Ethereum in a browser

Jirka Chadima, Fragaria

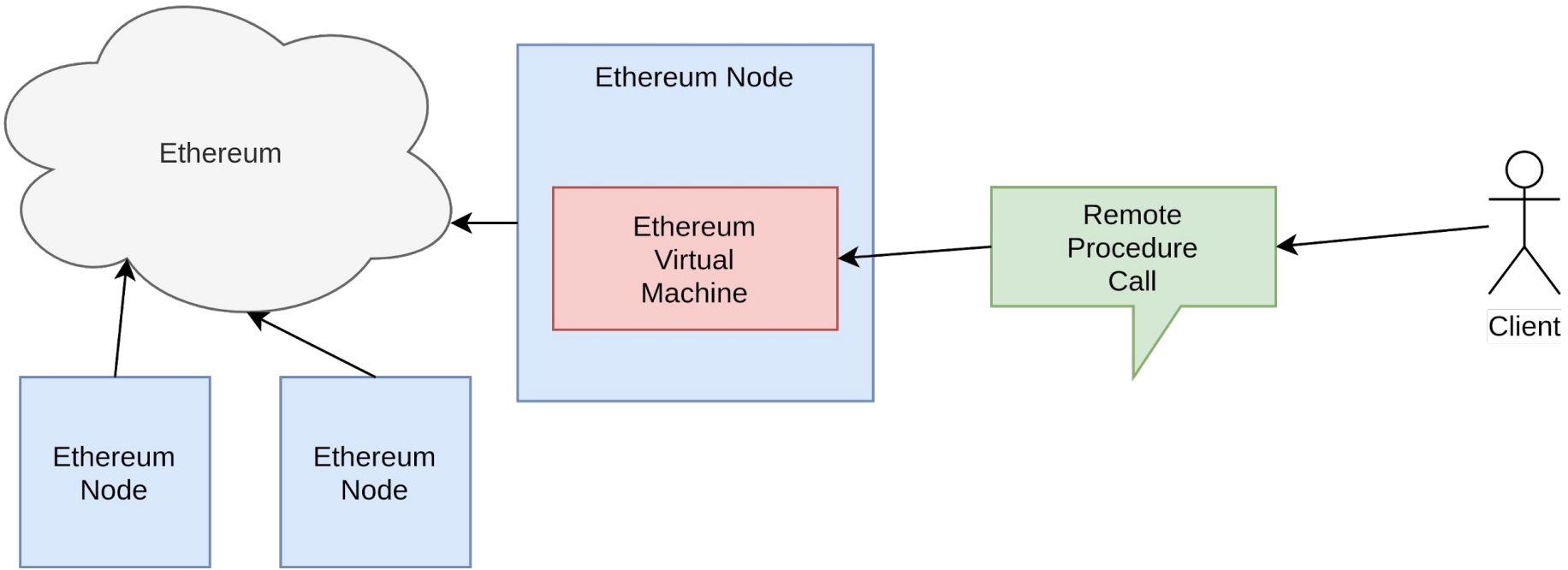
 @jirkachadima

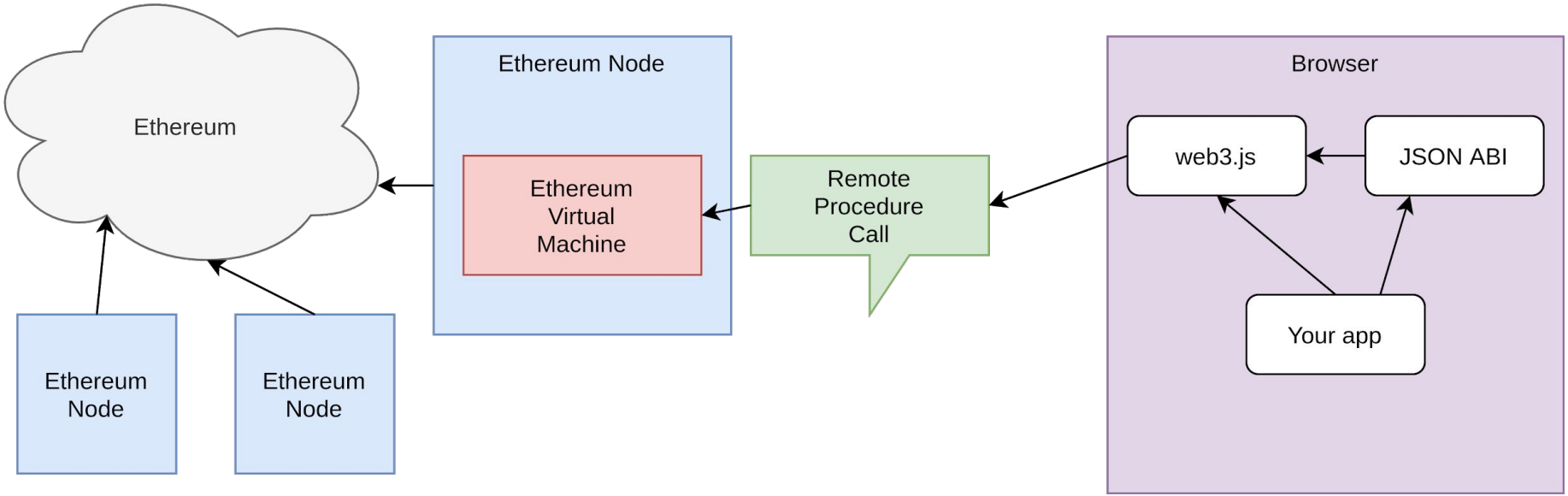
Who am I?

- Full stack developer at Fragaria
  - Web applications, prototypes
- Developer for Winding Tree (2018-2019)
  - Travel industry on Ethereum

Contents

- Getting a smart contract into a program
- Talking to Ethereum from a webpage with web3.js
- Building a P2P payment gateway for a token


Ethereum
Ethereum Node
Ethereum
Virtual
Machine
Remote
Procedure
Call
Client
Ethereum
Node
Ethereum
Node


Ethereum
Ethereum Node
Ethereum Virtual Machine
Remote Procedure Call
Browser
web3.js
JSON ABI
Your app
Ethereum Node
Ethereum Node

Application Binary Interface

- A list of methods available on a smart contract
- Usually distributed as a JSON file
- Product of smart contract compilation

```
50      function balanceOf(address account) public view override returns (uint256) {
51          return _balances[account];
52      }
53
54      /**
55       * @dev See {IERC20-transfer}.
56       *
57       * Requirements:
58       *
59       * - `recipient` cannot be the zero address.
60       * - the caller must have a balance of at least `amount`.
61       */
62      function transfer(address recipient, uint256 amount) public virtual override returns (bool) {
63          _transfer(_msgSender(), recipient, amount);
64          return true;
65      }
66
67      /**
68       * @dev See {IERC20-allowance}.
69       */
70      function allowance(address owner, address spender) public view virtual override returns (uint256) {
71          return _allowances[owner][spender];
72      }
73
74      /**
75       * @dev See {IERC20-approve}.
76       *
```

ERC20.sol

ERC20.sol

```
{
  "constant": false,
  "inputs": [
    {
      "name": "to",
      "type": "address"
    },
    {
      "name": "value",
      "type": "uint256"
    }
  ],
  "name": "transfer",
  "outputs": [
    {
      "name": "",
      "type": "bool"
    }
  ],
  "payable": false,
  "stateMutability": "nonpayable",
  "type": "function"
},
```

Method call

- Encode method signature and arguments based on the ABI
- You end up with a binary data
- That binary data can be executed by EVM

- Transactions are the same, but with a signature

Method call in web3.js

```
import IERC20 from "@openzeppelin/contracts/build/contracts/IERC20.json";
import Web3 from "web3";

const web3 = new Web3(rpcprovider);

const tokenOnRopstenAddress = '0xB6e225194a1C892770c43D4B529841C99b3DA1d7';
const tokenInstance = new web3.eth.Contract(IERC20.abi, tokenOnRopstenAddress);

const balance = tokenInstance
      .methods
      .balanceOf('0xB6e225194a1C892770c43D4B529841C99b3DA1d7')
      .call();
```

TX call in web3.js

```
tokenInstance.methods.transfer(
            '0x4435789eeddc628357a81cb43c1aa9e268457931',
            1000000000000
        ).send({
            from: window.ethereum.selectedAddress
            // value, gas, gasPrice, data, nonce
        })
        .once('transactionHash', (hash) => {
            console.log('tx got its hash - check etherscan', hash);
        }).once('receipt', (receipt) => {
            console.log('receipt', receipt);
        }).on('confirmation', (confNumber, receipt) => {
            console.log('confirmation', confNumber);
        }).then((receipt) => {
            console.log('tx has been mined', receipt)
        });
```

Getting a provider

- You need to sign transactions for EVM
- Key management is hard
- Use something that exists
  - Ethereum node - Infura
  - “Wallet” - Metamask

- Standardization is taking its time (EIP 1102, EIP 1193)

DEMO TIME

Tools?

- Plethora of libraries in various states of maturity and obsolescence

- Truffle and Truffle Boxes (https://www.trufflesuite.com/boxes)
- OpenZeppelin and Starter Kits (https://openzeppelin.com/starter-kits/)

Q&A

https://github.com/JirkaChadima/js-meetup-blockchain

PragueJS Meetup, 2020/02/25

 @jirkachadima

Resources

- Web3.js docs - https://web3js.readthedocs.io/
- Metamask - http://metamask.io/

- Online solidity IDE - https://remix.ethereum.org/
- Clickable Ethereum playground - https://eth.build/