Blockchain, applications concrètes et sécurité

Journée Sécurité Min2Rien – 9 novembre 2017

Renaud LIFCHITZ
renaud.lifchitz@digitalsecurity.fr

digital security | econocom

INTERVENANT

Renaud Lifchitz, IoT security expert, DIGITAL SECURITY

renaud.lifchitz@digitalsecurity.fr

Quelques activités de Digital Security

CONSEIL

Définition

En amont des projets :

- Stratégie, schéma directeur
- Cartographie des risques et plan de traitement
- Etudes prospectives et de cadrage
- Recherche d'opportunités

Construction & mise en œuvre

Ingénierie sécurité :

- Politique & système de management (processus sécurité)
- Conduite du changement (formation, communication, sensibilisation)
- Intégration de la sécurité dans les projets
- Tests et recette des solutions

AUDIT

Evaluation

Au cœur des vérifications

- Tests d'intrusion
- Audits d'architecture
- Audits de conformité
- Audits de maturité
- Audit de code
- Audit de configuration
- Exercices en mode red team
- Préparation aux certifications
- Laboratoire de test et d'essai IoT

CERT

Maintien en condition de sécurité

Accompagnement opérationnel

- Réponse à incidents / Aide à la réaction (traitement des alertes, analyse forensic & post-incident)
- Contrôle continu
- Aide à la détection (veille, surveillance)

ISO 27001 Lead Auditor, ISO 27005 Risk Manager, ISO 22301 Lead Implementor, ITIL, CMMI

Qualifié PASSI

TF-CSIRT Trusted Introducer

Introduction

Blockchain

- Registre global distribué

  (aucun point unique de défaillance)

- Transmission d'informations authentifiée, fiable et sûre

- Multiples usages

- Multiples intérêts

- Entièrement personnalisable selon le contexte métier

Blockchain

Intérêts

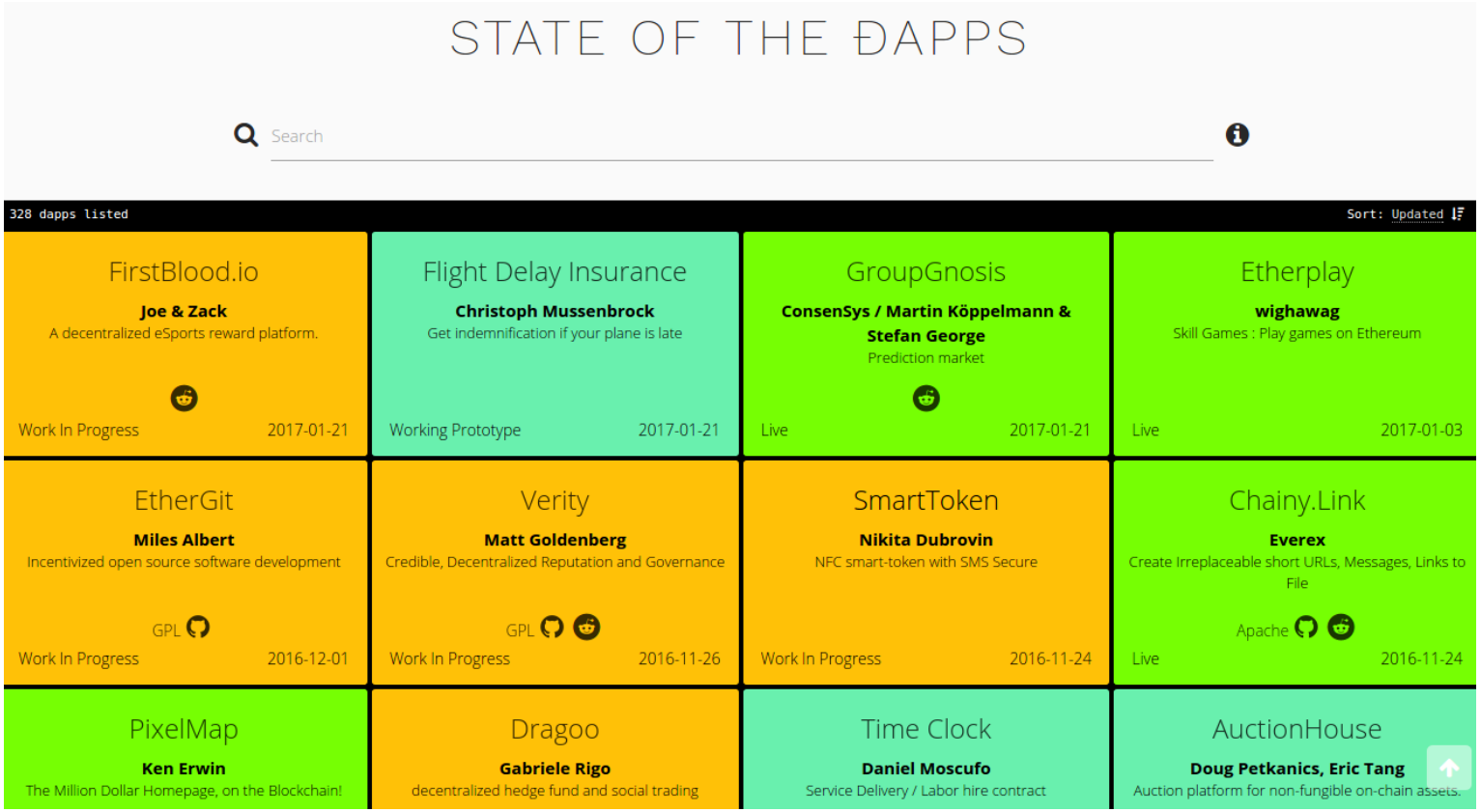
- Scalabilité : facilité pour déployer des nœuds
- Résilience : résistance aux attaques de tout type (réseau, applicatives, dénis de service, …)
- Intégrité et authenticité des données : données authentifiées et immuables
- Décentralisation : pas de point de défaillance unique, plus besoin de tiers de confiance
- Rapidité des transactions par rapport aux réseaux interbancaires (ex.: SWIFT)

Réseau de confiance

Smarts contracts

- Exécution automatisée, décentralisée, conditionnelle et sûre d'engagements (contrats) programmés à l'avance
- Contrats non modifiables une fois déployés sur la blockchain
- Exécution infalsifiable
- Grande variété de contrats modélisables
- Une partie, deux parties, ou contrats multipartites
- dApp : application web décentralisée se connectant à un ou des contrats sur une blockchain

Smarts contracts



« State of the dApps », un annuaire public de dApps Ethereum :

http://dapps.ethercasts.com/

Oracles

- Programmes jouant le rôle de passerelles
  entre une blockchain et le monde physique ou
  plus généralement le web
- Les conditions d'exécution d'un contrat
  dépendent très souvent d'indicateurs externes :
  météo, cours de bourse, actualités, résultat d'un match de sport, solde sur un compte…
- Un oracle se présente le plus souvent sous forme d'une fonction appelable
  depuis un smart contract

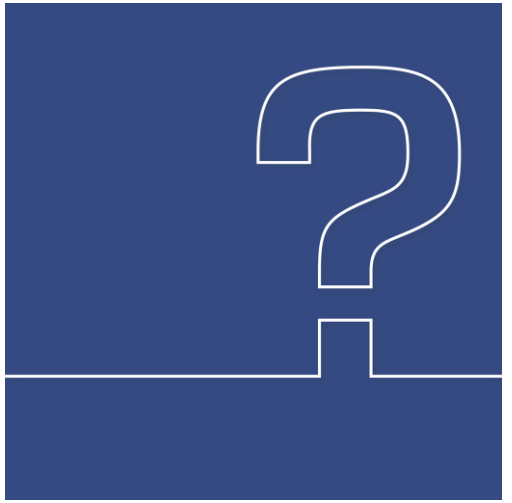
Une blockchain prometteuse : Ethereum

- Première version : 30 juillet 2015
- 15 secondes par bloc
- Des smart contracts très puissants (« Turing-complets »), contrairement à Bitcoin
- Un système d'oracle mâture et bien intégré : http://www.oraclize.it/ , apportant une preuve d'honnêteté (« TLSNotary »)
- Un bon support de la communauté et de quelques professionnels
- Une documentation riche
- Une majorité d'exemples et de démonstrations seront réalisés avec Ethereum lors de cette présentation
- Langage de développement des smart contracts : Solidity (variante typée de Javascript)

Cas d’usages

Pourquoi une blockchain ?

Ou pourquoi ne pas en abuser…



- De nombreux cas d'usage ne justifient pas l'usage d'une blockchain :
  - Transactions très limitées en taille et en nombre (Bitcoin est limité à 3-7 transactions par seconde, Ethereum à 7-15)
  - Système coûteux énergétiquement parlant (par rapport à une redondance informatique classique)
- Plusieurs facteurs favorisent et légitiment par contre l'adoption d'une blockchain :
  - Absence de confiance à priori entre participants
  - Ecriture par des acteurs indépendants
  - Bénéfices pour les participants
  - Désintermédiation

Cas d'usages généraux

- Banque
- Assurance
- Notariat
- Vote électronique
- Conservation de la preuve
- Collecte/Levée de fonds
- Exécution conditionnelle de transactions (contrats électroniques)

Cas d'usages généraux

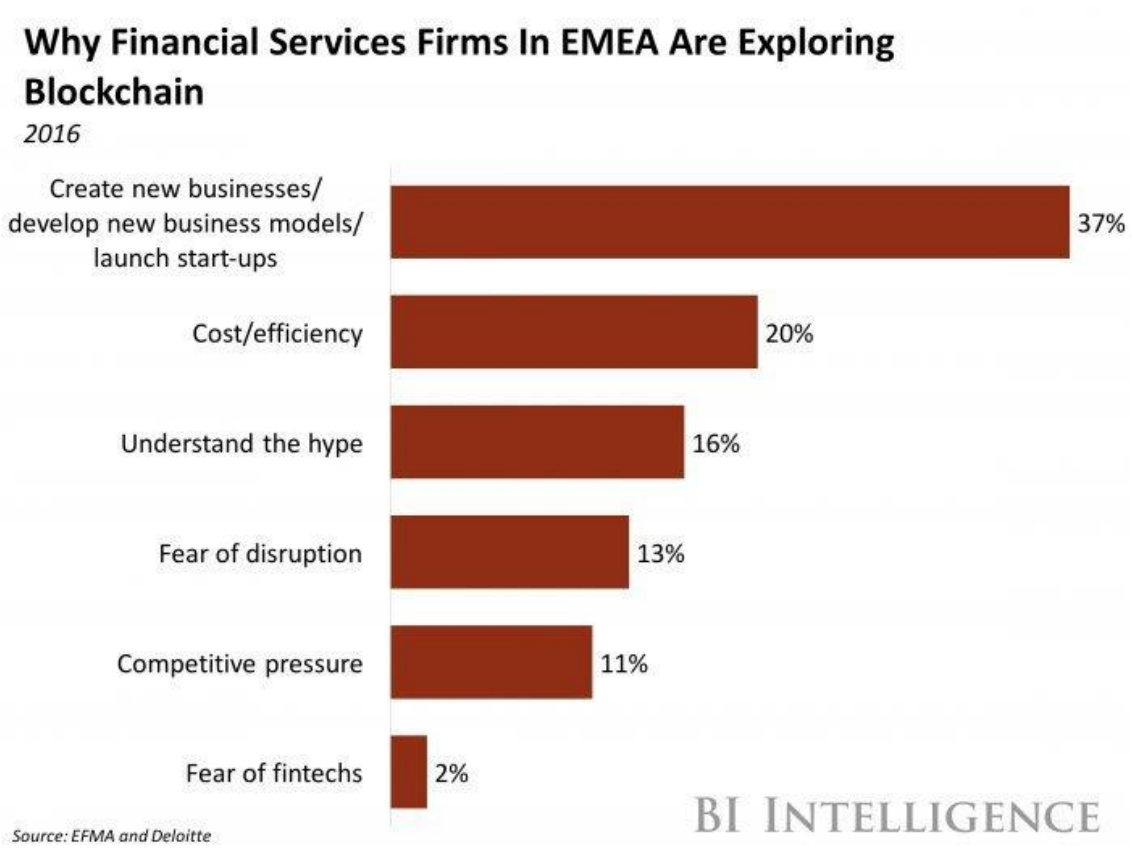
Vote électronique

```
pragma solidity ^0.4.0;
contract Ballot {

    struct Voter {
        uint weight;
        bool voted;
        uint8 vote;
        address delegate;
    }
    struct Proposal {
        uint voteCount;
    }

    address chairperson;
    mapping(address => Voter) voters;
    Proposal[] proposals;

    /// Create a new ballot with $(_numProposals) different proposals.
    function Ballot(uint8 _numProposals) {
        chairperson = msg.sender;
        voters[chairperson].weight = 1;
        proposals.length = _numProposals;
    }

    /// Give $(voter) the right to vote on this ballot.
    /// May only be called by $(chairperson).
    function giveRightToVote(address voter) {
        if (msg.sender != chairperson || voters[voter].voted) return;
        voters[voter].weight = 1;
    }

    /// Delegate your vote to the voter $(to).
    function delegate(address to) {
        Voter sender = voters[msg.sender]; // assigns reference
        if (sender.voted) return;
        while (voters[to].delegate != address(0) && voters[to].delegate != msg.sender)
            to = voters[to].delegate;
        if (to == msg.sender) return;
        sender.voted = true;
        sender.delegate = to;
        Voter delegate = voters[to];
        if (delegate.voted)
            proposals[delegate.vote].voteCount += sender.weight;
        else
            delegate.weight += sender.weight;
    }

    /// Give a single vote to proposal $(proposal).
    function vote(uint8 proposal) {
        Voter sender = voters[msg.sender];
        if (sender.voted || proposal >= proposals.length) return;
        sender.voted = true;
        sender.vote = proposal;
        proposals[proposal].voteCount += sender.weight;
    }

    function winningProposal() constant returns (uint8 winningProposal) {
        uint256 winningVoteCount = 0;
        for (uint8 proposal = 0; proposal < proposals.length; proposal++)
            if (proposals[proposal].voteCount > winningVoteCount) {
                winningVoteCount = proposals[proposal].voteCount;
                winningProposal = proposal;
            }
    }
}
```

Smart contract Ethereum de vote électronique
(Browser Solidity)

Cas d'usages généraux

Intérêts des services financiers EMEA dans la blockchain



Cas d'usages généraux

Démonstration

Notariat / Ancrage de données / Preuve d'antériorité sur la blockchain Bitcoin :

https://woleet.io/

Banques

Elles ont franchi le pas blockchain…

Banques

Un standard pour l'émission de jetons sur la blockchain ?

- Jeton : unité de valeur dont on souhaite contrôler l'émission, l'utilisation et/ou les contreparties

- Standard ERC-20 sur Ethereum (https://github.com/ethereum/EIPs/issues/20)

- Utilisation :
  - Monnaie électronique
  - Points de fidélité (enseignes commerciales)
  - Bons d'achat / bons de réduction
  - Preuves

Banques

Exemple : séquestre de fonds

X

The world's first proof-of-hodl

The safest uncapped ICO has begun. Pledge your Ether into our eighteen-line smart contract to prevent cashing out early and secure your ticket to the moon. **Ether will be withdrawable after Ethereum's fifth birthday (July 30, 2020).**

Deposit your Ether to

0x1bb28e79f2482df6bf60efc7a33365703bcf1536

(Don't send directly from an exchange — you must have access to your private key to withdraw.)

```
pragma solidity ^0.4.11;
contract hodlEthereum {
    event Hodl(address indexed hodler, uint indexed amount);
    event Party(address indexed hodler, uint indexed amount);
    mapping (address => uint) public hodlers;
    uint constant partyTime = 1596067200; // 30th July 2020
    function() payable {
        hodlers[msg.sender] += msg.value;
        Hodl(msg.sender, msg.value);
    }
    function party() {
        require (block.timestamp > partyTime && hodlers[msg.sender] > 0);
        uint value = hodlers[msg.sender];
        hodlers[msg.sender] = 0;
        msg.sender.transfer(value);
        Party(msg.sender, value);
    }
}
```

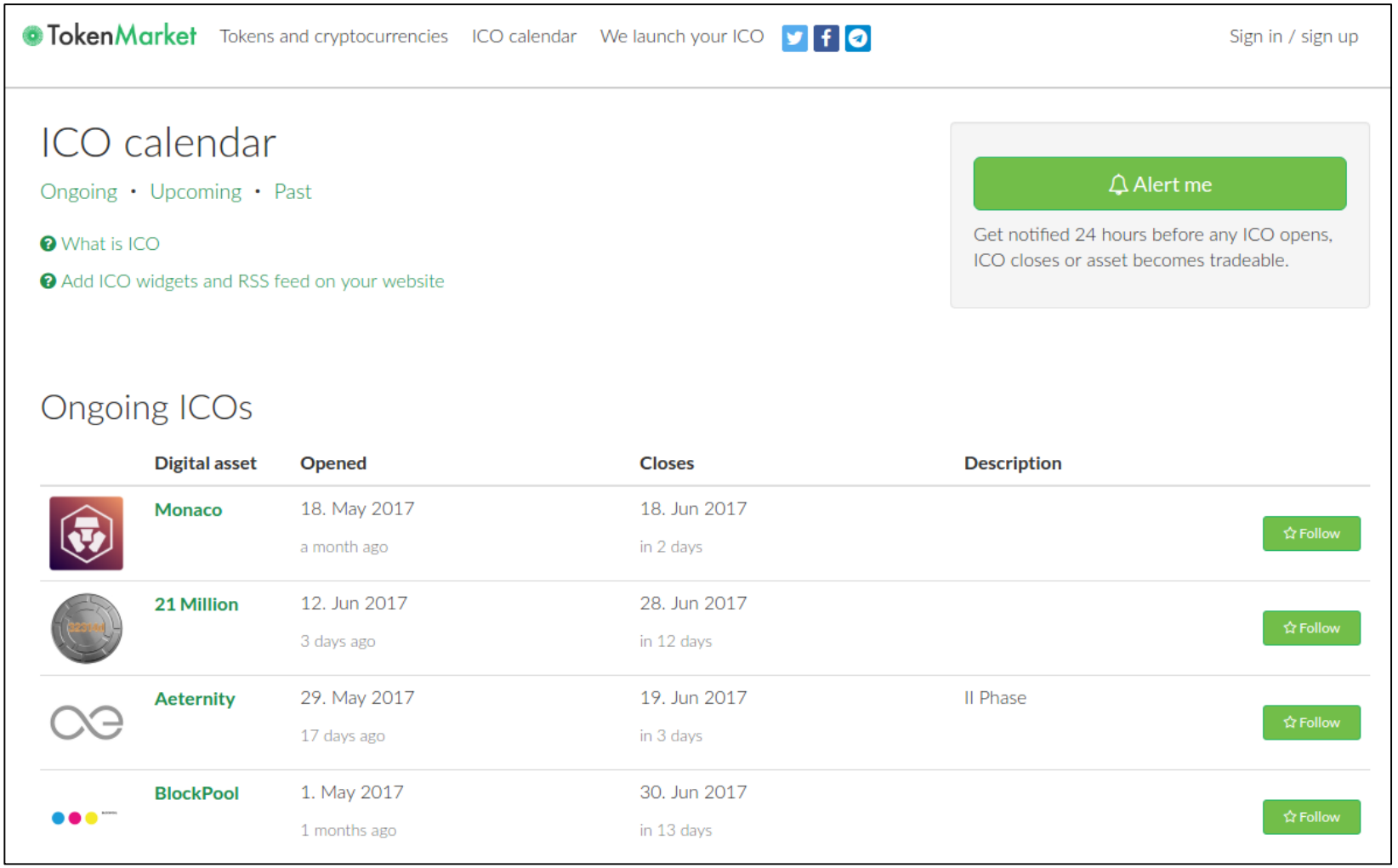
Warning: You won't be able to withdraw your balance for a long time.

Smart contract Ethereum de séquestre décentralisé de fonds :

https://hodlethereum.com/deposit

Banques

Exemple : levées de fonds décentralisées



Calendrier des levées de fonds Ethereum recensées (ICO : « Initial Coin Offering ») :

https://tokenmarket.net/ico-calendar

Assurances

Cas d'usage

- Automatisation du paiement des primes à échéance
- Assurances indicielles ou paramétriques : estimations actualisées des risques par oracle
- Garantie d'unicité de déclaration de sinistre
- Acquittement de sinistre par oracle
- Rationalisation du paiement des indemnisations

Assurances

Exemples

- Assurance couvrant les retards d'avion :
  « Flight Delays Suck! » : https://fdd.etherisc.com/

- Assurance couvrant les cultures contre les risques de sécheresse ou d'inondation :
  « Jamii Crop Insurance » : https://crop.etherisc.com/

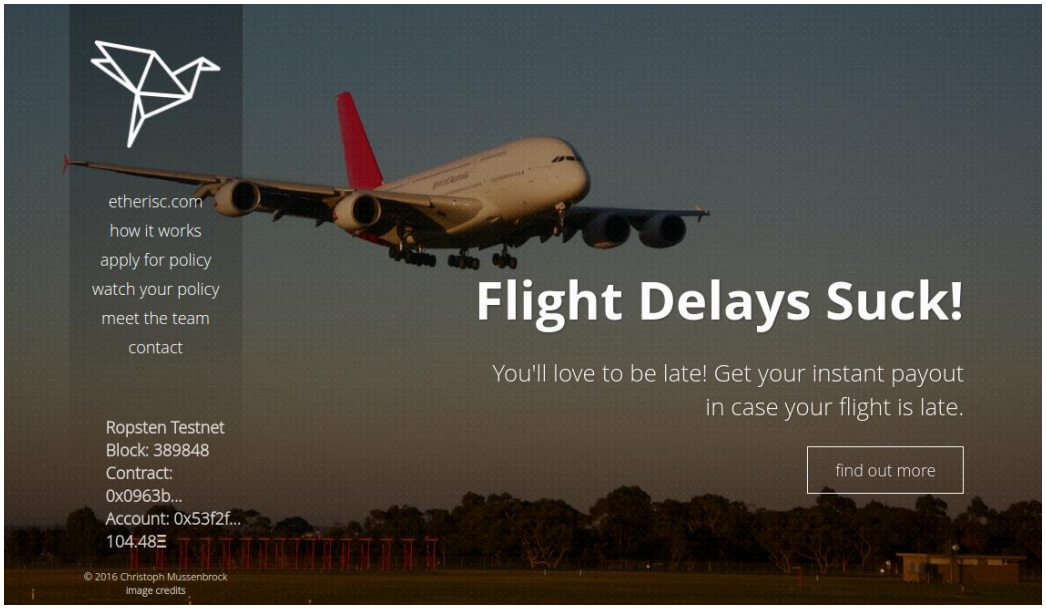
- Sécurité sociale décentralisée (en test) :
  « Etherisc Social Insurance » https://govhack.etherisc.com/

- Mise en oeuvre de swaps de risque de catastrophe naturelle, négociation facilitée des obligations catastrophe (Allianz Risk Transfer AG & Nephila Capital Limited)

- Développement de sidechains pour l'interopérabilité entre blockchains et le traitement de transactions massives (Axa Strategic Ventures & Blockstream)

Assurances

Démonstration



Assurance couvrant les retards d'avion :

« Flight Delays Suck! » : https://fdd.etherisc.com/

Sécurité

L'affaire « The DAO » (1/2)

- The DAO est un smart contract de levée de fonds (Organisation Décentralisée Autonome) développé par Slock.it (serrure connectée à la blockchain) et créé en mai 2016

- Analyse juridique de la contractualisation avec un smart contract via la société suisse DAO.LINK: https://www.ethereum-france.com/dao-link-permet-a-des-entreprises-de-contracter-avec-des-dao/

- Equivalent de plus de 150 millions d'euros collectés pour un projet initial qui ne nécessitait que quelques centaines de milliers d'euros (15% de la masse monétaire émise)

L'affaire « The DAO » (2/2)

- 17 juin 2016 : détournement du tiers par exploitation d'une vulnérabilité d'implémentation (appels récursifs) dans le contrat

- « Hard Fork » en juillet 2016 pour liquider le contrat et récupérer les fonds, puis naissance d'ETC : quid de la gouvernance ?

Impacts du choix de la technologie

La blockchain

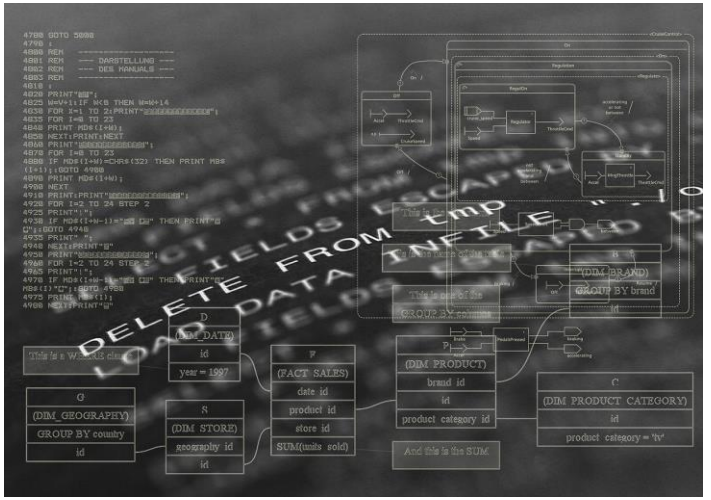
- Critères importants :
  - Maturité
  - Sécurité
  - Possibilité d'interopératibilité (oracles et sidechains)
  - Support
  - Puissance des smart contracts
  - Montée en charge (taille des transactions et délai entre les blocs)

- Quelques blockchains :
  Bitcoin, Ethereum, Zcash, Ripple, Sia, Lisk, Tezos, (DAG : Iota, Byteball) …

Impacts du choix de la technologie

Le langage de développement des smart contracts

- Langages impératifs :
  - Courants en développement
  - Plus simples à écrire
  - Plus complexes à vérifier par preuve formelle (effets de bord)



- Langages fonctionnels :
  - Peu communs
  - Complexes à écrire
  - Plus faciles à vérifier (pas d'effets de bord)

Bonnes pratiques de sécurité

Bonnes pratiques fonctionnelles

- Simplicité, modularité et réutilisabilité du code
- Ecriture de tests unitaires et de tests d'intégration
- Incitations économiques diverses :
  - Limites de montants traités
  - Bug bounties (ex. : https://bountyfactory.io)
  - Marchés de prédiction (ex. : https://gnosis.pm/ , https://augur.net/)
- Séparation des conditions et des actions dans le code (« Condition-Oriented programming »)

Bonnes pratiques de sécurité

Bonnes pratiques techniques

- Implémentation d'un « killswitch » dans les contrats
- Pré et post-conditions sur les fonctions
- Preuves formelles : plus faciles avec les langages fonctionnels (mais incitations économiques non prises en compte)
- Utilisation de « mocks » pour les tests
- Utilisation d'environnements de test (frameworks, testnets…)

Nos prestations de service blockchain orientées sécurité

Nos savoir-faire blockchain / sécurité

- Accompagnement à la conception et mise en œuvre de solutions blockchain
- Evaluation des risques techniques et juridiques
- Formation aux technologies blockchain
- Développement de preuves de concept
- Audit de primitives cryptographiques
- Développement de smart contracts
- Maîtrise des technologies Bitcoin, Ripple et Ethereum

Digital Security participe à la rédaction d'une étude sur la blockchain pour un ministère

Questions ? / Contact

Renaud LIFCHITZ
Consultant Sécurité Senior
renaud.lifchitz@digitalsecurity.fr

info@digitalsecurity.fr