

Analysis of Ethereum Smart Contracts and Opcodes

2nd Distributed Ledger Technology Workshop (DLT 2019)



Authors: *Stefano Bistarelli (Unipg)*
Gianmarco Mazzante
Matteo Micheletti
Leonardo Mostarda
Francesco Tiezzi

Speaker: Gianmarco Mazzante

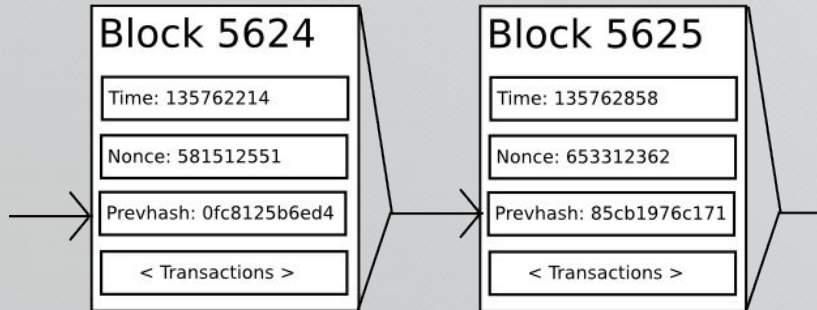
How can we spend less on smart contract fees?



How can we spend less on smart contract fees?

Chapters

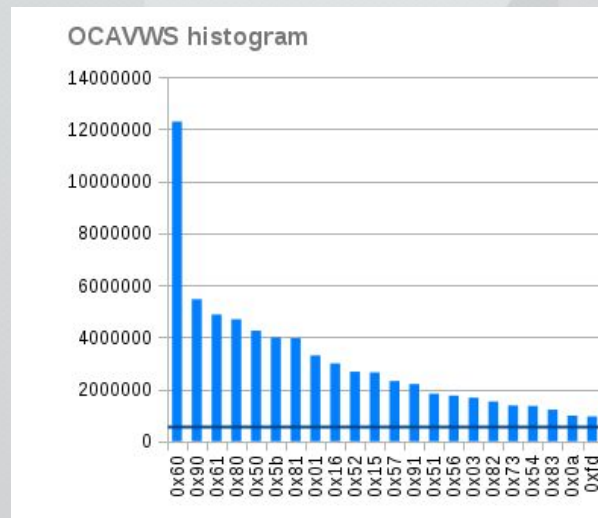
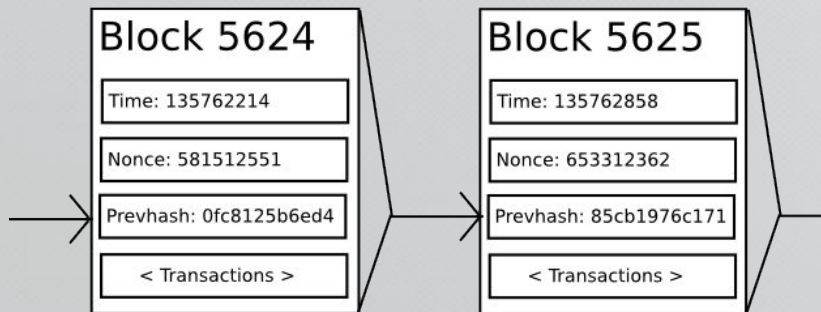
- 1) Get an understanding about *blockchain* and *smart contracts*



How can we spend less on smart contract fees?

Chapters

- 1) Get an understanding about *blockchain* and *smart contracts*
- 2) Perform a statistical analysis on *opcodes*



Blockchain and Ethereum

Smart contracts are programs deployed on the blockchain and executed in the network



Blockchain and Ethereum

Smart contracts are programs deployed on the blockchain and executed in the network



Ethereum introduced *turing-complete* contracts

Opcodes and gas

The contracts are translated into **opcodes** and executed in the **Ethereum Virtual Machine**

```
PUSH1 0x09  
PUSH1 0xFA  
MUL
```



Opcodes and gas

The contracts are translated into **opcodes** and executed in the **Ethereum Virtual Machine**

PUSH1	0x09	3 <i>gwei</i>
PUSH1	0xFA	3 <i>gwei</i>
MUL		5 <i>gwei</i>

Each opcode has a different execution cost in terms of **gas**

Opcodes and gas

The contracts are traded on the **Ethereum Virtual Machine**

Each opcode has a

Wei	1000000000000000000
Kwei, Ada, Femtoether	1000000000000000
Mwei, Babbage, Picoether	1000000000000
Gwei, Shannon, Nanoether, Nano	1000000000
Szabo, Microether, Micro	1000000
Finney, Milliether, Milli	1000
Ether	1
Kether, Grand, Einstein	0.001
Mether	0.000001
Gether	0.000000001
Tether	0.0000000000001
USD (at 245.827\$ p/ ether)	245.827
EUR (at 211.334€ p/ ether)	211.334

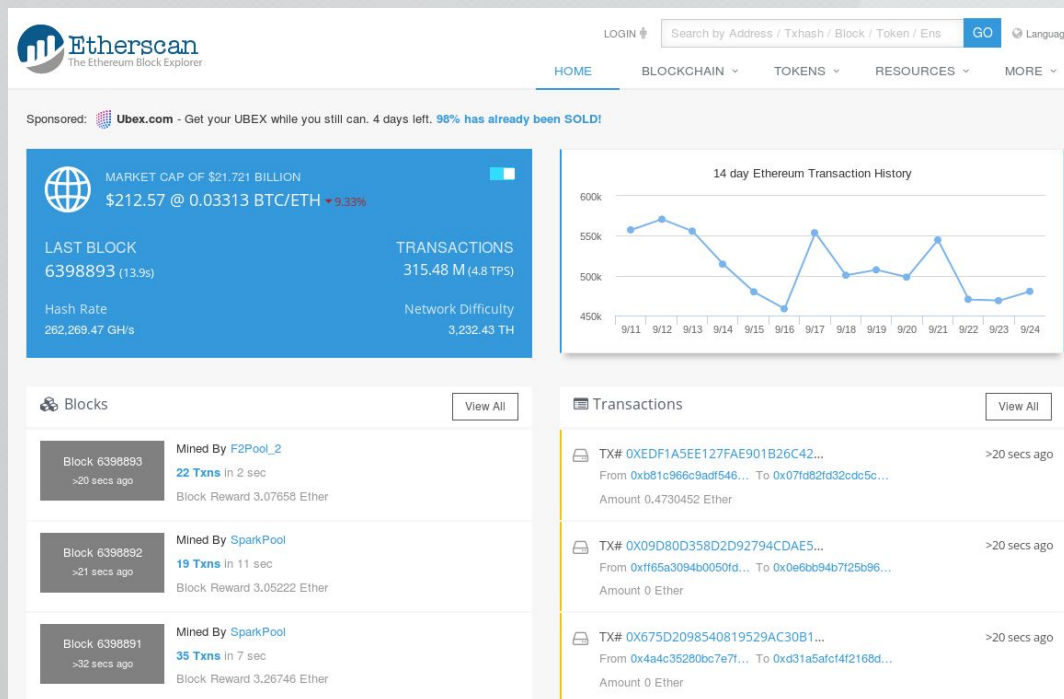
executed in the

terms of **gas**



Analysis

We used **Etherscan.io** to retrieve all the contracts surces



Analysis

We used **Etherscan.io** to retrieve all the contracts surces

Contract 0x8E4290B07bd2c7e02C2CD71e60660e2488E25e4a

Home / Accounts / Address

Sponsored:

Contract Overview

Balance:

Ether Value:

Transactions:

Token Tracker:

Transactions

Are you the

Switch To O

Contract Creation Code

Switch Back To Bytecodes View

```
PUSH1 0x80
PUSH1 0x40
MSTORE
PUSH1 0x04
CALLDATASIZE
LT
PUSH2 0x01cc
JUMPI
PUSH4 0xffffffff
PUSH29 0x0100000000000000000000000000000000000000000000000000000000000000
```

0x6060604
63095ea7b
79cc67901
80fd5b610
019050610
341561015857600080fd5b61018d00480803573fff169060200190919080359060200190919050506105d1565b604051
8082151515815260200191505060405180910390f35b34156101b257600080fd5b6101ba61065e565b6040518082815260200191505060405180910390f35b341561
01db57600080fd5b61022f600480803573fff1690602001909190803573fff
ff16906020019091908035906020019091905050610664565b6040518082151515815260200191505060405180910390f35b341561025457600080fd5b61025c6107
91565b604051808260ff1660ff16815260200191505060405180910390f35b341561028357600080fd5b61029960048080359060200190919050506107a4565b604051
8082151515815260200191505060405180910390f35b34156102be57600080fd5b6102ea600480803573fff16906020
0190919050506108a8565b6040518082815260200191505060405180910390f35b341561030b57600080fd5b610340600480803573fffffffffffffffffffffffffffffffff

0bf5780
3578063
a576000
2602081
390f35b

Analysis

On Etherscan is possible to **verify contracts** compiled with **So/C**

Etherscan
The Ethereum Block Explorer

LOGIN Language

HOME BLOCKCHAIN ▼ TOKENS ▼ RESOURCES ▼ MORE ▼

Verify Contract Code (version 2.0) ***Updated** [Home](#) / [Verify Contract](#)

Note: Check out the new [Verify Source Code API](#)

Contract Source Code

Verify and Publish your Solidity Source Code

Step 1 : Enter your Contract Source Code below.
Step 2 : If the Bytecode generated matches the existing Creation Address Bytecode, the contract is then Verified.
Step 3 : Contract Source Code is published online and publicly verifiable by anyone.

NOTES

- To verify Contracts that accept Constructor arguments, please enter the **ABI-encoded** Arguments in the last box below.
- For debugging purposes if it compiles correctly at [Browser Solidity](#), it should also compile correctly here.
- Contracts that use "imports" will need to have the code concatenated into one file as we do not support "imports" in separate files. You can try using the [Blockcat solidity-flattener](#) or [SolidityFlattener](#)
- We do not support contract verification for contracts created by another contract.
- There is a timeout of up to 45 seconds for each contract compiled. If your contract takes longer than this we will not be able to verify it.

Contract Address	Contract Name:	Compiler	Optimization
<input type="text" value="Contract Address"/> <input type="button" value="📄"/>	<input type="text" value="Contract Name"/> <input type="button" value="📄"/>	<input type="text" value="[Please select]"/>	<input type="text" value="Yes"/>

Enter the Solidity Contract Code below

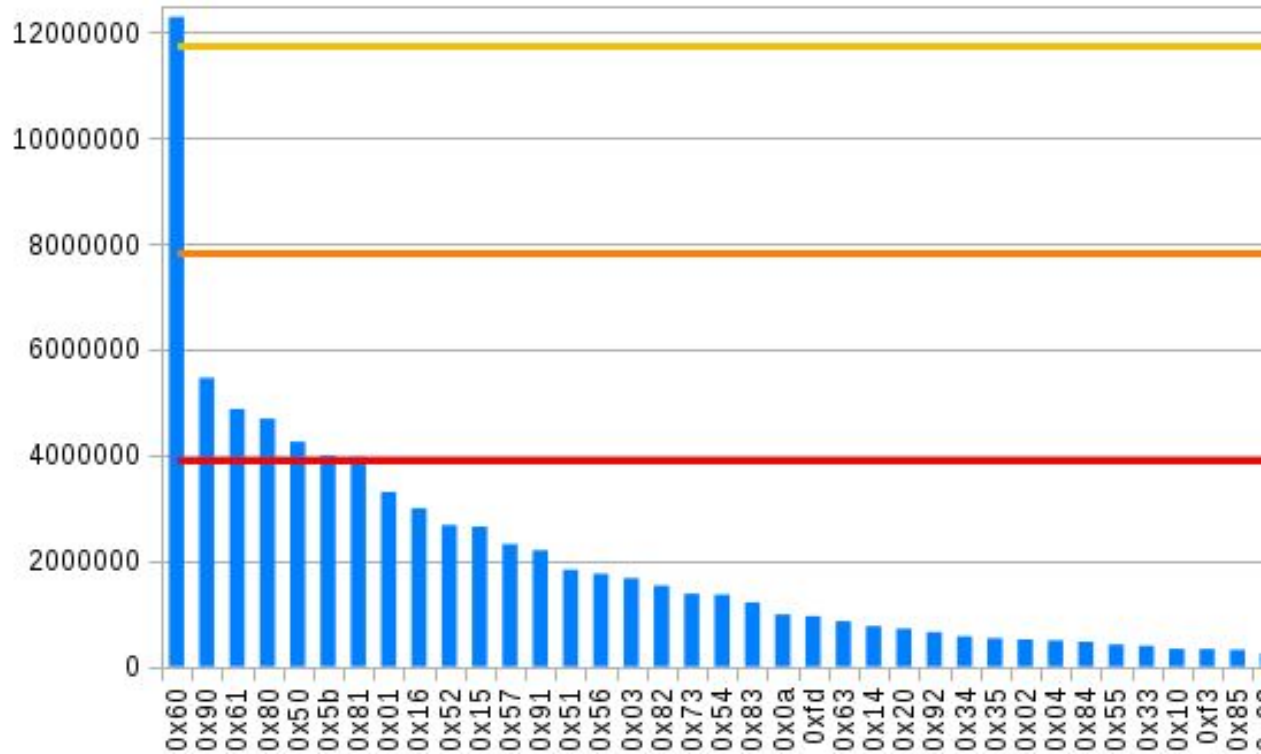
Analysis

Most used PUSH opcodes on verified contracts

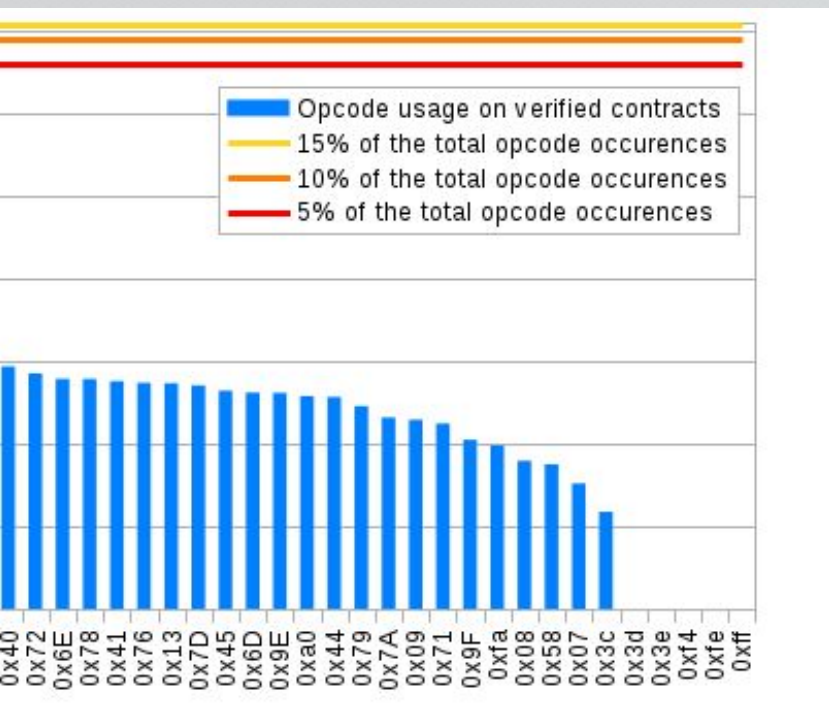
1	PUSH1
3	PUSH2
18	PUSH20
23	PUSH4
41	PUSH32

Most used opcodes on verified contracts

1	PUSH1
2	SWAP1
3	PUSH2
4	DUP1
5	POP
6	JUMPDEST
7	DUP2
8	ADD
9	AND
10	MSTORE

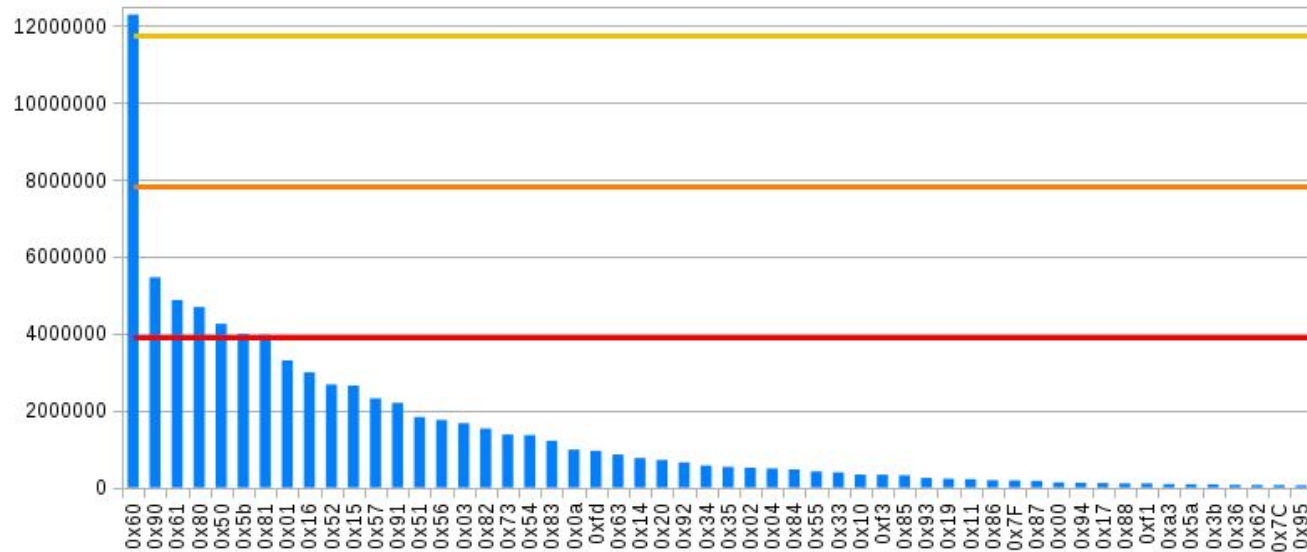


Analysis



Unused opcodes on verified contracts	
131	RETURNDATASIZE
132	RETURNDATACOPY
133	DELEGATECALL
134	INVALID
135	SELFDESTRUCT

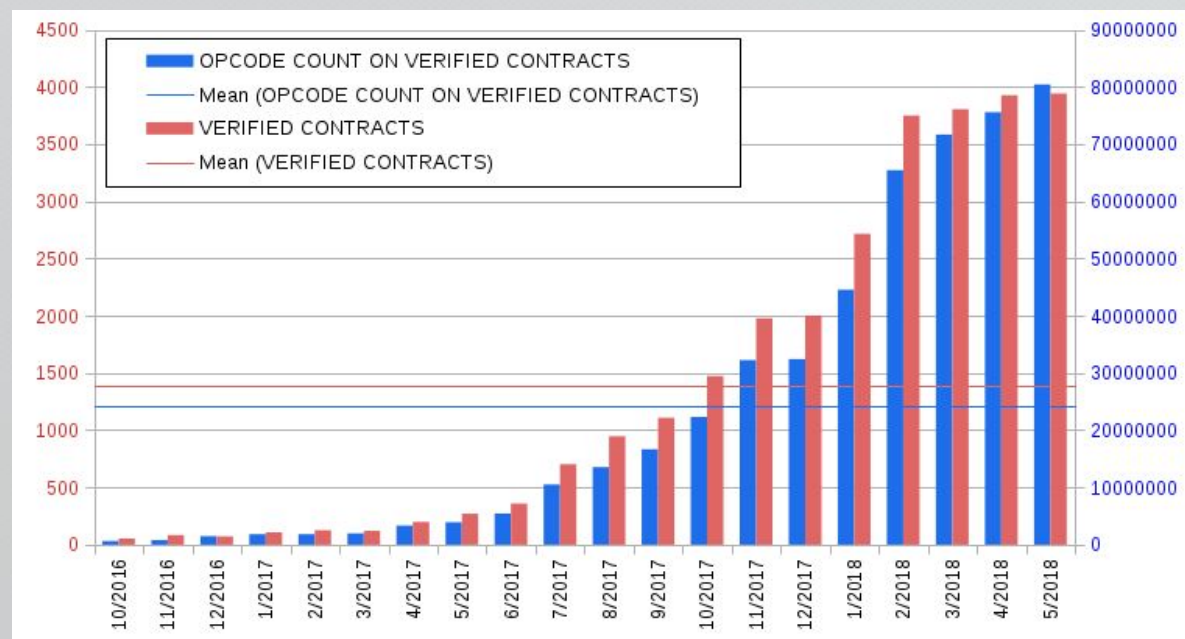
Analysis



Environmental Information opcodes on verified contracts	
27	CALLVALUE
28	CALLDATALOAD
33	CALLER
50	EXTCODESIZE
51	CALLDATASIZE
56	ADDRESS
59	CALLDATACOPY
68	CODECOPY
70	BALANCE
100	GASPRICE
104	CODESIZE
106	ORIGIN
130	EXTCODECOPY
131	RETURNDATASIZE
132	RETURNDATACOPY

Analysis

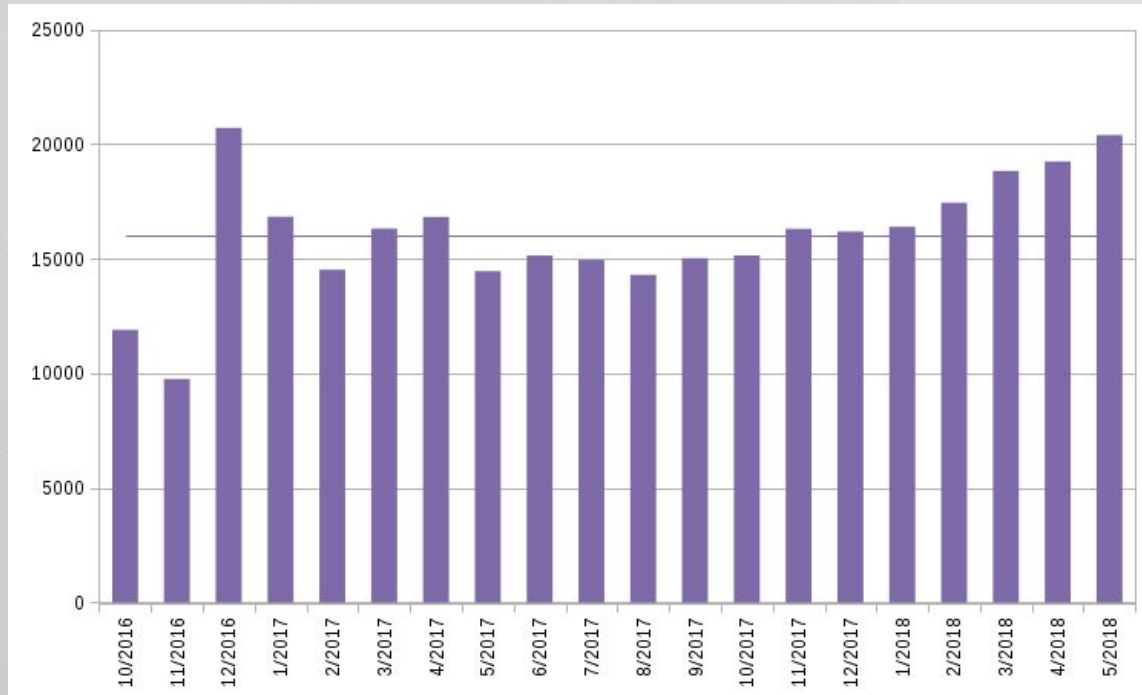
Opcode count per month and contract count per month



MONTH	VERIFIED CONTRACTS	OPCODE COUNT ON VERIFIED CONTRACTS
10/2016	53	630859
11/2016	83	809555
12/2016	72	1491497
1/2017	108	1818251
2/2017	126	1830664
3/2017	120	1958167
4/2017	198	3332301
5/2017	270	3903969
6/2017	359	5436532
7/2017	702	10495739
8/2017	947	13541032
9/2017	1108	16653251
10/2017	1473	22308628
11/2017	1977	32242058
12/2017	2002	32415653
1/2018	2716	44550116
2/2018	3749	65411651
3/2018	3804	71645646
4/2018	3926	75555729
5/2018	3941	80398664

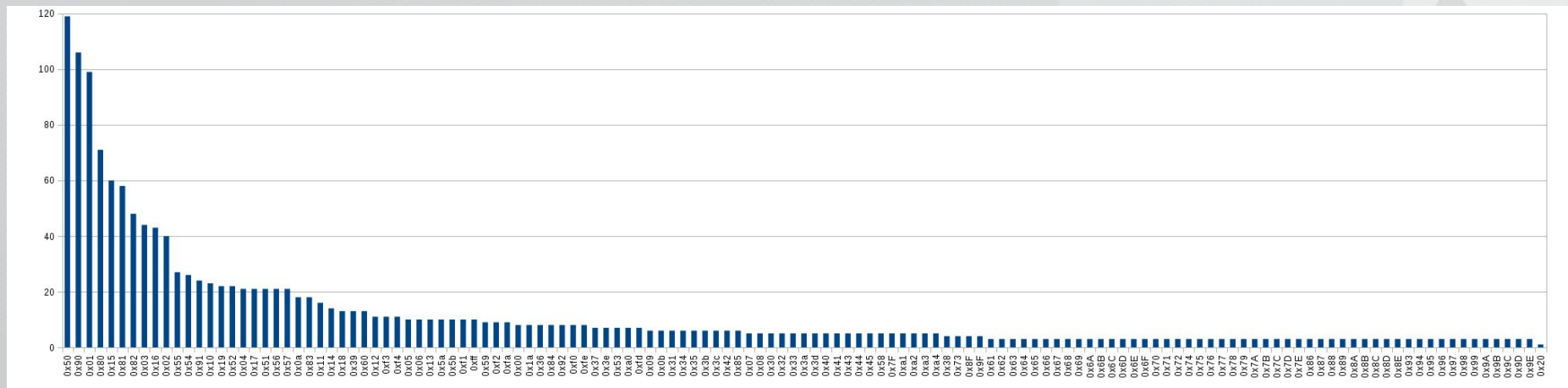
Analysis

Opcodes over contract count per month



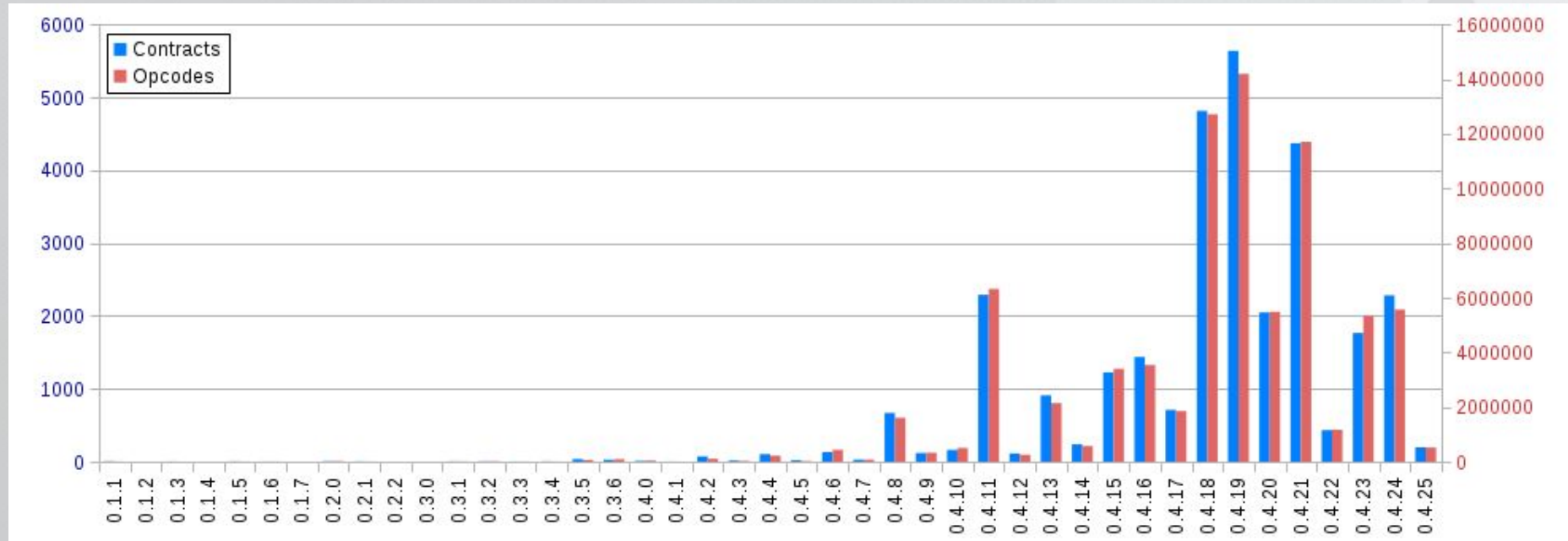
Analysis

Opcode occurrences on Solidity v0.4.19 source code



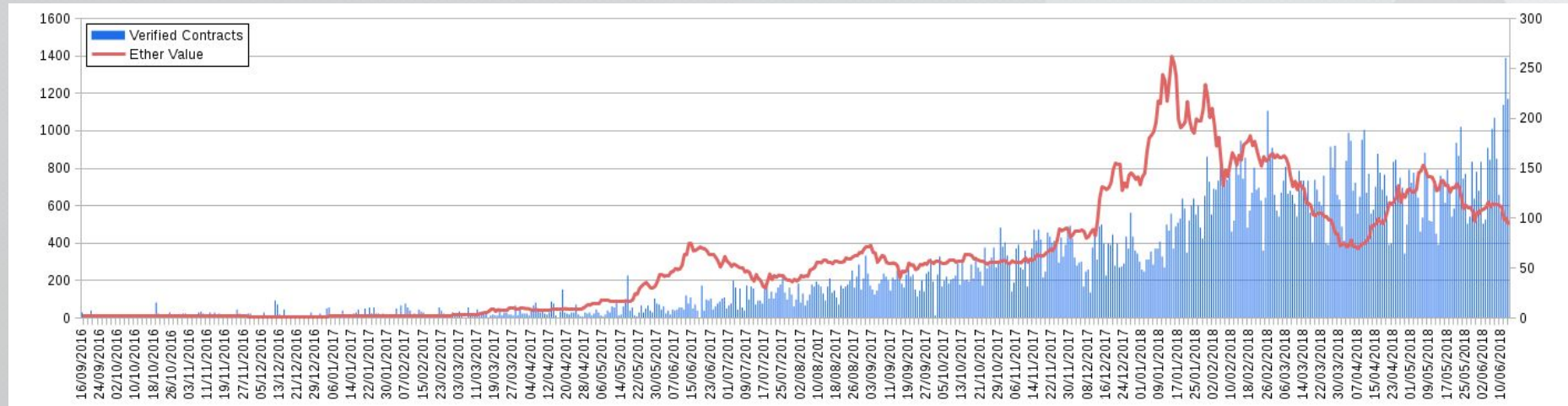
Analysis

Opcode count and contract deployment on different versions of Solidity



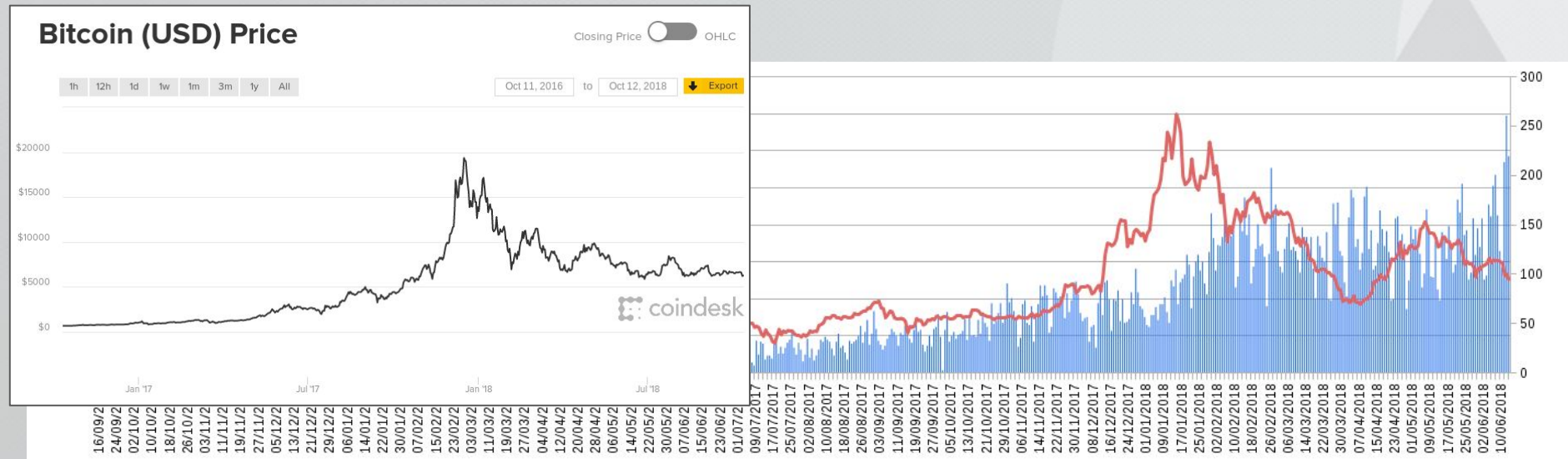
Analysis

Verified contracts per date and line chart of Ether value over time



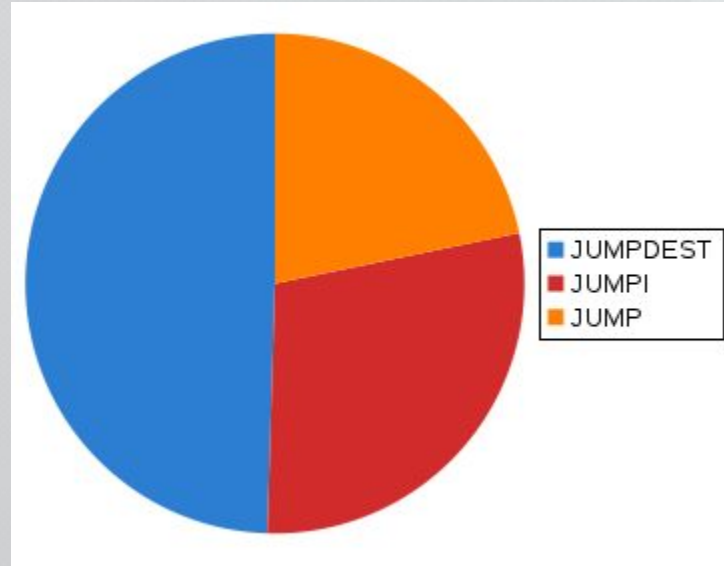
Analysis

Verified contracts per date and line chart of Ether value over time



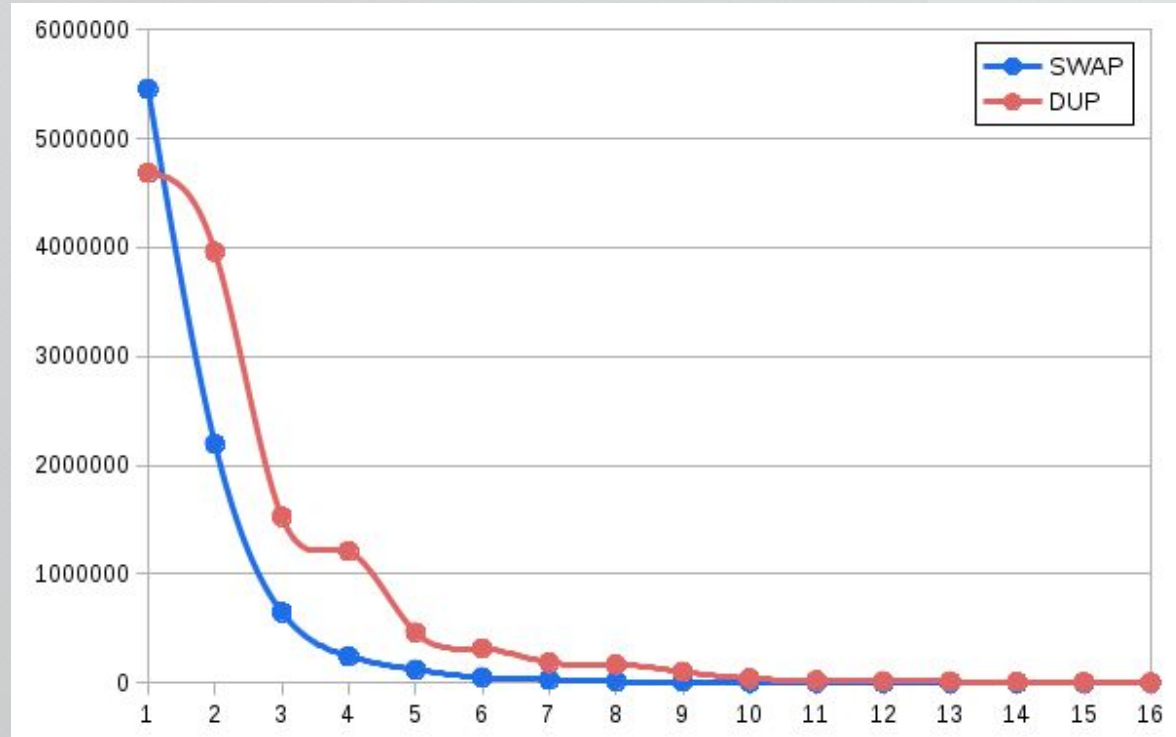
Analysis

JUMP, JUMPI and JUMPDEST opcodes occurrences percentage



Analysis

Comparison of SWAP and DUP occurrences



Conclusions and future works

By analysing the verified Ethereum smart contracts the last two years, we monitored opcodes usage.

We plan to:

- Investigate the correlation between opcodes usage and the corresponding Solidity code to identify relevant patterns.
- Extend our study to non-verified contracts.

Conclusions and future works

We plan to study and analyse the gas consumption of the contracts in order to

- support formal analyses on smart contracts
- define DSLs for specific application domains



**Thanks for
Your attention**

