VIETNAM WEB SUMMIT

Step 1.

Build and Secure your D-App on Public Blockchain

Hoàng Thanh Tùng

Co-Founder & Product Manager @ TomoChain

Outline

- Build a DApp?
- Secure your DApp?
- Conclusion
- Game

Step 0:
What’s Blockchain?

Blockchain là một database chỉ ghi thêm dữ liệu đúng đắn. Các bản ghi được sắp xếp theo một chật tự liên tục và móc lối chặt chẽ với nhau

Properties

- Transparent
- Trustless
- Immutable
- Decentralize

Ethereum

- Vitalik Buterin
- 30/7/2015

Smart Contract

- Execute Code
- Ethereum Virtual Machine
- Solidity

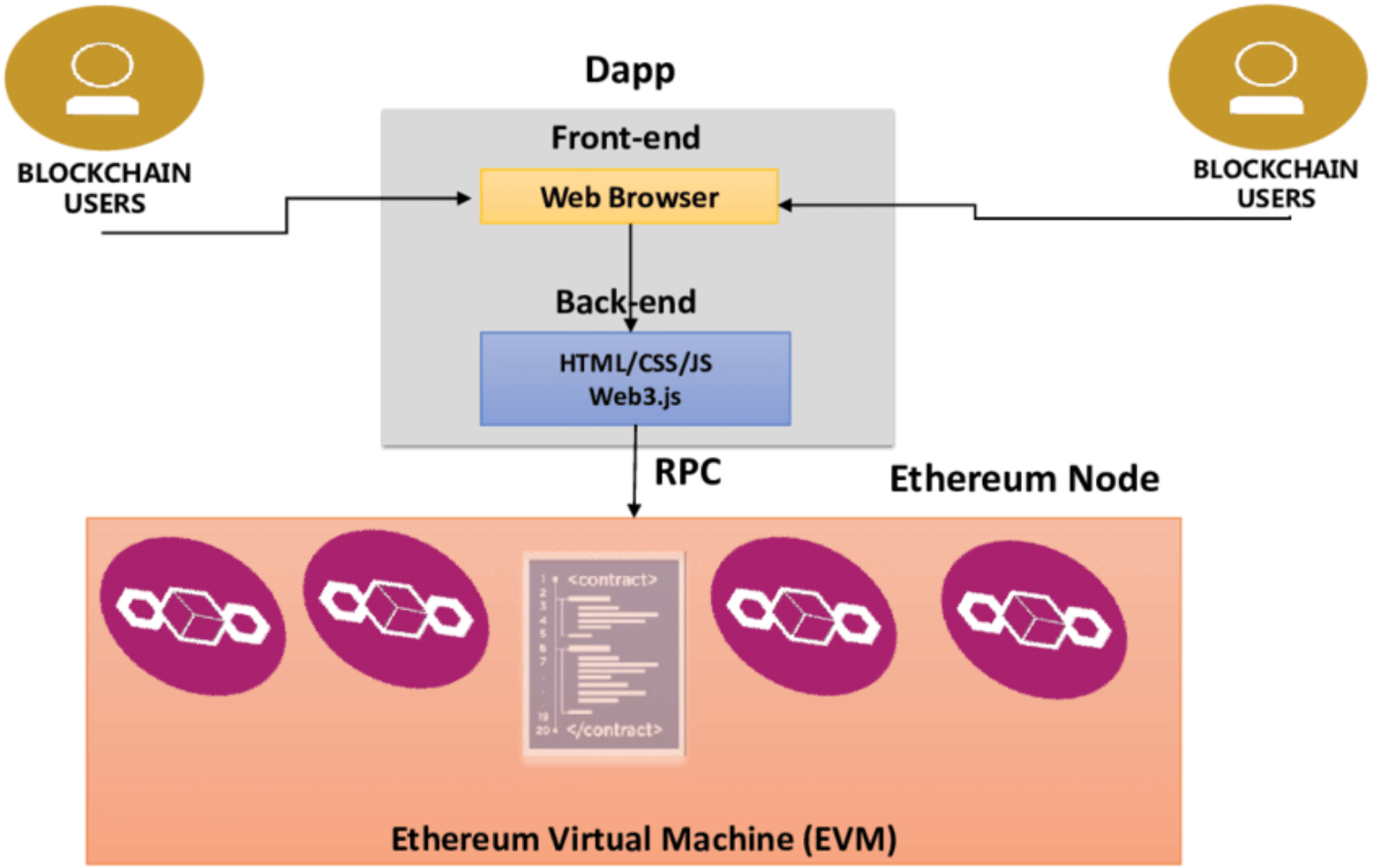
Use Case

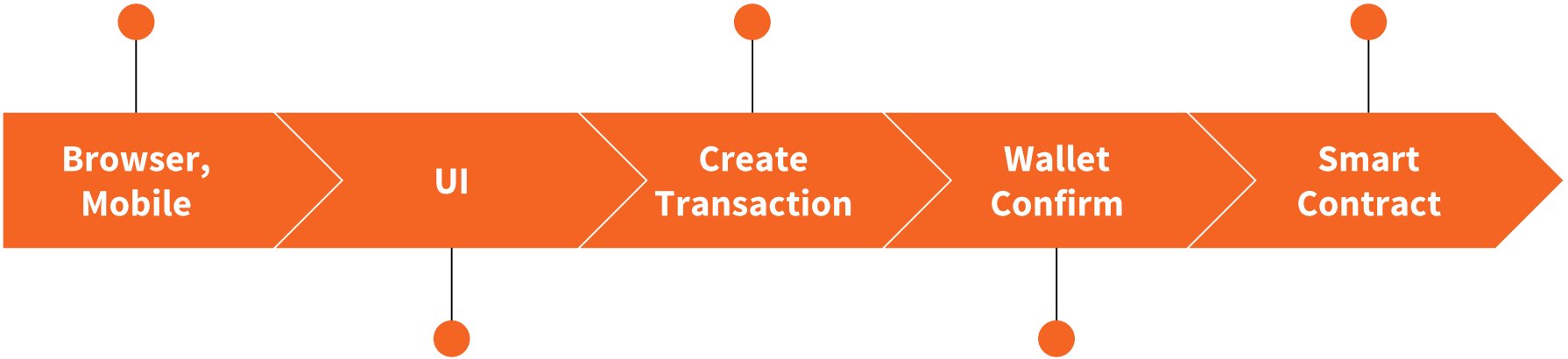
- Game
- Exchange
- Finance
- etc...

How to build a DApp?

Structure of a DApp

- Smart Contract
- Wallet
- Front-end


Dapp
Front-end
Web Browser
Back-end
HTML/CSS/JS
Web3.js
BLOCKCHAIN USERS
BLOCKCHAIN USERS
RPC
Ethereum Node
<contract>
</contract>
Ethereum Virtual Machine (EVM)


Browser, Mobile
UI
Create Transaction
Wallet Confirm
Smart Contract

Step by Step Build a DApp

- Smart Contract
- Unit Test
- Front-end

Lucky Game

- Nhập tên/Số điện thoại để tham gia
- Mỗi số điện thoại chỉ được tham gia một lần
- Sau 30 blocks sẽ ngẫu nhiên chọn ra 1 người may mắn
- Kết thúc

Write Smart Contract

- Solidity
- Remix.Ethereum.org

```
struct Player {
  bytes32 key;
  bytes32 name;
  uint luckyNumber;
}

Player[] public players;
bool haveWinner;
uint public winner;
mapping(bytes32 => uint) indexByKey;
mapping(bytes32 => bool) checkKey;
uint public finishBlock;
uint public durationBlock;
```

```
function join(bytes32 name, bytes32 key) public
  validate(name)
  onTime() {
  uint index = players.length;
  if (checkKey[key] == false) {
    players.push(Player({
      name: name,
      key: key,
      luckyNumber: index
    }));

    checkKey[key] = true;
    indexByKey[key] = index;
  }
}
```

```
function rand(uint max) private view returns (uint256 result) {
  uint256 lastBlockNumber = block.number - 1;
  return uint256(blockhash(lastBlockNumber)) % max;
}
```

```
function drawWinner() public
  isFinish() {
  if (haveWinner == false) {
    winner = rand(players.length);
    haveWinner = true;
    emit DrawWinner(
        players[winner].name,
        players[winner].key,
        players[winner].luckyNumber);
  }
}
```

Unit Test

Compile | Run | Analysis | Testing | Debugger | Settings

Unit Testing

Test your smart contract by creating a foo_test.sol file (open ballot_test.sol to see the example).
You will find more informations in the documentation Then use the stand alone NPM module remix-tests to run unit tests in your Continuous Integration https://www.npmjs.com/package/remix-tests.
For more details, see How to test smart contracts guide in our documentation.

Generate test file

No test file available

Run Tests

```
contract TestJoinGame {

    LuckyContract lucky;
    function beforeAll () public {
        lucky = new LuckyContract(2);
    }

    function joinGame1 () public {
        uint n = lucky.numberOfPlayers();
        lucky.join("tung", "1");
        Assert.equal(lucky.numberOfPlayers() > n, true, "should be true");
    }

    function joinGame2 () public {
        uint n = lucky.numberOfPlayers();
        lucky.join("hoang", "2");
        Assert.equal(lucky.numberOfPlayers() > n, true, "should be true");
    }

    function joinGameAlready () public {
        uint n = lucky.numberOfPlayers();
        lucky.join("hoang", "2");
        Assert.equal(lucky.numberOfPlayers() == n, true, "should be true");
    }

    function joinGameInvalidName () public {
        uint n = lucky.numberOfPlayers();
        lucky.join("", "2");
        Assert.equal(lucky.numberOfPlayers() == n, true, "should be true");
    }

    function joinGame3 () public {
        uint n = lucky.numberOfPlayers();
        lucky.join("Thanh", "3");
        Assert.equal(lucky.numberOfPlayers() > n, true, "should be true");
    }
}
```

browser/test_test.sol (TestJoinGame)

- ✓ (Join game1)
- ✓ (Join game2)
- ✓ (Join game already)
- ✓ (Join game3)
- ✓ (Join game invalid name)

browser/test_test.sol

5 passing (0s)

Full Code of LuckyContract

https://github.com/tunght91/LuckyGame

Deploy

- Remix.ethereum.org
- Metamask.io
- Ganache, Rinkeby, TomoChain Testnet

https://remix.ethereum.org/#optimize=false&version=soljson-v0.4.23+commit.124ca40d.js

browser/LuckyContract.sol

```
pragma solidity ^0.4.23;

contract LuckyContract {
  address owner;

  struct Player {
    bytes32 key;
    bytes32 name;
    uint luckyNumber;
  }

  Player[] public players;
  bool public haveWinner;
  uint public winner;
  mapping(bytes32 => uint) indexByKey;
  mapping(bytes32 => bool) checkKey;
  uint public finishBlock;
  uint public durationBlock;

  event Join(bytes32 name, bytes32 key, uint luckyNumber);
  event UpdateName(bytes32 oldName, bytes32 newName, bytes32 key);
  event DrawWinner(bytes32 name, bytes32 key, uint luckyNumber);
  event ForceDrawWinner(bytes32 name, bytes32 key, uint luckyNumbe
  event Reset(uint duration);

  constructor(uint _durationBlock) public {
    owner = msg.sender;
    haveWinner = false;
    winner = 0;
    durationBlock = _durationBlock;
    finishBlock = block.number + durationBlock;
  }


  modifier isOwner() {
    require(
      msg.sender == owner,
      "You don't have permission"
    );
```

TomoChain Testnet

MAS
0xaD12...B1eA

507.8505 TOMO
$89.58 USD

DEPOSIT SEND

History

#99 - 12/10/2018 at 16:20
Sent Ether -2000 TOMO
CONFIRMED -$352.80 USD

#98 - 12/10/2018 at 15:46
Contract Interaction -0 TOMO
CONFIRMED -$0.00 USD

#97 - 12/10/2018 at 15:20

Debugger Settings

Custom (89)

7.8505449337918804

wei

rom Address

tances to interact with.

[2] only remix transactions, script

Search transactions

Compile | Run | Analysis | Testing | Debugger | Settings

Environment: Injected Web3 — Custom (89)

Account: 0xad1...2b1ea (507.8505449337918804

Gas limit: 3000000

Value: 0 wei

LuckyContract

Deploy 10

or

At Address: Load contract from Address

TomoChain Testnet

MAS → New Contract

CONTRACT DEPLOYMENT

0

$0.00

DETAILS | DATA

Origin: remix.ethereum.org

Bytes: 3321

HEX DATA:

0x6080604052348015610010576000080fd5b506040516
02080610cd983398101806040528101908080519060200
1909291905050503360000806101000a81548173ffffffff
ffffffffffffffffffffffffffffffff021916908373ffffffffffffffffffff
ffffffffffffffffff1602179055506000600260006101000a8
1548160ff021916908315150217905550600060003819 0
5550806007819055506007544301600681905550506 1
0c21806100b86000396000f3006080604052600436106

REJECT | CONFIRM

Deployed Contracts

LuckyContract at 0xa0f...cadbf (blockchain)

drawWinner	
forceDrawWinner	
join	bytes32 name, bytes32 key
reset	uint256 _durationBlock
durationBlock	
finishBlock	
getPlayerByKey	bytes32 key
getWinner	
haveWinner	
numberOfPlayers	
players	uint256
winner	

https://scan.testnet.tomochain.com/address/0x37c757ca9cf87e2bac8eb175ffe148be0749d7a1#transactions

TomoScan | Home | Transactions | Accounts | Tokens | Blocks | Need help?

Home > Accounts
> 0x37c757ca9cf87e2bac8eb175ffe148be0749d7a1

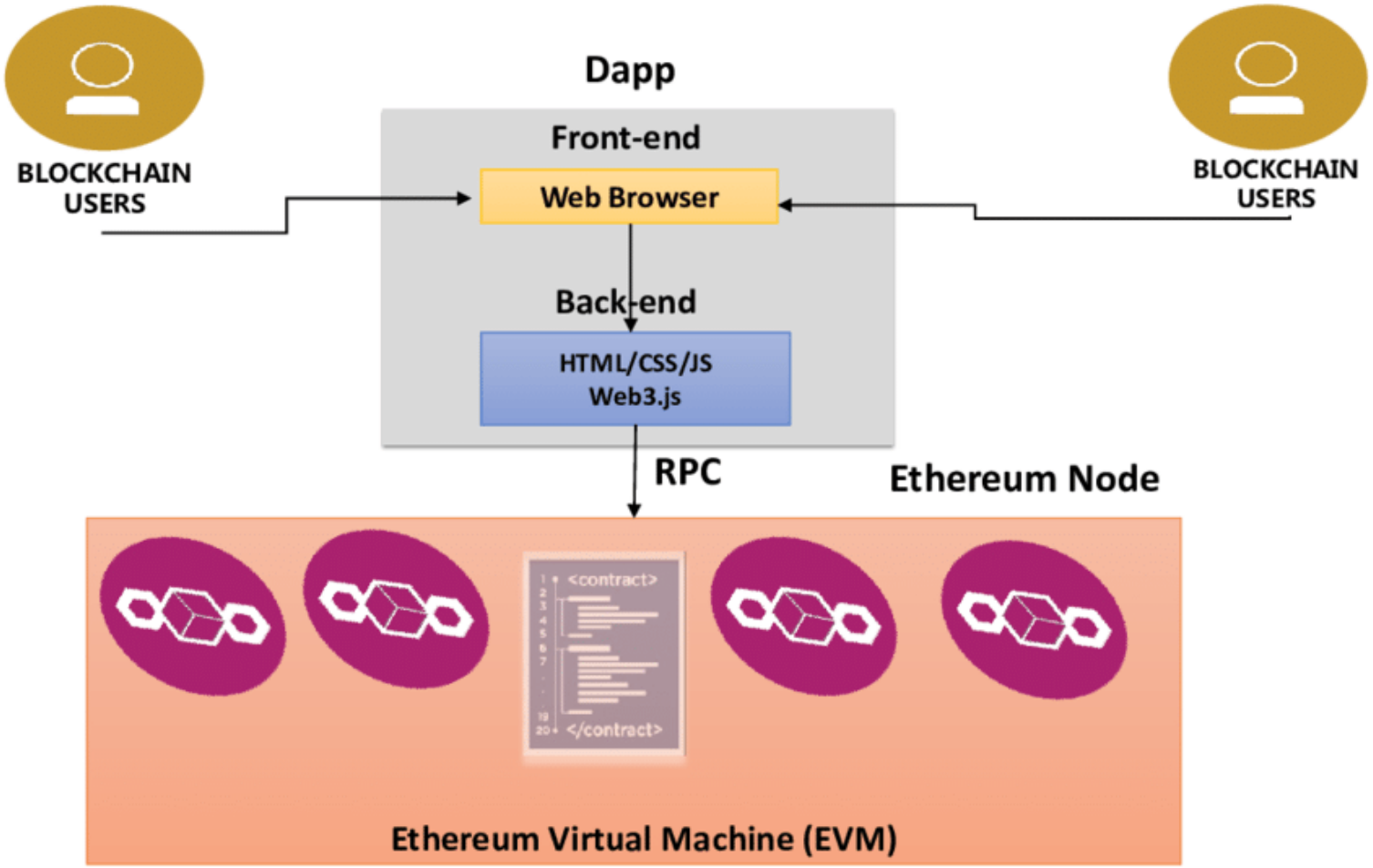
Search

Contract: 0x37c757ca9cf87e2bac8eb175ffe148be0749d7a1

Copy

TOMO Balance	0 TOMO
TOMO USD Value	0
Transactions	167 txns
Contract Creator	0xc412515c31273980222ff579ed85ca22a56d9683 at txns 0x1ddb574ae13d8316a54aeb4988ee0597aba03851f17ada1aad35928287a0d1d7

https://scan.testnet.tomochain.com/address/0x37c757ca9cf87e2bac8eb175ffe148be0749d7a1#transactions


Dapp
Front-end
Web Browser
Back-end
HTML/CSS/JS
Web3.js
BLOCKCHAIN USERS
BLOCKCHAIN USERS
RPC
Ethereum Node
<contract>
</contract>
Ethereum Virtual Machine (EVM)

Front-end

- Web3js
- HTML/CSS/JS
- ReactJS, VueJS

Web3js

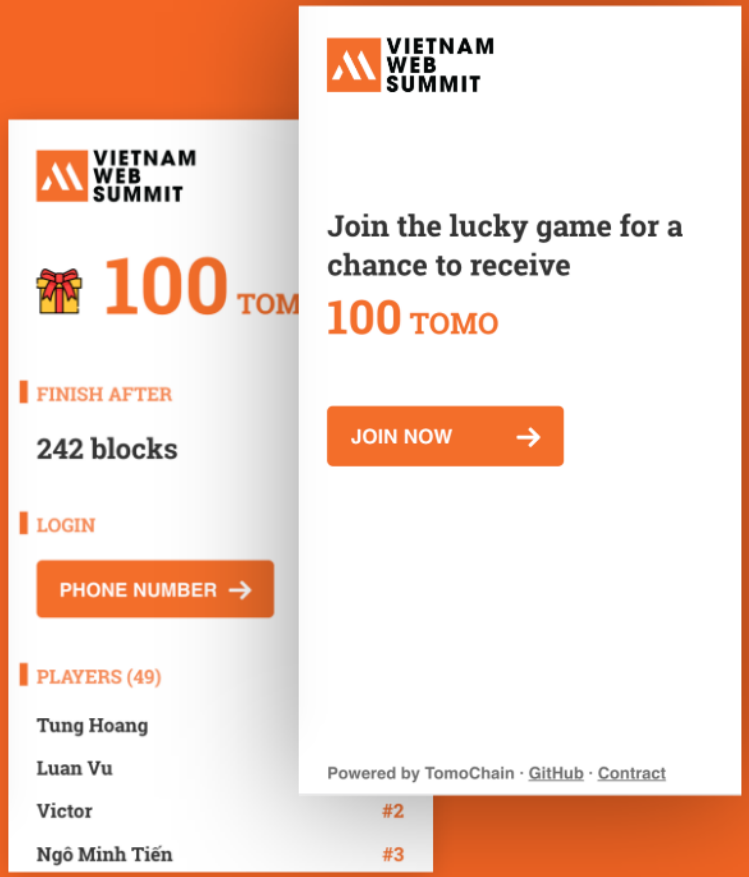
- ABI
- https://tesnet.tomochain.com

```
function join(bytes32 name, bytes32 key) public
  validate(name)
  onTime() {
  uint index = players.length;
  if (checkKey[key] == false) {
    players.push(Player({
      name: name,
      key: key,
      luckyNumber: index
    }));

    checkKey[key] = true;
    indexByKey[key] = index;
  }
}
```

```
{
  "constant": false,
  "inputs": [
    {
      "name": "name",
      "type": "bytes32"
    },
    {
      "name": "key",
      "type": "bytes32"
    }
  ],
  "name": "join",
  "outputs": [],
  "payable": false,
  "stateMutability": "nonpayable",
  "type": "function"
},
```

```
var Web3 = require('web3');
var abi = require('./abi');
var web3 = new Web3('https://testnet.tomochain.com');
var contractAddress = '0x37c757ca9cf87e2bac8eb175ffe148be0749d7a1';
var LuckyContract = new web3.eth.Contract(abi, contractAddress);

LuckyContract.methods
  .numberOfPlayers()
  .call()
  .then(v => parseInt(v));
```

Web3JS + VueJS =



How to secure your DApp?

The DAO Hack 2016

Code Issue Leads To $150 Milion Ether Theft

Smart Contract Security

- External calls
- Immutability
- Privacy
- Overflows & Underflows

External calls

Ethereum Contract can call other Contract

```
mapping (address => uint) private balances;

function withdraw() public {
  uint amount = balances[msg.sender];
  if (!(msg.sender.call.value(amount)())) {
    revert;
  }
  balances[msg.sender] = 0;
}
```

```
mapping (address => uint) private balances;

function withdraw() public {
  uint amount = balances[msg.sender];

  balances[msg.sender] = 0;
  if (!(msg.sender.call.value(amount)())) {
    revert;
  }
}
```

Immutability

A deployed contract can’t be patched

```
contract BlackHole {
    function () payable {}
    function getBalances() returns (uint) {
        return this.balance;
    }
}
```

Privacy

Remember that on-chain data is public

Overflows & Underflows

```
  0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
+ 0x00000000000000000000000000000001
----------------------------------
= 0x00000000000000000000000000000000
```

```
  0x00000000000000000000000000000000
- 0x00000000000000000000000000000001
----------------------------------
= 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
```

Conclusion

- Smart Contract
- Wallet
- Front-end

Conclusion

- Solidity
- Remix
- Metamask
- Unit Test
- Ganache, Rinkeby, TomoChain Testnet

Conclusion

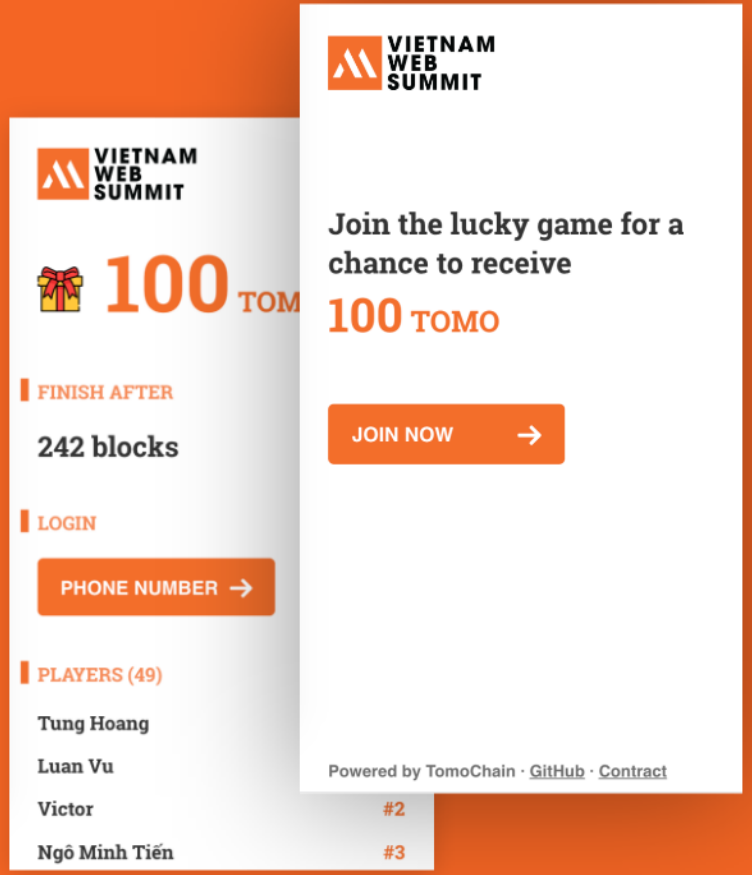
- Web3js
- ABI
- HttpProvider
- HTML/CSS/JS

Conclusion

- External calls
- Immutability
- Privacy
- Overflows & Underflows
- ....

Lucky Game

yesno.fun



Thanks