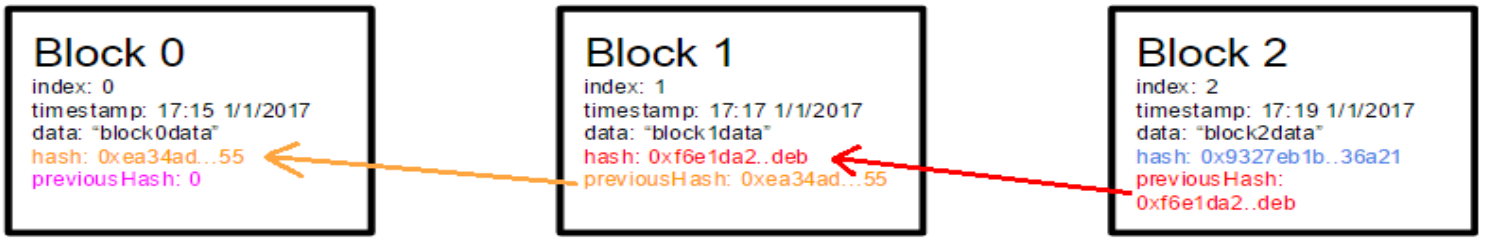
BLOCKCHAIN

- - Liste chaînés publiques :
  - Sécurité des chaînes publiques
  - Problème des généraux Byzantins
  - Cryptologie
  - Alternatives
  - Limites du modèle public (BTC)
    - Energie
    - Volume
    - Vitesse
- - Liste chaînées privées :
  - Concept et use case
  - Etat de l'art
  - Perspectives/applications

BLOCKCHAIN

Les concepts techniques sur lesquels repose le principe d'une blockchain :

- Une liste chaînée
- Un mécanisme qui garantit la non altération de la liste



- Demo du principe de blockchain publique :
  - https://anders.com/blockchain/hash.html

ynov CAMPUS

BLOCKCHAIN PUBLIQUE

La plupart des projets reposent sur le principe d'une blockchain publique. Dans la pratique, n'importe quel noeud peut-être amené à créer un nouveau bloc (sous certaines conditions). C'est ce choix d'architecture qui induit l'utilisation d'un système de sécurité basé sur une forme de tirage au sort connu sous le terme « minage ». Par exemple, le BTC résoud ce problème grâce à :

- Un mode p2p qui résoud le problème des généraux byzantins
- Cryptologie sha256()

Certaines alternatives existent, recherche de nombres premiers (Primecoin & Riecoin). Mais la plupart ont adopté un modèle de blockchain publique.

BLOCKCHAIN PUBLIQUE

Limites du modèle chaîne publique :

- Energie
- Volume
- Vitesse

- Le modèle de blockchain publique actuel semble condamné dans un monde limité.
- Il est possible d'utiliser le même principe de chaîne tout en remplaceant le système de sécurité basé sur le minage par un système de sécurité basé sur un réseau de confiance (ring of trust). Seuls les membres de ce réseau sont habilités à modifier la blockchain...

BLOCKCHAIN PRIVÉE

En utilisant les mêmes principes que les certificats SSL ou les clés SSH (clé publique/privée), il est possible de sécuriser une liste chaînée par des signatures numériques. Du coup, la modification de la liste chainée ne peut être réalisé que par un membre de ce réseau. Pour bien faire, l'idéal est d'intégrer la liste des membres ... dans la liste chaînée elle-même.

L'entrée dans ce type de cercle ne peut se faire que par « certification » d'un ou plusieurs membres.

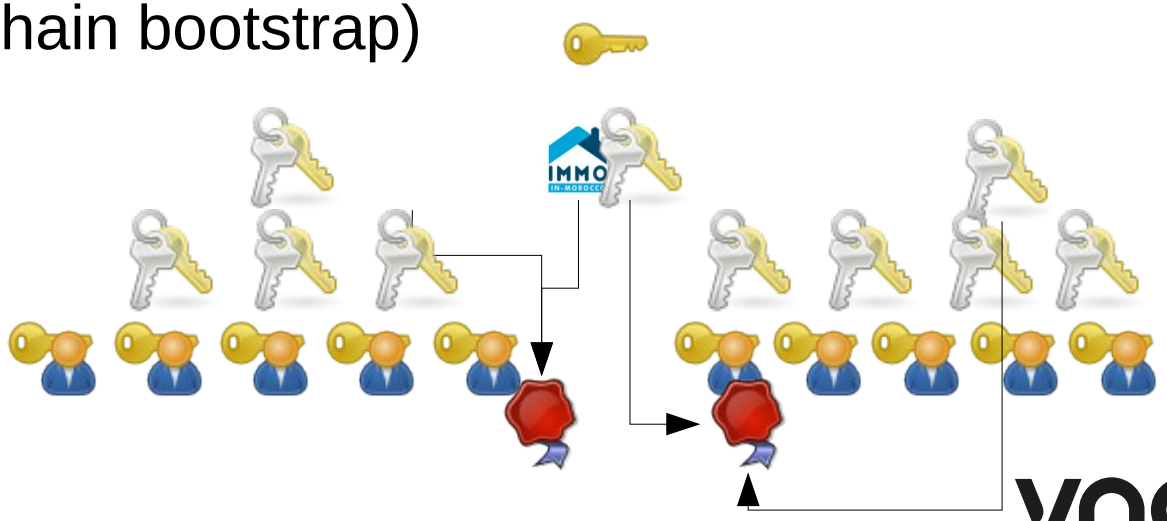
BLOCKCHAIN PRIVÉE

Concept :

A l'instar des Autorités de certification pour les certificats SSL serveur, le système intègre ses propres règles de « certification ». Définir si on doit être certifié par N membre pour devenir membre, le délai de validité du certificat, ...

Par exemple, pour devenir « membre » de ce réseau pour pouvoir créer de nouveaux blocs, il faut obtenir une nouvelle « clé » qui soit certifiée par deux certificats du niveaux supérieur :

- Level0 : root (blockchain bootstrap)
- Level1 : national
- Level2 : régional
- Level3 : communal



BLOCKCHAIN PRIVÉE

État de l'art :

- Cadastre, notaires :
  - ONG Bitland (Ghana)
  - BitFury (Géorgie)
  - Ubitquity (États-Unis)
  - Chromaway (Suède)
- Duniter :
  - Théorie de la monnaie relative
- ...

BLOCKCHAIN PRIVÉE

Applications :

- Documents officiels (diplômes, actes notariés, factures...)
- Histogramme (dépôts de brevets, preuves d'actes privés, associations...)
- Base documentaire répartie
- Cercles de confiance (authentification, habilitations...)
- ...

BLOCKCHAIN

Objectif du projet collaboratif :

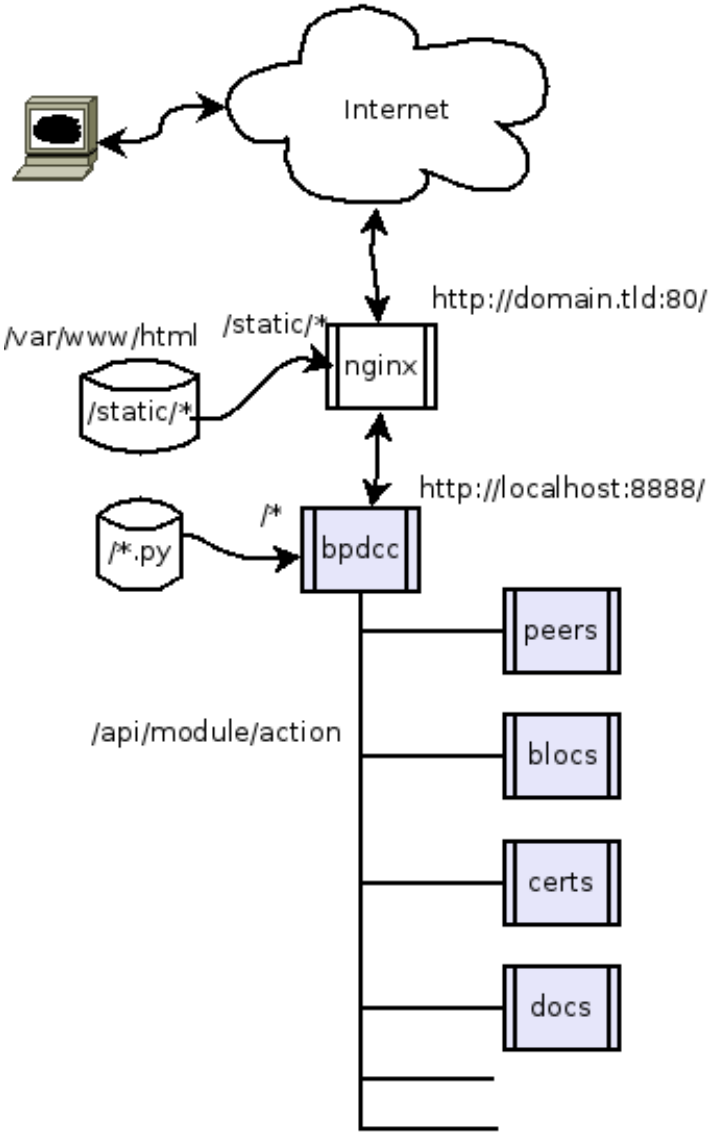
- Concevoir un modèle de sécurité alternatif (sans minage)
- Concevoir un modèle de stockage de documents certifiés conformes (en remplacement des transactions monétaires)

BLOCKCHAIN PROJECT

- Commons :
  - VPN Full-mesh
  - PKI intégrée
  - Répartition du stockage
- Exemples d'implémentation
  - Quelques pistes...
  - https://github.com/modulix/bpdcc

ynov CAMPUS

BLOCKCHAIN PROJECT



- Application WEB (wsgi)
- API REST
- POST JSON
- Access :
  only Members/Owner
- Initial rsync

ynov CAMPUS

BLOCKCHAIN PROJECT

Chaque service est un module Python qui contient les méthodes correspondantes à l'URL :

- /api/module/methode ? arguments
- Exemple : /api/doc/list.json?data={'id' : 'xxx', 'nbr' : 100}
  - ~/doc/__init__.py
  - ~/doc/index.py
    - add()
    - del()
    - list()
    - validate()
    - sign()
    - ...

BLOCKCHAIN PROJECT

De plus, si le module est un « service », un thread permanent, permet de réaliser les traitements en boucle :

- peers :
- ~/peers/__init__.py
  - ~/doc/peers.py
    - run()
      - for peer in peers :
        self.get_info(peer)
        sleep(3)

BLOCKCHAIN PROJECT

Structures JSON :

- Peers :
  - {
    'name' : 'node_0000',
    'ip' : '10.33.33.10',
    'quality' : 9999,
    'description' : 'xxxxx',
    'date' : yyyy-mm-dd-H24-MI-SS',
    'certs' : ['xxxxx', 'yyyyy']
    }