être sollicitées pour obtenir des données.*

Il n'y a en fait aucun serveur : il suffit de deux ordinateurs, qui constitueront les nœuds du réseau.*

Tous les nœuds ont le même rôle ; il n'y a pas de statut privilégié pour un nœud.

*Congestion

Augmentation du trafic provoquant un ralentissement global du réseau informatique.

*Nœuds

Postes connectés par un protocole réseau pair-à-pair.

Chaque utilisateur décide des partages sur son disque dur et des permissions qu'il octroie aux autres utilisateurs.

Une ressource partagée sur un ordinateur constituant un nœud apparait sur tous les autres ordinateurs qui sont connectés au réseau : c'est le concept de ***partage arbitraire.***

Le système peer-to-peer permet ainsi l'échange d'objets (fichiers, flux multimédias continus, le calcul réparti, services...) entre tous les ordinateurs du réseau sans passer par un serveur central.

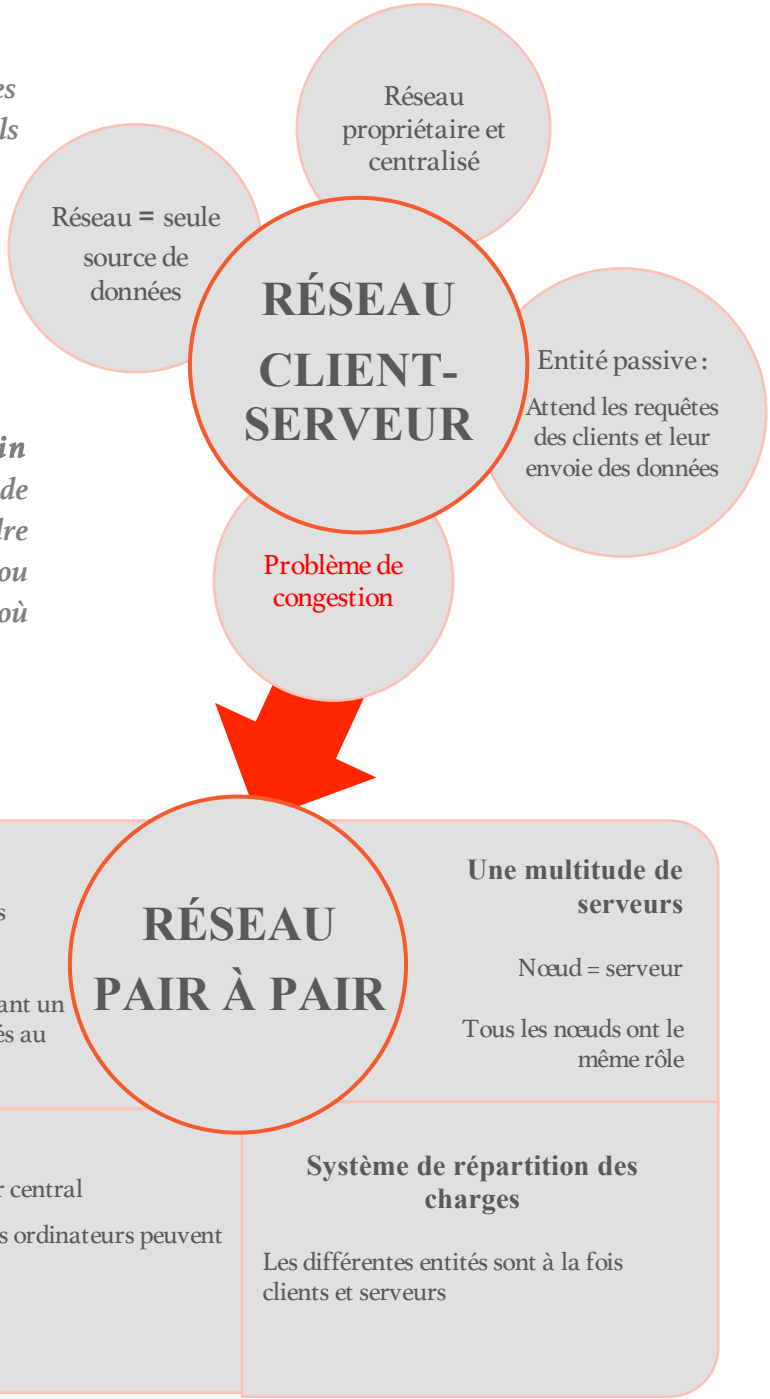
L'avantage : alors que dans un réseau client-serveur une panne du serveur empêche le réseau de fonctionner, dans un réseau P2P, si un nœud tombe en panne, il reste toujours tous les autres ordinateurs pour servir de ressource.

L'utilisation d'un système pair-à-pair nécessite pour chaque nœud l'utilisation d'un logiciel qui remplit à la fois les fonctions de client et de serveur. Ce logiciel est appelé ***servent****, ou encore « client ».*

Les systèmes pair-à-pair ***facilitent le partage d'informations*** *et rendent la censure ou les attaques légales ou pirates plus difficiles. Ce sont ainsi des outils de choix pour* ***décentraliser des services qui doivent assurer une haute disponibilité tout en permettant de faibles coûts d'entretien.***

Néanmoins, ces systèmes sont plus complexes à concevoir que les systèmes client-serveur.

Le système P2P favorise ***la coopération au sein d'une communauté.*** *Lorsqu'un grand nombre de nœuds propose des ressources, le système peut atteindre son plein potentiel. Le risque est alors qu'un ou plusieurs nœuds partagent des fichiers corrompus d'où la nécessité de s'équiper d'un firewall efficace.*

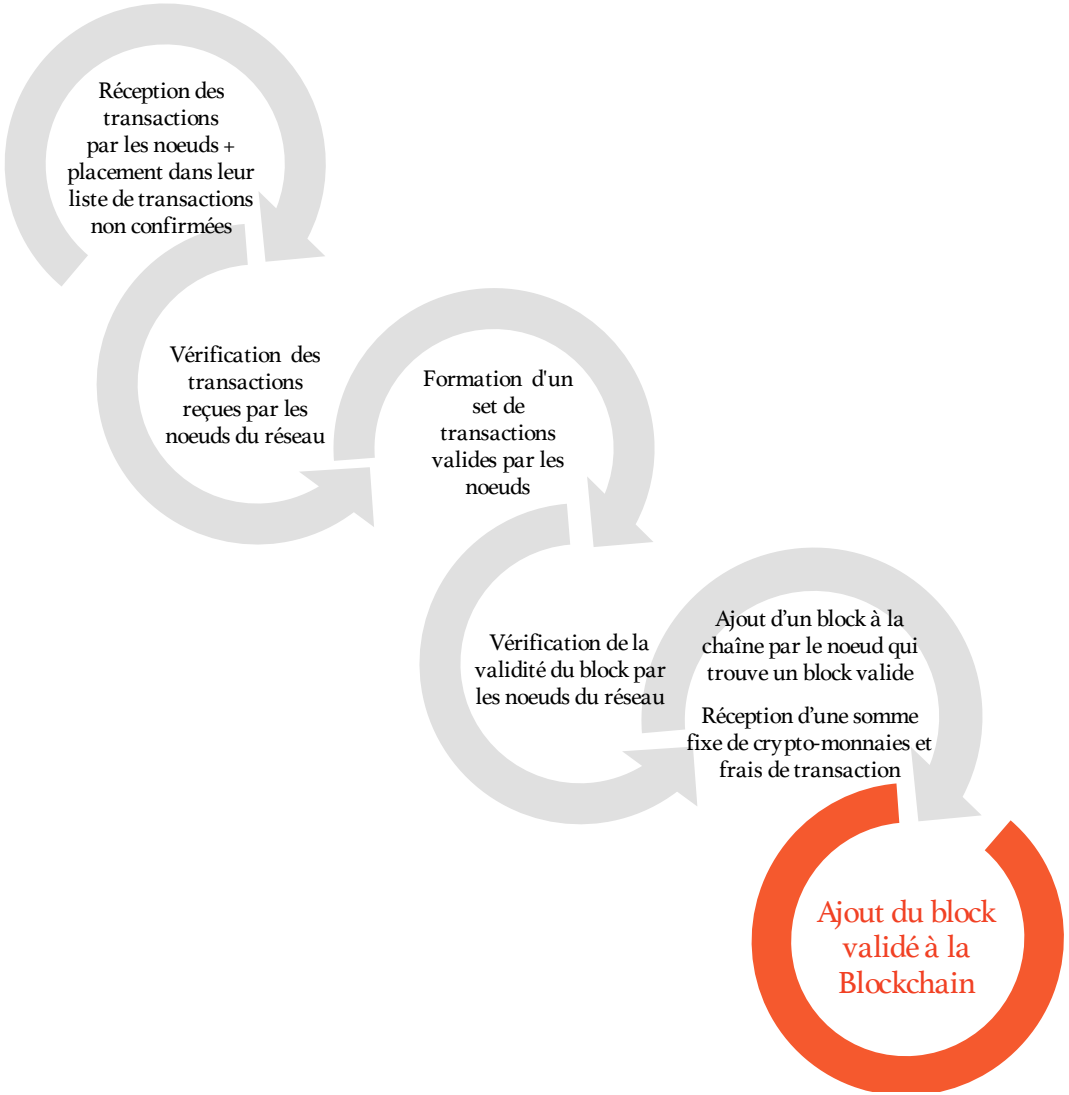


II. L'utilisation de la Blockchain dans les smart contracts

A. Définition et fonctionnement de la Blockchain

La traduction littérale du terme blockchain en français est « chaine de blocs ». La blockchain est un registre transparent, décentralisé et sécurisé permettant d'échanger et de stocker des informations. Elle constitue une base de données qui gère une liste d'enregistrements protégés contre les modifications ou la falsification et contenant l'historique de tous les échanges réalisés par ses différents utilisateurs depuis sa création. Mise à jour en temps réel, cette base de données est stockée sur les serveurs de chacun de ses utilisateurs (en principe). Ces derniers peuvent ainsi s'échanger des données sans intermédiaire et sans risque. On distingue les blockchains publiques qui sont accessibles à tous des blockchains privées, à l'accès limité.

La blockchain est construite autour d'un algorithme qui vérifie l'exactitude de chacune des données qui lui sont proposées avant de les enregistrer. Les données validées sont ensuite regroupées en blocs sur le réseau, tandis que les données erronées sont systématiquement rejetées. La décentralisation de la blockchain garantit l'impartialité du processus de validation.



B. Comment les smart-contracts fonctionnent avec la Blockchain ?

Ce rudimentaire exposé des éléments fondamentaux qui caractérisent la blockchain permet de saisir l'intérêt de son concours au bon fonctionnement des smart contracts. Chacun a déjà entendu parler d'au moins un smart contract ... le Bitcoin. Certes le contrat est basique et limité à une fonction monétaire, mais permet tout de même un transfert de valeur entre deux personnes.

Aujourd'hui déjà et a fortiori demain, la blockchain permettra à des agents autonomes de coder des contrats entièrement personnalisés et d'une grande complexité. Les conditions stipulées dans le contrat seront automatiquement exécutées grâce aux vérifications autonomes de la blockchain (sur un modèle si/quand alors).

<u>Exemple pratique : achat d'une automobile d'occasion</u>

Monsieur A veut acheter une voiture d'occasion. Pour financer son achat, il souhaite contracter un prêt pour lequel il doit fournir de nombreux documents.

Dans un monde non digitalisé :

- S'assurer que le vendeur est bien propriétaire
- S'assurer que les documents attestant du suivi régulier de l'automobile sont bien authentiques
- Effectuer le paiement des sommes dues à la livraison de l'automobile

→ Déficit de confiance
→ Délais longs
→ Intermédiaires

Dans un monde digitalisé :

- Les Smart contracts permettraient d'obtenir un prêt sans constituer de dossier car toutes les informations nécessaires sont disponibles sur la blockchain
- L'authenticité des actes de propriété et d'entretien serait également validée par la bockchain
- La blockchain organiserait le remboursement du prêt selon l'échéancier convenu et libérerait automatiquement les fonds destinés au vendeur du véhicule dès sa livraison

→Confiance restaurée
→ Délais raccourcis
→Plus d'intermédiaires

III. La distinction entre les smart contracts et les autres contrats

A. La distinction entre les smart contracts et les ricardian contracts

Le ricardian contract prend racine dans le travail de Ian Grigg, spécialiste de la cryptographie financiale. Il est complété au milieu des années 1990 par les contributions de Ricardo, un système de transfert d'actifs qui a été construit en 1995-1996. Le système et le modèle de conception ont été nommés d'après David Ricardo, en hommage à sa contribution formatrice à la théorie du commerce international.
Selon son créateur, un contrat ricardien est "un contrat numérique qui définit les termes et conditions d'une interaction, entre deux ou plusieurs pairs, qui est signé cryptographiquement et vérifié. Il est à la fois lisible par l'homme et par la machine et signé numériquement".

Un ricardian contract enregistre un document légalement valide et relié numériquement à un objet ou à une valeur déterminée. Sa mise en œuvre peut varier. Il place toutes les informations du document légal dans un format qui peut être exécuté par un logiciel. Il s'agit donc à la fois d'un accord juridique entre les parties et d'un protocole qui intègre un accord offrant un haut niveau de sécurité grâce à l'identification cryptographique.

Caractéristiques des ricardian contracts

Le document :

- Identifiable de manière sure
- Signé par l'émetteur
- Imprimable

L'analyse du document :

- Analysable par l'homme
- Programme analysable
- Formulaires équivalents

Leurs différences :

SMART CONTRACT	*RICARDIAN CONTRACT*
N'est jamais un ricardian contract	*N'est pas toujours un smart contract*
Pas nécessairement signé par l'homme	*Nécessairement signé par l'homme*
Executé par une machine	*Pas besoin d'être exécuté par une machine*

B. La distinction entre les smart contracts et les contrats classiques

	LES SMART CONTRACTS	***LES CONTRATS CLASSIQUES***
Mise en œuvre	*Immédiate et irrévocable*	*Pendant durée du contrat*
Lisibilité	*Difficile pour l'humain, facile pour la machine*	*Facile pour l'humain, difficile pour la machine*
Adaptabilité	*Difficile si blockchain publique*	*Facile si accord des parties*
Rédaction	*Rapide*	*Lente*
Sécurité	*Elevée*	*Limitée*
Archivage	*Facile*	*Difficile*
Extraction des données	*Immédiate*	*Lente*
Accesibilité du contrat	*Contrat public si blockchain publique*	*Bonne confidentialite*

IV. Un langage et une méthodologie

Les computational contracts (i.e. ricardian contract)

Le contrat de base ne fait que représenter un accord simple et logique : dans le contrat de vente, si une personne paye le prix de la chose, alors la remise de cette dernière survient. Un logiciel raisonne souvent de la même façon. Pour ouvrir ce petit guide, vous cliquez dessus, et en échange, le logiciel exécute la contrepartie en ouvrant le document.
On peut alors imaginer que le contenu d'un contrat puisse être transposé dans un code.

Il existe plusieurs sortes de langages. Parmi eux, le langage « object oriented » : on recherche des récurrences autour de l'objet du contrat, par eemple la vente d'une voiture. Ces éléments sont récurrents lorsqu'ils s'observent dans chaque contrat. Ces récurrences vont être définies comme des paramètres de l'objet qui seront identifiées par le code. Ceci rend le contrat et ses termes facilement manipulables.

Ensuite, on attribue une valeur variable à l'objet récurrent. Enfin, on insère ce paramètre dans le document final. Ainsi, le simple apport de la donnée nécessaire au contrat en cours d'élaboration changera l'acte final. En effet, on modifie la valeur variable du paramètre identifié, ce qui a pour effet direct de changer la valeur de cette propriété dans le document final. (Pour plus d'explications, voir le lexique digital contract sur le site d'Assas legal innovation.)

Prenons pour exemple l'automatisation d'une clause, courante, déterminant le droit applicable au contrat et de la juridiction devant laquelle sera portée le litige.
Il suffira de renseigner à la machine les paramètres qui changeront, soit directement (comme ci-dessous), soit en faisant appel à un fichier de données extérieur.

```
Ti=Article 8. Droit applicable - Litiges

// Paramètres

competence_Juridiction=françaises

loi_Applicable=française

// Computational contract

1.sec=Si des difficultés surviennent à l'occasion de l'interprétation ou de
l'exécution ou de la terminaison du présent contrat, les parties auront
recours à une conciliation amiable préalablement à toute instance judiciaire.

2.sec=Si les difficultés survenues à l'occasion de l'interprétation ou de
l'exécution ou de la terminaison du présent contrat n'ont pas été résolues par
voie de conciliation, le litige sera soumis aux juridictions
{competence_Juridiction} compétentes qui le trancheront conformément à la loi
{loi_Applicable}.

=[G/Z/ol/s2]
```

```
//Résultat clause

Article 8. Droit applicable - Litiges

Si des difficultés surviennent à l'occasion de l'interprétation ou de
l'exécution ou de la terminaison du présent contrat, les parties auront
recours à une conciliation amiable préalablement à toute instance judiciaire.

2.sec=Si les difficultés survenues à l'occasion de l'interprétation ou de
l'exécution ou de la terminaison du présent contrat n'ont pas été résolues par
voie de conciliation, le litige sera soumis aux juridictions françaises
compétentes qui le trancheront conformément à la loi française.
```

Le langage smart contracts

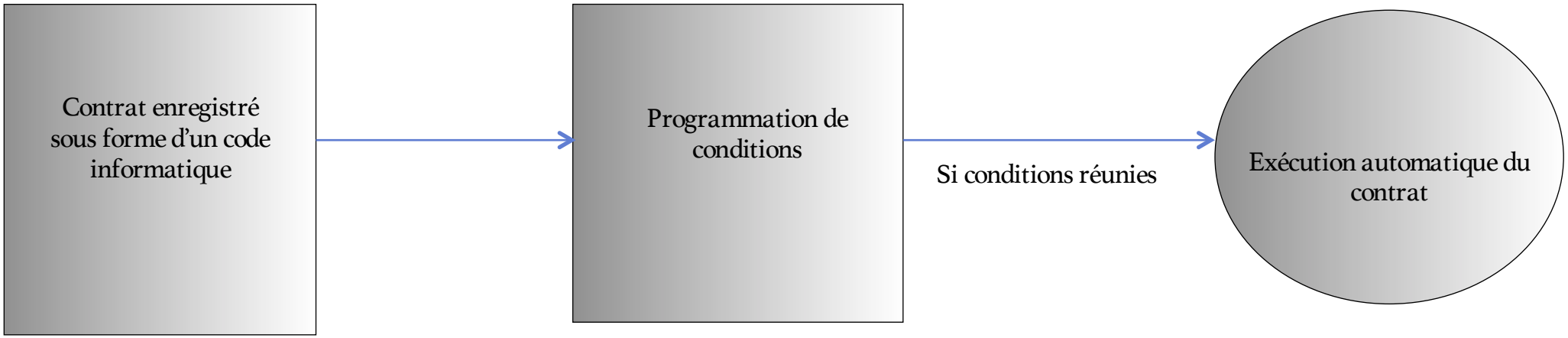
Ces "contrats intelligents" consistent à enregistrer des contrats sous la forme d'un code informatique qui s'activerait automatiquement lorsque certaines conditions seraient réunies. Lorsque ces éléments déclencheurs surviennent, le contrat encodé s'exécute.

Il y a de très nombreux langages pour les smart contracts. On en trouve plusieurs par plateforme. Ces langages peuvent revêtir différentes fonctions.

Déployés sur la blockchain Ethereum :

- *WaBi, qui met en interaction les producteurs, les consommateurs, et l'organisation responsable de l'authenticité et de la sécurité des produits de grande consommation comme la nourriture et les produits pharmacologiques de sorte à garantir la traçabilité du contrôle qualité via la technologie RFID.*
- *Ethereum bytecode → langage de programmation Ethereum agissant dans la blockchain. C'est un des langages qui rend les opérations faites via ethereum auto-exécutables sur la blockchain.*

"Les contrats intelligents Ethereum sont des ensembles d'instructions de programmation exécutées sur tous les nœuds exécutant un client Ethereum complet. La partie d'Ethereum qui exécute les instructions de contrat intelligent est appelée EVM. C'est une machine virtuelle qui lit une représentation de bas niveau de contrats intelligents appelée Ethereum bytecode. Le bytecode Ethereum est un langage d'assemblage composé de plusieurs opcodes. Chaque opcode effectue une certaine action sur la blockchain Ethereum".



V. L'utilisation des smart contracts dans différents domaines

A. Les contrats de location

Prenons l'exemple d'un contrat de location d'un appartement.

Il suffirait de connecter la serrure de la porte de son appartement à la blockchain et de lier cette serrure à un smart-contract de location.

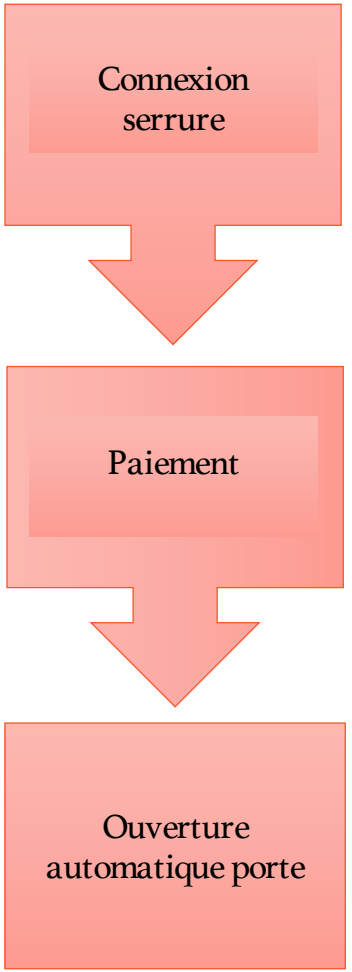
Lorsque quelqu'un veut utiliser votre appartement, il paye le prix de la location au contrat qui est lié à votre serrure sur la blockchain. Une fois le paiement effectué, la porte s'ouvre automatiquement pour la personne qui a réglé la location, pendant la durée correspondant au paiement.

En pratique, tout est géré par des smart-contracts de façon transparente. La location se fait par l'intermédiaire d'une dApp (une application décentralisée sur Ethereum) et le paiement se fait en ethers. Slock.it travaille cependant à l'intégration de moyens de paiement classiques comme la carte bleue avec des partenaires commerciaux. Dans ce cas les euros seront automatiquement convertis en cryptomonnaie, et vice versa.

Le tout est contrôlé par une application sur le smartphone ou un site internet, de façon transparente pour l'utilisateur qui aura à gérer uniquement une interface web classique.

Un fonctionnement similaire peut aussi être imaginé pour louer un vélo, une machine à laver, une voiture, etc.

Le tiers de confiance disparait donc de l'équation : les objets « se louent eux mêmes »



B. Les contrats d'assurance

L'utilisation des smart contacts permettrait d'apporter des réductions importantes de coûts pour les assureurs et les assurés, et d'améliorer l'expérience client.

<u>Gestion des réclamations</u> :

Le Smart Contract permettra une automatisation complète de la procédure de paiement des réclamations et donc un raccourcissement des délais de paiement pour les assurés. Au lieu d'attendre plusieurs semaines à plusieurs mois pour se faire dédommager, les clients se feront rembourser quasi-instantanément.

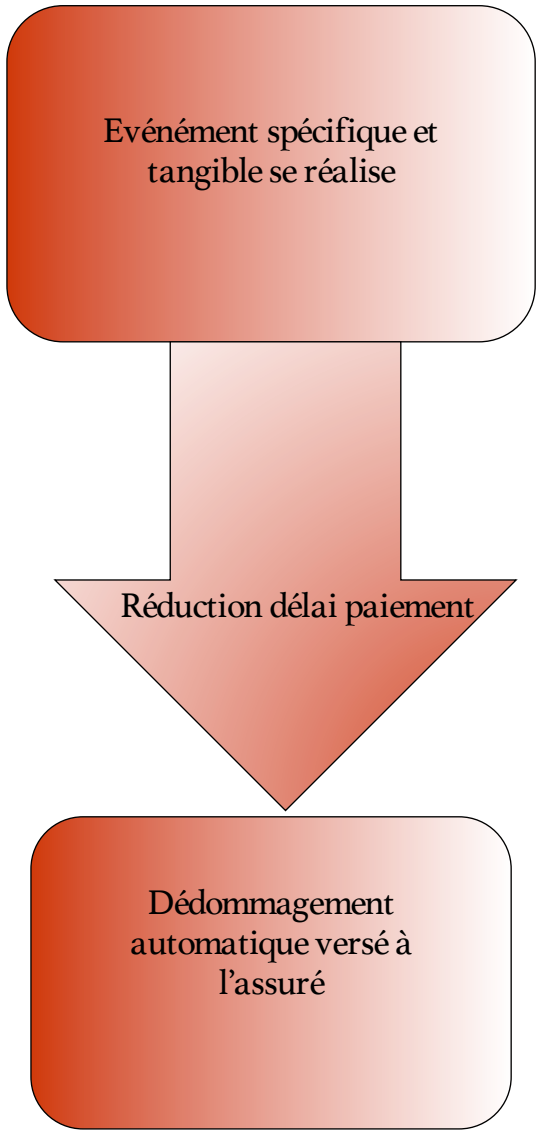
<u>Assurances paramétriques</u> :

L'utilisation du Smart Contract est particulièrement pertinente dans le cas de risques dits "paramétriques" dans lesquels la mesure d'un événement spécifique et tangible permet de générer automatiquement des dédommagements à l'assuré. C'est le cas des risques météorologiques par exemple : dans ces cas de figure, le paiement est activé par des sinistres naturels prévisibles (vitesse du vent, localisation d'un ouragan, magnitude d'un tremblement de terre, etc...).

Pour un exemple très concret : remboursement du billet d'avion en retard : https://fizzy.axa/fr/

<u>Objets connectés</u> :

Le déploiement d'objets connectés dans notre quotidien (voitures, maison) est à l'origine du développement de nouveaux produits d'assurance basés sur le Smart Contract. Le placement de capteurs dans les domiciles permet de mesurer les sinistres (rupture de canalisation, coupure de courant ou panne d'appareil) et d'envoyer cette information au Smart Contract, ce qui enclenche un dédommagement automatique de l'assuré.
Le Smart Contract pourrait donc bien bouleverser le paysage assurantiel tel qu'on le connaît. Plusieurs startups se sont déjà saisies des opportunités qu'offre cette technologie pour développer des produits assurantiels complètement novateurs.

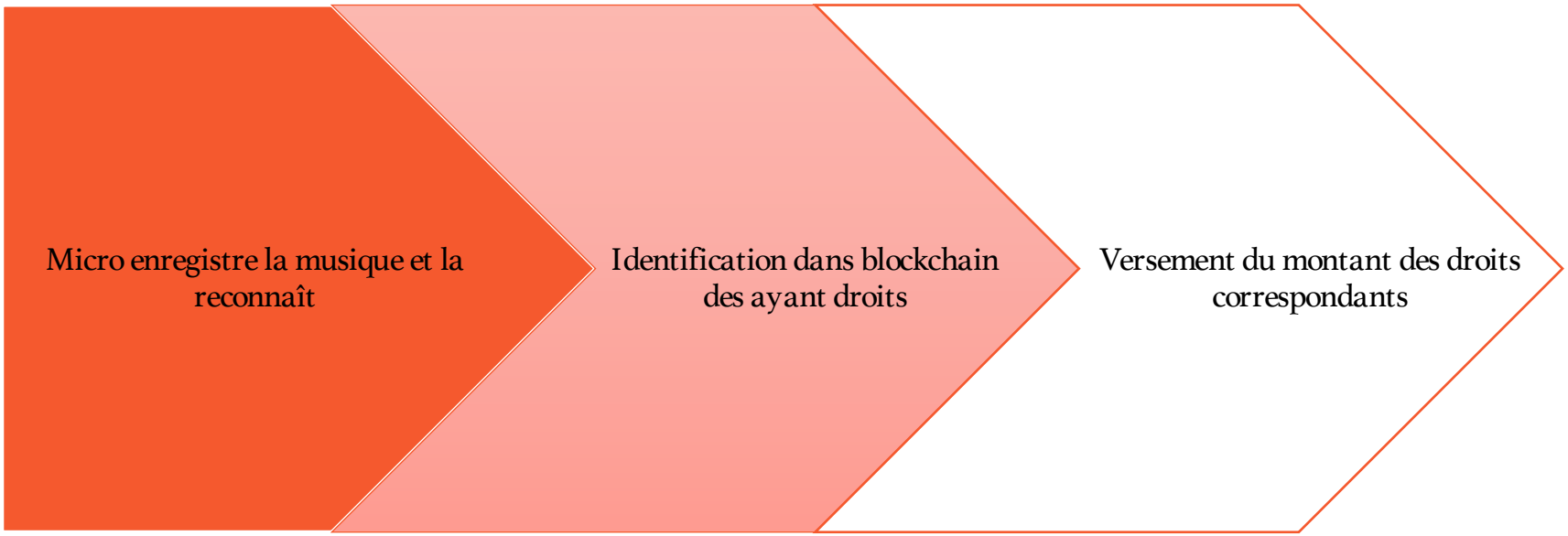


C. Les copyrights

Certaines start-up ont d'ores et déjà entamé la transformation en organisant par le biais de smart contracts la perception directe et immédiate par les musiciens des droits sur leur œuvre, avec pour perspective le contournement de la gestion collective des droits.

Le dispositif est décrit de la manière suivante : « un micro enregistre la musique diffusée, reconnaît le morceau, identifie dans la blockchain les ayants droit et exécute le contrat en leur reversant le montant des droits correspondants ».

L'avantage c'est qu'un artiste peut demander une rémunération pour une diffusion de sa musique à la radio et ne pas en demander aux boîtes de nuit et demander un prix moins élevé aux particuliers. Il pourrait ainsi ne pas faire payer des sites de streaming qui ont une approche éthique. Enfin, la blockchain et les smart contracts permettent de différencier la répartition des droits : 5% à tel musicien, 2% à tel autre... explique Clément Jeanneau, cofondateur de Blockchain France.



Le petit guide des Digital Contracts a été réalisé par
(dans l'ordre alphabétique) :

Sabrina Boukhatem

Magali Cadoret

Baptiste Etienne

Florian Imbert

Hulé Kéchichian

William Kocemba

Rosalie Lechat

Sarah Maïssa Belmekki