Scientific School on Blockchain
& distributed ledger technology

Web3

A platform for decentralized apps

14 Jun 2018 **Mauro Pili**

Andreas M. Antonopoulos

@aantonop

Bitcoin UX design is a mess

09:52 - 14 nov 2013

Web and Blockchain: **friends** or **foes**?

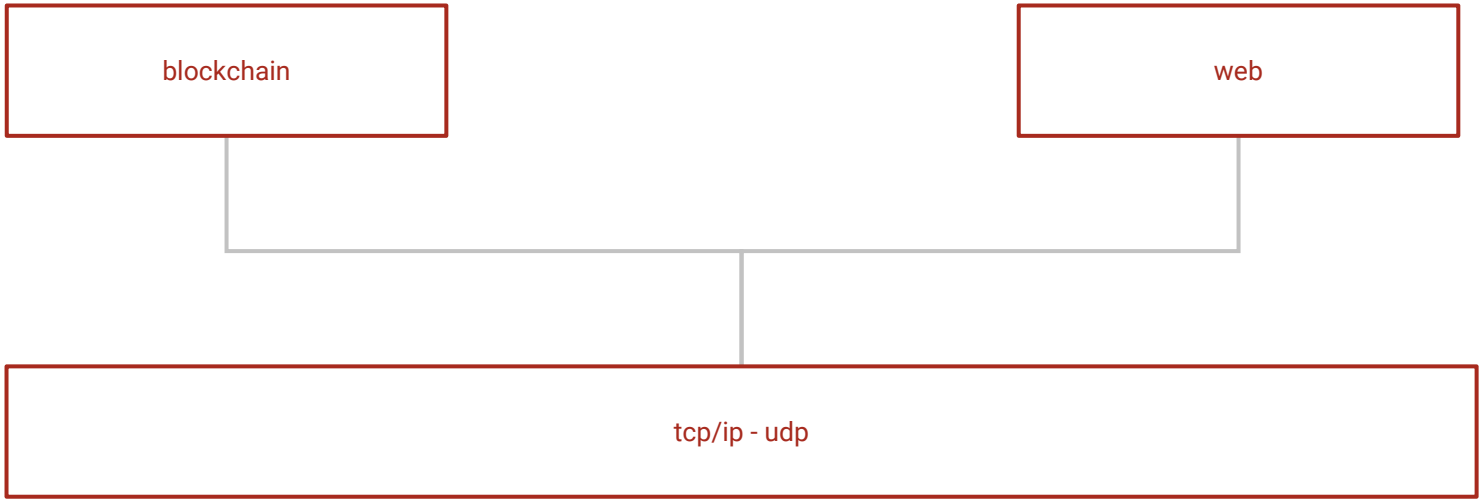
World Wide Web

World Wide Web has been central to the development of the Information Age and is the **primary tool billions of people use to interact** on the Internet

Blockchain

An open, **distributed** ledger that can record transactions between two parties efficiently and in a verifiable and permanent way

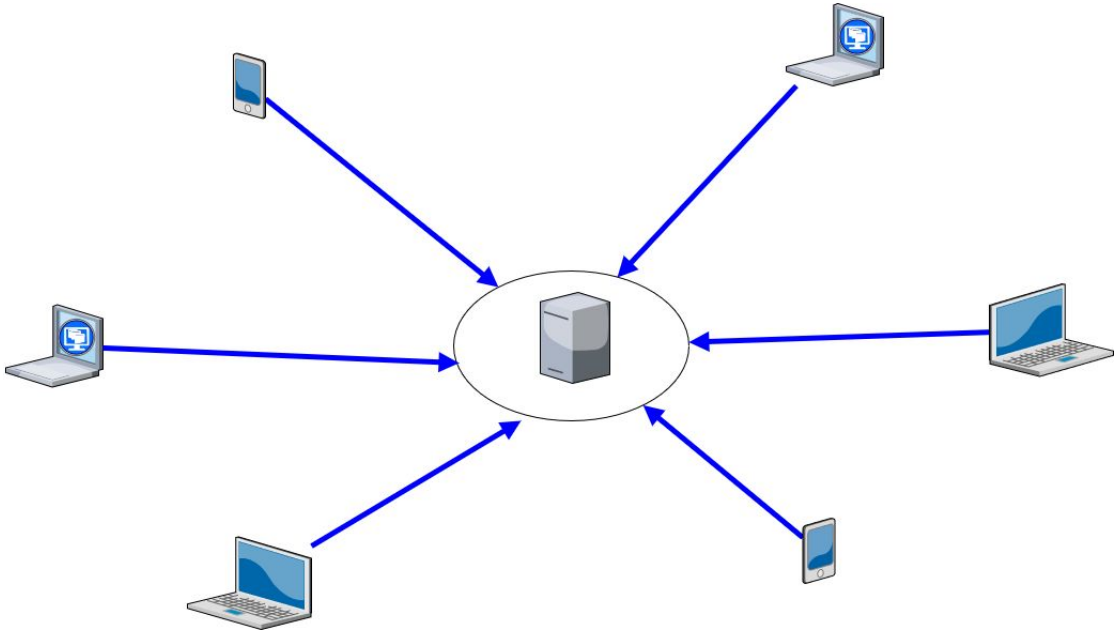
web and blockchain both *live* on the internet



World Wide Web

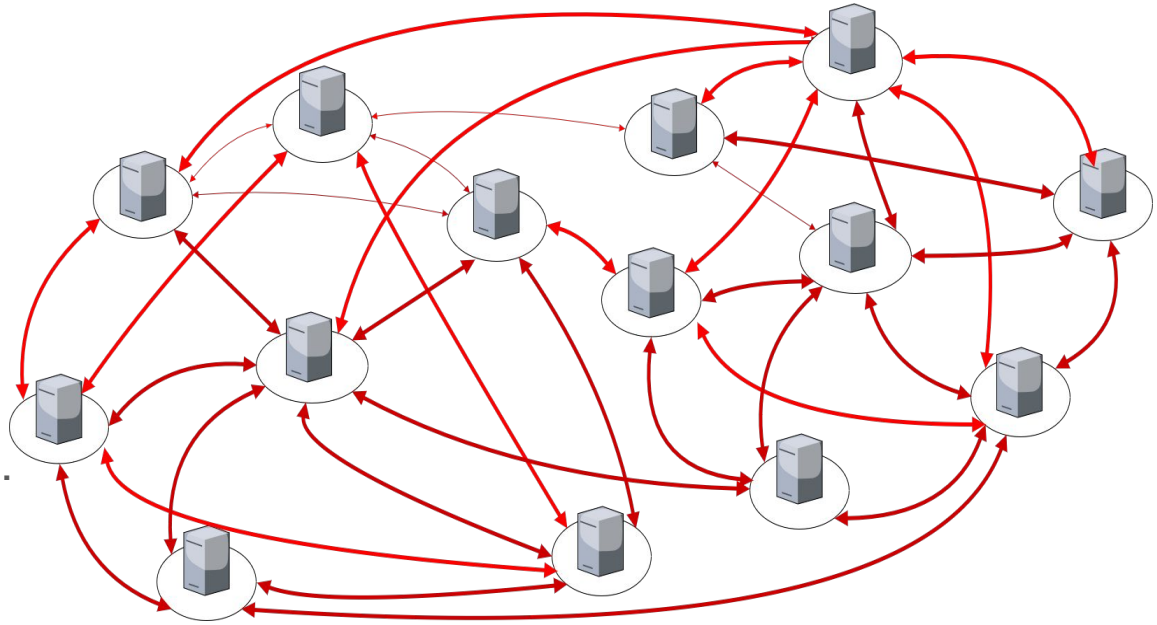
"... a web of hypertext documents to be viewed by browsers using a **client–server architecture**"

(Nov.1990 Tim Berners-Lee)



Peer to peer network

Peer-to-peer (P2P) computing or networking is a **distributed** application architecture that partitions tasks or workloads between peers. Peers are equally privileged, equipotent participants in the application. They are said to form a peer-to-peer network of nodes

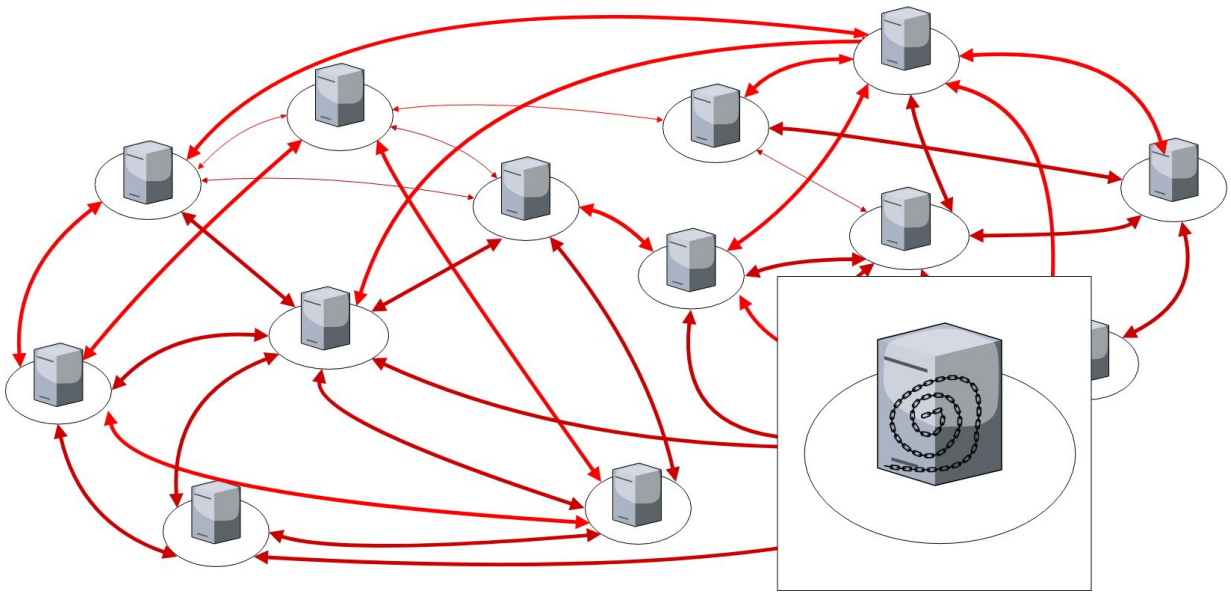


Bitcoin peer to peer network

Any computer that connects to the Bitcoin network is called a node.

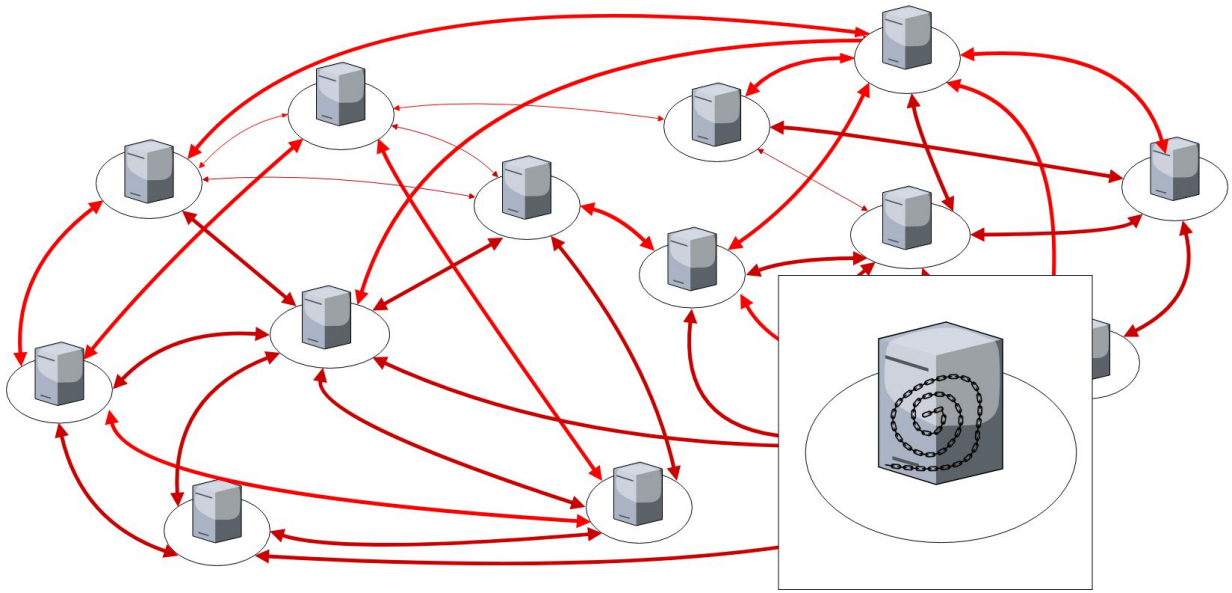
Nodes that fully verify all of the rules of Bitcoin are called full nodes.

Full nodes download every block and transaction and check them against Bitcoin's consensus rules



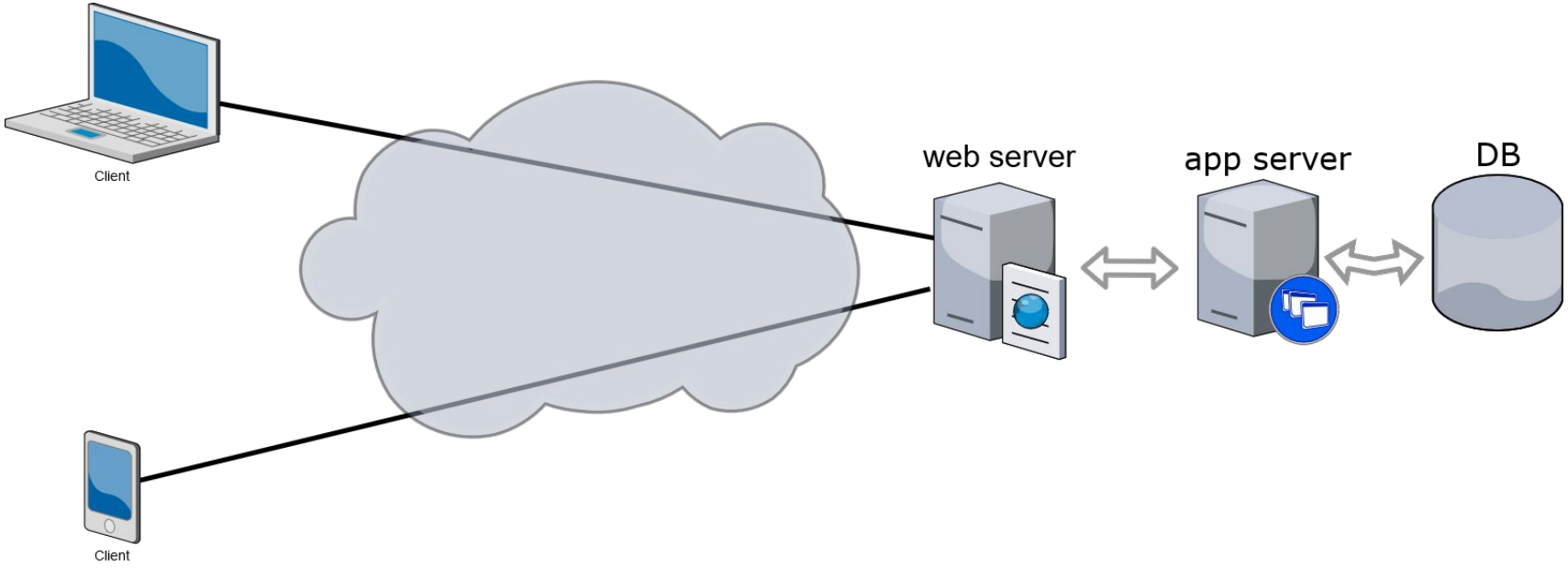
Ethereum a world computer

A globally **decentralized**, un-ownable, digital computer for executing peer-to-peer (smart) contracts

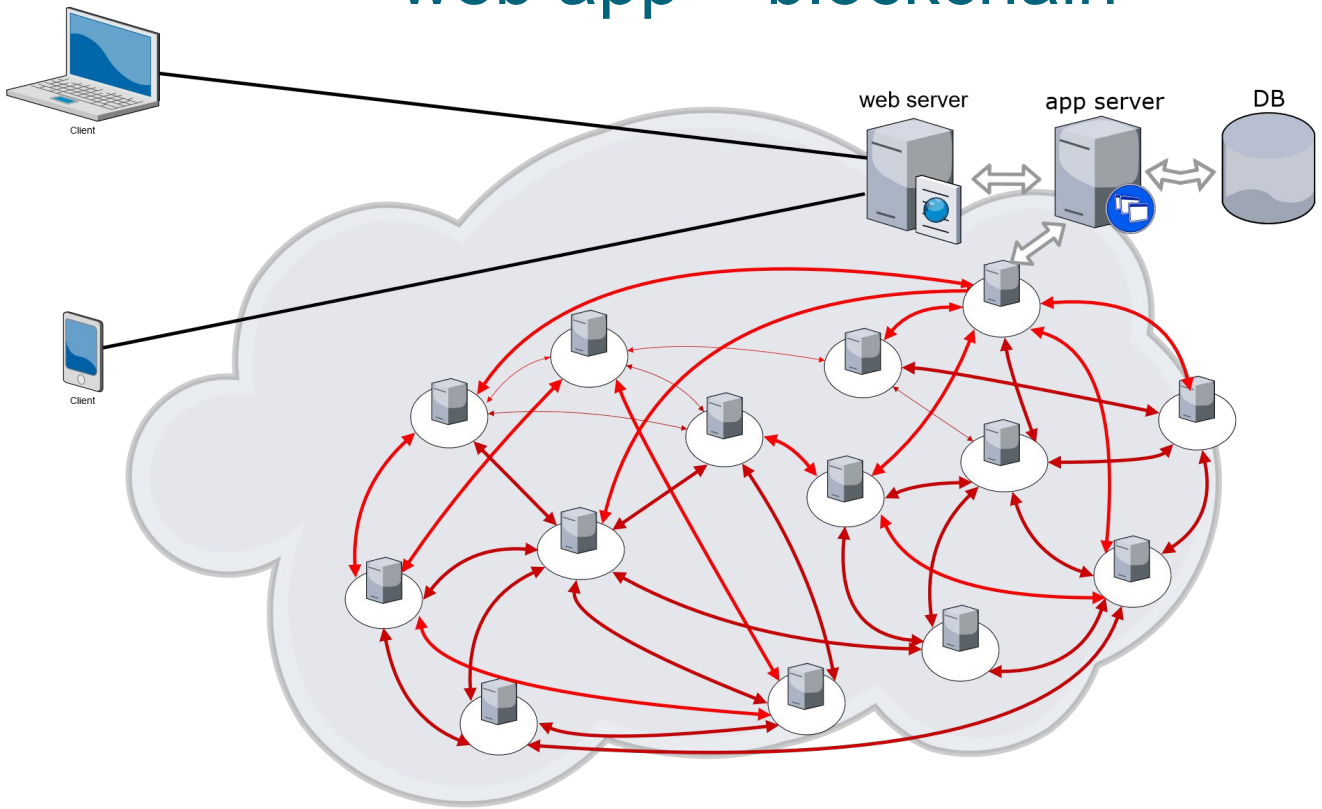


EVM
Ethereum virtual machine

Web app stack



web app + blockchain



Asymmetric cryptography

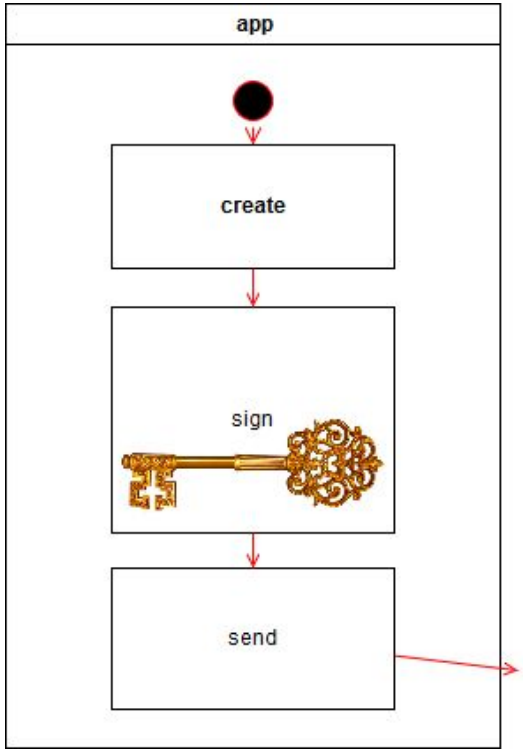
cryptographic system uses pairs of keys:

Public keys which may be disseminated widely

Private keys which are known only to the owner.
Private key should never be shared with anyone

Blockchain app

A blockchain app needs the **private key** to sign the transaction



Blockchain contains public keys

The blockchain is a huge DB of public keys and transactions

User

- create a transaction

- sign the transaction with his private key

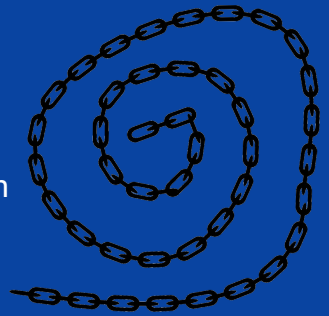
- send the transaction with the public key to the network

Miner

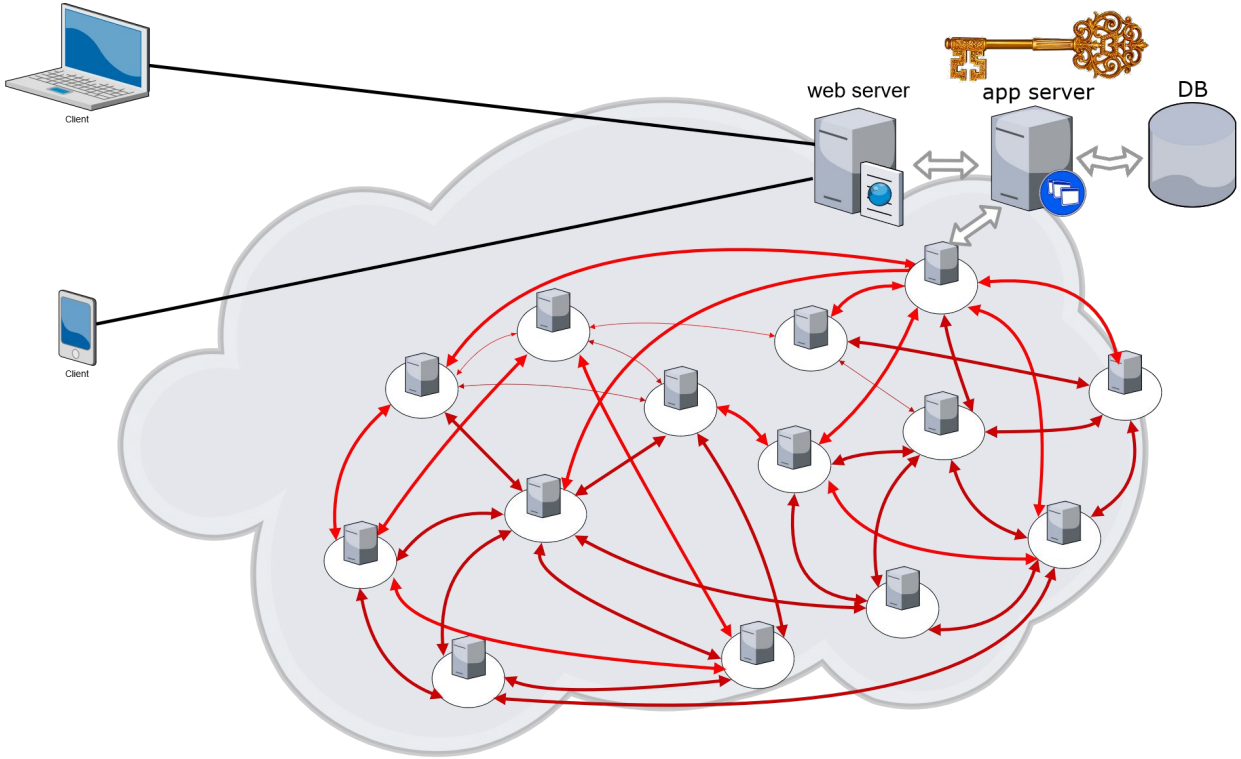
- collect the transactions from the pool
- mine a block with the transactions
- add the new minted block to the blockchain

Nodes

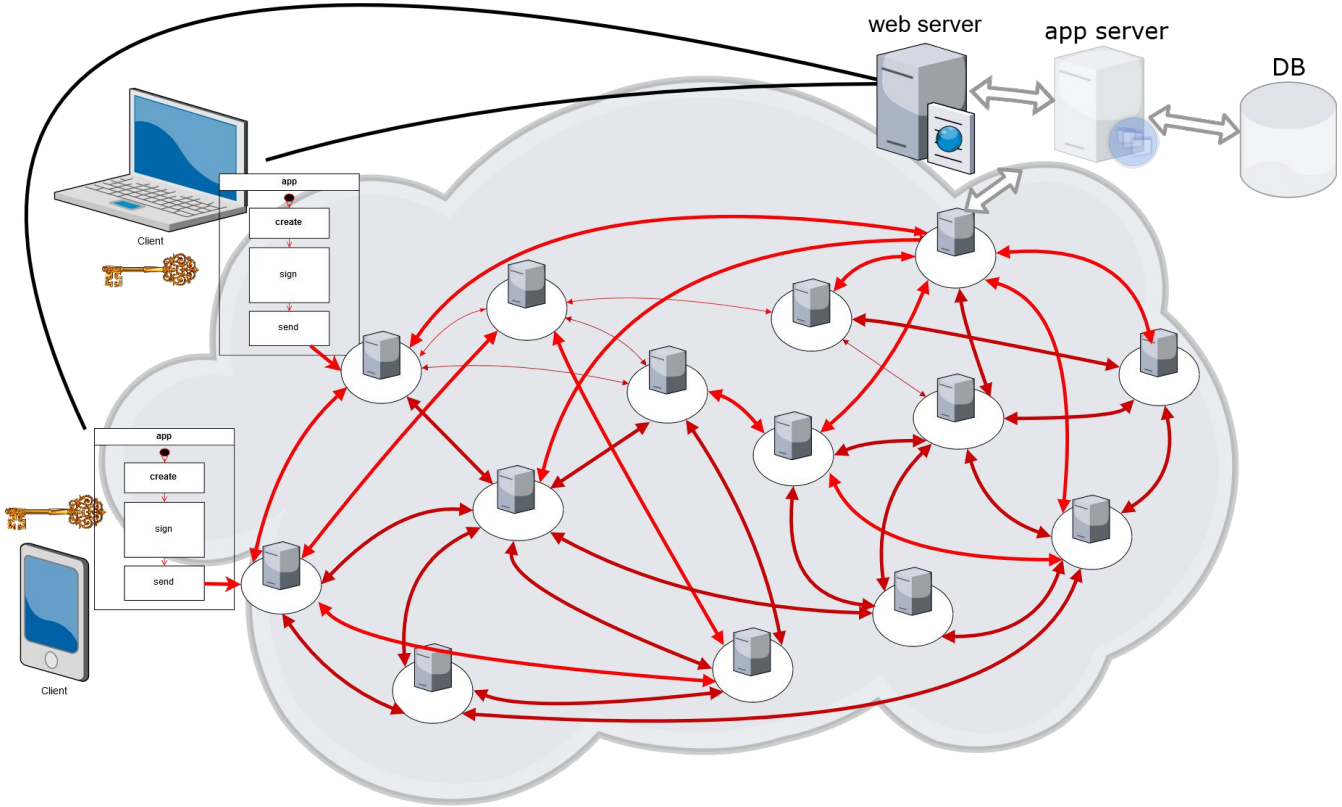
- verify and
- share the blockchain



Centralized (Fiduciary) app provider



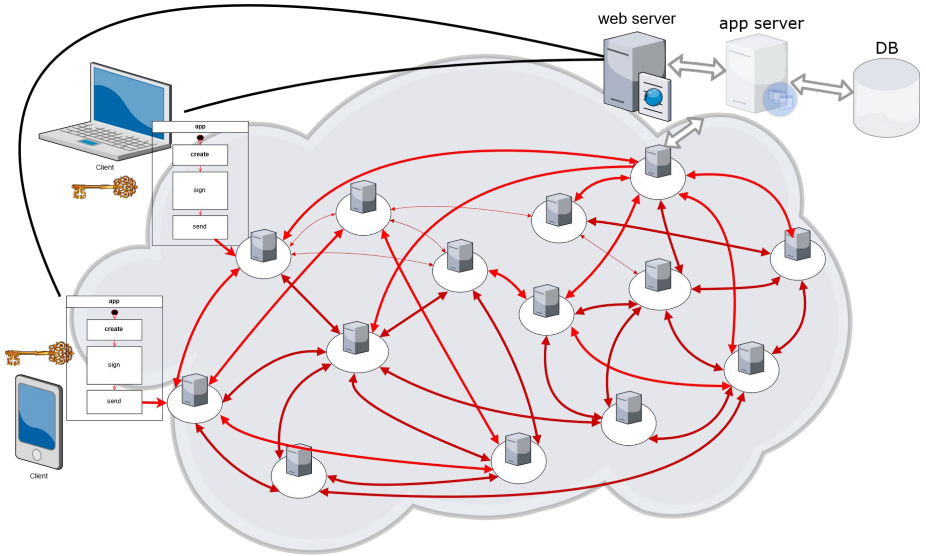
Decentralized app provider



Decentralized app provider

the UI is a web app (html, javascript, css),

- downloaded from a web server
- **runs in the web browser**
- connects to the ethereum network through web3js
- call the smart contract that runs in the ethereum network



the **smart contract** runs in the ethereum network

Web3.js

Web3.js definitions:

- a platform (collection of libraries) which allow you to interact with a local or remote ethereum node, using a HTTP or IPC connection

- “*the web without centralized servers*”

- a JavaScript framework for enabling communication with smart contracts from web apps

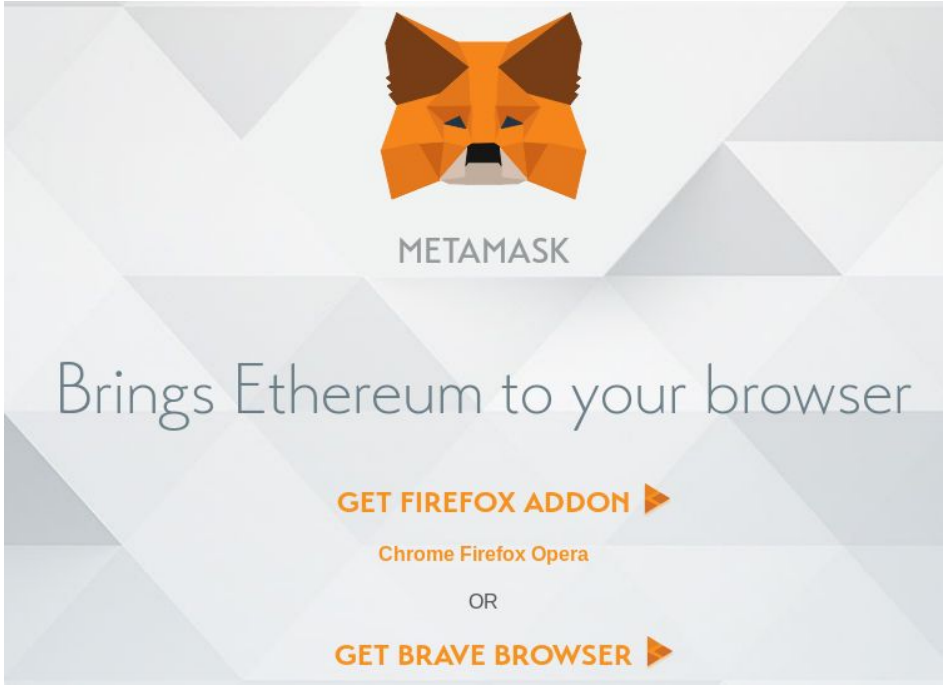
Web3-capable browser

Injection of Web3 object

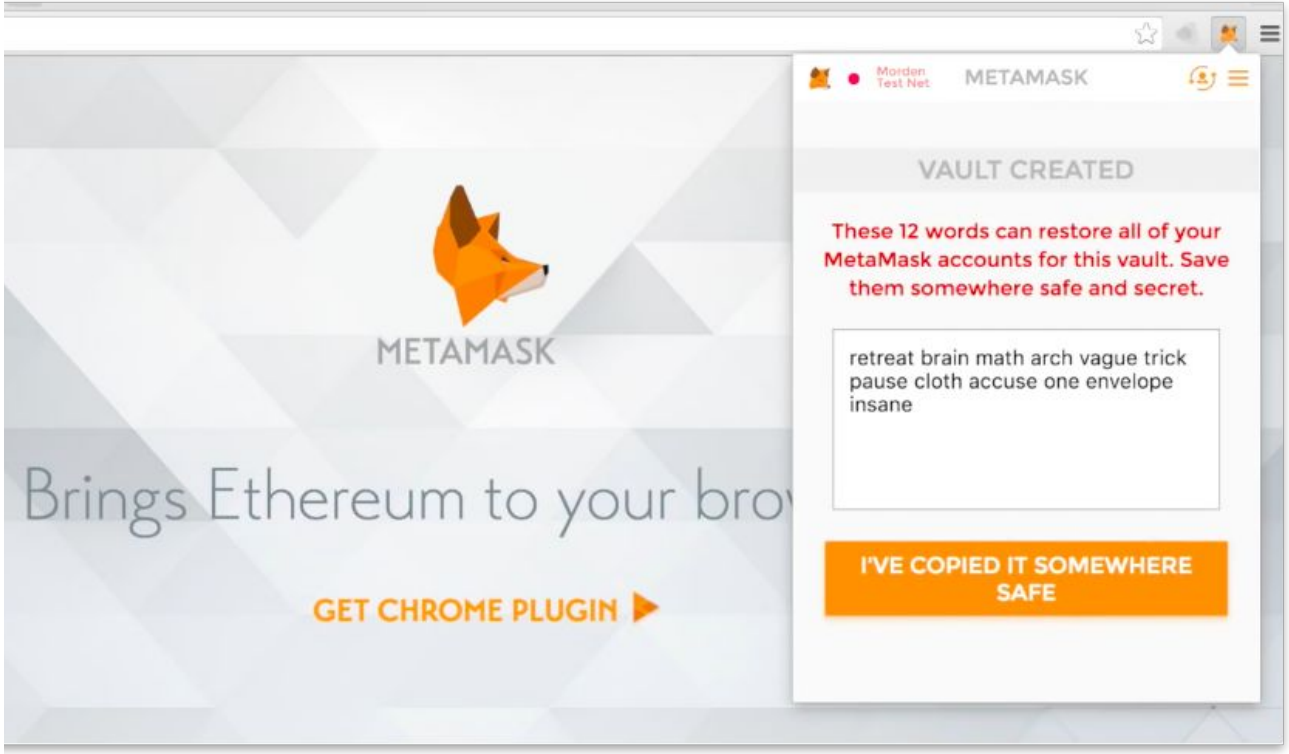
desktop:

- **metamask.io**
- Mist: the official Ethereum browser and the first dapp browser released
- parity.io

mobile: Toshi, Cipher, Trust



metamask plugin in the browser



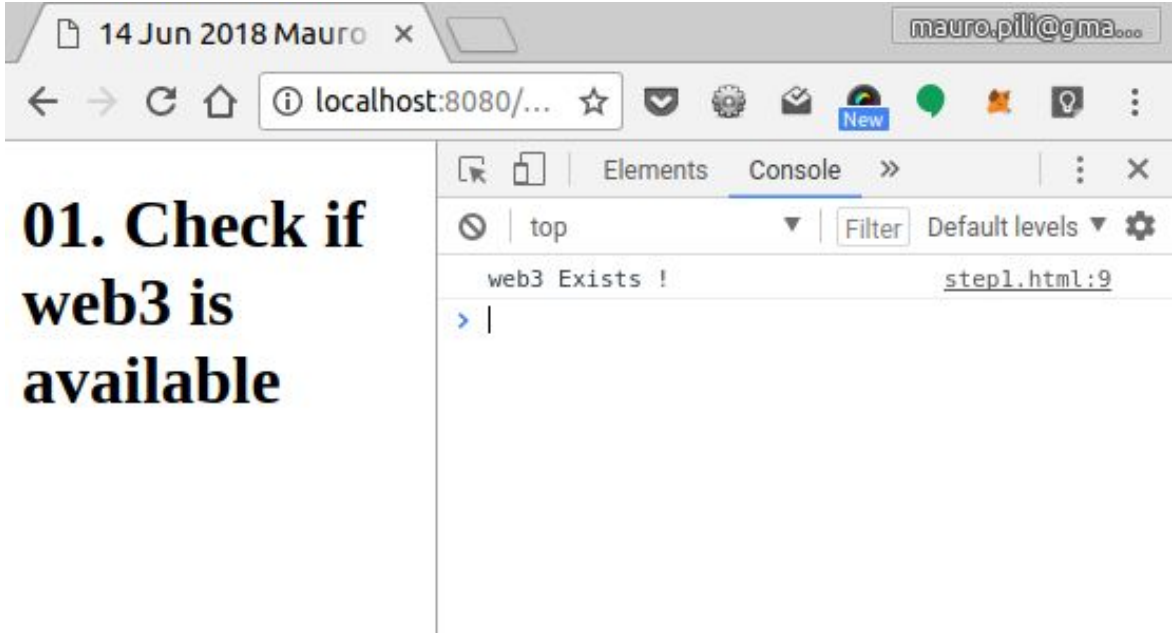
Test if Web3 is available

```
window.addEventListener('load', function() {
  // Checking if Web3 has been injected by the browser (Mist/MetaMask)
  if (typeof web3 !== 'undefined') {
    // Use Mist/MetaMask's provider
    web3js = new Web3(web3.currentProvider);
  } else {
    console.log('No web3? You should consider trying MetaMask!')
   // fallback - use your fallback strategy (local node / hosted node + in-dapp id mgmt / fail)
    web3js = new Web3(new Web3.providers.HttpProvider("http://localhost:8545"));
  }
  // Now you can start your app & access web3 freely:
  startApp()
})
```

https://github.com/MetaMask/faq/blob/master/DEVELOPERS.md

A simple html page to check web3js

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <script type="text/javascript">
     window.addEventListener('load',
function() {
        if (typeof web3 !== 'undefined') {
           console.log('web3 Exists !');
        } else {
           console.log('Sorry, web3 does
not exists');
        }
     })
  </script>
</head>
<body>
<h1>01. Check if web3 is available</h1>
</body>
</html>
```



check network type

```
function getNetworkName(networkId) {
   switch (networkId) {
       case '1': return 'Main net';
       case '2': return 'Morden test (deprecated)';
       case '3': return 'Ropsten test';
       case '4': return 'Rinkeby test';
       case '42': return 'Kovan test';
       default: return 'unknown network';
   }
}
if (typeof web3 !== 'undefined') {
   console.log('Ok web3 exists');
   web3.version.getNetwork(function(err, netId) {
           console.log('err:', err);
           console.log('network id:', netId, getNetworkName(netId));
       }
   )
} else {
   console.log('You are using a browser without Web3 capabilities.');
}
```

14 Jun 2018 Mauro Pili

localhost:8080/step02.html

Check the network type

Main Network

METAMASK

Elements Console Sources Network Performance Memory Application Security

top Filter Default levels Group simil

```
Ok web3 exists
▶MetaMask: web3 will be deprecated in the near future in favor of the ethereumProvider
https://github.com/MetaMask/faq/blob/master/detecting_metamask.md#web3-deprecation
err: null
network id: 1 Main net
>
```

A very simple smart contract

Remix - Solidity IDE - Firefox Developer Edition

Remix - Solidity IDE

https://remix.ethereum.org/#optimize=false&version=sol

browser/Message.sol

Compile | Run | Settings | Analysis | Debugger | Support

browser

config

```
pragma solidity ^0.4.24;

contract Message {
    function getMessage() public view returns(string) {
        return "Hello world!";
    }
}
```

Environment: Injected Web3 Ropsten (3)

Account: 0x6c2...0c37f (2.261602178999721

Gas limit: 3000000

Value: 0 wei

Message

Deploy

1F8dE0D441e4735A77F069F7fFBA At Address

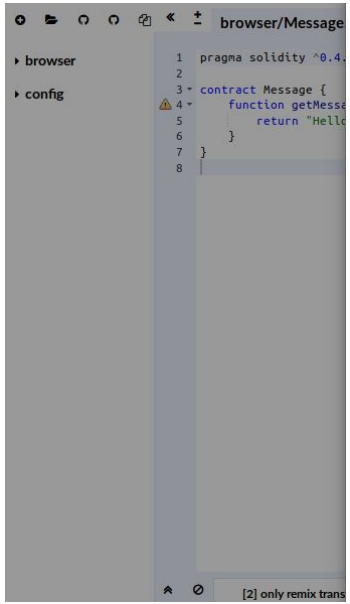
0 pending transactions

[2] only remix transactions, script

Search transac

ABI — Application Binary Interface

The ABI is a json that describes the deployed contract and its functions. It allows to contextualize the contract and call its functions



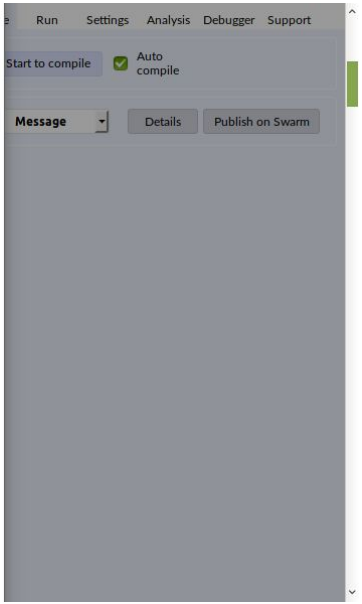

```
{
    "linkReferences": {},
    "object": "6080604052348015610010576000080fd5b5061013f806100206000396000f3006080604
    "opcodes": "PUSH1 0x80 PUSH1 0x40 MSTORE CALLVALUE DUP1 ISZERO PUSH2 0x10 JUMPI PUS
    "sourceMap": "26:113:0:-;;;;8:9:-1;5:2;;;30:1;27;20:12;5:2;26:113:0;;;;;;;"
}
```


ABI

```
0:
  constant: true
  inputs:
  name: getMessage
  outputs:
    0:
      name:
      type: string
  payable: false
  stateMutability: view
  type: function
```

WEB3DEPLOY

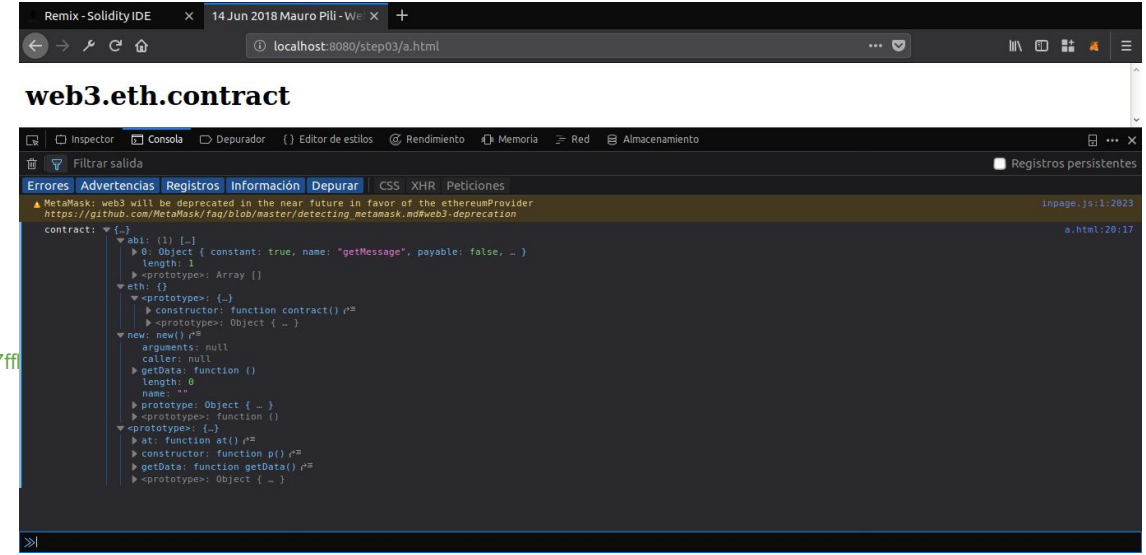
```
var messageContract = web3.eth.contract([{"constant":true,"inputs":[],"name":"getMessag
```



web3.eth.contract

```
pragma solidity ^0.4.24;

contract Message {
    function getMessage() public view returns(string) {
        return "Hello world!";
    }
}
```

```
window.addEventListener('load', function() {
  if (typeof web3 !== 'undefined') {
    var ABI = [ {
      "constant": true,
      "inputs": [],
      "name": "getMessage",
      "outputs": [ { "name": "", "type": "string" } ],
      "payable": false,
      "stateMutability": "view",
      "type": "function"
    } ];
    var address = "0xbc90e0fc10ed1f8de0d441e4735a77f069f7ff
    var contract = new web3.eth.contract(ABI, address);
    console.log('contract:', contract);
```

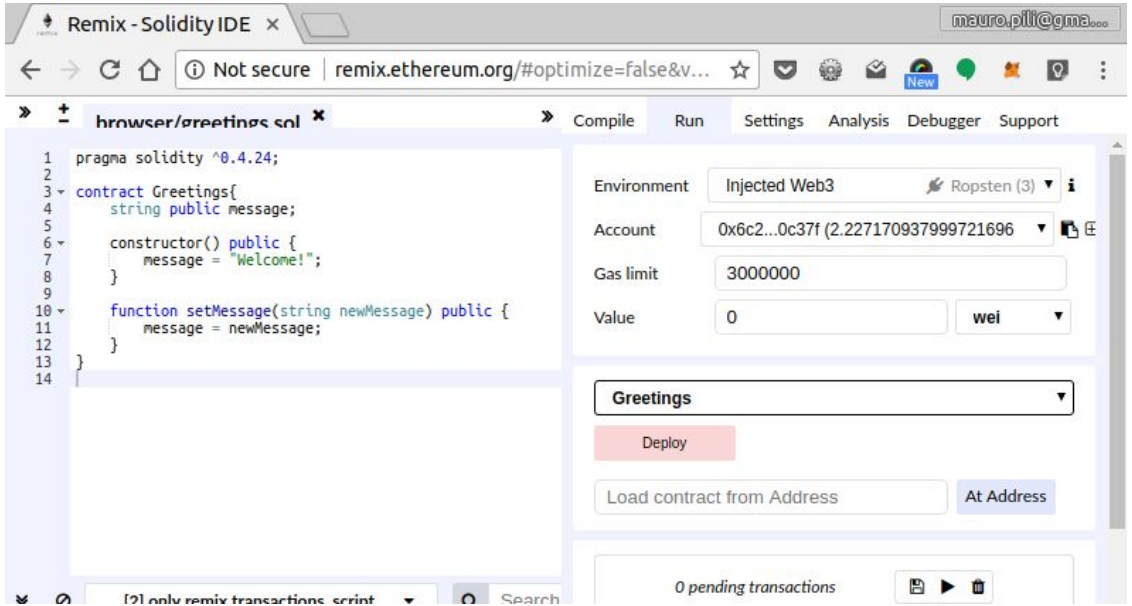


```html
<!DOCTYPE html>
<head>
  <script type="text/javascript">
      window.addEventListener('load', function() {
        if (typeof web3 !== 'undefined') {
          var ABI = [ {
            "constant": true, "inputs": [],
            "name": "getMessage",
            "outputs": [ { "name": "", "type": "string" } ],
            "payable": false, "stateMutability": "view",
            "type": "function"
          } ];
          var address = "0xbc90e0fc10ed1f8de0d441e4735a77f069f7ffba";
          const contract = (web3.eth.contract(ABI)).at(address);
          contract.getMessage(
           (err,res) => {
            console.log("err:",err,"res:",res);
            document.getElementById('message').innerText = res;
           }
          );
        }
      })
  </script>
</head>
<body>
<h1>Get a message from the smart contract</h1>
Message: <strong id="message"></strong>
</body>
</html>
```

```
pragma solidity ^0.4.24;

contract Message {
    function getMessage() public view returns(string) {
        return "Hello world!";
    }
}
```

A smart contract that **writes** in the blockchain

```
pragma solidity ^0.4.24;

contract Greetings{
   string public message;

   constructor() public {
       message = "Welcome!";
   }

   function
     setMessage(string newMessage)
     public {
       message = newMessage;
     }
}
```



A smart contract that **writes** in the blockchain

```
pragma solidity ^0.4.24;

contract Greetings{
   string public message;

   constructor() public {
       message = "Welcome!";
   }

   function
     setMessage(string newMessage)
     public {
       message = newMessage;
     }
}
```

```
var ABI = [
            {
                "constant": false,
                "inputs": [{"name": "newMessage", "type": "string" }],
                "name": "setMessage",
                "outputs": [],
                "payable": false,
                "stateMutability": "nonpayable",
                "type": "function"
            },
            {
                "inputs": [],
                "payable": false,
                "stateMutability": "nonpayable",
                "type": "constructor"
            },
            {
                "constant": true,
                "inputs": [],
                "name": "message",
                "outputs": [{"name": "","type": "string"}
                ],
                "payable": false,
                "stateMutability": "view",
                "type": "function"
            }
        ];
```

writing in the blockchain

```
<!DOCTYPE html>
<head>
   <script type="text/javascript">
      function getMessage() {...};
      function setMessage() {...};
   </script>
</head>
<body>
  <h1>Set a message in the smart contract</h1>
  Message: <strong id="message"></strong>
  <p>Enter a new message:
    <input id="newmessage" type="text">
  </p>
  <button id="button" onclick="setMessage()">
     set message
  </button>
</body>
</html>
```

Set a message in the smart contract

Message: **"Welcome!"**

Enter a new message: [ ]

[set message]

Set a message in the smart contract

Message: **"Welcome!"**

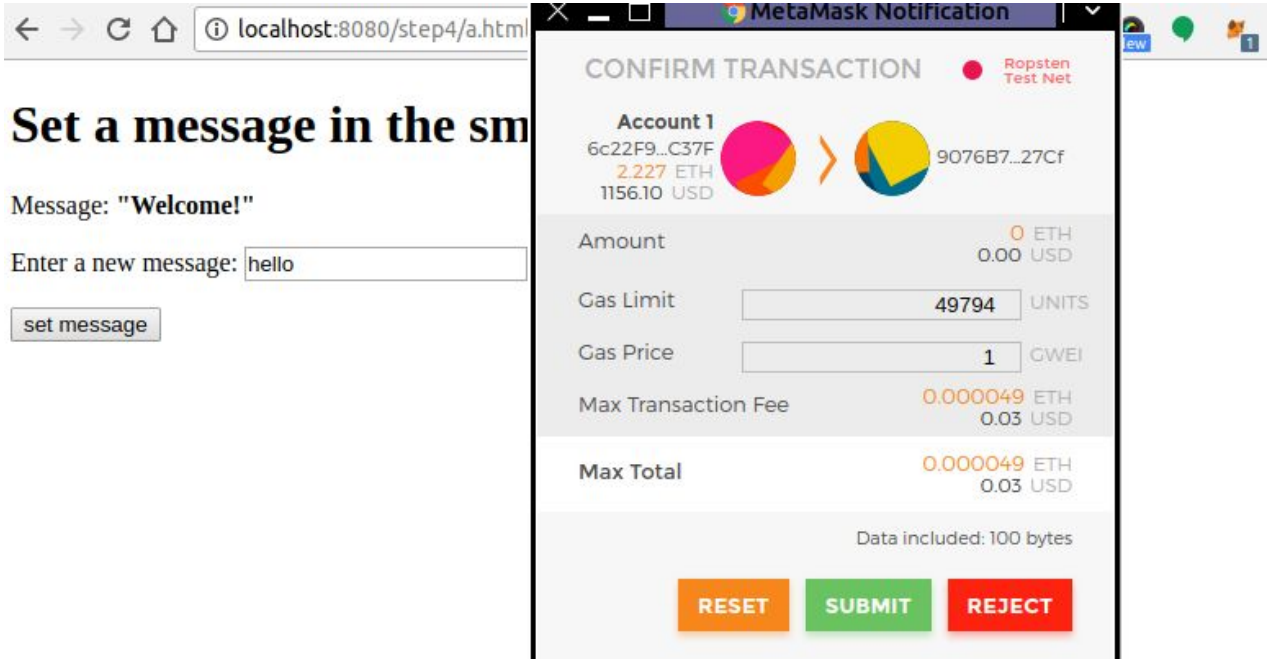
Enter a new message:

set message

```
<!DOCTYPE html>
<head>
  <script type="text/javascript">

var ABI = [...];
var address = "0x9076b714fac55e114a59501eddcc6fc432d727cf";

function getMessage() {
  const contract = (web3.eth.contract(ABI)).at(address);
  contract.message((err,res) => {
     console.log("err:",err,"res:",res);
     document.getElementById('message').innerText = res;
  });
};

function setMessage() {
  var newmessage =
""+document.getElementById('newmessage').value+"";
  const contract = (web3.eth.contract(ABI)).at(address);
  contract.setMessage(newmessage, (err,res) => {
     console.log("setMessage err:",err,"res:",res);
     getMessage();
  });
};

window.addEventListener('load', function() {
  if (typeof web3 !== 'undefined') {
     getMessage();
  } else {
     console.log('Are you using a browser without Web3 capabilities?');
  }
})

</head>
<body>
  <h1>Set a message in the smart contract</h1>
  Message: <strong id="message"></strong>
  <p>Enter a new message:
    <input id="newmessage" type="text">
  </p>
  <button id="button" onclick="setMessage()">
    set message
  </button>
</body>
</html>
```

writing in the blockchain

Web3 (contract.setMessage) calls Metamask that opens a window to interact with the user



pending transaction ...

Sicuro | https://ropsten.etherscan.io/address/0x9076b714fac55e114a59501eddcc6fc432d727cf

ROPSTEN Etherscan The Ethereum Block Explorer

ROPSTEN (Revival) TESTNET | Search by Address / Txhash / Block / Tok

HOME | BLOCKCHAIN | TOKEN | CHART

Contract 0x9076B714fac55E114a59501eDDCC6Fc432d727Cf

Home /

Contract Overview

Balance:	0 Ether
Transactions:	10 txns

Misc

Contract Creator:	0x6c22f997e8aae45... at txn 0x7fb717d5edc3da

Transactions | Code | Events

Latest 10 txns (+1 PendingTxn)

TxHash	Block	Age	From		To	Value
0x68f3f7f066a4d6b5...	(pending)	15 secs ago	0x6c22f997e8aae45...	IN	0x9076b714fac55e1...	0 Ether
0x64374ce3c5e2d5...	3418232	1 day 20 hrs ago	0x6c22f997e8aae45...	IN	0x9076b714fac55e1...	0 Ether
0x1634308c3bb692...	3418227	1 day 20 hrs ago	0x6c22f997e8aae45...	IN	0x9076b714fac55e1...	0 Ether

https://ropsten.etherscan.io/address/0x9076b714fac55e114a59501eddcc6fc432d727cf

Remix - Solidity IDE

Non sicuro | remix.ethereum.org/#opti...

browser/Multisig.sol | browser/Greetings.so

Compile | Run | Settings | Analysis | Debugger | Support

```
pragma solidity ^0.4.24;

contract Greetings{
    string public message;

    constructor() public {
        message = "Welcome!";
    }

    function setMessage(string newMessage) public {
        message = newMessage;
    }
}
```

0x9076b714fac55e114a59501eddcc6fc | At Address

0 pending transactions

Greetings at 0x907...727cf (blockchain)

setMessage | string newMessage

message

0: string: "hello"