ETHEREUM OVERVIEW

Created: Sept 2023
Last Edited: Sept 2023

UofT: CSCD71F22
-David Liu, Founder of dApp Technology Inc.

BLOCKCHAIN 1.0

- First concept of decentralization
- Focus on cryptocurrency
- Emergence of cryptocurrency wallets, mining rigs, mining software and decentralized blockchain computer

- Notable projects:
- ECash (by DigiCash 1983)
- Bitcoin (by Cypherpunks 2009)

BLOCKCHAIN 1.0 APPS

Constraints:

- Limited Transactions types
- Limited Data Types
- Small data storage size

Mitigation:

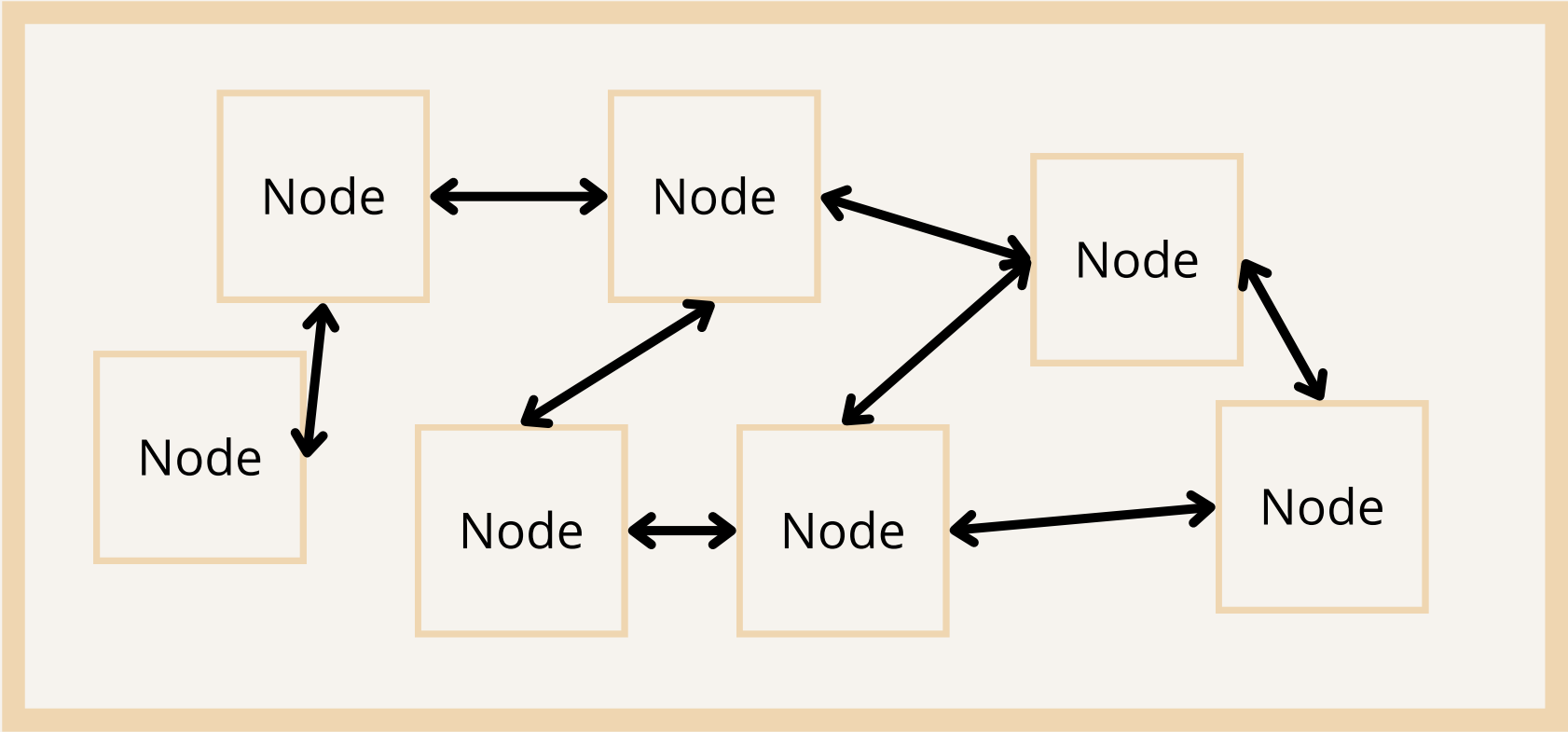
- Compute functionality off-chain

BLOCKCHAIN 2.0

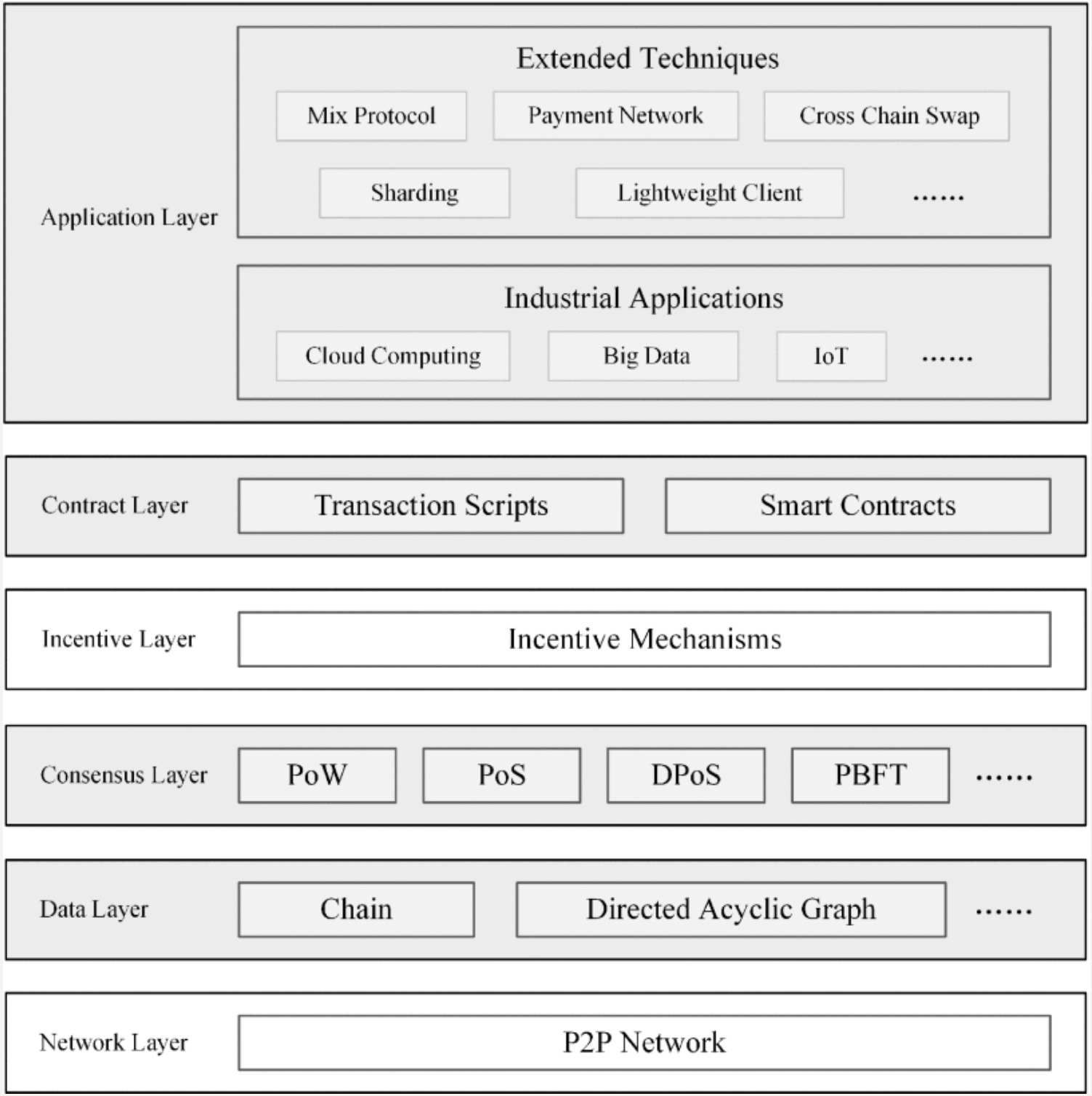
- Emergence of Decentralized Code (Smart Contract)
- Mass adoption of Decentralized Applications (dApps)

- Notable projects:
- Ethereum (Blockchain by Vitalik Buterin 2013)
- Uniswap (ETH dApp by Hayden Adams 2018)

ETHEREUM NETWORK



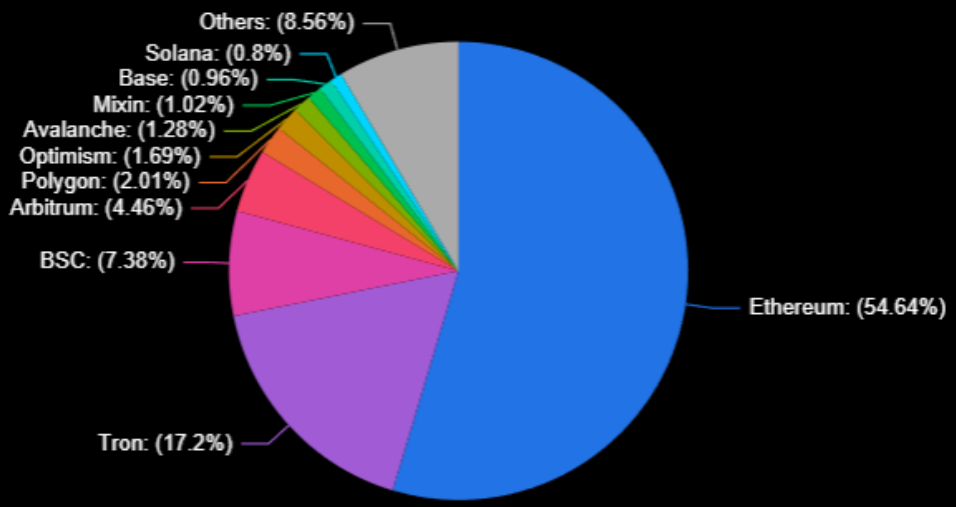
- Each Node stores a portion of the blockchain and runs the EVM to execute code from Smart Contracts
- Ethereum Validators receives data from the Nodes and adds new blocks to the blockchain


Application Layer
Extended Techniques
Mix Protocol
Payment Network
Cross Chain Swap
Sharding
Lightweight Client
.....
Industrial Applications
Cloud Computing
Big Data
IoT
.....
Contract Layer
Transaction Scripts
Smart Contracts
Incentive Layer
Incentive Mechanisms
Consensus Layer
PoW
PoS
DPoS
PBFT
.....
Data Layer
Chain
Directed Acyclic Graph
.....
Network Layer
P2P Network

ETHEREUM QUICK NOTES

- First Decentralized Blockchain with Smart Contracts (Turing Complete)
- Programs run on the Ethereum Virtual Machine (EVM)
- Ether is the native token used to pay for transactions (Gas Fees)
- Largest adopted blockchain for dApps and largest community support
- Solidity is the most popular coding language

Total Value Locked All Chains



Name	Protocols	Active Users	TVL	Stables	Mcap/TVL
1 Ethereum	912		$21.18b	$66.409b	9.33
2 Tron	25		$6.665b	$44.57b	1.13
3 BSC	648		$2.86b	$5.005b	11.69
4 Arbitrum	453	120,979	$1.73b	$1.658b	0.63
5 Polygon	475	295,951	$780.31m	$1.299b	6.5
6 Optimism	190	68,756	$655.24m	$581.65m	1.69
7 Avalanche	331	28,300	$496.54m	$1.224b	6.58
8 Mixin	10		$394.16m	$46.93m	
9 Base	123	70,713	$370.29m	$116.3m	
10 Solana	108	75,010	$310.43m	$1.521b	26.59

Sept 23, 2023
DeFi Llama

ETHEREUM NETWORK (SEPT 2023)

Ethereum today

The latest network statistics

TOTAL ETH STAKED

The total amount of ETH currently being staked and securing the network.

25.19M

30d 90d

TRANSACTIONS TODAY

The number of transactions successfully processed on the network in the last 24 hours.

1.022M

30d 90d

VALUE LOCKED IN DEFI (USD)

The amount of money in decentralized finance (DeFi) applications, the Ethereum digital economy.

$48.34B

30d 90d

NODES

Ethereum is run by thousands of volunteers around the globe, known as nodes.

7,875

30d 90d

Sept 2023
ethereum.org

ETHEREUM STATS (SEPT 2023)

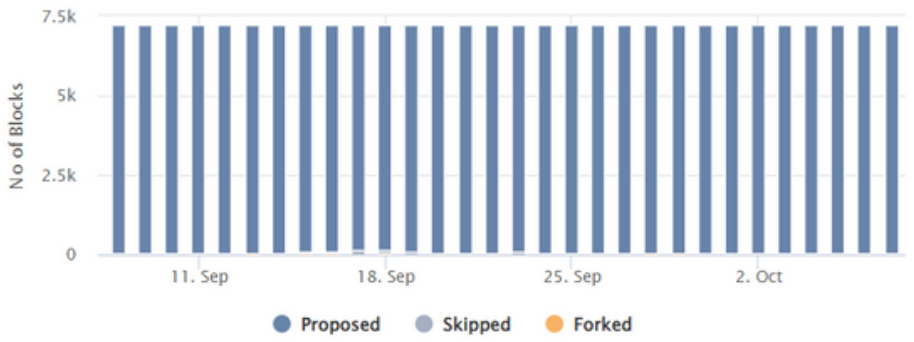
Showing the last 30 days

07 Sep 2023 - 06 Oct 2023

BLOCKS

View Details

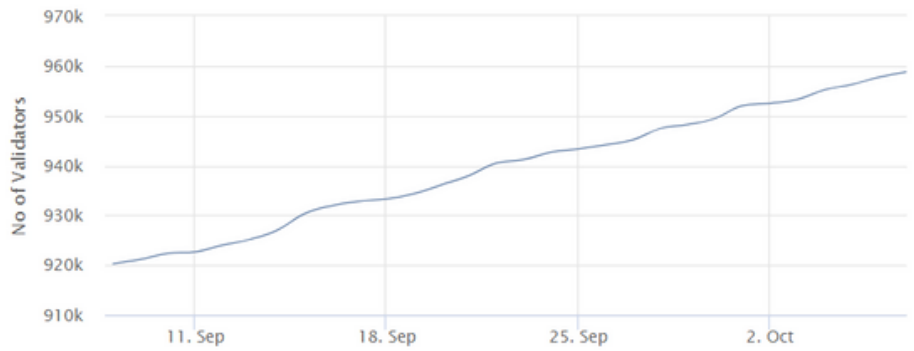
Blocks produced on the Beacon Chain.



VALIDATORS

View Details

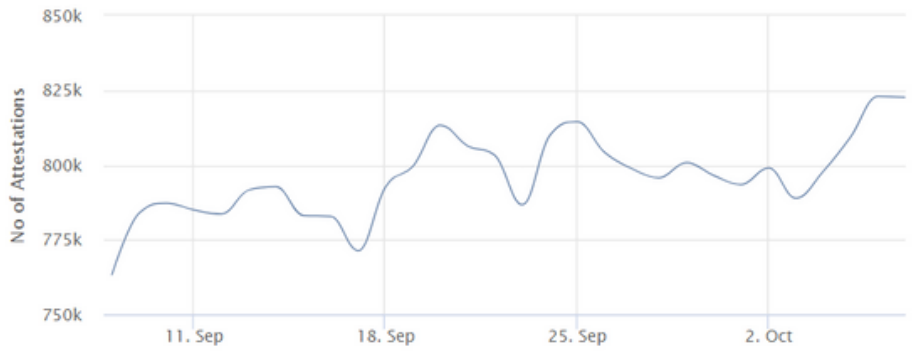
The number of validators being run on the Beacon Chain.



ATTESTATIONS

View Details

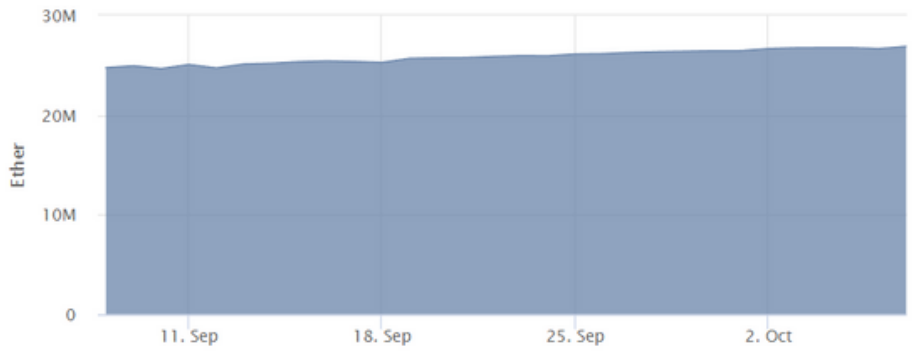
The votes on the validity of newly created blocks on the Beacon Chain.



ETHER VOTED

View Details

The number of ether staked in the participation on the Beacon Chain.



Sept 2023
https://beaconscan.com/statistics

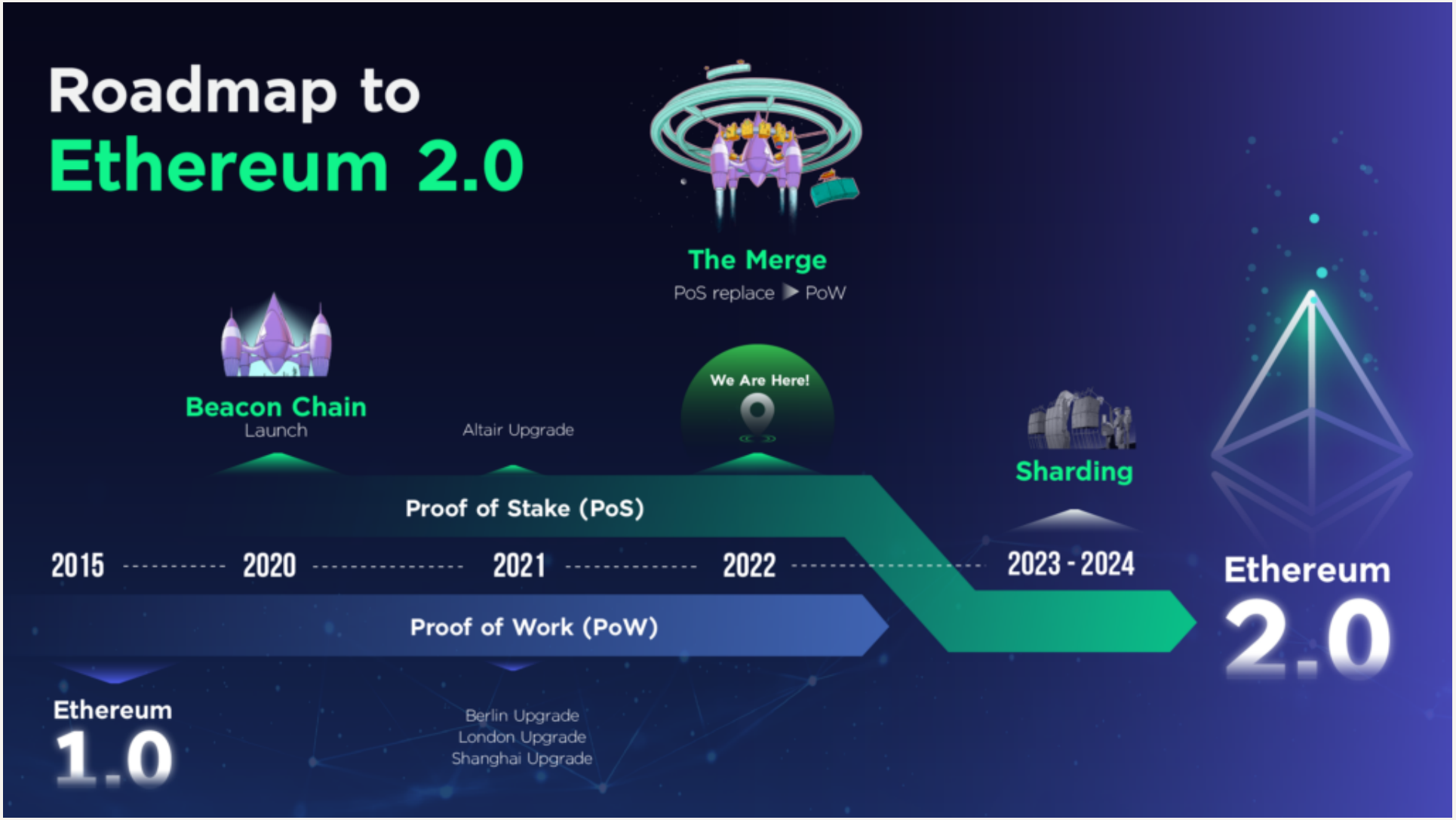
ETHEREUM STATS (SEPT 2023)

Ethereum in numbers

4k+	96M+	53.3M+
Projects built on Ethereum	Accounts (wallets) with an ETH balance	Smart contracts on Ethereum
$410B	$3.5B	1.022M
Value secured on Ethereum	Creator earnings on Ethereum in 2021	Number of transactions today

Sept 2023
ethereum.org

ETHEREUM TIMELINE

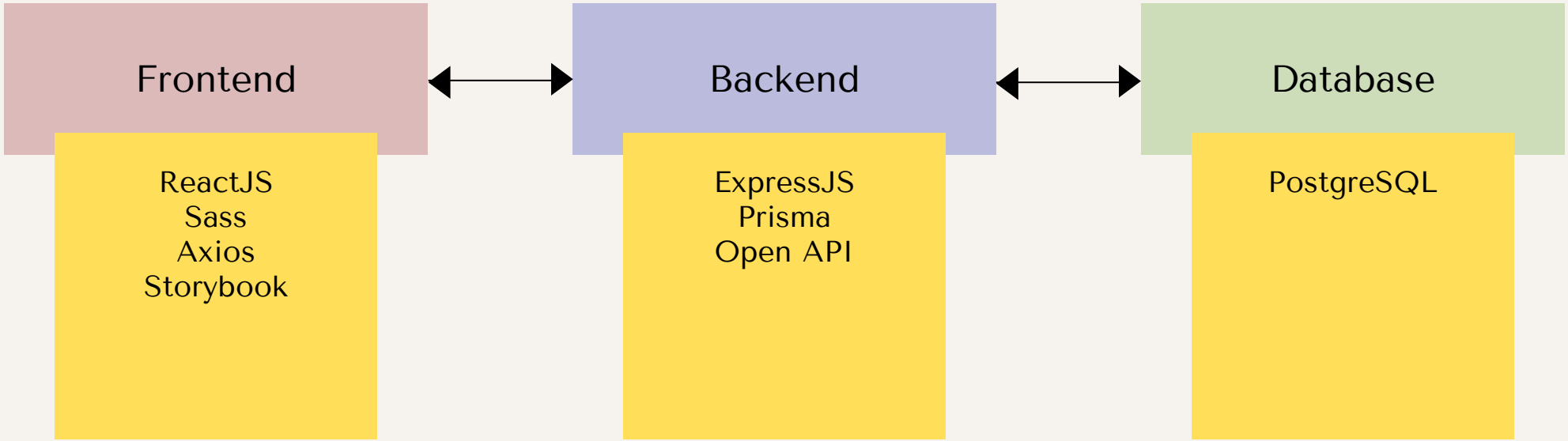


Sept 2023
https://content.bitazza.com/eth1-0-2-0/

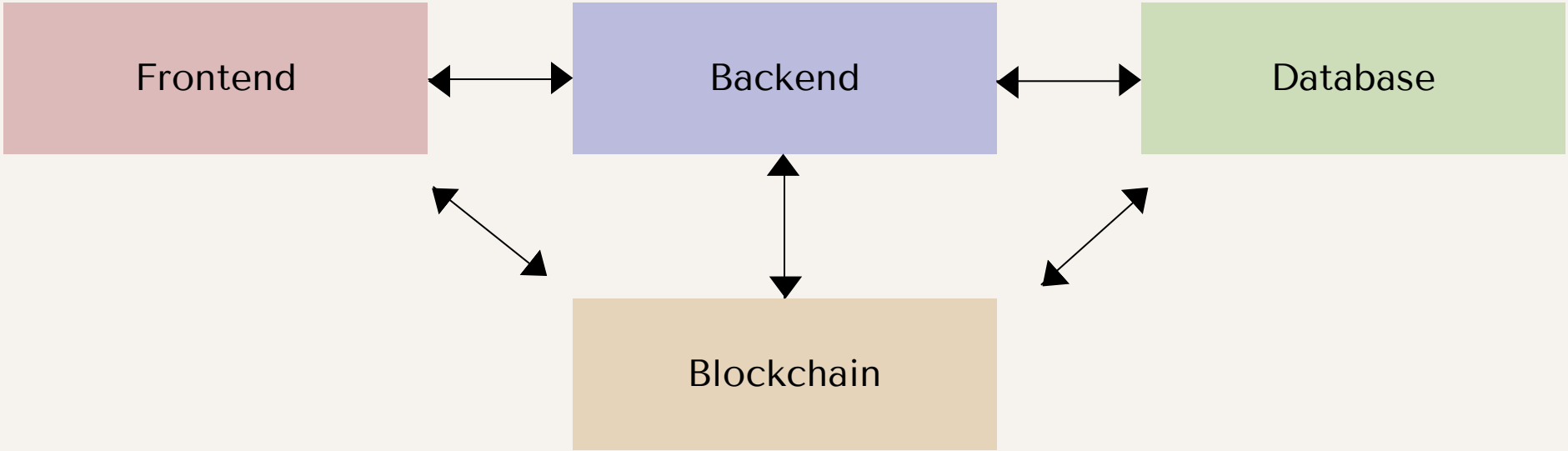
Blockchain Development

FULL STACK

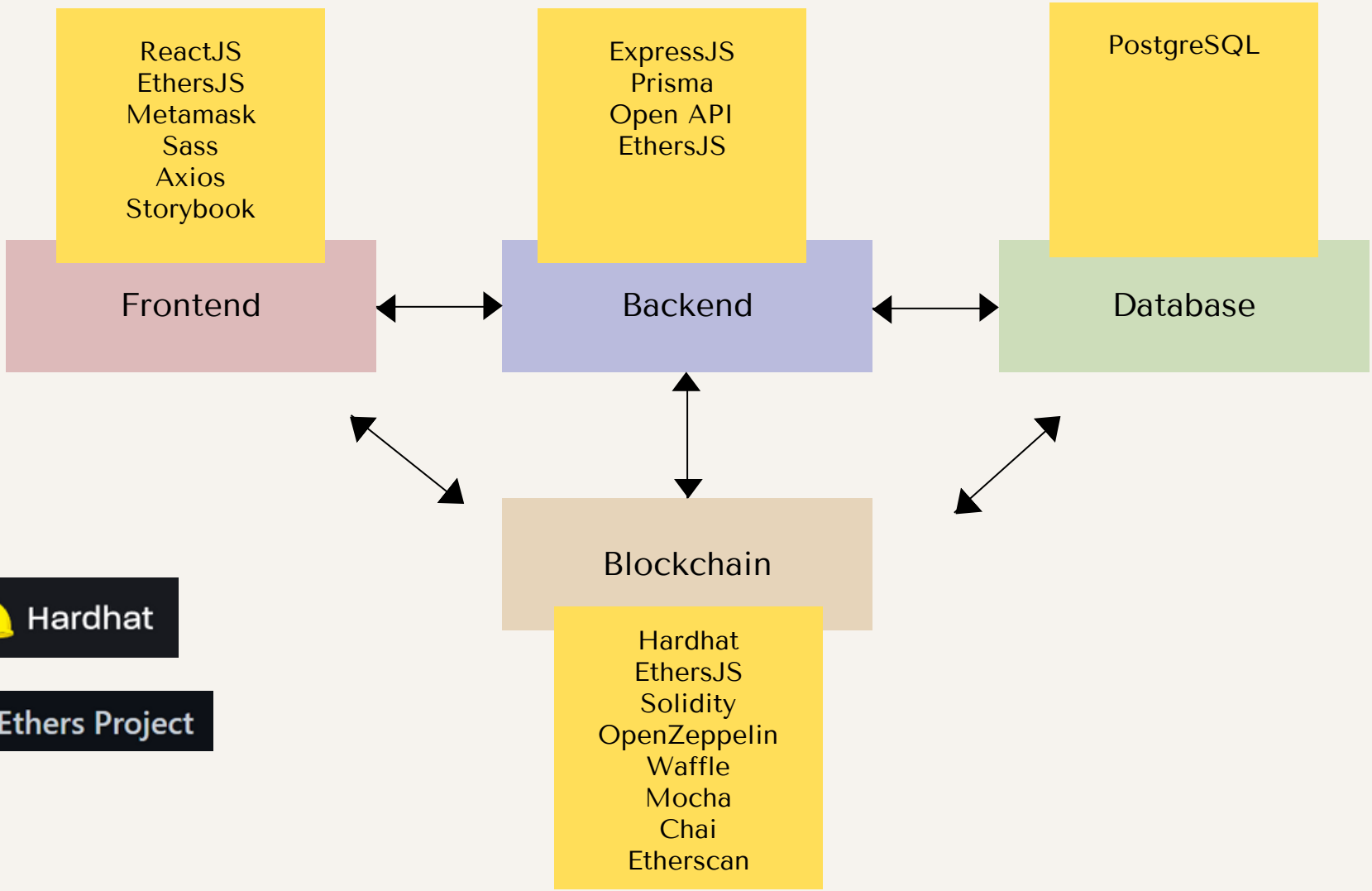
FULL STACK EXAMPLE



BLOCKCHAIN FULL STACK



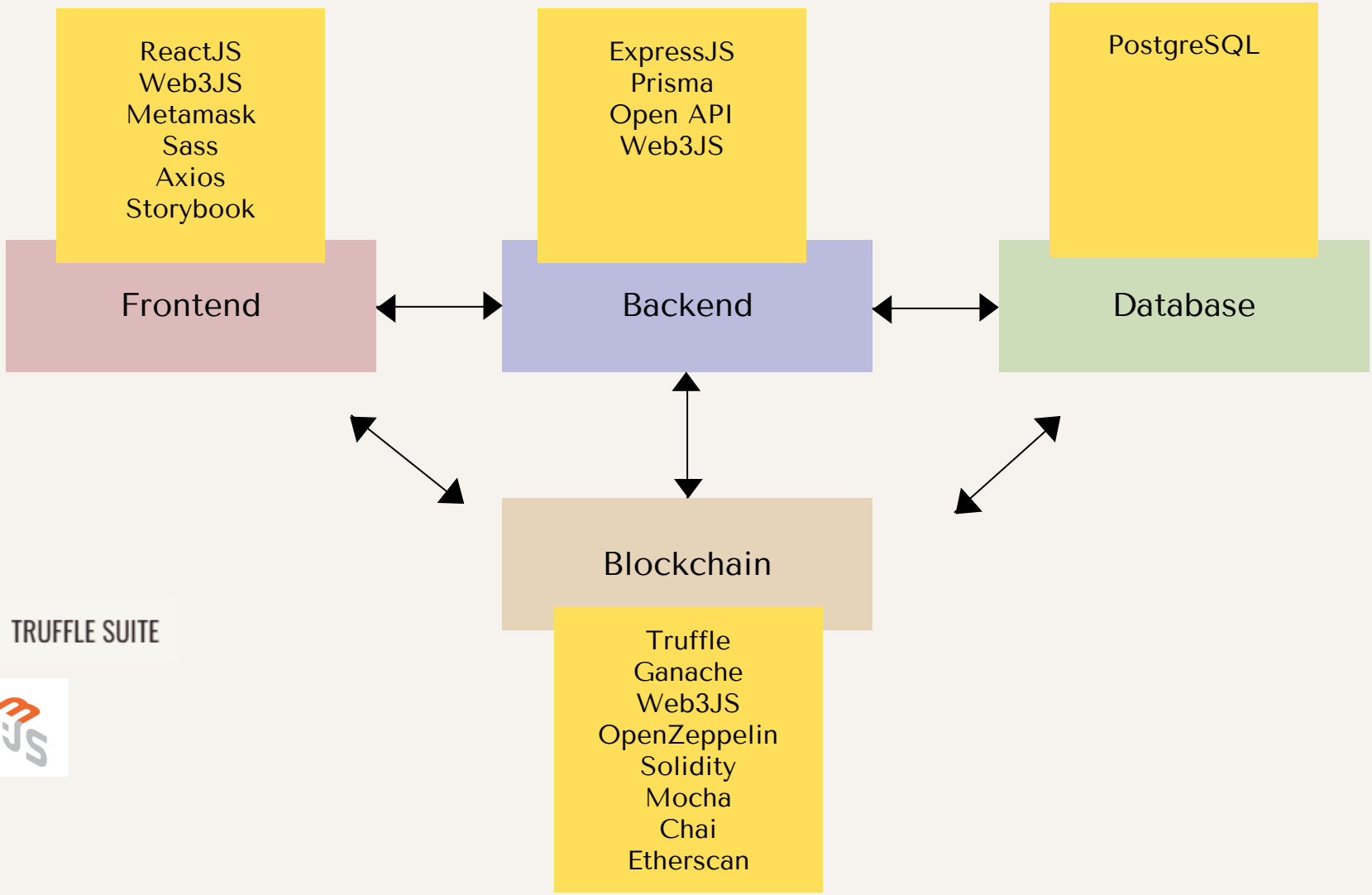
BLOCKCHAIN FULL STACK ETHERSJS EXAMPLE



Hardhat

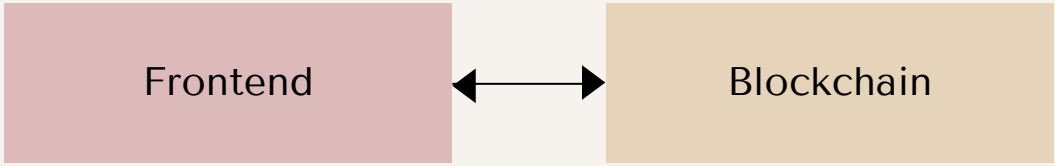
The Ethers Project

BLOCKCHAIN FULL STACK WEB3JS EXAMPLE



TRUFFLE SUITE

MINIMAL DAPP



DECENTRALIZED APPS

- Auditability (Open sourced code)
- High availability (Distributed)
- Transparency (Open transactions)
- Neutrality (Decentralized Governance)

BLOCKCHAIN INTERACTION TECH COMPARISON

FEATURES	ETHERSJS	WEB3JS
Community		●
Code readability	●	
Package Size	●	
Well tested	●	
Tutorial Materials		●

REMIX IDE

- A learning platform for developing, deploying and administering ETH Smart Contracts.

REMIX

PIGGY BANK CONTRACT

Variables		
admin	address	Admin of the Contract

Functions		
constructor	(admin: address)	Sets the admin
deposit	()	Deposit ETH to piggy bank.
withdraw	(receiver: address, amt: uint)	Admin withdraw ETH to any address

Events		
Deposit	(depositor: address, amt: uint)	A deposit has occurred
Withdraw	(receiver: address, amt: uint)	A withdraw has occurred

WHAT ARE SMART CONTRACTS?

“a set of promises, specified in digital form, including protocols within which parties perform on the other promises."
-Nick Szabo (1990s)

Definition has changed since the invention of Bitcoin (2009). Smart Contracts are not smart nor are they legally binding.

SMART CONTRACT DEFINITION

- Computer Programs: code that can run
- Immutable: when code is deployed to blockchain, it cannot change
- Deterministic: given the context of the execution transaction and blockchain state, the outcome can be determined
- EVM Context: can access all smart contracts' public data state, executing transaction's context and information about recent blocks
- Decentralized World Computer: EVM runs as a local instance on every node. All nodes combined to form a single "world computer"

SMART CONTRACT ADDRESS

- Each deployed smart contract has an address calculated via opcodes:
- CREATE (old version) or
- CREATE2 (Can be pre-calculated): keccak256(0xff, deployerContractAddress, salt, keccak256(init_code)) [12:]

- No private keys associated
- Smart Contract Creator receives no special privileges at protocol level

SMART CONTRACT DELETION

- Smart contract code can be deleted via SELFDESTRUCT opcode (must be manually coded in)
- Code is removed from internal state storage from its address
- Deletion does not remove its transaction history
- All smart contract's ETH will be sent to a specified address
- Deletion transaction sender will receive gas refund

ABI

- Application Binary Interface, a bridge between OS and user programs (2 program modules)
- Defines how functions and data structures are accessed in machine code
- Defines how to encode and decode data from machine code

EVM ABI

- Each smart contract has an associated ABI
- Encodes smart contract calls and read transaction call data
- A smart contract's ABI is represented as a JSON array of function descriptions and events

SMART CONTRACT DEPLOYMENT

Compile Smart Contract into Bytecode and Opcodes.

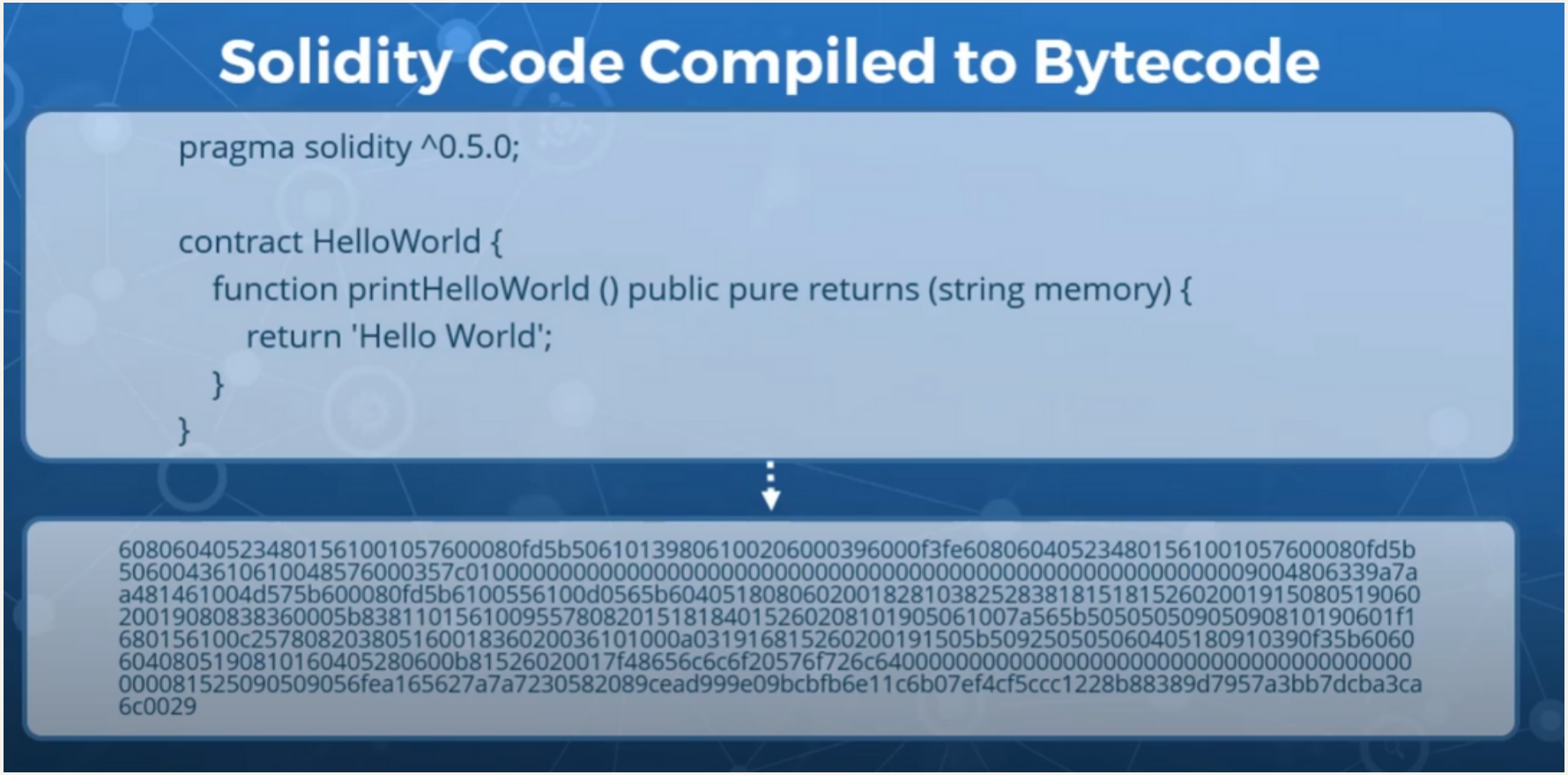
↓

Using an EOA, submit a special “Contract Creation" transaction with the Bytecode to address 0x0...0. Wait for it to be validated.

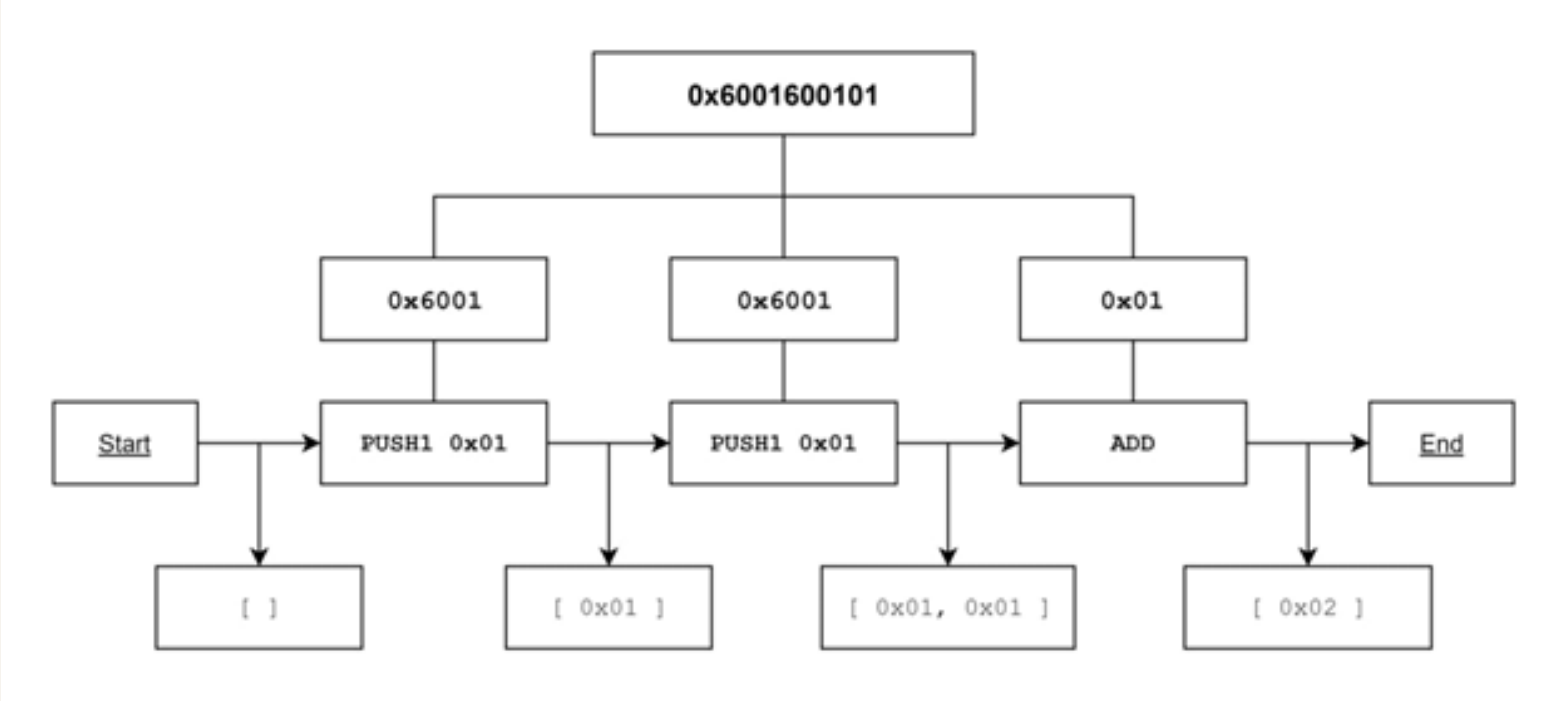
↓

[Optional] Verify the Smart Contract on a block explorer (most popular is Etherscan)

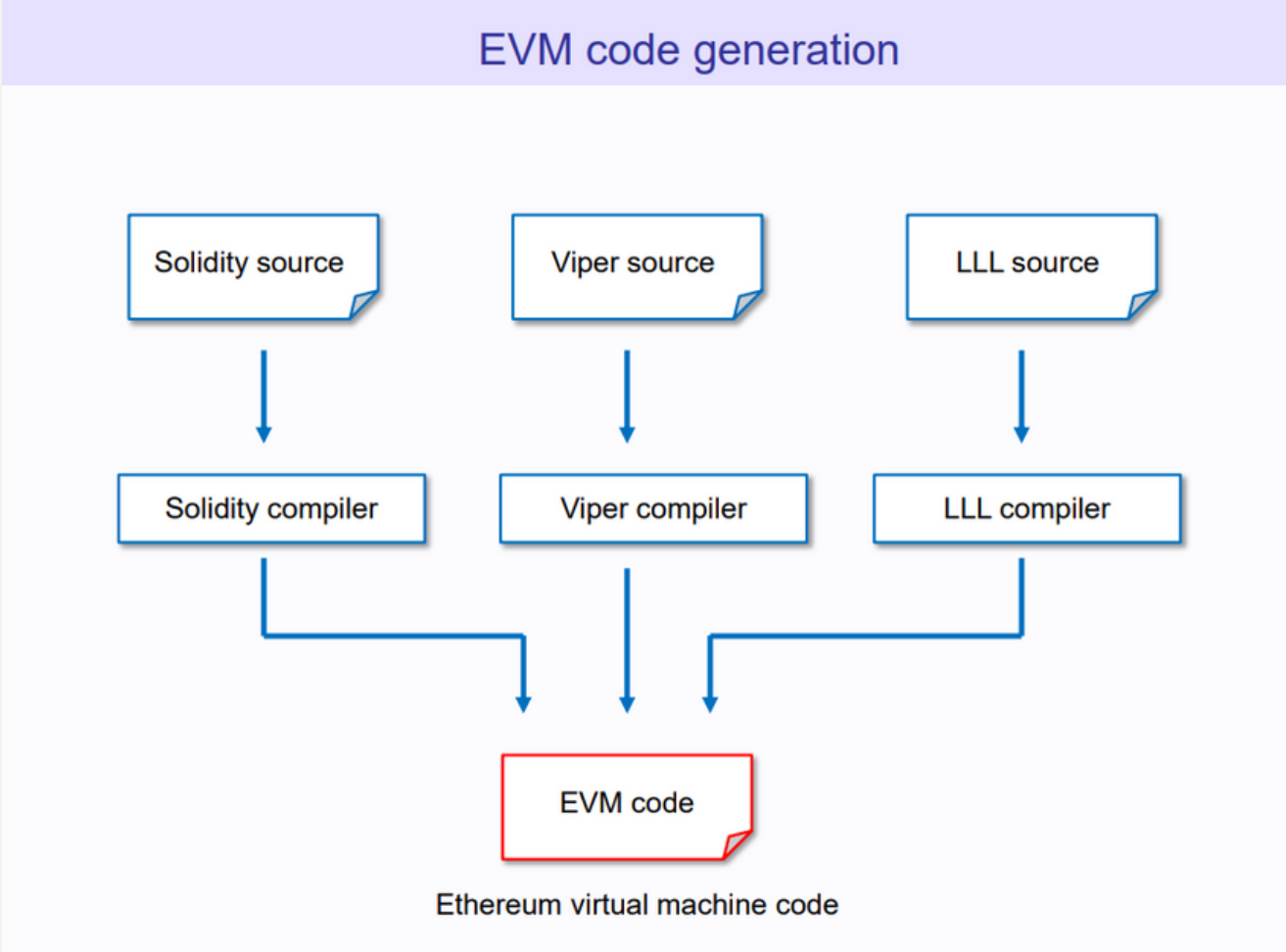
Special Bytecode called EVM Bytecode



PARSING BYTECODE INTO OPCODES



EVM SMART CONTRACT LANGAUGES

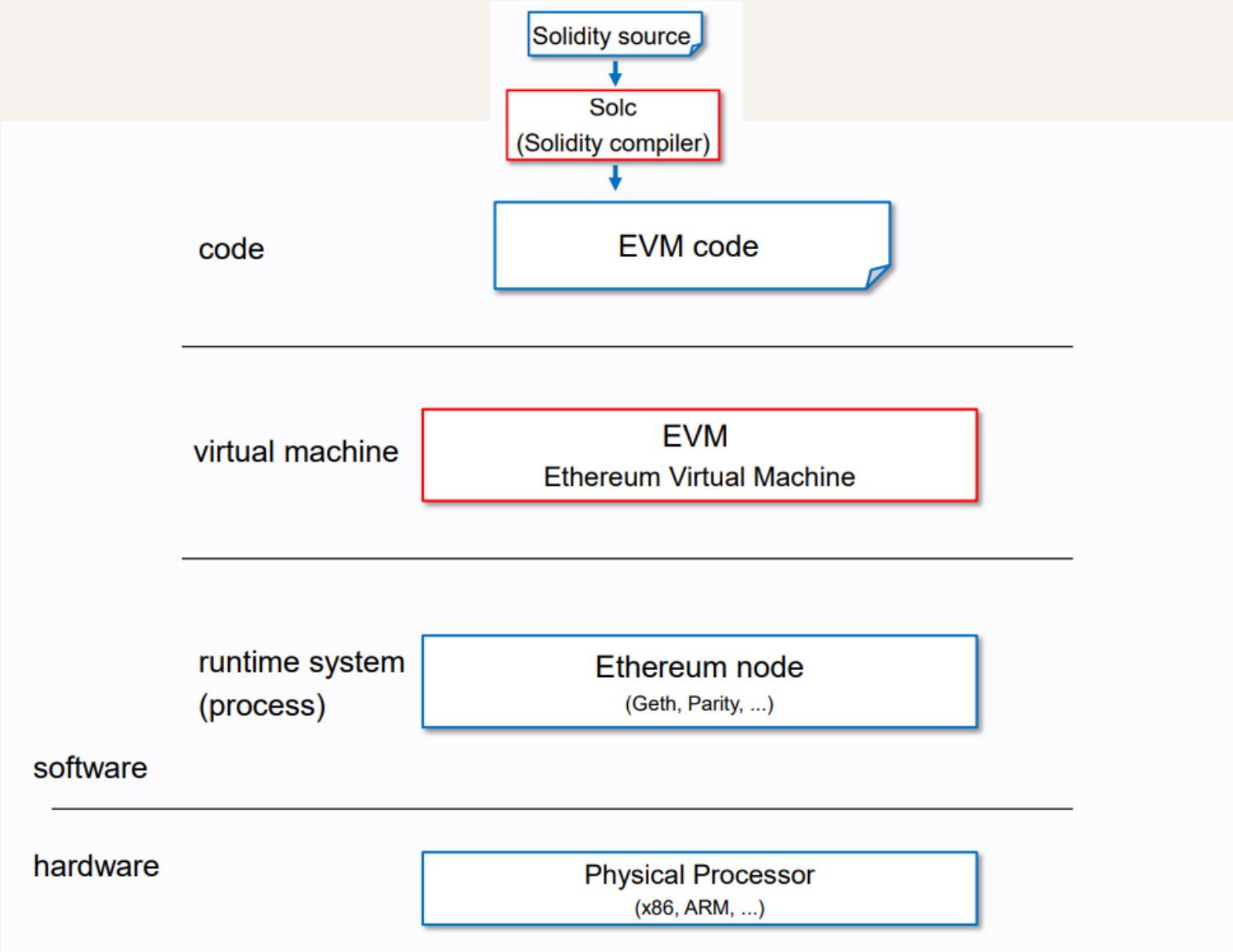


EVM SMART CONTRACT LANGAUGES

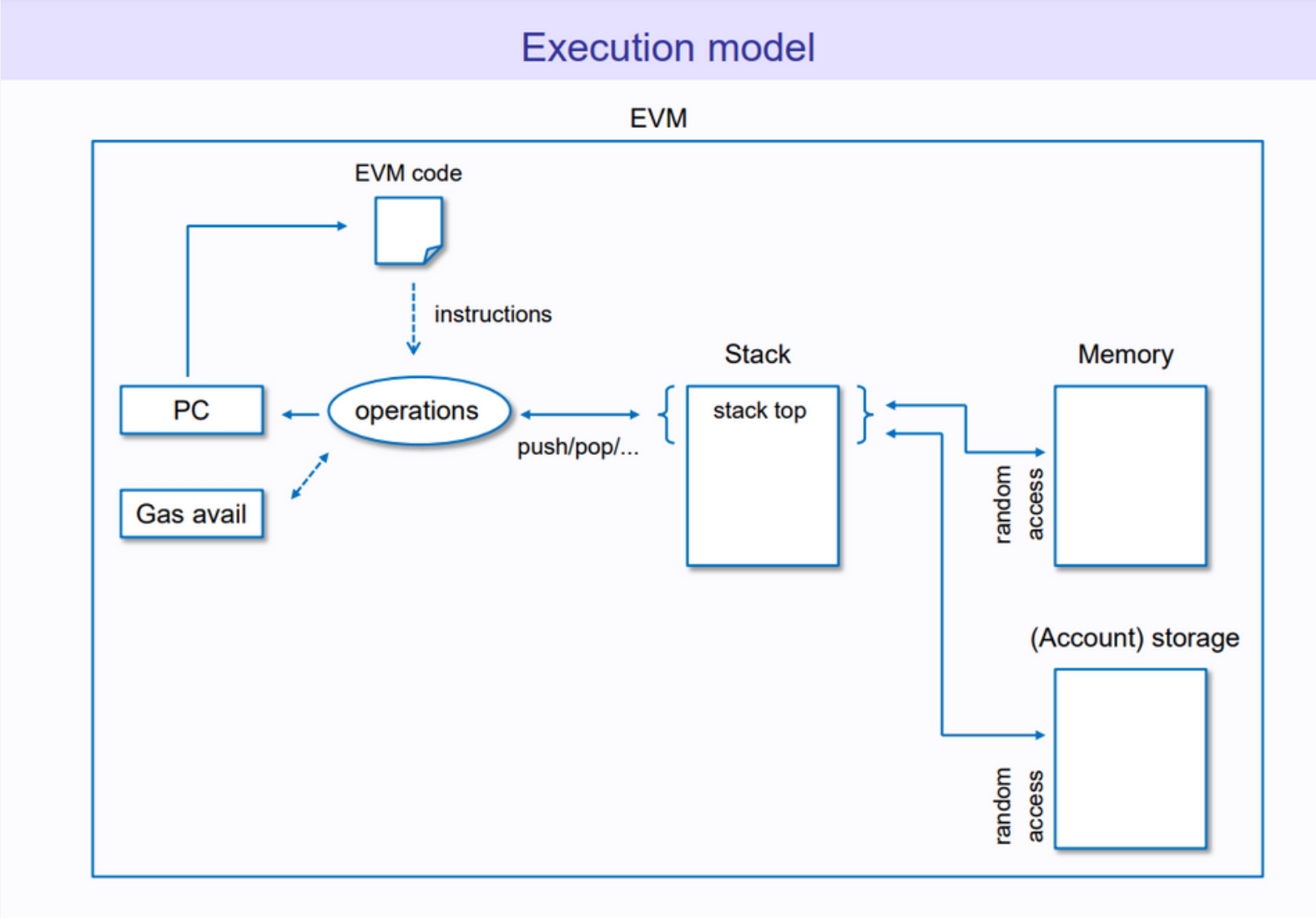
- Declarative Language: functions express the logic of a program but not its flow. There are no changes of state outside of a function. Ex: Haskell, SQL
- Imperative Language: functions combine logic and flow of a program. Ex: Java, C++
- Hybrid: combination of above. Ex: Javascript, Python

- LLL: Declarative Language, Lisp-like syntax, rarely used
- Solidity: Imperative Language, Javascript + Java like syntax, widely used (Course will use Solidity)
- Vyper: Imperative Language, Python like syntax, moderately used

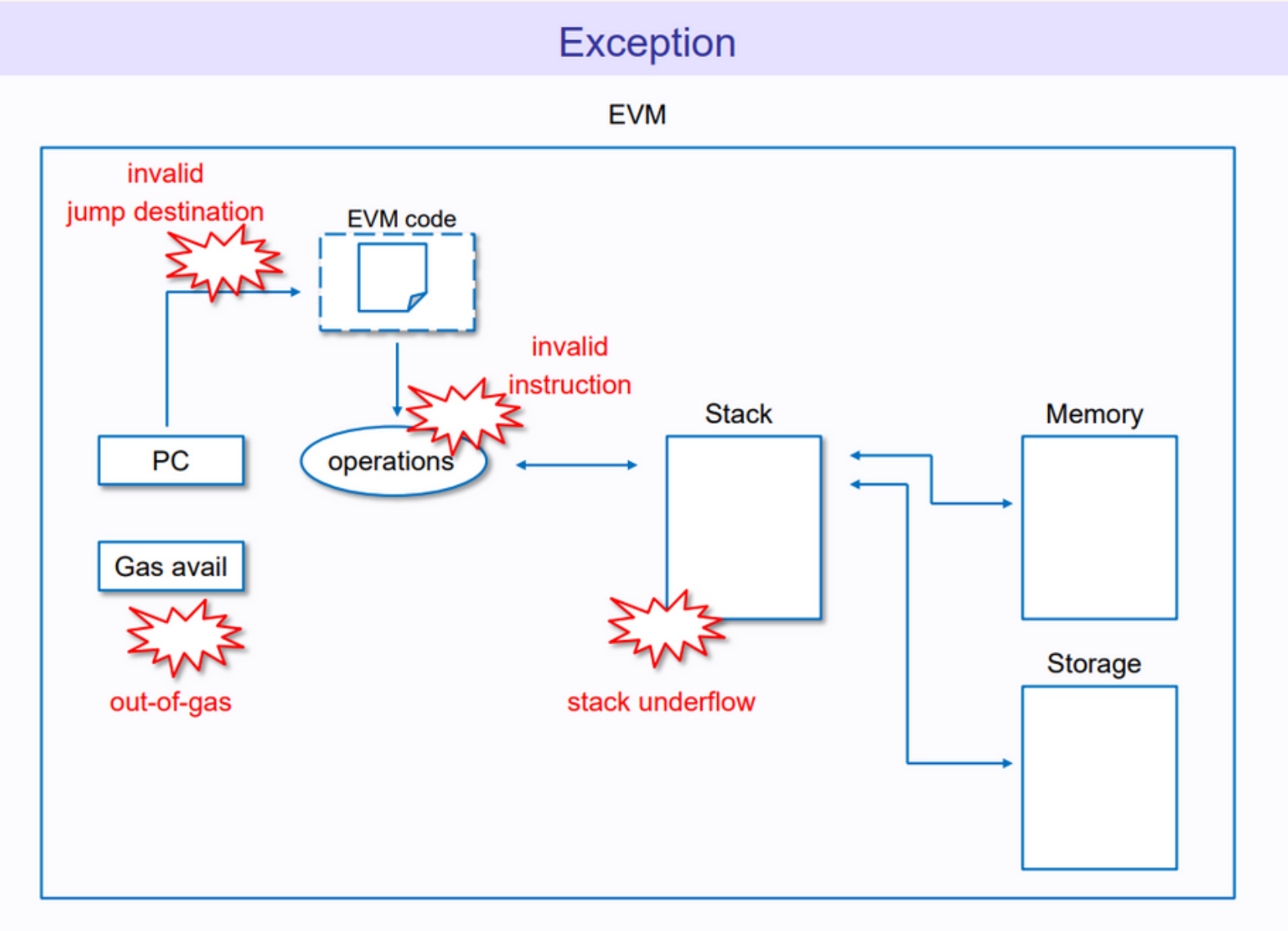
CODE EXECUTION



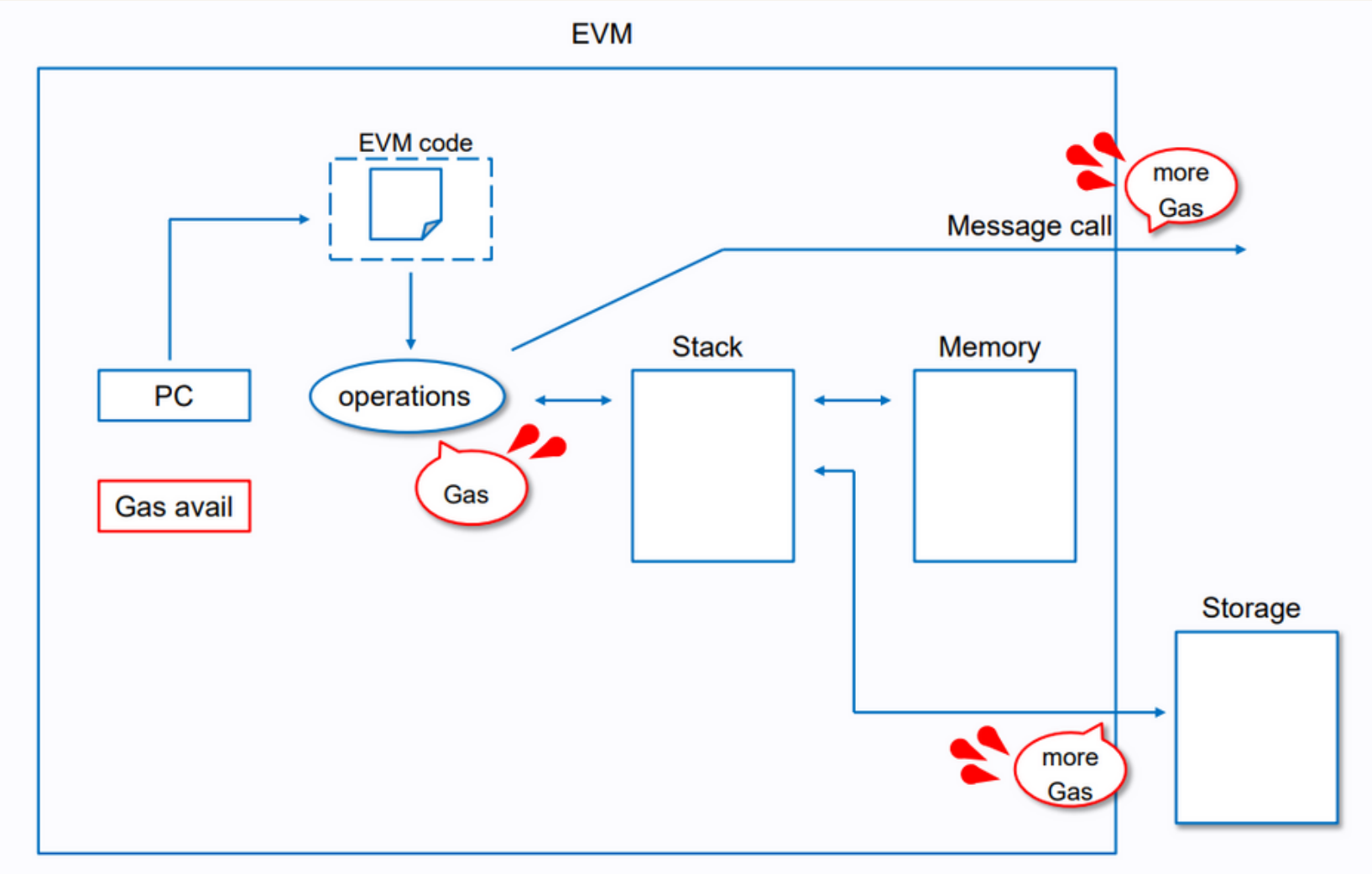
EVM


Execution model
EVM
EVM code
instructions
Stack
Memory
PC
operations
stack top
push/pop/...
Gas avail
random access
(Account) storage
random access

EVM

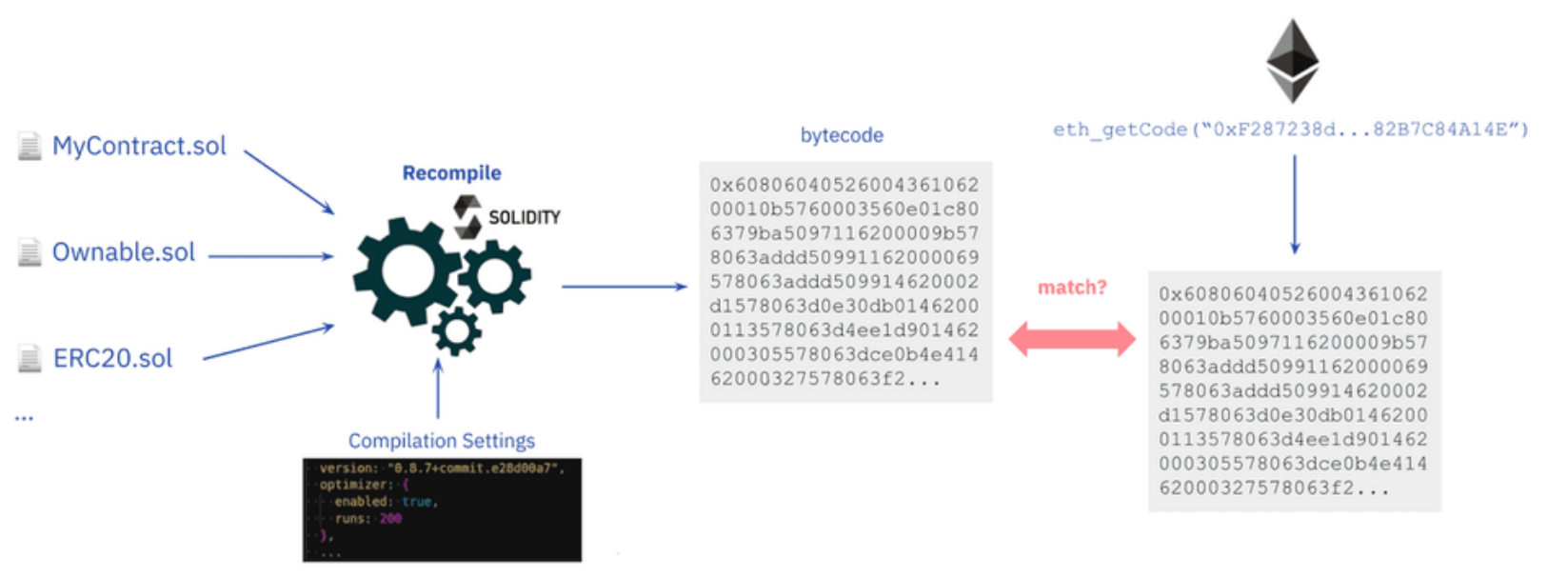

Exception
EVM
invalid
jump destination
EVM code
invalid
instruction
Stack
Memory
PC
operations
Gas avail
out-of-gas
stack underflow
Storage

EVM


EVM
EVM code
more
Gas
Message call
Stack
Memory
PC
operations
Gas
Gas avail
Storage
more
Gas

UNVERIFIED SMART CONTRACT

Verifying is done by sending the data used in the Contract Creation Transaction, source code of the smart contract and smart contract metadata to the Block Explorer. The Block Explorer will then compare the bytecode to the deployed smart contract.

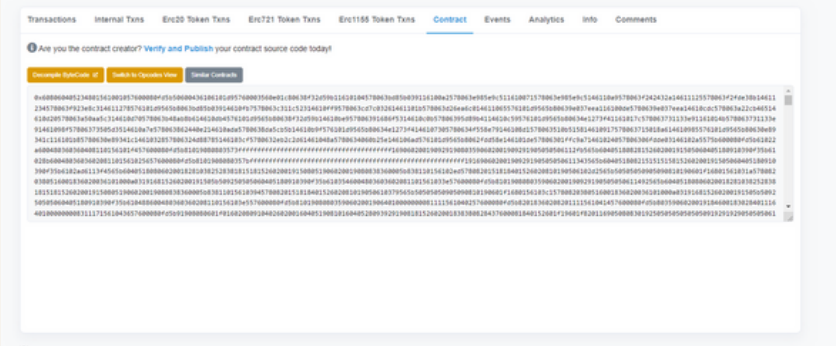


UNVERIFIED SMART CONTRACT

Metadata file


```
{
  "compiler": { "version": "0.8.4+commit.c7e474f2" },
  "language": "Solidity",
  "output": {
    // Application Binary Interface (ABI) describing how to interact with the contract,
    // and what functions and parameters are available.
    "abi": [
      {
        "inputs": [],
        "name": "retrieve",
        "outputs": [
          { "internalType": "uint256", "name": "", "type": "uint256" }
        ],
        "stateMutability": "view",
        "type": "function"
      },
      {
        "inputs": [
          { "internalType": "uint256", "name": "num", "type": "uint256" }
        ],
        "name": "store",
        "outputs": [],
        "stateMutability": "nonpayable",
```


Unverified Bytecode



Metadata appended to Bytecode

Bytecode of 0x00878Ac0D6B8d981ae72BA7cDC967eA0Fae69df4 (Görli)


```
6080604052348015610010576000080fd5b5061012f80610020600039600 0f3fe6080604052348015600f57600080fd5b5060043610603257600035 60e01c80632e64cec11460375780
636057361d146051575b600080fd5b603d6069565b60405160489190 60c2565b60405180910390f35b6067600480360381019060639190608f565b6072565b005b6000805490509056
5b8060008190555050565b600081359050606898160e5565b92915050565b600060208284031215 60a057600080fd5b600060ac84828501607c565b91505092915050565b60bc8160db
565b82525050565b600060208201905060d5600083018460b5565b92915050565b600081905091905 0565b60ec8160db565b811460f657600080fd5b5056fea2646970667358221220
c019e4614043d8adc295c3046ba5142c603ab309adeef171f330c51c38f1498964736f6c63430008040033
```

VERIFIED SMART CONTRACT

Transactions | Internal Txns | Erc20 Token Txns | Erc721 Token Txns | Contract | Events | Analytics | Comments

Code | Read Contract | Write Contract

Search Source Code

Similar Match Source Code

Note: This contract matches the deployed ByteCode of the Source Code for Contract 0x62051ee320385e5d8b...

Contract Name:	RaribleUserToken	Optimization Enabled:	Yes with 200 runs
Compiler Version	v0.5.17+commit.d19bba13	Other Settings:	istanbul EvmVersion, MIT license

Contract Source Code (Solidity)

Outline | More Options

```
/**
 *Submitted for verification at Etherscan.io on 2020-06-03
 */

pragma solidity ^0.5.0;
pragma experimental ABIEncoderV2;


/**
 * @title SafeMath
 * @dev Math operations with safety checks that throw on error
 */
library SafeMath {

    /**
    * @dev Multiplies two numbers, throws on overflow.
    */
    function mul(uint256 a, uint256 b) internal pure returns (uint256 c) {
        // Gas optimization: this is cheaper than asserting 'a' not being zero, but the
        // benefit is lost if 'b' is also tested.
        // See: https://github.com/OpenZeppelin/openzeppelin-solidity/pull/522
        if (a == 0) {
            return 0;
        }

```

Contract Security Audit

- No Contract Security Audit Submitted - Submit Audit Here

Contract ABI

Export ABI


```
[{"inputs":[{"internalType":"string","name":"name","type":"string"},{"internalType":"string","name":"symbol","type":"string"},{"internalType":"string","name":"contractURI","type":"string"},
{"internalType":"string","name":"tokenURIPrefix","type":"string"},
{"internalType":"address","name":"signer","type":"address"}],"payable":false,"stateMutability":"nonpayable","type":"constructor"},{"anonymous":false,"inputs":
[{"indexed":true,"internalType":"address","name":"_owner","type":"address"},{"indexed":true,"internalType":"address","name":"_operator","type":"address"},
{"indexed":false,"internalType":"bool","name":"_approved","type":"bool"}],"name":"ApprovalForAll","type":"event"},{"anonymous":false,"inputs":
[{"indexed":true,"internalType":"address","name":"creator","type":"address"},{"indexed":false,"internalType":"string","name":"name","type":"string"},
{"indexed":false,"internalType":"string","name":"symbol","type":"string"}],"name":"CreateERC1155_v1","type":"event"},{"anonymous":false,"inputs":
[{"indexed":true,"internalType":"address","name":"previousOwner","type":"address"},
{"indexed":true,"internalType":"address","name":"newOwner","type":"address"}],"name":"OwnershipTransferred","type":"event"},{"anonymous":false,"inputs":
[{"indexed":false,"internalType":"uint256","name":"tokenId","type":"uint256"},{"indexed":false,"internalType":"address[]","name":"recipients","type":"address[]"},
```


Contract Creation Code

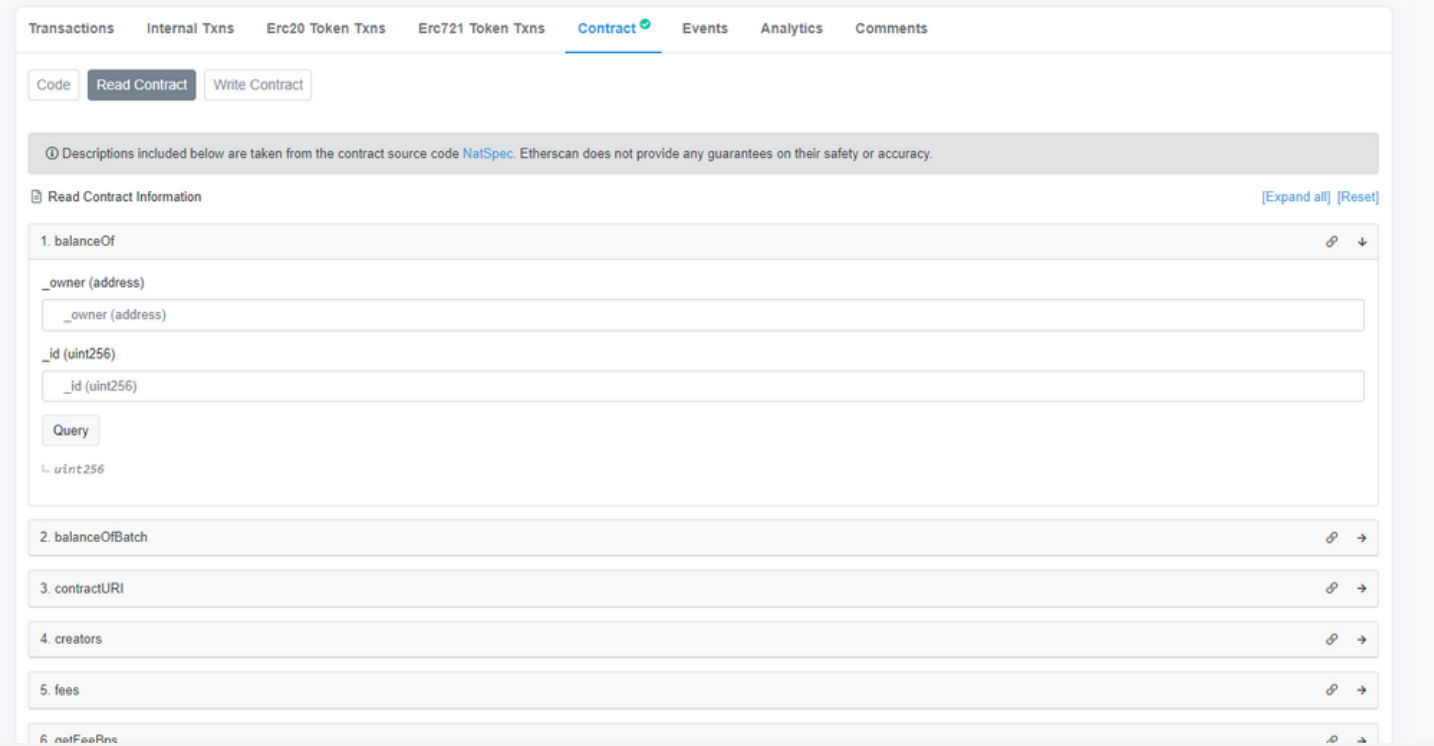
Decompile ByteCode | Switch to Opcodes View


```
6080604052348015620000115760008060fd5b5060405162003159e38038062003159e833981016040819052620000349162000489565b8484828585818181818062000059630 1ffc9a760e01b6001600160e01b03620002231656 5b62000074632
ddc656160e21b6001600160e01b03620002231656 5b60006200008960016001600160e01b03620002 7c16565b6001805460016001600160a01b0319166001600160a01b0383169081179091556040519192509060000907f8bc0079c531659141344cd1f
d8a4f28419497f9722a3daafe3b4186f6b6457e0908290a350620000fd620000ce6001600160e01b03620002 7c16565b6001600160e01b03620002831656 5b8051620001129060039060020840190620003 7e565b505081516200012a9150600
59060208401906200037e565b5062000146 63e8a3d48560e01b6001600160e01b03620002231656 5b5062000162636cdb3d1360e11b6001600160e01b03620002231656 5b505084516200017990600a90602088019062000037e565b50835162
00018f906000b90602087019062000037e565b5062000 1a483600160016 0e01b03620002831656 5b620001ce604051620001b690620006931656 5b604051908190039020600160016 0e01b03620002231656 5b505050505033600160016 0a01b031
67f658fd9a983a35f4a0b0697abb2c6971d688810d8bc2649828a164ae391b87472c868660405162000210929190620006a0565b604051809103908a250505050506200087d1565b6001600160e01b03198082161415620002595760405162461b
cd60e51b815260040162000250906200006e3565b604051809103908a250505050506200087d1565b6001600160e01b03198082161415620002595760405162461b
c565b604051600160016 0e01b038216987f47d1c22a25bb3a5d4e481b9b1c6944e2eadc3181a0a28b495cd61d35b5323f2490600090a250565b620002ca828260016 0e01b03620003 2f16565b156200030a576040516 2461bcd60e51b81
526004016200025090620006d1565b600160016 0a01b0316600090815260209190915260409020885460ff191660011790555565b600060016 0160a01b0383166200035a576040516 2461bcd60e51b815260040162000250906200006f5565b5
060016 0160a01b0383166600090815260208390526040902085460ff165b929150505655b8280546001816001161561010002031666002900490600052602060002090601f016020900481019282601f10620003c157805160ff191638800117 85
```


Deployed ByteCode Sourcemap

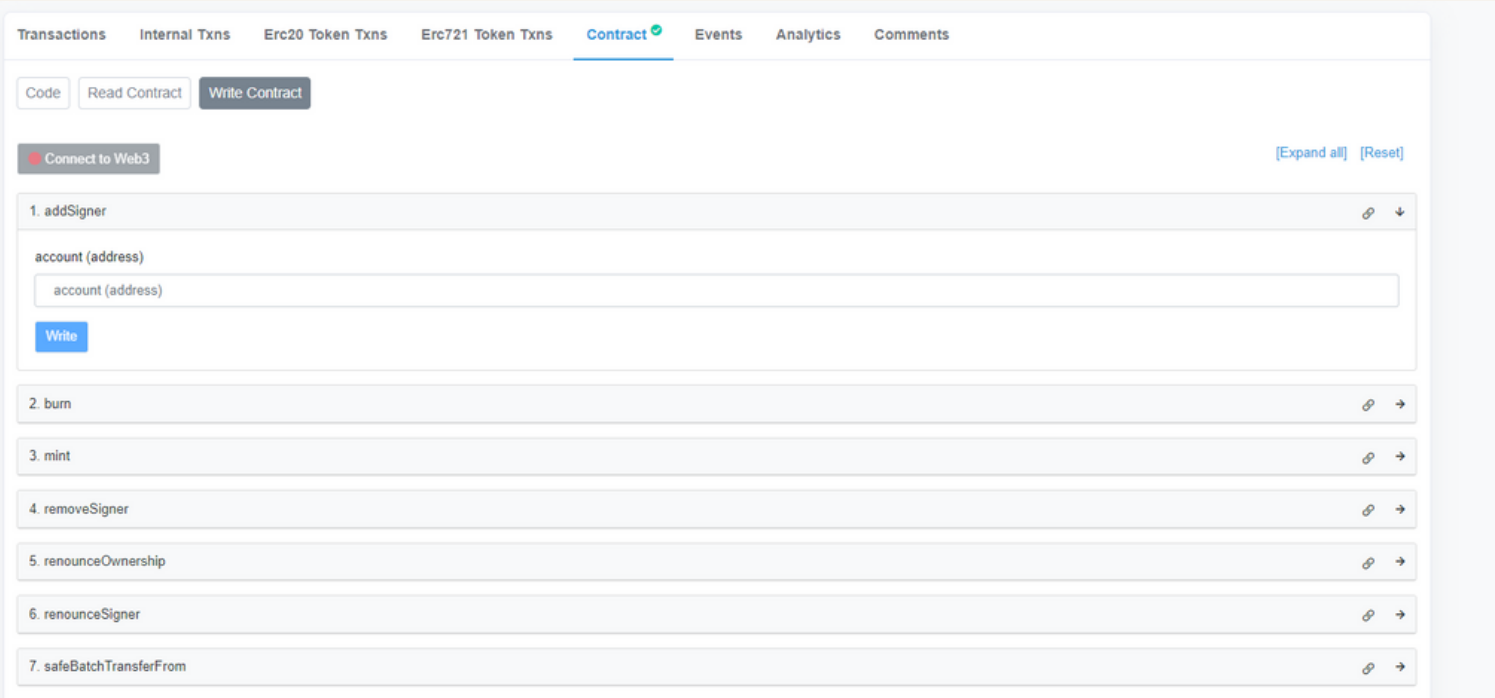
READ DATA FROM SMART CONTRACT

Reading from a verified Smart Contract on Etherscan



WRITING DATA TO SMART CONTRACT

Writing to a verified Smart Contract on Etherscan



SMART CONTRACT PLANNING

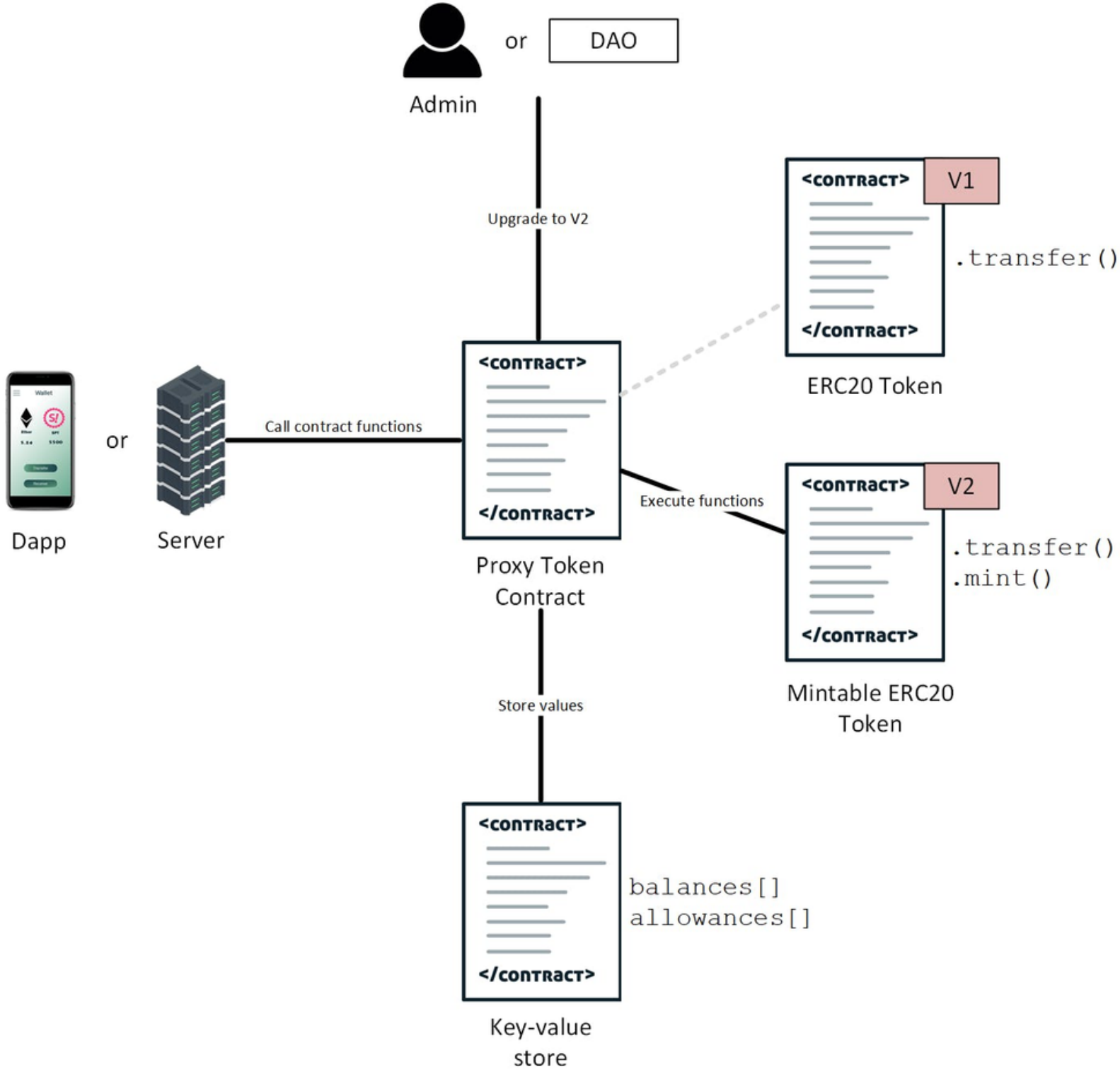
Smart contracts cannot be changed after it has been deployed!

They should be thoroughly planned out before coding.

SMART CONTRACT UPGRADABLE

Actually, there's a work around. We can have a proxy smart contract the maintains code versions via another smart contract.

Certain rules apply, and there a limit on what could be changed.


or
DAO
Admin
Upgrade to V2
<CONTRACT>
V1
.transfer()
</CONTRACT>
ERC20 Token
Dapp
or
Server
Call contract functions
<CONTRACT>
</CONTRACT>
Proxy Token Contract
Execute functions
<CONTRACT>
V2
.transfer()
.mint()
</CONTRACT>
Mintable ERC20 Token
Store values
<CONTRACT>
balances[]
allowances[]
</CONTRACT>
Key-value store

SMART CONTRACT PROPERTIES

Property	Description
Native Token Balance	ETH owned by the smart contract
Data State	Data stored in the variables via transactions
Code	Low Level compiled bytecode that could be executed
...	...

EXAMPLE SMART CONTRACT

```
// SPDX-License-Identifier: UNLICENSED
pragma solidity ^0.8.9;

// Import this file to use console.log
import "hardhat/console.sol";

contract Lock {
    uint public unlockTime;
    address payable public owner;

    event Withdrawal(uint amount, uint when);

    constructor(uint _unlockTime) payable {
        require(
            block.timestamp < _unlockTime,
            "Unlock time should be in the future"
        );

        unlockTime = _unlockTime;
        owner = payable(msg.sender);
    }

    function withdraw() public {
        console.log("Unlock time is %o and block timestamp is %o", unlockTime, block.timestamp);

        require(block.timestamp >= unlockTime, "You can't withdraw yet");
        require(msg.sender == owner, "You aren't the owner");

        emit Withdrawal(address(this).balance, block.timestamp);

        owner.transfer(address(this).balance);
    }
}
```

EIP

"Ethereum Improvement Proposals (EIPs) describe standards for the Ethereum platform, including core protocol specifications, client APIs, and contract standards."

-https://eips.ethereum.org/

EIP TYPES

EIPs are separated into a number of types, and each has its own list of EIPs:

- Standard Track (500)
- Core (189)
- Networking (13)
- Interface (42)
- ERC (256): Ethereum request for comment. These are application-level standards and conventions, including contract standards such as token standards (ERC20), and name registries (ERC137).
- Meta (18)
- Informational (6)

TOKENS

Many tokens are created using the ERC Smart Contract standards. Tokens can represent ownership of currencies or digital assets.

3 popular types of tokens:

- Fungible: Each commodity has the same value (Ex. Fiat Currency)
- Non-Fungible (NFT): Each commodity is unique (Ex. Driver's License)
- Semi-Fungible: Each set of commodity is unique (Ex. Pokemon Cards consisting of 5 Pikachus and 10 Charzards)

These tokens have widely accepted ERC standards:

- ERC20: Fungible (Ex. Any token on Uniswap, except ETH)
- ERC721: NFT (Any token on Foundation NFT)
- ERC1155: Semi-Fungible, also known as "NFT" by the general community (Ex. Some tokens on Opensea)

ERC20

```
pragma solidity ^0.4.24;

/**
 * @title ERC20 interface
 * @dev see https://github.com/ethereum/EIPs/issues/20
 */
interface IERC20 {
  function totalSupply() external view returns (uint256);

  function balanceOf(address who) external view returns (uint256);

  function allowance(address owner, address spender)
    external view returns (uint256);

  function transfer(address to, uint256 value) external returns (bool);

  function approve(address spender, uint256 value)
    external returns (bool);

  function transferFrom(address from, address to, uint256 value)
    external returns (bool);

  event Transfer(
    address indexed from,
    address indexed to,
    uint256 value
  );

  event Approval(
    address indexed owner,
    address indexed spender,
    uint256 value
  );
}
```

ERC721

```
pragma solidity ^0.4.24;

import "../../introspection/IERC165.sol";

/**
 * @title ERC721 Non-Fungible Token Standard basic interface
 * @dev see https://github.com/ethereum/EIPs/blob/master/EIPS/eip-721.md
 */
contract IERC721 is IERC165 {

  event Transfer(
    address indexed from,
    address indexed to,
    uint256 indexed tokenId
  );
  event Approval(
    address indexed owner,
    address indexed approved,
    uint256 indexed tokenId
  );
  event ApprovalForAll(
    address indexed owner,
    address indexed operator,
    bool approved
  );

  function balanceOf(address owner) public view returns (uint256 balance);
  function ownerOf(uint256 tokenId) public view returns (address owner);

  function approve(address to, uint256 tokenId) public;
  function getApproved(uint256 tokenId)
    public view returns (address operator);

  function setApprovalForAll(address operator, bool _approved) public;
  function isApprovedForAll(address owner, address operator)
    public view returns (bool);

  function transferFrom(address from, address to, uint256 tokenId) public;
  function safeTransferFrom(address from, address to, uint256 tokenId)
    public;

  function safeTransferFrom(
    address from,
    address to,
    uint256 tokenId,
    bytes data
  )
    public;
}
```

ERC1155

```
// SPDX-License-Identifier: MIT

pragma solidity >=0.6.2 <0.8.0;

import "../../introspection/IERC165.sol";

/**
 * @dev Required interface of an ERC1155 compliant contract, as defined in the
 * https://eips.ethereum.org/EIPS/eip-1155[EIP].
 *
 * _Available since v3.1._
 */
interface IERC1155 is IERC165 {

    event TransferSingle(address indexed operator, address indexed from, address indexed to, uint256 id, uint256 value);

    event TransferBatch(address indexed operator, address indexed from, address indexed to, uint256[] ids, uint256[] values);

    event ApprovalForAll(address indexed account, address indexed operator, bool approved);

    event URI(string value, uint256 indexed id);

    function balanceOf(address account, uint256 id) external view returns (uint256);

    function balanceOfBatch(address[] calldata accounts, uint256[] calldata ids) external view returns (uint256[] memory);

    function setApprovalForAll(address operator, bool approved) external;

    function isApprovedForAll(address account, address operator) external view returns (bool);

    function safeTransferFrom(address from, address to, uint256 id, uint256 amount, bytes calldata data) external;

    function safeBatchTransferFrom(address from, address to, uint256[] calldata ids, uint256[] calldata amounts, bytes calldata data) external;
}
```

KNOWING THE ECOSYSTEM

In blockchain, using deployed smart contract code and platform standards is encouraged because:

1. Avoid redundant development work
2. Security Audits and Testing already done
3. Lower learning curve for both Users and Developers due to similar Smart Contract APIs
4. Incorporate the larger amount of users and assets held by established dApps into a new dApp

KNOWING THE ECOSYSTEM

DeFi Example:

- ERC20
- Uniswap provides liquidity and facilitates trades for ERC20 tokens
- Web3 Startups make staking pools to incentivize people to provide liquidity to Uniswap for their token
- Staking Aggregators auto compound the staking rewards to earn a high APY
- Web3 Insurance offers automatic payout for Staking Aggregator smart contract hacks

SOLIDITY

- Strongly Typed
- Object Oriented (Inperitive)
- Similar to JavaScript and Java
- filename.sol
- Current Version (Sept 2023): 0.8.21

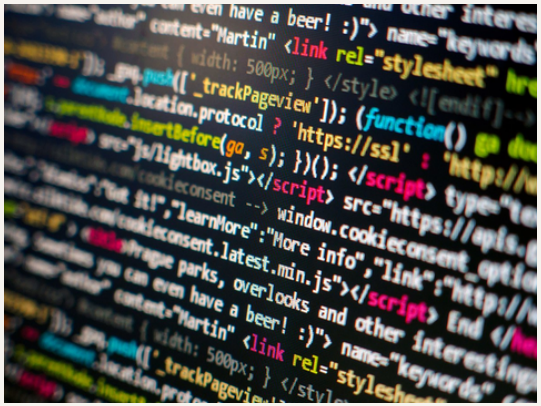
SOLIDITY VERSION

- Semantic Versioning: MAJOR.MINOR.PATCH
- Major: breaking changes
- Minor: backwards compatible changes
- Patch: backwards compatible bug fixes

- Versioning is for choosing “solc” version
- The bare min dev setup is a “solc” and a text editor

FUNCTION VISIBILITY AND TYPES

external	Function can only be called from other contracts
internal	Function can only be called from the current contract
private	Same as internal but additional not visible to derived contracts
public	Function can be called internally or externally
view	Additional Type. Function does not write data and only returns data
pure	Additional Type. Function does not read or write data, only returns data
payable	Additional Type. Function call may also have Ether attached
virtual	Additional Type. Function overrides an inherited function.



MODIFIERS

```
contract Mutex {
    bool locked;
    modifier noReentrancy() {
        require(
            !locked,
            "Reentrant call."
        );
        locked = true;
        _;
        locked = false;
    }

    function f() public noReentrancy returns (uint) {
        (bool success,) = msg.sender.call("");
        require(success);
        return 7;
    }
}
```

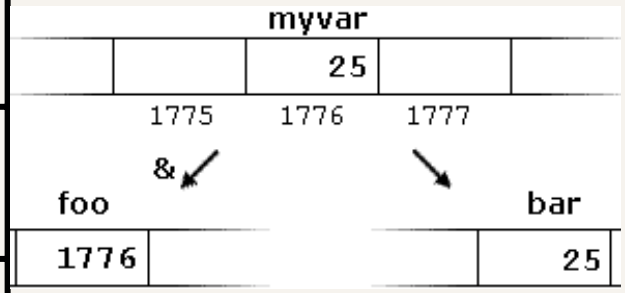
When a function uses a modifier, the functions code will be modified, as if the code is moved to where the _ is

VARIABLE VISIBILITY AND TYPES

public	Data can be accessed externally and internally. Also, getters are auto generated
internal	Data can only be accessed within the contract. NOTE: Data is still visible.
private	Same as internal but not visible in derived contracts. NOTE: Data is still visible.
string	A list of characters
bool	true or false
int	Positive or negative number. No Decimal
uint	Positive number. No Decimal
address	Unique identifier for accounts and contracts
enum	User defined type

REF TYPES

arrays	Fixed or dynamic sized lists
structs	User defined type
mapping	Hashmap with bytes, strings or enum as keys to any value
uint	Positive number. No Decimal
address	Unique identifier for accounts and contracts
enum	User defined type
bytes	bytes array
struct	user defined data containers



BUILT-IN FUNCTIONS

keccak256, sha256, sha3, ripemd160	hashing algorithms
ecrecover	recover the signing address from a signature
this	reference to the executing smart contract
selfdestruct	delete the executing contract and sending away containing ETH

TIME UNITS

NOTE: TIME IS STAGGERED BY BLOCK INCLUSION AND IS NOT CONTINUOUS

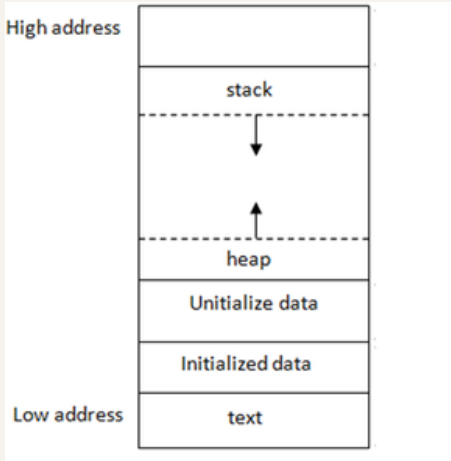
seconds	1
minutes	60 seconds
hours	3600 seconds
days	86400 seconds

ETHER UNITS

wei	1
szabo	10**12 wei
finney	10**16 wei
ether	10**18 wei
1eX	1**X

DATA LOCATION

Memory	Temporary store, lifetime is limited to a function call
Storage	Global data store
Calldata	Same as Memory but can only be used in function parameter declaration



MESSAGE CONTEXT

msg.sender	Address that called the smart contract function. Can be EOA or Smart Contract address
msg.value	Amount of Ether attached (valued in wei)
msg.data	Data payload of the smart contract call
msg.sig	The function selector, which is the 1st 4 bytes of data payload
msg.gas	Remaining gas supply. DEPRECATED, replaced with gasleft()

TRANSACTION CONTEXT

tx.gasprice	gas price in the calling transaction
tx.origin	address of original EOA for the initiating transaction. UNSAFE

BLOCK CONTEXT

block.coinbase	Recipient address of the current block's reward fee
block.randao	Random number generated by Beacon chain
block.gaslimit	Max gas all transactions in a block can spend
block.number	Current block number
block.timestamp	Timestamp of when the current block is added to the blockchain

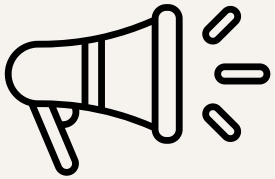
ERROR HANDLING

assert	Revert if given statement is false
require	Revert if the given statement is false, with an optional error message
revert	Revert the execution

Overview | Internal Txns | State | Comments

Transaction Hash:	0x67ec3acc5274a88c50d1e79e9b9d4c2c3d5e0e3ba3cc33b32d65f3fdb3b5a258
Status:	Fail
Block:	5602146 12739659 Block Confirmations
Timestamp:	1979 days 19 hrs ago (May-12-2018 06:33:58 PM +UTC)
Method:	Quick Convert
From:	0xfBd28a75d7593CC9b934878673a1BFc13831ae6f
To:	0xc6725aE749677f21E4d8f85F41cFB6DE49b9Db29 (Bancor: Converter #3) Warning! Error encountered during contract execution [Reverted]
Value:	0.073619703694185916 ETH $114.14 - [CANCELLED]
Transaction Fee:	0.002209125 ETH $3.43
Gas Price:	5 Gwei (0.000000005 ETH)

EVENTS



- Similar to calldata, stored on chain as a history record, but not as state data
- Used to announce data in a pub/sub like model

Overview

ETH BALANCE
0 ETH

ETH VALUE
$0.00

TOKEN HOLDINGS
$0.28 (11 Tokens)

More Info

PRIVATE NAME TAGS
+ Add

CONTRACT CREATOR
Bancor: Deployer 2 at txn 0xa7b634d766900a58...

Multi Chain

MULTICHAIN ADDRESSES
2 addresses found via Blockscan

Transactions | Internal Transactions | Token Transfers (ERC-20) | Contract | Events | Analytics | Comments

Latest 25 Contract Events

Tip: Logs are used by developers/external UI providers for keeping track of contract actions and for auditing

Txn Hash	Block	Age	Method	Logs
0x47729b9059924327...	5769020	1950 days 6 hrs ago	0xf0843ba9 quickConvert(address[...	Conversion (index_topic_1 address _fromToken, index_topic_2 address _toToken, index_topic_3 address [topic0] 0xcee13e282037fd063de72c4cf7955988112510b046cda38c12c47862c1785e7b [topic1] 0x000000000000000000000000c0829421c1d260bd3cb3e0f06cfe2d52db2ce315 [topic2] 0x000000000000000000000000001f573d6fb3f13d689ff844b4ce37794d79a7ff1c [topic3] 0x000000000000000000000000f20b9e713a33f61fa38792d2afaf1cd30339126a Hex → 000000000000000000000000000000000000000000000000000023d175255bd32 Hex → 000000000000000000000000000000000000000000000000014b729d1043356e Hex → 0000000000000000000000000000000000000000000000000000000000000000 Hex → 000000000000000000000000000000000000000a5b51eb300c18dc5889c0980 Hex → 0000000000000000000000000000000000000005fd64a1a1a1b233f1bf40d75e0

CALLING OTHER CONTRACTS

Instance Call	Call a reference instance of the smart contract
call	Low level call function to customize message gas, error handling and other parameters. msg.sender is changed to the address of the calling smart contract
delegatecall	Same as call, but the execution context is the calling contract

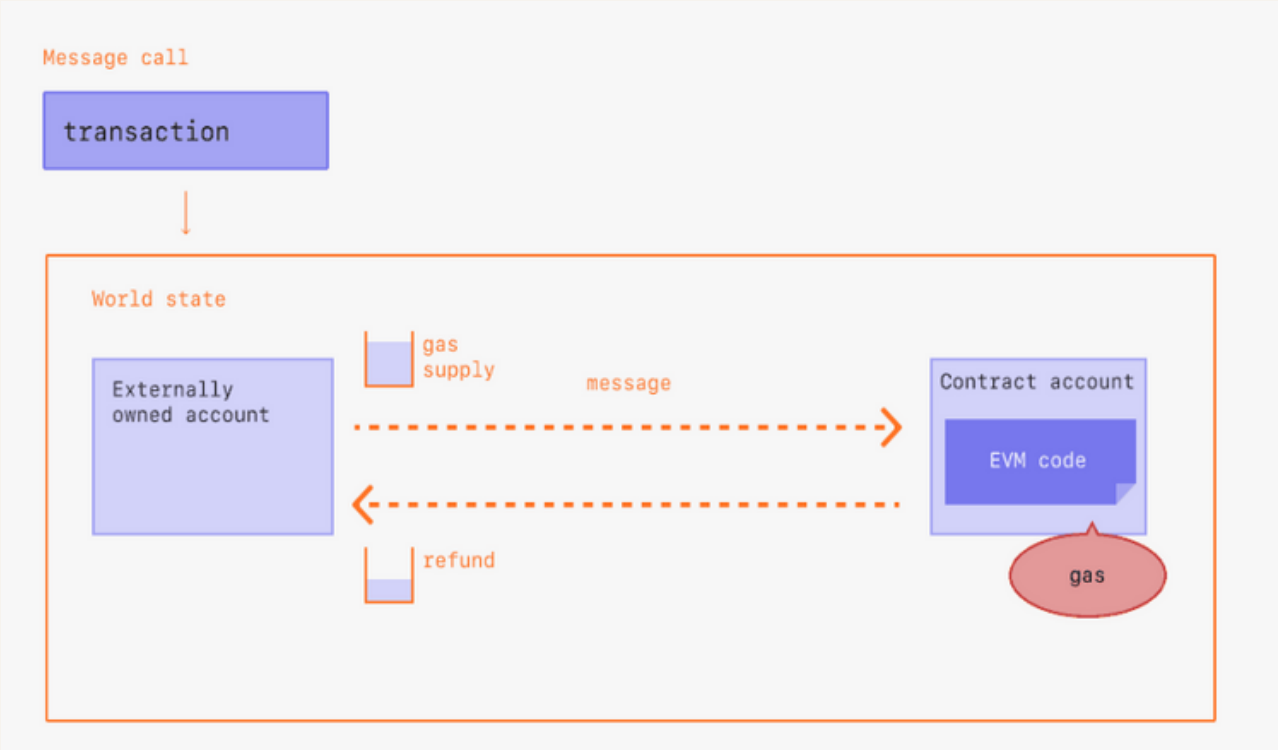
TIPS ON SAVING GAS

- avoid dynamically sized arrays
- avoid calls to other contracts
- minimize blockchain data updates
- estimate gas costs

GAS COSTS

Hex	Name	Gas	Stack	Mem / Storage
			top, bottom	
00	STOP	0		
01	ADD	3	`a, b` => `a + b`	
02	MUL	5	`a, b` => `a * b`	
03	SUB	3	`a, b` => `a - b`	
04	DIV	5	`a, b` => `a // b`	
05	SDIV	5	`a, b` => `a // b`	
06	MOD	5	`a, b` => `a % b`	
07	SMOD	5	`a, b` => `a % b`	
08	ADDMOD	8	`a, b, N` => `(a + b) % N`	
09	MULMOD	8	`a, b, N` => `(a * b) % N`	
0A	EXP	A1	`a, b` => `a ** b`	
0B	SIGNEXTEND	5	`b, x` => `SIGNEXTEND(x, b)`	
0C-0F	*invalid*			
10	LT	3	`a, b` => `a < b`	
11	GT	3	`a, b` => `a > b`	
12	SLT	3	`a, b` => `a < b`	

EIP-1559



Sept 2023
ethereum.org

EIP-1559

Gas Cost = base fee + tip

Refund = max fee - (base fee + tip)

BASE FEE CHANGE

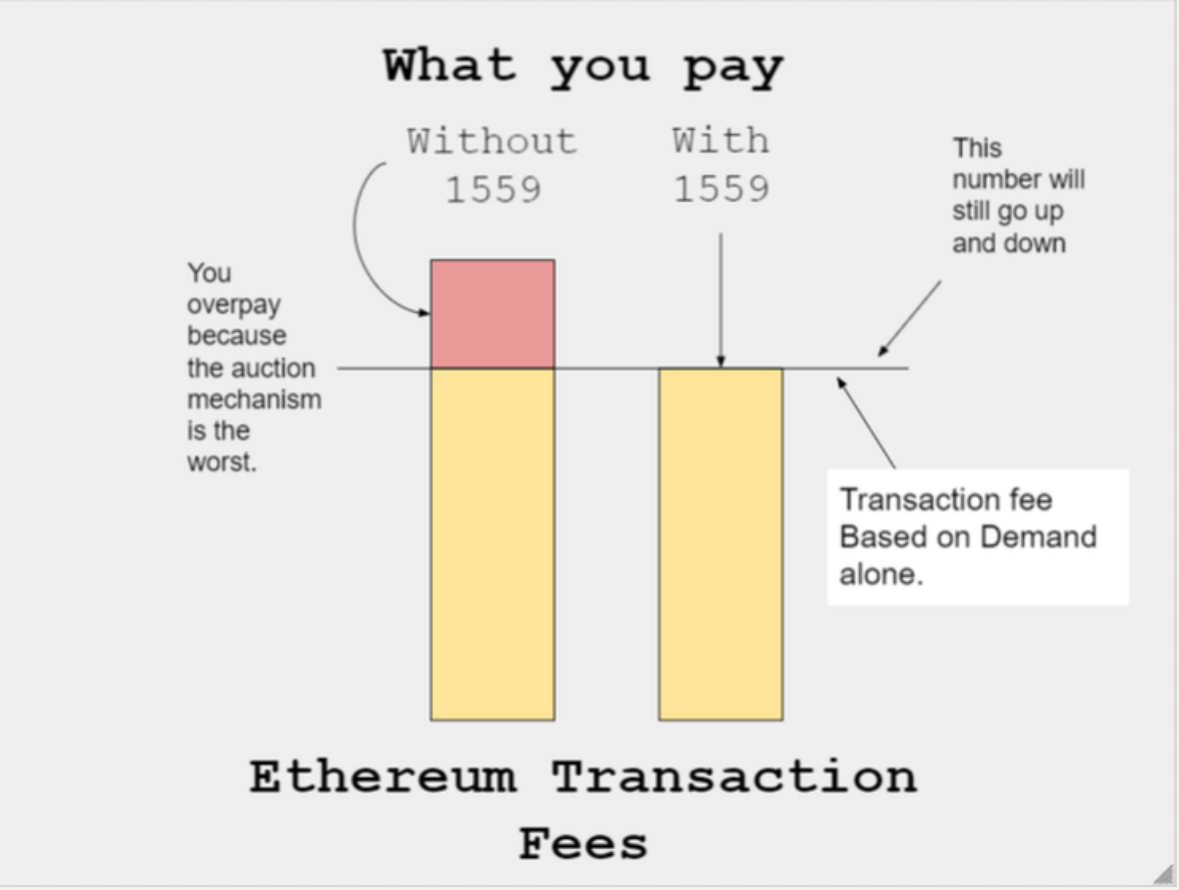
- Tips are awarded to Validators to include transaction into a block
- Base Fee ETH is burned

Last Block Capacity	Change of Base Fee
Exactly 50% full	No change
100% full	12.5% increase
Between 50% and 100%	< 12.5% Increase
Between 0% and 50%	< 12.5% Decrease
0% / Empty	12.5% Decrease

Sept 2023
docs.alchemy.org

BASE FEE CHANGE

Tips are less relevant as Base Fees exponentially increases



Sept 2023
hackmd.io/@tvanepps/1559-wallets

GAS LIMIT

Consider this lottery smart contract code:

```
if(lotteryDrawerCount != 100) {
  lotteryDrawerCount++;
} else {
  **calculate winner via complex calculations**
}
```

What will happen when I submit my transaction at lotteryDrawerCount = 99?

GAS UNITS

Unit	Denominations	
Wei	1	1
Kwei	1,000	10^3
Mwei	1,000,000	10^6
Gwei	1,000,000,000	10^9
Szabo	1,000,000,000,000	10^12
Finney	1,000,000,000,000,000	10^15
Ether	1,000,000,000,000,000,000	10^18
KEther	1,000,000,000,000,000,000,000	10^24
MEther	1,000,000,000,000,000,000,000,000	10^24
GEther	1,000,000,000,000,000,000,000,000,000	10^27
TEther	1,000,000,000,000,000,000,000,000,000,000	10^30

More resources Used:

https://coinsbench.com/about-evm-opcode-gas-ethereum-accounts-9f0896f09d04
https://ethereum.org/
https://hardhat.org/
https://docs.ethers.io/v5/
https://www.openzeppelin.com/
https://takenobu-hs.github.io/downloads/ethereum_evm_illustrated.pdf
https://www.skillsoft.com/