

# PD-2.0.0 First smart contract

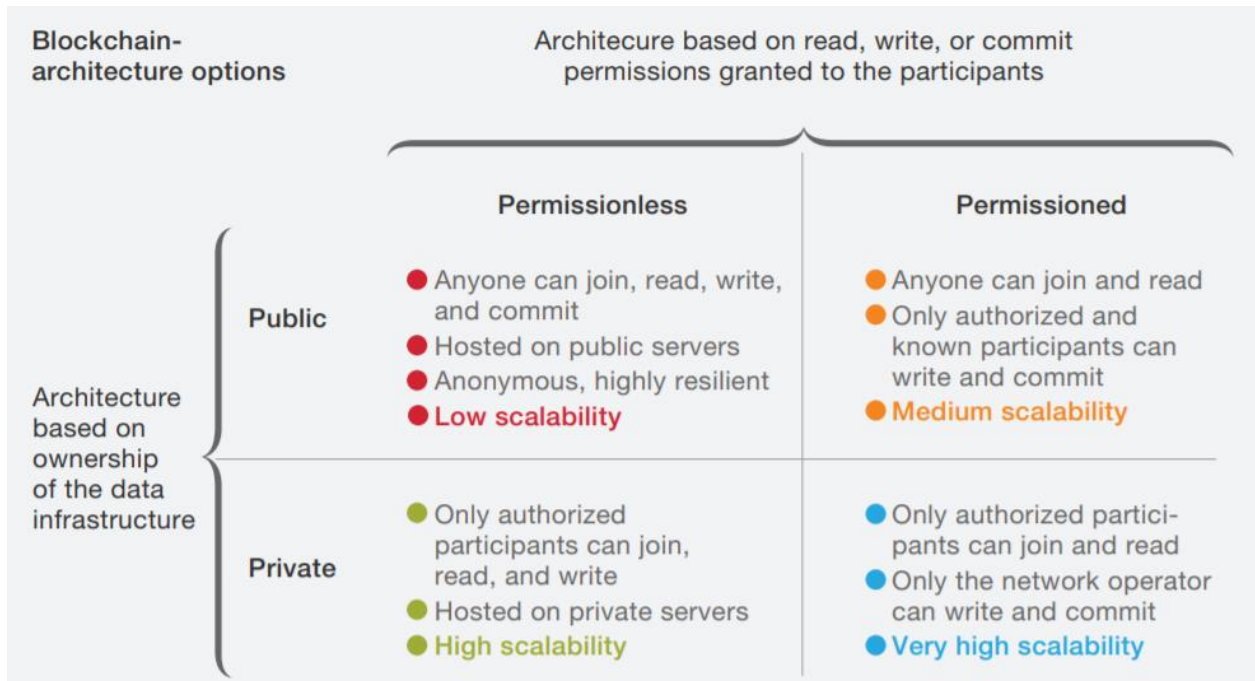
- PD-2.1 Architecture
- PD-2.2 Reading material
- PD-2.3 Prepare to use the play editor
- PD-2.4 Solidity

# PD-2.1.0 Architecture

- PD-2.1.1 Types of blockchains
- PD-2.1.2 Characteristics of blockchains
- PD-2.1.3 First generation blockchains
- PD-2.1.4 Second generation blockchains
- PD-2.1.5 Third generation blockchains
- PD-2.1.6 Objects and interactions
- PD-2.1.7 DAPP architecture

# PD-2.1.1 Types of blockchains

	Public	Private Permissioned	Public (read) Permissioned (write)
1 <sup>st</sup> gen	Bitcoin		
2 <sup>nd</sup> gen Smart contracts	Ethereum	Ethereum enterprise Hyperledger	Hyperledger Fabric
3 <sup>rd</sup> gen Multichain & upgradability	Polkadot Ethereum 2.0 Avalanche		



# PD-2.1.1 Different blockchains



<https://libra.org>



**HYPERLEDGER**

<https://www.hyperledger.org/resources/blockchain-showcase>



<https://marketplace.r3.com/solutions>



**HYPERLEDGER  
BESU**

<https://www.hyperledger.org/projects/besu>



**ENTERPRISE  
ETHEREUM  
ALLIANCE**

<https://entethalliance.org/publications/>



**Ava  
Labs.**

**AVALANCHE**

P-CHAIN

Coordinates validators  
Snowman Blockchain  
Create Subnets

X-CHAIN

Avalanche DAG  
Create Assets  
Exchange Assets

C-CHAIN

Executes EVM contracts  
Snowman Blockchain  
Ethereum RPCs

Ethereum 1



<https://www.stateofthedapps.com/>



<https://consensus.net/quorum>

<https://www.avalabs.org>

**EVM Pallet**

<https://substrate.dev/docs/en/knowledgebase/smart-contracts/evm-pallet>

*Polkadot.*

<https://polkadot.network>

<https://mathdapp.store/>



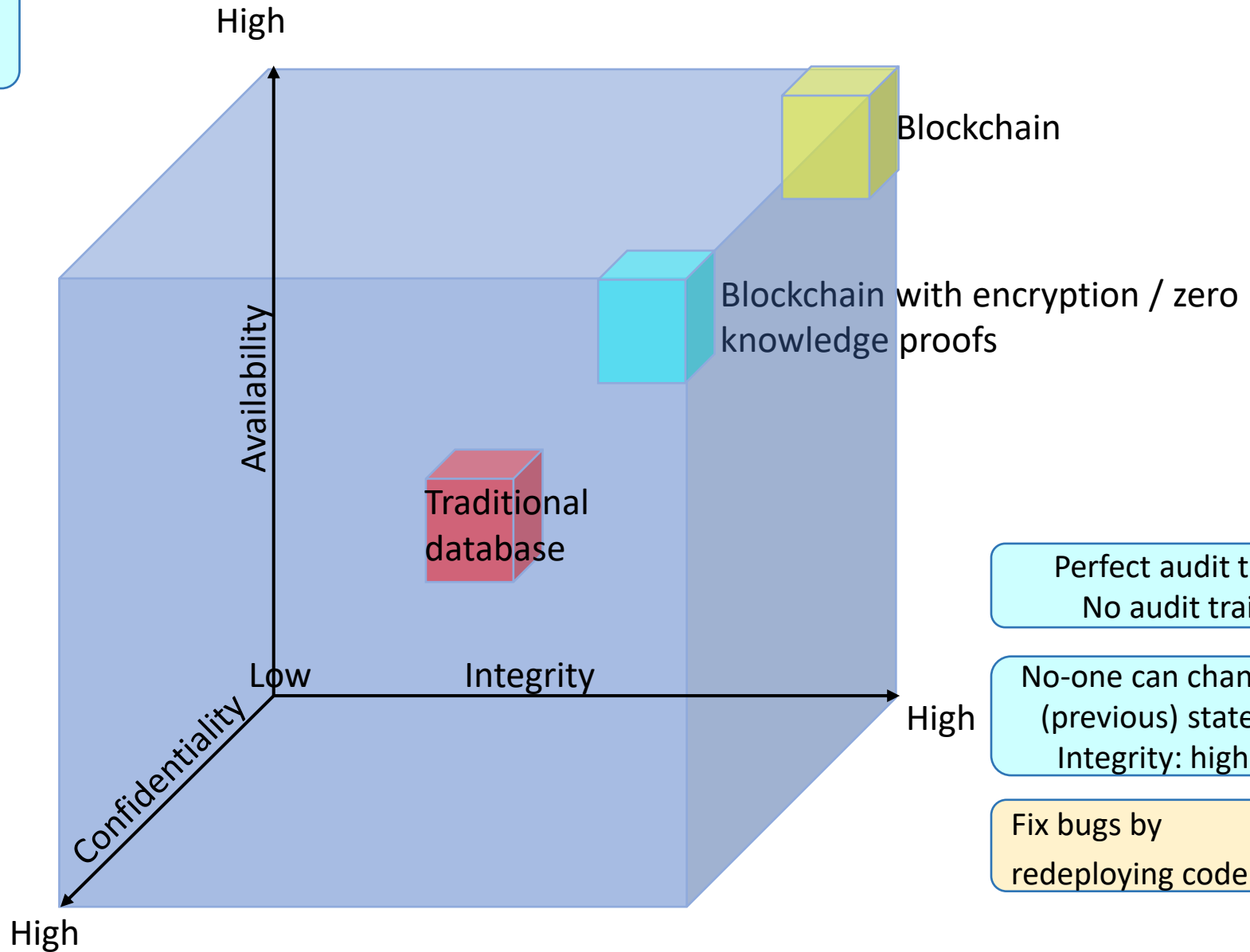
<https://www.lopp.net/bitcoin-information/data-anchor.html>

# PD-2.1.1 Why Ethereum?

- Permissionless & open
- Most used chain
- Reference chain (everyone compares to Ethereum 1)
- Other chain incorporate EVM or connect
- Network effects / money lego
  
- Disadvantage: high fees
  - Accelerates layer 2 solutions

# PD-2.1.2 Characteristics of blockchains

Very distributed database  
Availability: high



Blockchain

Blockchain with encryption / zero knowledge proofs

Traditional database

Perfect audit trail of writes  
No audit trail for reads

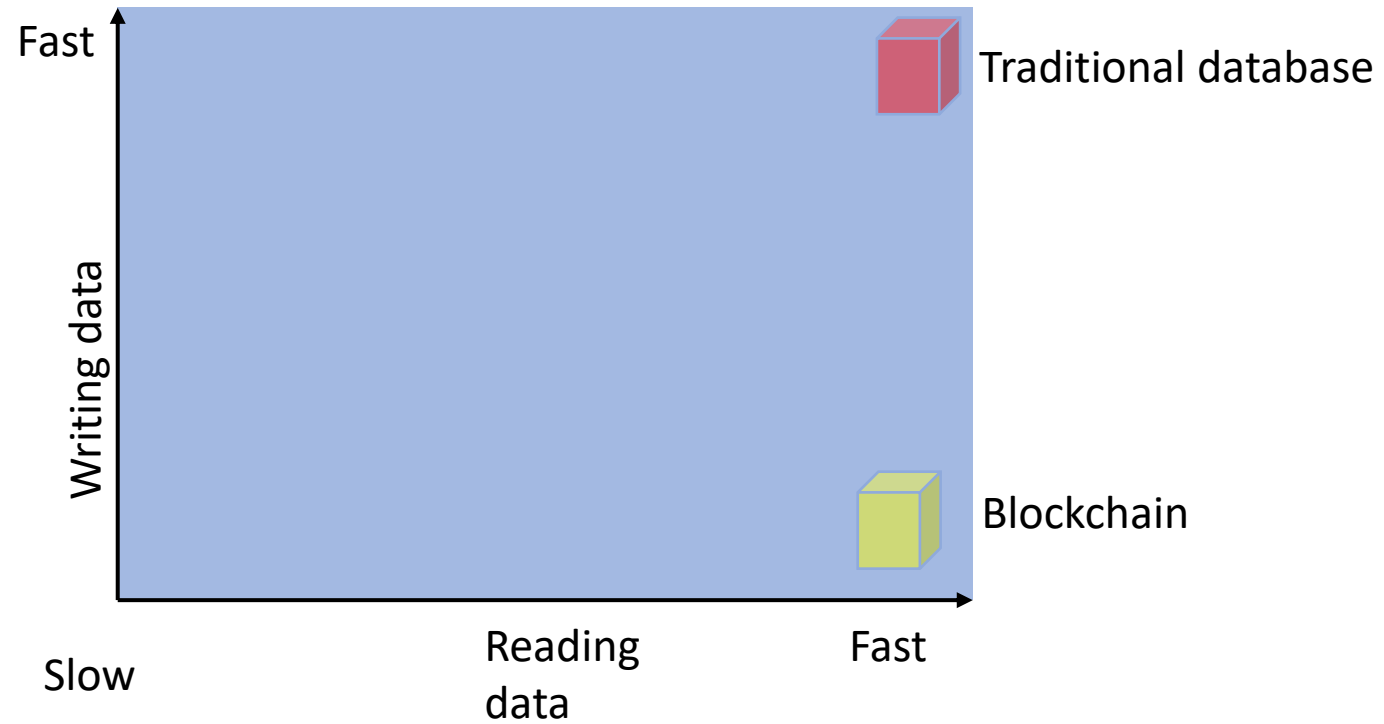
No-one can change (previous) state  
Integrity: high

Fix bugs by redeploying code

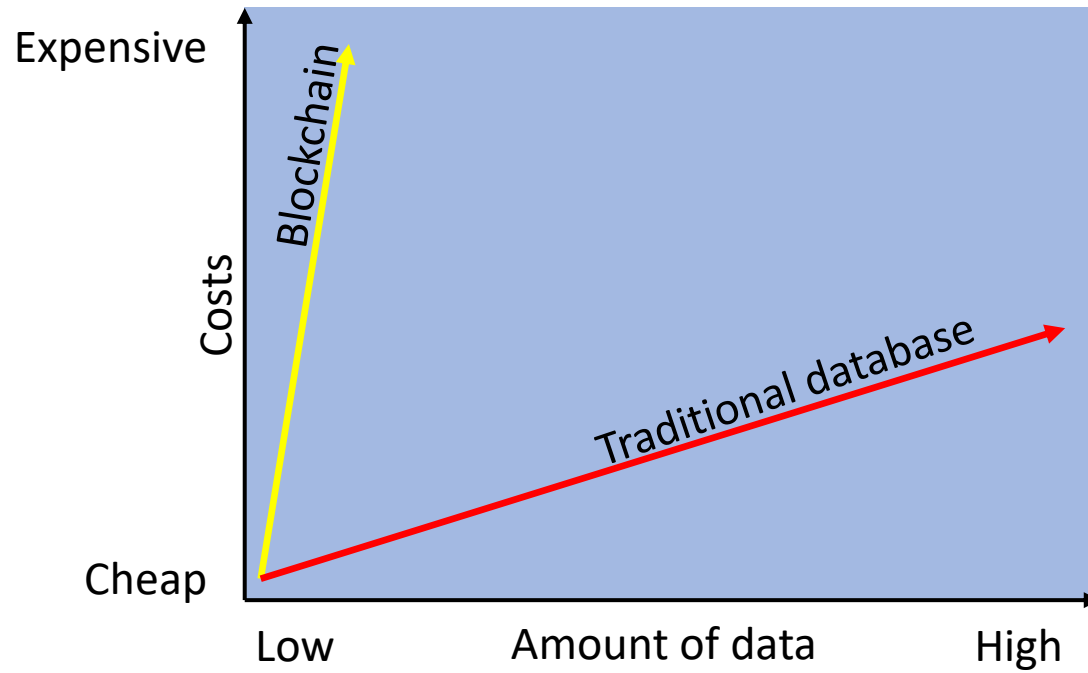
Anyone can read (everything)  
Confidentiality : low

Modules are re-used (also in unexpected ways)

# PD-2.1.2 Performance blockchain



# PD-2.1.2 Amounts of data vs Costs





# PD-2.1.2 Other characteristics

Money/value involved => bugs quickly security bugs

Anyone can deploy code

Anyone can interact with smart contract

Anyone can copy the entire chain (fork)

Key management important

Shared infrastructure & pay for usage

Hindrance by others

Denial of service possible (gas)

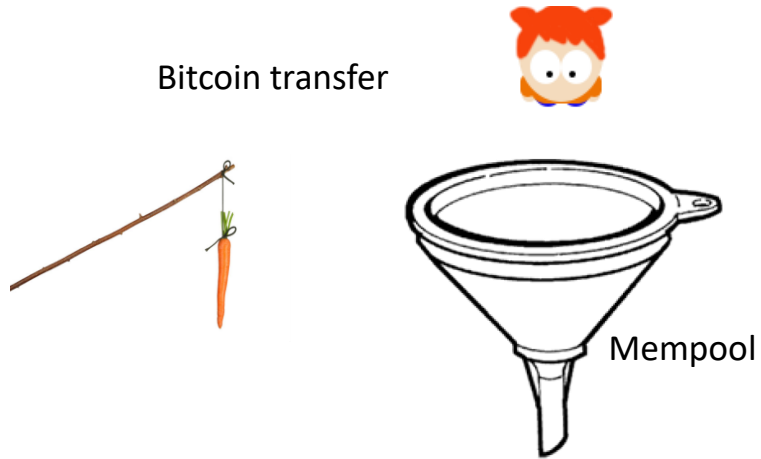
Immature, complex, changing rapidly => lots of bugs

Need to trust cryptography, protocols, software, network, incentive mechanisms

Trust is gained by proving all possible interactions

Incentive design important

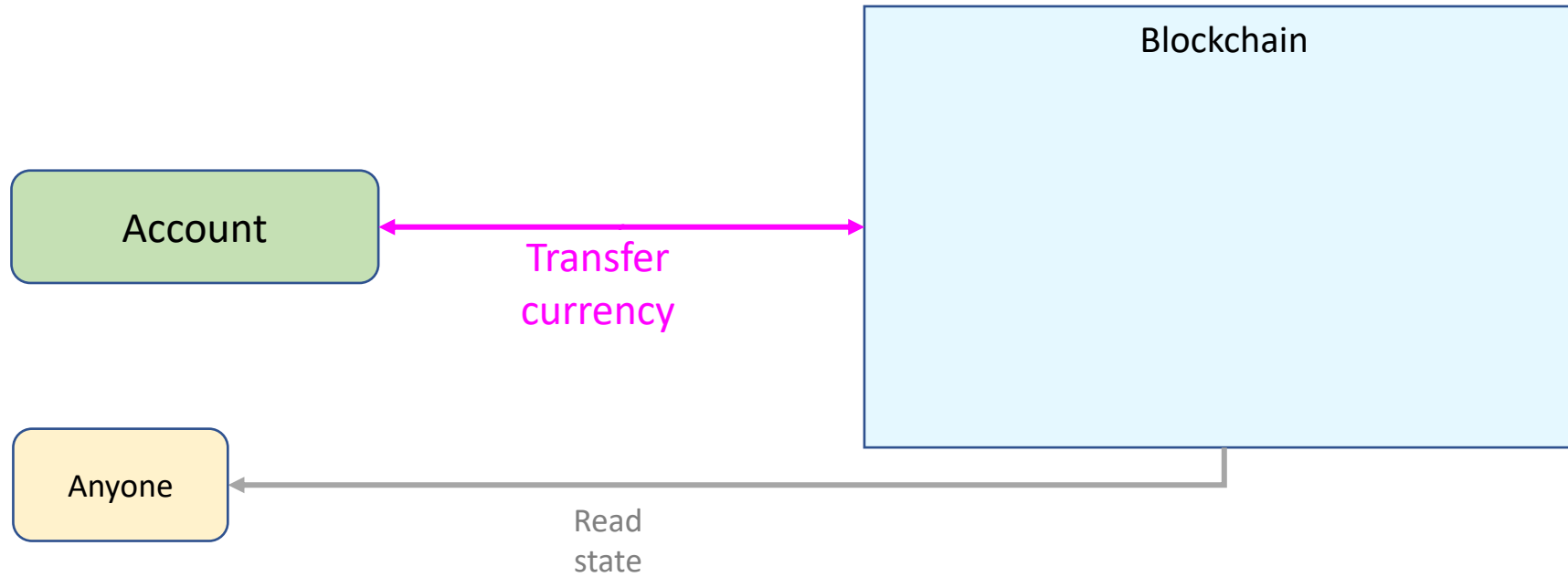
# PD-2.1.3 First generation blockchains



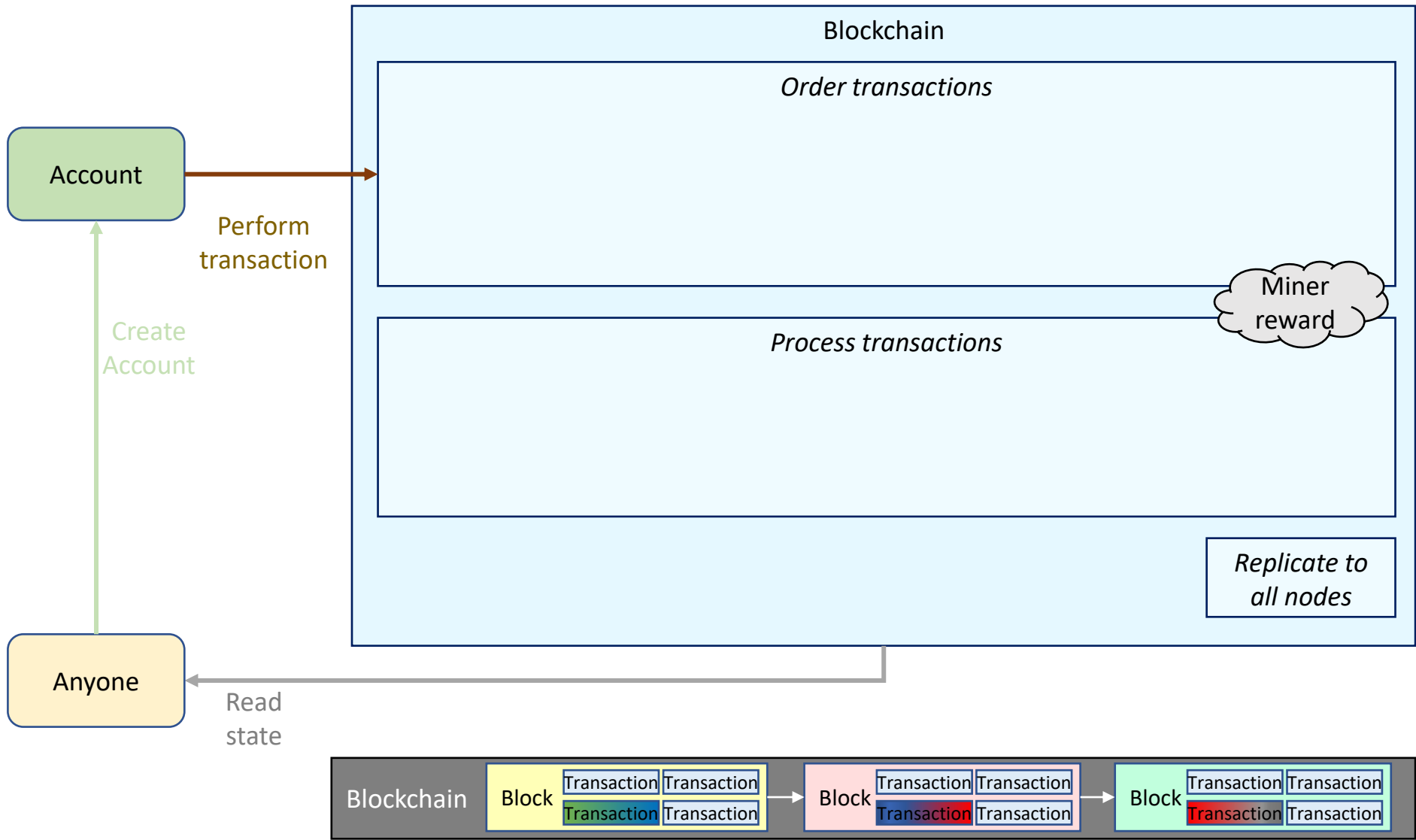
<https://txstreet.com/v/btc>

<https://dailyblockchain.github.io>

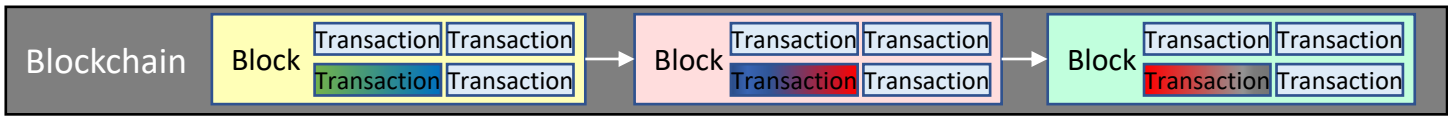
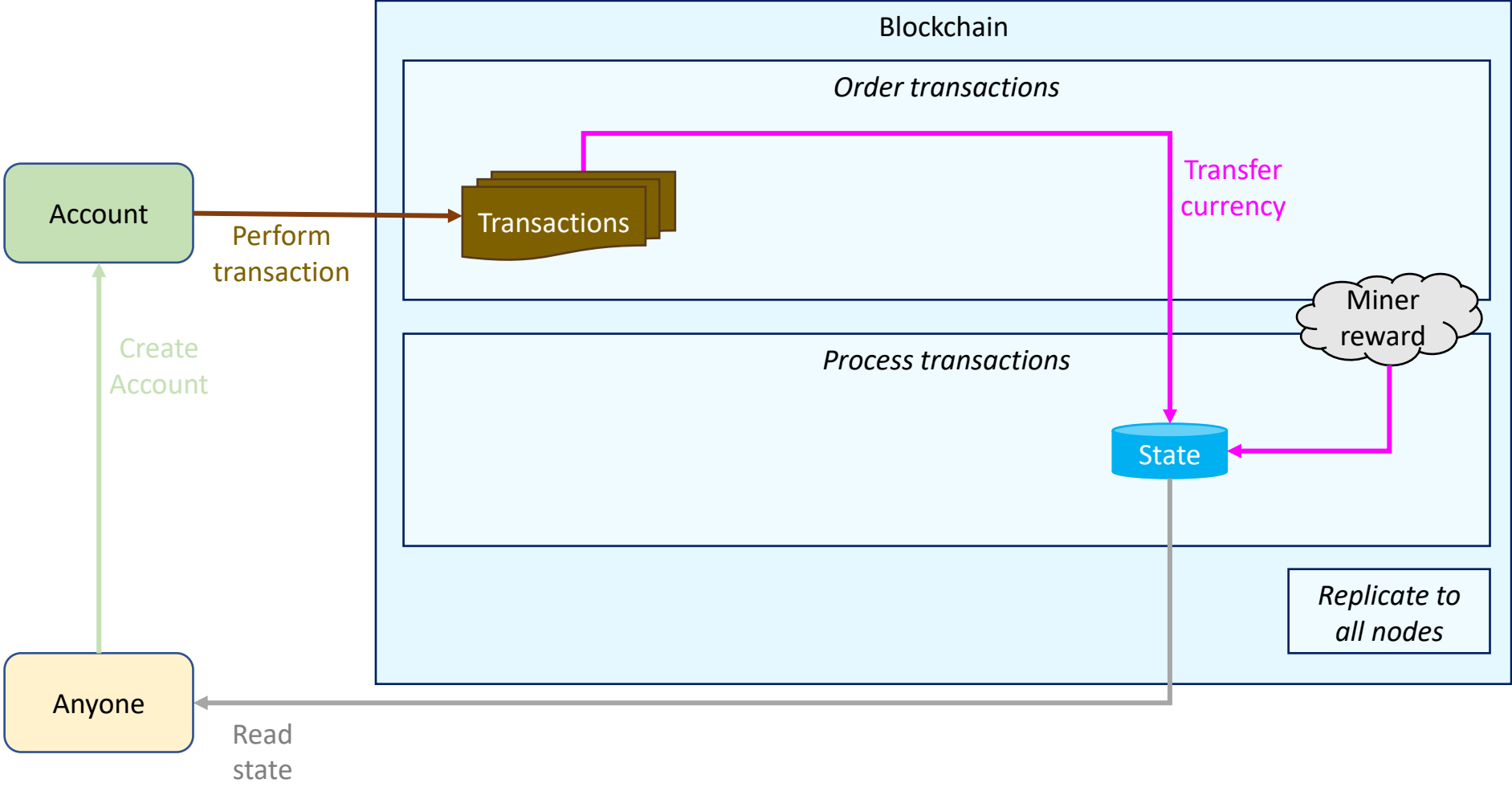
# PD-2.1.3 Black box 1<sup>st</sup> generation



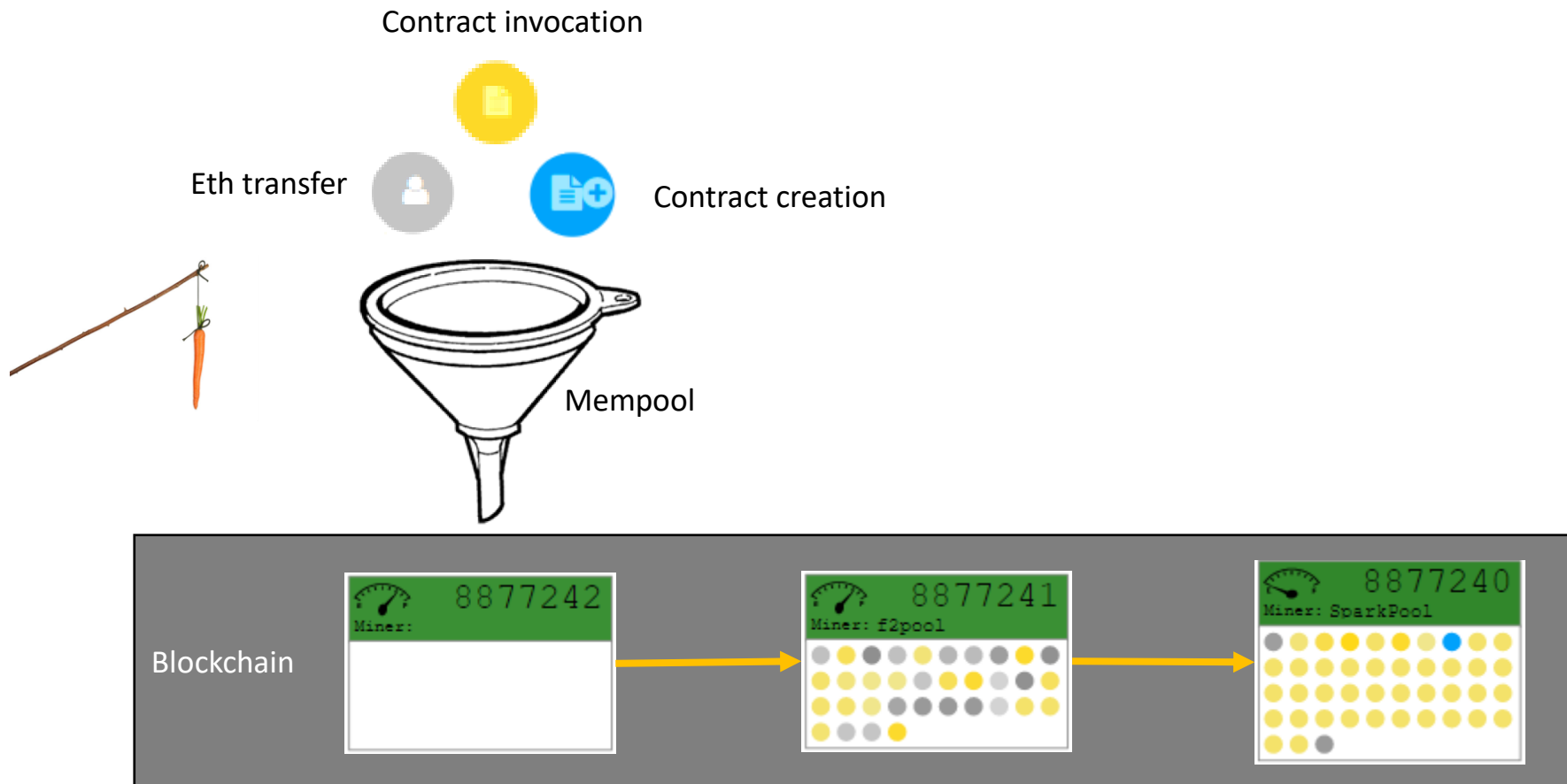
# PD-2.1.3 Architecture 1<sup>st</sup> generation



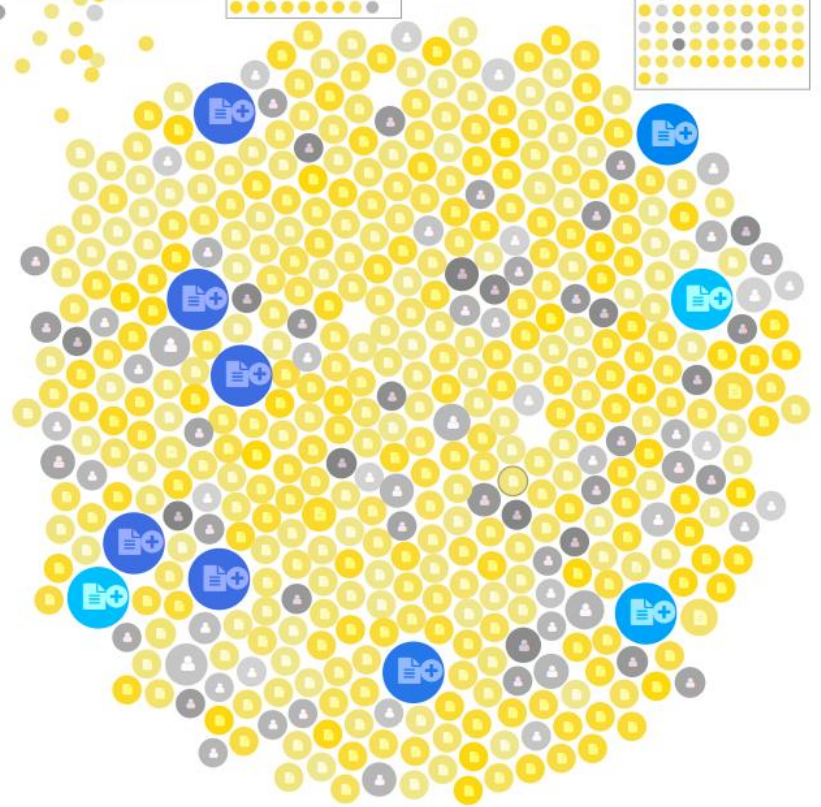
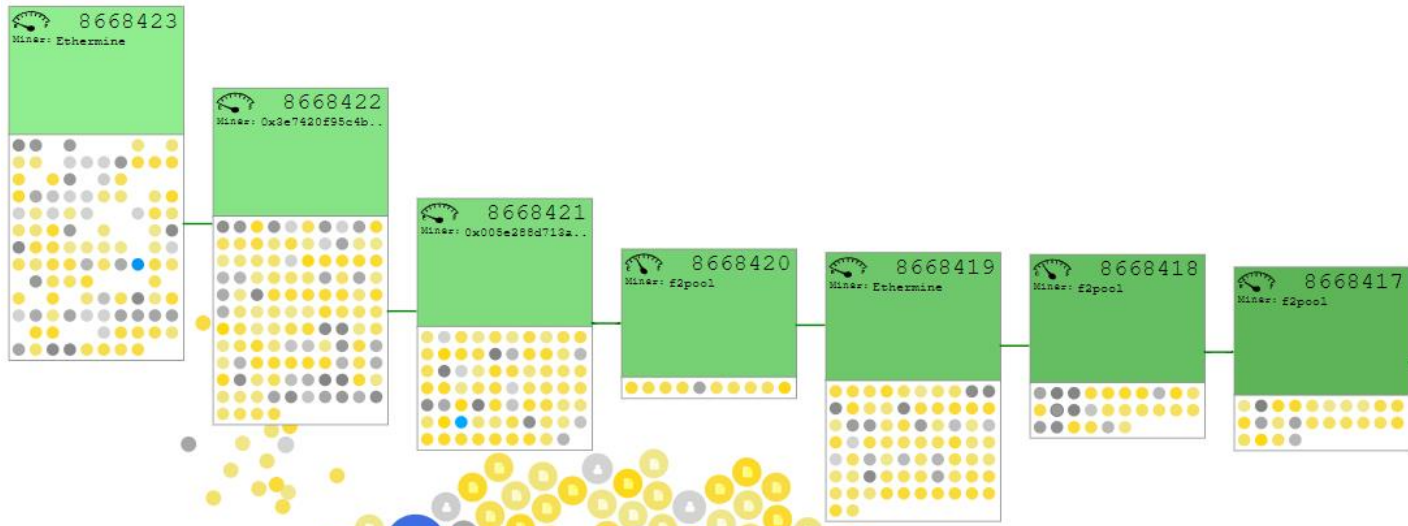
# PD-2.1.3 Architecture 1<sup>st</sup> generation



# PD-2.1.4 Second generation blockchains

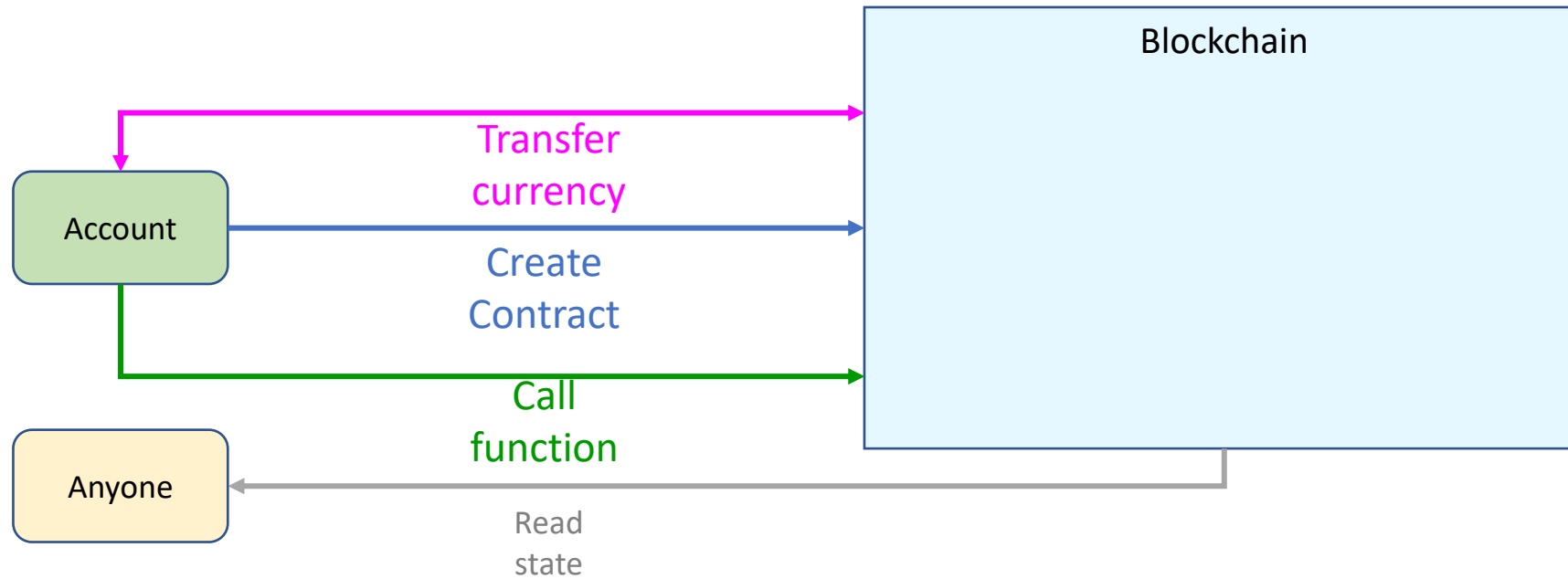


# PD-2.1.4 EthViewer



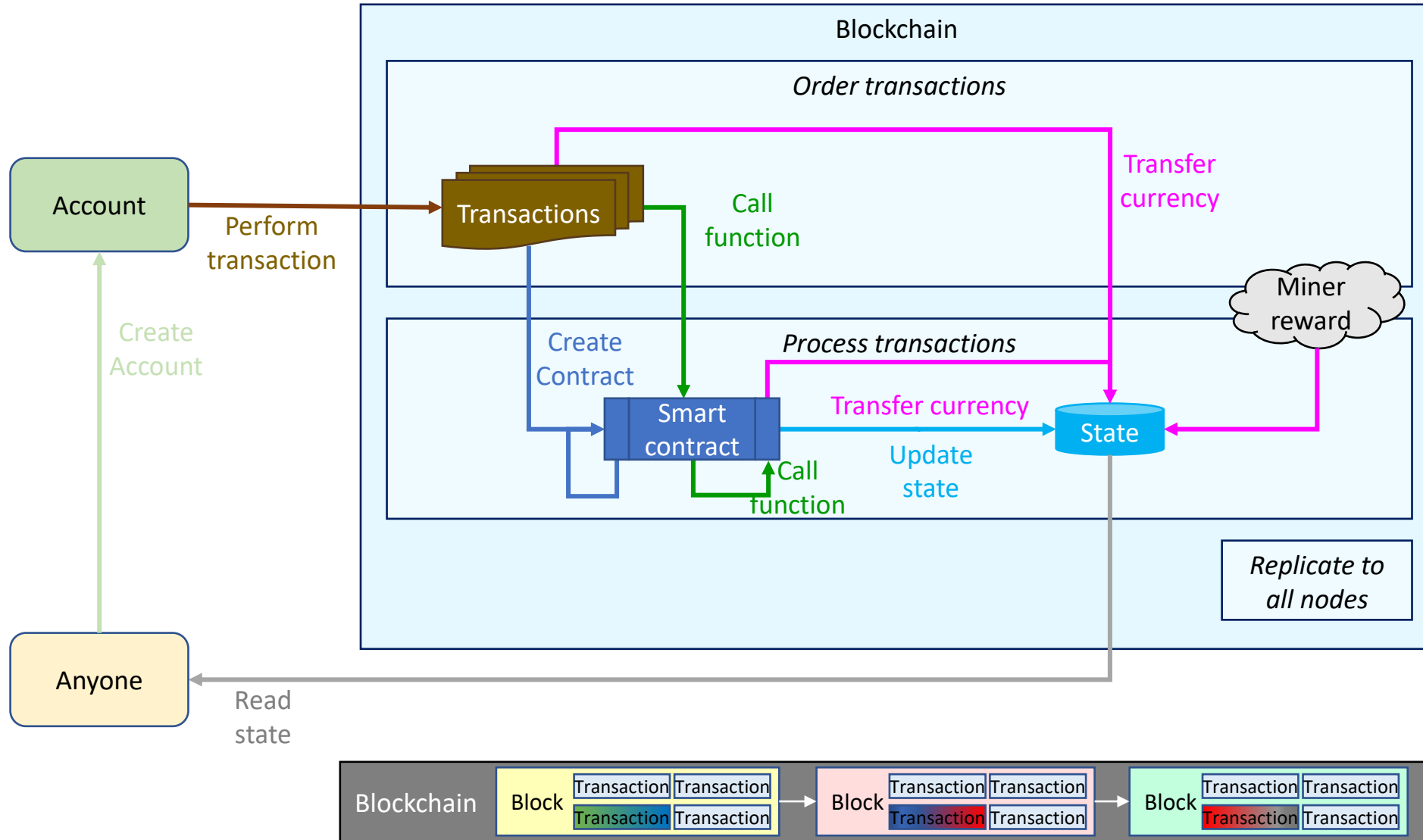
-  Contract invocation
-  Eth transfer
-  Contract creation

# PD-2.1.4 Black box 2<sup>nd</sup> generation

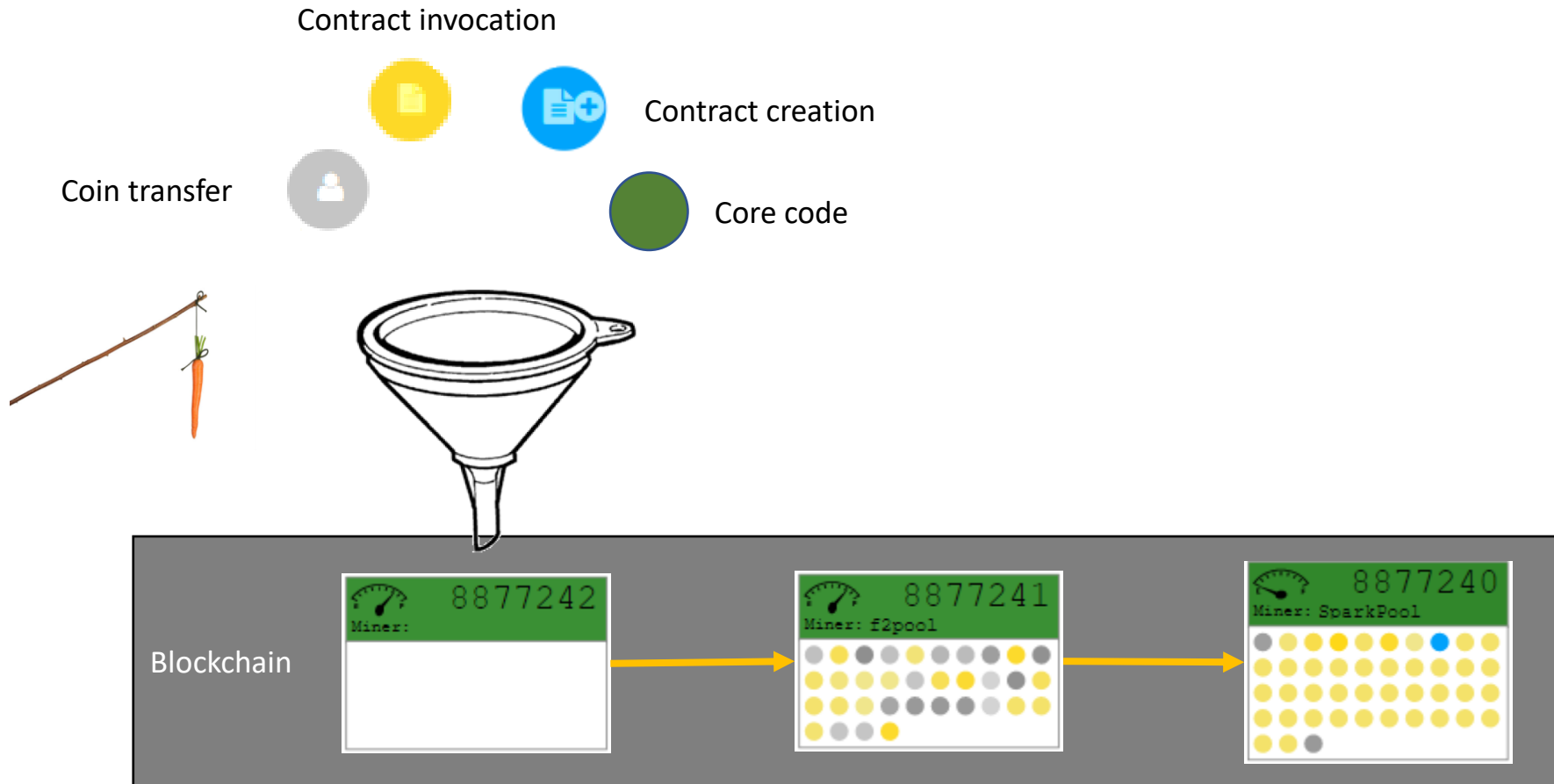




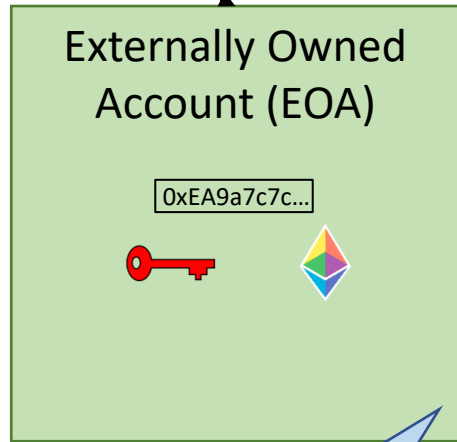
# PD-2.1.4 Architecture 2<sup>nd</sup> generation



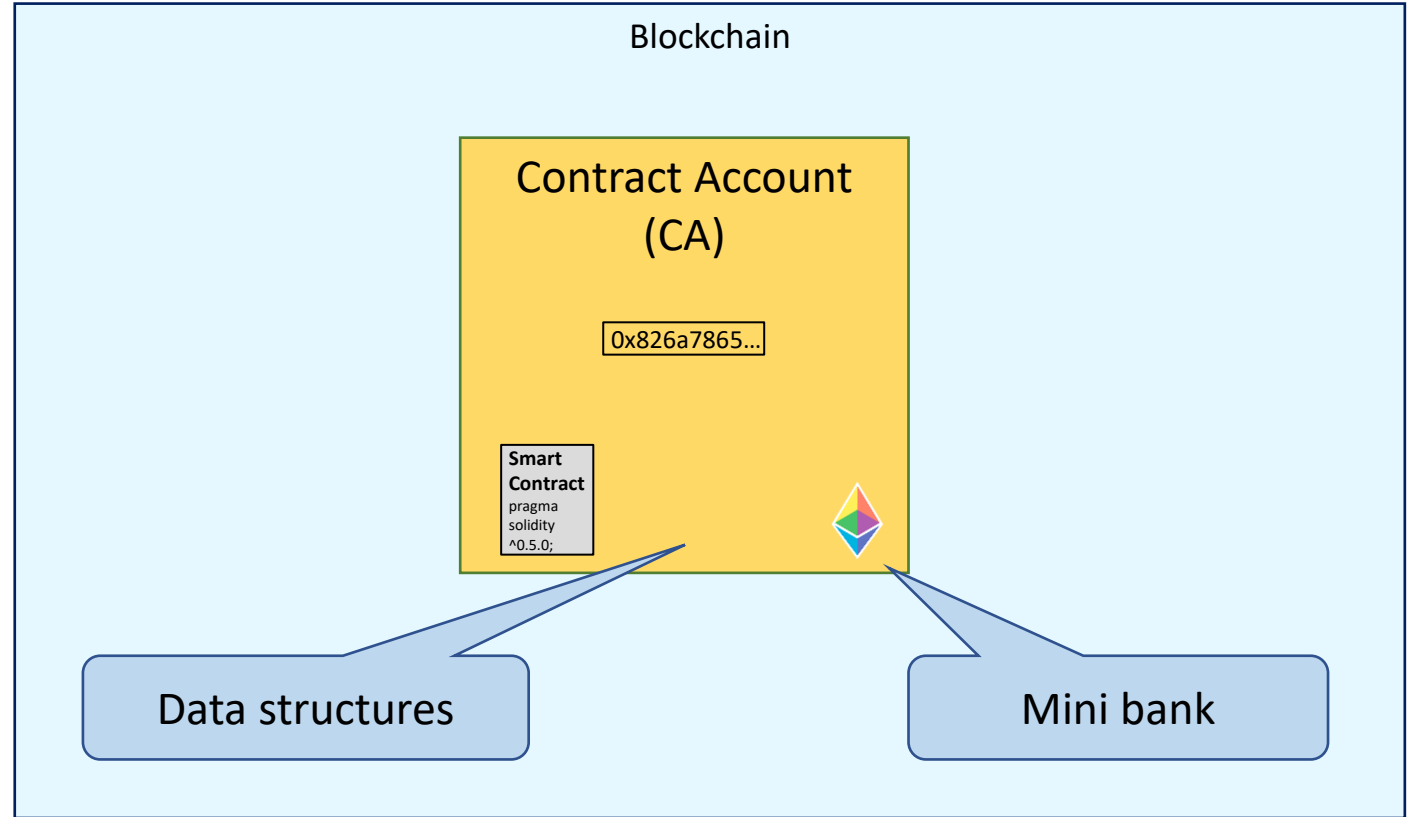
# PD-2.1.5 Third generation blockchains



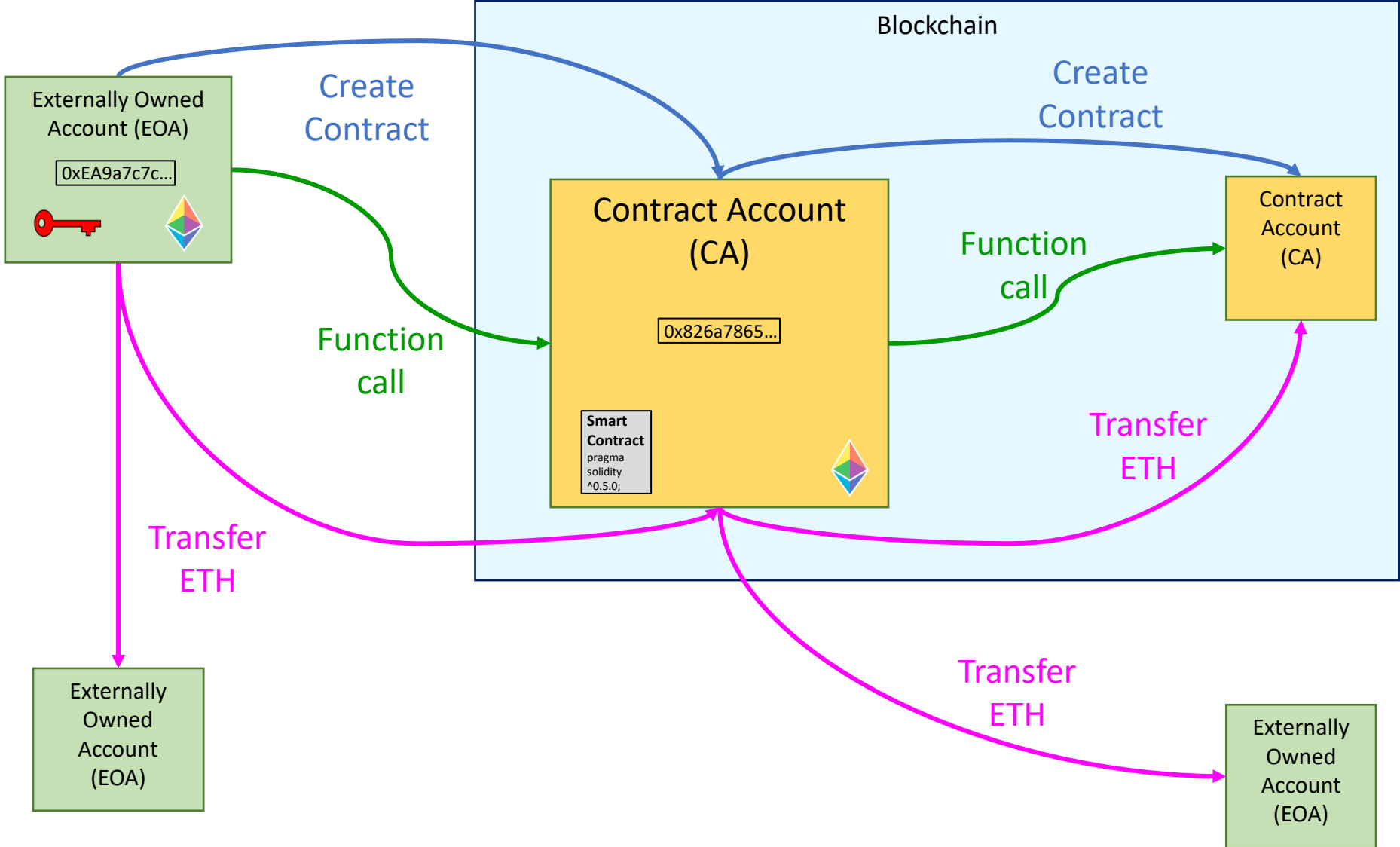
# PD-2.1.6 Objects and interactions



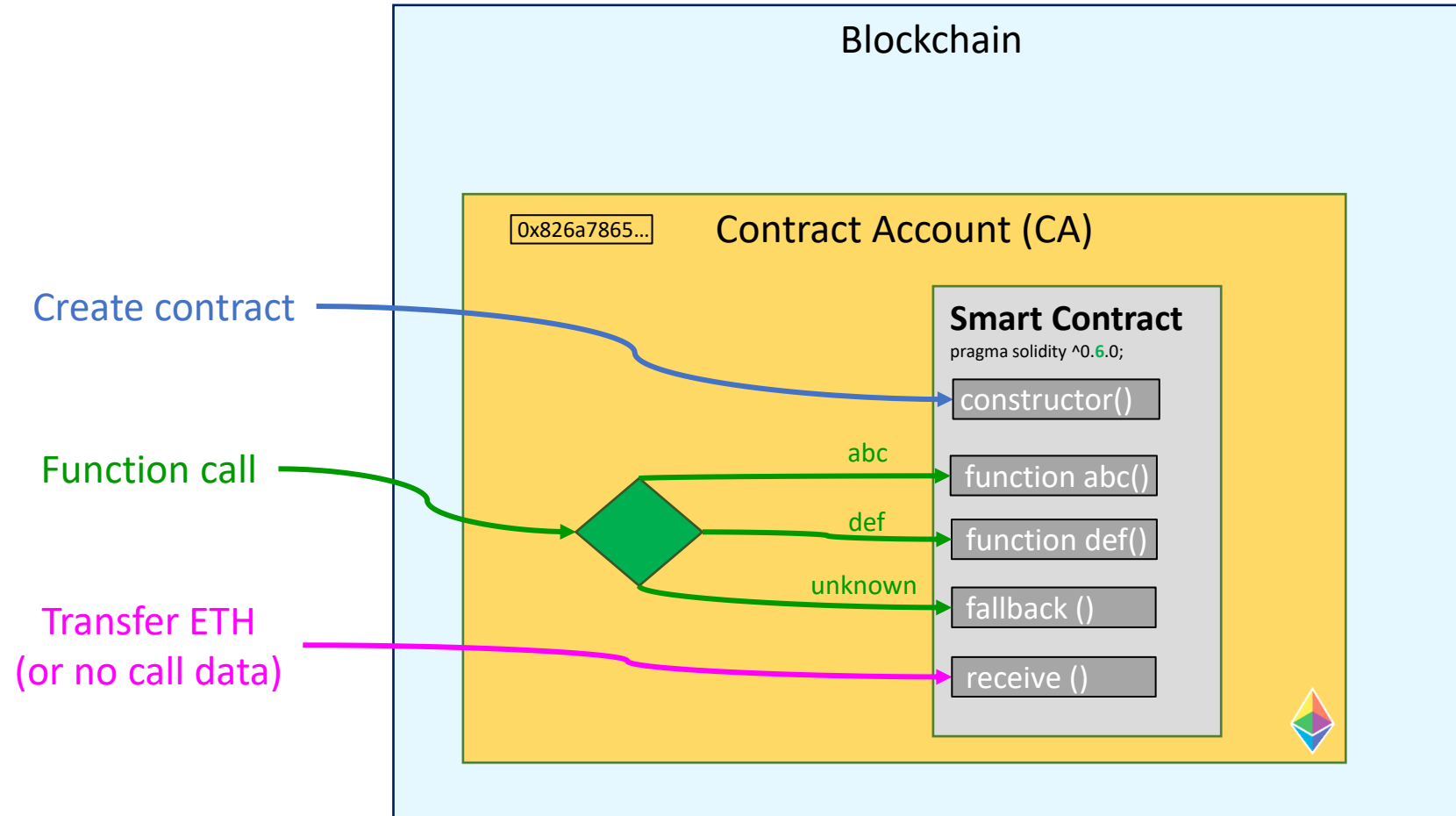
Bank balance



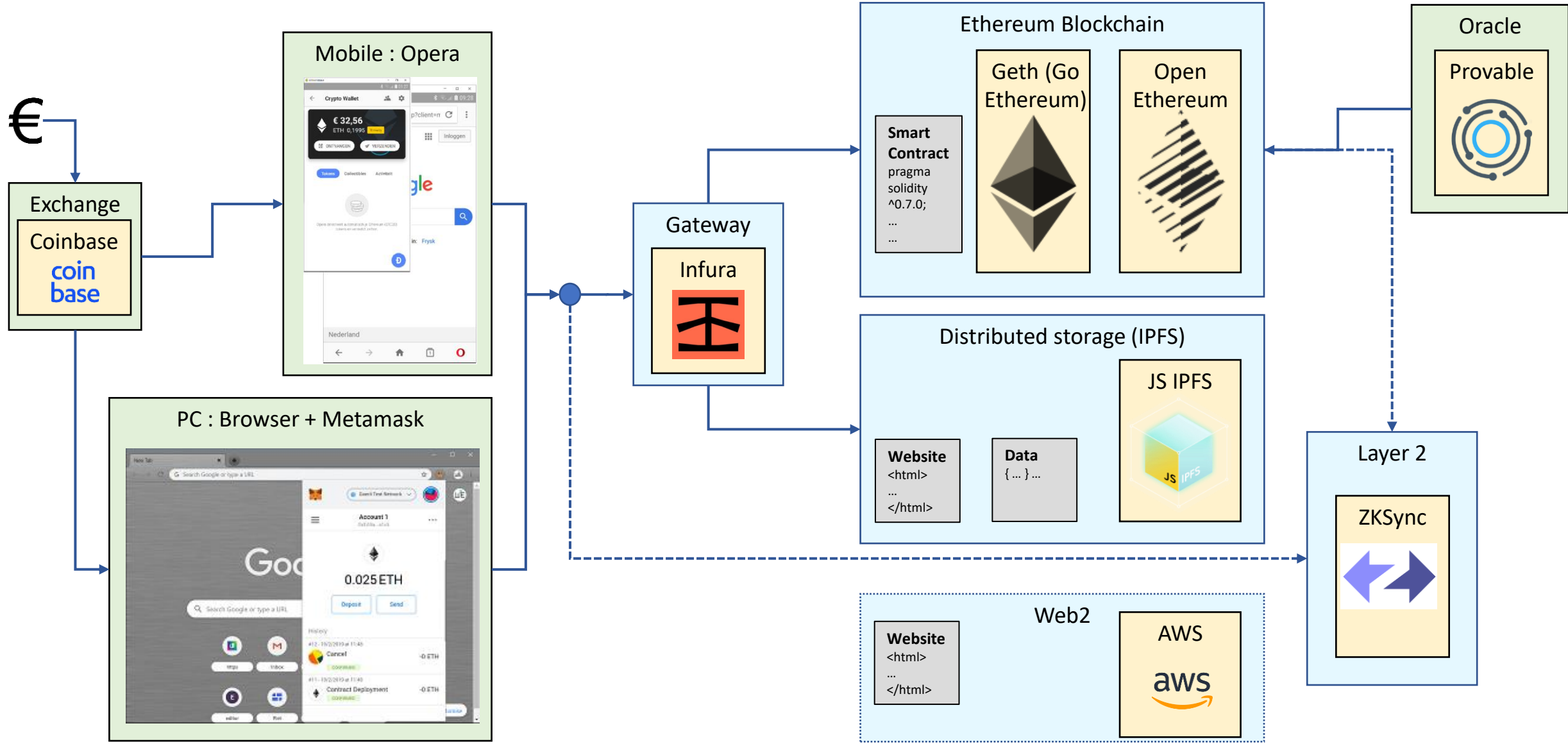
# PD-2.1.6 Interactions between addresses



# PD-2.1.6 Functions of a smart contract



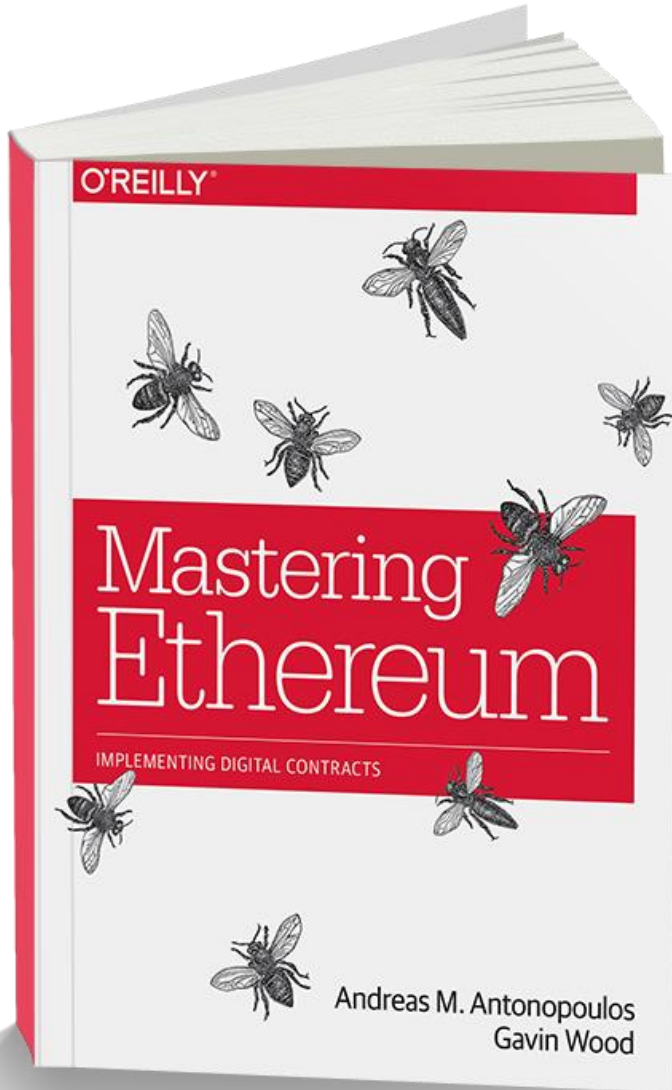
# PD-2.1.7 DAPP architecture



# PD-2.2 Reading material

- Mastering Ethereum (mandatory)
- Crypto Zombies (mandatory)
- Solidity manual (mandatory)
- Ethereum.org
- EthHub
- Consensys developer portal

# PD-2.2.1 Mastering Ethereum



<https://ethereumbook.info>

<https://github.com/ethereumbook/ethereumbook/blob/develop/01what-is.asciidoc>

<https://github.com/ethereumbook/ethereumbook/blob/develop/book.asciidoc>

<https://github.com/ethereumbook/ethereumbook/tree/develop/02intro.asciidoc>

<https://github.com/ethereumbook/ethereumbook/tree/develop/03clients.asciidoc>

<https://github.com/ethereumbook/ethereumbook/tree/develop/04keys-addresses.asciidoc>

<https://github.com/ethereumbook/ethereumbook/tree/develop/05wallets.asciidoc>

<https://github.com/ethereumbook/ethereumbook/tree/develop/06transactions.asciidoc>

<https://github.com/ethereumbook/ethereumbook/tree/develop/07smart-contracts-solidity.asciidoc>

<https://github.com/ethereumbook/ethereumbook/tree/develop/09smart-contracts-security.asciidoc>

<https://github.com/ethereumbook/ethereumbook/tree/develop/10tokens.asciidoc>

<https://github.com/ethereumbook/ethereumbook/tree/develop/11oracles.asciidoc>

<https://github.com/ethereumbook/ethereumbook/tree/develop/12dapps.asciidoc>

<https://github.com/ethereumbook/ethereumbook/blob/develop/13evm.asciidoc#turing-completeness-and-gas>

<https://github.com/ethereumbook/ethereumbook/tree/develop/14consensus.asciidoc>



# PD-2.2.2 Crypto Zombies



<https://cryptozombies.io/en/lesson/1/chapter/1>

<https://cryptozombies.io/en/lesson/2/chapter/1>

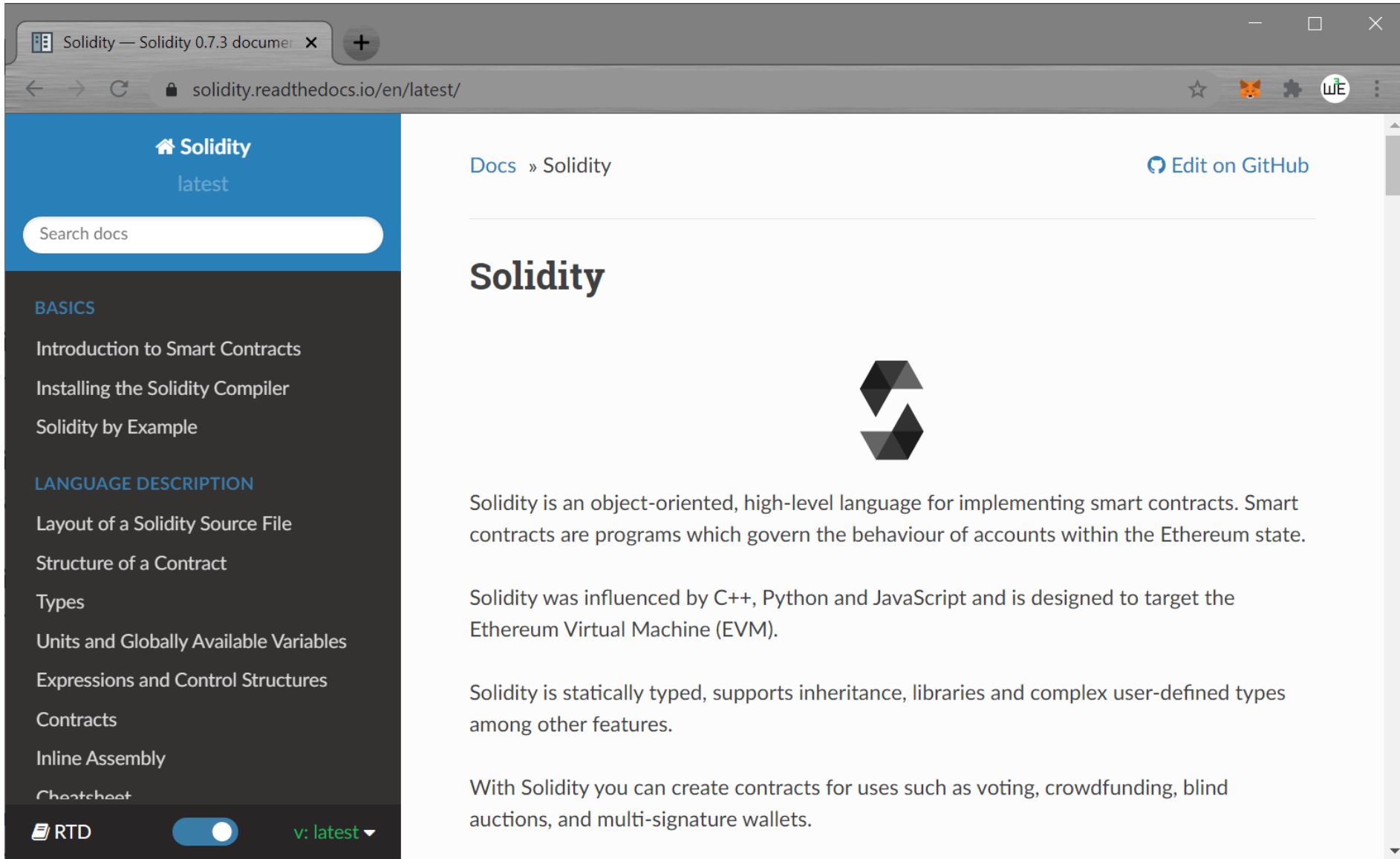
<https://cryptozombies.io/en/lesson/3/chapter/1>

<https://cryptozombies.io/en/lesson/4/chapter/1>

<https://cryptozombies.io/en/lesson/5/chapter/1>

<https://cryptozombies.io/en/lesson/6/chapter/1>

# PD-2.2.3 Solidity manual




The screenshot shows a web browser window displaying the Solidity 0.7.3 documentation page. The browser's address bar shows the URL `solidity.readthedocs.io/en/latest/`. The page has a dark blue sidebar on the left with a search bar and a navigation menu. The main content area is white and features the Solidity logo, a breadcrumb trail, and an "Edit on GitHub" link. The page title is "Solidity", and the content describes it as an object-oriented, high-level language for implementing smart contracts. The sidebar menu includes sections for "BASICS" and "LANGUAGE DESCRIPTION".

Solidity — Solidity 0.7.3 document

solidity.readthedocs.io/en/latest/

Docs » Solidity [Edit on GitHub](#)

## Solidity



Solidity is an object-oriented, high-level language for implementing smart contracts. Smart contracts are programs which govern the behaviour of accounts within the Ethereum state.

Solidity was influenced by C++, Python and JavaScript and is designed to target the Ethereum Virtual Machine (EVM).

Solidity is statically typed, supports inheritance, libraries and complex user-defined types among other features.

With Solidity you can create contracts for uses such as voting, crowdfunding, blind auctions, and multi-signature wallets.

<https://soliditylang.org>

<https://solidity.readthedocs.io/en/latest/>

Search docs

**Solidity**  
latest

**BASICS**

- Introduction to Smart Contracts
- Installing the Solidity Compiler
- Solidity by Example

**LANGUAGE DESCRIPTION**

- Layout of a Solidity Source File
- Structure of a Contract
- Types
- Units and Globally Available Variables
- Expressions and Control Structures
- Contracts
- Inline Assembly
- Cheatsheet

RTD  v: latest

# PD-2.2.4 Ethereum.org

## ETHEREUM DEVELOPER RESOURCES

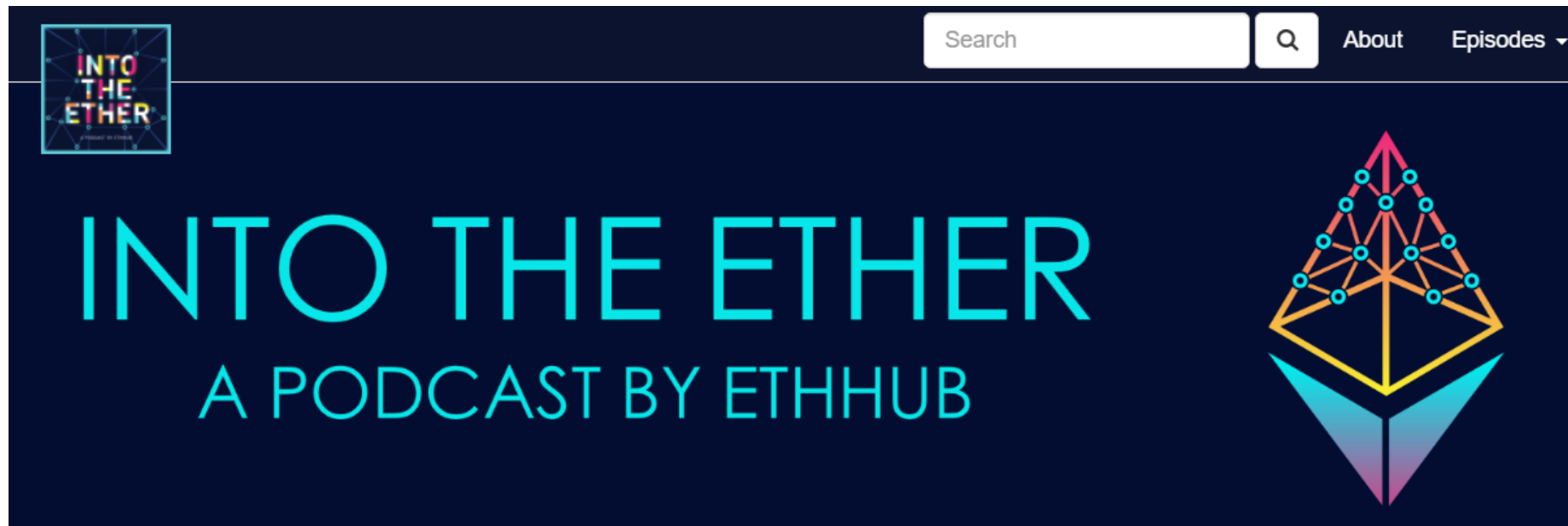
A builders manual for Ethereum. By  
builders, for builders.



# PD-2.2.5 EthHub



<https://docs.ethhub.io>



<https://podcast.ethhub.io>

# PD-2.2.6 Consensys developer portal

## Ethereum Developer Portal

### Documentation

Infura ›  
MetaMask ›  
Diligence (MythX) ›  
Quorum ›  
Codefi ›

### APIs

Infura ›  
MetaMask ›  
Diligence (MythX) ›  
Quorum ›

### Support

Infura ›  
MetaMask ›  
Diligence (MythX) ›  
Quorum ›  
Codefi ›

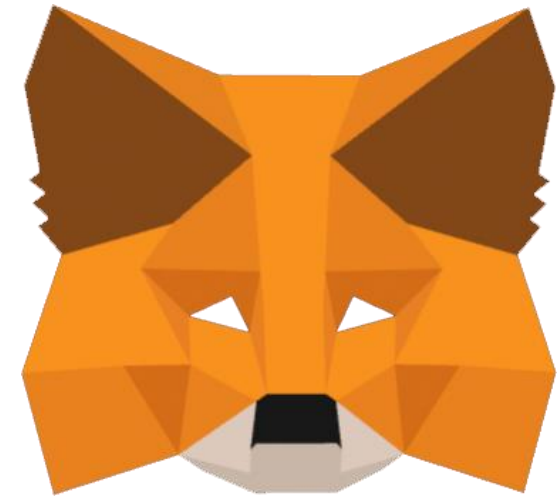
### Communities

Infura ›  
MetaMask ›  
Diligence (MythX) ›  
Quorum ›  
Codefi ›  
BUIDL Network ›

## PD-2.3 Prepare to use the play editor

- Install MetaMask
- Select testnetwork
- Get test ETH
- Play editor
- Etherscan
- Avalanche chain

# PD-2.3.1 Install Metamask



Location	Action	Object
<a href="https://www.google.com">https://www.google.com</a> – Search bar	Enter	metamask
<a href="https://www.google.com/search?q=metamask">https://www.google.com/search?q=metamask</a>	Click	MetaMask
<a href="https://metamask.io">https://metamask.io</a>	Click	get chrome extension
<a href="https://chrome.google.com/webstore/">https://chrome.google.com/webstore/...</a>	Click	Add to Chrome
Popup Add “MetaMask”?	Click	Add extension
chrome-extension://nkbi.../home.html#initialize/welcome	Click	Get Started
chrome-extension://nkbi.../home.html#initialize/select-action	Click	Create a wallet
chrome-extension://nkbi.../home.html#initialize/metametrics-opt-in	Click	I agree
Start menu	Start	{password manager}
Password manager	Do	Create random password
Password manager	Copy	Password
chrome-extension:... field: New password	Paste	{password}
chrome-extension:... field: Confirm password	Paste	{password}
chrome-extension:... checkbox: I have read ...	Click	{checkbox}
chrome-extension://nkbi.../home.html#initialize/seed-phrase	Click	Click here to reveal...
{ paper}	Write	{seed phrase}
	Click	Next
chrome-extension://nkbi.../home.html#initialize/seed-phrase/confirm	Click	{ All the words}
	Click	Confirm
chrome-extension://nkbi.../home.html#initialize/end-of-flow	Click	All Done
chrome-extension://nkbi.../home.html#	Close	{windows}

<https://www.youtube.com/watch?v=Wc-Hgn1QUjA>



<https://metamask.io/>

[http://web3examples.com/ethereum/install/Install\\_MetaMask\\_Windows.html](http://web3examples.com/ethereum/install/Install_MetaMask_Windows.html)

# PD-2.3.2 Goerli test ETH

Goerli Testnet Faucet

GOERLI FAUCET

0xEA9a7c7cD8d4Dc3acc6f0AaEc1506C8D6041a1c!

You can use <https://rpc.slock.it/goerli> RPC to connect to the Goerli testnet.

REQUEST 0.05 GÖETH

I'm not a robot

Powered by **Slock.it** - Send feedback to <https://github.com/slockit/goerli-faucet>

Goerli Testnet Faucet

GOERLI FAUCET

0xEA9a7c7cD8d4Dc3acc6f0AaEc1506C8D6041a1c!

You can use <https://rpc.slock.it/goerli> RPC to connect to the Goerli testnet.

REQUEST 0.05 GÖETH

I'm not a robot

Powered by **Slock.it** - Send feedback to <https://github.com/slockit/goerli-faucet>

Goerli Testnet Faucet

Success

0.05 GoETH has been successfully transferred to [0xEA9a7c7cD8d4Dc3acc6f0AaEc1506C8D6041a1c5](https://faucet.goerli.mudit.blog)

OK

I'm not a robot

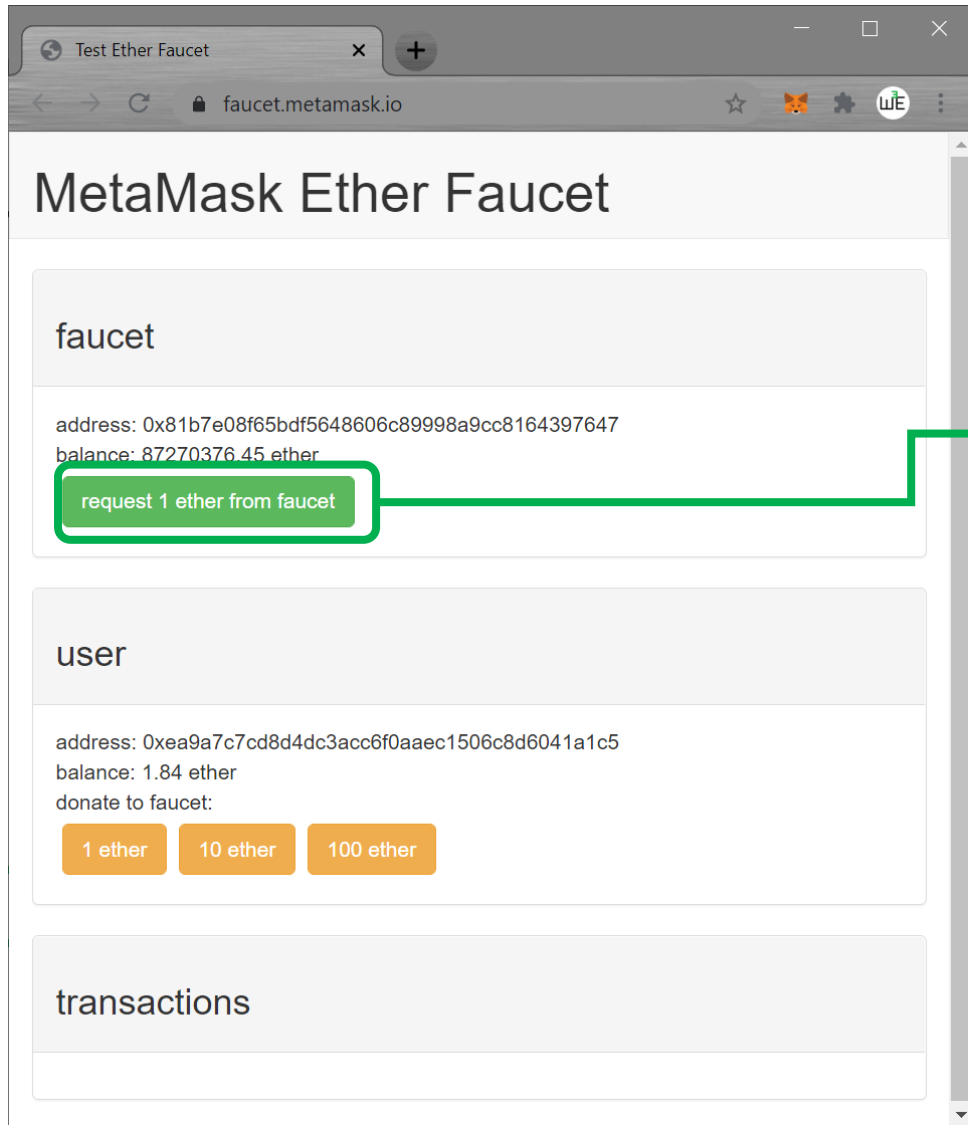
Powered by **Slock.it** - Send feedback to <https://github.com/slockit/goerli-faucet>

<https://goerli-faucet.slock.it>

<https://faucet.goerli.mudit.blog>



# PD-2.3.2.1 Ropsten test ETH



Test Ether Faucet

faucet.metamask.io

## MetaMask Ether Faucet

faucet

address: 0x81b7e08f65bdf5648606c89998a9cc8164397647  
balance: 87270376.45 ether

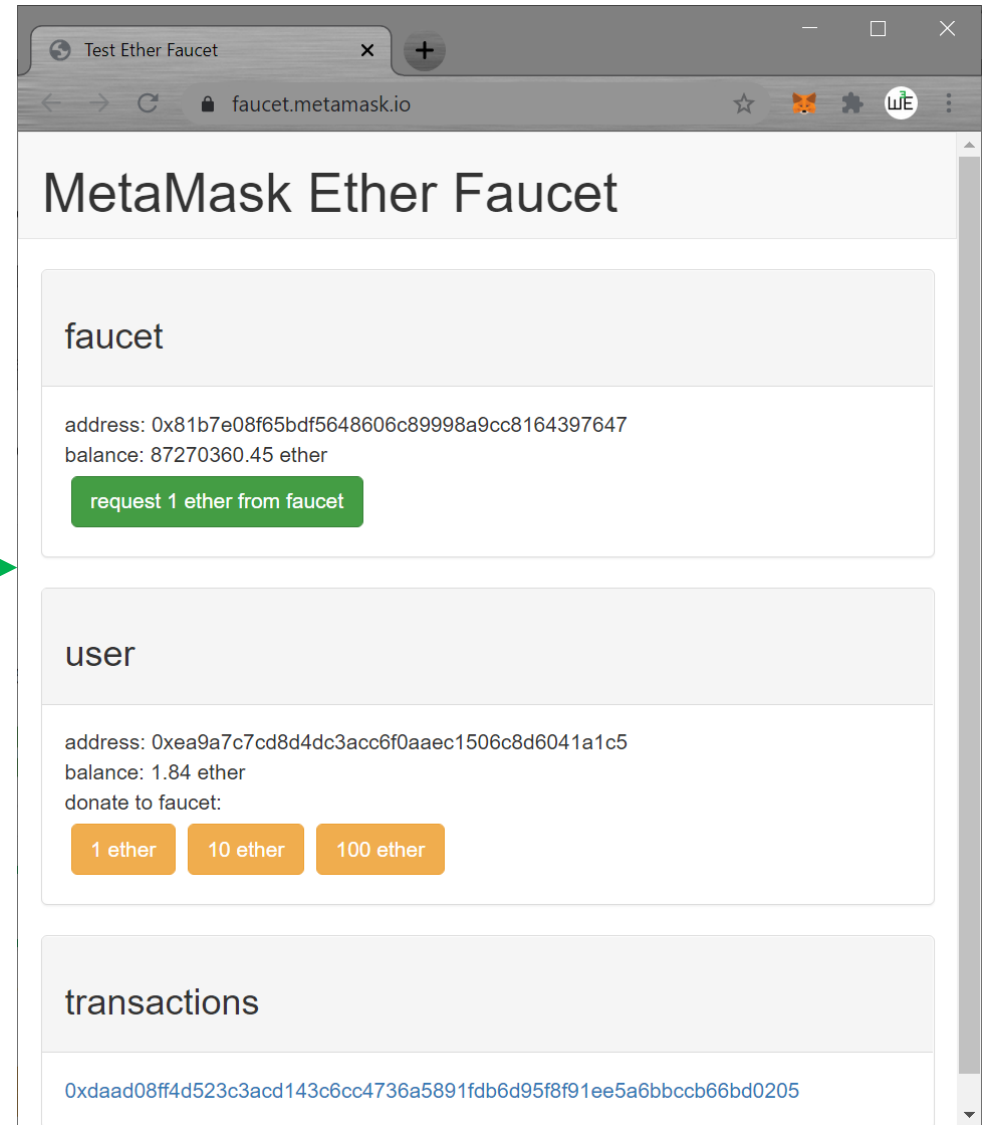
request 1 ether from faucet

user

address: 0xea9a7c7cd8d4dc3acc6f0aaec1506c8d6041a1c5  
balance: 1.84 ether  
donate to faucet:

1 ether 10 ether 100 ether

transactions



Test Ether Faucet

faucet.metamask.io

## MetaMask Ether Faucet

faucet

address: 0x81b7e08f65bdf5648606c89998a9cc8164397647  
balance: 87270360.45 ether

request 1 ether from faucet

user

address: 0xea9a7c7cd8d4dc3acc6f0aaec1506c8d6041a1c5  
balance: 1.84 ether  
donate to faucet:

1 ether 10 ether 100 ether

transactions

0xdaad08ff4d523c3acd143c6cc4736a5891fdb6d95f8f91ee5a6bbccb66bd0205



# PD-2.3.3 Play editor

The screenshot displays the Play editor interface. On the left, a code editor shows the Solidity source code for a contract named `SimpleStorage`. The code includes a pragma statement for Solidity 0.5.12, a `uint8` variable `storedData`, and two public functions: `set` and `get`. The `set` function takes a `uint8` parameter `x` and assigns it to `storedData`. The `get` function returns the value of `storedData`.

```
1
2 /*
3 You can use Play editor with any contract.
4 Paste it in the editor and wait for the preview to start interacting with it.
5
6 **To interact with the contract you will need a Metamask extension.
7 */
8
9
10 pragma solidity 0.5.12;
11
12 contract SimpleStorage {
13     uint8 storedData;
14
15     function set(uint8 x) public {
16         storedData = x;
17     }
18
19     function get() public view returns (uint8) {
20         return storedData;
21     }
22 }
23
24 }
25
```

On the right, a preview window titled "SimpleStorage" shows a user interface. At the top, there are "Publish" and "Connect" buttons. Below them, there are two function call boxes. The first box is for the `get` function, and the second box is for the `set` function. The `set` function box has an input field labeled `x` with the value `0` entered. A "Publish" button with a right-pointing arrow is highlighted with a green box in the top right corner of the preview area.

<https://playproject.io/play-ed/>

<https://github.com/playproject-io/play-ed>

# PD-2.3.4 Etherscan

TESTNET Goerli (GTH) Blockchain x

goerli.etherscan.io

Etherscan

Home Blockchain Tokens Misc Goerli

Gö Goerli Testnet Explorer

All Filters Search by Address / Txn Hash / Block / Token / Ens

Advertise your brand here! Start Today

### Latest Blocks

Bk	3492873 39 secs ago	Miner 0x22ea9f6b28db76a71... 1 txn in 15 secs	5.00009 Eth
Bk	3492872 54 secs ago	Miner 0x9d525e28fe5830ee9... 2 txns in 15 secs	5.00008 Eth
Bk	3492871 1 min ago	Miner 0xe0a2bd4258d276883... 0 txn in 15 secs	5 Eth
Bk	3492870 1 min ago	Miner 0xd9a5179f091d85051... 2 txns in 15 secs	5.00015 Eth
Bk	3492869 1 min ago	Miner 0x8b24eb4e6aae90605... 1 txn in 15 secs	5.00009 Eth
Bk	3492868 1 min ago	Miner 0x22ea9f6b28db76a71... 4 txns in 15 secs	5.00049 Eth

View all blocks

### Latest Transactions

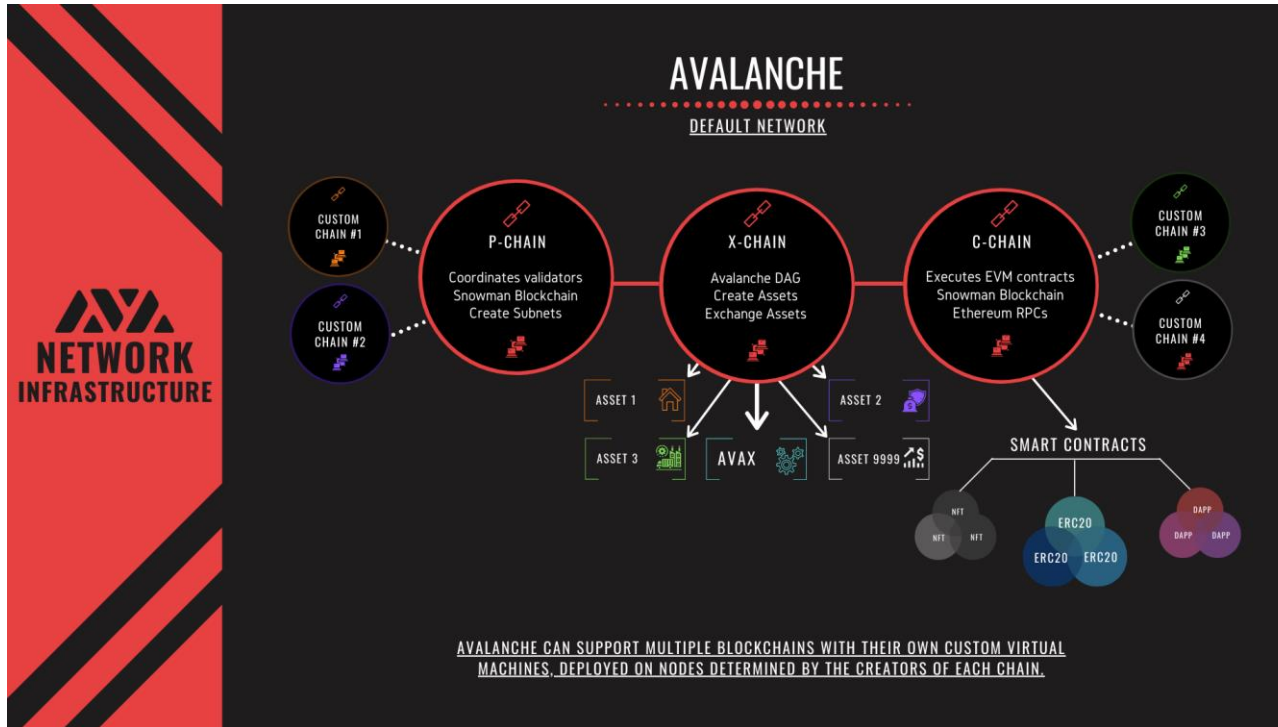
Tx	0xab5622e83b... 39 secs ago	From 0x256b25ff4bbf82c0f23... To 0x2b062e2934f96758a...	0 Eth
Tx	0x6943822114e... 54 secs ago	From 0x8ced5ad0d8da4ec21... To 0x830d1568a8adfea21f...	0.05 Eth
Tx	0x8ee9075cb31... 54 secs ago	From 0x0828d0386c1122e56... To 0x7753cfad258efbc52a...	0 Eth
Tx	0xb65c76e6567... 1 min ago	From 0x8ced5ad0d8da4ec21... To 0x144aa2ae61fbd332c...	0.05 Eth
Tx	0x90b1b7f8abba... 1 min ago	From 0xab1ca038afa53fb73b... To 0xbcccd7e437d3bf919...	0 Eth
Tx	0x1afadbce1629... 1 min ago	From 0x256b25ff4bbf82c0f23... To 0x2b062e2934f96758a...	0 Eth

View all transactions

Powered by Ethereum

Preferences

# PD-2.3.5 Avalanche C-Chain



The screenshot shows the Metamask Settings dialog for the Avalanche FUJI C-Chain. The settings are as follows:

- Network Name: Avalanche FUJI C-Chain
- New RPC URL: <https://api.avax-test.network/ext/bc/C/>
- Chain ID: 0xa869
- Symbol (optional): C-AVAX
- Block Explorer URL (optional): <https://cchain.explorer.avax-test.netwo>

Buttons for 'Cancel' and 'Save' are visible at the bottom of the dialog.

## FUJI Testnet Settings:

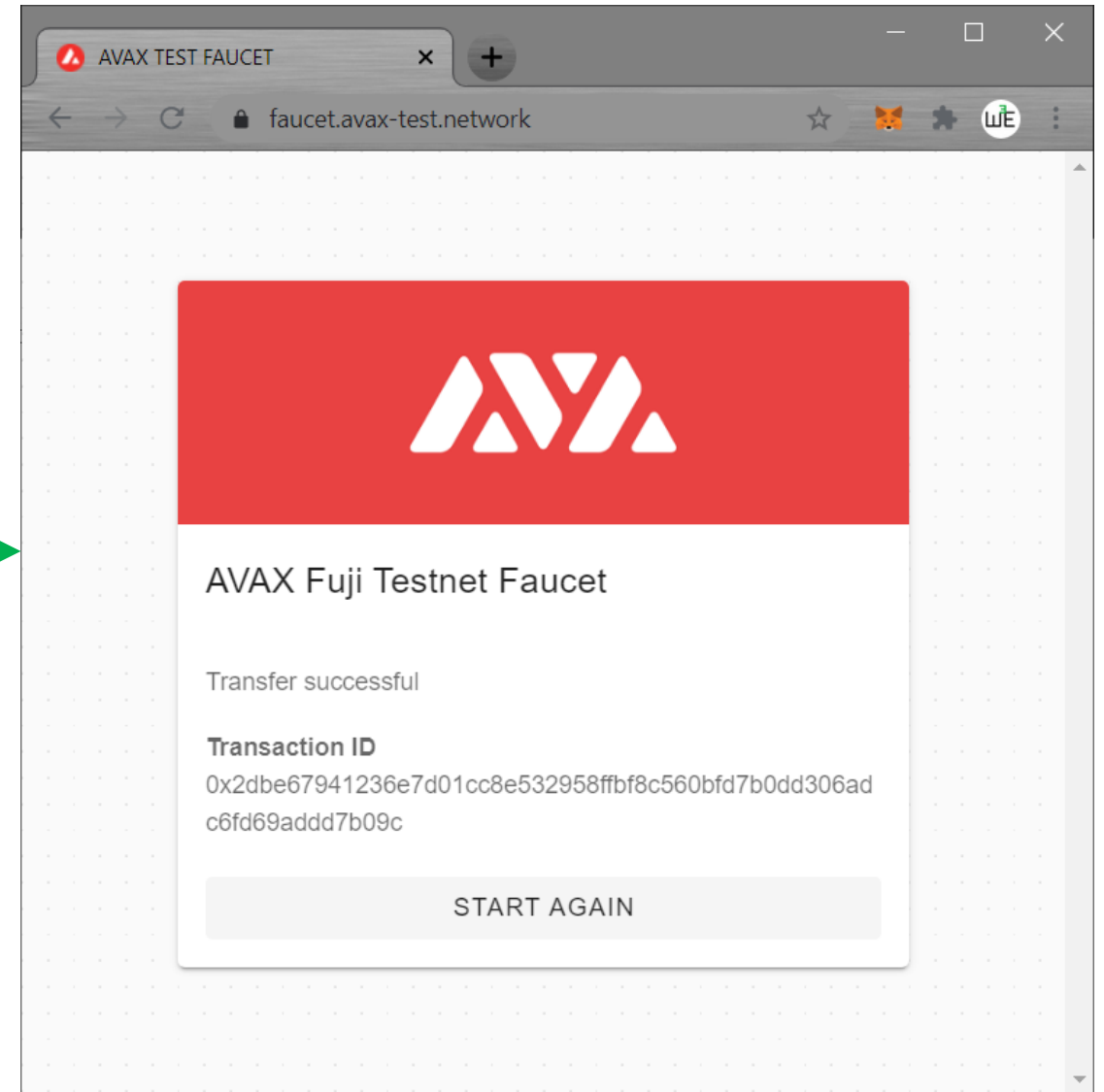
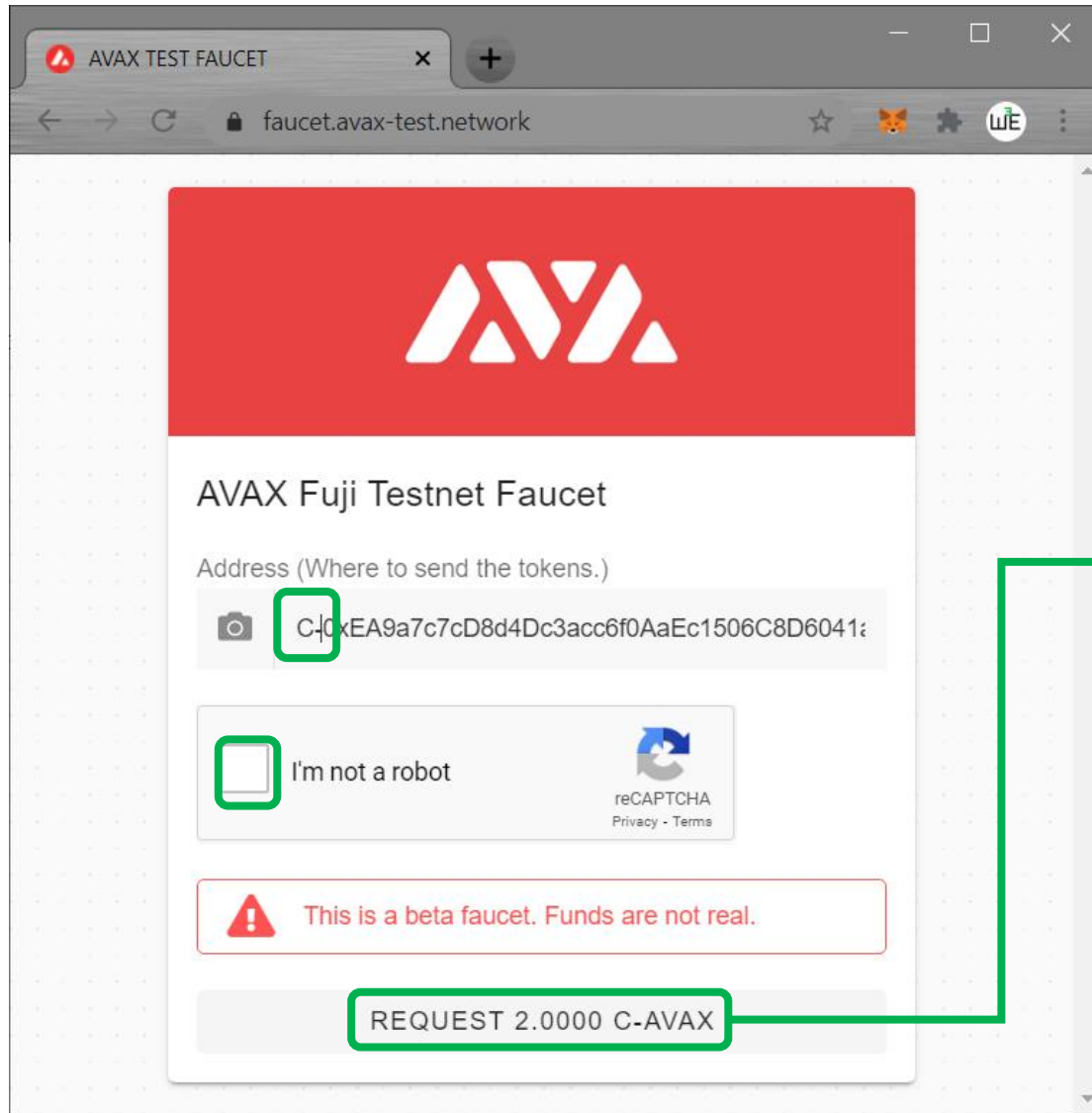
- **Network Name:** Avalanche FUJI C-Chain
- **New RPC URL:** <https://api.avax-test.network/ext/bc/C/rpc>
- **ChainID:** 0xa869
- **Symbol:** C-AVAX
- **Explorer:** <https://cchain.explorer.avax-test.network>

<https://api.avax-test.network/ext/bc/C/rpc>

<https://cchain.explorer.avax-test.network>

<https://docs.avax.network/build/tutorials/platform/deploy-a-smart-contract-on-avalanche-using-remix-and-metamask>

# PD-2.3.6 Avalanche C-Chain faucet



# PD-2.3.7 Avalanche C-Chain Play editor

The screenshot displays the Play editor interface for an Avalanche C-Chain contract. On the left, a code editor shows the Solidity source code for a SimpleStorage contract. The code includes a pragma statement for Solidity 0.5.9, a uint8 variable named storedData, a set function, and a get function. The right side of the interface features a preview of the SimpleStorage contract, showing its published status, address, and a 'get' function call with the value '133'. A Metamask account overlay is visible in the foreground, displaying the account name 'Account 1', the balance '1.9349 ETH', and a list of recent transactions including 'Contract Interaction' and 'Contract Deployment', both showing a balance change of '-0 ETH'.

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
contract.sol output.json
/*
 * You can use Play editor with any contract.
 * Paste it in the editor and wait for the preview to start interacting with it.
 * **To interact with the contract you will need a Metamask extension.
 */
pragma solidity 0.5.9;
contract SimpleStorage {
    uint8 storedData;
    function set(uint8 x) public {
        storedData = x;
    }
    function get() public view returns (uint8) {
        return storedData;
    }
}
```

SimpleStorage

published  
Wed Sep 30 2020 11:29:16  
contract address (homest  
0x4Ebd111fC91eA11c824C19  
published by  
0xEA9a7c7cD8d4Dc3acc6f0A

get

"133"

set

x ————— 133 ✓

→

{}]

Support chat

Avalanche C-Chain

Account 1  
0xEA9a...a1c5

1.9349 ETH

BUY SEND

Assets Activity

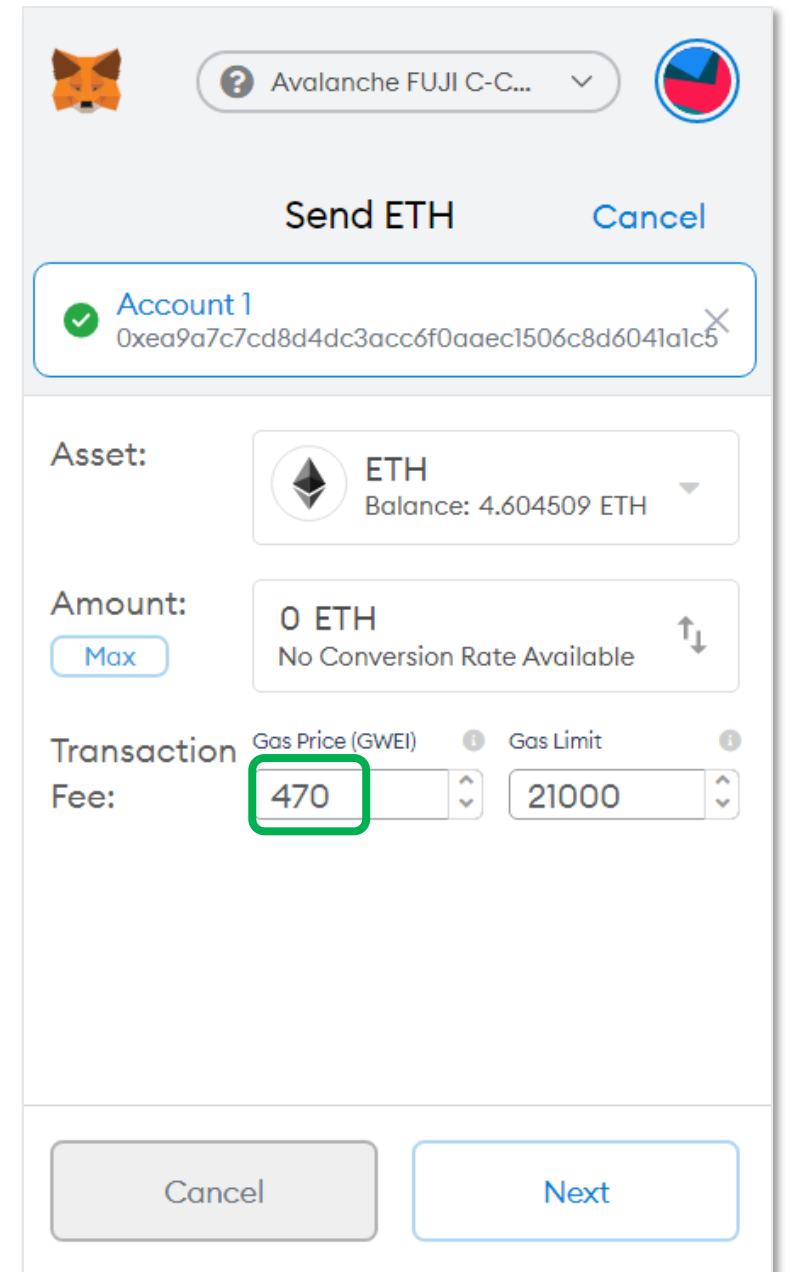
Contract Interaction  
Sep 30 · playproject.io -0 ETH -0 ETH

Contract Deployment  
Sep 30 · playproject.io -0 ETH -0 ETH

# PD-2.3.8 Avalanche GasPrice

The C-Chain gas price is  $4.7e-7$  AVAX/gas.

The C-Chain gas limit is  $10e8$ .



The screenshot shows a transaction confirmation screen for sending ETH. At the top, there is a header with the Avalanche logo, a dropdown menu set to 'Avalanche FUJI C-C...', and a profile picture. Below the header are two buttons: 'Send ETH' and 'Cancel'. The account selected is 'Account 1' with the address '0xea9a7c7cd8d4dc3acc6f0aaec1506c8d6041a1c5'. The 'Asset' is set to 'ETH' with a balance of 4.604509 ETH. The 'Amount' is '0 ETH' with a 'Max' button and a note 'No Conversion Rate Available'. The 'Transaction Fee' section shows 'Gas Price (GWEI)' set to '470' (highlighted with a green box) and 'Gas Limit' set to '21000'. At the bottom, there are 'Cancel' and 'Next' buttons.

# PD-2.4 Solidity

- Try out several examples

<http://web3examples.com/ethereum/#solidity>

[http://web3examples.com/ethereum/solidity\\_examples](http://web3examples.com/ethereum/solidity_examples)

<https://solidity-by-example.org>



# PD-2.4.1 SimpleStorage

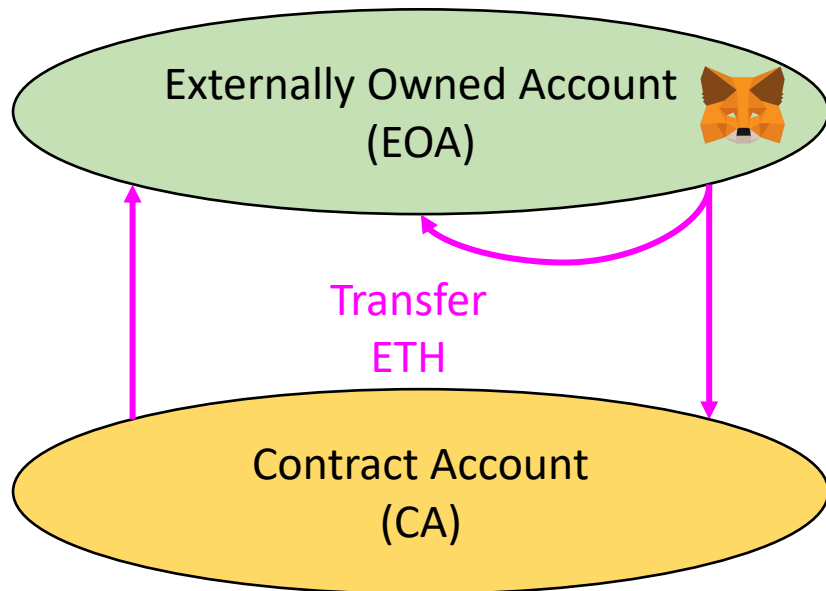
```
SimpleStorage.sol x
1
2  /*
3   You can use Play editor with any contract.
4   Paste it in the editor and wait for the preview to start interacting with it.
5
6   **To interact with the contract you will need a Metamask extension.
7   */
8
9
10  pragma solidity 0.5.9;
11
12  contract SimpleStorage {
13
14      uint8 storedData;
15
16      function set(uint8 x) public {
17          storedData = x;
18      }
19
20      function get() public view returns (uint8) {
21          return storedData;
22      }
23
24  }
```

# PD-2.4.2 Hello World

```
HelloWorld.sol x
1  pragma solidity ^0.5.11;
2
3  contract HelloWorld {
4      string public welcome = "Hello World!";
5  }
```

# PD-2.4.3 Transfer ETH

```
Transfer.sol
1 // https://raw.githubusercontent.com/web3examples/ethereum/master/solidity_examples/Transfer.sol
2 pragma solidity >=0.4.0 <0.7.0;
3 contract TestPay {
4     function ContractBalance() public view returns (uint) { return address(this).balance; }
5     function () external payable {}
6     function PayToContract() public payable {}
7     function RequestFromContract() public { msg.sender.transfer(0.1 ether); }
8     function Transfer(address payable to) public { to.transfer(0.05 ether); }
9 }
```



<http://web3examples.com/ethereum/demo/Play Transfer eth with contract.html>

[https://github.com/web3examples/ethereum/blob/master/solidity\\_examples/Transfer.sol](https://github.com/web3examples/ethereum/blob/master/solidity_examples/Transfer.sol)



# PD-2.4.4 Types

```
Types.sol
1 pragma solidity ^0.5.11;
2
3 contract Types {
4     bool public a; // note: public variables => getter
5     int public b1;
6     int256 public b256=1;
7     uint public b2=2;
8     // no floating point
9     address public c1;
10    address payable public c2;
11    byte public d0;
12    bytes1 public d1; // fixed byte array
13    bytes3 public d3;
14    bytes32 public d32;
15    uint[] public e1=[12,7,6,4];
16
17    function e1getCount() public view returns (uint count) {
18        return e1.length;
19    }
20
21    string public e2="Test";
22
23    enum Choices { A, B, C, D }
24    Choices public f1=Choices.D;
25 }
```

# PD-2.4.5 Arrays

```
Arrays.sol
1 pragma solidity ^0.5.11;
2
3 contract TestArray {
4     struct member {
5         uint data;
6         address sender;
7     }
8
9     member[] public List;
10    mapping(address => uint) public Map;
11
12    function add(uint x) public {
13        member memory temp;
14        temp.data=x;
15        temp.sender=msg.sender;
16        Map[msg.sender]=x;
17        List.push(temp);
18    }
19
20    function ListLength() public view returns (uint) {
21        return List.length;
22    }
23 }
```

# PD-2.4.6 Mappings

```
Mapping.sol x
1  pragma solidity ^0.5.11;
2
3  contract RegisterParticipants {
4      mapping(address => bool) public MapParticipant;
5      address[] public ListParticipant;
6      mapping(address => uint) public IndexInList;
7
8      constructor() public {
9          ListParticipant.push(address(0)); // "use" address 0, to make tests easier
10     }
11     function Participate(bool Join) public {
12         MapParticipant[msg.sender]=Join;
13         uint i=IndexInList[msg.sender];
14         if (i > 0) { // Delete previous participation entry
15             ListParticipant[i]=ListParticipant[ListParticipant.length-1]; // switch
16             IndexInList[msg.sender]=0;
17             IndexInList[ListParticipant[i]]=i;
18             ListParticipant.pop();
19         }
20         if (Join) {
21             ListParticipant.push(msg.sender);
22             IndexInList[msg.sender]=ListParticipant.length-1;
23         }
24     }
25     function NrOfParticipants() public view returns (uint) {
26         return ListParticipant.length-1;
27     }
28 }
```

# PD-2.4.7 Storage / Memory

```
Memory_Storage.sol x
1 // Shows the difference between memory and storage
2 pragma solidity >=0.5.11;
3
4 contract TicketSystem {
5     struct Ticket {uint val;}
6     Ticket[] public ticketArray;
7
8     constructor () public {
9         Ticket memory ticket = Ticket(1234);
10        ticketArray.push(ticket);
11    }
12    function StoreValue1 (uint value) public returns (uint) {
13        Ticket storage ticket = ticketArray[0]; // pointer
14        ticket.val = value;
15        return ticketArray[0].val; // returns the updated val
16    }
17    function StoreValue2 (uint value) public returns (uint) {
18        Ticket memory ticket = ticketArray[0]; // copy
19        ticket.val = value;
20        return ticketArray[0].val; // does not return the updated val
21    }
22 }
```