

Samedi 28 janvier 2023



# LOCKCHAINS

## Models de Transactions

# Sommaire

## 1. Intro

- Les Cypher-punks

## 2. Transactions Bitcoin

- Transaction non dépensés uTXO
- Bitcoin Script
- Les Limites du modèle Bitcoin

## 3. Etherum

- Avantages d'Etherum
- Similitudes Etherum - Bitcoin
- Différence Etherum - Bitcoin
- Exécution d'un contrat
- Applications

## 4. Références

- Références

# Les précurseurs du Bitcoin

- ▶ Hashcash (M. Back)
- ▶ Titres de propriété sécurisés (M. Szabo)
  - ▶ Bit gold ; *smart-contract*
- ▶ B-money (M. Dai)
- ▶ Employé de PGP (M. Finney)
  - ▶ 1er recipiendaire de BTC
- ▶ Voisin de Finney



# Les précurseurs du Bitcoin

- ▶ Hashcash (M. Back)
- ▶ Titres de propriété sécurisés (M. Szabo)
  - ▶ Bit gold ; *smart-contract*
- ▶ B-money (M. Dai)
- ▶ Employé de PGP (M. Finney)
  - ▶ 1er recipiendaire de BTC
- ▶ Voisin de Finney



Nick Szabo

# Les précurseurs du Bitcoin

- ▶ Hashcash (M. Back)
- ▶ Titres de propriété sécurisés (M. Szabo)
  - ▶ Bit gold ; *smart-contract*
- ▶ B-money (M. Dai)
- ▶ Employé de PGP (M. Finney)
  - ▶ 1er recipiendaire de BTC
- ▶ Voisin de Finney



Wei Dai

# Les précurseurs du Bitcoin

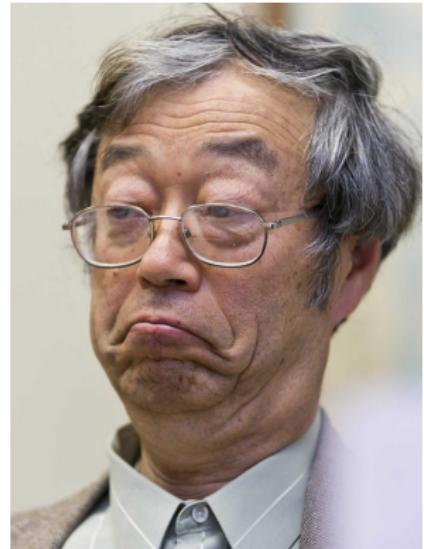
- ▶ Hashcash (M. Back)
- ▶ Titres de propriété sécurisés (M. Szabo)
  - ▶ Bit gold ; *smart-contract*
- ▶ B-money (M. Dai)
- ▶ Employé de PGP (M. Finney)
  - ▶ 1er recipiendaire de BTC
- ▶ Voisin de Finney



Hal Finney

# Les précurseurs du Bitcoin

- ▶ Hashcash (M. Back)
- ▶ Titres de propriété sécurisés (M. Szabo)
  - ▶ Bit gold ; *smart-contract*
- ▶ B-money (M. Dai)
- ▶ Employé de PGP (M. Finney)
  - ▶ 1er recipiendaire de BTC
- ▶ Voisin de Finney



Dorian Nakamoto

# Concept Cypher-punks

- ▶ Un Escrow : un garant, **contrat de séquestration** (**contrat de dépôt**)
- ▶ Signature Elliptiques (ECDSA)
  - ▶ Sécurité forte : de l'ordre de  $2^{\text{longueur}(d_A)}$  où  $d_A$  est la clef
  - ▶ Cryptographie asymétrique, pair de clef (privée, public)
  - ▶ Utile pour signer les transactions
- ▶ Merkle Tree :
  - ▶ taille actuelle de la **blockchain Bitcoin** : 450G
- ▶ Fonction Hash160 = ripemd160(sha256(data))

# Concept Cypher-punks

- ▶ Un Escrow : un garant, **contrat de séquestration** (contrat de dépôt)
- ▶ Signature Elliptiques (ECDSA)
  - ▶ Sécurité forte : de l'ordre de  $2^{\text{longueur}(d_A)}$  où  $d_A$  est la clef
  - ▶ Cryptographie asymétrique, pair de clef (privée, public)
  - ▶ Utile pour signer les transactions
- ▶ Merkle Tree :
  - ▶ taille actuelle de la **blockchain Bitcoin** : 450G
- ▶ Fonction Hash160 = ripemd160(sha256(data))

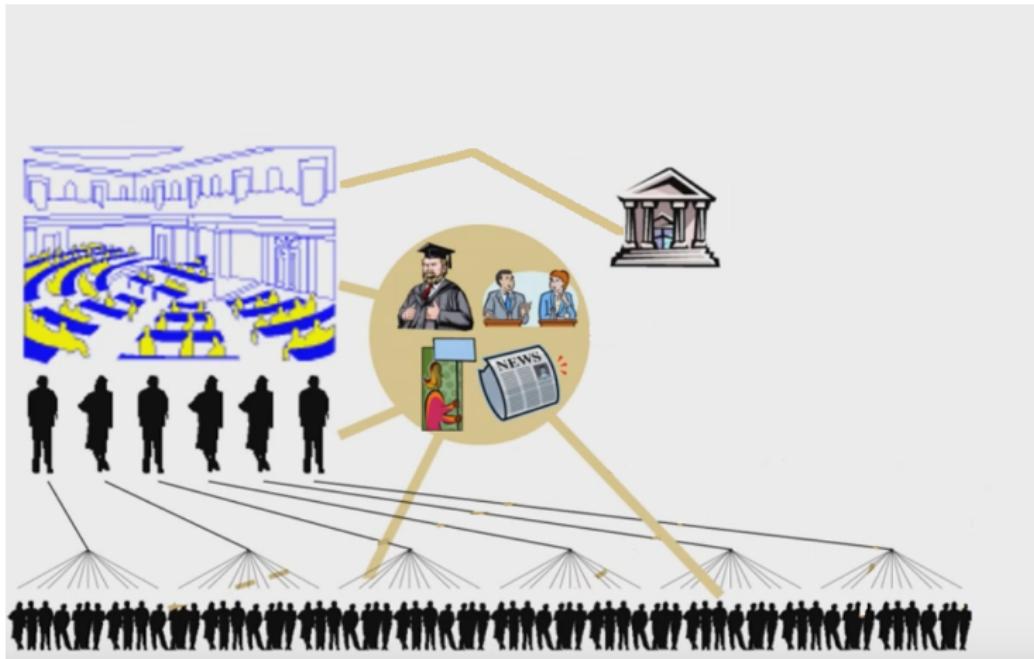
# Concept Cypher-punks

- ▶ Un Escrow : un garant, **contrat de séquestre** (contrat de dépôt)
- ▶ Signature Elliptiques (ECDSA)
  - ▶ Sécurité forte : de l'ordre de  $2^{\text{longueur}(d_A)}$  où  $d_A$  est la clef
  - ▶ Cryptographie asymétrique, pair de clef (privée, public)
  - ▶ Utile pour signer les transactions
- ▶ Merkle Tree :
  - ▶ taille actuelle de la **blockchain Bitcoin** : 450G
- ▶ Fonction  $\text{Hash160} = \text{ripemd160}(\text{sha256}(\text{data}))$

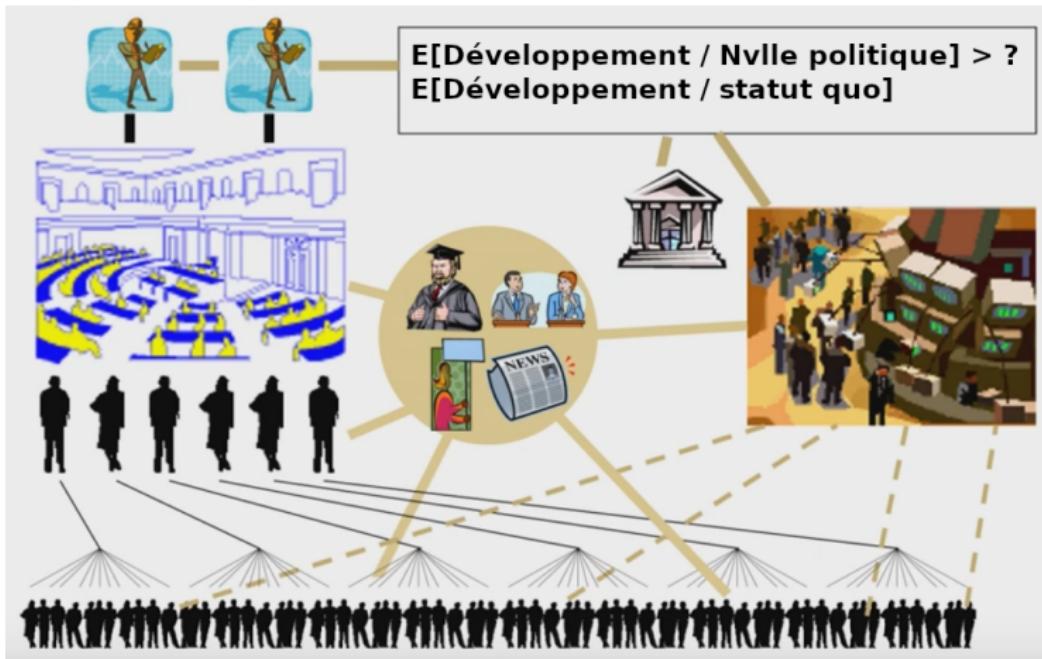
# Concept Cypher-punks

- ▶ Un Escrow : un garant, **contrat de séquestre** (contrat de dépôt)
- ▶ Signature Elliptiques (ECDSA)
  - ▶ Sécurité forte : de l'ordre de  $2^{\text{longueur}(d_A)}$  où  $d_A$  est la clef
  - ▶ Cryptographie asymétrique, pair de clef (privée, public)
  - ▶ Utile pour signer les transactions
- ▶ Merkle Tree :
  - ▶ taille actuelle de la **blockchain Bitcoin** : 450G
- ▶ Fonction  $\text{Hash160} = \text{ripemd160}(\text{sha256}(\text{data}))$

# Futarchy : Voter les valeurs en pariant l'espérance



# Futarchy : Voter les valeurs en pariant l'espérance



- Financiarisation des décisions politiques ([Robin Hanson](#))

# Sommaire

## 1. Intro

- Les Cypher-punks

## 2. Transactions Bitcoin

- Transaction non dépensés uTXO
- Bitcoin Script
- Les Limites du modèle Bitcoin

## 3. Etherum

- Avantages d'Etherum
- Similitudes Ethereum - Bitcoin
- Différence Ethereum - Bitcoin
- Exécution d'un contrat
- Applications

## 4. Références

- Références

# Détail d'un transaction non dépensée

```

{
  "hash": "5a42590fbe0a90ee8e8747244d6c84f0db1a3a24e8f1b95b10c9e050990b8b6b",
  "ver": 1,
  "vin_sz": 2,
  "vout_sz": 1,
  "lock_time": 0,
  "size": 404,
  "in": [
    {
      "prev_out": {
        "hash": "3be4ac9728a0823cf5e2deb2e86fc0bd2aa503a91d307b42ba76117d79280260",
        "n": 0
      },
      "scriptSig": "30440..."
    },
    {
      "prev_out": {
        "hash": "7508e6ab259b4df0fd5147bab0c949d81473db4518f81afc5c3f52f91ff6b34e",
        "n": 0
      },
      "scriptSig": "3f3a4ce81...."
    }
  ],
  "out": [
    {
      "value": "10.12287097",
      "scriptPubKey": "OP_DUP OP_HASH160 69e02e18b5705a05dd6b28ed517716c894b3d42e OP_EQUALVERIFY OP_CHECKSIG"
    }
  ]
}

```

# Sommaire

## 1. Intro

- Les Cypher-punks

## 2. Transactions Bitcoin

- Transaction non dépensés uTXO
- **Bitcoin Script**
- Les Limites du modèle Bitcoin

## 3. Etherum

- Avantages d'Etherum
- Similitudes Ethereum - Bitcoin
- Différence Ethereum - Bitcoin
- Exécution d'un contrat
- Applications

## 4. Références

- Références

# Un exemple

## Script "Payer à ... "

```
P_DUP  
OP_HASH160  
69e02e18...  
OP_EQAVLVERIFY  
OP_CHEKSIG
```

## Instructions

- ▶ programmation en pile
- ▶ 256 instructions
  - ▶ 15 désactivées
  - ▶ 75 réservées (pour plus tard)

(liste complète sur [en.bitcoin.it](http://en.bitcoin.it))

# Un exemple

## Script "Payer à ... "

<sig>

<pubKey>

---

P\_DUP

OP\_HASH160

<pubKeyHash ?>

OP\_EQAVLVERIFY

OP\_CHEKSIG

### Instructions

- ▶ programmation en pile
- ▶ 256 instructions
  - ▶ 15 désactivées
  - ▶ 75 réservées (pour plus tard)

(liste complète sur [en.bitcoin.it](http://en.bitcoin.it))

# Un exemple

## Script "Payer à ... "

<sig>

<pubKey>

---

P\_DUP

OP\_HASH160

<pubKeyHash ?>

OP\_EQAVLVERIFY

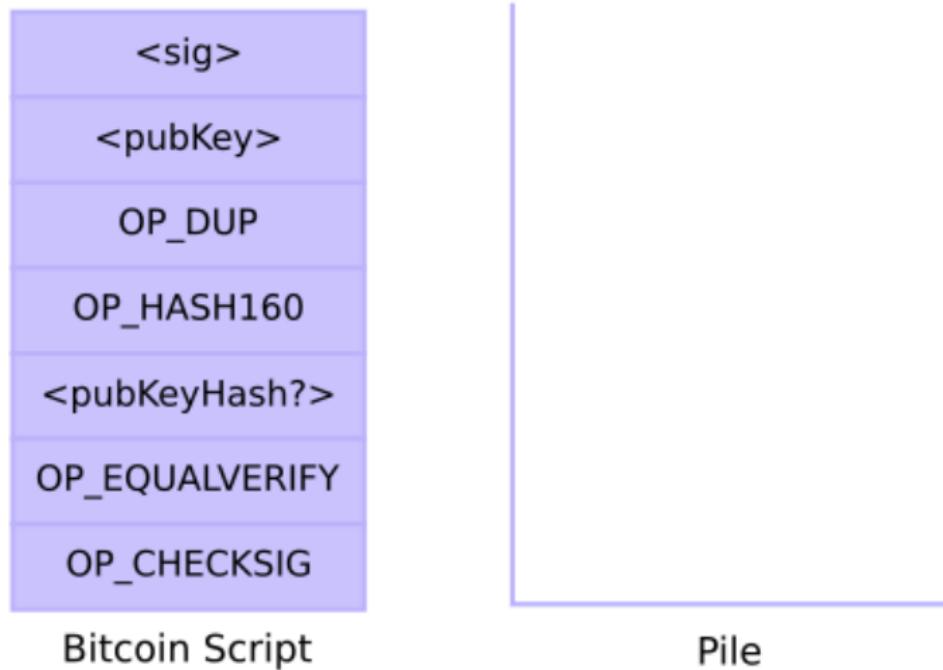
OP\_CHEKSIG

## Instructions

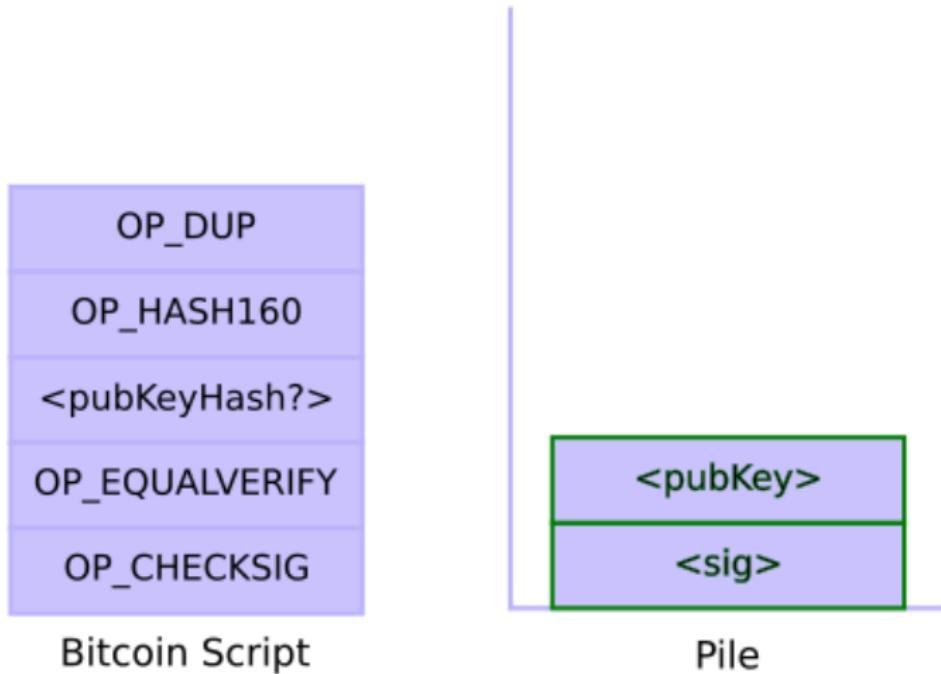
- ▶ programmation en pile
- ▶ 256 instructions
  - ▶ 15 désactivées
  - ▶ 75 réservées (pour plus tard)

([liste complète sur en.bitcoin.it](http://en.bitcoin.it))

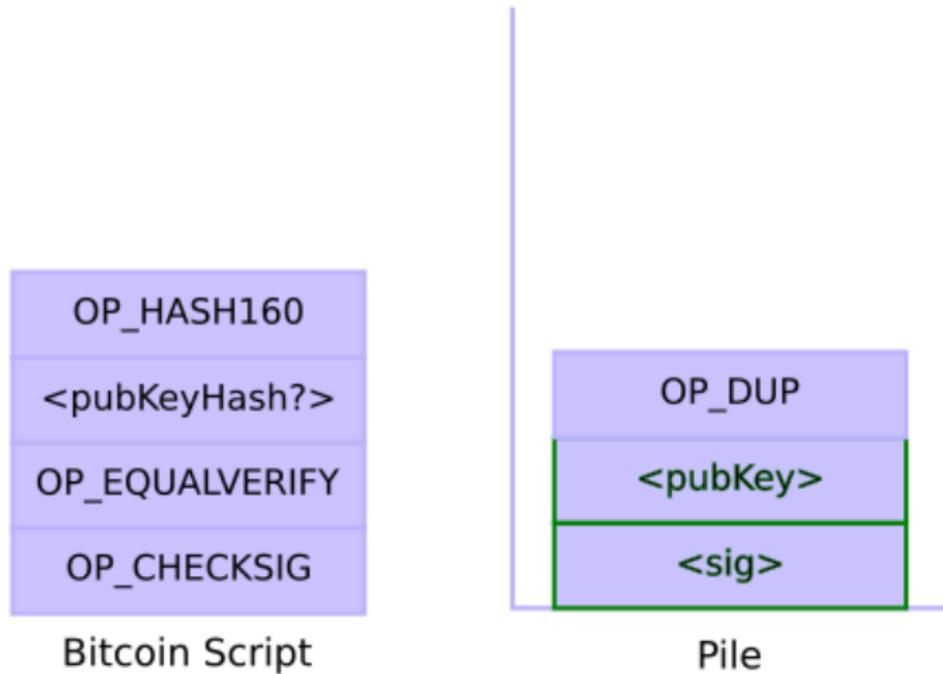
## Execution du Script "Payer à ..."



## Execution du Script "Payer à ..."



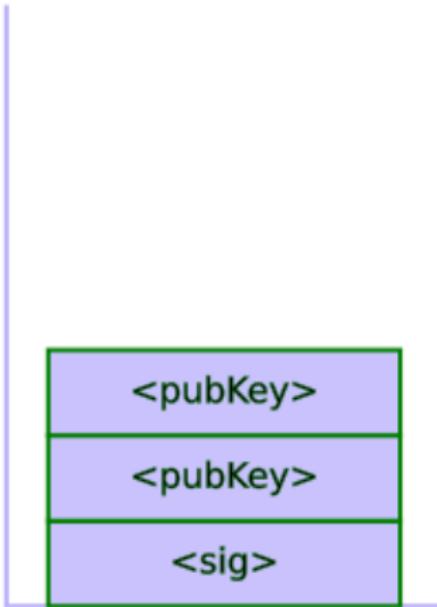
## Execution du Script "Payer à ..."



## Execution du Script "Payer à ..."

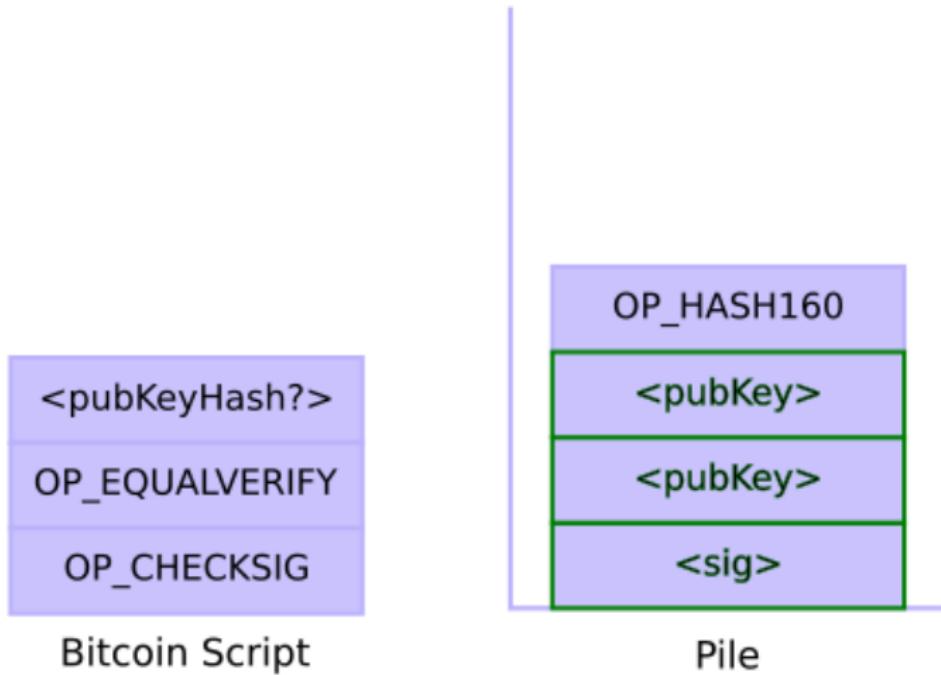
OP\_HASH160  
<pubKeyHash?>  
OP\_EQUALVERIFY  
OP\_CHECKSIG

Bitcoin Script



Pile

## Execution du Script "Payer à ..."



## Execution du Script "Payer à ..."

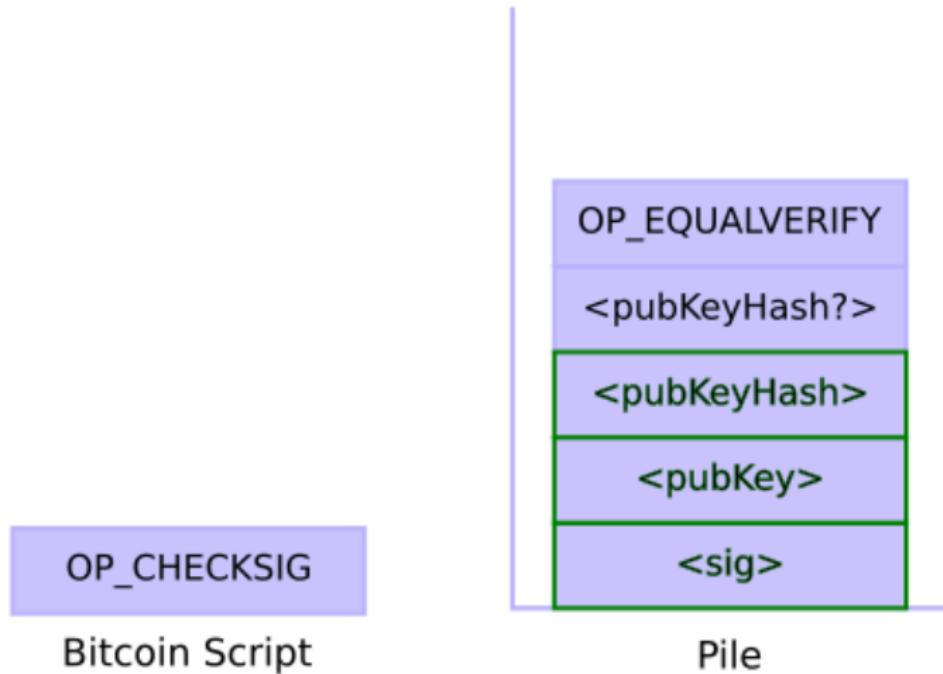
```
<pubKeyHash?>
OP_EQUALVERIFY
OP_CHECKSIG
```

Bitcoin Script

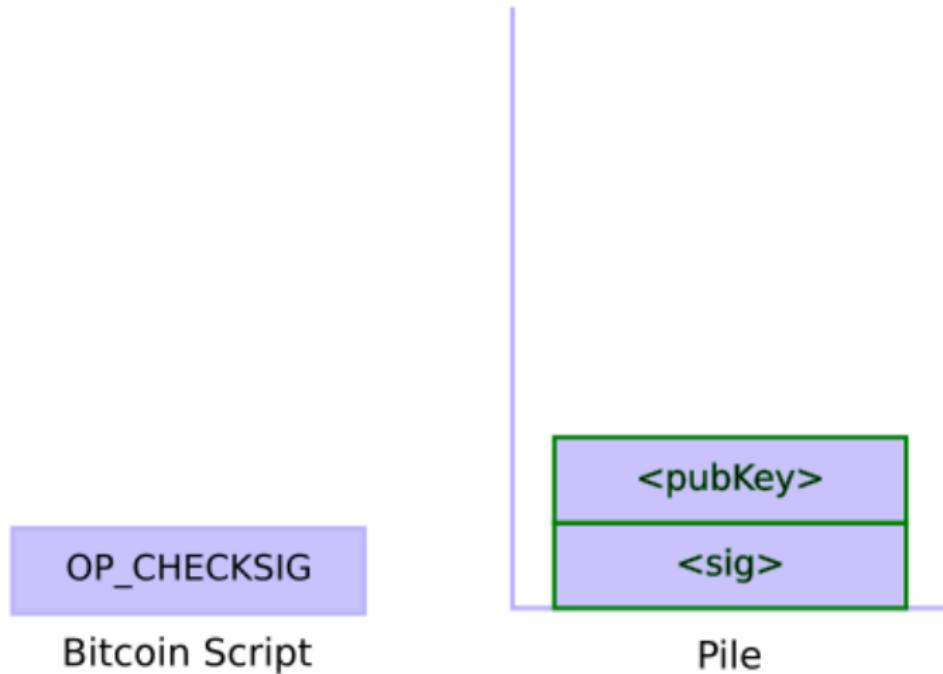
```
<pubKeyHash>
<pubKey>
<sig>
```

Pile

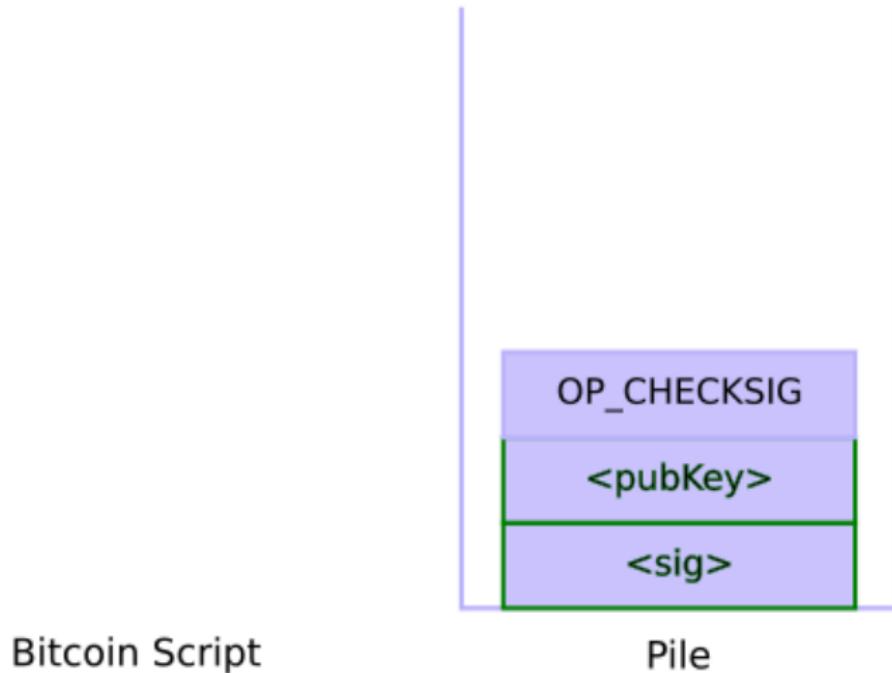
## Execution du Script "Payer à ..."



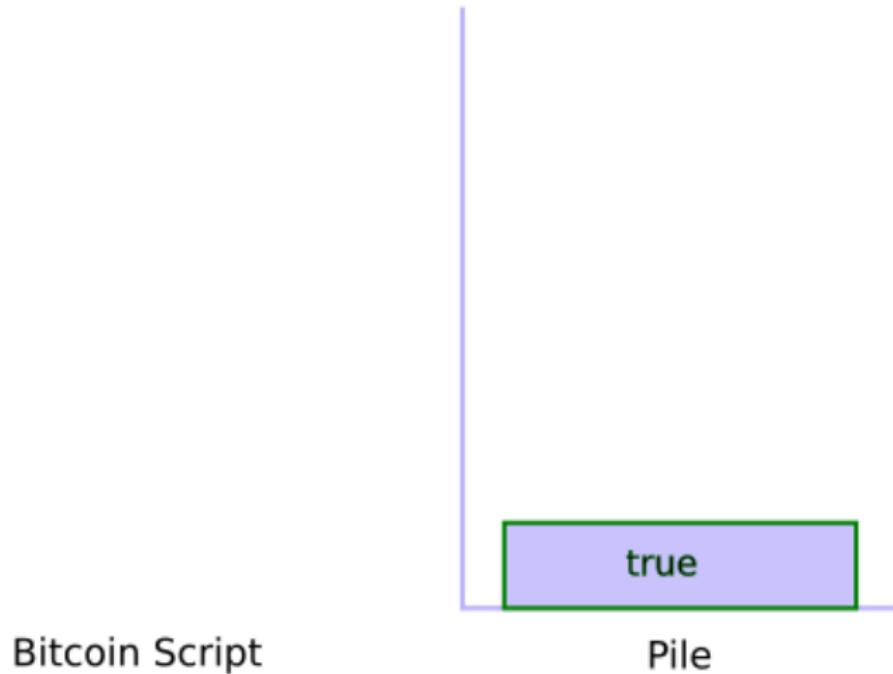
## Execution du Script "Payer à ..."



## Execution du Script "Payer à ..."



## Execution du Script "Payer à . . ."



# Sommaire

## 1. Intro

- Les Cypher-punks

## 2. Transactions Bitcoin

- Transaction non dépensés uTXO
- Bitcoin Script
- **Les Limites du modèle Bitcoin**

## 3. Etherum

- Avantages d'Etherum
- Similitudes Etherum - Bitcoin
- Différence Etherum - Bitcoin
- Exécution d'un contrat
- Applications

## 4. Références

- Références

- ▶ pas d'universalité
- ▶ Ne connais pas la valeur de l'utxo
- ▶ Une utxo est soit dépensée ou pas
  - ▶ Il n'y a pas d'état intermédiaire
- ▶ Le script n'a pas accès au nonce, ni aux infos de la blockchain

# Sommaire

## 1. Intro

- Les Cypher-punks

## 2. Transactions Bitcoin

- Transaction non dépensés uTXO
- Bitcoin Script
- Les Limites du modèle Bitcoin

## 3. Etherum

### ■ Avantages d'Etherum

- Similitudes Etherum - Bitcoin
- Différence Etherum - Bitcoin
- Exécution d'un contrat
- Applications

## 4. Références

- Références

# Ce qu'Etherum apporte de nouveau

- ▶ Accès à l'état de toute la blockchain
- ▶ Permet des Applications (Universelles)
- ▶ est moins énergivore (POW vs POS)

# Sommaire

## 1. Intro

- Les Cypher-punks

## 2. Transactions Bitcoin

- Transaction non dépensés uTXO
- Bitcoin Script
- Les Limites du modèle Bitcoin

## 3. Etherum

- Avantages d'Etherum
- **Similitudes Etherum - Bitcoin**
- Différence Etherum - Bitcoin
- Exécution d'un contrat
- Applications

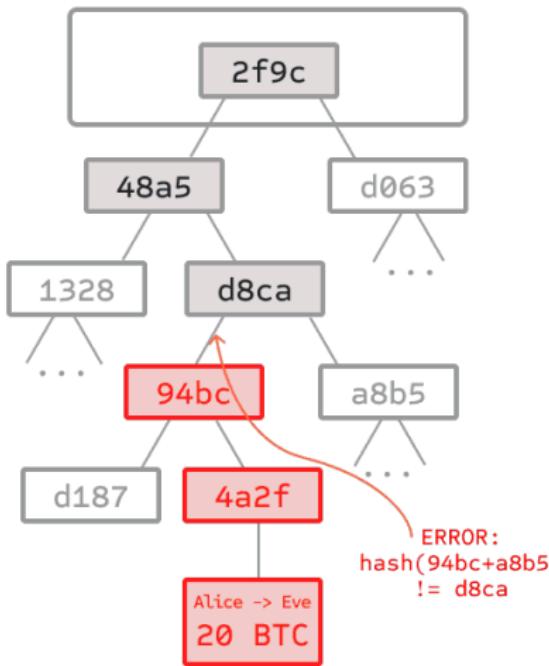
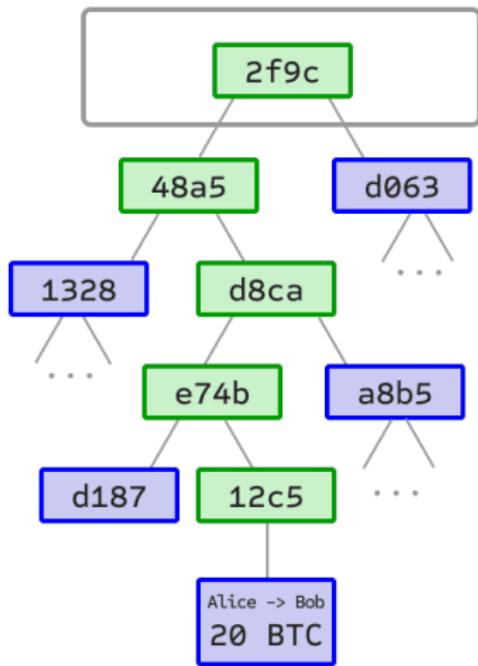
## 4. Références

- Références

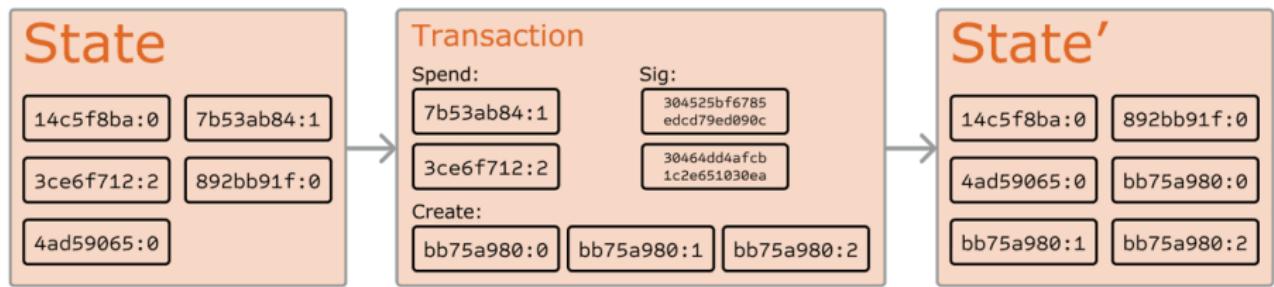
# Chaine de block



# Arbre de Merkle

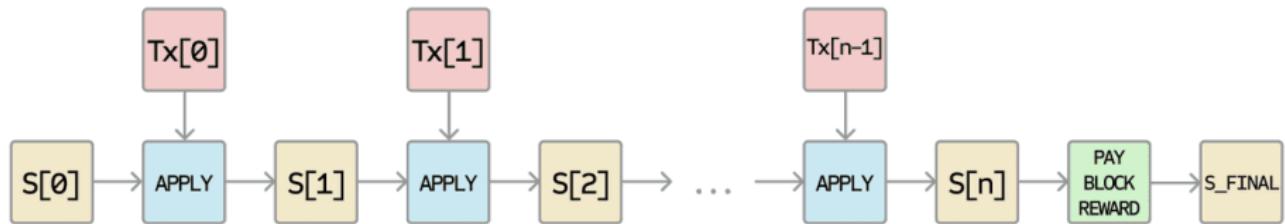


# Etats et transitions



Les blockchain Etherum et Bitcoin peuvent être vue comme des systèmes à transition d'états

# Etats et transitions



Les blockchain Ethereum et Bitcoin peuvent être vues comme des systèmes à transition d'états

# Sommaire

## 1. Intro

- Les Cypher-punks

## 2. Transactions Bitcoin

- Transaction non dépensés uTXO
- Bitcoin Script
- Les Limites du modèle Bitcoin

## 3. Etherum

- Avantages d'Etherum
- Similitudes Etherum - Bitcoin
- **Différence Etherum - Bitcoin**
- Exécution d'un contrat
- Applications

## 4. Références

- Références

# Un autre modèle de transaction

## Compte Etherum

- ▶ Nonce
- ▶ Balance du compte
- ▶ Code du contrat
- ▶ Espace de stockage
  - ▶ stack : FIFO
  - ▶ memory : un tableau de taille variable
  - ▶ storage : pour le long terme

### State'

14c5f8ba:  
- 1014 eth

bb75a980:  
- 5212 eth

```
if !contract.storage[tx.data[0]]:  
    contract.storage[tx.data[0]] = tx.data[1]  
[0, 235235, CHARLIE, ALICE ...]
```

892bf92f:  
- 0 eth

```
send(tx.value / 3, contract.storage[0])  
send(tx.value / 3, contract.storage[1])  
send(tx.value / 3, contract.storage[2])
```

[ALICE, BOB, CHARLIE]

4096ad65  
- 77 eth

Exemple d'un nouvel état

# Un autre modèle de transaction

## Compte Etherum

- ▶ Nonce
- ▶ Balance du compte
- ▶ Code du contrat
- ▶ Espace de stockage
  - ▶ stack : FIFO
  - ▶ memory : un tableau de taille variable
  - ▶ storage : pour le long terme

### State'

14c5f8ba:  
- 1014 eth

bb75a980:  
- 5212 eth

```
if !contract.storage[tx.data[0]]:  
    contract.storage[tx.data[0]] = tx.data[1]  
[0, 235235, CHARLIE, ALICE ...]
```

892bf92f:  
- 0 eth

```
send(tx.value / 3, contract.storage[0])  
send(tx.value / 3, contract.storage[1])  
send(tx.value / 3, contract.storage[2])
```

[ALICE, BOB, CHARLIE]

4096ad65  
- 77 eth

Exemple d'un nouvel état

# Messages ou transactions Etherum

Ils peuvent

- ▶ être créée *on-chain*
- ▶ contenir des données
- ▶ être des réponses à un autre message
- ▶ accéder à tout l'état de la blockchain

## Deux exemples de "smart contract"

Remplace une donnée sur la blockchain

```
if !contract.storage[msg.data[0]]:  
    contract.storage[msg.data[0]] = msg.data[1]
```

Envoie un montant à quelqu'un

```
def send(to, val):  
    if self.storage[msg.sender] >= val:  
        self.storage[msg.sender] = self.storage[msg.sender] - val  
        self.storage[to] = self.storage[to] + val
```

# Sommaire

## 1. Intro

- Les Cypher-punks

## 2. Transactions Bitcoin

- Transaction non dépensés uTXO
- Bitcoin Script
- Les Limites du modèle Bitcoin

## 3. Etherum

- Avantages d'Etherum
- Similitudes Etherum - Bitcoin
- Différence Etherum - Bitcoin
- Exécution d'un contrat**
- Applications

## 4. Références

- Références

# Exécution d'un contrat

## Objectif

- ▶ Les frais sont limités par STARTGAS x GASPRICE
- ▶ En cas d'erreur tout doit aller au mineur
- ▶ En cas de réussite le mineur ne prend que ce qui a été consommé et le reste retourne au créateur de la transaction

## Plus précisement

- 1 : Vérifier le format de la transactoin
- 2 : Prendre STARTGAS \* GASPRICE dans balance de l'envoyeur
- 3 : Enlève frais de la transaction dans les STARTGAS, 5 gas / byte
- 4 : Fais le transfert de valeur
- 5 : Fait tourner le code dans EVM
- 6 : calcule combien de GAS on été dépensé et renvoie le reste à l'envoyeur

# Sommaire

## 1. Intro

- Les Cypher-punks

## 2. Transactions Bitcoin

- Transaction non dépensés uTXO
- Bitcoin Script
- Les Limites du modèle Bitcoin

## 3. Etherum

- Avantages d'Etherum
- Similitudes Etherum - Bitcoin
- Différence Etherum - Bitcoin
- Exécution d'un contrat
- Applications

## 4. Références

- Références

# Application purement financières

- ▶ Sous-monnaies
  - ▶ stable coins
- ▶ produits dérivés
- ▶ contrat hedging
  - ▶ assurances, SchellingCoin
- ▶ Compte d'épargne
- ▶ Application notariale
- ▶ contrat de travail
- ▶ loterie
- ▶ marché distribués



# Applications semi-financières

## Monnaies et données

Être payés pour, ou fournir, des données

## gestions des identités

- ▶ gestion de nom de domaine (nameCoin)
- ▶ gestoin d'ID
- ▶ gestion de fichiers
- ▶ gestion de contenu
- ▶ réseau sociaux décentralisé



# Applications semi-financières

## Monnaies et données

Être payés pour, ou fournir, des données

## gestions des identités

- ▶ gestion de nom de domaine (nameCoin)
- ▶ gestoin d'ID
- ▶ gestion de fichiers
- ▶ gestion de contenu
- ▶ réseau sociaux décentralisé

## Autres Applications :

- ▶ Calcul distribué (seti@home, folding@home)

# Applications non-financières

- ▶ Vote en ligne
- ▶ Gouvernance décentralisé



# Applications Autonome et Décentralisé

## DA : Decentralised Autonomous

- ▶ DAC communauté (equal vote)
- ▶ DAC corporation (votre proportional to share)

## DAO : Decentralised Autonomous Organisation

Peut fonctionner avec des individus ne parlant pas la même langue

- ▶ Automatise de la gouvernance

Code modifiable car stoqué dans la zone de stoquage des contrats

- ▶ modifié à l'aide de pointeurs

# Divers

- ▶ ICO : Initial coin offering
- ▶ ASIC : Application specific integrated circuits

# Sommaire

## 1. Intro

- Les Cypher-punks

## 2. Transactions Bitcoin

- Transaction non dépensés uTXO
- Bitcoin Script
- Les Limites du modèle Bitcoin

## 3. Etherum

- Avantages d'Etherum
- Similitudes Ethereum - Bitcoin
- Différence Ethereum - Bitcoin
- Exécution d'un contrat
- Applications

## 4. Références

- Références

## Article fondateur des smart-contracts

- ▶ Whitepaper d'Ethereum ([pdf](#))

## CV de personnages célf

- ▶ Adam Back : [vidéo](#)
- ▶ Nick Szabo : [vidéo](#)
- ▶ Wei Dai

## Liens pour une introduction aux smart-contracts sur Cardano

- ▶ <https://play.marlowe-finance.io>
- ▶ <https://run.marlowe-finance.io/>