PAP URLICATOR

PHYSTECH BUSINESS SOLUTIONS CASE CHAMPIONSHIP

ВЫБОР ОДИН.

СОЗДАНИЕ АЛГОРИТМА КЛАСТЕРИЗАЦИИ СМАРТ-КОНТРАКТОВ

ПРОБЛЕМА

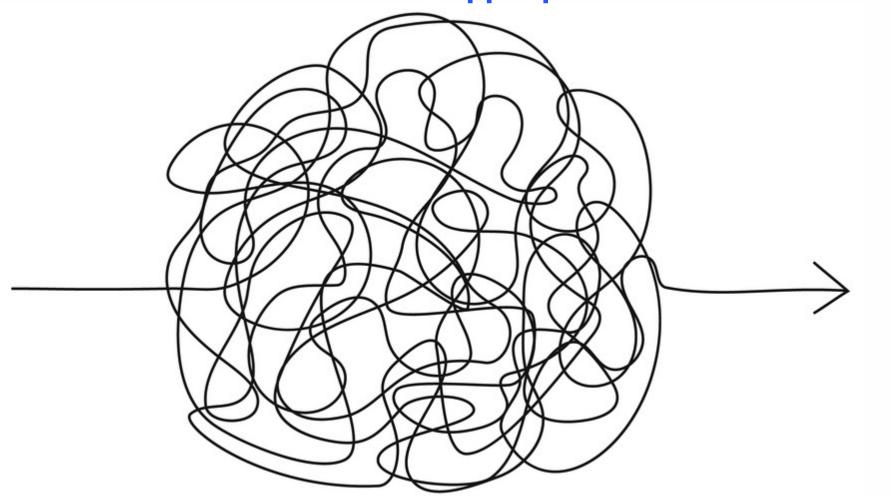
ОТСУТСТВИЕ ОБЪЕДИНЕНИЯ СМАРТ-КОНТРАКТОВ В ПРОЕКТЫ



СЛОЖНОСТЬ ПОНИМАНИЯ МЕХАНИЗМОВ ТРАНЗАКЦИЙ



АЛГОРИТМ КЛАСТЕРИЗАЦИИ ФОРМУЛЫ БУДУЩЕГО



СГРУППИРОВАННЫЕ СМАРТ-КОНТРАКТЫ

ВЫЗОВЫ ТРАНЗАКЦИИ

HALL AJICOPITM

- ОСНОВАН НА МЕТАДАННЫХ СМАРТ-КОНТРАКТОВ
- АНАЛИЗИРУЕТ ИХ ИСХОДНЫЙ КОД
- УЧИТЫВАЕТ МНОГООБРАЗИЕ РАЗРАБОТЧИКОВ БЛОКЧЕЙН-ИНДУСТРИИ
- ОТТАЛКИВАЕТСЯ ОТ СТАТИСТИЧЕСКИХ ЗАКОНОМЕРНОСТЕЙ

Почему сложно определить принадлежность смарт-контракта к проекту?

Потому что в сети разработчики не подписывают свой код. Или всё-таки подписывают?

ДЛЯ КАЖДОГО СМАРТ-КОНТРАКТА МЫ ПРЕДЛАГАЕМ ПРОВЕРЯТЬ

Наличие у него Public Name Tag на Etherscan

Токены

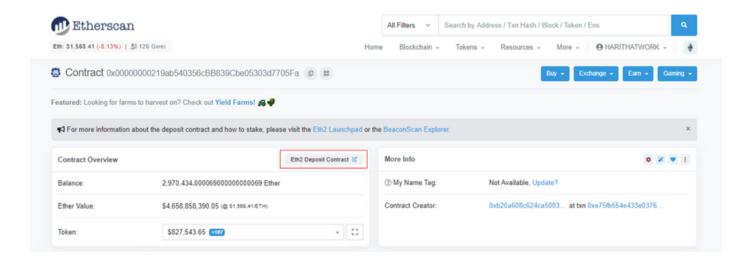
Имена методов

Закомментированные строки кода

I. Etherscan Public Name Tag

С помощью него Etherscan даёт возможность проектам помечать свои контракты

Public Name Tags indicates the owner of an address in the case of human-owned or Externally Owned Addresses (EOA) and the owner or purpose of an address in the case of contract addresses.



Name Tags can refer to an organization or an individual. In most cases, it comes with a URL link that points to the source of the Name Tag. Sometimes, we'll also add a public note on top for announcements or warnings related to the address.

Это гарантированно верное отождествление контракта с компанией, которая его создала

public_tags				
	url	tag	tag_link	
7f12f8b7fc532	https://etherscan.io/address/0xb4e16d0168e52d35cacd2c6185b44281ec28c9dc	Uniswap V2: USDC	https://v2.info.uniswap.org/pair/0xb4e16d0168e52d35cacd2c6185b44281ec28c9dc	
7f12f8b7fc532	https://etherscan.io/address/0xb1cd6e4153b2a390cf00a6556b0fc1458c4a5533	Bancor: ETHBNT Token	https://www.bancor.network/	
7f12f8b7fc532	https://etherscan.io/address/0x0d4a11d5eeaac28ec3f61d100daf4d40471f1852	Uniswap V2: USDT	https://v2.info.uniswap.org/pair/0x0d4a11d5eeaac28ec3f61d100daf4d40471f1852	
7f12f8b7fc532	https://etherscan.io/address/0x2af7ea6cb911035f3eb1ed895cb6692c39ecba97	InstaDApp: Event	https://instadapp.io/	
7f12f8b7fc532	https://etherscan.io/address/0x111111125434b319222cdbf8c261674adb56f3ae	1inch Network v2	https://app.1inch.io/#/r/0xb82F564B5E59041DaF46909dB12eA4858D1530A5	
8413ef929a5fbe	https://etherscan.io/address/0xaddc3e67a500f7037cd622b11df291a6351bfb64	•	0	
8413ef929a5fbe	https://etherscan.io/address/0xf7a232044c586d658f826e1932c37cad770b0687			
8413ef929a5fbe	https://etherscan.io/address/0x72012ada1d0a9e9cba22b3f13f3bc52b57ed6b66			
8413ef929a5fbe	https://etherscan.io/address/0xfd8a76dc204e461db5da4f38687adc9cc5ae4a86			
8413ef929a5fbe	https://etherscan.io/address/0x73d2ff81fcea9832fc9ee90521abde1150f6b52a			
8413ef929a5fbe	https://etherscan.io/address/0x612447e8d0bdb922059ce048bb5a7cef9e017812			
B413ef929a5fbe	https://etherscan.io/address/0xe7f4c89032a2488d327323548ab0459676269331			
8413ef929a5fbe	https://etherscan.io/address/0xd9e1ce17f2641f24ae83637ab66a2cca9c378b9f	SushiSwap: Router	https://app.sushi.com/swap	
8413ef929a5fbe	https://etherscan.io/address/0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2	Wrapped Ether	https://weth.io/	
8413ef929a5fbe	https://etherscan.io/address/0x2216d47494e516d8206b70fca8585820ed3c4946			

Мы спарсили страницу каждого контракта из рабочего датасета и уверенно идентифицировали 30% адресов

II. Токены

Bitquery API позволяет различать смарт-контракты токенов. Их имена часто содержат полное или краткое название проекта.

Однако мы должны отличать простые токены, участвующие во многих других проектах:

- 1) Они не вызываются через delegate call.
- 2) Мы знаем их имена:)

address	annotations	currencies
https://etherscan.io/address/0x64eda51d3ad40d56b9dfc5554e06f94e1dd786fd		Curve.fi tBTC/sbtcCrv
https://etherscan.io/address/0x6def55d2e18486b9ddfaa075bc4e4ee0b28c1545		Badger Sett Curve.fi renBTC/wBTC
https://etherscan.io/address/0x63cf44b2548e4493fd099222a1ec79f3344d9682		-
https://etherscan.io/address/0x2a8facc9d49fbc3ecff569847833c380a13418a8		-
https://etherscan.io/address/0x93054188d876f558f4a66b2ef1d97d16edf0895b		-
https://etherscan.io/address/0x444b860128b7bf8c0e864bdc3b7a36a940db7d88		-
https://etherscan.io/address/0x49849c98ae39fff122806c06791fa73784fb3675	Curve.fi renBTC/wBTC (crvRenWBTC), Curve.fi	Curve.fi renBTC/wBTC
https://etherscan.io/address/0x72012ada1d0a9e9cba22b3f13f3bc52b57ed6b66		-
https://etherscan.io/address/0x77655099f72484fa7d8cd701112c129b35b6ca6a		xWAIFUWETH
https://etherscan.io/address/0xb91bca4c6a607448a093803b3b2a9a4ed3e9f71e		xWAIFU
https://etherscan.io/address/0xbd4765210d4167ce2a5b87280d9e8ee316d5ec7c		-
https://etherscan.io/address/0x4e977830ba4bd783c0bb7f15d3e243f73ff57121		Aave stable debt bearing WETH
https://etherscan.io/address/0xd9e1ce17f2641f24ae83637ab66a2cca9c378b9f	Router, SushiSwap	-
https://etherscan.io/address/0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2	https://weth.io/, 0x Ecosystem, WrappedEther, Stableco	Wrapped Ether
https://etherscan.io/address/0x2216d47494e516d8206b70fca8585820ed3c4946	C	Waifus



III. Имена методов

Благодаря Bitquery API и GraphQL выгружаем список методов контракта

	-	Generic	withdrawToken, Contract Creation, depositEther
345dbdf3a34d4266cf5ebe9	Uniswap V2	DEX	approve, transferFrom, mint, balanceOf, transfer, burn, getReserves, token1, token0, swap, Cont
f23dc50a7bf38a8d6b233f6		Token	approve, transferFrom, symbol, balanceOf, transfer, burn, allowance, decimals, Contract Creation
	_	Generic	exitTokens, lockTokens, grantRole, Contract Creation, initialize, renounceRole

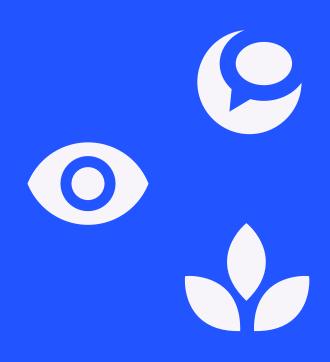
В приведённом примере контракт без метки имеет схожие с контрактом Uniswap названия функций. Действительно, ручной анализ показал, что они относятся к одному проекту (и конечно, кластеру в датасете P2P).

Таким образом, мы можем сопоставлять контракты по синтаксису, который использовался разработчиком. Делать это можно, например, с помощью библиотек SpaCy и NLTK.

IV. Закомментированные строки кода

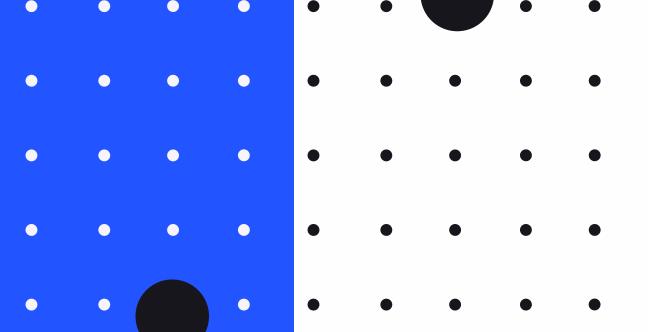
Недостающая информация о контракте может быть заполнена благодаря тому, что иногда разработчик упоминает имя проекта в комментариях. Как правило оно находится в одной строке со словами "author", "copyright", "developed" и тп

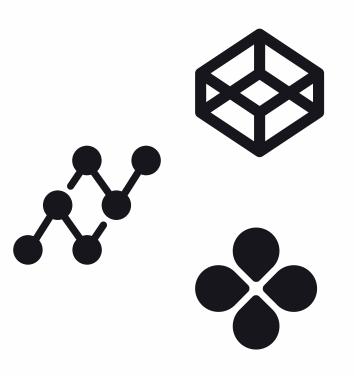




Сопоставлять их со списком блокчейн-проектов!

Данные собрали. А что с ними делать?





Мы сравниваем полученные нами данные на предмет их вхождения в список блокчейн-проектов.

В рамках решения кейса и в целях упрощения прототипирования мы взяли в качестве списков рейтинг ERC-20 и список разработчиков крипто-инструментов.

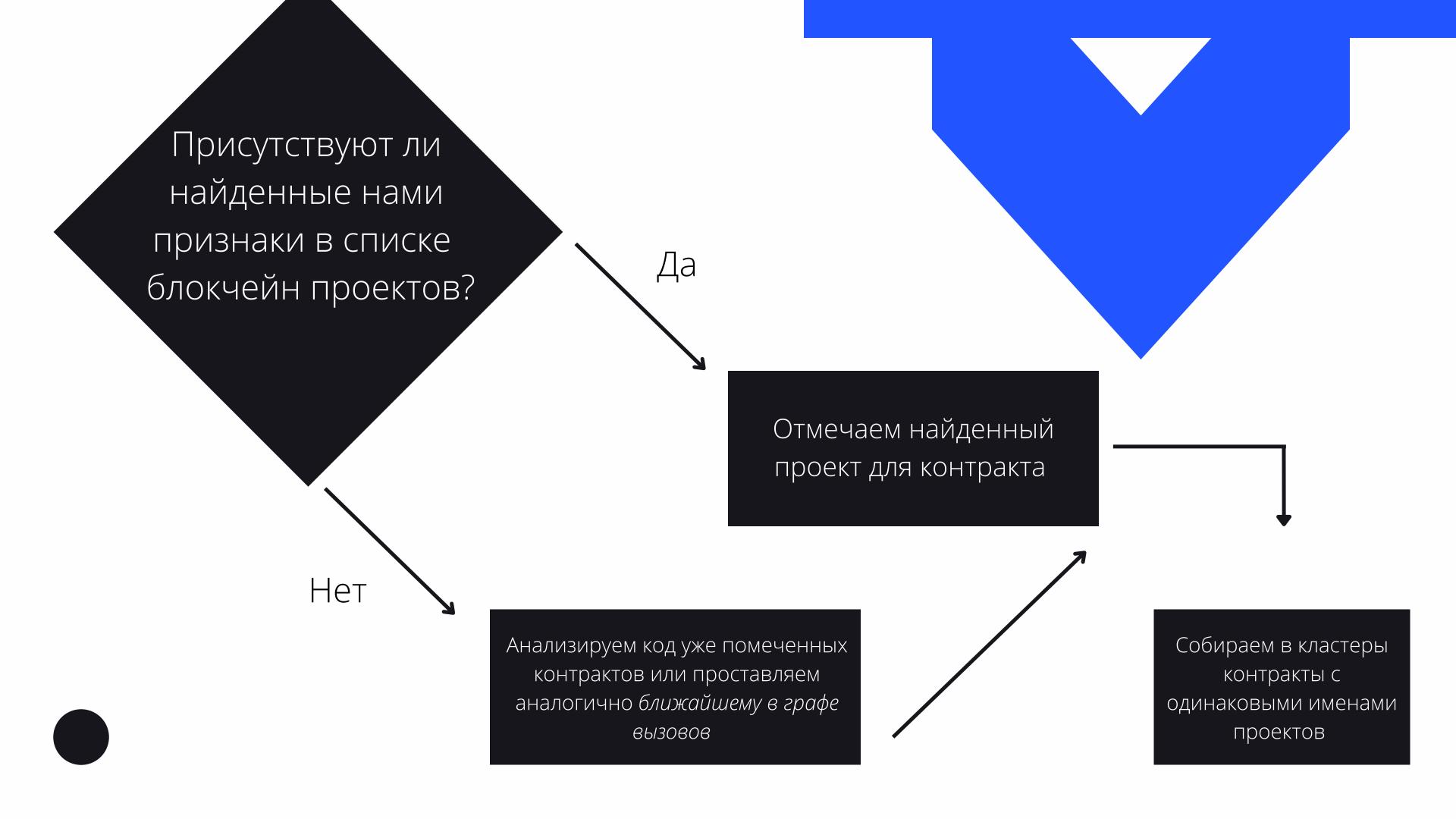
Так мы избавляемся от версионных символов и прочих побочных деталей и объединяем контракты в больший кластер.

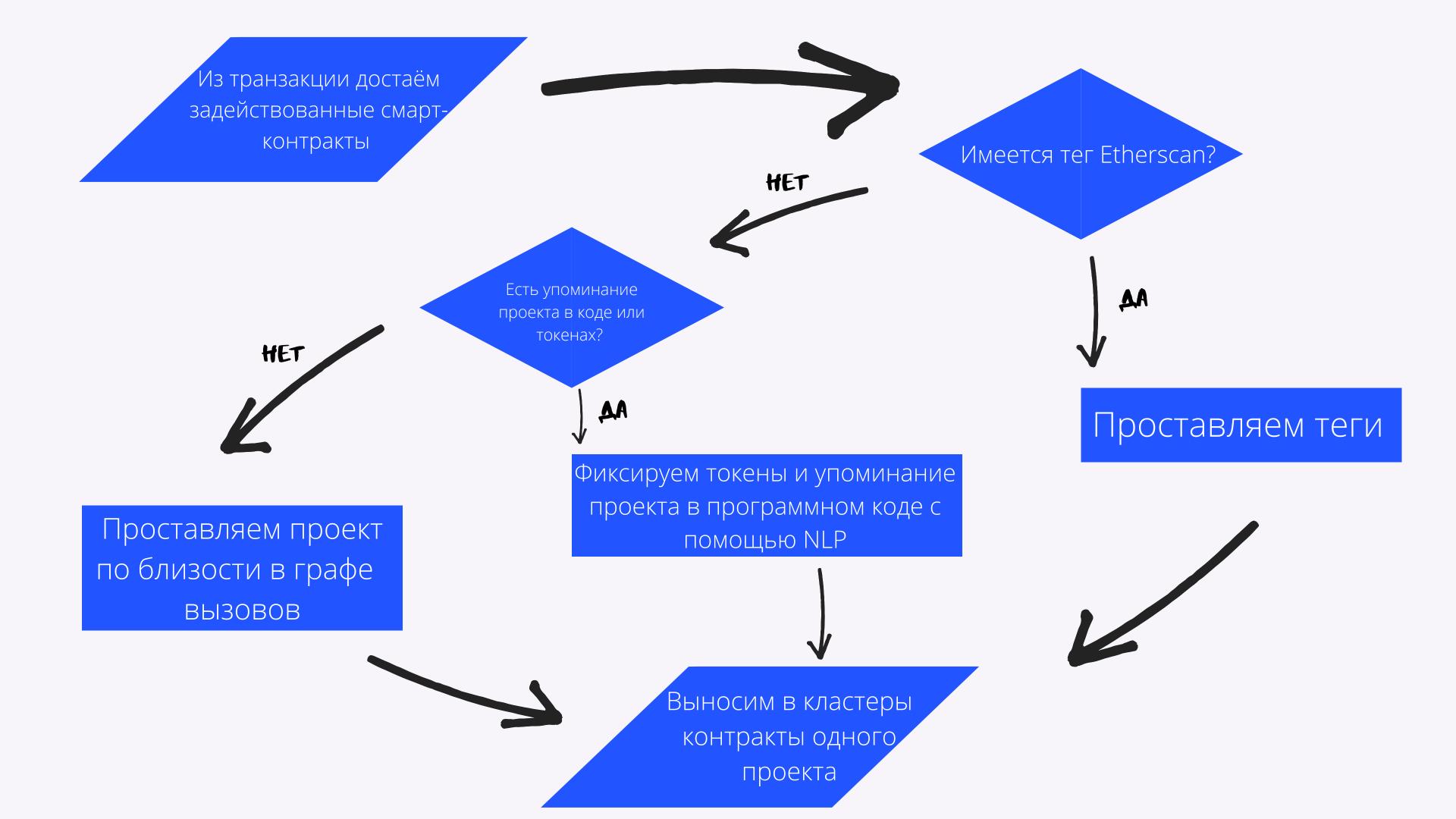
erc-20

erc-20			
erc20_name			
Tether USD			
BNB			
USD Coin			
Wrapped LUNA Token			
HEX			
Wrapped UST Token			
Binance USD			
SHIBA INU			
Wrapped BTC			
Matic Token			
stETH			
Wrapped liquid staked Ether 2	.0		
Cronos Coin			
Dai Stablecoin			
ChainLink Token			
TRON			
Bitfinex LEO Token			
Theta Token			
OKB			
Uniswap			

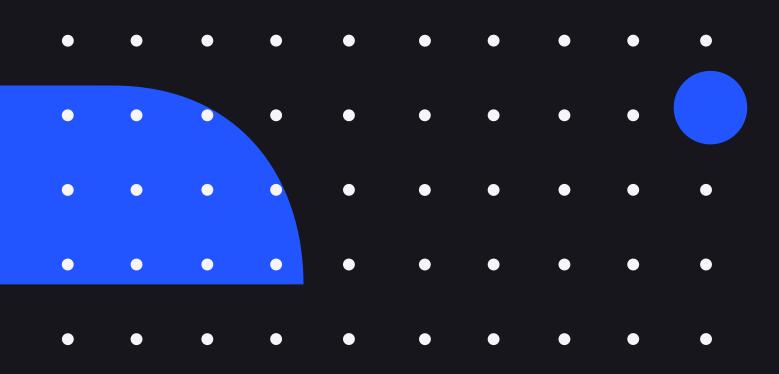
crypto_tools

tool_name		
Altrady		
AltSignals		
YouHodler		
Pionex		
Tokenplace		
Nexo		
MyEtherWallet (MEW)		
ECOS		
SHAMINING		
ACCOINTING.com		
Tickeron		
еТого		
Koinly		
HollaEx		
TradeStation		
CryptoTaxCalculator		
1inch		





Ещё одна вещь...



Смарт-контракты не обязательно кластеризовать по проектам

Статья <u>Assessing the Similarity of Smart</u>
<u>Contracts by Clustering their Interfaces</u>
натолкнула нас на множество идей



Статистический анализ

Оценка выполнения контрактов в каждой транзакции



Функционал смарт-контракта

Кластеризация по принципу действия, выделение стандартных интерфейсов



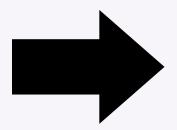
Граф вызовов между контрактами

Можно выделить адреса, играющие ключевую роль и строить кластеры вокруг них



Наш граф

Попарно считаем количество вызовов между контрактами



Благодаря **networkx** строим
взвешенный граф,
где вес рёбер число обращений
для каждой пары
контрактов

Контракты взаимодействуют больше других считаем их причастными к одному кластеру

МЕТОД КЛАСТЕРИЗАЦИИ НА ГРАФАХ*

0xd0fbb87e47da9987d845dbdf3a34d4266cf5ebe9

0xa0c68c638235ee32657e8f720a23cec1bfc77c77

0x3affcca64c2a6f4e3b6bd9c64cd2c969efd1ecbe

0x364c90218f6664f6c8b154ad9c3e31947cd3640c

0x53360799ab5f7d622f23dc50a7bf38a8d6b233f6

0x6abb753c1893194de4a83c6e8b4eadfc105fd5f5

0x1116898dda4015ed8ddefb84b6e8bc24528af2d8 0xa2327a938febf5fec13bacfb16ae10ecbc4cbdcf

.5c7adb3fd0df2d1822a36922dd941e16d2bf4e51

0x6def55d2e18486b9ddfaa075bc4e4ee0b28c1545

0xa0b86991c6218b36c1d19d4a2e9eb0ce3606eb48

0x046728da7cb8272284238bd3e47909823d63a58d

0xbafe01ff935c7305907c33bf824352ee5979b526

0xb3c9669a5706477a2b237d98edb9b57678926f04 0x939daad09fc4a9b8f8a9852a485dab2df4f4b3f8 0x2971adfa51b20e5a41bae5a708a8655a9c74f723

0xdac17f958d2ee523a2206206994597c13d831ec7

0x3d9819210a31b4961b30ef54be2aed79b9c9cd3b

0x41671ba1abcba387b9b2b752c205e22e916be6e3

0xfd8a76dc204e461db5da4f38687adc9cc5ae4a86

0x4c8a1beb8a87765788946d6b19c6c6355194abeb

0x4ddc2d193948926d02f9b1fe9e1daa0718270ed5

0x68b3465833fb72a70ecdf485e0e4c7bd8665fc45 0x0fc584529a2aefa997697fafacba5831fac0c22d 0x53edf7fc8bb9c249694ea0a2174043553b34db27

0x52f0feaa424fc209625f87f6c39ee1c1fa2440a9

0x5d3a536e4d6dbd6114cc1ead35777bab948e3643

0x73d2ff81fcea9832fc9ee90521abde1150f6b52a

0x0de05f6447ab4d22c8827449ee4ba2d5c288379b 0x220bda5c8994804ac96ebe4df184d25e5c2196d4

0xa035b9e130f2b1aedc733eefb1c67ba4c503491f

0xc02aaa39b223fe8d0a0 5c4f27ead9083c756cc2

0x6b175474e89094c44da98b954eedeac495271d0f

0x1f57cc62113c3a6346882dcf3ed49120411ac2d2

305178492165216213653530bff1848a515c8c5 0xe7f4c89032a2488d327323548ab09b50x607692862803:83627206fc5587b22c07c957718e3

0xddde1fa049209bc24b69d5fa316a56efec918d79

0xf63b34710400cad3e044cffdcab00a0f32e33pef50ba01

0xbe86f647b167567525ccaafcd6f881f1ee558216

0xc6845a5c768bf8d7681249f8927

0x6c3c78838c761c6ac7be9f59fe808ea2a6e43 0x030ba81f1c18d2806

0x63cH4b2548e4493fd099222a1ec79f3344d9682

0xa558ea1a875f8b576f0728d32c39f62158e49b92

0x612447e8d0bdb922059ce048bb5a7cef9e017812

0xe4922afab0bbadd8ab2a88e0c79d884ad337fca6

0x6354e79f21b56c11f48bcd7c451be456d7102a36

0x057835ad21a177dbdd3090bb1cae03eacf78fc6d

0x8a32f49ffba88aba6eff96f45d8bd1d4b3f35c7d

0x541dcd3f00bcd1a683cc73e1b2a8693b602201f4

0x3b2a77058a1eb4403a90b94585fab16bc512e703

0xd784927ff2f95ba542bfc624c8a8a98f3495f6b5

0x5b1b5fea1b99d83ad479df0c222f0492385381dd

0x4c9a2bd661d640da3634a4988a9bd2bc0f18e5a9

0x83d055d382f25e6793099713505c68a5c7535a35

0xe95a536cf5c7384ff1ef54819dc54e03d0ff1979

0xeb4c2781e4eba804ce9a9803c67d0893436bb27d

0xc011a73ee8576fb46f5e1c5751ca3b9fe0af2a6f

0x2f9ec37d6ccfff1cab21733bdadede11c823ccb0

0xe2d6ccac3ee3a21abf7bedbe2e107ffc0c037e80

*для простоты визуализации в Gephi отобразили от 3 и более

0x23d1b2755d6c243dfa9dd06624f1686b9c9e13eb

вызовов

Исходный код и полученные данные доступны на GitHub

https://github.com/dizpatcher/defianalytics

Результаты были достигнуты благодаря API Bitquery, Etherscan, языку GraphQL, библиотекам networkx, requests и BeautifulSoup (Python), а также Gephi.



К описанному алгоритму

Подключаем ИИ

Учимся правильно

Интерпретировать собранные данные

Проверяем на корректность

Дополнительные возможности

Выродов Артём

t.me/vyrodov

Казьмин Олег

t.me/oleg_ator02

Маркович Владимир

t.me/vm5_ball

