

afnic

Que faire de rigolo avec la blockchain ?

Stéphane Bortzmeyer

AFNIC

bortzmeyer@nic.fr

afnic

Que faire de rigolo avec la blockchain ?

Stéphane Bortzmeyer

AFNIC

bortzmeyer@nic.fr

Plan du tutoriel

- 1 Introduction
- 2 Bitcoin
- 3 Ethereum
- 4 Usages
- 5 Les contrats
- 6 Le langage Solidity
- 7 Registre de noms de domaine
- 8 Conclusion

Le buzzword du moment



Le thème fédérateur de la Cloud Week Paris 2016
Les usages et la valorisation des données



VISIONARIES' CONFERENCE

EUROCLOUD TROPHIES

ÉTATS GÉNÉRAUX OF THE CLOUD

FULL WEEK PROGRAM

REGISTER

La Blockchain a-t-elle les moyens de ses ambitions ? (Forum Atena)



Buzz du moment, la Blockchain, promet de combler un manque d'Internet dans les échanges entre pairs : la confiance.

Le buzz est-il un effet d'« Inflated expectations » ? La désillusion promise par le « Hype cycle » de Gartner va-t-elle suivre ?

Le 05 juillet dans le cadre de la Cloud Week, Forum ATENA s'entoure des grands experts du domaine pour sonder sur la belle mécanique du Blockchain autour de trois volets : pédagogie de la Blockchain, ambitions de la Blockchain et



afnic

Le buzzword du moment

- Depuis début 2016, la blockchain est partout,



Le thème fédérateur de la Cloud Week Paris 2016
Les usages et la valorisation des données



VISIONARIES' CONFERENCE

EUROCLOUD TROPHIES

ÉTATS GÉNÉRAUX OF THE CLOUD

FULL WEEK PROGRAM

REGISTER

La Blockchain a-t-elle les moyens de ses ambitions ? (Forum Atena)



Buzz du moment, la Blockchain, promet de combler un manque d'Internet dans les échanges entre pairs : la confiance.

Le buzz est-il un effet d'« Inflated expectations » ? La désillusion promise par le « Hype cycle » de Gartner va-t-elle suivre ?

Le 05 juillet dans le cadre de la Cloud Week, Forum ATENA s'entoure des grands experts du domaine pour sonder sur la belle mécanique du Blockchain autour de trois volets : pédagogie de la Blockchain, ambitions de la Blockchain et



afnic

Le buzzword du moment

- Depuis début 2016, la blockchain est partout,
- Toutes les étapes du Hype Cycle (de « la blockchain va guérir le cancer » à « la blockchain est responsable de la faim dans le monde »).



Le thème fédérateur de la Cloud Week Paris 2016

Les usages et la valorisation des données



VISIONARIES' CONFERENCE

EUROCLOUD TROPHIES

ÉTATS GÉNÉRAUX OF THE CLOUD

FULL WEEK PROGRAM

REGISTER

La Blockchain a-t-elle les moyens de ses ambitions ? (Forum Atena)



Buzz du moment, la Blockchain, promet de combler un manque d'Internet dans les échanges entre pairs : la confiance.

Le buzz est-il un effet d'« Inflated expectations » ? La désillusion promise par le « Hype cycle » de Gartner va-t-elle suivre ?

Le 05 juillet dans le cadre de la Cloud Week, Forum ATENA s'entoure des grands experts du domaine pour sonder sur la belle mécanique du Blockchain autour de trois volets : pédagogie de la Blockchain, ambitions de la Blockchain et



afnic

Le principe

Le principe

- 1 Des transactions cryptographiquement signées faisant passer d'un état à l'autre,

Le principe

- ① Des transactions cryptographiquement signées faisant passer d'un état à l'autre,
- ② Les transactions regroupées dans un **bloc**,

Le principe

- ① Des transactions cryptographiquement signées faisant passer d'un état à l'autre,
- ② Les transactions regroupées dans un **bloc**,
- ③ Les blocs sont chaînés (un bloc contient le condensat du bloc précédent)

Le principe

- ① Des transactions cryptographiquement signées faisant passer d'un état à l'autre,
- ② Les transactions regroupées dans un **bloc**,
- ③ Les blocs sont chaînés
- ④ La construction de la chaîne est répartie : chacun faisant tourner le même algorithme, on peut avoir un **consensus**,

Le principe

- ① Des transactions cryptographiquement signées faisant passer d'un état à l'autre,
- ② Les transactions regroupées dans un **bloc**,
- ③ Les blocs sont chaînés
- ④ La construction de la chaîne est répartie,
- ⑤ Le tout est **public**
 - ① Cela permet donc une validation **publique** du contenu de la chaîne,

Le principe

- ① Des transactions cryptographiquement signées faisant passer d'un état à l'autre,
- ② Les transactions regroupées dans un **bloc**,
- ③ Les blocs sont chaînés
- ④ La construction de la chaîne est répartie,
- ⑤ Le tout est **public**
 - ① Cela permet donc une validation **publique** du contenu de la chaîne,
 - ② On a donc enfin une structure de données construite en pair-à-pair et que tout le monde peut vérifier.

Attention aux gougnafiers

Attention aux gougnafiers

- Si c'est pair-à-pair, n'importe qui peut écrire sur la chaîne ?
Comment empêcher qu'elle ne grossisse démesurément ?

Attention aux gougnafiers

- Si c'est pair-à-pair, n'importe qui peut écrire sur la chaîne ?
Comment empêcher qu'elle ne grossisse démesurément ?
- Deux grandes classes de solutions :

Attention aux gougnafiers

- Si c'est pair-à-pair, n'importe qui peut écrire sur la chaîne ?
Comment empêcher qu'elle ne grossisse démesurément ?
- Deux grandes classes de solutions :
 - ① Preuve de travail (*proof-of-work*)

Attention aux gougnafiers

- Si c'est pair-à-pair, n'importe qui peut écrire sur la chaîne ?
Comment empêcher qu'elle ne grossisse démesurément ?
- Deux grandes classes de solutions :
 - ① Preuve de travail (*proof-of-work*)
 - ② Preuve de participation (*proof-of-stake*)

Attention aux gougnafiers

- Si c'est pair-à-pair, n'importe qui peut écrire sur la chaîne ?
Comment empêcher qu'elle ne grossisse démesurément ?
- Deux grandes classes de solutions :
 - ① Preuve de travail (*proof-of-work*)
 - ② Preuve de participation (*proof-of-stake*)
- Preuve de travail : résoudre un puzzle informatique, par exemple (pas très écologiste)

Attention aux gougnafiers

- Si c'est pair-à-pair, n'importe qui peut écrire sur la chaîne ?
Comment empêcher qu'elle ne grossisse démesurément ?
- Deux grandes classes de solutions :
 - 1 Preuve de travail (*proof-of-work*)
 - 2 Preuve de participation (*proof-of-stake*)
- Preuve de travail : résoudre un puzzle informatique, par exemple (pas très écologiste)
- Preuve de participation : un vote proportionnel à l'implication (l'argent, dans le cas d'une cybermonnaie)

Attention aux gougnafiers

- Si c'est pair-à-pair, n'importe qui peut écrire sur la chaîne ? Comment empêcher qu'elle ne grossisse démesurément ?
- Deux grandes classes de solutions :
 - ① Preuve de travail (*proof-of-work*)
 - ② Preuve de participation (*proof-of-stake*)
- Preuve de travail : résoudre un puzzle informatique, par exemple (pas très écologiste)
- Preuve de participation : un vote proportionnel à l'implication (l'argent, dans le cas d'une cybermonnaie)
- Les deux classes ont un problème commun : « l'attaque des 51 % » (dictature de la majorité)

Un peu d'histoire

Un peu d'histoire

- 1 Jusqu'en 2008 : quelques articles sans impact (exemple, 2005-2008, Nick Szabo « *Bit Gold* »)

Un peu d'histoire

- 1 Jusqu'en 2008 : quelques articles sans impact
- 2 2008-2009 Publication de l'article de Satoshi Nakamoto « *Bitcoin : A Peer-to-Peer Electronic Cash System* » La chaîne de blocs démarre réellement

Un peu d'histoire

- 1 Jusqu'en 2008 : quelques articles sans impact
- 2 2008-2009 Publication de l'article de Satoshi Nakamoto
« *Bitcoin : A Peer-to-Peer Electronic Cash System* »
- 3 2009, code source de Bitcoin publié, **genèse** (création du premier bloc)

Un peu d'histoire

- 1 Jusqu'en 2008 : quelques articles sans impact
- 2 2008-2009 Publication de l'article de Satoshi Nakamoto « *Bitcoin : A Peer-to-Peer Electronic Cash System* »
- 3 2009, code source de Bitcoin publié, **genèse** (création du premier bloc)
- 4 2011, sortie de Namecoin, *fork* de Bitcoin pour enregistrer des noms. 2013, sortie de Twister (*microblogging*, avec la chaîne de blocs pour avoir des noms uniques)

Un peu d'histoire

- 1 Jusqu'en 2008 : quelques articles sans impact
- 2 2008-2009 Publication de l'article de Satoshi Nakamoto « *Bitcoin : A Peer-to-Peer Electronic Cash System* »
- 3 2009, code source de Bitcoin publié, **genèse** (création du premier bloc)
- 4 2011, sortie de Namecoin, *fork* de Bitcoin pour enregistrer des noms. 2013, sortie de Twister
- 5 2013, NXT (monnaie sans minage) Ce n'est **pas** un *fork* de Bitcoin

Un peu d'histoire

- 1 Jusqu'en 2008 : quelques articles sans impact
- 2 2008-2009 Publication de l'article de Satoshi Nakamoto « *Bitcoin : A Peer-to-Peer Electronic Cash System* »
- 3 2009, code source de Bitcoin publié, **genèse** (création du premier bloc)
- 4 2011, sortie de Namecoin, *fork* de Bitcoin pour enregistrer des noms. 2013, sortie de Twister
- 5 2013, NXT
- 6 2013, premier article sur Ethereum (chaîne stockant des programmes et pas juste des données) , 2015 genèse d'Ethereum

Un peu d'histoire

- 1 Jusqu'en 2008 : quelques articles sans impact
- 2 2008-2009 Publication de l'article de Satoshi Nakamoto « *Bitcoin : A Peer-to-Peer Electronic Cash System* »
- 3 2009, code source de Bitcoin publié, **genèse** (création du premier bloc)
- 4 2011, sortie de Namecoin, *fork* de Bitcoin pour enregistrer des noms. 2013, sortie de Twister
- 5 2013, NXT
- 6 2013, premier article sur Ethereum, 2015 genèse d'Ethereum
- 7 Début 2016 : début du *hype*

Un peu d'histoire

- 1 Jusqu'en 2008 : quelques articles sans impact
- 2 2008-2009 Publication de l'article de Satoshi Nakamoto « *Bitcoin : A Peer-to-Peer Electronic Cash System* »
- 3 2009, code source de Bitcoin publié, **genèse** (création du premier bloc)
- 4 2011, sortie de Namecoin, *fork* de Bitcoin pour enregistrer des noms. 2013, sortie de Twister
- 5 2013, NXT
- 6 2013, premier article sur Ethereum, 2015 genèse d'Ethereum
- 7 Début 2016 : début du *hype*
- 8 Mi-2016 : plein de chaînes différentes, plein de développeurs, plein de chercheurs, plein de start-ups

Plan du tutoriel

- 1 Introduction
- 2 Bitcoin
- 3 Ethereum
- 4 Usages
- 5 Les contrats
- 6 Le langage Solidity
- 7 Registre de noms de domaine
- 8 Conclusion

La cybermonnaie

La cybermonnaie

- Première chaîne de blocs opérationnelle,

La cybermonnaie

- Première chaîne de blocs opérationnelle,
- Toujours en tête, en valorisation en € (10 milliards), ou bien en condensats/s,

La cybermonnaie

- Première chaîne de blocs opérationnelle,
- Toujours en tête, en valorisation en € (10 milliards), ou bien en condensats/s,
- Spécialisé dans la monnaie,

La cybermonnaie

- Première chaîne de blocs opérationnelle,
- Toujours en tête, en valorisation en € (10 milliards), ou bien en condensats/s,
- Spécialisé dans la monnaie,
- La chaîne fait aujourd'hui dans les 80 Go.

La chaine Bitcoin en temps réel

Blockchain Luxembourg S.A.R.L [LU] <https://blockchain.info>



BLOCKCHAIN
info

Home

Charts

Stats

Markets

API

Wallet

English ▾

Home Welcome to Blockchain

[More.](#)

Height	Age	Transactions	Total Sent	Relayed By	Size (kB)
419561	4 minutes	98	833.66422644 BTC	Kano CKPool	31
419560	4 minutes	316	2,078.15677391 BTC	BitFury	169
419559	6 minutes	1433	11,326.01 BTC	BitFury	830.93
419558	10 minutes	2838	34,819.72 BTC	AntPool	998.08
419557	32 minutes	736	8,972.17 BTC	Slush	317.45
419556	36 minutes	608	11,992.19 BTC	BTCC Pool	361.14

Latest Transactions

981be2ee20d6654bd67e0c75b...

< 1 minute

3.29866991 BTC

776f639afc2a7e262a292463b...

< 1 minute

0.0012671 BTC

9ac47d163696a5cd922c8eeb8...

< 1 minute

0.00103711 BTC

Search

You may enter a block height, address, block hash, transaction hash, hash160, or ipv4 address...

Search

NEWS

axmic

10 / 5

Minage et preuve de travail

Minage et preuve de travail

- Bitcoin utilise un système de preuve de travail : pour insérer un bloc, il faut le remplir avec une valeur qui donne un condensat ayant certaines caractéristiques

Minage et preuve de travail

- Bitcoin utilise un système de preuve de travail
- On est récompensé en bitcoins : c'est le **minage**

Minage et preuve de travail

- Bitcoin utilise un système de preuve de travail
- On est récompensé en bitcoins : c'est le **minage**
- Bitcoin est donc une « hashocratie » :-)

Minage et preuve de travail

- Bitcoin utilise un système de preuve de travail
- On est récompensé en bitcoins : c'est le **minage**
- Bitcoin est donc une « hashocratie » :-)
- La preuve de travail Bitcoin se fait avec des ASIC : aucune chance avec un CPU normal (ou même un GPU)

Fric

- Utilisation principale de Bitcoin : stocker et transférer des bitcoins

Fric

- Utilisation principale de Bitcoin : stocker et transférer des bitcoins
- Ils peuvent s'échanger contre les monnaies *fiat* (celles des États) sur des **places de marché** comme Paymium ou Kraken, ou dans un DAB comme celui de Vancouver

- Utilisation principale de Bitcoin : stocker et transférer des bitcoins
- Ils peuvent s'échanger contre les monnaies *fiat* (celles des États) sur des **places de marché** comme Paymium ou Kraken, ou dans un DAB comme celui de Vancouver
- Autres usages du bitcoin : dons à l'EFF, acheter un nom de domaine chez Gandi...

Bitcoin pour l'utilisateur

Bitcoin pour l'utilisateur

- Les transactions sont signés. Chacun doit donc avoir une bi-clé. Le condensat de la clé publique est l'**adresse**.

Bitcoin pour l'utilisateur

- Les transactions sont signés. Chacun doit donc avoir une bi-clé. Le condensat de la clé publique est l'**adresse**.
- M. Michu n'a en général pas un nœud complet. Il utilise un « client léger » (*wallet*) et fait donc confiance à un serveur.

Bitcoin pour l'utilisateur

- Les transactions sont signés. Chacun doit donc avoir une bi-clé. Le condensat de la clé publique est l'**adresse**.
- M. Michu n'a en général pas un nœud complet. Il utilise un « client léger » (*wallet*) et fait donc confiance à un serveur.

Merci d'envoyer des bitcoins à

1HtNJ6ZFUc9yu9u2qAwB4tGdGwPQasQGax



Un client Bitcoin

The screenshot shows a mobile application interface for a Bitcoin wallet. At the top, the status bar displays signal strength, 84% battery, and the time 11:37. The app header is labeled "Bitcoin" and includes a tag icon and a menu icon. The main display shows a balance of **mBTC 15.52** with a note **≈ EUR 9.39** and a QR code for scanning. A warning message states: "Vous devez sauvegarder votre portefeuille !" (You must back up your wallet!). Below this, a list of transactions is shown:

- 04/11/2015 16aq u8J7 L... + 14.50
- + 04/11/2015 16aq u8J7 L... + 14.50
Ce paiement a été renversé par son expéditeur.
- 22/10/2015 **Gandi SAS** - 59.58
≈ EUR - 14.19
4bDa9cQ84wUR5ZVYm2srV7 (via BitPay)
- 18/08/2014 14T4 2UBt u... + 60.70

Idées idiotes entendues

Idées idiotes entendues

- « On ne connaît pas Nakamoto »

Idées idiotes entendues

- « On ne connaît pas Nakamoto »
- « Ça ne durera pas »

Idées idiotes entendues

- « On ne connaît pas Nakamoto »
- « Ça ne durera pas »
- « C'est illégal »

Idées idiotes entendues

- « On ne connaît pas Nakamoto »
- « Ça ne durera pas »
- « C'est illégal »
- « C'est une monnaie virtuelle »

Idées idiotes entendues

- « On ne connaît pas Nakamoto »
- « Ça ne durera pas »
- « C'est illégal »
- « C'est une monnaie virtuelle »
- « Cela ne correspond pas à de la production réelle dans le vrai monde »

Idées idiotes entendues

- « On ne connaît pas Nakamoto »
- « Ça ne durera pas »
- « C'est illégal »
- « C'est une monnaie virtuelle »
- « Cela ne correspond pas à de la production réelle dans le vrai monde »
- « C'est anonyme »

Les explorateurs

Les explorateurs

- La chaîne de blocs est **publique**

Les explorateurs

- La chaîne de blocs est **publique**
- N'importe qui peut donc créer un **explorateur** qui va afficher la chaîne sous forme d'une jolie page Web

Les explorateurs

- La chaîne de blocs est **publique**
- N'importe qui peut donc créer un **explorateur** qui va afficher la chaîne sous forme d'une jolie page Web
- Attention, un explorateur public, c'est le *cloud* : il ne faut pas forcément lui faire confiance

Une transaction Bitcoin

Blockchain Luxembourg S.A.R.L [LU] <https://blockchain.info/tx/1253ceb4bb5870978f0cd9c94c3718e3bd1b8c200ad>



BLOCKCHAIN
info

[Home](#)

[Charts](#)

[Stats](#)

[Markets](#)

[API](#)

[Wallet](#)

English ▾

Transaction View information about a bitcoin transaction

1253ceb4bb5870978f0cd9c94c3718e3bd1b8c200ad20a68c12dc6b9e77d0444

1FrfMP6qHXsvdJWZtDjepdWrXx1197XSRe



1HtNJ6ZFUc9yu9u2qAwB4tGdGwPQasQGax

0.00077823 BTC

0.00077823 BTC

Summary

Size 191 (bytes)

Received Time 2016-06-18 13:28:39

Lock Time Block: 416259

Included In Blocks [417153](#) (2016-06-20 05:17:35 + 2,389 minutes)

Confirmations 2407 Confirmations

Relayed by IP [83.226.56.114 \(whois\)](#)

Inputs and Outputs

Total Input 0.00078015 BTC

Total Output 0.00077823 BTC

Fees 0.00000192 BTC

Estimated BTC Transacted 0.00077823 BTC

Scripts [Show scripts & coinbase](#)

afnic

17 / 5

Plan du tutoriel

- 1 Introduction
- 2 Bitcoin
- 3 Ethereum**
- 4 Usages
- 5 Les contrats
- 6 Le langage Solidity
- 7 Registre de noms de domaine
- 8 Conclusion

Les principes

Les principes

- Code fait de zéro,

Les principes

- Code fait de zéro,
- Chaîne **généraliste** : pas uniquement pour la monnaie, une plate-forme pour les développeurs,

Les principes

- Code fait de zéro,
- Chaîne **généraliste** : pas uniquement pour la monnaie, une plate-forme pour les développeurs,
- Séparation de la spécification et de la mise en œuvre,

Les principes

- Code fait de zéro,
- Chaîne **généraliste** : pas uniquement pour la monnaie, une plate-forme pour les développeurs,
- Séparation de la spécification et de la mise en œuvre,
- Ethereum a une monnaie, l'**ether** mais il a d'autres applications que la monnaie

La chaîne

```
> eth.getBlock(1000000)
{
  gasUsed: 50244,
  hash: "0x8e38b4dbf6b11fcc3b9dee84fb7986e29ca0a02cecd8977c161ff7333329
  parentHash: "0xb4fbadf8ea452b139718e2700dc1135cfc81145031c84b7ab27cd7
  transactions: ["0xea1093d492a1dcb1bef708f771a99a96ff05dcab81ca76c3194
  ...
}

> eth.getBlock(999999)
{
  gasUsed: 231000,
  hash: "0xb4fbadf8ea452b139718e2700dc1135cfc81145031c84b7ab27cd710394f
  parentHash: "0xd33c9dde9fff0ebaa6e71e8b26d2bda15ccf111c7af1b633698ac8
  transactions: ["0x22879e0bc9602fef59dc0602f9bc385f12632da5cb4eee4b813
  ...
}
```

Stocker du code dans la chaîne

Stocker du code dans la chaîne

- Bitcoin avait déjà un langage mais limité (volontairement, pour la sécurité)

Stocker du code dans la chaîne

- Bitcoin avait déjà un langage mais limité (volontairement, pour la sécurité)
- Pour faire autre chose, il fallait *forker* et modifier le langage (Namecoin, Twister)

Stocker du code dans la chaîne

- Bitcoin avait déjà un langage mais limité (volontairement, pour la sécurité)
- Pour faire autre chose, il fallait *forker* et modifier le langage (Namecoin, Twister)
- Ethereum a au contraire un **langage de Turing**

Stocker du code dans la chaîne

- Bitcoin avait déjà un langage mais limité (volontairement, pour la sécurité)
- Pour faire autre chose, il fallait *forker* et modifier le langage (Namecoin, Twister)
- Ethereum a au contraire un **langage de Turing**
- Langage du niveau d'un langage d'assemblage (EVM pour *Ethereum Virtual Machine*)

Stocker du code dans la chaîne

- Bitcoin avait déjà un langage mais limité (volontairement, pour la sécurité)
- Pour faire autre chose, il fallait *forker* et modifier le langage (Namecoin, Twister)
- Ethereum a au contraire un **langage de Turing**
- Langage du niveau d'un langage d'assemblage
- Les programmes sont exécutés par tous les nœuds de la chaîne

Stocker du code dans la chaîne

- Bitcoin avait déjà un langage mais limité (volontairement, pour la sécurité)
- Pour faire autre chose, il fallait *forker* et modifier le langage (Namecoin, Twister)
- Ethereum a au contraire un **langage de Turing**
- Langage du niveau d'un langage d'assemblage
- Les programmes sont exécutés par tous les nœuds de la chaîne
- Tout est donc possible : il « suffit » de programmer

Stocker du code dans la chaîne

- Bitcoin avait déjà un langage mais limité (volontairement, pour la sécurité)
- Pour faire autre chose, il fallait *forker* et modifier le langage (Namecoin, Twister)
- Ethereum a au contraire un **langage de Turing**
- Langage du niveau d'un langage d'assemblage
- Les programmes sont exécutés par tous les nœuds de la chaîne
- Tout est donc possible : il « suffit » de programmer
- Évidemment, conséquences en terme de sécurité et fiabilité

Stocker du code dans la chaîne

- Bitcoin avait déjà un langage mais limité (volontairement, pour la sécurité)
- Pour faire autre chose, il fallait *forker* et modifier le langage (Namecoin, Twister)
- Ethereum a au contraire un **langage de Turing**
- Langage du niveau d'un langage d'assemblage
- Les programmes sont exécutés par tous les nœuds de la chaîne
- Tout est donc possible : il « suffit » de programmer
- Évidemment, conséquences en terme de sécurité et fiabilité
- Une autre solution existe, les *sidechains* comme RootStock ou BlockStack (mixtes, la chaîne de Bitcoin comme référence, et la leur pour la souplesse)

Plan du tutoriel

- 1 Introduction
- 2 Bitcoin
- 3 Ethereum
- 4 Usages**
- 5 Les contrats
- 6 Le langage Solidity
- 7 Registre de noms de domaine
- 8 Conclusion

Ce qui est bien adapté à la chaîne de blocs

Tout ce qui a besoin de vérification publique, sans autorité centrale

- Registre (par exemple de noms)
- Cadastre
- Monnaie (notamment micro-paiements)

Ce qui n'est pas adapté à la chaîne de blocs

La chaîne est le plus lent et le plus cher calculateur du monde Inadaptée pour :

- Longs calculs (cryptographie...)
- Gros stockages (pas de vidéos dans la chaîne)

Plan du tutoriel

- 1 Introduction
- 2 Bitcoin
- 3 Ethereum
- 4 Usages
- 5 Les contrats**
- 6 Le langage Solidity
- 7 Registre de noms de domaine
- 8 Conclusion

Présentation

Présentation

- Un contrat (*smart contract* pour le marketing) est un programme,

Présentation

- Un contrat (*smart contract* pour le marketing) est un programme,
- Comme tout programme, il est écrit par un programmeur, puis compilé dans le langage de la machine virtuelle (EVM),

Présentation

- Un contrat (*smart contract* pour le marketing) est un programme,
- Comme tout programme, il est écrit par un programmeur, puis compilé dans le langage de la machine virtuelle (EVM),
- Il est exécuté par les nœuds de la chaîne (tous l'exécutent, et doivent trouver le même résultat).

Présentation

- Un contrat (*smart contract* pour le marketing) est un programme,
- Comme tout programme, il est écrit par un programmeur, puis compilé dans le langage de la machine virtuelle (EVM),
- Il est exécuté par les nœuds de la chaîne (tous l'exécutent, et doivent trouver le même résultat).
- Le contrat peut manipuler de l'argent (en recevoir, le transmettre. . .)

Sécurité des contrats

Sécurité des contrats

- Un contrat, c'est un programme

Sécurité des contrats

- Un contrat, c'est un programme
- Les logiciels ont des bogues

Sécurité des contrats

- Un contrat, c'est un programme
- Les logiciels ont des bogues
- Donc, les contrats ont des bogues

Sécurité des contrats

- Un contrat, c'est un programme
- Les logiciels ont des bogues
- Donc, les contrats ont des bogues
- Langages fonctionnels ? Vérifications formelles ?

Sécurité des contrats

- Un contrat, c'est un programme
- Les logiciels ont des bogues
- Donc, les contrats ont des bogues
- Langages fonctionnels ? Vérifications formelles ?
- Et les attaques par déni de service ? Protection par l'**essence** (qui propulse les contrats, et qu'il faut payer).

Pourquoi les contrats ne doivent pas être « smarts »

Pourquoi les contrats ne doivent pas être « smarts »

- Un contrat doit être sûr (ne pas contenir de bogues) :
problème difficile !

Pourquoi les contrats ne doivent pas être « smarts »

- Un contrat doit être sûr : problème difficile !
- Un contrat doit être vérifié par les utilisateurs (on n'envoie pas de l'argent à un contrat qu'on n'a pas vérifié).

Pourquoi les contrats ne doivent pas être « smarts »

- Un contrat doit être sûr : problème difficile !
- Un contrat doit être vérifié par les utilisateurs

Le contrat doit donc être **simple**. Un *dumb contract*.

Pourquoi les contrats ne doivent pas être « smarts »

- Un contrat doit être sûr : problème difficile !
- Un contrat doit être vérifié par les utilisateurs

Le contrat doit donc être **simple**. Un *dumb contract*. En prime, l'EVM exécute du code machine, pas le source que vous lisez. La vérification du code source ne suffit donc pas.

Le concept de DAO

DAO = *Decentralized Autonomous Organisation*, une entité stockée sur la chaîne (sous forme de contrats) et qui s'exécute automatiquement

Le concept de DAO

DAO = *Decentralized Autonomous Organisation*, une entité stockée sur la chaîne et qui s'exécute automatiquement

- Une « organisation » non enregistrée au RGS :-)

Le concept de DAO

DAO = *Decentralized Autonomous Organisation*, une entité stockée sur la chaîne et qui s'exécute automatiquement

- Une « organisation » non enregistrée au RGS :-)
- Sans intervention humaine au quotidien

Le concept de DAO

DAO = *Decentralized Autonomous Organisation*, une entité stockée sur la chaîne et qui s'exécute automatiquement

- Une « organisation » non enregistrée au RGS :-)
- Sans intervention humaine au quotidien
- Protégée des passions humaines et de l'arbitraire du pouvoir

Le concept de DAO

DAO = *Decentralized Autonomous Organisation*, une entité stockée sur la chaîne et qui s'exécute automatiquement

- Une « organisation » non enregistrée au RGS :-)
- Sans intervention humaine au quotidien
- Protégée des passions humaines et de l'arbitraire du pouvoir
- Mais si ça cafouille, quels recours ?

The DAO

The DAO

- The DAO est/était **une** DAO particulière

The DAO

- The DAO est/était **une** DAO particulière
- Un fond d'investissement : les investisseurs y mettent des ethers, des gens proposent des projets, les investisseurs votent

The DAO

- The DAO est/était **une** DAO particulière
- Un fond d'investissement : les investisseurs y mettent des ethers, des gens proposent des projets, les investisseurs votent
- The DAO a récolté l'équivalent de plus de 100 M€, le plus gros *crowdfunding* jamais fait

La crise de The DAO et les leçons

La crise de The DAO et les leçons

- Une bogue dans le contrat a permis à un voleur d'emporter un tiers des fonds,

La crise de The DAO et les leçons

- Une bogue dans le contrat a permis à un voleur d'emporter un tiers des fonds,
- Il s'agissait d'une bogue dans **un** contrat, pas une faille d'Ethereum,

La crise de The DAO et les leçons

- Une bogue dans le contrat a permis à un voleur d'emporter un tiers des fonds,
- Il s'agissait d'une bogue dans **un** contrat, pas une faille d'Ethereum,
- Débat : faut-il modifier le code d'Ethereum pour empêcher le voleur de migrer les fonds ?

« Gouvernance » de la chaîne

« Gouvernance » de la chaîne

- Attention, le mot « gouvernance » est souvent utilisé pour dire « prise de pouvoir par les gouvernements, qui ne supportent pas que les choses se fassent sans eux »

« Gouvernance » de la chaîne

- Attention, le mot « gouvernance » est souvent utilisé pour dire « prise de pouvoir par les gouvernements »
- Ici, « gouvernance » est au sens de « prise de décision »

« Gouvernance » de la chaîne

- Attention, le mot « gouvernance » est souvent utilisé pour dire « prise de pouvoir par les gouvernements »
- Ici, « gouvernance » est au sens de « prise de décision »
- Que fallait-il faire après le vol ?

« Gouvernance » de la chaîne

- Attention, le mot « gouvernance » est souvent utilisé pour dire « prise de pouvoir par les gouvernements »
- Ici, « gouvernance » est au sens de « prise de décision »
- Que fallait-il faire après le vol ?
- Terminologie (non consensuelle) : *fork* (décision qui va couper le chaîne en deux, ceux qui le suivent et les autres), *soft fork* (des transactions ex-valides sont refusées ; le vieux code continue à marcher), *hard fork* (des transactions ex-invalides sont désormais acceptées ; faut tout mettre à jour)

Plan du tutoriel

- 1 Introduction
- 2 Bitcoin
- 3 Ethereum
- 4 Usages
- 5 Les contrats
- 6 Le langage Solidity**
- 7 Registre de noms de domaine
- 8 Conclusion

Caractéristiques

Caractéristiques

- Langage impératif de haut niveau,

Caractéristiques

- Langage impératif de haut niveau,
- Compilé en EVM,

Caractéristiques

- Langage impératif de haut niveau,
- Compilé en EVM,
- Quelques fonctions pré-définies spécifiques à la chaîne de blocs (comme `send` pour envoyer des ethers).

Exemple Solidity trivial

```
contract Storage {  
    uint storedData;  
  
    function set(uint x) {  
        storedData = x;  
    }  
  
    function get() constant returns (uint retVal) {  
        return storedData;  
    }  
}
```

Exemple Solidity trivial

- `storage.sol`, un contrat qui stocke un entier (et permet de le récupérer)

```
contract Storage {
    uint storedData;

    function set(uint x) {
        storedData = x;
    }

    function get() constant returns (uint retVal) {
        return storedData;
    }
}
```

Exemple Solidity trivial

- `storage.sol`, un contrat qui stocke un entier (et permet de le récupérer)
- Aucune sécurité : toute écriture remplace la valeur précédente

```
contract Storage {
    uint storedData;

    function set(uint x) {
        storedData = x;
    }

    function get() constant returns (uint retVal) {
        return storedData;
    }
}
```

La console JavaScript du nœud geth

```
> eth.blockNumber
1265891
> eth.accounts
["0xaf8e19438e05c68cbdaf33ff15a439ce6742972", "0x2dda57ee99c806477ba05"]
> eth.getBalance(eth.accounts[0])
158160501120369773406
> eth.getBalance(soleau.address)
10060000000000000000

> storage.get()
11
> storage.set.sendTransaction("9",
                               {from: eth.accounts[0], gas: 100000})
"0x6617f5c5382dcb1657c10591c9563e4ec0d07445d628a882b09194fb4fbd6dd2"
> storage.get()
11
> storage.get()
9
```

La chaîne n'oublie rien

Accès aux anciennes valeurs depuis un client JavaScript :

```
% geth --exec 'loadScript("dump-storage.js")' attach ipc:/home/stephan
```

```
...
```

```
Block #1204641 (1466949752 Sun, 26 Jun 2016 16:02:32 CEST) : 100
```

```
Block #1204569 (1466948558 Sun, 26 Jun 2016 15:42:38 CEST) : 44
```

```
Block #1204524 (1466947850 Sun, 26 Jun 2016 15:30:50 CEST) : 10
```

```
...
```

```
contract = "0x6d363cd2eb21ebd39e50c9a2f94a9724bf907d13";
```

```
maxBlocks = 1000;
```

```
startBlock = eth.blockNumber;
```

```
for (var i = 1; i < maxBlocks; i++) { /* Be careful: we go *back* in time */
```

```
    current = web3.eth.getStorageAt(contract, 0, startBlock-i);
```

```
    if (current !== previous) {
```

```
        blockDate = new Date(web3.eth.getBlock(startBlock-i+1).timestamp * 1000);
```

```
        console.log("Block #" + (startBlock-i+1) + " (" +
```

```
            web3.eth.getBlock(startBlock-i+1).timestamp + " " +
```

```
            blockDate.toString() + ") : " + web3.toDecimal(previous));
```

```
        previous = current;
```

```
    }
```

```
}
```

The logo for afnic, consisting of the lowercase letters 'afnic' in a stylized, handwritten-style font.

Sécurité de Solidity

Sécurité de Solidity

- Langage impératif : difficile de raisonner dessus. Des contrats en OCaml validés avec Coq ?

Sécurité de Solidity

- Langage impératif : difficile de raisonner dessus.
- Les fonctions (comme `send`) peuvent échouer mais tester le code de retour n'est pas obligatoire

Sécurité de Solidity

- Langage impératif : difficile de raisonner dessus.
- Les fonctions (comme `send`) peuvent échouer mais tester le code de retour n'est pas obligatoire
- Un contrat peut en appeler un autre mais **c'est dans une autre transaction**

Sécurité de Solidity

- Langage impératif : difficile de raisonner dessus.
- Les fonctions (comme `send`) peuvent échouer mais tester le code de retour n'est pas obligatoire
- Un contrat peut en appeler un autre mais **c'est dans une autre transaction**
- Pas de distinction compte/contrat : on croit envoyer de l'argent à un compte, on appelle son code !

Sécurité de Solidity

- Langage impératif : difficile de raisonner dessus.
- Les fonctions (comme `send`) peuvent échouer mais tester le code de retour n'est pas obligatoire
- Un contrat peut en appeler un autre mais **c'est dans une autre transaction**
- Pas de distinction compte/contrat : on croit envoyer de l'argent à un compte, on appelle son code !
- Si votre fonction n'est pas **réentrante**, votre état peut changer pendant une transaction !

Plan du tutoriel

- 1 Introduction
- 2 Bitcoin
- 3 Ethereum
- 4 Usages
- 5 Les contrats
- 6 Le langage Solidity
- 7 Registre de noms de domaine**
- 8 Conclusion

Présentation détaillée d'un registre de noms en Solidity

Le cahier des charges :

- Permet à quiconque de créer un nom et de l'associer à des données, en payant 1 szabo (un millionième d'ether),
- Permet de récupérer les données associées à un nom, et l'adresse du titulaire (une sorte de whois),
- Permet de détruire un nom,
- Permet de transférer un nom à un nouveau titulaire,
- Aucun privilège pour le gérant du contrat, à part toucher l'argent, et, plus tard, énumérer facilement tous les noms.

Quelques choix techniques

Quelques choix techniques

- Utilisation des chaînes de caractères de Solidity (**très** limitées)

Quelques choix techniques

- Utilisation des chaînes de caractères de Solidity (**très** limitées)
- Deux *mappings*, des noms vers les enregistrements, d'un numéro vers les noms (pour l'énumération)

Quelques choix techniques

- Utilisation des chaînes de caractères de Solidity (**très** limitées)
- Deux *mappings*, des noms vers les enregistrements, d'un numéro vers les noms (pour l'énumération)
- Attention, les *mappings* de Solidity renvoient toujours une valeur, pour tout index

Le contrat Registry

```
contract Registry {  
  
    address public nic; // The Network Information Center  
  
    struct Record {  
        string value; // IP addresses, emails, etc. In the future,  
        // it will be more sophisticated  
        address holder;  
        bool exists; // Or a more detailed state, with an enum?  
        uint idx;  
    }  
    mapping (string => Record) records;  
    mapping (uint => string) index;  
  
    uint public maxRecords;  
    uint public currentRecords;
```

registry.sol, 2/4

```
// Constructor
function Registry() {
    nic = msg.sender;
    currentRecords = 0;
    maxRecords = 0;
    register("NIC", "Automatically created by for the registry");
}

function whois(string name) constant returns(bool exists, string value) {
    if (records[name].exists) {
        exists = true;
        value = records[name].value;
        holder = records[name].holder;
    } else {
        exists = false;
    }
}
```

registry.sol, 3/4

```
function register(string name, string value) {
    /* Payment not yet implemented */
    uint i;
    if (records[name].exists) {
        if (msg.sender != records[name].holder) { // Or use modifiers
            throw;
        }
        else {
            i = records[name].idx;
        }
    }
    else {
        records[name].idx = maxRecords;
        i = maxRecords;
        maxRecords++;
    }
    records[name].value = value;
    records[name].holder = msg.sender;
    records[name].exists = true;
    currentRecords++;
    index[i] = name;
}
```

registry.sol, 4/4

```
function transfer(string name, address to) {
    if (records[name].exists) {
        if (msg.sender != records[name].holder) { ... throw
        records[name].holder = to;
    }
    else ... throw
}

function remove(string name) {
    uint i;
    if (records[name].exists) {
        if (msg.sender != records[name].holder) {
            throw;
        }
        else {
            i = records[name].idx;
        }
    }
    ... else
    records[name].exists = false;
    currentRecords--;
}
```

Enregistrer un nom

```
> registry.register.sendTransaction("icann", "127.0.53.53",  
    {from: eth.accounts[1], gas:200000});  
"0x9f0fd5e5f58ff112ca3b98dcad99f5c20e4403623d7d1e19b42dd2841310bbed"  
> registry.whois("icann")  
[false, "", "0x0000000000000000000000000000000000000000000000000000000000000000"]  
  
> registry.whois("icann")  
[true, "127.0.53.53", "0xc90cd1fa9940a4d4a07a37c53bb4f423fd286945"]
```

Enregistrer un nom

```
> registry.register.sendTransaction("icann", "127.0.53.53",  
    {from: eth.accounts[1], gas:200000});  
"0x9f0fd5e5f58ff112ca3b98dcad99f5c20e4403623d7d1e19b42dd2841310bbed"  
> registry.whois("icann")  
[false, "", "0x0000000000000000000000000000000000000000000000000000000000000000"]  
  
> registry.whois("icann")  
[true, "127.0.53.53", "0xc90cd1fa9940a4d4a07a37c53bb4f423fd286945"]
```

Évidemment, dans la réalité, on passera par (Web ou EPP) puis JSON-RPC vers le nœud Ethereum.

Plan du tutoriel

- 1 Introduction
- 2 Bitcoin
- 3 Ethereum
- 4 Usages
- 5 Les contrats
- 6 Le langage Solidity
- 7 Registre de noms de domaine
- 8 Conclusion

Sur la chaîne de blocs

Sur la chaîne de blocs

- Une invention géniale

Sur la chaîne de blocs

- Une invention géniale
- Rend possible ce qui semblait impossible (noms uniques, choisis et sécurisés en pair-à-pair, par exemple)

Sur la chaîne de blocs

- Une invention géniale
- Rend possible ce qui semblait impossible (noms uniques, choisis et sécurisés en pair-à-pair, par exemple)
- Le *hype* actuel est en bonne partie justifié

Sur Bitcoin

« Bitcoin est fini »

Sur Bitcoin

« Bitcoin est fini »

- Beaucoup d'expérience et beaucoup d'argent,

Sur Bitcoin

« Bitcoin est fini »

- Beaucoup d'expérience et beaucoup d'argent,
- Un *mind share* sans égal

Sur Bitcoin

« Bitcoin est fini »

- Beaucoup d'expérience et beaucoup d'argent,
- Un *mind share* sans égal
- Donc, peut-être de l'avenir mais pas mal d'obstacles à surmonter (gouvernance. . .)

Sur Bitcoin

« Bitcoin est fini »

- Beaucoup d'expérience et beaucoup d'argent,
- Un *mind share* sans égal
- Donc, peut-être de l'avenir mais pas mal d'obstacles à surmonter (gouvernance. . .)
- Bitcoin peut être remplacé, sans que le chaîne de blocs soit remise en cause

Sur les crypto-monnaies en général

Sur les crypto-monnaies en général

- Une idée qui va rester

Sur les crypto-monnaies en général

- Une idée qui va rester
- Les crypto-monnaies actuelles ne tiendront peut-être pas la distance

Sur les crypto-monnaies en général

- Une idée qui va rester
- Les crypto-monnaies actuelles ne tiendront peut-être pas la distance
- Mais on n'arrête pas le progrès : des tas de gens savent que les systèmes centralisés ne sont pas les seuls

Sur Ethereum et les contrats

Sur Ethereum et les contrats

- Une très bonne idée

Sur Ethereum et les contrats

- Une très bonne idée
- Risquée (les programmes ont des bogues. . .)

Sur Ethereum et les contrats

- Une très bonne idée
- Risquée (les programmes ont des bogues. . .)
- « Si le Web a survécu à PHP et JavaScript, Ethereum peut survivre aux bogues de The DAO »

Sur Ethereum et les contrats

- Une très bonne idée
- Risquée (les programmes ont des bogues. . .)
- « Si le Web a survécu à PHP et JavaScript, Ethereum peut survivre aux bogues de The DAO »
- Même si Ethereum est remplacé, l'idée des contrats dans la chaîne va rester

Optimisme prudent

Optimisme prudent

- Grand succès technique

Optimisme prudent

- Grand succès technique
- Des perspectives très intéressantes

Optimisme prudent

- Grand succès technique
- Des perspectives très intéressantes
- Plein d'incertitudes

Optimisme prudent

- Grand succès technique
- Des perspectives très intéressantes
- Plein d'incertitudes
- Investisseurs, attention, c'est du risqué

Merci !

afnic

www.afnic.fr
contact@afnic.fr

afnic