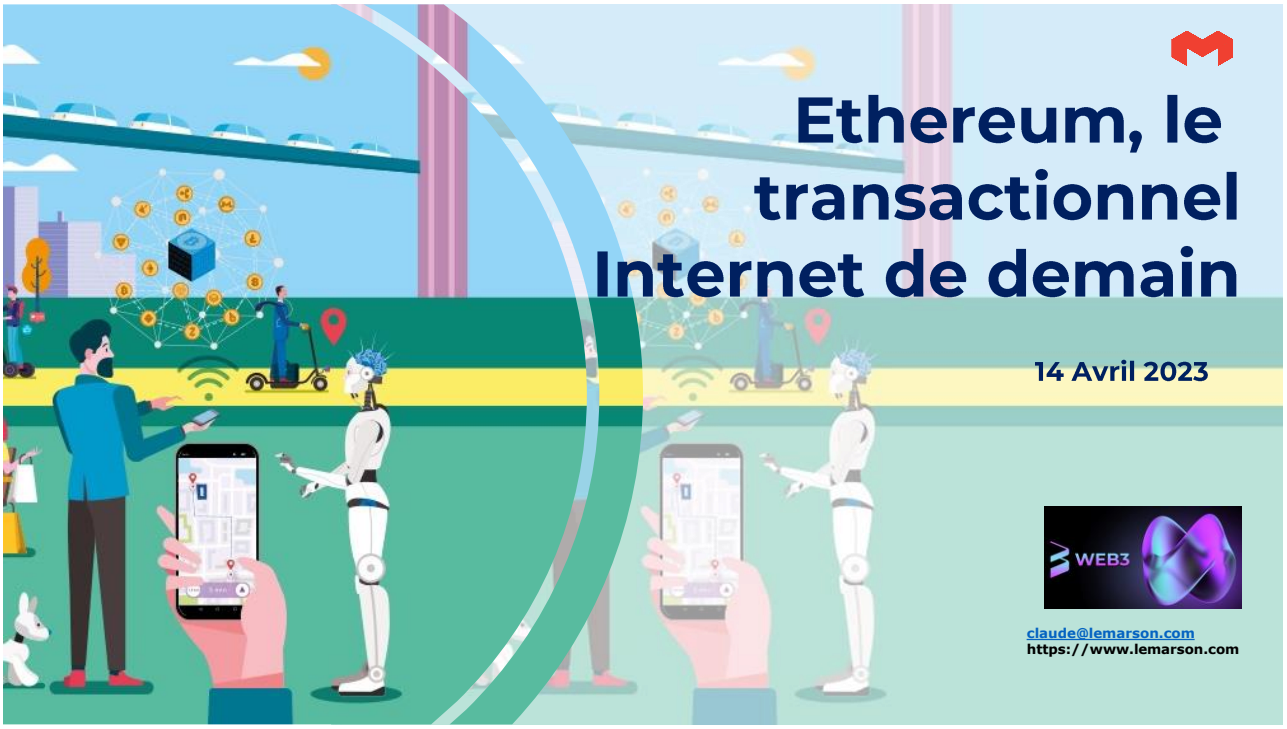


Ethereum, le transactionnel Internet de demain

14 Avril 2023

claude@lemarson.com
https://www.lemarson.com

Sommaire

Ethereum, le transactionnel de demain sur Internet

- *Ce qu'est Internet : asynchrone et communication*
- *L'avenir du transactionnel classique, peu de changements*
- *Les nouveaux besoins : transactionnel lourd distribué*
- *Le choix : IaaS Cloud ou Internet natif*
- *Les nécessités nouvelles (Web3) : blockchain, décentralisation, "smart contracts", monnaies cryptographiques, dématérialisation, sécurité*
- *Architecture et constituants d'Ethereum, au-delà des monnaies*
- *Crédibilité d'Ethereum sur le créneau Web3 ?*
- *Avantages et inconvénients*
- *Des réalisations encore très techniques*
- *Faut-il prendre le risque ?*

2 700 applications recensées sur Ethereum… de loin leader mondial du domaine

Ce qu'est Internet aujourd'hui

Une forte connotation asynchrone

- 95 % des applications ont ou auront un client Web : Internet et Intranet
- Un réseau de transport
- Des protocoles totalement "déconnectés" de la réalité sécuritaire des TI
- Des services de messagerie
- Connexion en totale ubiquité
- Outils de collaboration synchrone et visuelle : IM, vidéo (Teams, Zoom…), avec une couche de protocoles propriétaires
- Support envisageable d'espaces virtuels (metavers)
- Réseaux sociaux
- VoIP
- Applications de nature asynchrone
- Services de géolocalisation
- Des services de complément : financiers, logistique, éducation…

Les besoins de demain

Transactionnel lourd et distribué
Internet est-il adapté aux besoins modernes ?

- Une nouvelle génération
- Préparer l'arrivée des blockchains qui trouvent leur créneau, mais le saut est "quantique"…
    - La question est de savoir si les Dapps présentent un intérêt ou pas
- Internet doit être compatible avec les modes asynchrones et synchrones
- Une plate-forme mieux sécurisée : quadrature du cercle
- Un support de transactions, comme ont pu l'être les mainframes et les réseaux propriétaires : distinction entre transactionnd lourd et distribué
- Une véritable plate-forme d'activités qui va nécessiter que l'on puisse apporter la preuve des opérations : banque, médecine, logistique industrielle
- Doit être conforme aux besoins IoT : transport cellulaire et basse consommation

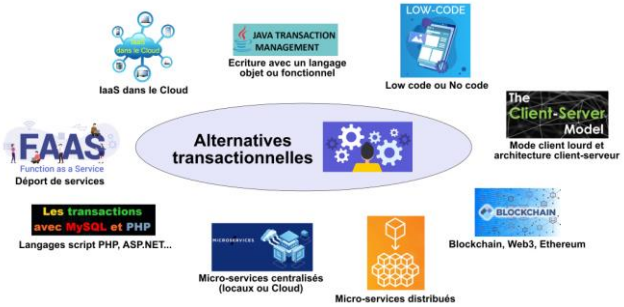
L'avenir du transactionnel classique

- Transaction : un bloc de code qui correspond à une fonction métier de forte granularité
  - Généralement peu de données, mais des traitements lourds
  - Classiquement 10 ou 20 000 lignes de code
  - Sur mainframe, servie par des services techniques, qui permettent au développeur de se concentrer sur la logique métier
- Aucune raison que cela change
  - Les mainframes "logiques" après portage sur des serveurs lourds Unix ou Linux ne seront pas remis en cause
  - Les mainframes physiques vont progressivement disparaître
  - Chez IBM, Ils ne représentent que 3 % du CA matériel installé… mais 10 % des services associés
- 500 milliards de lignes Cobol
  - Difficiles à reprendre : c'est une illusion
  - Les contraintes d'écriture : documentation, simplicité, architecture… n'ont pas toujours été respectées
  - Usage des Perform et Go To en Cobol
  - Architectures monolitiques impossibles à transcrire dans un contexte distribué : aucun intérêt
  - Manque de développeurs
  - L'essentiel des applications seront réécrites, mais pas transcrites
  - Le Web3 et Ethereum ne sont pas concernés, sauf pour des extensions spécifiques et "indépendantes"
- Les nouveaux transactionnels lourds sont des projets menés avec les API et langages modernes (Java, C#, fonctionnel…), qui nécessitent beaucoup de technicité.

```
000001 IDENTIFICATION DIVISION.
000006 PROGRAM-ID.                TESTPROG.
000095 ENVIRONMENT DIVISION.
000096 CONFIGURATION SECTION.
000103 DATA DIVISION.
000104 WORKING-STORAGE SECTION.
001667 01  TESTPROG-STRUCT.
001735                20  ARRAY-NAME    OCCURS 10.
001736                    25  RUECK-BUE       PIC S9(03) COMP-3.
001737                    25  RUECK-BKL       PIC 9(02).
001738                    25  RUECK-BS        PIC 9(03).
001739                    25  RUECK-SF        PIC X(12).
001737 01  baseaddr        PIC 9(10).
001737 01  thisaddr        PIC 9(10).
001737 01  difference      PIC 9(10).
001740
002211 PROCEDURE DIVISION.
004020     INITIALIZE  TESTPROG-STRUCT

           move address of TESTPROG-STRUCT to baseaddr
           display 'address of TESTPROG-STRUCT = ' baseaddr

           move address of RUECK-BKL(1) to thisaddr
           compute difference = thisaddr - baseaddr
           display 'offsetof(ARRAY-NAME,RUECK-BKL) = ' difference

           move address of RUECK-BS(1) to thisaddr
           compute difference = thisaddr - baseaddr
           display 'offsetof(ARRAY-NAME,RUECK-BS) = ' difference

           move address of RUECK-SF(1) to thisaddr
           compute difference = thisaddr - baseaddr
           display 'offsetof(ARRAY-NAME,RUECK-SF) = ' difference

           move address of RUECK-BUE(2) to thisaddr
           compute difference = thisaddr - baseaddr
           display 'sizeof(ARRAY-NAME(1)) = ' difference

           goback
```


Les alternatives au nouveau transactionnel

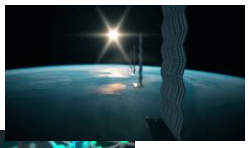
- Solution de facilité
  - IaaS dans le Cloud : forme moderne d'infogérance, on se débarrasse de tout…
- Réécriture en objet Java, C# ou autre
  - Les API batch et transactionnelles se sont fait attendre
  - Ce n'est pas leur cœur de métier et manquent d'efficacité
- Mini-applications non distribuées, locales ou Cloud, écrites avec un langage script : VBNet, low code… : fortes limitations fonctionnelles
- Transactionnel avec client lourd Windows ou Linux : difficiles à maintenir, complexité redoutable
- FaaS, en déportant les composants où sont concentrés les traitements les plus lourds
- Transcription Web PHP, ASP.NET, en s'appuyant sur des API dédiées transactionnelles : solution la plus logique, avec client Web
- Ecriture sous forme de micro-services centralisés (local ou Cloud) avec réutilisation des composants : suicidaire pour des applications généralement synchrones et traitements lourds
- Micro-services urbanisés et distribués : aucun intérêt, autre que de "complexifier" la conception des applications
- Mode blockchain, type Ethereum, à condition que le besoin existe et que les services techniques soient disponibles



La convergence technique

Permet d'envisager le transactionnel Web3

- Réseaux satellitaires à basse altitude avec une résilience très courte (entre 10 et 50 ms), qui assure à Internet une connectivité planétaire : autour de 2 milliards d'usagers "utiles"
- Des vitesses élevées grâce au déploiement des fibres optiques sous-marines et les générations cellulaires 5 et 6G.
- Des mécanismes nouveaux de protection en termes de confidentialité et d'intégrité : chiffrement quantique…
- Une puissance de calcul locale et dans le Cloud en croissance exponentielle.
- Des standards d'architectures distribuées DApps.
- Des moyens pour garantir la "vérité" transactionnelle par des contrats intelligents.
- Des moyens de paiement indépendants, en lien avec les monnaies actuelles.



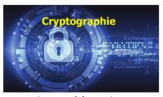
Les besoins nouveaux (Web3) :

Blockchain, décentralisation, monnaies cryptographiques, dématérialisation, sécurité

Distribution des fichiers dans une infrastructure décentralisée où chaque fichier est désigné par son contenu (caractérisé par son hash).

Web3 a généralement besoin d'une centrale d'identification qui puisse attribuer une identité aux différents acteurs susceptibles d'intervenir.

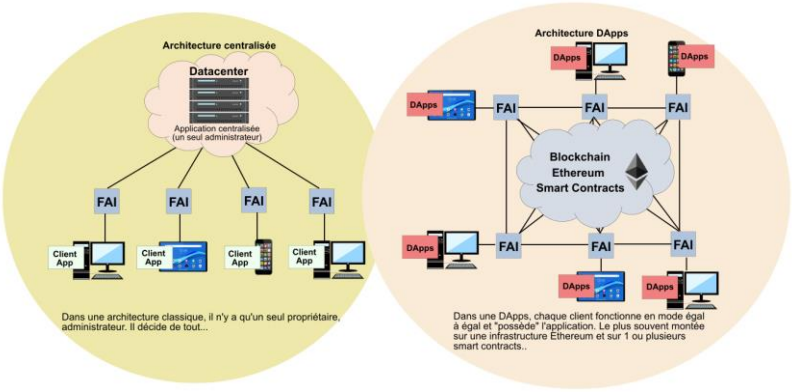
La distribution des applications (les Dapps) accompagne celle des fichiers. Dans ce contexte, chaque noeud est susceptible d'exercer un rôle actif dans l'exécution.



La cryptographie est une composante incontournable de Web3, que l'on retrouve dans la Blockchain, les techniques d'authentification et de NFT, les actifs non fongibles.

Les Blockchains sont très présentes dans le périmètre des Web3, le plus souvent à travers les "smart contracts" qui régissent les conditions de fonctionnement des acteurs de l'infrastructure.

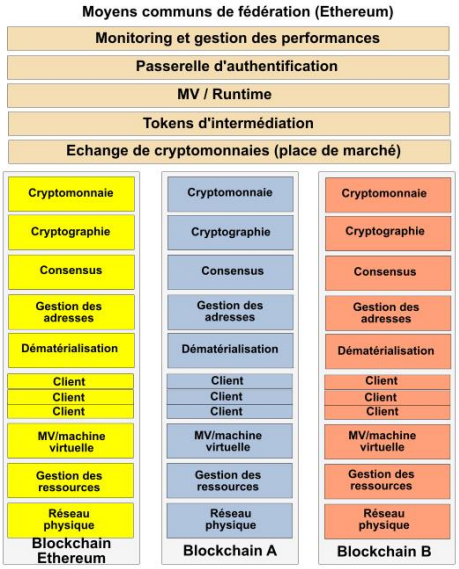
Les DApps : Applications distribuées



- DApps : applications distribuées qui fonctionnent sur un pied d'égalité entre des "ayant droits" et pas sous contrôle d'une entité unique.
- Applications réparties sur un grand nombre de serveurs et postes de travail, toute modification devant être approuvée par l'ensemble de la communauté, qui en est "propriétaire".
- Une DApp est généralement fondée sur un ou plusieurs "smart contracts", qui décrivent les mécanismes fonctionnels de l'application, avec une interface d'usage, un modèle distribué de stockage, un protocole de communication "peer to peer" et un système décentralisé de résolution de noms.

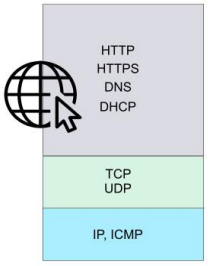
Le rôle possible d'Ethereum

- Fédérateur transactionnel d'applications décentralisées Blockchain.
- A terme, il y aura des milliers de blockchains, privées ou publiques, qui toutes auront pour finalité de gérer, pour "qui de droit", des vérités dans toutes leur diversité.
- Chaque blockchain peut être indépendante, avec des piles de services différents, voire incompatibles. C'est la certitude d'une pagaïe monstre, synonyme de perte d'efficacité…et d'abandon. D'où la fédération…
- Comme on ne pourra pas empêcher le besoin d'indépendance, le mieux sera de constituer une pile qui puisse agir indifféremment à différents niveaux, jusqu'à l'aspect applicatif qui "portera" leur spécificité : monnaie cryptographique, gestion de valeurs, tangibles ou non.
- En 2023, seul Ethereum de Vitalik Buterin peut jouer ce rôle.
- Ce qui ne préjuge en rien des futurs intervenants.
- Les entreprises doivent analyser l'intérêt de ces architectures : POC, essais pilotes, compétence à acquérir, réflexion sur les applications possibles.

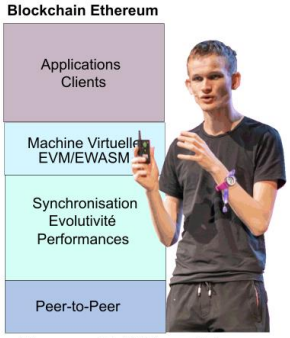


Une nouvelle référence de services

Le modèle OSI, 50 ans déjà...
(Hubert Zimmermann)



1983 : les briques Internet



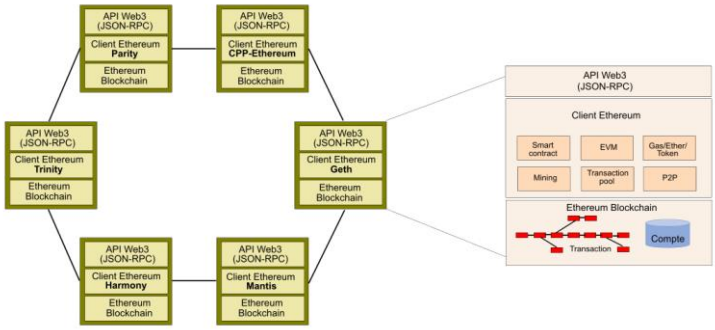
Ethereum et le XXI ème siècle : le futur
(Vitalik Buterin)

Les services Ethereum, dont certains pourraient constituer les briques communes à l'ensemble des applications de blockhain, sont centrés autour des réseaux "peer to peer", de la gestion du consensus répartie sur plusieurs niveaux et des clients dont le code s'exécute dans une machine virtuelle spécifique. A ces services dédiés Blockchain, il faut ajouter ceux plus classiques de monitoring et d'administration, qui lui viennent d'Hyperledger, issu de la mouvance Linux.

Architecture et constituants d'Ethereum

Au-delà des monnaies

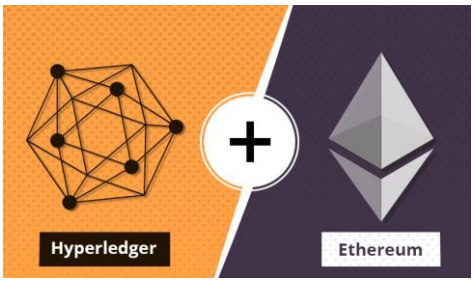
- Communication : réseau P2P ("Peer to Peer" ou "égal à égal"), chaque "peer" est un nœud appartenant à l'infrastructure.
  - Différentes configurations possibles : P2P centralisé ou chaque nœud est connecté à une autorité centrale.
  - P2P décentralisé, dans lequel l'autorité est elle-même constituée de plusieurs nœuds.
  - Ethereum comporte des services de "consensus" pour mettre à jour les nœuds, scindés en deux parties : "sharding" et synchronisation.
  - Le client, porté par chacun des nœuds, peut être "full", léger ou dédié à l'archivage.
    - Le mode "full" assure la gestion des données de la blockchain et prend sa part dans la validation des blocs et des transactions, le suivi des états, etc.
    - Le mode léger est destiné aux équipements qui ne sont pas armés pour stocker des volumes de données importants.



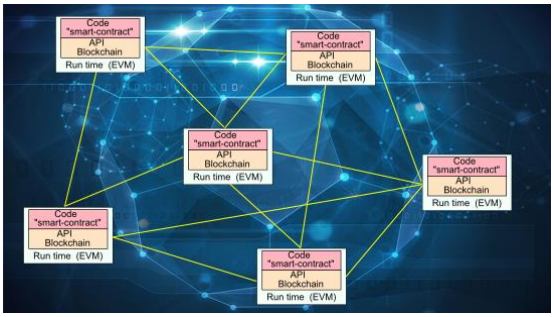
Architecture et constituants d'Ethereum

Au-delà des monnaies

- Applications : Ethereum s'appuie sur une gamme diversifiée de clients, développés par des tiers, comme Geth, le plus populaire avec 80 % des installations, écrit en Go, Parity (Rust), OpenEthereum (Rust), Nethermind (C#), Besu (Java), Trinity (Python), voire Erigon (Go).
- Les clients exécutent le code blockchain d'Ethereum, dans une machine virtuelle spécifique, un "run time" EVM (dont les "smart contracts"). Avec Ethereum 2, EVM est remplacé par EWASM, basé sur la technique du "Web Assembly" : plus performant et ouvert à d'autres langages que Solidity ou Vyper, tels que C++, C, JavaScript, Rust, etc.
- Il manque une couche de services génériques, qu'il trouvera dans Hyperledger, une DLT ("Distributed Ledger Technology") concurrente, issue de la mouvance Linux, dont il s'est rapproché pour rationaliser et mutualiser ses efforts de recherche et développement.
- Authentification des acteurs, administrateurs, utilisateurs et chargés des mise à jour de blocs
- Chiffrement des données confidentielles, autorisations, habilitations, etc.
- Certains services existaient déjà, mais Hyperledger les améliore fortement.



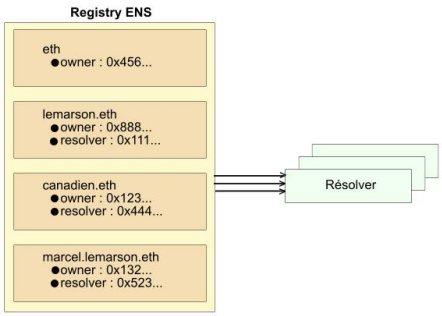
Les "smart contracts" avec Ethereum



- Les "smart contracts" permettent de programmer une relation de gré à gré entre deux acteurs.
- L'application est développée via un serveur Web indépendant, sur lequel on effectue les opérations d'édition et de compilation, puis de diffusion.
- Une fois le code écrit et testé sur une plate-forme concrète, qui peut être une Blockchain existante, il faudra le déployer dans les nœuds concernés par le « smart contract ». Opération qui se fera par une simple commande sur la plate-forme de développement, qui diffusera le code sur les comptes identifiés par une adresse.
  - Ex de l'outil Truffle, une suite complète, qui se charge, entre-autre, de la compilation solc, l'utilitaire le plus employé dans le monde Ethereum (commande truffle compile).
- Le code lui-même possède un identifiant, chaque contrat étant « localisé » de cette manière : 0xba0b295669a9fd37d5f28d9ec85e40f4cb697bae, par exemple.
- Une application est donc constituée de deux blocs : le code proprement dit et l'API Blockchain, sur laquelle il est posé. C'est cette API qui s'assure de la validité des données véhiculées et donc de la « vérité" portée par les nœuds.
- Deux familles de langages orientés contrats :
  - Dédiés à une blockchain spécifique (Solidity)
  - Ceux dont le code est compilé, pour produire un code exécutable sur une autre Blockchains.

Un système de résolution de noms

- ENS (Ethereum Name Service) est une architecture décentralisée qui traduit les adresses de sites, de portefeuilles crypto, les références des transactions et les empreintes numériques en noms de domaines compréhensibles.
- Les noms sont enregistrés sur la blockchain Ethereum (ex : l'adresse Ethereum 0x913a8f37c005cfF880e83a0Bc19A180b8e7d2c16 peut être renommée en une adresse lisible, lemarson.eth).
- Deux composants : registre et résolveurs.
  - **Le registre** est un *smart contract*, qui tient à jour la liste de tous les domaines et sous-domaines enregistrés et conserve certaines informations : propriétaire et résolveur. Le registre comprend aussi un *registrar*, qui permet d'attribuer les sous-domaines et spécifie certaines règles.
  - **Les résolveurs** sont responsables du processus de traduction de suites de chiffres en nom compréhensibles.
- Une procédure de résolution se déroule en deux étapes : demander au registre qui est responsable du nom de domaine et demander au résolveur la réponse à la requête.
- La principale différence avec les DNS est que l'ENS vend les noms de domaines sous la forme de NFT. Les jetons spéciaux ERC-721 sont utilisés par la blockchain de l'Ethereum et servent de certificats de propriété.
- La vente sous forme de NFT induit que les noms de domaines soient mis en vente sur des plateformes spéciales (OpenSea…). Une fois un nom de domaine NFT acheté, il est transférable à un autre utilisateur ou sur un autre wallet, comme un NFT classique.



Des réalisations Ethereum

- 2 700 applications Dapps recensées sur Ethereum (2022)
- Domaines souvent très techniques d'infrastructures pour la gestion des monnaies
- Encore peu d'applications de gestion classique transactionnelle

Des domaines qui se diversifient

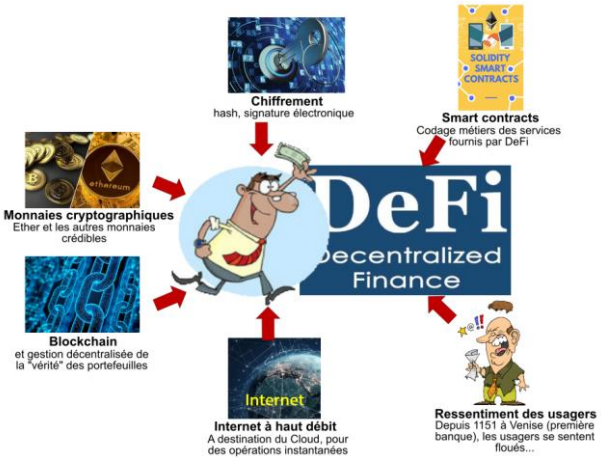
- Santé
- Logistique industrielle
- Contrats
- Gestion des copyright et droits associés
- Elections
- IoT
- Gestion des actifs financiers
- Propriétés et cadastres
- Publicité
- …

Avantages et inconvénients

- Présent depuis 2015 sur le créneau.
- Une réputation précoce de qualité et d'originalité.
- Une communauté très large et active.
- Des possibilités de développement très vastes.
- Une forte avance dans la création d'applications décentralisées et de smart contrats.
- Une couverture fonctionnelle qui va plus loin que la simple mise en œuvre d'une cryptomonnaie.
- La bonne réputation de Vitalik Buterin, fondateur de la monnaie Ether (fortune personnelle, au-delà d'1 milliard $...à 29 ans), 2ème acteur du marché derrière les bitcoins.
- L'image de marque reconnue auprès de plusieurs grands noms de l'économie et des finances (Axa, Commonwealth Bank of Australia, JP Morgan, Amazon, Microsoft Azure, le chinois Ant Financial, BNP Paribas, Citigroup…)

- Des failles de sécurité récurrentes.
- La réalité financière et industrielle d'Ethereum, qui peut être en "trompe l'œil" (syndrome de Docker).
- L'extrême volatilité des cryptomonnaies, malgré la capitalisation boursière supérieure à 210 milliards $.
- L'apprentissage du langage Solidity peut être jugé complexe. Des concurrents proposent le même service sans nécessiter un nouveau langage (sans doute la meilleure solution). Ethereum 2 a réglé ce problème.
- Les "smart contracts" de première génération émises par Ethereum présentent de nombreux problèmes d'évolutivité et des performances parfois médiocres. La situation s'est améliorée avec un meilleur algorithme de consensus : PoW à PoS.
- Ethereum n'est pas (encore) reconnu comme standard par les grands organismes mondiaux et la popularité des ERC-20 pour les "tokens" fongibles, ERC-721 pour les "tokens" non fongibles et ERC-1155 le compromis des deux précédents, n'en fait pas (encore) des normes officielles.
- Dépendance psychologique vis-à-vis de Vitalik Buterin.
- Tout reste à faire pour convaincre les clients à investir dans les Dapps : le portefeuille d'applications est trop orienté infrastructures, il doit se diversifier vers la gestion classique.

La crédibilité d'Ethereum portée par les DeFi



- Les applications financières vont tirer le marché (DeFi) : il en existe des centaines qui professionnalisent le domaine et "rassurent" les clients.
- DeFi est un ensemble de plates-formes indépendantes (ou non) fondées sur une blockchain, dans lesquelles les usagers communiquent via Internet pour accéder aux services financiers proposés.
- Chaque plate-forme constitue un monde en soi, avec sa communauté d'usagers, à qui elle assure des services sous une forme plus démocratique et égalitaire.
- Grâce aux blockchains, les opérations financières donnent lieu à des mises à jour dans chacun des portefeuilles, tout étant chiffré et validé par des hashs (signature électronique), les participants à une plate-forme donnée, disposant de l'ensemble des opérations effectuées depuis le démarrage. Avec toujours la même certitude qu'il sera (en principe) impossible de modifier un ou plusieurs de ces portefeuilles, pour les détourner.
- Les opérations peuvent se faire sur une cryptomonnaie avec des actifs qui auront une correspondance ou non avec les monnaies réelles : dollars, euros, yens, etc.
- Toutes ces couches contribueront à l'élaboration d'une sorte de système d'information financière, dont la principale caractéristique sera de ne plus dépendre des acteurs habituels de la finance.

Ethereum, le transactionnel Internet de demain

Nos prochains webinaires

21 avril 2023	**Backup et restauration des datacenters**
28 avril 2023	**Pourquoi l'IA est-elle stupide ? Elle nous imite**
5 mai 2023	**L'hyperautomatisation : les temps modernes du TI**
12 mai 2023	**Quand la biométrie sort des sentiers battus**
26 mai 2023	**Productivité, il n'y a pas qu'Office. Ah, bon ?**
2 juin 2023	**Les grandes utopies du TI : capitaliser sur nos erreurs**
9 juin 2023	**Les transports du futur : verts et sans pilotes**
16 juin 2023	**Sécurité : les reproches faits à la suite TCP/IP**
23 juin 2023	**La fédération d'identités : "you will never walk alone..."**
30 juin 2023	**Les grandes figures du TI... celles dont on parle moins**

claude@lemarson.com
https://www.lemarson.com