Ethereum

Demo, Transactions

Prof. Marco Comuzzi
Department of Industrial Engineering
Ulsan National Institute of Science and Technology (UNIST)
mcomuzzi@unist.ac.kr

What is Ethereum?

Conceived by Vitalik Buterin in 2013, live in 2015

It is a public blockchain

It is a cryptocurrency: it is used by its users to record the exchange of digital "tokens", denominated ETH

It has smart contracts, so…

The ETH balance of nodes is only a part of the Ethereum state

1.

Ethereum Demo

What do we need?

The Goerli Ethereum Test Network

A few accounts on the Goerli network

Some (fake) ETH for each account

A wallet

A smart contract (BC4C-CGM >> BC4C-goETH)

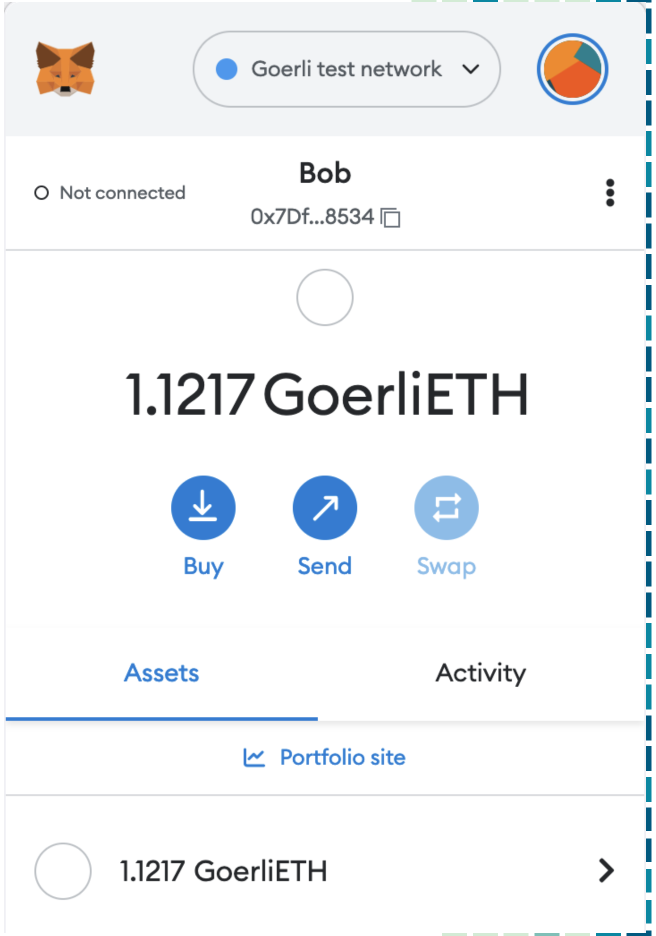
The “Metamask” wallet

Runs as a plugin of any browser

Can handle multiple accounts

Can connect to Ethereum main and test networks

Integrated with “Remix” smart contract IDE



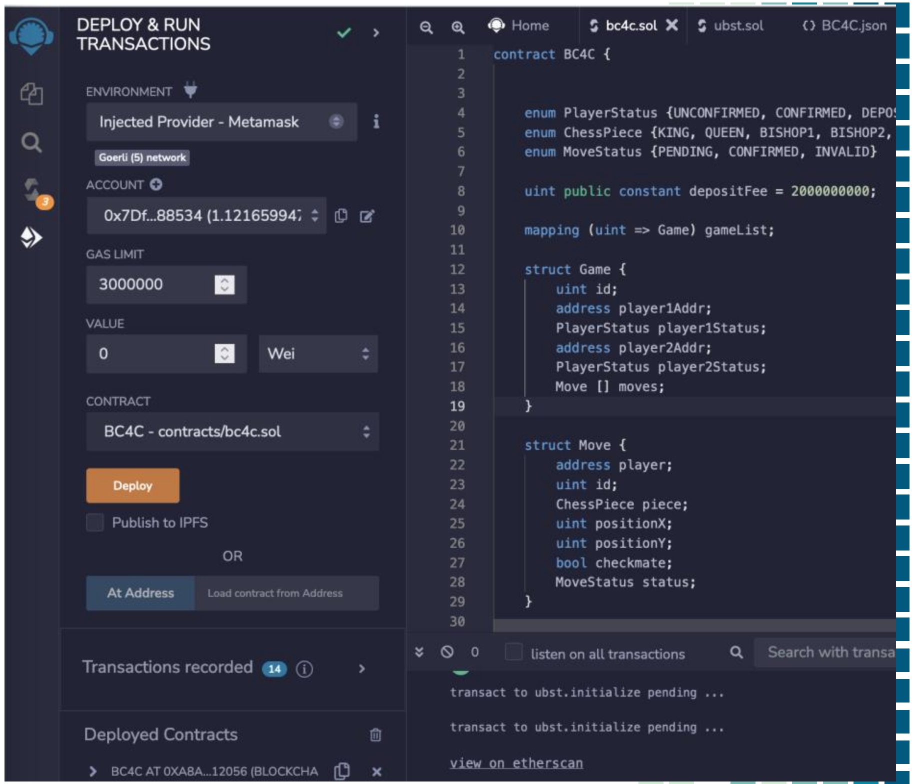
The “Remix” Smart Contract IDE

Runs in the browser (remix.ethereum.org)

Used to compile, deploy and call Ethereum smart contracts

Smart contract: BC4C-ETH

BC4C smart contract, with deposits paid in (fake) ETH



ENVIRONMENT: Injected Provider - Metamask
Goerli (5) network
ACCOUNT: 0x7Df...88534 (1.12165994
GAS LIMIT: 3000000
VALUE: 0 Wei
CONTRACT: BC4C - contracts/bc4c.sol

BC4C AT 0XA8A...12056 (BLOCKCHA

```
contract BC4C {


    enum PlayerStatus {UNCONFIRMED, CONFIRMED, DEPOS
    enum ChessPiece {KING, QUEEN, BISHOP1, BISHOP2,
    enum MoveStatus {PENDING, CONFIRMED, INVALID}

    uint public constant depositFee = 2000000000;

    mapping (uint => Game) gameList;

    struct Game {
        uint id;
        address player1Addr;
        PlayerStatus player1Status;
        address player2Addr;
        PlayerStatus player2Status;
        Move [] moves;
    }

    struct Move {
        address player;
        uint id;
        ChessPiece piece;
        uint positionX;
        uint positionY;
        bool checkmate;
        MoveStatus status;
    }

```

```
transact to ubst.initialize pending ...

transact to ubst.initialize pending ...

view on etherscan
```

Where do we get the fake ETH in Goerli?

https://goerli-faucet.pk910.de/

What did we do?

Transfer ETH among accounts

Create a new game

Confirm game creation

Pay deposits for a game to the smart contract

1.
Ethereum network and transactions

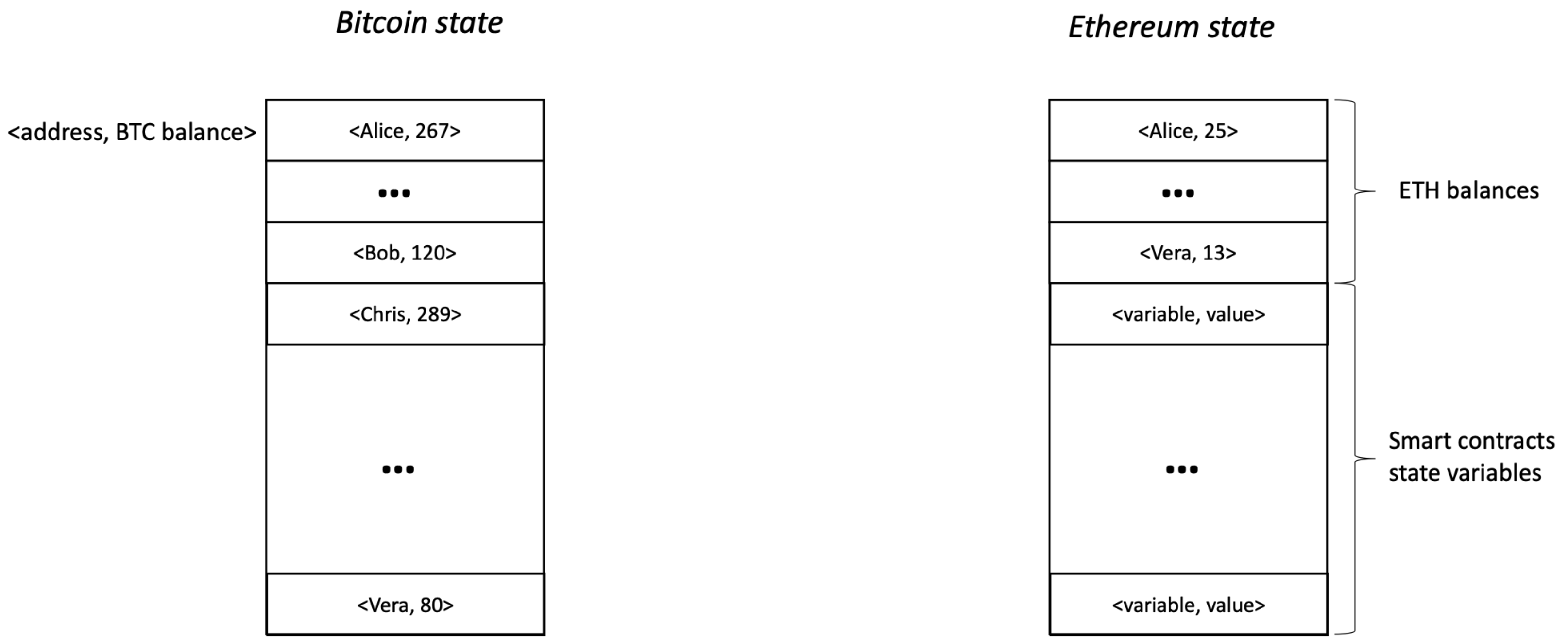
Ethereum network

A public network, like Bitcoin

Anybody can download a client at https://Ethereum.org and run a node

Ethereum state

ETH balance of nodes, State variables of smart contracts



Ethereum accounts

Each node (account) is associated with a unique address, like in Bitcoin

Two types of accounts:

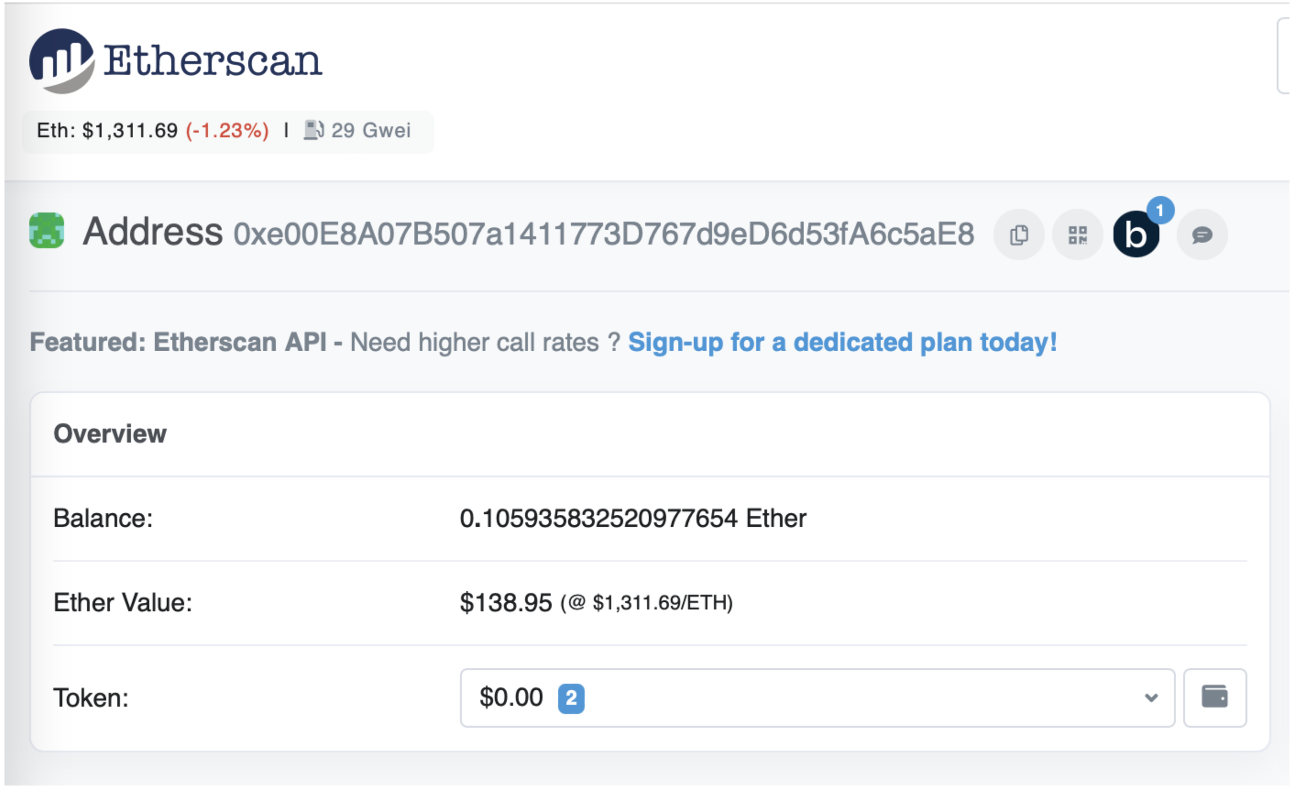
EOA – Externally Owned Accounts

The “normal” accounts, they issue digitally signed transactions

Smart Contract Accounts (SCA)

Identify uniquely a smart contract in the Ethereum network
Can only be the recipient of a tx, not the originator

EOA



Etherscan

Eth: $1,311.69 (-1.23%) | 29 Gwei

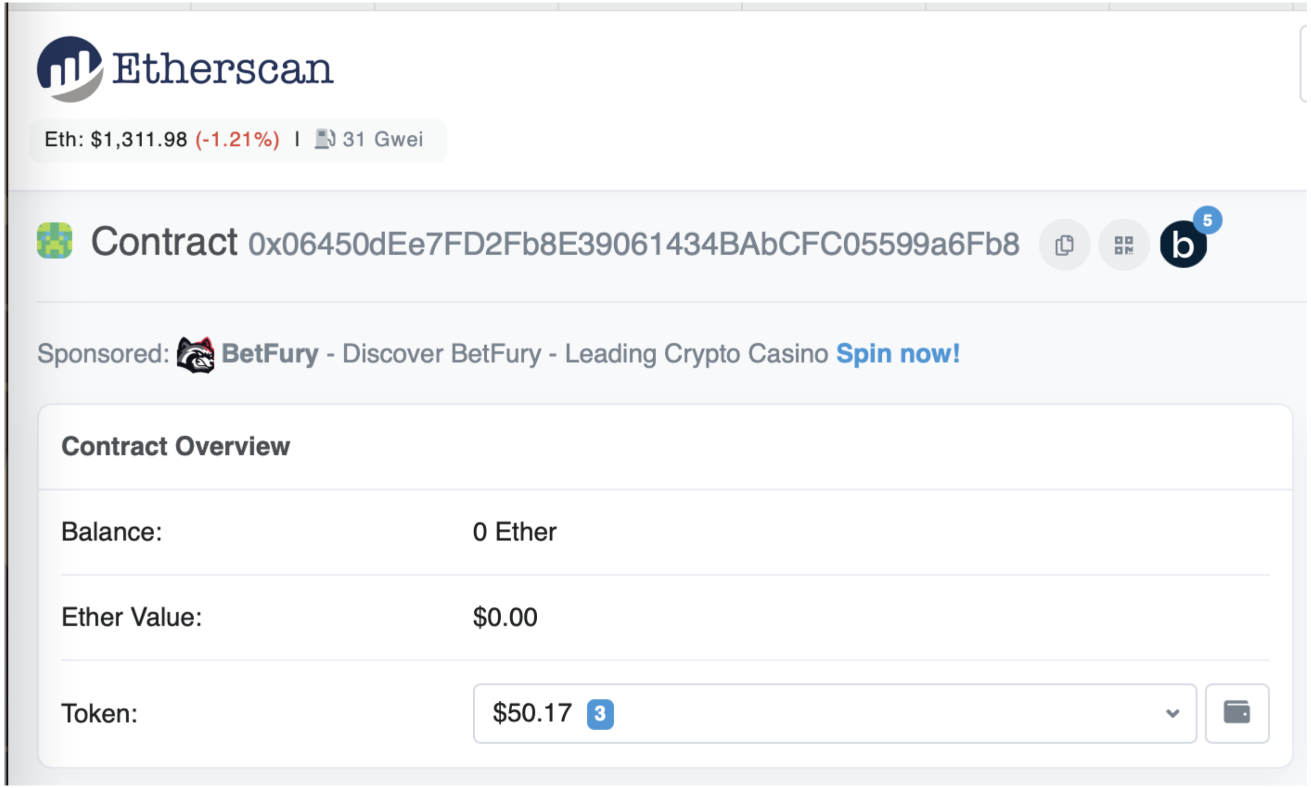
Address 0xe00E8A07B507a1411773D767d9eD6d53fA6c5aE8

Featured: Etherscan API - Need higher call rates ? Sign-up for a dedicated plan today!

Overview

Balance:	0.105935832520977654 Ether
Ether Value:	$138.95 (@ $1,311.69/ETH)
Token:	$0.00 2

Smart Contract Account


Etherscan

Eth: $1,311.98 (-1.21%) | 31 Gwei

Contract 0x06450dEe7FD2Fb8E39061434BAbCFC05599a6Fb8

Sponsored: BetFury - Discover BetFury - Leading Crypto Casino Spin now!

Contract Overview

Balance:	0 Ether
Ether Value:	$0.00
Token:	$50.17 3

Ethereum transactions: 3 types

1. Transactions that transfer ETH from one EOA (sender) to another EOA (Recipient)

2. Transactions that create and distribute a smart contract

3. Transactions that invoke an existing smart contract

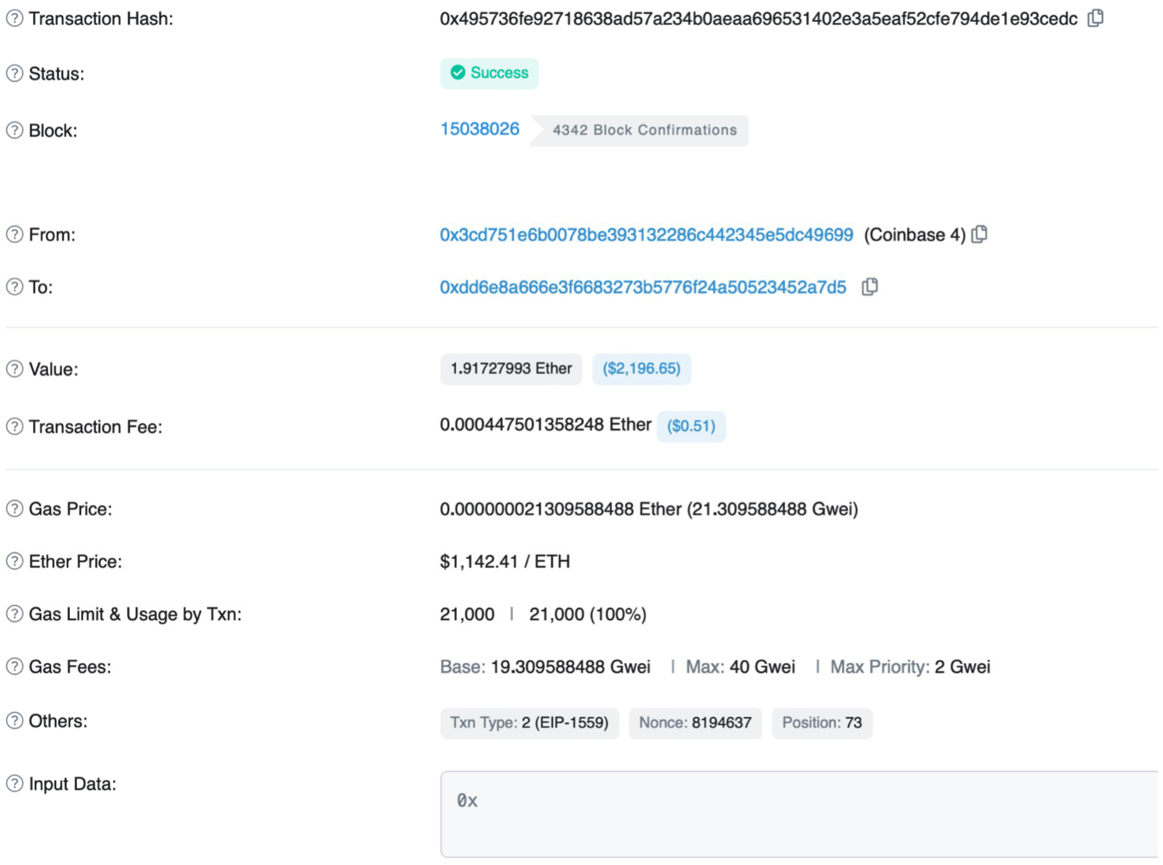
Digital signatures of transactions

Easy

Each transaction is digitally signed by the originator
(= the sender, the deployer of a contract, or the EOA invoking a smart contract)

No UTXOs or many-to-many transactions

Type 1: ETH transfer between EOA



Transaction Hash:	0x495736fe92718638ad57a234b0aeaa696531402e3a5eaf52cfe794de1e93cedc
Status:	Success
Block:	15038026 4342 Block Confirmations
From:	0x3cd751e6b0078be393132286c442345e5dc49699 (Coinbase 4)
To:	0xdd6e8a666e3f6683273b5776f24a50523452a7d5
Value:	1.91727993 Ether ($2,196.65)
Transaction Fee:	0.000447501358248 Ether ($0.51)
Gas Price:	0.000000021309588488 Ether (21.309588488 Gwei)
Ether Price:	$1,142.41 / ETH
Gas Limit & Usage by Txn:	21,000 | 21,000 (100%)
Gas Fees:	Base: 19.309588488 Gwei | Max: 40 Gwei | Max Priority: 2 Gwei
Others:	Txn Type: 2 (EIP-1559) Nonce: 8194637 Position: 73
Input Data:	0x

https://etherscan.io/tx/0x495736fe92718638ad57a234b0aeaa696531402e3a5eaf52cfe794de1e93cedc

(New) Elements of a transaction

Transaction nonce

Input data

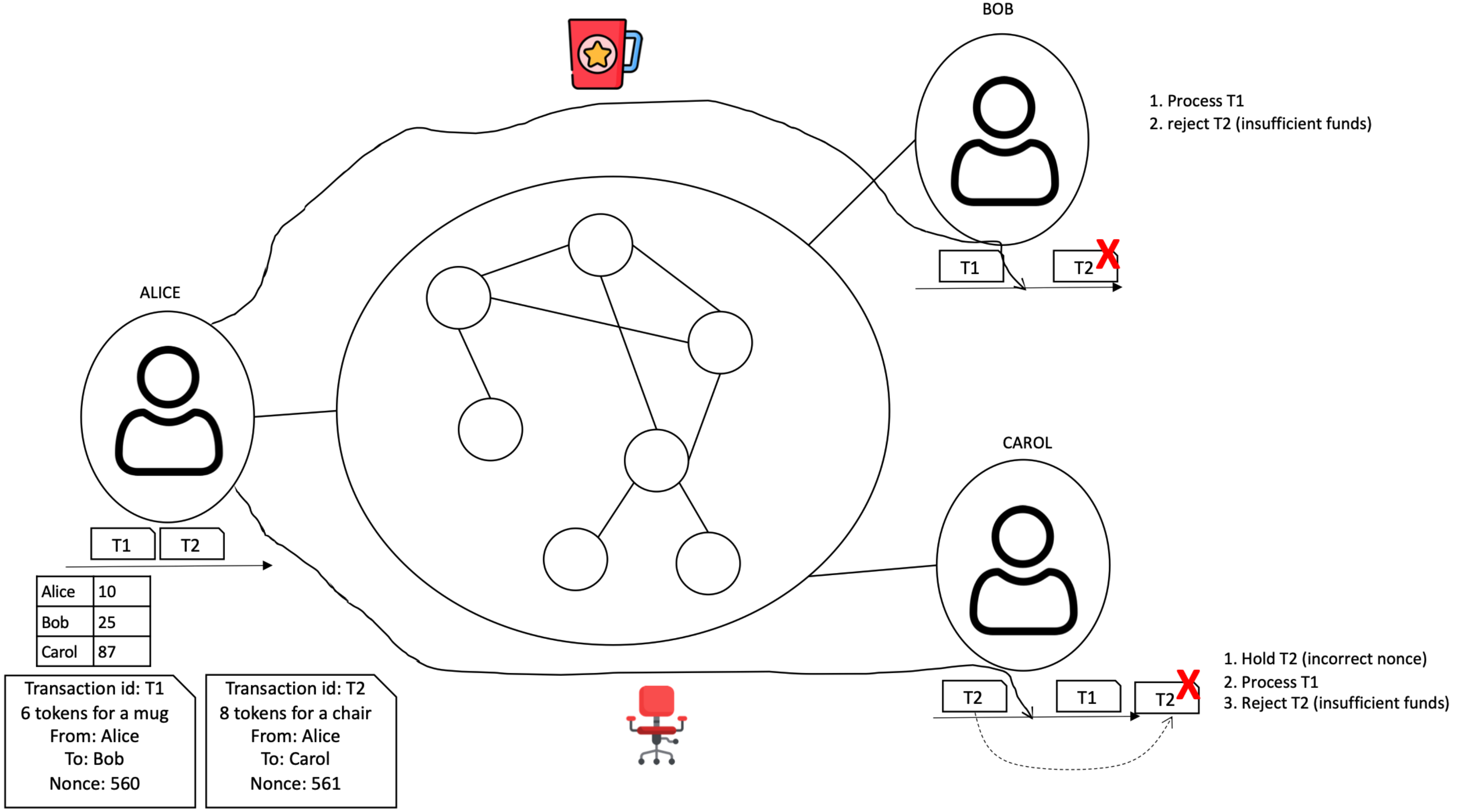
Gas (limit, price, used)

Transaction Nonce

No UTXOs in Ethereum, so how does Ethereum prevent double spending?

Include in each transaction a “nonce” equal to number of transactions issued by the originator until now

A transaction is valid only if its nonce is equal to:
Nonce of the last transaction issued by each originator + 1



1. Process T1
2. reject T2 (insufficient funds)

Alice	10
Bob	25
Carol	87

Transaction id: T1
6 tokens for a mug
From: Alice
To: Bob
Nonce: 560

Transaction id: T2
8 tokens for a chair
From: Alice
To: Carol
Nonce: 561

1. Hold T2 (incorrect nonce)
2. Process T1
3. Reject T2 (insufficient funds)

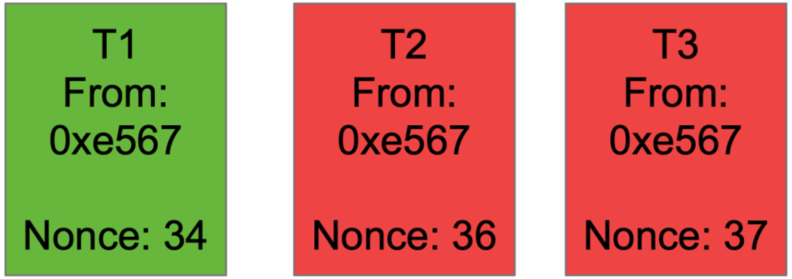
Transaction Nonce :problems

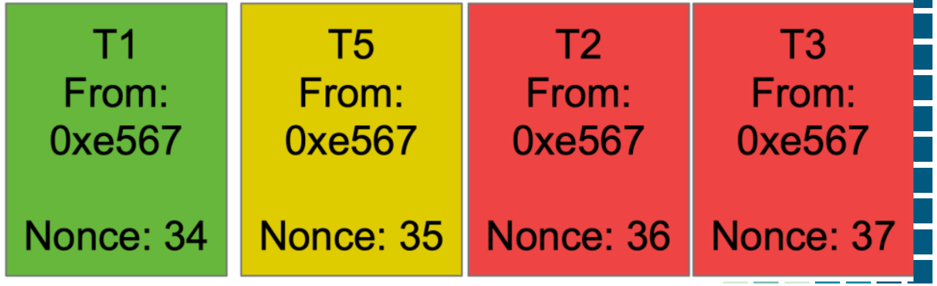
Transactions T2, T3 (and any subsequent one submitted by EOA 0xe567) will never be mined!

They are not valid (wrong nonce)

Once T5 is submitted, T5, T2 and T3 will all be mined eventually

- There is no way to “revoke” a transaction





Marco Comuzzi mcomuzzi@unistac.kr

How to set the transaction nonce?

If originator uses only client application, the client can do it

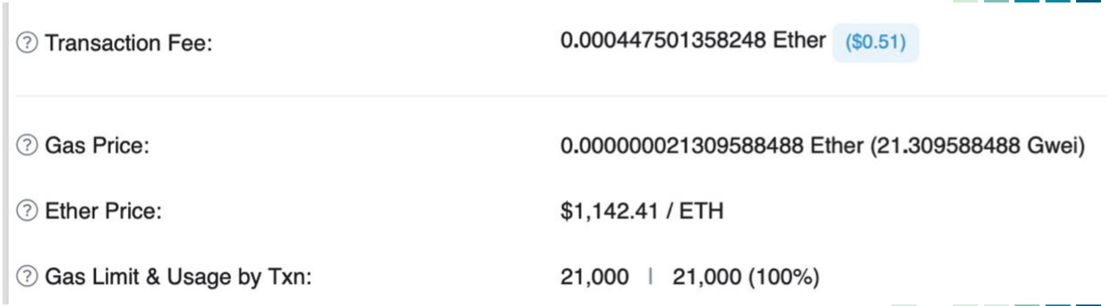
If multiple clients (wallets) are used, then some coordination is needed

In any case, the last valid transaction issued by an originator is somewhere in the Ethereum ledger

Transaction input data

Type of Ethereum transaction	***Content of input data field***
Transactions transferring ETH between EOAs	Ignored E.g. payment reason/reference
Transactions deploying a new smart contract	Byte code of the smart contract
Transactions invoking an existing smart contract	Function to invoke and (if necessary) input parameters

Transaction Gas



ⓘ Transaction Fee:	0.000447501358248 Ether ($0.51)
ⓘ Gas Price:	0.000000021309588488 Ether (21.309588488 Gwei)
ⓘ Ether Price:	$1,142.41 / ETH
ⓘ Gas Limit & Usage by Txn:	21,000 | 21,000 (100%)

"Gas" is the mechanism in Ethereum used to implement transaction fees

The execution of transactions "consumes" gas

When issued, the transactions "fills" a transaction with some gas

- Gas limit: the amount of gas (n. of units)
- Gas price: the price (in ETH) that the originator wants to pay for 1 unit of gas consumed

Gas used: the actual gas used by a node to process the transaction

- Can be less then the gas limit

The "fee" is simply the value in ETH of the gas used (n. units X gas price)

Transaction gas

Why a “variable” fee?

Why gas used may be less than the gas limit?

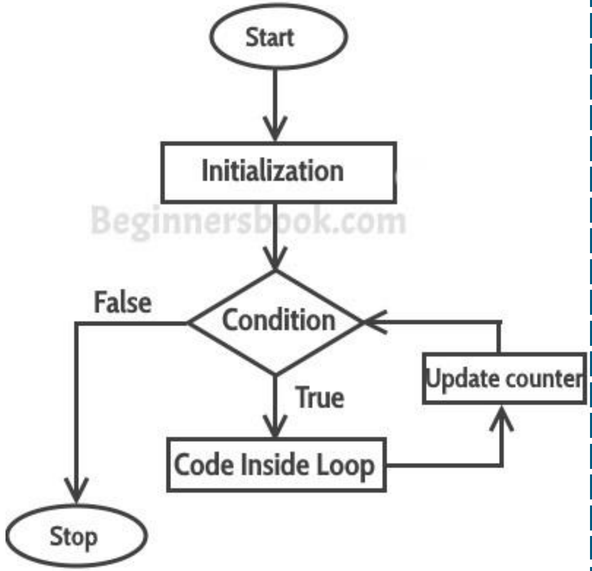
Gas and Smart Contract executions 1/2

A transaction can start the execution of a smart contract

Smart contract are written in "Turing complete" languages (they can express any computation, e.g., if-then-else, loops)

Processing a transaction invoking a smart contract may take more or less, depending on the execution path

(Transactions transferring ETH are simpler, no variability when they are processed)



Marco Comuzzi mcomuzzi@unistac.kr

Gas and Smart Contract executions 2/2

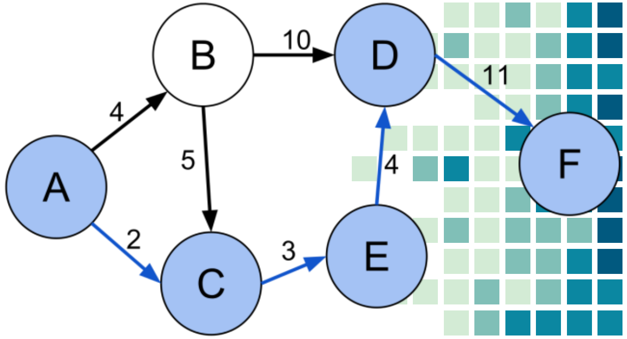
The gas mechanism allows variable fees, accounting for the **path** followed by the execution of a smart contract

The gas mechanism prevents that the smart contract execution gets stuck in "infinite" computation

When the gas limit is reached, the execution stops and an error is returned

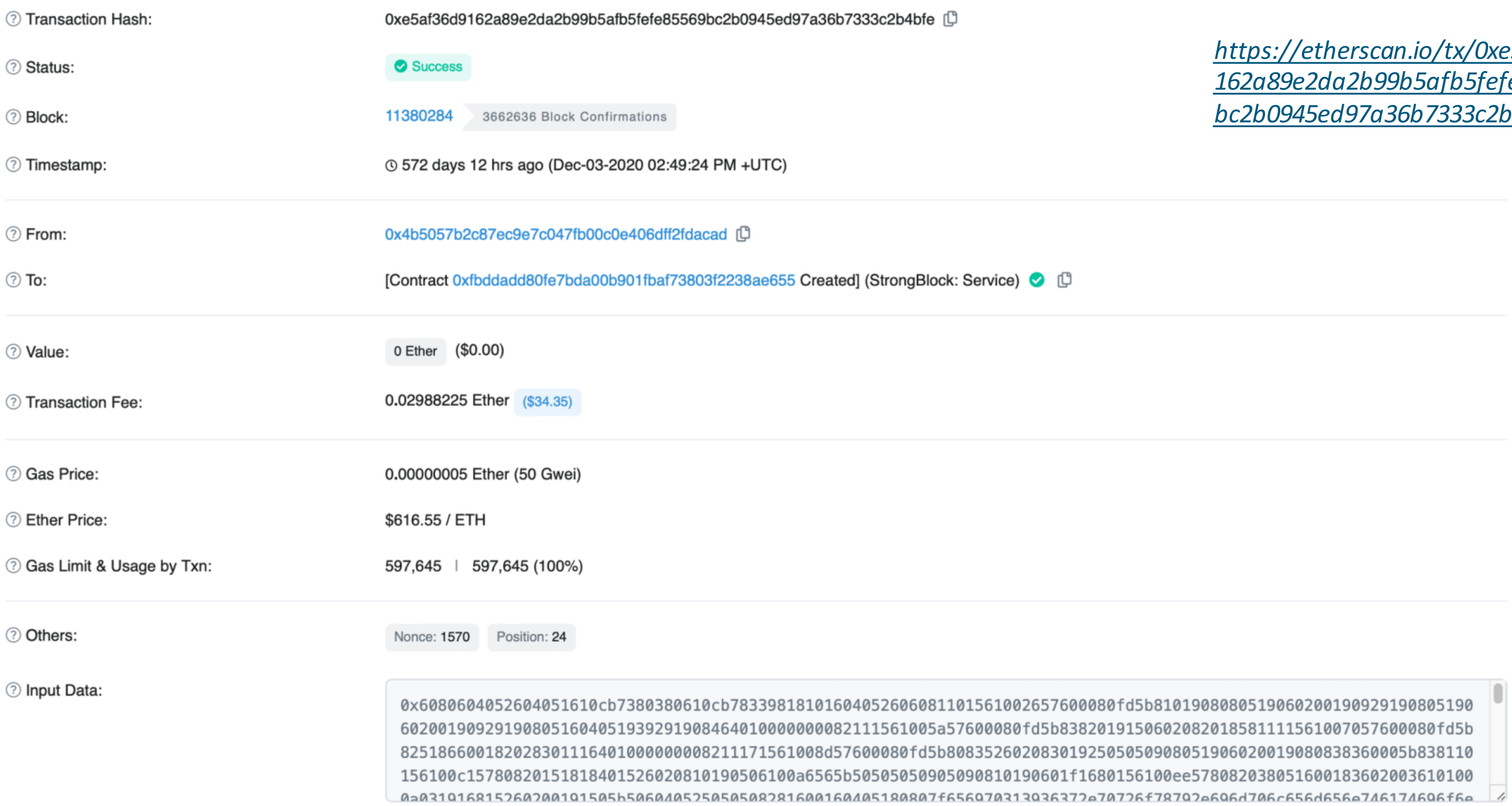
Ethereum clients can estimate the gas needed

Gas price can be improved to pay higher fees (and improve chance to be mined earlier)



Marco Comuzzi mcomuzzi@unistac.kr

Type 2: Deploying smart contracts



Transaction Hash:	0xe5af36d9162a89e2da2b99b5afb5fefe85569bc2b0945ed97a36b7333c2b4bfe
Status:	Success
Block:	11380284 3662636 Block Confirmations
Timestamp:	572 days 12 hrs ago (Dec-03-2020 02:49:24 PM +UTC)
From:	0x4b5057b2c87ec9e7c047fb00c0e406dff2fdacad
To:	[Contract 0xfbddadd80fe7bda00b901fbaf73803f2238ae655 Created] (StrongBlock: Service)
Value:	0 Ether ($0.00)
Transaction Fee:	0.02988225 Ether ($34.35)
Gas Price:	0.00000005 Ether (50 Gwei)
Ether Price:	$616.55 / ETH
Gas Limit & Usage by Txn:	597,645 | 597,645 (100%)
Others:	Nonce: 1570 Position: 24
Input Data:	

```
0x608060405260405161 0cb7380380610cb78339818101604052606081101561002657600080fd5b810190808051906020019092919080519 0
```

0x6080604052604051610cb7380380610cb783398181016040526060811015610026576000 80fd5b8101908080519060200190929190805190
602001909291908051604051939291908464010000000082111561005a57600080fd5b838201915060208201858111156100705760 0080fd5b
825186600182028301116401000000008211171561008d57600080fd5b80835260208301925050509080519060200190808383600 05b838110
156100c157808201518184015260208101905061 00a6565b5050505090509081019060 1f168015610 0ee5780820380516001836020036101 00
0a031916815260200191505b50604052505050828160016040518080 7f6569703139363372e70726f78792e696d706c656d656e746174696f6e

https://etherscan.io/tx/0xe5af36d9162a89e2da2b99b5afb5fefe85569bc2b0945ed97a36b7333c2b4bfe

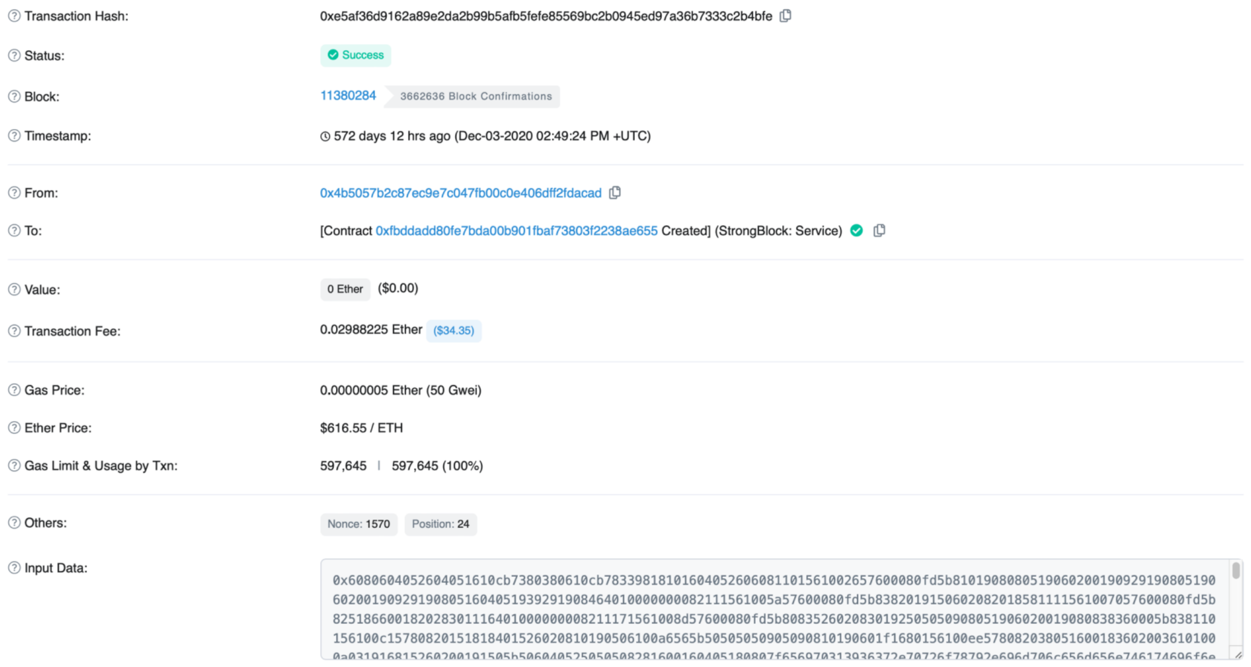
Transaction to deploy SCs

The recipient is the fictitious node “0”

The “input data” contain the byte code of the smart contract

The “value” of the transaction is the initial balance (in ETH) of the smart contract

The execution of this transaction creates a new Smart Contract Account

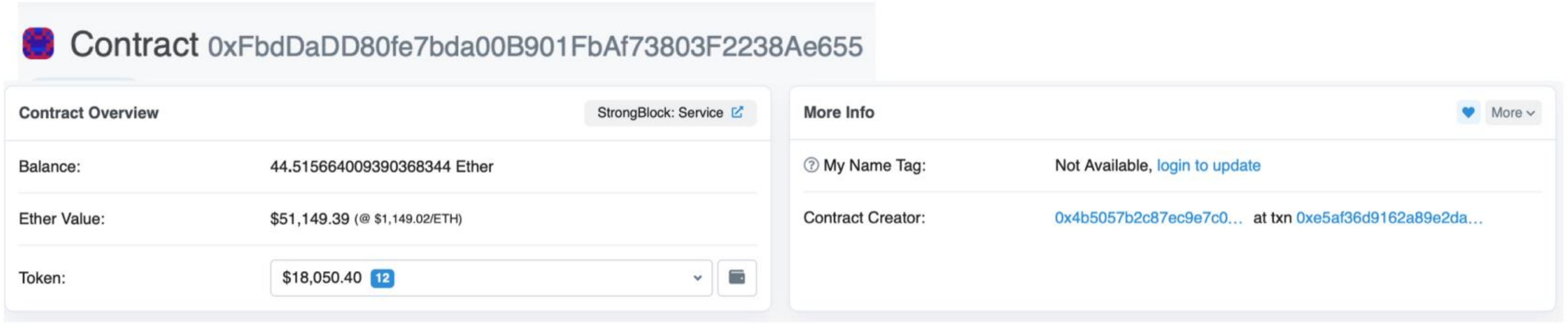


Transaction Hash:	0xe5af36d9162a89e2da2b99b5afb5fefe85569bc2b0945ed97a36b7333c2b4bfe
Status:	Success
Block:	11380284 3662636 Block Confirmations
Timestamp:	572 days 12 hrs ago (Dec-03-2020 02:49:24 PM +UTC)
From:	0x4b5057b2c87ec9e7c047fb00c0e406dff2fdacad
To:	[Contract 0xfbddadd80fe7bda00b901fbaf73803f2238ae655 Created] (StrongBlock: Service)
Value:	0 Ether ($0.00)
Transaction Fee:	0.02988225 Ether ($34.35)
Gas Price:	0.00000005 Ether (50 Gwei)
Ether Price:	$616.55 / ETH
Gas Limit & Usage by Txn:	597,645 | 597,645 (100%)
Others:	Nonce: 1570 Position: 24
Input Data:	

```
0x608060405260405161 0cb7380380610cb7833981810160405260608110156100265760008 0fd5b810190808051906020019092919080519 0
```

```
0x6080604052604051610cb7380380610cb78339818101604052606081101561002657600080fd5b81019080805190602001909291908051906020019092919080516040519392919084640100000000821115610 05a57600080fd5b838201915060208201858111156100705760008 0fd5b
```

https://etherscan.io/tx/0xe5af36d9162a89e2da2b99b5afb5fefe85569bc2b0945ed97a36b7333c2b4bfe



Contract 0xFbdDaDD80fe7bda00B901FbAf73803F2238Ae655

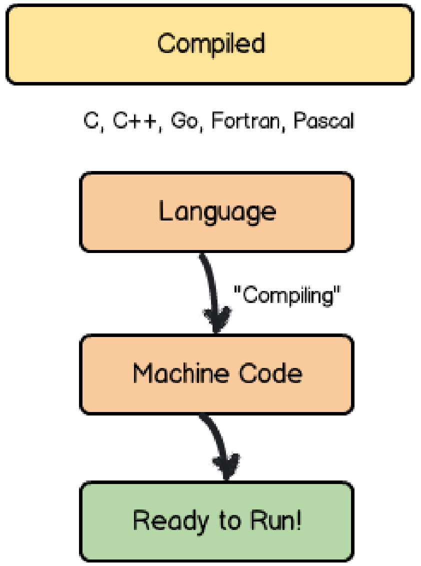
Contract Overview	StrongBlock: Service
Balance:	44.515664009390368344 Ether
Ether Value:	$51,149.39 (@ $1,149.02/ETH)
Token:	$18,050.40 12

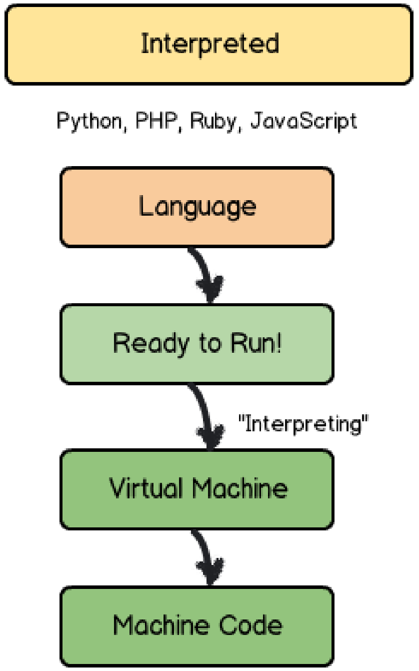
More Info	More
My Name Tag:	Not Available, login to update
Contract Creator:	0x4b5057b2c87ec9e7c0... at txn 0xe5af36d9162a89e2da...

What is the “byte code”?

Machine code is created for a specific execution environment (Macos or Windows)

Normally (much) faster than interpreted languages





The same code can run anywhere where there is an interpreter

(Syntax) errors may occur during execution

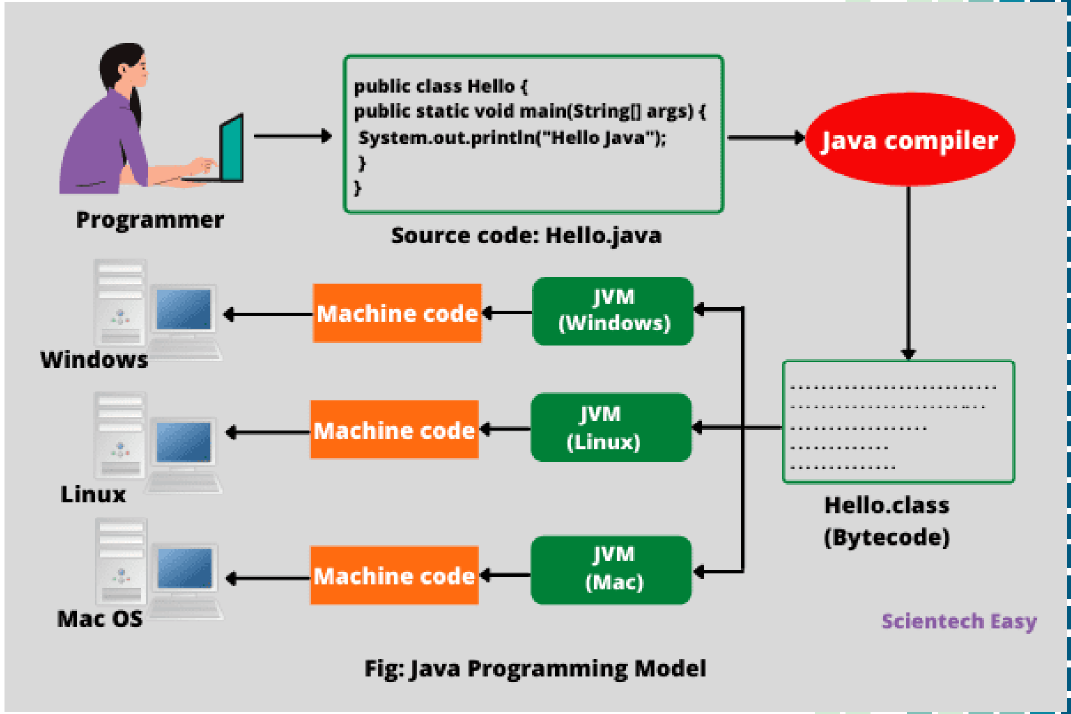
Slow

What is the "byte code"?

Interpreted + compiled (Java)

Source compiled into bytecode

Bytecode can execute everywhere there is a "Virtual Machine"



```
public class Hello {
public static void main(String[] args) {
 System.out.println("Hello Java");
 }
}
```

Fig: Java Programming Model

What is the “byte code”?

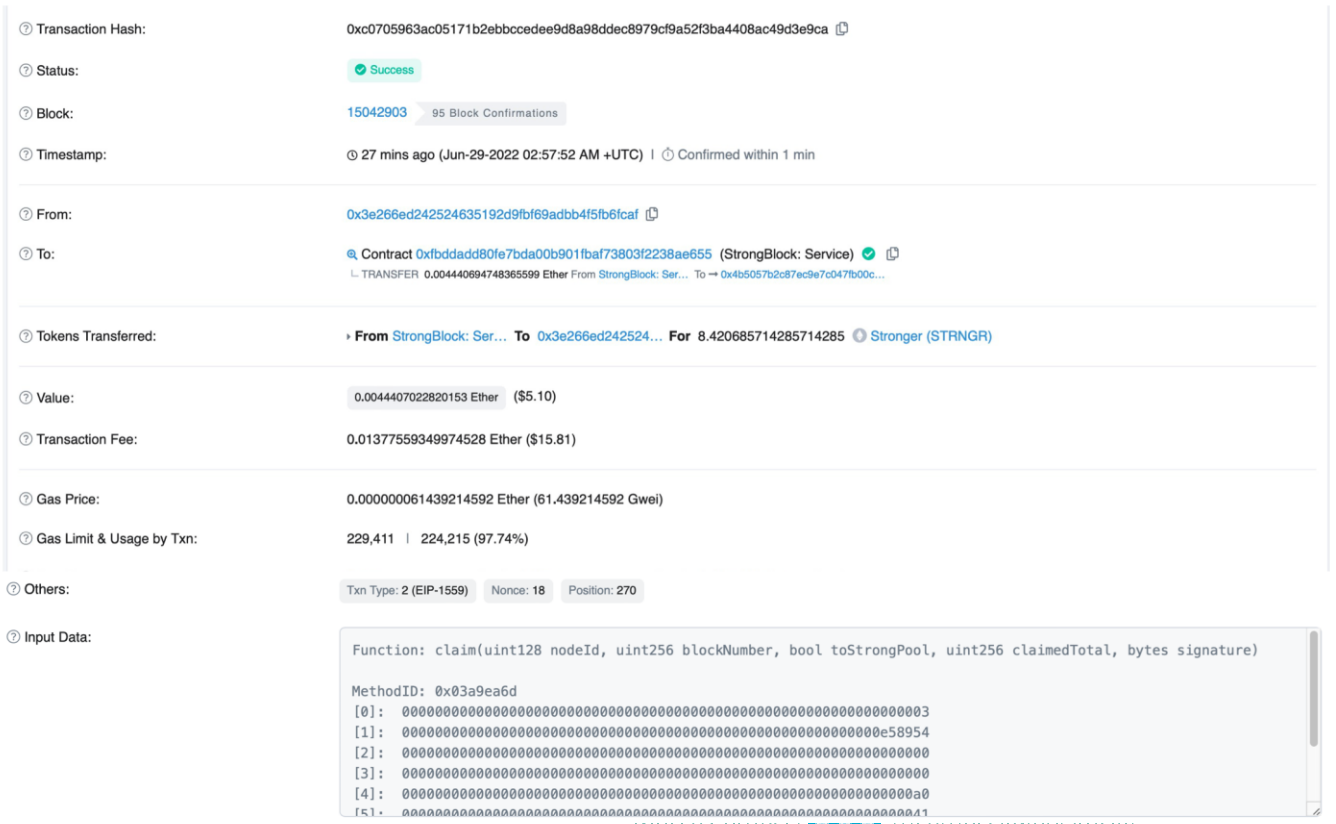
Ethereum Smart Contract languages (Solidity, Vyper, ...)

Compiled into Ethereum bytecode

Bytecode distributed via transactions (input data)

Bytecode can be executed by any node
(using the EVM – Ethereum Virtual Machine)

Type 3: Invoking smart contracts



Transaction Hash:	0xc0705963ac05171b2ebbccedee9d8a98ddec8979cf9a52f3ba4408ac49d3e9ca
Status:	Success
Block:	15042903 95 Block Confirmations
Timestamp:	27 mins ago (Jun-29-2022 02:57:52 AM +UTC) | Confirmed within 1 min
From:	0x3e266ed242524635192d9fbf69adbb4f5fb6fcaf
To:	Contract 0xfbddadd80fe7bda00b901fbaf73803f2238ae655 (StrongBlock: Service) ✓ └ TRANSFER 0.004440694748365599 Ether From StrongBlock: Ser... To → 0x4b5057b2c87ec9e7c047fb00c...
Tokens Transferred:	From StrongBlock: Ser... To 0x3e266ed242524... For 8.420685714285714285 Stronger (STRNGR)
Value:	0.0044407022820153 Ether ($5.10)
Transaction Fee:	0.01377559349974528 Ether ($15.81)
Gas Price:	0.000000061439214592 Ether (61.439214592 Gwei)
Gas Limit & Usage by Txn:	229,411 | 224,215 (97.74%)
Others:	Txn Type: 2 (EIP-1559) Nonce: 18 Position: 270
Input Data:	

```
Function: claim(uint128 nodeId, uint256 blockNumber, bool toStrongPool, uint256 claimedTotal, bytes signature)

MethodID: 0x03a9ea6d
[0]:  0000000000000000000000000000000000000000000000000000000000000003
[1]:  0000000000000000000000000000000000000000000000000000000000e58954
[2]:  0000000000000000000000000000000000000000000000000000000000000000
[3]:  0000000000000000000000000000000000000000000000000000000000000000
[4]:  00000000000000000000000000000000000000000000000000000000000000a0
[5]:  0000000000000000000000000000000000000000000000000000000000000041
```

https://etherscan.io/tx/0xc0705963ac05171b2ebbccedee9d8a98ddec8979cf9a52f3ba4408ac49d3e9ca

Transactions invoking smart contracts

The recipient is a smart contract account

The "input" data contain invocation parameters

(function + parameters)

"Value" > 0 can be used to transfer ETH to the contract
(when invoking a function "payable").

THANKS!

https://sites.google.com/site/marcocomuzziphd

http://iel.unist.ac.kr/

You can find me at:

@dr_bsad

mcomuzzi@unist.ac.kr