16 June 2018

Reverse Engineering Of Blockchain Smart Contracts

REcon Montreal 2018

QuoScient

Whoami

Patrick Ventuzelo

@Pat_Ventuzelo

QuoScient GmbH

- Security Researcher/Engineer

Quolab

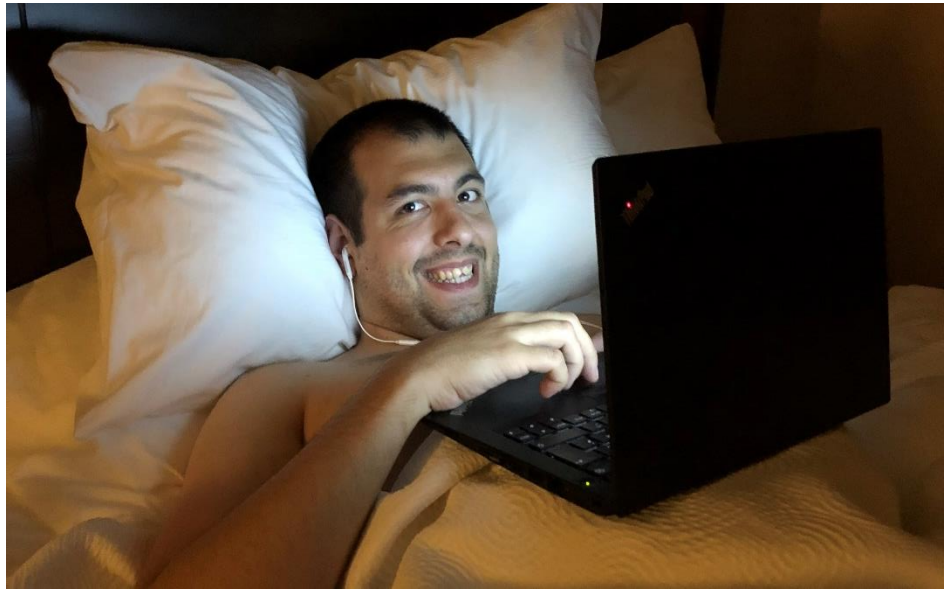
- Threat Intel & Response Platform
- Collaborative, Decentralized

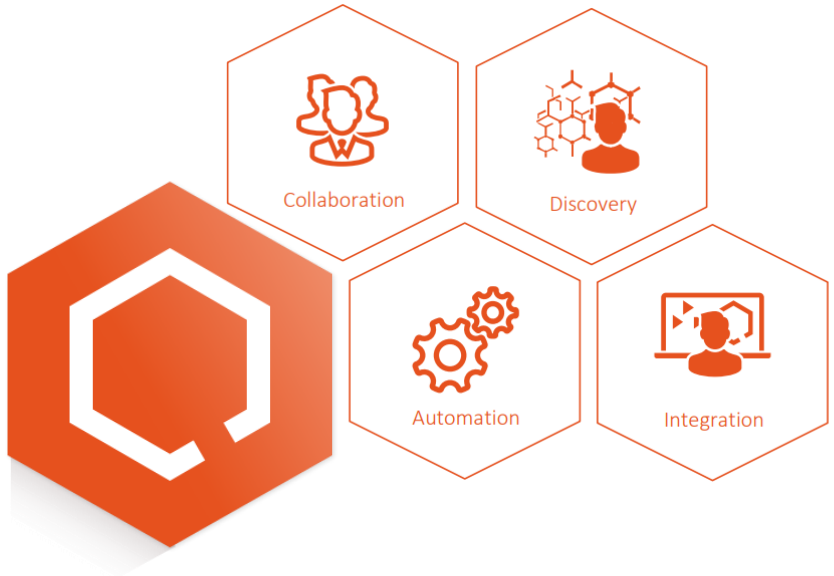
Research

- Smart contracts, WebAssembly, …

Support Intelligence Operations

- Transaction tracking
- Vulnerability analysis





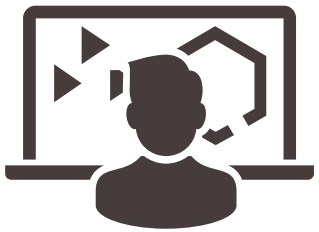
Smart contracts analysis for…

- Users/ICO
  - Due diligence
  - Understand the Logic

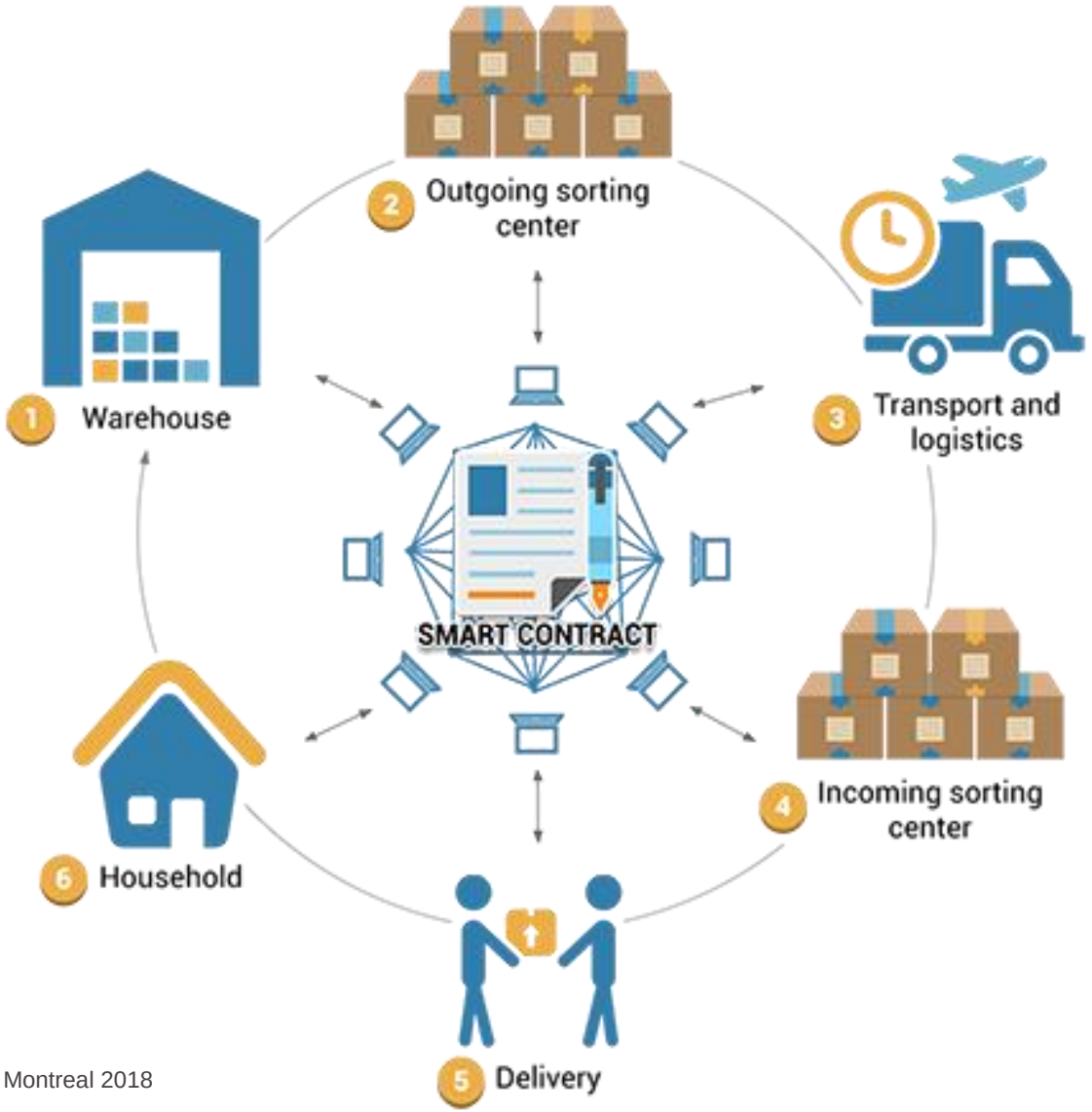
- Security researcher
  - Bug hunting
  - Vulnerability research

- Company
  - Security audit
  - Bytecode Optimization

- Threat intelligence team
  - Transaction tracking
  - Analyze smart contract interactions



Smart contract applications



Decentralized Escrow

by Random Directions

how to ditch a third party

Alice wants to sell a widget for $100.

Bob wants to buy a widget for $100.

Alice puts an ad for selling a widget, posting $10 as collateral.

An 2-2 transaction is created that gives Alice $110 and Bob $5.

Alice signs her half and ships the widget.

The funds are deposited into the address only after **both parties sign.**

Alice puts $10, and Bob puts $105 into a 2-2 multisig address.

Bob retains the power to not sign.

Here, both parties lose their collateral.

Bob signs his half when he receives the widget.

The 2-2 transaction completes, and

Alice is left with a net of $100, and Bob with his new widget.

brought to you by :

Random Directions

Today's talk…

- **WILL NOT** be about:
  - Blockchain concepts (PoW/PoS/BTF/Merkle trees/Mining)
  - Cryptography (SHA-3/Keccak/RIPEMD160)
  - TheDAO/Parity bug/[Insert Your Hack Here]
  - If you should HOLD or NOT…

- **WILL BE** about:
  - Smart contracts
  - VM implementations
  - Assembly language & Instructions set
  - Disassembly / CFG reconstruction

  - Octopus
    - https://github.com/quoscient/octopus

Top 100 Cryptocurrencies by Market Capitalization

Cryptocurrencies ▾ | Watchlist | USD ▾ | Next 100 → | View All

#	Name	Market Cap	Price	Volume (24h)	Circulating Supply	Change (24h)	Price Graph (7d)
1	Bitcoin	$131,630,663,771	$7,706.55	$4,817,210,000	17,080,362 BTC	0.86%	
2	Ethereum	$60,706,993,741	$607.52	$1,926,250,000	99,925,918 ETH	-0.39%	
3	Ripple	$26,563,609,126	$0.676926	$283,516,000	39,241,525,848 XRP *	0.19%	
4	Bitcoin Cash	$19,552,477,164	$1,138.72	$635,736,000	17,170,575 BCH	-0.76%	
5	EOS	$12,350,553,071	$13.78	$985,357,000	896,149,492 EOS *	-2.12%	
6	Litecoin	$6,884,229,680	$121.04	$356,839,000	56,876,598 LTC	-0.51%	
7	Stellar	$5,527,306,942	$0.297553	$52,481,500	18,575,873,684 XLM *	1.06%	
8	Cardano	$5,500,687,285	$0.212160	$83,969,500	25,927,070,538 ADA *	-2.43%	
9	IOTA	$4,764,253,882	$1.71	$80,402,000	2,779,530,283 MIOTA *	-0.07%	
10	TRON	$3,894,674,866	$0.059236	$178,789,000	65,748,111,645 TRX *	-0.98%	
11	NEO	$3,488,758,000	$53.67	$79,111,800	65,000,000 NEO *	-1.25%	

1.

3.

2.

ethereum

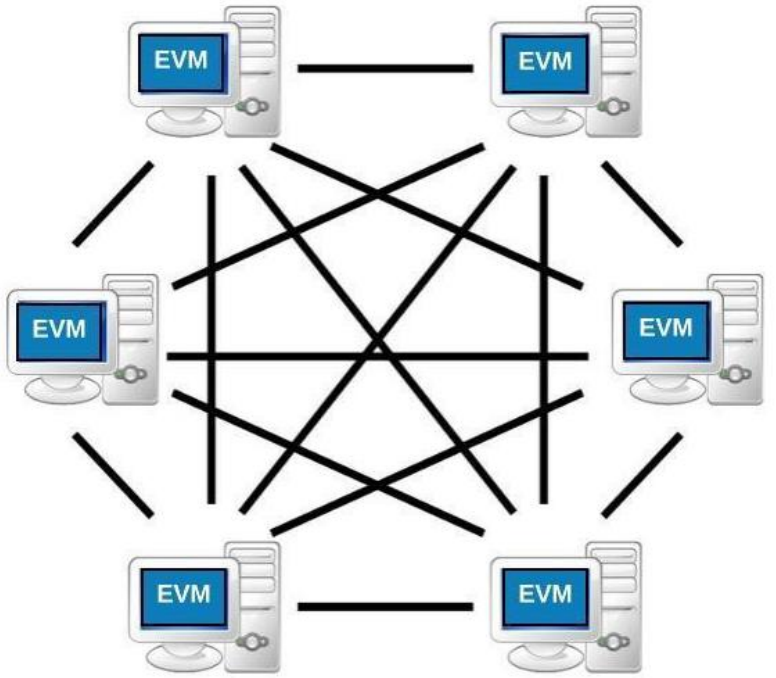
What is Ethereum?

- *“Ethereum is a decentralized platform that runs smart contracts: applications that run exactly as programmed without any possibility of downtime, censorship, fraud or third-party interference.”*

- Create by Vitalik Buterin & Gavin Wood - 2013
  - White paper: Description of the project
  - Yellow paper: Ethereum's formal specification (Technical)
  - Open source

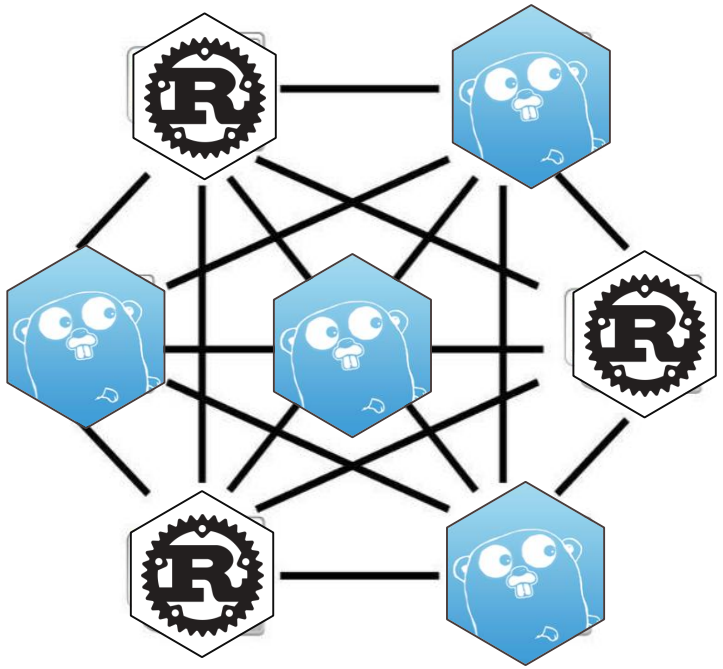
- Ethereum Virtual machine (EVM)

- Smart contracts
  - Application stored & execute on the blockchain
  - ĐApps (Decentralized Application)



Ethereum full node

- A node is:
  - Piece of Software
  - Connected to other nodes
  - Maintains locally a copy of the blockchain



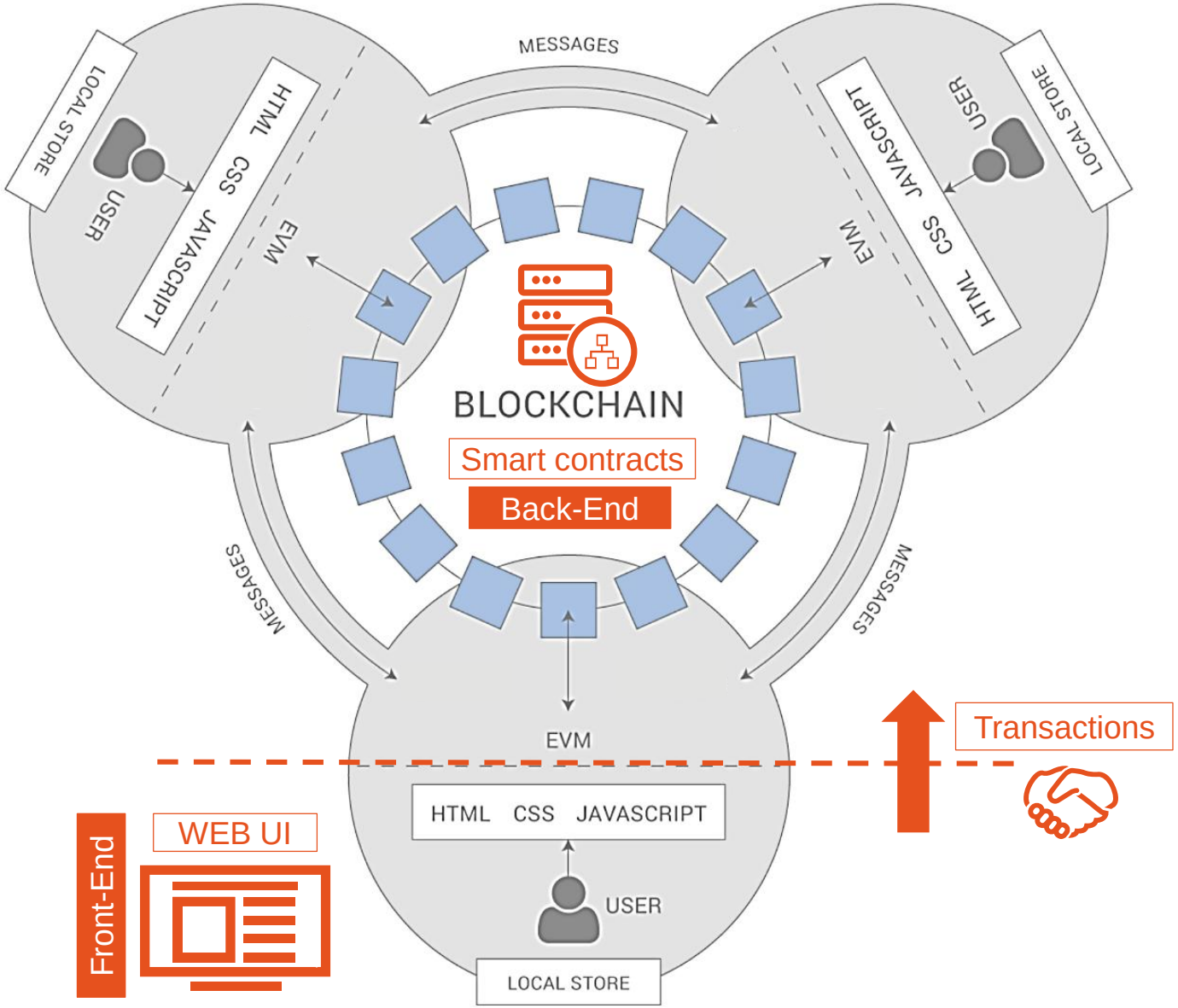
~ 60%

- Geth
  - Golang
  - Major full node
  - Ethereum foundation

~ 35%

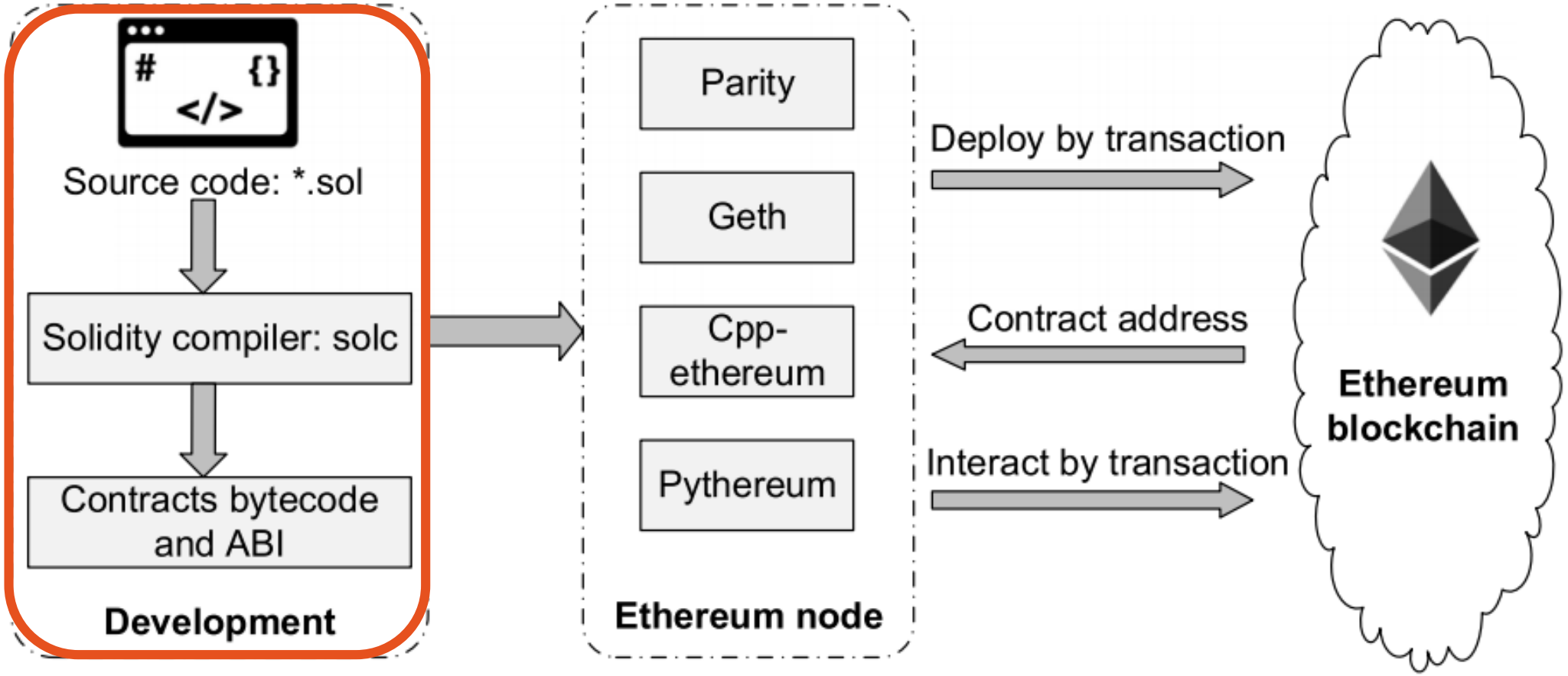
- Parity
  - Rust
  - Fast, secure, light
  - Parity team

Smart contract application - ĐApps



Smart contract creation process

Development, deployment & interaction



https://doi.org/10.1016/j.future.2017.08.020

Development languages

Solidity

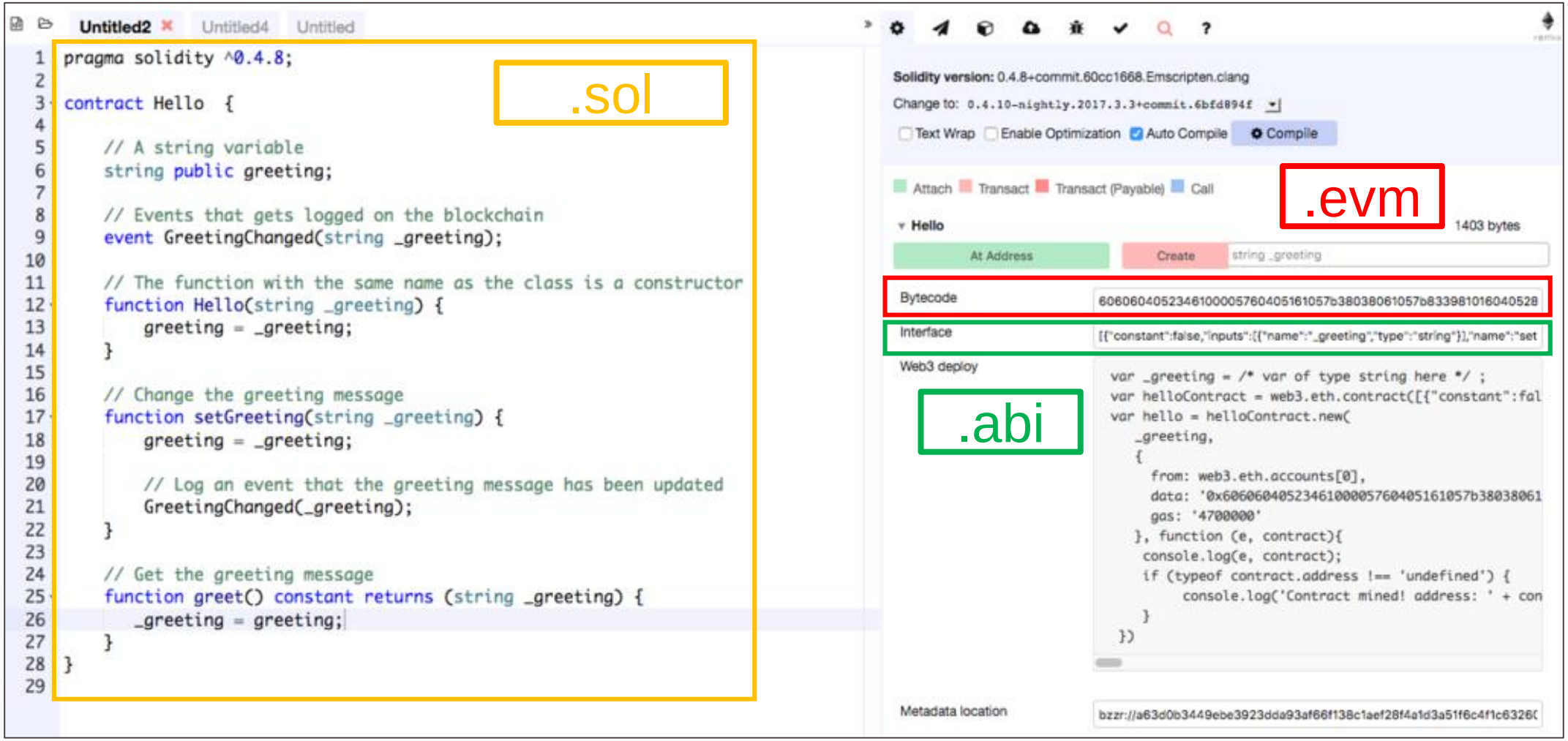
```
contract Mortal {
    /* Define variable owner of the type address */
    address owner;

    /* This function is executed at initialization
                and sets the owner of the contract */
    function Mortal() { owner = msg.sender; }

    /* Function to recover the funds on the contract */
    function kill() {
        if (msg.sender == owner)
            selfdestruct(owner);
    }
}

contract Greeter is Mortal {
    /* Define variable greeting of the type string */
    string greeting;

    /* This runs when the contract is executed */
    function Greeter(string _greeting) public {
        greeting = _greeting;
    }

    /* Main function */
    function greet() constant returns (string) {
        return greeting;
    }
}
```

Vyper

```
# Vyper Greeter Contract

greeting: bytes <= 20


@public
def __init__():
    self.greeting = "Hello"


@public
def setGreeting(x: bytes <= 20):
    self.greeting = x


@public
def greet() -> bytes <= 40:
    return self.greeting
```

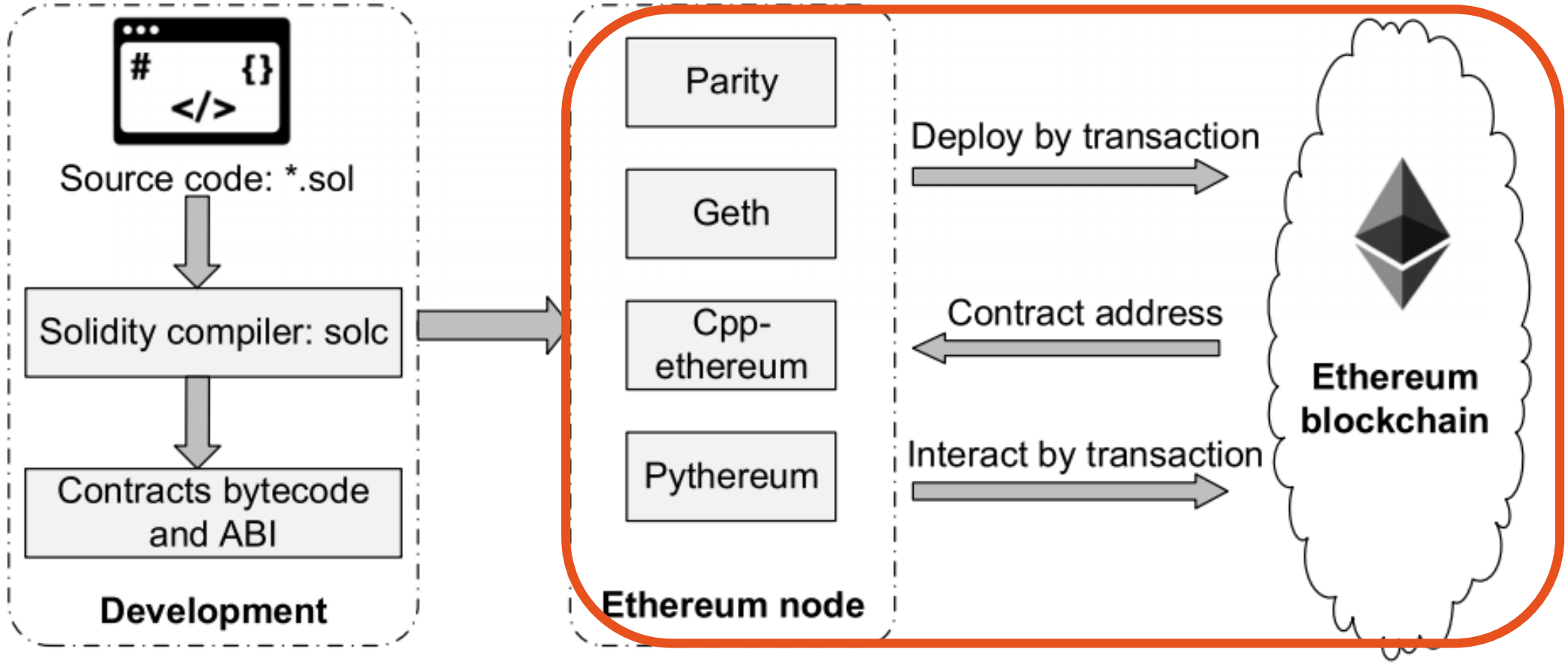
Remix

Online Web IDE for Solidity smart contracts development



Smart contract creation process

Development, deployment & interaction



https://doi.org/10.1016/j.future.2017.08.020

Existing contracts on etherscan.io

Only bytecode is mandatory to create a contract account

Bytecode decomposition

- Loader code
  - Run the contract constructor
  - Execute once to store the runtime code on the blockchain
  - Can be present in “Contract creation code” on etherscan.io

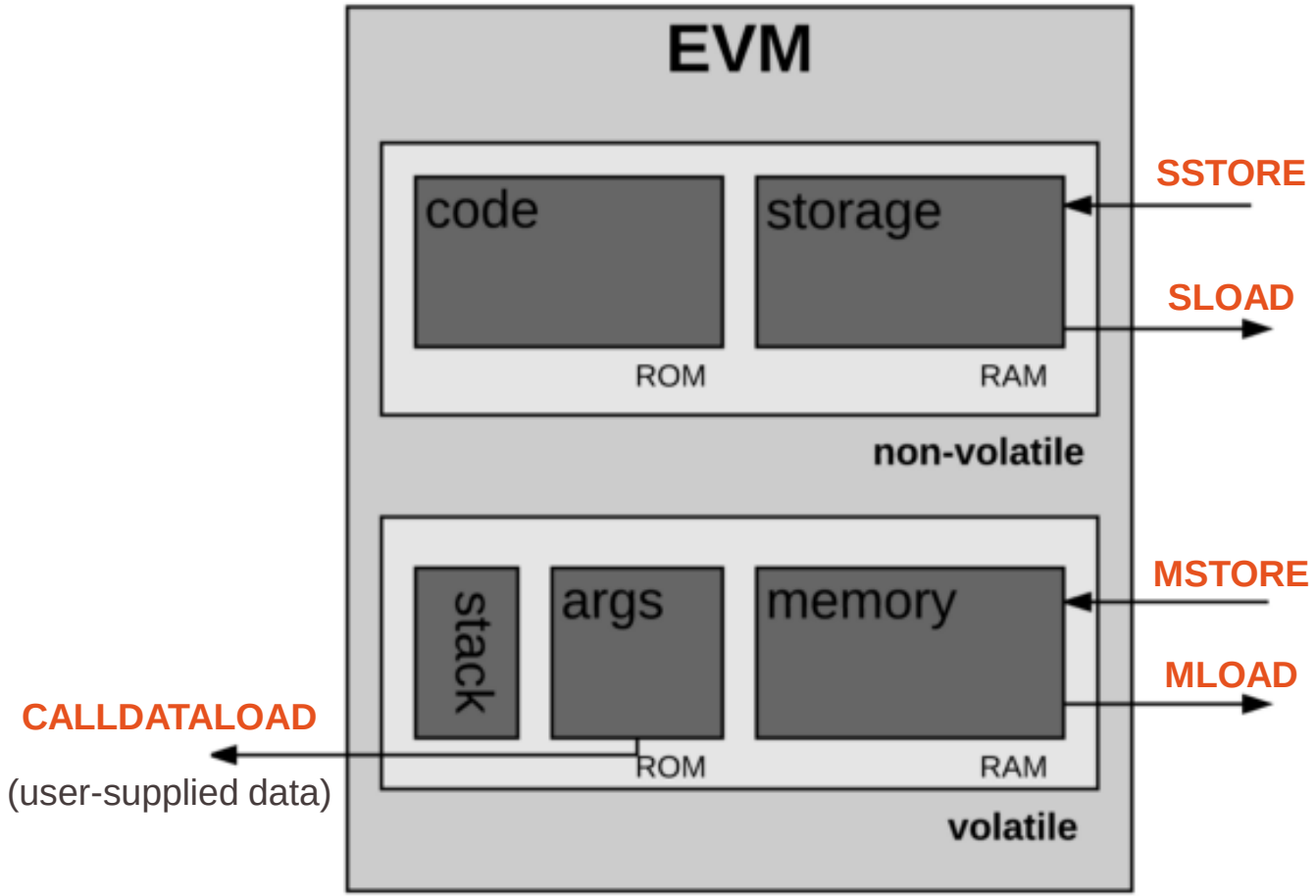
- Runtime code
  - Stored on the blockchain
  - Executed for each transaction with the contract

- Swarm Hash (a.k.a. bzzhash)
  - Merkle tree hash use to retrieve the content of the associated persistent storage of the contract
  - Concatenated at the end of the code
  - Magic number: 0x627a7a72 (bzzr)

```
6080604052348015610010576000 80fd5b5060405161039b380
38061039b8339810180604052810190808051820192919050 50
5033600080610100 0a81548173ffffffffffffffffffffffff
ffffffffffffffff02191690837 3ffffffffffffffffffffff
ffffffffffffffff1602179055508060019080519060200190 6
1008992919061009056 5b5050610135565b8280546001816001
16156101000203166002900490600052602060002090601f016
0209004801928260 1f106100d15780516 0ff19168380011785
556100ff565b828001600101855582156100ff579182015b828
111156100fe5782518255916020019190600101906100e3565b
5b50905061010c9190610110565b5090565b61013291905b808
2111561012e576000816000905550600101610116565b509056
5b90565b6102578061014460003960 00f300 6080604052 60043
61061004c576000357c0100000000000000000000000000000
0000000000000000000000000900463ffffffff16806341c0e
1b514610051578063cfae321714610068575b600080fd5b3480
1561005d57600080fd5b506100666100f8565b005b348015610
07457600080fd5b5061007d610189565b604051808060200182
8103825283818151815260200191508051906020019080838 36
0005b838110156100bd578082015181840152602081019050 61
00a2565b5050505090509081019060 1f1680156100ea578082 0
3805160018360200361010 00a031916815260200191505b5092
505050604051809103 90f35b600080905490610100 0a900473f
ffffffffffffffffffffffffffffffffffffffff1673ffffffff
ffffffffffffffffffffffffffffffff163373ffffffffffffff
ffffffffffffffffffffffff1614156101875760008090 54
906101000a900473ffffffffffffffffffffffffffffffffffff
ffff1673ffffffffffffffffffffffffffffffffffffffff16
ff5b565b6060600180546001816001161561010002031660029
00480601f0160208091040260200160405190810160405280 92
9190818152602001828054600181600116156101000203166 00
2900480156102215780601f106101f6576101008083540402 83
5291602001916102215 65b8201919060005260206000209 05b8
154815290600101906020018083116102045782900360 1f1682
01915b5050505050905090560 0a165627a7a72305820df97826
8dd1593a7bbc753bfb0404d8353b4c6ced383d8107c926d5003
e40c060029
```

Ethereum Virtual Machine

Architecture		
Stack machine		
Turing complete		
Instruction set		~180 Opcodes
Memory type		
Stack	volatile	byte-array (list [])
Memory	volatile	byte-array (list [])
Storage	persistent	key-value database (dictionary {})



EVM Instructions set

Opcodes value	Family	Examples
0x00 – 0x0B	Stop and Arithmetic Operations	**STOP**, ADD, SUB, MUL, **DIV**, EXP
0x10 – 0x1A	Comparison & Bitwise Logic Operations	**LT**, **GT**, **EQ**, **ISZERO**, AND, XOR
0x20	SHA3	**SHA3**
0x30 – 0x3E	Environmental Information	**ADDRESS**, **CALLER**, **CALLDATALOAD**
0x40 – 0x45	Block Information	**BLOCKHASH**, **COINBASE**, NUMBER
0x50 – 0x5B	Stack, Memory, Storage and Flow Operations	**POP**, MSTORE, **JUMP**, **JUMPI**, **JUMPDEST**
0x60 – 0x7F	Push Operations	**PUSH1** – PUSH32
0x80 – 0x8F	Duplication Operations	**DUP1** – DUP16
0x90 – 0x9F	Exchange Operations	**SWAP1** – SWAP16
0xA0 – 0xA4	Logging Operations	LOG0 – LOG4
0xF0 – 0xFF	System operations	CALL, **RETURN**, **DELEGATECALL**

Decoded Bytecode :

[1] PUSH1 0x80
[3] PUSH1 0x40
[4] MSTORE
[6] PUSH1 0x04
[7] CALLDATASIZE
[8] LT
[11] PUSH2 0x004c
[12] JUMPI
[14] PUSH1 0x00
[15] CALLDATALOAD
[45] PUSH29 0x0100000000000000000000000000000000000000000000000000000000
[46] SWAP1
[47] DIV
[52] PUSH4 0xffffffff
[53] AND
[54] DUP1
[59] PUSH4 0x41c0e1b5
[60] EQ
[63] PUSH2 0x0051
[64] JUMPI
[65] DUP1
[70] PUSH4 0xcfae3217
[71] EQ
[74] PUSH2 0x0068
[75] JUMPI
[76] JUMPDEST
[78] PUSH1 0x00
[79] DUP1

Control flow instructions

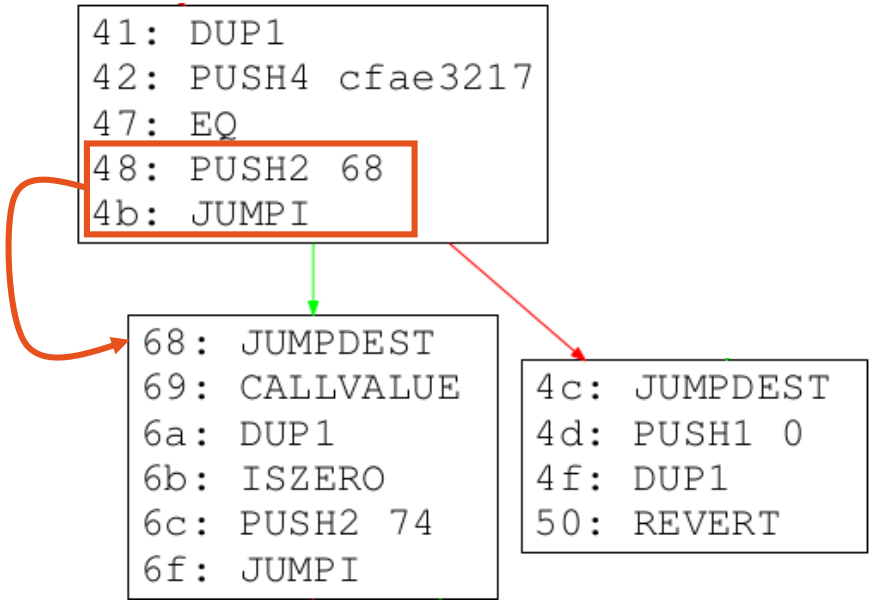
Opcode	Simplify description	Basic block position
JUMP	Unconditional jump	Last instruction
JUMPI	Conditional jump	Last instruction
RETURN , STOP INVALID SELFDESTRUCT , REVERT	Halt execution	Last instruction
JUMPDEST	Marks a position within the code that is a valid target destination for jumps	First instruction

EIP 615: Subroutines and Static Jumps for the EVM By *Greg Colvin*
New branch opcodes: JUMPTO, JUMPIF, JUMPSUB, JUMPSUBV, ….

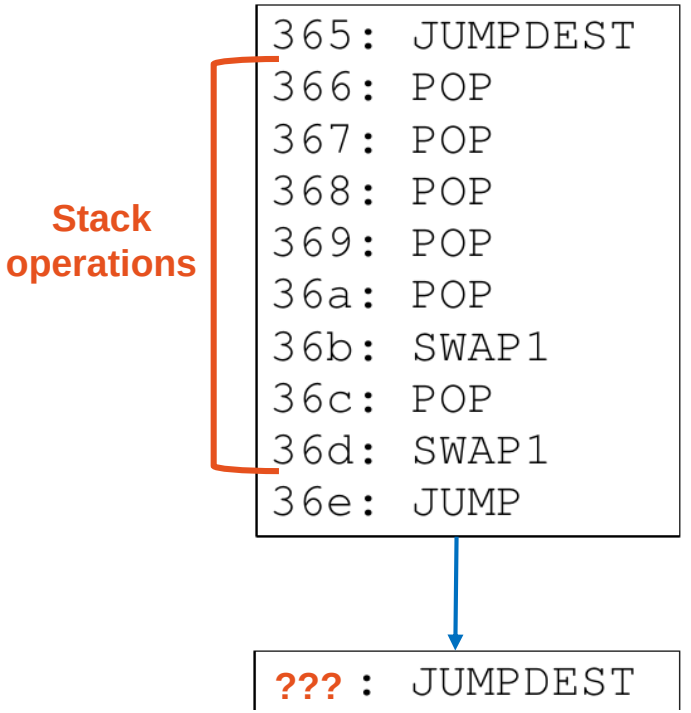
Edges identifications

Static analysis vs dynamic analysis

- Static analysis only works if:
  - Jump target offset is pushed on the stack
  - Just before the JUMP/I

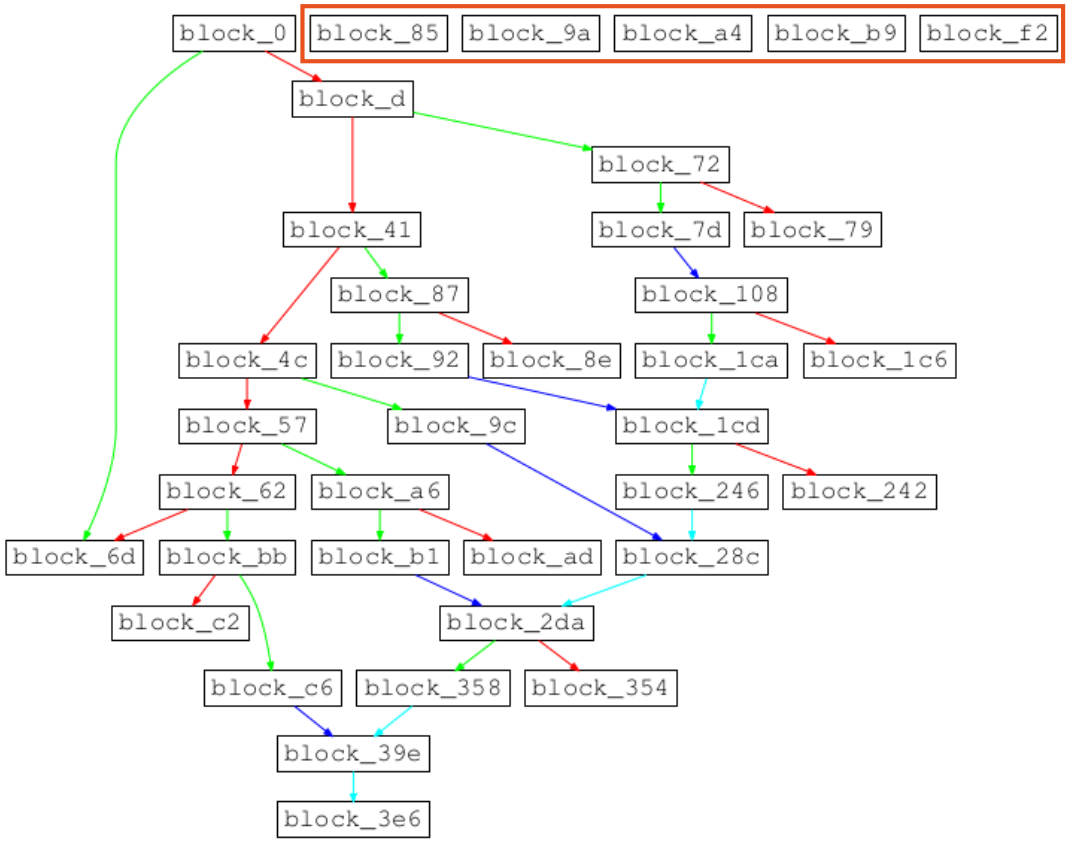


- And fails if:
  - Stack operations are used to put the jump target on top of the stack

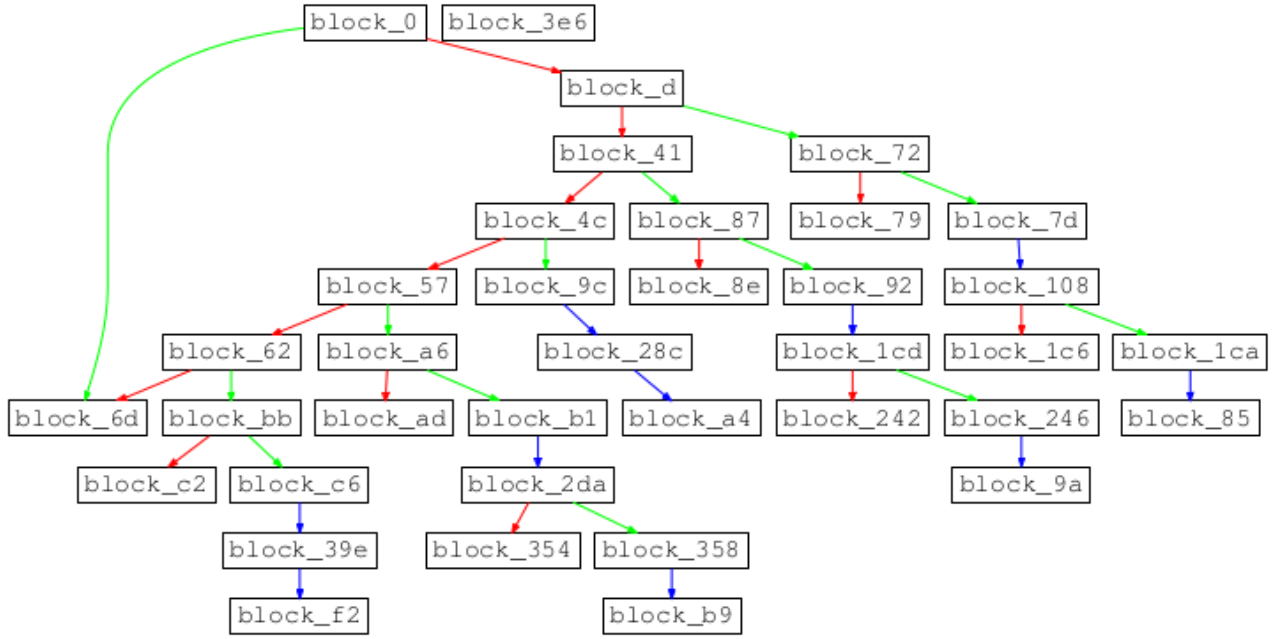


Control Flow Graph (CFG) reconstruction

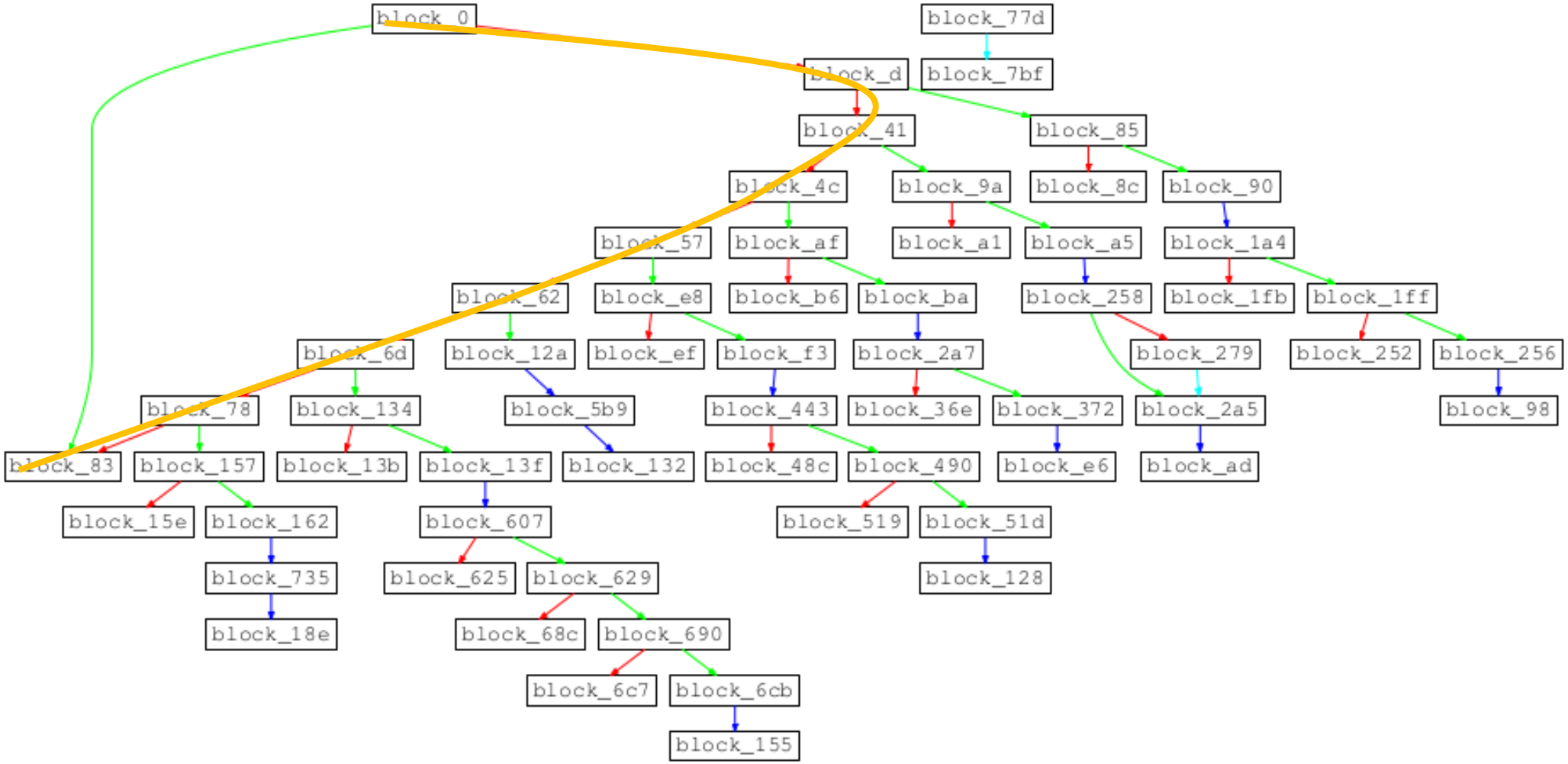
Static analysis



Dynamic analysis (stack evaluation)



Simplify visualization of the smart contract CFG

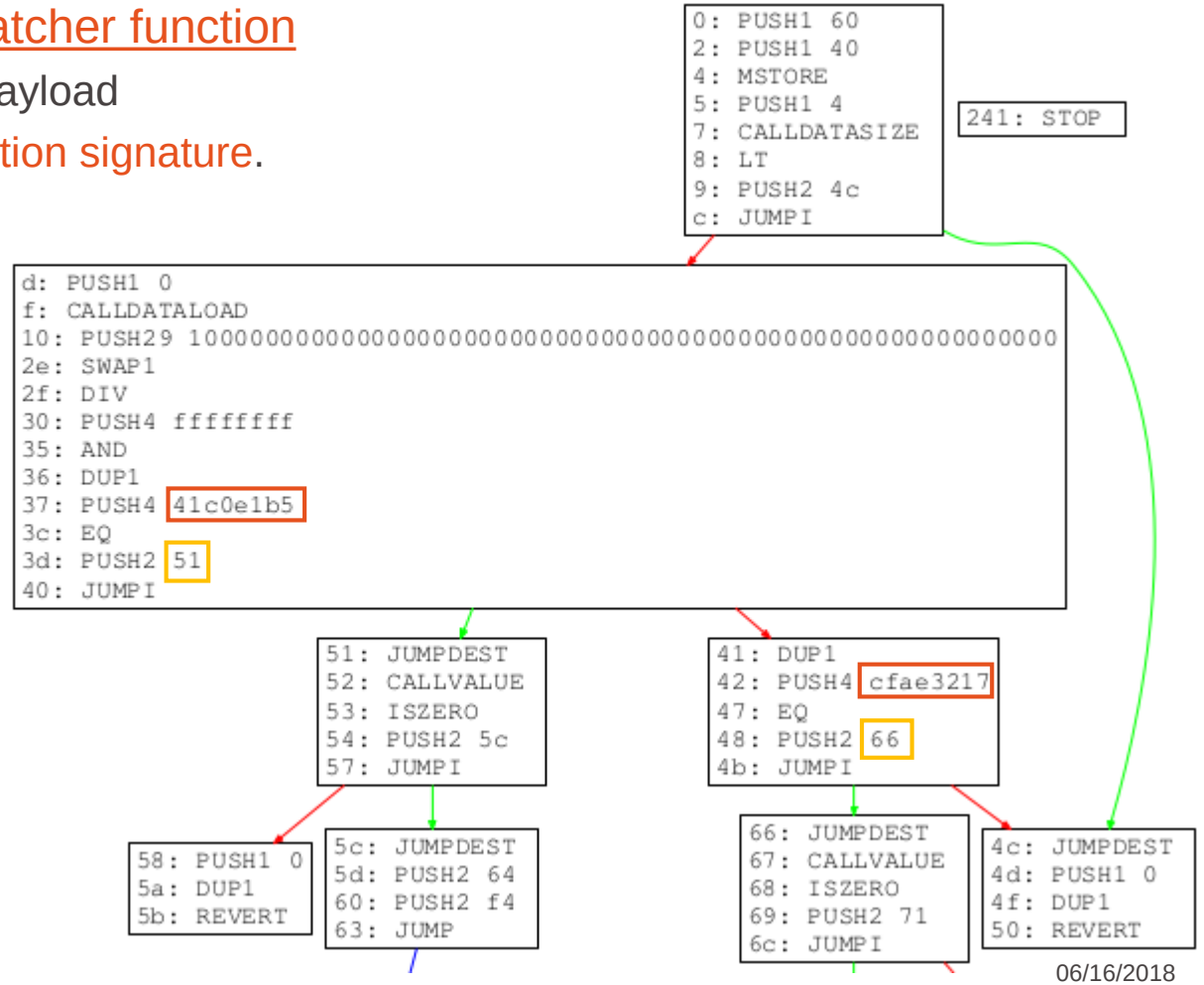


Dispatcher function

- Runtime code entry point is usually a Dispatcher function
  - Switch on the first 4 bytes of the transaction payload
  - execute the associated code of the given function signature.

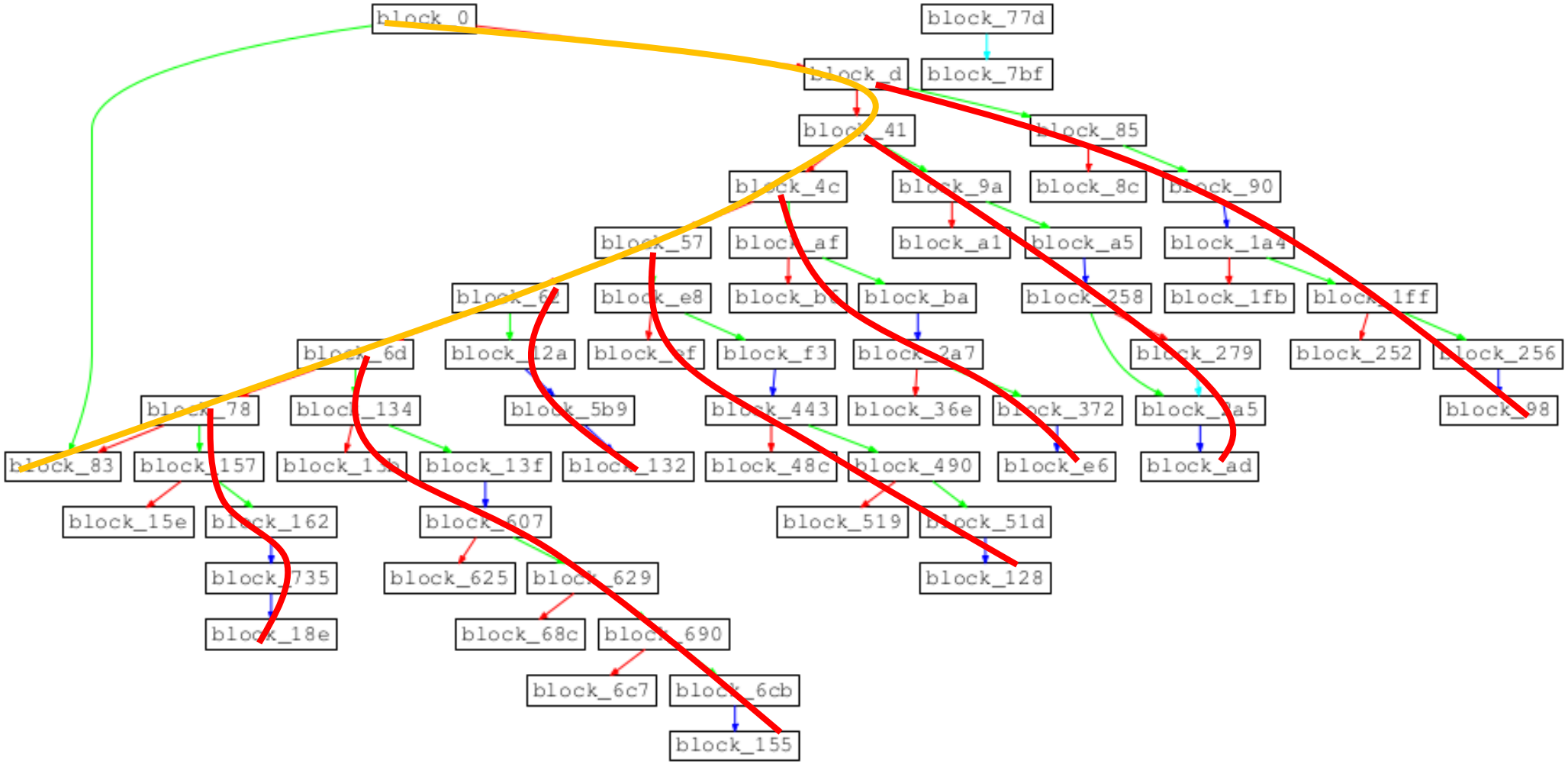
- Two functions signatures here:
  - **41c0e1b5**
  - **cfae3217**

```
41: DUP1
42: PUSH4   FUNC_HASH
47: EQ
48: PUSH2   FUNC_OFFSET
4b: JUMPI
```



Functions identification - Depth First Search

Dispatcher function & 7 callable functions



Transaction payload

- Function signatures/identifiers: First 4 bytes of the sha3 (keccak256) of the function prototype text

```
In [51]: explorer.web3_sha3('0x' + 'attack(address,uint8)'.encode("utf-8").hex())
Out[51]: '0x6ebb6d8020dbdaad3245b82b9ed99905876002f2e6cc8216cd475a481e0b7414'
```

- Composition :
  - Signature (function ID)
  - Argument #1 (address)
  - Argument #2 (uint)

Function prototype string	4-bytes signature
“kill()”	0x41c0e1b5
“greet()”	0xcfae3217
“attack(address,uint)”	**0x6ebb6d80**

- Arguments size: 256-bits words

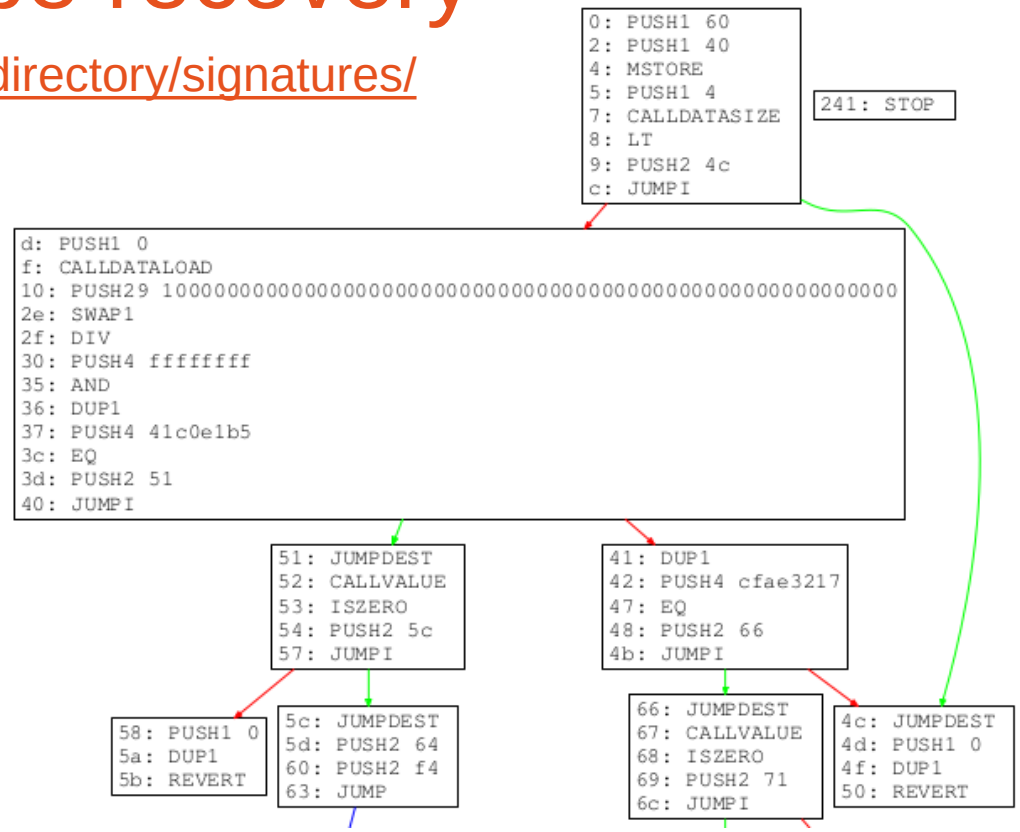
```
'input': '0x6ebb6d80000000000000000000000002983eedfbd560b6d65ffa47de6a0f6d6fee6e136000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000004',
```

Functions name & arguments type recovery

Function signature reverse lookup database - https://www.4byte.directory/signatures/

Search Signatures 0x70a08231 Search

ID	Text Signature	Bytes Signature
2009	greet()	0xcfae3217
1907	kill()	0x41c0e1b5



- Two functions signatures here:
  - **0x41c0e1b5 - "kill()"**
  - **0xcfae3217 - "greet()"**

ID	text signature	bytes signature
31808	distributeTokens(address[],uint256[])	0x4bd09c2a
31807	distributeTokens(address[],uint256)	0x256fa241

EVM Disassembler available

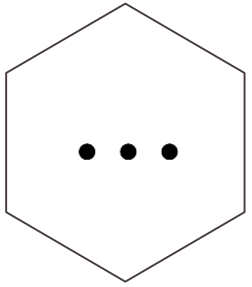
- Etherscan.io
  - ByteCode To Opcode Disassembler

- IDA-EVM
  - IDA Processor Module for the Ethereum Virtual Machine (EVM)

- Quolab
  - Octopus/IDA/BinaryNinja integrate

- Ethersplay
  - Binary ninja plugin

- Capstone
  - Support EVM

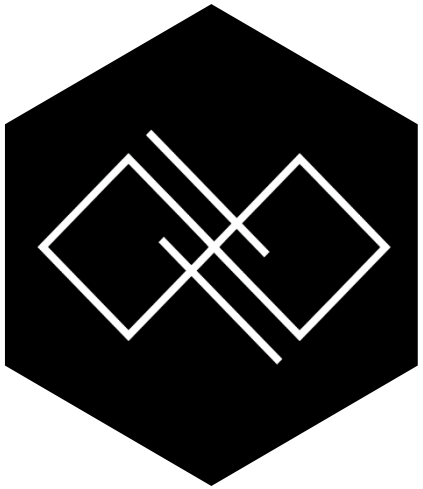


- evmdis
- ethdasm

Decompilation & IR SSA

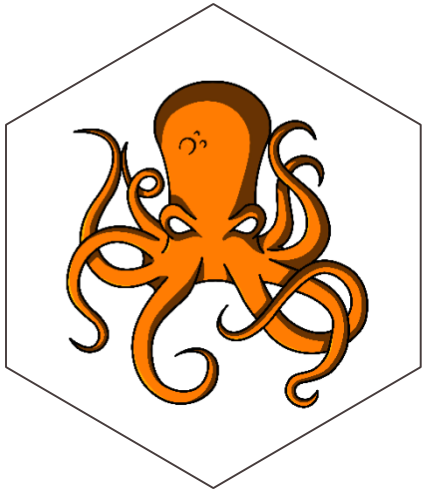
Porosity *by Comae*

- Decompiler
- Github



EthRays *by Ret2*

- Decompiler



Octopus *by QuoScient*

- IR SSA (WIP)
- Quolab or Github

Rattle *by Trail of bits*

- IR SSA

Simplify your analysis with IR

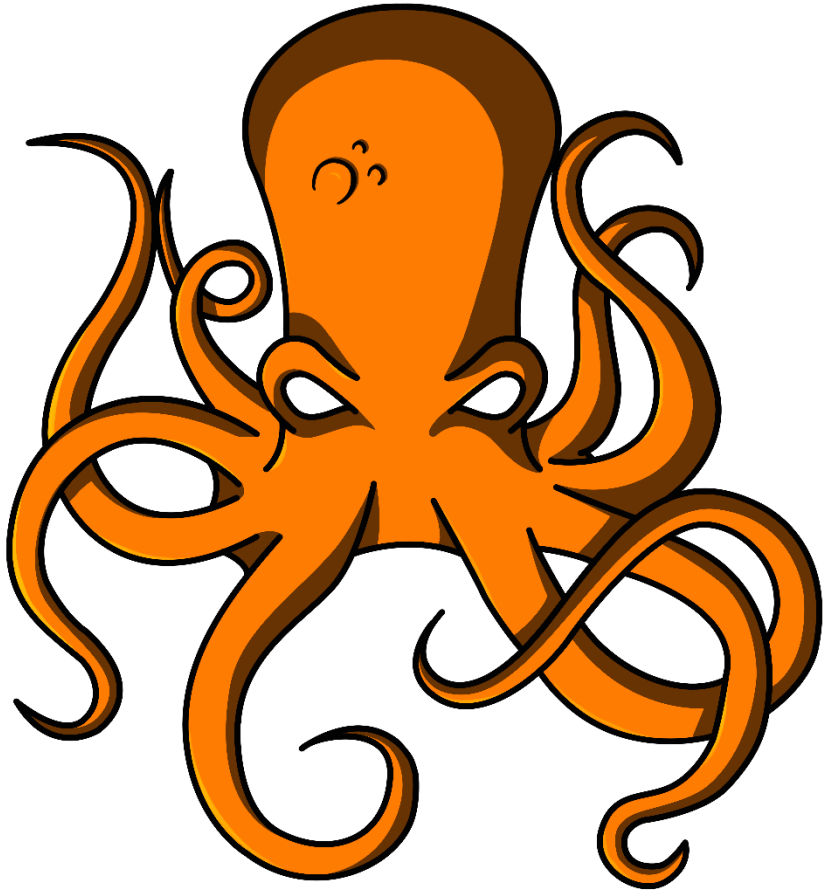
- Static single assignment (SSA)
  - IR (Intermediate representation)
  - each variable is assigned once
  - each variable is defined before being used

Instruction	SSA	Optimize SSA
PUSH1 0x03	%0 = #0x03	
PUSH1 0x05	%1 = #0x05	
ADD	%2 = ADD(%1, %0)	%0 = ADD(#0x05, #0x03)
PUSH1 0x09	%3 = #0x08	
MUL	%4 = EQ(%3, %2)	%1 = EQ(#0x08, %0)

- Ryan will give you more detail on that ;)

Octopus - Ethereum

```
from octopus.api.graph import CFGGraph
from octopus.platforms.ETH.cfg import EthereumCFG

# bytecode contract
bytecode_hex = "60606040526000357c...900480635f"

# create the CFG
cfg = EthereumCFG(bytecode_hex)

# generic visualization api
graph = CFGGraph(cfg)
graph.view()
```

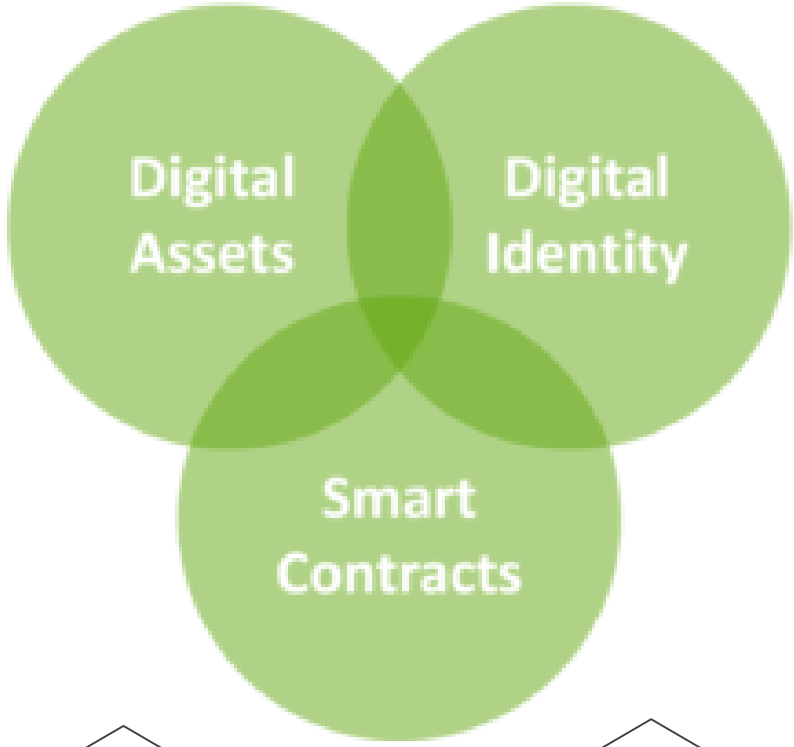


NEO

smart economy

Beginning of Neo

- AntShares (ANS) in 2014.
  - Founded by *Da Hongfei and Erik Zhang*
  - Rebranded as Neo in 2017.
  - Often named as:
    - *“China’s first blockchain platform”*
    - *“Chinese Ethereum”*
- Smart Economy
  - Digital assets
  - Digital identity
  - Smart contracts
- Đapps – Decentralized application
  - http://ndapp.org/



The Neo Project
Official Neo project

City Of Zion
Open-Source Community

NEX

by The NEX Team

NEX combines the NEO blockchain with an off-chain matching engine to enable much faster and more complex trades than existing decentralized exchanges

Website GitHub Whitepaper Twitter LinkedIn Medium

Moonlight

by The Moonlight Team

Moonlight is a distributed workforce and analytical project management platform featuring a global public ledger of contributor work experience and a new match-making algorithm to effectively fulfill project needs.

Website Github Whitepaper Twitter LinkedIn Medium

Red Pulse

by Red Pulse

Red Pulse Tokens (RPX) are NEO tokens issued by Red Pulse, an event-driven research firm covering market events impacting Chinese companies, sectors and the overall economy.

Website GitHub Reddit Contract

AdEx

by adex.network

AdEx is a decentralized ad exchange built on the blockchain and smart contracts. The core feature of AdEx will be the so-called AdEx User Profile - a personalized page that allows every end user to understand and control the ads delivered to them.

Website GitHub Medium Facebook Telegram Twitter Reddit

NeoAuth

by @NeoAuth

NeoAuth enables authentication over the NEO blockchain, allowing you to log in with a NEO address instead of an email and password.

Website Demo GitHub Whitepaper

Zeepin

by Zeepin

The Distributed Creative New Economy.
Zeepin, a decentralized innovation community, is dedicated to promoting highly efficient circulation of innovation assets.
Smart work, Creative life!

Website Telegram Twitter Facebook Reddit Medium GitHub Contract

Qlink

by Allen Li

Qlink, developed by Qlink Foundation in Singapore, adopts the blockchain technology and creates a decentralized mobile network for P2Peer WiFi sharing, mobile data converted content distribution, enterprise telecom services and crowd-sourcing base stations

GitHub Medium Page AMA summary Contract

Neo Smart IoT

by hal0x2328, phetter

Control IoT (Internet of Things) devices via Neo smart contracts (first device is an ESP8266).

Website GitHub Contract

Imusify

by DavidWalters123, geek96, Nikolaj-K, metachris

imusify is a free, blockchain based, incentified and decentralized platform for music related digital content such as audio, video, apps, images, and blogging where anyone can join, contribute and get paid $IMU.

Website GitHub

Chain Line

by notatestuser

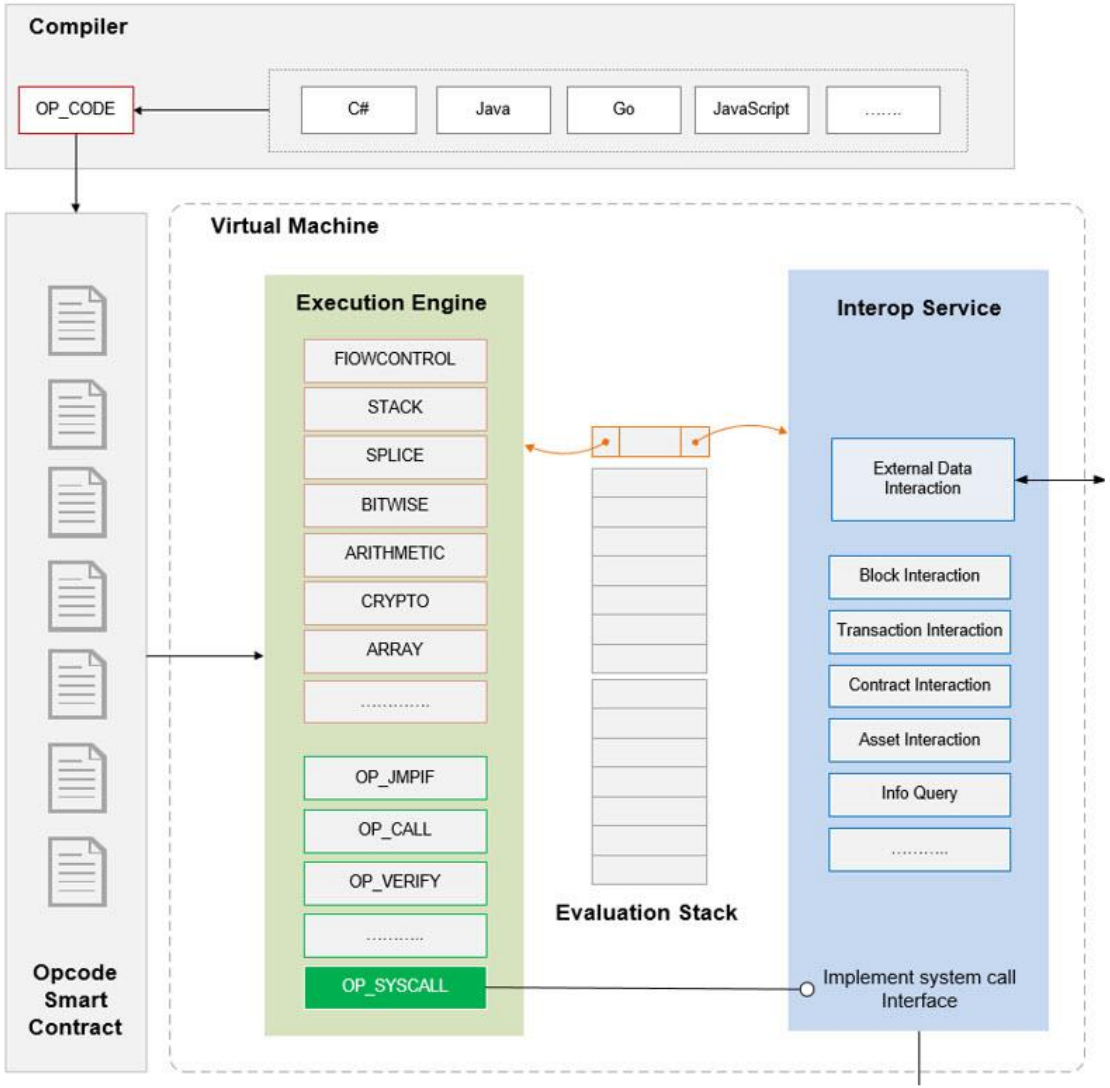
Peer-to-peer courier platform. Couriers transport items to fulfill demands and earn courier fees.

Website GitHub

NeoVM implementations & architecture

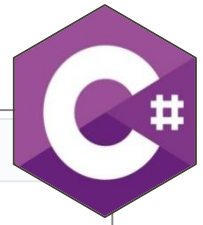
- Neo-vm
  - Official NEO Virtual Machine
  - The Neo project

- Neo-python
  - Python Node and SDK
  - City Of Zion



Compilation to AVM bytecode

neon



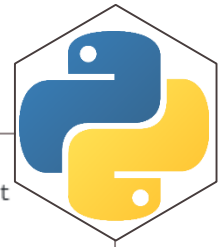
13 lines (11 sloc) | 271 Bytes

```
1  using Neo.SmartContract.Framework.Services.Neo;
2
3  namespace Neo.SmartContract
4  {
5      public class HelloWorld : Framework.SmartContract
6      {
7          public static void Main()
8          {
9              Storage.Put(Storage.CurrentContext, "Hello", "World");
10         }
11     }
12 }
```

neoj

```
import org.neo.smartcontract.framework.SmartContract;
import org.neo.smartcontract.framework.services.neo.Storage;

public class HelloWorld extends SmartContract {

    public static byte[] Main(String[] args){
        Storage.put(Storage.currentContext(), "Greeting to the World", "Hello World!")
        return Storage.get(Storage.currentContext(),"Greeting to the World");
    }

}
```

neo-boa



```
from boa.interop.Neo.Runtime import Log, Notify
from boa.interop.Neo.Storage import Get, Put, GetContext

def Main():
    context = GetContext()

    # This is the storage key we use in this example
    item_key = 'test-storage-key'

    # Try to get a value for this key from storage
    item_value = Get(context, item_key)
    msg = ["Value read from storage:", item_value]
    Notify(msg)

    if len(item_value) == 0:
        Notify("Storage key not yet set. Setting to 1")
        item_value = 1

    else:
        Notify("Storage key already set. Incrementing by 1")
        item_value += 1

    # Store the new value
    Put(context, item_key, item_value)
    msg = ["New value written into storage:", item_value]
    Notify(msg)

    return item_value
```

Contract d3cce84d0800172d09c88ccad61130611bd047a4

Hash	d3cce84d0800172d09c88ccad61130611bd047a4
Name	Lock
Version	2.0
Parameters	["Integer","PublicKey","Signature"]
Return Type	Boolean
Uses Storage	No
Author	Erik Zhang
Email	erik@neo.org
Description	Lock 2.0

Publish Transaction

▶ Invocation 4e84015258880ced0387f34842b1d96f605b9cc78b308e1f0d876933c2c9134b 8/20/2017, 12:36:57 PM

← Ae2d6qj91YL3LVUMkza7WQsaTYjzjHm4z1 -722.1202854 GAS

Ae2d6qj91YL3LVUMkza7WQsaTYjzjHm4z1 232.1202854 GAS (Spent) →

Published Lock

Network Fee: 0 GAS System Fee: 490 GAS

Script

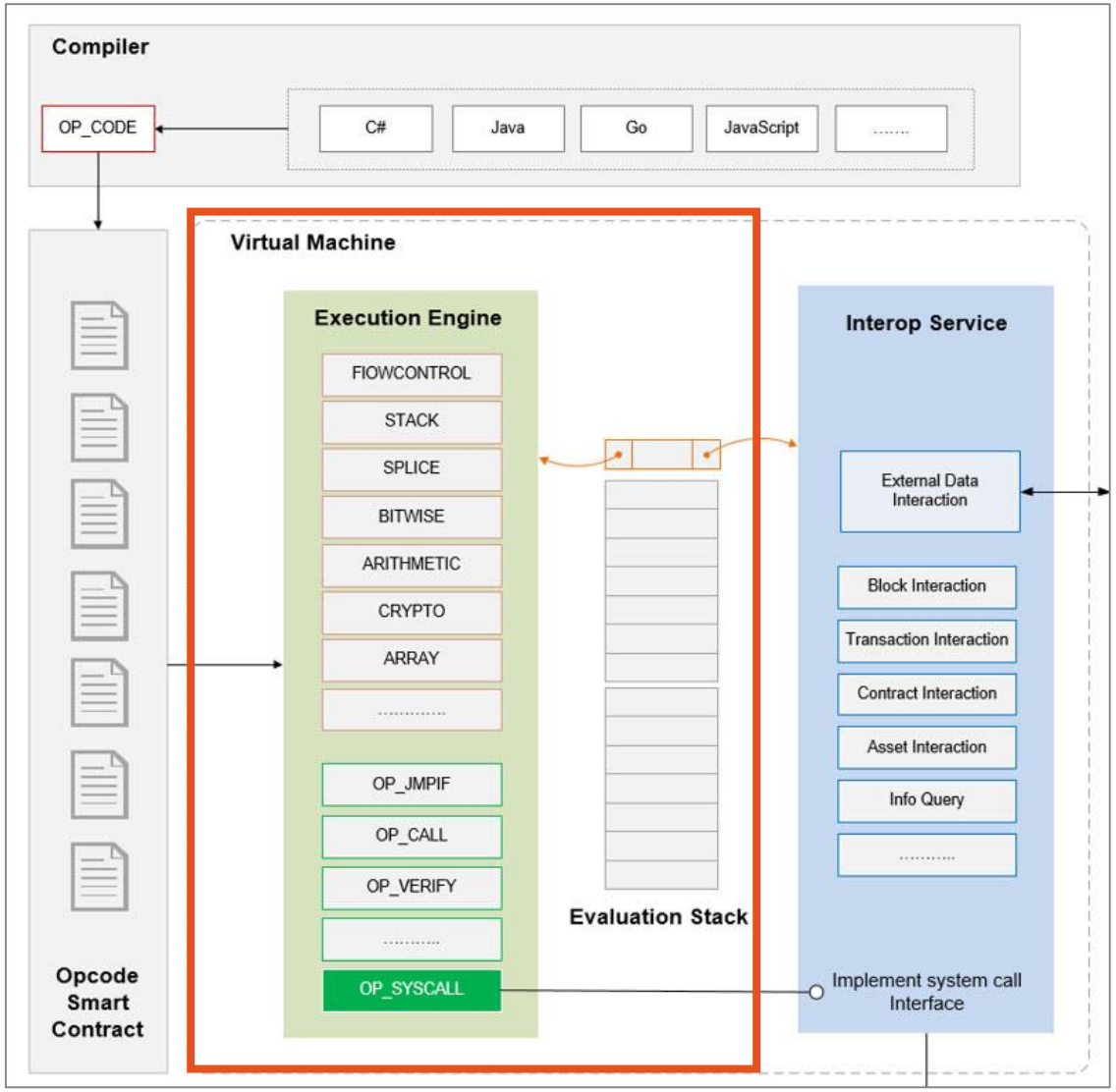
```
56c56b6c766b00527ac46c766b51527ac46c766b52527ac4616168184e656f2e426c6f636b636861696e2e476574486569676874616818 4e656f2e426c6f636b6368616
96e2e476574486561646572 6c766b53527ac46c766b00c36c766b53c36168174e656f2e4865616465722e47657454696d657374616d70a06c766b54527ac46c766b54c3
640e00006c766b55527ac4621a006c766b51c36c766b52c3617cac6c766b55527ac46203006c766b55c3616c7566
```

Neo bytecode (.avm)

Neo Virtual Machine #1

Execution engine & Stacks

Architecture		
Stack machine		
Turing complete		
Instruction sets		~180 Opcodes
4 type of Memory		
Evaluation Stack (Compute stack)	volatile	array (list [])
Alt Stack (Standby stack)	volatile	array (list [])
Invocation Stack (Call stack)	volatile	array (list [])
Storage	persistent	key-value database (dictionary {})



AVM Instructions set

Really similar to Bitcoin Script with Turing completeness

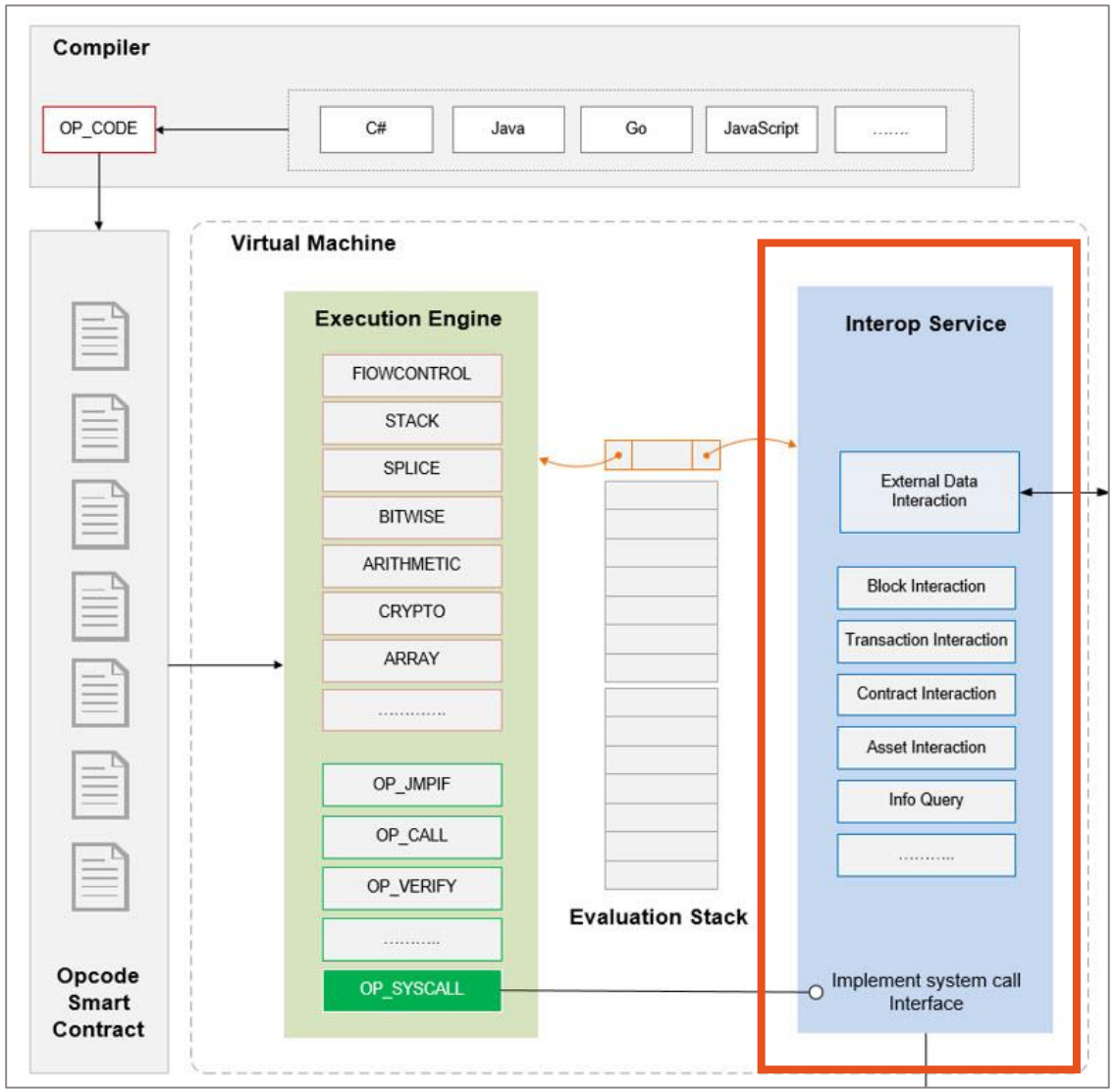
Opcodes value	Family	Examples
0x00 – 0x60	Constants	**PUSH0**, PUSHBYTES1 – PUSHBYTES75, PUSH1 – PUS16
0x61 – 0x69	**Flow control**	NOP, **JMP, JMPIF, JMPIFNOT, CALL, RET, SYSCALL**
0x6A - 0x7D	Stack	DUPFROMALTSTACK, **TOALTSTACK**, **DROP**, **DUP**, ROLL, **SWAP**
0x7E – 0x82	Splice	**CAT**, **SUBSTR**, LEFT, RIGHT, SIZE
0x83 – 0x8A	Bitwise logic	**INVERT**, AND, OR, **XOR, EQUAL**
0x8B – 0xA5	Arithmetic	**INC, DEC, SUB, MUL**, SHL, SHR, BOOLAND, LT, GT, **MAX, MIN**
0xA6 – 0xAE	Crypto	**SHA256, HASH160, CHECKSIG**
0xC0 – 0xCB	Array	**ARRAYSIZE**, UNPACK, SETITEM, **APPEND**
0xF0 – 0xF1	Exceptions	**THROW, THROWIFNOT**

- Neo-VM source code: https://github.com/neo-project/neo-vm/blob/master/src/neo-vm/OpCode.cs

Neo Virtual Machine #2

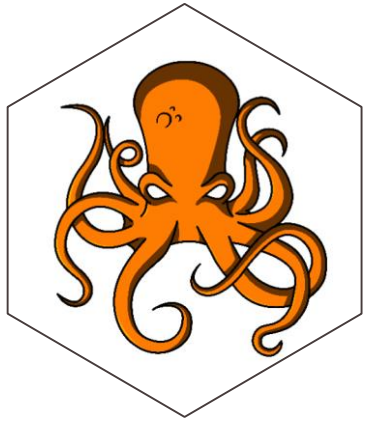
Interactive Service Layer

- **`SYSCALL`** opcode with API name as parameter
  - used to communicates with the outside
- NEO namespace: ~60 API methods
  - Query the blockchain
  - `Neo.Blockchain.GetHeight`
  - `Neo.Transaction.GetHash`
  - Manipulate the persistent store (Storage)
  - `Neo.Storage.Get, Neo.Storage.Put`
- System namespace: ~4 API methods
  - access of the execution environment
  - `System.ExecutionEngine.GetEntryScriptHash`



Disassembling

- neo-debugger-tools (C#)
  - CLI-Disassembler



- Octopus (Python)
  - NeoDisassembler Class
  - Quolab or github

```
0: PUSH15
1: NEWARRAY
2: TOALTSTACK
3: FROMALTSTACK
4: DUP
5: TOALTSTACK
6: PUSH0
7: PUSH2
8: ROLL
9: SETITEM
a: FROMALTSTACK
b: DUP
c: TOALTSTACK
d: PUSH1
e: PUSH2
f: ROLL
10: SETITEM
11: NOP
12: PUSHBYTES11 0x4f7065726174696f6e3a20
1e: FROMALTSTACK
1f: DUP
20: TOALTSTACK
21: PUSH0
22: PICKITEM
23: CAT
24: NOP
25: SYSCALL 0x4e656f2e52756e74696d652e4c6f67
36: NOP
37: FROMALTSTACK
38: DUP
39: TOALTSTACK
```

Decomposition into basic blocks & functions

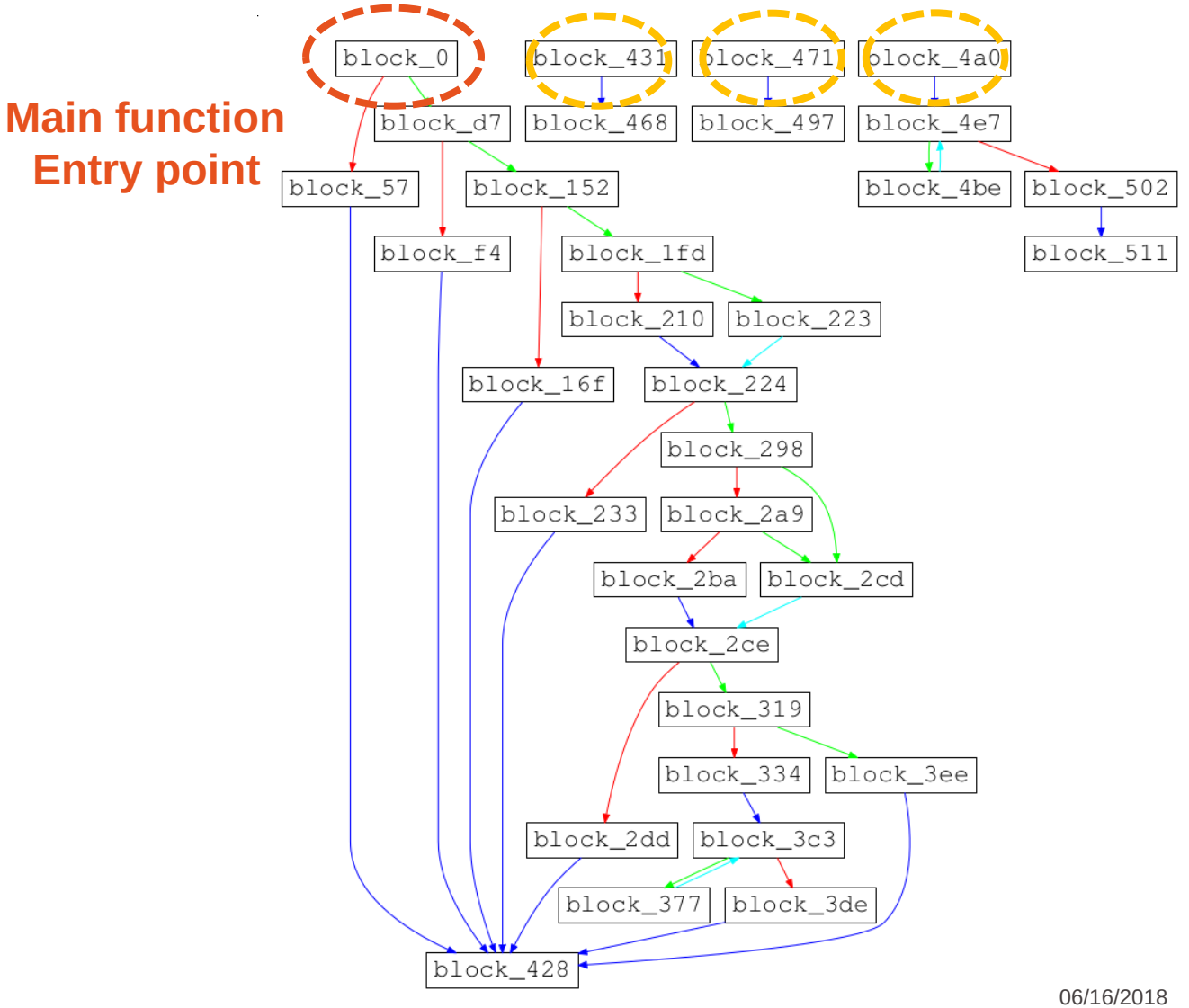
Opcode	Simplify description	Role
JMP	Unconditional jump	Last instruction
JMPIF, JMPIFNOT	Conditional jump	Last instruction
RET	Halt execution	Last instruction
THROW, THROWIFNOT	Halt execution	Last instruction

```
2dd: NOP
2de: PUSH2
2df: NEWARRAY
2e0: DUP
2e1: PUSH0
2e2: FROMALTSTACK
2e3: DUP
2e4: TOALTSTACK
2e5: PUSH0
2e6: PICKITEM
2e7: SETITEM
2e8: DUP
2e9: PUSH1
2ea: FROMALTSTACK
2eb: DUP
2ec: TOALTSTACK
2ed: PUSH1
2ee: PICKITEM
2ef: PUSH0
2f0: PICKITEM
2f1: SETITEM
2f2: NOP
2f3: SYSCALL 4e656f2e52756e74696d652e4e6f74696679
307: NOP
308: FROMALTSTACK
309: DUP
30a: TOALTSTACK
30b: PUSH1
30c: PICKITEM
30d: PUSH0
30e: PICKITEM
30f: FROMALTSTACK
310: DUP
311: TOALTSTACK
312: PUSH4
313: PUSH2
314: ROLL
315: SETITEM
316: JMP 1201
```

```
319: FROMALTSTACK
31a: DUP
31b: TOALTSTACK
31c: PUSH0
31d: PICKITEM
31e: PUSHBYTES5 4172726179
324: EQUAL
325: FROMALTSTACK
326: DUP
327: TOALTSTACK
328: PUSH11
329: PUSH2
32a: ROLL
32b: SETITEM
32c: FROMALTSTACK
32d: DUP
32e: TOALTSTACK
32f: PUSH11
330: PICKITEM
331: JMPIFNOT bd00
```

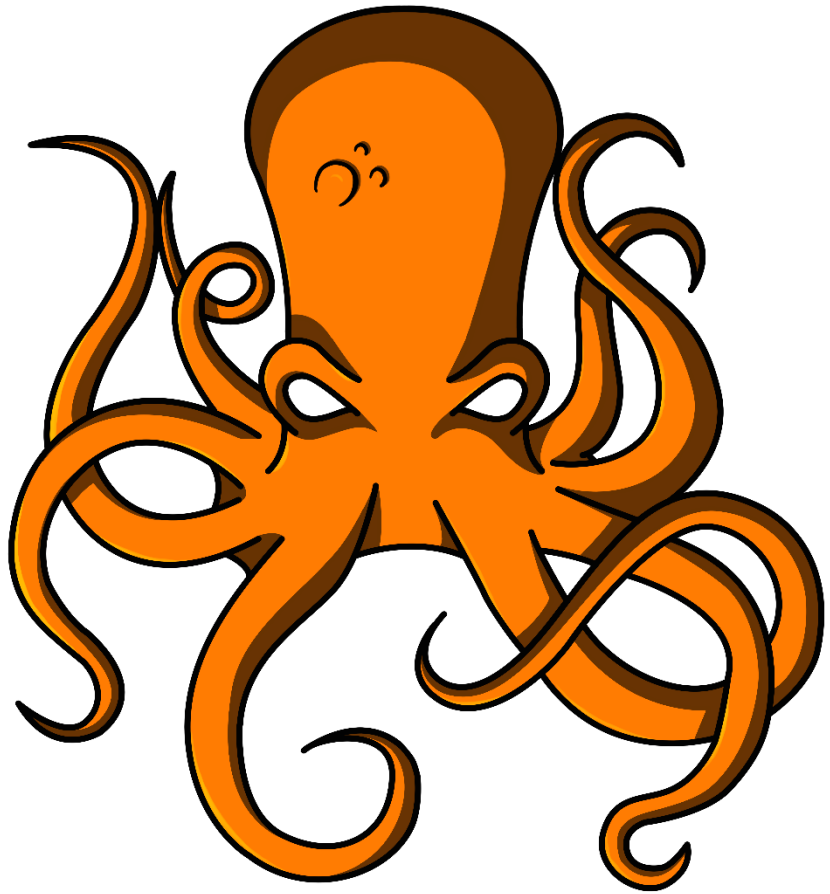
CFG using static analysis

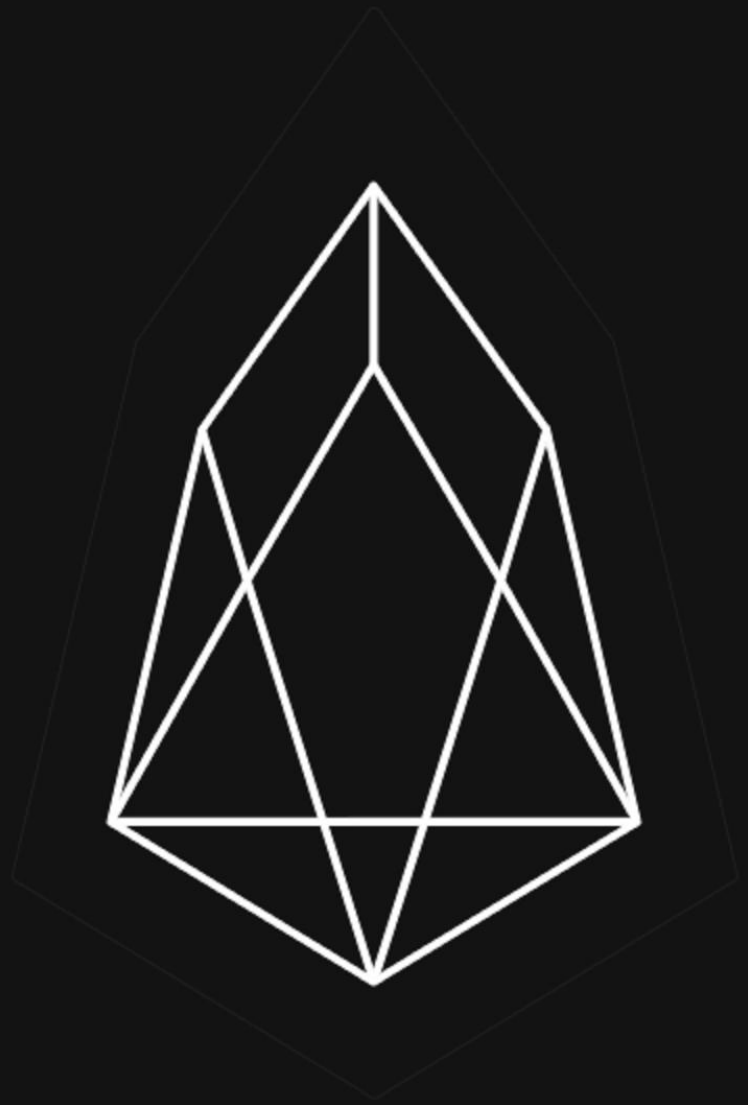
- AVM functions
  - First function is the Main function
  - Other function are following
- Function code
  - Linear list of instructions.
  - End with RET instruction



Octopus - Neo

```
from octopus.api.graph import CFGGraph
from octopus.platforms.NEO.cfg import NeoCFG

# lock contract
bytecode_hex = "56c56b6c7...66b55c3616c7566"

# create the CFG
cfg = NeoCFG(bytecode_hex,
             static_analysis=True)

# visualization
graph = CFGGraph(cfg)
graph.view()
```

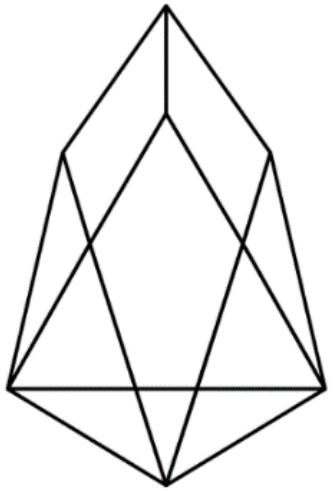




EOS

What's EOS?

- *"The Most Powerful Infrastructure for Decentralized Applications"*
  - Open source smart contract platform
  - Mainnet will be launch ***soon***
- Created by *Daniel Larimer*
  - *White paper in 2017*
- Smart contract
  - Written in C++
  - Compiled to WebAssembly



EOS

SCALABLE

Supports thousands of Commercial Scale DApps
Inter-blockchain Communication
Separates Authentication from Execution

FLEXIBLE

Freeze and Fix Broken Applications
Generalized Role-based Permissions
Web Assembly

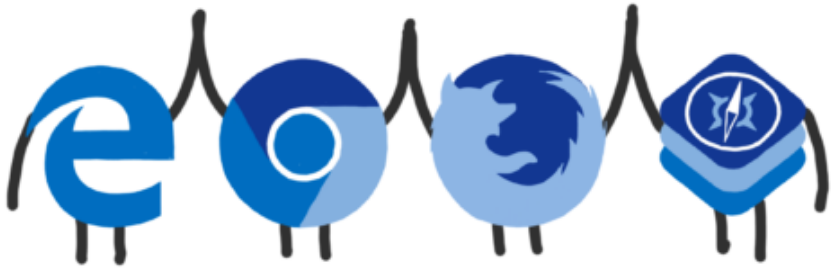
USABLE

Web Toolkit for Interface Development
Self Describing Interfaces
Declarative Permission Scheme

WebAssembly (Wasm)

- *"WebAssembly (abbreviated Wasm) is a low-level assembly-like language with a compact binary format that runs with near-native performance and provides languages with low-level memory models such as C++ and Rust with a compilation target so that they can run on the web."*

- WebAssembly goals:
  - Be fast, efficient, and portable (near-native speed)
  - Easily readable and debuggable (Wast)
  - Keep secure (safe, sandboxed execution environment)
  - Don't break the web



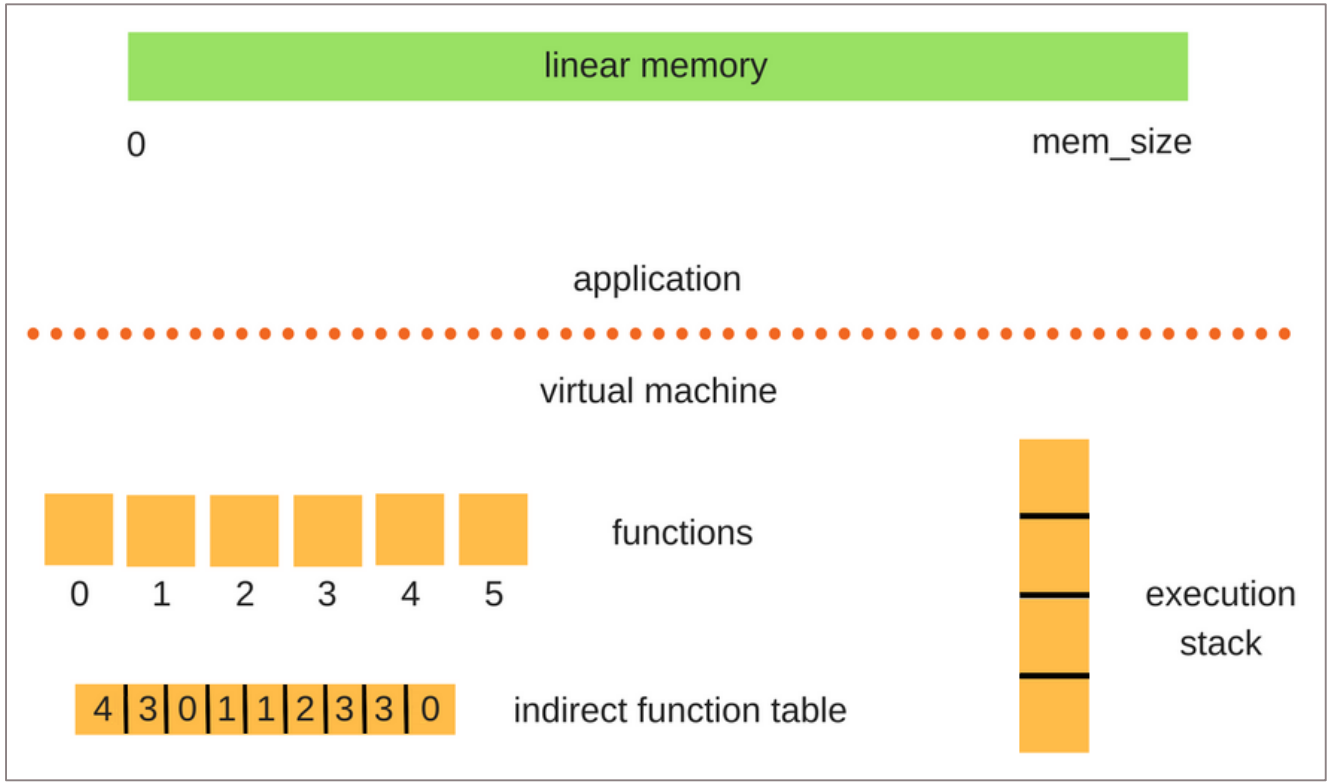
- Supported by LLVM
  - LLVM Backend

Virtual machine specification

- Stack machine
- Compiled code is immutable at runtime
- Abnormal behavior triggers traps

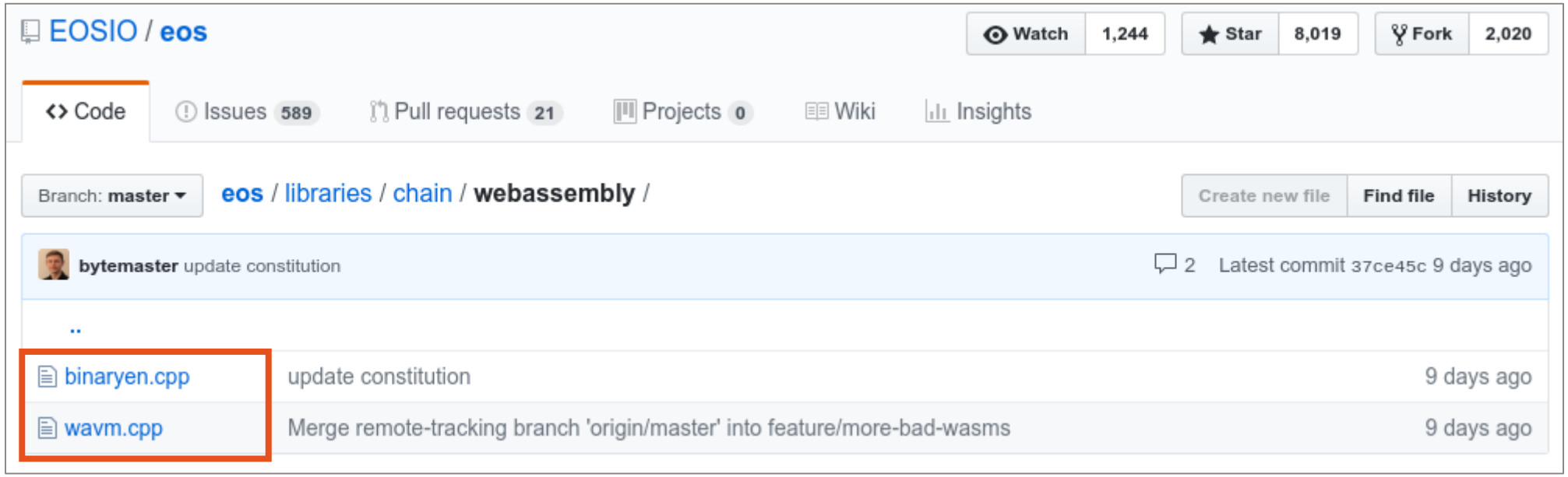
- Linear memory
  - Bounds-checked array
- Functions
  - Calls specify by index using static index table
  - Type signature check at runtime

- Control-flow integrity (CFI)
  - Direct/Indirect call targets are valid functions
  - Branches point to a valid destination



WebAssembly third-party libraries in EOS

- Binaryen
  - compiler infrastructure and toolchain library for WebAssembly
- WAVM
  - standalone VM for WebAssembly



EOS Node RCE Vulnerability *by Qihoo 360*

Date: 05/29/18

- EOS WASM Contract Function Table Array Out of Bounds
  - http://blogs.360.cn/blog/eos-node-remote-code-execution-vulnerability/

- assert() only works in Debug build but doesn't work in a Release build.

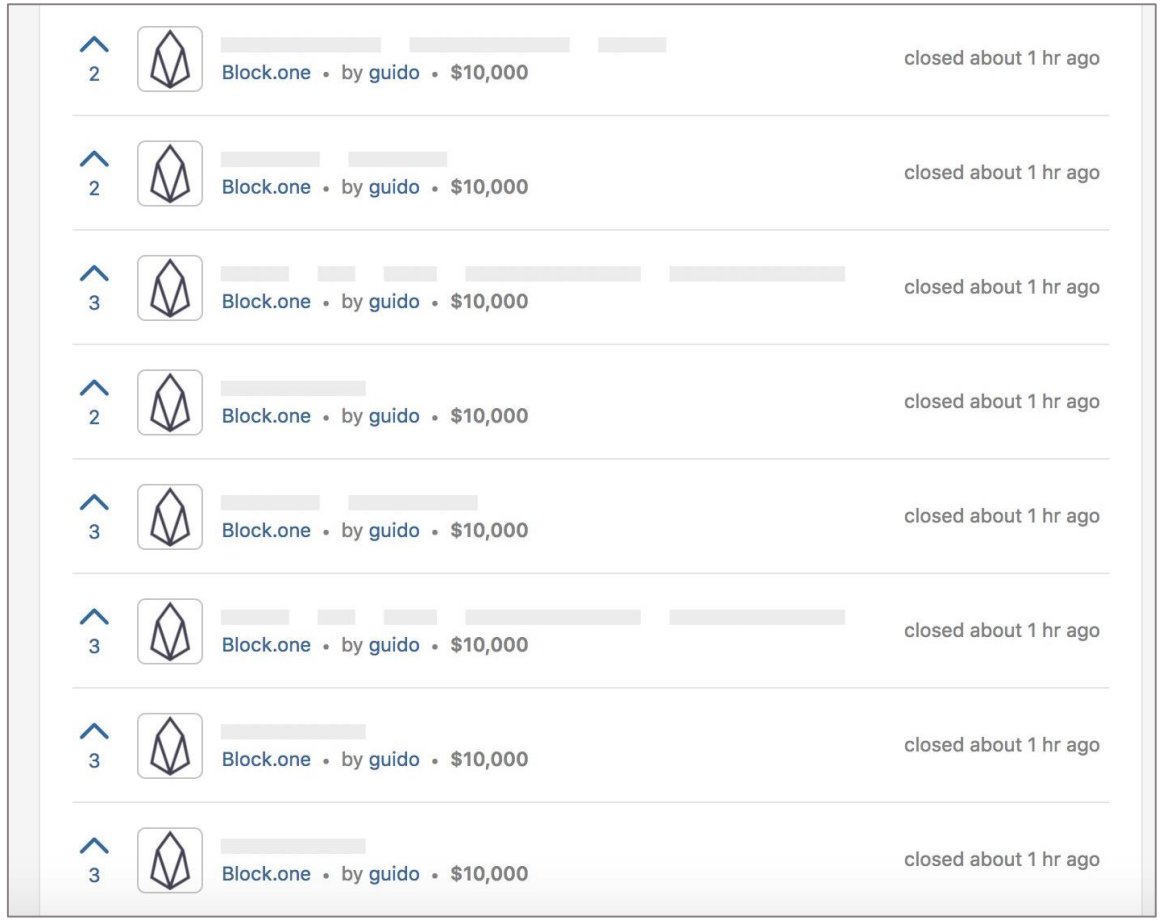
2 libraries/chain/webassembly/binaryen.cpp — View

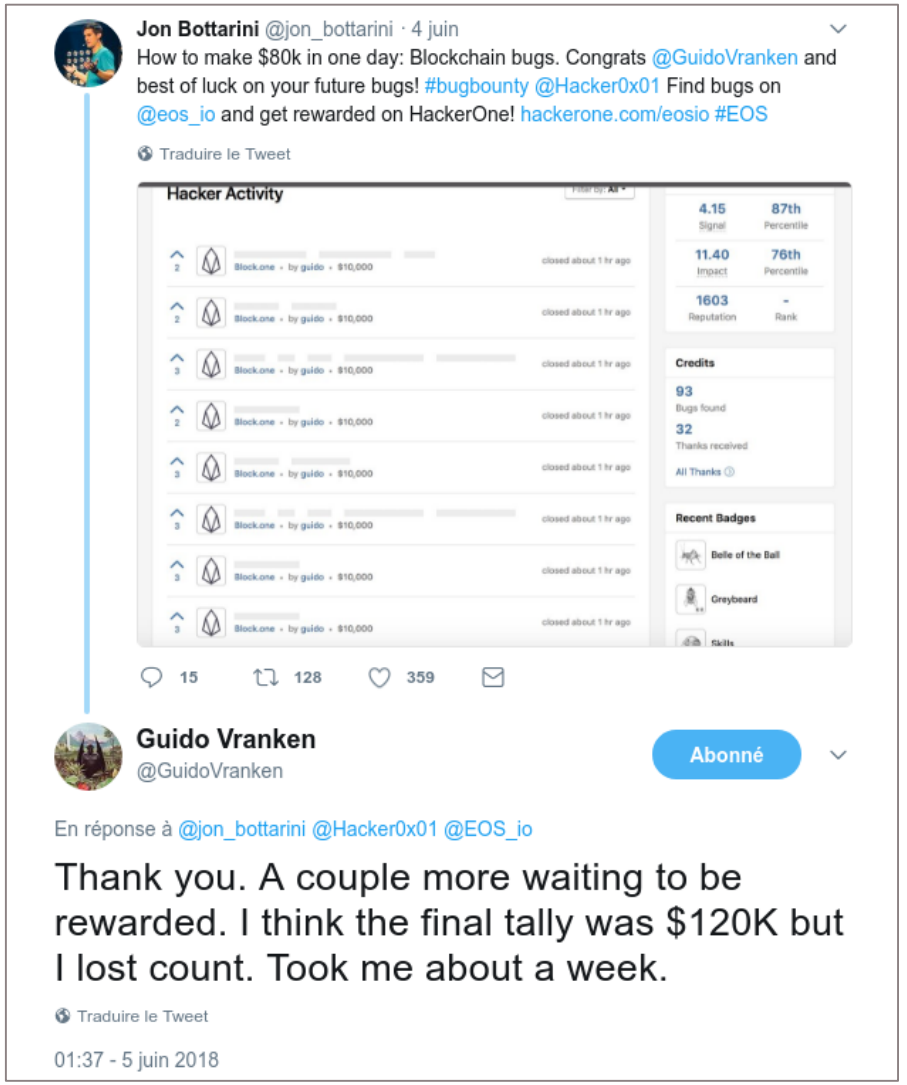
```
@@ -73,7 +73,7 @@ std::unique_ptr<wasm_instantiated_module_interface> binaryen_runtime::instantiat
73 73        table.resize(module->table.initial);
74 74        for (auto& segment : module->table.segments) {
75 75           Address offset = ConstantExpressionRunner<TrivialGlobalManager>(globals).visit(segment.offset).value.geti32();
76    -        assert(offset + segment.data.size() <= module->table.initial);
   76 +        FC_ASSERT(offset + segment.data.size() <= module->table.initial);
77 77           for (size_t i = 0; i != segment.data.size(); ++i) {
78 78              table[offset + i] = segment.data[i];
79 79           }
```

Out Of Bound Write

More than 12 bugs in one week *by guido*

Date: 06/04/18





Source code / Wasm / Wast

- Source code
  - Eosio standard library
- Wasm module
  - Magic number '\0asm'
  - Binary format
  - Compact
  - Easy to verify
  - 11 different sections
- Wast
  - Text format
  - Body of a function (Text column) is a linear list of low-level instructions.

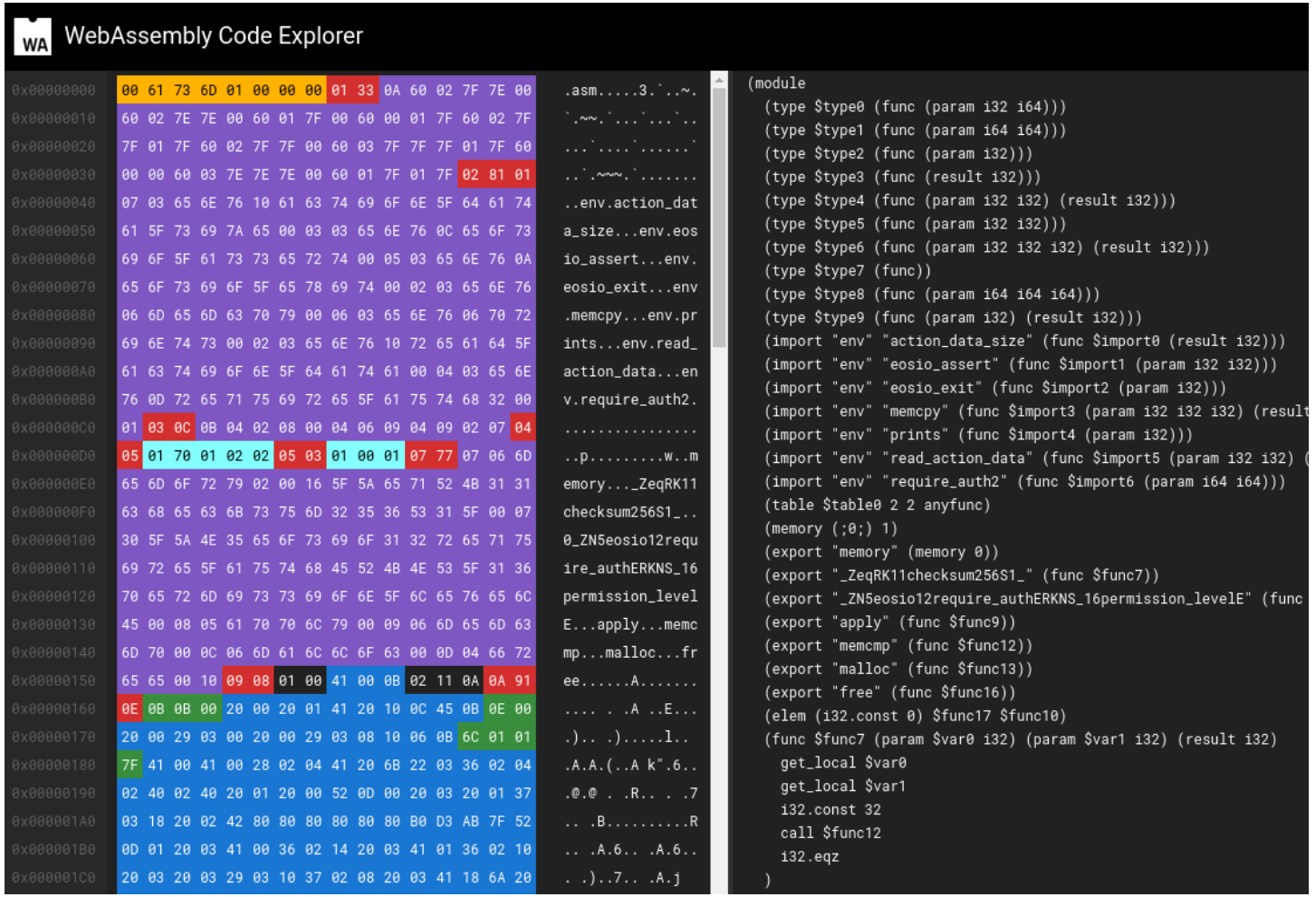
C++	Binary	Text

`int factorial(int n) {`
`  if (n == 0)`
`  return 1;`
`  else`
`  return n * factorial(n-1);`
`}`
	`20 00`
	`42 00`
	`51`
	`04 7e`
	`42 01`
	`05`
	`20 00`
	`20 00`
	`42 01`
	`7d`
	`10 00`
	`7e`
	`0b`
		`get_local 0`
		`i64.const 0`
		`i64.eq`
		`if i64`
		`  i64.const 1`
		`else`
		`  get_local 0`
		`  get_local 0`
		`  i64.const 1`
		`  i64.sub`
		`  call 0`
		`  i64.mul`
		`end`

Decode & Disassembly

Convert .wasm to .wast

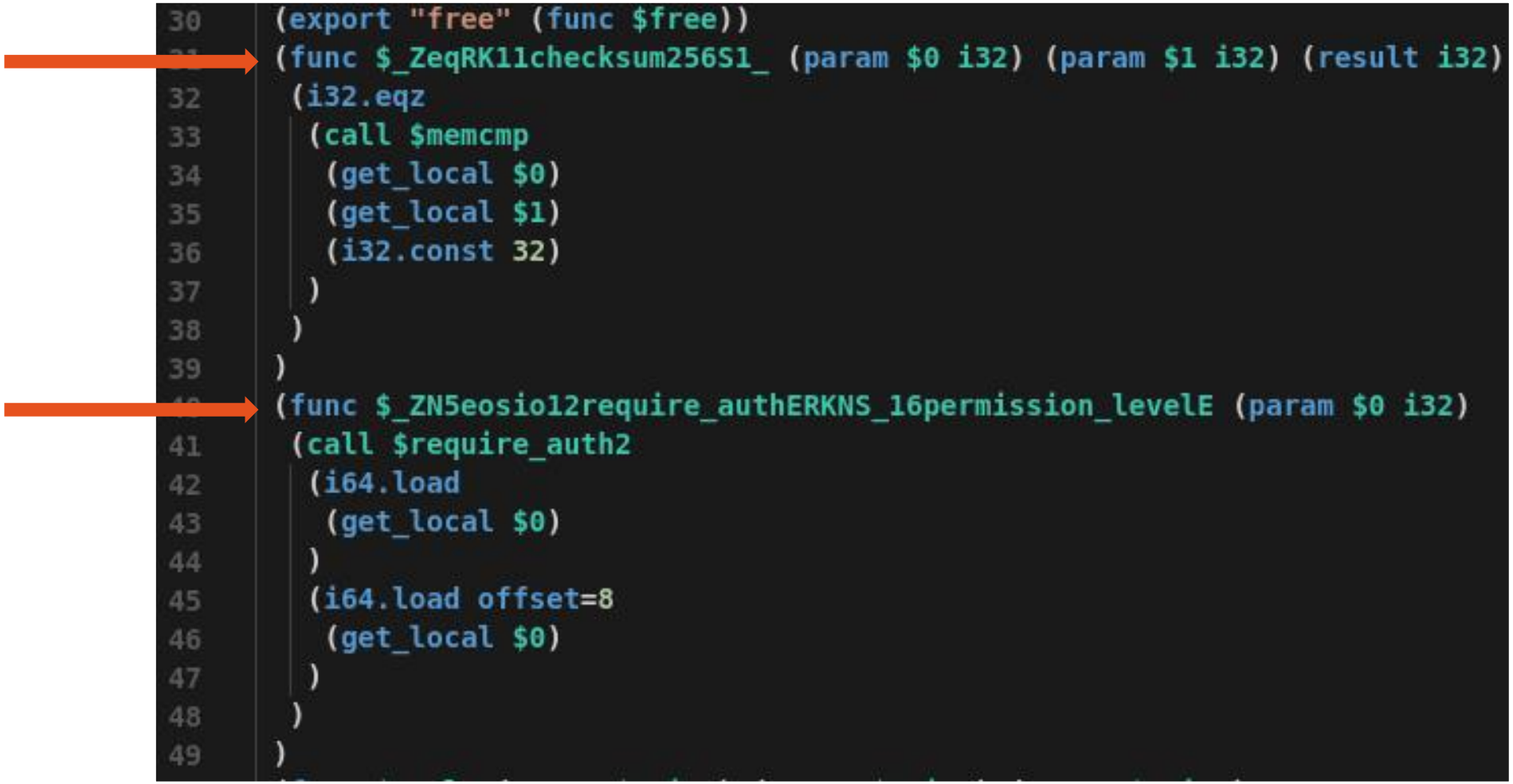
- Wasm decoder & disassembler library
  - Python module
  - Octopus
    - EosDisassembler class
- WABT: The WebAssembly Binary Toolkit
  - wasm-objdump
  - Wasmcodeexplorer
    - Simple WebAssembly binary file explorer
    - Available online



EOS module text representation - sections

```
(module
 (type $FUNCSIG$vij (func (param i32 i64)))
 (type $FUNCSIG$vjj (func (param i64 i64)))
 (type $FUNCSIG$vi (func (param i32)))
 (type $FUNCSIG$i (func (result i32)))
 (type $FUNCSIG$iii (func (param i32 i32) (result i32)))
 (type $FUNCSIG$vii (func (param i32 i32)))
 (type $FUNCSIG$iiii (func (param i32 i32 i32) (result i32)))
 (type $FUNCSIG$v (func))
 (import "env" "action_data_size" (func $action_data_size (result i32)))
 (import "env" "eosio_assert" (func $eosio_assert (param i32 i32)))
 (import "env" "eosio_exit" (func $eosio_exit (param i32)))
 (import "env" "memcpy" (func $memcpy (param i32 i32 i32) (result i32)))
 (import "env" "prints" (func $prints (param i32)))
 (import "env" "read_action_data" (func $read_action_data (param i32 i32) (result i32)))
 (import "env" "require_auth2" (func $require_auth2 (param i64 i64)))
 (table 2 2 anyfunc)
 (elem (i32.const 0) $__wasm_nullptr $_ZN13ping_contract4pingEy)
 (memory $0 1)
 (data (i32.const 4) "`a\00\00")
 (data (i32.const 16) "read\00")
 (data (i32.const 32) "Pong\00")
 (data (i32.const 8448) "malloc_from_freed was designed to only be called after _heap was completely allocated\00")
 (export "memory" (memory $0))
 (export "_ZeqRK11checksum256S1_" (func $_ZeqRK11checksum256S1_))
 (export "_ZN5eosio12require_authERKNS_16permission_levelE" (func $_ZN5eosio12require_authERKNS_16permission_levelE))
 (export "apply" (func $apply))
 (export "memcmp" (func $memcmp))
 (export "malloc" (func $malloc))
```

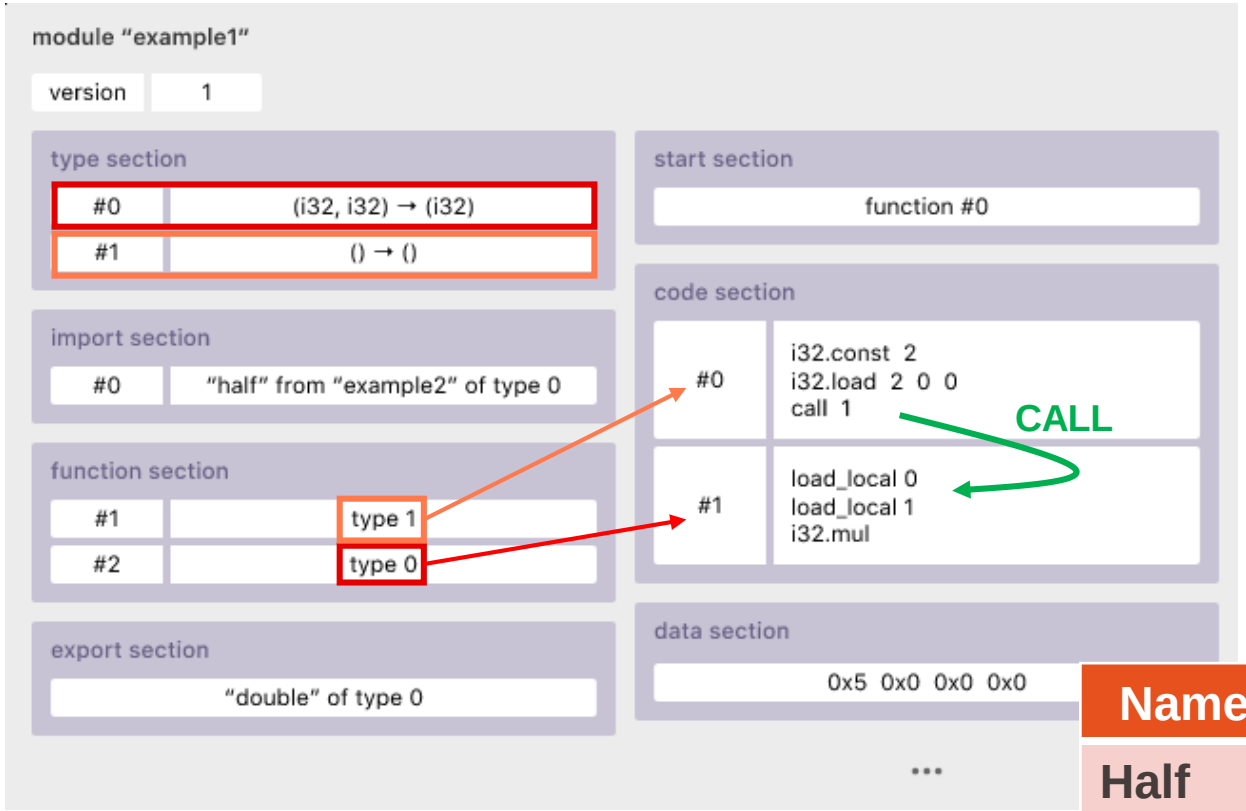
EOS module text representation – function codes



```
30  (export "free" (func $free))
    (func $_ZeqRK11checksum256S1_ (param $0 i32) (param $1 i32) (result i32)
32   (i32.eqz
33    (call $memcmp
34     (get_local $0)
35     (get_local $1)
36     (i32.const 32)
37    )
38   )
39  )
    (func $_ZN5eosio12require_authERKNS_16permission_levelE (param $0 i32)
41   (call $require_auth2
42    (i64.load
43     (get_local $0)
44    )
45    (i64.load offset=8
46     (get_local $0)
47    )
48   )
49  )
```

Function analysis

Name, parameters & return types, call flow



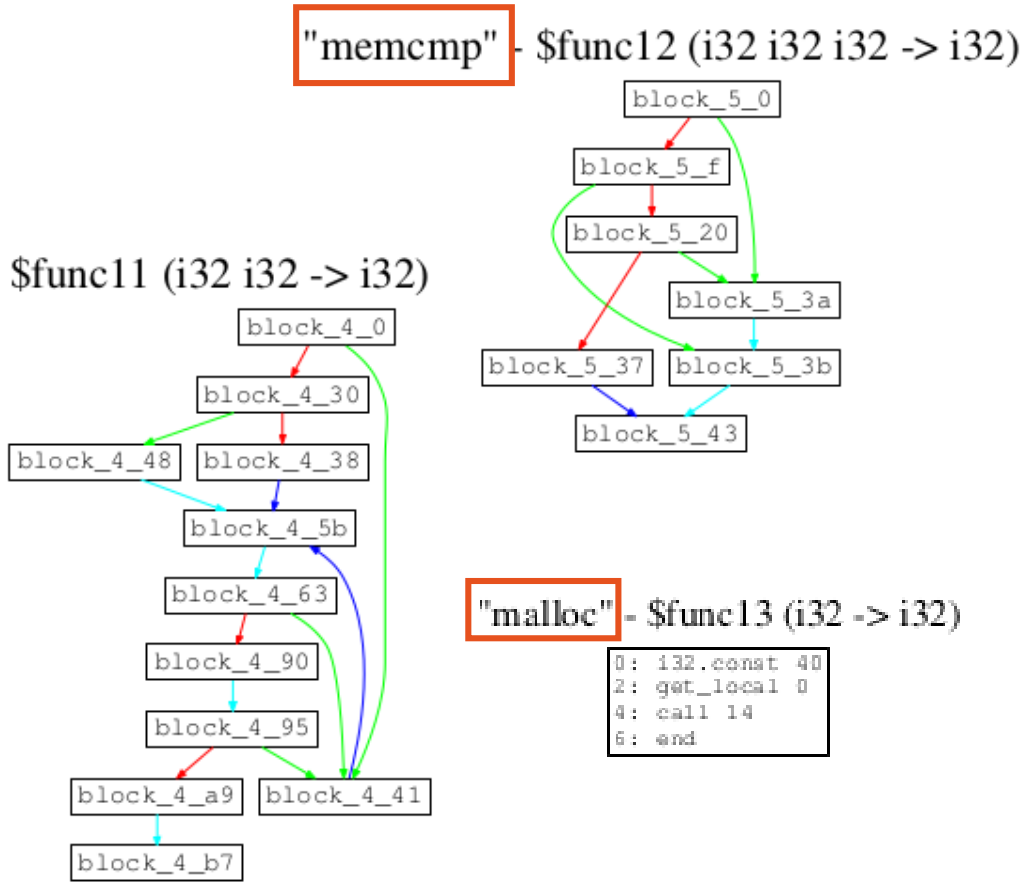
- Function body
  - Linear list of instructions
- Parameters & return types
  - `type_sec[func_sec[func_id]]`
  - Only one return value ATM
- Call flow
  - `call <func_id>`

Name	Arg & return types	Type
Half	**(i32, i32) → i32**	Imported function
???	**() → ()**	Start function (Internal)
double	**(i32, i32) → i32**	Exported function

CFG reconstruction

Control flow operators (described here)

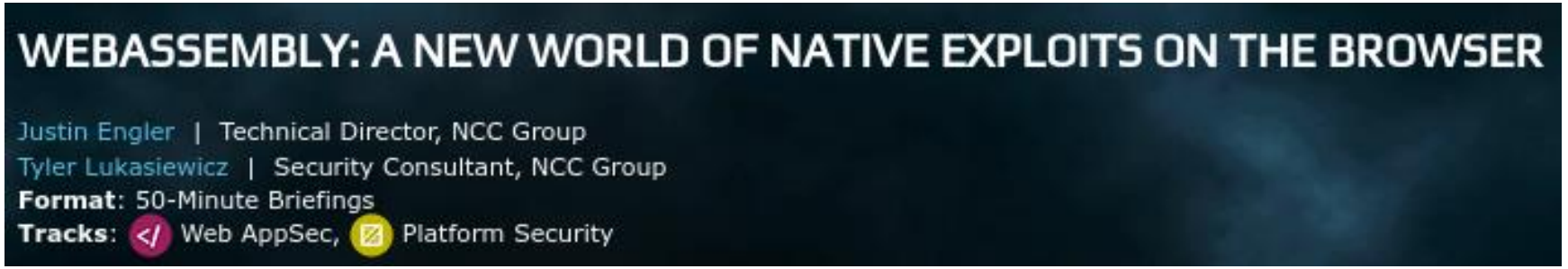
Name	Opcode	Immediates	Description
`unreachable`	`0x00`		trap immediately
`nop`	`0x01`		no operation
`block`	`0x02`	sig : `block_type`	begin a sequence of expressions, yielding 0 or 1 values
`loop`	`0x03`	sig : `block_type`	begin a block which can also form control flow loops
`if`	`0x04`	sig : `block_type`	begin if expression
`else`	`0x05`		begin else expression of if
`end`	`0x0b`		end a block, loop, or if
`br`	`0x0c`	relative_depth : `varuint32`	break that targets an outer nested block
`br_if`	`0x0d`	relative_depth : `varuint32`	conditional break that targets an outer nested block
`br_table`	`0x0e`	see below	branch table control flow construct
`return`	`0x0f`		return zero or one value from this function



Security of WebAssembly

- *Natalie Silvanovich* – Google Project Zero

- *Justin Engler & Tyler Lukasiewicz* – NCC Group

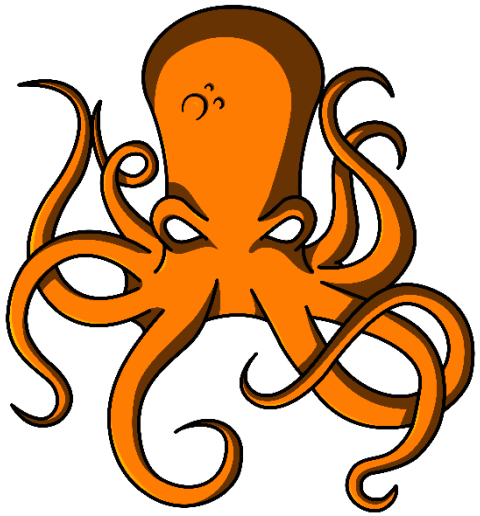


- WebAssembly security documentation: https://webassembly.org/docs/security/

Conclusion

- Analysis of Blockchain Smart contracts is a good way:
  - to understand how blockchain work in general
  - to discover new languages & ASM
  - to start vulnerability research on blockchain nodes/VMs
- Future of Octopus:
  - IDA plugin
  - SSA for all platforms
  - ETH Symbolic Execution
  - Decompilation
- Just choose one to play with…

Thanks & Question



RECON MONTREAL 2018

- Patrick Ventuzelo / @Pat_Ventuzelo / patrick.ventuzelo@quoscient.io
- Octopus - https://github.com/quoscient/octopus

CONTACT

QuoScient

Radilostrasse 43
60489 Frankfurt
Germany
+49 69 33 99 79 38
curious@quoscient.io
www.quoscient.io

QuoScient

Digital Active Defense