Introduction à la Blockchain Ethereum

Jean-Patrick GELAS
Université Claude Bernard – Lyon 1
Février 2021

Agenda

Nous ne parlerons pas (ou peu) des...

- Algorithmes de consensus (PoW, PoS, PoA, DPoS,...)
- Plateformes d'échanges (Binance, Kraken,...)
- Comment miner de la crypto monnaie
- Comment devenir **crypto millionaire** ! :-)

Blockchain : Rappel

- Structure de données simple
- La technologie Blockchain : « Base de données » sécurisée et décentralisée.

Démo : https://anders.com/blockchain/

Les mineurs

- Héberge une copie de la blockchain
- Ajoutent de nouvelles liste de transactions (*i.e.* des blocs) à la chaîne.
- Vérifient l'intégrité de la blockchain
- Génèrent de nouveaux *coins*
- (Exécutent les Smart Contracts)

Ethereum : Définition

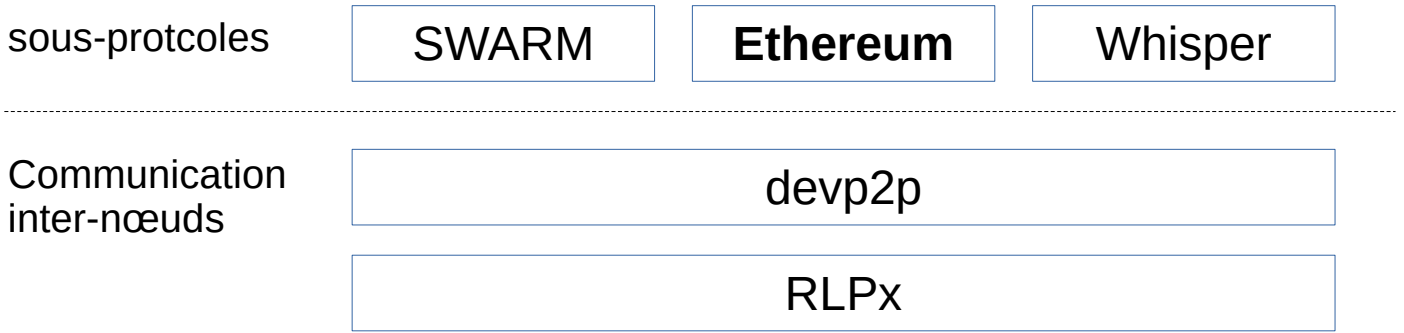
« Protocole d'échanges décentralisés permettant la création par les utilisateurs de contrats intelligents grâce à un langage Turing-complet. »

- Wikipedia

Ethereum

- Blockchain de seconde génération.
- Développée par ***Vitalik Buterin***, lancée en juillet 2015.
- Fréquence moyenne des blocs : 14-15 secondes
- Taille des blocs : dynamique
- Symbole boursier : ETH
- Quantité maximale : non limitée

Architecture Ethereum



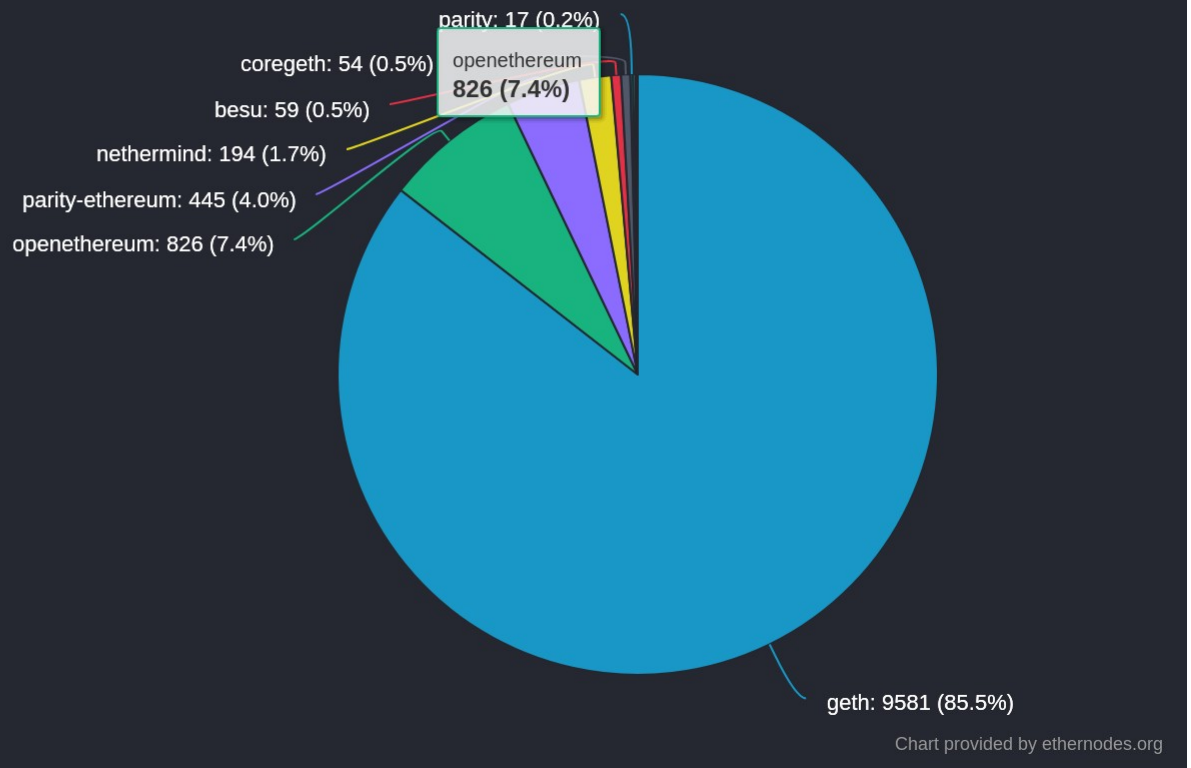
- ***devp2p***: Abstraction de la couche matériel pour communiquer entre les nœuds.
- **Ethereum** : La blockchain que nous connaissons permettant de stocker la logique des *Dapps* (smart-contract).
- **Swarm** : Un système de fichiers décentralisé proche de IPFS permettant de servir les *Dapps*.
- **Whisper** : Un protocole de communication décentralisé pour communiquer entre les *Dapps*.

Ethereum Mainnet Statistics – CLIENTS – https://wwv.ethernodes.org

11000 nœuds contre 10500 pour Bitcoin (février 2021) https://bitnodes.io/

CLIENTS

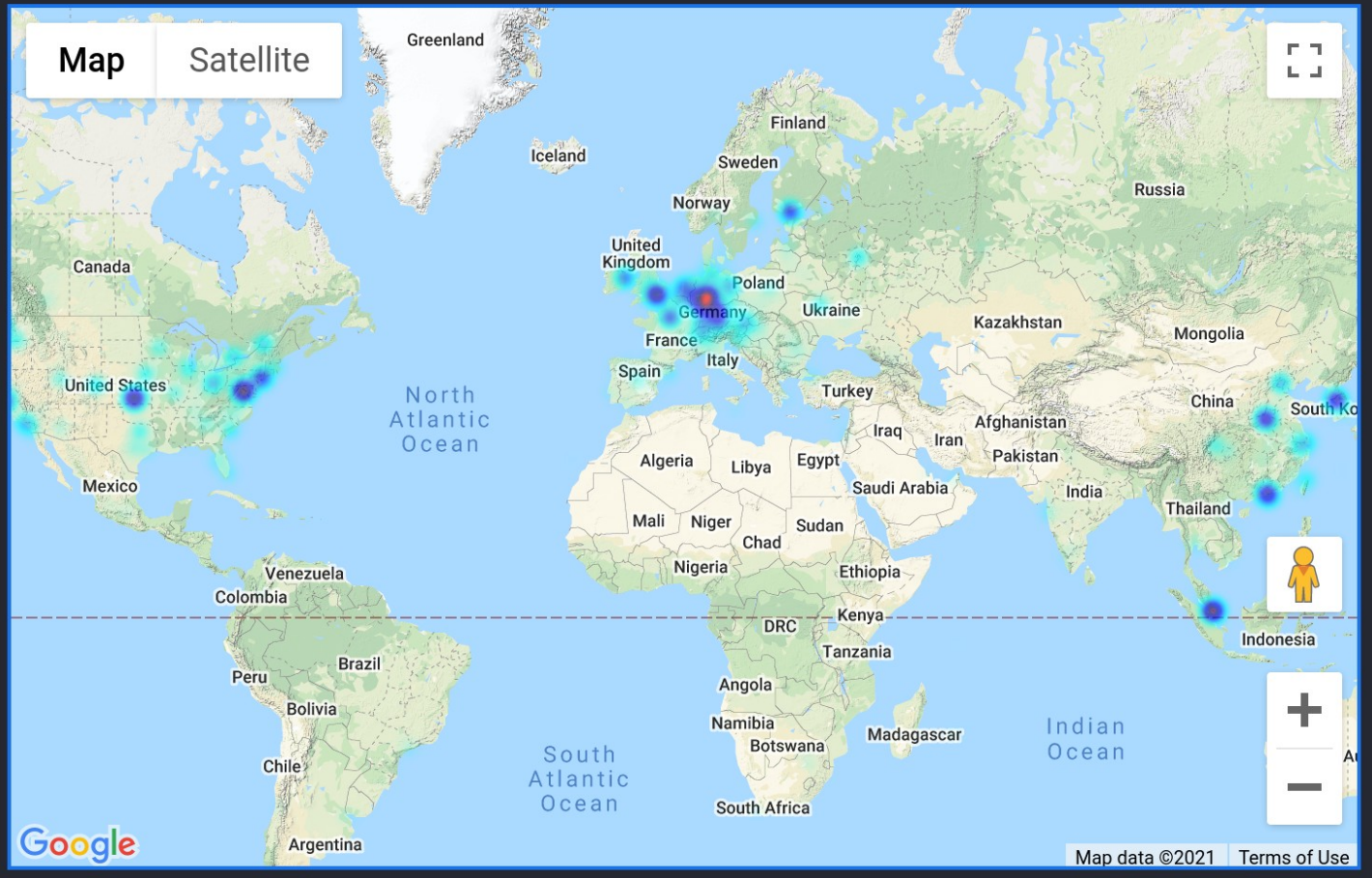
Total	11203 (100%)
geth	9581 (85.52%)
openethereum	826 (7.37%)
parity-ethereum	445 (3.97%)
nethermind	194 (1.73%)
besu	59 (0.53%)
coregeth	54 (0.48%)
parity	17 (0.15%)
multigeth	16 (0.14%)



Ethereum Mainnet Statistics – COUNTRIES – https://ww.ethernodes.org

COUNTRIES

Total	11176 (100%)
United States	2949 (26.39%)
Germany	2084 (18.65%)
China	931 (8.33%)
Singapore	604 (5.40%)
United Kingdom	525 (4.70%)
Japan	417 (3.73%)
South Korea	369 (3.30%)
France	294 (2.63%)
Hong Kong	272 (2.43%)



Smart Contract

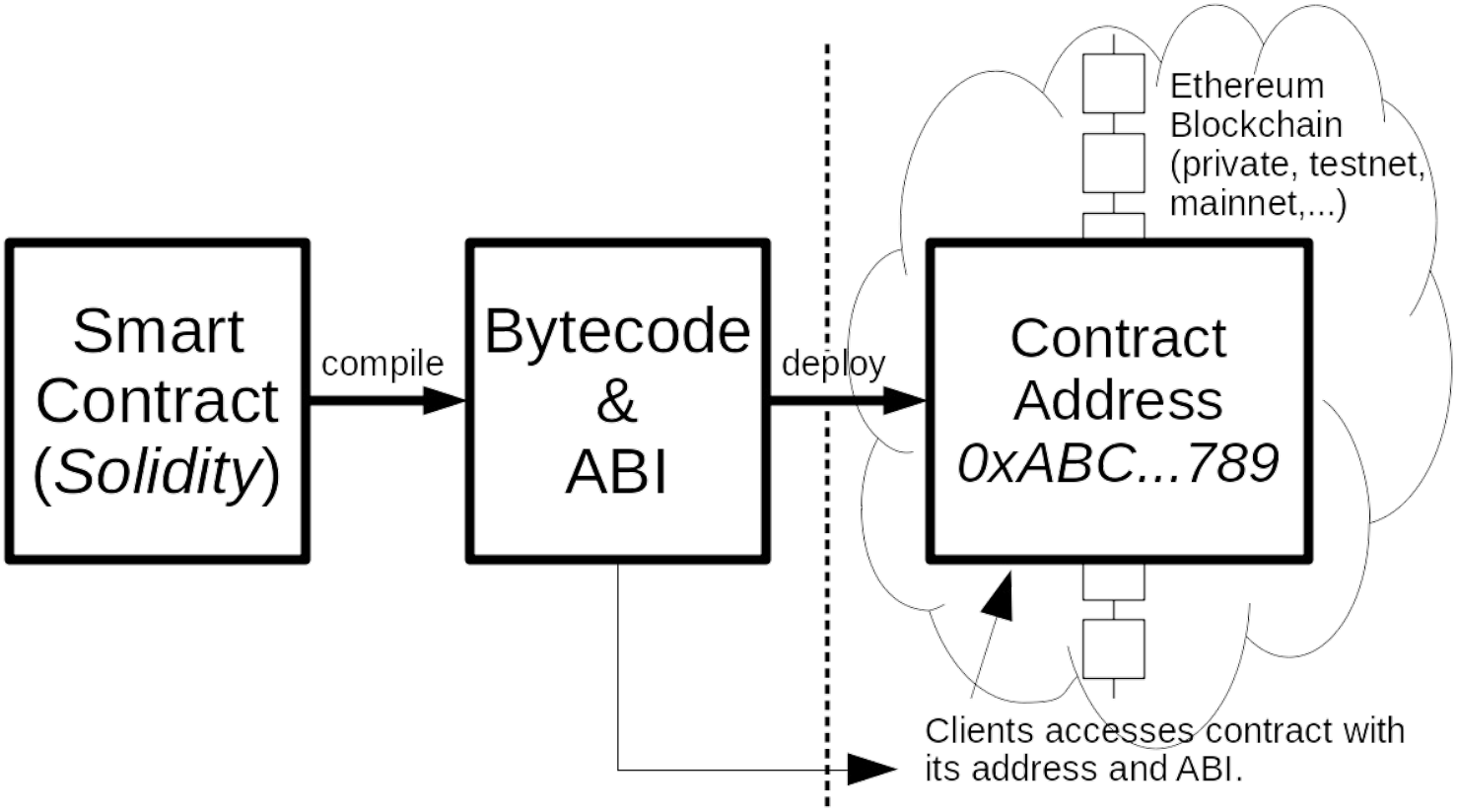
- Programme autonome
- Déployé et répliqué
- Non modifiable
- Adapté pour gérer des transactions

Smart Contract

- Du *bytecode* stocké dans la blockchain
- Rédigé dans un langage de haut niveau : *Solidity*
- Compilé (solc)
- Accessible via une adresse codée sur 160 bits
- Exécuté dans l'Ethereum Virtual Machine (EVM)

0x71c7656ec7ab88b098defb751b7401b5f6d8976f

Cycle de vie du Smart Contract



ABI : Application Binary Interface

Code machine

"opcodes": "PUSH1 0x80 PUSH1 0x40 MSTORE
CALLVALUE DUP1 ISZERO PUSH2 0x10 JUMPI PUSH1 0x0
DUP1 REVERT JUMPDEST POP PUSH1 0x40 MLOAD PUSH1
0x20 DUP1 PUSH2 0x487 DUP4 CODECOPY DUP2 ADD
PUSH1 0x40 SWAP1 DUP2 MSTORE SWAP1 MLOAD PUSH1..."

"object":
"6080604052348015610010576000 80fd5b5060405160
208061048783398101604090815290516000805460016
0a060020a031916331780825560160a060020a031681
526001602081905292902091909155 60ff81166100606..."

Smart Contract

- Le *bytecode* est déployé sur Ethereum dans une transaction (dans le champ *data* de la transaction).
- Une fois sur la blockchain le contrat ne peut être ni modifié ni supprimé.
- Il est accessible sur tous les nœuds à jour.
- Pour interagir avec le contrat il faut une machine virtuelle qui interprète le *bytecode*.
- L'EVM (Ethereum Virtual Machine) mise à disposition par les clients Ethereum qui gère un nœuds (ex : Geth, Parity,...)

Clients Ethereum

- Coté client on a un *provider* qui se connecte à un nœud en particulier et qui communique avec l'EVM avec des requêtes au format JSON-RPC.
- Les clients Ethereum proposent une API avec un ensemble de méthodes JSON-RPC.

curl -X POST --data '{"jsonrpc":"2.0","method":"**web3_clientVersion**","params":[],"id":67}' http://127.0.0.1:8545

{"id":67,"jsonrpc":"2.0","result":"**EthereumJS TestRPC/v2.3.3/ethereum-js**"}

Communiquer avec Ethereum

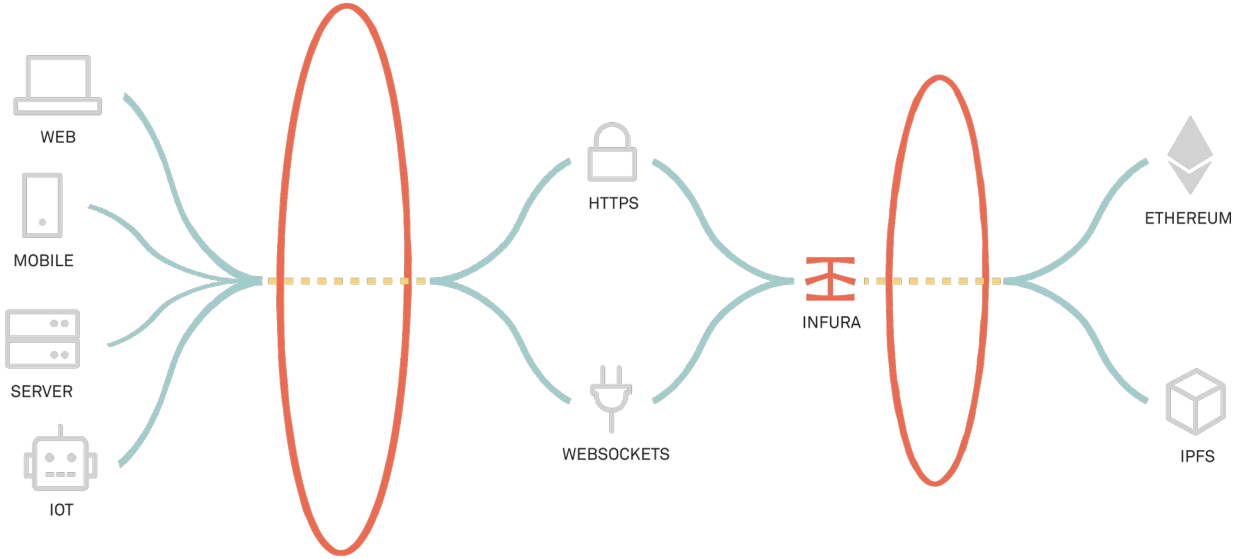
- Envoyez des requêtes JSON-RPC avec son propre *provider* n'est pas forcément simple à mettre en place et à manipuler.
- *Web3.js :* Un des projets les plus utilisé sur Ethereum propose d'englober ce *provider* et ces requêtes avec une API JavaScript (existe aussi dans d'autres langages).

- Une alternative populaire à Web3 est *ethers.js*

https://blog.infura.io/ethereum-javascript-libraries-web3-js-vs-ethers-js-part-i/

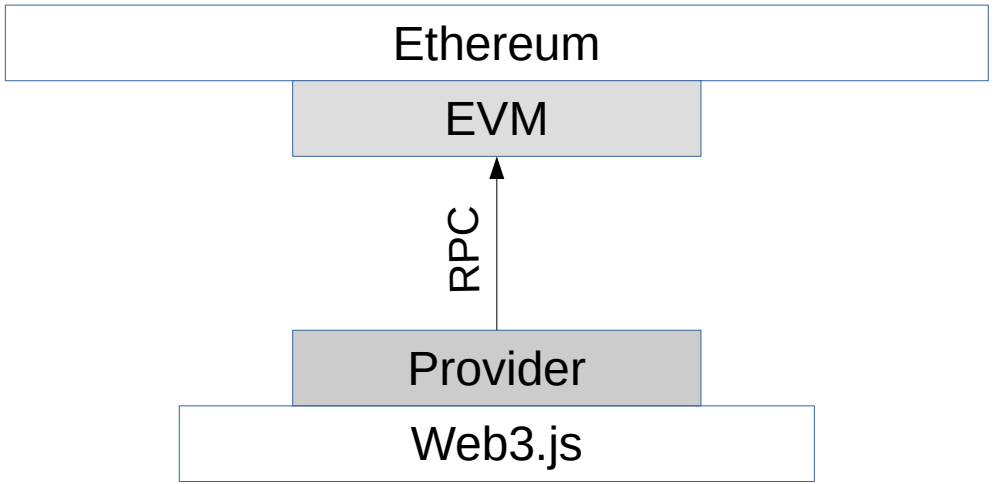
Web3.js / Ethers.js

- Librairies JavaScript simple d'utilisation permettant d'interagir avec un nœud.
- La première chose à faire est de leur donner un *provider* avec l'adresse du nœud à contacter.
- Idéalement le nœud auquel se connecte Web3 ou Ethers tourne sur la machine de l'utilisateur.
- Mais de plus en plus les utilisateurs se connectent à des nœuds distants grâce à des services comme Infura (https://infura.io/)



INFURA

Web3 pour interagir avec la blockchain



JavaScript API
(Client/Serveur)

Dapps

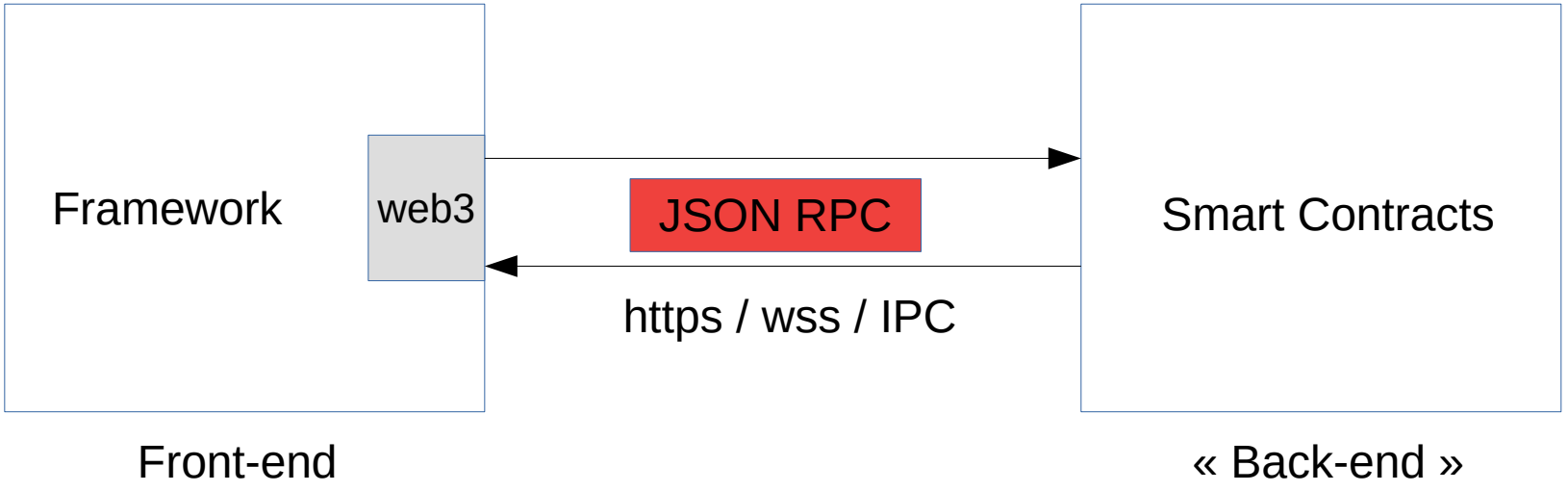
- Application accessible de façon décentralisée (en principe).

Pas de serveur centralisé

  - ni pour gérer une « base de données »,
  - *ni pour servir l'application,*
  - *ni pour communiquer entre plusieurs applications.*

- Le projet Ethereum propose une solution pour chacun de ces points
- *Swarm*, *Whisper* ne sont pas encore suffisamment mature...
- Une Dapp se "limite" donc à la connexion entre une application communiquant avec des smart-contracts sans s'appuyer sur un serveur pour gérer une base de données centralisée.

Dapp


Framework
web3
JSON RPC
https / wss / IPC
Smart Contracts
Front-end
« Back-end »

Le langage Solidity

- Langage de haut niveau
- Influencé par C++, Python et Javascript.
- Typé statiquement.
- Supporte l'héritage,
- l'appel à des bibliothèques,
- la définition de type complexe par les utilisateurs.

Hello World

```
pragma solidity ^0.4.18;

contract Hello {
	string message = "Default message";

	function getMessage () public view returns (string) {
		return message;
	}
	function setMessage (string _message) public payable {
		message = _message;
	}
}
```

Simple Token

```
pragma solidity ^0.4.20;

contract MyToken {
    /* This creates an array with all balances */
    mapping (address => uint256) public balanceOf;

    /* Initializes contract with initial supply tokens to the creator of the contract */
    function MyToken(
        uint256 initialSupply
        ) public {
        balanceOf[msg.sender] = initialSupply;              // Give the creator all initial tokens
    }

    /* Send coins */
    function transfer(address _to, uint256 _value) public returns (bool success) {
        require(balanceOf[msg.sender] >= _value);           // Check if the sender has enough
        require(balanceOf[_to] + _value >= balanceOf[_to]); // Check for overflows
        balanceOf[msg.sender] -= _value;                    // Subtract from the sender
        balanceOf[_to] += _value;                           // Add the same to the recipient
        return true;
    }
}
```

Nexium (NXC) – Token ERC-20

```
contract Nexium {
    ...
    function Nexium() {
        initialSupply = 100000000000;
        balanceOf[msg.sender] = initialSupply;
        name = 'Nexium';
        symbol = 'NxC';
        decimals = 3;
        burnAddress = 0x1b320000000000000000000000000000000000
    }
    function totalSupply() returns(uint){
        return initialSupply - balanceOf[burnAddress];
    }
    function transfer(address _to, uint256 _value) ...
    function transferFrom(address _from, ...
```

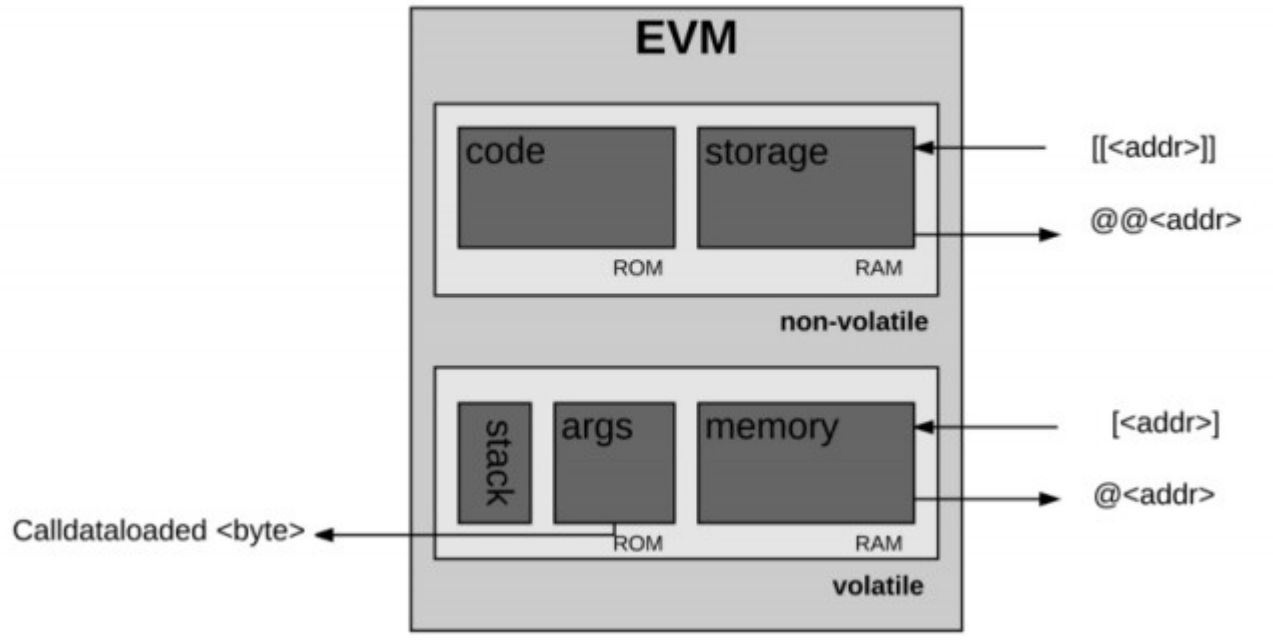
0x (ZRX) – Token ERC-20

```
contract ZRXToken is UnlimitedAllowanceToken {

    uint8 constant public decimals = 18;
    uint public totalSupply = 10**27; // 1 billion tokens
    string constant public name = "0x Protocol Token";
    string constant public symbol = "ZRX";

    function ZRXToken() {
        balances[msg.sender] = totalSupply;
        ...
```

Ethereum Virtual Machine



Ethereum Virtual Machine

- Machine (*quasi-*) Turing complète
- Environnement d'exécution des Smart Contracts
- Émule une machine 256 bits avec des pseudo-registres
- Registres émulés par une *stack* virtuelle.

Ethereum et unités de mesure

- **Ether** (ETH) : le nom de la crypto monnaie
- **Wei** : une fraction d'Ether (1 ETH = 10^{18} Wei)
- **GAS** : unité de mesure en terme de quantité de calcul
- **GAS price** : défini le prix (en GWei) que vous êtes prêt à payer au mineur.
- **GAS limit** : Quantité maximum de GAS que vous êtes prêt à payer pour une transaction.

Analogie

- GAS limit : capacité du réservoir d'une voiture en litre.
- GAS price le prix du litre de carburant.
  - Voiture : 1,50 EUR (prix) par litre (unité)
  - Ethereum : 20 GWei (prix) par GAS (unité)
- Pour remplir le réservoir il faut :
  - 50 litres à 1,50 EUR = 75 EUR
  - 21000 unités de GAS à 20 GWei = 0.00042 ETH

Remarques

- Fixer un *GAS limit* évite de dépenser une fortune en cas de problème.
- La quantité de GAS requise est définie par le coût et la quantité d'instructions exécutées.
- Fixer un *GAS limit* trop petit a peu d'intérêt.

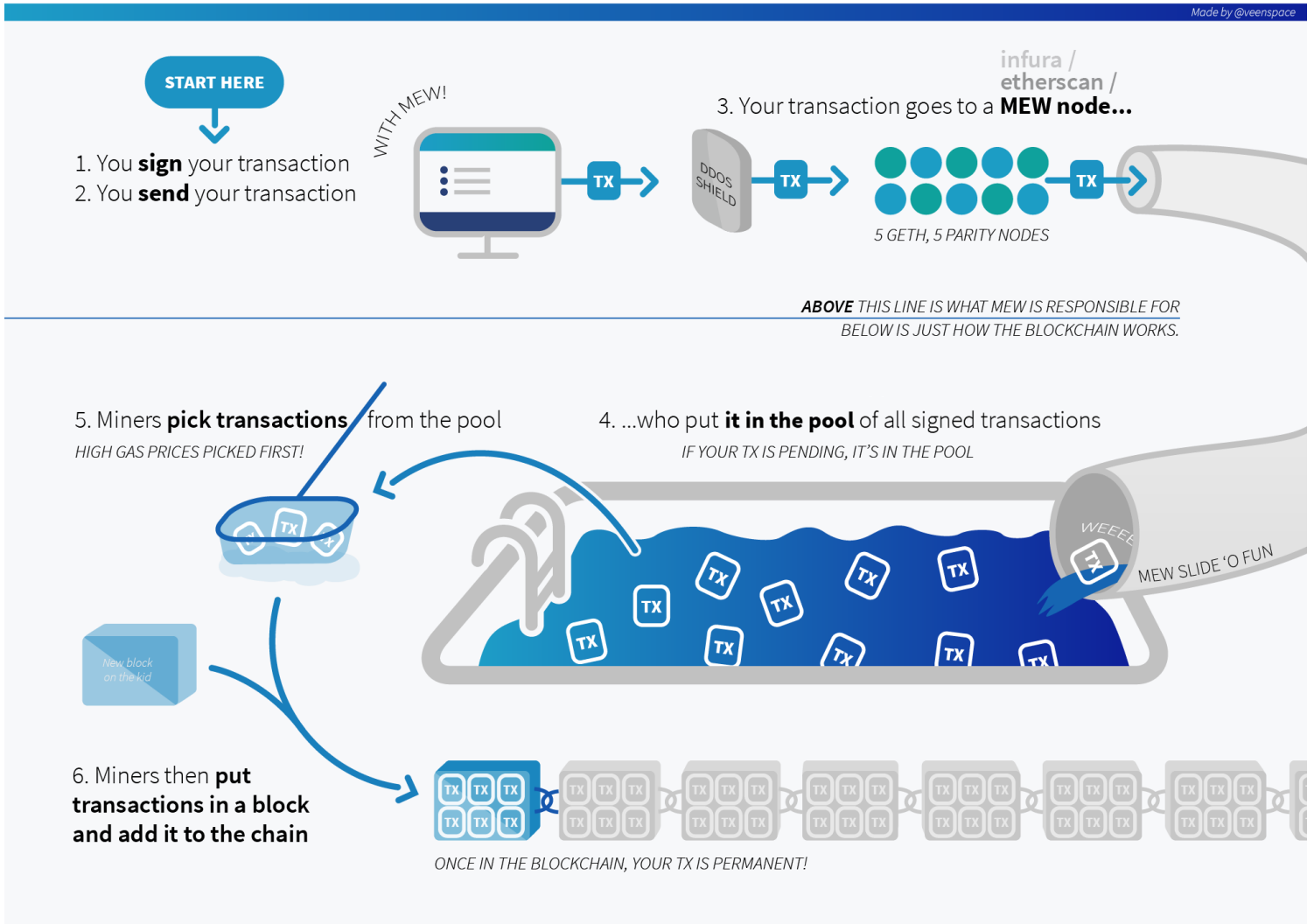
Coût des instructions

Value	Mnemonic	Gas Used	Subset	Removed from stack	Added to stack	Notes
0x00	STOP	0	zero	0	0	Halts execution.
0x01	ADD	3	verylow	2	1	Addition operation
0x02	MUL	5	low	2	1	Multiplication operation.
0x03	SUB	3	verylow	2	1	Subtraction operation.
0x04	DIV	5	low	2	1	Integer division operation.
0x51	MLOAD	3	verylow	1	1	Load word from memory.
0x52	MSTORE	3	verylow	2	0	Save word to memory
0x53	MSTORE8	3	verylow	2	0	Save byte to memory.
0x54	SLOAD	200		1	1	Load word from storage
0x55	SSTORE	((value != 0) && (storage_location == 0)) ? 20000 : 5000		1	1	Save word to storage.

Coût d’une transaction

- Coût total d'une transaction = *GAS_price x GAS_used*
- Priorité aux transactions avec un *GAS_price* élevé.
- Plus l'utilisateur est prêt à payer, plus vite la transaction sera traitée.

MyEtherWallet Behind-The-Scenes



https://ethgasstation.info/

Recommended Gas Prices in Gwei

133	TRADER < ASAP
133	FAST < 2m
111	STANDARD < 5m

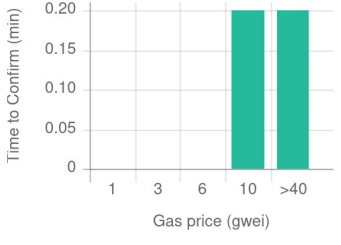
Gas Internals

Median Wait Times

seconds	blocks
14	1

Transaction Count by Gas Price

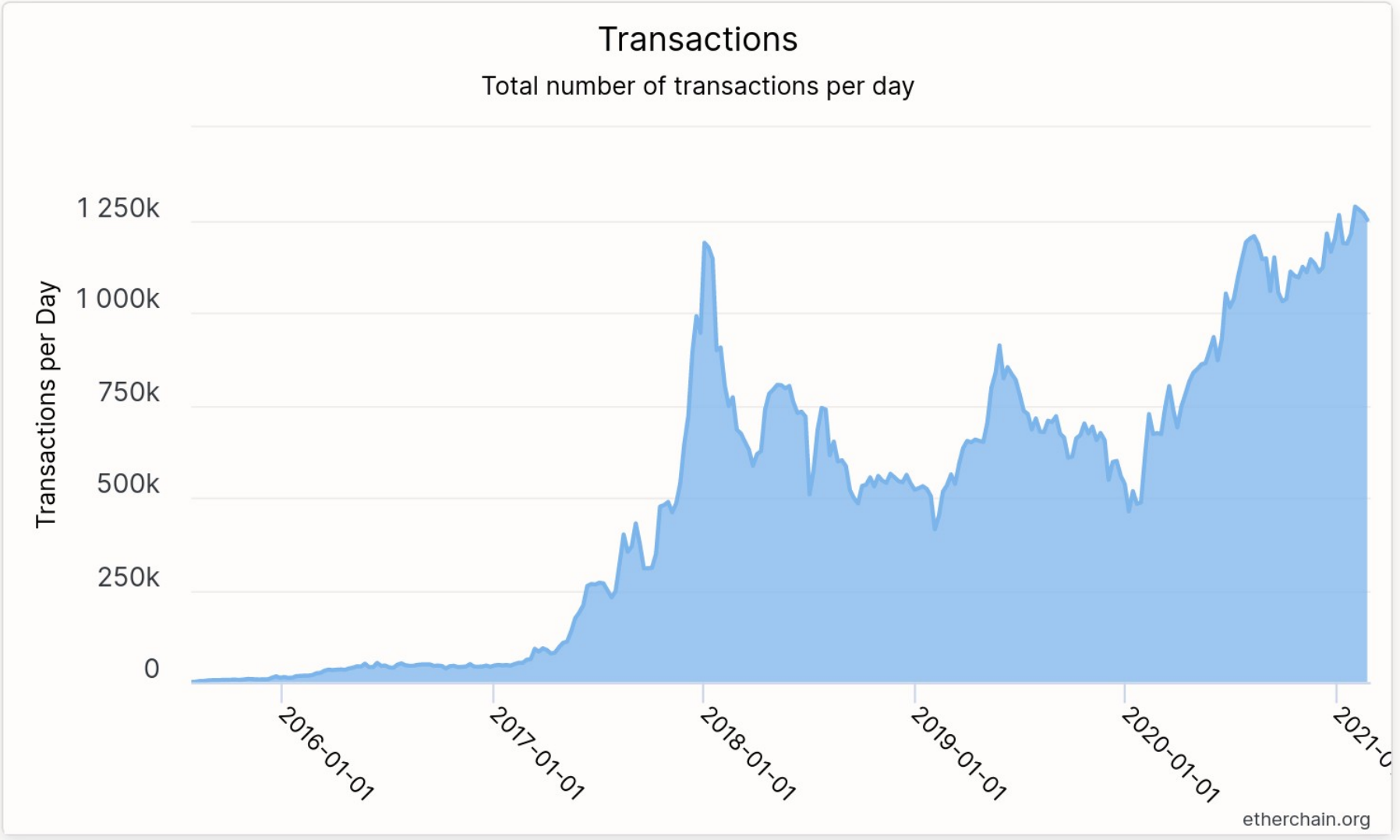
Confirmation Time by Gas Price

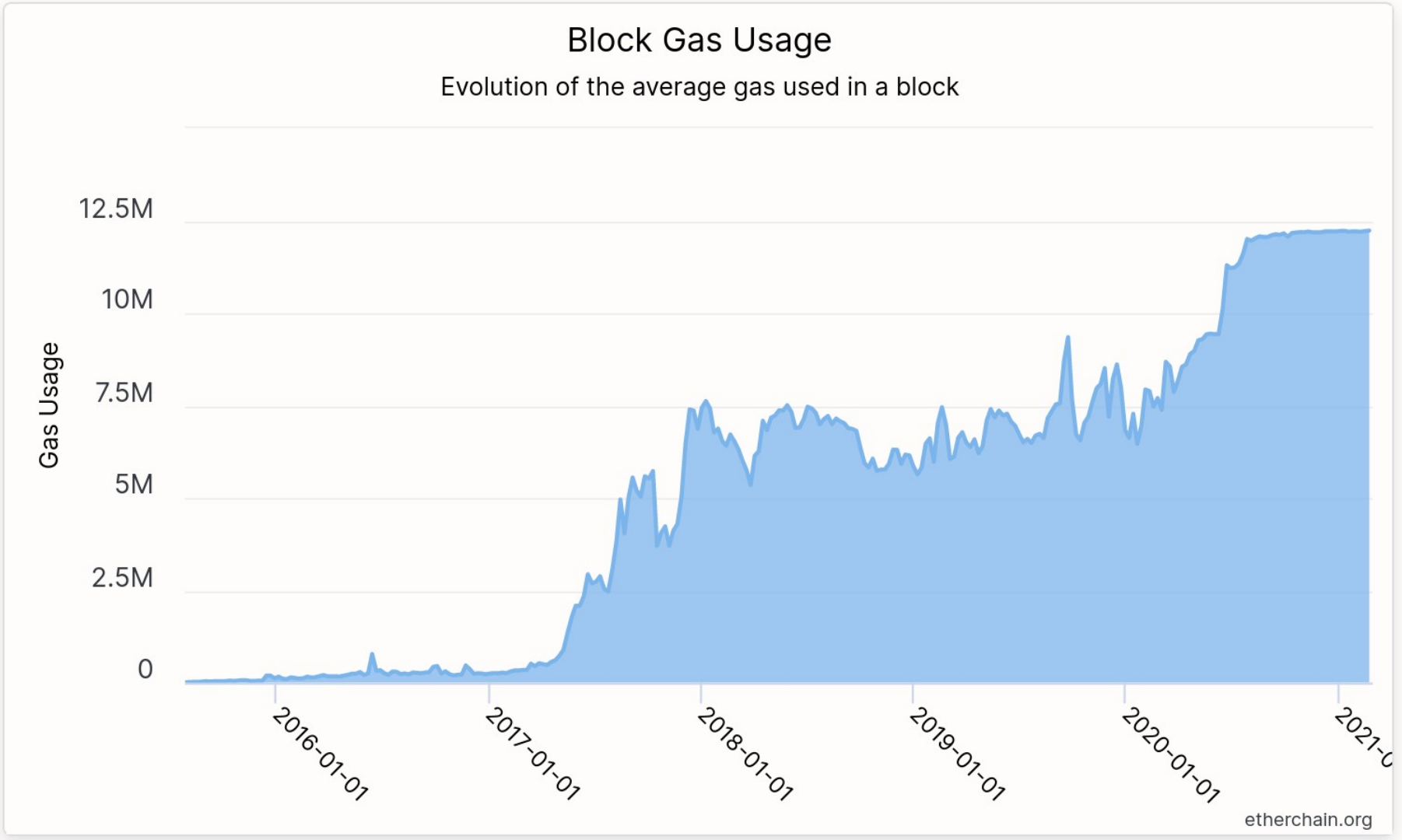


Real Time Gas Use

Last Block: 11951737

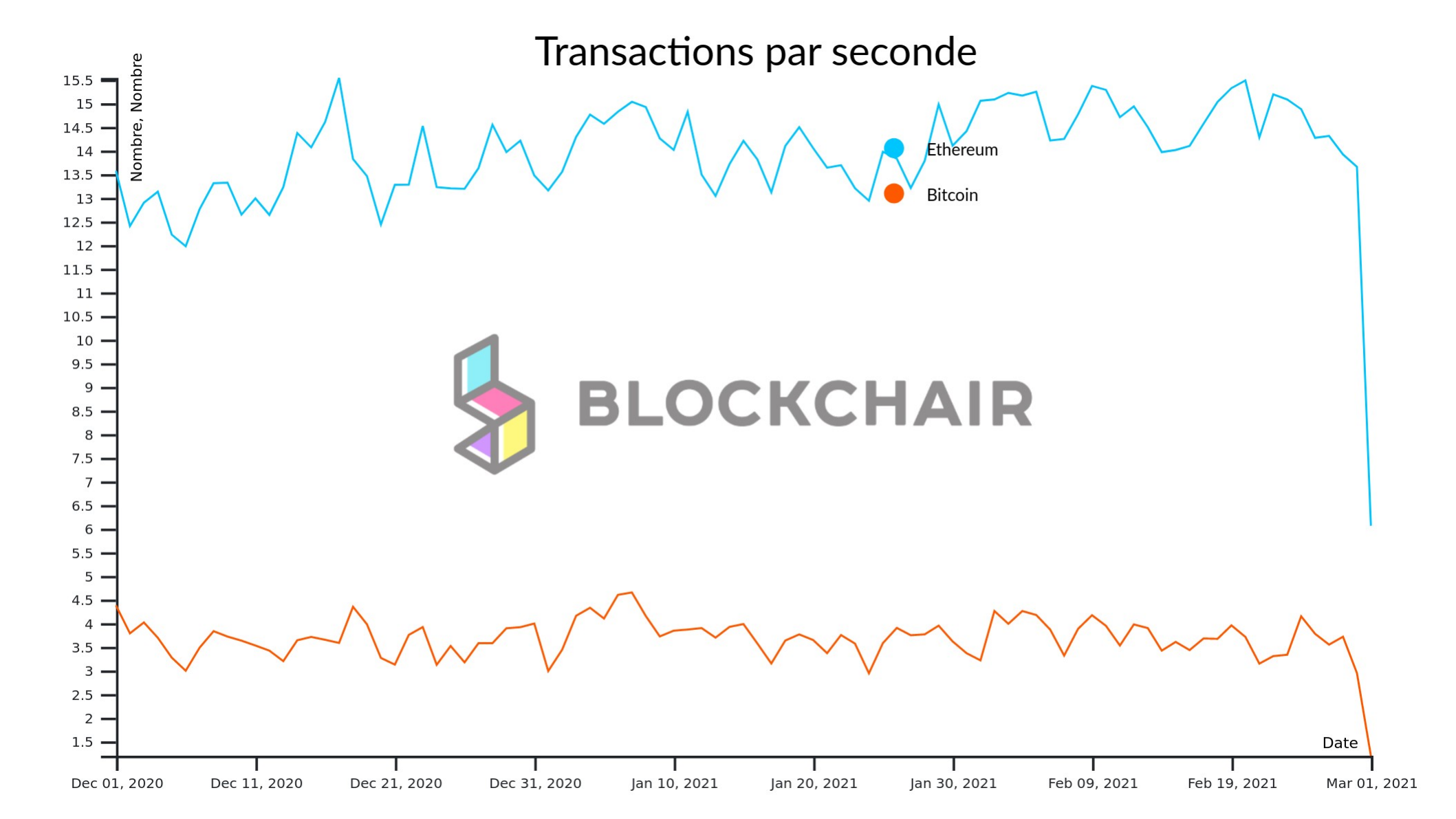
Nombre de GAS par bloque : 12,500,000
soit 3.5 ETH par bloque (février 2021)


Transactions
Total number of transactions per day
Transactions per Day
1 250k
1 000k
750k
500k
250k
0
2016-01-01
2017-01-01
2018-01-01
2019-01-01
2020-01-01
etherchain.org


Block Gas Usage
Evolution of the average gas used in a block
Gas Usage
12.5M
10M
7.5M
5M
2.5M
0
2016-01-01
2017-01-01
2018-01-01
2019-01-01
2020-01-01
etherchain.org

Performances actuelles

- **block time 15 sec, 4 blocks/min**
- 5800 block/day, 2 000 000 block/year
- **Block Gas limit 12 500 000**
- Daily Gas cap 72 500 000 000
- **15 tx/sec**


Transactions par seconde
Nombre, Nombre
15.5
15
14.5
14
13.5
13
12.5
12
11.5
11
10.5
10
9.5
9
8.5
8
7.5
7
6.5
6
5.5
5
4.5
4
3.5
3
2.5
2
1.5
Ethereum
Bitcoin
BLOCKCHAIR
Date
Dec 01, 2020
Dec 11, 2020
Dec 21, 2020
Dec 31, 2020
Jan 10, 2021
Jan 20, 2021
Jan 30, 2021
Feb 09, 2021
Feb 19, 2021
Mar 01, 2021

En résumé : Pour le calcul...

- L'infrastructure Ethereum est un énorme ordinateur Turing complet distribué
- Ultra tolérant aux pannes
- Très mal exploité car tous les nœuds exécutent les mêmes instructions avec les même données :-(

En résumé : Pour le stockage...

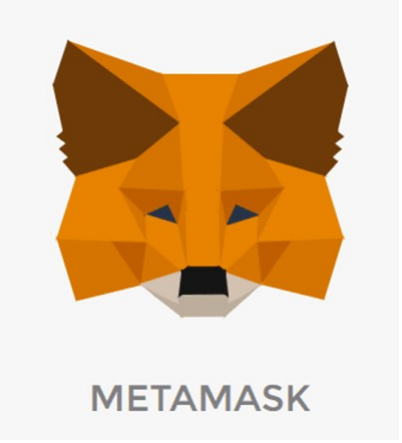
On est limité par conception à :

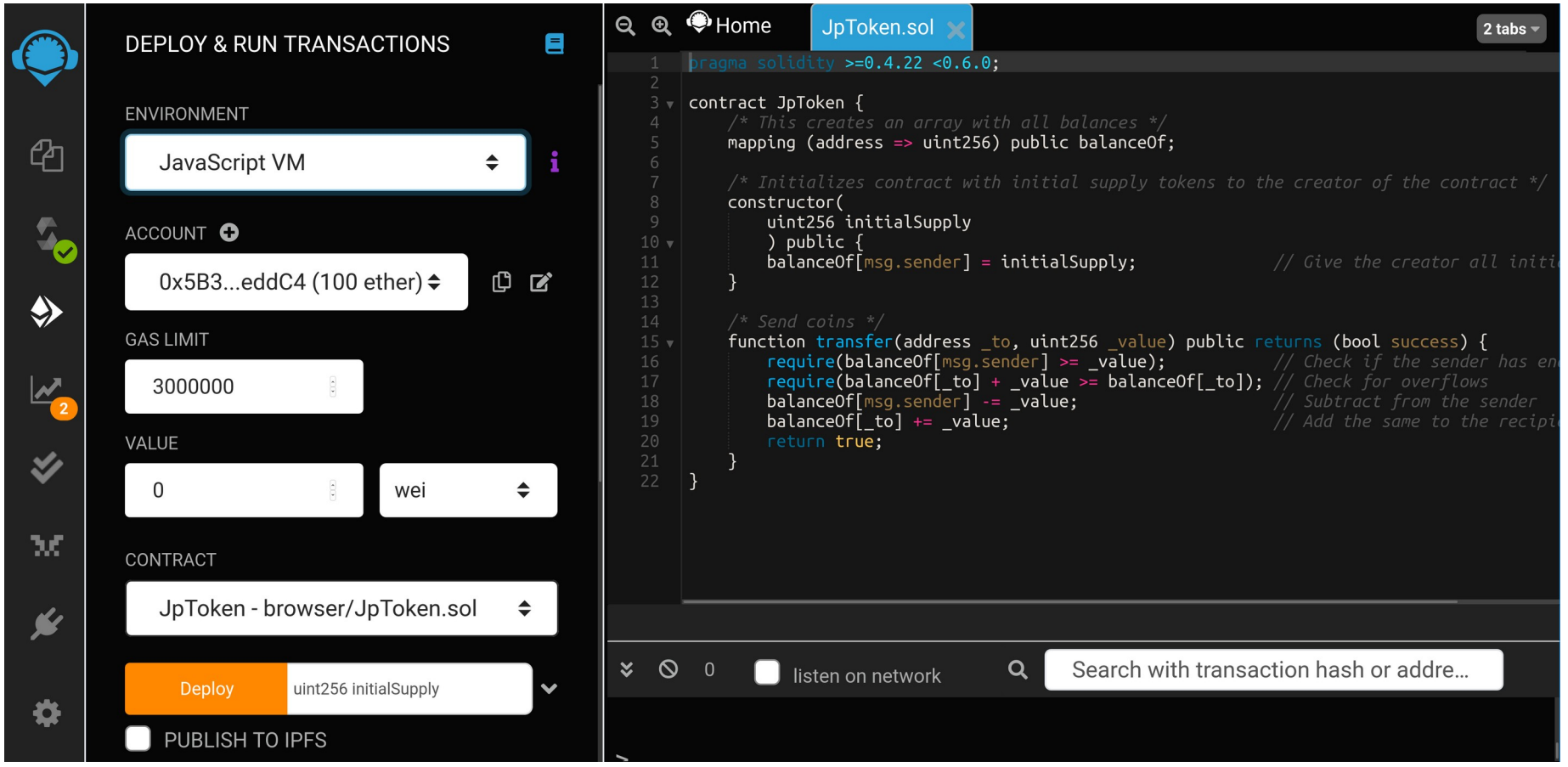
- la capacité de stockage des nœuds
- le débit (140 kB / 15 sec => 75 kbits/s)
- le prix du GAS qui fluctue en fonction de l'Ether
- SSTORE : 20.000 GAS/Word -> 640.000 GAS/KB
- 100 GWei / GAS -> 0.64 ETH/KB
- 1500 $/ETH -> 96 $/KB

96,000,000 USD / GB
Temps de transfert : ~30 heures
(février 2021 - ~1500 USD/ETH)

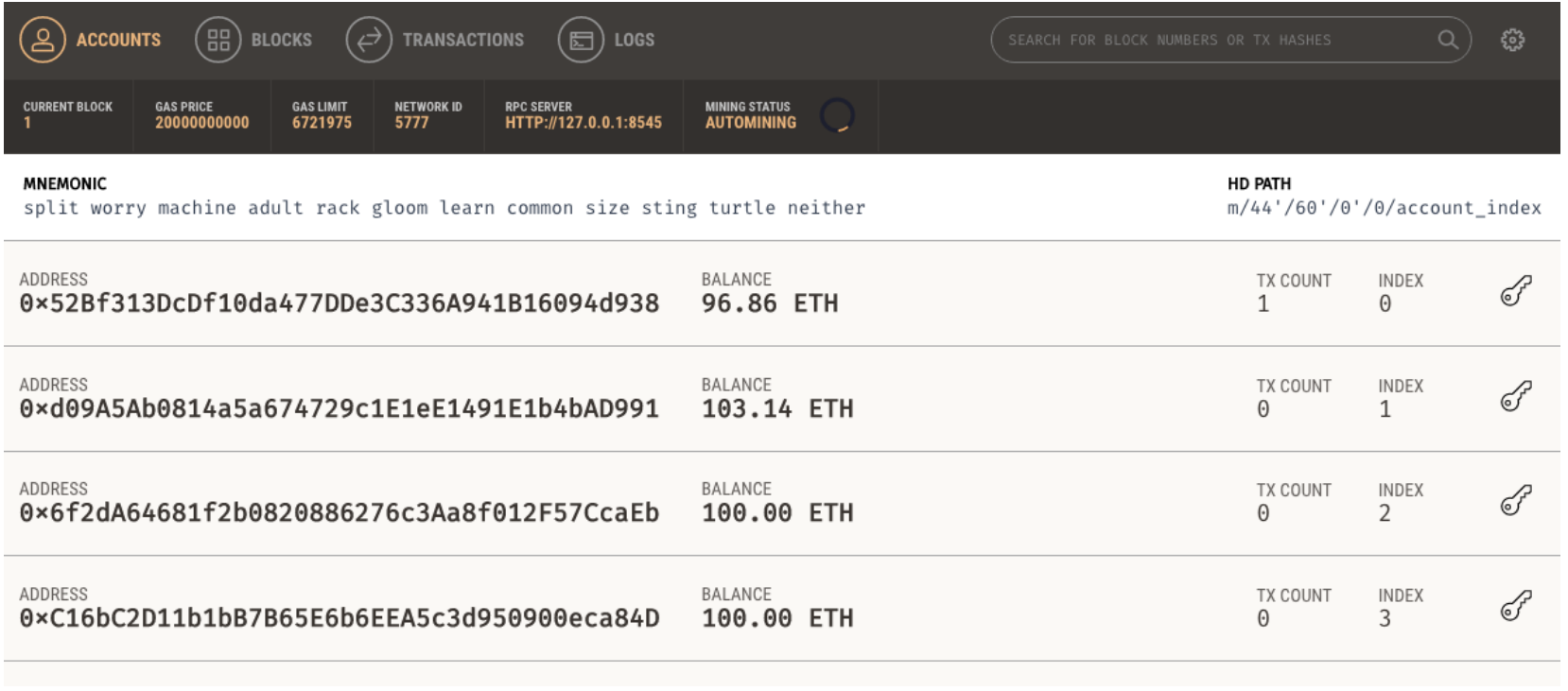
Outils de développement

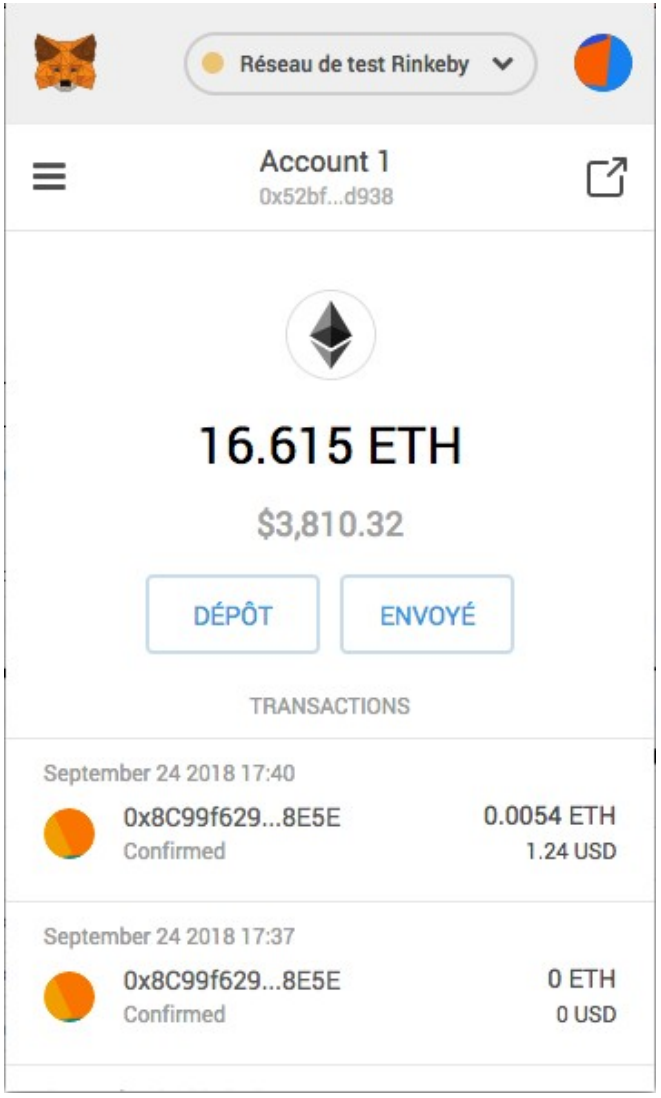
- Metamask
- Remix
- Ganache
- Truffle

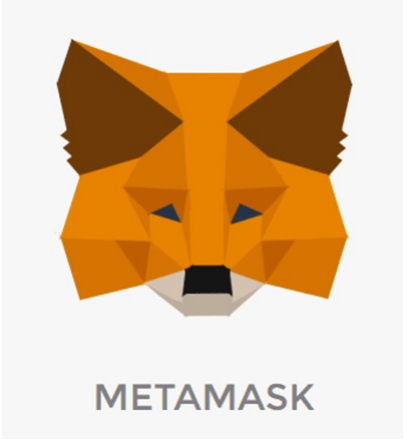


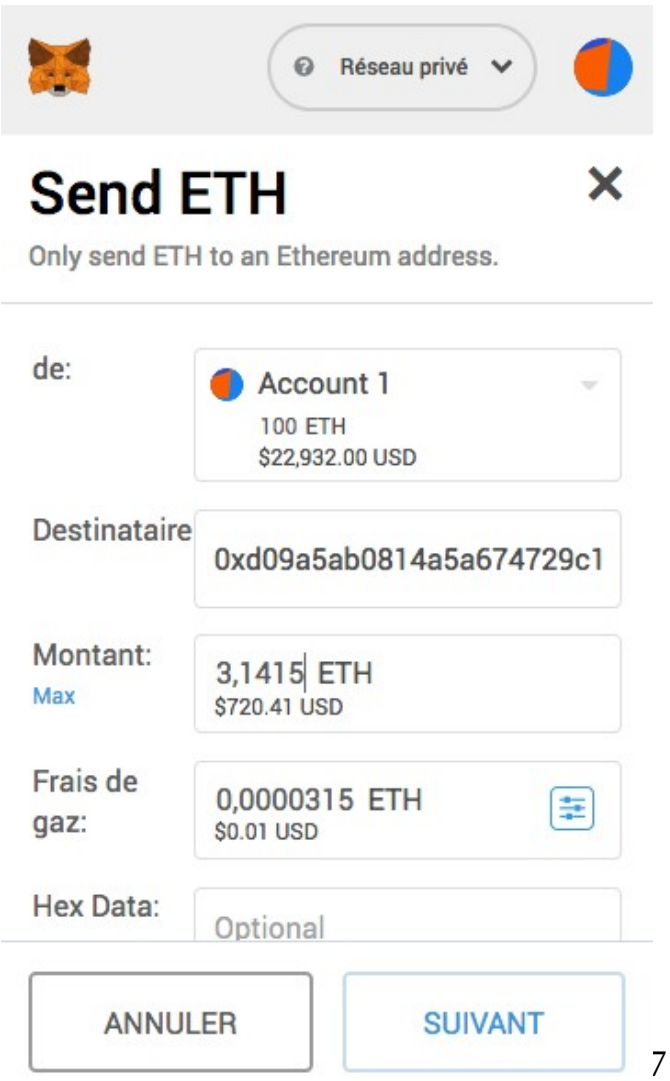

DEPLOY & RUN TRANSACTIONS
ENVIRONMENT
JavaScript VM
ACCOUNT
0x5B3...eddC4 (100 ether)
GAS LIMIT
3000000
VALUE
0
wei
CONTRACT
JpToken - browser/JpToken.sol
Deploy
uint256 initialSupply
PUBLISH TO IPFS
Home
JpToken.sol
2 tabs
pragma solidity >=0.4.22 <0.6.0;
contract JpToken {
/* This creates an array with all balances */
mapping (address => uint256) public balanceOf;
/* Initializes contract with initial supply tokens to the creator of the contract */
constructor(
uint256 initialSupply
) public {
balanceOf[msg.sender] = initialSupply; // Give the creator all initi
}
/* Send coins */
function transfer(address _to, uint256 _value) public returns (bool success) {
require(balanceOf[msg.sender] >= _value); // Check if the sender has en
require(balanceOf[_to] + _value >= balanceOf[_to]); // Check for overflows
balanceOf[msg.sender] -= _value; // Subtract from the sender
balanceOf[_to] += _value; // Add the same to the recipi
return true;
}
}
0
listen on network
Search with transaction hash or addre...

Ganache

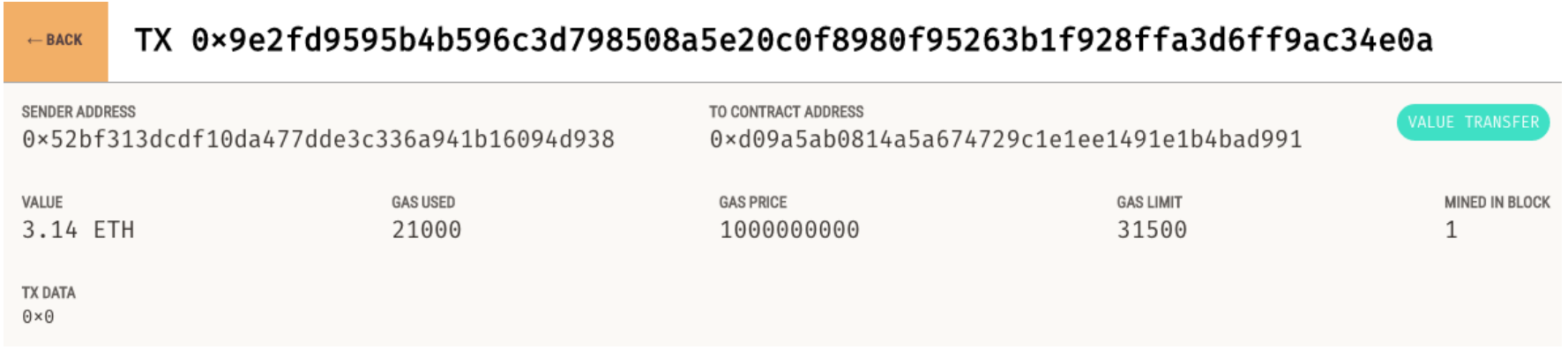

ACCOUNTS
BLOCKS
TRANSACTIONS
LOGS
SEARCH FOR BLOCK NUMBERS OR TX HASHES
CURRENT BLOCK 1
GAS PRICE 20000000000
GAS LIMIT 6721975
NETWORK ID 5777
RPC SERVER HTTP://127.0.0.1:8545
MINING STATUS AUTOMINING
MNEMONIC
split worry machine adult rack gloom learn common size sting turtle neither
HD PATH
m/44'/60'/0'/0/account_index
ADDRESS 0×52Bf313DcDf10da477DDe3C336A941B16094d938
BALANCE 96.86 ETH
TX COUNT 1
INDEX 0
ADDRESS 0×d09A5Ab0814a5a674729c1E1eE1491E1b4bAD991
BALANCE 103.14 ETH
TX COUNT 0
INDEX 1
ADDRESS 0×6f2dA64681f2b0820886276c3Aa8f012F57CcaEb
BALANCE 100.00 ETH
TX COUNT 0
INDEX 2
ADDRESS 0×C16bC2D11b1bB7B65E6b6EEA5c3d950900eca84D
BALANCE 100.00 ETH
TX COUNT 0
INDEX 3


Réseau de test Rinkeby
Account 1
0x52bf...d938
16.615 ETH
$3,810.32
DÉPÔT
ENVOYÉ
TRANSACTIONS
September 24 2018 17:40
0x8C99f629...8E5E
Confirmed
0.0054 ETH
1.24 USD
September 24 2018 17:37
0x8C99f629...8E5E
Confirmed
0 ETH
0 USD


METAMASK


Réseau privé
Send ETH
Only send ETH to an Ethereum address.
de:
Account 1
100 ETH
$22,932.00 USD
Destinataire
0xd09a5ab0814a5a674729c1
Montant:
Max
3,1415 ETH
$720.41 USD
Frais de gaz:
0,0000315 ETH
$0.01 USD
Hex Data:
Optional
ANNULER
SUIVANT

Visualisation d’une transaction



Transaction 0x512322644a8bb57fe3fe00977ba462daef2da571a9ee267fc74a969973eae7df

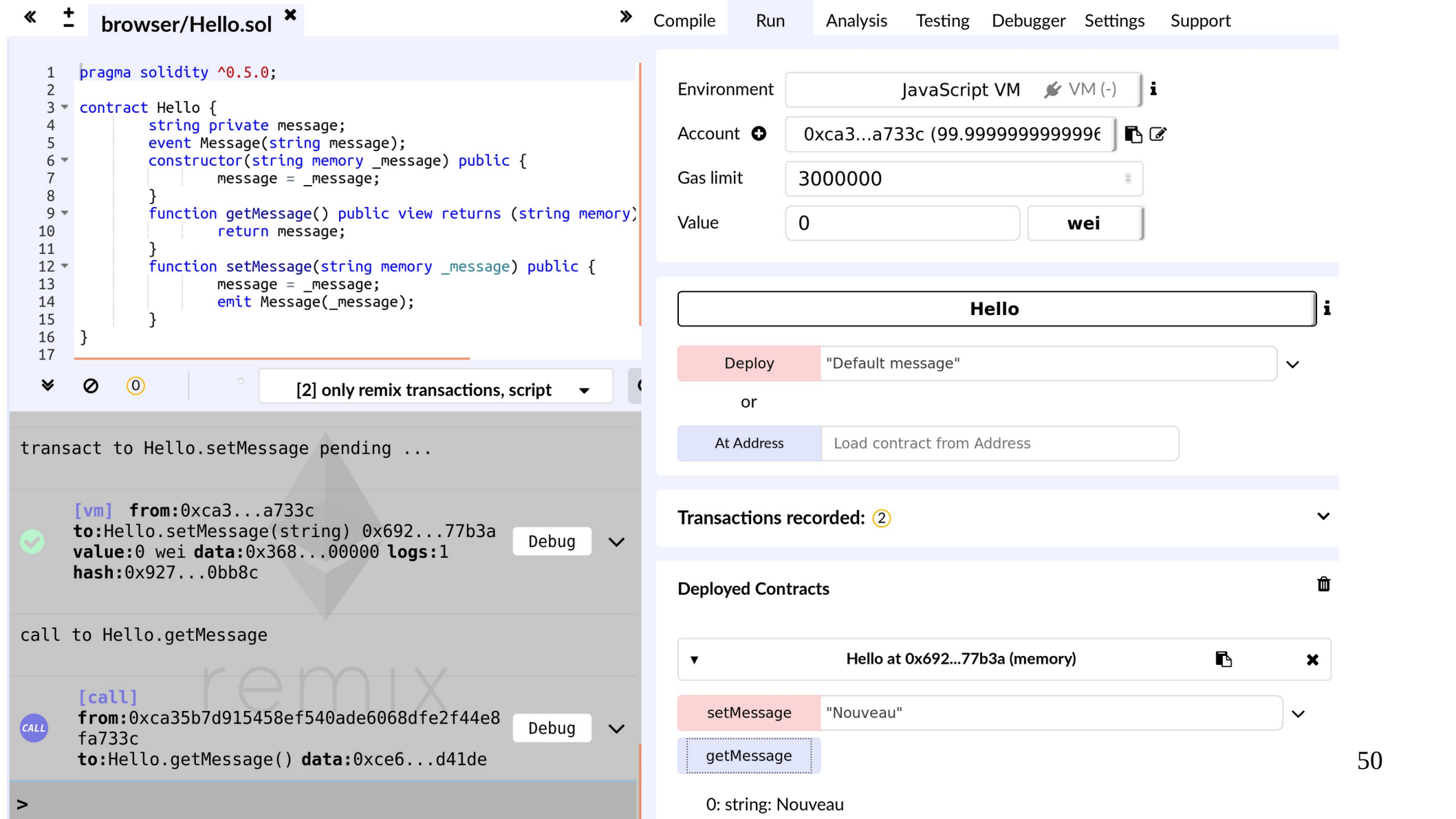
Overview | Comments

Buy | Crypto Loan

Transaction Information

Tools & Utilities

TxHash:	0x512322644a8bb57fe3fe00977ba462daef2da571a9ee267fc74a969973eae7df
TxReceipt Status:	Success
Block Height:	6686011 (1 Block Confirmation)
TimeStamp:	17 secs ago (Nov-11-2018 06:00:36 PM +UTC)
From:	0xc98f4c64c63ce7d10cb7615e0100ffcfc912aba3
To:	0x5d2c3da03b5bc33673b921e8cf5bbae2100e7dc0
Value:	0.054343840721857432 Ether ($11.37)
Gas Limit:	21000
Gas Used By Transaction:	21000 (100%)
Gas Price:	0.000000001401000001 Ether (1.401000001 Gwei)
Actual Tx Cost/Fee:	0.00002942100002 Ether ($0.006158)
Nonce & {Position}:	1 | {123}


browser/Hello.sol
Compile
Run
Analysis
Testing
Debugger
Settings
Support
pragma solidity ^0.5.0;
contract Hello {
string private message;
event Message(string message);
constructor(string memory _message) public {
message = _message;
}
function getMessage() public view returns (string memory)
return message;
}
function setMessage(string memory _message) public {
message = _message;
emit Message(_message);
}
}
[2] only remix transactions, script
transact to Hello.setMessage pending ...
[vm] from:0xca3...a733c
to:Hello.setMessage(string) 0x692...77b3a
value:0 wei data:0x368...00000 logs:1
hash:0x927...0bb8c
Debug
call to Hello.getMessage
[call]
from:0xca35b7d915458ef540ade6068dfe2f44e8fa733c
to:Hello.getMessage() data:0xce6...d41de
Debug
Environment
JavaScript VM
VM (-)
Account
0xca3...a733c (99.999999999996
Gas limit
3000000
Value
0
wei
Hello
Deploy
"Default message"
or
At Address
Load contract from Address
Transactions recorded: 2
Deployed Contracts
Hello at 0x692...77b3a (memory)
setMessage
"Nouveau"
getMessage
0: string: Nouveau

Conclusion

- Par conception la blockchain Ethereum garantie :
  - l'immutabilité des données,
  - l'exécution et l'accès sans censure possible.
- Analyser un Smart Contract en terme de consommation de GAS pour maîtriser le coût opérationnel
- Maximiser les calculs et le stockage *offchain*.

Liens utiles

- https://www.openlearning/courses/bliss-mooc
- https://hackernoon.com/ether-purchase-power-df40a38c5a2f
- OPCODE list + GAS : https://docs.google.com/spreadsheets/d/1m89CVujrQe5LAFJ8-YAUCcNK950dUzMQPMJBxRtGCqs/edit#gid=0
- https://www.ethernodes.org/
- https://www.etherchain.org/
- https://etherscan.io/
- https://medium.com/coinmonks/storing-on-ethereum-analyzing-the-costs-922d41d6b316