

# Ethereum et smart contracts

ALYRA



# Un peu d'histoire

---

Vitalik Buterin

Ethereum a été inventé fin 2013 par Vitalik Buterin

Nick Szabo

est un informaticien, juriste et cryptographe connu pour ses travaux de recherche sur les contrats numériques et la monnaie numérique

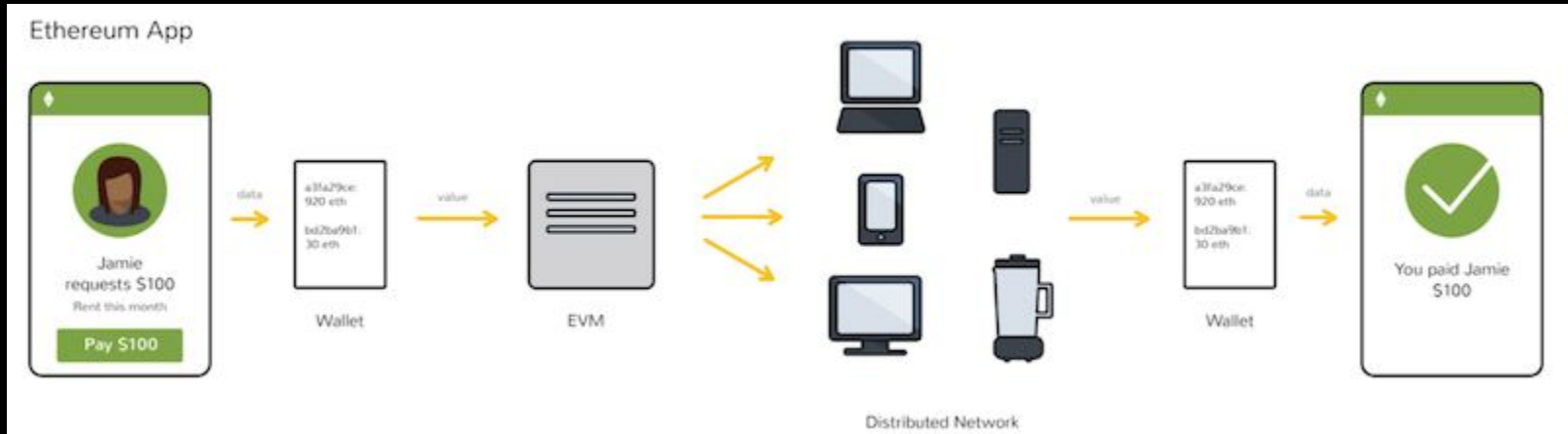


# Ethereum

---

- Ethereum est un réseau géant qui consiste en un énorme nombre d'ordinateurs connecté entre eux.
- Ce réseau est appelé Ethereum Virtual Network (EVN) et fonctionne un peu comme un super-ordinateur, où toutes les transactions sont enregistrées sur chacun des ordinateurs du réseau.
- L'ETH est la crypto-monnaie permettant de faire fonctionner le réseau en étant utilisée comme "carburant".
- Une des fonctionnalités innovantes de la blockchain d'ethereum était l'introduction des smart contracts.

# Application Ethereum



# Qu'est ce que le gas

---

Le gas est une unité qui sert à mesurer le temps de travail informatique nécessaire pour faire fonctionner un smart contract sur le réseau Ethereum.

## Pour simplifier

C'est un système à peu près équivalent à celui des kilowatts pour mesurer la consommation électrique. On finira au final par payer en €, mais on mesure l'électricité consommée en KWH (kilowatts par heure).

# Bitcoin vs Ethereum

---

« La blockchain de Bitcoin a été conçue spécifiquement pour des applications monétaires, alors qu'Ethereum permet de créer tout type d'applications »

le fondateur d'Ethereum, Vitalik Buterin.

# Smart Contract

---

- Programme informatique qui est chargé d'exécuter les termes d'un contrat une fois que toutes les conditions sont réunies
- Les smart contracts sont enregistrés et exécutés sur la blockchain

```
contract Escrow {
    address PartyA;
    address payable PartyB;
    address notary;
    mapping(address => uint256) private _deposits;
    bool complete;

    constructor () {
        notary = msg.sender;
    }

    function depositsOf(address payee) public view returns (uint256) {
        return _deposits[payee];
    }

    function deposit(address payee) public payable {
        uint256 amount = msg.value;
        _deposits[payee] = _deposits[payee] + amount;
    }

    function validate() public payable {
        require(msg.sender == notary);
        require(_deposits[PartyA]);
        complete = true
    }

    function withdraw() public onlyPrimary {
        require(complete && msg.sender == PartyB);
        PartyB.transfer(address(this).balance);
    }
}
```

# Smart Contract

---

## Caractéristiques

- Une adresse publique sans clé privée
- Code déterminé

## Limites

- Pas de mécanisme de mise à jour
- Pas de récupération d'information
- Pas de déclenchement automatique



# remix.ethereum.org

The screenshot displays the Remix IDE interface. On the left is a vertical sidebar with icons for file explorer, console, and other tools. The main workspace is divided into several sections:

- FILE EXPLORERS**: A sidebar on the left showing a file tree with files like `SceneOuvrte.sol`, `scenario.json`, `Casino.sol`, and `ballot.sol`.
- Home**: A central panel with a blue header announcing "The new layout has arrived" with buttons for "Learn more" and "Use previous version".
- Environments**: A section with buttons for "Solidity", "Vyper", and "Workshops".
- File**: A section with links for "New File", "Open Files", "Connect to Localhost", and "Import From:" with buttons for "Gist", "GitHub", "Swarm", "Ipfs", "https", and "Resolver-engine".
- Featured Plugins**: A section with links for "Pipeline" and "Debugger", and a button for "See all Plugins".
- Resources**: A section with links for "Documentation", "Gitter channel", "Medium Posts", and "Tutorials".
- Bottom Panel**: A console area with a search bar and a list of items, including "listen on network" and a search bar with the text "Search with transaction hash or address". Below this is a code editor with a snippet of Solidity code: `Use exports.register(key, obj).remove(key).clear() to register and reuse object across script executions.`



# Scène ouverte

---

```
pragma solidity ^0.5.6;  
  
contract SceneOuverte {  
  
}
```

# Scène ouverte

---

```
pragma solidity ^0.5.6;

contract SceneOuverte {
    string[12] public passagesArtistes;
    uint public creneauxLibres = 12;
}
```

# Scène ouverte

---

```
pragma solidity ^0.5.6;

contract SceneOuverte {
    string[12] public passagesArtistes;
    uint public creneauxLibres = 12;

    function sInscrire(string memory nomDArtiste) public {
        passagesArtistes[12-creneauxLibres] = nomDArtiste;
        creneauxLibres -= 1;
    }
}
```

# Scène ouverte

---

```
pragma solidity ^0.5.6;

contract SceneOuverte2 {

    string[12] public passagesArtistes;
    uint public creneauxLibres = 12;
    uint tour;

    function sInscrire(string memory nomDArtiste) public {
        if(creneauxLibres>0){
            passagesArtistes[12-creneauxLibres] = nomDArtiste;
            creneauxLibres -= 1;
        }
    }
}
```

# Scène ouverte

---

```
pragma solidity ^0.5.5;

contract SceneOuverte {

    string[12] public passagesArtistes;
    uint public creneauxLibres = 12;
    uint tour;

    function sInscrire(string memory nomDArtiste) public {
        if(creneauxLibres>0){
            passagesArtistes[12-creneauxLibres] = nomDArtiste;
            creneauxLibres -= 1;
        }
    }

    function passerArtisteSuivant() public {...
    }

    function artisteEnCours() public view returns (string memory){...
    }
}
```

# Scène ouverte

---

```
pragma solidity ^0.5.6;

contract SceneOuverte {

    string[12] public passagesArtistes;
    uint public creneauxLibres = 12;
    uint tour;

    function sInscrire(string memory nomDArtiste) public {
        if(creneauxLibres>0){
            passagesArtistes[12-creneauxLibres] = nomDArtiste;
            creneauxLibres -= 1;
        }
    }

    function passerArtisteSuivant() public {
        tour += 1;
    }

    function artisteEnCours() public view returns (string memory){
        if (tour< (12 - creneauxLibres)){
            return passagesArtistes[tour];
        } else {
            return "FIN";
        }
    }
}
```



# Lire l'état

---



```
pragma solidity ^0.5.6;

contract SceneOuvrte {

    string[12] public passagesArtistes;
    uint public creneauxLibres = 12;
    uint tour;

    function sInscrire(string memory nomDArtiste) public {
        if(creneauxLibres>0){
            passagesArtistes[12-creneauxLibres] = nomDArtiste;
            creneauxLibres -= 1;
        }
    }

    function passerArtisteSuivant() public {
        tour += 1;
    }

    function artisteEnCours() public view returns (string memory){
        if (tour< (12 - creneauxLibres)){
            return passagesArtistes[tour];
        } else {
            return "FIN";
        }
    }
}
```



# Compile

---

**SOLIDITY COMPILER**

Compiler

0.5.11+commit.c082d0b


Include nightly builds ☐

Language

Solidity

EVM Version

compiler default

 Compile SceneOuverte.sol

Compiler Configuration

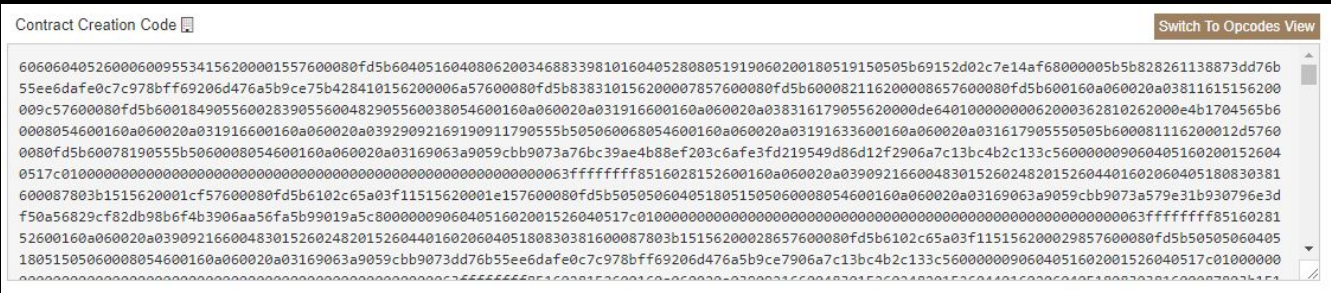
☒ Auto compile

☐ Enable optimization

☐ Hide warnings



**Error**

```
:21:3: DeclarationError: Identifier not found or not unique.  
  Participant[] participants;  
  ^-----^
```





# Déploiement

Environment

JavaScript VM  VM (-) ▼ 

Account +

0xca3...a733c (99.99999999999956765) ▼  


Gas limit

3000000

Value

0

wei ▼


SceneOuvrte ▼ 


Deploy

or



At Address 

Load contract from Address

Transactions recorded: 1 















Deployed Contracts 

▶

SceneOuvrte at 0x692...77b3a (memory)  

# Déploiement

✓ [vm] from:0xca3...a733c to:Scene0uverte.(constructor) value:0 wei data:0x608...00029  
logs:0 hash:0x78c...1fb77

status	0x1 Transaction mined and execution succeed
transaction hash	0x78c73c35acffaa075b6cb2ea9635de6d0ba643af2cbb01205a0f65b65681fb77 
contract address	0x692a70d2e424a56d2c6c27aa97d1a86395877b3a 
from	0xca35b7d915458ef540ade6068dfe2f44e8fa733c 
to	Scene0uverte.(constructor) 
gas	3000000 gas 
transaction cost	432342 gas 
execution cost	290718 gas 
hash	0x78c73c35acffaa075b6cb2ea9635de6d0ba643af2cbb01205a0f65b65681fb77 
input	0x608...00029 
decoded input	{ } 
decoded output	- 
logs	[ ]  
value	0 wei 

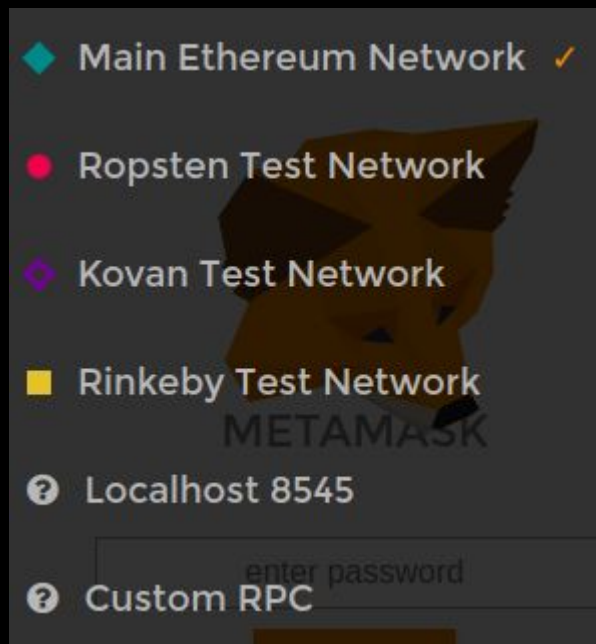
# Les réseaux

---

Main : réseau principal

Ropsten, Kovan, Rinkeby, Sokol réseaux de test

Localhost 8545 : noeud local ( réseau principal, de test, ou local selon options )



# Déploiement

CONFIRM TRANSACTION

Ropsten Test Net

Main Xaxa

8D76B0...7468

5.993 ETH

>

New Contract

Amount

0 ETH

Gas Limit

UNITS

Gas Price

GWEI

Max Transaction Fee

0.020200 ETH

Max Total

0.020200 ETH

Data included: 812 bytes

RESET

ACCEPT

REJECT

Ropsten Test Net

METAMASK

MetaMask

Main Xaxa

0x8D76B...

5.993 ETH

BUY

SEND

HISTORY

May 31 2017 20:13

0x467782A5...71F0

0 ETH

May 31 2017 20:10

Contract Published

0 ETH

# Liens intéressants

---

- [Qu'est ce que Ethereum ?](#)
- [Qu'est ce qu'un smart contract ?](#)
- [Le fonctionnement des smart contracts sur Ethereum](#)
- [Ethereum gas, gas price et gas limit](#)