Back To The Future

Of Software Security

Developing Secure Smart Contracts

Final - OWASP Toronto

January 23, 2019

Whoami

- Jamie Baxter, M. Eng., OSCP, OSCE, CISSP, GPEN
- Independent Information Security Consultant focusing on security assessments (applications, infrastructures and smart contracts)
- Previously worked in aerospace, government and finance sectors
- CTF’er, pen-tester, red teamer, appsec

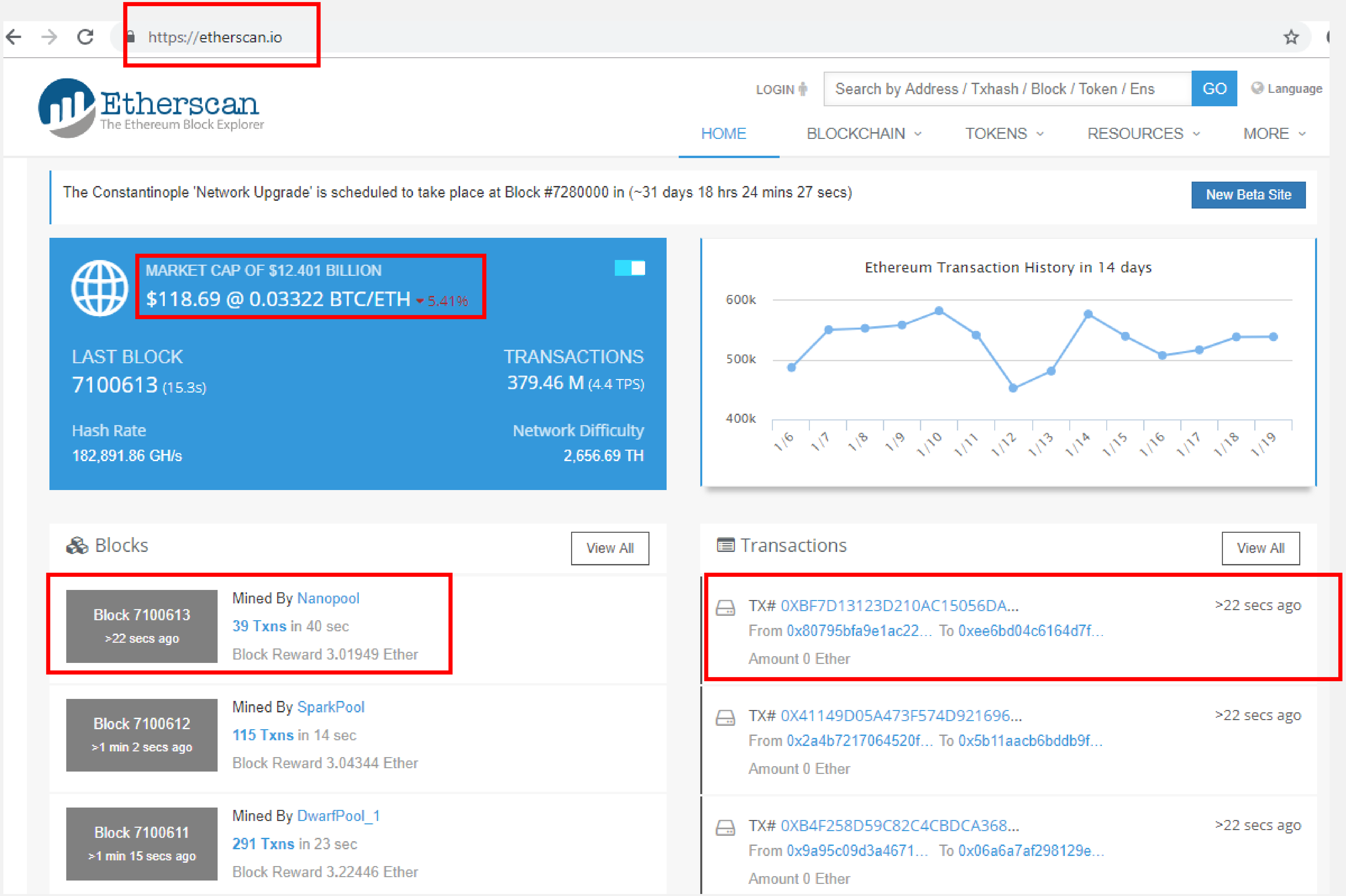
Tonight

- What is a Smart Contract?
- Ethereum Overview
- Smart Contract Introduction
- Smart Contract Vulnerabilities
- Resources And Capture The Flags

What are Ethereum Smart Contracts?

- Def: A Ethereum Smart Contract is a program that defines a general purpose computation which takes place on a blockchain or distributed ledger
- Term originally coin by Nick Szabo
- The smart contract code facilitates, verifies, and enforces the negotiation or performance of an agreement or transaction.
- While self-verifying, self-executing and tamper resistant smart contracts may contain bugs, from programmer errors to flaws in the compiler & toolchain to the platform itself.

Source:
https://blockchainhub.net/smart-contracts/
https://en.wikipedia.org/wiki/Smart_contract/



The Constantinople 'Network Upgrade' is scheduled to take place at Block #7280000 in (~31 days 18 hrs 24 mins 27 secs)

MARKET CAP OF $12.401 BILLION
$118.69 @ 0.03322 BTC/ETH ▾ 5.41%

LAST BLOCK
7100613 (15.3s)

TRANSACTIONS
379.46 M (4.4 TPS)

Hash Rate
182,891.86 GH/s

Network Difficulty
2,656.69 TH

Blocks

Block 7100613
>22 secs ago
Mined By Nanopool
39 Txns in 40 sec
Block Reward 3.01949 Ether

Block 7100612
>1 min 2 secs ago
Mined By SparkPool
115 Txns in 14 sec
Block Reward 3.04344 Ether

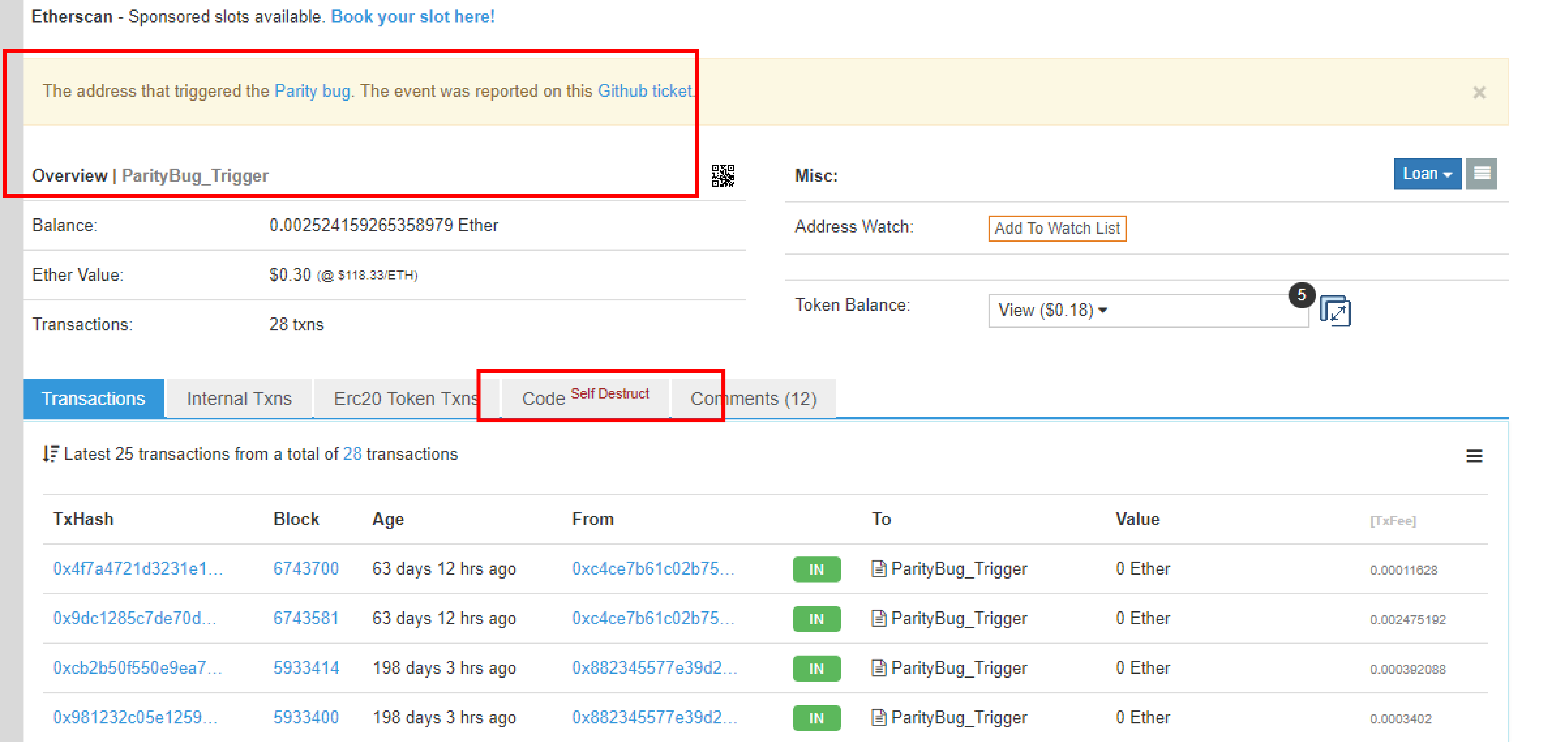
Block 7100611
>1 min 15 secs ago
Mined By DwarfPool_1
291 Txns in 23 sec
Block Reward 3.22446 Ether

Transactions

TX# 0XBF7D13123D210AC15056DA... >22 secs ago
From 0x80795bfa9e1ac22... To 0xee6bd04c6164d7f...
Amount 0 Ether

TX# 0X41149D05A473F574D921696... >22 secs ago
From 0x2a4b7217064520f... To 0x5b11aacb6bddb9f...
Amount 0 Ether

TX# 0XB4F258D59C82C4CBDCA368... >22 secs ago
From 0x9a95c09d3a4671... To 0x06a6a7af298129e...
Amount 0 Ether



The address that triggered the Parity bug. The event was reported on this Github ticket.

Overview | ParityBug_Trigger

Balance: 0.002524159265358979 Ether

Ether Value: $0.30 (@ $118.33/ETH)

Transactions: 28 txns

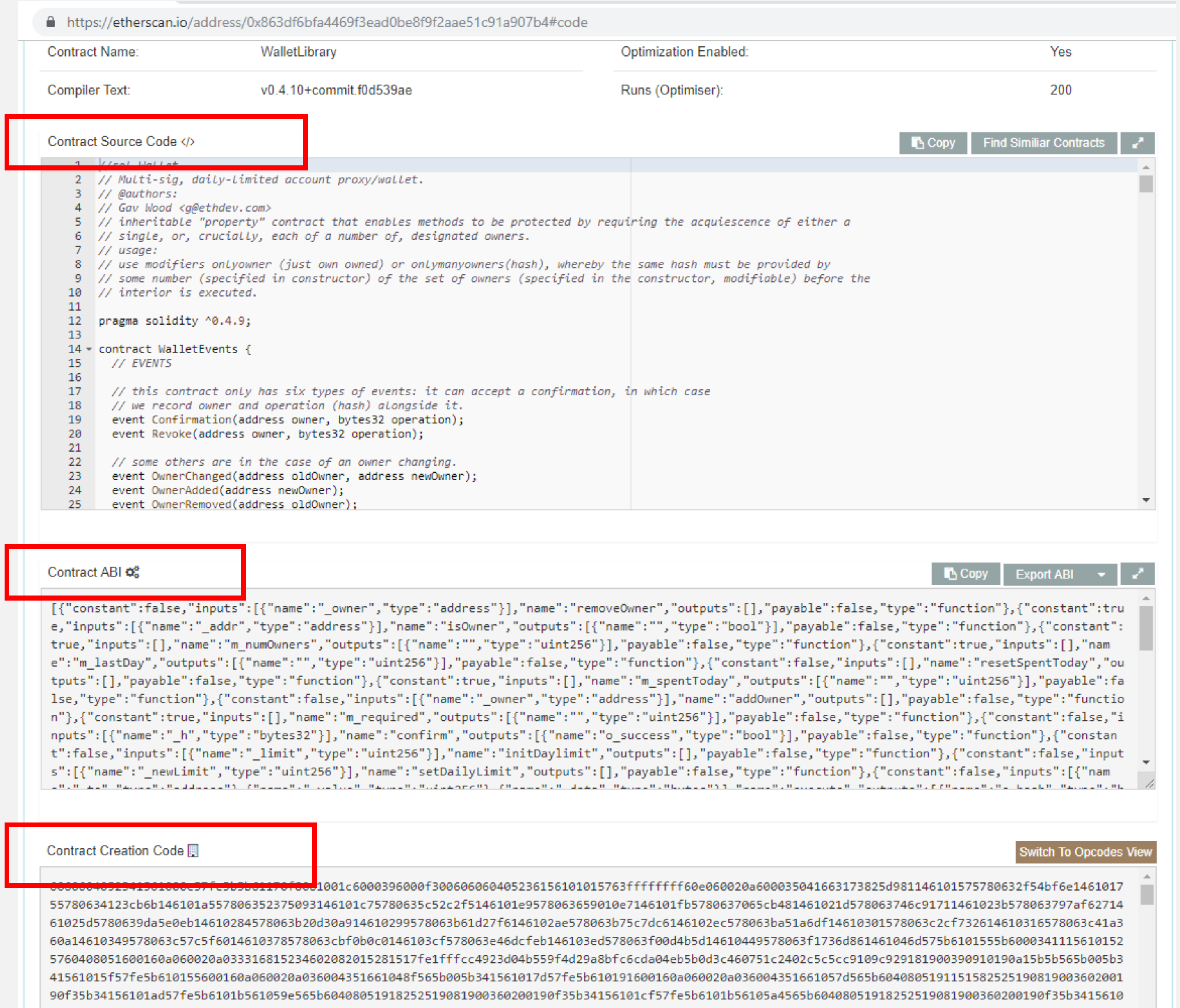
Misc:

Address Watch: Add To Watch List

Token Balance: View ($0.18)

Latest 25 transactions from a total of 28 transactions

TxHash	Block	Age	From		To	Value	[TxFee]
0x4f7a4721d3231e1...	6743700	63 days 12 hrs ago	0xc4ce7b61c02b75...	IN	ParityBug_Trigger	0 Ether	0.00011628
0x9dc1285c7de70d...	6743581	63 days 12 hrs ago	0xc4ce7b61c02b75...	IN	ParityBug_Trigger	0 Ether	0.002475192
0xcb2b50f550e9ea7...	5933414	198 days 3 hrs ago	0x882345577e39d2...	IN	ParityBug_Trigger	0 Ether	0.000392088
0x981232c05e1259...	5933400	198 days 3 hrs ago	0x882345577e39d2...	IN	ParityBug_Trigger	0 Ether	0.0003402



https://etherscan.io/address/0x863df6bfa4469f3ead0be8f9f2aae51c91a907b4#code

Contract Name:	WalletLibrary	Optimization Enabled:	Yes
Compiler Text:	v0.4.10+commit.f0d539ae	Runs (Optimiser):	200

Contract Source Code </>

Copy
Find Similiar Contracts

```
//sol Wallet
// Multi-sig, daily-limited account proxy/wallet.
// @authors:
// Gav Wood <g@ethdev.com>
// inheritable "property" contract that enables methods to be protected by requiring the acquiescence of either a
// single, or, crucially, each of a number of, designated owners.
// usage:
// use modifiers onlyowner (just own owned) or onlymanyowners(hash), whereby the same hash must be provided by
// some number (specified in constructor) of the set of owners (specified in the constructor, modifiable) before the
// interior is executed.

pragma solidity ^0.4.9;

contract WalletEvents {
  // EVENTS

  // this contract only has six types of events: it can accept a confirmation, in which case
  // we record owner and operation (hash) alongside it.
  event Confirmation(address owner, bytes32 operation);
  event Revoke(address owner, bytes32 operation);

  // some others are in the case of an owner changing.
  event OwnerChanged(address oldOwner, address newOwner);
  event OwnerAdded(address newOwner);
  event OwnerRemoved(address oldOwner);
```

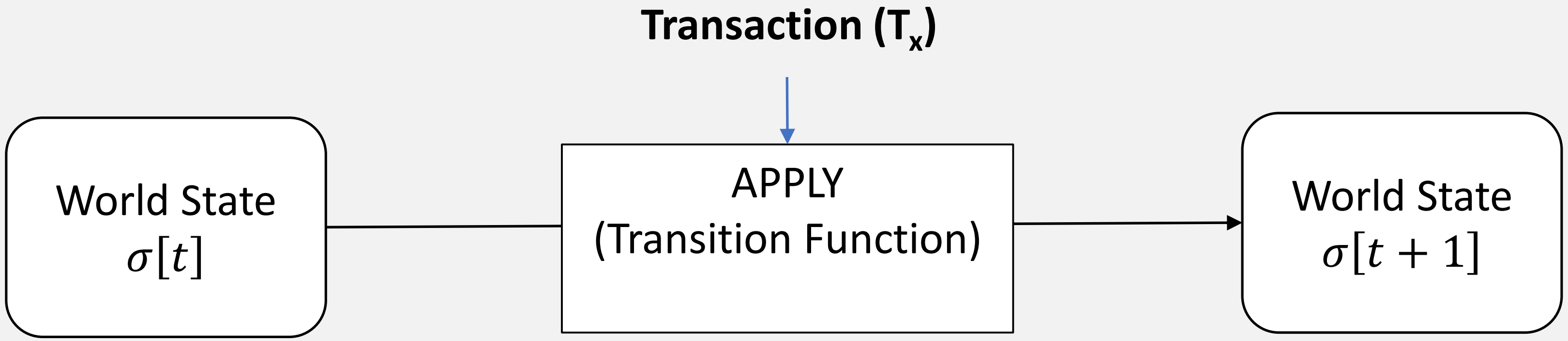
Contract ABI

[{"constant":false,"inputs":[{"name":"_owner","type":"address"}],"name":"removeOwner","outputs":[],"payable":false,"type":"function"},{"constant":true,"inputs":[{"name":"_addr","type":"address"}],"name":"isOwner","outputs":[{"name":"","type":"bool"}],"payable":false,"type":"function"},{"constant":true,"inputs":[],"name":"m_numOwners","outputs":[{"name":"","type":"uint256"}],"payable":false,"type":"function"},{"constant":true,"inputs":[],"name":"m_lastDay","outputs":[{"name":"","type":"uint256"}],"payable":false,"type":"function"},{"constant":false,"inputs":[],"name":"resetSpentToday","outputs":[],"payable":false,"type":"function"},{"constant":true,"inputs":[],"name":"m_spentToday","outputs":[{"name":"","type":"uint256"}],"payable":false,"type":"function"},{"constant":false,"inputs":[{"name":"_owner","type":"address"}],"name":"addOwner","outputs":[],"payable":false,"type":"function"},{"constant":true,"inputs":[],"name":"m_required","outputs":[{"name":"","type":"uint256"}],"payable":false,"type":"function"},{"constant":false,"inputs":[{"name":"_h","type":"bytes32"}],"name":"confirm","outputs":[{"name":"o_success","type":"bool"}],"payable":false,"type":"function"},{"constant":false,"inputs":[{"name":"_limit","type":"uint256"}],"name":"initDaylimit","outputs":[],"payable":false,"type":"function"},{"constant":false,"inputs":[{"name":"_newLimit","type":"uint256"}],"name":"setDailyLimit","outputs":[],"payable":false,"type":"function"},{"constant":false,"inputs":[{"name":"_to","type":"address"},{"name":"_value","type":"uint256"},{"name":"_data","type":"bytes"}],"name":"execute","outputs":[{"name":"o_hash","type":"b

Contract Creation Code

6060604052341561000c57fe5b5b61170f8061001c6000396000f30060606060405236156101015763ffffffff60e060020a600035041663173825d981146101575780632f54bf6e1461017
55780634123cb6b146101a557806352375093146101c75780635c52c2f5146101e9578063659010e7146101fb5780637065cb481461021d578063746c91711461023b578063797af62714
61025d5780639da5e0eb14610284578063b20d30a914610299578063b61d27f6146102ae578063b75c7dc6146102ec578063ba51a6df14610301578063c2cf732614610316578063c41a3
60a14610349578063c57c5f6014610378578063cbf0b0c0146103cf578063e46dcfeb146103ed578063f00d4b5d14610449578063f1736d861461046d575b6101555b6000341115610152
5760408051600160a060020a033316815234602082015281517fe1fffcc4923d04b559f4d29a8bfc6cda04eb5b0d3c460751c2402c5c5cc9109c929181900390910190a15b5b565b005b3
41561015f57fe5b610155600160a060020a036004351661048f565b005b341561017d57fe5b610191600160a060020a036004351661057d565b6040805191151582525190819003602001
90f35b34156101ad57fe5b6101b561059e565b60408051918252519081900360200190f35b34156101cf57fe5b6101b56105a4565b60408051918252519081900360200190f35b3415610

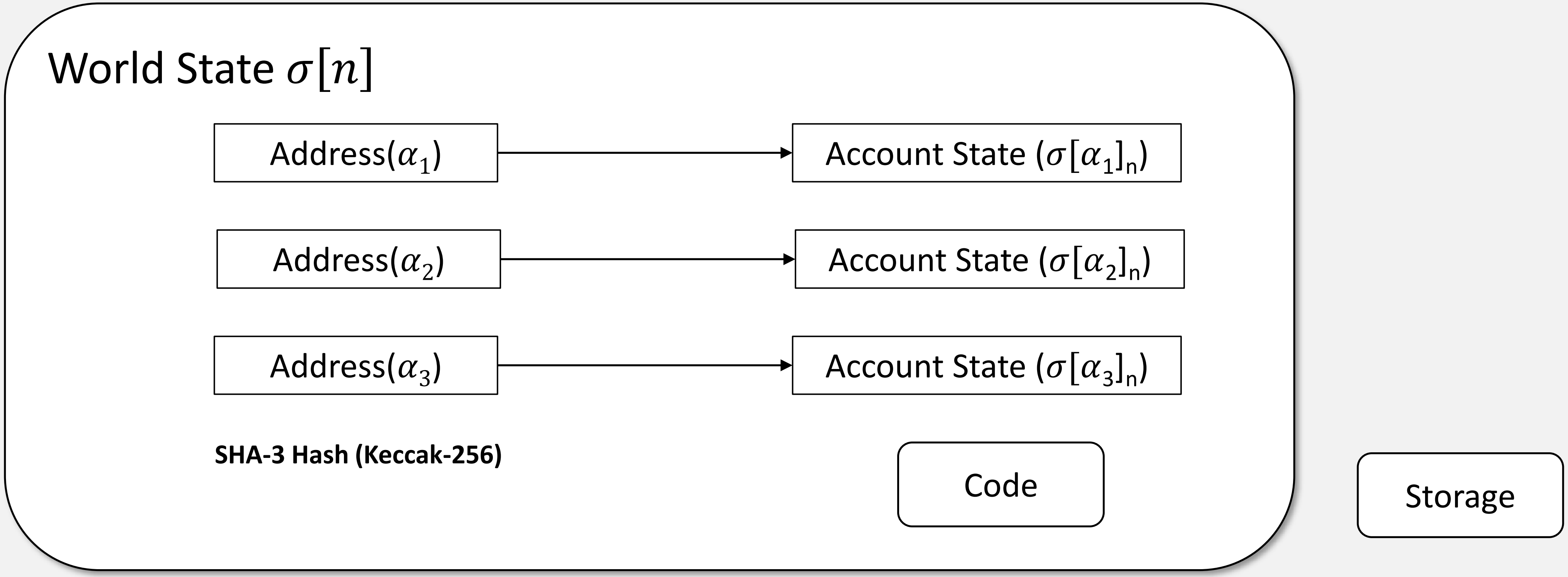
Ethereum is a Transaction Based State Machine



A transaction is a single cryptographically-signed instruction

What is a World State ($\boldsymbol{\sigma}$)?

- It is the mapping between addresses and their account state at a given time



World State $\sigma[n]$

Address(α_1) → Account State ($\sigma[\alpha_1]_n$)

Address(α_2) → Account State ($\sigma[\alpha_2]_n$)

Address(α_3) → Account State ($\sigma[\alpha_3]_n$)

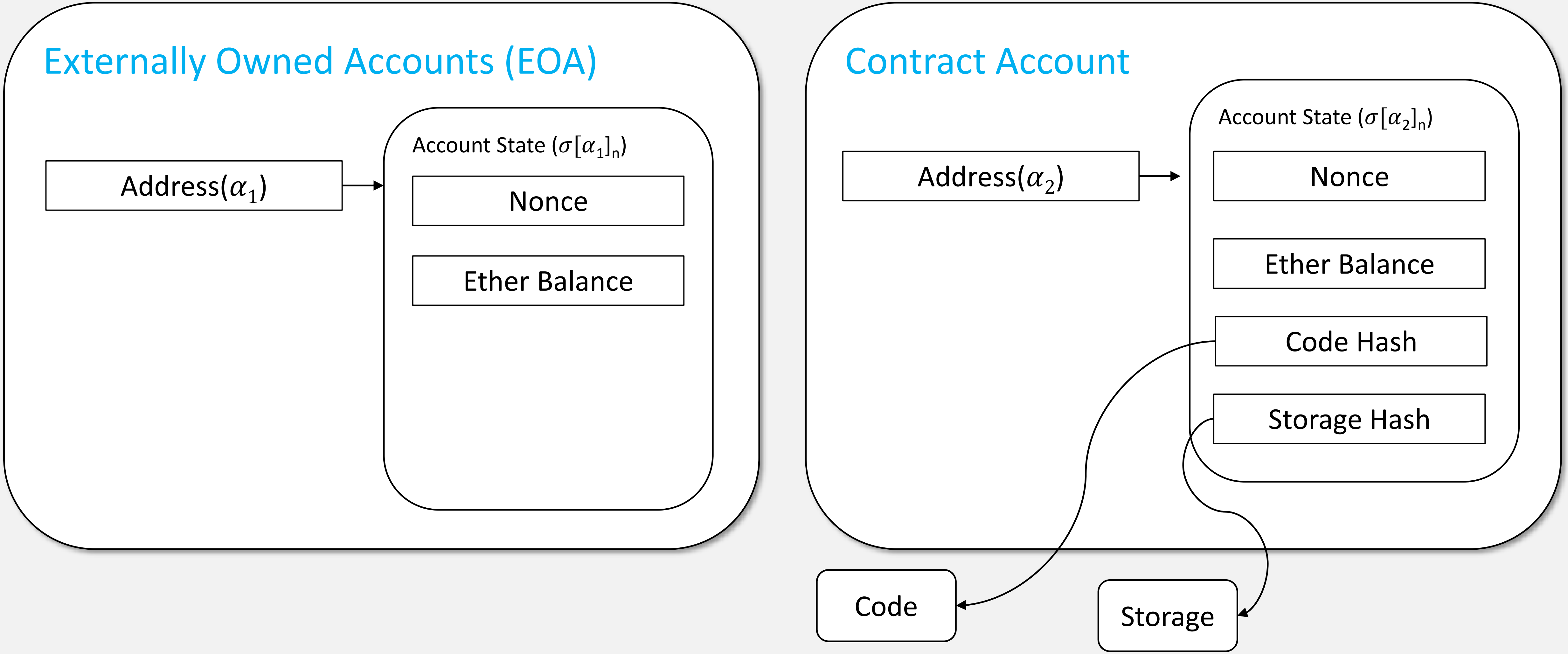
SHA-3 Hash (Keccak-256)

Code

Storage

What's in an Account?

There's actually two types of accounts



Externally Owned Accounts (EOA)

Address(α_1)

Account State ($\sigma[\alpha_1]_n$)

- Nonce
- Ether Balance

Contract Account

Address(α_2)

Account State ($\sigma[\alpha_2]_n$)

- Nonce
- Ether Balance
- Code Hash
- Storage Hash

Code

Storage

A Word on Addresses

Externally Owned Account (EOA) Address (A)

$A = B_{96..255}(KEC(PUBKEY(p_r)))$ Where p_r is the private key

Contract Accounts Address (A)

$$A = B_{96..255}(KEC(Sender\ Address, Nonce))$$

Account Type Summary

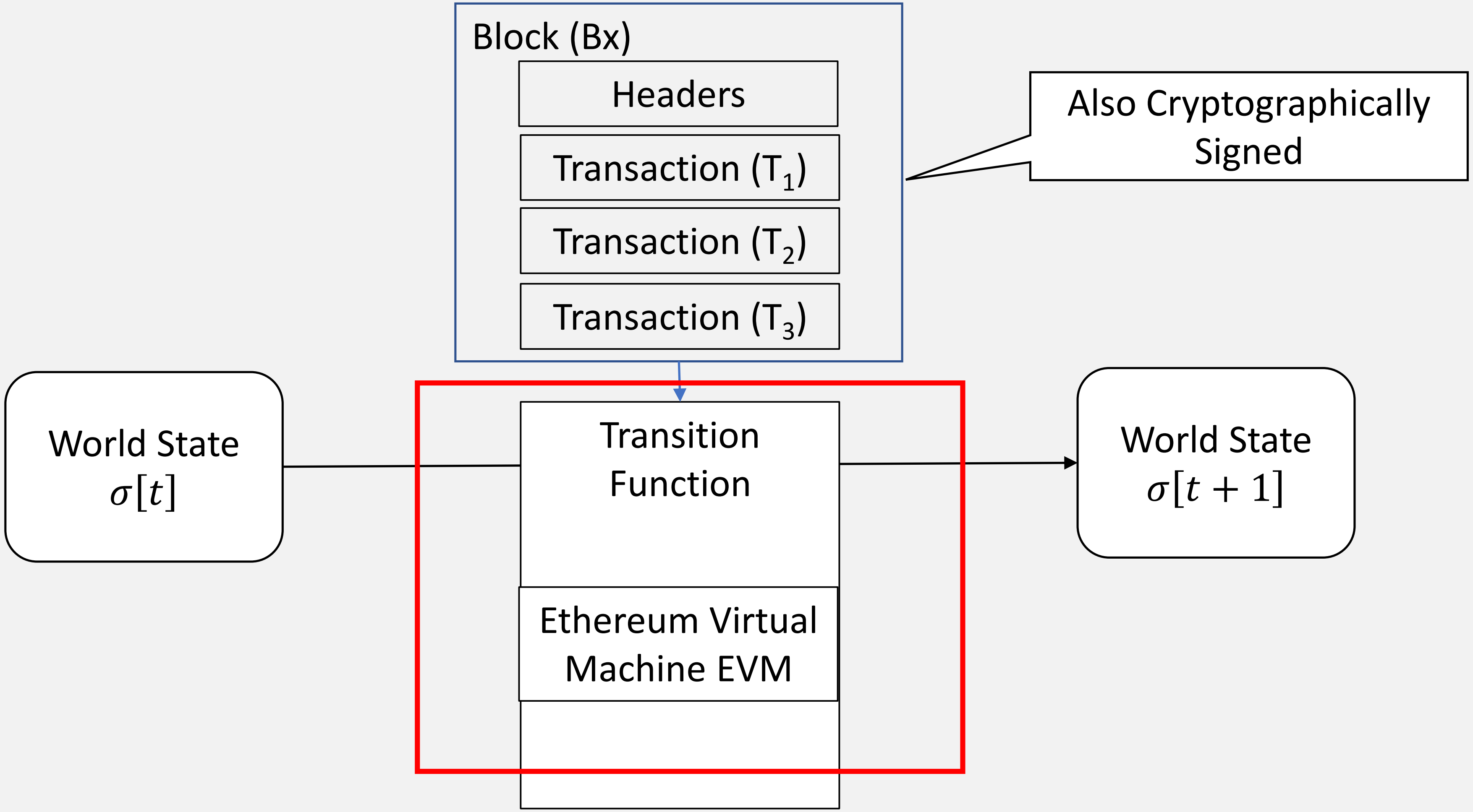
Externally Owned Accounts

- Have a nonce
- Have an Ether balance
- Can send transactions
  - Transfers
  - Messages to Contracts or other EOAs
- Only EOA can initiate transactions

Contract Accounts

- Have a nonce
- Have an Ether balance
- Code hash
- Code execution is triggered by a transaction
- Can call other contracts

Multiple Transactions are Combined in a Block



Block (Bx)

Headers

Transaction (T_1)

Transaction (T_2)

Transaction (T_3)

Also Cryptographically Signed

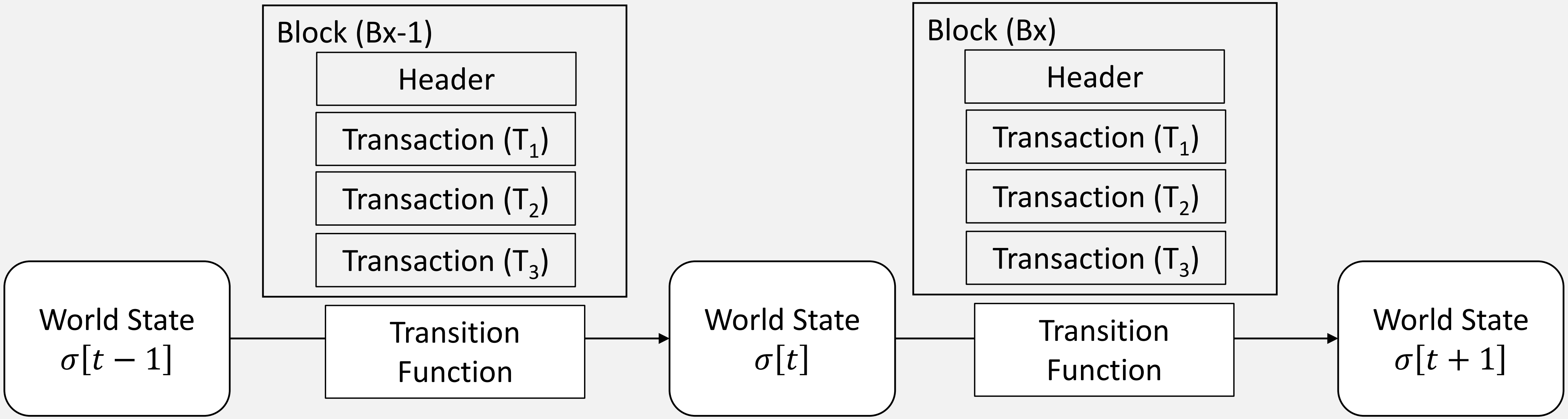
World State $\sigma[t]$

Transition Function

Ethereum Virtual Machine EVM

World State $\sigma[t+1]$

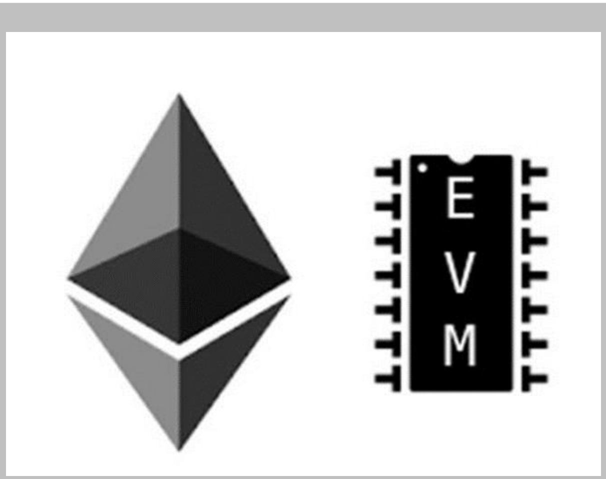
The Sequence of Blocks and World States

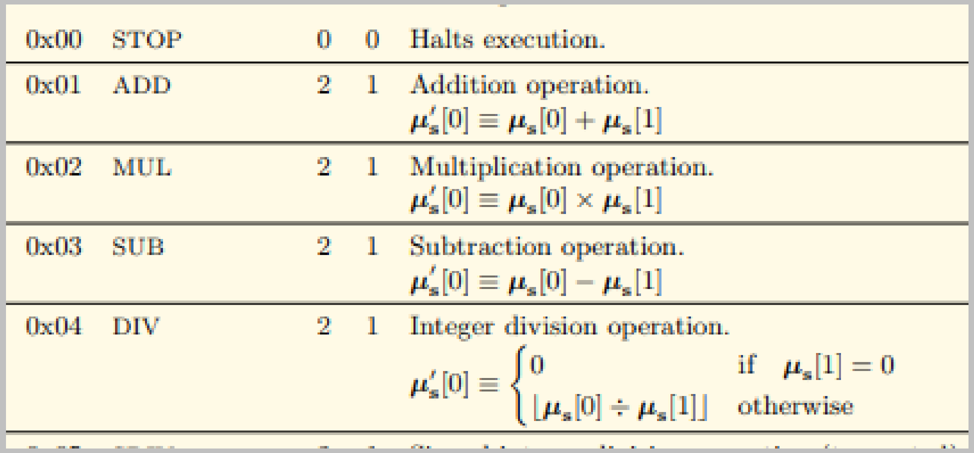


…is the Blockchain!

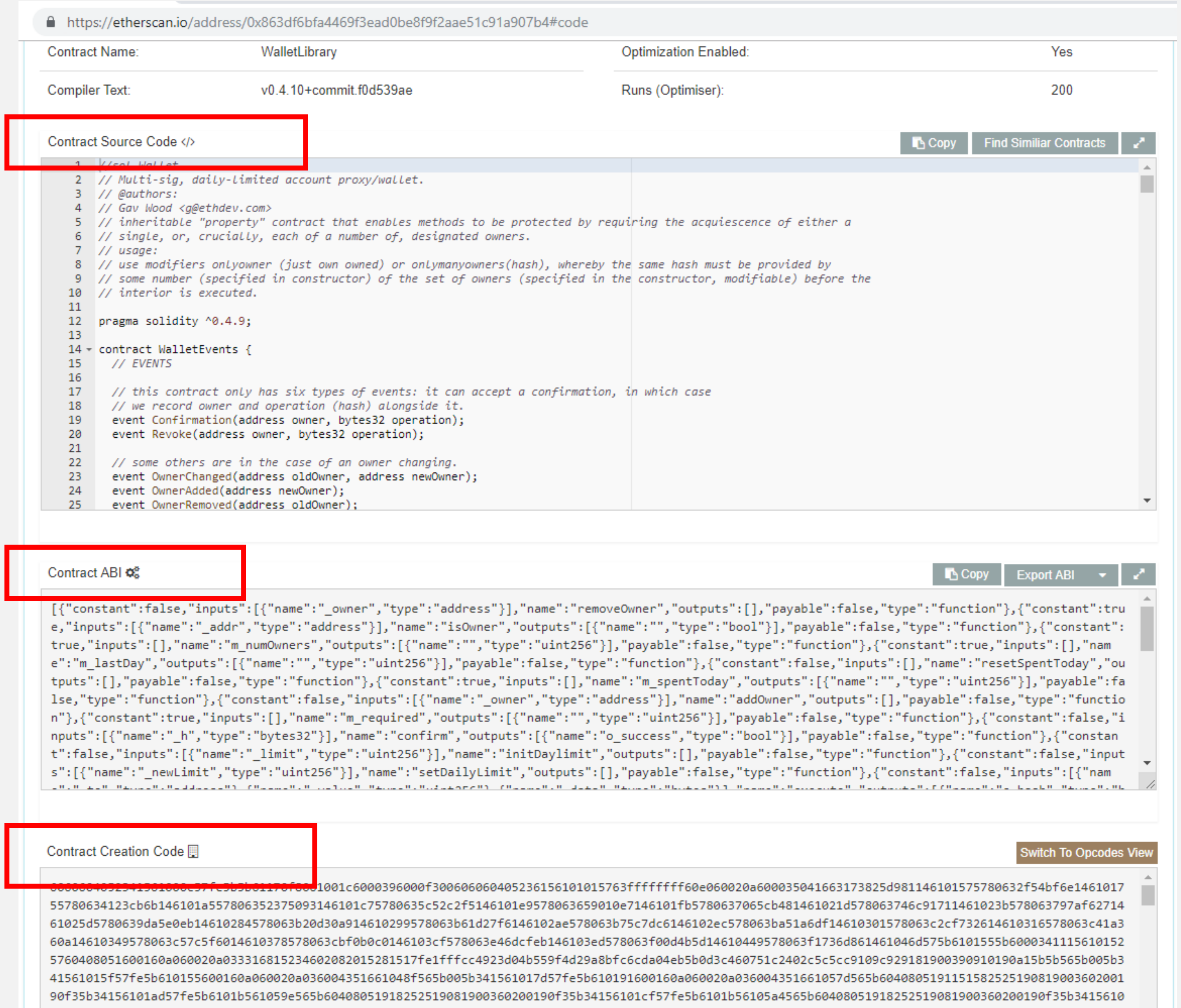
The Transition Function - Ethereum Virtual Machine (EVM)

- Turing complete instruction set 2^8 Op Codes, Fixed Length)
- 256-bit word machine
- 1024 element stack (of 256 bits each)
- 8-Bit opcodes
- No registers (purely stack based)
- Storage (persistent / per account)
- Memory (volatile)
- It’s purpose is run EVM Byte Code (aka Smart Contracts)





0x00	STOP	0	0	Halts execution.
0x01	ADD	2	1	Addition operation. $\boldsymbol{\mu}'_s[0] \equiv \boldsymbol{\mu}_s[0] + \boldsymbol{\mu}_s[1]$
0x02	MUL	2	1	Multiplication operation. $\boldsymbol{\mu}'_s[0] \equiv \boldsymbol{\mu}_s[0] \times \boldsymbol{\mu}_s[1]$
0x03	SUB	2	1	Subtraction operation. $\boldsymbol{\mu}'_s[0] \equiv \boldsymbol{\mu}_s[0] - \boldsymbol{\mu}_s[1]$
0x04	DIV	2	1	Integer division operation. $\boldsymbol{\mu}'_s[0] \equiv \begin{cases} 0 & \text{if} \quad \boldsymbol{\mu}_s[1] = 0 \\ \lfloor \boldsymbol{\mu}_s[0] \div \boldsymbol{\mu}_s[1] \rfloor & \text{otherwise} \end{cases}$



https://etherscan.io/address/0x863df6bfa4469f3ead0be8f9f2aae51c91a907b4#code

Contract Name:	WalletLibrary	Optimization Enabled:	Yes
Compiler Text:	v0.4.10+commit.f0d539ae	Runs (Optimiser):	200

Contract Source Code </>

Copy
Find Similiar Contracts

```
//sol Wallet
// Multi-sig, daily-limited account proxy/wallet.
// @authors:
// Gav Wood <g@ethdev.com>
// inheritable "property" contract that enables methods to be protected by requiring the acquiescence of either a
// single, or, crucially, each of a number of, designated owners.
// usage:
// use modifiers onlyowner (just own owned) or onlymanyowners(hash), whereby the same hash must be provided by
// some number (specified in constructor) of the set of owners (specified in the constructor, modifiable) before the
// interior is executed.

pragma solidity ^0.4.9;

contract WalletEvents {
  // EVENTS

  // this contract only has six types of events: it can accept a confirmation, in which case
  // we record owner and operation (hash) alongside it.
  event Confirmation(address owner, bytes32 operation);
  event Revoke(address owner, bytes32 operation);

  // some others are in the case of an owner changing.
  event OwnerChanged(address oldOwner, address newOwner);
  event OwnerAdded(address newOwner);
  event OwnerRemoved(address oldOwner);
```

Contract ABI

[{"constant":false,"inputs":[{"name":"_owner","type":"address"}],"name":"removeOwner","outputs":[],"payable":false,"type":"function"},{"constant":true,"inputs":[{"name":"_addr","type":"address"}],"name":"isOwner","outputs":[{"name":"","type":"bool"}],"payable":false,"type":"function"},{"constant":true,"inputs":[],"name":"m_numOwners","outputs":[{"name":"","type":"uint256"}],"payable":false,"type":"function"},{"constant":true,"inputs":[],"name":"m_lastDay","outputs":[{"name":"","type":"uint256"}],"payable":false,"type":"function"},{"constant":false,"inputs":[],"name":"resetSpentToday","outputs":[],"payable":false,"type":"function"},{"constant":true,"inputs":[],"name":"m_spentToday","outputs":[{"name":"","type":"uint256"}],"payable":false,"type":"function"},{"constant":false,"inputs":[{"name":"_owner","type":"address"}],"name":"addOwner","outputs":[],"payable":false,"type":"function"},{"constant":true,"inputs":[],"name":"m_required","outputs":[{"name":"","type":"uint256"}],"payable":false,"type":"function"},{"constant":false,"inputs":[{"name":"_h","type":"bytes32"}],"name":"confirm","outputs":[{"name":"o_success","type":"bool"}],"payable":false,"type":"function"},{"constant":false,"inputs":[{"name":"_limit","type":"uint256"}],"name":"initDaylimit","outputs":[],"payable":false,"type":"function"},{"constant":false,"inputs":[{"name":"_newLimit","type":"uint256"}],"name":"setDailyLimit","outputs":[],"payable":false,"type":"function"},{"constant":false,"inputs":[{"nam

Contract Creation Code

6060604052341561000e57fe5b5b61170f8061001c6000396000f30060606040523615610101576 3ffffffff60e060020a600035041663173825d98114610157578063 2f54bf6e1461017
55780634123cb6b146101a55780635237509314610 1c75780635c52c2f5146101e9578063659010e7146101fb5780637065cb481461021d578063746c91711461023b578063797af62714
61025d5780639da5e0eb14610284578063b20d30a914610299578063b61d27f6146102ae578063b75c7dc6146102ec578063ba51a6df14610301578063c2cf73261461031657 8063c41a3
60a14610349578063c57c5f6014610378578063cbf0b0c0146103cf578063e46dcfeb146103ed578063f00d4b5d14610449578063f1736d861461046d575b6101555b600034111561015 2
5760408051600160a060020a033316815234602082015281517fe1fffcc4923d04b559f4d29a8bfc6cda04eb5b0d3c460751c2402c5c5cc9109c929181900390910190a15b5b565b005b3
41561015f57fe5b610155600160a060020a036004351661048f565b005b341561017d57fe5b610191600160a060020a036004351661057d565b6040805191151582525190819003602001
90f35b34156101ad57fe5b6101b561059e565b60408051918252519081900360200190f35b34156101cf57fe5b6101b56105a4565b60408051918252519081900360200190f35b3415610

What are Ethereum Smart Contracts?

- Smart Contracts are very similar to classes in C++ or Java
- All Smart Contracts are **bound to an address** and **have an ether balance** associated with them
- Smart Contracts have a constructor (no overloading though)
- Solidity supports inheritance and polymorphism
- Other objected orientated concepts like visibility (private, public), state variables and interfaces also all apply
- Compiled to EVM Bytecode and stored in the world state indexed by code hash
- Contracts can be killed (suicide)
- Usually written in Solidity. But other languages exist ex: LLL

Life Cycle of a Smart Contract

Transaction to Create

- Issued by a EOA or another Smart Contract (contracts can create contracts)

Execution Driven by Transactions

- Receive transactions (calls, delegate calls)
- Perform actions
- Functions called from other functions

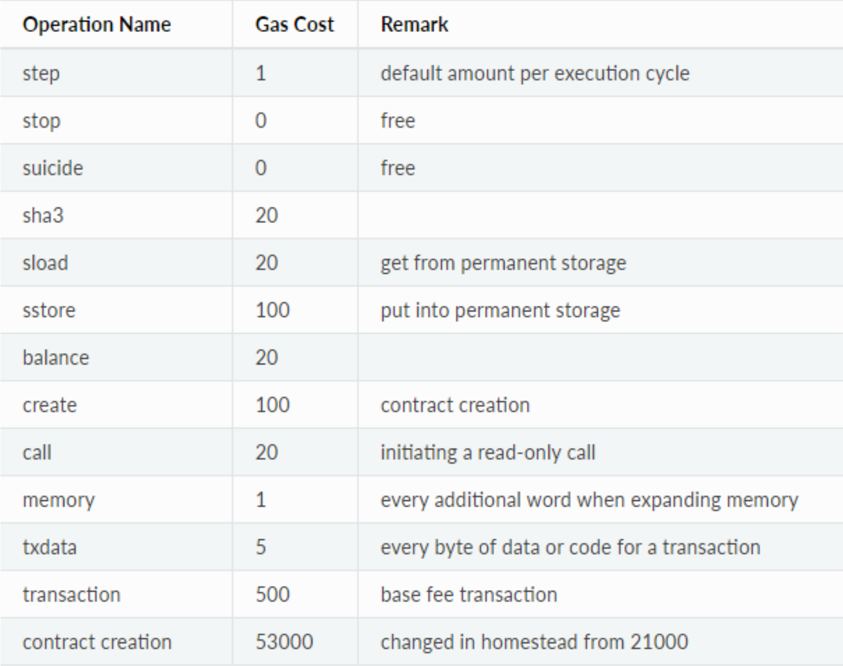
Suicide or “Freeze”

Every Contract is stored within the world state.

Contract Execution - Everything has a Price!

- Cost is measured in "GAS"
- The unit price of GAS in Ether is defined by the initiator of the transaction.
- Creating a contract costs GAS
- All execution steps cost GAS
- The more complex the execution the greater the cost
- Each transaction is provided a GAS stipend to begin execution
- Each block is subject to the GAS limit of 8 million.
  - Consider an expensive transaction like SSTORE (20000 Gas) means a block can write to store 400 times
  - Ethereum network can process about 25 transactions per second. Though multiple initiatives are underway to greatly increase that

Partial List of GAS costs



Operation Name	Gas Cost	Remark
step	1	default amount per execution cycle
stop	0	free
suicide	0	free
sha3	20	
sload	20	get from permanent storage
sstore	100	put into permanent storage
balance	20	
create	100	contract creation
call	20	initiating a read-only call
memory	1	every additional word when expanding memory
txdata	5	every byte of data or code for a transaction
transaction	500	base fee transaction
contract creation	53000	changed in homestead from 21000

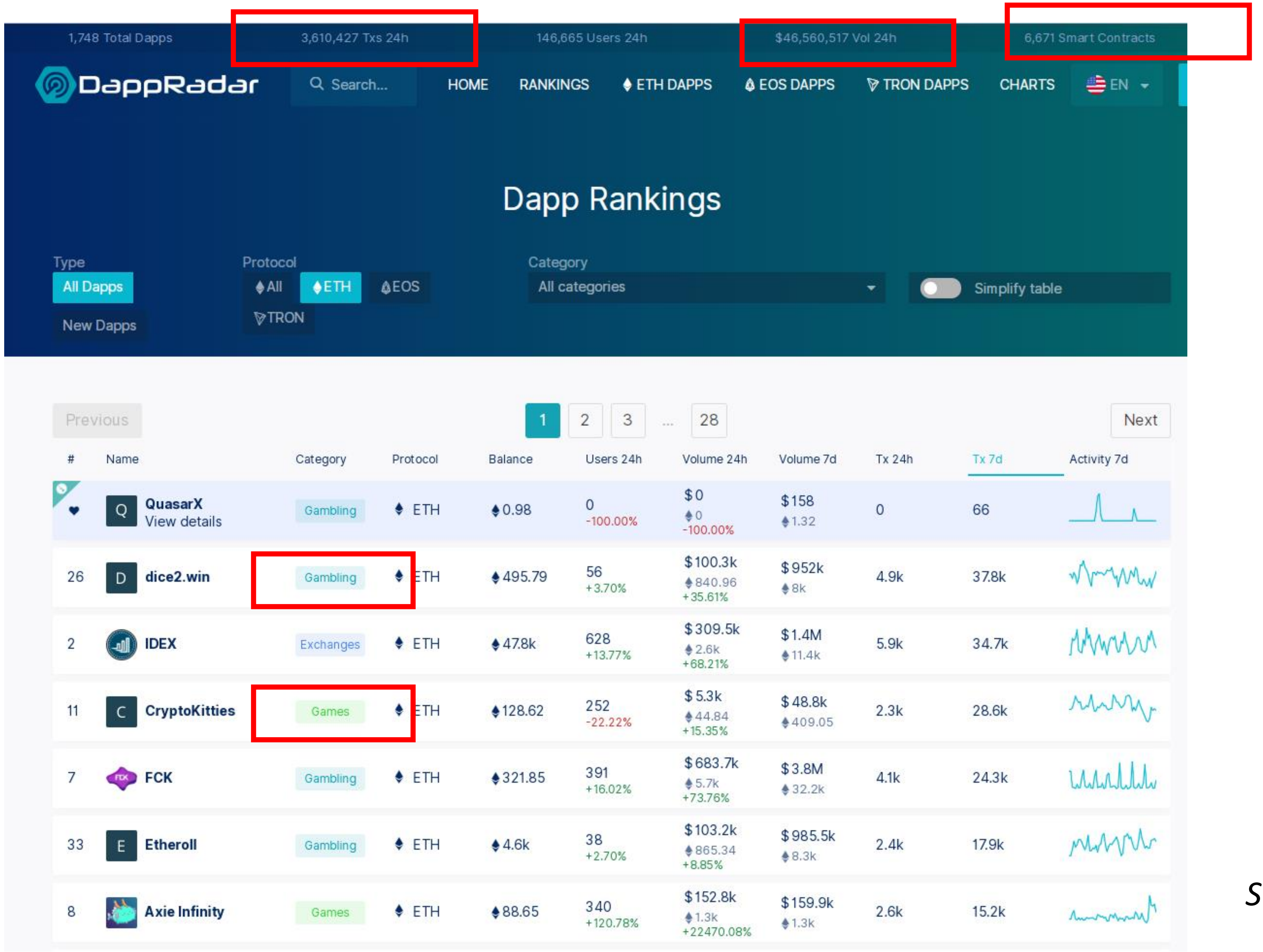
Distributed Applications (dApps)
(Simplified)

Contract(s) Backend

Web Gui Front End

An Example dApp - CryptoKitties!

A recent Dapp Ranking



1,748 Total Dapps | 3,610,427 Txs 24h | 146,665 Users 24h | $46,560,517 Vol 24h | 6,671 Smart Contracts

Dapp Rankings

#	Name	Category	Protocol	Balance	Users 24h	Volume 24h	Volume 7d	Tx 24h	Tx 7d	Activity 7d
	QuasarX View details	Gambling	♦ ETH	♦0.98	0 -100.00%	$0 ♦0 -100.00%	$158 ♦1.32	0	66	
26	dice2.win	Gambling	♦ ETH	♦495.79	56 +3.70%	$100.3k ♦840.96 +35.61%	$952k ♦8k	4.9k	37.8k	
2	IDEX	Exchanges	♦ ETH	♦47.8k	628 +13.77%	$309.5k ♦2.6k +68.21%	$1.4M ♦11.4k	5.9k	34.7k	
11	CryptoKitties	Games	♦ ETH	♦128.62	252 -22.22%	$5.3k ♦44.84 +15.35%	$48.8k ♦409.05	2.3k	28.6k	
7	FCK	Gambling	♦ ETH	♦321.85	391 +16.02%	$683.7k ♦5.7k +73.76%	$3.8M ♦32.2k	4.1k	24.3k	
33	Etheroll	Gambling	♦ ETH	♦4.6k	38 +2.70%	$103.2k ♦865.34 +8.85%	$985.5k ♦8.3k	2.4k	17.9k	
8	Axie Infinity	Games	♦ ETH	♦88.65	340 +120.78%	$152.8k ♦1.3k +22470.08%	$159.9k ♦1.3k	2.6k	15.2k	

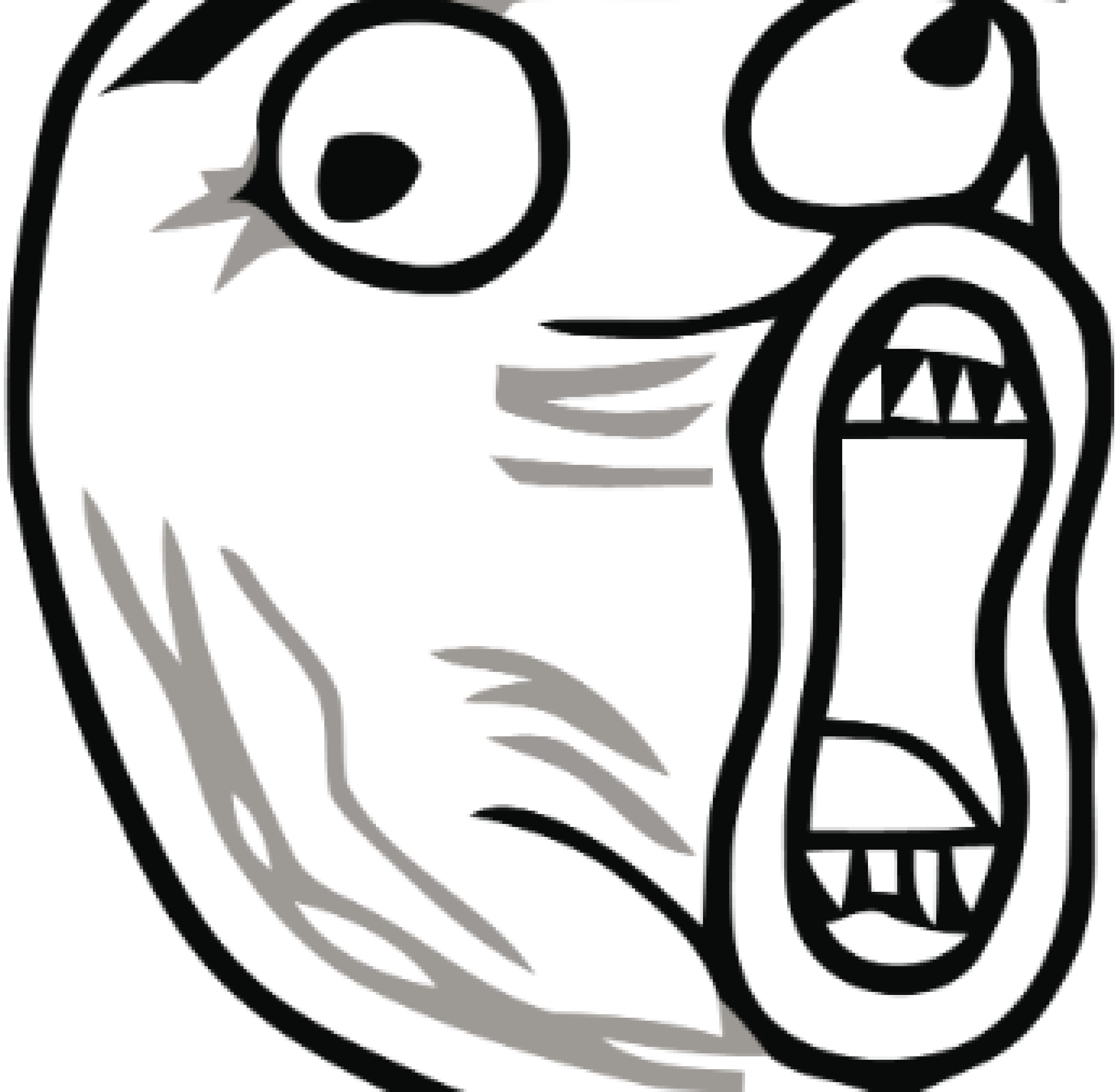
Source: http://dappradar.com

Tools – A Sampling

Tool	Descriptions	Comments
Metamask	A Browser Extension for Running dApps	Wallet Integration
Mist	Dedicated Dapp Browser	Wallet Integration
Ganache	Ethereum Personal Blockchain (Now you can have a blockchain too!)	“Ganache is a personal blockchain for Ethereum development you can use to deploy contracts, develop your applications, and run tests”
Truffle	Smart Contract Development Suite	Compile and Deploy Smart Contracts
Remix	IDE	Online
Geth	Ethereum Node Controller (can join main or multiple test and special purpose nets)	**geth** is the the command line interface for running a full ethereum node implemented in Go.

So, of course, all the past lessons in software security have been applied and Smart Contracts are now *bug free*…

Thanks for coming out!



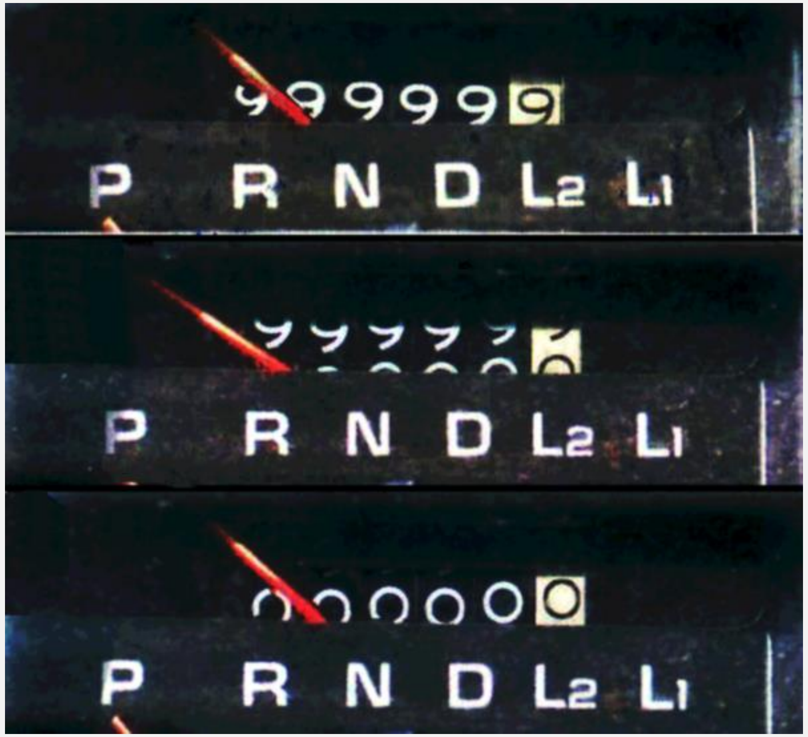
LOL

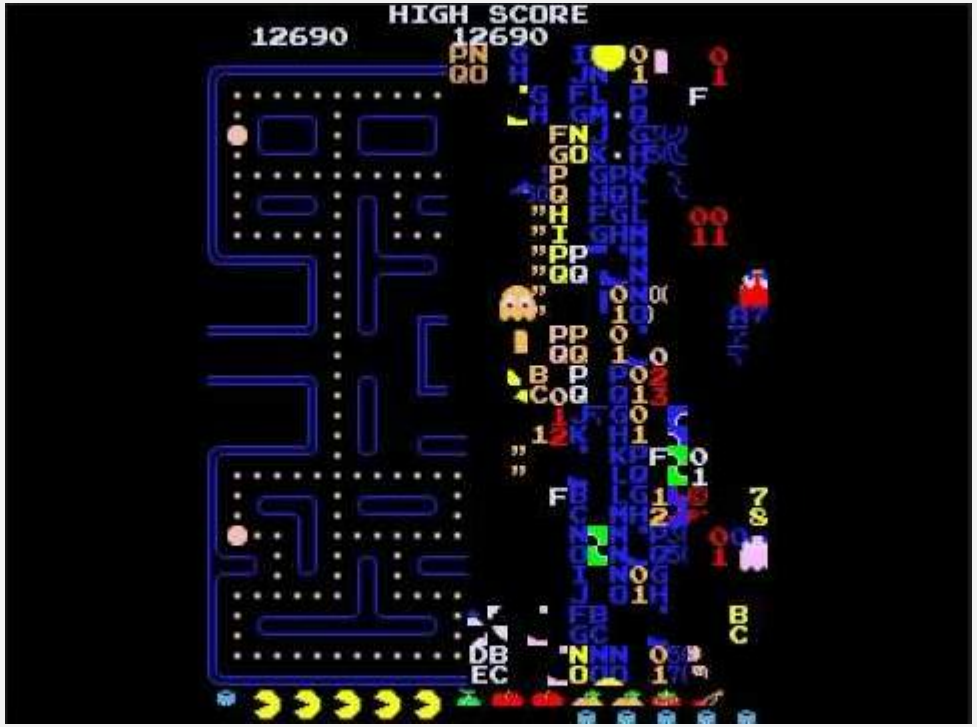
Everything old is new again!

- Integer Underflow / Overflow (SWC-101)
- Unprotected Sensitive Functions (Self-Destruct) (SWC-106)
- Exposed Private Data
- Bad Randomness (SWC-120)
- Re-Entrancy (SWC-107)
- Unsafe Authorization (SWC-115)
- Unsafe Contract Constructors (SWC-115)
- Out-Of-Bounds Write-Anywhere (SWC-124)
- Unprotected Withdrawal

There are currently 29 weakness patterns identified in Smart Contracts:
Source: https://en.wikipedia.org/wiki/Integer_overflow

Integer Overflows have been with us…for a *long*, *long* time!





Source: https://en.wikipedia.org/wiki/Integer_overflow

Integer Overflow (Simple) - (SWC-101)

```
pragma solidity ^0.4.24;


contract OverflowAdd {
    uint256 private balance = 1;

    function add(uint256 deposit) public {
        balance = balance + deposit;
    }
```

Execution Run #1

balance = 1
add(100)
balance = 101

Execution Run #2

balance = 2^256
add(1)
balance = 0

Source: https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-101

Integer Overflow (Simple) - (SWC-101)

```
pragma solidity ^0.4.24;

contract Overflow_Add {
    uint256 private Balance = 1;

    function AddSafe(uint256 deposit) public {
       uint256 newBalance = balance + deposit;
       require(newBalance >= deposit, “OVERFLOW DETECTED”);

       balance += deposit;
    }

}
```

Execution Run #1
Balance = 1
AddSafe(100)
balance = 101

Execution Run #2
Balance = 2^256
AddSafe(1)
Balance = 0 ‘ Exception Thrown

Source: https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-101

Integer Overflow (More Complex) - (SWC-101)

```
pragma solidity ^0.4.5;

contract MegaTokenBank{
   mapping(address => uint256) public Ledger;
   uint256 constant PRICE_PER_TOKEN = 10000;

   function MegaTokenBank(address _player) public payable {
      require(msg.value == 1);
   }

   function buy(uint256 numTokens) public payable {
      require(msg.value == numTokens * PRICE_PER_TOKEN);

      Ledger[msg.sender] += numTokens;
   }

   function sell(uint256 numTokens) public {
      require(balanceOf[msg.sender] >= numTokens);

      Ledger[msg.sender] -= numTokens;
      msg.sender.transfer(numTokens * PRICE_PER_TOKEN);
   }
}
```

Problem:
Arithmetic Results in Integer Overflow

Solution
Ensure sanity checks are applied after arithmetic

Consider a library like **SafeMath**
(Source: https://github.com/OpenZeppelin/openzeppelin-solidity/tree/master/contracts/math)

Source: https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-101

Exposed Private Data

There are no secrets on the blockchain

```
pragma solidity ^0.4.5;

contract SecretHolder {
   uint256 constant MySecretValue=
0xABCDEF1010;

    function GetSecret() public
payable {
        require(msg.sender = owner);
    }

}
```

Problem:
The **World State** is stored in each synced node.

Hence your secret value is available by manual inspection

Unprotected Self-Destruct (SWC-106)

```
contract SuicideMultiTxFeasible {
    uint256 private initialized = 0;
    uint256 public count = 1;

    function init() public {
        initialized = 1;
    }

    function run(uint256 input) {
        if (initialized == 0) {
            return;
        }

        selfdestruct(msg.sender);
    }
}
```

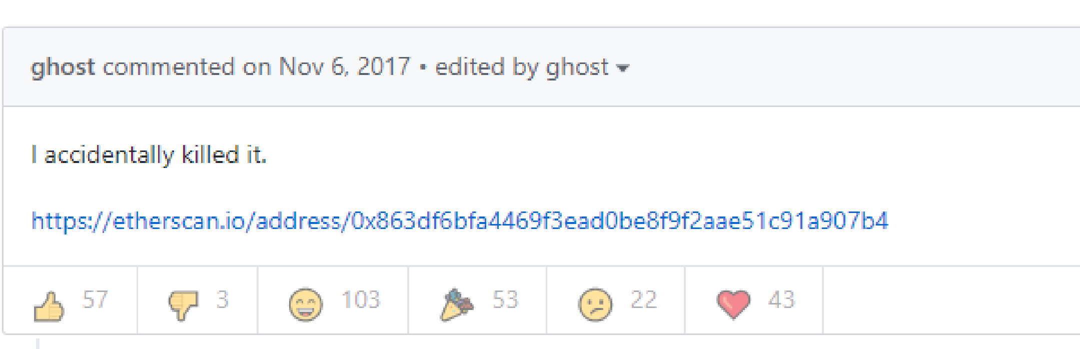
Problem:
The self-destruct will destroy the contract and freeze any ether attached to the contract address.

Whether it's $1 dollar or $150 Million dollars

https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-106

Unprotected Self-Destruct (SWC-106) - Parity

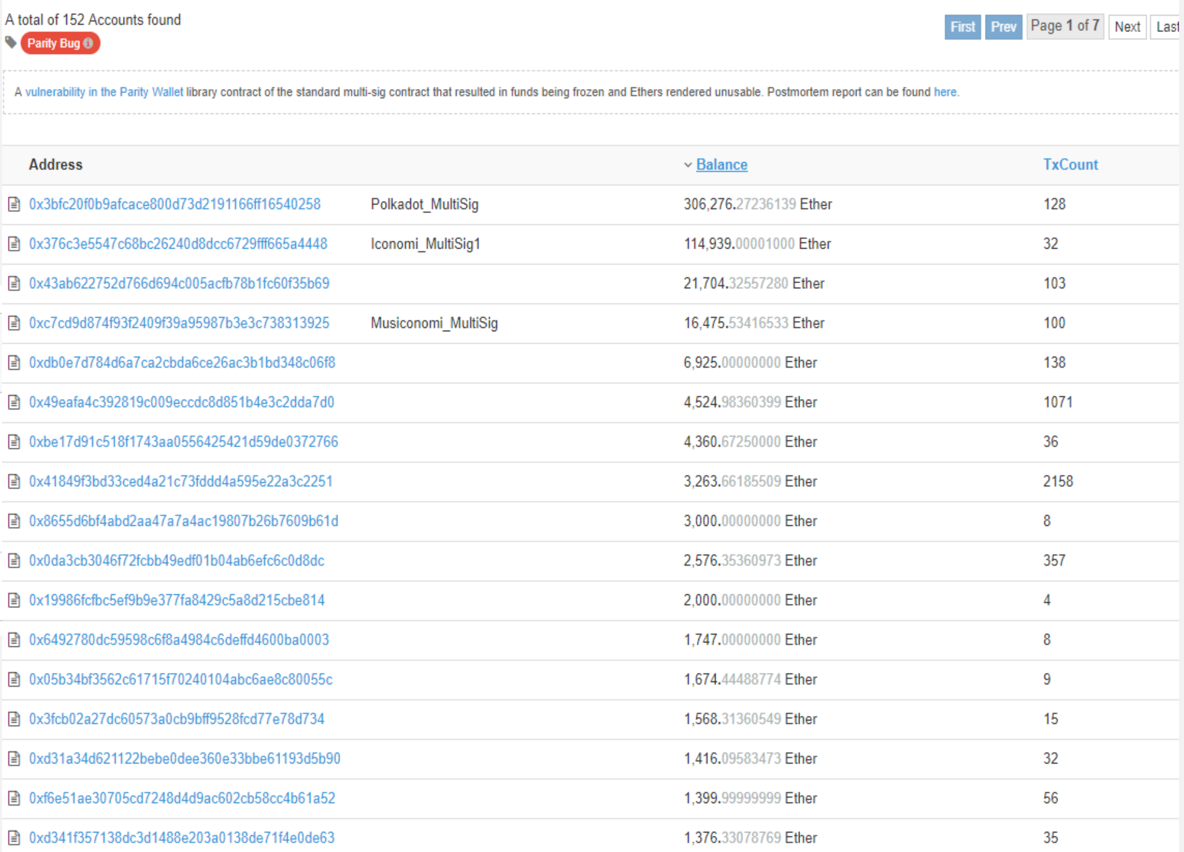
“anyone can kill your contract #6995” – devops199
https://github.com/paritytech/parity-ethereum/issues/6995



ghost commented on Nov 6, 2017 • edited by ghost ▾

I accidentally killed it.

https://etherscan.io/address/0x863df6bfa4469f3ead0be8f9f2aae51c91a907b4



A total of 152 Accounts found

Parity Bug

A vulnerability in the Parity Wallet library contract of the standard multi-sig contract that resulted in funds being frozen and Ethers rendered unusable. Postmortem report can be found here.

Address		Balance	TxCount
0x3bfc20f0b9afcace800d73d2191166ff16540258	Polkadot_MultiSig	306,276.27236139 Ether	128
0x376c3e5547c68bc26240d8dcc6729fff665a4448	Iconomi_MultiSig1	114,939.00001000 Ether	32
0x43ab622752d766d694c005acfb78b1fc60f35b69		21,704.32557280 Ether	103
0xc7cd9d874f93f2409f39a95987b3e3c738313925	Musiconomi_MultiSig	16,475.53416533 Ether	100
0xdb0e7d784d6a7ca2cbda6ce26ac3b1bd348c06f8		6,925.00000000 Ether	138
0x49eafa4c392819c009eccdc8d851b4e3c2dda7d0		4,524.98360399 Ether	1071
0xbe17d91c518f1743aa0556425421d59de0372766		4,360.67250000 Ether	36
0x41849f3bd33ced4a21c73fddd4a595e22a3c2251		3,263.66185509 Ether	2158
0x8655d6bf4abd2aa47a7a4ac19807b26b7609b61d		3,000.00000000 Ether	8
0x0da3cb3046f72fcbb49edf01b04ab6efc6c0d8dc		2,576.35360973 Ether	357
0x19986fcfbc5ef9b9e377fa8429c5a8d215cbe814		2,000.00000000 Ether	4
0x6492780dc59598c6f8a4984c6deffd4600ba0003		1,747.00000000 Ether	8
0x05b34bf3562c61715f70240104abc6ae8c80055c		1,674.44488774 Ether	9
0x3fcb02a27dc60573a0cb9bff9528fcd77e78d734		1,568.31360549 Ether	15
0xd31a34d621122bebe0dee360e33bbe61193d5b90		1,416.09583473 Ether	32
0xf6e51ae30705cd7248d4d9ac602cb58cc4b61a52		1,399.99999999 Ether	56
0xd341f357138dc3d1488e203a0138de71f4e0de63		1,376.33078769 Ether	35

Roughly 150-300 Million remains “Frozen”

Source: https://etherscan.io/address/0x863df6bfa4469f3ead0be8f9f2aae51c91a907b4#code

Bad Randomness (SWC-120)

On the blockchain nothing is truly random

```
/*
 * @source:
https://capturetheether.com/challenges/lotteries/guess-the-
random-number/
 * @author: Steve Marx
 */

pragma solidity ^0.4.21;

contract GuessTheRandomNumberChallenge {
    uint8 answer;

    function GuessTheRandomNumberChallenge() public payable
{
        require(msg.value == 1 ether);
        answer =
uint8(keccak256(block.blockhash(block.number - 1), now));
    }

    function isComplete() public view returns (bool) {
        return address(this).balance == 0;
    }

    function guess(uint8 n) public payable {
        require(msg.value == 1 ether);

        if (n == answer) {
            msg.sender.transfer(2 ether);
        }
    }
}
```

Problems:
Miners can manipulate block numbers.
PC are far faster than Ethereum and can “run ahead” of the block chain.

Source: https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-120

Bad Randomness (SWC-120)

On the blockchain nothing is truly random

```
// Stage one commit
// Guess the modulo of the blockhash 20 blocks from your guess
    function guess(uint8 _guess) public payable {
        require(msg.value == 1 ether);
        commitedGuess = _guess;
        commitBlock = block.number;
        guesser = msg.sender;
    }
    function recover() public {
      //This must be called after the guessed block and before
commitBlock+20's blockhash is unrecoverable
      require(block.number > commitBlock + 20 && commitBlock+20
> block.number - 256);
      require(guesser == msg.sender);

      if(uint(blockhash(commitBlock+20)) == commitedGuess){
        msg.sender.transfer(2 ether);
      }
    }
```

Solution:
Only generate the “random” number AFTER the guesses are committed.

This call RANDAO or Commit Pattern.

Source: https://github.com/randao/randao

Source: https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-120

Re-Entrancy (SWC-107)

```
/*
 * @source: http://blockchain.unica.it/projects/ethereum-survey/attacks.html#simpledao
 * @author: Atzei N., Bartoletti M., Cimoli T
 * Modified by Josselin Feist
 */
pragma solidity 0.4.24;

contract SimpleDAO {
  mapping (address => uint) public credit;

  function donate(address to) payable public{
    credit[to] += msg.value;
  }

  function withdraw(uint amount) public{
    if (credit[msg.sender]>= amount) {
      require(msg.sender.call.value(amount)()); // Calls Sender Code
      credit[msg.sender]-=amount;
    }
  }

  function queryCredit(address to) view public returns(uint){
    return credit[to];
  }
}
```

Problem:
Ether is sent via call on the senders amount() function before it is actually deducted of the balance.

Withdraw can be called over and over again in amount() before the amount is deducted.

Source: https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-107

Re-Entrancy (SWC-107)

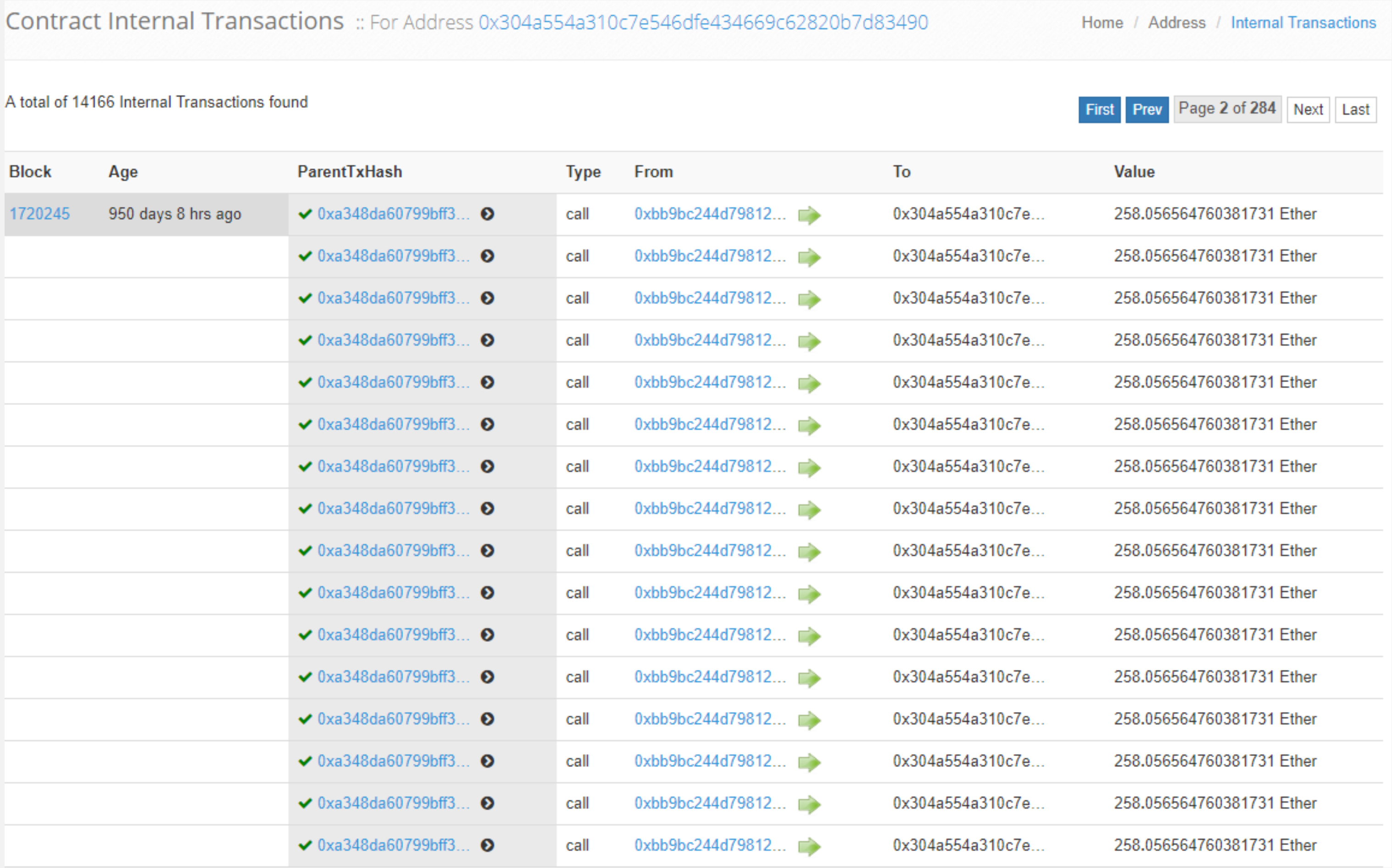
```
 function withdraw(uint amount) public{
   if (credit[msg.sender]>= amount) {
       credit[msg.sender]-=amount; // Update Balance First
       require(msg.sender.call.value(amount)()); // Calls Sender
Code
    }
 }

 function queryCredit(address to) view public returns(uint){
   return credit[to];
 }
}
```

Solution:
Update value before calling sender contracts code.

Ideally use send() or transfer() as opposed to calling the senders code

Source: https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-107

The DAO Hack – Re-Entrancy



Contract Internal Transactions :: For Address 0x304a554a310c7e546dfe434669c62820b7d83490

A total of 14166 Internal Transactions found

Block	Age	ParentTxHash	Type	From	To	Value
1720245	950 days 8 hrs ago	✓ 0xa348da60799bff3...	call	0xbb9bc244d79812... ➡	0x304a554a310c7e...	258.056564760381731 Ether
		✓ 0xa348da60799bff3...	call	0xbb9bc244d79812... ➡	0x304a554a310c7e...	258.056564760381731 Ether
		✓ 0xa348da60799bff3...	call	0xbb9bc244d79812... ➡	0x304a554a310c7e...	258.056564760381731 Ether
		✓ 0xa348da60799bff3...	call	0xbb9bc244d79812... ➡	0x304a554a310c7e...	258.056564760381731 Ether
		✓ 0xa348da60799bff3...	call	0xbb9bc244d79812... ➡	0x304a554a310c7e...	258.056564760381731 Ether
		✓ 0xa348da60799bff3...	call	0xbb9bc244d79812... ➡	0x304a554a310c7e...	258.056564760381731 Ether
		✓ 0xa348da60799bff3...	call	0xbb9bc244d79812... ➡	0x304a554a310c7e...	258.056564760381731 Ether
		✓ 0xa348da60799bff3...	call	0xbb9bc244d79812... ➡	0x304a554a310c7e...	258.056564760381731 Ether
		✓ 0xa348da60799bff3...	call	0xbb9bc244d79812... ➡	0x304a554a310c7e...	258.056564760381731 Ether
		✓ 0xa348da60799bff3...	call	0xbb9bc244d79812... ➡	0x304a554a310c7e...	258.056564760381731 Ether
		✓ 0xa348da60799bff3...	call	0xbb9bc244d79812... ➡	0x304a554a310c7e...	258.056564760381731 Ether
		✓ 0xa348da60799bff3...	call	0xbb9bc244d79812... ➡	0x304a554a310c7e...	258.056564760381731 Ether
		✓ 0xa348da60799bff3...	call	0xbb9bc244d79812... ➡	0x304a554a310c7e...	258.056564760381731 Ether
		✓ 0xa348da60799bff3...	call	0xbb9bc244d79812... ➡	0x304a554a310c7e...	258.056564760381731 Ether
		✓ 0xa348da60799bff3...	call	0xbb9bc244d79812... ➡	0x304a554a310c7e...	258.056564760381731 Ether
		✓ 0xa348da60799bff3...	call	0xbb9bc244d79812... ➡	0x304a554a310c7e...	258.056564760381731 Ether

Source: https://etherscan.io

- Abused “split” function of DAO contract
- $3.6 million Ether stolen
- $420 million to date
- Due to the way the contract was structured a 27 day hold was in place
- Community majority (89%) voted to “Hard Fork” (creating the divide between Ether and Ether Classic)
- Actors who stole the ether were actively involved in trying to the influence the community to <u>not</u> hard fork

Unsafe Authorization (SWC-115)

```
contract MyContract {

    address owner;

    function MyContract() public {
        owner = msg.sender; // Properly set in constructor
    }

    function sendTo(address receiver, uint amount) public
{
        require(tx.origin == owner); // Improper Check
        receiver.transfer(amount);
    }

}
```

Problem:
A crafted blockheader with chosen tx.origin may be mined

If the block is “mined” a an actor may take over the contract then.

Source: https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-115

Unsafe Authorization (SWC-115)

```
contract MyContract {

    address owner;

    function MyContract() public {
        owner = msg.sender; // Properly set in constructor
    }

    function sendTo(address receiver, uint amount) public
{
        require(msg.sender == owner); // Improper Check
        receiver.transfer(amount);
    }

}
```

Solution:
Use msg.sender to validate who sent the message

Source: https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-115

Unsafe Contract Constructors (SWC-118)

```
/*
 * @source: https://github.com/trailofbits/not-so-smart-
contracts/blob/master/wrong_constructor_name/incorrect_constructor.sol
 * @author: Ben Perez
 * Modified by Gerhard Wagner
 */


pragma solidity 0.4.24;

contract Missing{
    address private owner;

    modifier onlyowner {
        require(msg.sender==owner);
        _;
    }

    function missing()
        public
    {
        owner = msg.sender;
    }

    function () payable {}

    function withdraw()
        public
        onlyowner
    {
       owner.transfer(this.balance);
    }
}
```

Problem:
By mis-spelling the constructor name a default constructor is auto-generated without the expected checks.

Source: https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-118

Unsafe Contract Constructors (SWC-118)

```
/*
 * @source: https://github.com/trailofbits/not-so-smart-
contracts/blob/master/wrong_constructor_name/incorrect_constructor.sol
 * @author: Ben Perez
 * Modified by Gerhard Wagner
 */


pragma solidity 0.4.24;

contract Missing{
    address private owner;

    modifier onlyowner {
        require(msg.sender==owner);
        _;
    }

    function missing()
        public
    {
        owner = msg.sender;
    }

    function () payable {}

    function withdraw()
        public
        onlyowner
    {
       owner.transfer(this.balance);
    }
}
```

Solution:
Making sure the names match in spelling and case. Review output from static analysis tools and compiler.

Source: https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-118

Out-Of-Bounds Write-Anywhere (SWC-124)

```
function UpdateLedgerAtIndex(uint idx, uint entry) public {
                Ledger[idx] = entry;
 }
```

Problem:
Without appropriate bounds check index offsets called directly or arrays will write into nearby storage.

Often this includes over-writing the owner variable potentially changing the owner of the contract or modify other information on the stack.

Will Smart Contract Control Flow Exploitation become a thing? (We haven’t seen the first buffer overflow yet).

Source: https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-124

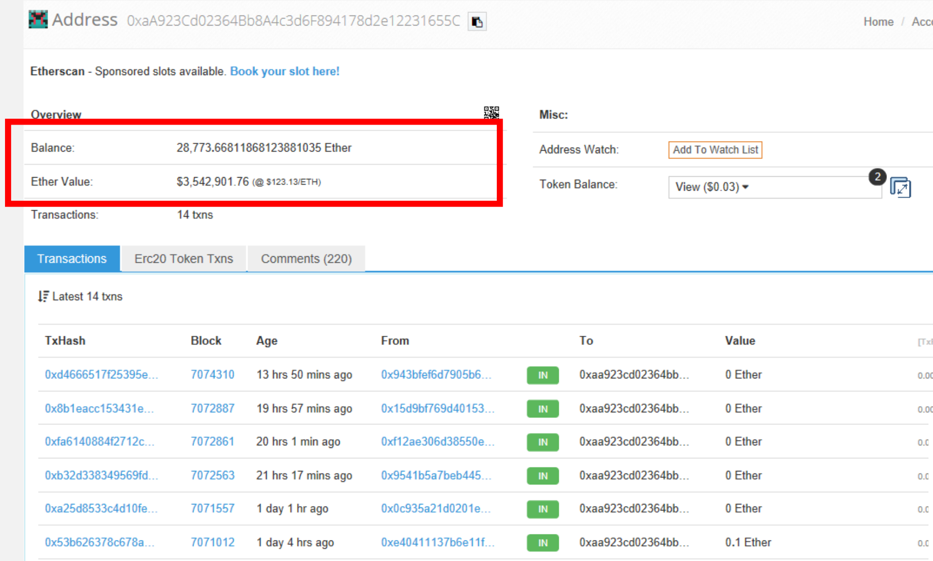
Out-Of-Bounds Write-Anywhere (SWC-124)

```
function UpdateLedgerAtIndex(uint idx, uint entry) public {
        require(idx < Ledger.length);
        Ledger[idx] = entry;
 }
```

Solution:
Ensure adequate bounds checking

Source: https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-124

And of course, there’s exchange hacks!



Address 0xaA923Cd02364Bb8A4c3d6F894178d2e12231655C

Etherscan - Sponsored slots available. Book your slot here!

Overview

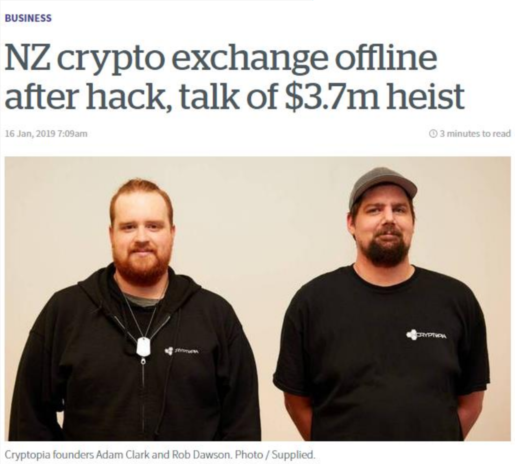
Balance:	28,773.66811868123881035 Ether
Ether Value:	$3,542,901.76 (@ $123.13/ETH)
Transactions:	14 txns

Misc:

Address Watch:	Add To Watch List
Token Balance:	View ($0.03)

Latest 14 txns

TxHash	Block	Age	From		To	Value	[TxFee]
0xd4666517f25395e...	7074310	13 hrs 50 mins ago	0x943bfef6d7905b6...	IN	0xaa923cd02364bb...	0 Ether	0.0001008
0x8b1eacc153431e...	7072887	19 hrs 57 mins ago	0x15d9bf769d40153...	IN	0xaa923cd02364bb...	0 Ether	0.00010959375
0xfa6140884f2712c...	7072861	20 hrs 1 min ago	0xf12ae306d38550e...	IN	0xaa923cd02364bb...	0 Ether	0.0
0xb32d338349569fd...	7072563	21 hrs 17 mins ago	0x9541b5a7beb445...	IN	0xaa923cd02364bb...	0 Ether	0.0
0xa25d8533c4d10fe...	7071557	1 day 1 hr ago	0x0c935a21d0201e...	IN	0xaa923cd02364bb...	0 Ether	0.0
0x53b626378c678a...	7071012	1 day 4 hrs ago	0xe40411137b6e11f...	IN	0xaa923cd02364bb...	0.1 Ether	0.0
	2	1 day 7 hrs ago	0xc8b75986014954...	IN	0xaa923cd02364bb...	88.585373952818115 Ether	0.0
	2	1 day 8 hrs ago	0xc8b75986014954...	IN	0xaa923cd02364bb...	7,507.851507620849139 Ether	0.0
	4	2 days 11 hrs ago	0xc8b75986014954...	IN	0xaa923cd02364bb...	3.173312292029139 Ether	0.0
	4	2 days 12 hrs ago	0xc8b75986014954...	IN	0xaa923cd02364bb...	1,499.756017054949832 Ether	0.0



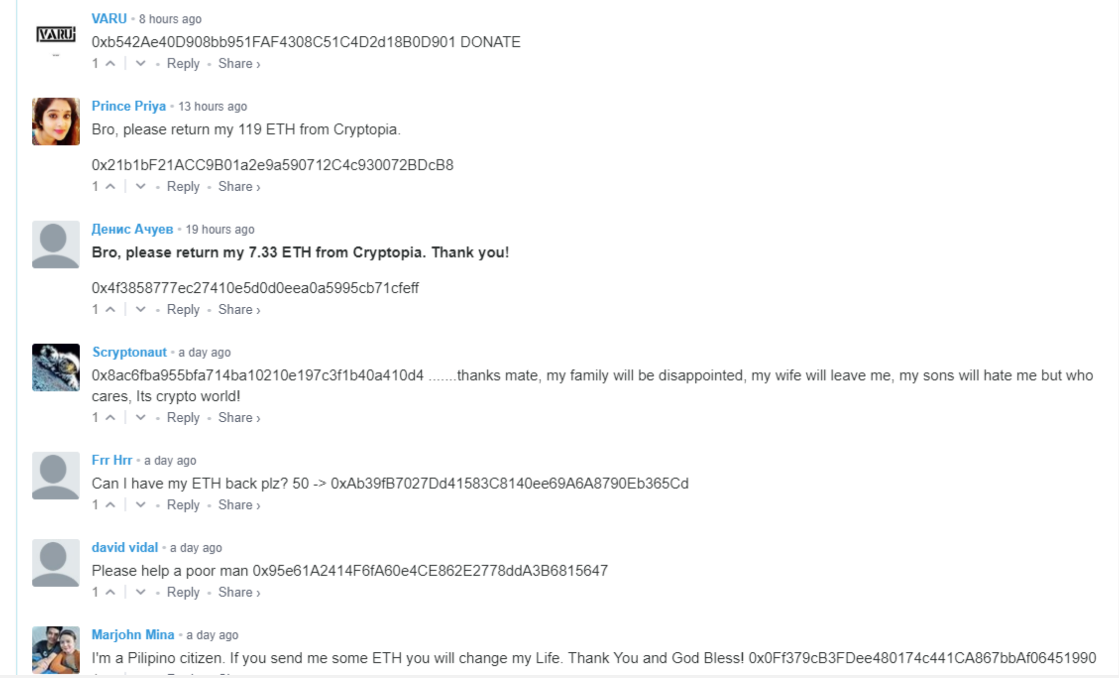
BUSINESS

NZ crypto exchange offline after hack, talk of $3.7m heist

16 Jan, 2019 7:09am

3 minutes to read

Cryptopia founders Adam Clark and Rob Dawson. Photo / Supplied.



VARU • 8 hours ago

0xb542Ae40D908bb951FAF4308C51C4D2d18B0D901 DONATE

1 ^ | v • Reply • Share ›

Prince Priya • 13 hours ago

Bro, please return my 119 ETH from Cryptopia.

0x21b1bF21ACC9B01a2e9a590712C4c930072BDcB8

1 ^ | v • Reply • Share ›

Денис Ачуев • 19 hours ago

Bro, please return my 7.33 ETH from Cryptopia. Thank you!

0x4f3858777ec27410e5d0d0eea0a5995cb71cfeff

1 ^ | v • Reply • Share ›

Scryptonaut • a day ago

0x8ac6fba955bfa714ba10210e197c3f1b40a410d4 ....thanks mate, my family will be disappointed, my wife will leave me, my sons will hate me but who cares, Its crypto world!

1 ^ | v • Reply • Share ›

Frr Hrr • a day ago

Can I have my ETH back plz? 50 -> 0xAb39fB7027Dd41583C8140ee69A6A8790Eb365Cd

1 ^ | v • Reply • Share ›

david vidal • a day ago

Please help a poor man 0x95e61A2414F6fA60e4CE862E2778ddA3B6815647

1 ^ | v • Reply • Share ›

Marjohn Mina • a day ago

I'm a Pilipino citizen. If you send me some ETH you will change my Life. Thank You and God Bless! 0x0Ff379cB3FDee480174c441CA867bbAf06451990

Honey Pots

Contracts that appear vulnerable but are not

- Just have to send a little bit of Ether in… ☺
- Use of anti-disassembly tricks to hinder analysis

Great talk on research to detect such contracts

- Smart Contracts honeypots for profit (and probably fun) - Ben Schimdt
- *Source: https://www.youtube.com/watch?v=Lj0J7_a1AVQ*

Security Tools

IDE

- Remix (online IDE) - https://remix.ethereum.org/

Smart Contract Static Analysis

- Slither - https://github.com/trailofbits/slither

Smart Contract Dynamic Analysis (Symbolic Execution)

- Mithril Classic - https://github.com/ConsenSys/mythril-classic
- Manticore - https://github.com/trailofbits/manticore

Smart Contract Dynamic Analysis (Fuzzing)

- Echidna - https://github.com/trailofbits/echidna

To The Future

- Smart Contract development is still very new
- Increased use of design patterns in Smart Contract development to address challenges like upgrading
- Educate developers on types of weaknesses
- Better tooling
- Use of standards when implementing Tokens (ERC* series tokens)

References

1) Smart Contract Weakness Classification
https://smartcontractsecurity.github.io/SWC-registry/

2) Trail Of Bits – Not So Smart Contracts
https://github.com/trailofbits/not-so-smart-contracts

3) Smashing Ethereum Smart Contracts for Fun and ACTUAL Profit
https://github.com/b-mueller/smashing-smart-contracts

4) Smart Contract Best Practices
https://consensys.github.io/smart-contract-best-practices/

5) Ethereum Yellow and Beige Papers
Yellow Paper - http://gavwood.com/paper.pdf
Beige Paper - https://github.com/chronaeon/beigepaper

Challenges!

1) Capture The Ether (By Steve Marx @smarx)

https://capturetheether.com/challenges/

2) Security Innovation Blockchain CTF (By Security Innovation)

https://blockchain-ctf.securityinnovation.com/

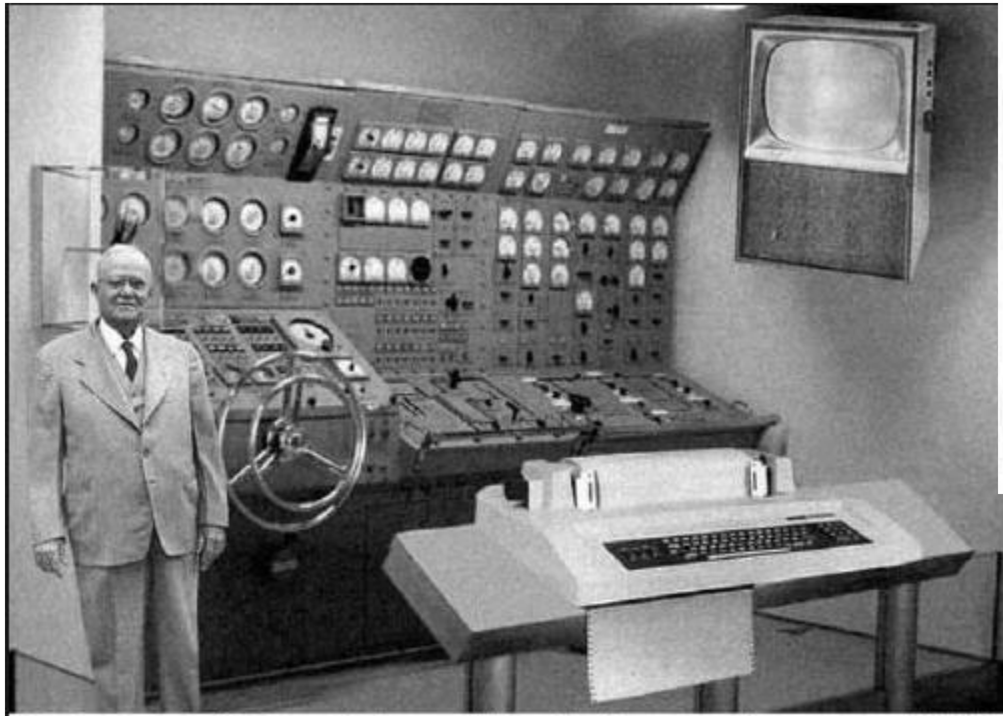
3) EtherNaut CTF (@ZeppelinOrg)

https://ethernaut.zeppelin.solutions/

Thank you!

- Thank you to Judy (@daarkprincess) for bringing the cookies!
- Thank you to OWASP Toronto and George Brown for hosting!
- Thank you to everyone for attending!

Questions?



*Scientists from the **RAND** Corporation have created this model to illustrate how a "home computer" could look like in the year 2004. However the needed technology will not be economically feasible for the average home. Also the scientists readily admit that the computer will require not yet invented technology to actually work. but 50 years from now scientific progress is expected to solve these problems. With teletype interface and the Fortran language. the computer will be easy to use.*

I'm listening…